CHAPTER 1

INTRODUCTION

1.1 Background of the Study

With the advancement of wireless technologies, wireless sensor networks can greatly expand our ability to monitor and track conditions of patients in the healthcare area. High performance and fault tolerant wireless devices can now be employed to eliminate medical errors, reduce workload and increase the efficiency of hospital staff, and improve the comfort of patients. Thus, there has been increased interest among research groups in developing wireless recording and monitoring for real-time physiological parameters (e.g. Electrocardiogram (ECG), Electroencephalography (EEG), Electrooculography (EOG), Electromyography (EMG) Neural, Blood Flow, Blood Pressure, Pulse Rate etc.) from a patient's body in medical environments . Existing wireless data collection systems use standards such as ZigBee (IEEE 802.15.4) or Bluetooth (IEEE 802.15.1). A Wireless Body Area Network (WBAN) based on a low cost wireless sensor network technology could greatly benefit patient monitoring systems in hospitals, residential and work environments (Poon & Zhang, 2006).

A WBAN system allows easy internetworking with other devices and networks, thus offering health care worker easy access to patient's critical and non-critical data. One of the main advantages of a WBAN is to monitor patients remotely using an intranet or the Internet. A WBAN could be seen as a special purpose wireless sensor network with a number of additional system design requirements. A WBAN is mostly likely to incorporate wearable and implantable node operating in two different frequencies. An implantable node is most likely to operate at 400 MHz using the Medical Implant Communications Service (MICS) medical band whereas the wearable node could operate in some other band.

Many patients can benefit from continuous monitoring as a part of a diagnostic procedure, optimal maintenance of a chronic condition or during supervised recovery from an acute disease or surgical procedure (Park & Jayaraman, 2003).

A Personal Area Network (PAN) or Body Area Network (BAN) can be achieved by integrating a vital sign monitoring sensor into a user's clothing. This monitoring system is suitable for patients suffering from very chronic diseases such as stroke, diabetes, hypertension, etc because many of them will be practically immobile and therefore stationed in one place due to the seriousness of their ailment and in such a situation, there is need for a continuous unobtrusive monitoring of these vital signs. However, Body Area Network is unsuitable for lengthy, continuous monitoring, particularly during normal activity, intensive training or computer-assisted rehabilitation. Recent technology advances in wireless networking, micro-fabrication, and integration of physical sensors, embedded microcontrollers and radio interfaces on a single chip, promise a new generation of wireless sensors suitable for many applications. However, the existing telemetric devices either use wireless communication channels exclusively to transfer digitized data from sensors to the monitoring station, or use standard high-level wireless protocols such as Bluetooth or ZigBee to transfer the data to the monitoring devices. Simple, accurate means of monitoring daily activities outside of the laboratory are not available at the present, only estimates can be obtained from questionnaires, measures of heart rate, video assessment, and use of pedometers or accelerometers. Finally, records from individual monitoring sessions can be integrated into research databases that would provide support for data mining and knowledge discovery relevant to specific conditions and patient categories (Istepanian et al., 2004).

Increased system processing power allows sophisticated real-time data processing from the data acquired from the sensors. The results obtained from the systems can support biofeedback and generation of warnings. The use of biofeedback techniques has gained increased attention among researchers in the field of physical medicine and tele-rehabilitation. Intensive practice schedules have been shown to be important for recovery of motor function. Unfortunately, an aggressive approach to rehabilitation involving extensive therapist-supervised motor training is not a realistic expectation in today's health care system where individuals are typically seen as outpatients about twice a week for no longer than 30–45 min. Results from the Wireless Body Area Network (WBAN) technology and biofeedback systems appear to be a valid alternative, as they reduce the extensive time to set-up a patient before each session and require limited time involvement of physicians and therapists. Furthermore, WBAN technology could potentially address a second factor that hinders enthusiasm for rehabilitation, namely the fact that setting up a patient for the procedure is rather time-consuming. This is because tethered sensors need to be positioned on the subject, attached to the equipment, and a software application needs to be started before each session. WBAN technology allows sensors that will be positioned on the subject for prolonged periods, therefore eliminating the need to position them for every training session. Instead, a personal server such as a Personal Digital Assistant (PDA) can almost instantly initiate a new training session whenever the subject is ready and willing to exercise. In addition to home rehabilitation, this setting also may be beneficial in the clinical setting, where precious time of physicians and therapists could be saved. Moreover, the system can issue timely warnings or alarms to the patient, or to a specialized medical response service in the event of significant deviations of the norm or medical emergencies. However, as for all systems, regular, routine maintenance (verifying configuration and thresholds) by a specialist is required (Martin et al., 2000).

Typical examples of possible applications include stroke rehabilitation, physical rehabilitation after hip or knee surgeries, myocardial infarction rehabilitation, and traumatic brain injury rehabilitation. The assessment of the effectiveness of rehabilitation procedures has been limited to the laboratory setting; relatively little is known about rehabilitation in real-life situations. Miniature, wireless, Body Area Network (BAN) technology offers a tremendous opportunity to address this issue.

Several models of multiparametric monitors are networkable, i.e., they can send their output to a Central Intensive care Unit (CICU) monitoring station, where a single staff member can observe and respond to several bedside monitors simultaneously. Ambulatory telemetry can also be achieved by portable, battery-operated models which are carried by the patient and which transmit their data via a wireless data connection.

Persons requiring critical care; patients who have undergone surgery or persons with chronic ailments require continuous health monitoring and real-time feedback for immediate action in emergency situations. The patients would be subjected to discomfort and inconvenience due to prolonged hospitalization. Frequent visits to hospitals may also be required for follow up treatment and care. Use of Body Sensor Network can provide an alternative solution for remote monitoring of patients residing in the comfort of the homes. Patients can move about and follow their daily routine without the necessity of being confined to their beds. Data obtained over a long period of time in the patient's natural environment would offer the doctors a better insight into the patient's health condition and such data can be analyzed to arrive at the correct diagnosis and provide the right care (Park & Jayaraman, 2003).

Communication of health related information between sensors in BSN and the remote medical server has to be strictly private and confidential to protect patient privacy The sensor data sent using Internet and wireless transmission is prone to different types of attack such as eavesdropping, sending false values or replay of previous data. Medical professionals have to be certain that the data are not tampered in transit or at a point of origin as proper diagnosis requires accurate data.

The Internet of Things (IoT) which is the network of physical devices, vehicles, buildings and other items—embedded with electronics, software, sensors, actuators, and network connectivity enable these objects to collect and exchange data. In 2013 the Global Standards Initiative on Internet of Things (IoT-GSI) defined the IoT as "the infrastructure of the information society." The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit; when IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure.

In the internet of things (IoT), devices gather and share information directly with each other and the cloud, making it possible to collect, record and analyze new data streams faster and more accurately. That suggests all sorts of interesting possibilities across a range of industries: cars that sense wear and tear and self-schedule maintenance or trains that dynamically calculate and report projected arrival times to waiting passengers.

The internet of things (IoT) plays a significant role in healthcare applications where it can be used to manage chronic diseases and store patients' medical records. Internet of things in remote medical monitoring offers great promise to patients because it can be used to monitor, track and store patients' medical records in the cloud for easy access.

Remote medical monitoring of patients requires monitoring the physiological state of patients with acute or chronic conditions or chronic disease states which predominantly derive decided prognosis advantages from intensive condition tracking. More particularly, the invention is directed to a condition monitoring system which includes one or more remote modular testing units and a central station. The remote units include physiological parameter testing modules to acquire data from one or possibly many patients and communicate with a central station typically capable of interfacing with a large number of patient-operated units or clinician-operated units testing many patients. The central station, in turn, may interface and communicate with any number of other devices as by networking. Parameters checked may include but are not limited to blood pressure, pulse rate, blood oxygen saturation, weight, blood glucose, temperature, prothrombin (clotting) time and pulmonary function, including respiratory rate and depth. Other functions, such as ECG (electrocardiograph) traces and infant breathing monitoring for detection of SIDS (sudden infant death syndrome) onset are also contemplated (Park & Jayaraman, 2003)

A remote monitoring system operates with an instruction set to provide automated administration of health care to a patient. In a preferred embodiment of the invention a central monitoring station receives data from a plurality of patients connected with vital sign monitoring sensors which enter the queuing model from where the results are analysed and kept in the server. The doctor can view the server from anywhere to see the results of the patients' vital signs. A particular patient's graph can also be viewed by the doctor to see the progression of the patient's vital signs readings. Medical procedures are then administered to the patient and results taken as data. The data is made available to the central monitor so that proper medical interpretation is enabled. A number of novel steps in the programming of the system are taken to assure that the right patient is being monitored, that the patient is being tested properly and that the system is being monitored appropriately.

Body network sensors have become wearable computing results from placing computers and sensors on the body to create a body area network (BAN) that can sense, process, and report on some set of the wearer's attributes. Proactive computing and wearable computing working in tandem let computers fade into the woodwork, enriching quality of life and engendering independence.

1.2 Statement of the Problem

In a typical Nigerian hospital, the vital signs of patients are only checked by nurses on duty. The nurses check these vital signs two times a day and because the vital signs fluctuate throughout the day, a patient who was checked a while ago may have the vital signs deteriorate terribly after some time. The nurse having recorded the readings for that particular patient will not get to know except when the vital signs are checked again in the evening. This results in the avoidable death of the patients most of the time. Therefore, there is the need to have these patients (especially those suffering from chronic diseases) monitored remotely and continuosly so that at every point in time, the readings of the different vital signs are taken and sent to a central location and anytime the reading becomes abnormal, the central location notifies the medical personnel immediately so that they can swing into action. This project is designed to have a continuous unobtrusive monitoring of the vital signs, have a real time medical interpretation and prescription and this will drastically reduce the mortality rate in Nigerian hospitals.

1.3 Objectives of the Study

The objective of this thesis is to develop a remote medical monitoring system that when fully deployed, will be capable of:

i. Simulating and monitoring the vital signs (such as blood pressure, and pulse rate) of patients with very chronic diseases like diabetes, hypertension, stroke, etc, based on known causes of abnormal blood pressure and pulse rate and send the results to the central location.

ii. Utilizing a Wireless Application Protocol (WAP) which can run on a Personal Digital Assistant (PDA), or smart phone based on the simulated data.

iii. Utilizing supplementary software in the server which collects these vital signs from a queuing model, analyse them, make predictions and recommendations and the doctor can view the patients' records from anywhere and at anytime.

iv. Storing such records as archived documents for future reference by medical experts. These stored records can also be used in data mining.

v. Producing report on medical failures and events which can be used for decision making by medical experts in their absence from the hospital.

vi. Ensuring that the data being transferred are adequately secured as they move between the patient and the medical personnel.

1.4 Significance of the Study

Many sick people in Nigeria today are in the remote areas where adequate health facility is not readily available. Some of these sick people do not necessarily need to be in the hospital environment in order to get treated.

Most of the time, some of these patients die due to lack of adequate care when these deaths can be avoidable. Part of the reason people die in hospitals is latency between ward visits, absence of medical personnel in critical situations for people who are sick of chronic diseases and no health care facilities for people in the remote areas. These avoidable deaths need to be stopped by deploying a technology that is dynamic and responsive to stimuli in the hospital rooms.

If this is not done, more Nigerians will die when such deaths are expected to be avoidable.

Loss of Nigerian citizens through causes that can be prevented is detestable. Any effort made to avert these avoidable deaths is to be lauded. This project if implemented will assist in saving many lives that would have been lost and will also extend health care services to people in the rural areas. It is for this reason that this dissertation is very significant.

1.5 Scope of the Study

i. This thesis is limited to the blood pressure and pulse rate monitoring which are two of the major vital signs of humans. There is also a particular interest in patients suffering from chronic diseases like diabetes, hypertension and stroke.

ii. This thesis produced a prototype Remote Medical Monitoring system using Central Hospital Warri as a case study. It is expected that this prototype when deployed in any other hospital with little or no amendment whether the hospital is a private or public one will be able to achieve similar results.

1.6 Limitations of the Study

The limitations of this thesis are as follows:

i. It was difficult getting a blood pressure and pulse rate monitoring device which can monitor the blood pressure and pulse rate and which can send the result to a mobile phone using Bluetooth. To overcome this limitation, the readings of the blood pressure and the pulse rate were simulated using a computer program.
ii. During the feasibility study, the nurses at the Central Hospital were not very supportive in divulging information about how the system work initially but after much persuasion, the nurses became friendly and volunteered reasonable information.

Remote Medical Monitoring is a new area in Nigeria, so it was not easy getting materials especially during the literature review.

1.7 Definition of Terms

i. Blood pressure

Blood pressure (BP) is a force exerted by circulating blood on the walls of blood vessels, and is one of the principal vital signs. During each heartbeat, BP varies between a maximum (systolic) and a minimum (diastolic) pressure (Klabunde, 2005). The mean BP, due to pumping by the heart and resistance in blood vessels, decreases as the circulating blood moves away from the heart through arteries. It has its greatest decrease in the small arteries and arterioles, and continues to decrease as the blood moves through the capillaries and back to the heart through veins. Gravity, valves in veins, and pumping from contraction of skeletal muscles, are some other influences on BP at various places in the body.

The term *blood pressure* usually refers to the pressure measured at a person's upper arm. It is measured on the inside of an elbow at the brachial artery, which is the upper arm's major blood vessel that carries blood away from the heart. A person's BP is usually expressed in terms of the systolic pressure and diastolic pressure, for example 120/80, where 120 is the systolic pressure and 80 is the diastolic pressure.

ii. Bluetooth

Bluetooth is a proprietary open wireless protocol for exchanging data over short distances (using short length radio waves) from fixed and mobile devices, creating personal area networks (PANs). It was originally conceived as a wireless alternative to RS-232 data cables (Stallings, 2005). It can connect several devices, overcoming problems of synchronization.

iii. Body Area Network

WBAN or BAN, short for (Wireless) Body Area Network, consists of a set of mobile and compact intercommunicating sensors, either wearable or implanted into the human body, which monitor vital body parameters and movements.

These devices, communicating through wireless technologies, transmit data from the body to a home base station, from where the data can be forwarded to a hospital, clinic or elsewhere, real-time (Charles & Jeffrey, 2004).

iv. Data Hub

In data communications, a hub is a place of convergence where data arrives from one or more directions and is forwarded out in one or more other directions. A hub usually includes a switch of some kind. (And a product that is called a "switch" could usually be considered a hub as well.) The distinction seems to be that the hub is the place where data comes together and the switch is what determines how and where data is forwarded from the place where data comes together.

v. Electrocardiography

Electrocardiography (ECG or EKG from Greek: *kardia*, meaning heart) is a transthoracic (across the thorax or chest) interpretation of the electrical activity of the heart over a period of time, as detected by electrodes attached to the surface of the skin and recorded by a device external to the body. The recording produced by this noninvasive procedure is termed an electrocardiogram (ECG or EKG).

An ECG is used to measure the rate and regularity of heartbeats, as well as the size and position of the chambers, the presence of any damage to the heart, and the effects of drugs or devices used to regulate the heart, such as a pacemaker (Van Mieghem et al., 2004).

Most ECGs are performed for diagnostic or research purposes on human hearts, but may also be performed on animals, usually for diagnosis of heart abnormalities or research.

vi. Electroencephalography (EEG)

Electroencephalography (EEG) is the recording of electrical activity along the scalp. EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain. In clinical contexts, EEG refers to the recording of the brain's spontaneous electrical activity over a short period of time, usually 20–40 minutes, as recorded from multiple electrodes placed on the scalp. Diagnostic applications generally focus on the spectral content of EEG, that is, the type of neural oscillations that can be observed in EEG signals. In neurology, the main diagnostic application of EEG is in the case of epilepsy, as epileptic activity can create clear abnormalities on a standard EEG study. A secondary clinical use of EEG is in the diagnosis of coma, encephalopathies, and brain death. A third clinical use of EEG is for studies of sleep and sleep disorders where recordings are typically done for one full night, sometimes more. EEG used to be a first-line method for the diagnosis of tumors, stroke and other focal brain disorders, but this use has decreased with the advent of anatomical imaging techniques with high (<1 mm) spatial resolution such as MRI and CT. Despite limited spatial resolution, EEG continues to be a valuable tool for research and diagnosis, especially when millisecond-range temporal resolution (not possible with CT or MRI) is required (Niedermeyer & Da Silva, 2004).

vii. Electromyography (EMG)

Electromyography (EMG) is a technique for evaluating and recording the electrical activity produced by skeletal muscles. EMG is performed using an instrument called an electromyograph, to produce a record called an electromyogram. An electromyograph detects the electrical potential generated by muscle cells when these cells are electrically or neurologically activated. The signals can be analyzed to detect medical abnormalities, activation level, recruitment order or to analyze the biomechanics of human or animal movement (Kamen, 2004).

viii. Electrooculography (EOG/E.O.G.)

Electrooculography (EOG/E.O.G.) is a technique for measuring the resting potential of the retina. The resulting signal is called the electrooculogram. The main applications are in ophthalmological diagnosis and in recording eye movements. Unlike the electroretinogram, the EOG does not represent the response to individual visual stimuli (Brown et al., 2006).

Eye movement measurements: Usually, pairs of electrodes are placed either above and below the eye or to the left and right of the eye. If the eye is moved from the center position towards one electrode, this electrode "sees" the positive side of the retina and the opposite electrode "sees" the negative side of the retina. Consequently, a potential difference occurs between the electrodes.

Assuming that the resting potential is constant, the recorded potential is a measure for the eye position.

ix. General Packet Radio Service (GPRS)

General packet radio service (GPRS) is a packet oriented mobile data service on the second generation (2G) and third generation (3G) cellular communication systems global system for mobile communications (GSM). The service is available to users in over 200 countries worldwide. GPRS was originally standardized by European Telecommunications Standards Institute (ETSI) in response to the earlier CDPD and i-mode packet switched cellular technologies. It is now maintained by the 3rd Generation Partnership Project (3GPP).

x. Global Positioning System (GPS)

The Global Positioning System (GPS) is a space-based global navigation satellite system that provides reliable location and time information in all weather and at all times and anywhere on or near the Earth when and where there is an unobstructed line of sight to four or more GPS satellites. It is maintained by the United States government and is freely accessible by anyone with a GPS receiver (Hoffmann-Wellenhof et al., 1994).

GPS consists of three parts: the space segment, the control segment, and the user segment. The U.S. Air Force develops, maintains, and operates the space and control segments. GPS satellites broadcast signals from space, which each GPS receiver uses to calculate its three-dimensional location (latitude, longitude, and altitude) plus the current time (Kaplan, 1996).

xi. Pager

A pager is a small telecommunications device that receives (and, in some cases, transmits) alert signals and/or short messages. This type of device is convenient for people expecting telephone calls, but who are not near a telephone set to make or return calls immediately.

A typical one-way pager fits easily in a shirt pocket; some are as small as a wristwatch. A miniature, short-range wireless receiver captures a message, usually accompanied by a beep. (This is why the device is also known as a beeper). The simplest one-way pagers display the return-call telephone number of the person who sent the message. Alternatively, a code can be displayed that indicates which of several designated parties is requesting a return phone call. Sophisticated one-way pagers can display short text messages.

xii. Personal Area Network

A personal area network (PAN) is a computer network used for communication among computer devices, including telephones and personal digital assistants, in proximity to an individual's body. The devices may or may not belong to the person in question. The reach of a PAN is typically a few meters. PANs can be used for communication among the personal devices themselves (intrapersonal communication), or for connecting to a higher level network and the Internet (an uplink).

Personal area networks may be wired with computer buses such as USB and FireWire. A wireless personal area network (WPAN) can also be made possible with network technologies such as Infrared Data Association (IrDA), Bluetooth, Ultra-wide Band (UWB), Z-Wave and ZigBee (Charles & Jeffrey, 2004).

xiii. Personal Digital Assistant

A personal digital assistant (PDA), also known as a palmtop computer, is a mobile device which functions as a Personal information manager and connects to the internet. The PDA has an electronic visual display enabling it to include a web browser, but some newer models also have audio capabilities, enabling them to be used as mobile phones or portable media players (Viken, 2009). Many PDAs can access the internet, intranets or extranets via Wi-Fi, or Wireless Wide Area Networks (WWANs). Many PDAs employ touch screen technology.

xiv. **Pulse**

The pulse is the physical expansion of the artery. Its rate is usually measured either at the wrist or the ankle and is recorded as beats per minute. The pulse commonly taken is the radial artery at the wrist. Sometimes the pulse cannot be taken at the wrist and is taken at the opposite of the elbow (brachial artery), at the neck against the carotid artery (carotid pulse), behind the knee (popliteal artery), or in the foot dorsalis pedis or posterior tibial arteries. The pulse rate can also be measured by listening directly to the heartbeat using a stethoscope. The pulse varies with age (Lauralee, 2006). A newborn or infant can have a heart rate of about 130-150 beats per minute. A toddler's heart will beat about 100-120 times per minute, an older child's heartbeat is around 90-110 beats per minute, adolescents around 80-100 beats per minute, and adults pulse rate is anywhere between 50 and 80 beats per minute.

xv. Pulse oximetry

Pulse oximetry (or ~ oxymetry in the UK) is a non-invasive method allowing the monitoring of the oxygenation of a patient's haemoglobin (Mower et al., 1997).

A sensor is placed on a thin part of the patient's anatomy, usually a fingertip or earlobe, or in the case of a neonate, across a foot, and a light containing both red and infrared wavelengths is passed from one side to the other. Changing absorbance of each of the two wavelengths is measured, allowing determination of the absorbances due to the pulsing arterial blood alone, excluding venous blood, skin, bone, muscle, fat, and (in most cases) fingernail polish. Based upon the ratio of changing absorbance of the red and infrared light caused by the difference in color between oxygen-bound (bright red) and oxygen unbound (dark red or blue, in severe cases) blood haemoglobin, a measure of oxygenation (the per cent of haemoglobin molecules bound with oxygen molecules) can be made (Mower et al., 1998).

xvi. Radio

Radio is the transmission of signals by modulation of electromagnetic waves with frequencies below those of visible light. Electromagnetic radiation travels by means of oscillating electromagnetic fields that pass through the air and the vacuum of space. Information is carried by systematically changing (modulating) some property of the radiated waves, such as amplitude, frequency, phase, or pulse width. When radio waves pass an electrical conductor, the oscillating fields induce an alternating current in the conductor (Clint & Gervelis, 2003). This can be detected and transformed into sound or other signals that carry information.

xvii. Respiratory rate

Varies with age, but the normal reference range for an adult is 12-20 breaths/minute The value of respiratory rate as an indicator of potential respiratory dysfunction has been investigated but findings suggest it is of limited value (Tortora & Anagnostakos, 1990).

xviii. Satellite link

A satellite modem is not the only device needed to establish a communication channel. Other equipment that are essential for creating a satellite link include satellite antennas and frequency converters.

Data to be transmitted are transferred to a modem from Data terminal equipment (e.g. a computer). The modem usually has Intermediate frequency

(IF) output (that is, 50-200 MHz), however, sometimes the signal is modulated directly to L-band. In most cases frequency has to be converted using an up converter before amplification and transmission.

xix. Satellite modem

A satellite modem or sat modem is a modem used to establish data transfers using a communications satellite as a relay.

There is a wide range of satellite modems from cheap devices for home internet access to expensive multifunctional equipment for enterprise use.

A "modem" stands for "modulator-demodulator". A satellite modem's main function is to transform an input bitstream to a radio signal and vice versa. There are some devices that include only a demodulator (and no modulator, thus only allowing data to be downloaded by satellite) that are also referred to as "satellite modems". These devices are used in satellite Internet access (in this case uploaded data is transferred through a conventional Public Switched Telephone Network (PSTN) modem or an Asymetric Digital Subscriber Line (ADSL) modem).

xx. Temperature

Temperature recording gives an indication of core body temperature which is normally tightly controlled (thermoregulation) as it affects the rate of chemical reactions (Chang, 2004).

Temperature can be recorded in order to establish a baseline for the individual's normal temperature for the site and measuring conditions. The main reason for checking body temperature is to solicit any signs of systemic infection or inflammation in the presence of a fever (temp > 38.5°C or sustained temp > 38°C), or elevated significantly above the individuals normal temperature.

xxi. The pill box

The 'pill box' helps keep track of the patient's medication by sending a signal to his or her mobile phone every time a pill is removed. If a patient forgets to take medication or is taking too many pills, he or she is sent a reminder via mobile phone to follow the prescribed doses.

xxii. The Weighted Average

A weighted average is an average that takes into account the proportional relevance of each component rather than treating each component equally.

It is an average in which each quantity to be averaged is assigned a weight. The weightings determine the relative importance of each quantity on the average (James, 2006). Weightings are the equivalent of having that many like items with the same value are involved in the average.

xxiii. The wristband blood-pressure monitoring device

The wristband blood-pressure monitoring device can also check other vital signs such as heart rate, and is activated by simply pressing a button. Blood-pressure readings, for example, are gathered from one or more sensors via a Bluetooth short-range radio connection. Once transmitted, secure access ensures only authorised medical personnel see the patient's data. If an unusual reading comes through, either a reminder can be sent to the patient to take his or her medication or a new prescription can be made, depending on the doctor's diagnosis.

xxiv. Transmission Control Protocol

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet Protocol Suite. TCP is one of the two original components of the suite (the other being Internet Protocol, or IP), so the entire suite is commonly referred to as *TCP/IP* (Comer, 2006). Whereas IP handles lower-level transmissions from computer to computer as a message makes its way across the Internet, TCP operates at a higher level, concerned only with the two end systems, for example a Web browser and a Web server.

xxv. Very Small Aperture Terminal (VSAT)

A Very Small Aperture Terminal (VSAT), is a two-way satellite ground station or a stabilized maritime VSAT antenna with a dish antenna that is smaller than 3 meters. The majority of VSAT antennas range from 75 cm to 1.2 m. Data rates typically range from 56 Kbit/s up to 4 Mbit/s. VSATs access satellites in geosynchronous orbit to relay data from small remote earth stations (terminals) to other terminals (in mesh configurations) or master earth station "hubs" (in star configurations).

xxvi. Vital signs

Vital signs are measures of various physiological statistics, often taken by health professionals, in order to assess the most basic body functions. Vital signs are an essential part of a case presentation. The act of taking vital signs normally entails recording Body temperature, Pulse rate (or heart rate), Blood pressure, and Respiratory rate, but may also include other measurements. Vital signs often vary by age.

The equipment needed are: a thermometer, a sphygmomanometer, and a watch. Though a pulse can often be taken by hand, a stethoscope may be required for a patient with a very weak pulse (Gao, 2005).

CHAPTER 2
LITERATURE REVIEW

2.1 Remote Medical Monitoring

The commoditization of computer hardware and software has enabled a new computing paradigm whereby computers will sense, calculate, and act on our behalf, either with or without human interaction as best fits the circumstances. Further, this will occur in an everyday environment, not just when a person is working at a desk (Andrew & Alfred, 2008).

Andrew and Alfred (2008) went on to discuss proactive computing, where computers anticipate the needs of people around them, and the implications its emergence will have for the public health. Their focus, as the article implies, is on networks of sensors that infuse the patient's milieu, including home and work and anywhere they go. They predict that telemedicine (the delivery of primary care in situ, wherever the patient may be, via information and communications technologies, or ICT) will be revolutionized by the decreasing cost and increasing power of the ICT that are becoming ubiquitous (many computers, one person) in our lives.

This paradigm shift was made possible by the inexorable increase in computing capabilities as we moved from mainframes (one computer, many people) to the personal computer (one computer, one person) to ubiquitous computing (many computers, one person), (Andrew & Alfred, 2008).
It is not uncommon to find a single person managing a desktop PC, laptop, cell phone, PDA, and portable media player. Today, these devices are discrete and managed individually. But as ubiquitous computing evolves, the computers will become both more numerous and less visible; they will be integrated into everyday life in a way that does not call attention to their presence (Andrew & Alfred, 2008).

In the context of medicine, ubiquitous computing presents an exciting challenge and a phenomenal opportunity. *Proactive computing* is a form of ubiquitous computing in which computers anticipate the needs of people around them. Wearable computing results from placing computers and sensors on the body to create a *body area network* (BAN) that can sense, process, and report on some set of the wearer's attributes. Proactive computing and wearable computing working in tandem let computers fade into the woodwork, enriching quality of life and engendering independence (Andrew & Alfred, 2008).

2.1.1 System Architecture

Andrew and Alfred (2008) from the University of Virginia developed a Remote Medical Monitoring system which consisted of three tiers, as shown in Figure 2.1 where each tier is distinguished by its locality and functionality within the broader system.

The first tier is the set of sensors that discern signals of interest, and relay information to each other and the data hub.

Tier two, the data hub, is a device that provides more computational capacity, allowing data to be stored or further processed before transmission to some outside medical network via the Internet, GSM, or some other means.

The third tier is the medical network, which is operated by a healthcare provider such as a hospital or telemedicine center where the staff can handle emergency situations. The remote medical monitoring system is seen by Andrew and Alfred (2008) as a hybrid of a broadcast service and a 9-1-1 service. In broadcast mode, it can provide periodic data updates to physicians regarding a patient's health; in 9-1-1 mode, it can autonomously raise a red flag whenever it detects a dangerous anomaly in monitored data (Andrew & Alfred, 2008).

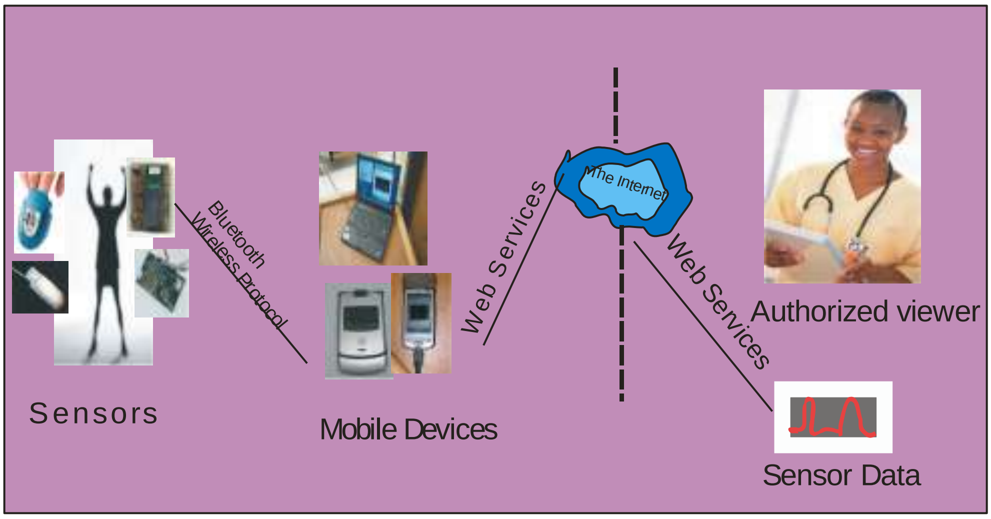


Figure 2.1: Remote medical monitoring system which consists of three tiers: one or more sensors that capture information about the patient, a data hub (such as a PDA, laptop or cell phone) for local data processing and display, and a medical network that records and analyzes information to detect anomalies.

Source: IEEE Computer Society. April 2008 Vol. 41 No.4

2.1.2 System Issues

The major system issues in the design of Andrew and Alfred is that their system did not make provisions for adequate security of the patient's data as it travels across networks and this can have adverse effects on the patient's data.

Also there is no form of authentication of the patient's data thereby making the system porous.

2.1.3 Amon

Anliker et al (2003) developed a medical device called AMON (alert portable telemedical monitor); which encapsulates many sensors (Pulse oximetry, ECG, accelerometer, and skin temperature) into one wrist-worn device that is connected directly to a telemedicine center via a GSM network, allowing direct contact with the patient if necessary.

The unit promises to be an important early prototypical remote medical monitoring system, but the results of testing in a medical study called for more research because most sensor outputs couldn't be used in a clinical setting—especially the ECG, which couldn't be detected reliably at the wrist.

Remote medical monitoring is currently in its infancy, but its future is bright. Patients with chronic diseases will be outfitted with appropriate sensors from which data will be transported to hubs for local processing; the data ultimately will be forwarded to secure medical networks for visualization and analysis by physicians, aided by software agents that monitor the data stream. Self-sufficient patients will use this technology to stay independent longer, and patients in assisted-living environments will benefit from continuous monitoring and a faster, better-informed medical response to adverse events. The push to take healthcare home in an inconspicuous and minimally invasive fashion equips individuals to be attuned to their health, encouraging a healthier and more well-informed society (Andrew & Alfred, 2008).

2.2 Telemedicine

Telemedicine is a rapidly developing application of clinical medicine where medical information is transferred through the phone or the Internet and sometimes other networks for the purpose of consulting, and sometimes remote medical procedures or examinations (Sachpazidis, 2008).

Telemedicine may be as simple as two health professionals discussing a case over the telephone, or as complex as using satellite technology and video-conferencing equipment to conduct a real-time consultation between medical specialists in two different countries (Nakajima et al., 2006). Telemedicine generally refers to the use of communications and information technologies for the delivery of clinical care.

Care at a distance (also called *in absentia* care), is an old practice which was often conducted via post (George et al., 1992). There has been a long and successful history of in absentia health care which, thanks to modern communication technology, has evolved into what we know as modern telemedicine.

In its early manifestations, African villagers used smoke signals to warn people to stay away from the village in case of serious disease. In the early 1900s, people living in remote areas in Australia used two-way radios, powered by a dynamo driven by a set of bicycle pedals, to communicate with the Royal Flying Doctor Service of Australia.

Angaran (2009) stated that the terms e-health and telehealth are at times wrongly interchanged with telemedicine. Like the terms "medicine" and "health care", telemedicine often refers only to the provision of clinical services while the term telehealth can refer to clinical and non-clinical services such as medical education, administration, and research. The term e-health is often, particularly in the UK and Europe, used as an umbrella term that includes telehealth, electronic medical records, and other components of health IT (Robert, 1995).

2.2.1 Types of Telemedicine

Telemedicine can be broken into three main categories: (Dennis et al., 2003).

i. store-and-forward,
ii. Remote monitoring and
iii. interactive services.

Store-and-forward telemedicine involves acquiring medical data (like medical images, biosignals etc) and then transmitting this data to a doctor or medical specialist at a convenient time for assessment offline. It does not require the presence of both parties at the same time (Dennis et al., 2003). Dermatology (cf: teledermatology), radiology, and pathology are common specialties that are conducive to asynchronous telemedicine. A properly structured Medical Record preferably in electronic form should be a component of this transfer. A key difference between traditional in-person patient meetings and telemedicine encounters is the omission of an actual physical examination and history. The store-and-forward process requires the clinician to rely on a history report and audio/video information in lieu of a physical examination.

Remote monitoring, also known as self-monitoring/testing, enables medical professionals to monitor a patient remotely using various technological devices (Traynor, 2010). This method is primarily used for managing chronic diseases or specific conditions, such as heart disease, diabetes mellitus, or asthma. These services can provide comparable health outcomes to traditional in-person patient encounters, supply greater satisfaction to patients, and may be cost-effective.

Interactive telemedicine services provide real-time interactions between patient and provider, to include phone conversations, online communication and home visits. Many activities such as history review, physical examination, psychiatric evaluations and ophthalmology assessments can be conducted comparably to those done in traditional face (Lisa et al., 2011) to-face visits. In addition, “clinician-interactive” telemedicine services may be less costly than in-person clinical visits.

Monitoring a patient at home using known devices like blood pressure monitors and transferring the information to a caregiver is a fast growing emerging service (Arora et al., 2011). These remote monitoring solutions have a focus on current high morbidity chronic diseases and are mainly deployed for the First World. In developing countries a new way of practicing telemedicine is emerging better known as Primary Remote Diagnostic (Kontaxakis et al., 2006). This new technology and principle of practicing medicine holds big promises to solving major health care delivery problems in for instance Nigeria because Primary Remote Diagnostic Consultations not only monitors an already diagnosed chronic disease, but has the promise to diagnosing and managing the diseases a patient will typically visit a general practitioner for (Weinstein et al., 2009).

2.2.2 Practical Implementation of Telemedicine with Special Emphasis on Remote Medical Monitoring Around the World.

i. Telemedicine Connects Earthquake-Ravaged Haiti to the World

February 18, 2010 — Just days after a devastating earthquake struck Haiti on January 12, medical volunteers from the University of Miami in Florida arrived and set up a tent hospital in Port-au-Prince (Kumar, 2010), Along with drugs and other medical supplies, the healthcare personnel brought a vital link to the rest of the world.

"Everything was destroyed over there, and in the first days there was no Internet in Port-au-Prince," team member Antonio Marttos Jr, MD, assistant professor of surgery at the University of Miami School of Medicine and director of trauma telemedicine at Jackson Memorial Medical Center's William Lehman Injury Research Center in Miami, Florida, told *Medscape Medical News*. "But we were able to connect to our trauma center in Miami" (Kumar, 2010).

Although telemedicine is now commonplace in the United States, it may surprise some that it is being used in poor, rural countries such as Haiti. Telemedicine experts say that high-tech advances in equipment make global telehealth possible and that use of this technology can improve diagnoses, speed treatment, and lower costs. It also comes in handy when a disaster has damaged or destroyed other communications.

2.2.3 Improvised Remote Medical Monitoring

When the devastating earthquake struck Haiti in January 12, 2010, some US physicians improvised a way to consult with specialists at their home hospital.
Some of the patients had horrific injuries and with limited access to lab findings, some difficult patient care decisions were taken. Some of these decisions were heartbreaking and irreversible, like having an amputation for the patients' concerned instead of treating the patients' said Dr Ashburn, a professor in the Department of Anesthesiology and Critical Care Medicine at the Hospital of the University of Pennysylvania in Philadelphia.

Even though live telemedicine was not possible in this situation, there was improvised remote medical monitoring where they used cameras on cell phones with internet access to take photos of patients' X–ray films and wounds and emailed them to stateside specialists, including a pediatric trauma surgeon Kumar (2010).

2.3 Design of a Wearable Sensor Network for Home Monitoring System

Eliasz et al. (2011) described a wearable ubiquitous healthcare monitoring system that integrates electrocardiogram (ECG device) and an accelerometer sensor with a mobile device in a Bluetooth-based body surface network (BSN). Their research focused on the right connection of the hardware units, combination of the detection of QRS complexes [The QRS complex is the name for the combination of three of the graphical deflctions seen on a typical electrocardiogram (ECG)], calculation of heart rate (HR) and the detection of human falls. The main aim of Eliasz et al. (2011) research was the early detection of abnormal situations (high/low HR, a fall) and the heart rate variability analysis.

The system integrates 12-leads ECG signal transmitter (ASPEKT 500, which is a digital unit designated for wireless ECG signal transmission to PC), an accelerometer sensor and a mobile device in a Bluetooth-based body surface network (BSN) (Eliasz et al., 2011).

2.3.1 Hardware Units

Eliasz et al. (2011) used ASPEKT 500 manufactured by Aspel which is a digital unit designated for wireless ECG signal transmission to PC or mobile device. The transmitter allows a free patients movement up to 10 m from receiver. Small dimensions and weight make the examination more comfortable for a patient.

Figure 2.2:ASPEKT 500 - Wireless ECG signal transmitter

Source: Aspel ASPEKT 500 technical documentation: http://www.aspel.com.pl/index.php?lang=pl&pg=372

The ASPEKT 500 - Wireless ECG signal transmitter is equipped with ten electrode cable. In order to receive 12- leads (Einthoven, Goldberger, Wilson) the electrodes are connected as shown in fig. 2.5

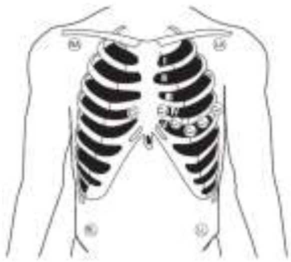


Figure 2.3: Electrodes location on the patient

Source: Aspel ASPEKT 500 technical documentation: http://www.aspel.com.pl/index.php?lang=pl&pg=372

ECG data is sampled at 500 Hz frequency. Battery operated time is about 12 hours.

2.3.2 Methods

The most important aim in the monitoring system is the detection of the QRS complexes in real-time (Eliasz et al., 2011). With this it was easy for them to calculate the heart rate and observe the heart rate variability. The data processing is crucial to extract the correct part of the signal, the QRS complex. The ECG waveform contains also P, T, sometimes U waves and a lot of noise (60 Hz power line noise, EMG, motion artifacts).

One of the most popular and often cited QRS detection algorithms that works in the time domain is the Pan and Tomkins algorithm that was proposed in 1985 (Tompkins

& Pan,1985). The QRS detection algorithm is based on analysis of the slope, amplitude and width of the QRS complex which refers to the depolarization of the right and left ventricles

One of the project goals is to detect a fall (Eliasz et al., 2011). Fall detection could be achieved by analyzing the accelerometer data. The most common method of detecting a fall is calculating the absolute sum of ACC signal in different direction as shown in eq. (2.1).

$$FALL = \sqrt{ACC_X^2 + ACC_Y^2 + ACC_Z^2} \tag{2.1}$$

Source: Biomedical Signal Analysis, IEEE Press, Vol. BME-32, NO. 3, 1985

2.3.3 System Architecture

Eliaz et al. (2011) developed a monitoring system prototype and the analyzed software was implemented in order to check if it was possible to monitor human activity remotely. The wireless ElectroCardioGram (ECG, which records heart activity) recorder transmits data to mobile device via wireless network based on Bluetooth technology. Data is analyzed by implemented software and forwarded using general packet radio service to Internet database, where medical data is accessible to a doctor.
Implemented software is organized into 4 modules: ECG signal analysis, Fall detection, Heart Rate Variability and Database.

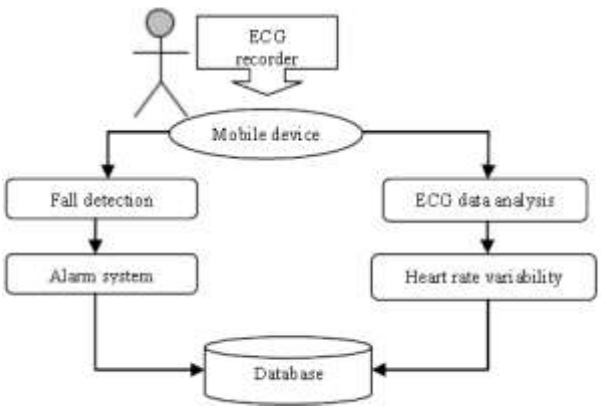


Figure 2.4: Monitoring system architecture.

Source: Proceedings of the Federated Conference on Computer Science and Information Systems.

The most important module of the monitoring system is the ECG data analysis module which analyses the incoming signal data from the ECG recorder. The output of the module is transmitted to the heart rate variability module (Eliasz et at., 2011).

Eliasz et al's fall detection module is responsible for analyzing data from accelerometer and calculating the FALL factor (FALL factor is the ratio of the height (h) a person falls and the stretch of the length (l) available to absorb the energy of the fall.

The FALL factor is given by F = $\frac{h}{l}$ (2.2)

2.3.4 Tests and Results

One of the most important stages in the implementation of the monitoring system is its verification. The QRS detection algorithm using ECG signals was tested by Eliasz et al (2011) at the 402 Proceedings of the Federated Conference on Computer Science and Information Systems.

Eliasz et al (2011) presented four cases of QRS detection possibilities as shown in Table 2.1.

Table 2.1:QRS complex detection possibilities

QRS detection	Absence of QRS complex	QRS complex occurs
Detected	FP (False Positive)	TP (True Positive)
Undetected	TN (True Negaive)	FN (False Negative)

The effectiveness of the application as performed by Eliasz et al (2011) was by calculating the four cases (FP, TP, TN, FN) mentioned in Table 2.1 for every file from the database.

The most common used equation to demonstrate the effectiveness of the QRS detection algorithm is the sensitivity (Rangayyan, 2010).

The sensitivity equation is given as:

Sensitivity = $\left[\frac{TP}{TP+FN}\right]100$ (2.3)

Where TP isTrue positive and

FP is False Negative

Table 2.2 presents results for selected files as shown by Eliasz et al (2011)

Table 2.2: Results for the database files

Signal Nr.	Total QRS complex (FN + TP)	TP	Sensitivity
100	2273	2273	100 [%]
101	1865	1863	99.8 [%]
105	2572	2541	98.6 [%]
113	1795	1711	95.2 [%]
117	1535	1498	97.4 [%]

The proposed method of detecting falls did not not give satisfactory results in performed tests. It often provided with false positive detections especially during rapid movements or stops. Further research is needed in order to elaborate more accurate algorithm (Rangayyan, 2010).

2.3.5 Findings

The proposed healthcare monitoring system can help to monitor health conditions (heart rate, heart rate variability) and support elderly, sick and disabled people in their independent living (Eliasz et al., 2011).

An SMS messaging module was integrated with the monitoring system. After detecting a nonstandard situation a short notice is sent. Eliasz et al. (2011) stated that future works should be to develop more advanced algorithms for detecting non-standard situations and improve fall detection algorithm.

2.4 Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare

Juan et al. (2009) presented a paper on a distributed telemonitoring system aimed at improving healthcare and giving assistance to dependent people at their homes. Their system implemented a SOA-based (Service Oriented Architecture) platform which is capable of allowing heterogeneous wireless sensor networks to communicate in a distributed way independently of time and location restrictions. Their approach provides the system with a higher ability to recover from errors and a better flexibility to change their behaviour at execution time.

They looked at the different ways technology can be handled and according to Jayaputera (2007), stated that the continuous advancement in mobile computing makes it possible to obtain information about the context and react physically to it in more innovative ways. However they stated that in situations where technology is difficult to handle, the Ambient intelligence (AmI) tries to adapt the technology to the people's needs by proposing three basic concepts: ubiquitous computing, ubiquitous communication and intelligent user interfaces.

Anderson (2000) stated that the World Health Organization determined that in the year 2025 there will be 1 billion people in the world over the age of 60 and twice as many by 2050, with nearly 80% concentrated in developed countries. In fact, people over 60 years old represent more than 21% of the European population (Anderson, 2000) and people over 65 are the fastest growing segment of the population in the United States of America (Anderson, 2000). Furthermore, over 20% of those people over 85 have a limited capacity for independent living, requiring continuous monitoring and daily assistance (Pennifer, 1995). The importance of developing new and more reliable ways of providing care and support for the elderly is underscored by this trend, and the creation of secure, unobtrusive and adaptable environments for monitoring and optimizing health care will become vital (Corchado et al., 2008).

The telemonitoring system was aimed at improving healthcare and assistance to dependent people at their homes. This system makes use of the *Services laYers over Light PHysical devices* (SYLPH) platform. SYLPH is based on a Service-Oriented Architecture (SOA) model for integrating heterogeneous Wireless Sensors Networks

(WSNs) into AmI systems (Ahmed et al., 2006). SYLPH focuses on distributing the systems'functionalities into independent functionalities (i.e. services).

This model provides a flexible distribution of resources and facilitates the inclusion of new functionalities in highly dynamic environments. Wireless sensor networks provide an infrastructure capable of supporting the distributed communication needed in highly dynamic environments, in this case an AmI-based telemonitoring system for a dependency scenario, increasing mobility, flexibility and efficiency since resources can be accessed regardless their physical location (Gast & Gast, 2002).

2.4.1 Motivation and Problem Description

Ardissono et al. (2004) discussed the emergence of Ambient Intelligence which involves substantial changes in the design of functional architectures, since it is necessary to provide features which enable a ubiquitous computing and communication and also an intelligent interaction with users. This section discusses some of the most important problems of existent functional architectures, including their suitability for constructing intelligent environments according to the Ambient Intelligence paradigm (Ardissono et al., 2004).

Excessive centralization of services negatively affects the systems' functionalities, overcharging or limiting their capabilities. Classical functional architectures are characterized by trying to find modularity and a structure oriented to the system itself. Modern functional architectures like SOA consider integration and performance aspects that must be taken into account when functionalities are created outside the system. A SOA-based system is a network of independent services, machines, the people who operate, affect, use, and govern those services as well as the suppliers of equipment and personnel to these people and services (Gast & Gast, 2002).
Cerami (2002) defined service as a mechanism that facilitates the access to one or more functionalities (e.g. functions, network capabilities, etc.). The SOA model is aimed at the interoperability between different systems, distribution of resources, and the lack of dependency of programming languages (Cerami, 2002). Services are

linked by means of standard communication protocols that must be used by applications in order to share resources in the services network (Ardissono et al., 2004). The compatibility and management of messages that the services generate to provide their functionalities is an important and complex element in any of these approaches. A distributed architecture provides more flexible ways to move functions to where actions are needed, thus obtaining better responses at execution time, autonomy, services continuity, and superior levels of flexibility and scalability than centralized architectures (Camarinha-Matos & Afsarmanesh, 2007). Unfortunately, the difficulty in developing a distributed architecture is higher (Rigole et al., 2002). This way, it is necessary to have a more complex system analysis and design, which implies more time to reach the implementation stage. AmI-based developments will require the use of several sensors and actuators strategically distributed in the environment. This provides the systems with context-aware capabilities in order to change its behavior automatically. It is possible to make a difference between sensor networks: wired and wireless. There are several technologies for creating wired sensors networks, such as X10, LonWorks or KNX. However, wired networks are not as flexible as wireless sensors networks and require more infrastructural support (Ahmed et al., 2006) (Meong-hun & Hyun, 2007). On the other hand, wireless technologies enable easier deployments than the wired ones, avoiding the need of wiring homes or hospitals and decreasing the costs and drawbacks of the setup phase.

The ZigBee standard allows operating in the frequency range belonging to the radio band known as ISM (Industrial, Scientific and Medical), especially in the 868MHz band in Europe, the 915MHz in the USA and the 2.4GHz in almost all over the world (Huang & Pang, 2007).The underlying IEEE 802.15.4 standard is designed to work with low-power and limited computational resources nodes (Singh et al, 2008). ZigBee incorporates additional network, application and security layers over the 802.15.4 standard (Baronti et al., 2007). The ZigBee standard allows up to 65,534 nodes connected in a star, tree or mesh topology network. Another standard to deploy wireless sensor networks is Bluetooth. This standard allows multiple Wireless Personal Area Networks (WPAN) and Wireless Body Area Networks (WBAN) applications for interconnecting mobile phones, earphones, personal computers, printers, etc. Bluetooth operates also in the ISM 2.4GHz band. It allows creating star topology networks of up to 8 devices in which one of them acts as master and the rest as slaves. Several Bluetooth networks can be interconnected by means of Bluetooth devices that belong simultaneously to two or more networks creating more extensive networks (Ilyas, 2002). Although there are plenty of options for creating (Wireless Sensor Networks) WSNs, the main problem is the difficulty for integrating devices from different technologies in a single network (Shu et al, 2006). In addition, the lack of a common architecture may lead to additional costs due to the necessity of deploying non-transparent interconnection elements between networks. Moreover, the developed elements (e.g. devices) are too dependent on the application to which they belong, thus complicating their reutilization. Some developments try to reach the devices integration by implementing middleware layers as reduced versions of virtual machines (e.g. Squawk Java Virtual Machine) (Simon & Cifuentes, 2005). These developments require devices with high computational power and large memory microcontrollers. For these reasons, there are needed more expensive devices with larger size or more costly miniaturization. These drawbacks are very important regarding WSNs, as it is essential to deploy applications with reduced resources and low infrastructural cost especially in home care scenarios. The SYLPH platform integrates a SOA approach for facilitating the distribution and management of resources (i.e. services) into heterogeneous WSNs. There are several attempts to integrate WSNs and a SOA approach (Meshkova et al., 2008; Moeller & Sleman, 2008; Prinsloo et al., 2006; Sleman & Moeller, 2008; Song & Kang, 2007). In SYLPH, unlike those approaches, services are directly embedded on the WSN nodes and can be invoked from other nodes in the same network or other network connected to the former one.

It is necessary to provide efficient solutions that allow building AmI environments for providing dependent people healthcare at their homes. One of the key aspects for the construction of these environments is obtaining context information through sensor networks. There are several healthcare developments for telemonitoring based on

WSNs (Andrew & Alfred, 2008; Fass, 2007; Varshney, 2008). However, these developments do not take into account their integration with other systems and are difficult to be adapted to new situations. The use of SYLPH is proposed in order to face some of the issues found when integrating heterogeneous wireless sensor networks.

2.4.2 The SYLPH Platform

Camarinha-Matos and Afsarmanesh (2007) described the *Service laYers over Light PHysical devices* (SYLPH) platform as a distributed architecture which integrates a SOA approach over Wireless Sensors Networks for building systems based on the Ambient Intelligence paradigm. The main objective is to distribute resources over multiple WSNs by modeling the functionalities as independent services. A service oriented approach has been chosen because such architectures are asynchronous and non-dependent on context (i.e. previous states of the system, which must not be confused with context-aware environments). Thus, devices working on them do not take up continuously processing time and are free to do other tasks or consume less energy. SYLPH can be executed over multiple wireless devices independently of their microcontroller or the programming language they use.

2.4.3 The Telemonitoring System

Fass (2007) describes the main features of a telemonitoring system which is aimed at improving healthcare of dependent people at their homes. The system makes use of several WSNs in order to gather context information in an automatic and ubiquitous way.

Several functionalities are directly embedded on the WSN nodes and can be invoked from other nodes in the same network or other network connected to the former one by means of the SYLPH platform. SYLPH Gateways are used in order to interconnect different heterogeneous WSNs. This way, SYLPH contemplates the possibility of connecting WSNs based on different radio and link technologies (e.g. ZigBee, Bluetooth, Wi-Fi, etc.), whilst other approaches do not. In addition, SYLPH focuses specially on devices with small resources in order to save microcontrollers' computing time, memory data size and energy consumption.

Biomedical sensors (e.g. electrocardiogram, blood pressure, body temperature, etc.) and automation sensors (e.g. building temperature, light, humidity, etc.) have significant differences especially on how they collect data. Biomedical sensors obtain continuous information about vital signs whose samples are important and should not be lost (Jayaputera et al., 2007). On the other hand, automation sensors obtain information at a relatively lower frequency compared to biomedical sensors (Sarangapani, 2007). In addition, biomedical sensors should be smaller and easier to wear. It is necessary to interconnect several WSNs from different radio technologies in a telemonitoring scenario (Andrew & Alfred, 2008). Having a compatible distributed platform for deploying healthcare applications over the different networks facilitates the developers' work and the integration of the heterogeneous devices.
Huang and Pang (2007) described the basic communication and infrastructure schema of the telemonitoring system. A network of ZigBee devices was been designed to cover the home of each patient to be monitored. There is a ZigBee remote control carried by the monitored patient that incorporates a button which can be pressed in case of remote assistance or urgent help. Moreover, there are a set of ZigBee sensors that obtain information about the home environment (e.g. light, smoke, temperature, doors' states, etc.) in which the user lives and that physically responds to the changes (e.g. light dimmers, fire alarms or door locks) (Huang & Pang, 2007).

2.4.4 Results

According to Corchado et al. (2009) the telemonitoring system presented in their paper improves security at home to dependents. Corchado et al. (2009) performed several test cases in order to evaluate the overall performance of the system, especially the management of emergency situations. The tests, which involved 13 patients and 6 caregivers, allowed them to evaluate the system. Specifically, the results of the SYLPH system were studied over a period of 4 weeks. Their approach was evaluated in terms of the four main objectives defined for WSN applied in healthcare developments: Minimize error rates; Conduct diagnosis with real time patient data; Improve efficiency and Reduce costs (Fass, 2007).

The initial tests showed an error rate above 19%. This percentage was primarily due to errors in the use of the system by the caregivers. After the third test, the error rate was reduced to 11%. From that point the error rate remained stable at 4%. This error could be reduced with a higher level of training in the use of the telemonitoring system (Corchado et al., 2009). False positives occurred during the test period of the alert subsystem.

Finally, the efficiency of the telemonitoring system has been enhanced and the infrastructure cost has been reduced.

An important issue they had was to improve the security and privacy of their system.

2.5 A Wireless Body Area Network of Intelligent Motion Sensors for Computer Assisted Physical Rehabilitation

Istepanian et al. (2004) stated that wearable health monitoring systems integrated into a telemedicine system are novel information technology that will be able to support early detection of abnormal conditions and prevention of its serious consequences.

A wearable health-monitoring device using a Personal Area Network (PAN) or Body Area Network (BAN) can be integrated into a user's clothing (Park & Jayaraman, 2003). This system organization, however, is unsuitable for lengthy, continuous monitoring, particularly during normal activity (Martin et al., 2000), intensive training or computer-assisted rehabilitation. Recent technology advances in wireless networking (Otis, 2003) micro-fabrication (Ghovanloo, 2002) and integration of physical sensors, embedded microcontrollers and radio interfaces on a single chip (Raskovic et al., 2004) promise a new generation of wireless sensors suitable for many applications (Raskovic et al., 2004). However, the existing telemetric devices either use wireless communication channels exclusively to transfer raw data from sensors to the monitoring station, or use standard high-level wireless protocols such as Bluetooth that are too complex, power demanding, and prone to interference by other devices operating in the same frequency range. These characteristics limit their use for prolonged wearable monitoring. Simple, accurate means of monitoring daily activities outside of the laboratory are not available (Aminian et al., 1999) at the present, only estimates can be obtained from questionnaires, measures of heart rate, video assessment, and use of pedometers (Aminian et al., 1999) or accelerometers (Milenkovic, 2002). Finally, records from individual monitoring sessions are rarely integrated into research databases that would provide support for data mining and knowledge discovery relevant to specific conditions and patient categories.

Lee and Mase (2002) described the typical examples of possible applications to include stroke rehabilitation, physical rehabilitation after hip or knee surgeries, myocardial infarction rehabilitation, and traumatic brain injury rehabilitation. The assessment of the effectiveness of rehabilitation procedures has been limited to the laboratory setting; relatively little is known about rehabilitation in real-life situations. Miniature, wireless, wearable technology offers a tremendous opportunity to address this issue (Lee & Mase, 2002).

Otto et al. (2005) proposed a wireless BAN composed of off-the-shelf sensor platforms with application-specific signal conditioning modules (Otto et al., 2005). In this paper, they presented a general system architecture and described a recently developed activity sensor "ActiS" (Pappas et al., 2004). ActiS is based on a standard wireless sensor platform and a custom sensor board with a one-channel bio amplifier and two accelerometers (Steele et al., 2003) stated that as a heart sensor, ActiS can be used to monitor heart activity and position of the upper trunk. The same sensor can be used to monitor position and activity of upper and lower extremities. A wearable system with ActiS sensors would also allow one to assess metabolic rate and cumulative energy expenditure as a valuable parameter in the management of many medical conditions (Pappas et al., 2004). Warren (2003) gave insight into an early version of the ActiS which is based on a custom developed wireless intelligent sensor and custom wireless protocols in the license-free 900 MHz Scientific and Medical Instruments (ISM) band (Warren, 2003). Warren (2003) stated that their initial experience indicated the importance of standard sensor platforms with ample processing power, minute power consumption, and standard software support. Such platforms were not available on the market during the design of the first prototype system (Warren, 2003). Standard hardware and software architecture facilitate interoperable systems and devices that are expected to significantly influence next generation health systems (Tharion et al., 2004). This trend can also be observed in recently developed physiological monitors systems from Harvard (Van et al., 2003) and (Welch-Allen et al., 2003).

2.5.1 System Architecture

According to Jovanov et al. (2000), continuous technological advances in integrated circuits, wireless communication, and sensors enable development of miniature, non-invasive physiological sensors that communicate wirelessly with a personal server and subsequently through the Internet with a remote emergency, weather forecast or medical database server; Jovanov et al. (2000) using baseline (medical database), sensor (WBAN) and environmental (emergency or weather forecast) information, algorithms may result in patient-specific recommendations. Jovanov et al. (2000) designed a personal server, running on a PDA or a 3 G cell phone, which provides the human-computer interface and communicates with the remote server(s). Figure 2.5 gives a generalized overview of a multi-tier system architecture; the lowest level encompasses a set of intelligent physiological sensors; the second level is the personal server (Internet enabled PDA, cell-phone, or home computer); and the third level encompasses a network of remote health care servers and related services (Caregiver, Physician, Clinic, Emergency, Weather). Each level represented a fairly complex subsystem with a local hierarchy employed to ensure efficiency, portability, security, and reduced cost. Figure 2.5 showed an example of information flow in an integrated WBAN system.

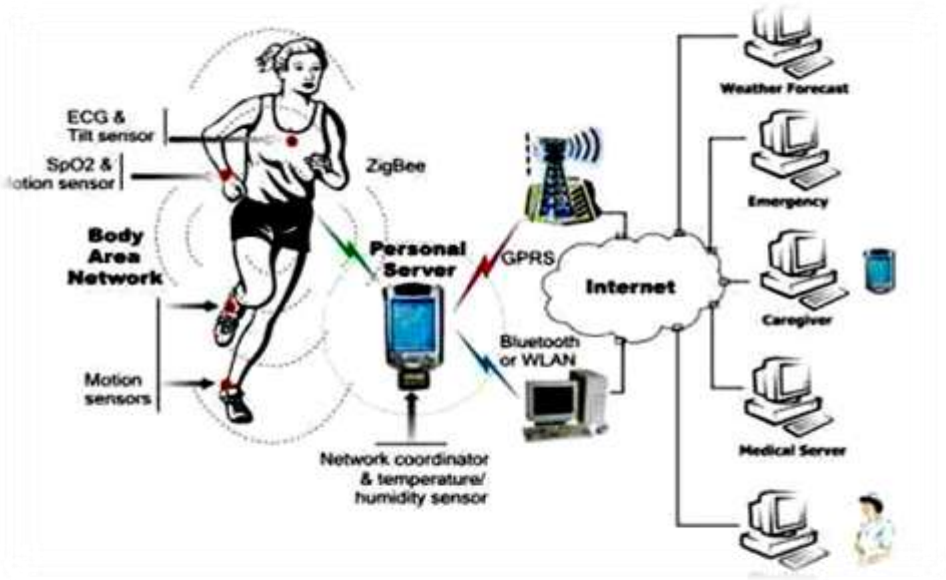


Figure 2.5: Wireless Body Area Network of Intelligent Sensors for Patient Monitoring
Source: Journal of Neuro Engineering and Rehabilitation 2005, **2**:6 doi:10.1186/1743-0003-2-6.

2.5.2 Sensor level

Jovanov et al. (2000) went further to describe a WBAN which will include a number of physiological sensors depending on the end-user application. Information of several sensors can be combined to generate new information such as total energy expenditure. An extensive set of physiological sensors may include the following:

i. an ECG (electrocardiogram) sensor for monitoring heart activity
ii. an EMG (electromyography) sensor for monitoring muscle activity
iii. an EEG (electroencephalography) sensor for monitoring brain electrical activity
iv. a blood pressure sensor
v. a tilt sensor for monitoring trunk position
vi. a breathing sensor for monitoring respiration
vii. movement sensors used to estimate user's activity
viii. a "smart sock" sensor or a sensor equipped shoe insole used to delineate phases of individual steps.

These physiological sensors typically generate analog signals that are interfaced to standard wireless network platforms that provide computational, storage, and communication capabilities. Multiple physiological sensors can share a single wireless network node. In addition, physiological sensors can be interfaced with an intelligent sensor board that provides on-sensor processing capability and communicates with a standard wireless network platform through serial interfaces (Otto et al., 2005).
The wireless sensor nodes should satisfy the following requirements: minimal weight, miniature form-factor, low-power operation to permit prolonged continuous monitoring, seamless integration into a WBAN, standard-based interface protocols, and patient-specific calibration, tuning, and customization. These requirements represent a challenging task, (Otto et al., 2005) especially where one vendor creates all components. Only hybrid systems implemented by combining off-the-shelf, commodity hardware and software components, manufactured by different vendors promise proliferation and dramatic cost reduction (Otto et al., 2005).

The wireless network nodes (Otto et al., 2005) designed was implemented as tiny patches or incorporated into clothes or shoes. The network nodes continuously collect and process raw information, store them locally, and send them to the personal server. Type and nature of a healthcare application will determine the frequency of relevant events (sampling, processing, storing, and communicating). Ideally, sensors periodically transmit their status and events, therefore significantly reducing power consumption and extending battery life. When local analysis of data is inconclusive or indicates an emergency situation, the upper level in the hierarchy can issue a request to transfer raw signals to the upper levels where advanced processing and storage is available (Raskovic et al., 2004).

2.5.3 Personal Server Level

The personal server performs the following tasks:

i. Initialization, configuration, and synchronization of WBAN nodes

ii. Control and monitor operation of WBAN nodes

iii. Collection of sensor readings from physiological sensors

iv. Processing and integration of data from various physiological sensors providing better insight into the users state

v. Providing an audio and graphical user-interface that can be used to relay early warnings or guidance (e.g., during rehabilitation)

vi. Secure communication with remote healthcare provider servers in the upper level using Internet services (Raskovic et al., 2004).

The personal server can be implemented on an off-the-shelf Internet-enabled PDA (Personal Digital Assistant) or 3 G cell phone, or on a home personal computer. Multiple configurations are possible depending on the type of wireless network employed (Raskovic et al., 2004). For example, the personal server can communicate with individual WBAN nodes using the Zigbee wireless protocol that provides low-power network operation and supports virtually an unlimited number of network nodes.

Otto et al. (2005) believed that relying on off-the-shelf mobile computing platforms is crucial, as these platforms will continue to grow in their capabilities and quality of services. The challenging tasks are to develop robust applications that provide simple and intuitive services (WBAN setup, data fusion, questionnaires describing detailed symptoms, activities, secure and reliable communication with remote medical servers, etc). Total information integration will allow patients to receive directions from their healthcare providers based on their current conditions (Otto et al., 2005).

2.5.4 Medical Services

Otto et al. (2005) envisioned various medical services in the top level of the tiered hierarchy. A healthcare provider runs a service that automatically collects data from individual patients, integrates the data into a patient's medical record, processes them, and issues recommendations, if necessary (Otto et al., 2005). These recommendations are also documented in the electronic medical record. If the received data are out of range or indicate an imminent medical condition, an emergency service can be notified (this can also be done locally at the personal server level). The exact location of the patient can be determined based on the Internet access entry point or directly if the personal server is equipped with a GPS sensor (Lee & Mase., 2002). Medical professionals can monitor the activity of the patient and issue altered guidance based on the new information, other prior known and relevant patient data, and the patient's environment (e.g., location and weather conditions) (Lee & Mase, 2002).

2.5.5 ActiS – Activity Sensor

Pappas et al. (2004) developed ActiS sensor specifically for WBAN-based, wearable computer-assisted, rehabilitation applications. They integrated a one-channel bio-amplifier and three accelerometer channels with a low power microcontroller into an intelligent signal processing board that can be used as an extension of a standard wireless sensor platform. ActiS consists of a standard sensor platform, (Pappas et al., 2004) Telos, from Moteiv and a custom Intelligent Signal Processing Module – ISPM.

Figure 2.6: Telos wireless platform with intelligent signal processing daughtercard ISPM

Source: Journal of Neuro Engineering and Rehabilitation 2005, **2**:6 doi:10.1186/1743-0003-2-6.

The Telos platform is an ideal fit for this application due to small footprint and open source system software support. A second generation of the Telos platform features an 8 MHz MSP430F1611 microcontroller with integrated 10 KB of RAM and 48 KB of flash memory, a USB (Universal Serial Bus) interface for programming and communication, and an integrated wireless ZigBee compliant radio with on-board antenna (Otto et al, 2005).

System Design Issues

The main system design issues (Pappas et al, 2004) encountered include:

i. types of sensors
ii. power source
iii. size and weight of sensors
iv. wireless communication range and transmission characteristics of wearable sensors
v. sensor location and mounting
vi. seamless system configuration
vii. automatic uploads to the patient's electronic medical record
viii. intuitive and simple user interface (Pappas IPI et al., 2004).

2.5.6 Types of sensors

As for sensors, accelerometers and gyroscopes offer greater sensitivity and are more applicable for monitoring of motion since they generate output *(*Krause et al., 2003) found that frequency of human induced activity ranges from 1 to 18 Hz. Sampling rates in the existing projects vary from 10 – 100 Hz. Almost all projects in the last five years use MEMS accelerometers or a combination of accelerometers and gyroscopes (Pappas, IPI et al., 2004).

2.5.7 Power Source, Size/Weight, and transmission characteristics

To be unobtrusive, the sensors must be lightweight with small form factor. The size and weight of sensors is predominantly determined by the size and weight of batteries. Requirements for extended battery life directly oppose the requirement for small form factor and low weight (Kern et al., 2003).This implies that sensors have to be extremely power efficient, as frequent battery changes for multiple WBAN sensors would likely hamper users' acceptance and increase the cost. In addition, low power consumption is very important as we move toward future generations of implantable sensors that would ideally be self-powered, using energy extracted from the environment (Kern et al., 2003).

2.5.8 Location of Sensors

Although the purpose of the measurement does influence sensor location, researchers seem to disagree on the ideal body location for sensors. A motion sensor attached to an ankle is the most discriminative single position for state recognition, while a combination of hip and ankle sensors discriminates the states even more (Aminian et al., 2001). In a study of the relationship between metabolic energy expenditure and various activities, researchers at Eindhoven University of Technology, the Netherlands, placed tri-axial accelerometers on a subject's back waistline (Melnick et al., 2002). Krause *et al* use two accelerometers on the Sense Wear armband (Pappas et al., 2004). Park and Jayaraman (2003) placed accelerometer sensors in the subject's thigh pocket in order to measure angular position and velocity of the thigh. Doing so, they were able to accurately monitor a subject's activity and with the assistance of gyroscopes and compass headings were able to successfully estimate a subject's change in location.

2.5.9 Seamless System Configuration

The intelligent WBAN sensors should allow users to easily assemble a robust ad-hoc WBAN, depending on the user's state of health (Van et al., 2003) proposed a standard off-the-shelf sensors, manufactured by different vendors, and sold "over-the-counter" (Van et al, 2003). Each sensor should be able to identify itself and declare its operational range and functionality. In addition, they should support easy customization for a given application.

2.5.10 Social Issues

Melnick et al. (2002) stated that the social issues of WBAN systems include privacy/security and legal issues and due to the communication of health-related information between sensors and servers, all communication over WBAN and Internet should be encrypted to protect user's privacy. Legal regulation will be necessary to regulate access to patient-identifiable information.

2.5.11 Possible Applications

According to Winters et al. (2003), the WBAN technology can be used for computer-assisted physical rehabilitation in ambulatory settings and monitoring of trends during recovery. An integrated system can synergize the information from multiple sensors, warn the user in the case of emergencies, and provide feedback during supervised recovery or normal activity (Winters et al., 2003). Candidate applications include post-stroke rehabilitation, orthopaedic rehabilitation (e.g. hip/knee replacement rehabilitation), and supervised recovery of cardiac patients (Pappas et al., 2004). In the case of orthopaedic rehabilitation the system can measure forces and accelerations at different points and provide feedback to the user in real-time. Unobtrusive monitoring of cardiac patients can be used to estimate intensity of activities in user's daily routine and correlate it with the heart activity (Pappas et al., 2004).

2.5.12 Findings

Jovanov et al. (2000) designed a wearable Wireless Body Area Network (WBAN) of physiological sensors which when integrated into a telemedical system holds the promise to become a key infrastructure element in remotely supervised, home-based patient rehabilitation. It has the potential to provide a better and less expensive alternative for rehabilitation healthcare and may provide benefit to patients, physicians, and society through continuous monitoring in the ambulatory setting, early detection of abnormal conditions, supervised rehabilitation, and potential knowledge discovery through data mining of all gathered information (Jovanov et al., 2000).

2.6 A Wireless Medical Monitoring Over a Heterogeneous Sensor Network

Mehmet et al. (2007) built a Wireless Body Sensor Network (WBSN) based on the newly available 402-405 MHz MICS (Medical Implant ommunication Service) band. The project used both implanted and on-body (i.e. external) nodes. According to Mehmet et al. (2007), instead of applying other standards such as Bluetooth, ZigBee or WLANs for data collections from sensors, this band was particularly chosen in their design to eliminate the strong interference from other devices as reliable communication and accurate monitoring are very crucial for patients' lives. The MICS do not also cause interference to other users of the electromagnetic spectrums (Tekin et al., 2005). Another advantage they gave was that when nodes used are implantable, it provided small size, low-power, faster data transfer and longer communication range (Tekin et al., 2006).

Combining all these features with the acceptance of the 402-405 MHz band internationally offers an attractive frequency choice for the targeted WBSN application.

Previously Yuce *et al* (2007) reported a MICS based WBSN in that it has the capability of simplex communication and only allows one patient's condition to be monitored locally. In this work they presented a multi-patient monitoring system with data transfer ability over a network or the internet to a remote computer. Hardware and software designs were developed to reach their goals. A media access layer (MAC) has been successfully implemented to support multi-patient monitoring facility. Moreover, data accessibility has been successfully extended to a remote computer that maybe located within a local area network (i.e. LAN) or to the internet. Hardware modules have been designed to support both indoor and outdoor environments which means that a patient can be either in a medical center or in his/her house or an outside location that is close to the internet or a mobile communication network.

2.6.1 Multi-Patient Body Sensor Network System

Mehmet et al (2007) proposed a WBSN which comprises of sensor nodes, a Central Control Unit (CCU) which transmits data to a local PC and a receiver station at a medical center. The sensor nodes are responsible for obtaining raw data from a human body. After processing, they transmit those data to the CCU via the wireless RF link using the MICS band (Scanlon et al., 2000). The CCU repackages the data and transmits to the local PC. The data collected at the local PC is transferred to a remote PC across the network in a medical center or through internet if it is at a different location than the Medical centre. Fig 2.7 describes the architecture of the heterogeneous wireless network system designed by Mehmet et al (2007).

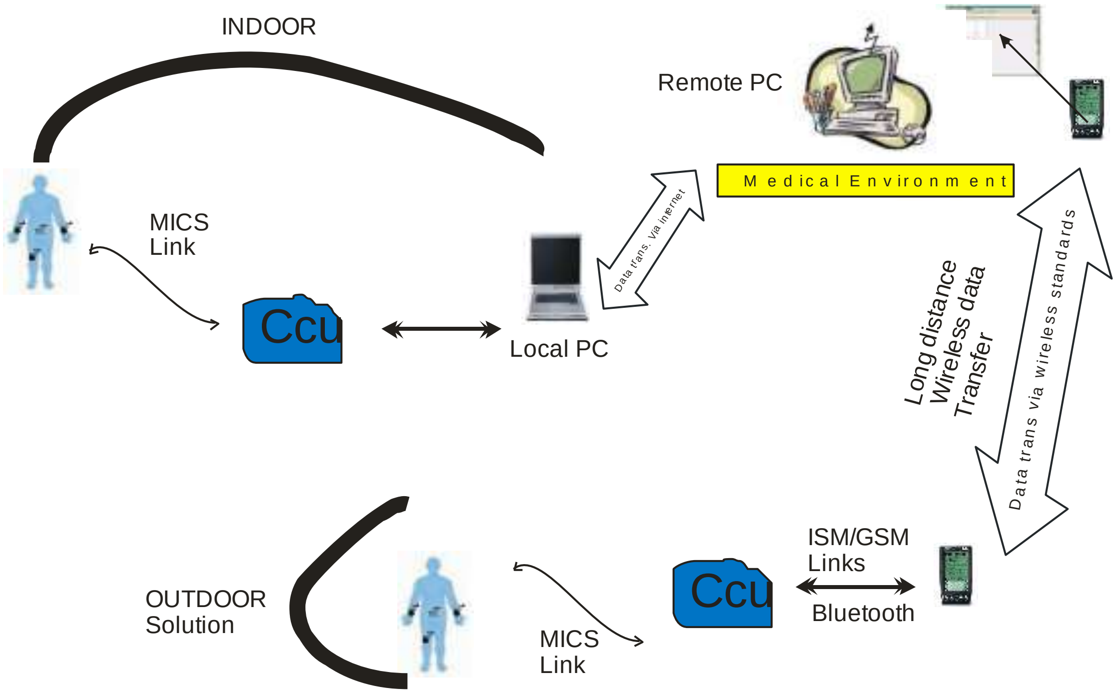


Figure 2.7: A heterogeneous wireless network system for medical monitoring.
Source: http://www.wirelessis.com , April 2007.

The receiver station (i.e. the remote PC) displayed all the received data on a User Display Graphic (GUI) and is also capable of storing all the data in the database system of a medical center (Tekin et al., 2005). The CCU is targeted to be worn around a patient's waist. For the patients with limited mobility, the CCU will be replaced at an accessible location at a distance of up to 10 meter in a house. In the later case, the CCU is connected to a local PC that displays the real-time information received from the sensor nodes and records the information to a database locally, the same way that is done at the remote PC of the medical center (Tekin et al., 2006). The stored information can be sent via internet to the database of the medical center on a periodic basis. When more than one patient is accommodated in a room as in the case of a hospital, the necessary software packages are installed in the CCU and the local PC in order to obtain physiological signals from sensors of each patient. Data gathering is performed at some certain time intervals assigned to patients. Pulse rate and temperature are the two vital signs selected to be monitored in the prototype system to demonstrate a multi user network (Yuce *et al., 2007).*

2.6.2 **Sensor Nodes and CCU Hardware Designs**

Yuce et al (2007) designed sensor nodes which collect raw signals from a human body. The signal from a human body is usually weak and coupled with noise. The signal go through amplification and filtering process to increase the signal strength, and to remove unwanted signals and noise (Yuce *et al., 2007).* After which, it will go through an analog to digital conversion (ADC) stage to be converted into digital for digital processing. The digitized signal is then processed and stored in the microprocessor. The microprocessor will then pack those data and transmit over the air via a transmitter. The overview of a Pulse Rate Sensor Node is shown in Fig. 2.8 (Yuce *et al., 2007).* The pulse rate sensor node comprises of Microcontroller PIC16F877 and the transceiver AMIS-5210 are selected in the project because of the following reasons: low-power consumption, size, and the suitability operating at the MICS band and for the physiological data processing (Yuce *et al.*, 2007). Fig. 2.8 shows the hardware implementation of sensor nodes. Both temperature and pulse rate sensor nodes were built on a common PCB circuit. Thus the electronics can be used interchangeable. The antennas for this project are designed as a loop printed around the prototyping boards. The CCU also requires a micro-controller and a wireless transceiver chip to coordinate all activities similar to the sensor nodes. The CCU hardware is made of the same transceiver chip from AMI semiconductor (AMI52100 IC) and the microcontroller PIC16F87 (Yuce *et al., 2007)*. The targeted wireless distance between sensors and the CCU (the MICS link) is 1-10 meters. The CCU can thus be located at the waist of the patient or at an easily accessible place.

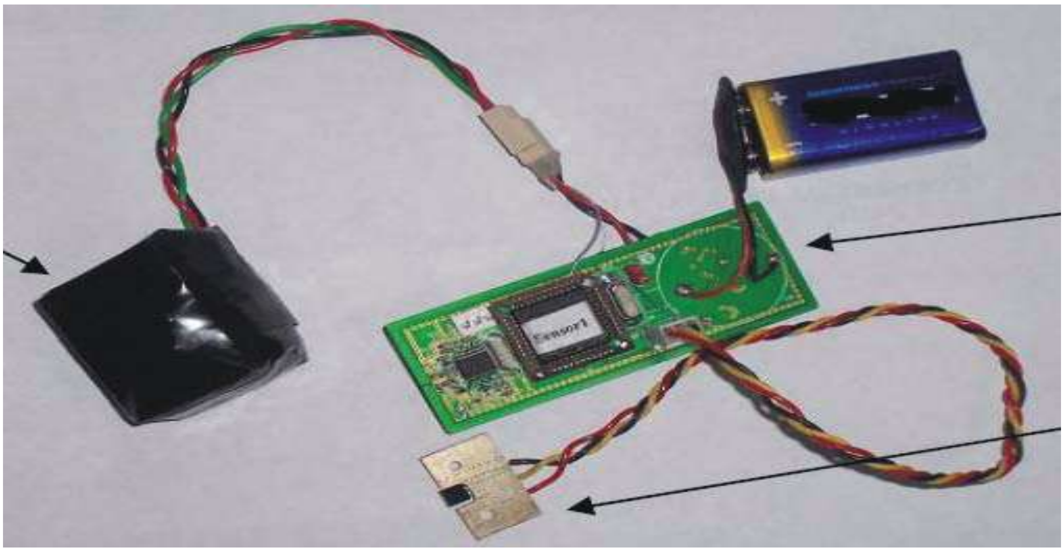


Figure 2.8: Hardware design of sensor nodes.
Source: http://www.wirelessis.com , April 2007.

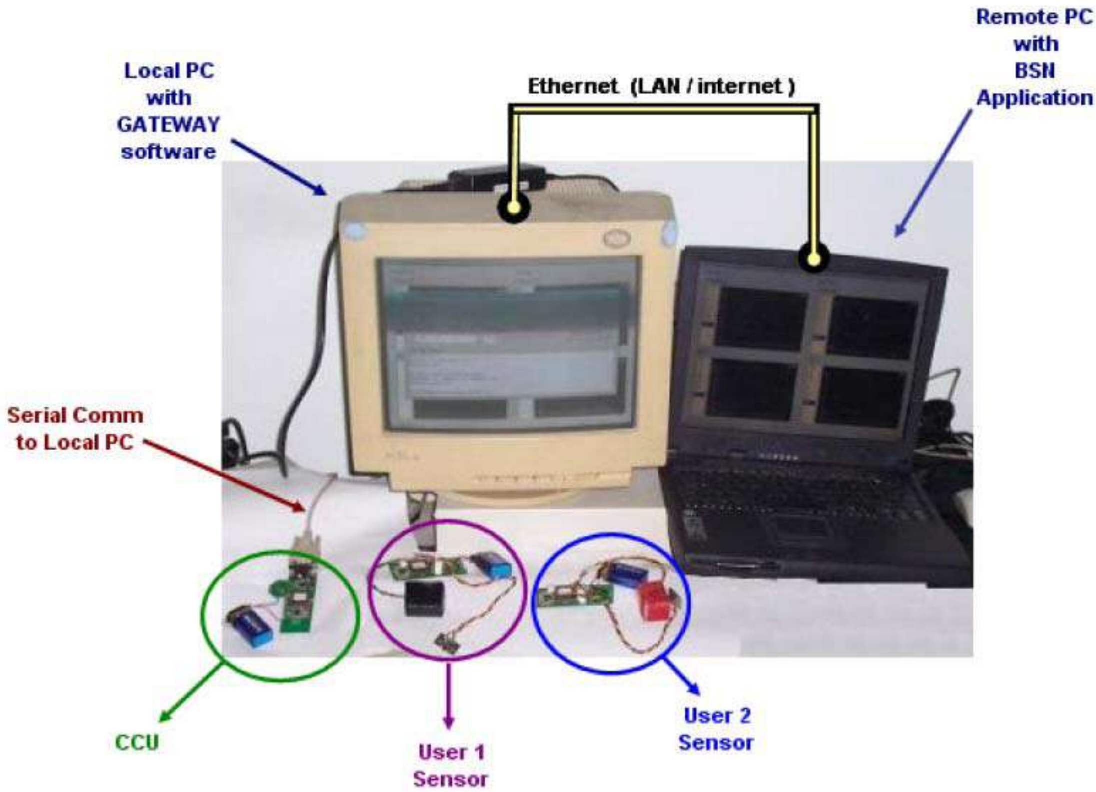


Figure 2.9**:** Set up of the multi-patient body sensor network system.
Source: http://www.wirelessis.com , April 2007.

2.6.3 **Data Base, Software Programs and Monitoring**

According to Mehmet et al (2007), since all sensor nodes of a body communicate with the same CCU, the data is prefixed with an identifier that is used to identify source of data. To reduce collisions further between data sent, a firmware (media access layer (MAC) protocol) is written to control data transmissions. The communication between the sensors and the CCU is bidirectional as to support a multi user (i.e. multi-patient) communications to illustrate a multi user implementation (Mehmet et al., 2007).

Mehmet et al (2007) developed a media access layer (MAC) protocol for firmware at both sensor node and the CCU to prove the bidirectional communication. The system was designed using the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) MAC protocol to accommodate multi-patient medical data transfer.

Mehmet et al (2007) also developed a software (defined as GATEWAY) at the local PC to communicate with the CCU to get readings from sensors and then forward them through the network/internet to an application on a remote PC (at medical center). While performing this task, the GATEWAY also verifies the data integrity and schedules retransmission if required. Another software program is developed at the remote PC (called BSN) which gets readings from GATEWAY via network/internet. These readings are stored in the remote PC for analysis. The program is also able to display readings for multiple users.

2.6.4 Performance Evaluation

Mehmet et al (2007) conducted a test to evaluate the bit-error rate (BER) performance of the wireless transmission with the distance. The BER indicates the reliability of a channel. For the test, 625000 bytes (5X106 bits) of data were sent continuously from a sensor node to the CCU. If any bit of the received data is in error, the whole 8 bits in the byte were considered to be in error (Mehmet et al.,2007).

2.6.5 **Findings**

Mehmet et al. (2007) developed a wireless sensor network system for monitoring physiological parameters from multiple patient bodies. A prototype system that is able to acquire readings from multiple patients was presented by Mehmet et al. (2007). It has been shown that after obtaining data from sensors, the data can be transferred to a remote PC through a local area network or the internet for further analysis. Such a wireless body sensor network system is very suitable to be used in hospital environments. Human errors will be reduced and health professionals will spend their time more on other important issues. In addition, such systems result in an increase in patients' comfort level as they no longer need to be wakening up for periodic checks in the hospital environments. Another advantage is that patients can stay in their home while their conditions can still be monitored by medical staff. More wards could thus be available to patients in medical centers.

Mehmet et al. (2007) suggests future work which will involve including more sensors to monitor other vital signs such as blood pressure, oxygen saturation level, ECG, etc. The software at the local P should also be able to evaluate the physiological parameters before being transmitted to the medical centre. They also propose the possibility of extending the capability to automate alerts which can get attention of the health professionals as soon as they are needed.

2.7 Wireless Healthcare Monitoring System for Home

Populations of industrialized nations of the world are ageing at a rapid rate. This has a profound implication on the healthcare systems of these nations. A large number of the elderly citizens will be spending their lives in the confines of their home, suffering from chronic illnesses that need continuous monitoring (Hui et al., 2007).

An effective system for home-based monitoring of the health status of such patients will go a long way in relieving the load on the healthcare system, by reducing the number of outpatient visits to hospitals. It will also avoid a great deal of trouble for the elderly patients and their caregivers.

Hui et al. (2007) presented a paper that described the system design of the wireless electrocardiogram (WECG) system and focused more on the personal server hardware and software design.

2.7.1 Overview of WECG System

Hui et al. (2007) proposed a wireless health monitoring system for home usage which is shown in fig. 2.10. Wireless sensor nodes collect vital health parameters from the body of the patient.

The data from the sensor nodes is sent to a personal server – which is worn by/kept close to the patient. The data from the personal server can be examined by a doctor for further diagnosis. The data can also be sent to a home server from where it can be transferred to a healthcare network through a public telephone network or through the Internet. The scheme of this wireless healthcare monitoring system is illustrated in Fig. 210.

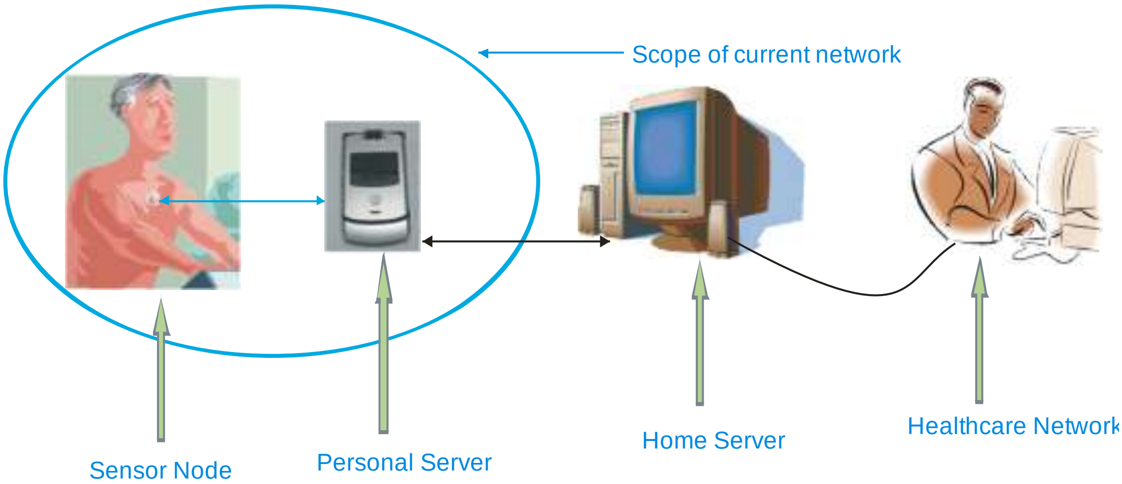


Figure 2.10: Scheme of the wireless healthcare monitoring system

Source: World Academy of Science, Engineering and Technology Vol. 42 2008

The technical approach by Hui et al. (2007) chosen to implement the health monitoring system is as follows:

i. Design and Develop Analog, Digital & Radio Frequency Integrated Circuits, optimized for ultralow power applications

ii. Use these key components to develop ultra-low power Wireless Sensor nodes (WSN) which are used to collect vital body parameters from a patient's body.

iii. Data from the WSN will be transmitted to a Personal Server (PDA with hardware & software extensions). This will be a shortrange link (< 5 m), which facilitates conserving power in the WSN.

iv. The data from the Personal Server can be transferred to the external world through conventional links, for further analysis.

The WECG system consisted of

1. A micro-power wireless sensor node to acquire the bio-signal from biosensors, encode and modulate it via ultra low-power ASK transmitter.
2. A personal server to acquire the wireless bio-signal streamed out from the sensor node.
3. GUI software to view and log the acquired bio-signal.

2.7.2 Personal Server

The personal server was built around a Window Mobile Smartphone and a Secure Digital Input Output (SDIO) wireless receiver module. The wireless module was connected to the Smartphone via microSD slot that is commonly found on today's mobile phones.

i. Receiver Module

The receiver module provided the necessary control to select the desired Amplitude Shift Keying (ASK) frequency, convert the received Manchester encoded signal into valid data, and transmit the data and Received Signal Strength Indicator (RSSI) signal to the Smartphone software via SDIO interface (William, 1984).

The front end of the receiver module consisted of a double superhet ASK receiver, Melexis MLX71122, that could support data rate up to 100 kbps and cover frequency ranges 300 to 930 MHz with channel raster as low as 10 kHz. Its RF dynamic range ranges -10 dBm and -112 dBm. Despite of this wide flexibility, in their application they fixed the channel of the receiver at the 433 MHz-ISM band and configured its data filter to accept channel data rate of 9600 bps.

The configuration of the receiver front-end was Serial Peripheral Interface (SPI) programmable by the Microcontroller Unit (MCU), MicroChip PIC18F25K20. The RSSI was sampled through the MCU's built-in Analog-to-Digital Converter (ADC), and the received data (DTAO) was passed as digital input. The MCU communicate with the mobile software residing on the Smartphone through the Universal Asynchronous Receiver/Transmitter (UART) to SDIO controller, the Arasan AC2600 chip. The Electrical Erasable Programmable Read-Only Memory (EEPROM) provides the configuration and initialization settings for the UART to SDIO controller.

The firmware running on the MCU is also responsible for the bit and frame synchronization of the received data stream, and handles exceptions like parity error and synchronization error. A resynchronization algorithm has also been in cooperated in the firmware to improve the reliability of the wireless link. After proper synchronization, the original biodata is then decoded and extracted out from the data stream and supplied the sampled bio-data to the Smartphone via SDIO interface for display and logging. To ease the design of bit and frame synchronization of the receiver, the transmission framing is actually following the concept of UART framing, with predefined length of start and stop bits. The transmitted data stream is Manchester encoded to ensure equal transitions of '1's and '0's (William, 1984).

2.7.3 Smartphone Software

The purpose of the Smartphone software is to provide data logging and viewing, and to relay the acquired data to remote server via mobile network.

The features include start / stop / pause / resume acquisition ECG signals, requesting remote device identity (ID), requesting current state, requesting current baud rate and setting baud rate.

The functions of the software include: wrapping the user-initiated communication protocol, reading/writing through the RS232 serial port, handling the received data packet from the receiver module, suppressing the 50 Hz interference, which human body is prone to couple it from near-by domestic power lines, by using a digital notch filter analyzing the signals to extract the critical data processing to display the finalized received signals monitoring and controlling the receiver module (Hui et al., 2007).

2.7.4 Development and Results

Hui et al, (2007) developed and tested the prototype of the personal server which includes the receiver module and Smartphone software.

The receiver module consumes approximately 25 mA during active mode and consumes less than 1 mA during stand-by mode. This current is about 50% of a typical Bluetooth handset consumption. The consumption of the micro-sensor node is estimated to be around 3 mA. A supply voltage of 3.3 V was used during the testing of receiver module.

They proposed further work to link up the proposed system with a health care network through the Internet.

2.8 Internet of things in Remote Medical Monitoring Systems

The Internet of Things (IoT) offers great promise in the field of healthcare where its principles are being applied to improve access to care, increase the quality of care and most importantly reduce the cost of care.

As the technology for collecting, analyzing and transmitting data in the IoT continues to mature, more and more exciting new IoT driven healthcare applications and systems continue to emerge.

Wireless sensor-based systems are at work today, gathering patient medical data that was never before available for analysis and delivering care to people for whom care wasn’t previously accessible. In these ways, IoT-driven systems are making it possible to radically reduce costs and improve health by increasing the availability and quality of care. The ability of devices to gather data on their own removes the limitations of human-entered data automatically obtaining the data doctors’ need, at the time and in the way they need it. The automation reduces the risk of errors. Fewer errors can mean increased efficiency, lower costs and improvements in quality in just about any industry. But it's of particular interest/need in healthcare, where human error can literally be the difference between life and death. The IoT plays a significant role in a broad range of healthcare applications, from managing chronic diseases at one end of the spectrum to preventing disease at the other. Here are some examples of how its potential is already playing out:

i. Clinical care: Hospitalized patients whose physiological status requires close attention can be constantly monitored using IoT-driven, noninvasive monitoring. This type of solution employs sensors to collect comprehensive physiological information and uses gateways and the cloud to analyze and store the information and then send the analyzed data wirelessly to caregivers for further analysis and review. It reduces the process of having a health professional come by at regular intervals to check the patient's vital signs, instead providing a continuous automated flow of information. In this way, it simultaneously improves the quality of care through constant attention and lowers the cost of care by eliminating the need for a caregiver to actively engage in data collection and analysis.
ii. Remote monitoring: There are people all over the world whose health may suffer because they don't have ready access to effective health monitoring. But small powerful wireless solutions connected through the IoT are now making it possible for monitoring to come to these patients instead and vice versa. These solutions can be used to securely capture patient health data from a variety of sensors, apply complex algorithms to analyze the data and then share it through wireless connectivity with medical professionals who can make appropriate health recommendations.

2.9 Blood pressure

Blood pressure (BP) is the pressure exerted by circulating blood upon the walls of blood vessels, and is one of the principal vital signs. Blood pressure usually refers to the arterial pressure of the systemic circulation. During each heartbeat, BP varies between a maximum (systolic) and a minimum (diastolic) pressure (Klabunde, 2005). The mean BP, due to pumping by the heart and resistance to flow in blood vessels, decreases as the circulating blood moves away from the heart through arteries. Blood pressure drops most rapidly along the small arteries and arterioles, and continues to decrease as the blood moves through the capillaries and back to the heart through veins (Klabunde, 2005). Gravity, valves in veins, and pumping from contraction of skeletal muscles are some other influences on BP at various places in the body.

The measurement blood pressure refers to the systemic arterial pressure measured at a person's upper arm. It is measured on the inside of an elbow at the brachial artery, which is the upper arm's major blood vessel that carries blood away from the heart. A person's BP is usually expressed in terms of the systolic pressure over diastolic pressure (mmHg), for example 140/90.

i. Normal

While average values for arterial pressure could be computed for any given population, there is often a large variation from person to person; arterial pressure also varies in individuals from moment to moment. Additionally, the average of any given population may have a questionable correlation with its general health; thus the relevance of such average values is equally questionable. However, in a study of 100 human subjects with no known history of hypertension, an average blood pressure of 112/64 mmHg was found, (Eguchi et al., 2007) which are the normal values.

Various factors, such as age and gender influence average values, influence a person's average BP and variations. In children, the normal ranges are lower than for adults and depend on height (Eguchi et al., 2007). As adults age, systolic pressure tends to rise and diastolic tends to fall (Eguchi et al., 2007). In the elderly, BP tends to be above the normal adult range, (Eguchi et al., 2007) largely because of reduced flexibility of the arteries. Also, an individual's BP varies with exercise, emotional reactions, sleep, digestion and time of day.

Differences between left and right arm BP measurements tend to be random and average to nearly zero if enough measurements are taken. However, in a small percentage of cases there is a consistent difference greater than 10 mmHg which may need further investigation, e.g. for obstructive arterial disease (Agarwal et al., 2008) (Appel et al., 2006). The risk of cardiovascular disease increases progressively above 115/75 mmHg (Appel et al., 2006). In the past, hypertension was only diagnosed if secondary signs of high arterial pressure were present, along with a prolonged high systolic pressure reading over several visits. Regarding hypotension, in practice blood pressure is considered too low only if noticeable symptoms are present (Rosenson et al., 2004).

Clinical trials demonstrate that people who maintain arterial pressures at the low end of these pressure ranges have much better long term cardiovascular health. The principal medical debate concerns the aggressiveness and relative value of methods used to lower pressures into this range for those who do not maintain such pressure on their own. Elevations, more commonly seen in older people, though often considered normal, are associated with increased morbidity and mortality.

Average blood pressure in (mmHg):

Table 2.3: Average blood pressure

1 year	**6-9years**	**Adults**
95/65	100/65	110/65 – 140/90

2.9.1 Physiology

There are many physical factors that influence arterial pressure. Each of these may in turn be influenced by physiological factors, such as diet, exercise, disease, drugs or alcohol, stress, obesity, and so-forth (Klabunde, 2007).

Some physical factors are:

i. Rate of pumping. In the circulatory system, this rate is called heart rate, the rate at which blood (the fluid) is pumped by the heart. The volume of blood flow from the heart is called the cardiac output which is the heart rate (the rate of contraction) multiplied by the stroke volume (the amount of blood pumped out from the heart with each contraction). The higher the heart rate, the higher the mean arterial pressure, assuming no reduction in stroke volume or central venous return (Rosenson et al., 2004).

ii. Volume of fluid or blood volume, the amount of blood that is present in the body. The more blood present in the body, the higher the rate of blood return to the heart and the resulting cardiac output. There is some relationship between dietary salt intake and increased blood volume, potentially resulting in higher arterial pressure, though this varies with the individual and is highly dependent on autonomic nervous system response and the renin-angiotensin system (Klabunde, 2007).

iii. Resistance. In the circulatory system, this is the resistance of the blood vessels. The higher the resistance, the higher the arterial pressure upstream from the resistance to blood flow. Resistance is related to vessel radius (the larger the radius, the lower the resistance), vessel length (the longer the vessel, the higher the resistance), blood viscosity, as well as the smoothness of the blood vessel walls. Smoothness is reduced by the build up of fatty deposits on the arterial walls. Substances called vasoconstrictors can reduce the size of blood vessels, thereby increasing BP. Vasodilators (such as nitroglycerin) increase the size of blood vessels, thereby decreasing arterial pressure. Resistance, and its relation to volumetric flow rate (Q) and pressure difference between the two ends of a vessel are described by Poiseuille's Law (Rosenson et al., 2004).

iv. Viscosity, or thickness of the fluid. If the blood gets thicker, the result is an increase in arterial pressure. Certain medical conditions can change the viscosity of the blood. For instance, anemia (low red blood cell concentration), reduces viscosity, whereas increased red blood cell concentration increases viscosity. It had been thought that aspirin and related "blood thinner" drugs decreased the viscosity of blood, but instead studies found (Klabunde, 2007) that they act by reducing the tendency of the blood to clot.

In practice, each individual's autonomic nervous system responds to and regulates all these interacting factors so that, although the above issues are important, the actual arterial pressure response of a given individual varies widely because of both split-second and slow-moving responses of the nervous system and end organs. These responses are very effective in changing the variables and resulting BP from moment to moment.

Moreover, blood pressure is the result of cardiac output increased by peripheral resistance: *blood pressure = cardiac output X peripheral resistance*. As a result, an abnormal change in blood pressure is often an indication of a problem affecting the heart's output, the blood vessels' resistance, or both. Thus, knowing the patient's blood pressure is critical to assess any pathology related to output and resistance (Klabunde, 2007).

2.9.2 Arm–leg gradient

The *arm–leg (blood pressure) gradient* is the difference between the blood pressure measured in the arms and that measured in the legs. It is normally less than 10 mmHg, (O'Rourke, 1995) but may be increased in e.g. coarctation of the aorta

2.9.3 Vascular resistance

The larger arteries, including all large enough to see without magnification, are conduits with low vascular resistance (assuming no advanced atherosclerotic changes) with high flow rates that generate only small drops in pressure. The smaller arteries and arterioles have higher resistance, and confer the main drop in blood pressure along the circulatory system (O'Rourke, 1995).

2.9.4 Vascular pressure wave

Modern physiology developed the concept of the vascular pressure wave (VPW). This wave is created by the heart during the systole and originates in the ascending aorta. Much faster than the stream of blood itself, it is then transported through the vessel walls to the peripheral arteries. There the pressure wave can be palpated as the peripheral pulse. As the wave is reflected at the peripheral veins, it runs back in a centripetal fashion. Where the crests of the reflected and the original wave meet, the pressure inside the vessel is higher than the true pressure in the aorta. This concept explains why the arterial pressure inside the peripheral arteries of the legs and arms is higher than the arterial pressure in the aorta, (Mitchell, 2006), (Klabunde, 2007) (Booth,1977), and in turn for the higher pressures seen at the ankle compared to the arm with normal ankle brachial pressure index values.

2.9.5 Regulation

The endogenous regulation of arterial pressure is not completely understood, but the following mechanisms of regulating arterial pressure have been well-characterized:

i. Baroreceptor reflex: Baroreceptors in the high pressure receptor zones (mainly in the aortic arch and carotid sinus) detect changes in arterial pressure. These baroreceptors send signals ultimately to the medulla of the brain stem, specifically to the Rostral ventrolateral medulla (RVLM). The medulla, by way of the autonomic nervous system, adjusts the mean arterial pressure by altering both the force and speed of the heart's contractions, as well as the total peripheral resistance. The most important arterial baroreceptors are located in the left and right carotid sinuses and in the aortic arch (Deakin & Low, 2000).
ii. Renin-angiotensin system (RAS): This system is generally known for its long-term adjustment of arterial pressure. This system allows the kidney to compensate for loss in blood volume or drops in arterial pressure by activating an endogenous vasoconstrictor known as angiotensin II.

iii. Aldosterone release: This steroid hormone is released from the adrenal cortex in response to angiotensin II or high serum potassium levels. Aldosterone stimulates sodium retention and potassium excretion by the kidneys. Since sodium is the main ion that determines the amount of fluid in the blood vessels by osmosis, aldosterone will increase fluid retention, and indirectly, arterial pressure (Deakin & Low, 2000).

iv. Baroreceptors in low pressure receptor zones (mainly in the venae cavae and the pulmonary veins, and in the atria) result in feedback by regulating the secretion of antidiuretic hormone (ADH/Vasopressin), renin and aldosterone. The resultant increase in blood volume results an increased cardiac output by the Frank–Starling law of the heart, in turn increasing arterial blood pressure.

These different mechanisms are not necessarily independent of each other, as indicated by the link between the RAS and aldosterone release. Currently, the RAS is targeted pharmacologically by ACE inhibitors and angiotensin II receptor antagonists. The aldosterone system is directly targeted by spironolactone, an aldosterone antagonist. The fluid retention may be targeted by diuretics; the antihypertensive effect of diuretics is due to its effect on blood volume. Generally, the baroreceptor reflex is not targeted in hypertension because if blocked, individuals may suffer from orthostatic hypotension and fainting (Deakin & Low, 2000).

2.9.6 Measurement

Arterial pressure is most commonly measured via a sphygmomanometer, which historically used the height of a column of mercury to reflect the circulating pressure (Deakin & Low, 2000). BP values are generally reported in millimetres of mercury (mmHg), though aneroid and electronic devices do not use mercury.

For each heartbeat, BP varies between systolic and diastolic pressures. Systolic pressure is peak pressure in the arteries, which occurs near the end of the cardiac cycle when the ventricles are contracting. Diastolic pressure is minimum pressure in the arteries, which occurs near the beginning of the cardiac cycle when the ventricles are filled with blood. An example of normal measured values for a resting, healthy adult human is 120 mmHg systolic and 80 mmHg diastolic (written as 120/80 mmHg, and spoken [in the US and UK] as "one-twenty over eighty").

Systolic and diastolic arterial BPs are not static but undergo natural variations from one heartbeat to another and throughout the day (in a circadian rhythm). They also change in response to stress, nutritional factors, drugs, disease, exercise, and momentarily from standing up. Sometimes the variations are large. Hypertension refers to arterial pressure being abnormally high, as opposed to hypotension, when it is abnormally low. Along with body temperature, respiratory rate, and pulse rate, BP is one of the four main vital signs routinely monitored by medical professionals and healthcare providers (Elliot, 2011). Measuring pressure invasively, by penetrating the arterial wall to take the measurement, is much less common and usually restricted to a hospital setting.

2.9.7 Noninvasive

The noninvasive auscultatory and oscillometric measurements are simpler and quicker than invasive measurements, require less expertise, have virtually no complications, are less unpleasant and less painful for the patient. However, noninvasive methods may yield somewhat lower accuracy and small systematic differences in numerical results. Noninvasive measurement methods are more commonly used for routine examinations and monitoring.

2.9.8 Palpation

A minimum systolic value can be roughly estimated by palpation, most often used in emergency situations, but should be used with caution (Jhalani et al., 2005). It has been estimated that, using 50% percentiles, carotid, femoral and radial pulses are present in patients with a systolic blood pressure > 70 mmHg, carotid and femoral pulses alone in patients with systolic blood pressure of > 50 mmHg, and only a carotid pulse in patients with a systolic blood pressure of > 40 mmHg (Jhalani et al., 2005). A more accurate value of systolic BP can be obtained with a sphygmomanometer and palpating the radial pulse. Diastolic blood pressure cannot be estimated by this method. The American Heart Association recommends that palpation be used to get an estimate before using the auscultatory method.

The auscultatory method (from the Latin word for "listening") uses a stethoscope and a sphygmomanometer. This comprises an inflatable (*Riva-Rocci*) cuff placed around the upper arm at roughly the same vertical height as the heart, attached to a mercury or aneroid manometer. The mercury manometer, considered the gold standard, measures the height of a column of mercury, giving an absolute result without need for calibration and, consequently, not subject to the errors and drift of calibration which affect other methods. The use of mercury manometers is often required in clinical trials and for the clinical measurement of hypertension in high-risk patients, such as pregnant women.

A cuff of appropriate size is fitted smoothly and snugly, then inflated manually by repeatedly squeezing a rubber bulb until the artery is completely occluded. Listening with the stethoscope to the brachial artery at the elbow, the examiner slowly releases the pressure in the cuff. When blood just starts to flow in the artery, the turbulent flow creates a "whooshing" or pounding (first Korotkoff sound). The pressure at which this sound is first heard is the systolic BP. The cuff pressure is further released until no sound can be heard (fifth Korotkoff sound), at the diastolic arterial pressure.

The auscultatory method is the predominant method of clinical measurement.

2.9.9 Oscillometric

The oscillometric method was first demonstrated in 1876 and involves the observation of oscillations in the sphygmomanometer cuff pressure which are caused by the oscillations of blood flow, i.e., the pulse (Jhalani et al, 2005). The electronic version of this method is sometimes used in long-term measurements and general practice. It uses a sphygmomanometer cuff, like the auscultatory method, but with an electronic pressure sensor (transducer) to observe cuff pressure oscillations, electronics to automatically interpret them, and automatic inflation and deflation of the cuff. The pressure sensor should be calibrated periodically to maintain accuracy.

Oscillometric measurement requires less skill than the auscultatory technique and may be suitable for use by untrained staff and for automated patient home monitoring.

The cuff is inflated to a pressure initially in excess of the systolic arterial pressure and then reduced to below diastolic pressure over a period of about 30 seconds. When blood flow is nil (cuff pressure exceeding systolic pressure) or unimpeded (cuff pressure below diastolic pressure), cuff pressure will be essentially constant. It is essential that the cuff size is correct: undersized cuffs may yield too high a pressure; oversized cuffs yield too low a pressure. When blood flow is present, but restricted, the cuff pressure, which is monitored by the pressure sensor, will vary periodically in synchrony with the cyclic expansion and contraction of the brachial artery, i.e., it will oscillate. The values of systolic and diastolic pressure are computed, not actually measured from the raw data, using an algorithm; the computed results are displayed.

Oscillometric monitors may produce inaccurate readings in patients with heart and circulation problems, which include arterial sclerosis, arrhythmia, preeclampsia, pulsus alternans, and pulsus paradoxus.

In practice the different methods do not give identical results; an algorithm and experimentally obtained coefficients are used to adjust the oscillometric results to give readings which match the auscultatory results as well as possible. Some equipment uses computer-aided analysis of the instantaneous arterial pressure waveform to determine the systolic, mean, and diastolic points. Since many oscillometric devices have not been validated, caution must be given as most are not suitable in clinical and acute care settings.

The term NIBP, for non-invasive blood pressure, is often used to describe oscillometric monitoring equipment.

2.9.10 White-coat hypertension

For some patients, BP measurements taken in a doctor's office may not correctly characterize their typical BP (Pickering et al., 2005). In up to 25% of patients, the office measurement is higher than their typical BP. This type of error is called white-coat hypertension (WCH) and can result from anxiety related to an examination by a health care professional (Pickering et al., 2005). The misdiagnosis of hypertension for these patients can result in needless and possibly harmful medication. WCH can be reduced (but not eliminated) with automated BP measurements over 15 to 20 minutes in a quiet part of the office or clinic (Mancia et al., 2007).

Debate continues regarding the significance of this effect. Some reactive patients will react to many other stimuli throughout their daily lives and require treatment. In some cases a lower BP reading occurs at the doctor's office (Niiranen et al., 2006).

2.9.11 Home monitoring

Ambulatory blood pressure devices that take readings every half hour throughout the day and night have been used for identifying and mitigating measurement problems like white-coat hypertension. Except for sleep, home monitoring could be used for these purposes instead of ambulatory blood pressure monitoring (Shimbo, 2007). Home monitoring may be used to improve hypertension management and to monitor the effects of lifestyle changes and medication related to blood pressure (Shimbo, 2007). Compared to ambulatory blood pressure measurements, home monitoring has been found to be an effective and lower cost alternative, (Kate et al., 2011; Shimbo, 2007; Struijk et al., 2008) but ambulatory monitoring is more accurate than both clinic and home monitoring in diagnosing hypertension. Ambulatory monitoring is recommended for most patients before the start of antihypertensive drugs (Struijk et al., 2008).

Aside from the white-coat effect, BP readings outside of a clinical setting are usually slightly lower in the majority of people. The studies that looked into the risks from hypertension and the benefits of lowering BP in affected patients were based on readings in a clinical environment.

When measuring BP, an accurate reading requires that one not drink coffee, smoke cigarettes, or engage in strenuous exercise for 30 minutes before taking the reading. A full bladder may have a small effect on BP readings; if the urge to urinate exists, one should do so before the reading. For 5 minutes before the reading, one should sit upright in a chair with one's feet flat on the floor and with limbs uncrossed. The BP cuff should always be against bare skin, as readings taken over a shirt sleeve are less accurate. During the reading, the arm that is used should be relaxed and kept at heart level, for example by resting it on a table (Sharon & Emily, 2006). Since BP varies throughout the day, measurements intended to monitor changes over longer time frames should be taken at the same time of day to ensure that the readings are comparable. Suitable times are:

i. immediately after awakening (before washing/dressing and taking breakfast/drink), while the body is still resting,
ii. immediately after finishing work.

Automatic self-contained BP monitors are available at reasonable prices, some of which are capable of Korotkoff's measurement in addition to oscillometric methods, enabling irregular heartbeat patients to accurately measure their blood pressure at home.

2.10 Pulse Rate

In medicine, one's pulse represents the tactile arterial palpation of the heartbeat by trained fingertips. The pulse may be palpated in any place that allows an artery to be compressed against a bone, such as at the neck (carotid artery), at the wrist (radial artery), behind the knee (popliteal artery), on the inside of the elbow (brachial artery), and near the ankle joint (posterior tibial artery). The pulse can also be measured by listening to the heart beat directly (auscultation), traditionally using a stethoscope (Robergs & Landwehr, 2002).

2.10.1 Physiology

The pulse is a decidedly low tech/high yield and antiquated term still useful at the bedside in an age of computational analysis of cardiac performance. Claudius Galen was perhaps the first physiologist to describe the pulse (Gellish et al., 2007). The pulse is an expedient tactile method of determination of systolic blood pressure to a trained observer. Diastolic blood pressure is non-palpable and unobservable by tactile methods, occurring between heartbeats (Gellish et al., 2007)

Pressure waves generated by the heart in systole moves the arterial walls. Forward movement of blood occurs when the boundaries are pliable and compliant. These properties form enough to create a palpable pressure wave (Wohlfart & Farazdaghi, 2003).

The heart rate may be greater or lesser than the pulse rate depending upon physiologic demand. In this case, the heart rate is determined by auscultation or audible sounds at the heart apex, in which case it is not the pulse. The *pulse deficit* (difference between heart beats and pulsations at the periphery) is determined by simultaneous palpation at the radial artery and auscultation at the heart apex (Farazdaghi & Wohlfart, 2001).

Pulse velocity, pulse deficits and much more physiologic data are readily and simplistically visualized by the use of one or more arterial catheters connected to a transducer and oscilloscope. This invasive technique has been commonly used in intensive care since the 1970s.

The rate of the pulse is observed and measured by tactile or visual means on the outside of an artery and is recorded as beats per minute or BPM (Karvonen et al., 1998).

The pulse may be further indirectly observed under light absorbances of varying wavelengths with assigned and inexpensively reproduced mathematical ratios. Applied capture of variances of light signal from the blood component hemoglobin under oxygenated vs. deoxygenated conditions allows the technology of pulse oximetry.

2.10.2 Normal Pulse Rates

Normal pulse rates at rest, in beats per minute (BPM:

Table 2.4:Normal Pulse Rates
Source: Physiol Vol. 84, Issue 6, 2042-2051, June 1998

New Born	1 -12 months	1 -2 years	2 – 6 years	7 – 12 years	13 years – adult	Adult athletes
120 – 160	80 – 140	80 – 130	75 – 120	75 – 110	60 – 100	40 – 60

The pulse rate can be used to check overall heart health and fitness level. Generally lower is better, but bradycardias can be dangerous. Symptoms of a dangerously slow heartbeat include weakness, loss of energy and fainting.

2.10.3 Evaluation

Several pulse patterns can be of clinical significance. These include:

i. Pulsus alternans: an ominous medical sign that indicates progressive systolic heart failure. To trained fingertips, the examiner notes a pattern of a strong pulse followed by a weak pulse over and over again. This pulse signals a flagging effort of the heart to sustain itself in systole.
ii. Pulsus bigeminus: indicates a pair of hoofbeats within each heartbeat. Concurrent auscultation of the heart may reveal a gallop rhythm of the native heartbeat.
iii. Pulsus bisferiens: an unusual physical finding typically seen in patients with aortic valve diseases. If the aortic valve does not normally open and close, trained fingertips will observe two pulses to each heartbeat instead of one.
iv. Pulsus tardus et parvus: a slower than normal rise in the tactile pulse caused by an increasingly stiff aortic valve. Loss of compliance in the aortic valve makes it progressively harder to open, thus requiring increased generation of blood pressure in the left ventricle.

v. Pulsus paradoxus: a condition in which some heartbeats cannot be detected at the radial artery during the inspiration phase of respiration. It is caused by an exaggerated decrease in blood pressure during this phase, and is diagnostic of a variety of cardiac and respiratory conditions of varying urgency.
vi. Tachycardia: an elevated resting heart rate. In general an electrocardiogram (ECG) is required to identify the type of tachycardia.

The strength of the pulse can also be reported:

i. 0 = Absent
ii. 1 = Barely palpable
iii. 2 = Easily palpable
iv. 3 = Full
v. 4 = Aneurysmal or Bounding pulse

2.10.4 Mobile Devices

A mobile device (also known as cellphone device, handheld device, handheld computer, "Palmtop" or simply handheld) is a pocket-sized computing device, typically having a display screen with touch input or a miniature keyboard (Chet Hosmer et al., 2011). In the case of the personal digital assistant (PDA) the input and output are combined into a touch-screen interface. Smartphones and PDAs are popular amongst those who require the assistance and convenience of a conventional computer, in environments where carrying one would not be practical. Enterprise digital assistants can further extend the available functionality for the business user by offering integrated data capture devices like barcode, RFID and smart card readers (Lotz, 2007).

Types of mobile devices

Mobile devices have been designed for many applications and include: (Lotz, 2007).

i. Mobile computers

ii. Personal digital assistant/enterprise digital assistant
iii. Graphing calculator
iv. Handheld game consoles
v. Digital still camera (DSC)
vi. Digital video camera (DVC or digital camcorder)
vii. Portable media player
viii. e-book reader
ix. Mobile phone
x. Pager
xi. Personal navigation devices (PNDs)

Usage

Handheld devices have become ruggedized for use in mobile field management situations to record information in the field. They are used to achieve a variety of tasks for increasing efficiency that include digitizing notes, sending and receiving invoices, asset management, recording signatures, managing parts and scanning barcodes (Lotz, 2007).

Handheld computers used at work have molded over time into a variety of form factors, including smartphones on the low end, handheld PDAs, Ultra Mobile PCs, tablet PCs, and even notebook computers.

2.11 Web Service

Web services are typically application programming interfaces (API) or web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services (Benslimane et al., 2008).

In common usage the term refers to clients and servers that communicate over the Hypertext Transfer Protocol (HTTP) protocol used on the web. Such services tend to fall into one of two camps: Big Web Services and RESTful Web Services.

"Big Web Services" use Extensible Markup Language (XML) messages that follow the Simple Object Access Protocol (SOAP) standard and have been popular with traditional enterprise. In such systems, there is often a machine-readable description of the operations offered by the service written in the Web Services Description Language (WSDL). The latter is not a requirement of a SOAP *endpoint*, but it is a prerequisite for automated client-side code generation in many Java and .NET SOAP frameworks (frameworks such as Spring, Apache Axis2 and Apache CXF being notable exceptions). Some industry organizations, such as the WS-I, mandate both SOAP and WSDL in their definition of a web service.

2.11.1 Wireless/Remote Health Monitoring Devices

i. **Lifelink**

A mobile real-time telemonitoring and diagnostic facility to command and control remote medical devices through mobile phones. The whole process is phone-based (Obrenovic et al, 2002).

LifeLink Monitoring offers a powerful suite of telemonitoring services to combat congestive heart failure, diabetes and hypertension. It provides immediate feedback to patients and accurate, objective data to clinicians.

LifeLink telemonitoring services enable patients to monitor blood pressure and heart rate, weight, blood glucose and haemoglobin at home, and to send objective data to clinicians over the telephone. Patients also report their subjective symptoms and medication compliance (Obrenovic et al, 2002).

ii. Corventis

Corventis develops wireless cardiovascular solutions that offer visibility into a patient's health status

PiiX™: An unobtrusive, water-resistant, patient-worn device that adheres to the skin and automatically collects and transmits physiological information zLink™: A small portable device (similar to a cell phone) that wirelessly transmits information received from PiiX to Corventis (Riedel, 2003).

Corventis Web Services: A hosted application for data analysis and storage that also includes the Corventis website, which enables secure access to physiological trends and clinical event information for interpretation and diagnosis.

iii. Cardionet

CardioNet is the world's leading supplier of mobile cardiac outpatient telemetry. CardioNet provides the next-generation ambulatory cardiac monitoring service with beat-to-beat, real time analysis, automatic arrhythmia detection and wireless ECG transmission (Riedel, 2003).

CardioNet provides a comprehensive suite of post-symptom, looping, and auto trigger event monitors as part of its turn-key cardiac event monitoring service.

iv. Biotronik

Home Monitoring is an automatic telecardiologic service for efficient therapy management using the cellular phone network.

Cardio Messenger: The CardioMessenger® II connects the implant with the BIOTRONIK® service center. This exterior device receives the data sent by the implant (trend, event, or patient-initiated message) and forwards the information to the service center via the cellular phone network (Schwiebert et al, 2001).

The wireless/remote health monitoring systems/devices developed by companies such as Toumaz, Lifelink, Corventis, CardioNet, MedNet, Biotronik and Triage Wireless can monitor vital signs continuously and keeps clinicians connected to patients but none of these devices have any form of security embedded in them considering the fact that they collect vital sign data from patients and transmit them wirelessly to the central station thereby making the patients medical data vulnerable and susceptible to all forms of electronic attacks. Also these patients data are transmitted raw as they are collected without any form of analyses on the data.

2.11.2 Other Devices Which Can Be Used In Remote Monitoring

i. Thermometers

Digital Thermometers come in several different forms, including oral, rectal, ear, and forehead (Riedel, 2003). They are all easier to read than traditional thermometers, and don't contain mercury, which is a known health risk if the thermometer breaks.

The more expensive models promised results in less time, an especially attractive feature for those models aimed at young children or infants. On one end, the Timex AccuCurve takes 30 seconds to produce results, and at the other, Braun and Vicks each has a model that takes only one second (Riedel, 2003).

Skylark Device & Systems makes a thermometer that is built into a pacifier, with memory for the last measured temperature.

ii. Heart Rate Monitors

Most heart rate monitors resemble wristwatches, and many come with features that have been on digital watches for a long time. They include a chest strap that does the actual measurement, allowing the readout device to stay on the wrist.

Reebok sells several heart rate monitors, ranging in price from $60 to $120. The highest-end model comes with a pyramid display for exercise intensity, 99-lap memory stopwatch, average heart rate, and other features. None of the monitors has the ability to interface with a PC.

iii. Blood Pressure Monitors

Blood pressure monitors typically have a cuff that goes around a finger, the wrist, or the upper arm.

Omron makes several kinds of blood pressure monitors. Features include fast measurement, memory, a printer, and different sized cuffs. Some models have electronic inflation (automatic), and some require you to use a pump (manual).

Users can wear a vital signs monitor directly on their bodies in the form of a wireless device that monitor heartbeat, heart and breathing rates, and attitude (i.e. notice if the wearer falls) (Riedel, 2003). The monitor sends the information to a base station next to user's PC, which then relays the information to a hospital or other monitoring organization, over the Internet. The remote monitoring group can then intervene if necessary.

iv. Pedometers / Step Counters

Pedometers and step counters serve a dual purpose. First, they measure an exercise regimen, such that if it advised a 2-mile walk every other day, the pedometer could inform the wearer when that objective had been met. Second, they allow the wearer to keep track of walking patterns during a regular day, outside of an exercise program (Obrenovic et al, 2002).

In principle, a pedometer measures the distance walked, while a step-counter counts the number of steps taken. Each uses the same technology (motion of your waist rising and falling while you walk [California WIC Association], but a pedometer allows input of the stride width, which then enables a simple calculation of distance travelled. Since it takes an average stride width, the distance reported is inevitably an estimate, although devices don't report a margin of error. In any case, the measurement becomes problematic if gait is a changing parameter, for example when alternating between running, jogging, and walking.

There are three requirements to use wireless technology in remote health monitoring: (Obrenovic et al, 2002).

1. Small Form Factor
2. Extremely Low Power
3. Protocol for Low Latency / High Scalability / High Responsiveness

With these in place, devices are possible that monitor vital signs on individuals 24/7, no matter where they go or for how long.

One such system (Schwaibold et al, 2004) is based on Bluetooth communication among sensory units on the patient, a mobile information unit or base station, a rescue service, and a digital online patient record. It is able to monitor vital parameters and patient location, save that data to the patient's permanent data record, allow for manual input of patient actions, such as food intake or exercise, and provide a context-sensitive reaction, such as local emergency information.

The PHMon system [PHMon] has an identical structure, with a central database more clearly defined, and with messages going over the public Internet. The project description points out the potential for large savings at the hospital, if patients can be efficiently monitored remotely.

Alternately, the cell phone could be used as the transmission device (Abascal & Civit, 2001). In response to a single button press, measurements could be taken and immediately sent over the user's cell phone to an emergency service. This could be done in conjunction with other functions that the cell phone would have during regular usage.

Agent technology can be used for applications in telemedicine, in which a presentation agent goes over a stream of data from a medical device (such as a blood pressure monitor) and renders it in a graphical format that is meaningful to someone monitoring the patient or user (Obrenovic et al, 2002). By moving the heavy graphic processing to a PC, the device is free to only be involved in data collection, thus reducing the necessary size and alleviating some of the interface issues.

While personal devices today are largely if not completely external, the next generation may be implanted under the skin. Such devices (Schwiebert et al, 2001) could include artificial retinas, glucose monitors, organ monitors, cancer detectors, and general health monitors. They include many unique challenges but have the potential to give people an unprecedented view of what goes on inside their bodies.

The 'pill box' helps keep track of the patient's medication by sending a signal to his or her mobile phone every time a pill is removed. If a patient forgets to take medication or is taking too many pills, he or she is sent a reminder via mobile phone to follow the prescribed doses (Schwiebert et al., 2001).

Since vital measurements are readily available and can be monitored on-the-go, it is expected that both these solutions will increase patient mobility and help reduce the need for patients to visit the doctor. Regular and frequent monitoring can also bring peace of mind to people with chronic illnesses or those recovering from an operation. These devices are already of great interest to the pharmaceutical industry, which will be able to conduct field trials of new drugs more accurately.

These solutions are examples of how technology from the emerging field of information-based medicine can be used to deliver medical information in real time to help the medical profession and the health industry improve the quality of patient care (Obrenovic et al., 2002).

Patients exhibiting any of a variety of serious, chronic conditions which require monitoring of medications, or the like, benefit from frequent testing. Many conditions which have traditionally required weekly or even daily clinic visits for testing may be tracked using remote monitoring with a customized testing frequency employed. This approach generally exceeds the benefits of care offered by frequent clinic visits at a fraction of the cost. In addition, hospital, assisted living and hospice units, for example, can also benefit greatly from portable, multiple-test patient monitoring units that monitor vital signs and conduct other tests and which are integrated into one management system including a central monitoring system capable of interfacing with many other devices (Schwaibold et al., 2004).

Pulse oximetry is now a well-developed and widely used noninvasive technique of assessing oxygen saturation in pulmonary and cardiac patients. The valuable data obtained, including heart rate, has stimulated its use in acute, as well as chronic, care facilities treating pulmonary and cardiac diseases. Pulse oximetry can be used to indicate early signs of worsening cardiac failure if used to monitor patient desaturation during sleep (Schwaibold et al., 2004). As left ventricular failure worsens, pulmonary congestion worsens and gas exchange is increasingly impaired, leading to a decrease in the arterial oxygen saturation as well as increase in heart rate. Timely reversal of this condition by relieving the left ventricle from excess afterload is essential and monitoring pulse oximetry also provides a measure of the improvements in cardiac output and reduction in left ventricular filling pressure that occurs with therapy.

Cardiac failure manifests itself by fluid retention and weight gain. Systemic venous congestion, another condition typical of CHF, is also reflected by salt and water retention which produces weight gain and an inappropriate increase in systemic blood pressure. Chronic systemic hypertension is also often the primary cause of failing heart and kidneys. Thus, closely monitoring patient weight also tracks the congestive state. Complementing this with blood pressure, heart rate and pulse oximetry monitoring provides the health professional multiple objective measures (Schwaibold et al., 2004).

2.11.3 Remote Health Monitoring System

Fu et al. (2005) described a remote health monitoring system with a central monitoring station which receives data from different remote testing units. Each of the testing units is designed for a particular patient and provides care for that patient alone. A monitoring unit mayalso take care of multiple patients and transmit the data received to the central monitoring station using telephone lines. Their system connected the testing units to the central monitoring station using only telephone lines. They did not discuss any security measures to safeguard these patients' data as they travel between the testing units and the central monitoring station. Also no analysis was done on the data before it is sent to the central monitoring station.

The system described by Lee (2000) is a home medical surveillance system which includes a large number of patient subscriber apparatuses that interface with a central station. The patient's data is taken at a particular predetermined time and transmitted to the central station. Lee's system did not entail a continuous monitoring of the patient's data. Thay also did not state the method of interfacing their apparatus used with the central station and did not mention the name of this apparatus. The patients's data in the system described by Lee is taken at a particular predetermined time and transmitted to the central station. This predetermined time is not stated and therefore it is not quantifiable.

Gallant et al (2001) described a microprocessor based patient monitoring system which used a plurality of devices for measuring patients' parameters. The data obtained is transmitted from the monitoring units over the telephone lines to a PC. Gallant et al's system only transmitted the patient's data using telephone lines. The name of the devices used for the measurement was not mentioned, there is no analyses on the data collected before being transmitted and no security measures were taken to safeguard the patient's data.

Policastro et al (2002) further describes a portable microprocessor-controlled apparatus for monitoring, storing and transmitting patients' data from stored memory to a remote location over a telephone line or to a built-in graphic display or printer. Again Policastro et al's system transmitted the patients' data over a telephone line to a graphic display or printer. Their system did not have any database where the patient's data are stored and no security measure was taken to safeguard the patient's data.

2.11.4 Internet

The Internet is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide.

It is a *network of networks* that consists of millions of private, public, academic, business, and government networks of local to global scope that are linked by a broad array of electronic and optical networking technologies (Hafner, 1998). The Internet carries a vast array of information resources and services, most notably the inter-linked hypertext documents of the World Wide Web (WWW) and the infrastructure to support electronic mail.

Most traditional communications media, such as telephone and television services, are reshaped or redefined using the technologies of the Internet, giving rise to services such as Voice over Internet Protocol (VoIP). Newspaper publishing has been reshaped into Web sites, blogging, and web feeds. The Internet has enabled or accelerated the creation of new forms of human interactions through instant messaging, Internet forums, and social networking sites (Cohen et al., 2001).

The Internet has no centralized governance in either technological implementation or policies for access and usage; each constituent network sets its own standards (Walter et al., 2002). Only the overreaching definitions of the two principal name spaces in the Internet, the Internet Protocol address space and the Domain Name System, are directed by a maintainer organization, the Internet Corporation for Assigned Names and Numbers (ICANN) (Comer, 2006). The technical underpinning and standardization of the core protocols (IPv4 and IPv6) is an activity of the Internet Engineering Task Force (IETF), a non-profit organization of loosely affiliated international participants that anyone may associate with by contributing technical expertise (Barab´asi et al., 2002).

2.11.5 Bluetooth

Bluetooth is a proprietary open wireless protocol for exchanging data over short distances (using short length radio waves) from fixed and mobile devices, creating personal area networks (PANs). It was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization.

- **Implementation**

Bluetooth uses a radio technology called frequency-hopping spread spectrum, which chops up the data being sent and transmits chunks of it on up to 79 bands of 1 MHz width in the range 2402-2480 MHz. This is in the globally unlicensed Industrial, Scientific and Medical (ISM) 2.4 GHz short-range radio frequency band

Bluetooth provides a secure way to connect and exchange information between devices such as faxes, mobile phones, telephones, laptops, personal computers, printers, Global Positioning System (GPS) receivers, digital cameras, and video game consoles.

The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group (SIG). The Bluetooth SIG consists of companies in the areas of telecommunication, computing, networking, and consumer electronics.

To be marketed as a bluetooth device, it must be qualified to standards defined by the SIG (Stallings, 2005).

i. Uses

Bluetooth is a standard communications protocol primarily designed for low power consumption, with a short range (power-class-dependent: 100 m, 10 m and 1 m, but ranges vary in practice) based on low-cost transceiver microchips in each device. Because the devices use a radio (broadcast) communications system, they do not have to be in line of sight of each other.

ii. Computer requirements

A personal computer must have a Bluetooth adapter in order to communicate with other Bluetooth devices (such as mobile phones, mice and keyboards). While some desktop computers and most recent laptops come with a built-in Bluetooth adapter, others will require an external one in the form of a dongle.

Unlike its predecessor, IrDA, which requires a separate adapter for each device, Bluetooth allows multiple devices to communicate with a computer over a single adapter.

2.11.6 ZigBee

ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on the IEEE 802.15.4-2003 standard for wireless personal area networks (WPANs), such as wireless headphones connecting with cell phones via short-range radio. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth. ZigBee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life, and secure networking (Stallings, 2005).

The ZigBee Alliance is a group of companies that maintain and publish the ZigBee standard.

2.11.7 TCP/IP (Network function)

TCP provides a communication service at an intermediate level between an application program and the Internet Protocol (IP). That is, when an application program desires to send a large chunk of data across the Internet using IP, instead of breaking the data into IP-sized pieces and issuing a series of IP requests, the software can issue a single request to TCP and let TCP handle the IP details.

IP works by exchanging pieces of information called packets. A packet is a sequence of bytes and consists of a *header* followed by a *body*. The header describes the packet's destination and, optionally, the routers to use for forwarding until it arrives at its final destination. The body contains the data which IP is transmitting (Andrew, 2002).

Due to network congestion, traffic load balancing, or other unpredictable network behaviour, IP packets can be lost, duplicated, or delivered out of order. TCP detects these problems, requests retransmission of lost packets, rearranges out-of-order packets, and even helps minimize network congestion to reduce the occurrence of the other problems. Once the TCP receiver has finally reassembled a perfect copy of the data originally transmitted, it passes that datagram to the application program. Thus, TCP abstracts the application's communication from the underlying networking details.

TCP is used extensively by many of the Internet's most popular applications, including the World Wide Web (WWW), E-mail, File Transfer Protocol, Secure Shell, peer-to-peer file sharing, and some streaming media applications.

TCP is optimized for accurate delivery rather than timely delivery, and therefore, TCP sometimes incurs relatively long delays (in the order of seconds) while waiting for out-of-order messages or retransmissions of lost messages. It is not particularly suitable for real-time applications such as Voice over IP. For such applications, protocols like the Real-time Transport Protocol (RTP) running over the User Datagram Protocol (UDP) are usually recommended instead (Vinton & Robert, 1974)

2.11.8 Omron HEM-790it

The *Omron HEM-790IT i*s the ultimate tool for keeping accurate track of your blood pressure; day in and day out.

This automatic blood pressure monitor is the highest spec of the Omron home use BP monitors with arm cuff. It combines ease of use, precision and comfort along with making use of the ultimate in computer technology to allow you to see any fluctuations in your blood pressure in one easy glance.

Ideal for both medium and large sized arms, the **Omron HEM-790IT with Advanced Omron Health Management Software** is fitted with the innovative ‘Comfit Cuff’, specifically designed for quick and correct positioning along with ultimate comfort. No need for a second person to aid you, simply wrap the cuff around your arm and, at a single press of a button, your blood pressure, pulse rate and any cardiac arrhythmias (heart beat irregularities) are calculated.

The monitor benefits from Omron’s ‘IntelliSense’ technology.

This is a great feature which takes your blood pressure into account and inflates the cuff to the individual needs at that exact point. This is especially important for those people with various arrhythmias or cardiac conditions which can cause blood pressure fluctuations. It also avoids unnecessary over inflation of the cuff which can prove uncomfortable. The Omron 790-it’s included Omron Health Management Software enables you to easily download your blood pressure results via USB to your computer. Within a few seconds you will have a simple to understand graph which will show the exact trends of both your blood pressure and pulse rate over a period of time; perfect for noting any times of day when hypertension seems to be at a maximum. This software supports two separate users and clearly shows both systolic and diastolic blood pressures.

Figure 2.11: Omron HEM-790IT blood pressure monitor

2.11.8.1 Omron HEM-790IT Features

i. User friendly, single person, one button operation
ii. Clear and easy to read digital display
iii. Comfortable and pre-formed Comfit Cuff to fit both male and female arms (9” – 17” in diameter)
iv. IntelliSense technology ensures correct cuff inflation for your individual needs

v. Specifically designed to aid detection of early morning hypertension (cleared by the FDA)
vi. Displays the previous 3 readings taken within 10 minutes of each other
vii. Large memory which holds 200 readings along with the time and date taken
viii. Includes the Omron Health Management Software for easy tracking of your blood pressure via your computer
ix. Can help in the prediction of CVA's (stroke) and/or cardiac arrhythmias
x. CD ROM and manual provide clear instructions for use
xi. Can run on 4 AA batteries (included) or via an AC adaptor (included)
xii. Carry case for protection if you need to carry it with you
xiii. 5 year limited warranty

2.11.8.2 Omron HEM-790it Comparison

The Good

i. One of the easiest to use automatic blood pressure monitors on the market today
ii. Clinically proven IntelliSense technology gives accurate blood pressure readings time and time again.
iii. Large buttons and large digital readout for excellent visibility. The numbers are over 1" in size for blood pressure and ½" for pulse
iv. Simple, 3 step instructions for correct fitting of the Comfit Cuff
v. Detects morning hypertension, unlike Omron's more budget models, the Omron HEM-712C blood pressure monitor and Omron HEM-711DLX
vi. Once you have downloaded the Omron Health Management software onto your computer you just connect up the monitor via a USB and download. The monitor can store up to 200 readings so this only needs to be done once every few weeks if you prefer
vii. This model is really the best for computer connections. By showing the averages of the readings for different times of the day you can quickly spot

any changes. Print off the graphs from your computer or even transport them to a Microsoft Excel worksheet and take them along to the surgery when you have your check ups. The more data your doctor has, the more accurate his analysis.

viii. This software is also compatible with other Omron products which support this feature.

ix. You can track readings for two people. This is done by a simple slide button – person A and person B. Both the monitor and the software allow for this so making this an ideal monitor for a couple who both need to track their blood pressure.

x. It has the ability to detect any arrhythmias (irregular heart beats) and will show this on the screen by displaying the relevant symbol for that particular irregularity.

The Cons

i. The Comfit Cuff should be replaced once a year.

ii. Because this automatic blood pressure monitor is highly accurate and sensitive, you do need to ensure that you place the cuff in the same place each and every time. Any variations can result in widely varying blood pressure results so you do need to pay attention to the instructions to make sure you use this correctly.

2.12 Summary of the Limitations of Existing Systems

Several works have been done on remote medical monitoring by several people and a lot of constraints have been identified.

Andrew and Alfred (2008) designed a remote medical monitoring system which consisted of three tiers. Tier 1 is the set of sensors that discern signals of interest and relay information to each other and the data hub. Tier 2 is the data hub which provides a more computational capacity allowing data to be stored or further processed before transmission to some outside medical network via the internet.

The third tier is the medical network which is operated by a healthcare provider such as a hospital or telemedicine center where staff can handle emergency situations.

The major system issues in the design of Andrew and Alfred (2008) is that their system did not make provisions for adequate security of the patient's data as it travels across networks. Also there is no form of authentication of the patient's data thereby making the system porous.

Eliasz et al (2011) described a wearable healthcare monitoring system that integrates electrocardiogram and an accelerometer sensor. Their research focused on the right connection of the hardware units, combination of the detection of the QRS complexes and the detection of human falls. (QRS complexes is the name for the combination of three of the graphical deflections seen on a typical electrocardiogram (ECG), calculation of heart rate and the detection of human falls).

The limitation of their work is that the proposed method of detecting falls did not give satisfactory results in performed tests. It often provided with false positive detections especially during rapid movements or stops. Eliasz et al (2011) stated that future works should be to develop more advanced algorithms and improve fall detection algorithm.

Mehmet et al (2007) built a Wireless Body Sensor Network (WBSN) based on the medical implant communication service band. The project used both implanted and on-body nodes.

Mehmet et al (2007) suggests that future work should include more sensors to monitor other vital signs such as blood pressure, pulse rate, oxygen saturation level, ECG etc. They also proposed the possibility of extending the capability to automate alerts which can get the attention of health professionals as soon as they are needed.

Hui et al (2007) presented a paper that described the system design of a wireless electrocardiogram (WECG) system and focused more on the personal server hardware and software design.

They proposed further work to link up the proposed system with a health care network through the internet.

2.12.1 Deduction from Reviewed Works

The following are deductions from the reviewed works:

i. The different people who worked on remote medical monitoring system did not consider the security and privacy of the patient's records as they travel through the networks. The security of patients' data is very important because the patients' medical records need to be safeguarded against unauthorized access.
ii. The method proposed by Eliasz et al (2011) to detect human falls did not give satisfactory result in performed tests.
iii. The fall detection algorithm designed by Eliasz et al (2011) was quite simple and needs to be improved on.
iv. Mehmet et al (2007) suggests that more robust systems should be designed to include more sensors to monitor blood pressure, pulse rate, oxygen saturation level, ECG etc.
v. Mehmet et al (2007) also proposed the possibility of extending the capability to automate alerts which can get the attention of the health professionals as soon as they are needed.
vi. The system designed by Istepanian et al (2004) is unsuitable for lengthy continuous monitoring particularly during normal activity, intensive training or computer-assisted rehabilitation.
vii. There is need to link up the system proposed by Hui et al (2007) with a health care network through the internet.

2.12.2 Expectations from the New System

The new system is an enhanced secured remote medical monitoring system which is able to:

i. Monitor the vital signs of patients from a remote location and send the results to the server which is located in the hospital.
ii. The vital signs to be monitored include the blood pressure, the pulse rate and the mean arterial pressure.
iii. These vital signs are simulated using a simulated mobile phone.
iv. The new system also addressed the major gap identified in other remote medical monitoring systems which is the security of the patients' data as they travel through the networks. A computer program was written which encrypts and decrypts the patients' data. A 16-bit symmetric key encryption algorithm which employs the Advanced Encryption Standard was used to achieve the encryption.

CHAPTER 3

METHODOLOGY AND SYSTEM ANALYSIS

3.1 Introduction

In system analysis, the system is surveyed and planned. This involves the study and analysis of the existing system, the definition of system requirements and priorities for the new or improved system.

The components of the existing system are studied and analysed. This is done by identifying all the problems in the existing system, the causes of the problems and then state the actions to be taken in order to produce a better system.

Feasibility study is carried out to determine whether the project is feasible or not. Data flow diagrams and information flow diagrams are also drawn at this stage. The methodologies to be used for the project are also stated here.

Some methodologies used in systems analysis are:

i. Structured Systems Analysis and Design Methodology (SSADM)
ii. Object Oriented Analysis and Design Methodology OOADM)
iii. Expert Systems Approach to Software Engineering
iv. Prototyping

3.2 Adopted Method for this Research

The methods adopted for this research are the Structured Systems Analysis and Design Methodology, Object Oriented Analysis and Design Methodology (OOADM) and Prototyping.

This is because SSADM is best suited for a detailed design and analysis of an information system through which an improved system can be developed from an existing system.

Object Oriented Analysis and Design Methodology is adopted because since it models a system as a group of interacting objects where each object is characterized by its class, its state and its behavior, and the proposed system is going to be a web-based system, and the programming language to be deployed will also be web-based, it is imperative that an object oriented methodology be adopted.

Prototyping is also employed in this research because, the proposed system requires that a partial but functional version of the application be developed and prototyping is well suited for this purpose.

3.3 Data Gathering

The Central Hospital Warri was used as a case study for this research and the data used for the analysis were gathered from this hospital.

The Central Hospital Warri is a very big hospital and therefore before the researcher set out to collect any data from there, she had to seek for their permission. A sample of the letter to that effect is attached in the appendix.

A survey type questionnaire and a face to face interview were used for the data gathering. The questions in the questionnaire were divided into two sections, section 1 for the nurses and section 2 for the patients.

It was discovered that majority of the patients were illiterates and therefore the face to face interview was used for them. The face to face interview was done for the patients by using the questions in the questionnaire.

The questionnaire was used for the nurses and the chief matron. The chief matron was also interviewed.

3.3.1 The Use of Questionnaire

Questionnaires were administered to the nurses and patients in the central hospital Warri and their responses were analyzed.

The questions in the questionnaire were analyzed and grouped under the following headings:

i. Doctor’s response to emergency calls
ii. Efficiency
iii. Redundancy
iv. Cost Effectiveness
v. Commitment to duty
vi. Dependence on power
vii. Life Saving
viii. Doctor’s on call
ix. Reliability
x. Disturbance
xi. Faulty Instruments
xii. Number of times a day the blood pressure and pulse rate is checked
xiii. Failure to check the blood pressure and pulse rate

The users’ responses is presented in Table 3.1

Table 3.1: Attributes of the users’ responses

S/N	ATTRIBUTES	VERY OFTEN	NOT SO OFTEN	NOT AT ALL	TOTAL
1.	Response to emergency calls	94	28	58	180
2.	Efficiency	56	112	12	180
3.	Redundancy	90	6	84	180
4.	Cost Effectiveness	91	29	60	178
5.	Commitment to duty	50	22	24	96
6.	Dependence on power	28	0	152	180
7.	Life saving	152	4	12	168
8.	Doctor’s on call	90	42	48	180
9.	Reliability	100	64	16	180
10.	Disturbance	68	23	5	96
11.	Faulty Instruments	56	30	16	102
12.	Number of times a day the vital sign is checked	148	12	20	180
13.	Failure to check Bp and Pulse rate	0	78	30	108

The weighted average was used to analyze the results of the questionnaire and the interview. The weight 3 was assigned to represent very often, 2 to represent not so often and 1 to represent not at all. The responses for each of the attributes was entered in a table and the weighted average was calculated. The value of the result got after calculating the weighted average determines which of the otions to take for a particular attribute.

The weighted average is given by the equation:

$$\bar{x} = \frac{\sum fx}{\sum f} \quad (3.1)$$

3.3.2 Calculation of the different attributes

i. Doctor's response to emergency calls

Table 3.2: Doctor's response to emergency calls

X	3	2	1	Total
F	94	28	58	180
Fx	282	56	58	396

3: Doctors respond very often to emergency calls
2: Doctors do not respond very often to emergency calls
1: Doctors do not respond to emergency calls at all.

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{396}{180} = 2.2$$

From the result got from the calculation, doctors do not respond very often to emergency calls.

ii. Efficiency

Table 3.3: Efficiency

X	3	2	1	Total
F	56	112	12	180
Fx	168	224	12	404

3: The manual method of monitoring patient's blood pressure and pulse rate is very efficient
2: the manual method of monitoring patient's blood pressure and pulse rate is not very efficient
1: the manual method of monitoring patient's blood pressure and pulse rate is not efficient at all.

To calculate the weighted average:

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{404}{180} = 2.24$$

From the result from the weighted average calculation, the manual method of monitoring patient's blood pressure and pulse rate is not very efficient.

iii. Redundancy

Table 3.4: Redundancy

X	3	2	1	Total
F	90	6	84	180
Fx	270	12	84	366

3: the nurses will feel redundant if the job of monitoring the vital signs are taken away from them
2: the nurses will not feel so redundant if the job of monitoring the vital signs are taken away from them
1: the nurses will not feel redundant at all if the job of monitoring the vital signs are taken away from them

To calculate the weighted average:

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{366}{180} = 2.03$$

Since the result from the calculation of weighted average for this attribute is 2.03, the option 2 is taken which is: the nurses will not feel so redundant if the job of monitoring the vitalsigns are taken away from them.

iv. Cost Effectiveness

Table 3.5: Cost Effectiveness

X	3	2	1	Total
F	91	29	60	180
Fx	273	58	60	391

3: the remote medical monitoring will be very cost effective
2: the remote medical monitoring will not be very cost effective
1: the remote medical monitoring will not be cost effective at all

To calculate the weighted average

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{391}{180} = 2.17$$

Based on the calculation, it is stated that the remote medical monitoring will not be very cost effective.

v. Commitment of the nurses to duty

Table 3.6: Commitment of the nurses to duty

X	3	2	1	Total
F	50	22	24	96
Fx	150	44	24	218

3: the nurses are so committed to their duty of monitoring the vital signs
2: the nurses are not so committed to their duty of monitoring the vital signs
1: the nurses are not committed to their duty of monitoring the vital signs at all.

To calculate the weighted average

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{218}{96} = 2.27$$

From the analysis of the questionnaire and the result from the weighted average calculated, the nurses are not so committed to their duty of monitoring the vital signs of patients.

vi. Dependence on power

Table 3.7: Dependence on power

X	3	2	1	Total
F	28	0	152	180
Fx	84	0	152	236

3: the instrument for the manual monitoring of blood pressure and pulse rate depends on power all the time.
2: the instrument for the manual monitoring of blood pressure and pulse rate does not depend on power all the time.
1: the instrument for the manual monitoring of blood pressure and pulse rate does not depend on power at all.

To calculate the weighted average

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{236}{180} = 1.31$$

From the calculation of the weighted average, the result is 1.31which corresponds to the number 1 above which states that the instrument for the manual monitoring of blood pressure and pulse rate does not depend on power at all.

vii. Life saving

Table 3.8: Life saving

X	3	2	1	Total
F	152	4	12	168
Fx	456	8	12	476

3: the remote monitoring of the vital signs of patients will very likely save the lives of the patients
2: the remote monitoring of the vital signs of patients will not likely save the lives of the patients
1: the remote monitoring of the vital signs of patients will not save the lives of the patients at all.

To calculate the weighted average

$$\overline{x} = \frac{\sum fx}{\sum f}$$

$$\overline{x} = \frac{476}{168} = 2.83 \cong 3$$

From the result of the weighted average, it is stated that the remote monitoring of the vital signs of patients will very likely save the lives of the patients.

viii. Doctors on call

Table 3.9: Doctors on call

X	3	2	1	Total
F	90	42	48	180
Fx	270	84	48	402

3: the doctors on call stay in their offices very often
2: the doctors on call do not stay in their offices very often
1: the doctors on call do not stay in their offices at all.

To calculate the weighted average

$$\overline{x} = \frac{\sum fx}{\sum f}$$

$$\overline{x} = \frac{402}{180} = 2.23$$

From the result, it is stated that the doctors on call do not stay in their offices very often.

ix. Reliability

Table 3.10: Reliability

X	3	2	1	Total
F	100	64	16	180
Fx	300	128	16	444

3: the instruments for the manual monitoring of the vital signs are very reliable

2: the instruments for the manual monitoring of the vital signs are not very reliable

1: the instruments for the manual monitoring of the vital signs are not reliable at all.

To calculate the weighted average:

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{444}{180} = 2.46$$

Therefore it is stated that the instruments for the manual monitoring of the vital signs are not very reliable.

x. Disturbance

Table 3.11: Disturbance

X	3	2	1	Total
F	68	23	5	96
Fx	204	46	5	255

3: Patients feel disturbed every time their blood pressure and pulse rate are measured
2: Patients do not feel disturbed every time their blood pressure and pulse rate are measured
1: Patients do not feel disturbed at all when their blood pressure and pulse rate are measured.

To calculate the weighted average:

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{255}{96} = 2.65 \cong 3$$

Based on the analysis of the questionnaire and the result got from the calculation of the weighted average, it is stated that patients feel disturbed every time their blood pressure and pulse rate are measured.

xi. Faulty Instruments

Table 3.12: Faulty Instruments

X	3	2	1	Total
F	56	30	16	102
Fx	168	60	16	244

3: the instruments for measuring the blood pressure and pulse rate can be faulty all the time
2: the instruments for measuring the blood pressure and pulse rate can be faulty sometimes
1: the instruments for measuring the blood pressure and pulse rate cannot be faulty at all.

To calculate the weighted average:

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{244}{102} = 2.39$$

From the results, it is stated that the instruments for measuring the blood pressure and pulse rate can be faulty sometimes

xii. Number of times a day the blood pressure and pulse rate is checked

Table 3.13: Number of times the blood pressure and pulse rate are checked

X	3	2	1	Total
F	148	12	20	180
Fx	444	24	20	488

3: the blood pressure and pulse rate of the patients are checked two times a day

2: the blood pressure and pulse rate of the patients are checked three times a day

1: the blood pressure and pulse rate of the patients are checked more than three times a day

To calculate the weighted average:

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{488}{180} = 2.71 \cong 3$$

From the result, it is stated that the blood pressure and pulse rate of the patients are checked two times a day.

xiii. Failure to check blood pressure and pulse rate

Table 3.14: Failure to check blood pressure and pulse rate

X	3	2	1	Total
F	0	78	30	108
Fx	0	156	30	186

3: the nurses fail to monitor the blood pressure and pulse rate of the patients all the time

2: the nurses fail to monitor the blood pressure and pulse rate of the patients sometimes

1: the nurses do not fail to monitor the blood pressure and pulse rate of the patients at all.

To calculate the weighted average:

$$\bar{x} = \frac{\sum fx}{\sum f}$$

$$\bar{x} = \frac{186}{108} = 1.72 \cong 2$$

From the result it is stated that the nurses fail to monitor the blood pressure and pulse rate of the patients sometimes.

3.4 Problem Identification

In Nigerian hospitals, the monitoring of patients is done by the nurses. The nurses monitor these vital signs, record the results and wait for the doctors to come round, read the report and then make the prescriptions.

The problems identified in the hospital are as follows:

i. Inadequate and inefficient monitoring of the patients by the nurses left to do this monitoring. The nurses check the patients two times in a day, morning and evening and that is not enough for the aged and chronically ill because their condition may change during the day without anybody knowing and before the nurses can monitor the vital signs in the evening, it may have been too late. This has led to the death of many patients in Nigerian hospitals.
ii. The patients' medical records are stored in the individual patient's files and accessible by every staff in the various departments thereby not protecting the patient's privacy. The patients' medical records are not secured since they are accessible to all staff of the departments. This lack of security of patients' medical records could be very dangerous as some staff could compromise and tamper with patients' medical records.
iii. Doctors do not have real time access to patients' medical records. The only way they can have access to patients' records is by calling for their files which may not be readily available.

iv. The lackadaisical attitude on the part of the nurses. It was observed that the nurses put in charge of monitoring these patients most of the time are very careless and unperturbed about the conditions of their patients. They are not devoted to the monitoring of these patients and reporting unsafe situations to the doctors until it is too late. Sometimes even when the situation is critical, they care less and this results to the death of many patients.

v. There is always delay in time between when the nurses monitor the vital signs and when any attention is given to the patient. This also results in the death of many patients.

vi. The health attendants whose duties are to call the doctors when the situations are critical do not do their work well. They show little or no interest in carrying out their duties.

vii. The instruments used for this monitoring can be faulty at times thereby giving wrong results. The interpretation of these wrong results as they are can have adverse effects on the patients.

3.5 Feasibility Studies

A feasibility study was carried out in the Central Hospital Warri and the following was ascertained.

3.5.1 Required Technology

The technology for this project are as follows:

i. Internet
ii. Vital sign sensors
iii. Routers
iv. Vsat (Very Small Aperture Terminal)

The Internet can be made available. This means that, with the installation of a Vsat (Very Small Aperture Terminal), in the hospital, internet can be configured and access given to all authorised users.

Routers are available and also affordable. The hospitals can get as many routers as possible installed at different locations to capture the vital signs sent out from the sensors, alternatively, masts (like the MTN masts) should be built at strategic locations to capture these vital signs from the patients at different locations.

3.5.2 Required Resources

The resources required are:

i. Manpower: Programmers, testers and debuggers
ii. Software and hardware

The programmers will develop the software needed for the project as well as test and debug the software. Programmers can be employed from the IT industry to do this job of software development, testing and debugging.

The hardware needed are the computers and the network hardware. All these are also available and affordable.

This system can therefore be said to be technically feasible.

3.6 Analysis of the Present System

3.6.1 Present Procedure for Patient Monitoring In Central Hospital Warri

In a typical Nigerian hospital (e.g. Central Hospital), a sick patient visits the hospital, obtains a card for a fee, sees the doctor who examines the patient and determines whether there should be further investigation on the case of the patient or to simply prescribe drugs for the patient.

If the patient’s case requires further investigation, the patient is sent to the laboratory for tests, the result of which is sent to the doctor. At this point, the doctor determines whether the patient’s case requires admission and the monitoring of the patient’s vital signs. If there is need for that, the doctor instructs the nurse assigned with the duty of monitoring the vital signs.

The nurse swings into action, monitors the patient’s vital signs morning and evening, records the readings in the chart provided which the doctor examines anytime the doctor comes around. If the patient’s condition is deteriorating, the frequency of the vital sign monitoring increases. It can get to monitoring the vital signs every thirty minutes.

3.6.2 Merits of the Existing System

From the analysis of the questionnaire administered and the oral interviews conducted, the merits of the existing system include:

i. Patients feel attended to since the nurses perform a routine check of their vital signs on a daily basis.
ii. The nurses monitor the vital signs and the readings taken, so there is no problem of network failure.
iii. All the vital signs can be monitored manually, so there is no issue with power failure. This enhances efficiency since the results of their vital signs will always be available.
iv. Since the vital signs are monitored manually, the doctors always try to make themselves available to attend to the patients when the need arises.
v. The nurses feel fulfilled because since they do the monitoring of these vital signs by themselves, they feel relevant to the system.

3.6.3 Demerits of the Existing System

The existing system has a lot of demerits which are outlined below:

i. The nurses that monitor the patients most times do not show enough commitment to duty. Sometimes they do not monitor the patients when they ought to and the result can be fatal.
ii. The health attendants whose duty it is to call the doctor once there is an unsafe situation do not do their work well. Sometimes they attend to their own selfish needs when they are instructed to get the doctor to attend to the patients.
iii. The doctors on their own part do not stay in their offices most times when they are on call and when their attention is needed, they will be nowhere to be found. The health attendants also will not bother to put a call to the doctors since there is no provision for that in this hospital and they will never use their own cell phones to call the doctors even when there is an emergency. This can result to the death of the patient involved.

iv. The instruments used for the monitoring can be faulty and thereby give very inaccurate results.

v. There is always delay in time between when the nurses monitor the vital signs and when any attention is given to the patient. This also results in the death of many patients.

3.6.4 The Data Flow Diagram of the Existing System

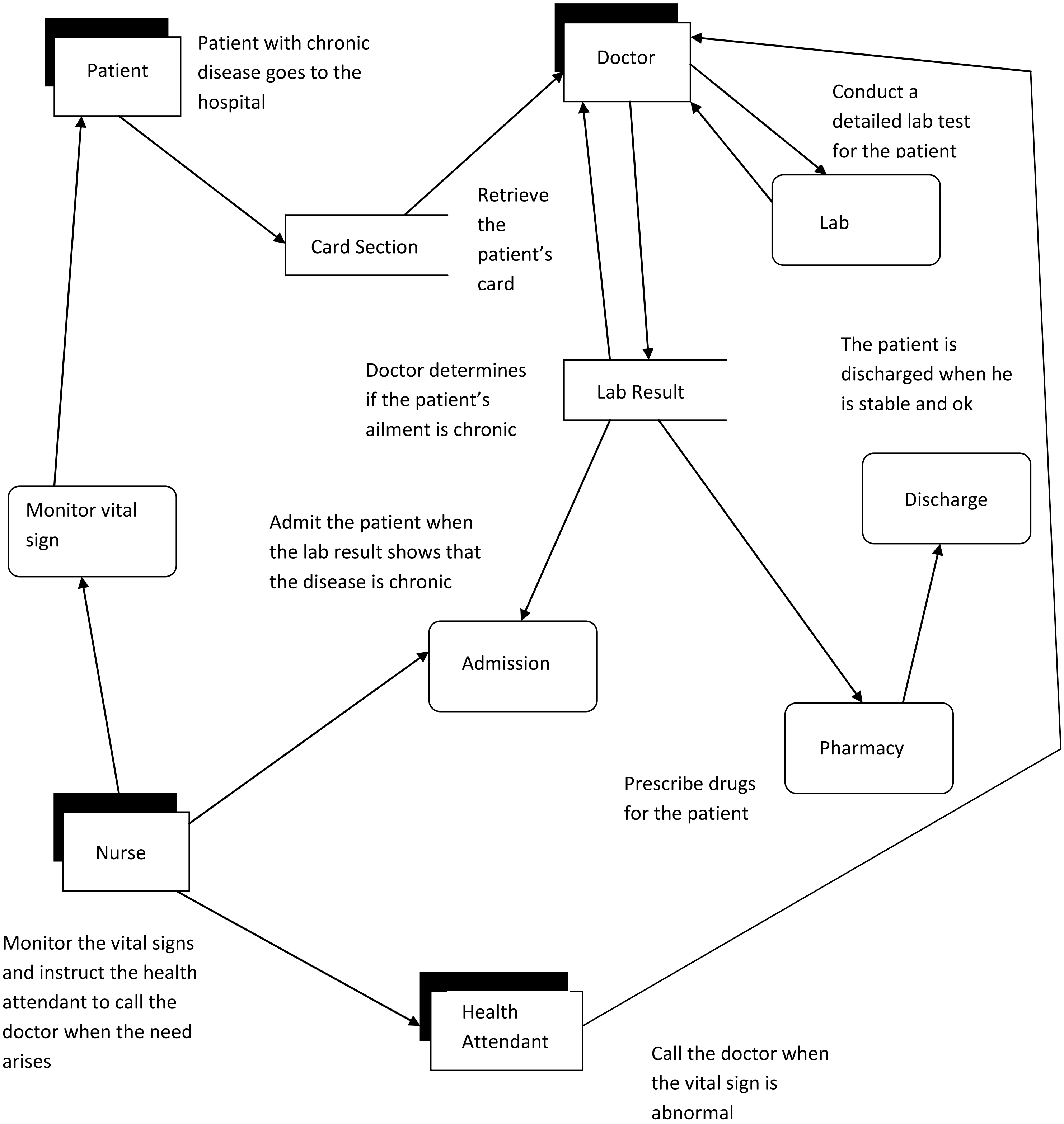


Figure 3.1: Data Flow Diagram of the present system

In the existing system, a sick patient goes to the hospital and proceeds to the card section, obtains a card and goes to see the doctor. The doctor sends the patient to the lab and when the lab results come out, the doctor looks at the result and determines the seriousness of the ailment. If the ailment is not serious, the doctor sends the patient to the pharmacy from where the patient purchases the prescribed drugs and is discharged. But if the ailment is serious, the patient is placed of admission. If the patient's ailment is not chronic, the patient remains on admission where treatment is given until the patient becomes well and gets discharged. If the patient's ailment is chronic, the monitoring and the recording of the vital signs begin. When the vital sign becomes abnormal, the nurse instructs the health attendant to call the attention of the doctor. The doctor comes around and proffers an expert advice and solution.

3.6.5 Information Flow in the Existing System

In a typical Nigerian Hospital, the flow of information is divided into two sections:

i. Admin
ii. Patient

i. Admin information Flow

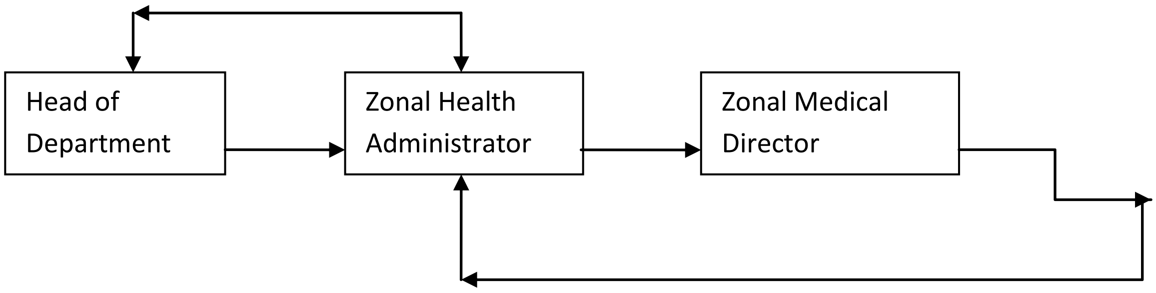


Figure 3.2: Admin information flow

In the Admin information flow, the head of every department reports directly to the zonal health administrator who in turn reports directly to the zonal medical director. The zonal medical director instructs the zonal health administrator on how to solve any given proble. The zonal health admistrator inturn instructs the haed of department on what to do.

ii. Patient Information Flow

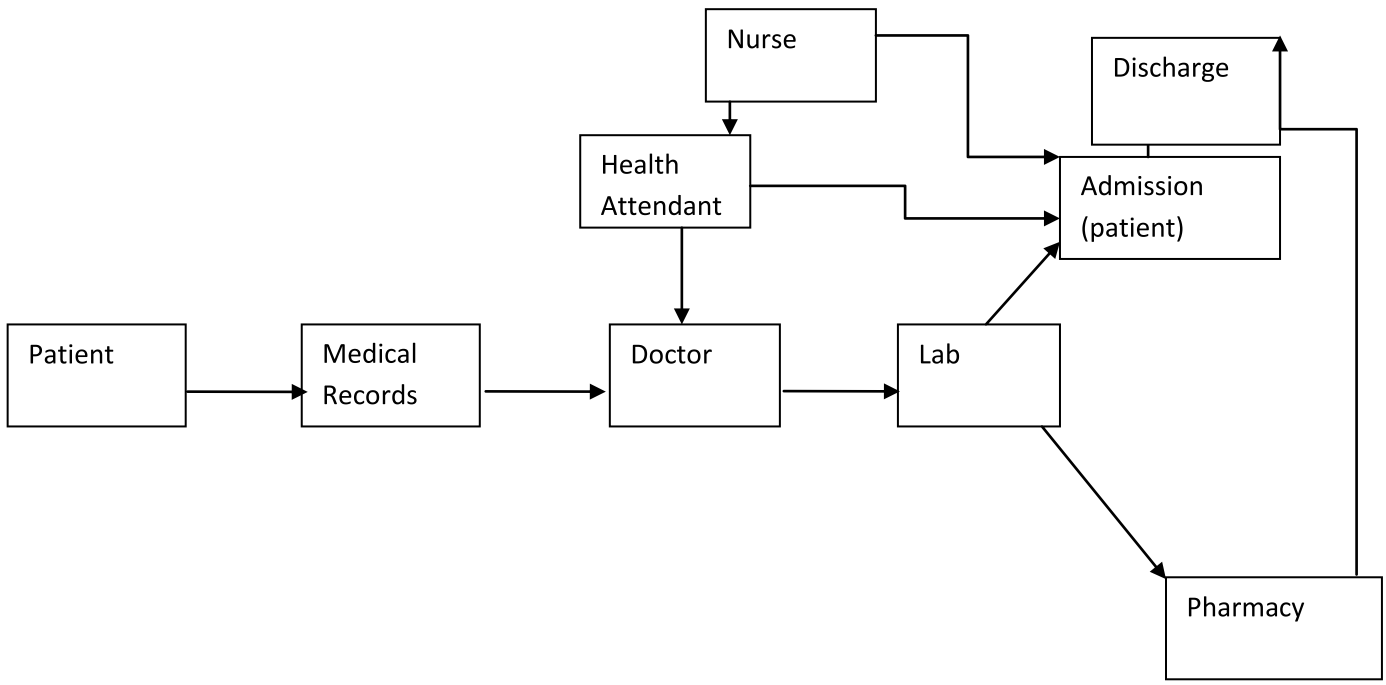


Figure 3.3 Patient information flow

In the patient information flow, the patient goes to the medical records to obtain card and goes to see the doctor. The doctor sends the patient to the lab. The patient goes to the pharmacy if the ailment is not chronic, get the drugs and gets discharged. If the patiet's ailment is chronic, the patient is placed on admission and the vital signs monitored and recorded. When the vital sign becomes abnormal, the nurse instructs the health attendant who calls the doctor to proffer medical advice and solution.

3.7 Analysis of the Proposed System

3.7.1 Proposed Procedure for Patient Monitoring In the New System

In this new system, a sick patient goes to the hospital, fills the registration form, the hospital staff captures the biometrics of the patient and uploads it into the computer. All the computers are connected through a network and therefore the doctors can view the patient's information from any of the computers. The patient is sent to the doctor with the specialty the patient needs. The doctor examines the patient, sends the patient to the lab and when the lab result comes out, the doctor determines whether the patient should be admitted or treated as an out patient. If the patient is an out patient, the doctor prescribes the drugs, sends the patient to the pharmacy to collect drugs after payment and go home. If the patient is on admission, the doctor again determines if the patient is suffering from a chronic disease. If the patient is not suffering from a chronic disease, the patient remains on admission where treatment is administered until the patient is fit to go home. But if the patient’s ailment is chronic, the vital signs of the patient is simulated using the software that runs in the simulated phone to generate the blood pressure and the pulse rate which is sent tothe server and new set of readings are generated every twenty five seconds and sent to the server.

In the server there is a queuing model which accepts these readings, analyse them before sending them to the database. Also for security reasons, as the patient’s data travel from the mobile phone to the server, they are encrypted using the encryption key and can only be decrypted in the server using the decryption key by an authorised staff. This is to safeguard the patient’s health records from unauthorised acess and use. The doctor from time to time can check his computer and view the patient’s record and provide suggestions and solution. The doctor can also view each patient’s vital sign reading graph to see the progression of the readings.

When a patient’s vital sign becomes abnormal, the nurse can send a message to the doctor. The doctor can also see it in his computer and since these readings get to the server every twenty five minutes, the doctor will always have a long range of readings to look at at every point in time and this will aid the doctor in taking accurate and timely decisions.

3.7.2 Merits of the New System

The merits of the new system are outlined below:

i. The sensors collect the readings for the vital signs of patients and forward it through the Internet to the server, which then report any unsafe situation to the medical network for medical interpretation and intervention. This saves the nurses a lot of trouble, so they will have more time for other clinical activities in the hospital.

ii. Monitoring the patients manually cause a lot of disturbance to the patients as they may be sleeping at the time or they may be very weak and may need to be allowed to rest. So in this situation where the signs are monitored without the patient knowing it, it gives the patients enough time to relax and recuperate.
iii. Any unsafe situation is reported to the medical team in charge immediately so that they can swing into action. This solves the problem of not getting the doctors in their offices when they are needed as the message will be relayed right into their phones.
iv. The readings are more accurate and reliable than the ones gotten from manual monitoring of patients.
v. The sensors periodically transmit their status and events, therefore significantly reducing power consumption and extending battery life.
vi. High performance and fault tolerant wireless devices can now be employed to eliminate medical errors, to reduce workload and increase the efficiency of hospital staff, and to improve the comfort of patients.
vii. There is better performance in the new system.
viii. Records from individual monitoring sessions can be integrated into research databases that would provide support for data mining and knowledge discovery relevant to specific conditions and patient categories.
ix. Death rate is drastically reduced to the barest minimum since the patients are attended to at the appropriate time.
x. Continuous monitoring of the patients tell a much deeper story about what is going on with the patient, revealing early signs of trouble that can trigger life-saving intervention.
xi. There is reduced loss of life in the new system since there is a real time monitoring of the patients' vital signs.
xii. The new system will provide automated working with less human interaction.
xiii. It enables villagers and chronically ill patients to have their vital signs monitored since they can be anywhere other than the hospital and their vital signs will still be checked.

xiv. More hospital beds will be made available since the patient's do not need to be in the hospitals before their vital signs can be monitored. The hospital beds can thus be reserved for emergency situations.

xv. It enables the doctors to take more timely and accurate decisions since they always have a long range of vital signs readings of each patient at every point in time.

3.7.3 Demerits of the New System

The demerits of this new system are as follows:

i. The major problem this new system will face is the network problem. Since the data stored in the mobile devices are sent over the web to the medical network, there may not be network in all the places this technology will be used and that will pose a problem for the new system.

ii. The sensors use batteries and they will tend to be discharging quickly since the sensors will be on the patients all the time.

iii. The patients may not understand this system and therefore they may tend to feel that they have been completely abandoned.

iv. The nurses may feel very uncomfortable with this system as they may tend to think that the major part of their work have been taken away from them thereby making them redundant.

v. Sometimes there may be interference between the Bluetooth devices and the medical equipment).

3.7.4 Justification for the New System

This system will be of immense help to health care in Nigeria generally despite the disadvantages. This is because:

i. The problem of network can always be solved by using the Bluetooth as a means of communication between the mobile devices where the data is stored and the medical network since the Bluetooth technology can work anywhere.

The Bluetooth connection covers a distance of 10 meters which is good since the patient has the devices on with his mobile phone having the Bluetooth connection within the same area.

ii. Ideally, sensors periodically transmit their status and events, thereby significantly reducing power consumption and extending battery life.

iii. When the patients are sensitized on the benefits of this new system, they will readily appreciate it and also prefer it knowing fully well that even though nobody is monitoring their vital signs, the devices in their bodies are doing that.

iv. Since the monitoring is continuous, it goes a long way to help the doctors to do their assessment of the patients and to provide the best medical treatment. This is because the blood pressure for instance fluctuates at different times of the day and so the doctors can do analysis of the readings obtained within a period of time to determine the average blood pressure of the patient.

v. Also when the nurses are assured that the new system will not make them redundant as they will use their time to perform other clinical activities efficiently, they will appreciate it.

vi. This system is very important and will be beneficial to Nigerian Hospitals because already, there are a lot of issues in the manual monitoring as discussed above resulting to high mortality rate in Nigeria. This system therefore will go a long way to reduce the mortality rate in Nigerian Hospitals.

3.7.5 DATAFLOW DIAGRAM OF THE PROPOSED SYSTEM

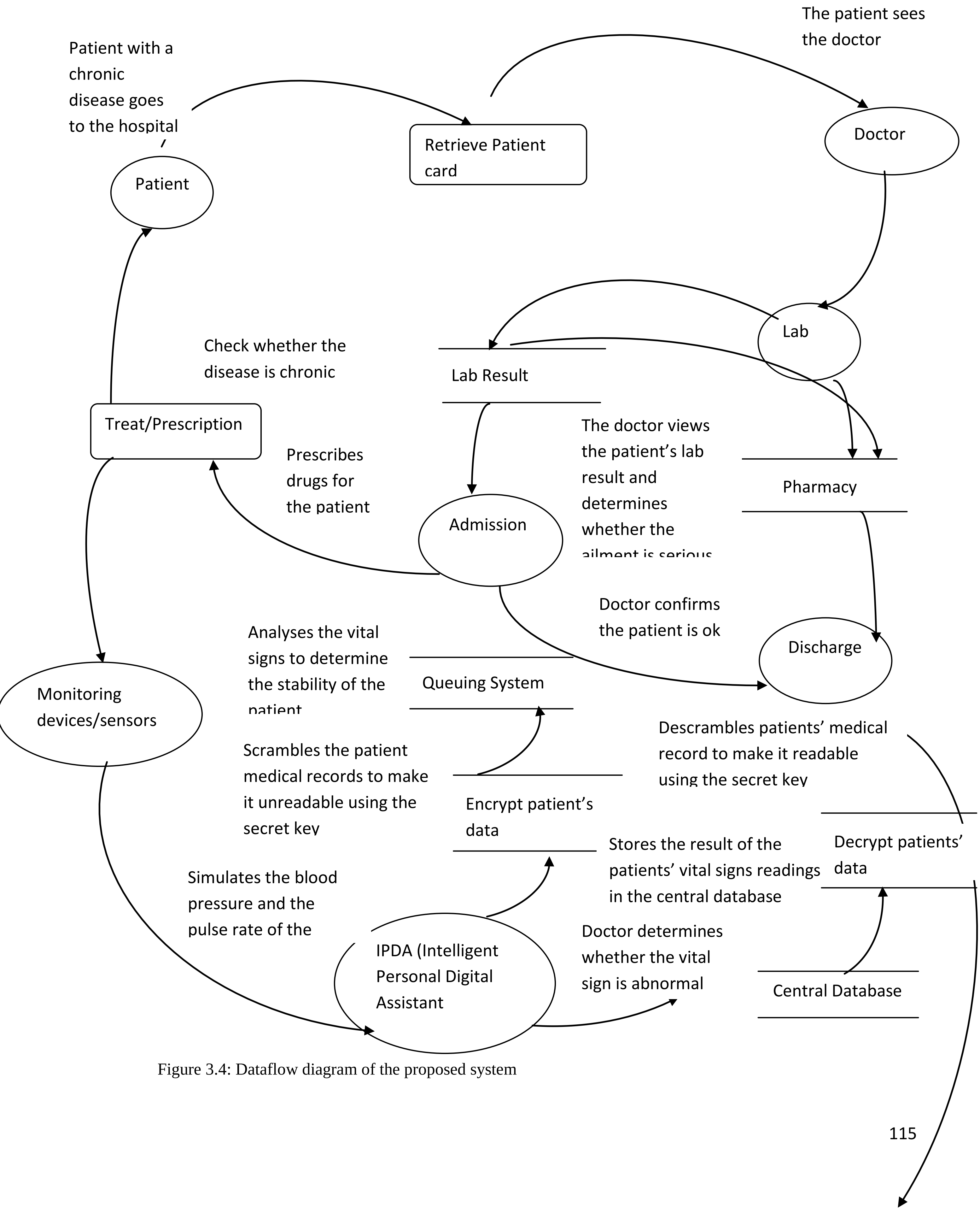


Figure 3.4: Dataflow diagram of the proposed system

A sick patient goes to the medical records, retrieves the card and proceeds to see the doctor. The doctor examines the patient and sends the patient to the lab. From the lab result, the doctor decides whether to put the patient on admission or to treat the patient as an outpatient depending on the nature of the patient's ailment.

If the patient's ailment is chronic, the monitoring of the vital signs will commence and readings are sent to the server. When the vital sign reading becomes abnormal, the doctor is notified immediately and the doctor provides solution on the next line of action.

3.7.6 Information Flow in the Proposed System

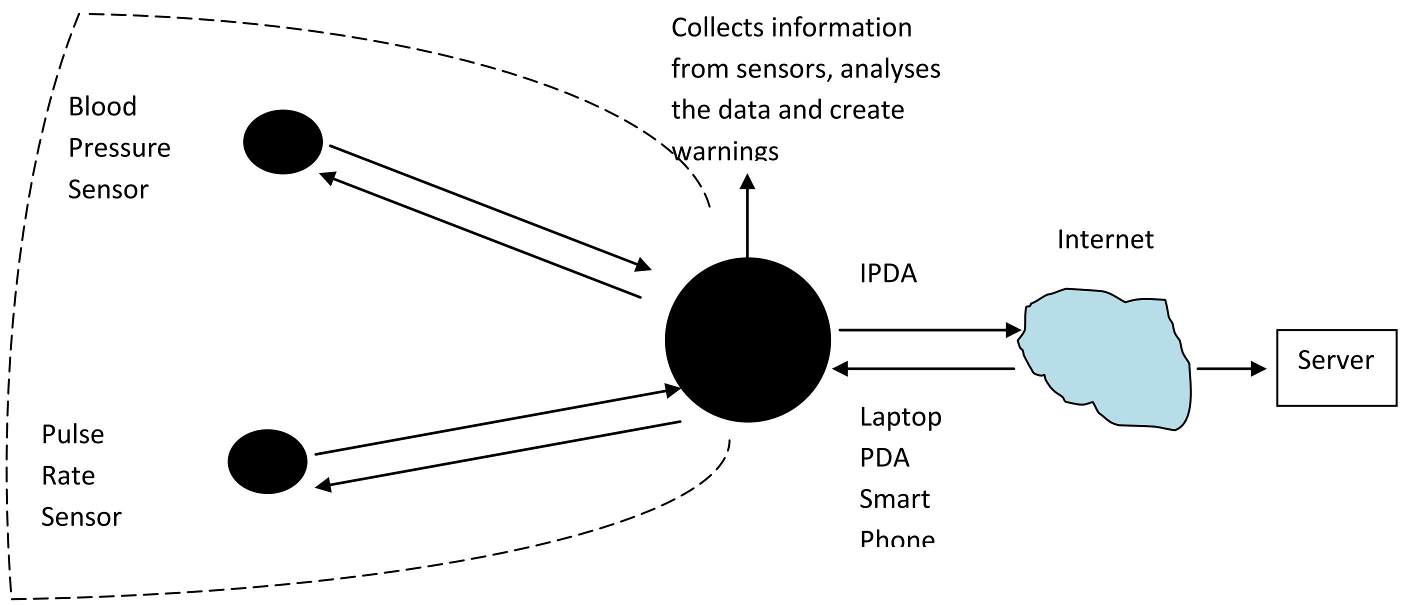


Figure 3.5: Information flow in the proposed system

In the proposed system, the information (readings) from the sensors, the blood pressure and the pulse rate sensors in this case are simulated in the Intelligent Personal Digital Assistant (IPDA) using data structures such as name, age, smoking habits, alcohol intakes etc. This IPDA which can be a laptop, a PDA or a Smartphone sends the simulated readings through the Internet to the server. The information enters the server. The queuing model queues the data according to how they are sent. The server also contains the database. The block diagram of the information flow in the proposed system is shown in Figure 3.6

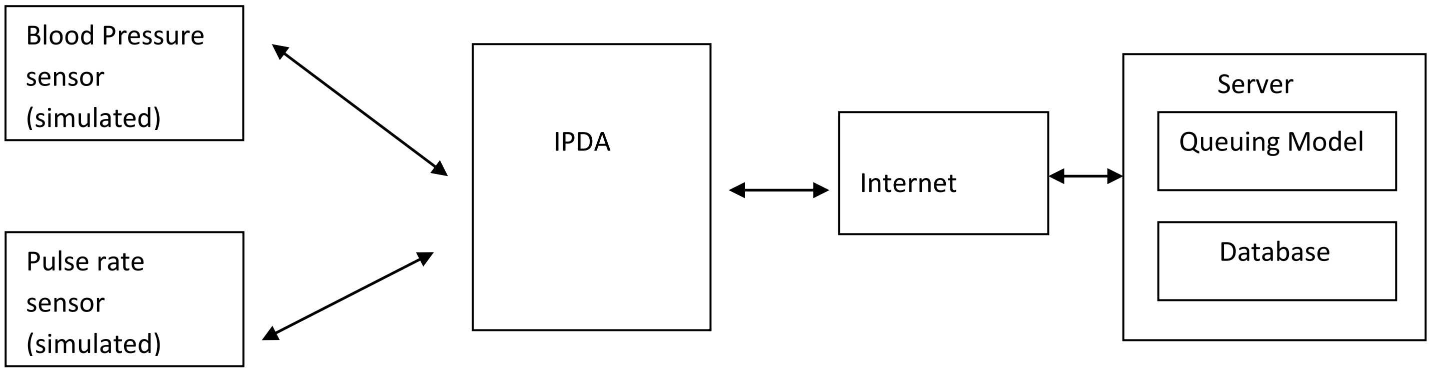


Figure 3.6: Block diagram of the information flow in the proposed system

3.7.7 High Level Models of the Proposed System

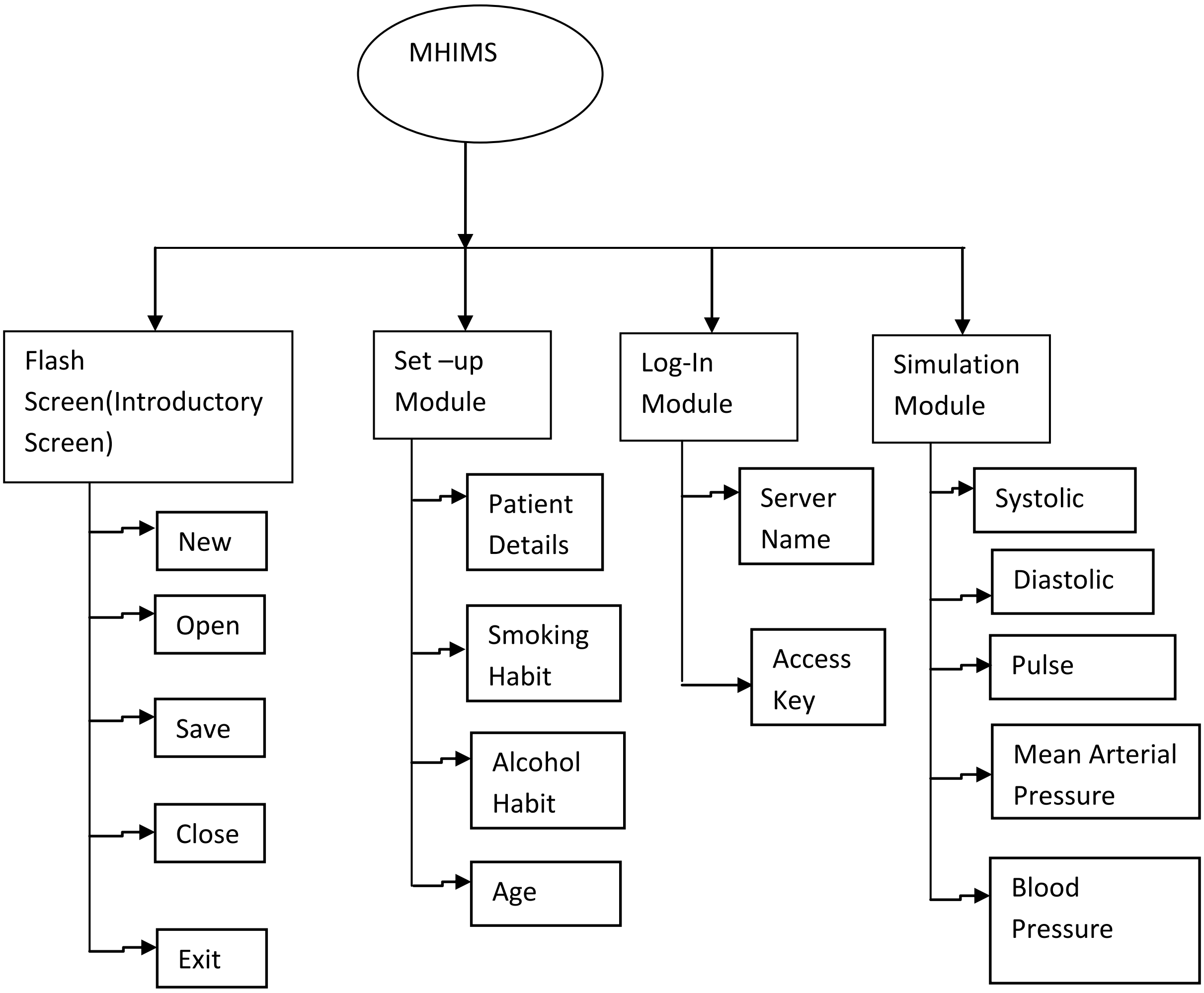


Figure 3.7. High level model of the proposed system (MHIMS)

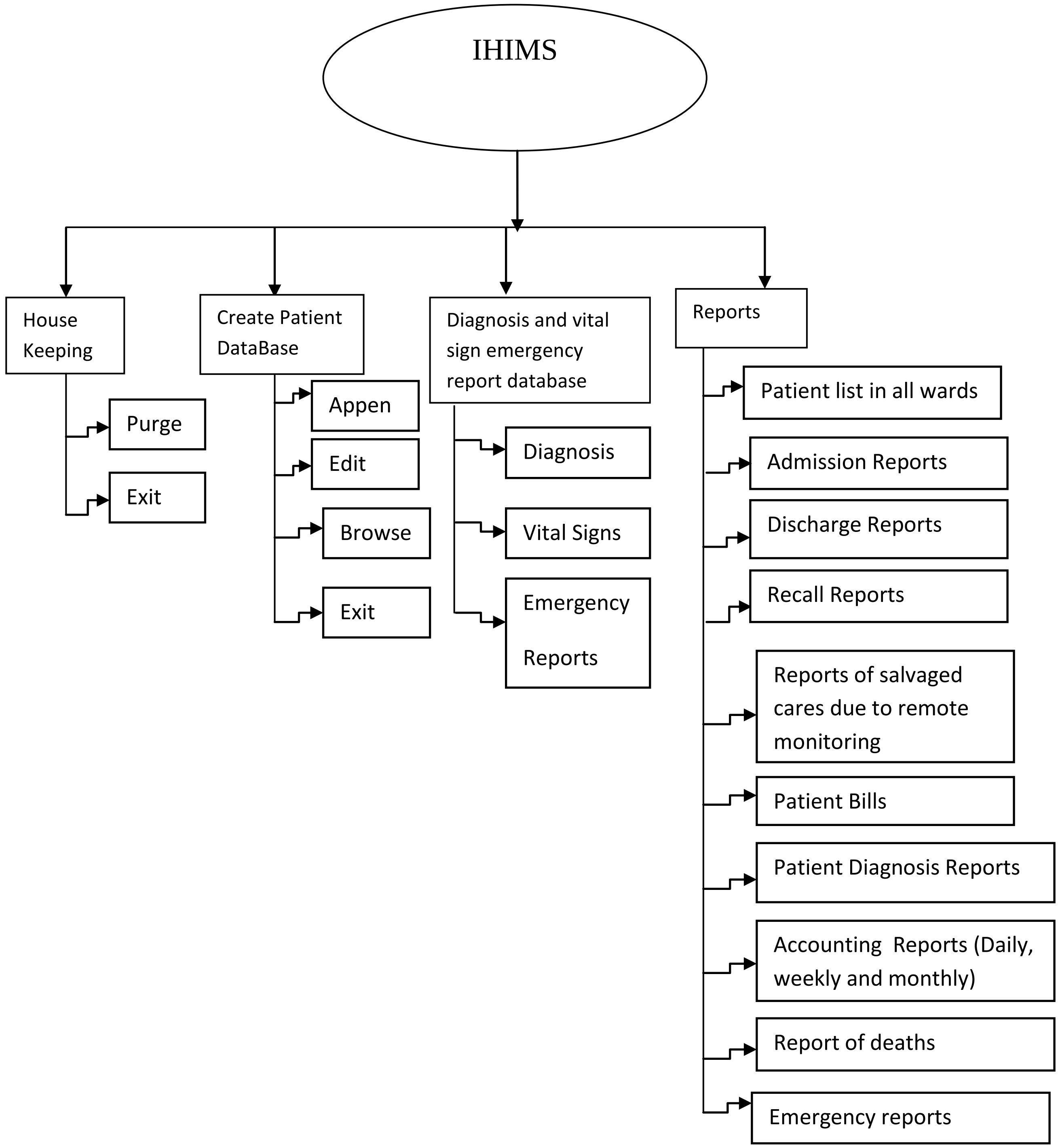


Figure 3.8: High Level Model of the Proposed System (IHIMS)

The two High Level Models are used to demonstrate the two malor ations in this project. One being the Mobile Health Information Management System (MHIMS) and the other being the Integrated Hospital Information Management System (IHIMS).

CHAPTER 4

SYSTEM DESIGN

4.1 Introduction

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

i. Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, which involves a simplistic (and sometimes graphical) representation of an actual system. In the context of systems design, modelling can undertake the following forms, including:

i. Data flow diagrams
ii. Entity Life Histories
iii. Entity Relationship Diagrams
iv. Physical design

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed as output.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc.

4.2 Design Considerations

In the development of the system, certain factors were considered to enhance the optimal performance of the new system. The factors considered include:

i. **Security:** in the course of transferring patients' medical records from one place to another, the records are exposed to great danger because unauthorized users can tamper with the records and falsify them which can have devastating consequences both the hospital and the patients' including the patients' families. In view of this, adequate security measures are considered.

   As the patients' medical records travel from the IPDA (Intelligent Personal Digital Assistant), to the server, the records are encrypted to prevent unauthorized access. This encryption is done using the symmetric key encryption algorithm which uses a 16 bit key for the encryption. The records are decrypted in the server by an authorized user using the same 16 bit key. This prevents unauthorized users from having access to the patients' records.

ii. **Reusability:** A segment of the source code can be used again to add new functionality with slight or no modification. The design of this system is done in such a way that it is not restricted to a particular organization. It can be used in any other hospital.

iii. **Modularity:** This system is designed in modules for easy enhancement. The functionality of the system is separated into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. Conceptually, modules represent a separation of concerns and improve maintainability by enforcing logical boundaries between components. Modules are typically incorporated into the program through interfaces. The elements defined in the interface are detectable by other modules.

Information security is a vital issue in the case of medical applications when the patient data is used for either real time diagnostic purposes or long term analysis of chronic conditions.

4.3 Main Menu

There are two major applications in this project:

i. Client End (Simulated Data)
ii. Server End (which hosts the main program)

4.3.1 The Client End (Simulated Data)

This application is divided into different modules as shown:

Flash Screen Module (Introduces the application)

i. New
ii. Open
iii. Save
iv. Close
v. Exit

Set-up Module

i. Name
ii. Age
iii. Blood pressure type
iv. Smoking Habit
v. Alcohol Intake

Log –In Module

i. Server name
ii. Access Key

Simulation Module

i. Patient Id
ii. Memo 1
iii. Memo 2
iv. Date count
v. Systolic
vi. Time count
vii. Diastolic
viii. Irregular Heart Beat
ix. Pulse rate
x. Telephone Number

4.3.1.1 Physiological Parameters

The physiological parameters that are monitored are blood pressure and pulse rate. Table 4.1illustrates the specifications of the physiological signals being monitored.

Table 4.1: Physiological Parameters of the Patient being monitored

S/N	Physiological Parameter	Specifications
1.	Blood Pressure (BP)	Systolic: 60 -200mmHg Diastolic: 50 – 110 mmHg
2.	Pulse Rate	0 – 200bpm

4.3.1.2 Intelligent Personal Digital Assistant (IPDA)

This is used as a personal server. The personal server interfaces the Wireless Body Area Network (WBAN) nodes using bluetooth. It holds patient authentication information and is configured with the server IP address in order to interface the medical services. It collects physiological vital signals from WBAN, processes them, and prioritizes the transmission of critical data when there is sudden clinical change in the current patient condition and data content for example when the physiological parameters exceeds the ones on the specification.

A simulation is done here which generates the readings for the blood pressure and the pulse rate. This is done by using the known causes for high and low blood pressure and high and low pulse rate. Several factors and conditions may play a role in its development, including:

i. Smoking
ii. Being overweight or obese
iii. Lack of physical activity
iv. Too much salt in the diet
v. Too much alcohol consumption (more than 1 to 2 drinks per day)
vi. Stress
vii. Older age
viii. Genetics
ix. Family history of high blood pressure
x. Chronic kidney disease
xi. Adrenal and thyroid disorders

Of all these factors, the ones used in the simulation were name, age, phone number, smoking habits, alcohol intake, salt intake and weight.

4.3.1.3 Systemic arterial blood pressure

Table 4.2: Systemic arterial blood pressure

Classification of Blood Pressure for Adults		
Category	**Systolic, mmHg**	**Diastolic, mmHg**
Hypotension	< 90	< 60
Desirable	90 – 119	60 – 79
Prehypertension	120 – 139	80 – 89
Stage 1 Hypertension	140 – 159	90 – 99
Stage 2 Hypertension	160 -179	100 – 109
Hypertensive Crisis	≥ 180	≥ 120

i. Mean Arterial Pressure

The mean arterial pressure (MAP) is the average over a cardiac cycle and is determined by the cardiac output (CO), systemic vascular resistance (SVR), and central venous pressure (CVP), (Klabunde, 2007).

$$\mathrm{MAP} = (\mathrm{CO} \cdot \mathrm{SVR}) + \mathrm{CVP}. \qquad (4.1)$$

MAP can be approximately determined from measurements of the systolic pressure P_{sys} and the diastolic pressure P_{dias} while there is a normal resting heart rate, (Klabunde, 2007).

$$\mathrm{MAP} \approxeq P_{\mathrm{dias}} + \frac{1}{3}(P_{\mathrm{sys}} - P_{\mathrm{dias}}). \qquad (4.2)$$

ii. Pulse pressure

The up and down fluctuation of the arterial pressure results from the pulsatile nature of the cardiac output, i.e. the heartbeat. The pulse pressure is determined by the interaction of the stroke volume of the heart, compliance (ability to expand) of the aorta, and the resistance to flow in the arterial tree. By expanding under pressure, the aorta absorbs some of the force of the blood surge from the heart during a heartbeat. In this way, the pulse pressure is reduced from what it would be if the aorta wasn't compliant. (Klabunde, 2007). The loss of arterial compliance that occurs with aging explains the elevated pulse pressures found in elderly patients.

The pulse pressure can be simply calculated from the difference of the measured systolic and diastolic pressures, (Klabunde, 2007).

$$P_{\mathrm{pulse}} = P_{\mathrm{sys}} - P_{\mathrm{dias}}. \qquad (4.3)$$

4.3.1.4 Sensor Configuration

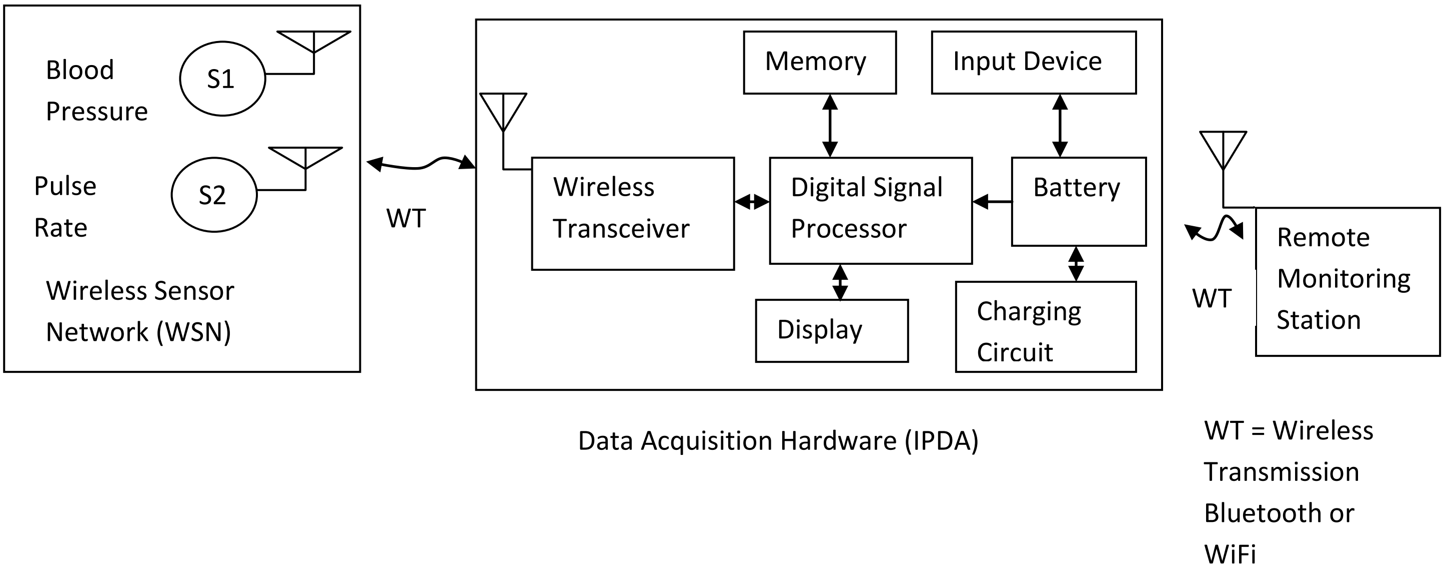


Figure 4.1: Architecture of a Wireless Sensor Network

A Wireless Sensor Network (WSN) consists of base stations and a number of wireless sensors which are simple, tiny in size, inexpensive and a tiny battery powered. The WSN for the physiological monitoring consists of three modules namely sensor nodes, data acquisition hardware and remote monitoring station. Figure 4.2 illustrates the detailed block diagram of the wireless sensor network.

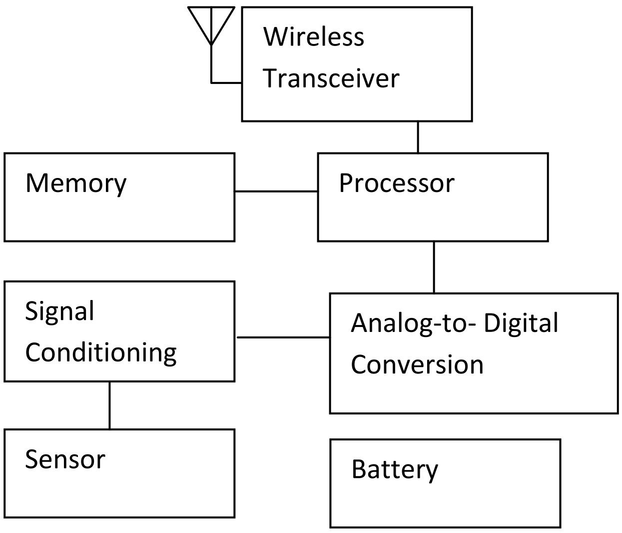


Figure 4.2: Block diagram of the wireless sensor node

4.3.1.5 Sensor Node

The sensor nodes are responsible for acquiring the physiological data and transmitting it to the data acquisition hardware (IPDA). The sensor nodes are strategically placed at various locations on the patient to acquire and transmit the physiological data to the data acquisition hardware (IPDA).

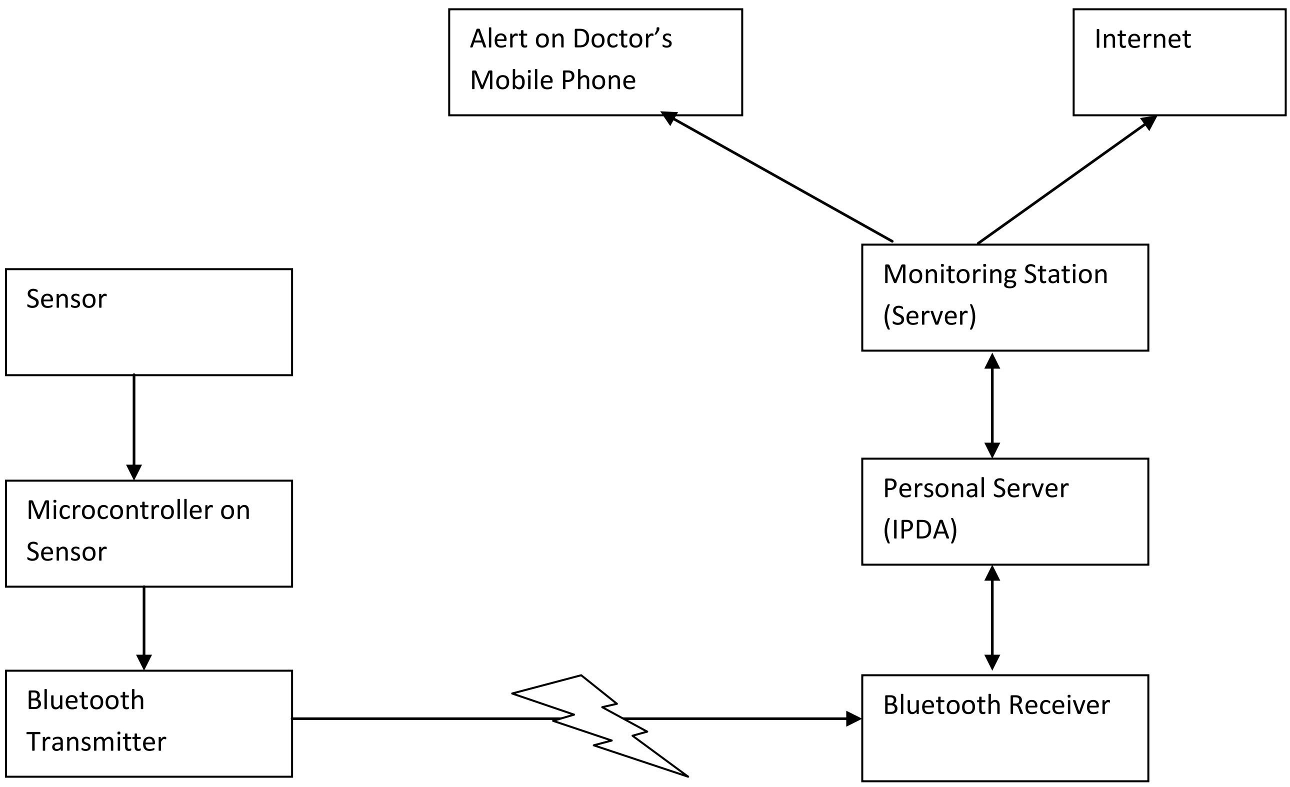


Figure 4.3: Transmission Flowchart

Sensor: It is a sensing chip to sense physiological data from the patient's body. The results of the sensor are simulated using known causes for normal and abnormal blood pressure and pulse rate.

Microcontroller: it is used to perform local data processing such as data compression and it also controls the functionality of other components in the sensor node.

Bluetooth Transmitter: this transmits the signal from the sensor to the IPDA.

Bluetooth Receiver: This is responsible for receiving the signal from the sensor. The Bluetooth receiver is in the IPDA.

4.3.1.6 Priority Scheduling and Data Compression

Table 4.3: Priority scheduling

Physiological Signs	Data Rate	Latency	Priority
Blood Pressure	High	Low	1
Pulse Rate	Low	High	2

4.3.1.7 Interpretation of Results

Blood pressure measurement gives two readings:

i. Systolic BP – the 'top'value, which indicates the force exerted against the walls of the arteries by the blood as it is being pumped from the heart. The normal range for this value would be around 110 -140 millimetres of mercury (mmHg).

ii. Diastolic BP – the 'lower'value, which indicates the pressure exerted by the elastic recoil of the arteries during the relaxation phase between heart beats.

The normal range for diastolic BP would be around 60 – 90mmHg.

4.3.1.8 Blood Pressure Ranges

Table 4.4: Blood Pressure Ranges

	Low	Normal	Borderline	High
Systolic	< 100	100 – 140	141– 160	>160
Diastolic	< 60	60 – 90	91 – 110	>110

Blood pressure will fluctuate throughout the day, and sometimes even fluctuate between readings taken within minutes of each other. This could be due to factors such as the anxiety and apprehension of having the blood pressure actually taken.

4.3.1.9 The IPDA Software

The software that runs on the IPDA is called the **Mobile Health Management Information System** and the structure is shown in the figure below:

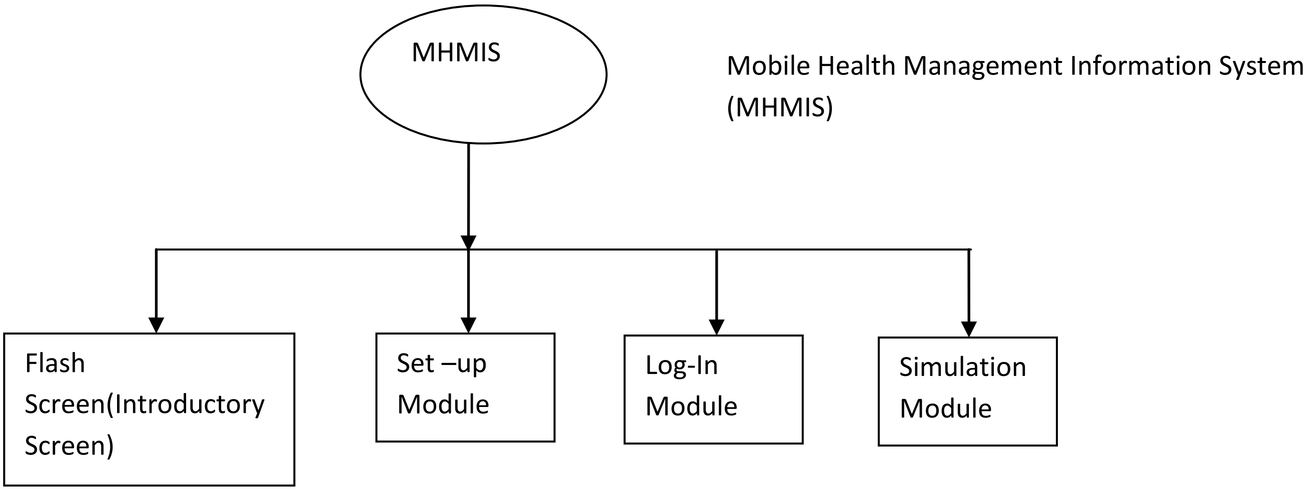


Figure 4.4: Mobile Health Information Management System Structure

4.3.1.10 Features of the Mobile Health Information Management System

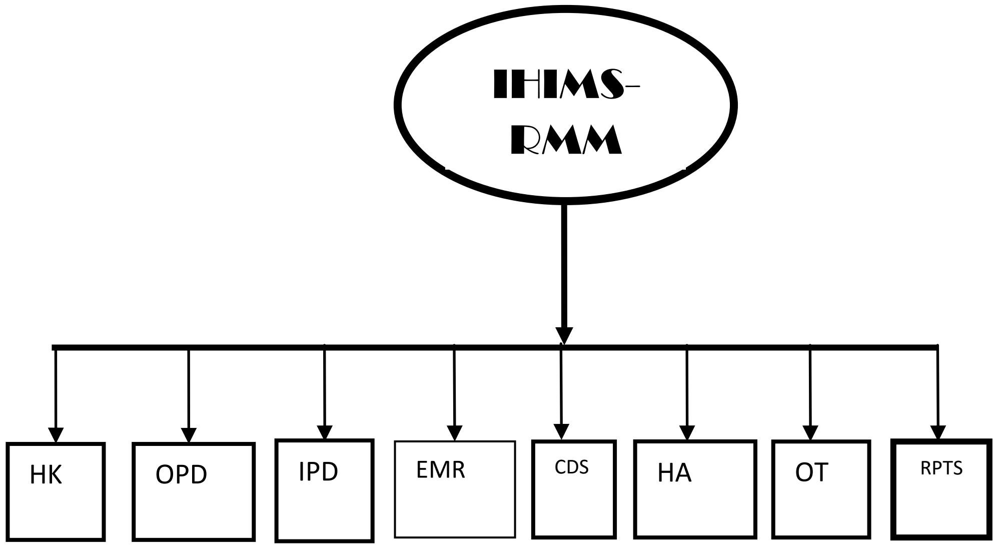
i. MHIMS is a complete wireless application
ii. It generates the simulated vital sign data (blood pressure and pulse rate)
iii. It automatically sends the simulated data to the server through the wireless router or the Internet.
iv. It analyses the results and stores them in its memory
v. It uses a Bluetooth connection to send the result to the server.
vi. The Bluetooth connection can cover a distance of 10 meters.

4.3.2 The Server End

The server end hosts the main program. This application is divided into different modules as shown:

Main MODULES (Control Centre)

- Out Patient Department (OPD)
- In-Patient Department (IPD)
- Clinical Support
- Hospital Administration
- Ancillary Services
- Operation Theatre
- EMR
- CDSS



NOTES: HK = House keeping which will involve purging of out-of-date data

OPD = Out Patient Department Module
IPD = In-Patient Department Module
CS = Clinical Support Module
HA = Hospital Administration Module
AS = Ancillary Services Module
OT = Operation Theatre Module
Exit = Exit module. Exit to Windows or Dos
EMR = Medical Records software system that keeps track of patient's history
CDS = Computerized Decision Support System

Figure 4.5: The Control Centre

4.3.2.1 Overview of the Features and Modules

4.3.2.2 Out Patient Department (OPD)

Appointment Scheduling

Reception & Enquiry

OPD Registration

Test Booking

Pathological Test

Radiological Test

Billing (OPD)

4.3.2.3 In-Patient Department (IPD)

i. Registration
ii. Discharge
iii. Ward Management
    i. Bed Transfer
    ii. Daily Update (Billing)
    iii. Monitor Patient
    iv. Lab Test Booking
    v. OT Booking
    vi. Allergy History
    vii. Social History
    viii. Case History
    ix. Prescription
    x. Medicine
iv. Billing (IPD)
v. Doctor Master
vi. TPA

4.3.2.4 Clinical Support

i. Laboratory
ii. Electronic Medical Record (EMR)
iii. Operation Theatre
iv. Duty Roaster

4.3.2.5 Hospital Administration

i. Stock Management
ii. MIS
iii. Security & Admin

4.3.2.6 Operation Theatre

i. OT Scheduling
ii. Procedure Master
iii. OT Master
iv. OT Booking
v. OT Patient details
vi. OT Notes
vii. OT Billing

4.4 Reports Production Module with customization

Patient Admissions Master List – This class will produce a report with the following attributes or data dictionary (Patient Registration Module)

i. Patient Hospital No. (Key Identifier)
ii. Full name of patient, surname, middle name and first name.
iii. Address details: Home and Office
iv. Telephone no.
v. Email address
vi. Age
vii. Gender

viii. Nationality
ix. Religion
x. Disease diagnosed on admission
xi. Date of Admission
xii. Doctor to which patient is assigned
xiii. Total bill accumulated to date
xiv. Total payment to date
xv. Balance outstanding
xvi. Treatment administered (eg. Surgery, physiotherapy etc.)
xvii. Current status of health
xviii. Projected date of Discharge
xix. Personal attendee name and details:

4.5 Emergency And Doctor's Reaction Report From Remote Medical Monitoring

i. Patient Hospital Identity
ii. Location and Ward
iii. Name
iv. Nature of Emergency
v. Doctors Reaction
vi. Current Health Condition

4.6 Hospital Discharge List

i. Patient Hospital No. (Key Identifier)
ii. Full name of patient
iii. Date Admitted
iv. Health status on discharge
v. Planned visit schedules
vi. Statement of account status (balance outstanding if any)
vii. Doctors comment

4.6.1 **CDSS (**Computerized Decision Support System)

The need of an automated analysis, decision and response in case of an emergency scenario leads to the inclusion of a CDSS. CDSS is a piece of software, which analyses the patient's physiological data (e.g. ECG, blood pressure, body temperature etc.) in order to find out symptoms of any abnormality. These symptoms are used by the CDSS to estimate the current health situation of the patient. The decision support system is also capable of making decisions based on the diagnosis of estimated health situation

4.6.2 EMR is Electronic Medical Records software *system* that keeps track of patient's history. Integrated with CDSS, it provides initial data (previous medical record of concerned patent) to the CDSS (Computerized Decision Support System) for comparison with current data. EMR stores the new data as well as the results of the analysis performed by CDSS.

4.7 **Other Important Reports from which a Choice may be Made**:

i. Canned reports
ii. Patient/ patient class registration detail & summary
iii. Department/service header booking/service provided
iv. Doctor/Diagnostic test/Misc. service rates & summary
v. With/without names
vi. List of services available
vii. Bed Occupancy Report
viii. Patient/patient class registration detail & summary
ix. Day count of diagnostic tests
x. Patient dues/discount given/Invoice details
xi. Department/ service/patient revenue statement

4.7.1 Features Of The Integrated Hospital Management System (IHMS)

i. Proposed Model is a fully integrated online system.
ii. IHMS supports multi currency system, which is converted to the base currency for accounting purposes.
iii. IHMS supports multi location implementation where the hospital has outpatient, Inpatient and other services in many physical locations.
iv. Doctors can get the investigation reports online at their terminal (or any terminal) without waiting for the physical report reaching them. Doctors can see the previous investigation results for a patient for better diagnosis.
v. Doctors can view their daily appointment schedules for any day from any terminal.
vi. IHMS is highly secure and flexible. It allows the administrator to configure different information access privileges for different individuals depending on their role in the Hospital.
vii. Inventory valuation is done in FIFO (First In First Out) with earliest expiry date in Pharmacy.
viii. IHMS provides the following reports and more at the click of button daily cash collections, patient turnover, billing and medical records related reports etc which enables management to take better decisions
ix. GIS/GPRS is deployed to identify the location of a patient in a distributed Community-based system. The patient's location is identified via a map.

4.8 Output Specification

4.8.1 Output Format

The output format shows exactly how the output form of the program will look like. Tables 4.5 to 4.11 shows the format of the output of the two applications. While Table 4.5 shows the output of the Mobile User Log File which gives a tabular view of the output format of the Mobile Health Information Management System, Tables 4.6 to 4.11 gives the output format of the second application, the Integrated Hospital Information Management System.

4.8.2 Mobile Health Information Management System

Mobile User Log File

As at 99.99.9999

Table 4.5: Mobile User Log File

Patient Name	Telephone No	Patient Id	Systolic Pressure	Diastolic Pressure	Pulse Rate	Suggestions
Okeke Ngozi	08012345678	PID001	200	100	80	High Blood pressure/High Pulse rate
Ojiewulu Nneka	09023415678	PID002	160	110	90	High Blood Pressure/High Pulse rate
Adebayo Olufemi	08023418769	PID003	110	70	60	Normal Blood Pressure/Normal Pulse Rate

4.8.3 REPORTS (OUTPUT FORMATS)

PATIENTS ADMISSION MASTER LIST REPORT

Automated Intelligent Hospitals Ltd

Patient Admissions Master List

as at 99.99.999

Table 4.6: Patient Admission Master List

Hospital Id	Patient Name	Address	Tel	Email	Age	Gender	Date Admitted	Diagnosis	Ward	Bed No	Doctor In C
HSC021	Johnson P	9 Mbonu Str	0803	drosu@ymai	60	M	1/1/10	Malaria	Male	22	Dr Cynthia
HSC022	Dayo A	19 Nwado Str	0803	mrn@ymail	30	F	14/2/10	HIV	Female	26	Dr Ohu
HSC023	Oblor O	20 MCC Rd	0803	odro@ymail	26	M	16/4/10	Hepatis B	Male	28	Dr Osuagwu
HSC024	Blessing E	16 Igirita Str	0803	blosu@ymai	31	F	18/5/10	Malaria	Female	29	Dr Kens
HSC025	Uche C	9 Ojike Str	0803	uchu@ymail	32	M	21/6/10	Cholera	Male	30	Dr Obe

4.8.4 REMOTE MEDICAL REPORTS FROM IPDAs

Automated Intelligent Hospitals Ltd

Remote Medical Reports from PDAs

as at 99.99.9999

Table 4.7: Remote Medical Reports from IPDA

Hospital Id	**Patient Name**	**Address**	**Tel**	**Email**	**Age**	**Gender**	**Date Admitt**	**Diagnosis**	**Ward**	**BedNo**	**Doctor-In(**	**Emergency Report**
HSC021	Johnson P	9 Mbonu Ojik	080:	drosu@ymail	60	M	1/1/10	Malaria	Male	22	Dr Cynthia	BP = 250/150
HSC022	Dayo A	19 Nwado Str	080:	mrndo@ymail	30	F	14/2/1(	HIV	Female	26	Dr Ohu	Acute Head ach
HSC023	Oblor O	20 MCC Rd	080:	modrosu@yma	26	M	16/4/1(	Hepatis B	Male	28	Dr Osuagv	Blood Sugar = 400mpd
HSC024	Blessking I	16 Igirita Str	080:	bldrosu@ymail	31	F	18/5/1(	Malaria	Female	29	Dr Kens	Stroke
HSC025	Uche C	9 Mbonu Ojik	080:	uchdrosu@yma	32	M	21/6/1(	Colera	Male	30	Dr Obe	Quick breath

4.8.5 **Automated Intelligent Hospitals Ltd**

Hospital Discharge Report

as at 99.99.9999

Table 4.8: Hospital Discharge List

Hospita Id	**Patient Name**	**Address**	**Tel**	**Email**	**Age**	**Gende**	**Date Admi**	**Diagnos**	**Ward**	**Bed No**	**Doctor -In C**	**Dischar Date**	**Recall**	**Bill outstand**	**Date Discha**
HSC021	Johnson F	9 Mbonu O	0803	drosu@ymail	60	M	1/1/10	Malaria	Male	22	Dr Cyn	2/2/10	30/2/10	N20,000	3/3/10
HSC022	Dayo A	19 Nwado S	0803	mrndo@ymai	30	F	14/2/10	HIV	Femal	26	Dr Ohu	30/3/10	30/4/10	N1500	10/6/10
HSC023	Oblor O	20 MCC R	0803	modrosu@ym	26	M	16/4/10	Hepatitis	Male	28	Dr Osu	1/5/10	1/6/10	N7000	13/7/10
HSC024	Blessking	16 Igirita S PHonu Ojil	0803	bldrosu@yma	31	F	18/5/10	Malaria	Femal	29	Dr Ken	30/6/10	30/7/10	N12000	20/8/10
HSC025	Uche C	9 Mbonu O	0803	uchdrosu@ym	32	M	21/6/10	Cholera	Male	30	Dr Obe	20/7/10	30/8/10	Nil	30/9/10

4.8.6 **Automated Intelligent Hospitals Ltd**

Hospital Drug Inventory Report

as at 99.99.9999

Table 4.9: Hospital Drug Inventory Report

Drug Id	**Description**	**Qty Purchased**	**Date Purchased**	**Qty Used up**	**Stock Bal**	**Reorder Level**	**Location**	**Naira Value Of Bal**	**RMKS**
PaC021	|PanadolExtra	100 pks	1/1/10	80	20 pks	25	Store A	N200	In quick demand
Ana022	Anagine 500 r	300 pks	6/2/10	250	50	40	Store B	N2000	Sell sluggish
Supra023	Supradeem	20 pks	10/3/10	12	8	10	Store C	\N180	In quick demand
InjC024	Injection nidle	16 pieces	26/4/10	10	6	7	Store D	N110	To reduce Reorde Level
SplkinC025	Splina	20 bottles	2/11/10	18	2	7	Store F	N105	Direct Markting

4.8.7 **Automated Intelligent Hospitals Ltd**

Hospital Equipment Listing Report

as at 99.99.9999

Table 4.10: Hospital Equipment Listing Report

ME Id	Description	Qty Purchased	Date Purchased	Qty Sold	ME Bal	Status of ME	Location	Naira Value Of Bal
XrayPaC021	Xray Machine	100	1/1/10	80	20	All Ok	Store A	N200
BPAna022	BP Checker	300	6/2/10	250	50	40 OK	Store B	N2000
ECGra023	ECG	20	10/3/10	12	8	All Ok	Store C	N180
EyeT024	Electric Tough	16	26/4/10	10	6	All Ok	Store D	N110
BSCC025	Blood Sugar Check	20 bottles	2/11/10	18	2	1 ok 1 faulty	Store F	N105

4.8.8 **Automated Intelligent Hospitals Ltd**

Referral Hospitals Master Listing Report

as at 99.99.9999

Table 4.11: Referral Hospital Master Listing

Hosp Id	**Name**	**Location/ Address**	**Tel No.**	**Email Adress**	**URL WWW**	**SpecializtionU**	**Ownership**
BS021	Bright Hosp Ltd	Igirita	08037101792	80	www.yx.com	Brain Surgery	private
CAN022	GenderEq Hosp Ltd	Buguma	08037101793/	250	www.xlin.com0	Cancer	Religious bod
HTR023	Sick Bay Ltd	GRA	08067101792	12	www..TechH.vom	Heart Transplai	publiv
EyeT024	Ekeniia Hosp. Ltd	Artilary	08077101793	10	www.BHX.com	Optometry	Civil servants
TropD025	Heart Foundation L	Owerri Rd	08057101792/	18	www.EzuhuH.cor	Tropical Diseas	Club

4.8.9 **Automated Intelligent Hospitals Ltd**

Medical Experts Data Base Listing Report

as at 99.99.9999

Table 4.12: Medical DataBase Expert Listing Report

Expert Id	**Name**	**Hospital**	**Location/ Address**	**Tel No.**	**Email Adress**	**URL WWW**	**SpecializtionU**	**Quaslif.**
BS021	Prof. E Osuagw	Bright Hosp Ltd	Igirita	08037101792	profosuagwu@yshoo.co	www.yx.com	Brain Surgery	M.D, FWACS
CAN022	Prof Nwachukw	GenderEq Hosp L	Buguma	0803710179 31	profnwa@yahoo.com	www.xlin.com0	Cancer	MD. RCS Lon
HTR023	Prof Anigbogu	Sick Bay Ltd	GRA	0806710179 2	profaniwu@yshoo.	www..TechH.vo	Heart Transplan	MD. FWACS
EyeT024	Prof Onibere	Ekeniia Hosp. Ltd	Artilary	0807710179 3	profoni@yshoo.	www.BHX.com	Optometry	MD. FICSs
TropD025	Prof Chiemeke	Heart Foundation	Owerri Rd	0805710179 2/	profchimeke@yshoo.	www.EzuhuH.co	Tropical Diseas	MD. FACS

4.9 The Modules

This consists of the different forms in the hospital. It consists of the Out Patient Department (OPD) module, In Patient Department (IPD) module, the Clinical support module, hospital administration, ancilliary services, operation theatre, EMR, Expert services, drug services, mechanical equipment and hospital registration.

4.9.1 In - Patient Department/Out Patient Department

This is the in patient/out patient department module where the patient's personal information is entered.

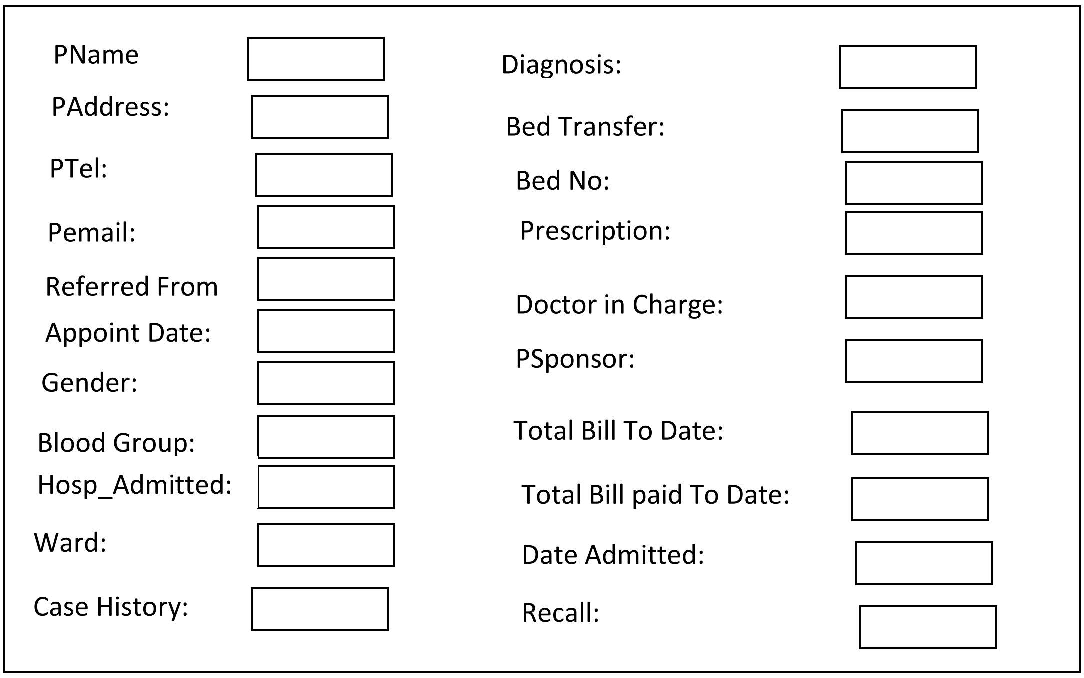


Figure 4.6: IPD/OPD Module

4.9.2 CLINICAL_SUP

This is the clinical support module which consists of the staff name, function, ward,

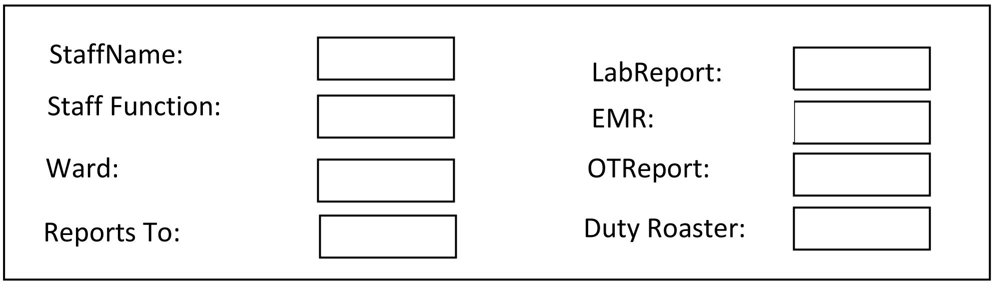


Figure 4.7: Clinical Support Module

4.9.3 Hospital_Adm

This is the hospital administration module which consists of staff name, rank, function and chain of control.

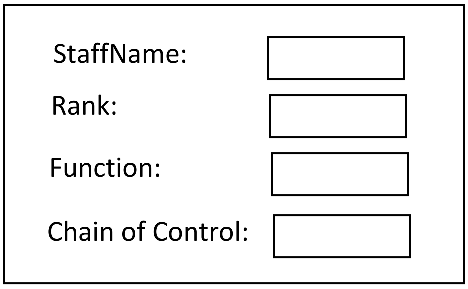


Figure 4.8: Hospital Administration Module

4.9.4 AncillaryServices (Diagnostic Module)

This is the diagnostic module which takes care of the laboratory investigations.

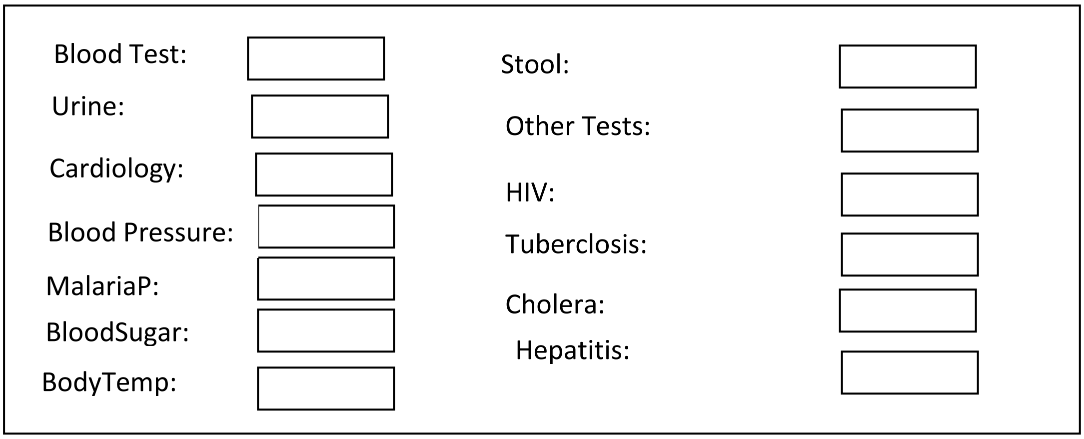


Figure 4.9: Ancilliary Services Module

4.9.5 OPT

This is the operation theatre module which is where surgery details of the patient are kept.

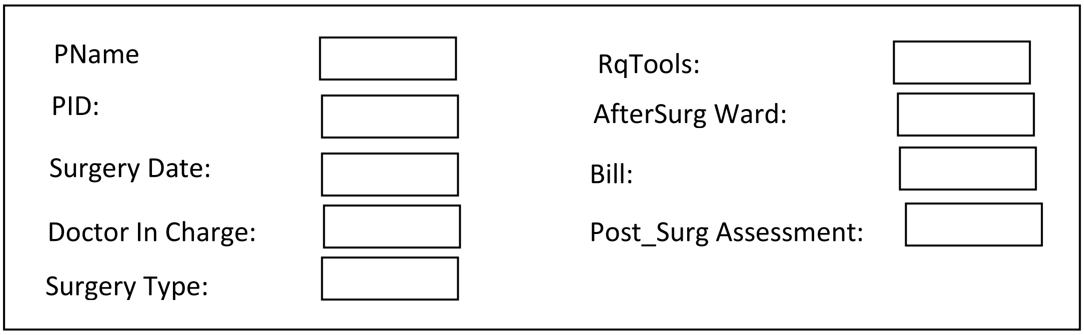


Figure 4.10: Operation Theatre Module

4.9.6 EMR

This is the Electronic Medical Record where the electronic records of patients are kept.

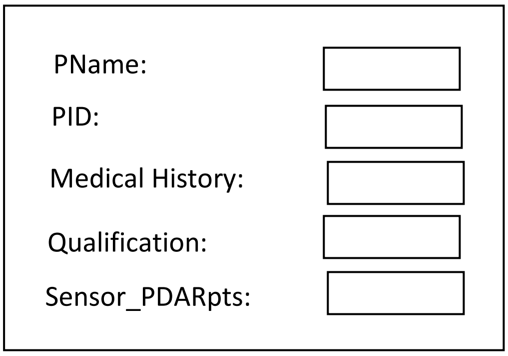


Figure 4.11: Electronoic Medical Records Module

4.9.7 EC

This is the Experts module and is where the list of the experts in the hospital are kept. It contains the details of the experts and the days they are available in the hospital.

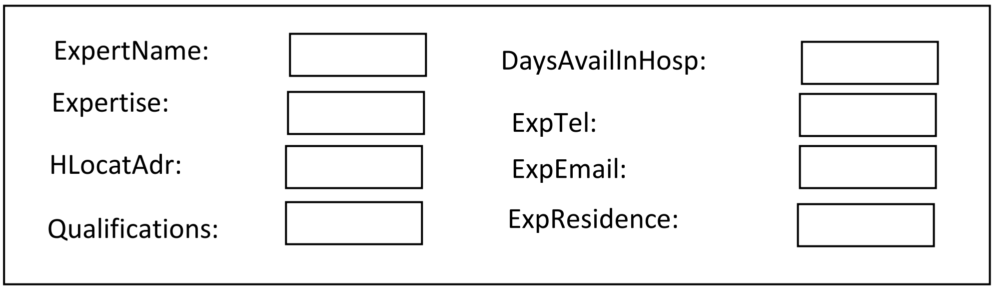


Figure 4.12: Experts Module

4.9.8 HR

This is the hospital registration module that takes care of the registration of a new hospital.

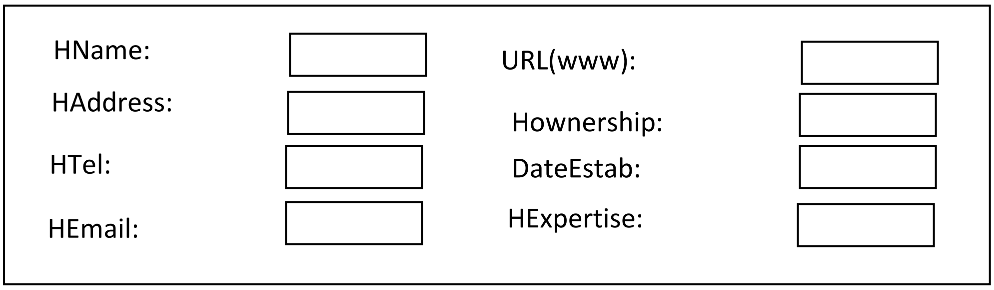


Figure 4.13: Hospital Registration Module

4.9.9 ME

This is the Mechanical Equipment module that is responsible for keeping records of all the mechanical equipment in the hospital.

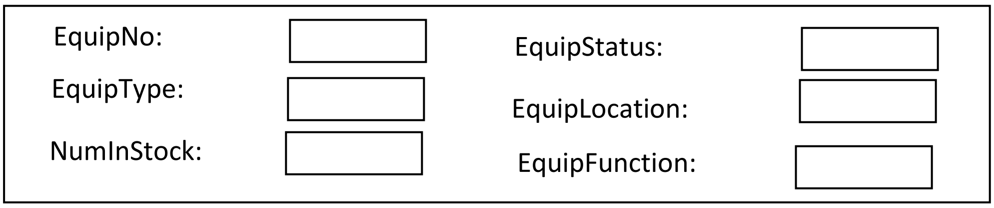


Figure 4.14: Mechanical Equipment Module

4.9.10 DRUG

This is the drug module where all the records of the drugs are kept.

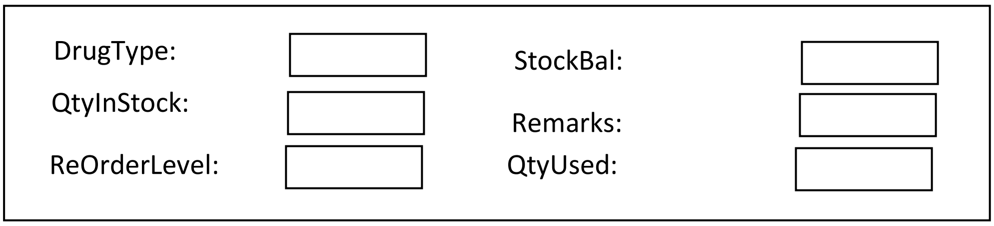


Figure 4.15: Drug Module

4.10 THE FLOWCHARTS

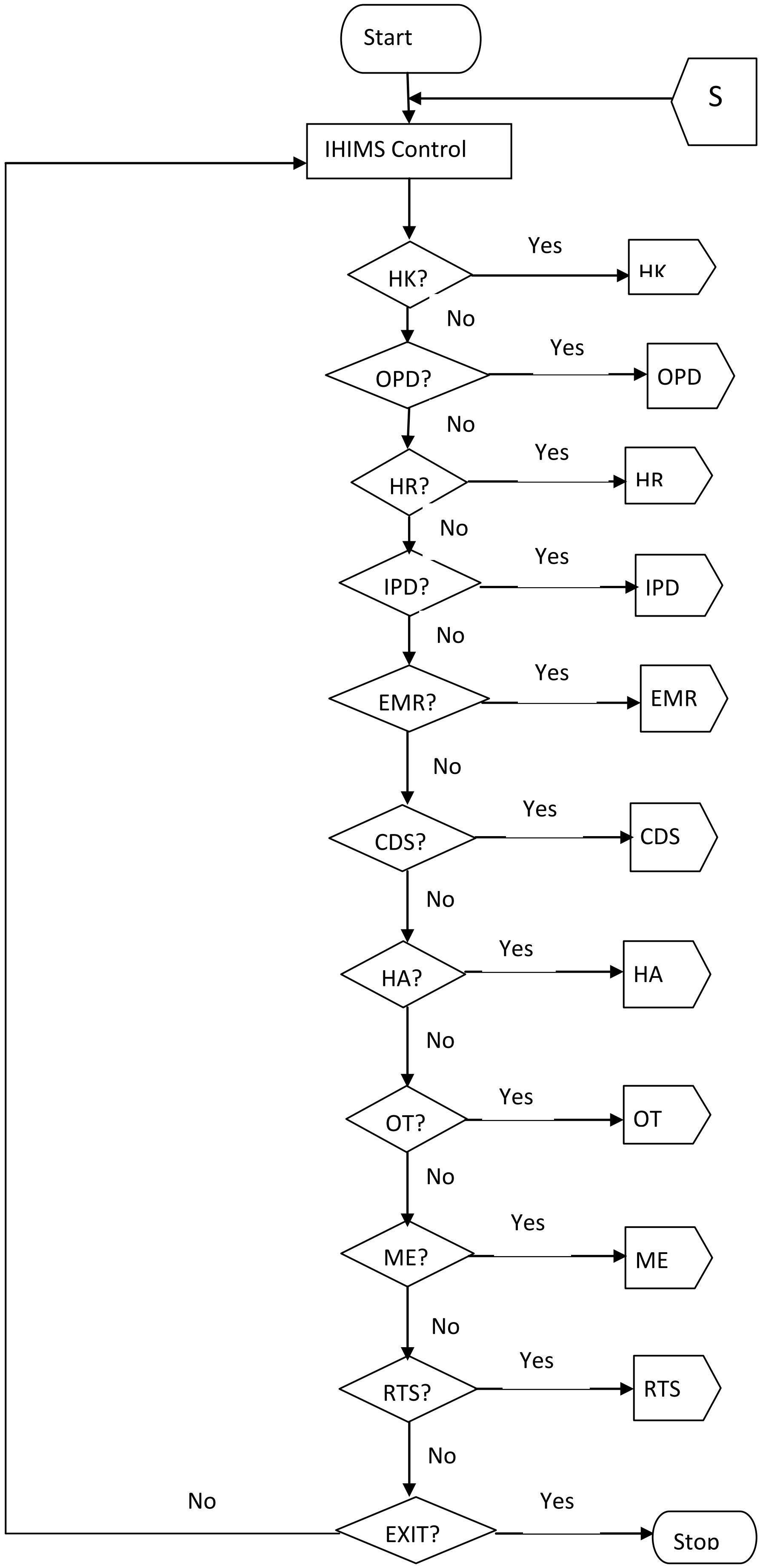


Figure 4.16: IHIMS Control Flowchart

4.10.1 HK (HOUSE KEEPING)

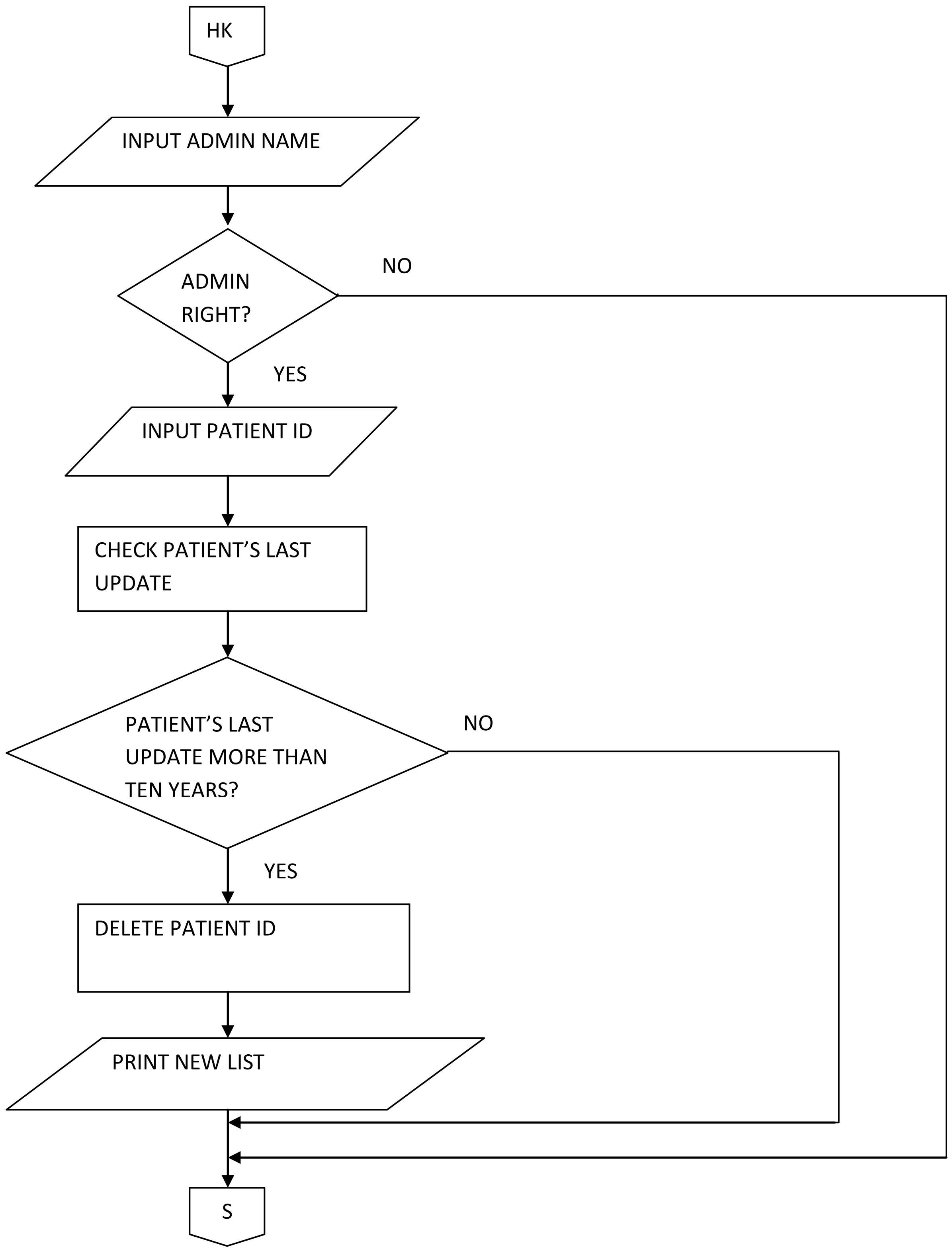


Figure 4.17: House Keeping Flowchart

4.10.2 OPD (OUT PATIENT DEPARTMENT)

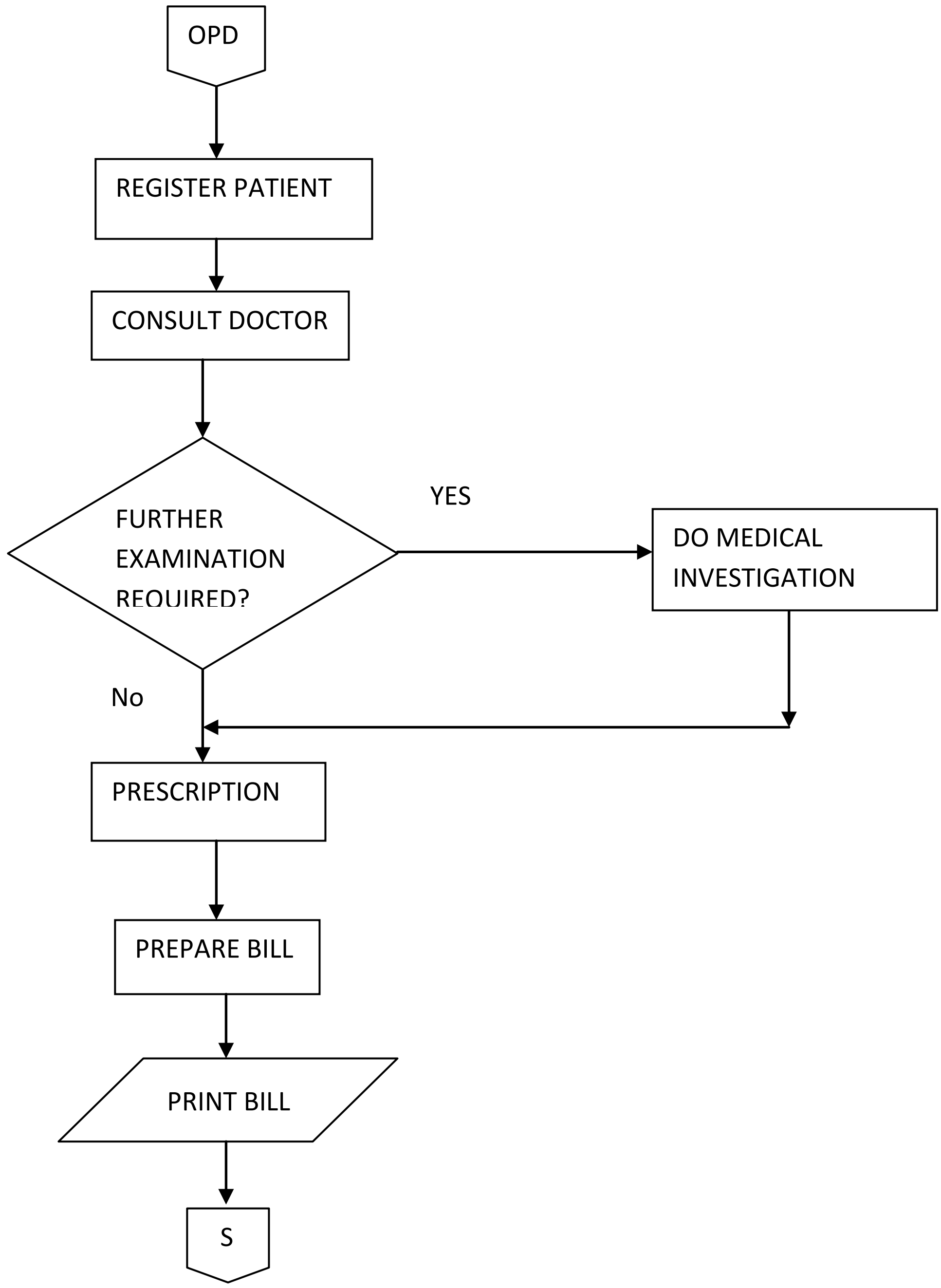


Figure 4.18: Out-Patient Department Flowchart

4.10.3 HR (HOSPITAL REGISTRATION)

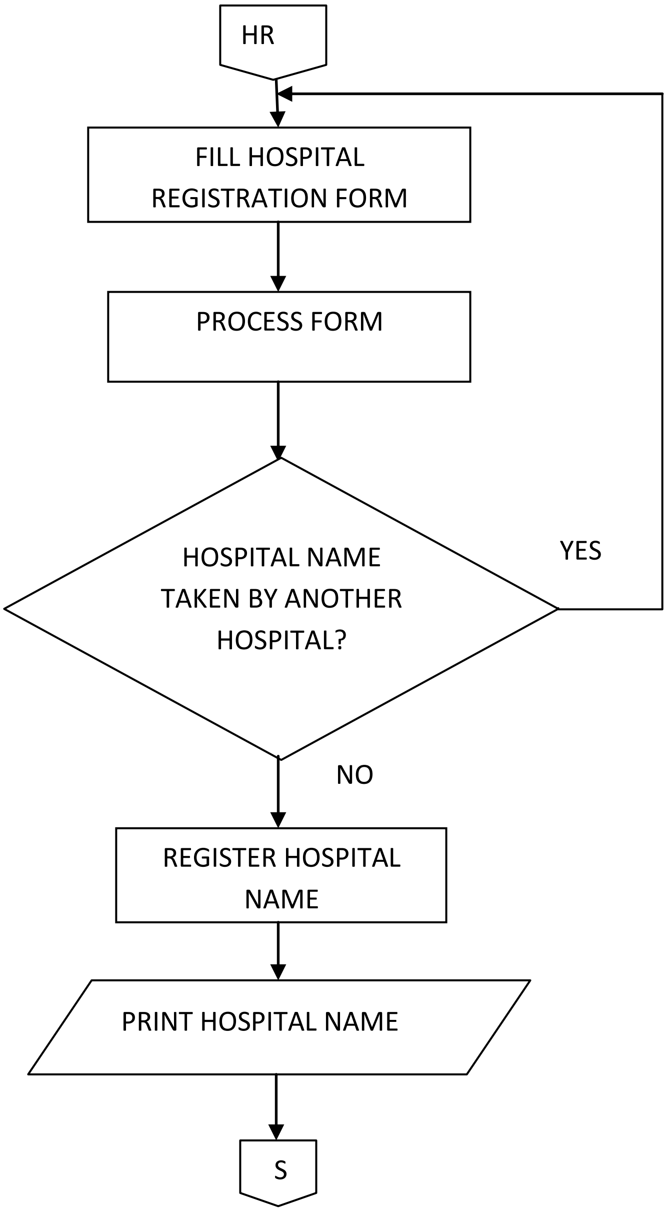


Figure 4.19: Hospital Registration Flowchart

4.10.4 IPD (IN PATIENT DEPARTMENT)

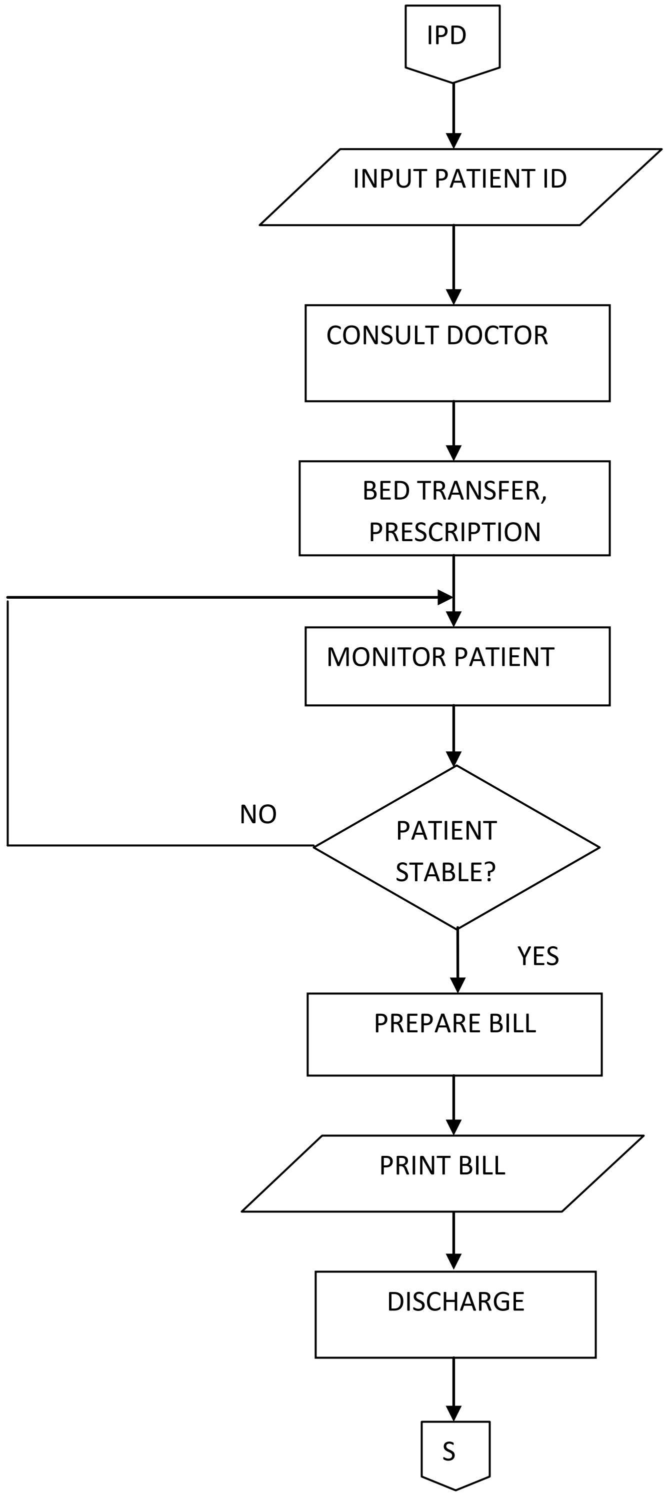


Figure 4.20: In-Patient Department Flowchart

4.10.5 EMR (ELECTRONIC MEDICAL RECORDS)

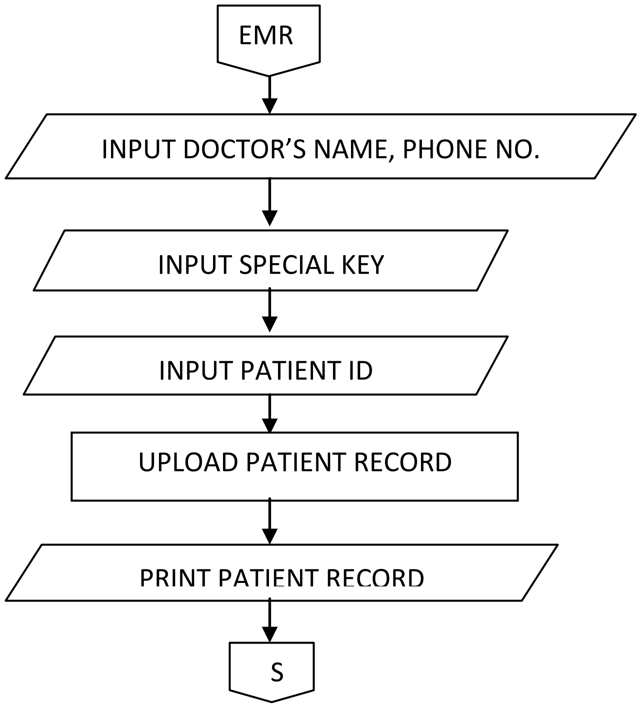


Figure 4.21: Electronic Medical Record Flowchart

4.10.6 CDSS (Computerized Decision Support System)

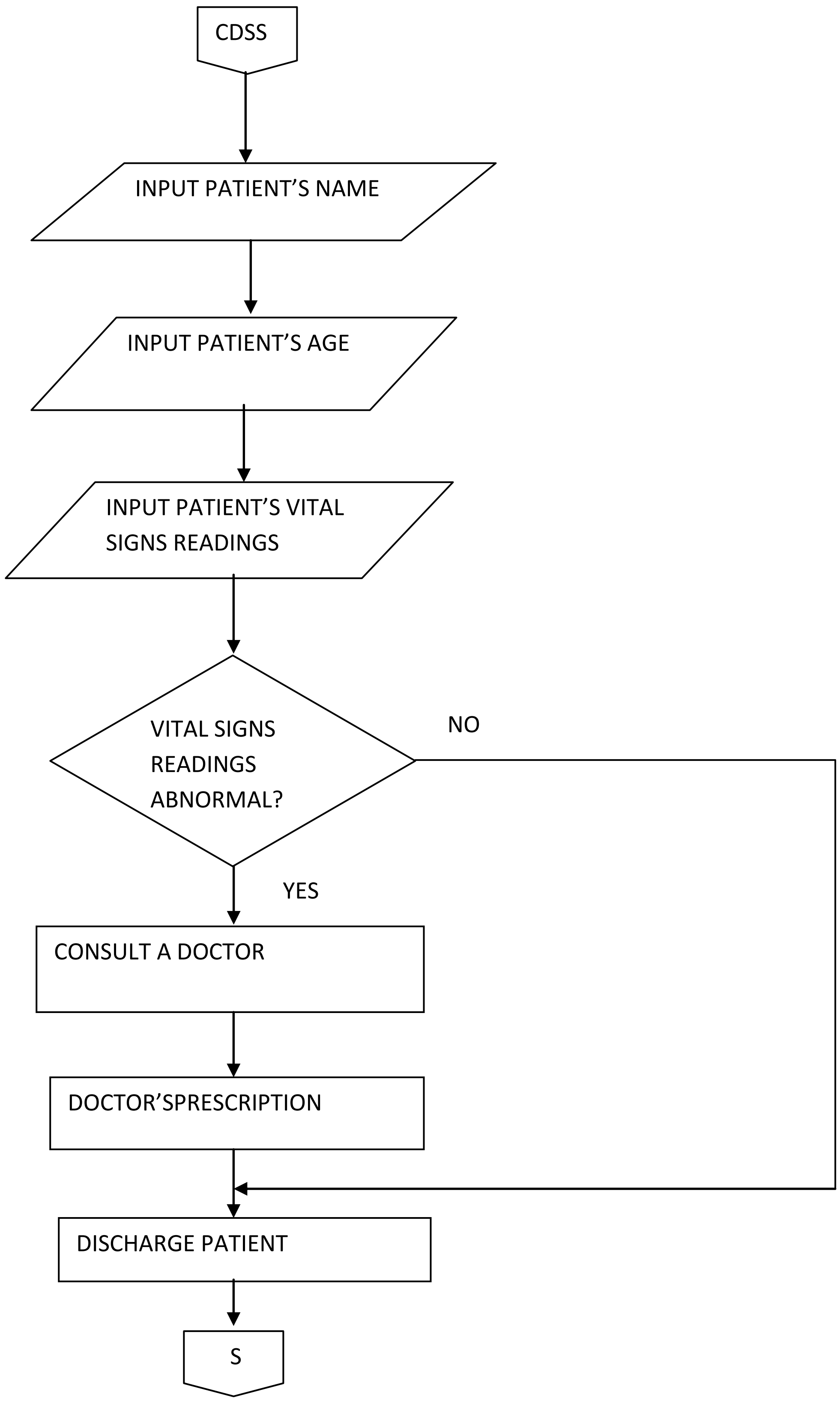


Figure 4.22: Computerized Decision Support System Flowchart

4.10.7 OT (OPERATION THEATRE)

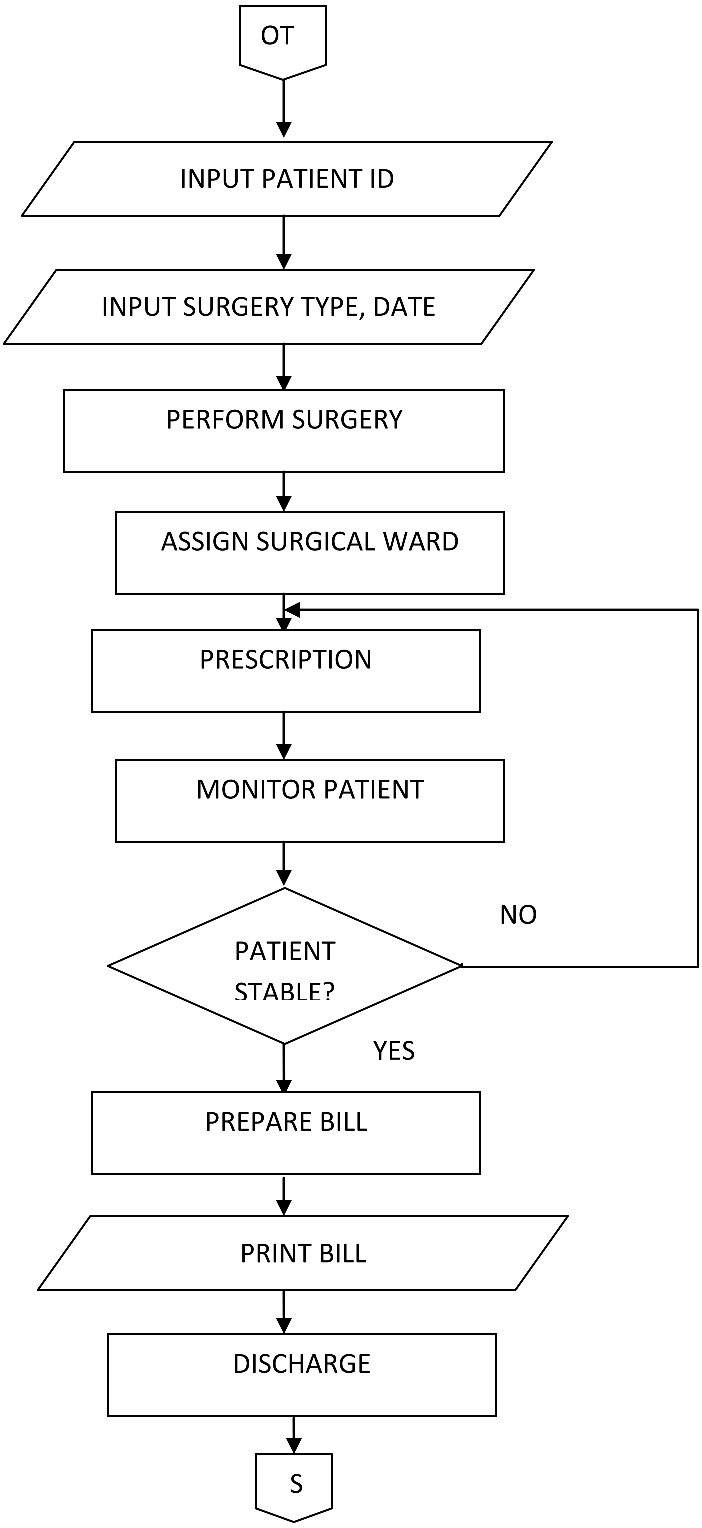


Figure 4.23: Operation Theatre Flowchart

4.10.8 HA (HOSPITAL ADMINISTRATION)

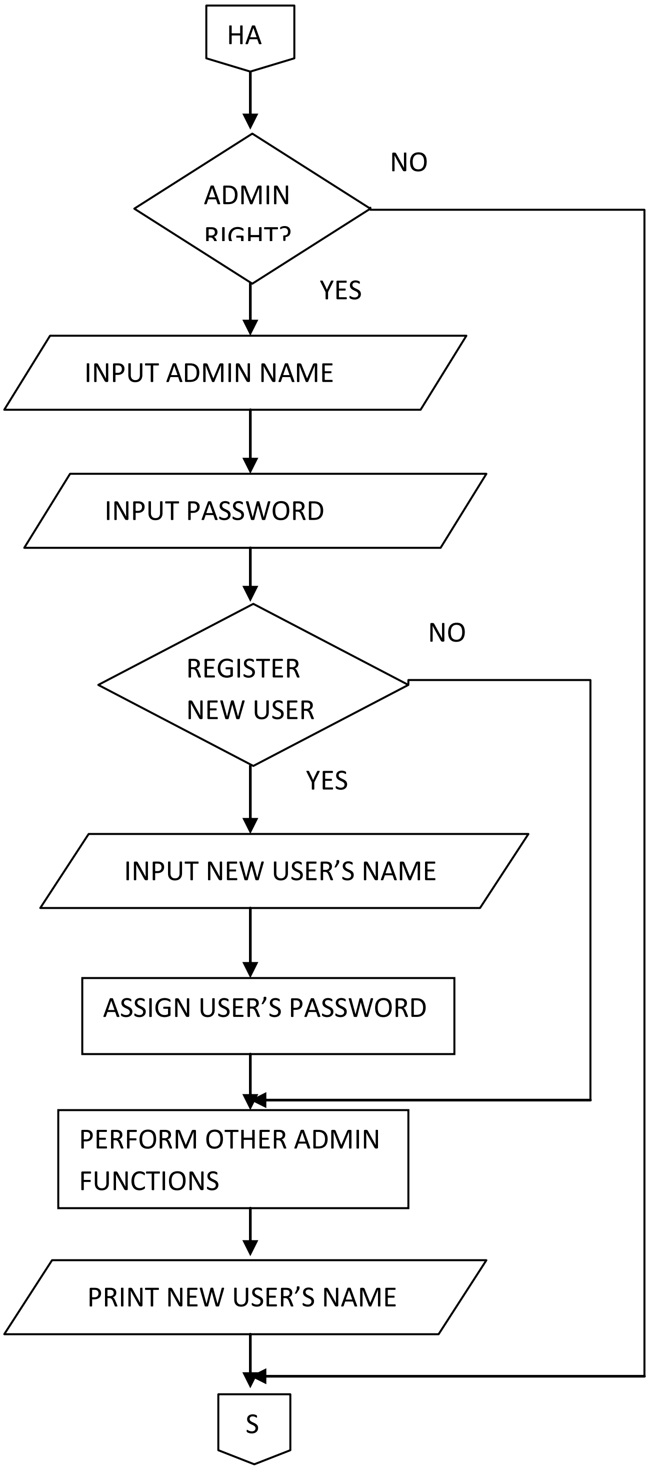


Figure 4.24: Hospital Administration Flowchart

4.10.9 ME (MECHANICAL EQUIPMENT)

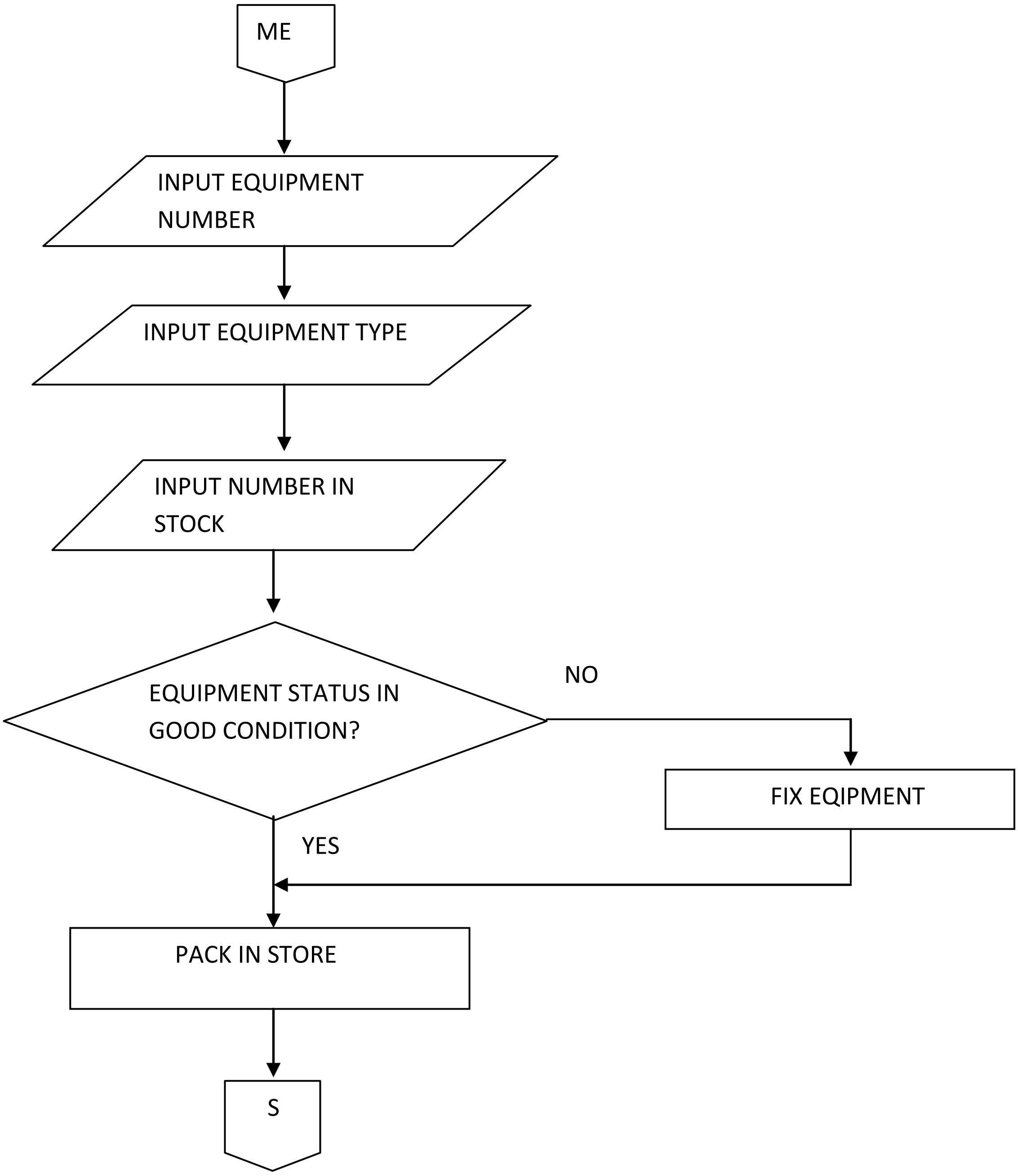


Figure 4.25: Mechanical Equipment Flowchart

4.10.10 REPORTS

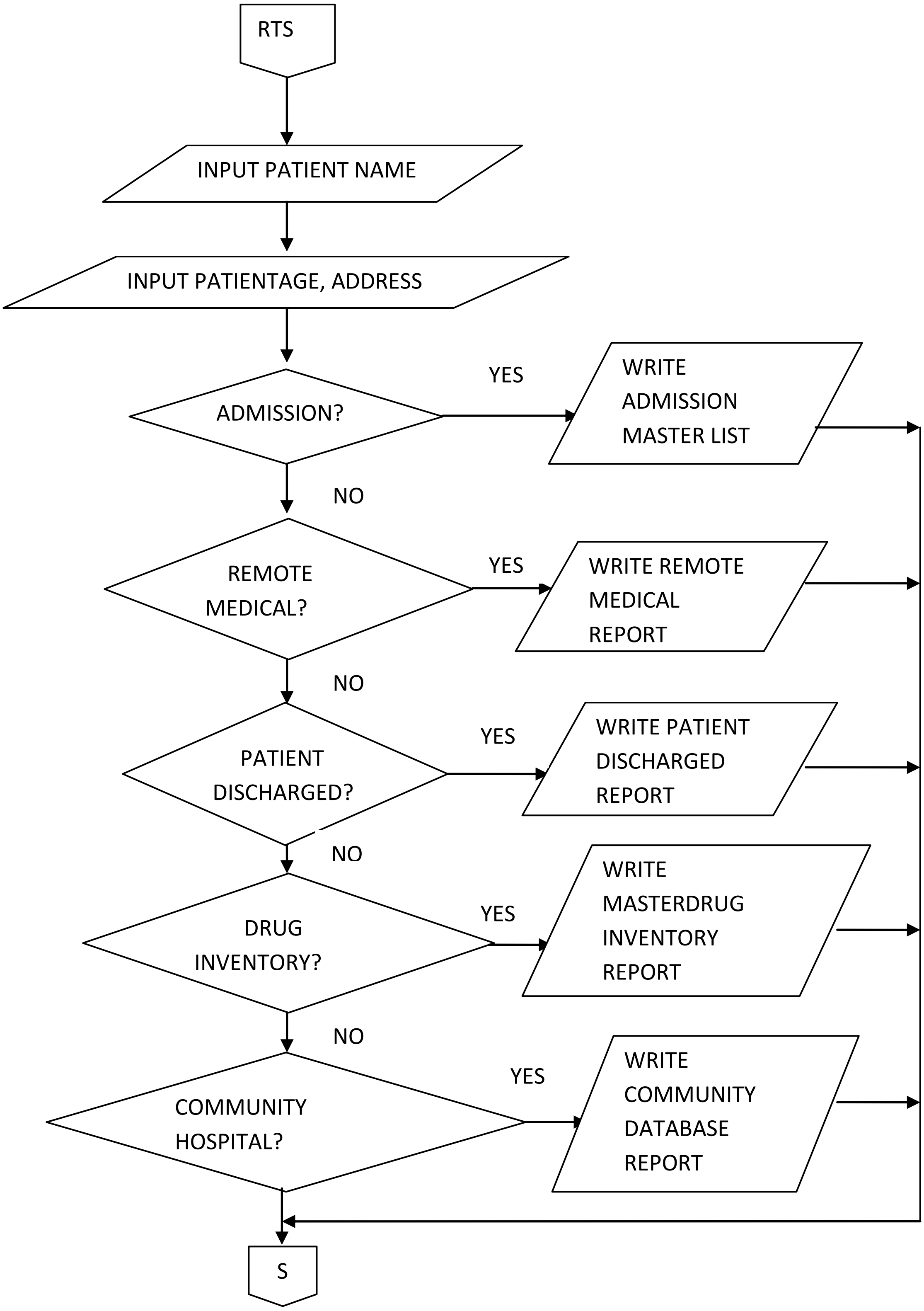


Figure 4.26: Reports Flowchart

4.11 **Input Specification**

The database specification for the Mobile Health Information Management System is shown below:

i. Mobile Patient Data (MPD)

The specification of the data used for the mobile patient data is shown below:

Table 4.13: Mobile Patient Data (MPD)

Name	Data Type	Width	Dec
Pid	Char	10	
Memo 1	Char	100	
Memo 2	Char	100	
Date Count	Char	15	
Systolic	Int	3	
Time Count	Char	15	
Diastolic	Int	3	
Irregular Heart Beat	Char	1	
Pulse	Int	3	
Telephone Number	Char	12	

The specification of the data used in the hospital is shown below:

ii. In – Patient Department (IPD)

Table 4.14: IPD

Name	Data Type	Width	Dec
PName	Char	10	
PAddress	Char	50	
PTel	Char	12	
Pemail	Char	50	
RefferedFrom	Int	50	
DateAdmtd	Char	12	
Gender	Char	6	
HospAdmtd	Char	50	
HWard	Char	10	
BedNo	Char	5	
Diagnosis	Char	100	
Prescription	Char	100	
Treatment	Char	100	
DocIc	Char	50	

Sponsor	Char	50	
STel	Char	12	
Semail	Char	50	
TBilltoDate	Char	10	
TBillPdtoDate	Int	10	
DischargeDate	Char	30	
DischargeStatus	Char	50	
PStatus	Char	50	
RecallDate	Char	15	

iii. Out – Patient Department (OPD)

Table 4.15: OPD

Name	Data Type	Width	Dec
PName	Char	25	
PAddress	Char	50	
PTel	Char	12	
Pemail	Char	50	
ReferedFrom	Char	50	
DateAdmtd	Char	12	
HiD	Char	10	
Gender	Char	6	
BloodGp	Char	3	
HospAdmtd	Char	50	
HWard	Char	10	
Diagnosis	Char	100	
Prescription	Char	100	
Treatment	Char	100	
DocIc	Char	50	
Sponsor	Char	50	
STel	Char	12	
Semail	Char	50	
TBilltoDate	Int	10	
DischargeDate	Char	30	

iv. Clinical Support (ClinicalSupt)

Table 4.16: ClinicalSupt

Name	Data Type	Width	Dec
Staff_NameIc	Char	25	
Function	Char	15	
Ward	Char	5	
Reports_to	Char	15	

Lab	Char	100	
EMR	Char	100	
OT	Char	100	
Duty_Roaster	Char	100	

v. Hospital Administration (HospAdmin)

Table 4.17: HospAdmin

Name	Data Type	Width	Dec
Staff_NameIc	Char	25	
Function	Char	15	
Rank	Char	15	
ChainOfCntrl	Char	15	

vi. Ancilliary Services (Ancilliary_Serv.DBF, Diagnosis Module)

Data utilized by CDS module for Clinical decision and stored in EMR.

Table 4.18: Ancilliary_Serv

Name	Data Type	Width	Dec
BloodTest	Char	100	
Urine	Char	100	
Cardiology	Char	100	
BloodPrsure	Char	25	
BloodSugar	Char	25	
Temp	Char	15	
Malaria	Char	25	
HIV	Char	25	
Tuberculosis	Char	25	
Cholera	Char	25	
Hepatitis	Char	100	
Eye	Char	100	
Ear	Char	100	
Mouth	Char	100	
Stool	Char	100	
OtherTests	Char	100	

vii. Operation_Theatre

Table 4.19: Operation_Theatre

Name	Data Type	Width	Dec
PName	Char	25	
PiD	Char	10	
SurgeryDate	Char	20	
DoctorIc	Char	25	
SurgeryType	Char	15	
Tools_Req	Char	100	
After_SurgWard	Char	15	
Bill	Int	10	
EvaluationOfSuccess	Char	100	
Recall	Char	12	

viii. EMR

Table 4.20: EMR

Name	Data Type	Width	Dec
PName	Char	25	
PiD	Char	25	
MedicalHistory	Char	50	
SensorReports(PDA)	Char	50	
Remarks(fromCDS)	Char	50	

ix. Expert (EC)

Table 4.21: EC

Name	Data Type	Width	Dec
ExpertName	Char	25	
Expertise	Char	15	
HospLocation	Char	25	
Qualifications	Char	100	
ETel	Char	12	
Expertemail	Char	50	
Eresidence	Char	50	
HospAvailability	Char	50	

x. Hospital Registration (HR)

Table 4.22: HR

Name	Data Type	Width	Dec
HospName	Char	30	
Location	Char	50	
www	Char	25	
Hemail	Char	30	
Hownership	Char	25	
DateEstab	Char	12	
HExpertise	Char	100	
CMO	Char	25	
CMOQualif	Char	100	

xi. Mechanical Equipment (ME including vehicles, furniture, fittings and medical equipments)

Table 4.23:ME

Name	Data Type	Width	Dec
EquipNo(id)	Char	10	
EquipType	Char	25	
QtyPurchased	Int	10	
QtySold	Int	10	
QtyBal	Int	10	
EquipStatus	Char	15	
Location	Char	15	
DatePurchased	Char	15	

xii. Drug

Table 4.24: Drug

Name	Data Type	Width	Dec
DrugID	Char	5	
Qty_in_Stock	Int	10	
ReorderLevel	Int	10	
QtyOrder	Int	10	
Bal	Int	10	
NairaValueBal	Int	10	

4.12 Entity Relationship Diagram

Entity relationship diagram (ERD) is a kind of diagram for presenting visually the structure of relational database. ERDs illustrate the logical structure of databases. Peter Chen developed ERDs in 1976. Since then Chaples Bachman and James Martin have added some slight refinements to the basic ERD principles.

There are three basic elements in ER models:

i. Entities are the "things" about which we seek information.
ii. Attributes are the data we collect about the entities.
iii. Relationships provide the structure needed to draw information from multiple entities.

4.12.1 The Entity Relationship Diagram

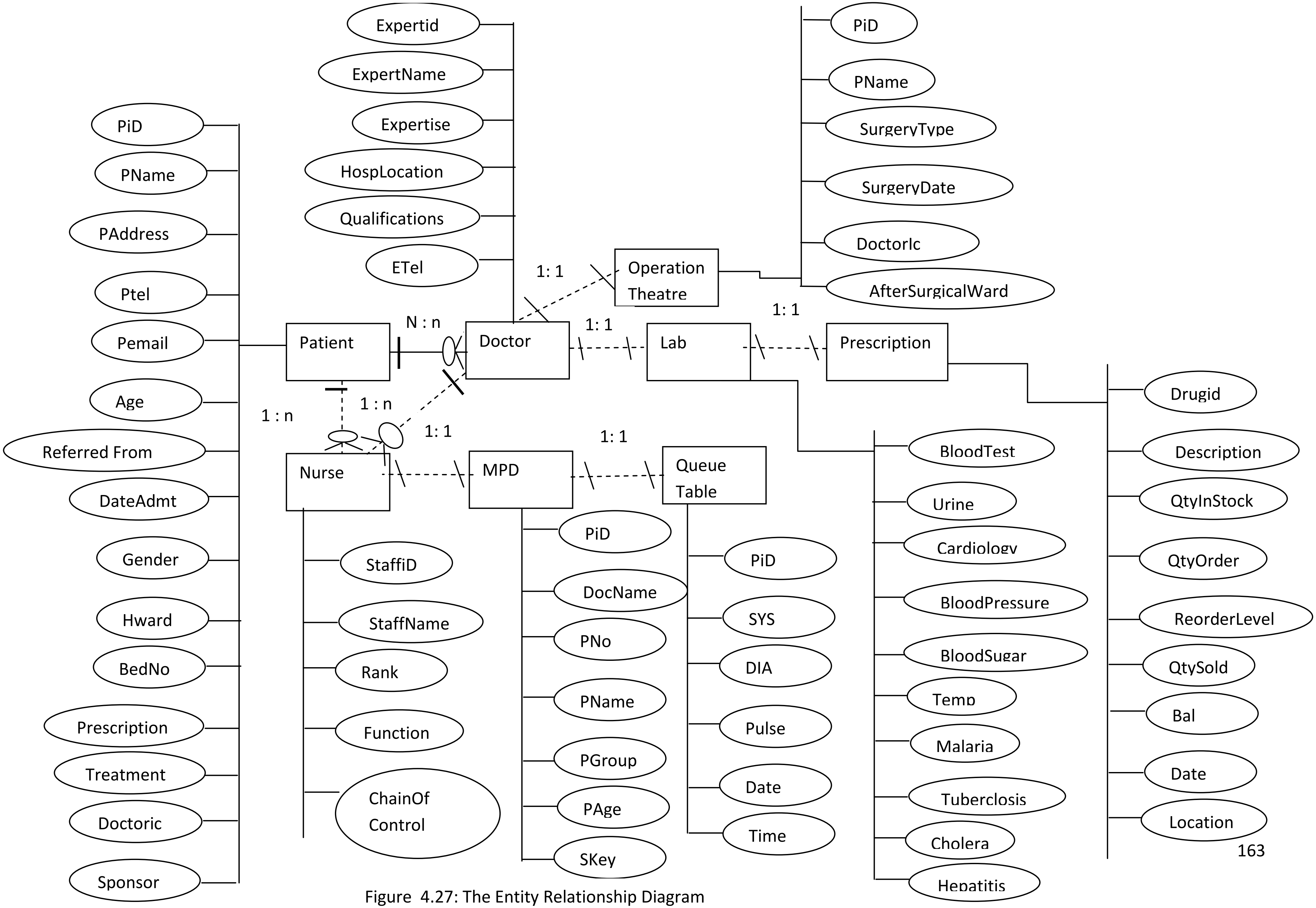


Figure 4.27: The Entity Relationship Diagram

4.13 Program Module Specification

Table 4.25: Program Module Specification

S/N	PROGRAM MODULE	FUNCTIONS
1.	Control_Centre Prg	Calls subprograms for execution
2.	HK.prg	Drives the house-keeping operations of IHIMS such as purging of unwanted files
3.	HPExit.prg	Exits from the purge module and returns control to the centre
4.	OPD.prg	Creates and manipulates Out-Patient DataBase
5.	IPD.prg	Creates and manipulates In-Patient DataBase
6.	EMR.prg	Creates a memo field used in capturing and storing information on patient records and those captured by sensors via PDAs
7.	CDS.prg	This is the program module used for analysis of medical data from labs and sensors which assists medical decision making on particular patients
8.	HA.prg	Creates inventory data base for inserting, updating and manipulation. It also handles other aspects of the Hospital's IMS
9.	HR.prg	Creates a data base for data capture, editing, deletion and manipulation of list of associated community hospitals
10.	EC.prg	Creates a data base of expert medical consultants, their location and expertise, edits, deletes and manipulates it and return control to the centre
11.	DrugInv.prg	Creates Drug Inventory data base and tracks usage, reorder and balance of stock and value and returns control to the centre
12.	OT.prg	This module creates a data base for operations theatre including surgery date, type, patient bill etc and returns control to the centre.

4.13.1 Mathematical Specification

This relate only to cost of drugs and number of days admitted and volumes of treatment received.

Table 4.26: Mathematical Specification

Drug/Treatment	Unit Price	Quantity	Total Price
Antibiotic	N10 per cap	200	N2000
Surgery	N25000	1	N25000
Multi-Vitamin	N10 per cap	200	N2000
Bed occupancy by night	N5000 per night	7	N35000
Food	N500 per meal	500x3x7	N10500
Other drugs			N10000
Total Bill			N84500
Advance Payment			N30000

Bal payable by patient before discharge			N54500

The equation might be presented thus:

TBp = SDuQSDp = Bb

Where DuQ = item debits x unit price x Quantity = Total Bill to Patient

Dp = Deposits by patient

Bb = Bill Balance

TBp = Total Bill to the patient

4.14 **Data Dictionary**

Table 4.27: Data Dictionary

Variable Name	Description/Function
1. PName	Patient Name: surname+middle name+First name. Refers to full name of admitted patient for treatment of an ailment.
2. Address	Street Name+House No+Area+City+State+Country
3. Tel	Telephone Number of patient
4. email	Email address of patient
5. Referred from	Hospital name and address from where the patient was transferred to present hospital
6. Appoint_Date	Date designated for treatment of patient
7. HiD	A hospital identity number for which a hospital can be accessed in a data base
8. Gender	Male or Female (Mr, Mrs/Miss)

9. Blood_Group	Eg. O, A, As etc.
10. Hosp_Admited	Community Hospital in which patient is admitted + Street Name+ House no.
11. Ward	Hospital Ward (eg. Male or female ward) + Bed no. where the patient is stationed for treatment
12. Case_History	A memo field which records archived information fro EMR from commencement of diagnosis and treatment to date.
13. Bed_No.	The identity number of the bed allocated to the patient + ward +Hospital location
14. BedTransfer	The identity number of the bed transferedthe patient + ward + New Hospital location
15. Diagnosis	A memo field which archives results of medical diagnosis to aid Doctors decision, prescription, treatment and management.
16. Prescription	Drugs recommended for administration based on medical diagnosis
17. DoctorIc	The Doctor in charge of a ward and a patient
18. Patient_Sponsor	The person in charge of paying the hospital bill of a patient
19. PSTel	Patient’s Sponsors Telephone number
20. PSEmail	Patient’s Sponsors Email Address
21. TotalBilToD	Total Bill charged the patient to date
22. TotalBilPdToD	Total payment to date by patient

23. DateAdmited	Date of patient's initial admission into the hospital
24. TretUndergone	Treatment administered on the patient so far
25. DateDischarged	Date of Patients discharge from the hospital
26. Recall	New date given to a discharged patient for checkup
27. StaffNameIc	Staff Name in charge of patient and the ward
28. StaffFunction	The role of the staff in charge in the ward. Is the staff a Medical Doctor, Nurse, Hospital Assistant, Messenger, Driver etc.?
29. ReportsTo	The higher authority to whom the staff reports
30. LabRpt	Memo field archived Report of Lab Tests results
31. EMR	Electronic Medical Records. This is an archived memo field for storing electronic medical reports on individual patient such as ECG, pulse, blood sugar, BP etc.
32 OTRpt	Archived memo field for operations theatre reports stored in EMR
33. DutyRoster	Archived memo field staff duty roster for the hospital
34. CMCic	The name of the chief medical Consultant in charge
35. Rank	Official Rank of the CMC in charge
36. SofCtrl	Span of Control of the CMC in charge
37. BloodTest	Result of Blood Test stored in the EMR
38. Urine	Result of Urine Test stored in the EMR
39. Cardiology	ECG examination Results stored in the EMR

40. BloodPresu	Result of blood pressure tests stored in the EMR
41. Malaria	Result of Widel Test stored in the EMR
42. BloodSugar	Fasting blood sugar test results stored in the EMR
43. Temp	Temperature tests results stored in the EMR
44. HIV	Results of HIV tests stored in the EMR
45. Turbaclosis	Results of Tuberculosis Lab Tests stored in the EMR
46. Cholera	Cholera tests results stored in the EMR
47. Hepatitis	Hepatitis tests results stored in the EMR
48. Eye	Optometry Tests on the eye results stored in the EMR
49. Ear	Ear tests results stored in the EMR
50. Stool	Stool Tests results stored in the EMR
51. Other tests	A memo field reserved for stored other tests results not capture above

4.15 **Dataflow Diagram**

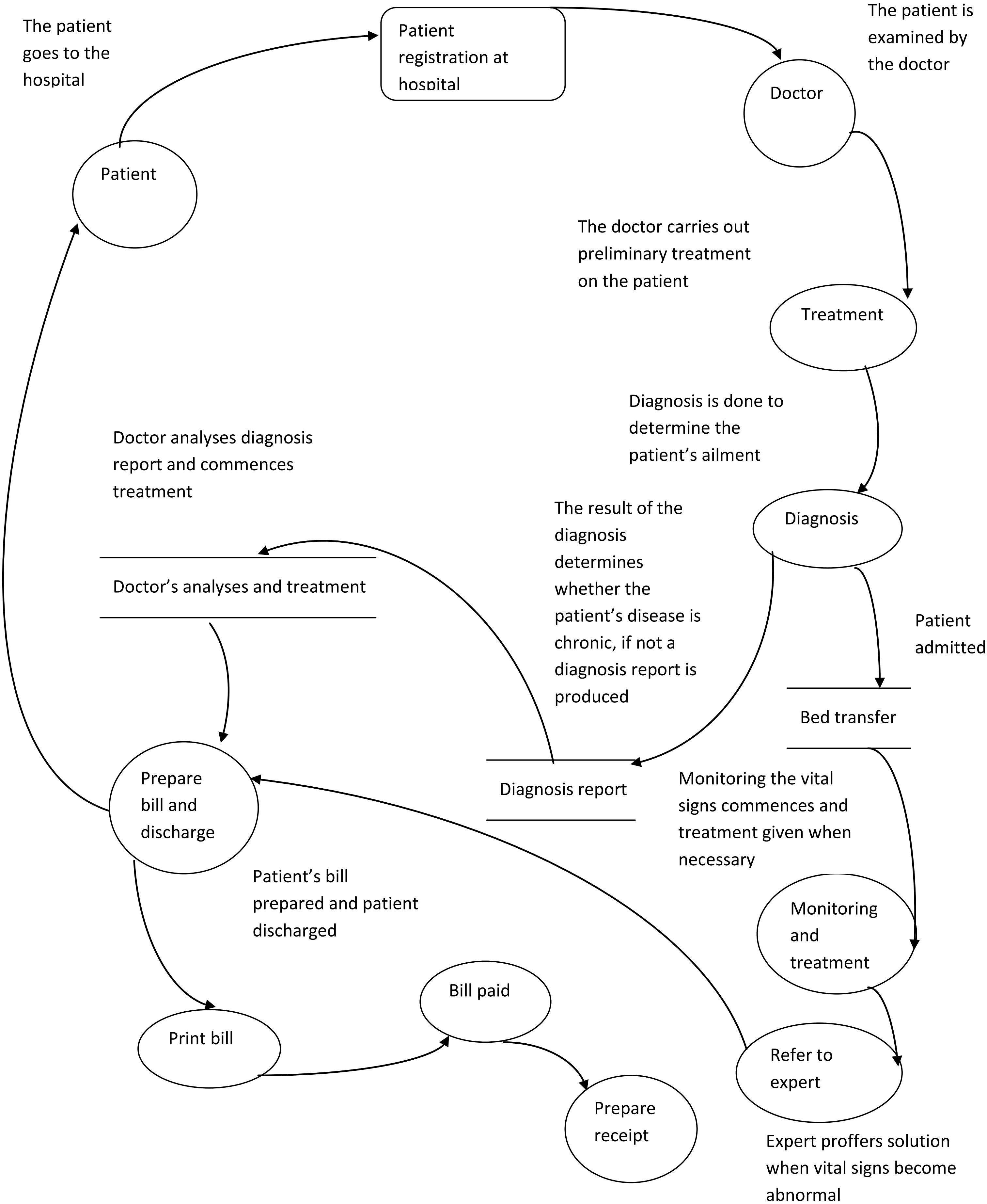


Figure 4.28: Overall Data Flow Diagram of Integrated Hospital Management System

4.16 System Flowchart

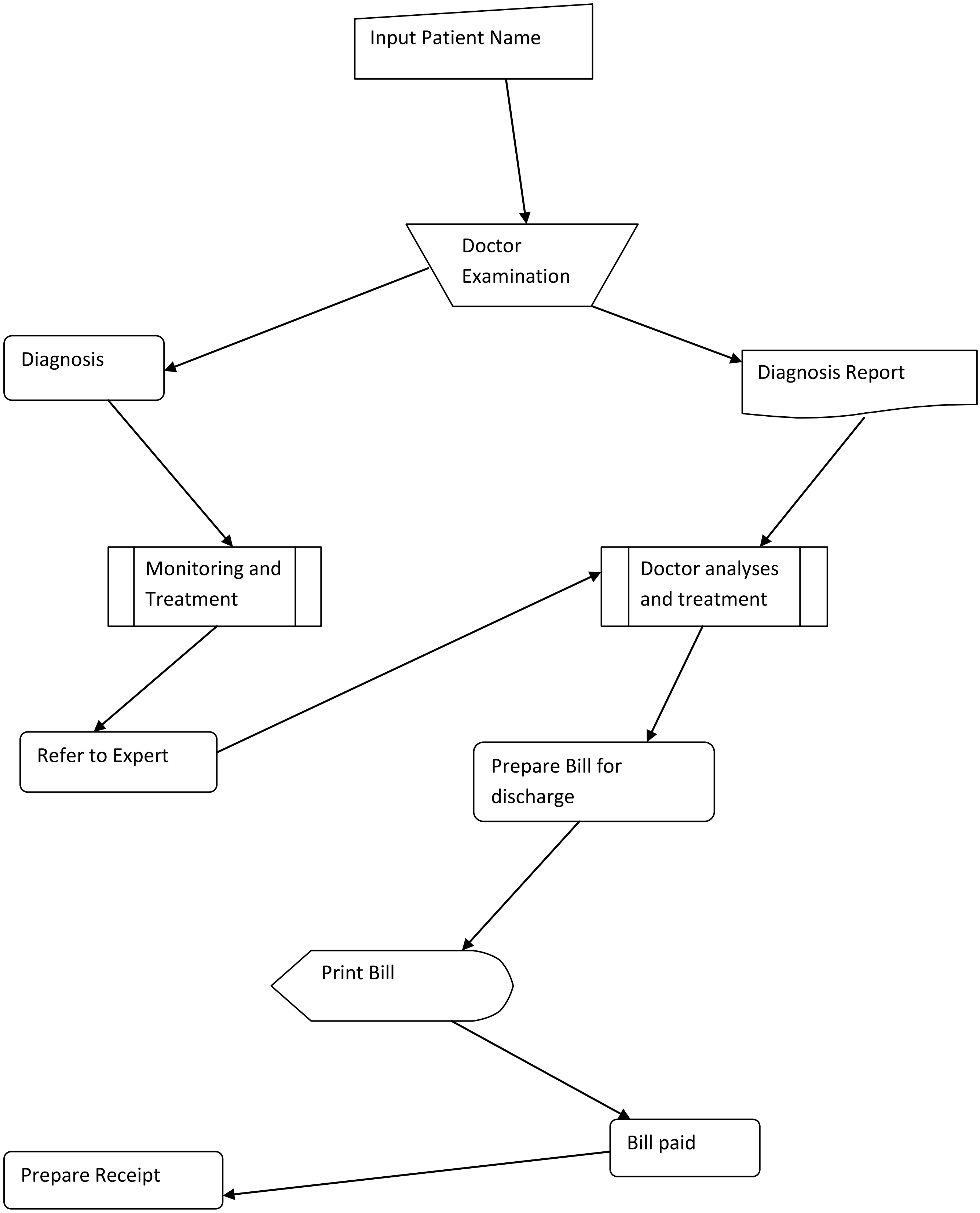


Figure 4.29: Systems Flowchart

4.17 **Pseudocode**

```
IHIMS

Select Menu Item

CASE menu Item OF
        HK: House Keeping
        OPD: Out Patient Department
        HR: Hospital Registration
        IPD: In Patient Department
        EMR: Electronic Medical Records
        CDS: Computerized Decision Support System
        HA: Hospital Administration
        OT: Operation Theatre
        ME: Medical Equipment
        RTS: Reports
ENDCASE
IF      Housekeeping THEN
        Input Admin name
IF      Admin Right THEN
        Input patient id
        Check patient last update
    ELSEIF   patient last update more than ten years THEN
        Delete patient id
        Print New list
ENDIF
        Check Outpatient department
IF      Outpatient department THEN
        Register patient
        Consult doctor
    ELSEIF further examination required THEN
        Do medical investigation
        Prescribe drugs
        Prepare bill
        Print bill
ELSE discharge patient
ENDIF
        Check Hospital Registration
IF      Hospital Registration THEN
        Fill hospital registration form
        Process form
```

ELSEIF Hospital name taken by another hospital
Register hospital name
Print hospital name
ELSE Check In patient department
ENDIF
IF Inpatient department THEN
Input patient id
Consult doctor
Transfer bed
Make prescription
Monitor patient
Prepare bill
Print bill
Discharge
ELSE Check Electronic Medical Records Department
IF Electronic Medical Records THEN
Input Doctor's name, phone number
Input special key
Input patient id
Upload patient record
Print patient record
ELSE Check computerized decision support system
ENDIF
IF Computerized Decision Support System THEN
Input patient's name, patient's age
Input patient's vital sign readings
IF vital sign readings abnormal THEN
Consult a doctor
Doctor's prescription
Monitor patient
Discharge patient
ELSE Check Hospital Administration
IF Admin Right THEN
Input Admin name
Input password
ELSEIF New user THEN
Input new user's name
Assign user's password
Perform other admin functions
Print new user's name

```
ELSE check Operations Theatre
ENDIF
IF      Operations Theatre THEN
        Input patient id
        Input surgery type, date
        Perform surgery
        Assign surgical ward
        Prescription
        Monitor patient
IF      patient stable THEN
        Prepare bill
        Print bill
        Discharge
ELSE check Mechanical Equipment
ENDIF
IF      Mechanical Equipment THEN
        Input Equipment Number
        Input Equipment Type
        Input Number in Stock
IF      Equipment status in good condition THEN
        Pack in store
ELSE    fix equipment
ENDIF
IF      Reports THEN
        Input patient name
        Input patient age, address
IF      patient for admission THEN
        Write Admission Master List
ELSEIF Remote Medical THEN
          Write Remote Medical Report
ELSEIF Patient Discharged THEN
        Write Patient discharged report
ELSEIF Drug Inventory THEN
        Write drug inventory report
ELSE
        Write Community Hospital Database Report
ENDIF
 ENDIF
ENDIF
ENDIF
```

```
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
STOP
```

4.18 Components of the Design

4.18.1 System Architecture

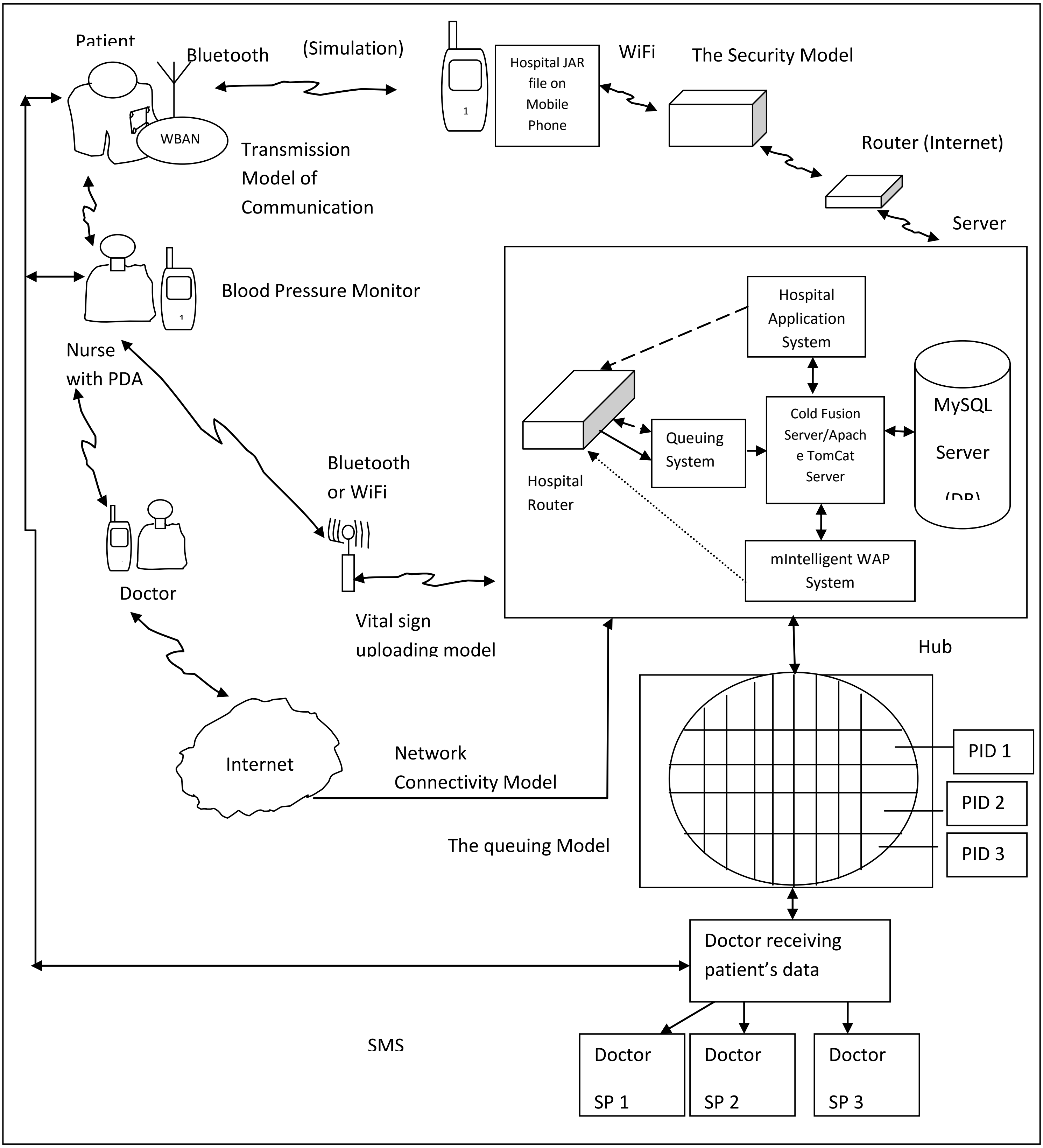


Figure 4.30: Total System Architecture for Remote Medical

PID 1 Patient ID 1: This can be from Pid 1 to Pid n.

SP 1 Specialist 1: This can also be form Sp 1 to Sp n.

Below is the detailed explanation of the above architecture:

i. **The Patient**

The patient is embedded with the blood pressure and the pulse rate sensor, which gathers information on the blood pressure and the pulse rate of the patient, sends this information through the Bluetooth to the Intelligent Personal Digital Assistant (IPDA). The IPDA can either be a laptop, a smart phone or Personal Digital Assistant. This creates a Wireless Body Area Network (WBAN). In this project, the readings of the blood pressure and the pulse rate are simulated using the known causes of high and low blood pressure and pulse rate respectively.

ii. **The IPDA**

The Intelligent Personal Digital Assistant contains the **Mobile Health Information Management System software** that collects the vital signs from the patient and sends it through WiFi to the router which also transmits this data to the server. This IPDA is a simulated phone which uses the data structure as patient's name, phone number, hospital id, blood pressure category, age, smoking habits and alcohol intake to simulate the blood pressure and the pulse rate readings. These readings are simulated every 25 seconds and sent directly to the queuing model in the server.

iii. **The Server**

The server has four main applications viz: the complete hospital application, the database, the Electronic Medical records, and the queuing model. The complete hospital application contains the following modules: Medical Experts, Medical Equipment, Patient Registration, Dynamic Staff Worklist, Patient Discharged, In-Patient, Out-Patient, Pharmacy/Portal Reports, Drug Dispensory Unit, Hospital Administrator, Associated Communities Hospital, Operation Theatres, Drug Inventory and Medical Data Analyser.

The database was created using My SQL.

The Electronic medical Records consists of the records of the patients, which is simulated by the mobile phone (IPDA) and sent to the server. It contains the Mobile Users module where all the records of the patients' data simulated are kept. The records contains the patient's id, name, phone number, systolic pressure, diastolic pressure, pulse, Mean Arterial Pressure (MAP), Blood Pressure, suggestions and date and time. These are the readings generated when the simulation is done in the mobile phone. These readings are generated every 25 seconds and sent to this mobile users module. The doctor can look at the mobile users module at any time to see the detailed records of the patients.

The queuing model contains the Network Queues of Mobile Users module which contains the details of all the patients' records simulated by the IPDA. The simulation once done, will continue to send the records to the queuing model every 25 seconds and the total number of records sent will continue to increase until the simulation is stopped. If the simulation is done for a whole day, the readings will continue to be generated and the total number of records sent to the queuing model will continue to be increased. The doctor can easily look at the queuing model to see the summary of the readings generated for each patient. A detailed graph for each patient is also generated in the queuing model so that the doctor can look at a particular patient's graph to see the progression of his vital signs and will be able to make timely and well informed decision and prescription.

iv. The Hub

The hub acts as a data repository where the information on the vital signs of patients are stored. The hub collects this information and sends it to the doctor. There is a queuing model in the hub which puts the patients' information in a queue and send these information to the doctor when there is an abnormal situation. The type of the queuing model used here is the M/M/1 which uses FIFO i.e. First In First Out. The doctor is able to view the server from anywhere.

v. The Doctor

There are two categories of doctors in this architecture. One is the doctor who receives the information on the vital sign of the patient and acts accordingly while the other is the doctor that does not necessarily receive information from the hub but who accesses the server to look at the database of the patients' records. The doctor accesses the server though the internet wherever he is. He can also send an SMS to the nurse informing her on what to do.

vi. The Nurse

The nurse gets information from the doctor directing her on what to do. The nurse also communicates to the patient through SMS if the patient is in a remote location or by direct oral communication if the patient is within the hospital premises. The nurse can also use the blood pressure loader to upload the patient's vital sign monitored over a period of time to the server and can send an SMS to the doctor whenever there is an abnormal situation.

4.19 Sequence Diagram of the Architecture

A sequence diagram is an object diagram that shows object interactions arranged in time sequence. In particular, it shows the objects participating in the interaction and the sequence of messages exchanged. A sequence diagram includes time sequences but does not include object relationships. A sequence diagram can exist in a generic form (describes all possible scenarios) and in an instance form (describes one actual scenario).

4.19.1 The Sequence Diagram of the Remote Medical Monitoring

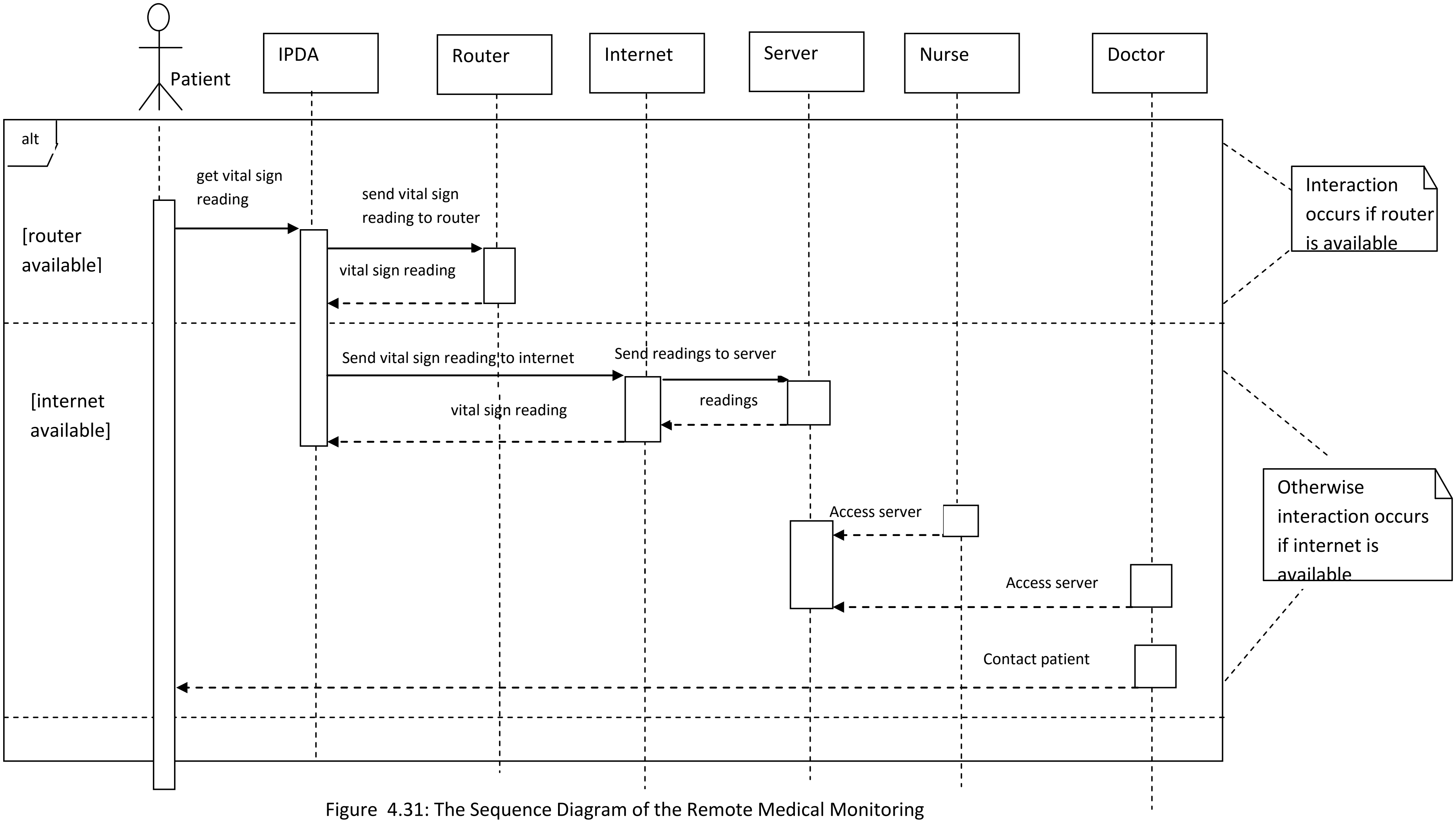


Figure 4.31: The Sequence Diagram of the Remote Medical Monitoring

The procedure of how the messages interact among components within the system is shown as follows:

i. The patient enters the parameters for the generation of his vital signs (blood pressure and pulse rate). The parameters to enter are name, phone number, age, blood pressure category, alcohol intake and smoking habits.
ii. The Intelligent Personal Digital Assistant (IPDA) simulates the readings for the blood pressure and pulse rate. The readings are simulated every 25 seconds.
iii. If router is available, then the readings of the blood pressure and pulse rate simulated are sent through the router using wifi (which is present in the IPDA and the router) to the server.
iv. If on the other hand, internet is available, the simulated readings are sent through the internet to the server.
v. The nurse can access the server from wherever she is and view the patient's records.
vi. The doctor can also access the server wherever he is and can contact the patient if the need arises.

4.20 The Queuing Model for distribution of Patient data

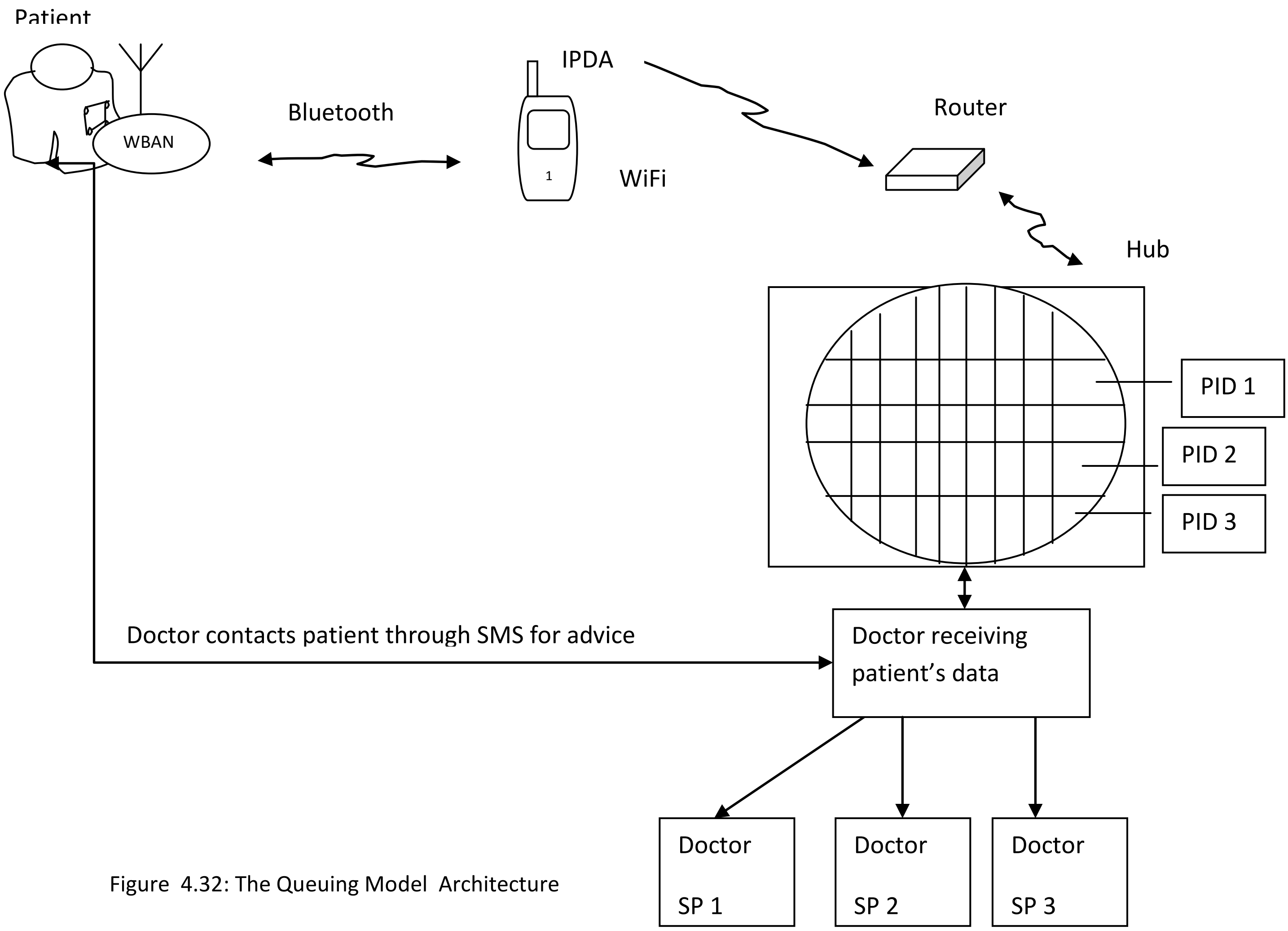


Figure 4.32: The Queuing Model Architecture

PID 1 Patient ID 1: This can be from Pid 1 to Pid n.

SP 1 Specialist 1: This can also be form Sp 1 to Sp n.

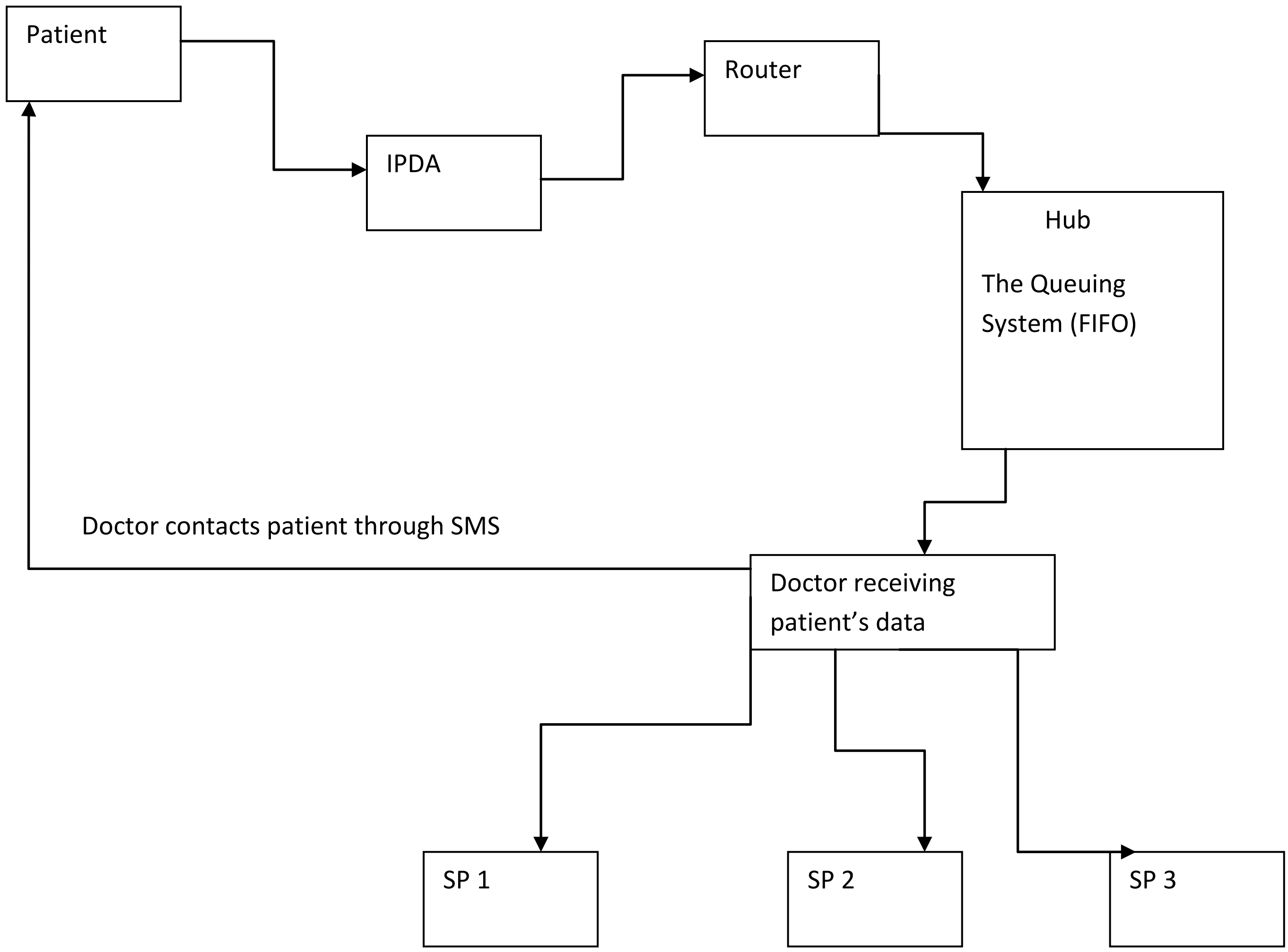


Figure 4.33: Block diagram of the queuing model

The M/M/1 queue consists of a server which provides service for the packets of data from the patients who arrive at the system and depart. It is a single-server queuing system with exponential interarrival times, exponential service times and first-in-first-out queue discipline. If a packet of data from a patient arrives when the server is busy, it joins the queue (the waiting line). There are two types of events: arrival events (A) and departure events (D). The following quantities are used in representing the model:

AT = arrival time

DT = depature time

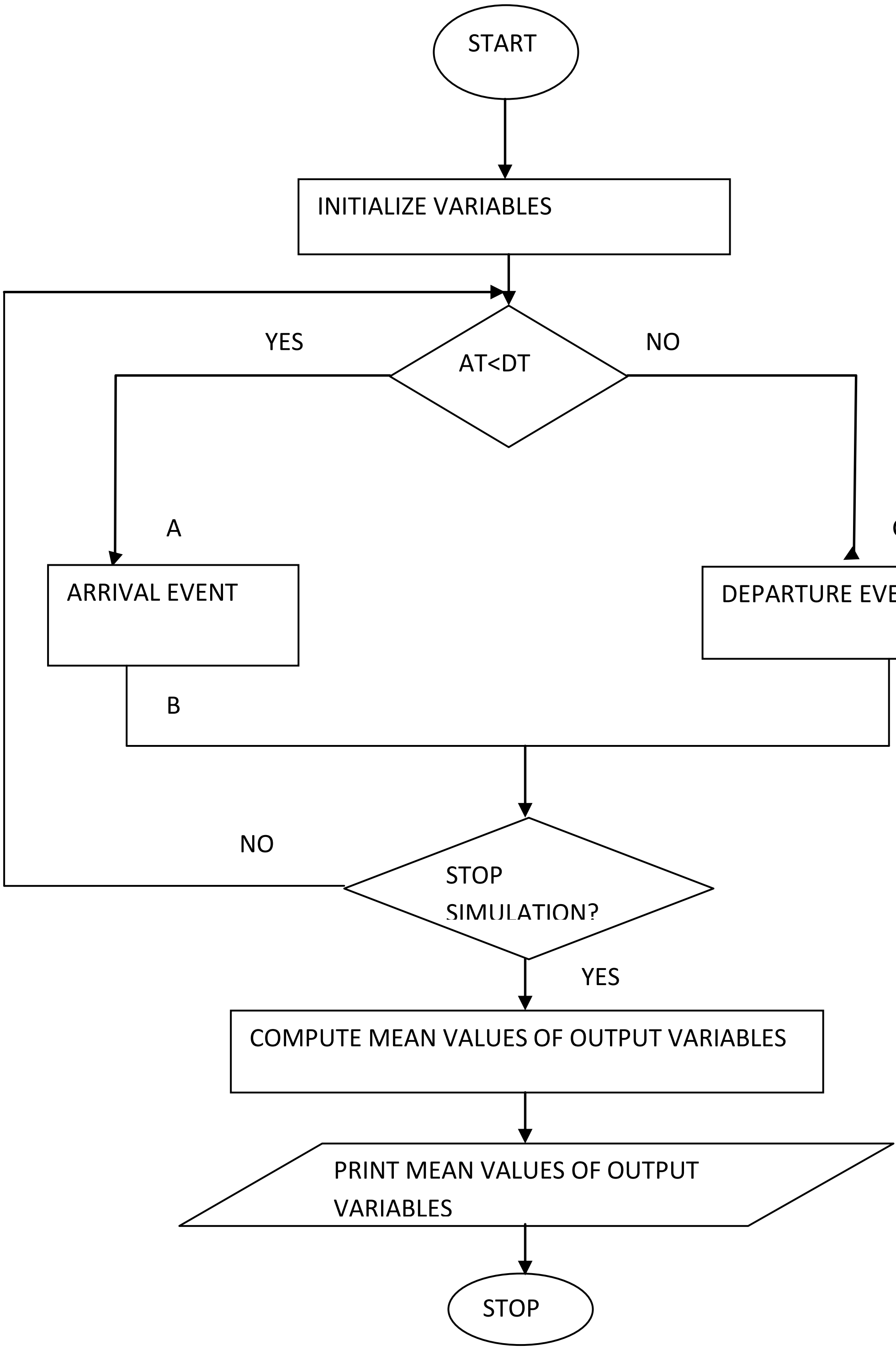


Figure 4.34: Fowchart of the queue simulation

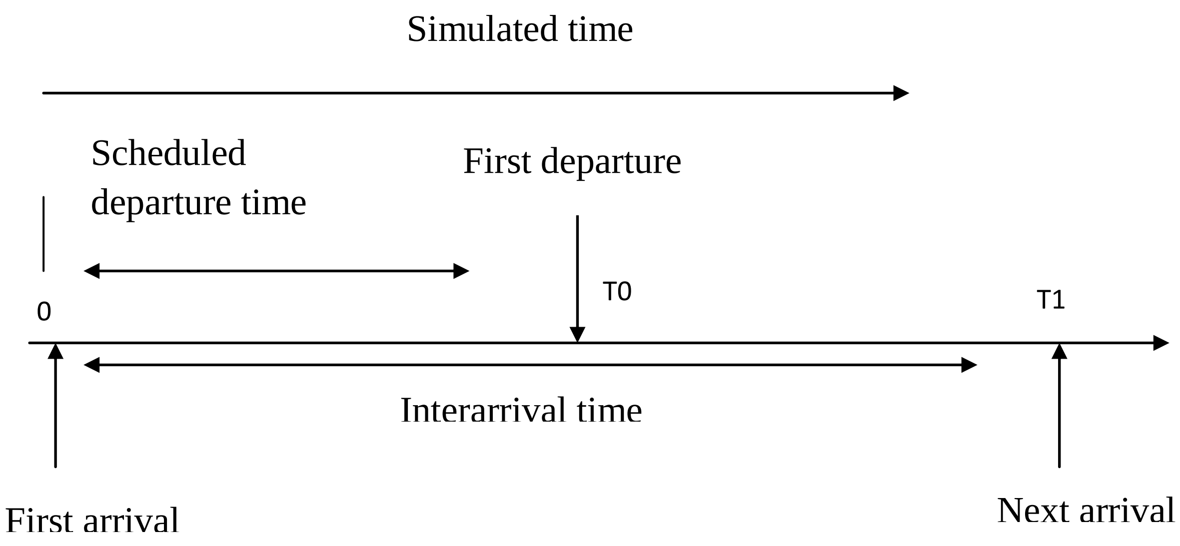


Figure 4.35: The first few events in simulation

In the first few events in simulation, the packets of data from the patient arrive the queuing model as the first arrival in time 0 and the interarrival time which is the time between the first arrival and the next attival is denoted by T0 while the next arrival is denoted by T1. The first packets of data to arrive the queue is also the first to depart from the queue which is the first departure.

In the queuing model, vital signs are collected by the sensor on the patients' body, sent through the Bluetooth (This data is simulated in the IPDA) to the IPDA which transmits this data by WiFi to the router which further transmits the data wirelessly to the hub which is in the server.

The hub acts as a data repository where these data are stored and sent to the doctor when there is an abnormal situation.

4.20.1 The Queueing Model

Queueing models can be represented using Kendall's notation

A/B/S/K/N/D

Where

A is the interarrival time distribution

B is the service time distribution

S is the number of servers

K is the system capacity

N is the calling population

D is the service discipline assumed

vii. The Arrival Rate

The data arrive as packets of data from different patients wearing the sensors into the hub.

Let Ci be the interarrival time between the arrivals of the (i – 1)th and the ith patients, the mean(or expected) inter-arrival time is denoted by E(C) and is called β; = 1/E(C) the arrival frequency. The data is simulated in the IPDA and the interarrival time is 25 seconds and therefore β = $\frac{1}{25}$ = 0.4.

viii. Service Mechanism

This is specified by the number of servers (denoted by s) each server having its own queue or a common queue and the probability distribution of the patient's service time.

Let Si be the service time of the ith patient, the mean service time of a customer is denoted by E(S) = $\mu = \frac{1}{E(S)}$ the service rate of a server. The service time here is 10 seconds and therefore μ = $\frac{1}{10}$ = 0.1.

ix. Queue Discipline

Discipline of a queuing system means the rule that a server uses to choose the next patient from the queue (if any) when the server completes the service of the current patient.

The queue discipline for this system is

Single Server- (FIFO) First In First Out i.e. patients data are worked on according to when they came to the queue.

x. Measures of Performance for the Queuing System

Let

Di be the delay in queue of the ith patient

Wi be the waiting time in the system of the ith patient

F(t) be the number of patients in queue at time t

G(t) be the number of patients in the system at time t = F(t) + No of patients served at t.

Then the measures,

$$D = \lim_{n\to\infty} \frac{\sum_{1=1}^{i=n} Di}{n} \text{ and} \qquad (4.4)$$

$$W = \lim_{n\to\infty} \frac{\sum_{1=1}^{i=n} Wi}{n} \qquad (4.5)$$

are called the steady state average delay and the steady state average waiting time in the system. Also the measures,

$$F = \lim_{n\to\infty} \frac{1}{T} \int_0^T F(t).dt \text{ and} \qquad (4.6)$$

$$G = \lim_{n\to\infty} \frac{1}{T} \int_0^T G(t).dt \qquad (4.7)$$

are called the steady state time average number in queue and the steady state time average number in the system.

xi. Single Channel Queue

[M/M/1] : {FCFS or FIFO} Queue System

xii. Arrival Time Distribution

This model assumes that the number of arrivals occurring within a given interval of time t, follows a poisson distribution with parameter $(\beta)t$. This parameter $(\beta)t$ is the average number of arrivals in time t which is also the variance of the distribution. If n denotes the number of arrivals within a time interval t, then the probability function p(n) is given by

$$P(n) = \frac{(\beta)^n}{n!} e^{-x} \qquad n = 0,1,2\ldots. \qquad (4.8)$$

The arrival process is called poisson input

The probability of no(zero) arrival in the interval [0,t] is,

Pr (zero arrival in [0,t]) = $e^{-\beta t}$ = p(0)

Also

P(zero arrival in [0,t]) = P(next arrival occurs after t)

= P(time between two successive arrivals exceeds t)

Therefore the probability density function of the inter- arrival times is given by,

$e^{-\beta t}$ for t > 0

This is called the negative exponential distribution with parameter β or simply exponential distribution. The mean inter-arrival time and standard deviation of this distribution are both $1/(\beta)$ where, $(\beta)$is the arrival time.

xiii. Performance Measures

The average number of units in the system G can be found from

G = sum of [n*Pn] for n = 1 to ∞

$$G = \frac{\beta}{\mu-\beta} = \frac{P}{1-P} \text{ where } P = \frac{\beta}{\mu} \qquad (4.9)$$

The average number in the queue is

$$F = (G - (1 - P_0) \qquad (4.10)$$

Sum of [(n-1)*Pn] for n = 1 to ∞

$$F = \frac{\beta^2}{\mu(\mu-\beta)} = \frac{P^2}{(1-P)} \qquad (4.11)$$

The average waiting time in the system (time in the system) can be obtained from

$$W = \frac{G}{\beta} = \frac{1}{\mu-\beta} \text{ and} \qquad (4.12)$$

$$D = W - \frac{1}{\mu} = \frac{\beta}{\mu.(\mu-\beta)}. \qquad (4.13)$$

The traffic intensity P (sometimes called occupancy) is defined as the average arrival rate (lambda) divided by the average service rate (mu). P is the probability that the server is busy.

$$P = \frac{\beta}{\mu} \qquad (4.14)$$

The mean number of customers in the system (N) can be found using the following equation:

$$N = \frac{\rho}{1-\rho} \qquad (4.15)$$

You can see from the above equation that as p approaches 1 number of customers would become very large. This can be easily justified intuitively. p will approach 1 when the average arrival rate starts approaching the average service rate. In this situation, the server would always be busy hence leading to a queue build up (large N).

Lastly we obtain the total waiting time (including the service time):

$$T = \frac{1}{\mu - \beta} \tag{4.16}$$

4.21 The Transmission model of communication

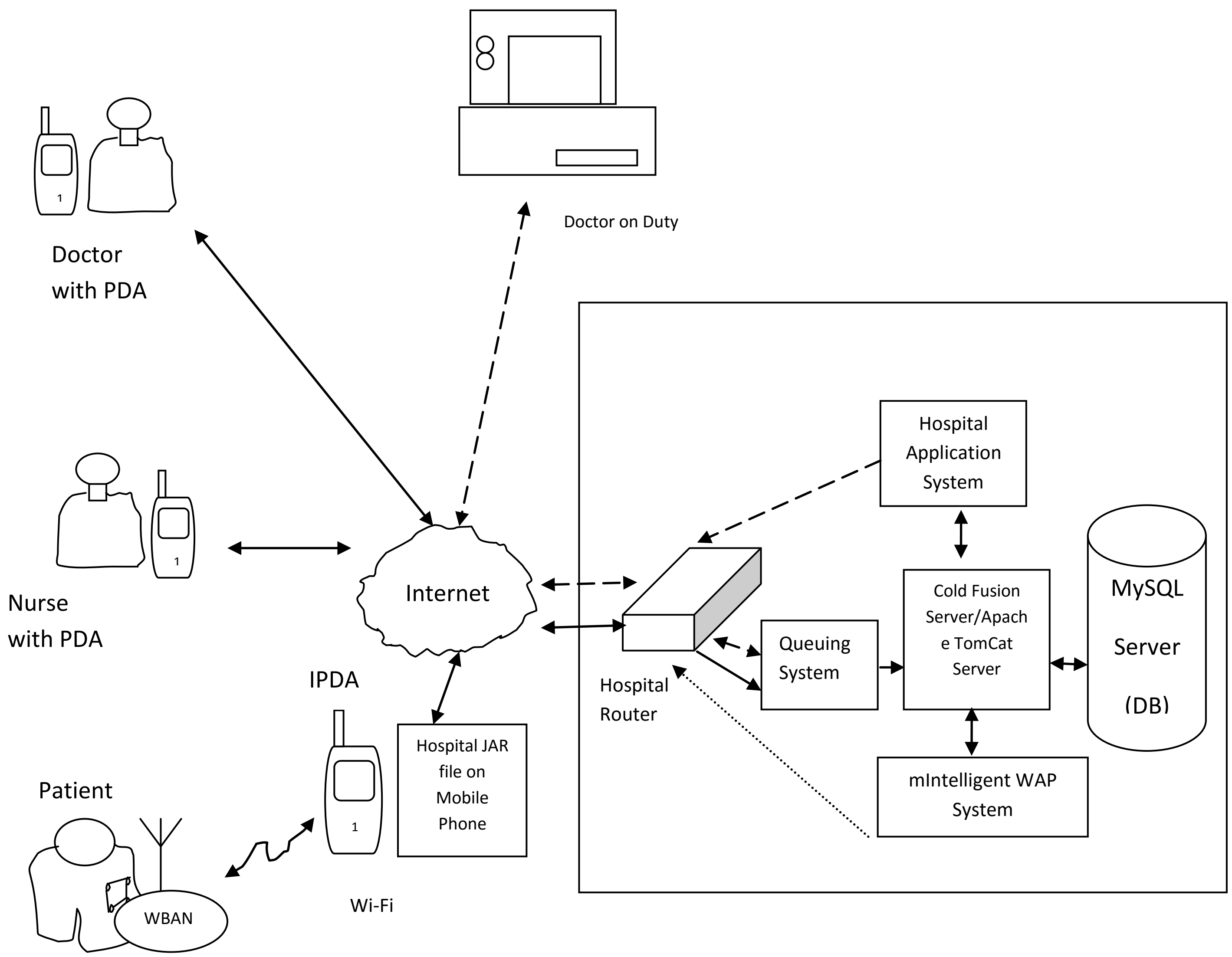


Figure 4.36: Data Transmission Model Architecture

The patient is embedded with different sensors, which gathers the information on the vital signs of the patient creating a wireless body area network (WBAN) and sends it to the Intelligent Personal Digital Assistant (IPDA) through the Bluetooth. This is done by simulation.

In the IPDA, there is software called the mobile health information management system which collects these vital signs, analyses them and sends them through routers or the internet depending on the location of the patient to the hospital router which transfers it to the queuing model in the server.

The queuing model is responsible for putting the packets of data as they arrive in a queue and sending them to the doctor (when there is an abnormal situation) through an SMS to his phone.

The doctor in turn access the server from wherever he is through the internet, checks the patient’s record and them sends an SMS to the nurse instructing her on what to do. He can also send an SMS to the patient giving his advice.

The nurse can also contact the patient through SMS.

4.22 The Vital Sign Uploading Model

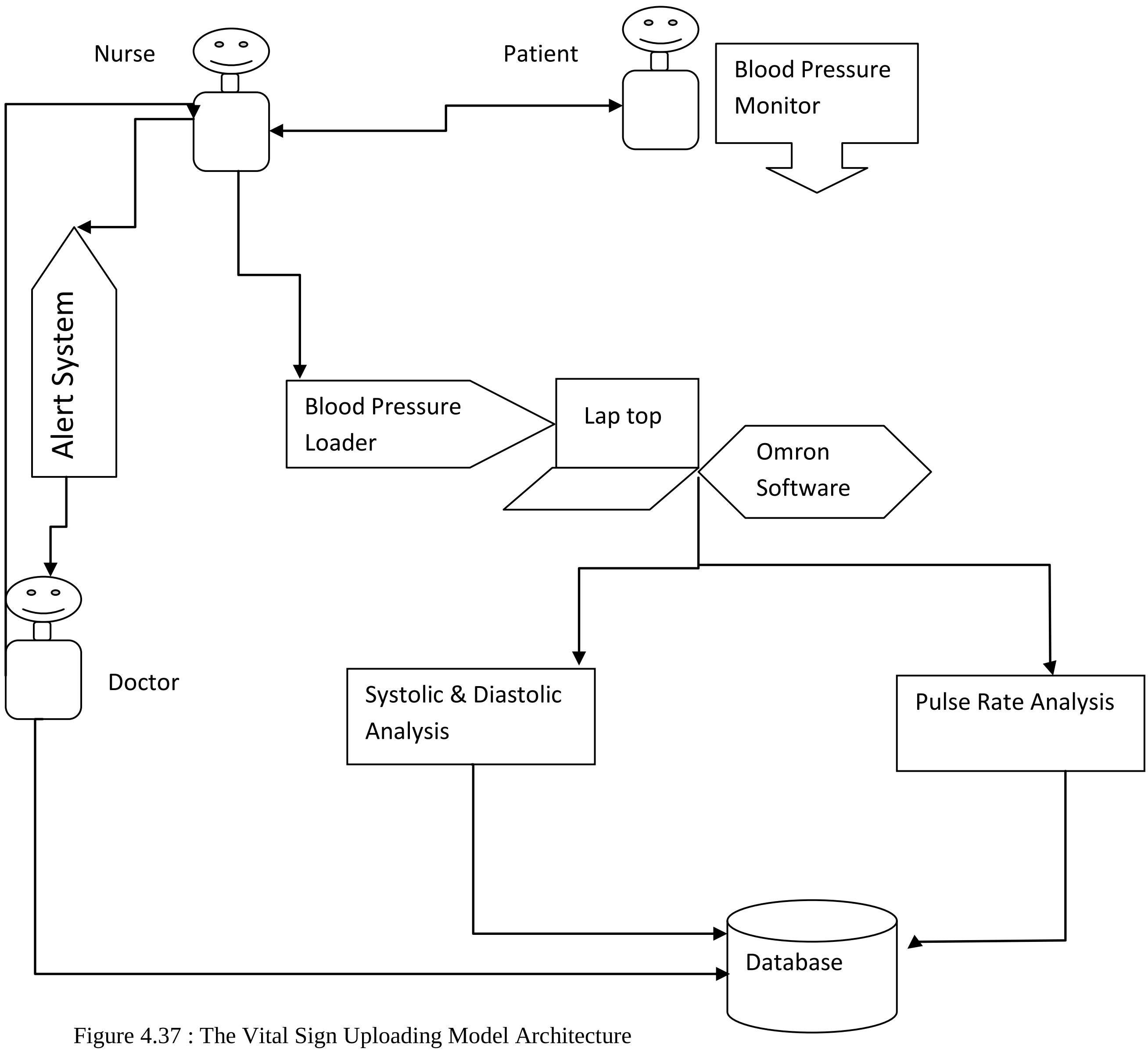


Figure 4.37 : The Vital Sign Uploading Model Architecture

The vital sign uploading model is used to upload the vital signs of the patient to the server. This model is used for patients who are in the hospital or for other patients who are not chronically ill but need to have their vital signs monitored.

This is achieved by using the Omron 760IT blood pressure monitor which monitors the blood pressure and the pulse rate, stores the readings which can be transferred to the computer.

The readings are checked at intervals and uploaded to the server every day or every week depending on the condition of the patient or the reason the blood pressure is monitored.

The nurse uploads the readings to the server and sends an alert to the doctor if there is any abnormal reading and the doctor can view the database anywhere he is and proffer his solution.

4.22.1 The Architecture

Figure 4.38 shows the overview of the architecture of the blood pressure monitoring and alert system.

The patient fixes the cuff of the blood pressure monitor on his upper left arm and presses the start button. This activates the blood pressure monitor which inflates the cuff automatically until when it gets to its peak, after which the cuff will start to deflate. After the deflation, the blood pressure and the pulse rate readings appear on the window of the monitor. If the patient is at home, he can take the readings from time to time until after about one week, when he takes the monitor to the nurse in the hospital for the analysis. The patient can transfer the readings to the laptop using the USB cable provided. If the patient is in the hospital, the nurse can take the readings at an interval depending on the condition of the patient.

The nurse or the patient transfers the readings to the computer using the USB cable provided. A CSV (Comma-Separated Values) file will be created which puts the readings in a tabular form.

A program called the Blood Pressure Loader (BPL) was written in Visual Basic which uploads the CSV file in the computer automatically into the database and is able to send an alert into the doctor’s phone so that wherever the doctor is, he will be able to proffer a solution. The doctor on getting the alert can check the database wherever he is to view the patient’s vital sign history and also proffers a solution.

The nurse on receiving the patient’s readings transfers it into the computer. She launches the blood pressure loader and uploads the patient’s readings into the computer. She can also send an alert to the doctor if the patient has abnormal readings. If the patient is in the hospital, the nurse will be the one to take the readings and transfer them into the computer as the case may be.

The blood pressure loader communicates with a mobile phone which has a Bluetooth so that a connection can be established between the blood pressure loader and the mobile phone. Once a connection is established, an SMS can be sent to the doctor's phone informing him of any abnormal reading.

A smart phone is used to initiate a connection with the computer using a Bluetooth. Therefore the port number used to establish this connection must be stated. (the port number assigned to the smart phone I am using is com port 5). If the patient is only uploading his data without sending an alert, he will select 'Anyone'. In that case he does not need to type the doctor's name and phone number, he simply uploads his data, but if he is sending it to a particular doctor, then he needs to select "A doctor" and then types the doctor's name , phone number, the key and his message. He will then open his already saved CSV file and click ''Upload CSV Data Now''. The patient's information is uploaded automatically into the hospital's server.

Because this data travels over networks, it is encrypted. The patient enters the encryption key under ''key 1'' which appears as a plus sign. The key used for this encryption appears on the doctor's phone when he gets the message so that he can use that key to decrypt it.

4.22.2 The Sequence Diagram for the Blood Pressure (BP) Monitoring System

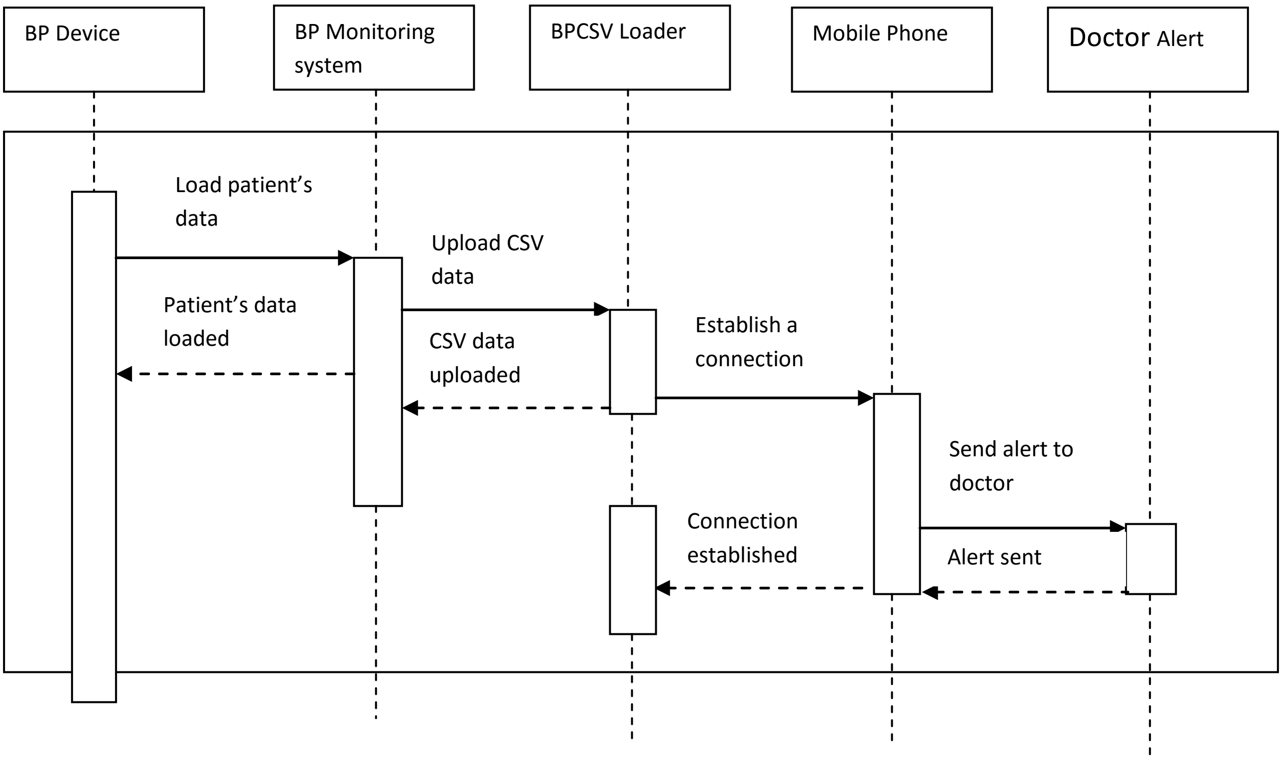


Figure 4.38: Sequence diagram for BP Monitoring System

The procedure of how the messages interact among components within the system is shown as follows:

i. The blood pressure and pulse rate readings are measured using the Omron 760IT blood pressure monitor. A Comma Separated Values (CSV) file is created of the readings which are then loaded in the blood pressure monitoring system.

ii. In the BP monitoring system, the patient's CSV data is uploaded into the Blood Pressure Comma Separated Values (BPCSV) loader.

iii. During the uploading of the readings to the BPCSV loader, the com port of the mobile phone used is set in the column provided and a connection is established with the mobile phone using the Bluetooth.

iv. Once a connection is established, an alert is sent to the doctor informing him of the patient's vital signs readings.

4.22.3 BP Monitoring Service Implementation

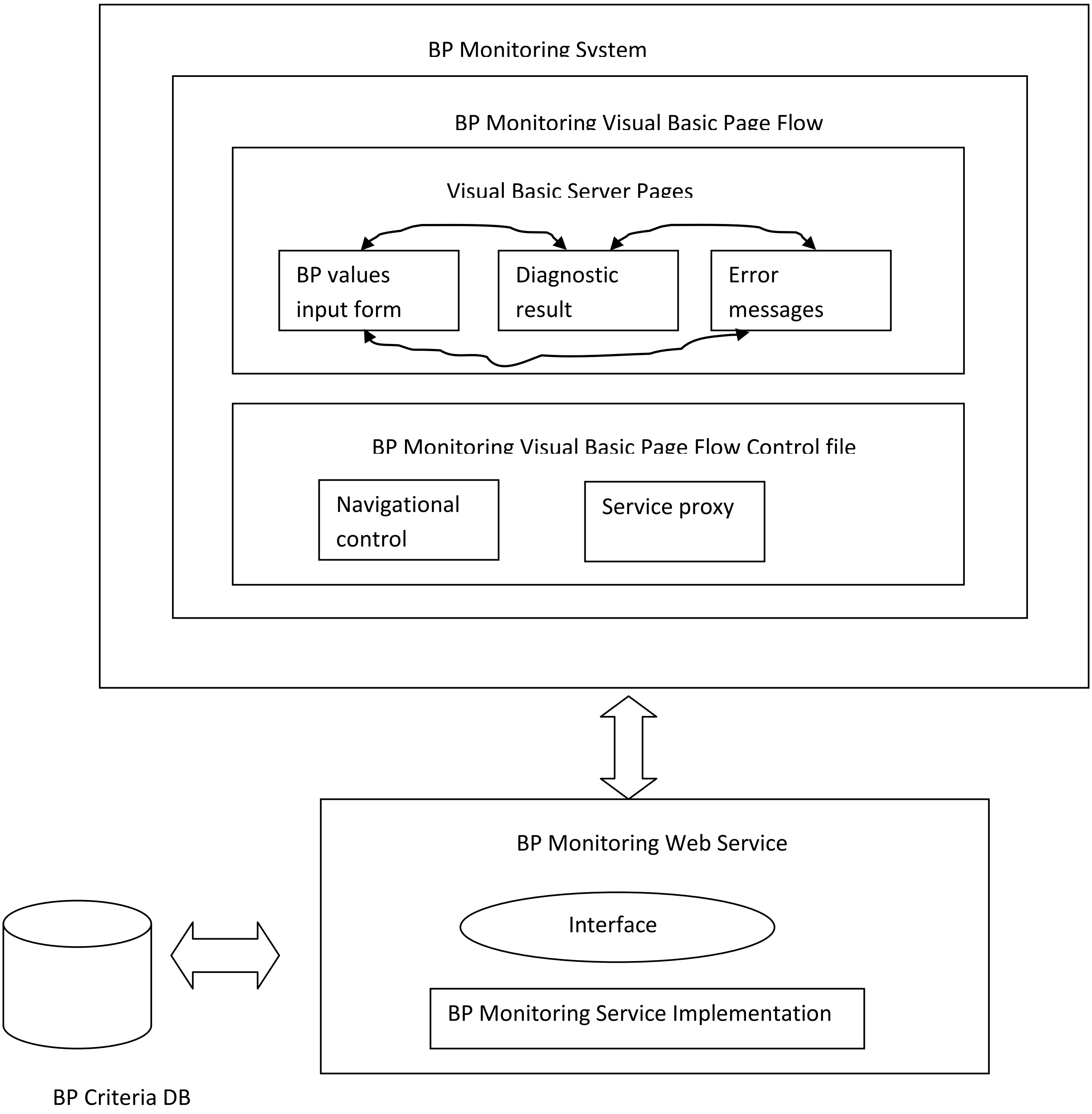


Figure 4.39: BP Monitoring Service Implementation

4.23 Applying Internet of things in Remote Patient Medical Monitoring System

In a remote patient medical monitoring system, it is pertinent to have the patients' medical records stored in the cloud for a more secured access. As a result, patients with chronic diseases may be less likely to develop complications, and acute complications may be diagnosed earlier than they would be otherwise. For example, patients suffering from cardiovascular diseases who are being treated with digitalis could be monitored around the clock to prevent drug intoxication. Arrhythmias that are randomly seen on an EKG could be easily detected and EKG data indicating heart hypoxemia could lead to faster detection of cardiac issues. The data collected may also enable a more preventive approach to healthcare by providing information for people to make healthier choices.

The patients are fitted with sensors that monitor the blood pressure and the pulse rate which in this case are simulated by the Intelligent Patient Digital Assistant (IPDA) which is a simulated mobile phone responsible for simulating these results.

The simulated data from the patients are sent through the hub to the cloud. These patients' medical records are stored in the cloud and both the nurses and the doctors can access the cloud to check the patients' data and make informed and timely decisions.

The Internet of Things in remote medical monitoring is shown in figure 4.40 where the patients' medical records are simulated in the Intelligent Patient Digital Assistant (IPDA). These records are encrypted in the security model, sent through the hub to the cloud where they are stored. The doctors can access the cloud when there is need to check on any patient's data and proffer solutions. This is also illustrated with the block diagram in figure 4.41 where the blood pressure, the pulse rate and the mean arterial pressure are simulated, sent through the hub which is the gateway to the cloud where the results are stored for easy access and reference.

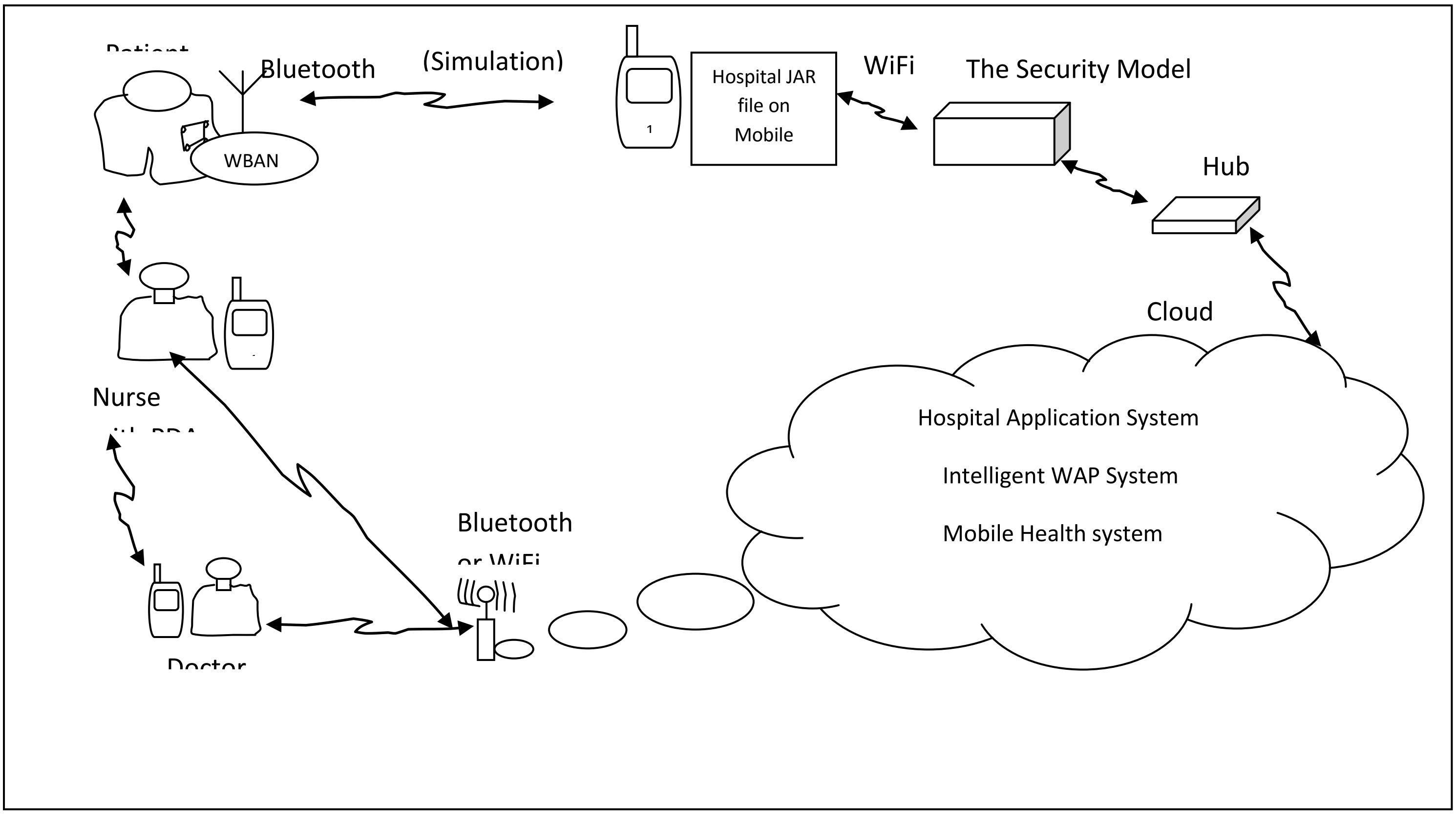


Figure 4.40: Internet of things in Remote Medical Monitoring

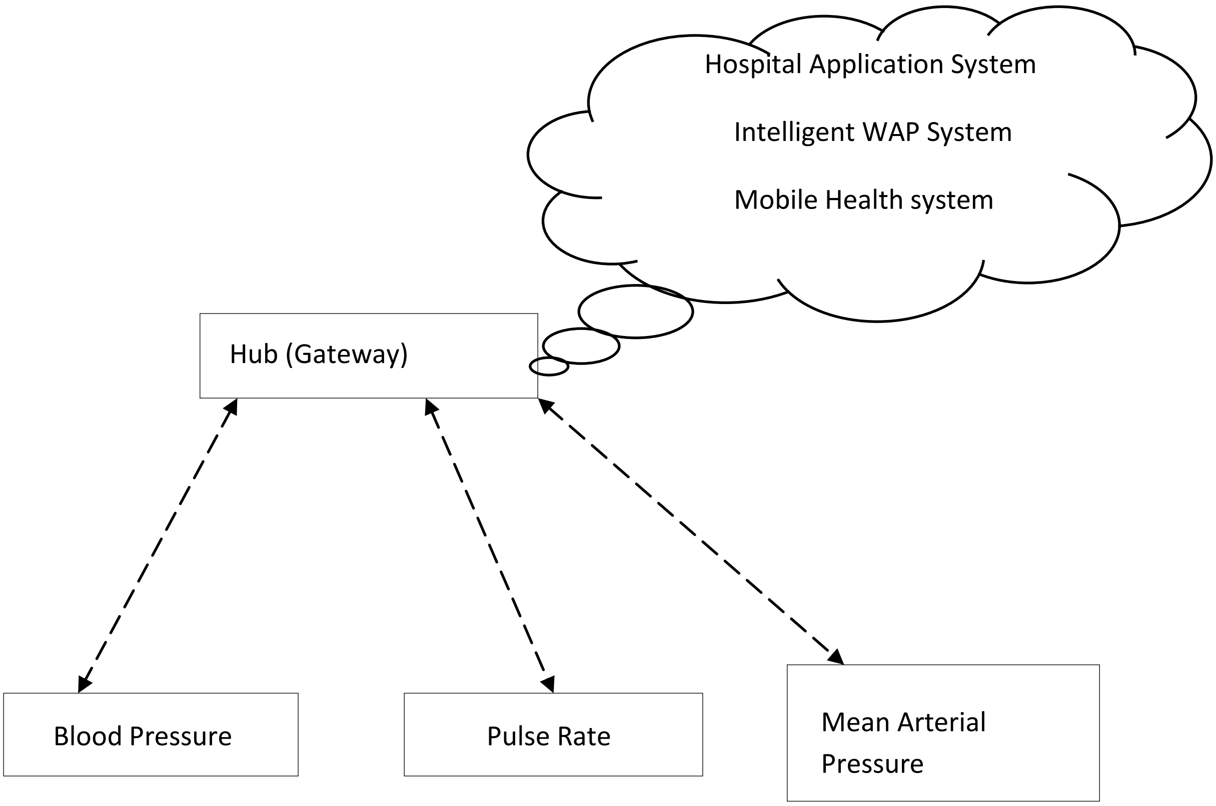


Figure 4.41: Block diagram of internet of things in remote medical monitoring

The blood pressure, the pulse rate and the mean arterial pressure are simulated, generated and sent to the hub. The hub in turn captures patient data from a variety of sensors and securely stores it in the cloud, where it can be accessed by those engaged in the patients' care.

Gateways are the information hubs that collect sensor data, analyze it and then communicate it to the cloud via wide area network (WAN) technologies. Gateways can be designed for clinical or home settings, in the latter, they may be part of a larger connectivity resource that also manages energy, entertainment and other systems in the home.

Wireless networking removes the physical limitations on networking imposed by traditional wired solutions like ethernet and USB.

4.24 The Security Architecture

As the patients' medical records travel across networks, it is petinent that they are protected against unauthorized access which might intercept the data along the way.

In view of this, the patients' medical records are encrypted as they travel across the networks, from the Intelligent Personal Digital Assitant to the server. This encryption is done by using the symmetric key encryption algorithm. In the encryption, the original text called the plaintext is encrypted using an encryption algorithm, to generate the encrypted text called the ciphertext that can only be read if decrypted.

The encryption scheme uses a pseudo-random encryption key generated by the algorithm. In the symmetric key scheme, the encryption and decryption keys are the same, thus communicating parties must have the same key before they can achieve secret communication.

The encrypted text which is the ciphertext is decrypted in the server by only authorized users using the decryption key which is the same key as the encrypted key.

In the security architecture, each patient is assigned a unique ID called PatientID (PID) which is generated using random number generation. The patientid generated is automatically encrypted. The patients’ data are simulated with the Simulated Patient Data Software(SPDS) and sent to the Patient Data Acquisition Centre (PDAC) through wireless connection for consolidation and further transmission to the Central Hospital Monitoring System (CHMS) for processing and follow up by the medical care givers.

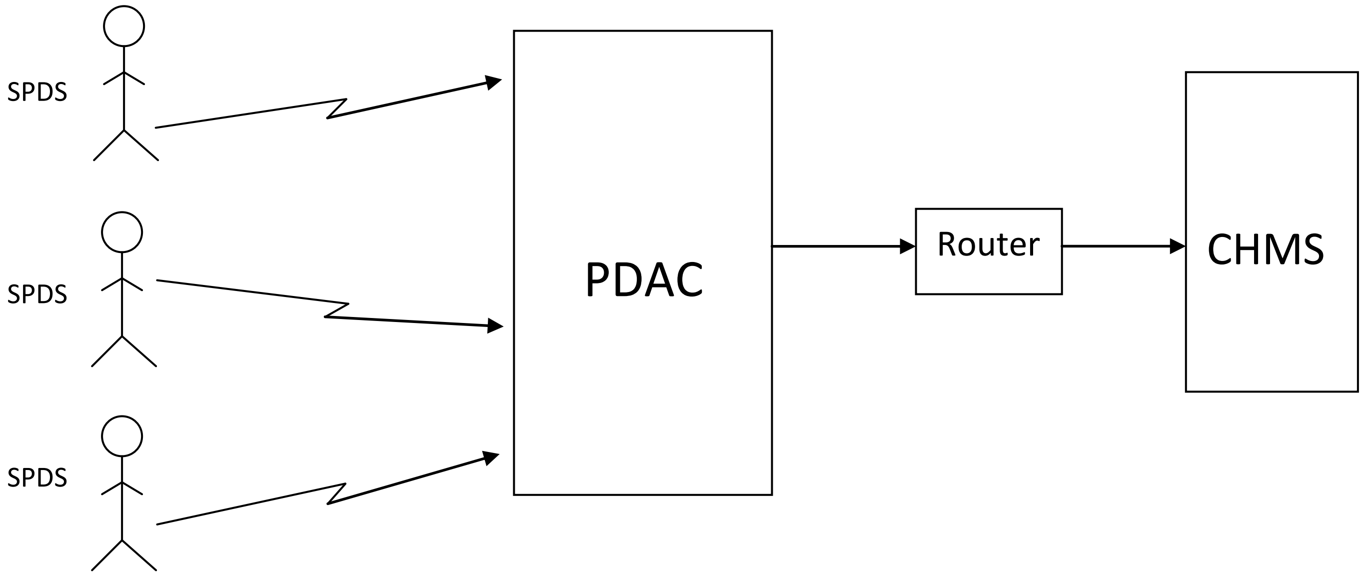


Figure 4.42: The Security Architecture

Each patient is identified with a unique ID called Patient ID (PID). The PID identifies the origin of the data.

The transmission of patient medical data between the PDAC and the CHMS can be carried out by using a wireless router or the internet. The patients can be at home carrying out their normal duties while their physiological parameters are being monitored. Medical personnel can also monitor the patient data and alert the experts in emergency situations.

Wireless transmission can be insecure and prone to data loss. Therefore the security of the patient data is of utmost importance and this is ensured by encryption and decryption of the patient data as it travels from the PDAC to the CHMS.

Two physiological parameters namely Blood Pressure and Pulse Rate are monitored.

The two parameters (Blood Pressure and Pulse Rate is generated for continuous monitoring of the patients and the encryption and decryption process was done by using the Symmetric Key Encryption Algorithm.

4.24.1 Components of a Security Model

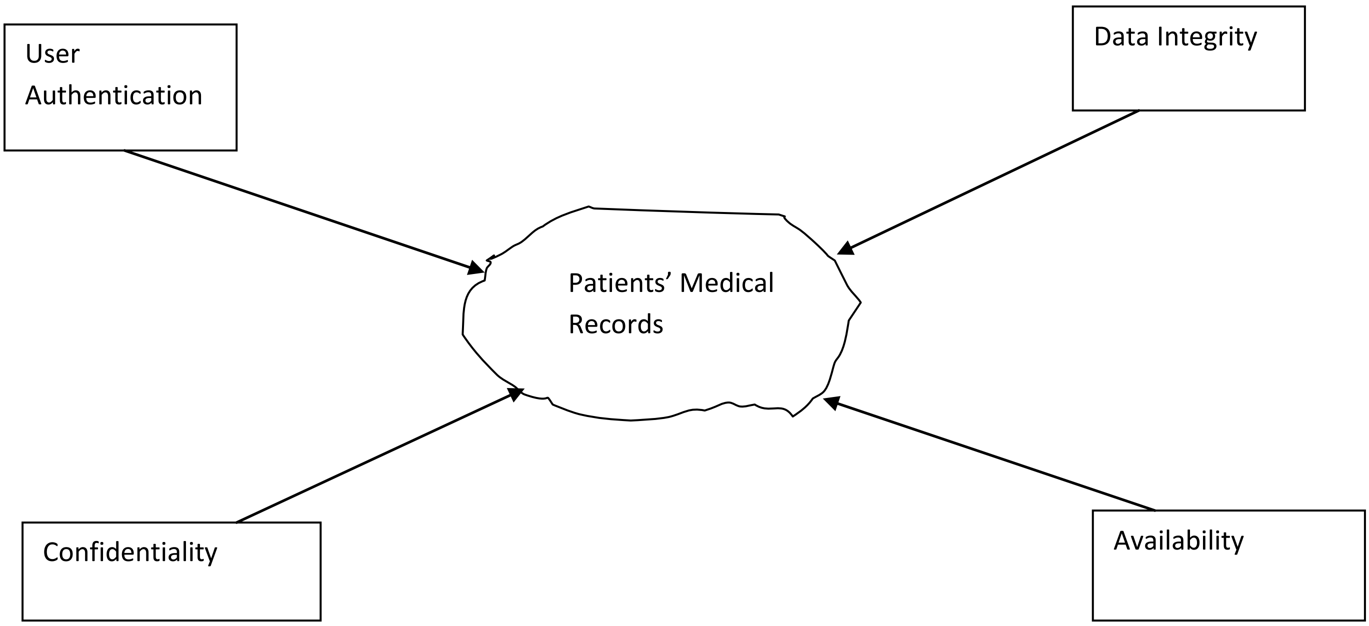


Figure 4.43 : Components of the Security Model

i. User Authentication: To provide privacy to patient data, the medical staff is validated using passwords.

ii. Data Integrity: To avoid data loss, corruption or malicious tampering, integrity of data is ensured by encryption algorithm.Integrity is usually provided by message authentication code or hashes.

iii. Confidentiality: This does not only apply to transmitted data but also to "secrets" held by the devices and is achieved using encryption. Encryption algorithms are used to convert plain text into cipher text and the equivalent decryption algorithm is used to convert the cipher text back to plain text. Symmetric encryption algorithms use the same key for encryption and decryption, while asymmetric algorithms use a public/private key pair.

iv. Availability: This ensures that the system can perform intended function without being disrupted by various technical or malicious cases such as mobile data service latency or quality of service problems.

4.24.2 Data Encryption and Authentication

The patient data sent from the PDAC to CHMS is encrypted using the symmetric key encryption algorithm and the secret key used is the generated key k_s. The block diagram of the encryption and authentication is given in figure 4.44.

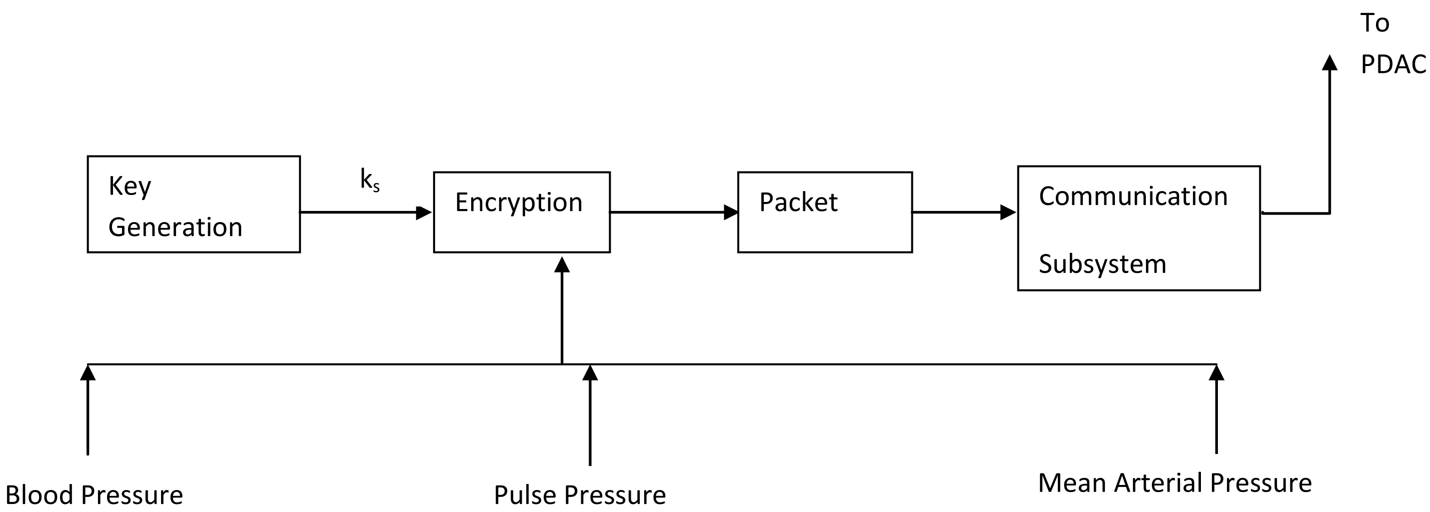


Figure 4.44: Encryption/Authentication at SPDS

The SPDS simulates the data for the various patients, consolidates them and encrypts the data using a secret key. The patient ID (PID) is also encrypted with the patients' data. The secret key, the patient ID with the patient data are all combined into a data packet and transmitted to PDAC.

On receiving the packet, the PDAC uses its secret key to compare and verify with the received secret key. This ensures the authenticity of the received data packet. PDAC then decrypts the packet. The block diagram of the decryption and authentication at PDAC is given in figure 4.45.

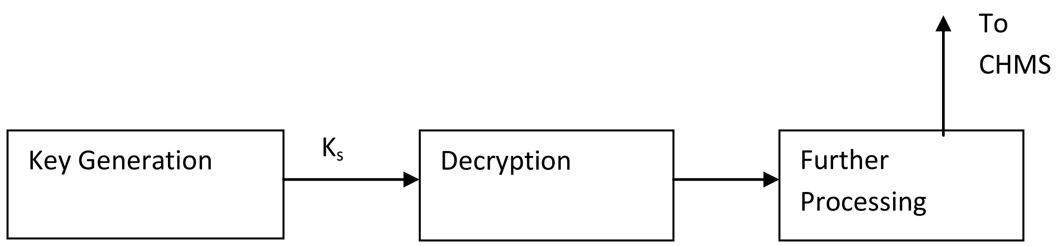


Figure 4.45: Decryption/Authentication at PDAC

PDAC also verifies the PID transmitted with the patient's data. It compares the received PID with the list of valid PID. If the PID is not valid, the packet is discarded.

4.24.3 Encryption Algorithm

Encryption algorithm, or cipher, is a mathematical function used in the encryption and decryption process – series of steps that mathematically transforms plaintext or other readable information into unintelligible cipher text. A cryptographic algorithm works in combination with a key (a number, word or phrase) to encrypt and decrypt data. To encrypt, the algorithm mathematically combines the information to be protected with a supplied key. The result of this combination is the encrypted data. To decrypt, the algorithm performs a calculation combining the encrypted data with a supplied key. The result of this is the decrypted data. If either the key or the data is modified, the algorithm produces a different result. The goal of the encryption algorithm is to make it as difficult as possible to decrypt the generated ciphertext without using the key.

Each algorithm uses a string of bits known as a "key" to perform the calculations. The larger the key (the more bits), the greater the number of potential patterns can be created, thus making it harder to break the code and descramble the contents. A cryptographic algorithm or cipher is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key – a word, number or phrase to encrypt the plaintext. The same plaintext encrypts to different ciphertext with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key.

4.24.4 A Symmetric Key Algorithm by Ayushi, Lecturer, Hindu College of Engineering 2010

4.24.4.1 Encryption Algorithm

Step 1: Generate the ASCII value of the letter

Step 2: Generate the corresponding binary value of it. [Binary value should be 8 digits]

Step 3: Reverse the 8 digit's binary number

Step 4: Take a 4 digits divisor (>= 1000) as the key

Step 5: Divide the reversed number with the divisor

Step 6: Store the remainder in first 3 digits and quotient in next 5 digits (remainder and quotient wouldn't be more than 3 digits and 5 digits long respectively. If any of these are less than 3 and 5 digits respectively we need to add required number of 0s (zeros) in the left hand side. So this would be the cipertext i.e encrypted text.

Now store the remainder in first 3 digits and quotient in next 5 digits.

4.24.4.2 Decryption Algorithm

Step 1: Multiply last 5 digits of the ciphertext by the key

Step 2: Add first 3 digits of the ciphertext with the result produced in the previous step

Step 3: if the result produced in the previous step i.e. step 2 is not an 8-bit number, we need to make it an 8-bit number

Step 4: Reverse the number to get the original text i.e. the plain text.

The aim of this symmetric algorithm written by Ayushi was to make it cost effective since according to him previous symmetric key algorithm did not consider that fact but unfortunately, the algorithm was simply designed for a small amount of data and will not work well when the data is large. It uses an 8 bit binary number to work which is quite small. Since the algorithm has an 8-bit key, the number of potential patterns to be created is very small thus making it easier to break the code and descramble the contents.

4.24.5 The Encryption/Decryption Algorithm

1.) Create the instance of javax.crypto.Cipher.

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

2.) Create the instance of sun.misc.BASE64Decoder.

BASE64Decoder d64 = new BASE64Decoder();

3.) Decode encryption key using decodeBuffer method of sun.misc.BASE64Decoder which will return a byte array.

byte[]b = d64.decodeBuffer(ENCRYPTION_KEY);

4.) Now create javax.crypto.spec.SecretKeySpec object using the key byte array and "AES".

SecretKeySpec key = new SecretKeySpec(strPassword.getBytes(), "AES");

5.) Now create javax.crypto.spec.IvyParameterSpec object using your IV spec.

AlgorithmParameterSpec paramSpec = new IvParameterSpec (strPassword.getBytes());

6.) Call the init method of Cipher Instance using encryption mode, key spec (created at step 4), IVParameter Spec(created at step 5).

cipher.init(Cipher.ENCRYPT_MODE, key, paramSpec);

7.) Call doFinal method of Cipher by passing the byte array of data which needs to be encrypted.

encrypted = cipher.doFinal(input.getBytes());

8.) Now encode the encrypted data using sun.misc.BASE64Encoder.

output = new BASE64Encoder().encode(encrypted)

NOTE ABOUT KEY SIZE:

The AES algorithm requires that the key size must be 16bytes. So if you provide a key whose size is not equal to 16bytes, a java.security.InvalidKeyException will be thrown.

This algorithm uses cipher which is a sprat class in the javax.crypto package that is specially designed for encryption and decryption. It provides the functionality of a cryptographic cipher for encryption and decryption. To create a cipher object, the application calls the cipher's getInstance method and passes the name of the requested transformation to it. The cipher getInstance method employs a random length padding which prevents an attacker from knowing the exact length of the plaintext message. This strengthens the algorithm because it prevents the cryptanalyst from using predictability to find known plaintext that can aid in breaking the encryption.

The base64Decoder converts the base64 encoded data into binary data and returns a byte array. The base64Decoder is responsible for displaying the encrypted files in the form of a ciphertext.

The algorithm uses a 16 bit key which is entered by the user and encrypted. A secret key spec and an algorithm parameter spec is created which validates the key input to make sure that it is the right key for the Advanced Encryption Standard algorithm to take place.

An init method which is found in the cipher class is called upon which ties the key to the encryption. The init method checks that the key supplied tallies with the key for the encryption.

The do final method is also a method inside the cipher class which is the final step of encryption. It gets the users input and converts it to byte which it stores in the byte array.

The Encode which is a method inside the base64Encoder displays the encrypted output.

This algorithm uses a 16-bit key and therefore the number of potential patterns to be created is quite large and this makes it difficult to break the code and descramble the contents.

This algorithm uses block ciphers which encrypts the data one fixed-size block at a time. The block cipher spreads a single bit of the plaintext across the entire encrypted block. Modes are employed in the encryption to combine the block cipher with the plaintext. The cipher block chaining (CBC) is the type of mode used in this encryption. CBC combines the previous block of cipher text with the next block of plaintext before encrypting it. A random value – an initialization vector is applied to the first block before it is encrypted. This chaining mechanism means the encryption of each block depends on the encryption of all previous blocks. CBC helps detect modifications to the cipher text.

The algorithm performs key distribution by using a public key to encrypt a randomly generated secret key. Then the recipient uses the corresponding private key to decrypt it.

Since the algorithm employs a random length padding (AES/CBC/PKCS5Padding), it prevents an attacker from knowing the exact length of the plaintext message. This strengthens the algorithm because it prevents the cryptanalyst from using predictability to find known plaintext that can aid in breaking an encryption.

Padding is used to prepare for the encryption using a specification such as PKCS5. This is achieved by appending random material to the message to make it long enough for the primitive. This padding scheme aims to ensure that the attacker cannot manipulate the plaintext to exploit the mathematical structure of the primitive that breaking the padding scheme is as hard as solving the hard problem underlying the primitive.

In the PKCS5 (Public Key Cryptographic Standards 5) padding used in this encryption, the last block is padded with integers, each of which is the number of bytes used to pad the message. The PKCS5 is used by padding with bytes all of the same value as the number of padding bytes. The input is padded with a padding string of between 1 and 8 bytes to make the total length an exact multiple of 8 bytes. The value of each byte of the padding string is set to the number of bytes added: i.e 8 bytes of value 0x08, 7 bytes of value 0x07, 6 bytes of value 0x06, 5 bytes of value 0x05, 4 bytes of value 0x04, 3 bytes of value 0x03, 2 bytes of value 0x02 or 1 byte of value 0x01.

4.24.6 **Block Cipher**

Block cipher is one of the methods of producing ciphertext (and is the type used for this algorithm), the other being stream cipher. Block ciphers are designed to encrypt data in chunks of a specific size as shown in figure 4.46 below. A block cipher specification will identify how much data should be encrypted on each pass (called a block) as well as what size key should be applied to each block. There exist a number of different algorithms that can be used when processing block cipher encryption. The most basic is to simply take the data and break it up into blocks while applying the key to each block.

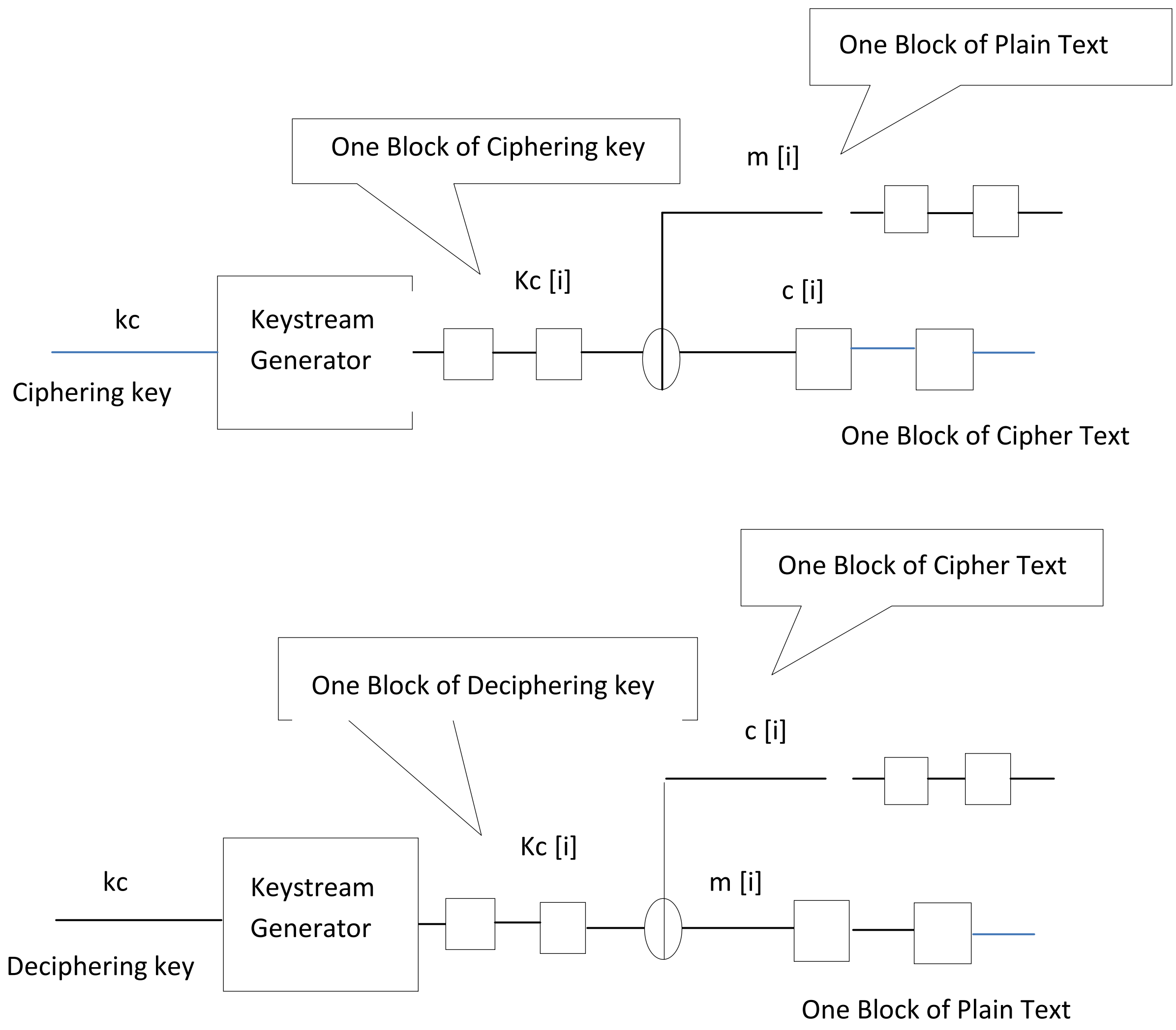


Figure 4.46: Block ciphering and deciphering

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Introduction

Generally implementation of the software is considered as the actual creation of the software. Since system design stage usually suggest that the interface, data and actual output are created, the implementation stage brings them all together.

It is also at this point that differences of Software Development Life Cycle (SDLC) models are noticed. Some treat system design as the actual creation of the software.

The different types of designs are placed together here. The final constructions of the designs are also combined at this stage.

The system implementation is actually the stage where the different resources needed to make the software operational are determined. At this, stage, the different hardware and software requirements for the project are stated, the choice of the programming language used and why that programming language is preferred is also stated.

During the system implementation, the changeover procedure is stated, the users to be trained and the type of training to be given to the users are also determined. The installation manual and the procedure for the installation of the software are also added at this stage.

This stage becomes an important phase of the project since it will be the first time that bugs and errors in the system are determined. During this stage, different type of errors will be determined. It will gauge how far the developer has advanced in building the software. It is also proper that these errors are fixed.

Even though it is not mentioned before, this stage will mark importance of documentation of the software.

Documentation is very important especially in this stage as bugs are determined. Sometimes the developer will even find out that certain expected functions are not working. Instead of scrutinizing the software all over again, the developer will go back to the documentation and see which part of the program has gone wrong. Documentation will also give the developer a chance to re-evaluate the process he/she has created.

5.2 System Requirements

5.2.1 Hardware Requirements

i. Processor

In addition to the minimum processor required by the operating system you are running, this program requires an 800 MHz Pentium III processor or equivalent.

ii. Graphics Display Card/Sound Card

A standard graphics display card that can display at 1024x768 resolution is adequate. A sound card is optional for audio playback if necessary.

iii. RAM

In addition to the minimum RAM required by the operating system you are running, this program requires a minimum of 512 MB of RAM; However 1 GB is recommended.

iv. Disk Space

Your computer must have a minimum of 1.5 GB of hard disk space available.

5.2.2 NetBeans™ IDE Version 7.0 (All Bundle)

This program includes the installer for the NetBeans IDE and requires the following:

NetBeans IDE is written in pure java. It should run on any working implementation of java 2 SDK, Standard Edition. To use the NetBeans IDE installer you must have the J2SDK installed.

i. Most recent (6.0) version of Sun's Java™ 2 SDK installed, prior to installing this software.
ii. The NetBeans IDE is supported on Microsoft Windows XP Professional SP3 and Windows Vista. The IDE is known to run on Microsoft Windows 2000 Professional SP4 as long as the hardware requirements are met.
iii. The IDE's minimum screen resolution is 1024x768 pixels.

Hardware Requirements

Minimum

i. 800 MHz Intel Pentium III workstation or equivalent
ii. 512 MB of memory
iii. 750 MB of free disk space
iv. The IDE's minimum screen resolution is 1024x768 pixels.

Recommended:

i. 2.6 GHz Intel Pentium III workstation or equivalent
ii. 1 GB of memory
iii. 1 GB of free disk space

5.2.3 Software Requirements

Operating System

You must be using one of the following versions of Windows:

i. Windows 7
ii. Windows Vista
iii. Windows XP (with Service Pack 2)

The Software developed

Two types of software were developed:

The client end (Mobile Health Information Management System)

The server end (Integrated Health Information Management System)

The client end is the software on the mobile phone and the size of this software is 48kb, while the server end is the software on the server and the size of this software is 50mb.

5.3 Program Development

This research is analysed using the methodologies: Structured Systems Analysis and Design Methodology (SSADM) and Object Oriented Analysis and Design Methodology (OOADM).

The modules that are used are:

Main MODULES (Control Centre)

i. Out Patient Department (OPD)
    - i. In-Patient Department (IPD)
    - ii. Clinical Support
    - iii. Hospital Administration
    - iv. Ancillary Services
    - v. Operation Theatre

ii. EMR

iii. CDSS

After the design, the program was coded using the java programming language. Java was chosen because the program is a web based program and so java is the best programming language to use in this case.

Java was also chosen because of its many benefits which include:

i. Write Once, Run Anywhere

Sun identifies "Write once, run anywhere" as the core value proposition of the Java platform. Translated from business jargon, this means that the most important promise of Java technology is that you only have to write your application once--for the Java platform--and then you'll be able to run it *anywhere*.

Anywhere, that is, that supports the Java platform. Fortunately, Java support is becoming ubiquitous. It is integrated, or being integrated, into practically all major operating systems. It is built into the popular web browsers, which places it on virtually every Internet-connected PC in the world. It is even being built into consumer electronic devices, such as television set-top boxes, PDAs, and cell phones.

ii. Security

Another key benefit of Java is its security features. Both the language and the platform were designed from the ground up with security in mind. The Java platform allows users to download untrusted code over a network and run it in a secure environment in which it cannot do any harm: it cannot infect the host system with a virus, cannot read or write files from the hard drive, and so forth. This capability alone makes the Java platform unique.

The Java 2 Platform takes the security model a step further. It makes security levels and restrictions highly configurable and extends them beyond applets. As of Java 1.2, any Java code, whether it is an applet, a servlet, a JavaBeans component, or a complete Java application, can be run with restricted permissions that prevent it from doing harm to the host system.

The security features of the Java language and platform have been subjected to intense scrutiny by security experts around the world. Security-related bugs, some of them potentially serious, have been found and promptly fixed. Because of the security promises Java makes, it is big news when a new security bug is found. Remember, however, that no other mainstream platform can make security guarantees nearly as strong as those Java makes. If Java's security is not yet perfect, it has been proven strong enough for practical day-to-day use and is certainly better than any of the alternatives.

iii. Network-centric Programming

Sun's corporate motto has always been "The network is the computer." The designers of the Java platform believed in the importance of networking and designed the Java platform to be network-centric. From a programmer's point of view, Java makes it unbelievably easy to work with resources across a network and to create network-based applications using client/server or multitier architectures. This means that Java programmers have a serious head start in the emerging network economy.

iv. Dynamic, Extensible Programs

Java is both dynamic and extensible. Java code is organized in modular object-oriented units called *classes*. Classes are stored in separate files and are loaded into the Java interpreter only when needed. This means that an application can decide as it is running what classes it needs and can load them when it needs them. It also means that a program can dynamically extend itself by loading the classes it needs to expand its functionality.

The network-centric design of the Java platform means that a Java application can dynamically extend itself by loading new classes over a network. An application that takes advantage of these features ceases to be a monolithic block of code. Instead, it becomes an interacting collection of independent software components. Thus, Java enables a powerful new metaphor of application design and development.

v. Internationalization

The Java language and the Java platform were designed from the start with the rest of the world in mind. Java is the only commonly used programming language that has internationalization features at its very core, rather than tacked on as an afterthought.

While most programming languages use 8-bit characters that represent only the alphabets of English and Western European languages, Java uses 16-bit Unicode characters that represent the phonetic alphabets and ideographic character sets of the entire world. Java's internationalization features are not restricted to just low-level character representation, however. The features permeate the Java platform, making it easier to write internationalized programs with Java than it is with any other environment.

vi. Performance

As I described earlier, Java programs are compiled to a portable intermediate form known as byte codes, rather than to native machine-language instructions. The Java Virtual Machine runs a Java program by interpreting these portable byte-code instructions. This architecture means that Java programs are faster than programs or scripts written in purely interpreted languages, but they are typically slower than C and C++ programs compiled to native machine language. Keep in mind, however, that although Java programs are compiled to byte code, not all of the Java platform is implemented with interpreted byte codes. For efficiency, computationally intensive portions of the Java platform--such as the string-manipulation methods--are implemented using native machine code.

Although early releases of Java suffered from performance problems, the speed of the Java VM has improved dramatically with each new release. The VM has been highly tuned and optimized in many significant ways. Furthermore, many implementations include a just-in-time compiler, which converts Java byte codes to native machine instructions on the fly. Using sophisticated JIT compilers, Java programs can execute at speeds comparable to the speeds of native C and C++ applications.

Java is a portable, interpreted language; Java programs run almost as fast as native, non-portable C and C++ programs. Performance used to be an issue that made some programmers avoid using Java. Now, with the improvements made in Java 1.2, performance issues should no longer keep anyone away. In fact, the winning combination of performance plus portability is a unique feature no other language can offer.

vii. Programmer Efficiency and Time-to-Market

The final, and perhaps most important, reason to use Java is that programmers like it. Java is an elegant language combined with a powerful and well-designed set of APIs. Programmers enjoy programming in Java and are usually amazed at how quickly they can get results with it. Studies have consistently shown that switching to Java increases programmer efficiency. Because Java is a simple and elegant language with a well-designed, intuitive set of APIs, programmers write better code with fewer bugs than for other platforms, again reducing development time.

5.4 Program Testing

The program will be tested by testing the different modules using test data to make sure they are doing what they are supposed to do.

The application on the mobile phone is tested by filling in the form, typing your user name and password and clicking on submit.

This generates the values for the blood pressure and pulse rate and sends it to the server.

The application in the server is tested by supplying raw data to the different modules, which generates the result. These are all shown in the appendix.

5.5 Changeover Procedure

The parallel changeover is used in this case whereby both the old system and the new system are run concurrently using the same inputs. The outputs are compared and reasons for differences resolved.

The changeover procedure involved in using the new system includes:

i. Getting the different monitoring devices to monitor the vital signs of the patients. The hospitals shall acquire as many monitoring devices as possible.
ii. Acquiring computers
iii. Connecting the computers through a Local Area Networks or a Wide Area Networks as the case may be.
iv. Creating a website for the hospital, hosting the website and then uploading the information to the website.
v. Training of the staff responsible for managing the system.
    a. The staff to be trained include the nurses, the doctors and the care givers.
    b. As the need arises, patients and sometimes their relations need to be educated on how to use the monitoring devices because in some cases the patients are at home or in a remote location and therefore need to know how to operate their phones in order for the vital signs to be monitored.

c. Providing adequate security to the program and the database.

5.5.1 **Training of Users**

As listed above, the users to be trained are: the nurses, the doctors and the care givers. The users are to be trained on how:

i. Install the mobile health management information system in the patient's mobile phone.
ii. Access the server to get information concerning the patients when the need arises.
iii. The doctor is to be trained on how to access the queuing model and how to respond when he gets any information regarding the patient.
iv. The nurse is to be educated on how the mobile application works and she in turn should always train the patients on how to use the mobile application on their mobile phones.
v. The patients or care givers should be trained on how to fill the form on their mobile phones when the application is installed there.

vi. Fill in their user name and password in the phone when prompted to do so.

5.6 **System Security**

The information that will be gotten from this program is the health information of patients and therefore need to be safe guarded against unauthorised access. The security of the entire system must therefore be of utmost concern.

The security of the system is achieved by doing the following:

i. Limiting the access to the information so that it is only authorised personnel with password that can have access to the information.
ii. The patient's information that is sent from the mobile phone via the router to the server is encrypted. It is also decrypted in the queuing model and which ever doctor getting information about a patient's data is given the decryption key with which he can access the data.

5.7 **Users Guide**

5.7.1 Installation Arrangement

i. Installing the Software
ii. Installing NetBean 7.0
iii. Copying Application and Setting the path for the hospital database

Step 1: Copying Application

Copy the folder inthospital to C: prompt or any other hard drive you choose to run the application from.

Step 2:

i. Load the control panel application
ii. Double click the control panel
iii. Double click the administrative tools
iv. Double click the data source (ODBC) icon
v. Click on the add button

vi. Select Microsoft Access Driver (*.mdb) option
vii. Click finish
viii. On the Data source label, type inthospitalDB
ix. Click on the select button and browse to the inthospital/src/inthospital/inthospitalDB folder in the hard disk (C:). Select inthospitalDB.mdb.
x. Click ok and ok again to complete the task.

Running Application (How to use the software)

Step 1: Start the Netbean 7.0 from the start button

Select Open project from the file menu
Browse the hard drive and locate the inthospital folder
Click the open project folder button

Step 2: Building and compiling application

Click on the build menu and select Build Main Project or Clean and build main project.
Click on Run menu and select run main project

This will compile all classes used and debug the application to acquire the new http port. This automatically makes the system the active server machine.

Step 3: Using/Test the application in Client Computer(s)

On the client computer, do the following:

i. Load any web browser e.g. Internet explorer, Mozilla fire fox, Opera etc.

ii. At the browser address label, type http://localhost:8084/inthospital or http://127.0.0.1:8084/inthospital.
iii. Use appropriate links to navigate.
iv. Connect client(s) to the server computer
v. Make sure a working network installation of the system is on.
vi. At the client web browser address textbox, type the server IP address e.g. http://192.168.0.1:8084/inthospital/ if the server have been configured with 192.168.0.1 IP address.

Special Information:

Make sure other server applications are not using the http 8080 otherwise you must change to the displayed http port number as generated by the server machine.

Running the Mobile Application

i. The steps to follow are the same with running the application on the server. The only difference is that instead of clicking Inthospital, you click Mobile.
ii. Click on run and click clean and build
iii. Click on run again and click run main project
iv. The mobile phone prototype appears on the screen
v. Fill the form in the phone and click submit
vi. The application generates the blood pressure and the pulse rate and sends it to the server.

5.8 Hosting on the Internet

Website hosting begins with purchasing your own URL (uniform resource locator), and to do that you visit an online company offering domains for sale.

Next you have to upload the site to a web-hosting company that will display it on the Internet, 24 hours a day, seven days a week. Thousands of such web-hosting companies exist, and there are now web-hosting directories that enable you to select by cost, platform type, facilities, etc. — all of which are explained by on-site notes.

You make your choice of hosting company, click through to their site, pay their hosting fee, and can then upload your site to that company's server. The hosting company will provide instructions. It's very simple, but you'll need a cheap or free piece of software called an ftp program. This you can obtain from any software supplier, and use it to maintain your site thereafter. Once uploaded, your site goes 'live'. You're on the Internet.

CHAPTER 6

SYSTEM TESTING AND EVALUATION

6.1 Introduction

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

6.2 Test Plan

6.2.1 Program Testing Procedure:

6.2.1.1 Loading the Mobile Health Software

i. Lunch the NetBeans IDE 7.0 application
ii. Click on File-Open Project
iii. Locate the Project folder “MobileHos” in drive C.
iv. Click Open Project button

v. Click on Run-Run Main Project or Press F6. This compiles the MobileHos application and also establish connection to the tomcat server and the Application IDE will be lunched in the default web browser (Internet Explorer, Firefox, Opera, Netscape Navigator or Apple Mosiac).

6.2.1.2 Activating the Server

i. Lunch the NetBeans IDE 7.0 application
ii. Click on File-Open Project
iii. Locate the Project folder "inthospital" in drive C.
iv. Click Open Project button
v. Click on Run-Run Main Project or Press F6. This compiles the server application and also establish connection to the tomcat server and the Application IDE will be lunched in the default web browser (Internet Explorer, Firefox, Opera, Netscape Navigator or Apple Mosiac).

6.2.1.3 Activiating client connection

i. If the system is on a LAN, the IP address for the server will be http://localhost:8084/inthospital/. In most cases, the server true IP may look like this http://192.168.1.1:8084/inthospital . In a LAN setting, the client(s) IP address should be configured as 192.168.1.N where N range from 2 to 255 and the file sharing and printing options must have been enabled. This will help for the smooth running of this application in a LAN. For a WAN setting, the application and its corresponding modules must be published in the internet.
ii. Lunch the client browser and type the IP address of the server in this format: http://192.168.1.1:8084/inthospital.
iii. Navigate the interface(s) via the hyperlink at both the top and the left side of the display.
    - The following interfaces and their corresponding program codes highlight the operations of the test application.

6.3 Test Data

The test data that is used for this thesis is generated by using a simulation to get the blood pressure and the pulse rate using known causes for both high and low blood pressure and high and low pulse rate. Different parameters are used to generate these test data. The parameters used are:

i. Name
ii. GSM/Phone No
iii. Hospital Id
iv. Category of Blood pressure (1 – High, 2 – Normal, 3 – Low)
v. Age
vi. Drinking Habits
vii. Smoking Habits

These parameters are used for the simulation of the blood pressure and the pulse rate.

6.4 Actual Versus Expected Test Result

Actual Test Result

Table 6.1: Actual Test Result

Patient Name	Telephone No	Patient Id	Age	Category of Blood Pressure	Drinking Habits	Smoking Habits
Okeke Ngozi	08012345678	PID001	89	1	Y	Y
Ojiewulu Nneka	09023415678	PID002	70	2	N	N
Adebayo Olufemi	08023418769	PID003	76	2	Y	N

6.5 Expected Test Result

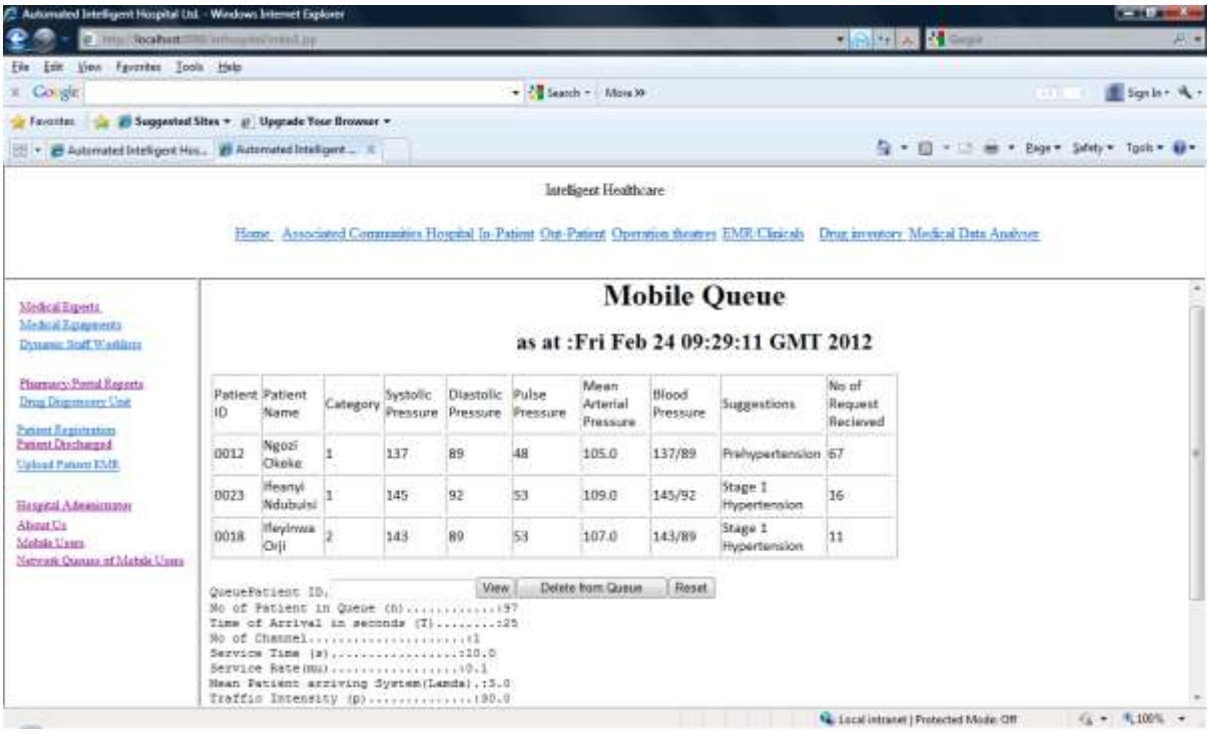


Figure 6.1: Expected Result

6.6 Performance Evaluation

The expected result performs a simulation using the actual test result to generate the parameters for the systolic pressure, the diastolic pressure, the pulse pressure, the mean arterial pressure and the blood pressure. It also gives suggestions on the state of the patient’s blood pressure specifies the date and time the simulation was carried out.

It gives analysis of the patients’ record on the queue like the number of patients in queue, the time of arrival, the number of channels, the service time, service rate and traffic intensity.

6.7 Limitations

The limitations of this project are as follows:

i. It was very difficult getting a monitoring device that can monitor the vital signs of patients which has a Bluetooth and which is able to transmit the result automatically to the IPDA. Most of the monitoring devices in the market can monitor the vital signs but cannot send to any monitoring station. To overcome this limitation the researcher had to do simulation in order to generate the readings.

ii. During the feasibility study, the nurses in the Central Hospital were not very supportive in divulging information about how their system work initially but after much persuasion, the researcher got all the information she needed about how their system work.

iii. Remote Medical Monitoring is a very new area here in Nigeria, so it was very difficult getting materials especially during the literature review but eventually, the researcher was able to gather some material.

CHAPTER 7

SUMMARY AND CONCLUSION

7.1 Summary

This project has been designed to monitor the vital signs of patients from a remote location for the doctor in charge to be aware of the health status of the patient at any point in time and to proffer a solution. There are many of these vital signs among which are blood pressure, pulse rate, respiration, temperature, ECG etc. Out of these vital signs, the researcher worked on blood pressure and pulse rate.

The researcher was able to do a simulation to get the readings for the blood pressure and the pulse rate. This simulation was done on a mobile phone and transferred to a queuing model built by the researcher which is in the server. The queuing model is also able to send health information concerning any patient when it gets an abnormal result concerning that patient to the medical personnel for immediate medical intervention.

Different models were built to achieve this, ranging from the queuing model, the transmission model of communication, the security model and the network model.

Two different types of software were developed. They are: the **Mobile Health Information Management System** which is the software that runs on the mobile phone and which is able to simulate the patients' vital sign and send it to the queuing model in the server. The other software is the **Integrated Hospital Information Management System** which runs on the server and is able to perform other hospital functions. It is made up of the hospital application, the Electronic Medical Records (where the patients' health information are stored for referencing) and the hospital database. It also contains the vital sign uploading model where the nurse can manually send the vital sign of patients within the hospital premises to the doctor anywhere he is by sending an alert to his phone.

A third software was also developed called the **Simulated Patient Data Software (SPDS),** which generates patient id and encrypts it togheter with the patients' medical records before sending it to the server.

7.2 Review of Achievement

The system developed has the capability of doing the following:

i. Remote monitoring of the vital signs of patients. The patients may be in a remote location or the doctor may be in a remote location from where he can monitor the vital signs of these patients and proffer immediate solution. This is possible as long as there is telephone or internet facility in the location.
ii. Establish a feedback mechanism in which a message can be sent to the doctor's phone if there is an emergency and the doctor in turn can send a message to the caregiver instructing him on what to do.
iii. The automated mobile health information management system is able to simulate the readings of the blood pressure and the pulse rate of patients and send them to the queuing model in the server
iv. The automated intelligent hospital software designed is able to capture the results queuing model directly into its database.
v. There is real time monitoring of the patients' blood pressure and pulse rate. these readings are generated every 25 seconds and sent directly to the queuing model in the server.
vi. The blood pressure loader developed has an alert system which is activated once the patient's readings becomes abnormal. In such a situation, an alert is sent directly to the doctor so that wherever he is, he can proffer a solution on what should be done.

7.3 Areas of Application

The software will be useful in all hospitals and clinics both big or small. It will also be applied in health centers and in any health institution where there are patients and there is need to monitor their vital signs.

It will be very useful in these health institutions because it will drastically reduce the incidence of avoidable deaths in Nigerian hospitals and health institutions.

7.4 Suggestions for Further Work

i. The researcher has been able to simulate and model a remote medical monitoring system using simulated values and developing two major applications to both simulate and monitor these vital signs.

ii. The researcher is therefore suggesting that further work should be done to be able to design and implement a remote medical monitoring system whereby the monitoring device is designed and developed and the patient wears this monitoring device and the device is able to collect the data and send it to computer using Bluetooth for onward transmission to the doctor's laptop for immediate medical intervention.

iii. Further work also involves including more sensors to monitor other vital signs such as oxygen saturation level, ECG, respiration etc.

iv. Also remote medical monitoring is just one of the three ways telemedicine can be achieved. The others being store and forward and interactive services. Therefore the researcher is suggesting that work should be done in these areas so as to enhance the use of telemedicine in Nigeria.

7.5 Contributions of the Work to the Body of Knowledge in Computer Science

i. The major contribution of this thesis to knowledge is the incorporation of the security architecture which safeguards and protects the patients' medical records as they travel between networks. This was achieved by developing a software called Simulated Patient Data Software (SPDS) which encrypts the patients' medical records before they are sent to the database in the server and decrypts it by carrying out an authentication process to determine if the user is an authorized personnel. The encryption was done by using a 16-bit symmetric key encryption algorithm which employed the Advanced Encryption Standard to encrypt and decrypt the patients' records. A novel encryption/decryption algorithm was developed to achieve this. This bridges the gap identified in the work done by Andrew and Alfred (2008), Eliasz et al (2011), Mehmet et al (2007) and Istepanian et al (2004). These people, even though they worked on Remote Medical Monitoring, did not consider the security of the patients' medical records as they travel through the networks. When patients' medical records are not secured, they are prone to unauthorized access, eavesdropping and data loss. This thesis has been able to secure the patients' medical records as they travel through the networks giving the medical personnel confidence that their patients' medical records are protected.

ii. The queuing model was enhanced and adapted into the application and used for the analysis and evaluation of the vital sign parameters before being transmitted to the medical center. This will enable the doctor to take timely decisions that has the capacity to save lives. This bridges the gap in the work designed by Mehmet et al (2007) where they suggested that future work should include the monitoring of other vital signs like blood pressure, oxygen saturation level and ECG. Mehmet et al (2007) also suggested that future work should be to develop a software which will be able to evaluate the physiological parameters before being transmitted to the medical centre. The queuing model adapted in this thesis was able to evaluate physiological parameters before being transmitted to the medical center. Blood pressure was also one of the physiological parameters simulated in this thesis.

7.6 Recommendations

Remote medical monitoring is a vital component of safe health care delivery across the globe and so its importance cannot be over emphasized. Therefore it is recommended that schools, homes, government offices etc should be involved in this clarion call to save lives. This can be achieved by:

i. Government getting involved in this call to save lives by distributing the vital signs monitoring instruments to homes, schools, offices etc to enable them use it when the need arises.
ii. Educating the masses on the importance of the monitoring devices and how to use them by organizing seminars, lectures, and sensitizing the market women, the labourers and all skilled and unskilled labour on the need for remote monitoring.
iii. Getting the GSM operators to put their networks in the most interior villages so that everybody can have access to this service and enjoy the benefits.
iv. Making the monitoring devices available in all hospitals and health institutions in large quantities, sending the nurses and caregivers on courses on telemedicine for them to know the importance and how to apply it.
v. Having internet connectivity in all the government hospitals and health centres and encouraging the private clinics to follow suit.
vi. These software (**Mobile Health Information Management System** and **Intelligent Hospital Information Management System)** designed and implemented should be adopted by all the Nigerian hospitals, clinics and all health institutions in the country as it will enhance the quality of life of the Nigerian citizenry and save many lives that could have been lost.

7.7 Conclusion

In conclusion, a remote medical monitoring system for monitoring physiological parameters from multiple patient bodies has been developed. A prototype system that is able to acquire readings from multiple patients has been presented. It has been shown that after simulating the data the IPDA which can be the the laptop, the mobile phone or the Personal Digital Assistant encrypts the packets of data and sends the simulated data automatically to the queuing system in the server which queues the data according to how they arrive and is able to send the data to the medical doctor wherever he is when there is an abnormal situation. The doctor accesses the server from anywhere and also an alert can be sent on the doctor's phone when there is an emergency and the doctor can also send a feedback instructing the caregiver on what to do.

Such a wireless body sensor network system is very suitable to be used in hospital environments. Human errors will be reduced and health professionals will spend their time more on other important issues. In addition, such systems result in an increase in patients' comfort level as they no longer need to be woken up for periodic checks in the hospital environments. Another advantage is that patients can stay in their home while their conditions can still be monitored by medical staff. More wards could thus be available to patients in medical centres.
This will enhance the quality of lives of the patients and reduce the number of avoidable deaths in Nigerian hospitals.

REFERENCES

Abascal, J., Civit, A. (2001). Universal Access to Mobile Telephony as a Way to Enhance the Autonomy of Elderly People: Proceedings of the Workshop on Universal Usability of Ubiquitious Computing. New York ACM Press, 66-72.

Agarwal, R., Bunaye, Z., Bekele, D. (2008). Prognostic significance of between-arm blood pressure differences: Hypertension 51 (3): 657–62.

Ahmed, A., Ali, J., Raza, A., Abbas, G. (2006). Wired Vs Wireless Deployment Support for Wireless Sensor Networks: TENCON 2006: 2006 IEEE Region 10 Conference, 1-3.

Aminian, K., Robert, P., Buchser, E., Rutschmann, D., Hayoz, M. (1999). Physical activity monitoring based on accelerometry: validation and comparison with video observation. Medical & Biological Engineering & Computing 37(3): 304-308.

Aminian, K., Najafi, B., Büla, C., Leyvraz, P. (2001). Ambulatory Gait Analysis Using Gyroscopes: In 25th Annual Meeting of the American Society of Biomechanics. San Diego.

Anderson, R., N. (2000). A method for constructing complete annual U.S. life tables: Vital and Health Statistics. Series 2, Data Evaluation and Methods Research,. 1-28.

Andrew, S. (2002). Computer Networks: Prentice Hall

Andrew, J., Alfred, W. (2008). Remote health care monitoring devices: Computer 41(4): 96 – 99.

Angaran, D., M. (1999). Telemedicine and Telepharmacy: Current Status and Future Implications", American Journal of Health-System Pharmacy, 56: 1405-26

Anliker, U. (2004). AMON: a werable multiparameter medical monitoring and alert system, in IEEE Trans. Information Tech. In Biomedicine, 8.

Appel, L., Brands, M., Daniels, S., Karanja, N., Elmer, P., Sacks, F. (2006). Dietary approaches to prevent and treat hypertension: a scientific statement from the American Heart Association. Hypertension 47 (2): 296–308.

Ardissono, L., Giovanna, P., Marino, S. (2004). A Conversational Approach to the Interaction with Web Services: Computational Intelligence, 20,   693-709.

Arora, S., Karla, T., Glen, M., Paulina, D., Summers, K., Denise, D., ... Clifford, Q. (2011). Outcomes of Treatment for Hepatitis C Virus Infection by Primary Care Providers: New England Journal of Medicine:

Baronti, P., Pillai, P., Chook, V., Chessa, S., Gotta, A., Hu, Y. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards, Comput. Commun., 30, 1655-1695.

Barab´asi, A., Albert, R. (2002). Statistical mechanics of complex networks: Rev. Mod. Phys. 74: 47–94.

Benslimane, D., Schahram, D., Amit, S. (2008). Services Mashups: The New Generation of Web Applications: IEEE Internet Computing, Institute of Electrical and Electronics Engineers. 12(5), 13–15.

Booth, J. (1977). A short history of blood pressure measurement: Proceedings of the Royal Society of Medicine 70 (11):793–9.

Brown, M., Marmor, M., Vaegan (2006). ISCEV Standard for Clinical Electro-oculography (EOG) in: Documenta Ophthalmologica, 113(3): 205—212.

Camarinha-Matos, L., Afsarmanesh, H. (2007). A comprehensive modelling framework for collaborative networked organizations: Journal of Intelligent Manufacturing, 18: 529-542.

Cerami, E. (2002). Web Services Essentials: Distributed Applications with XML-RPC, Soap, UDDI and Wsdl, O'Reilly Media.

Chang, H. (2004). Inventing Temperature: Measurement and Scientific Progress. Oxford: Oxford University Press.

Charles, D., Jeffrey, M. (2004): IrDA Principles and Protocols.

Chet, H., Carlton, J., Matthew, D., Thomas, M. (2011). Use of Mobile Technology for Information Collection and Dissemination: Data & Analysis Center for Software.

Clint, S., Curt, G. (2003). Wireless Network Performance Handbook: McGraw-Hill Professional.

Cohen, R., Erez, K., Ben-Avraham, D., Havlin, S. (2001). Breakdown of the Internet under intentional attack: Phys. Rev. Lett 86 (16): 3682–5.

Comer, D. E. (2006). Internetworking with TCP/IP:Principles, Protocols, and Architecture: Prentice Hall. 34 -40

Corchado, J., Bajo, J., Paz, Y., Tapia, D. (2009). Intelligent environment for monitoring Alzheimer patients, agent technology for health care: Decis. Support Syst., 44: 382-396.

Deakin, C., D., Low, J., L. (2000). Accuracy of the advanced trauma life support guidelines for predicting systolic blood pressure using carotid, femoral, and radial pulses: observational study. BMJ. 321 (7262): 673–4.

Dennis, C., Heather, B., Heaton, k., Haberman, D., Gill, H. (2003). Provision of Pharmacy Services to Underserved Populations Via Remote Dispensing and Two-Way Videoconferencing: American Journal of Health-System Pharmacy 60: 2577-2582

Eguchi, K., Yacoub, M., Jhalani, J., Gerin, W., Schwartz, J., Pickering, T. (2007). Consistency of blood pressure differences between the left and right arms: Arch Intern Med 167 (4): 388–93.

Eliasz, K., Jaworek, J., Augustyniak, P. (2011). Design of a wearable Sensor Network for Home Monitoring System: Computer Science and Information Systems: 401-403.

Elliot, V., S. (2011). Blood pressure readings often unreliable: American Medical News.

Farazdaghi, G., Wohlfart, B. (2001). Reference values for the physical work capacity on a bicycle ergometer for women between 20 and 80 years of age: Clin Physiol 21 (6): 682–7

Fass, L. (2007). Patient-Centric Healthcare: Medical Electrical Devices and Technology, 3rd Institution of Engineering and technology International Conference 77-109.

Gao, T. (2005). Vital Signs Monitoring and Patient Tracking Over a Wireless Network: IEEE-EMBS 27th Annual Int. Conference of the Eng. in Medicine and Biology 102 – 105.

Gast, M. (2002). 802.11 Wireless Networks: The Definitive Guide. O'Reilly.

Gellish, R., Goslin B., Oslon, R., McDonald, A., Russi, G., Moudgil, V. (2007). Longitudinal Modeling of the Relationship between Age and Maximal Heart Rate: Medicine & Science in Sports & Exercise (American College of Sports Medicine) 39 (5): 822–828.

George, R., Schwartz, G., Gene C., Febiger, L.(1992). Principles and Practice of Emergency Medicine 2:.3202

Ghovanloo, M., Najafi, K. (2002). A BiCMOS Wireless Stimulator Chip for Micromachined Stimulating Microprobes: Proceedings of the Second Joint EMBS/BMES Conference 2113-2114.

Hafner, K. (1998). Where Wizards Stay Up Late: The Origins Of The Internet. Simon & Schuster.

Hoffmann-Wellenhof, B., Lichtenegger, H., Collins, J. (1994). GPS: Theory and Practice. 3rd ed.New York: Springer-Verlag.

Huang, Y., Pang, A. (2007). A comprehensive study of low-power operation in IEEE 802.15.4: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems: Chania, Crete Island, Greece: ACM, 405-408.

Hui, T., Lim, G., David, S., Tan, K., Gopalakrishnan, P., Rajinder, S. (2007). Ultra low power sensor node for wireless health monitoring system: IEEE International Symposium on Circuits and Systems (ISCAS), 229–232, New Orleans, USA.

Istepanian, R., Jovanov, E., Zhang, Y. (2004). Guest Editorial Introduction to the Special Section on M-Health: Beyond Seamless Mobility and Global Wireless Health-Care Connectivity. IEEE Transactions on Information Technology in Biomedicine 8:405–414.

Ilyas, M. (2002). The handbook of ad hoc wireless networks: CRC Press.

Jackson, P. (1998). Introduction to Expert Systems: (3 ed.). Addison Wesley, 2.

James, F. (2006). Statistical Methods in Experimental Physics (2^{nd} ed). Singapore: World Scientific 324

Jayaputera, G., Zaslavsky, A., and Loke, S. (2007). Enabling run-time composition and support for heterogeneous pervasive multi-agent systems: J. Syst. Softw. 80 2039-2062.

Jhalani J., Goyal, T., Clemow, L., Schwartz, J., Pickering, T., Gerin, W. (2005). Anxiety and outcome expectations predict the white-coat effect: Blood Pressure Monitoring: 10 (6): 317–9.

Jovanov, E., Price, D., Raskovic, K., Kavi, T., Martin, R. (2000). Wireless Personal Area Networks in Telemedical Environment: Proc 3rd International Conference on Information technology in Biomedicine ITAB-ITIS 22–27.

Juan, M., Javier, B., Tapia, D., Ajith, A.(2009). Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare. IEEE Transactions on Information Technology in Biomedicine: 14(2): 234 – 240

Kamen, G. (2004). Electromyographic Kinesiolog: In Robertson, Research Methods in Biomechanics. Champaign, IL: Human Kinetics Publ.

Kaplan, E. (1996). Understanding GPS: Principles and Applications. Boston: Artech House Publishers.

Karvonen, M., Kentala, E., Mustala, O. (1998). The effects of on heart rate a longitudinal study: Ann Med Exp Fenn 1998; 35(3): 307-15.

Kate, L., Sue, J., Pelham, B., Mark, C. (2011). Cost-effectiveness of options for the diagnosis of high blood pressure in primary care: a modelling study, The Lancet.

Kern, N., Schiele, B., Schmidt, A. (2003). Multi-Sensor Activity Context Detection for Wearable Computing: European Symposium on Ambient Intelligence (EUSAI): Eindhoven, The Netherlands; 34 – 38.

Klabunde, R. E. (2005). Cardiovascular Physiology Concepts: Lippincott Williams & Wilkins 93–4.

Klabunde, R. E. (2007). Cardiovascular Physiology Concepts: Mean Arterial Pressure. http://www.cvphysiology.com/Blood%20Pressure/BP006.htm

Kontaxakis, G., Visvikis, D., Ohl, R., Sachpazidis, I., Suarez, J., Selby, B. (2006). Integrated Telemedicine Applications and Services for Oncological Positron Emission Tomography: Oncology Reports, 15.

Krause, A., Siewiorek, D., Smailagic, A., Farringdon, J. (2003). Farringdon J: Unsupervised, Dynamic Identification of Physiological and Activity Context in Wearable Computing: In Proc 7th International Symposium on Wearable Computers. White Plains, NY; 88-97.

Kumar, S. (2010). Telepathology: An Audit, In: Telepathology (Springer-Verlag Berlin Heidelberg): 225–229.

Lauralee, S. (2006). Fundamentals of Physiology: A Human Perspective. Thomson Brooks/Cole, 380.

Lee, S., Mase, K. (2002). Activity and Location Recognition Using Wearable Sensors: Pervasive Computing, 1(3): 24-32.

Lisa,K., Linda, L., Victoria, S., Karen, L. (2011). An E-rehabilitation Team Helps Caregivers Deal with Stroke: The Internet Journal of Allied Health Sciences and Practice.

Lotz, A. (2007). The Television Will Be Revolutionized. New York, NY: New York University Press 65-66.

Mancia, G., Backer, G., Dominiczak, A. (2007). Guidelines for the management of arterial hypertension: The Task Force for the Management of Arterial Hypertension of the European Society of Hypertension (ESH) and of the European Society of Cardiology (ESC). Eur Heart J 28 (12): 1462–536.

Martin, T., Jovanov, E., Raskovic, D. (2000). Issues in Wearable Computing for Medical Monitoring Applications: A Case Study of a Wearable ECG Monitoring Device. Proc of the International Symposium on Wearable Computers ISWC Atlanta, Georgia. 43–50.

Mehmet, R., Peng, C., Jamil, Y., Wentai, L., Chin, L. (2007). A Wireless Medical Monitoring Over a Heterogeneous Sensor Network: Conference Proceedings of IEEE Engineering Medical Biology Society: 5895 – 9.

Melnick, M., Radtka, S., Piper, M. (2002). Gait Analysis and Parkinson's Disease: Rehab Management, the Interdisciplinary Journal of Rehabilitation.

Meshkova, E., Riihijärvi, J., Oldewurtel, F., Jardak, C., Mähönen, P. (2008). Service-Oriented Design Methodology for Wireless Sensor Networks: A View through Case Studies, Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing. 01, IEEE Computer Society, 146-153.

Meong-hun, L., Hyun, Y. (2007). Comparative Analysis and Design of Wired and Wireless Integrated Networks for Wireless Sensor Networks: Software Engineering Research, Management & Applications, SERA. 5th ACIS International Conference on. 518-522.

Mitchell, G. (2006). Triangulating the peaks of arterial pressure: Hypertension 48 (4): 543–5. http://www.cvphysiology.com/Blood%20Pressure/BP012.htm.

Milenkovic, E., Jovanov, E., Chapman, J. Raskovic, D., Price, J. (2002). An Accelerometer-Based Physical Rehabilitation System: The 34th Southeastern Symposium on System Theory (SSST) 57-60.

Moeller, R., Sleman, A. (2008). Wireless networking services for implementation of ambient intelligence at home: Devices, Circuits and Systems, ICCDCS. 7th International Caribbean Conference 1-5.

Mower, Sachs, Nicklin, Baraff (1997). Pulse oximetry as a fifth pediatric vital sign: Pediatrics 99 (5): 681–6.

Mower, W., Nicklin, E., Baraff, L., Sachs, C. (1998). Pulse oximetry as a fifth vital sign in emergency geriatric assessment: Acad Emerg Med **5** (9): 858–65.

Nakajima, I., Sastrokusumo, U., Mishra, S., Komiya, R., Malik, A., Tanuma, T. (2006). The Asia Pacific Telecommunity's Telemedicine Activities: IEEE Xplore.com website 280 – 282.

Niedermeyer E., Da Silva F.L. (2004). Electroencephalography: Basic Principles, Clinical Applications, and Related Fields. Lippincot Williams & Wilkins.

Niiranen, T., J., Kantola, I., M. (2006). A comparison of home measurement and ambulatory monitoring of blood pressure in the adjustment of antihypertensive treatment: Am J Hypertens. 19 (5): 468–74.

Obrenovic, Z., Starcevic, D., Jovanov, E., Radivojevic, V. (2002). An Agent Based Framework for Virtual Medical Devices: Autonomous Agents & Multi-Agent Systems, Bologna, Italy. McGraw Hill Company. 89 - 95

O'Rourke, M. (1995). Mechanical principles in arterial disease: Hypertension 26 (1): 2–9.

Otis, B., Rabaey, J. (2003). A 300-μW 1.9-GHz CMOS Oscillator Utilizing Micromachined Resonators: IEEE Journal of Solid-State Circuits. 38(7): 1271-1274.

Otto, C., Gober, J., McMurtrey, R., Milenkoviæ, A., Jovanov, E. (2005). An Implementation of Hierarchical Signal Processing on Wireless Sensor in TinyOS Environment: 43rd Annual ACM Southeast Conference ACMSE.

Pappas, I., Keller, T., Mangold, S., Popovic, M., Dietz, V., Morari, M. (2004). A Reliable Gyroscope-Based Gait-Phase Detection Sensor Embedded in a Shoe Insole: IEEE Sensors Journal, 4(2): 268-274.

Park, S., Jayaraman, S. (2003). Enhancing the Quality of Life Through Wearable Technology: IEEE Engineering in Medicine and Biology Magazine. 22: pp 41–48.

Pennifer, E. (1995). Years of healthy life. ([Hyattsville, Md.] (6525 Belcrest Rd., Hyattsville 20782)): U.S. Dept. of Health and Human Services, Public Health Service, Centers for Disease Control and Prevention, National Center for Health Statistics.

Pickering T. G., Hall J. E., Appel L.J., Falkner B. E., Graves J., Hill M. N., ... Roccella E. J. (2005). Blood pressure measurement in humans: American Heart Association ouncil on High Blood Pressure Research. 45 (5): 142 – 61.

Poon, C., Zhang, Y. (2006). A Novel Biometrics to Secure Wireless Body Area Sensor Networks for Telemedicine and MHealth: IEEE Communications Magazine, 44 (4) 73-81.

Prinsloo, J., Schulz, C., Kourie, D., Theunissen, W., Strauss, T., Heever, R., Grobbelaar, S. (2006). A service oriented architecture for wireless sensor and actor network applications: Proceedings of the annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries. Somerset West, South Africa: South African Institute for Computer Scientists and Information Technologists, 145-154.

Rngayyan R. M. (2010). Pan-Tompkins algorithm to detect QRS complex in ECG signal, Biomedical Signal Analysis, IEEE press at http://www.docstoc.com/docs/22491202/Pan-Tompkins-algorithm-algorithm-to-detect-QRS-complex-in-ECG

Raskovic, D., Martin, T., Jovanov, E. (2004). Medical Monitoring Applications for Wearable Computing: *The Computer Journal 47: 495–504.*

Riedel, T. (2003). Health wiring goes remote using low-power RF, Wireless: Proceedings of ACM SIG Mobile, ACM Press, New York. 180-190.

Rigole, P., Holvoet, T., Berbers, Y. (2002). Using Jini to Integrate Home Automation in a Distributed Software-System: Distributed Communities on the Web, 185-232.

Robergs, R., Landwehr, R. (2002). The Surprising History of the HRmax=220-age Equation: Journal of Exercise Physiology 5 (2): 1–10.

Robert, H. (1995). Teletransmission of ECG Waveform: An indegenous low priced technique - Indian heart journal 34(6): 45 -50.

Rosenson, R., Wolff, D., Green, D., Boss, A., Kensey, K. (2004). Aspirin: Aspirin does not alter native blood viscosity. J. Thromb. Haemost. 2 (2): 340–1.

Sarangapani, J. (2007). Wireless Ad hoc and Sensor Networks: Protocols, Performance, and Control, CRC.

Sachpazidis, I. (2008). Image and Medical Data Communication Protocols for Telemedicine and Teleradiology (dissertation): Department of Computer Science, Technical University of Darmstadt, Germany.

Scanlon, W. G., Burns, J. B., Evans, N., E. (2000). Radiowave propagation from a tissue implanted source at 418 MHz and 916. 5 MHz: IEEE Trans. Biomedical Engineering, 47:527-534.

Schwaibold, M., Gmelin, M., Wagner, G., Sch¨ochlin, J., Bolz, A. (2004). Key factors for personal health monitoring and diagnosis devices: Workshop of Mobile Computing in Medicine, New York. ACM Press 20 -24

.

Schwiebert, L., Gupta, S., Weinmann, J., (2001). Research Challenges in Wireless Networks of Biomedical Sensors: Proceedings of ACM SIG Mobile: New York. ACM Press, 151-165.

Sharon, S., M., Emily, S., M. (2006). Foundations of Maternal-Newborn Nursing: 476

Shimbo, D., Abraham, D., Spuruill, T., Thomas, Pickering, T. (2007). The Relative Utility of Home, Ambulatory, and Office Blood Pressures in the Prediction of End-Organ Damage: Am J Hypertens 20 (5): 476–82.

Shu, L., Hui, X., Wu, X., Zhang, L., Jinsung, C., Sungyoung, L. (2006). VIP Bridge: Integrating Several Sensor Networks into One Virtual Sensor Network. Internet Surveillance and Protection,. ICISP. International Conference on, 2.

Simon, D., Cifuentes, C. (2005). The squawk virtual machine: Java™ on the bare metal, Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, San Diego, CA, USA: ACM, 150-151.

Singh, C., Kumar, A., Ameer, P. (2008). Performance evaluation of an IEEE 802.15.4 sensor network with a star topology: Wireless Networks, 14: 543-568.

Sleman, A., Moeller, R. (2008). Integration of Wireless Sensor Network Services into other Home and Industrial networks: Using Device Profile for Web Services (DPWS), Information and Communication Technologies: From Theory to Applications, ICTTA. 3rd International Conference 1-5.

Song, E., Kang, L. (2007). Smart Transducer Web Services Based on the IEEE 1451.0 Standard: Instrumentation and Measurement Technology Conference Proceedings, IMTC. IEEE, 1-6.

Stallings, W. (2005). Wireless communications & networks: Pearson Prentice Hall 10 -15

Steele, B., Belza, B., Cain, K., Warms, C., Coppersmith, J., Howard, J. (2003). Bodies in motion: Monitoring daily activity and exercise with motion sensors in people with chronic pulmonary disease. Journal of Rehabilitation Research & Development. ***40**: 45–58.*

Struijk, P., C., Mathews, V., J., Loupas, T., Stewart P., A. (2008). Blood pressure estimation in the human fetal descending aorta: Ultrasound Obstet Gynecol 32 (5): 673–81

Tekin, A., Yuce, M., Liu, W. (2005). A Low power MICS band transceiver architecture for implantable devices: In Proc. IEEE Wireless and Microwave Technology Conference, 55-58.

Tekin, A., Yuce, M., Liu, W. (2006). Integrated VCO Design for MICS Transceivers: In Proc. *IEEE* Custom Integrated Circuits Conference, 765 – 768.

Tharion, W., Yokota, M., Buller, M., DeLany, P., Hoyt, R. (2004). Total Energy Expenditure Estimated Using a Foot-Contact Pedometer: Medical Science Monitor 10 (9): 504-509.

Tompkins, W., Pan, J. (1985). A real-time QRS detection algorithm, Biomedical Signal Analysis: IEEE Press, BME- 32(3).

Tortora, G., Anagnostakos, N. (1990). Principles of Anatomy and Physiology: 6th edition, New York: Harper-Collins, 707.

Traynor, K. (2010). Navy takes Telepharmacy Worldwide: American Journal of Health-System Pharmacy 67:1134-1136.

Van, L., Kern, N., Gellersen, H., Schiele, B. (2003). Towards A Wearable Inertial Sensor Network: IEE EuroWearable (EuroWearable) Birmingham, UK. 12 – 20.

Van Mieghem, C; Sabbe, M; Knockaert, D (2004). The clinical value of the ECG in noncardiac conditions: Chest 125 (4): 1561–76.

Varshney, U. (2008). Improving Wireless Health Monitoring Using Incentive-Based Router Cooperation: Computer, 41:56-62.

Viken, A. (2009). The History of Personal Digital Assistants: 1980 – 2000. Agile Mobility.

Vinton, G., Robert, E. (1974). A Protocol for Packet Network Intercommunication: IEEE Transactions on Communications. 22(5): 637-648.

Walter, W., Ramesh, G., Sugih, J., Vern, P., Scott, S. (2002). Scaling phenomena in the Internet: In Proceedings of the National Academy of Sciences, 99(1): 2573–2580.

Warren, S. (2003). Beyond Telemedicine: Infrastructures for Intelligent Home Care Technology: In Pre-ICADI Workshop on Technology for Aging, Disability, and Independence. The Royal Academy of Engineering, Westminster, London.

Weinstein, R. S., Graham, Richter, Barker, Krupinski, Lopez, Yagi, Gilbertson (2009). Overview of telepathology, virtual microscopy and whole slide imagining: Prospects for the future, *Hum Pathol* 40: 1057–1069.

Welch, J., Guilak, F., Baker, S. (2003). A Wireless ECG Smart Sensor for Broad Application in Life Threatening Event Detection: Proc of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 3447-3449.

William, S. (1984). Digital signaling techniques: a comparison of various encoding techniques with an eye to superior performance, IEEE Communications Magazine, 22 (12).

Winters, J., Wang, Y., Winters, J. (2003). Wearable Sensors and Telerehabilitation: Integrating Intelligent Telerehabilitation Assistants with a Model for Optimizing Home Therapy. IEEE Engineering in Medicine and Biology Magazine 22(3):56-65.

Wohlfart, B., Farazdaghi, G. (2003). Reference values for the physical work capacity on a bicycle ergometer for men -- a comparison with a previous study on women: Clin Physiol Funct Imaging 23 (3): 166–70.

Yuce, M., R., Ng, P., C., Lee, C., K., Khan, J., Y., Liu, W. (2007). A wireless medical monitoring over a heterogeneous sensor network: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Conference Proceedings, Lyon, France 120-132.

APPENDIX I

Sample Remote Monitoring System Interview Guide

RESEARCH TOPIC: **MODELLING REMOTE PATIENT MEDICAL MONITORING SYSTEM IN NIGERIAN HOSPITALS**

RESEARCHER: **OJI IFEOMA VIVIAN**

SCHOOL: **NNAMDI AZIKIWE UNIVERSITY AWKA**

TO WHOM IT MAY CONCERN

I wish to thank you in a special way for sparing your precious time to grant me this interview today. My name is Oji Ifeoma Vivian from Nnamdi Azikiwe University, Awka Anambra State. I will like to talk to you about the manual Medical Monitoring System in your hospital with a view to ascertaining the merits and the demerits of the system and determining how to improve on it.

The interview should take less than one hour. I will be taking some notes during the session so please try to speak up so that I don't miss your comments.

All responses will be kept confidential. Remember you don't have to talk about anything you don't want to and you may end the interview at any time.

Thank you.

____________________	____________________	____________________
Interviewee	Witness	Date

Department of Computer Science,
Nnamdi Azikiwe University,
Awka.
2nd November, 2012.

The Medical Director,
Central Hospital,
Warri.

Dear Sir,

REQUEST FOR PERMISSION TO CARRY OUT A SURVEY ON MANUAL MEDICAL MONITORING IN CENTRAL HOSPITAL, WARRI

I wish to request for your permission to carry out a survey on Manual Medical Monitoring in your hospital.

I am Oji Ifeoma Vivian, a Ph.d student of the Nnamdi Azikiwe University, Awka. I am working on Remote Medical Monitoring where the vital signs of patients can be monitored remotely, and so I need to understand how these vital signs are monitored manually in your hospital with a view to improving on it.

I will be very grateful if my request is granted.

Thank you.

Yours Faithfully,

Oji Ifeoma Vivian

THE QUESTIONNAIRE

This questionnaire is aimed at studying and evaluating the manual system of vital signs monitoring in the Central Hospital Warri. The target are the patients who are old (i.e. patients from 70 years and above) and other patients suffering from any of the chronic diseases like diabetes, hypertension, stroke, cancer etc with a view to identifying the problems of this system and determining better ways to solve the problem.

The questionnaire is divided into two parts, part 1 for the nurses and part 2 for the patients.

Instructions

Please tick ✓ in the appropriate box.

Part 1

For the Nurses

1. How many times is the vital signs of your patients checked in a day?
   a. Two times in a day ☐ b. Three times a day ☐ c. More than three times a day ☐
2. How quickly does the doctor respond to emergency calls?
   a. very quickly ☐ b. Not so quickly ☐ c. Not at all ☐.
3. Does the instrument for monitoring the vital sign depend on power always?
   a. All the time ☐ b. Sometimes ☐ c. Not at all ☐
4. How often do the doctors on call stay in their offices?
   a. Very often ☐ b. Not so often ☐ c. Not at all ☐
5. How efficient is the manual monitoring of patient blood pressure and pulse rate?
   a. Very efficient ☐ b. Not so efficient ☐ c. Not efficient at all ☐.
6. How reliable are the instruments for measuring blood pressure and pulse rate?
   a. Very reliable ☐ b. Not so reliable ☐ c. Not reliable at all ☐

7. Do you think monitoring the vital signs of patients continuously even when they are in their homes is better than having all the patients in the hospital and monitoring their vital signs?
   a. Yes ☐ b. No ☐ .

8. Do you think the nurses will be redundant if the job of monitoring the vital signs of patients is taken away from them?
   a. Very likely ☐ b. Not so likely ☐ c. Not at all ☐
9. Do you think the patients will prefer this automatic continuous monitoring to the time to time manual monitoring?
   a. Very likely ☐ b. Not so likely ☐ c. Not at all ☐
10. How efficient do you think this remote monitoring will be?
    a. Very efficient ☐ b. Not so efficient ☐ c. Not efficient at all ☐
11. How cost effective do you think the remote medical monitoring will be?
    a. Very cost effective ☐ b. Not so cost effective ☐ c. Not at all ☐
12. Will this remote monitoring help in saving the lives of the patients.
    a. Very likely ☐ b. Not so likely ☐ c. Not at all ☐

Part 2

For the Patients

Please tick ✓ on the chronic condition(s) you have:

Diabetes ☐ High blood pressure ☐ Stroke ☐ Cancer ☐ Other chronic condition ☐ Specify --------------------- Age --------- Sex -----------

1. How often do the nurses check your blood pressure and pulse rate?
   a. Very often ☐ b. Not so often ☐ c. Not at all ☐
2. Has there been any time the nurses failed in their duty to check your blood pressure and pulse rate?
   a. Most of the time ☐ b. Sometimes ☐ c. Not at all ☐
3. How often do you feel disturbed when your blood pressure and pulse rate are being checked?
   a. Very often ☐ b. Not so often ☐ c. Not at all ☐
4. Do you think the instruments for checking the blood pressure and pulse rate can be faulty?
   a. Most likely ☐ b. Not likely ☐ c. Not at all ☐
5. Do you think it will be good to use a remote monitoring system where you can be in your home and your blood pressure and pulse rate can be monitored automatically?
   a. Very good ☐ b. Not so good ☐ c. Not good at all ☐

6. How often does the doctor go through your vital signs readings?
   a. Very often ☐ b. Not so often ☐. Not at all ☐
7. Do you think the nurses show enough commitment to their job in monitoring your blood pressure and pulse rate?
   a. Very committed ☐b. Not so committed ☐ Not at all ☐
8. How will you rate the vital sign monitoring in this hospital?
   a. Very good ☐b. Not so good ☐c. Not good at all ☐ .

APPENDIX II

PROGRAM LISTING

Mobile Hospital Application source code

```
package loginscreenexample;
import javax.microedition.io.*;         // package required for J2ME
import java.io.*;
import java.lang.*;
import javax.microedition.lcdui.*;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Ticker;
import javax.microedition.midlet.*;
import org.netbeans.microedition.lcdui.LoginScreen;
import org.netbeans.microedition.lcdui.SplashScreen;
import org.netbeans.microedition.lcdui.WaitScreen;
import org.netbeans.microedition.util.SimpleCancellableTask;
public class LoginScreenExample extends MIDlet implements CommandListener {
    private boolean midletPaused = false;
    private String url = "";//http://localhost:8084/inthospital/mobile/getConnection";
    private TextField tb;
    private String Age;
    private String BloodPressure;
private Form menu;
private TextField tb1;
private TextField tb2;
private Command exit = new Command("EXIT", Command.EXIT, 1);;
private Command connect = new Command("Connect", Command.SCREEN, 1);
private Display display;
public String pa;
//<editor-fold defaultstate="collapsed" desc=" Generated Fields ">
private WaitScreen waitScreen;
private SplashScreen splashScreen;
private LoginScreen loginScreen;
private Alert alertSuccess;
private Alert alertFailure;
private TextBox textBox;
private TextBox textBox1;
private Form form;
private DateField dateField;
private Alert alert;
private Command exitCommand;
private Command itemCommand;
private Command okCommand;
private Command okCommand1;
private Command okCommand2;
private Command okCommand3;
private Command okCommand4;
private Command okCommand5;
private Command backCommand;
private SimpleCancellableTask task;
private Ticker ticker;
```

```
private Ticker ticker1;
private Image image1;
private Image image;
//</editor-fold>
    /**
     * The LoginScreenExample constructor.
     */
    public LoginScreenExample() {
    }
    //<editor-fold defaultstate="collapsed" desc=" Generated Methods ">
    //</editor-fold>
    //<editor-fold defaultstate="collapsed" desc=" Generated Method: initialize ">
    /**
     * Initializes the application.
     * It is called only once when the MIDlet is started. The method is called before the <code>startMIDlet</code> method.
     */
    private void initialize() {
        // write pre-initialize user code here
        textBox = new TextBox("Instruction for Users", "Name;\nGSM/PHONE NO;\nHospital ID;\n [1-H, 2-Normal, 3-
Low];\nAge;\nDrink?Y/N.\nSmoke?Y/N.", 155, TextField.ANY);
        textBox.setTicker(getTicker1());
        textBox.addCommand(getOkCommand());
        textBox.setCommandListener(this);
        ticker = new Ticker("");
        // write post-initialize user code here
    }
    //</editor-fold>

    //<editor-fold defaultstate="collapsed" desc=" Generated Method: startMIDlet ">
    /**
     * Performs an action assigned to the Mobile Device - MIDlet Started point.
     */
    public void startMIDlet() {
        // write pre-action user code here
        switchDisplayable(null, getSplashScreen());
        // write post-action user code here
    }
    //</editor-fold>
    //<editor-fold defaultstate="collapsed" desc=" Generated Method: resumeMIDlet ">
    /**
     * Performs an action assigned to the Mobile Device - MIDlet Resumed point.
     */
    public void resumeMIDlet() {
        // write pre-action user code here
        // write post-action user code here
    }
    //</editor-fold>
    //<editor-fold defaultstate="collapsed" desc=" Generated Method: switchDisplayable ">
    /**
     * Switches a current displayable in a display. The <code>display</code> instance is taken from
<code>getDisplay</code> method. This method is used by all actions in the design for switching displayable.
     * @param alert the Alert which is temporarily set to the display; if <code>null</code>, then
<code>nextDisplayable</code> is set immediately
     * @param nextDisplayable the Displayable to be set
     */
    public void switchDisplayable(Alert alert, Displayable nextDisplayable) {
        // write pre-switch user code here
        Display display = getDisplay();
```

```
    if (alert == null) {
        display.setCurrent(nextDisplayable);
    } else {
        display.setCurrent(alert, nextDisplayable);
    }
    // write post-switch user code here
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: task ">
/**
 * Returns an initiliazed instance of task component.
 * @return the initialized component instance
 */
public SimpleCancellableTask getTask() {
    if (task == null) {
        // write pre-init user code here
        task = new SimpleCancellableTask();
        task.setExecutable(new org.netbeans.microedition.util.Executable() {
            public void execute() throws Exception {
                login();
            }
        });
        // write post-init user code here
    }
    return task;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Method: commandAction for Displayables ">
/**
 * Called by a system to indicated that a command has been invoked on a particular displayable.
 * @param command the Command that was invoked
 * @param displayable the Displayable where the command was invoked
 */
public void commandAction(Command command, Displayable displayable) {
    // write pre-action user code here
    if (displayable == form) {
        if (command == backCommand) {
            // write pre-action user code here
            switchDisplayable(null, getLoginScreen());
            // write post-action user code here
        }
    } else if (displayable == loginScreen) {
        if (command == LoginScreen.LOGIN_COMMAND) {
            // write pre-action user code here
            switchDisplayable(null, getWaitScreen());
            // write post-action user code here
        } else if (command == exitCommand) {
            // write pre-action user code here
            exitMIDlet();
            // write post-action user code here
        } else if (command == okCommand1) {
            // write pre-action user code here
            switchDisplayable(null, getTextBox1());
            // write post-action user code here
        } else if (command == okCommand4) {
            // write pre-action user code here
            switchDisplayable(null, getForm());
```

```
                // write post-action user code here
            }
        } else if (displayable == splashScreen) {
            if (command == SplashScreen.DISMISS_COMMAND) {
                // write pre-action user code here
                switchDisplayable(null, textBox);
                // write post-action user code here
            }
        } else if (displayable == textBox) {
            if (command == okCommand) {
                // write pre-action user code here
                switchDisplayable(null, getLoginScreen());
                // write post-action user code here
            }
        } else if (displayable == textBox1) {
            if (command == okCommand2) {
                // write pre-action user code here
                switchDisplayable(null, getLoginScreen());
                // write post-action user code here
            }
        } else if (displayable == waitScreen) {
            if (command == WaitScreen.FAILURE_COMMAND) {
                // write pre-action user code here
                switchDisplayable(getAlertFailure(), getLoginScreen());
                // write post-action user code here
            } else if (command == WaitScreen.SUCCESS_COMMAND) {
                // write pre-action user code here
                switchDisplayable(getAlertSuccess(), getLoginScreen());
                // write post-action user code here
            }
        }
        // write post-action user code here
    }
    //</editor-fold>
    //<editor-fold defaultstate="collapsed" desc=" Generated Getter: waitScreen ">
    /**
     * Returns an initiliazed instance of waitScreen component.
     * @return the initialized component instance
     */
    public WaitScreen getWaitScreen() {
        if (waitScreen == null) {
            // write pre-init user code here
            waitScreen = new WaitScreen(getDisplay());
            waitScreen.setTitle("wait Screen");
            waitScreen.setCommandListener(this);
            waitScreen.setText("Please Wait ...");
            waitScreen.setTask(getTask());
            // write post-init user code here
        }
        return waitScreen;
    }
    //</editor-fold>
public int Systolic;
public int Diastolic;
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: exitCommand ">
/**
 * Returns an initiliazed instance of exitCommand component.
 * @return the initialized component instance
```

```
 */
public Command getExitCommand() {
   if (exitCommand == null) {
          // write pre-init user code here
       exitCommand = new Command("Exit", Command.EXIT, 0);
          // write post-init user code here
   }
   return exitCommand;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: loginScreen ">
/**
 * Returns an initiliazed instance of loginScreen component.
 * @return the initialized component instance
 */
public LoginScreen getLoginScreen() {
   if (loginScreen == null) {
          // write pre-init user code here
       loginScreen = new LoginScreen(getDisplay());
       loginScreen.setLabelTexts("Connect to", "Access key:");
       loginScreen.setTitle("Mobile Login");
       loginScreen.setTicker(ticker);
       loginScreen.addCommand(LoginScreen.LOGIN_COMMAND);
       loginScreen.addCommand(getExitCommand());
       loginScreen.addCommand(getOkCommand1());
       loginScreen.addCommand(getOkCommand4());
       loginScreen.setCommandListener(this);
       loginScreen.setBGColor(-3355444);
       loginScreen.setFGColor(0);
       loginScreen.setPassword("root");
       loginScreen.setUsername("http://localhost:8084/inthospital/mobile/getconnection.jsp");
       loginScreen.setUseLoginButton(false);
       loginScreen.setLoginButtonText("inthospitaldb");
       menu = new Form("Connect to Database");
try{
tb = new TextField("Please enter database: ","inthospitaldb",30,TextField.ANY );
tb1 = new TextField("Please enter username: ","root",30,TextField.ANY);
tb2 = new TextField("Please enter password: ","",30,TextField.PASSWORD);
menu.append(tb);
menu.append(tb1);
menu.append(tb2);
menu.addCommand(exit);
menu.addCommand(connect);
menu.setCommandListener(this);
display.setCurrent(menu);
}catch (Exception e) {
showAlert(e.getMessage());
}
//loginScreen = new LoginScreen(display);
          // write post-init user code here
   }
   return loginScreen;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: alertFailure ">
/**
 * Returns an initiliazed instance of alertFailure component.
 * @return the initialized component instance
```

```
 */
public Alert getAlertFailure() {
    if (alertFailure == null) {
            // write pre-init user code here
        alertFailure = new Alert("alert", "Wrong username or password", null, null);
        alertFailure.setTimeout(Alert.FOREVER);
            // write post-init user code here
    }
    return alertFailure;
}
//</editor-fold>
public String[] DData=new String [10];
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: alertSuccess ">
/**
 * Returns an initiliazed instance of alertSuccess component.
 * @return the initialized component instance
 */
public Alert getAlertSuccess() {
    if (alertSuccess == null) {
            // write pre-init user code here
        alertSuccess = new Alert("Logged In", "Successfuly logged in", null, null);
        alertSuccess.setTimeout(Alert.FOREVER);
            // write post-init user code here
            String pa;
            StringBuffer sb = new StringBuffer();
            StringBuffer sb1 = new StringBuffer();
       String DataSource=textBox1.getString();
       int a=0;
       int u=0;
       char av=';';
       url = loginScreen.getUsername();//"http://localhost:8084/inthospital/getConnection";
        String user=loginScreen.getPassword();//"root";
        String db="inthospitaldb";
        String pwd="";
        System.out.println(DataSource.toString());
if (DataSource.length()!=0 ){
    for (int v=0;v<DataSource.length();v++){
        if (DataSource.charAt(v)==av){
            u=v+1;
            a++;
        }else{
            if (a<8){
            DData[a]=DataSource.substring(u, v+1);
            }
        }
    }
}
    for (int m=0;m<8;m++){
        System.out.println(DData[m]);
        }

 System.out.println(url);
String opt1=DData[3];
 System.out.println("Option selected-"+opt1);
if (opt1=="1"){
     Systolic=120;//Math.round()*120;// High Systolic Pressure
     Diastolic=80;
}else if (opt1=="3"){
```

```
    Systolic=60;//Math.round()*120;// High Systolic Pressure
     Diastolic=40;
}else {

    Systolic=90;//
    Diastolic=60;
}

 //PulseRate=Systolic-Diastolic;
 //MAP=Diastolic+(0.033*(Systolic-Diastolic));
DData[6]=Systolic+";\n";
DData[7]=Diastolic+";\n";
for (int j=0;j<8;j++){
     sb.append(DData[j]+"");
 }
sb1.append(db+";\n");
 sb1.append(user+";\n");
 sb1.append(pwd+";\n");
 sb1.append("Name:"+DData[0]+";\n");
 sb1.append("Tel.No:"+DData[1]+";\n");
 sb1.append("PID:"+DData[2]+";\n");
 sb1.append("Status:"+DData[3]+";\n");
 sb1.append("Age:"+DData[4]+"\n");
  sb1.append("Smoke:"+DData[5]+"\n");
 sb1.append("Sys:"+DData[6]+"\n");
 sb1.append("Dia:"+DData[7]+"\n");
 //pa=sb.toString();
        //String pa1=sb1.toString();
            connect2DB(url);
            alertSuccess = new Alert(pb,pa1,null,null);
    }
    return alertSuccess;
}
//</editor-fold>
public String pa1;
public String pb;
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: splashScreen ">
/**
 * Returns an initiliazed instance of splashScreen component.
 * @return the initialized component instance
 */
public SplashScreen getSplashScreen() {
    if (splashScreen == null) {
            // write pre-init user code here
        splashScreen = new SplashScreen(getDisplay());
        splashScreen.setTitle("Mobile Hospital");
        splashScreen.setCommandListener(this);
        splashScreen.setFullScreenMode(true);
        splashScreen.setImage(getImage1());
        splashScreen.setText("Mobile Health MIS");
            // write post-init user code here
    }
    return splashScreen;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: itemCommand ">
/**
```

```
 * Returns an initiliazed instance of itemCommand component.
 * @return the initialized component instance
 */
public Command getItemCommand() {
    if (itemCommand == null) {
            // write pre-init user code here
        itemCommand = new Command("Item", Command.ITEM, 0);
            // write post-init user code here
    }
    return itemCommand;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: ticker1 ">
/**
 * Returns an initiliazed instance of ticker1 component.
 * @return the initialized component instance
 */
public Ticker getTicker1() {
    if (ticker1 == null) {
            // write pre-init user code here
        ticker1 = new Ticker("");
            // write post-init user code here
    }
    return ticker1;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand ">
/**
 * Returns an initiliazed instance of okCommand component.
 * @return the initialized component instance
 */
public Command getOkCommand() {
    if (okCommand == null) {
            // write pre-init user code here
        okCommand = new Command("Ok", Command.OK, 0);
            // write post-init user code here
    }
    return okCommand;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand1 ">
/**
 * Returns an initiliazed instance of okCommand1 component.
 * @return the initialized component instance
 */
public Command getOkCommand1() {
    if (okCommand1 == null) {
            // write pre-init user code here
        okCommand1 = new Command("Profile", Command.OK, 0);
            // write post-init user code here
    }
    return okCommand1;
}
//</editor-fold>
    //<editor-fold defaultstate="collapsed" desc=" Generated Getter: textBox1 ">
    /**
     * Returns an initiliazed instance of textBox1 component.
     * @return the initialized component instance
```

```
 */
public TextBox getTextBox1() {
    if (textBox1 == null) {
        // write pre-init user code here
        textBox1 = new TextBox("User Profile (Use ; to seperate item.)", "", 100, TextField.ANY);
        textBox1.addCommand(getOkCommand2());
        textBox1.setCommandListener(this);
        // write post-init user code here
    }
    return textBox1;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand2 ">
/**
 * Returns an initiliazed instance of okCommand2 component.
 * @return the initialized component instance
 */
public Command getOkCommand2() {
    if (okCommand2 == null) {
        // write pre-init user code here
        okCommand2 = new Command("Return", Command.OK, 0);
        // write post-init user code here
    }
    return okCommand2;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image1 ">
/**
 * Returns an initiliazed instance of image1 component.
 * @return the initialized component instance
 */
public Image getImage1() {
    if (image1 == null) {
        // write pre-init user code here
        try {
            image1 = Image.createImage("/FemaleDoctoatComputer.png");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image1;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand3 ">
/**
 * Returns an initiliazed instance of okCommand3 component.
 * @return the initialized component instance
 */
public Command getOkCommand3() {
    if (okCommand3 == null) {
        // write pre-init user code here
        okCommand3 = new Command("Ok", Command.OK, 0);
        // write post-init user code here
    }
    return okCommand3;
}
//</editor-fold>
```

```
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand4 ">
/**
 * Returns an initiliazed instance of okCommand4 component.
 * @return the initialized component instance
 */
public Command getOkCommand4() {
    if (okCommand4 == null) {
        // write pre-init user code here
        okCommand4 = new Command("About", Command.OK, 0);
        // write post-init user code here
    }
    return okCommand4;
}
//</editor-fold>

//<editor-fold defaultstate="collapsed" desc=" Generated Getter: image ">
/**
 * Returns an initiliazed instance of image component.
 * @return the initialized component instance
 */
public Image getImage() {
    if (image == null) {
        // write pre-init user code here
        try {
            image = Image.createImage("/StaffPic.png");
        } catch (java.io.IOException e) {
            e.printStackTrace();
        }
        // write post-init user code here
    }
    return image;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: form ">
/**
 * Returns an initiliazed instance of form component.
 * @return the initialized component instance
 */
public Form getForm() {
    if (form == null) {
        // write pre-init user code here
        form = new Form("About Author", new Item[] { getDateField() });
        form.addCommand(getBackCommand());
        form.setCommandListener(this);
        // write post-init user code here
    }
    return form;
}
//</editor-fold>
//<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand5 ">
/**
 * Returns an initiliazed instance of okCommand5 component.
 * @return the initialized component instance
 */
public Command getOkCommand5() {
    if (okCommand5 == null) {
        // write pre-init user code here
```

```
            okCommand5 = new Command("Return", Command.OK, 0);
            // write post-init user code here
        }
        return okCommand5;
    }
    //</editor-fold>
    //<editor-fold defaultstate="collapsed" desc=" Generated Getter: dateField ">
    /**
     * Returns an initiliazed instance of dateField component.
     * @return the initialized component instance
     */
    public DateField getDateField() {
        if (dateField == null) {
            // write pre-init user code here
            dateField = new DateField("dateField", DateField.DATE_TIME);
            dateField.setDate(new java.util.Date(System.currentTimeMillis()));
            // write post-init user code here
        }
        return dateField;
    }
    //</editor-fold>
    //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand ">
    /**
     * Returns an initiliazed instance of backCommand component.
     * @return the initialized component instance
     */
    public Command getBackCommand() {
        if (backCommand == null) {
            // write pre-init user code here
            backCommand = new Command("Back", Command.BACK, 0);
            // write post-init user code here
        }
        return backCommand;
    }
    //</editor-fold>
    //<editor-fold defaultstate="collapsed" desc=" Generated Getter: alert ">
    /**
     * Returns an initiliazed instance of alert component.
     * @return the initialized component instance
     */
    public Alert getAlert() {
        if (alert == null) {
            // write pre-init user code here
            alert = new Alert("alertConnect", null, null, AlertType.CONFIRMATION);
            alert.setTimeout(Alert.FOREVER);
            // write post-init user code here
        }
        return alert;
    }
    //</editor-fold>
    public Display getDisplay() {
this.display=display;
       return Display.getDisplay(this);
    }
    /**
     * Exits MIDlet.
     */
    public void exitMIDlet() {
```

```
        switchDisplayable(null, null);
        destroyApp(true);
        notifyDestroyed();
    }

    /**
     * Called when MIDlet is started.
     * Checks whether the MIDlet have been already started and initialize/starts or resumes the MIDlet.
     */
    public void startApp() {
        if (midletPaused) {
            resumeMIDlet();
        } else {
            initialize();
            startMIDlet();
        }
        midletPaused = false;
    }

    /**
     * Called when MIDlet is paused.
     */
    public void pauseApp() {
        midletPaused = true;
    }
public Alert connect2DB(String url){
  StringBuffer  sb;
   String user=loginScreen.getPassword();//"root";
        String db="inthospitaldb";
        String pwd="";
System.out.println("------------------");
    System.out.println(DData[0]);
 try {
HttpConnection c = (HttpConnection) Connector.open(url);
c.setRequestProperty("User-Agent","Profile/MIDP-1.0,Configuration/CLDC-1.0");
c.setRequestProperty("Content-Language","en-US");
c.setRequestMethod(HttpConnection.POST);
DataOutputStream os = (DataOutputStream)c.openDataOutputStream();
os.writeUTF(db.trim());
os.writeUTF(user.trim());
os.writeUTF(pwd.trim());
os.writeUTF(DData[0].trim());
os.writeUTF(DData[1].trim());
os.writeUTF(DData[2].trim());
os.writeUTF(DData[3].trim());
os.writeUTF(DData[4].trim());
os.writeUTF(DData[5].trim());
os.writeUTF(DData[6].trim());
os.writeUTF(DData[7].trim());
System.out.println("---------Connection was established to :"+os.toString());
os.flush();
os.close();
System.out.println("---------pa========");
// Get the response back from the servlet page.
DataInputStream is =(DataInputStream)c.openDataInputStream();
//is = c.openInputStream();
int ch;
sb = new StringBuffer();
```

```
while ((ch = is.read()) != -1) {
sb.append((char)ch);
}
pa=sb.toString();
pa1=sb.toString();
System.out.println("---------pa=========");
System.out.println(sb);
pb="Mobile info Sent";
 alert = new Alert(pb,pa.toString(),null,null);
is.close();
c.close();
return alert;
} catch (Exception e) {
    System.out.println(e.getMessage());
    pb="Mobile Patient Error:";
    pa1=e.getMessage();
    alert = new Alert(pb,pa1,null,null);
    return alert;
}
}
    /**
     * Called to signal the MIDlet to terminate.
     * @param unconditional if true, then the MIDlet has to be unconditionally terminated and all resources has to be released.
     */
    public void destroyApp(boolean unconditional) {
    }
        private void login() throws Exception {
        StringBuffer sb = new StringBuffer();
        String db="inthospitaldb";
        String pa;
            StringBuffer sb1 = new StringBuffer();
        String pwd="";
       String DataSource=textBox1.getString();
       int a=0;
       int u=0;
       char av=';';
       url = loginScreen.getUsername();//"http://localhost:8084/inthospital/getConnection";
        String user=loginScreen.getPassword();//"root";
        System.out.println(DataSource.toString());
 //double MAP;
if (DataSource.length()!=0 ){
    for (int v=0;v<DataSource.length();v++){
        if (DataSource.charAt(v)==av){
            u=v+1;
            a++;
        }else{
            if (a<8){
            DData[a]=DataSource.substring(u, v+1);
            }
        }
    }
}
    for (int m=0;m<8;m++){
        System.out.println(DData[m]);
        }
```

```
System.out.println(url);
 System.out.println(url);
String opt1=DData[3];
  System.out.println("Option selected-"+opt1);
if (opt1=="1"){
     Systolic=120;//Math.round()*120;// High Systolic Pressure
     Diastolic=80;
}else if (opt1=="3"){
    Systolic=60;//Math.round()*120;// High Systolic Pressure
     Diastolic=40;
}else {
    Systolic=90;//
    Diastolic=60;
}
//PulseRate=Systolic-Diastolic;
 //MAP=Diastolic+(0.033*(Systolic-Diastolic));
DData[6]=Systolic+"\n";
 DData[7]=Diastolic+"\n";
for (int j=0;j<8;j++){
     sb.append(DData[j]+"");
 }
sb1.append(db+";\n");
 sb1.append(user+";\n");
 sb1.append(pwd+";\n");
 sb1.append("Name:"+DData[0]+";\n");
 sb1.append("Tel.No:"+DData[1]+";\n");
 sb1.append("PID:"+DData[2]+";\n");
 sb1.append("Status:"+DData[3]+";\n");
 sb1.append("Age:"+DData[4]+"\n");
 sb1.append("Smoke:"+DData[5]+"\n");
 sb1.append("Sys:"+DData[6]+"\n");
 sb1.append("Dia:"+DData[7]+"\n");
 pa=sb.toString();
       String pa1=sb1.toString();
 try {
HttpConnection c = (HttpConnection) Connector.open(url);
c.setRequestProperty("User-Agent","Profile/MIDP-1.0,Configuration/CLDC-1.0");
c.setRequestProperty("Content-Language","en-US");
c.setRequestMethod(HttpConnection.POST);
DataOutputStream os = (DataOutputStream)c.openDataOutputStream();
os.writeUTF(db.trim());
os.writeUTF(user.trim());
os.writeUTF(pwd.trim());
os.writeUTF(DData[0].trim());
os.writeUTF(DData[1].trim());
os.writeUTF(DData[2].trim());
os.writeUTF(DData[3].trim());
os.writeUTF(DData[4].trim());
os.writeUTF(DData[5].trim());
os.writeUTF(DData[6].trim());
os.writeUTF(DData[7].trim());
os.flush();
os.close();
// Get the response back from the servlet page.
DataInputStream is =(DataInputStream)c.openDataInputStream();
//is = c.openInputStream();
int ch;
sb = new StringBuffer();
```

```
while ((ch = is.read()) != -1) {
sb.append((char)ch);
}
showAlert(sb.toString());
is.close();
c.close();
} catch (Exception e) {
showAlert(e.getMessage());
}
}
  private void showAlert(String err) {
Alert a = new Alert("");
a.setString(err);
a.setTimeout(Alert.FOREVER);
//display.setCurrent(a);
}
}
```

Source Code for Blood Pressure Loader

```
Private Sub MDIForm_Load()
   Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
   Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
   Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 14500)
   Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 10500)
'   LoadNewDoc
   'Initialize Flag for Excel running or not - assume it was unless we start it
   blnExcelWasNotRunning = False
   blnExcelVisible = True
   gstrExcel = "Excel.Application"
End Sub
Private Sub MDIForm_Unload(Cancel As Integer)
   If Me.WindowState <> vbMinimized Then
      SaveSetting App.Title, "Settings", "MainLeft", Me.Left
      SaveSetting App.Title, "Settings", "MainTop", Me.Top
      SaveSetting App.Title, "Settings", "MainWidth", Me.Width
      SaveSetting App.Title, "Settings", "MainHeight", Me.Height
   End If
   'If Excel was not running close it down
   If blnExcelWasNotRunning Then
      'Don't have Excel bother us with dialog boxes
      gobjExcel.DisplayAlerts = False
      'Close down Excel
      gobjExcel.Quit
   End If
End Sub
Private Sub mnDelete_Click()
'OpenMainFiles
OpenOUTPUT
msgRec = InputBox("Enter Patient ID to delete.", "Deletion  Routine")
If msgRec = Null Or msgRec = "" Then
Exit Sub
End If
Set recSelect = dbUBS2.Execute("SELECT * FROM emr where PiD=" & "'" & UCase(msgRec) & "'")
If recSelect.EOF Then
MsgBox ("Record not found.")
Exit Sub
Else
mgsrec2 = MsgBox("Delete this Patient ID now!", vbYesNo)
If mgsrec2 = vbYes Then
dbUBS2.Execute ("delete * FROM emr where PiD=" & "'" & UCase(msgRec) & "'")
```

```
End If
End If
End Sub
Private Sub mnuHelpAbout_Click()
   'To Do
   MsgBox "Still under construction."
End Sub
Private Sub mnuViewOptions_Click()
   'To Do
   MsgBox "Options Dialog Code goes here!"
End Sub
Private Sub mnuViewStatusBar_Click()
   If mnuViewStatusBar.Checked Then
      sbStatusBar.Visible = False
      mnuViewStatusBar.Checked = False
   Else
      sbStatusBar.Visible = True
      mnuViewStatusBar.Checked = True
   End If
End Sub
Private Sub mnuViewToolbar_Click()
   If mnuViewToolbar.Checked Then
      tbToolBar.Visible = False
      mnuViewToolbar.Checked = False
   Else
      tbToolBar.Visible = True
      mnuViewToolbar.Checked = True
   End If
End Sub
Private Sub tbToolBar_ButtonClick(ByVal Button As ComctlLib.Button)
   Select Case Button.Key
      Case "OPEN"
         mnuFileNew_Click
      Case "Open"
         mnuFileOpen_Click
      Case "Save"
         mnuFileSave_Click
      Case "Print"
         mnuFilePrint_Click
   End Select
End Sub
'Code for WB!
Private Sub VBABuildXYZ()
'On Error GoTo 10
   'This procedure specifies the ABCs of the XYZ model and solves the model.
   'This procedure is similar to BuildXYZ in the XYZVBA.XLS file with the
   'difference being that from VB 5.0 an Excel object must be used.
   Dim Result As Integer
   Dim dblObjective As Double
   Dim intStatus As String
   Dim intSolutionStatus As Integer
   'Get a handle to Excel
   Result = GetExcel(gobjExcel)
   'Open the workbook
   gobjExcel.workbooks.Open sFile, UpdateLinks:=3
 '   gobjExcel.Run "Button2_Click"
   ShowExcel
10:
'MsgBox ("This Module Restore scores cannot be found on the loaded Dataset. Check to see if the appropriate file have been loaded.")
End Sub
'Code for WB!
Private Sub ShowExcel()
```

```
    On Error Resume Next
    'Set the Excel Visible flag to TRUE
    blnExcelVisible = True
    'See if there is an excel running
    Set gobjExcel = GetObject(, gstrExcel)
    If Err.Number <> 0 Then
        Err.Clear
    Else
        'Make excel visible
        gobjExcel.Visible = blnExcelVisible
    End If
End Sub
'Code for WB!
Private Sub HideExcel()
    On Error Resume Next
    'Set the Excel Visible flag to FALSE
    blnExcelVisible = False
    'See if there is an excel running
    Set gobjExcel = GetObject(, gstrExcel)
    If Err.Number <> 0 Then
        Err.Clear
    Else
        'Make excel invisible
        gobjExcel.Visible = blnExcelVisible
    End If
End Sub
'Code for WB!
Function GetExcel(objExcel As Object) As Integer
    'This function Finds Excel and runs it if it is not currently
    'It returns 1 if it was sucessful or 0 if not
    Dim intResult As Integer
    'Assume everything will work out fine
    GetExcel = 1
    On Error Resume Next
    'First see if there is an excel running
    Set objExcel = GetObject(, gstrExcel)
    If Err.Number <> 0 Then
        Err.Clear
        'Excel was not running
        blnExcelWasNotRunning = True
        'Assume
 NOT visible since we started Excel
        blnExcelVisible = False
        Set objExcel = CreateObject(gstrExcel)
        If Err.Number <> 0 Then
            'Life is hopeless!
            MsgBox "Unable to get a handle to Excel"
            GetExcel = 0
        End If
    End If
End Function
Private Sub mnuHelpContents_Click()
    Dim nRet As Integer
    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hWnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBox Err.Description
```

```
      End If
   End If
End Sub
Private Sub mnuHelpSearch_Click()
   Dim nRet As Integer
   'if there is no helpfile for this project display a message to the user
   'you can set the HelpFile for your application in the
   'Project Properties dialog
   If Len(App.HelpFile) = 0 Then
      MsgBox "Unable to display Help Contents. There is no Help associated with this project.", vbInformation, Me.Caption
   Else
      On Error Resume Next
      nRet = OSWinHelp(Me.hWnd, App.HelpFile, 261, 0)
      If Err Then
         MsgBox Err.Description
      End If
   End If
End Sub
Private Sub mnuWindowArrangeIcons_Click()
   Me.Arrange vbArrangeIcons
End Sub
Private Sub mnuWindowCascade_Click()
   Me.Arrange vbCascade
End Sub
Private Sub mnuWindow
NewWindow_Click()
   'To Do
   MsgBox "New Window Code goes here!"
End Sub
Private Sub mnuWindowTileHorizontal_Click()
   Me.Arrange vbTileHorizontal
End Sub
Private Sub mnuWindowTileVertical_Click()
   Me.Arrange vbTileVertical
End Sub
Private Sub mnuViewRefresh_Click()
   'To Do
   MsgBox "Refresh Code goes here!"
End Sub
Private Sub mnuEditCopy_Click()
   'To Do
   MsgBox "Copy Code goes here!"
End Sub
Private Sub mnuEditCut_Click()
   'To Do
   MsgBox "Cut Code goes here!"
End Sub
Private Sub mnuEditPaste_Click()
   'To Do
   MsgBox "Paste Code goes here!"
End Sub
Private Sub mnuEditPasteSpecial_Click()
   'To Do
   MsgBox "Paste Special Code goes here!"
End Sub
Private Sub mnuEditUndo_Click()
   'To Do
   MsgBox "Undo Code goes here!"
End Sub
Private Sub mnuFileOpen_Click()
   With dlgCommonDialog
      'To Do
      'set the flags and attributes of the
```

```
        'common dialog control
        .Filter = "All Files (*.CSV*)|*.CSV"
        .ShowOpen
        If Len(.FileName) = 0 Then
           Exit Sub
        End If
        sFile = .FileName
'       gobjExcel.Run "Button2_Click"
mfile = (UCase(Left$(dlgCommonDialog.FileTitle, (Len(dlgCommonDialog.FileTitle) - 4))))
        VBABuildXYZ
        frmAPP.Command1.Enabled = True
    End With
    If Not UCase(Right$(sFile, 3)) = "CSV" Then
    MsgBox ("Selected file is invalid.")
    sFile = ""
    Exit Sub
    End If
'OpenDBFiles
End Sub
Private Sub mnuFileClose_Click()
    Set gobjExcel = Nothing
sFile = ""
End Sub
Private Sub mnuFileSave_Click()
    'To Do
    MsgBox "Save Code goes here!"
End Sub
Private Sub mnuFilePageSetup_Click()
    dlgCommonDialog.ShowPrinter
End Sub
Private Sub mnuFilePrint_Click()
   SearchString = InputBox("Enter the Patient ID to print", "PaientID")
   If SearchString = "" Then
   MsgBox ("No Entry")
   Exit Sub
   End If
 '  OpenMainFiles
    OpenOUTPUT
   Set recSelect = dbUBS2.Execute("Select  * FROM emr WHERE PiD=" & "'" & UCase(Trim(SearchString)) & "'" & "ORDER BY
DateCount")
If recSelect.EOF Then
MsgBox ("End of File Reach or the Patient ID was not found in the database")
Exit Sub
Else
With Datarep
    .Caption = "Automatd Intelligent Hospital LTD "
    .BottomMargin = 400
   .TopMargin = 100
   ' .Width = 800
   .RightMargin = 0.5
   .LeftMargin = 0.5
     Set .DataSource = recSelect
        .Sections("Section4").Controls(2).Caption = "Patient Blood Pressure Report"
        .Sections("Section4").Controls(1).Caption = "Automated Intelligent Hospital LTD "
         .Sections("Section2").Controls(1).Caption = recSelect!PiD
     .Show
End With
'===========
End If
End Sub
Private Sub mnuFileSend_Click()
    'To Do
    MsgBox "Send Code goes here!"
```

```
End Sub
Private Sub mnuFileMRU_Click(Index As Integer)
   'To Do
   MsgBox "MRU Code goes here!"
End Sub
Private Sub mnuFileExit_Click()
On Error Resume Next
   Dim Result As Integer
   Dim dblObjective As Double
   Dim intStatus As String
   Dim intSolutionStatus As Integer
   'Get a handle to Excel
   Result = GetExcel(gobjExcel)
   'Open the workbook
   'gobjExcel.workbooks.Open App.Path & "\GPFILES.XLS", UpdateLinks:=3
 '   gobjExcel.workbooks.Open sFile, UpdateLinks:=3
   gobjExcel.Run "Button1_Click"
   gobjExcel.workbooks.Close sFile, UpdateLinks:=3
10:
'MsgBox ("This Module Restore scores cannot be found on the loaded Dataset. Check to see if the appropriate file have been loaded.")
   'unload the form
   Unload Me
  End
End Sub
Private Sub mnuFileNew_Click()
If sFile = "" Then
MsgBox ("Use File OPEN command to select a Worksheet file")
Exit Sub
Else
MsgBox ("Still under construction.")
  ' LoadNewDoc
End If
End Sub
'Code for WB!
Private Sub mnuShow_Click()
VBABuildXYZ
   'Show Excel
   ShowExcel
End Sub
'Code for WB!
Private Sub mnuHide_Click()
   'Hide Excel
   HideExcel
End Sub
Option Explicit
Private Sub cmdApply_Click()
   MsgBox "Place code here to set options w/o closing dialog!"
End Sub
Private Sub cmdCancel_Click()
'    Unload Me
End Sub
Private Sub Command1_Click()
On Error Resume Next
Recfound = False
OpenMainFiles
OpenOUTPUT
If mfile = Null Or mfile = "" Then
MsgBox ("No file selected yet. Use File Open command to load a CSV file.")
Exit Sub
End If
Set recSelect2 = dbUBS.Execute("SELECT * FROM [" & mfile & "$]")
If recSelect2.EOF Then
```

```
Else
recSelect2.MoveFirst
Do
Fd1 = recSelect2.Fields("User Name")
'If IsEmpty(recSelect2.Fields("Memo 1")) = False Or Len(recSelect2.Fields("Memo 1")) < 1 Then recSelect2.Fields("Memo
1") = "-"
Fd2 = recSelect2.Fields("Memo 1")
'If IsEmpty(recSelect2.Fields("Memo 2")) = False Or Len(recSelect2.Fields("Memo 2")) < 1 Then recSelect2.Fields("Memo
2") = "-"
Fd3 = recSelect2.Fields("Memo 2")
Fd4 = recSelect2.Fields("Date")
Fd5 = recSelect2.Fields("Time")
Fd6 = recSelect2.Fields("SYS (mmHg)")
Fd7 = recSelect2.Fields("DIA (mmHg)")
Fd8 = recSelect2.Fields("Pulse (Pulses/min)")
Fd9 = recSelect2.Fields("Irregular Heartbeat (y or n)")
Fd10 = recSelect2.Fields("Excessive Movement (y or n)")
Fd11 = recSelect2.Fields("Memo (y or n)")
Fd12 = recSelect2.Fields("MsCnt1")
Fd13 = recSelect2.Fields("MsCnt2")
Fd14 = recSelect2.Fields("MsCnt3")
Fd15 = recSelect2.Fields("InputType")
Fd16 = recSelect2.Fields("MeasureType")
Set recSelect = dbUBS2.Execute("Select * FROM emr WHERE PiD=" & "'" & Fd1 & "'" & " AND (DateCount=" & CDate(Fd4) &
" AND TimeCount=" & "'" & Fd5 & "')" & " ORDER BY DateCount")
If recSelect.EOF Then
dbUBS2.Execute " INSERT INTO emr " _
& "(PiD,Memo1,Memo2,DateCount,TimeCount,SYS,DIA,Pulse,IrregularHeartbeat," _
& "ExcessiveMovement,Memo,MsCnt1,MsCnt2,MsCnt3,InputType,MeasureType) VALUES " _
& "('" & Fd1 & "'" & ",'" & Fd2 & "'" _
& ",'" & Fd3 & "'" & ",'" & Fd4 & "'" _
& ",'" & Fd5 & "'" & "," & Val(Fd6) & "" _
& "," & Val(Fd7) & "" & "," & Val(Fd8) & "" _
& ",'" & Fd9 & "'" & ",'" & Fd10 & "'" _
& ",'" & Fd11 & "'" & "," & Val(Fd12) & "" _
& "," & Val(Fd13) & "" & "," & Val(Fd14) & "" _
& "," & Val(Fd15) & "" & "," & Val(Fd16) & "" _
 & ");"
End If
' BALSpecialSAve = SumSpecialTemp
'Ncount = Ncount + 1
recSelect2.MoveNext
Loop Until recSelect2.EOF
End If
MsgBox ("Completing Uploading  Blood pressure records.")
Set recSelect = Nothing
 mfile = ""
Set recSelect2 = Nothing
'dbUBS2.Close
'dbUBS.Close
sns = MsgBox("Send Txt Message to a docor!", vbYesNo, "Doctor Alert")
If sns = vbYes Then
'frmBrowser.Show
End If
Command1.Enabled = False
End Sub
Private Sub Form_Load()
   'center the form
   'Me.Move (Screen.Width - Me.Width) / 2, (Screen.Height - Me.Height) / 2
End Sub
Option Explicit
Dim Allowed As Boolean
Dim TheDate As Date
```

```
Dim Msg, BuyNow
Private Sub Form_KeyPress(KeyAscii As Integer)
'   Unload Me
End Sub
Private Sub Form_Load()
  '  lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." & App.Revision
    Label2.Caption = App.ProductName
   ' lblCopyright.Caption = App.LegalCopyright
    TheDate = CDate("6/30/20")
    Msg = DateDiff("d", Now, TheDate)
    Allowed = True
End Sub
Private Sub Frame1_Click()
 '   Unload Me
End Sub
Private Sub Timer1_Timer()
'MsgBox Msg
 If Timer1.Interval = 5000 Then
    If Allowed = True Then
        Unload Me
       frmAPP.Show
        Timer1.Enabled = False
    End If
    End If
End Sub
    Global dbUBS As ADODB.Connection
Global dbUBS2 As ADODB.Connection
Global recSelect As ADODB.Recordset
Global recSelect2 As ADODB.Recordset
Public fMainForm As frmMain
Global sFile, sns As String
Global Fd1 As String
Global Fd2 As String
Global Fd3 As String
Global Fd4 As String
Global Fd5 As String
Global Fd6 As String
Global Fd7 As String
Global Fd8 As String
Global Fd9 As String
Global Fd10 As String
Global Fd11 As String
Global Fd12 As String
Global Fd13 As String
Global Fd14 As String
Global Fd15 As String
Global Fd16 As String
Global Strcnn As String
Global mfile As String
Global Recfound As Boolean
Global gobjExcel As Object
Global blnExcelWasNotRunning As Boolean
Global blnExcelVisible As Boolean
Global gstrExcel As String
' Declare necessary API routines:
Public Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal hWnd&, ByVal HelpFile$, ByVal wCommand%,
dwData As Any)
Sub Main()
    frmSplash.Show
    Set fMainForm = New frmMain
    fMainForm.Show
End Sub
Public Sub OpenOUTPUT()
```

```
'Strcnn = "Provider=Microsoft.Jet.OLEDB.4.0;Persist Security Info=False;"
'Strcnn = Strcnn & "Data Source=" & App.Path & "\OUTPUTTB.mdb"
Strcnn = "Provider=MSDASQL.1;Persist Security Info=False;Data Source=inthospitalDB"
Set dbUBS2 = New ADODB.Connection
dbUBS2.Open Strcnn
End Sub
Public Sub OpenMainFiles()
Strcnn = "Provider=MSDASQL.1;Persist Security Info=False;Extended Properties=" _
& "DBQ=" & sFile _
& ";Driver={Microsoft Excel Driver (*.xls)};DriverId=790;FIL=excel
8.0;MaxBufferSize=2048;MaxScanRows=8;PageTimeout=5;ReadOnly=1;SafeTransactions=0;Threads=3;UID=admin;UserCo
mmitSync=Yes;"
Set dbUBS = New ADODB.Connection
dbUBS.Open Strcnn
End Sub
```

Program listing of Java web Application

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package inthospital;
import java.util.Date;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.*;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Properties;
import sun.jdbc.odbc.JdbcOdbcDriver;
public class HospitalClass
{
   private String HID;
  private String PatStatus;
  private String systemDate;
  private String userName;
  private String userPassWord;
   private  String Location="";
     private ArrayList items;
   private String[] isite;
   private String[] iAssociate;
  int trycount=0;
   String  success="N";
  private String ExpertNamEemail="";
   private String History="";
   private String AfterSurgWard="";
   private int Bal=0;
   private String BedNo="";
   private String BedTransfer="";
   private int Bill=0;
   private String BloodGp="";
   private String BloodPressure="";
   private String BloodSugar="";
   private String BloodTest="";
   private String Cardiology="";
   private String ChainOfCntrl="";
   private String Cholera="";
   private String CMO="";
```

```
private String CMOQualif="";
private String DateAppt="";
private String DateEstab="";
private String Diagnosis="";
private String DischargeDate="";
private String DocIc="";
private String DoctorIc="";
private String DrugType="";
private String DutyRoster="";
private String Ear="";
private String Eemail="";
private String EMR="";
private String EquipNoid="";
private String EquipStatus="";
private String EquipType="";
private String EResidence="";
private String ETel="";
private String EvaluationOfSuccess="";
private String Expertise="";
private String ExpertName="";
private String Eye="";
private String Function="";
private String Gender="";
private String Hemail="";
private String Hepatitis="";
private String HExpertise="";
private String HiD="";
private String HIV="";
private String HospAdmtd="";
private String HospAvailability="";
private String HospLocation="";
private String HospNameAdr="";
private String Hownership="";
private String HWard="";
private String Lab="";
private String Malaria="";
private String MedicalHistory="";
private String Mouth="";
private int NairaValueBal=0;
private String OT="";
private String OtherTests="";
private String PAddress="";
private String Pemail="";
private String PiD="";
private String Pname="";
private String PName="";
private String Remarks;
private String Prescription="";
private String PTel="";
private int Qty=0;
private int QtyinStock=0;
private int QtyIssuedOut=0;
private String Qualifications="";
private String Rank="";
private String Recall="";
private String ReferedFrom="";
private String RemarksfromCDS="";
private int ReorderLevel=0;
private String Reportsto="";
private String Semail="";
private String SensorReportsPDA="";
private String Sponsor="";
private String StaffNameIc="";
```

```
    private String STel="";
    private String Stool="";
    private String SurgeryDate="";
    private String SurgeryType="";
    private int TBillPdtoDate=0;
    private int TBilltoDate=0;
    private String Temp="";
    private String Duty_Roster="";
    private String Reports_To="";
    private String Tools_Req="";
    private String Treatment="";
    private String DrugID="";
    private String Tuberculosis="";
     private String Description="";
    private String Urine="";
    private String Ward="";
    private String WWW="";
     private String PSat="";
     private String DateDischarged="";
      private String Transfer="";
      private String DateAdmtd="";
     private  String dbName = "inthospitalDB";
 //    String driver = new String("org.apache.derby.jdbc.ClientDriver" );
  String driver = new String("sun.jdbc.odbc.JdbcOdbcDriver" );
  String protocol = new String("jdbc:odbc:"+dbName);
//String protocol = new String("jdbc:derby://localhost:1527/"+dbName);
//jdbc:derby://localhost:1527/inthospitalDB
   String framework = "embedded";
   // private String driver = "org.apache.derby.jdbc.EmbeddedDriver";
    //private String protocol = "jdbc:derby:";
       ResultSet rs = null;
     Connection connCS = null;
    public Connection conn = null;
        ArrayList statements = new ArrayList(); // list of Statements, PreparedStatements
       PreparedStatement psInsert = null;
        PreparedStatement psUpdate = null;
       Statement s = null;
//public static final String CSdriver = new String("org.apache.derby.jdbc.EmbeddedDriver");
//public static final String dbURLCS = new String("jdbc:derby:/inthospitalDB");
public HospitalClass()
{
        go();
        System.out.println("Intelligent Hospital finished.");
    }
   public void go()
    {
        try
        {
       loadDriver();
       Properties p = new Properties();
p.put("user", "admin");
p.put("password", "");
  conn = DriverManager.getConnection(protocol);
  System.out.println("Successfully got the database connection...");
        } catch (SQLException e) {
                    System.out.println ("FAIL -- unexpected exception: " + e.toString());
                     e.printStackTrace();
             } catch (Exception e) {
                     System.out.println ("FAIL -- unexpected exception: " + e.toString());
                     e.printStackTrace();
             }
    }
        private void reportFailure(String message) {
```

```
    System.err.println("\nData verification failed:");
    System.err.println('\t' + message);
  }
  /**
   * Prints details of an SQLException chain to <code>System.err</code>.
   * Details included are SQL State, Error code, Exception message.
   *
   * @param e the SQLException from which to print details.
   */
  public static void printSQLException(SQLException e)
  {
    // Unwraps the entire exception chain to unveil the real cause of the
    // Exception.
    while (e != null)
    {
      System.err.println("\n----- SQLException -----");
      System.err.println("  SQL State:  " + e.getSQLState());
      System.err.println("  Error Code: " + e.getErrorCode());
      System.err.println("  Message:    " + e.getMessage());
      // for stack traces, refer to derby.log or uncomment this:
      //e.printStackTrace(System.err);
      e = e.getNextException();
    }
  }
 public ArrayList getItems() {
  return items;
 }
public void addItem(String id) {
   items.add(id.intern());
 }
  public int getSize() {
  int size = 0;
  if (items != null)
    return size = items.size();
  return size;
 }
 public String getOutHiD(HttpServletRequest request,HttpServletResponse
      response){
   if (request.getMethod().equals("POST")) {
  HiD=request.getParameter("HiD");
  PatStatus=request.getParameter("Select1");
  SearchOutForID(request,response);
  }
  return "y";
  }
  public String getOPTHiD(HttpServletRequest request,HttpServletResponse
       response){
    if (request.getMethod().equals("POST")) {
   HiD=request.getParameter("HiD");
   PatStatus=request.getParameter("Select1");
   SearchOPTForID(request,response);
   }
   return "y";
   }

 public String getInHiD(HttpServletRequest request,HttpServletResponse
     response){
 if (request.getMethod().equals("POST")) {
HiD=request.getParameter("HiD");
PatStatus=request.getParameter("Select1");
SearchInForID(request,response);
}
return "y";
```

```
  }
int opt=0;
  public String getAddEMR(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
    response.setContentType("text/html");
PrintWriter out = response.getWriter();
   if (request.getMethod().equals("POST")) {
  Pname=request.getParameter("PName");
  PiD=request.getParameter("PiD");
  History=request.getParameter("History");
  SensorReportsPDA=request.getParameter("SensorReportsPDA");
  RemarksfromCDS=request.getParameter("RemarksFromCDS");
  System.out.print(PiD+" "+Pname+" "+History+ " "+SensorReportsPDA+" "+RemarksfromCDS) ;
  flag=false;
  try{
 prepStmt = conn.prepareStatement("select * from emr where PiD=?");
                        prepStmt.setString(1,PiD);
                          rs=prepStmt.executeQuery();
                          flag=rs.next();
                        if (flag){
                          System.out.println("About to connec and update");
    prepStmt2 = conn.prepareStatement(
                        "update emr set Pname=?, PiD=?, History=?, SensorReportsPDA=?, "+
                       "RemarksfromCDS=? "+ "where PiD=?");
                         prepStmt2.setString(1,Pname);
                         prepStmt2.setString(2,PiD);
                           prepStmt2.setString(3,History);
                         prepStmt2.setString(4,SensorReportsPDA);
                         prepStmt2.setString(5,RemarksfromCDS);
                         prepStmt2.setString(6,PiD);
                         prepStmt2.executeUpdate();
                         out.println("<html>");
                         out.println("<head><title>Action completed.</title>");
                         out.println("</head>");
                         out.println("<body>");
                         out.println("<H1 ALIGN=\"CENTER\">Record Update was successful</H1>");
                         out.println("<P>");
                         out.println("<P>");
                         out.println("</body></html>");
                       }
                       if (!flag){
                         System.out.println("About to connec and insert");
                         prepStmt = conn.prepareStatement("insert into emr values(?,?,?,?,?)");
                         prepStmt.setString(1,PName);
                         prepStmt.setString(2,PiD);
                         prepStmt.setString(3,History);
                         prepStmt.setString(4,SensorReportsPDA);
                         prepStmt.setString(5,RemarksfromCDS);
                         prepStmt.executeUpdate();
                         out.println("<html>");
                         out.println("<head><title>Action completed.</title>");
                         out.println("</head>");
                         out.println("<body>");
                         out.println("<H1 ALIGN=\"CENTER\">Record Insertion was successful</H1>");
                         out.println("<P>");
                         out.println("<P>");
                         out.println("</body></html>");
                       }
                         } catch (Exception ex) {
                           out.println("<html>");
                      out.println("<head><title>Action completed.</title>");
                      out.println("</head>");
                      out.println("<body>");
```

```
                    out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                    out.println("<P>");
                    out.println("<P>");
                    out.println("</body></html>");
                              }
//  AddEMR();
  }
  return "y";
  }
  public String DateOfAnalysis="";
  public String getDateOfAnalysis(){
    return DateOfAnalysis;
  }
  public String getAddCDS(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
   response.setContentType("text/html");
PrintWriter out = response.getWriter();
  if (request.getMethod().equals("POST")) {
 PiD=request.getParameter("PiD");
BloodTest=request.getParameter("BloodTest");BloodSugar=request.getParameter("BloodSugar");
 Urine=request.getParameter("Urine"); Cardiology=request.getParameter("Cardiology");
BloodPressure=request.getParameter("BloodPressure");
 Temp=request.getParameter("Temp"); Malaria=request.getParameter("Malaria"); HIV=request.getParameter("HIV");
 Tuberculosis=request.getParameter("Tuberculosis"); Cholera=request.getParameter("Cholera");
Hepatitis=request.getParameter("Hepatitis");
 Eye=request.getParameter("Eye"); Eye=request.getParameter("Eye"); Mouth=request.getParameter("Mouth");
 Stool=request.getParameter("Stool"); OtherTests=request.getParameter("OtherTests");
Remarks=request.getParameter("Remarks");
 DateOfAnalysis=request.getParameter("DateOfAnalysis");
 flag=false;
 try{
prepStmt = conn.prepareStatement("select * from CDS where PiD=?");
                    prepStmt.setString(1,PiD);
                      rs=prepStmt.executeQuery();
                      flag=rs.next();
                    if (flag){
                      System.out.println("About to connec and update");
    prepStmt2 = conn.prepareStatement(
                     "update CDS set PiD=?, BloodTest=?, Urine=?, "+
                    "Cardiology=?, BloodPressure=?, BloodSugar=?, Temp=?, "+
                    "Malaria=?, HIV=?, Tuberculosis=?, Cholera=?, "+
                    "Hepatitis=?, Eye=?, Ear=?, Mouth=?, "+
                    "Stool=?, Other_Tests=?, Remarks=?, DateOfAnalysis=? "+
                     "where PiD=?");
prepStmt2.setString(1,PiD);prepStmt2.setString(2,BloodTest);
prepStmt2.setString(3,Urine);prepStmt2.setString(4,Cardiology);
prepStmt2.setString(5,BloodPressure);prepStmt2.setString(6,BloodSugar);
prepStmt2.setString(7,Temp);prepStmt2.setString(8,Malaria);
prepStmt2.setString(9,HIV);prepStmt2.setString(10,Tuberculosis);
prepStmt2.setString(11,Cholera);prepStmt2.setString(12,Hepatitis);
prepStmt2.setString(13,Eye);prepStmt2.setString(14,Ear);
prepStmt2.setString(15,Mouth);prepStmt2.setString(16,Stool);
prepStmt2.setString(17,OtherTests);prepStmt2.setString(18,Remarks);
prepStmt2.setString(19,DateOfAnalysis);prepStmt2.setString(20,PiD);
prepStmt2.executeUpdate();
                     out.println("<html>");
                     out.println("<head><title>Action completed.</title>");
                     out.println("</head>");
                     out.println("<body>");
                     out.println("<H1 ALIGN=\"CENTER\">Record Update was successful</H1>");
                     out.println("<P>");
                     out.println("<P>");
                     out.println("</body></html>");
```

```
                    }
                    if (!flag){
                     System.out.println("About to connec and insert");
                     prepStmt2 = conn.prepareStatement("insert into CDS values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
                     prepStmt2.setString(1,PiD);prepStmt2.setString(2,BloodTest);
    prepStmt2.setString(3,Urine);prepStmt2.setString(4,Cardiology);
    prepStmt2.setString(5,BloodPressure);prepStmt2.setString(6,BloodSugar);
    prepStmt2.setString(7,Temp);prepStmt2.setString(8,Malaria);
    prepStmt2.setString(9,HIV);prepStmt2.setString(10,Tuberculosis);
    prepStmt2.setString(11,Cholera);prepStmt2.setString(12,Hepatitis);
    prepStmt2.setString(13,Eye);prepStmt2.setString(14,Ear);
    prepStmt2.setString(15,Mouth);prepStmt2.setString(16,Stool);
    prepStmt2.setString(17,OtherTests);prepStmt2.setString(18,Remarks);
    prepStmt2.setString(19,DateOfAnalysis);
    prepStmt2.executeUpdate();
                     out.println("<html>");
                     out.println("<head><title>Action completed.</title>");
                     out.println("</head>");
                     out.println("<body>");
                     out.println("<H1 ALIGN=\"CENTER\">Record Insertion was successful</H1>");
                     out.println("<P>");
                     out.println("<P>");
                     out.println("</body></html>");
                    }
                     } catch (Exception ex) {
                      out.println("<html>");
                   out.println("<head><title>Action completed.</title>");
                   out.println("</head>");
                   out.println("<body>");
                   out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                   out.println("<P>");
                   out.println("<P>");
                   out.println("</body></html>");
                           }
//  AddEMR();
 }
 return "y";
 }
 public String getHiD(HttpServletRequest request,HttpServletResponse response) throws ServletException, IOException {
  response.setContentType("text/html");
PrintWriter out = response.getWriter();
 if (request.getMethod().equals("POST")) {
 PiD=request.getParameter("PiD"); PStatus=request.getParameter("PStatus");
flag=false;
 try{
prepStmt = conn.prepareStatement("select * from ipd where PiD=?");
                    prepStmt.setString(1,PiD);
         //         prepStmt.setString(2,PStatus);
                     rs=prepStmt.executeQuery();
                     flag=rs.next();
if (flag){
 PiD=rs.getString(1);
 Pname=rs.getString(2);
 PAddress=rs.getString(3);
PTel=rs.getString(4);
 Pemail=rs.getString(5);
 Age=rs.getString(6);
ReferedFrom=rs.getString(7);
 DateAdmtd=rs.getString(8);
 Gender=rs.getString(9);
 HospAdmtd=rs.getString(10);
 HWard=rs.getString(11);
 BedNo=rs.getString(12);
```

```
 Diagnosis=rs.getString(13);
 Prescription=rs.getString(14);
 Treatment=rs.getString(15);
DoctorIc=rs.getString(16);
 Sponsor=rs.getString(17);
 STel=rs.getString(18);
Semail=rs.getString(19);
 TBilltoDate=Integer.parseInt(rs.getString(20));
 TBillPdtoDate=Integer.parseInt(rs.getString(21));
DischargeDate=rs.getString(22);
 DischargeStatus=rs.getString(23);
 PStatus=rs.getString(24);
}else{
 PiD=PiD;
TBilltoDate=0;
 TBillPdtoDate=0;
     }
   out.println("<head><title>Action completed.</title>");
out.println("</head>");
out.println("<jsp:useBean id=\"ClientList\" scope=\"session\" class=\"inthospital.HospitalClass\" />");
                          out.println("<body onLoad=\"LoadMe()\">");
                          out.println("<script language='JavaScript' type='text/JavaScript'>");
                               out.println("function LoadMe(){");
                               //   out.println("window.open('DrugPage.jsp');");
                  out.println("location.href='PatientRegistration.jsp';");
                               out.println("};");
                                out.println("</script>");
                         out.println("</body></html>");
                        } catch (Exception ex) {
                          out.println("<html>");
                     out.println("<head><title>Action completed.</title>");
                     out.println("</head>");
                     out.println("<body>");
                     out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                     out.println("<P>");
                     out.println("<P>");
                     out.println("</body></html>");
                                 }
//  AddEMR();
 }
 return "y";
 }
 public String Age="";
 public String getAge(){
   return Age;
 }
 public String DischargeStatus="";
  public String getDischargeStatus(){
    return DischargeStatus;
  }
      public String PStatus="";
  public String getPStatus(){
    return PStatus;
  }
 public String getAddPatient(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
   response.setContentType("text/html");
PrintWriter out = response.getWriter();
  if (request.getMethod().equals("POST")) {
 PiD=request.getParameter("PiD"); Pname=request.getParameter("PName");PAddress=request.getParameter("PAddress");
 PTel=request.getParameter("PTel"); Pemail=request.getParameter("Pemail"); Age=request.getParameter("Age");
 ReferedFrom=request.getParameter("ReferedFrom"); DateAdmtd=request.getParameter("DateAdmtd");
Gender=request.getParameter("Gender");
```

```
 HospAdmtd=request.getParameter("HospAdmtd"); HWard=request.getParameter("HWard");
BedNo=request.getParameter("BedNo");
 Diagnosis=request.getParameter("Diagnosis"); Prescription=request.getParameter("Prescription");
Treatment=request.getParameter("Treatment");
 DoctorIc=request.getParameter("DoctorIc"); Sponsor=request.getParameter("Sponsor");
STel=request.getParameter("STel");
 Semail=request.getParameter("SEmail"); Recall=request.getParameter("Recall");
 TBilltoDate=Integer.parseInt(request.getParameter("TBilltoDate"));
 TBillPdtoDate=Integer.parseInt(request.getParameter("TBillPdtoDate"));
DateDischarged=request.getParameter("DateDischarged"); DischargeStatus=request.getParameter("DischargeStatus");
PStatus=request.getParameter("PStatus");
System.out.print(DischargeDate);
System.out.print(DateAdmtd);
System.out.print(Recall);
flag=false;
 try{
prepStmt = conn.prepareStatement("select * from ipd where PiD=?");
                        prepStmt.setString(1,PiD);
                         rs=prepStmt.executeQuery();
                         flag=rs.next();
                        if (flag){
                         System.out.println("About to connec and update");
   prepStmt2 = conn.prepareStatement(
                        "update ipd set PiD=?, Pname=?, Paddress=?, "+
                       "PTel=?, Pemail=?, Age=?, ReferedFrom=?, "+
                       "DateAdmtd=?, Gender=?, HospAdmtd=?, HWard=?, "+
                       "BedNo=?, Diagnosis=?, Prescription=?, Treatment=?, "+
                       "DoctorIc=?, Sponsor=?, STel=?, "+
                      "Semail=?, TBilltoDate=?, TBillPdtoDate=?, DischargeDate=?, "+
                       "DischargeStatus=?, PStatus=?, RecallDate=? where PiD=?");
prepStmt2.setString(1,PiD);prepStmt2.setString(2,Pname);
prepStmt2.setString(3,PAddress);prepStmt2.setString(4,PTel);prepStmt2.setString(5,Age);
prepStmt2.setString(6,Pemail);prepStmt2.setString(7,ReferedFrom);
prepStmt2.setString(8,DateAdmtd);prepStmt2.setString(9,Gender);
prepStmt2.setString(10,HospAdmtd);prepStmt2.setString(11,HWard);
prepStmt2.setString(12,BedNo);prepStmt2.setString(13,Diagnosis);
prepStmt2.setString(14,Prescription);prepStmt2.setString(15,Treatment);
prepStmt2.setString(16,DoctorIc);prepStmt2.setString(17,Sponsor);
prepStmt2.setString(18,STel);prepStmt2.setString(19,Semail);
 prepStmt2.setInt(20,TBilltoDate);prepStmt2.setInt(21,TBillPdtoDate);
 prepStmt2.setString(22,DischargeDate);prepStmt2.setString(23,DischargeStatus);
prepStmt2.setString(24,PStatus);prepStmt2.setString(25,Recall);prepStmt2.setString(26,PiD);
prepStmt2.executeUpdate();
                         out.println("<html>");
                         out.println("<head><title>Action completed.</title>");
                         out.println("</head>");
                         out.println("<body>");
                         out.println("<H1 ALIGN=\"CENTER\">Record Update was successful</H1>");
                         out.println("<P>");
                         out.println("<P>");
                         out.println("</body></html>");
                       }
                       if (!flag){
                         System.out.println("About to connec and insert");
 prepStmt2 = conn.prepareStatement("insert into ipd values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
             prepStmt2.setString(1,PiD);prepStmt2.setString(2,Pname);
             prepStmt2.setString(3,PAddress);prepStmt2.setString(4,PTel);
              prepStmt2.setString(5,Pemail); prepStmt2.setString(6,Age);
            prepStmt2.setString(7,ReferedFrom);
             prepStmt2.setString(8,DateAdmtd);prepStmt2.setString(9,Gender);
             prepStmt2.setString(10,HospAdmtd);prepStmt2.setString(11,HWard);
             prepStmt2.setString(12,BedNo);prepStmt2.setString(13,Diagnosis);
             prepStmt2.setString(14,Prescription);prepStmt2.setString(15,Treatment);
```

```
            prepStmt2.setString(16,DoctorIc);prepStmt2.setString(17,Sponsor);
            prepStmt2.setString(18,STel);prepStmt2.setString(19,Semail);
             prepStmt2.setInt(20,TBilltoDate);prepStmt2.setInt(21,TBillPdtoDate);
             prepStmt2.setString(22,DischargeDate);prepStmt2.setString(23,DischargeStatus);
            prepStmt2.setString(24,PStatus);prepStmt2.setString(25,Recall);
              prepStmt2.executeUpdate();
                    out.println("<html>");
                    out.println("<head><title>Action completed.</title>");
                    out.println("</head>");
                    out.println("<body>");
                    out.println("<H1 ALIGN=\"CENTER\">Record Insertion was successful</H1>");
                    out.println("<P>");
                    out.println("<P>");
                    out.println("</body></html>");
                   }
                    } catch (Exception ex) {
                      out.println("<html>");
                  out.println("<head><title>Action completed.</title>");
                  out.println("</head>");
                  out.println("<body>");
                  out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                  out.println("<P>");
                  out.println("<P>");
                  out.println("</body></html>");
                           }
//  AddEMR();
 }
 return "y";
 }
  public String getAddStaff(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
  response.setContentType("text/html");
PrintWriter out = response.getWriter();
  if (request.getMethod().equals("POST")) {
 StaffID=request.getParameter("StaffID");
 StaffName=request.getParameter("StaffName");
 Function=request.getParameter("Function");
 Rank=request.getParameter("Rank");
 ChainOfCntrl=request.getParameter("ChainOfCntrl");
 flag=false;
 try{
prepStmt = conn.prepareStatement("select * from hospadmin where StaffID=?");
                    prepStmt.setString(1,StaffID);
                      rs=prepStmt.executeQuery();
                      flag=rs.next();
                    if (flag){
                      System.out.println("About to connec and update");
   prepStmt2 = conn.prepareStatement(
                    "update hospadmin set StaffID=?, StaffName=?, Function=?, Rank=?, "+
                   "ChainOfCntrl=? "+ "where PiD=?");
                     prepStmt2.setString(1,StaffID);
                     prepStmt2.setString(2,StaffName);
                       prepStmt2.setString(3,Function);
                     prepStmt2.setString(4,Rank);
                     prepStmt2.setString(5,ChainOfCntrl);
                     prepStmt2.setString(6,StaffID);
                     prepStmt2.executeUpdate();
                     out.println("<html>");
                     out.println("<head><title>Action completed.</title>");
                     out.println("</head>");
                     out.println("<body>");
                     out.println("<H1 ALIGN=\"CENTER\">Record Update was successful</H1>");
                     out.println("<P>");
```

```
                    out.println("<P>");
                    out.println("</body></html>");
                   }
                   if (!flag){
                    System.out.println("About to connec and insert");
                    prepStmt2 = conn.prepareStatement("insert into hospadmin values(?,?,?,?,?)");
                    prepStmt2.setString(1,StaffID);
                     prepStmt2.setString(2,StaffName);
                       prepStmt2.setString(3,Function);
                     prepStmt2.setString(4,Rank);
                     prepStmt2.setString(5,ChainOfCntrl);
                     prepStmt2.executeUpdate();
                    out.println("<html>");
                    out.println("<head><title>Action completed.</title>");
                    out.println("</head>");
                    out.println("<body>");
                    out.println("<H1 ALIGN=\"CENTER\">Record Insertion was successful</H1>");
                    out.println("<P>");
                    out.println("<P>");
                    out.println("</body></html>");
                   }
                    } catch (Exception ex) {
                      out.println("<html>");
                  out.println("<head><title>Action completed.</title>");
                  out.println("</head>");
                  out.println("<body>");
                  out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                  out.println("<P>");
                  out.println("<P>");
                  out.println("</body></html>");
                          }
//  AddEMR();
 }
 return "y";
 }
  public String getAddEC(HttpServletRequest request,HttpServletResponse response) throws ServletException, IOException
{
    response.setContentType("text/html");
 PrintWriter out = response.getWriter();
   if (request.getMethod().equals("POST")) {
  ExpertID=request.getParameter("ExpertID");   ExpertName=request.getParameter("ExpertName");
  Expertise=request.getParameter("Expertise");
  Qualifications=request.getParameter("Qualification");
  Location=request.getParameter("Location");   ETel=request.getParameter("ETel");
  Eemail=request.getParameter("Eemail");     EResidence=request.getParameter("EResidence");
      HospAvailability=request.getParameter("HospAvailability");
      System.out.println(ExpertID+" "+ExpertName+" "+Qualifications);
         System.out.println(Location+" "+ETel+" "+Eemail);
            System.out.println(EResidence+" "+HospAvailability+" "+Expertise);
 flag=false;
  try{
 prepStmt = conn.prepareStatement("select * from ec where ExpertID=?");
                     prepStmt.setString(1,ExpertID);
                       rs=prepStmt.executeQuery();
                       flag=rs.next();
                     if (flag){
                       System.out.println("About to connec and update");
    prepStmt2 = conn.prepareStatement(
                     "update ec set ExpertID=?, ExpertName=?, Expertise=?, HospLocation=?, "+
                     "Qualification=?, ETel=?, ExpertEmail=?, "+
                     "Eresidence=?, HospAvailability=? where ExpertID=?");
                      prepStmt2.setString(1,ExpertID);
                      prepStmt2.setString(2,ExpertName);
```

```
                    prepStmt2.setString(3,Expertise);
                  prepStmt2.setString(4,Location);
                  prepStmt2.setString(5,Qualifications);
                  prepStmt2.setString(6,ETel);
                    prepStmt2.setString(7,Eemail);
                      prepStmt2.setString(8,EResidence);
                        prepStmt2.setString(9,HospAvailability);
                          prepStmt2.setString(10,ExpertID);
                  prepStmt2.executeUpdate();
                  out.println("<html>");
                  out.println("<head><title>Action completed.</title>");
                  out.println("</head>");
                  out.println("<body>");
                  out.println("<H1 ALIGN=\"CENTER\">Record Update was successful</H1>");
                  out.println("<P>");
                  out.println("<P>");
                  out.println("</body></html>");
                }
                if (!flag){
                  System.out.println("About to connec and insert");
                  prepStmt2 = conn.prepareStatement("insert into ec values(?,?,?,?,?,?,?,?,?)");
                  prepStmt2.setString(1,ExpertID);
                   prepStmt2.setString(2,ExpertName);
                     prepStmt2.setString(3,Expertise);
                   prepStmt2.setString(4,Location);
                   prepStmt2.setString(5,Qualifications);
                   prepStmt2.setString(6,ETel);
                     prepStmt2.setString(7,Eemail);
                       prepStmt2.setString(8,EResidence);
                         prepStmt2.setString(9,HospAvailability);
                          prepStmt2.executeUpdate();
                  out.println("<html>");
                  out.println("<head><title>Action completed.</title>");
                  out.println("</head>");
                  out.println("<body>");
                  out.println("<H1 ALIGN=\"CENTER\">Record Insertion was successful</H1>");
                  out.println("<P>");
                  out.println("<P>");
                  out.println("</body></html>");
                }
                  } catch (Exception ex) {
                    out.println("<html>");
               out.println("<head><title>Action completed.</title>");
               out.println("</head>");
               out.println("<body>");
               out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
               out.println("<P>");
               out.println("<P>");
               out.println("</body></html>");
                              }
//  AddEMR();
  }
  return "y";
  }
  public int OrderQuantity=0;
  public int getOrderQuantity(){
    return OrderQuantity;
  }
  public String getAddDrug(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
    response.setContentType("text/html");
 PrintWriter out = response.getWriter();
   if (request.getMethod().equals("POST")) {
```

```
DrugID=request.getParameter("DrugID");   Description=request.getParameter("Description");
OrderQuantity=Integer.parseInt(request.getParameter("OrderQuantity"));
QtyinStock=Integer.parseInt(request.getParameter("QtyinStock"));
ReorderLevel=Integer.parseInt(request.getParameter("ReorderLevel"));
Location=request.getParameter("Location");   QtySold=Integer.parseInt(request.getParameter("QtySold"));
Remarks=request.getParameter("Remarks");     NairaValueBal=Integer.parseInt(request.getParameter("NairaValueBal"));
     DatePurchased=request.getParameter("DatePurchased");
     flag=false;
try{
prepStmt = conn.prepareStatement("select * from drug where DrugID=?");
                    prepStmt.setString(1,DrugID);
                     rs=prepStmt.executeQuery();
                     flag=rs.next();
                    if (flag){
                     int TempBal=rs.getInt(7) ;
                     System.out.println("About to connec and update");
   prepStmt2 = conn.prepareStatement(
                    "update drug set DrugID=?, Description=?, QtyinStock=?, "+
                    "QtyOrder=?, ReorderLevel=?, QtySold=?, Bal=?, NairaValueBal=?, "+
                    "DatePurchased=?, Location=?, Remarks=?  where DrugID=?");
        prepStmt2.setString(1,DrugID); prepStmt2.setString(2,Description);
        prepStmt2.setInt(3,(OrderQuantity+TempBal)-QtySold); prepStmt2.setInt(4,OrderQuantity);
        prepStmt2.setInt(5,ReorderLevel);    prepStmt2.setInt(6,QtySold);
        prepStmt2.setInt(7,(OrderQuantity+TempBal)-QtySold);
        prepStmt2.setInt(8,NairaValueBal);prepStmt2.setString(9,DatePurchased);
          prepStmt2.setString(10,Location);prepStmt2.setString(11,Remarks);
        prepStmt2.setString(12,DrugID);
                     prepStmt2.executeUpdate();
                     out.println("<html>");
                     out.println("<head><title>Action completed.</title>");
                     out.println("</head>");
                     out.println("<body>");
                     out.println("<H1 ALIGN=\"CENTER\">Record Update was successful</H1>");
                     out.println("<P>");
                     out.println("<P>");
                     out.println("</body></html>");
                    }
                    if (!flag){
                     System.out.println("About to connec and insert");
                     prepStmt2 = conn.prepareStatement("insert into drug values(?,?,?,?,?,?,?,?,?,?,?)");
                     prepStmt2.setString(1,DrugID); prepStmt2.setString(2,Description);
            prepStmt2.setInt(3,(OrderQuantity-QtySold)); prepStmt2.setInt(4,OrderQuantity);
            prepStmt2.setInt(5,ReorderLevel);    prepStmt2.setInt(6,QtySold);
            prepStmt2.setInt(7,(OrderQuantity-QtySold));
            prepStmt2.setInt(8,NairaValueBal);prepStmt2.setString(9,DatePurchased);
              prepStmt2.setString(10,Location);prepStmt2.setString(11,Remarks);
              prepStmt2.executeUpdate();
                     out.println("<html>");
                     out.println("<head><title>Action completed.</title>");
                     out.println("</head>");
                     out.println("<body>");
                     out.println("<H1 ALIGN=\"CENTER\">Record Insertion was successful</H1>");
                     out.println("<P>");
                     out.println("<P>");
                     out.println("</body></html>");
                    }
                     } catch (Exception ex) {
                       out.println("<html>");
                   out.println("<head><title>Action completed.</title>");
                   out.println("</head>");
                   out.println("<body>");
                   out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                   out.println("<P>");
```

```
                        out.println("<P>");
                        out.println("</body></html>");
                                    }
//  AddEMR();
   }
   return "y";
   }
   public String SearchForDrugs(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
     response.setContentType("text/html");
 PrintWriter out = response.getWriter();
    if (request.getMethod().equals("POST")) {
   DrugID=request.getParameter("DrugID");
   System.out.print(DrugID);
  flag=false;
   try{
  prepStmt = conn.prepareStatement("select * from drug where DrugID=?");
                         prepStmt.setString(1,DrugID);
                          rs=prepStmt.executeQuery();
                          flag=rs.next();
                         if (flag){
                         DrugID=rs.getString(1);Description=rs.getString(2);
                         QtyinStock=rs.getInt(7);Location=rs.getString(10);
                         Remarks=rs.getString(11);
                        DatePurchased=rs.getString(9);
                          }else{
                          DrugID=DrugID;
                          Description="";
                             QtyinStock=0;Location="";
                             Remarks="";
                         }
                         out.println("<html>");
out.println("<head><title>Action completed.</title>");
out.println("</head>");
out.println("<jsp:useBean id=\"ClientList\" scope=\"session\" class=\"inthospital.HospitalClass\" />");
                          out.println("<body onLoad=\"LoadMe()\">");
                          out.println("<script language='JavaScript' type='text/JavaScript'>");
                               out.println("function LoadMe(){");
                                //   out.println("window.open('DrugPage.jsp');");
                               out.println("location.href='DrugPage.jsp';");
                               out.println("};");
                                 out.println("</script>");
                         out.println("</body></html>");
} catch (Exception ex) {
                           out.println("<html>");
                        out.println("<head><title>Action completed.</title>");
                        out.println("</head>");
                        out.println("<body>");
                        out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                        out.println("<P>");
                        out.println("<P>");
                        out.println("</body></html>");
                                    }
   }
   return "y";
   }
public String SearchForEMR(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
     response.setContentType("text/html");
 PrintWriter out = response.getWriter();
    if (request.getMethod().equals("POST")) {
   PiD=request.getParameter("PiD");
   System.out.print(PiD);
```

```
  flag=false;
       out.println("<html>");
       out.println("<head><title>Hospital EMR/Clinical</title>");
       out.println("<style type=\"text/css\">");
       out.println("<!--");
       out.println(".style2 {font-size: 10px}");
       out.println("-->");
       out.println("</style>");
       out.println("</head>");
       out.println("<body>");
       out.println("<P>");
     out.print("<form name=\"form1\" method=\"post\">");
     out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
     out.println("<H1 ALIGN=\"CENTER\">Hospital EMR/Clinical Reports</H1>");
      out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
       out.print("<table width=\"731\" border=\"1\">");
   try{
  prepStmt = conn.prepareStatement("select * from emr where PiD=?");
                              prepStmt.setString(1,PiD);
 rs=prepStmt.executeQuery();
          out.print("<tr>");
        //  out.println("<H7");
          out.print("<td width=\"50\"><span class=\"style2\">Patient ID</span></td>");
           out.print("<td width=\"20\"><span class=\"style2\">Date</span></td>");
            out.print("<td width=\"20\"><span class=\"style2\">Time</span></td>");
            out.print("<td width=\"20\"><span class=\"style2\">Sys</span></td>");
             out.print("<td width=\"30\"><span class=\"style2\">DIA</span></td>");
               out.print("<td width=\"20\"><span class=\"style2\">Irr. Heatbeat</span></td>");
                out.print("<td width=\"20\"><span class=\"style2\">Pulse</span></td>");
                  out.print("<td width=\"20\"><span class=\"style2\">Excess Moves</span></td>");
                    out.print("<td width=\"10\"><span class=\"style2\">Mscnt1</span></td>");
                      out.print("<td width=\"10\"><span class=\"style2\">MscCnt2</span></td>");
                        out.print("<td width=\"10\"><span class=\"style2\">MsCnt3</span></td>");
                          out.print("<td width=\"20\"><span class=\"style2\">Input Typpe</span></td>");
                            out.print("<td width=\"20\"><span class=\"style2\">Meaure Type</span></td>");
                              out.print("<td width=\"70\"><span class=\"style2\">Memo</span></td>");
               out.print("</tr>");
             while( rs.next() ){
            out.print("<tr>");
             out.print("<tr>");
             out.print("<td>"+rs.getString(3)+"</td>");
            out.print("<td>"+rs.getString(4)+"</td>");
            out.print("<td>"+rs.getString(5)+"</td>");
             out.print("<td>"+rs.getString(6)+"</td>");
             out.print("<td>"+rs.getString(7)+"</td>");
              out.print("<td>"+rs.getString(8)+"</td>");
               out.print("<td>"+rs.getString(9)+"</td>");
                out.print("<td>"+rs.getString(10)+"</td>");
                 out.print("<td>"+rs.getString(12)+"</td>");
                  out.print("<td>"+rs.getString(13)+"</td>");
                   out.print("<td>"+rs.getString(14)+"</td>");
                    out.print("<td>"+rs.getString(15)+"</td>");
                     out.print("<td>"+rs.getString(16)+"</td>");
                      out.print("<td>"+rs.getString(11)+"</td>");
            out.print("</tr>");
//out.println("</H7>");
             }
} catch (Exception ex) {
                                 out.println("<html>");
                            out.println("<head><title>Action completed.</title>");
                            out.println("</head>");
                            out.println("<body>");
```

```
                    out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                    out.println("<P>");
                    out.println("<P>");
                    out.println("</body></html>");
                              }
  }
  return "y";
  }
  public String getAddHR(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
    response.setContentType("text/html");
 PrintWriter out = response.getWriter();
   if (request.getMethod().equals("POST")) {
  HiD=request.getParameter("HiD");      HospName=request.getParameter("HospName");
  Location=request.getParameter("Location");   WWW=request.getParameter("WWW");
  Hemail=request.getParameter("Hemail");      Hownership=request.getParameter("Hownership");
  DateEstab=request.getParameter("DateEstab");        HTel=request.getParameter("HTel");
  CMO=request.getParameter("CMO");    CMOQualif=request.getParameter("CMOQualif");
    HExpertise=request.getParameter("HExpertise");
 flag=false;
  try{
 prepStmt = conn.prepareStatement("select * from HR where HiD=?");
 prepStmt.setString(1,HiD);      rs=prepStmt.executeQuery();   flag=rs.next();
                    if (flag){
                      System.out.println("About to connec and update");
    prepStmt2 = conn.prepareStatement(
                     "update HR set HiD=?, HospName=?, Location=?, WWW=?, "+
                     "Hemail=?, Hownership=?, DateEstab=?, "+
                     "HTel=?, CMO=?, CMOQualif=?, HExpertise=? where HiD=?");
                      prepStmt2.setString(1,HiD);  prepStmt2.setString(2,HospName);
                        prepStmt2.setString(3,Location); prepStmt2.setString(4,WWW);
                      prepStmt2.setString(5,Hemail); prepStmt2.setString(6,Hownership);
                        prepStmt2.setString(7,DateEstab); prepStmt2.setString(8,HTel);
                            prepStmt2.setString(9,CMO); prepStmt2.setString(10,CMOQualif);
                            prepStmt2.setString(11,HExpertise);prepStmt2.setString(12,HiD);
                      prepStmt2.executeUpdate();
                      out.println("<html>");
                      out.println("<head><title>Action completed.</title>");
                      out.println("</head>");
                      out.println("<body>");
                      out.println("<H1 ALIGN=\"CENTER\">Record Update was successful</H1>");
                      out.println("<P>");
                      out.println("<P>");
                      out.println("</body></html>");
                    }
                    if (!flag){
                      System.out.println("About to connec and insert");
                      prepStmt2 = conn.prepareStatement("insert into HR values(?,?,?,?,?,?,?,?,?,?,?)");
                      prepStmt2.setString(1,HiD);  prepStmt2.setString(2,HospName);
                        prepStmt2.setString(3,Location); prepStmt2.setString(4,WWW);
                      prepStmt2.setString(5,Hemail); prepStmt2.setString(6,Hownership);
                        prepStmt2.setString(7,DateEstab); prepStmt2.setString(8,HTel);
                            prepStmt2.setString(9,CMO); prepStmt2.setString(10,CMOQualif);
                            prepStmt2.setString(11,HExpertise);
                              prepStmt2.executeUpdate();
                      out.println("<html>");
                      out.println("<head><title>Action completed.</title>");
                      out.println("</head>");
                      out.println("<body>");
                      out.println("<H1 ALIGN=\"CENTER\">Record Insertion was successful</H1>");
                      out.println("<P>");
                      out.println("<P>");
                      out.println("</body></html>");
```

```
                    }
                    } catch (Exception ex) {
                      out.println("<html>");
                    out.println("<head><title>Action completed.</title>");
                    out.println("</head>");
                    out.println("<body>");
                    out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                    out.println("<P>");
                    out.println("<P>");
                    out.println("</body></html>");
                            }
//  AddEMR();
  }
  return "y";
  }
  public String DatePurchase="";
  public String getDatePurchase(){
   return DatePurchase;
  }
  public String getAddMe(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
      response.setContentType("text/html");
  PrintWriter out = response.getWriter();
     if (request.getMethod().equals("POST")) {
    EquipID=request.getParameter("EquipID");     EquipType=request.getParameter("EquipType");
    QtyPurchase=Integer.parseInt(request.getParameter("QtyPurchase"));
    QtySold=Integer.parseInt(request.getParameter("QtySold"));
    QtyBal=Integer.parseInt(request.getParameter("QtyBal"));
    QtyStatus=request.getParameter("QtyStatus");
    Location=request.getParameter("Location");
    DatePurchase=request.getParameter("DatePurchase");
   flag=false;
    try{
   prepStmt = conn.prepareStatement("select * from Me where EquipID=?");
   prepStmt.setString(1,EquipID);     rs=prepStmt.executeQuery();   flag=rs.next();
                      if (flag){
                        int tempBal=rs.getInt(5) ;
                        System.out.println("About to connec and update");
     prepStmt2 = conn.prepareStatement(
                      "update Me set EquipID=?, EquipType=?, QtyPurchase=?, QtySold=?, "+
                      "QtyBal=?, QtyStatus=?, Location=?, "+
                      "DatePurchase=? where EquipID=?");
                       prepStmt2.setString(1,EquipID);  prepStmt2.setString(2,EquipType);
                         prepStmt2.setInt(3,QtyPurchase); prepStmt2.setInt(4,QtySold);
                       prepStmt2.setInt(5,(QtyPurchase+tempBal-QtySold));
                       prepStmt2.setString(6,QtyStatus);
                         prepStmt2.setString(7,Location); prepStmt2.setString(8,DatePurchase);
                             prepStmt2.setString(9,EquipID);
                       prepStmt2.executeUpdate();
                       out.println("<html>");
                       out.println("<head><title>Action completed.</title>");
                       out.println("</head>");
                       out.println("<body>");
                       out.println("<H1 ALIGN=\"CENTER\">Record Update was successful</H1>");
                       out.println("<P>");
                       out.println("<P>");
                       out.println("</body></html>");
                     }
                     if (!flag){
                       System.out.println("About to connec and insert");
                       prepStmt2 = conn.prepareStatement("insert into Me values(?,?,?,?,?,?,?,?)");
                       prepStmt2.setString(1,EquipID);  prepStmt2.setString(2,EquipType);
                             prepStmt2.setInt(3,QtyPurchase); prepStmt2.setInt(4,QtySold);
```

```
                              prepStmt2.setInt(5,(QtyPurchase-QtySold));
                              prepStmt2.setString(6,QtyStatus);
                                prepStmt2.setString(7,Location); prepStmt2.setString(8,DatePurchase);
                              prepStmt2.executeUpdate();
                          out.println("<html>");
                          out.println("<head><title>Action completed.</title>");
                          out.println("</head>");
                          out.println("<body>");
                          out.println("<H1 ALIGN=\"CENTER\">Record Insertion was successful</H1>");
                          out.println("<P>");
                          out.println("<P>");
                          out.println("</body></html>");
                        }
                          } catch (Exception ex) {
                            out.println("<html>");
                        out.println("<head><title>Action completed.</title>");
                        out.println("</head>");
                        out.println("<body>");
                        out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                        out.println("<P>");
                        out.println("<P>");
                        out.println("</body></html>");
                                }
//  AddEMR();
   }
   return "y";
   }
  public String getAddOPT(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
     response.setContentType("text/html");
  PrintWriter out = response.getWriter();
    if (request.getMethod().equals("POST")) {
   PiD=request.getParameter("PiD");   PName=request.getParameter("PName");
   SurgeryDate=request.getParameter("Expertise");
   DoctorIc=request.getParameter("Qualification");
   AfterSurgWard=request.getParameter("Location");   EvaluationOfSuccess=request.getParameter("ETel");
   Bill=Integer.parseInt(request.getParameter("Bill"));     Tools_Req=request.getParameter("ToolsReq");
       Recall=request.getParameter("Recall");
        SurgeryType=request.getParameter("SurgeryType");
  flag=false;
   try{
  prepStmt = conn.prepareStatement("select * from OPT where PiD=?");
                         prepStmt.setString(1,PiD);
                           rs=prepStmt.executeQuery();
                           flag=rs.next();
                         if (flag){
                           System.out.println("About to connec and update");
     prepStmt2 = conn.prepareStatement(
                         "update OPT set PiD=?, Pname=?, SurgeryDate=?, DoctorIc=?, "+
                         "AfterSurgWard=?, EvaluationOfSuccess=?, Bill=?, Recall=?, "+
                         "ToolsReq=?, SurgeryType=? where PiD=?");
                          prepStmt2.setString(1,PiD);
                          prepStmt2.setString(2,PName);
                            prepStmt2.setString(3,SurgeryDate);
                          prepStmt2.setString(4,DoctorIc);
                          prepStmt2.setString(5,AfterSurgWard);
                          prepStmt2.setString(6,EvaluationOfSuccess);
                            prepStmt2.setInt(7,Bill);
                              prepStmt2.setString(8,Recall);
                                prepStmt2.setString(9,Tools_Req);
                                prepStmt2.setString(10,SurgeryType);
                                prepStmt2.setString(11,PiD);
                          prepStmt2.executeUpdate();
```

```
                         out.println("<html>");
                         out.println("<head><title>Action completed.</title>");
                         out.println("</head>");
                         out.println("<body>");
                         out.println("<H1 ALIGN=\"CENTER\">Record Update was successful</H1>");
                         out.println("<P>");
                         out.println("<P>");
                         out.println("</body></html>");
                        }
                        if (!flag){
                         System.out.println("About to connec and insert");
                         prepStmt2 = conn.prepareStatement("insert into OPT values(?,?,?,?,?,?,?,?,?,?)");
                         prepStmt2.setString(1,PiD);
                          prepStmt2.setString(2,PName);
                               prepStmt2.setString(3,SurgeryType);
                           prepStmt2.setString(4,SurgeryDate);
                          prepStmt2.setString(5,DoctorIc);
                          prepStmt2.setString(6,AfterSurgWard);
                                prepStmt2.setInt(7,Bill);
                          prepStmt2.setString(8,EvaluationOfSuccess);
                            prepStmt2.setString(9,Recall);
                         prepStmt2.setString(10,Tools_Req);
                              prepStmt2.executeUpdate();
                         out.println("<html>");
                         out.println("<head><title>Action completed.</title>");
                         out.println("</head>");
                         out.println("<body>");
                         out.println("<H1 ALIGN=\"CENTER\">Record Insertion was successful</H1>");
                         out.println("<P>");
                         out.println("<P>");
                         out.println("</body></html>");
                        }
                         } catch (Exception ex) {
                          out.println("<html>");
                        out.println("<head><title>Action completed.</title>");
                        out.println("</head>");
                        out.println("<body>");
                        out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                        out.println("<P>");
                        out.println("<P>");
                        out.println("</body></html>");
                                 }
//  AddEMR();
    }
    return "y";
    }
   public String getDeleteEMR(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
      response.setContentType("text/html");
  PrintWriter out = response.getWriter();
  if (request.getMethod().equals("POST")) {
  PiD=request.getParameter("PiD");
 flag=false;
 try{
 prepStmt = conn.prepareStatement("select * from emr where PiD=?");
                       prepStmt.setString(1,PiD);
                        rs=prepStmt.executeQuery();
                        flag=rs.next();
                       if (flag){
                        System.out.println("About to connec and delete");
                        prepStmt2 = conn.prepareStatement(
                       "delete * from emr where PiD=?");
                        prepStmt2.setString(1,PiD);
```

```
                        prepStmt2.executeUpdate();
                        out.println("<html>");
                      out.println("<head><title>Action completed.</title>");
                      out.println("</head>");
                      out.println("<body>");
                      out.println("<H1 ALIGN=\"CENTER\">Record Deletion was successful</H1>");
                      out.println("<P>");
                      out.println("<P>");
                      out.println("</body></html>");
                        }
                      } catch (Exception ex) {
                        out.println("<html>");
                      out.println("<head><title>Action completed.</title>");
                      out.println("</head>");
                      out.println("<body>");
                      out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                      out.println("<P>");
                      out.println("<P>");
                      out.println("</body></html>");
                    }
  }
  return "y";
  }
  public String getDeleteCDS(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
      response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    if (request.getMethod().equals("POST")) {
    PiD=request.getParameter("PiD");
   flag=false;
   try{
  prepStmt = conn.prepareStatement("select * from CDS where PiD=?");
                        prepStmt.setString(1,PiD);
                          rs=prepStmt.executeQuery();
                          flag=rs.next();
                        if (flag){
                          System.out.println("About to connec and delete");
                          prepStmt2 = conn.prepareStatement(
                        "delete * from CDS where PiD=?");
                         prepStmt2.setString(1,PiD);
                         prepStmt2.executeUpdate();
                         out.println("<html>");
                      out.println("<head><title>Action completed.</title>");
                      out.println("</head>");
                      out.println("<body>");
                      out.println("<H1 ALIGN=\"CENTER\">Record Deletion was successful</H1>");
                      out.println("<P>");
                      out.println("<P>");
                      out.println("</body></html>");
                        }
                      } catch (Exception ex) {
                        out.println("<html>");
                      out.println("<head><title>Action completed.</title>");
                      out.println("</head>");
                      out.println("<body>");
                      out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                      out.println("<P>");
                      out.println("<P>");
                      out.println("</body></html>");
                    }
   }
   return "y";
   }
```

```
   public String getDeleteDrug(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
        response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      if (request.getMethod().equals("POST")) {
      DrugID=request.getParameter("DrugID");
     flag=false;
     try{
    prepStmt = conn.prepareStatement("select * from drug where DrugID=?");
                              prepStmt.setString(1,DrugID);
                                rs=prepStmt.executeQuery();
                                flag=rs.next();
                              if (flag){
                                System.out.println("About to connec and delete");
                                prepStmt2 = conn.prepareStatement(
                              "delete * from drug where DrugID=?");
                               prepStmt2.setString(1,DrugID);
                               prepStmt2.executeUpdate();
                               out.println("<html>");
                           out.println("<head><title>Action completed.</title>");
                           out.println("</head>");
                           out.println("<body>");
                           out.println("<H1 ALIGN=\"CENTER\">Record Deletion was successful</H1>");
                           out.println("<P>");
                           out.println("<P>");
                           out.println("</body></html>");
                             }
                           } catch (Exception ex) {
                             out.println("<html>");
                           out.println("<head><title>Action completed.</title>");
                           out.println("</head>");
                           out.println("<body>");
                           out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                           out.println("<P>");
                           out.println("<P>");
                           out.println("</body></html>");
                       }
     }
     return "y";
     }
  public String getDeleteEC(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
       response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    if (request.getMethod().equals("POST")) {
    ExpertID=request.getParameter("ExpertID");
   flag=false;
   try{
  prepStmt = conn.prepareStatement("select * from ec where ExpertID=?");
                           prepStmt.setString(1,ExpertID);
                             rs=prepStmt.executeQuery();
                             flag=rs.next();
                           if (flag){
                             System.out.println("About to connec and delete");
                             prepStmt2 = conn.prepareStatement(
                           "delete * from ec where ExpertID=?");
                            prepStmt2.setString(1,ExpertID);
                            prepStmt2.executeUpdate();
                            out.println("<html>");
                        out.println("<head><title>Action completed.</title>");
                        out.println("</head>");
                        out.println("<body>");
                        out.println("<H1 ALIGN=\"CENTER\">Record Deletion was successful</H1>");
```

```
                    out.println("<P>");
                    out.println("<P>");
                    out.println("</body></html>");
                      }
                    } catch (Exception ex) {
                      out.println("<html>");
                    out.println("<head><title>Action completed.</title>");
                    out.println("</head>");
                    out.println("<body>");
                    out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                    out.println("<P>");
                    out.println("<P>");
                    out.println("</body></html>");
                  }
   }
   return "y";
   }
   public String getDeleteMe(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
       response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    if (request.getMethod().equals("POST")) {
    EquipID=request.getParameter("EquipID");
   flag=false;
   try{
  prepStmt = conn.prepareStatement("select * from Me where EquipID=?");
                      prepStmt.setString(1,EquipID);
                        rs=prepStmt.executeQuery();
                        flag=rs.next();
                      if (flag){
                        System.out.println("About to connec and delete");
                        prepStmt2 = conn.prepareStatement(
                      "delete * from Me where EquipID=?");
                       prepStmt2.setString(1,EquipID);
                       prepStmt2.executeUpdate();
                       out.println("<html>");
                    out.println("<head><title>Action completed.</title>");
                    out.println("</head>");
                    out.println("<body>");
                    out.println("<H1 ALIGN=\"CENTER\">Record Deletion was successful</H1>");
                    out.println("<P>");
                    out.println("<P>");
                    out.println("</body></html>");
                      }
                    } catch (Exception ex) {
                      out.println("<html>");
                    out.println("<head><title>Action completed.</title>");
                    out.println("</head>");
                    out.println("<body>");
                    out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                    out.println("<P>");
                    out.println("<P>");
                    out.println("</body></html>");
                  }
   }
   return "y";
   }
   public String getDeleteHR(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
       response.setContentType("text/html");
    PrintWriter out = response.getWriter();
```

```
 if (request.getMethod().equals("POST")) {
 HiD=request.getParameter("HiD");
flag=false;
try{
prepStmt = conn.prepareStatement("select * from HR where HiD=?");
                    prepStmt.setString(1,HiD);
                      rs=prepStmt.executeQuery();
                      flag=rs.next();
                    if (flag){
                      System.out.println("About to connec and delete");
                      prepStmt2 = conn.prepareStatement(
                    "delete * from HR where HiD=?");
                     prepStmt2.setString(1,HiD);
                     prepStmt2.executeUpdate();
                     out.println("<html>");
                  out.println("<head><title>Action completed.</title>");
                  out.println("</head>");
                  out.println("<body>");
                  out.println("<H1 ALIGN=\"CENTER\">Record Deletion was successful</H1>");
                  out.println("<P>");
                  out.println("<P>");
                  out.println("</body></html>");

                    }
                  } catch (Exception ex) {
                    out.println("<html>");
                  out.println("<head><title>Action completed.</title>");
                  out.println("</head>");
                  out.println("<body>");
                  out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                  out.println("<P>");
                  out.println("<P>");
                  out.println("</body></html>");
                }
}
return "y";
}
public String getDeleteStaff(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
        response.setContentType("text/html");
     PrintWriter out = response.getWriter();
     if (request.getMethod().equals("POST")) {
     StaffID=request.getParameter("StaffID");
    flag=false;
    try{
   prepStmt = conn.prepareStatement("select * from hospadmin where StaffID=?");
                    prepStmt.setString(1,StaffID);
                      rs=prepStmt.executeQuery();
                      flag=rs.next();
                    if (flag){
                      System.out.println("About to connec and delete");
                      prepStmt2 = conn.prepareStatement(
                    "delete * from hospadmin where StaffID=?");
                     prepStmt2.setString(1,StaffID);
                     prepStmt2.executeUpdate();
                     out.println("<html>");
                  out.println("<head><title>Action completed.</title>");
                  out.println("</head>");
                  out.println("<body>");
                  out.println("<H1 ALIGN=\"CENTER\">Staff Deletion was successful</H1>");
                  out.println("<P>");
                  out.println("<P>");
                  out.println("</body></html>");
```

```
                }
              } catch (Exception ex) {
                out.println("<html>");
              out.println("<head><title>Action completed.</title>");
              out.println("</head>");
              out.println("<body>");
              out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
              out.println("<P>");
              out.println("<P>");
              out.println("</body></html>");
            }
    }
    return "y";
    }
    public String getDeleteOPT(HttpServletRequest request,HttpServletResponse response) throws ServletException,
IOException {
          response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        if (request.getMethod().equals("POST")) {
        PiD=request.getParameter("PiD");
        flag=false;
        try{
       prepStmt = conn.prepareStatement("select * from OPT where PiD=?");
                  prepStmt.setString(1,PiD);
                    rs=prepStmt.executeQuery();
                    flag=rs.next();
                  if (flag){
                    System.out.println("About to connec and delete");
                    prepStmt2 = conn.prepareStatement(
                  "delete * from OPT where PiD=?");
                   prepStmt2.setString(1,PiD);
                   prepStmt2.executeUpdate();
                   out.println("<html>");
                out.println("<head><title>Action completed.</title>");
                out.println("</head>");
                out.println("<body>");
                out.println("<H1 ALIGN=\"CENTER\">Patient with ID "+PiD+" informtin Deletion was
successful</H1>");
                out.println("<P>");
                out.println("<P>");
                out.println("</body></html>");
                  }
                } catch (Exception ex) {
                  out.println("<html>");
                out.println("<head><title>Action completed.</title>");
                out.println("</head>");
                out.println("<body>");
                out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                out.println("<P>");
                out.println("<P>");
                out.println("</body></html>");
              }
        }
        return "y";
        }
        public String getDeletePatient(HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {
          response.setContentType("text/html");
       PrintWriter out = response.getWriter();
       if (request.getMethod().equals("POST")) {
       PiD=request.getParameter("PiD");
      flag=false;
      try{
```

```
prepStmt = conn.prepareStatement("select * from ipd where PiD=?");
                    prepStmt.setString(1,PiD);
                     rs=prepStmt.executeQuery();
                     flag=rs.next();
                    if (flag){
                     System.out.println("About to connec and delete");
                     prepStmt2 = conn.prepareStatement(
                    "delete * from ipd where PiD=?");
                     prepStmt2.setString(1,PiD);
                     prepStmt2.executeUpdate();
                     out.println("<html>");
                  out.println("<head><title>Action completed.</title>");
                  out.println("</head>");
                  out.println("<body>");
                  out.println("<H1 ALIGN=\"CENTER\">Patient with ID "+PiD+" informtin Deletion was
successful</H1>");
                  out.println("<P>");
                  out.println("<P>");
                  out.println("</body></html>");
                    }
                  } catch (Exception ex) {
                   out.println("<html>");
                  out.println("<head><title>Action completed.</title>");
                  out.println("</head>");
                  out.println("<body>");
                  out.println("<H1 ALIGN=\"CENTER\">"+ex.getMessage()+"</H1>");
                  out.println("<P>");
                  out.println("<P>");
                  out.println("</body></html>");
                }
 }
 return "y";
 }
 public String  getCDS(HttpServletRequest request, HttpServletResponse
      response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Hospital CDS Information.</title>");
    out.println("<style type=\"text/css\">");
    out.println("<!--");
    out.println(".style2 {font-size: 11px}");
    out.println("-->");
    out.println("</style>");
    out.println("</head>");
    out.println("<body>");
    out.println("<P>");
   out.print("<form name=\"form1\" method=\"post\">");
   out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
   out.println("<H1 ALIGN=\"CENTER\">Remotes Medical Reports from CDS/Sensor</H1>");
   out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
    out.print("<table width=\"731\" border=\"1\">");
    try {
       prepStmt = conn.prepareStatement("select * from CDS");
       rs=prepStmt.executeQuery();
       out.print("<tr>");
        out.print("<td width=\"44\"><span class=\"style2\">Patint's ID </span></td>");
        out.print("<td width=\"111\"><span class=\"style2\">Blood Test</span></td>");
         out.print("<td width=\"44\"><span class=\"style2\">Urine Test </span></td>");
         out.print("<td width=\"61\"><span class=\"style2\">Cardiology </span></td>");
          out.print("<td width=\"63\"><span class=\"style2\">Blood Pressure </span></td>");
           out.print("<td width=\"68\"><span class=\"style2\">Blood Sugar</span></td>");
```

```
out.print("<td width=\"57\"><span class=\"style2\">Temp</span></td>");
out.print("<td width=\"69\"><span class=\"style2\">HIV</span></td>");
out.print("<td width=\"90\"><span class=\"style2\">Tuberculosis</span></td>");
out.print("<td width=\"60\"><span class=\"style2\">Cholera </span></td>");
out.print("<td width=\"60\"><span class=\"style2\">Other Test</span></td>");
out.print("<td width=\"90\"><span class=\"style2\">Date of Analysis</span></td>");
out.print("<td width=\"60\"><span class=\"style2\">Remarks </span></td>");
out.print("</tr>");
while( rs.next() ){
out.print("<tr>");
out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(4)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(6)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(7)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(9)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(10)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(11)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(17)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(19)+"</span></td>");
out.print("<td><span class=\"style2\">"+rs.getString(18)+"</span></td>");
out.print("</tr>");
}
out.print("</table>");
out.print("</span>");
out.print("<TR><TD> <div align=\"center\">");
out.print("<INPUT TYPE=\"button\" NAME=\"Addnew\" value=\"Add/Edit Medical CDS\" onclick=\"LoadMe()\">");
out.print("<INPUT TYPE=\"button\" NAME=\"DeleteRec\" value=\"Delete Record\" onclick=\"DeleteMe()\">");
out.println("<script language='JavaScript' type='text/JavaScript'>");
out.println("function LoadMe(){");
//    out.println("window.open('CDSPage.jsp');");
out.println("location.href='CDSPage.jsp';");
out.println("};");
out.println("</script>");
out.println("<script language='JavaScript' type='text/JavaScript'>");
out.println("function DeleteMe(){");
// out.println("window.open('DeleteCDS.jsp');");
out.println("location.href='DeleteCDS.jsp';");
out.println("};");
out.println("</script>");
out.print("</div></TD>");
out.print("</form>");
} catch (Exception ex) {
    System.out.println(ex.getMessage());
out.println("System resources is temporally out of service.");
}
out.println("<P>");
out.println("</body></html>");
return "Thanks for your visit";
}
public String  getIN(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>Hospital In-Patient's Information.</title>");
        out.println("<style type=\"text/css\">");
        out.println("<!--");
        out.println(".style2 {font-size: 11px}");
        out.println("-->");
        out.println("</style>");
```

```
                    out.println("</head>");
                    out.println("<body>");
                    out.println("<P>");
                  out.print("<form name=\"form1\" method=\"post\">");
                  out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
                  out.println("<H1 ALIGN=\"CENTER\">Patient Admission Master List</H1>");
                   out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
                    out.print("<table width=\"731\" border=\"1\">");
                    try {
                       prepStmt = conn.prepareStatement("select * from ipd where DischargeStatus='NO'");
                       rs=prepStmt.executeQuery();
                       out.print("<tr>");
                        out.print("<td width=\"44\"><span class=\"style2\">Hospital ID </span></td>");
                        out.print("<td width=\"111\"><span class=\"style2\">Patient Name</span></td>");
                          out.print("<td width=\"111\"><span class=\"style2\">Patient Address</span></td>");
                         out.print("<td width=\"44\"><span class=\"style2\">Tel </span></td>");
                         out.print("<td width=\"61\"><span class=\"style2\">Email </span></td>");
                          out.print("<td width=\"63\"><span class=\"style2\">Age </span></td>");
                           out.print("<td width=\"68\"><span class=\"style2\">Gender</span></td>");
                           out.print("<td width=\"57\"><span class=\"style2\">Date Admin</span></td>");
                         out.print("<td width=\"69\"><span class=\"style2\">Diagnosis</span></td>");
                         out.print("<td width=\"90\"><span class=\"style2\">Ward</span></td>");
                         out.print("<td width=\"60\"><span class=\"style2\">Bed No</span></td>");
                         out.print("<td width=\"60\"><span class=\"style2\">Doctor IN Charge</span></td>");
                          out.print("</tr>");
                          while( rs.next() ){
                         out.print("<tr>");
                         out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
                         out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
                         out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
                          out.print("<td><span class=\"style2\">"+rs.getString(4)+"</span></td>");
                          out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
                          out.print("<td><span class=\"style2\">"+rs.getString(6)+"</span></td>");
                           out.print("<td><span class=\"style2\">"+rs.getString(8)+"</span></td>");
                          out.print("<td><span class=\"style2\">"+rs.getString(7)+"</span></td>");
                           out.print("<td><span class=\"style2\">"+rs.getString(10)+"</span></td>");
                           out.print("<td><span class=\"style2\">"+rs.getString(11)+"</span></td>");
                           out.print("<td><span class=\"style2\">"+rs.getString(9)+"</span></td>");
                           out.print("<td><span class=\"style2\">"+rs.getString(12)+"</span></td>");
                           out.print("</tr>");
                          }
                               out.print("</table>");
                          out.print("</span>");
                     out.print("<TR><TD> <div align=\"center\">");
                   out.print("<INPUT TYPE=\"button\" NAME=\"Addnew\" value=\"Add/Edit Patient\" onclick=\"LoadMe()\">");
                    out.print("<INPUT TYPE=\"button\" NAME=\"DeleteRec\" value=\"Delete Record\"
onclick=\"DeleteMe()\">");
                out.println("<script language='JavaScript' type='text/JavaScript'>");
             out.println("function LoadMe(){");
               //  out.println("window.open('PatientPage.jsp');");
             out.println("location.href='PatientPage.jsp';");
             out.println("};");
              out.println("</script>");
             out.println("<script language='JavaScript' type='text/JavaScript'>");
             out.println("function DeleteMe(){");
           //  out.println("window.open('DeletePatient.jsp');");
            out.println("location.href='DeletePatient.jsp';");
            out.println("};");
             out.println("</script>");
             out.print("</div></TD>");
              out.print("</form>");
                          } catch (Exception ex) {
```

```
                System.out.println(ex.getMessage());
            out.println("System resources is temporally out of service.");
              }
        out.println("<P>");
        out.println("</body></html>");
        return "Thanks for your visit";
         }
         public String  getOutPatient(HttpServletRequest request, HttpServletResponse
                                        response) throws ServletException, IOException {
                                       response.setContentType("text/html");
                                       PrintWriter out = response.getWriter();
                                       out.println("<html>");
                                       out.println("<head><title>Hospital Out-Patient's Information.</title>");
                                       out.println("<style type=\"text/css\">");
                                       out.println("<!--");
                                       out.println(".style2 {font-size: 11px}");
                                       out.println("-->");
                                       out.println("</style>");
                                       out.println("</head>");
                                       out.println("<body>");
                                       out.println("<P>");
                                     out.print("<form name=\"form1\" method=\"post\">");
                                     out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
                                     out.println("<H1 ALIGN=\"CENTER\">Hospital Out-Patient Master List</H1>");
                                      out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
                                      out.print("<table width=\"731\" border=\"1\">");
                                       try {
                                          prepStmt = conn.prepareStatement("select * from ipd where pStatus='Not
Admitted'");
                                          rs=prepStmt.executeQuery();
                                          out.print("<tr>");
                                           out.print("<td width=\"44\"><span class=\"style2\">Hospital ID
</span></td>");
                                           out.print("<td width=\"111\"><span class=\"style2\">Patient
Name</span></td>");
                                            out.print("<td width=\"111\"><span class=\"style2\">Patient
Address</span></td>");
                                           out.print("<td width=\"44\"><span class=\"style2\">Tel </span></td>");
                                           out.print("<td width=\"61\"><span class=\"style2\">Email </span></td>");
                                            out.print("<td width=\"63\"><span class=\"style2\">Age </span></td>");
                                             out.print("<td width=\"68\"><span class=\"style2\">Gender</span></td>");
                                             out.print("<td width=\"57\"><span class=\"style2\">Date
Admin</span></td>");
                                           out.print("<td width=\"69\"><span class=\"style2\">Diagnosis</span></td>");
                                           out.print("<td width=\"90\"><span class=\"style2\">Ward</span></td>");
                                           out.print("<td width=\"60\"><span class=\"style2\">Bed No</span></td>");
                                           out.print("<td width=\"60\"><span class=\"style2\">Doctor IN
Charge</span></td>");
                                            out.print("</tr>");
                                            while( rs.next() ){
                                           out.print("<tr>");
                                           out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
                                           out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
                                           out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
                                            out.print("<td><span class=\"style2\">"+rs.getString(4)+"</span></td>");
                                            out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
                                            out.print("<td><span class=\"style2\">"+rs.getString(6)+"</span></td>");
                                             out.print("<td><span class=\"style2\">"+rs.getString(8)+"</span></td>");
                                            out.print("<td><span class=\"style2\">"+rs.getString(7)+"</span></td>");
                                             out.print("<td><span class=\"style2\">"+rs.getString(10)+"</span></td>");
                                             out.print("<td><span class=\"style2\">"+rs.getString(11)+"</span></td>");
                                             out.print("<td><span class=\"style2\">"+rs.getString(9)+"</span></td>");
```

```
                                                  out.print("<td><span class=\"style2\">"+rs.getString(12)+"</span></td>");
                                                  out.print("</tr>");
                                                  }
                                                      out.print("</table>");
                                                  out.print("</span>");
                                        out.print("</form>");
                                                  } catch (Exception ex) {
                                                     System.out.println(ex.getMessage());
                                                out.println("System resources is temporally out of service.");
                                                  }
                                             out.println("<P>");
                                             out.println("</body></html>");
                                             return "Thanks for your visit";
                                              }
                    public String  getINPatient(HttpServletRequest request, HttpServletResponse
                                  response) throws ServletException, IOException {
                                 response.setContentType("text/html");
                                 PrintWriter out = response.getWriter();
                                 out.println("<html>");
                                 out.println("<head><title>Hospital Admission Information.</title>");
                                 out.println("<style type=\"text/css\">");
                                 out.println("<!--");
                                 out.println(".style2 {font-size: 11px}");
                                 out.println("-->");
                                 out.println("</style>");
                                 out.println("</head>");
                                 out.println("<body>");
                                 out.println("<P>");
                               out.print("<form name=\"form1\" method=\"post\">");
                               out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
                               out.println("<H1 ALIGN=\"CENTER\">Hospital (In-Patient) Admissions  Master List</H1>");
                                out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
                                out.print("<table width=\"731\" border=\"1\">");
                                 try {
                                    prepStmt = conn.prepareStatement("select * from ipd where pStatus='Admitted'");
                                    rs=prepStmt.executeQuery();
                                    out.print("<tr>");
                                     out.print("<td width=\"44\"><span class=\"style2\">Hospital ID </span></td>");
                                     out.print("<td width=\"111\"><span class=\"style2\">Patient Name</span></td>");
                                       out.print("<td width=\"111\"><span class=\"style2\">Patient Address</span></td>");
                                      out.print("<td width=\"44\"><span class=\"style2\">Tel </span></td>");
                                      out.print("<td width=\"61\"><span class=\"style2\">Email </span></td>");
                                       out.print("<td width=\"63\"><span class=\"style2\">Age </span></td>");
                                        out.print("<td width=\"68\"><span class=\"style2\">Gender</span></td>");
                                        out.print("<td width=\"57\"><span class=\"style2\">Date Admin</span></td>");
                                      out.print("<td width=\"69\"><span class=\"style2\">Diagnosis</span></td>");
                                      out.print("<td width=\"90\"><span class=\"style2\">Ward</span></td>");
                                      out.print("<td width=\"60\"><span class=\"style2\">Bed No</span></td>");
                                      out.print("<td width=\"60\"><span class=\"style2\">Doctor IN Charge</span></td>");
                                       out.print("</tr>");
                                       while( rs.next() ){
                                      out.print("<tr>");
                                      out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
                                      out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
                                      out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
                                       out.print("<td><span class=\"style2\">"+rs.getString(4)+"</span></td>");
                                       out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
                                       out.print("<td><span class=\"style2\">"+rs.getString(6)+"</span></td>");
                                        out.print("<td><span class=\"style2\">"+rs.getString(8)+"</span></td>");
                                       out.print("<td><span class=\"style2\">"+rs.getString(7)+"</span></td>");
                                        out.print("<td><span class=\"style2\">"+rs.getString(10)+"</span></td>");
                                        out.print("<td><span class=\"style2\">"+rs.getString(11)+"</span></td>");
```

```
                                                out.print("<td><span class=\"style2\">"+rs.getString(9)+"</span></td>");
                                                out.print("<td><span class=\"style2\">"+rs.getString(12)+"</span></td>");
                                                out.print("</tr>");
                                               }
                                                    out.print("</table>");
                                               out.print("</span>");
                                      out.print("</form>");
                                               } catch (Exception ex) {
                                                  System.out.println(ex.getMessage());
                                              out.println("System resources is temporally out of service.");
                                               }
                                          out.println("<P>");
                                          out.println("</body></html>");
                                          return "Thanks for your visit";
                                           }
                                           public String  getDischargedPatient(HttpServletRequest request, HttpServletResponse
                                                                 response) throws ServletException, IOException {
                                                                response.setContentType("text/html");
                                                                PrintWriter out = response.getWriter();
                                                                out.println("<html>");
                                                                out.println("<head><title>Hospital Discharged.</title>");
                                                                out.println("<style type=\"text/css\">");
                                                                out.println("<!--");
                                                                out.println(".style2 {font-size: 11px}");
                                                                out.println("-->");
                                                                out.println("</style>");
                                                                out.println("</head>");
                                                                out.println("<body>");
                                                                out.println("<P>");
                                                              out.print("<form name=\"form1\" method=\"post\">");
out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
                                                              out.println("<H1 ALIGN=\"CENTER\">Hospital Discharge
Reports</H1>");
                                                               out.println("<H2 ALIGN=\"CENTER\">as at :"+
getSystemDate()+"</H2>");
                                                                out.print("<table width=\"731\" border=\"1\">");
                                                                try {
                                                                    prepStmt = conn.prepareStatement("select * from ipd where
DisChargeStatus='YES'");

                                                                    rs=prepStmt.executeQuery();
                                                                    out.print("<tr>");
                                                                     out.print("<td width=\"44\"><span class=\"style2\">Hospital ID
</span></td>");
                                                                     out.print("<td width=\"111\"><span class=\"style2\">Patient
Name</span></td>");
                                                                      out.print("<td width=\"111\"><span class=\"style2\">Patient
Address</span></td>");
                                                                     out.print("<td width=\"44\"><span class=\"style2\">Tel
</span></td>");
                                                                     out.print("<td width=\"61\"><span class=\"style2\">Email
</span></td>");
                                                                      out.print("<td width=\"63\"><span class=\"style2\">Age
</span></td>");
                                                                       out.print("<td width=\"68\"><span
class=\"style2\">Gender</span></td>");
                                                                       out.print("<td width=\"57\"><span class=\"style2\">Date
Admin</span></td>");
                                                                     out.print("<td width=\"69\"><span
class=\"style2\">Diagnosis</span></td>");
                                                                     out.print("<td width=\"90\"><span
class=\"style2\">Ward</span></td>");
```

```
out.print("<td width=\"60\"><span class=\"style2\">Bed
No</span></td>");
out.print("<td width=\"60\"><span class=\"style2\">Doctor IN
Charge</span></td>");
out.print("<td width=\"60\"><span class=\"style2\">Discharged
Date</span></td>");
out.print("<td width=\"60\"><span class=\"style2\">Recall
Date</span></td>");
out.print("<td width=\"60\"><span class=\"style2\">Bill
Outstanding</span></td>");
out.print("</tr>");
while( rs.next() ){
out.print("<tr>");
out.print("<td><span
class=\"style2\">"+rs.getString(1)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(2)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(3)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(4)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(5)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(6)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(8)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(7)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(10)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(11)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(9)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(12)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(22)+"</span></td>");
out.print("<td><span
class=\"style2\">"+rs.getString(25)+"</span></td>");
out.print("<td><span
class=\"style2\">"+(Integer.parseInt(rs.getString(20))- Integer.parseInt(rs.getString(21)))+"</span></td>");
out.print("</tr>");
}
    out.print("</table>");
out.print("</span>");
out.print("</form>");
} catch (Exception ex) {
   System.out.println(ex.getMessage());
out.println("System resources is temporally out of service.");
}
out.println("<P>");
out.println("</body></html>");
return "Thanks for your visit";
}
private void loadDriver() {
    try {
     Class.forName(driver).newInstance();
       System.out.println("Loaded the appropriate driver");
    } catch (ClassNotFoundException cnfe) {
       System.err.println("\nUnable to load the JDBC driver " + driver);
       System.err.println("Please check your CLASSPATH.");
```

```
        //  cnfe.printStackTrace(System.err);
       } catch (InstantiationException ie) {
          System.err.println(
                   "\nUnable to instantiate the JDBC driver " + driver);
          ie.printStackTrace(System.err);
       } catch (IllegalAccessException iae) {
          System.err.println(
                   "\nNot allowed to access the JDBC driver " + driver);
          iae.printStackTrace(System.err);
       }
    }
public void CallInsertBookLog(HttpServletRequest request, HttpServletResponse
     response){
   int i=0;      ResultSet rs=null;    int b=0;    boolean flag=false;Connection conn = null;
try {
  Class.forName(driver).newInstance(); conn = DriverManager.getConnection(protocol);
 System.out.println(conn);
 System.out.println("Connection was successful");
 PreparedStatement prepStmt = conn.prepareStatement("select * from BookedIPLogTB where HiD=?");
 prepStmt.setString(1,HiD);
 rs=prepStmt.executeQuery();
 flag=rs.next();
 if(!flag){
 try {
  System.out.println("About to Insert a Record");
 prepStmt = conn.prepareStatement("insert into BookedIPLogTB values(?,?,?,?,?,?)");
 prepStmt.setString(1,HiD);
 prepStmt.setInt(2,trycount);
 prepStmt.setString(3,"Yes");
 prepStmt.setString(4,systemDate);
 prepStmt.setString(5,userName);
 prepStmt.setString(6,userPassWord);
 prepStmt.executeUpdate();
  System.out.println("Record Insertion was successful.");
 } catch (Exception ex) {
 System.out.print("Un able to Insert data."); }
 }else{
 try {
        System.out.println("Update Record");
      //PreparedStatement prepStmt = conn.prepareStatement("UPDATE GradeTable WHERE StudMatricNo=?");
     //  ("update phone set crd=crd+? where pno=?");
 prepStmt = conn.prepareStatement("Update BookedIPLogTB set NoofTry=?,Booked=? where HiD=?");
 prepStmt.setInt(1,trycount);
 prepStmt.setString(2,"Yes");
 prepStmt.setString(3,HiD);
 prepStmt.executeUpdate();
  System.out.println("Record Update was successful.");
 } catch (Exception ex) {
 System.out.print("Un able to Update data."); }
 }
  } catch (Exception ex) {
       System.out.println(ex.getMessage());
        System.out.println("System resources is temporally out of service.");
    }
}
  public void SearchInForID(HttpServletRequest request, HttpServletResponse
      response){
    int i=0;      ResultSet rs=null;    int b=0;    boolean flag=false;Connection conn = null;
 try {
   Class.forName(driver).newInstance(); conn = DriverManager.getConnection(protocol);
   System.out.println(conn);
   System.out.println("Connection was successful");
   PreparedStatement prepStmt = conn.prepareStatement("select * from ipd where HiD=? and pSat=?");
```

```
    prepStmt.setString(1,HiD);     prepStmt.setString(2,PatStatus);
    rs=prepStmt.executeQuery();
            flag=rs.next();
            if(flag){
              getFromDB();
              getInTablesss(request,response);
     //         }
           }else{
            getHiDS(request,response);
           // }
            }
      } catch (Exception ex) {
          System.out.println(ex.getMessage());
           System.out.println("System resources is temporally out of service.");
      }
   }
   public void SearchOPTForID(HttpServletRequest request, HttpServletResponse
      response){
     int i=0;      ResultSet rs=null;    int b=0;    boolean flag=false;Connection conn = null;
  try {
    Class.forName(driver).newInstance(); conn = DriverManager.getConnection(protocol);
    System.out.println(conn);
    System.out.println("Connection was successful");
    PreparedStatement prepStmt = conn.prepareStatement("select * from operation_theatre where PiD=?");
    prepStmt.setString(1,HiD);
    rs=prepStmt.executeQuery();
            flag=rs.next();
            if(flag){
              getOPDB();
              getOPTablesss(request,response);
     //         }
           }else{
            getHiDS(request,response);
           // }
            }
      } catch (Exception ex) {
          System.out.println(ex.getMessage());
           System.out.println("System resources is temporally out of service.");
      }
   }
 public PreparedStatement prepStmt=null;
   public PreparedStatement prepStmt2=null;
   public void SearchOutForID(HttpServletRequest request, HttpServletResponse
        response){
    try {
 prepStmt = conn.prepareStatement("select * from ipd where HiD=? and pSat=?");
      prepStmt.setString(1,HiD);     prepStmt.setString(2,PatStatus);
      rs=prepStmt.executeQuery();
              flag=rs.next();
              if(flag){
                getFromDB();
                getInTablesss(request,response);
       //         }
             }else{
              getHiDS(request,response);
             // }
              }
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
             System.out.println("System resources is temporally out of service.");
        }
     }
 public     int i=0;
```

```
    public int b=0;
    public boolean flag=false;
  public void SearchForID(HttpServletRequest request, HttpServletResponse
    response){
try {
 prepStmt = conn.prepareStatement("select * from ipd where HiD=? and pSat=?");
  prepStmt.setString(1,HiD);     prepStmt.setString(2,PatStatus);
  rs=prepStmt.executeQuery();
        flag=rs.next();
        if(flag){
          opt=1;
          PatStatus="Admitted";
          flag=true;
           getFromDB();
              getHiDS(request,response);
        }else{
         flag=false;
         PatStatus="Admitted";
        getHiDS(request,response);
        }
    } catch (Exception ex) {
       System.out.println(ex.getMessage());
        System.out.println("System resources is temporally out of service.");
    }
}
  public void getInTablesss(HttpServletRequest request, HttpServletResponse
      response) throws ServletException, IOException {
     response.setContentType("text/html");
     PrintWriter out = response.getWriter();
     if (PatStatus=="Admitted"){
        out.println("<html>");
         out.println("<head><title>A new Record ");
             out.println("</title></head>");
          out.println("<body>");
         out.println("<script language='JavaScript' type='text/JavaScript'>");
       out.println("location.href='InPatient.jsp';");
         out.println("</script>");
     out.println("</body>");
          out.println("</html>");
     }
       if (PatStatus!="Admitted") {
          out.println("<html>");
        out.println("<head><title>A new Record ");
            out.println("</title></head>");
         out.println("<body>");
        out.println("<script language='JavaScript' type='text/JavaScript'>");
      //   out.println("window.open('OutPatient.jsp');");
         out.println("location.href='PatientRegistrationOutPatient.jsp';");
        out.println("</script>");
     out.println("</body>");
         out.println("</html>");
        }
   }
   public void getOPTablesss(HttpServletRequest request, HttpServletResponse
         response) throws ServletException, IOException {
         response.setContentType("text/html");
         PrintWriter out = response.getWriter();
              out.println("<html>");
             out.println("<head><title>");
                 out.println("</title></head>");
              out.println("<body>");
             out.println("<script language='JavaScript' type='text/JavaScript'>");
           out.println("location.href='OperationPatient.jsp';");
```

```
            out.println("</script>");
          out.println("</body>");
             out.println("</html>");
          }
    public void getHiDS(HttpServletRequest request, HttpServletResponse
     response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    if (flag==true && opt==1){
   out.println("<html>");
    out.println("<head><title>Existing In Patient.");
       out.println("</title></head>");
     out.println("<body>");
    out.println("<script language='JavaScript' type='text/JavaScript'>");
  out.println("location.href='PatientRegistrationInPatient.jsp';");
    out.println("</script>");
out.println("</body>");
     out.println("</html>");
   }else if (flag==true && opt==2){
      out.println("<html>");
       out.println("<head><title>Exsting Out patient.");
          out.println("</title></head>");
        out.println("<body>");
       out.println("<script language='JavaScript' type='text/JavaScript'>");
     out.println("location.href='PatientRegistrationOutPatient.jsp';");
       out.println("</script>");
   out.println("</body>");
        out.println("</html>");
   }else {
     out.println("<html>");
   out.println("<head><title>A new Record ");
       out.println("</title></head>");
    out.println("<body>");
   out.println("<script language='JavaScript' type='text/JavaScript'>");
 //   out.println("window.open('PatientRegistrationOutPatient.jsp');");
    out.println("location.href='PatientRegistrationOutPatient.jsp';");
   out.println("</script>");
out.println("</body>");
    out.println("</html>");
   }
  }
  public String getAssociatedComm(HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head><title>Associated Communities Hospitals:.</title>");
out.println("</head>");
out.println("<body>");
out.println("<H1 ALIGN=\"CENTER\">Associated Communities Hospitals:</H1>");
out.println("<P>");
try {
    prepStmt = conn.prepareStatement("select * from associatecom");
    rs=prepStmt.executeQuery();
out.print("<table width=\"200\" border=\"1\" cellpadding=\"0\">");
              while( rs.next() ){
                out.print("<tr>");
             out.print("<td>"+rs.getString(1)+"</td>");
             String kk=rs.getString(2);
             out.println("<td><a href=\""+kk+"\">"+kk+"</a></td>");
             out.print(" </tr>");
             }
         out.print("</table>");
       } catch (Exception ex) {
```

```
        System.out.println(ex.getMessage());
    out.println("System resources is temporally out of service.");
      }
out.println("<P>");
out.println("</body></html>");
return "Thanks for your visit";
  }
 public String getAuthor(HttpServletResponse response) throws ServletException, IOException {
   response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head><title>About the author.</title>");
out.println("</head>");
out.println("<body>");
   out.println("<H1 ALIGN=\"CENTER\">AUTOMATED INTELLIGENT HOSPTITAL LTD.</H1>");
   out.println("<P>");
out.println("<H3 ALIGN=\"CENTER\">WAS DESIGNED</H3>");
out.println("<P>");
out.println("<H3 ALIGN=\"CENTER\">By</H3>");
out.println("<P>");
out.println("<H2 ALIGN=\"CENTER\">Oji Ifeoma Vivian</H2>");
out.println("<H2 ALIGN=\"CENTER\">Reg No. 2007517008P </H2>");
out.println("<P>");
out.println("<P>");
out.println("<P>");
out.println("<H5 ALIGN=\"CENTER\">Being A Ph.D. Project Submitted to</H5>");
out.println("<H4 ALIGN=\"CENTER\">School of Post Graduate Studies</H4>");
out.println("<H4 ALIGN=\"CENTER\">Nnamdi Azikiwe University, Akwa</H4>");
out.println("</body></html>");
return "Thanks for your visit";
  }
public void getFromDB(){
   int i=0;
     ResultSet rs=null;
   int b=0;
   boolean flag=false;
   Connection conn = null;
   try{
      prepStmt = conn.prepareStatement("select * from ipd where HiD=? and pSat=?");
         prepStmt.setString(1,HiD);     prepStmt.setString(2,PatStatus);
        rs=prepStmt.executeQuery();
        System.out.println("About to connec");
     while( rs.next() ){
     PName=rs.getString(1);PAddress=rs.getString(2);PTel=rs.getString(3);
     Pemail=rs.getString(4);ReferedFrom=rs.getString(6);HiD=rs.getString(7);
     Gender=rs.getString(8);BloodGp=rs.getString(9);HospAdmtd=rs.getString(10);
     HWard=rs.getString(11);BedNo=rs.getString(12);Diagnosis=rs.getString(13);
     Prescription=rs.getString(14);Treatment=rs.getString(15);Treatment=rs.getString(16);
     DocIc=rs.getString(17);Sponsor=rs.getString(18);STel=rs.getString(19);
     Semail=rs.getString(20);TBilltoDate=rs.getInt(21);TBillPdtoDate=rs.getInt(22);
     DischargeDate=rs.getString(23);PSat=rs.getString(24);
          }
          } catch (Exception ex) {
         System.out.print("No data base found");
   }
}

  public void getAncServicesDB(){
     int i=0;
       ResultSet rs=null;
     int b=0;
     boolean flag=false;
     Connection conn = null;
```

```
try{
  prepStmt = conn.prepareStatement("select * from ancService where PID=?");
      prepStmt.setString(1,PiD);
     rs=prepStmt.executeQuery();
     System.out.println("About to connec");
 while( rs.next() ){
 PiD=rs.getString(1);BloodTest=rs.getString(2);Urine=rs.getString(3);
 Cardiology=rs.getString(4);BloodPressure=rs.getString(5);BloodSugar=rs.getString(6);Temp=rs.getString(7);
 Malaria=rs.getString(8);HIV=rs.getString(9);Tuberculosis=rs.getString(10);
 Cholera=rs.getString(11);Hepatitis=rs.getString(12);Eye=rs.getString(13);
 Ear=rs.getString(14);Mouth=rs.getString(15);Stool=rs.getString(16);OtherTests=rs.getString(17);
          }
     } catch (Exception ex) {
     System.out.print("No data base found");
  }
}
public void getMEDB(){
   int i=0;
    ResultSet rs=null;
   int b=0;
   boolean flag=false;
   Connection conn = null;
   try{
     prepStmt = conn.prepareStatement("select * from ME where EquipID=?");
         prepStmt.setString(1,DrugType);
        rs=prepStmt.executeQuery();
        System.out.println("About to connec");
    while( rs.next() ){
    EquipID=rs.getString(1);EquipType=rs.getString(2);QtyPurchased=rs.getInt(3);
    QtySold=rs.getInt(4);QtyBal=rs.getInt(5);EquipStatus=rs.getString(6);
    Location=rs.getString(7);DatePurchased=rs.getString(8);
   }
        } catch (Exception ex) {
        System.out.print("No data base found");
   }
}

public String getEquipID(){
 return EquipID;
}
public int getQtyPurchased(){
 return QtyPurchased;
}
public int getQtySold(){
 return QtySold;
}
public int getQtyBal(){
 return QtyBal;
}
public String EquipID="";
public int QtyPurchased=0;
public int QtySold=0;
public int QtyBal=0;
public void getDrugDB(){
   int i=0;
    ResultSet rs=null;
   int b=0;
   boolean flag=false;
   Connection conn = null;
   try{
     prepStmt = conn.prepareStatement("select * from Drug where DrugType=?");
         prepStmt.setString(1,DrugType);
        rs=prepStmt.executeQuery();
```

```
            System.out.println("About to connec");
         while( rs.next() ){
         DrugType=rs.getString(1);QtyinStock=rs.getInt(2);ReorderLevel=rs.getInt(3);
         QtyIssuedOut=rs.getInt(4);BalStock=rs.getInt(5);Nairavalue=rs.getInt(6);
         RemarksID=rs.getString(7);
         DrugLocation=rs.getString(8);DatePurchased=rs.getString(9);
        }
              } catch (Exception ex) {
             System.out.print("No data base found");
        }
    }
    public int Nairavalue=0;
public int getNairavalue(){
return Nairavalue;
}
    public int BalStock=0;
public int getBalStock(){
return BalStock;
}
    public String DatePurchased="";
    public String getDatePurchased(){
  return DatePurchased;
}
public String DrugLocation="";
 public String getDrugLocation(){
   return DrugLocation;
 }
public String RemarksID="";
  public void getOPDB(){
      try{
            prepStmt = conn.prepareStatement("select * from operation_theatre where PiD=?");
            prepStmt.setString(1,HiD);     prepStmt.setString(2,PatStatus);
           rs=prepStmt.executeQuery();
           System.out.println("About to connec");
       while( rs.next() ){
       PName=rs.getString(1);HiD=rs.getString(2);
       SurgeryDate=rs.getString(3);DoctorIc=rs.getString(4);SurgeryType=rs.getString(5);
       Tools_Req=rs.getString(6);AfterSurgWard=rs.getString(7);Bill=rs.getInt(8);
       EvaluationOfSuccess=rs.getString(8);Recall=rs.getString(9);
            }
            } catch (Exception ex) {
           System.out.print("No data base found");
      }
 }
 public void getEMRDB(){
       try{
            prepStmt = conn.prepareStatement("select * from emrt where PiD=?");
            prepStmt.setString(1,PiD);
           rs=prepStmt.executeQuery();
           System.out.println("About to connec");
        while( rs.next() ){
        PName=rs.getString(1);PiD=rs.getString(2);
        History=rs.getString(3);SensorReportsPDA=rs.getString(4);RemarksfromCDS=rs.getString(5);
        }
             } catch (Exception ex) {
            System.out.print("No data base found");
       }
  }
  public static String getSystemDate() {
Date LogInDates=new Date();
  return LogInDates.toString() ;
  }
public String  getME(HttpServletRequest request, HttpServletResponse
```

```
     response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Hospital Equipments.</title>");
    out.println("<style type=\"text/css\">");
    out.println("<!--");
    out.println(".style2 {font-size: 11px}");
    out.println("-->");
    out.println("</style>");
    out.println("</head>");
    out.println("<body>");
    out.println("<P>");
 out.print("<form name=\"form1\" method=\"post\">");
 out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
 out.println("<H1 ALIGN=\"CENTER\">Hospital Medical Equipments Reports</H1>");
  out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
   out.print("<table width=\"731\" border=\"1\">");
    try {
        prepStmt = conn.prepareStatement("select * from Me");
        rs=prepStmt.executeQuery();
        out.print("<tr>");
              out.print("<td width=\"44\"><span class=\"style2\">Equip Id </span></td>");
         out.print("<td width=\"111\"><span class=\"style2\">Equip Type</span></td>");
          out.print("<td width=\"44\"><span class=\"style2\">Qty Purchase </span></td>");
          out.print("<td width=\"61\"><span class=\"style2\">Qty Sold</span></td>");
           out.print("<td width=\"63\"><span class=\"style2\">Qty Balance</span></td>");
            out.print("<td width=\"68\"><span class=\"style2\">Equip Status</span></td>");
            out.print("<td width=\"57\"><span class=\"style2\">Location</span></td>");
             out.print("</tr>");
           while( rs.next() ){
          out.print("<tr>");
         out.print("<tr>");
           out.print("<td>"+rs.getString(1)+"</td>");
          out.print("<td>"+rs.getString(2)+"</td>");
          out.print("<td>"+rs.getString(3)+"</td>");
           out.print("<td>"+rs.getString(4)+"</td>");
           out.print("<td>"+rs.getString(5)+"</td>");
           out.print("<td>"+rs.getString(6)+"</td>");
            out.print("<td>"+rs.getString(7)+"</td>");
            out.print("</tr>");
            }
                out.print("</span>");
                       out.print("</table>");
                       out.print("<div align=\"center\">");
                            out.print("<INPUT TYPE=\"button\" NAME=\"Addnew\" value=\"Add Equipment\"
onclick=\"LoadMe()\">");
                             out.print("<INPUT TYPE=\"button\" NAME=\"DeleteRec\" value=\"Delete Equipment\"
onclick=\"DeleteMe()\">");
                          out.println("<script language='JavaScript' type='text/JavaScript'>");
                       out.println("function LoadMe(){");
                       //    out.println("window.open('MePage.jsp');");
                       out.println("location.href='MePage.jsp';");
                       out.println("};");
                         out.println("</script>");
                       out.println("<script language='JavaScript' type='text/JavaScript'>");
                       out.println("function DeleteMe(){");
                    //    out.println("window.open('DeleteMe.jsp');");
                      out.println("location.href='DeleteMe.jsp';");
                      out.println("};");
                       out.println("</script>");
                       out.print("</div>");
```

```
out.print("</form>");
   } catch (Exception ex) {
      System.out.println(ex.getMessage());
 out.println("System resources is temporally out of service.");
   }
out.println("<P>");
out.println("</body></html>");
return "Thanks for your visit";
  }
  public String  getEMR(HttpServletRequest request, HttpServletResponse
     response) throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Hospital EMR/Clinical</title>");
    out.println("<style type=\"text/css\">");
    out.println("<!--");
    out.println(".style2 {font-size: 10px}");
    out.println("-->");
    out.println("</style>");
    out.println("</head>");
    out.println("<body>");
    out.println("<P>");
  out.print("<form name=\"form1\" method=\"post\">");
  out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
  out.println("<H1 ALIGN=\"CENTER\">Hospital EMR/Clinical Reports</H1>");
   out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
   out.print("<table width=\"731\" border=\"1\">");
    try {
        prepStmt = conn.prepareStatement("select * from EMR");
        rs=prepStmt.executeQuery();
        out.print("<tr>");
      //  out.println("<H7");
        out.print("<td width=\"50\"><span class=\"style2\">Patient ID</span></td>");
         out.print("<td width=\"20\"><span class=\"style2\">Date</span></td>");
          out.print("<td width=\"20\"><span class=\"style2\">Time</span></td>");
          out.print("<td width=\"20\"><span class=\"style2\">Sys</span></td>");
           out.print("<td width=\"30\"><span class=\"style2\">DIA</span></td>");
             out.print("<td width=\"20\"><span class=\"style2\">Irr. Heatbeat</span></td>");
              out.print("<td width=\"20\"><span class=\"style2\">Pulse</span></td>");
                out.print("<td width=\"20\"><span class=\"style2\">Excess Moves</span></td>");
                  out.print("<td width=\"10\"><span class=\"style2\">Mscnt1</span></td>");
                    out.print("<td width=\"10\"><span class=\"style2\">MscCnt2</span></td>");
                      out.print("<td width=\"10\"><span class=\"style2\">MsCnt3</span></td>");
                        out.print("<td width=\"20\"><span class=\"style2\">Input Typpe</span></td>");
                          out.print("<td width=\"20\"><span class=\"style2\">Meaure Type</span></td>");
                            out.print("<td width=\"70\"><span class=\"style2\">Memo</span></td>");
             out.print("</tr>");
           while( rs.next() ){
          out.print("<tr>");
           out.print("<tr>");
           out.print("<td>"+rs.getString(3)+"</td>");
          out.print("<td>"+rs.getString(4)+"</td>");
          out.print("<td>"+rs.getString(5)+"</td>");
           out.print("<td>"+rs.getString(6)+"</td>");
           out.print("<td>"+rs.getString(7)+"</td>");
            out.print("<td>"+rs.getString(8)+"</td>");
             out.print("<td>"+rs.getString(9)+"</td>");
              out.print("<td>"+rs.getString(10)+"</td>");
               out.print("<td>"+rs.getString(12)+"</td>");
                out.print("<td>"+rs.getString(13)+"</td>");
                 out.print("<td>"+rs.getString(14)+"</td>");
```

```
                    out.print("<td>"+rs.getString(15)+"</td>");
                     out.print("<td>"+rs.getString(16)+"</td>");
                      out.print("<td>"+rs.getString(11)+"</td>");
            out.print("</tr>");
//out.println("</H7>");
             }
                     out.print("</span>");
                           out.print("</table>");
            out.print(" <div align=\"center\">");
         out.print("<INPUT TYPE=\"button\" NAME=\"Addnew\" value=\"View Specific Patient ID\" onclick=\"LoadMe()\">");
        //  out.print("<INPUT TYPE=\"button\" NAME=\"DeleteRec\" value=\"Delete Record\" onclick=\"DeleteMe()\">");
       out.println("<script language='JavaScript' type='text/JavaScript'>");
   out.println("function LoadMe(){");
   //    out.println("window.open('EMRPage.jsp');");
  // out.println("location.href='EMRPage.jsp';");
  out.println("location.href='EMRPageView.jsp';");
   out.println("};");
     out.println("</script>");
   out.println("<script language='JavaScript' type='text/JavaScript'>");
   out.println("function DeleteMe(){");
  //  out.println("window.open('EMRDelete.jsp');");
  out.println("location.href='EMRDelete.jsp';");
  out.println("};");
   out.println("</script>");
   out.print("</div>");
          out.print("</form>");
             } catch (Exception ex) {
                 System.out.println(ex.getMessage());
           out.println("System resources is temporally out of service.");
             }
       out.println("<P>");
       out.println("</body></html>");
       return "Thanks for your visit";
         }
     public String  getEC(HttpServletRequest request, HttpServletResponse
     response) throws ServletException, IOException {
   response.setContentType("text/html");
   PrintWriter out = response.getWriter();
   out.println("<html>");
   out.println("<head><title>Medical Experts.</title>");
   out.println("<style type=\"text/css\">");
   out.println("<!--");
   out.println(".style2 {font-size: 11px}");
   out.println("-->");
   out.println("</style>");
   out.println("</head>");
   out.println("<body>");
   out.println("<P>");
 out.print("<form name=\"form1\" method=\"post\">");
 out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
 out.println("<H1 ALIGN=\"CENTER\">Medical experts Data Base Listing Reports</H1>");
  out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
  out.print("<table width=\"731\" border=\"1\">");
   try {
       prepStmt = conn.prepareStatement("select * from EC");
       rs=prepStmt.executeQuery();
       out.print("<tr>");
           out.print("<td width=\"44\"><span class=\"style2\">Expert ID</span></td>");
        out.print("<td width=\"111\"><span class=\"style2\">Expert Name</span></td>");
         out.print("<td width=\"44\"><span class=\"style2\">Expertise </span></td>");
         out.print("<td width=\"61\"><span class=\"style2\">Hospital Location </span></td>");
          out.print("<td width=\"63\"><span class=\"style2\">Qualification</span></td>");
```

```
         out.print("<td width=\"68\"><span class=\"style2\">Expertise Tel</span></td>");
         out.print("<td width=\"57\"><span class=\"style2\">Email</span></td>");
        out.print("<td width=\"69\"><span class=\"style2\">Expert Residence</span></td>");
        out.print("<td width=\"90\"><span class=\"style2\">Hospital Availability</span></td>");
         out.print("</tr>");
         while( rs.next() ){
        out.print("<tr>");
        out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
        out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
        out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
         out.print("<td><span class=\"style2\">"+rs.getString(4)+"</span></td>");
         out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
         out.print("<td><span class=\"style2\">"+rs.getString(6)+"</span></td>");
          out.print("<td><span class=\"style2\">"+rs.getString(7)+"</span></td>");
         out.print("<td><span class=\"style2\">"+rs.getString(8)+"</span></td>");
          out.print("<td><span class=\"style2\">"+rs.getString(9)+"</span></td>");
         out.print("</tr>");
         }
         out.print("</span>");
            out.print("</table>");
              out.print(" <div align=\"center\">");
            out.print("<INPUT TYPE=\"button\" NAME=\"Addnew\" value=\"Add new Record\" onclick=\"LoadMe()\">");
             out.print("<INPUT TYPE=\"button\" NAME=\"DeleteRec\" value=\"Delete Record\" onclick=\"DeleteMe()\">");
         out.println("<script language='JavaScript' type='text/JavaScript'>");
     out.println("function LoadMe(){");
    //     out.println("window.open('ECPage.jsp');");
     out.println("location.href='ECPage.jsp';");
     out.println("};");
       out.println("</script>");
     out.println("<script language='JavaScript' type='text/JavaScript'>");
     out.println("function DeleteMe(){");
   //  out.println("window.open('ECDelete.jsp');");
    out.println("location.href='ECDelete.jsp';");
    out.println("};");
     out.println("</script>");
     out.print("</div>");
      out.print("</form>");
            out.print("</table>");
         } catch (Exception ex) {
             System.out.println(ex.getMessage());
       out.println("System resources is temporally out of service.");
         }
   out.println("<P>");
   out.println("</body></html>");
   return "Thanks for your visit";
     }
    public String  getHR(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
       response.setContentType("text/html");
       PrintWriter out = response.getWriter();
       out.println("<html>");
       out.println("<head><title>Associated medical Hospital.</title>");
       out.println("<style type=\"text/css\">");
       out.println("<!--");
       out.println(".style2 {font-size: 11px}");
       out.println("-->");
       out.println("</style>");
       out.println("</head>");
       out.println("<body>");
       out.println("<P>");
    out.print("<form name=\"form1\" method=\"post\">");
    out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
```

```
out.println("<H1 ALIGN=\"CENTER\">Associated medical hospital Reports</H1>");
 out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
  out.print("<table width=\"731\" border=\"1\">");
   try {
       prepStmt = conn.prepareStatement("select * from HR");
       rs=prepStmt.executeQuery();
       out.print("<tr>");
                out.print("<td width=\"44\"><span class=\"style2\">Hosp. ID</span></td>");
        out.print("<td width=\"111\"><span class=\"style2\">Hosp. Name</span></td>");
         out.print("<td width=\"44\"><span class=\"style2\">Loation</span></td>");
         out.print("<td width=\"61\"><span class=\"style2\">WWW </span></td>");
          out.print("<td width=\"63\"><span class=\"style2\">Hosp. email </span></td>");
           out.print("<td width=\"68\"><span class=\"style2\">Hosp. Ownership</span></td>");
           out.print("<td width=\"57\"><span class=\"style2\">Date Established</span></td>");
         out.print("<td width=\"69\"><span class=\"style2\">Hosp. Telephone</span></td>");
         out.print("<td width=\"90\"><span class=\"style2\">CMO</span></td>");
           out.print("<td width=\"90\"><span class=\"style2\">CMO Qualification</span></td>");
             out.print("<td width=\"90\"><span class=\"style2\">Hosp. Expertise</span></td>");
          out.print("</tr>");
          while( rs.next() ){
         out.print("<tr>");
         out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
         out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
         out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
          String www=rs.getString(4);
           out.print("<td><span class=\"style2\"><a href=\""+www+"\">"+www+"</a></span></td>");
          out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
          out.print("<td><span class=\"style2\">"+rs.getString(6)+"</span></td>");
           out.print("<td><span class=\"style2\">"+rs.getString(7)+"</span></td>");
          out.print("<td><span class=\"style2\">"+rs.getString(8)+"</span></td>");
           out.print("<td><span class=\"style2\">"+rs.getString(9)+"</span></td>");
             out.print("<td><span class=\"style2\">"+rs.getString(10)+"</span></td>");
               out.print("<td><span class=\"style2\">"+rs.getString(11)+"</span></td>");
          out.print("</tr>");
          }
          out.print("</span>");
            out.print("</table>");
                out.print("<div align=\"center\">");
             out.print("<INPUT TYPE=\"button\" NAME=\"Addnew\" value=\"Add new Record\" onclick=\"LoadMe()\">");
              out.print("<INPUT TYPE=\"button\" NAME=\"DeleteRec\" value=\"Delete Record\" onclick=\"DeleteMe()\">");
          out.println("<script language='JavaScript' type='text/JavaScript'>");
      out.println("function LoadMe(){");
    //      out.println("window.open('hrPage.jsp');");
      out.println("location.href='hrPage.jsp';");
      out.println("};");
        out.println("</script>");
      out.println("<script language='JavaScript' type='text/JavaScript'>");
      out.println("function DeleteMe(){");
     // out.println("window.open('DeleteHR.jsp');");
     out.println("location.href='DeleteHR.jsp';");
     out.println("};");
      out.println("</script>");
      out.print("</div>");
       out.print("</form>");
          } catch (Exception ex) {
              System.out.println(ex.getMessage());
        out.println("System resources is temporally out of service.");
          }
   out.println("</body></html>");
   return "Thanks for your visit";
     }
 public String  getOPT(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
```

```
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head><title>Operation theatre informtion.</title>");
out.println("<style type=\"text/css\">");
out.println("<!--");
out.println(".style2 {font-size: 11px}");
out.println("-->");
out.println("</style>");
out.println("</head>");
out.println("<body>");
out.println("<P>");
out.print("<form name=\"form1\" method=\"post\">");
out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
out.println("<H1 ALIGN=\"CENTER\">Medical Operation theatre Reports</H1>");
out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
out.print("<table width=\"731\" border=\"1\">");
try {
    prepStmt = conn.prepareStatement("select * from OPT");
    rs=prepStmt.executeQuery();
    out.print("<tr>");
        out.print("<td width=\"44\"><span class=\"style2\">Patient ID</span></td>");
     out.print("<td width=\"111\"><span class=\"style2\">Patient Name</span></td>");
         out.print("<td width=\"44\"><span class=\"style2\">Surgery Type </span></td>");
      out.print("<td width=\"44\"><span class=\"style2\">Surgery Date </span></td>");
      out.print("<td width=\"61\"><span class=\"style2\">Doctor in charge</span></td>");
       out.print("<td width=\"63\"><span class=\"style2\">After Surgical Ward</span></td>");
        out.print("<td width=\"68\"><span class=\"style2\">Bill</span></td>");
        out.print("<td width=\"57\"><span class=\"style2\">Recall Date</span></td>");
      out.print("<td width=\"69\"><span class=\"style2\">Evaluation of Success</span></td>");
      out.print("<td width=\"90\"><span class=\"style2\">Tools Required</span></td>");
       out.print("</tr>");
       while( rs.next() ){
      out.print("<tr>");
      out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
      out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
      out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
       out.print("<td><span class=\"style2\">"+rs.getString(4)+"</span></td>");
       out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
       out.print("<td><span class=\"style2\">"+rs.getString(6)+"</span></td>");
        out.print("<td><span class=\"style2\">"+rs.getString(7)+"</span></td>");
       out.print("<td><span class=\"style2\">"+rs.getString(9)+"</span></td>");
        out.print("<td><span class=\"style2\">"+rs.getString(8)+"</span></td>");
             out.print("<td><span class=\"style2\">"+rs.getString(10)+"</span></td>");
       out.print("</tr>");
       }
       out.print("</span>");
           out.print("</table>");
          out.print("<div align=\"center\">");
        out.print("<INPUT TYPE=\"button\" NAME=\"Addnew\" value=\"Add new Record\" onclick=\"LoadMe()\">");
         out.print("<INPUT TYPE=\"button\" NAME=\"DeleteRec\" value=\"Delete Record\" onclick=\"DeleteMe()\">");
       out.println("<script language='JavaScript' type='text/JavaScript'>");
   out.println("function LoadMe(){");
   //    out.println("window.open('OPTPage.jsp');");
   out.println("location.href='OPTPage.jsp';");
   out.println("};");
     out.println("</script>");
   out.println("<script language='JavaScript' type='text/JavaScript'>");
   out.println("function DeleteMe(){");
  // out.println("window.open('DeleteOPT.jsp');");
  out.println("location.href='DeleteOPT.jsp';");
```

```
    out.println("};");
     out.println("</script>");
     out.print("</div>");
      out.print("</form>");
           out.print("</table>");
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
       out.println("System resources is temporally out of service.");
        }
   out.println("<P>");
   out.println("</body></html>");
   return "Thanks for your visit";
    }
     public String  getStaff(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
       response.setContentType("text/html");
       PrintWriter out = response.getWriter();
       out.println("<html>");
       out.println("<head><title>Medical Staff.</title>");
       out.println("<style type=\"text/css\">");
       out.println("<!--");
       out.println(".style2 {font-size: 11px}");
       out.println("-->");
       out.println("</style>");
       out.println("</head>");
       out.println("<body>");
       out.println("<P>");
    out.print("<form name=\"form1\" method=\"post\">");
    out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
    out.println("<H1 ALIGN=\"CENTER\">Medical Staff Reports</H1>");
     out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
      out.print("<table width=\"731\" border=\"1\">");
       try {
           prepStmt = conn.prepareStatement("select * from hospadmin");
           rs=prepStmt.executeQuery();
           out.print("<tr>");
               out.print("<td width=\"44\"><span class=\"style2\">Staff ID</span></td>");
            out.print("<td width=\"111\"><span class=\"style2\">Staff Name</span></td>");
             out.print("<td width=\"44\"><span class=\"style2\">Function </span></td>");
             out.print("<td width=\"61\"><span class=\"style2\">Rank </span></td>");
              out.print("<td width=\"63\"><span class=\"style2\">Chain of Control</span></td>");
             out.print("</tr>");
              while( rs.next() ){
             out.print("<tr>");
             out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
             out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
             out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
              out.print("<td><span class=\"style2\">"+rs.getString(4)+"</span></td>");
              out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
              out.print("</tr>");
              }
              out.print("</span>");
                          out.print("</table>");
                   out.print(" <div align=\"center\">");
                out.print("<INPUT TYPE=\"button\" NAME=\"Addnew\" value=\"Add new Record\" onclick=\"LoadMe()\">");
                 out.print("<INPUT TYPE=\"button\" NAME=\"DeleteRec\" value=\"Delete Record\" onclick=\"DeleteMe()\">");
              out.println("<script language='JavaScript' type='text/JavaScript'>");
          out.println("function LoadMe(){");
         //     out.println("window.open('StaffPage.jsp');");
          out.println("location.href='StaffPage.jsp';");
          out.println("};");
            out.println("</script>");
```

```
out.println("<script language='JavaScript' type='text/JavaScript'>");
out.println("function DeleteMe(){");
// out.println("window.open('DeleteStaff.jsp');");
out.println("location.href='DeleteStaff.jsp';");
out.println("};");
out.println("</script>");
out.print("</div>");
out.print("</form>");
    out.print("</table>");
  } catch (Exception ex) {
     System.out.println(ex.getMessage());
  out.println("System resources is temporally out of service.");
  }
out.println("<P>");
out.println("</body></html>");
return "Thanks for your visit";
 }
public String  AddEMR1(HttpServletRequest request, HttpServletResponse
  response) throws ServletException, IOException {
 response.setContentType("text/html");
 PrintWriter out = response.getWriter();
 out.println("<html>");
 out.println("<head><title>Hospital Equipments.</title>");
 out.println("<style type=\"text/css\">");
 out.println("<!--");
 out.println(".style2 {font-size: 11px}");
 out.println("-->");
 out.println("</style>");
 out.println("</head>");
 out.println("<body>");
 out.println("<P>");
out.print("<form name=\"form1\" method=\"post\">");
out.println("<H1 ALIGN=\"CENTER\">Hospital Medical Equipments Reports</H1>");
out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
out.print("<table width=\"731\" border=\"1\">");
 try {
    prepStmt = conn.prepareStatement("select * from HR");
    rs=prepStmt.executeQuery();
    out.print("<tr>");
    flag=rs.next();
    if(flag){
      out.print("<td width=\"44\"><span class=\"style2\">Med ID </span></td>");
     out.print("<td width=\"111\"><span class=\"style2\">Description</span></td>");
     out.print("<td width=\"44\"><span class=\"style2\">Qty Purchase </span></td>");
     out.print("<td width=\"61\"><span class=\"style2\">Date Purchased </span></td>");
     out.print("<td width=\"63\"><span class=\"style2\">Qty Sold </span></td>");
      out.print("<td width=\"68\"><span class=\"style2\">Bal</span></td>");
      out.print("<td width=\"57\"><span class=\"style2\">Up</span></td>");
     out.print("<td width=\"69\"><span class=\"style2\">Status of Med equip</span></td>");
     out.print("<td width=\"90\"><span class=\"style2\">Location</span></td>");
     out.print("<td width=\"60\"><span class=\"style2\">Naira Value </span></td>");
     out.print("</tr>");
     while( rs.next() ){
     out.print("<tr>");
     out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
     out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
     out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
     out.print("<td><span class=\"style2\">"+rs.getString(4)+"</span></td>");
     out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
     out.print("<td><span class=\"style2\">"+rs.getString(6)+"</span></td>");
      out.print("<td><span class=\"style2\">"+rs.getString(7)+"</span></td>");
     out.print("<td><span class=\"style2\">"+rs.getString(8)+"</span></td>");
      out.print("<td><span class=\"style2\">"+rs.getString(9)+"</span></td>");
```

```
                out.print("<td><span class=\"style2\">"+rs.getString(10)+"</span></td>");
                out.print("</tr>");
               }
               out.print("<tr>");
               out.print("</tr>");
               }else{
              out.print("<tr>");
              out.print("<td><span class=\"style2\">"+"No valid Record found"+"</span></td>");
              out.print("</tr>");
              out.print("<TR><TD> <div align=\"center\">");
          out.print("<INPUT TYPE=\"Submit\" NAME=\"Addnew\" value=\"Add new Record\" onclick=\"<%");
        out.println("<script language='JavaScript' type='text/JavaScript'>");
     out.println("function LoadMe(){");
     out.println("window.open('hrpage.jsp');");
      out.println("};");
      out.println("</script>");
      out.println("%>");
         out.print("</div></TD>");
               }
            out.print("</table>");
           out.print("</form>");
               out.print("</table>");
             } catch (Exception ex) {
                 System.out.println(ex.getMessage());
            out.println("System resources is temporally out of service.");
               }
        out.println("<P>");
        out.println("</body></html>");
        return "Thanks for your visit";
         }
         public String  getDrug(HttpServletRequest request, HttpServletResponse
              response) throws ServletException, IOException {
             response.setContentType("text/html");
             PrintWriter out = response.getWriter();
             out.println("<html>");
             out.println("<head><title>Drug Inventory.</title>");
             out.println("<style type=\"text/css\">");
             out.println("<!--");
             out.println(".style2 {font-size: 11px}");
             out.println("-->");
             out.println("</style>");
             out.println("</head>");
             out.println("<body>");
             out.println("<P>");
          out.print("<form name=\"form1\" method=\"post\">");
          out.print("<h1 align=\"center\"> Automated Intelligent Hospitals Ltd</h1>");
out.print("<h4 align=\"center\"> No. 18 Zander Street , Warri, Delta State</h4> </p>");
          out.println("<H1 ALIGN=\"CENTER\">Hospital Drug Inventory Report </H1>");
           out.println("<H2 ALIGN=\"CENTER\">as at :"+ getSystemDate()+"</H2>");
            out.print("<table width=\"731\" border=\"1\">");
             try {
                 prepStmt = conn.prepareStatement("select * from Drug");
                 rs=prepStmt.executeQuery();
                 out.print("<tr>");
                 flag=rs.next();
                     out.print("<td width=\"44\"><span class=\"style2\">Drug ID </span></td>");
                  out.print("<td width=\"111\"><span class=\"style2\">Description</span></td>");
                   out.print("<td width=\"44\"><span class=\"style2\">Qty In stock</span></td>");
                out.print("<td width=\"69\"><span class=\"style2\">Qty Ordered</span></td>");
                   out.print("<td width=\"69\"><span class=\"style2\">Reorder Level</span></td>");
                   out.print("<td width=\"61\"><span class=\"style2\">Qty Issued out/Sold </span></td>");
                               out.print("<td width=\"69\"><span class=\"style2\">Balance</span></td>");
                    out.print("<td width=\"63\"><span class=\"style2\">Naira Value Bal </span></td>");
```

```
                out.print("<td width=\"68\"><span class=\"style2\">Date purchased</span></td>");
                out.print("<td width=\"90\"><span class=\"style2\">Location</span></td>");
               out.print("<td width=\"60\"><span class=\"style2\">Remarks </span></td>");
               out.print("</tr>");
                while( rs.next() ){
               out.print("<tr>");
               out.print("<td><span class=\"style2\">"+rs.getString(1)+"</span></td>");
               out.print("<td><span class=\"style2\">"+rs.getString(2)+"</span></td>");
               out.print("<td><span class=\"style2\">"+rs.getString(3)+"</span></td>");
                out.print("<td><span class=\"style2\">"+rs.getString(4)+"</span></td>");
                out.print("<td><span class=\"style2\">"+rs.getString(5)+"</span></td>");
                out.print("<td><span class=\"style2\">"+rs.getString(6)+"</span></td>");
                out.print("<td><span class=\"style2\">"+rs.getString(7)+"</span></td>");
                out.print("<td><span class=\"style2\">"+rs.getString(8)+"</span></td>");
                out.print("<td><span class=\"style2\">"+rs.getString(9)+"</span></td>");
                out.print("<td><span class=\"style2\">"+rs.getString(10)+"</span></td>");
                out.print("<td><span class=\"style2\">"+rs.getString(11)+"</span></td>");
                out.print("</tr>");
                }
                out.print("</table>");
                out.print("</span>");
        out.print("<TR><TD> <div align=\"center\">");
       out.print("<INPUT TYPE=\"button\" NAME=\"Addnew\" value=\"Add/Edit Drug\" onclick=\"LoadMe()\">");
       out.print("<INPUT TYPE=\"button\" NAME=\"DeleteRec\" value=\"Delete Drug\" onclick=\"DeleteMe()\">");
     out.println("<script language='JavaScript' type='text/JavaScript'>");
 out.println("function LoadMe(){");
 //    out.println("window.open('DrugInt.jsp');");
 out.println("location.href='DrugInt.jsp';");
 out.println("};");
   out.println("</script>");
 out.println("<script language='JavaScript' type='text/JavaScript'>");
 out.println("function DeleteMe(){");
// out.println("window.open('DeleteDrug.jsp');");
out.println("location.href='DeleteDrug.jsp';");
out.println("};");
 out.println("</script>");
 out.print("</div></TD>");
 out.print("</form>");
               } catch (Exception ex) {
                   System.out.println(ex.getMessage());
              out.println("System resources is temporally out of service.");
                }
            out.println("<P>");
            out.println("</body></html>");
            return "Thanks for your visit";
             }
             public void AddEMR(){
                   try{
                         prepStmt = conn.prepareStatement("select * from EMR where PiD=?");
                         prepStmt.setString(1,PiD);
                        rs=prepStmt.executeQuery();
                        System.out.println("About to connec");
                        flag=rs.next() ;
                        if (flag){
prepStmt = conn.prepareStatement("update EMR set Pname=?, History=?, SensorReportsPDA=?, RemarksFromCDS=?
where PiD=?");
                         prepStmt.setString(1,PName);
                         prepStmt.setString(2,History);
                         prepStmt.setString(3,SensorReportsPDA);
                         prepStmt.setString(4,RemarksfromCDS);
                         prepStmt.setString(5,PiD);
                         rs=prepStmt.executeQuery();
                        }
```

```
                    if (!flag){
                      prepStmt = conn.prepareStatement("insert into EMR value(?,?,?,?,?)");
                      prepStmt.setString(1,PName);
                      prepStmt.setString(2,PiD);
                      prepStmt.setString(3,History);
                      prepStmt.setString(4,SensorReportsPDA);
                      prepStmt.setString(5,RemarksfromCDS);
                      rs=prepStmt.executeQuery();
                    }
                      } catch (Exception ex) {
                     System.out.print("No data base found");
               }
            }
            public void AddHR(){
                  try{
                       prepStmt = conn.prepareStatement("select * from HR where HospID=?");
                       prepStmt.setString(1,HospID);
                      rs=prepStmt.executeQuery();
                      System.out.println("About to connec");
                      flag=rs.next() ;
                      if (flag){
                        prepStmt = conn.prepareStatement("update HR Set HospNameAdr=?,Location=?, WWW=?,
Hemail=?,"+
                        " HOwnership=?,DataEstab=?,Hexpertise=?,CMO=?,CMOQualif=?,HospID=? where HospID=?");
                        prepStmt.setString(1,HospNameAdr);
                        prepStmt.setString(2,Location);
                        prepStmt.setString(3,WWW);
                        prepStmt.setString(4,Hemail);
                        prepStmt.setString(5,Hownership);
                        prepStmt.setString(6,DateEstab);
                        prepStmt.setString(7,HExpertise);
                        prepStmt.setString(8,CMO);
                        prepStmt.setString(9,CMOQualif);
                        prepStmt.setString(10,HospID);
                        prepStmt.setString(11,HospID);
                        rs=prepStmt.executeQuery();
                      }
                      if (!flag){
                        prepStmt = conn.prepareStatement("insert into HR value(?,?,?,?,?,?,?,?,?,?)");
                        prepStmt.setString(1,HospNameAdr);
                        prepStmt.setString(2,Location);
                        prepStmt.setString(3,WWW);
                        prepStmt.setString(4,Hemail);
                        prepStmt.setString(5,Hownership);
                        prepStmt.setString(6,DateEstab);
                        prepStmt.setString(7,HExpertise);
                        prepStmt.setString(8,CMO);
                        prepStmt.setString(9,CMOQualif);
                        prepStmt.setString(10,HospID);
                        rs=prepStmt.executeQuery();
                      }
                        } catch (Exception ex) {
                       System.out.print("No data base found");
                  }
              }
 public void AddStaff(){
       try{
             prepStmt = conn.prepareStatement("select * from HospAdmin where StaffID=?");
             prepStmt.setString(1,StaffID);
            rs=prepStmt.executeQuery();
            System.out.println("About to connec");
            flag=rs.next() ;
            if (flag){
```

```
            prepStmt = conn.prepareStatement("update HospAdmin Set StaffNameIC=?,Function=?, Rank=?,
ChainOfCntrl=?"+
                                    " where StaffID=?");
            prepStmt.setString(1,StaffNameIc);
            prepStmt.setString(2,Function);
            prepStmt.setString(3,Rank);
            prepStmt.setString(4,ChainOfCntrl);
            prepStmt.setString(5,StaffID);
            rs=prepStmt.executeQuery();
           }
           if (!flag){
            prepStmt = conn.prepareStatement("insert into HospAdmin value(?,?,?,?,?)");
            prepStmt.setString(1,StaffID);
            prepStmt.setString(2,StaffNameIc);
            prepStmt.setString(3,Function);
            prepStmt.setString(4,Rank);
            prepStmt.setString(5,ChainOfCntrl);
            rs=prepStmt.executeQuery();
           }
            } catch (Exception ex) {
           System.out.print("No data base found");
       }
   }
   public void AddClinicalSupt(){
        try{
             prepStmt = conn.prepareStatement("select * from ClinicalSupt where StaffID=?");
             prepStmt.setString(1,StaffID);
            rs=prepStmt.executeQuery();
            System.out.println("About to connec");
            flag=rs.next() ;
            if (flag){
              prepStmt = conn.prepareStatement("update ClinicalSupt Set StaffNameIC=?,Function=?, Ward=?,"+
              "Reportsto=?,EMR=?,OT=?,Duty_Roster=?,Lab=? where StaffID=?");
              prepStmt.setString(1,StaffNameIc);
              prepStmt.setString(2,Function);
              prepStmt.setString(3,Ward);
              prepStmt.setString(4,Reportsto);
              prepStmt.setString(5,EMR);
              prepStmt.setString(6,OT);
              prepStmt.setString(7,Duty_Roster);
              prepStmt.setString(8,Lab);
              prepStmt.setString(9,StaffID);
              rs=prepStmt.executeQuery();
            }
            if (!flag){
              prepStmt = conn.prepareStatement("insert into ClinicalSupt value(?,?,?,?,?,?,?,?,?)");
              prepStmt.setString(1,StaffID);
              prepStmt.setString(2,StaffNameIc);
              prepStmt.setString(3,Function);
              prepStmt.setString(4,Ward);
              prepStmt.setString(5,Reportsto);
              prepStmt.setString(6,EMR);
              prepStmt.setString(7,OT);
              prepStmt.setString(8,Duty_Roster);
              prepStmt.setString(9,Lab);
              rs=prepStmt.executeQuery();
            }
              } catch (Exception ex) {
             System.out.print("No data base found");
        }
     }
     public void Addancservice(){
             try{
```

```
 prepStmt = conn.prepareStatement("select * from ancservice where HiD=?");
 prepStmt.setString(1,HiD);
rs=prepStmt.executeQuery();
System.out.println("About to connec");
flag=rs.next() ;
if (flag){
  prepStmt = conn.prepareStatement("update ancservice Set BloodTest=?,Urine=?, Cardiology=?,"+
  "BloodPressure=?,BloodSuger=?,Temp=?,Malaria=?,HIV=?,"+
  "Tuberculosis=?,Cholera=?,Hepatitis=?,Eye=?,Ear=?,"+
  "Mouth=?,Stool=?,Other_Test=? where StaffID=?");
  prepStmt.setString(1,BloodTest);
  prepStmt.setString(2,Urine);
  prepStmt.setString(3,Cardiology);
  prepStmt.setString(4,BloodPressure);
  prepStmt.setString(5,BloodSugar);
  prepStmt.setString(6,Temp);
  prepStmt.setString(7,Malaria);
  prepStmt.setString(8,HIV);
  prepStmt.setString(9,Tuberculosis);
  prepStmt.setString(10,Cholera);
 prepStmt.setString(11,Hepatitis);
  prepStmt.setString(12,Eye);
 prepStmt.setString(13,Ear);
 prepStmt.setString(14,Mouth);
 prepStmt.setString(15,Stool);
 prepStmt.setString(16,OtherTests);
  prepStmt.setString(17,HiD);
  rs=prepStmt.executeQuery();
}
if (!flag){
  prepStmt = conn.prepareStatement("insert into ClinicalSupt"+
  " value(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
  prepStmt.setString(1,BloodTest);
  prepStmt.setString(2,Urine);
  prepStmt.setString(3,Cardiology);
  prepStmt.setString(4,BloodPressure);
  prepStmt.setString(5,BloodSugar);
  prepStmt.setString(6,Temp);
  prepStmt.setString(7,Malaria);
  prepStmt.setString(8,HIV);
  prepStmt.setString(9,Tuberculosis);
  prepStmt.setString(10,Cholera);
 prepStmt.setString(11,Hepatitis);
  prepStmt.setString(12,Eye);
 prepStmt.setString(13,Ear);
 prepStmt.setString(14,Mouth);
 prepStmt.setString(15,Stool);
 prepStmt.setString(16,OtherTests);
  prepStmt.setString(17,HiD);
  rs=prepStmt.executeQuery();
}
  } catch (Exception ex) {
 System.out.print("No data base found");
    }
}
  public void AddMe(){
      try{
        prepStmt = conn.prepareStatement("select * from Me where EquipID=?");
        prepStmt.setString(1,EquipID);
       rs=prepStmt.executeQuery();
       System.out.println("About to connec");
       flag=rs.next() ;
       if (flag){
```

```
int OldBalance=rs.getInt(5) ;
prepStmt = conn.prepareStatement("insert into Me"+
" value(?,?,?,?,?,?,?)");
prepStmt.setString(1,EquipID);
prepStmt.setString(2,EquipType);
prepStmt.setInt(3,QtyPurchased);
prepStmt.setInt(4 ,QtySold);
prepStmt.setInt(5,(OldBalance+QtyPurchased)-QtySold);
prepStmt.setString(6,EquipStatus);
prepStmt.setString(7,Location);
rs=prepStmt.executeQuery();
}
if (!flag){
prepStmt = conn.prepareStatement("insert into Me"+
                    " value(?,?,?,?,?,?,?)");
prepStmt.setString(1,EquipID);
prepStmt.setString(2,EquipType);
prepStmt.setInt(3,QtyPurchased);
prepStmt.setInt(4 ,QtySold);
prepStmt.setInt(5,QtyPurchased-QtySold);
prepStmt.setString(6,EquipStatus);
prepStmt.setString(7,Location);
rs=prepStmt.executeQuery();
}
} catch (Exception ex) {
System.out.print("No data base found");
}
}
public void AddOT(){
    try{
        prepStmt = conn.prepareStatement("select * from operation_theatre where PiD=?");
        prepStmt.setString(1,PiD);
          rs=prepStmt.executeQuery();
        System.out.println("About to connec");
        flag=rs.next() ;
        if (flag){
         prepStmt = conn.prepareStatement("update operation_theatre Set PName=?,"+
         "PiD=? SurgeryDate=?, DoctorIc=?, SurgeryType=?,Tools_Req=?,After_SurgWard=?,Bill=?,"+
         "EvaluationOfSuccess=?,Recall=? where Pid=?");
         prepStmt.setString(1,PName);
         prepStmt.setString(2,PiD);
         prepStmt.setString(3,History);
         prepStmt.setString(4,SensorReportsPDA);
         prepStmt.setString(5,RemarksfromCDS);
         prepStmt.setString(6,PiD);
         rs=prepStmt.executeQuery();
        }
         } catch (Exception ex) {
        System.out.print("No data base found");
    }
  }
   public String HospID="";
  public String getHospID() {
 return HospID;
 }
  public String HTel="";
  public String ExpertID="";
  public String getExpertID() {
return ExpertID;
}
  public String getHTel() {
return HTel;
}
```

```
public void getECDB(){
      try{
            prepStmt = conn.prepareStatement("select * from EC where ExpertID=?");
            prepStmt.setString(1,ExpertID);
           rs=prepStmt.executeQuery();
           System.out.println("About to connec");
        while( rs.next() ){
        ExpertID=rs.getString(1);ExpertName=rs.getString(2);
        Expertise=rs.getString(3);HospLocation=rs.getString(4);Qualifications=rs.getString(5);
          ETel=rs.getString(6);Eemail=rs.getString(7);EResidence=rs.getString(8);
          HospAvailability=rs.getString(9);
        }
             } catch (Exception ex) {
            System.out.print("No data base found");
      }
  }
public void getClinicalDB(){
     try{
          prepStmt = conn.prepareStatement("select * from ClinicalSupt where Staff_NameIc=?");
          prepStmt.setString(1,StaffNameIc);
         rs=prepStmt.executeQuery();
         System.out.println("About to connec");
      while( rs.next() ){
      StaffNameIc=rs.getString(1);Function=rs.getString(2);
      Ward=rs.getString(3);Reports_To=rs.getString(4);Lab=rs.getString(5);
      EMR=rs.getString(6);OT=rs.getString(7);Duty_Roster=rs.getString(8);
           }
           } catch (Exception ex) {
          System.out.print("No data base found");
    }
}
public void getStaffDB(){
     try{
          prepStmt = conn.prepareStatement("select * from hospadmin where Staff_NameIc=?");
          prepStmt.setString(1,StaffNameIc);
         rs=prepStmt.executeQuery();
         System.out.println("About to connec");
      while( rs.next() ){
      StaffNameIc=rs.getString(1);Function=rs.getString(2);
      Rank=rs.getString(3);ChainOfCntrl=rs.getString(4);
           }
           } catch (Exception ex) {
          System.out.print("No data base found");
     }
 }
public void setHiD(String newHiD) {
 if ( newHiD!=null) {
  HiD = newHiD;
  }
}
public void setPName(String newPName) {
if ( newPName!=null) {
  PName = newPName;
  }
}
public String getPName() {
  return Pname;
}
public String getPAddress() {
   return PAddress;
  }
  public String getPTel() {
     return PTel;
```

```
    }
  public String getReferedFrom() {
       return ReferedFrom;
      }
      public String getPemail() {
         return Pemail;
        }
        public String getDateAppt() {
           return DateAppt;
          }
          public String getDateAdmtd() {
           return DateAdmtd;
          }
     public String getGender() {
            return Gender;
           }
           public String getDateDischarged() {
       return DateDischarged;
     }
           public String getBloodGp() {
              return BloodGp;
             }
             public String getHospAdmtd() {
                return HospAdmtd;
               }
               public String getHWard() {
                  return HWard;
                 }
                 public String getBedNo() {
                    return BedNo;
                   }
                   public String getBedTransfer() {
                    return BedNo;
                  }
                   public String getDiagnosis() {
                      return Diagnosis;
                     }
                     public String getPrescription() {
                        return Prescription;
                       }
                       public String getTreatment() {
                          return Treatment;
                         }
                         public String getDocIc() {
                            return DocIc;
                           }
                           public String getSponsor() {
                              return Sponsor;
                             }
                             public String getSTel() {
                                return STel;
                               }
                               public String getSEmail() {
                                  return Semail;
                                 }
                                 public int getTBilltoDate() {
                                    return TBilltoDate;
                                   }
      public int getTBillPdtoDate() {
 return TBillPdtoDate;
}
public String getTDischargedDate() {
return DischargeDate;
```

```
}
 public String getStaffID() {
return StaffID;
}
 public String getStaffName() {
return StaffName;
}
 public String getFunction() {
return Function;
}
 public String getWard() {
return Ward;
}
 public String getReportsto() {
return Reportsto;
}
 public String getEMR() {
return EMR;
}
 public String getOT() {
return OT;
}
 public String getDuty_Roster() {
return DutyRoster;
}
 public String getEquipType() {
return EquipType;
}
 public String getEquipStatus() {
return EquipStatus;
}
 public String getHospNameAdr() {
return HospNameAdr;
}
 public String getLocation() {
return Location ;
}
 public String getWWW() {
return WWW;
}
 public String getHemail() {
return Hemail;
}
 public String getHiDs() {
return HiD;
}
 public String getHownership() {
return Hownership;
}
 public String getDateEstab() {
return DateEstab;
}
 public String getCMO() {
return CMO;
}
 public String getCMOQualif() {
return CMOQualif;
}
 public String getRank() {
return Rank;
}
 public String getChainOfCntrl() {
return ChainOfCntrl;
```

```
}
 public String getPiD() {
return PiD;
}
 public String getHistory() {
return History;
}
 public String getSensorReportsPDA() {
return SensorReportsPDA;
}
 public String getRemaksID() {
return RemarksID;
}
 public String getRemarks() {
return Remarks;
}
 public String getRemarksFromCDS() {
return RemarksfromCDS;
}
 public String getExpertName() {
return ExpertName;
}
 public String getExpertise() {
return Expertise;
}
 public String getHospLocation() {
return HospLocation;
}
 public String getQualifications() {
return Qualifications;
}
 public String getETel() {
return ETel;
}
 public String getExpertNamEemail() {
return ExpertNamEemail;
}
 public String getEemail() {
return Eemail;
}
 public String getEResidence() {
return EResidence;
}
 public String getHospAvailability() {
return HospAvailability;
}
 public String getDrugType() {
return DrugType;
}
 public int getQtyinStock() {
return QtyinStock;
}
public int QtyPurchase=0;
 public int getQtyPurchase() {
return QtyPurchase;
}
  public int getNairaValueBal() {
return NairaValueBal;
}
  public String QtyStatus="";
    public String getQtyStatus() {
  return QtyStatus;
  }
```

```
 public int getReorderLevel() {
return ReorderLevel;
}
 public int getQtyIsuedOut() {
return QtyIssuedOut;
}
 public int getBal() {
return Bal;
}
 public String getBloodTest() {
return BloodTest;
}
 public String getUrine() {
return Urine;
}
 public String getCardiologys() {
return Cardiology;
}
 public String getBloodPressure() {
return BloodPressure;
}
 public String getBloodSugar() {
return BloodSugar;
}
 public String getTemp() {
return Temp;
}
 public String getMalaria() {
return Malaria;
}
 public String getHIV() {
return HIV;
}
 public String getTuberculosis() {
return Tuberculosis;
}
 public String getCholera() {
return Cholera;
}
 public String getHepatitis() {
return Hepatitis;
}
 public String getEye() {
return Eye;
}
 public String getEquiptype() {
return EquipType;
}
 public String getEar() {
return Ear;
}
 public String getMouth() {
return Mouth;
}
 public String getStool() {
return Stool;
}
 public String getOthertests() {
return OtherTests;
}
 public String getSurgeryDate() {
return SurgeryDate;
}
```

```
 public String getDoctorIc() {
return DoctorIc;
}
 public String getSurgeryType() {
return SurgeryType;
}
 public String getToolsReq() {
return Tools_Req;
}
 public String getAfter_SurgWard() {
return AfterSurgWard;
}
 public int getBill() {
return Bill;
}
 public String getEvaluationOfSuccess() {
return EvaluationOfSuccess;
}
 public String getRecall() {
return Recall;
}
public String getPSat() {
return PatStatus;
}
  public String getAfterSurgWard() {
 return AfterSurgWard;
 }
   public String getReport_To() {
  return Reports_To;
  }
   public String HospName="";
     public String getHospName() {
    return HospName;
    }
    public String HTell="";
        public String getHTell() {
       return HTell;
       }
 public String StaffID="";
 public String StaffName="";
        public String getHExpertise() {
       return HExpertise;
       }
              public String getDrugID() {
             return DrugID;
             }
             public String getDescription() {
                         return Description;
                         }
                             public String getCardiology() {
                                            return Cardiology;
                                            }
                                            public String getOtherTests() {
                                                                  return OtherTests;
                                                                  }
}
```

Java codes for the encryption/decryption software

```
package org.eitse.view;

import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import org.eitse.checkMedStatus.CheckMedicalStatusDialog;
import org.eitse.registrationViews.PIDEncryptionDialog;
import java.awt.Color;
import javax.swing.JLabel;
import javax.swing.ImageIcon;

public class ApplicationMain extends JFrame {

	/**
	 *
	 */
	private static final long serialVersionUID = 1L;
	private JPanel contentPane;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					ApplicationMain frame = new ApplicationMain();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */
	public ApplicationMain() {
		setTitle("AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/DECRYPTION SOFTWARE");
		setResizable(false);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 700, 600);

		JMenuBar menuBar = new JMenuBar();
```

```
		setJMenuBar(menuBar);

		JMenu mnFile = new JMenu("File");
		menuBar.add(mnFile);

		JMenuItem mntmRegisterNewUser = new JMenuItem("Register New User");
		mntmRegisterNewUser.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				PIDEncryptionDialog obj = new PIDEncryptionDialog();
				obj.setVisible(true);
			}
		});
		mnFile.add(mntmRegisterNewUser);

		JMenuItem mntmCheckMedicalStatus = new JMenuItem("Check Medical Status");
		mntmCheckMedicalStatus.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				CheckMedicalStatusDialog obj = new CheckMedicalStatusDialog();
				obj.setVisible(true);
			}
		});
		mnFile.add(mntmCheckMedicalStatus);

		JMenuItem mntmExit = new JMenuItem("Exit");
		mnFile.add(mntmExit);

		JMenu mnHelp = new JMenu("Help");
		menuBar.add(mnHelp);

		JMenuItem mntmAbout = new JMenuItem("About");
		mnHelp.add(mntmAbout);
		contentPane = new JPanel();
		contentPane.setBackground(new Color(153, 204, 102));
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);
		contentPane.setLayout(null);

		JLabel lblNewLabel = new JLabel("");
		lblNewLabel.setIcon(new ImageIcon("C:\\Users\\itse\\OracleEclipseEEWorkspace\\Medical Status
Checker\\files\\medstatchecker.png"));
		lblNewLabel.setBounds(0, 0, 694, 551);
		contentPane.add(lblNewLabel);
	}
}
```

APPENDIX III
SAMPLE OUTPUTS

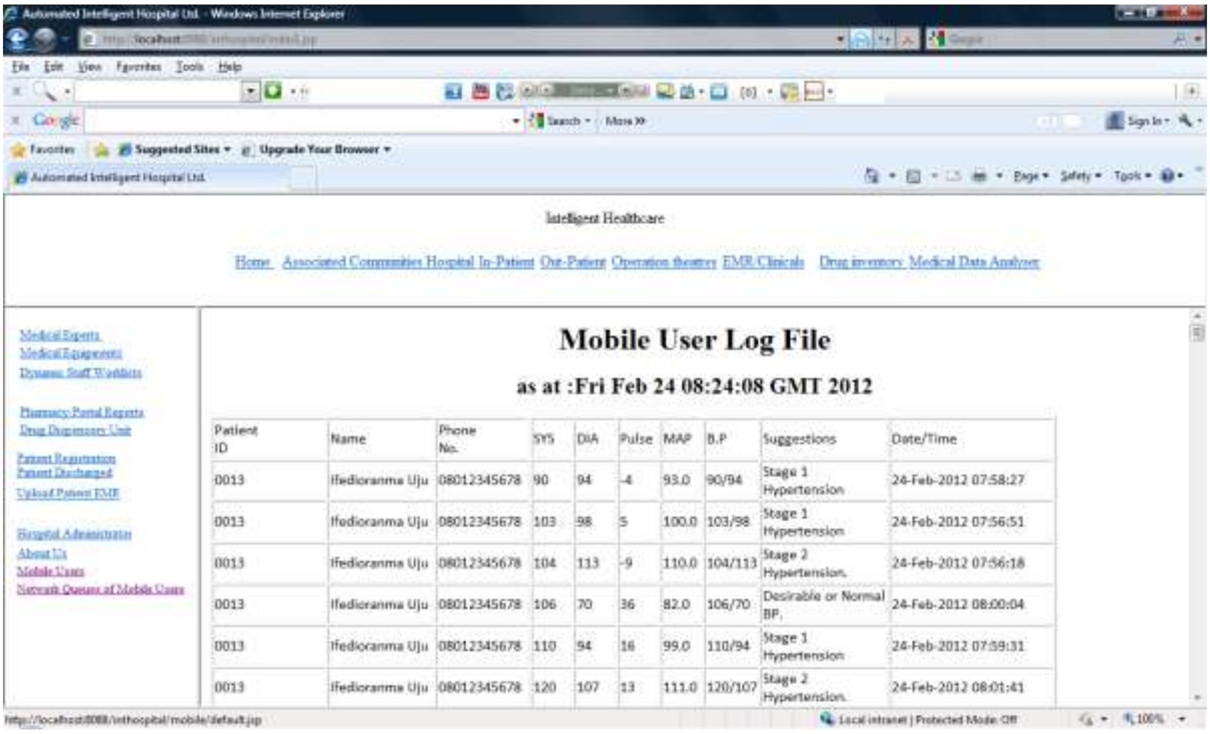

ClamshellCldcPhone1 (I...
Application View Help
Phone number: 123456789
Mobile Login
Connect to http://lo
Access key: ****
Menu
1 Setup
2 Login
Back
Sun
microsystems
12 Microsoft O...


ClamshellCldcPhone1 (I...
Application View Help
Phone number: 123456789
Mobile info Sent
Name:Ifeyinwa Okafor
Tel. No.:08034044149
Sys:162
DIA:69
Pulse:93
Done
Sun
microsystems
mobile - NetBeans I...


ClamshellCldcPhone1 (I...
Application View Help
Phone number: 123456789
Mobile Login
Connect to http://lo
Access key: ****
Exit
Menu
Sun
microsystems
12 Microsoft O...


ClamshellCldcPhone1 (I...
Application View Help
Phone number: 123456789
BP Transmission
Application is Active
Exit
Sun
microsystems
mobile - NetBeans I...

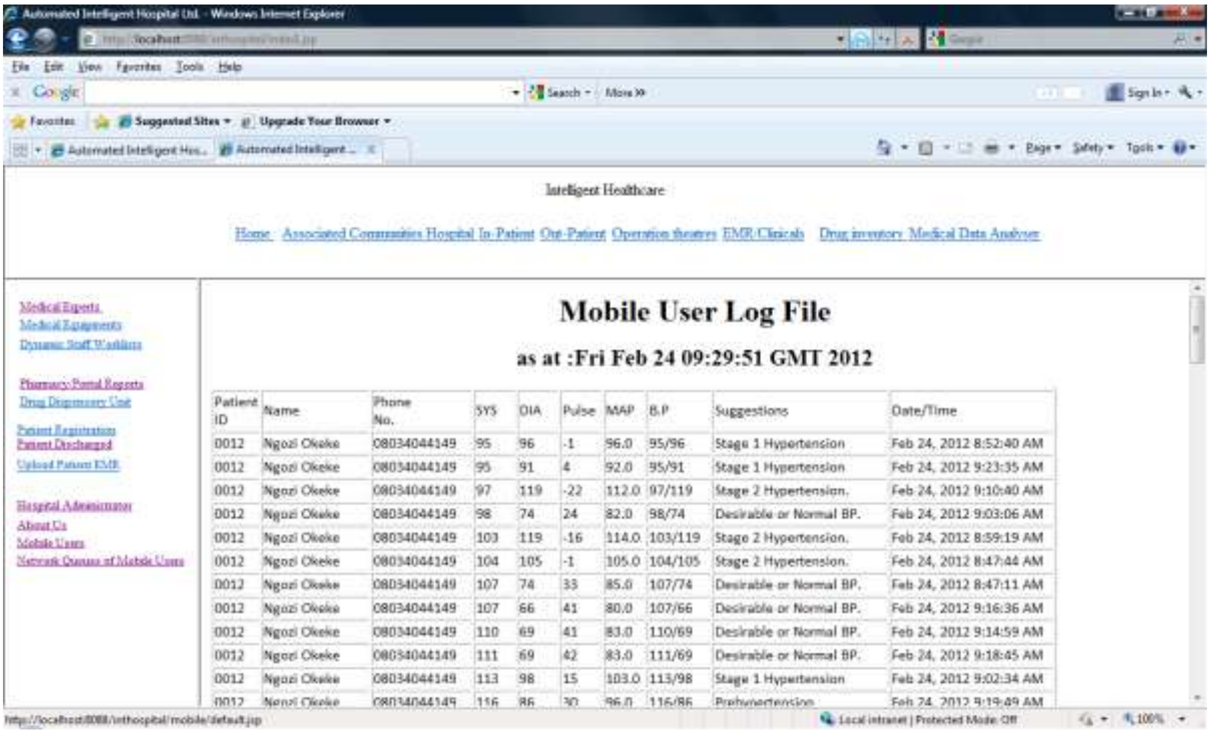

ClamshellCldcPhone1 (I...
Application View Help
Phone number: 123456789
Mobile info Sent
Name:Ifeanyi Okonkwo
Tel. No.:08012398234
Sys:103
DIA:85
Pulse:18
Done
Sun
microsystems
mobile - NetBeans I...



Mobile User Log File

as at :Fri Feb 24 08:24:08 GMT 2012

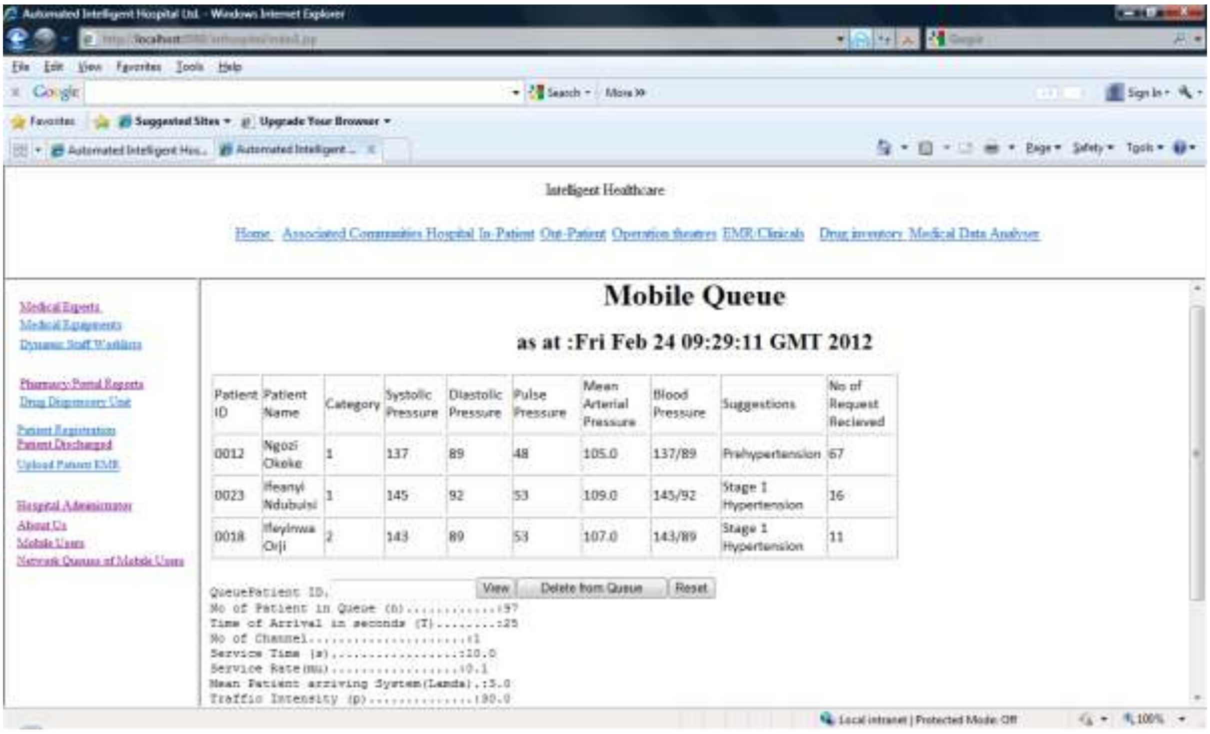
Patient ID	Name	Phone No.	SYS	DIA	Pulse	MAP	B.P	Suggestions	Date/Time
0013	Ifedioranma Uju	08012345678	90	94	-4	93.0	90/94	Stage 1 Hypertension	24-Feb-2012 07:58:27
0013	Ifedioranma Uju	08012345678	103	98	5	100.0	103/98	Stage 1 Hypertension	24-Feb-2012 07:56:51
0013	Ifedioranma Uju	08012345678	104	113	-9	110.0	104/113	Stage 2 Hypertension.	24-Feb-2012 07:56:18
0013	Ifedioranma Uju	08012345678	106	70	36	82.0	106/70	Desirable or Normal BP.	24-Feb-2012 08:00:04
0013	Ifedioranma Uju	08012345678	110	94	16	99.0	110/94	Stage 1 Hypertension	24-Feb-2012 07:59:31
0013	Ifedioranma Uju	08012345678	120	107	13	111.0	120/107	Stage 2 Hypertension.	24-Feb-2012 08:01:41

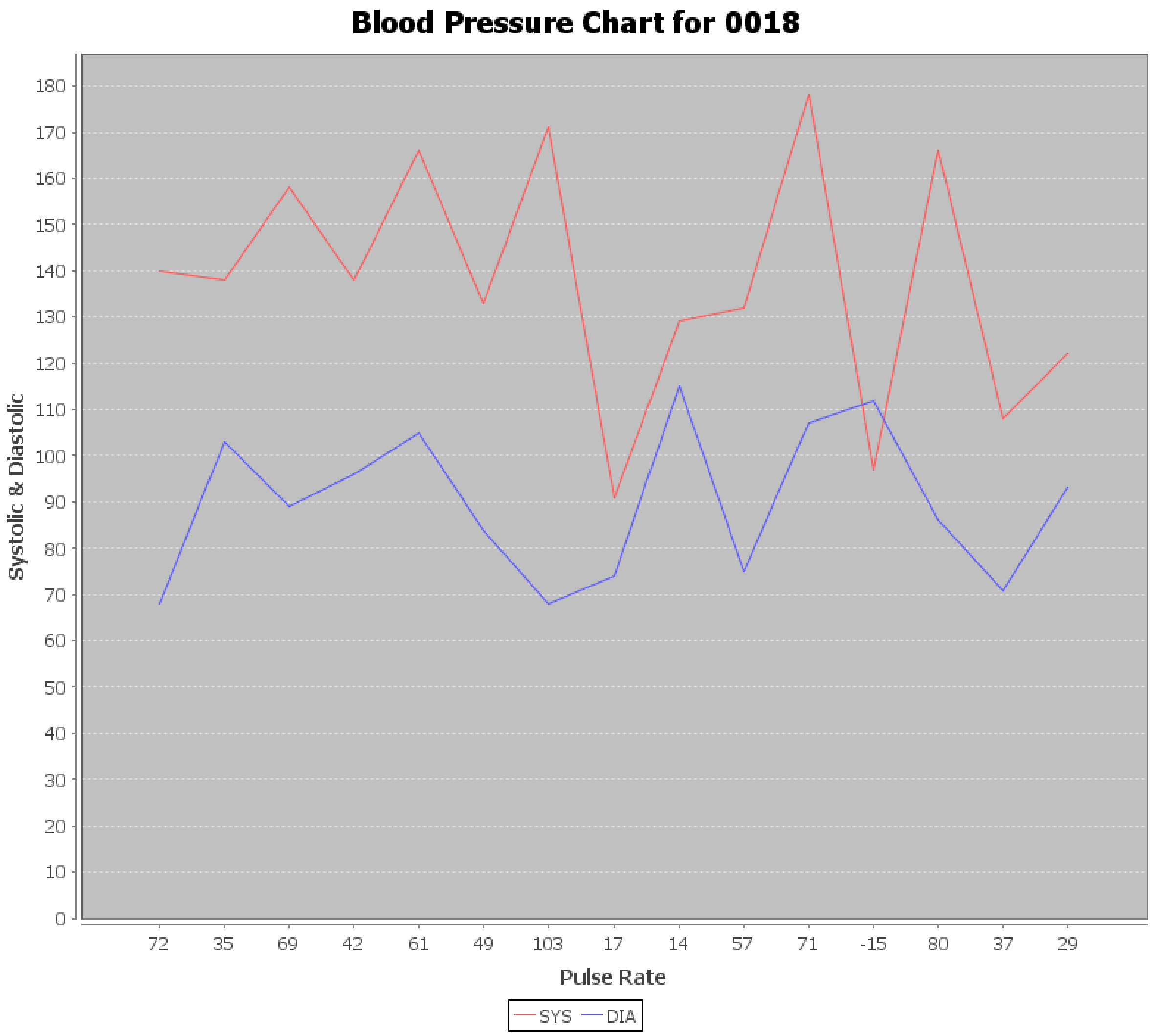


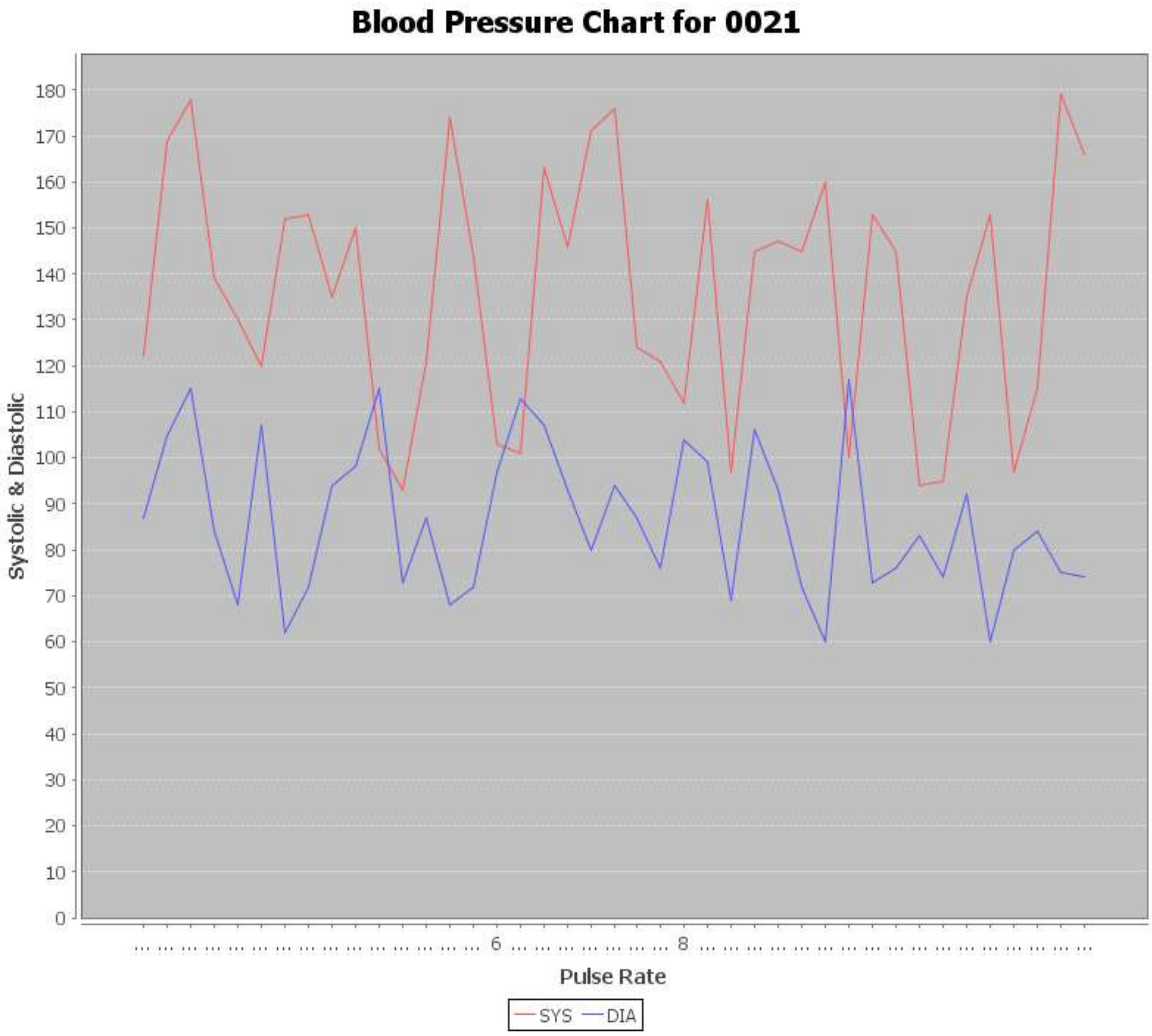
Mobile User Log File

as at :Fri Feb 24 09:29:51 GMT 2012

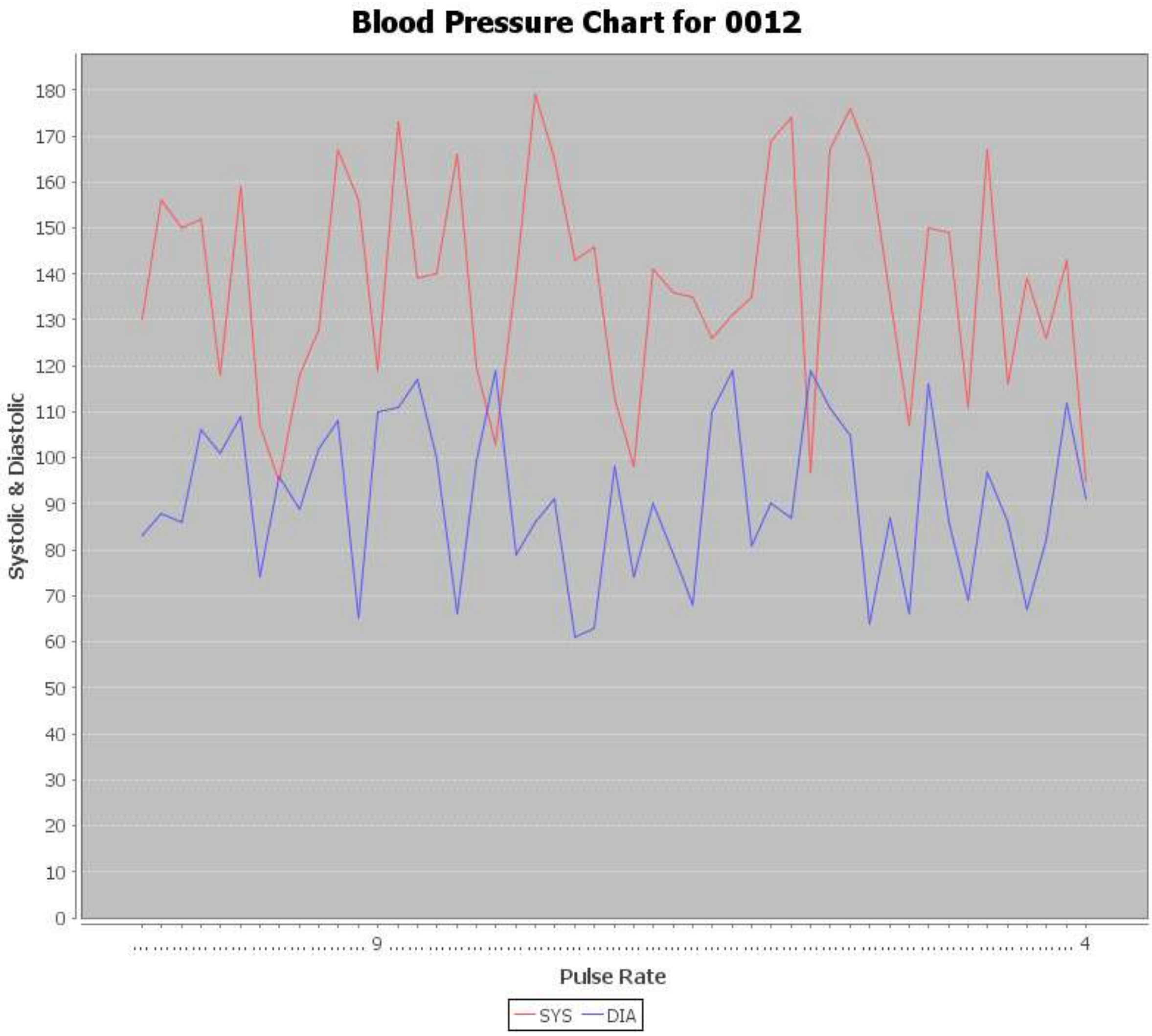
Patient ID	Name	Phone No.	SYS	DIA	Pulse	MAP	B.P	Suggestions	Date/Time
0012	Ngozi Okeke	08034044149	95	96	-1	96.0	95/96	Stage 1 Hypertension	Feb 24, 2012 8:52:40 AM
0012	Ngozi Okeke	08034044149	95	91	4	92.0	95/91	Stage 1 Hypertension	Feb 24, 2012 9:23:35 AM
0012	Ngozi Okeke	08034044149	97	119	-22	112.0	97/119	Stage 2 Hypertension.	Feb 24, 2012 9:10:40 AM
0012	Ngozi Okeke	08034044149	98	74	24	82.0	98/74	Desirable or Normal BP.	Feb 24, 2012 9:03:06 AM
0012	Ngozi Okeke	08034044149	103	119	-16	114.0	103/119	Stage 2 Hypertension.	Feb 24, 2012 8:59:19 AM
0012	Ngozi Okeke	08034044149	104	105	-1	105.0	104/105	Stage 2 Hypertension.	Feb 24, 2012 8:47:44 AM
0012	Ngozi Okeke	08034044149	107	74	33	85.0	107/74	Desirable or Normal BP.	Feb 24, 2012 8:47:11 AM
0012	Ngozi Okeke	08034044149	107	66	41	80.0	107/66	Desirable or Normal BP.	Feb 24, 2012 9:16:36 AM
0012	Ngozi Okeke	08034044149	110	69	41	83.0	110/69	Desirable or Normal BP.	Feb 24, 2012 9:14:59 AM
0012	Ngozi Okeke	08034044149	111	69	42	83.0	111/69	Desirable or Normal BP.	Feb 24, 2012 9:18:45 AM
0012	Ngozi Okeke	08034044149	113	98	15	103.0	113/98	Stage 1 Hypertension	Feb 24, 2012 9:02:34 AM
0012	Ngozi Okeke	08034044149	116	86	30	96.0	116/86	Prehypertension	Feb 24, 2012 9:19:49 AM

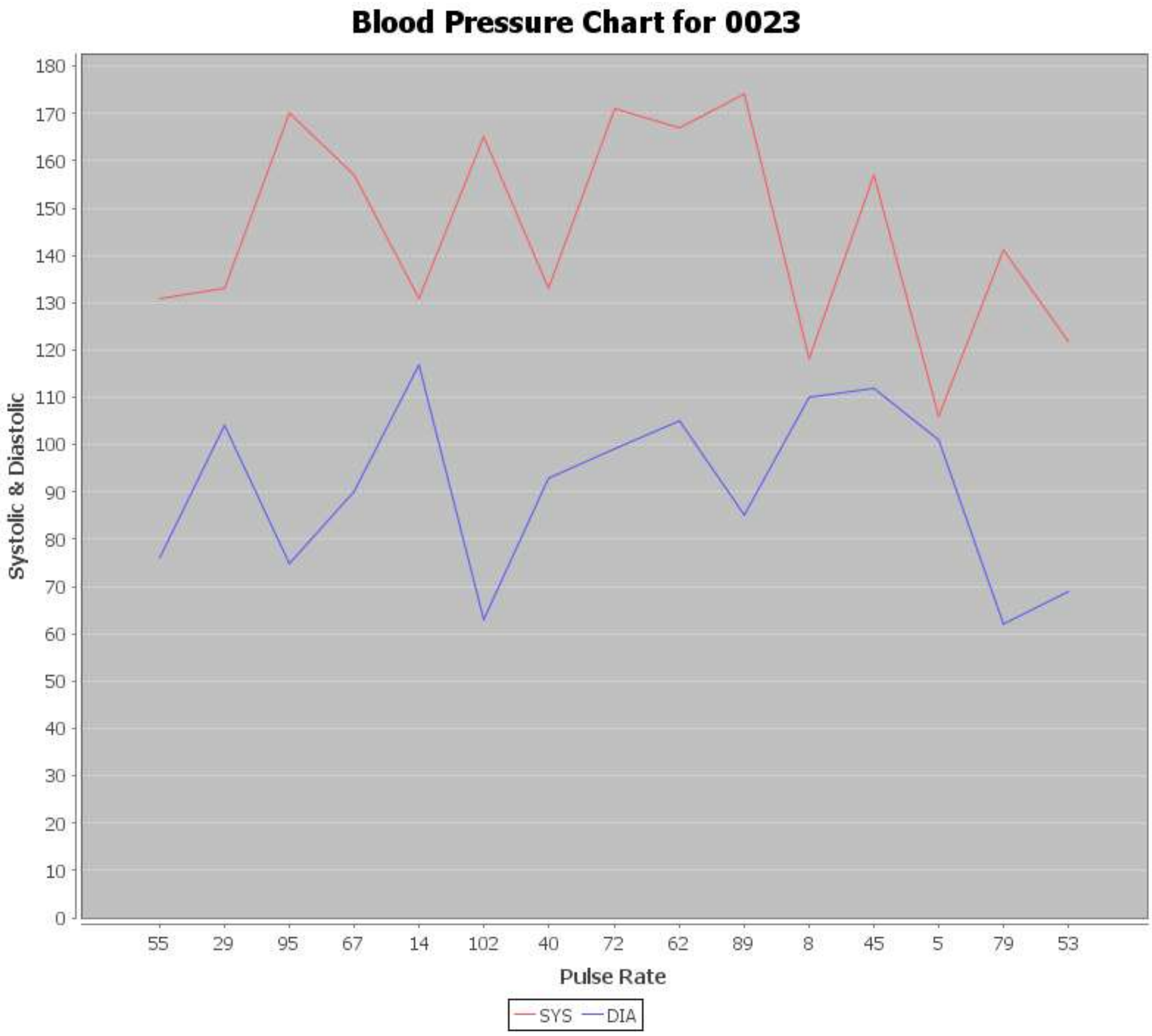

Automated Intelligent Hospital Ltd. - Windows Internet Explorer
Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Medical Equipments
Dynamic Staff Workloads
Pharmacy Portal Reports
Drug Dispensary Unit
Patient Registration
Patient Discharged
Upload Patient EMR
Hospital Administrator
About Us
Mobile Users
Network Queues of Mobile Users
Mobile Queue
as at :Fri Feb 24 09:29:11 GMT 2012
Patient ID | Patient Name | Category | Systolic Pressure | Diastolic Pressure | Pulse Pressure | Mean Arterial Pressure | Blood Pressure | Suggestions | No of Request Recieved
0012 | Ngozi Okoke | 1 | 137 | 89 | 48 | 105.0 | 137/89 | Prehypertension | 67
0023 | Ifeanyi Ndubuisi | 1 | 145 | 92 | 53 | 109.0 | 145/92 | Stage 1 Hypertension | 16
0018 | Ifeyinwa Orji | 2 | 143 | 89 | 53 | 107.0 | 143/89 | Stage 1 Hypertension | 11
QueuePatient ID,
View
Delete from Queue
Reset
No of Patient in Queue (n)..............:97
Time of Arrival in seconds (T).........:25
No of Channel.......................:1
Service Time (s)..................:10.0
Service Rate(mu)...................:0.1
Mean Patient arriving System(Lamda).:5.0
Traffic Intensity (p)...............:90.0
Local intranet | Protected Mode: Off
100%

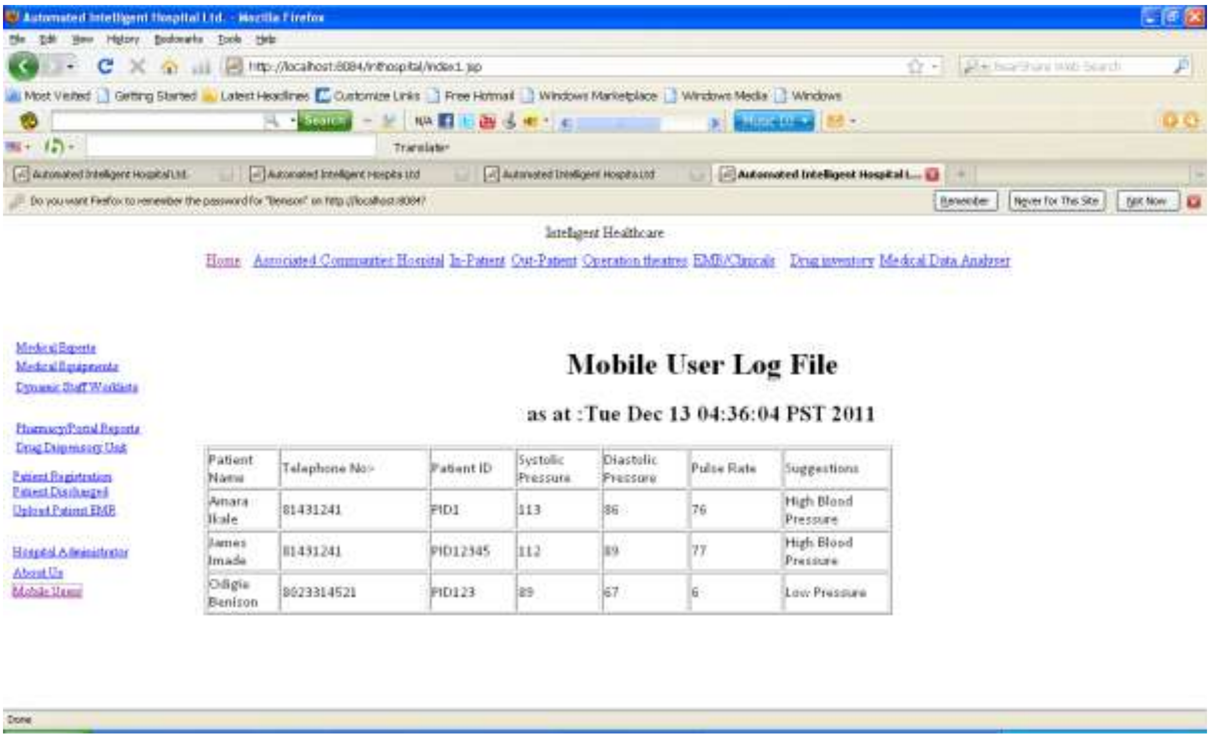

Blood Pressure Chart for 0018
Systolic & Diastolic
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
72 35 69 42 61 49 103 17 14 57 71 -15 80 37 29
Pulse Rate
SYS
DIA



Blood Pressure Chart for 0021
Systolic & Diastolic
Pulse Rate
SYS
DIA

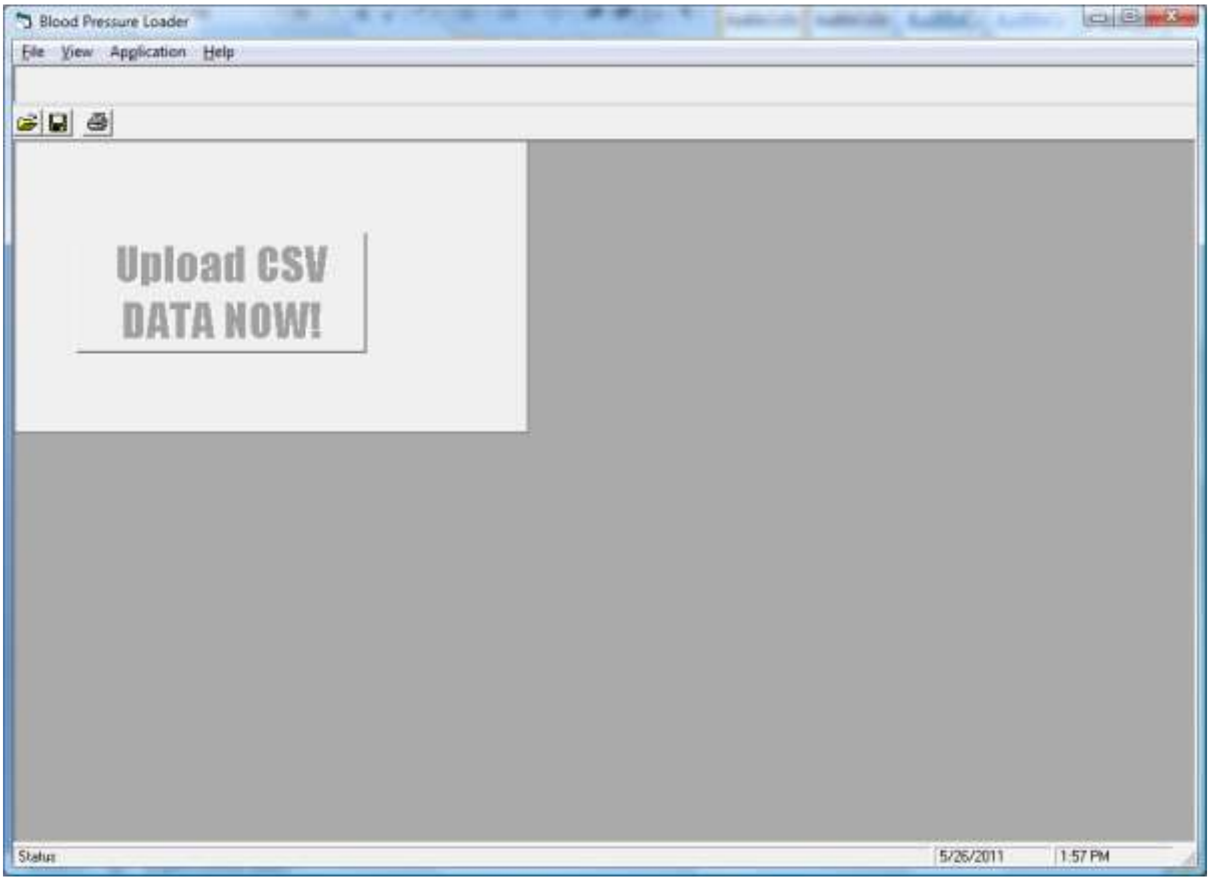

Blood Pressure Chart for 0012
Systolic & Diastolic
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
9
4
Pulse Rate
SYS
DIA

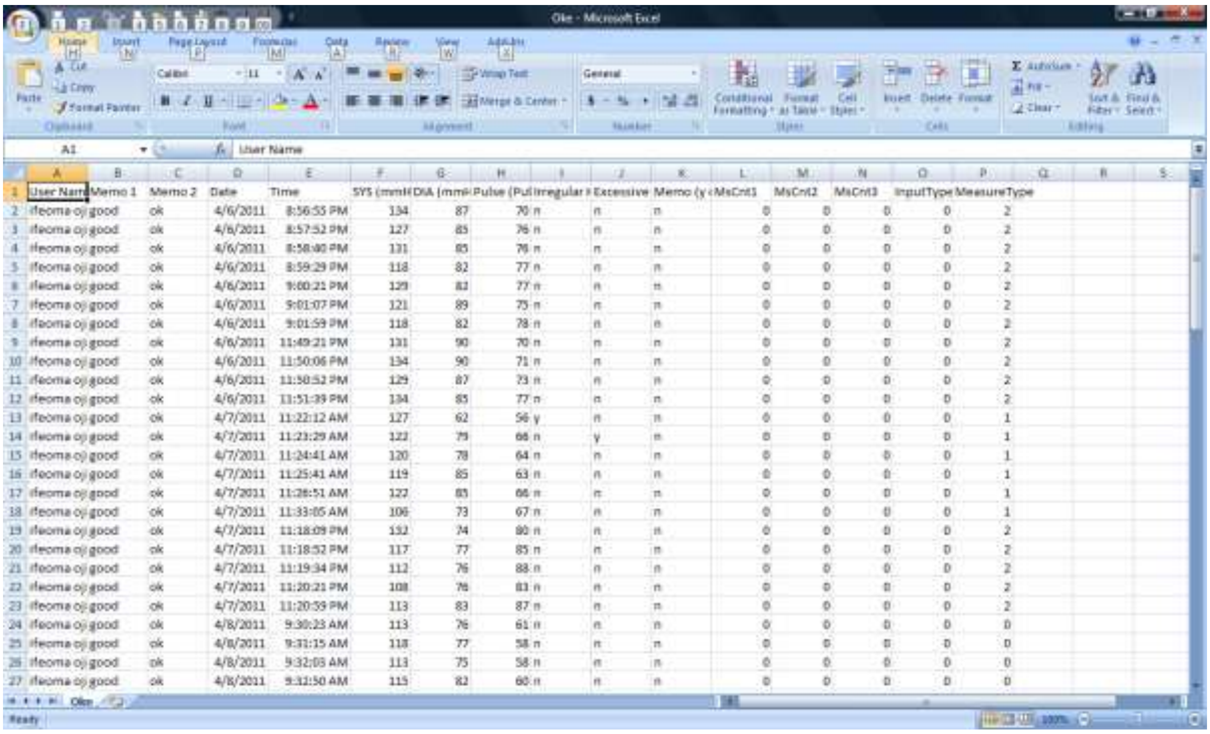




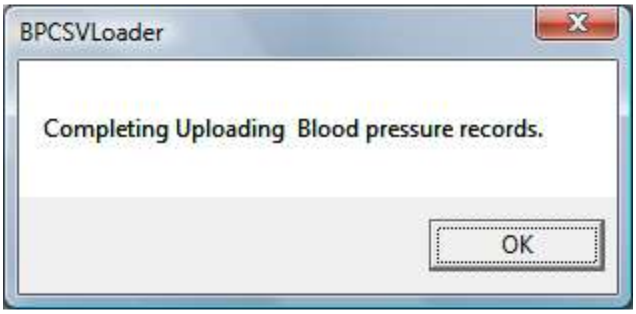
Patient Name	Telephone No>	Patient ID	Systolic Pressure	Diastolic Pressure	Pulse Rate	Suggestions
Amara Ikale	81431241	PID1	113	86	76	High Blood Pressure
James Imade	81431241	PID12345	112	89	77	High Blood Pressure
Odigie Benison	8023314521	PID123	89	67	6	Low Pressure

Program Output 1: Interfaces of Blood Pressure Loader

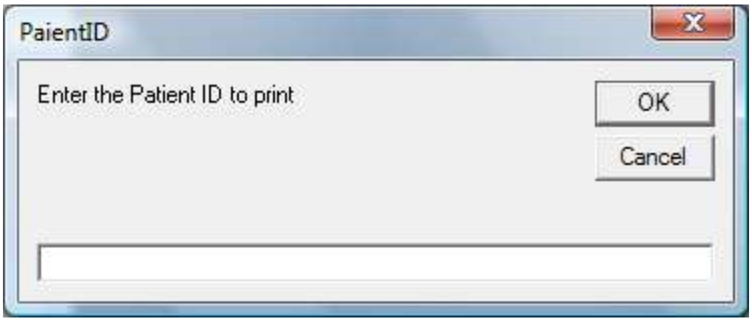




	User Name	Memo 1	Memo 2	Date	Time	SYS (mmH	DIA (mmH	Pulse (Pul	Irregular H	Excessive	Memo (y	MsCnt1	MsCnt2	MsCnt3	InputType	MeasureType
2	ifeoma oji	good	ok	4/6/2011	8:56:55 PM	134	87	70	n	n	n	0	0	0	0	2
3	ifeoma oji	good	ok	4/6/2011	8:57:52 PM	127	85	76	n	n	n	0	0	0	0	2
4	ifeoma oji	good	ok	4/6/2011	8:58:40 PM	131	85	76	n	n	n	0	0	0	0	2
5	ifeoma oji	good	ok	4/6/2011	8:59:29 PM	118	82	77	n	n	n	0	0	0	0	2
6	ifeoma oji	good	ok	4/6/2011	9:00:21 PM	129	82	77	n	n	n	0	0	0	0	2
7	ifeoma oji	good	ok	4/6/2011	9:01:07 PM	121	89	75	n	n	n	0	0	0	0	2
8	ifeoma oji	good	ok	4/6/2011	9:01:59 PM	118	82	78	n	n	n	0	0	0	0	2
9	ifeoma oji	good	ok	4/6/2011	11:49:21 PM	131	90	70	n	n	n	0	0	0	0	2
10	ifeoma oji	good	ok	4/6/2011	11:50:06 PM	134	90	71	n	n	n	0	0	0	0	2
11	ifeoma oji	good	ok	4/6/2011	11:50:52 PM	129	87	73	n	n	n	0	0	0	0	2
12	ifeoma oji	good	ok	4/6/2011	11:51:39 PM	134	85	77	n	n	n	0	0	0	0	2
13	ifeoma oji	good	ok	4/7/2011	11:22:12 AM	127	62	56	y	n	n	0	0	0	0	1
14	ifeoma oji	good	ok	4/7/2011	11:23:29 AM	122	79	68	n	y	n	0	0	0	0	1
15	ifeoma oji	good	ok	4/7/2011	11:24:41 AM	120	78	64	n	n	n	0	0	0	0	1
16	ifeoma oji	good	ok	4/7/2011	11:25:41 AM	119	85	63	n	n	n	0	0	0	0	1
17	ifeoma oji	good	ok	4/7/2011	11:28:51 AM	122	85	66	n	n	n	0	0	0	0	1
18	ifeoma oji	good	ok	4/7/2011	11:33:05 AM	106	73	67	n	n	n	0	0	0	0	1
19	ifeoma oji	good	ok	4/7/2011	11:18:09 PM	132	74	80	n	n	n	0	0	0	0	2
20	ifeoma oji	good	ok	4/7/2011	11:18:52 PM	117	77	85	n	n	n	0	0	0	0	2
21	ifeoma oji	good	ok	4/7/2011	11:19:34 PM	112	76	88	n	n	n	0	0	0	0	2
22	ifeoma oji	good	ok	4/7/2011	11:20:21 PM	108	76	83	n	n	n	0	0	0	0	2
23	ifeoma oji	good	ok	4/7/2011	11:20:59 PM	113	83	87	n	n	n	0	0	0	0	2
24	ifeoma oji	good	ok	4/8/2011	9:30:23 AM	113	76	61	n	n	n	0	0	0	0	0
25	ifeoma oji	good	ok	4/8/2011	9:31:15 AM	118	77	58	n	n	n	0	0	0	0	0
26	ifeoma oji	good	ok	4/8/2011	9:32:03 AM	113	75	58	n	n	n	0	0	0	0	0
27	ifeoma oji	good	ok	4/8/2011	9:32:50 AM	115	82	60	n	n	n	0	0	0	0	0


BPCSVLoader
Completing Uploading Blood pressure records.
OK


Doctor Alert
Send Txt Message to a docor!
Yes
No


PaientID
Enter the Patient ID to print
OK
Cancel

Automatd Intelligent Hospital LTD

Zoom 100%

Automated Intelligent Hospital LTD

Patient Blood Pressure Report

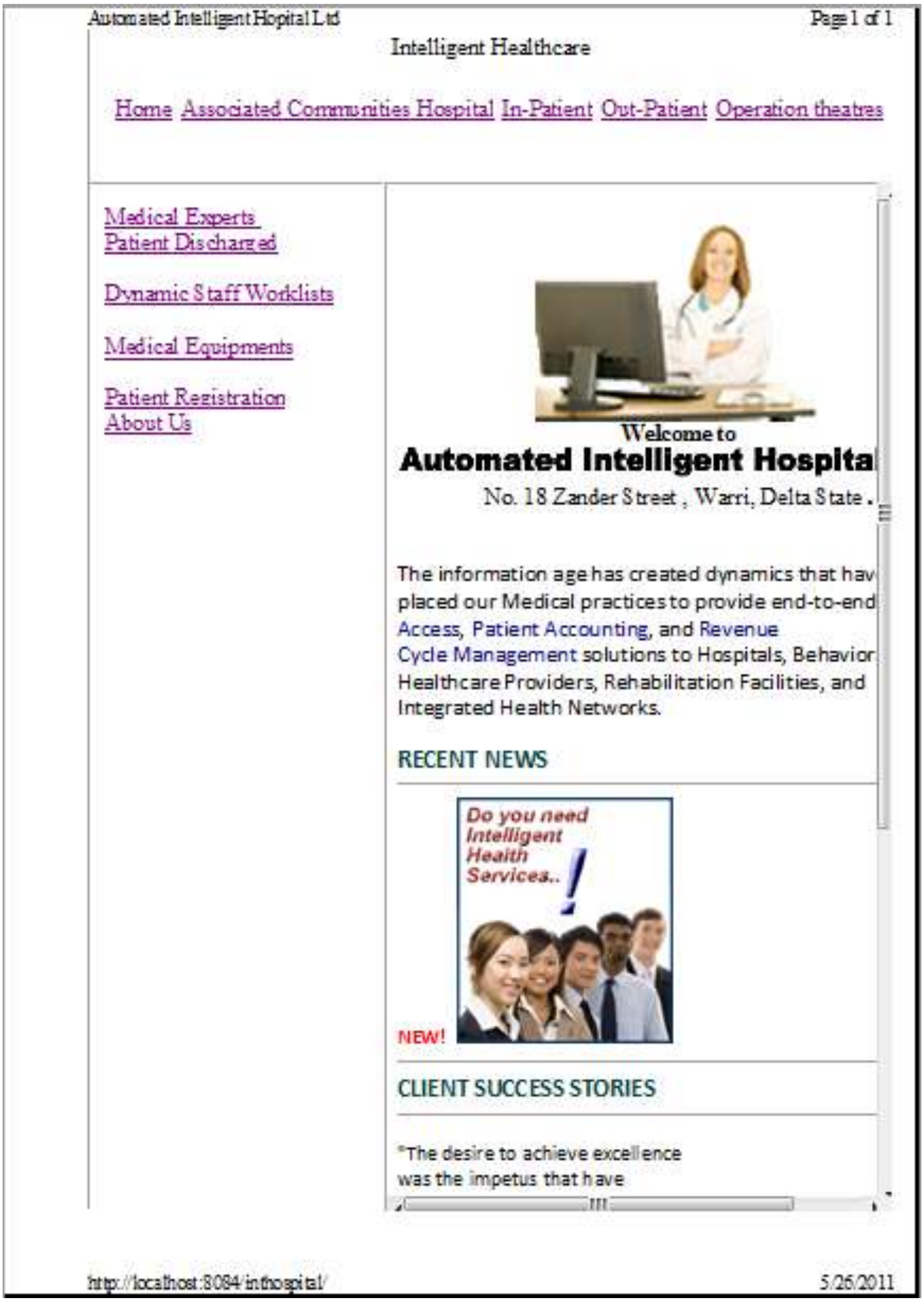
5/26/2011

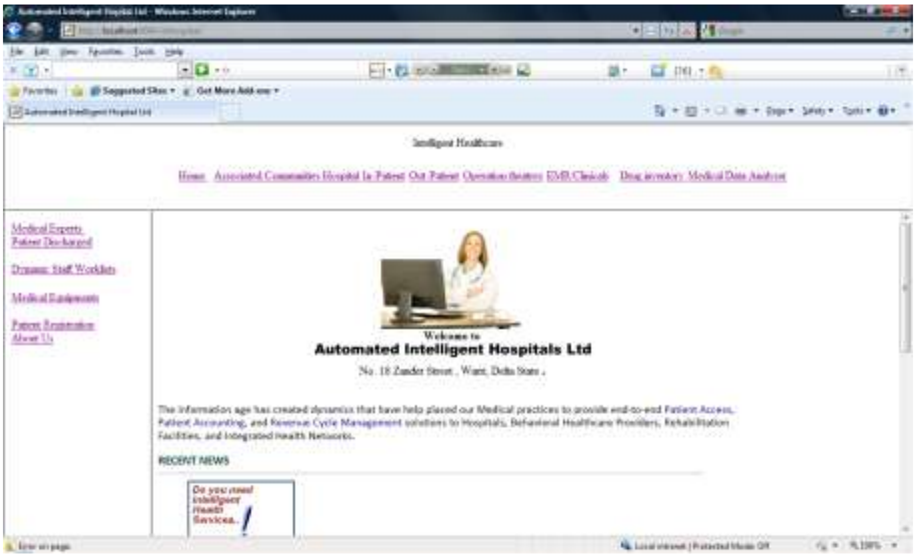
ifeoma oji

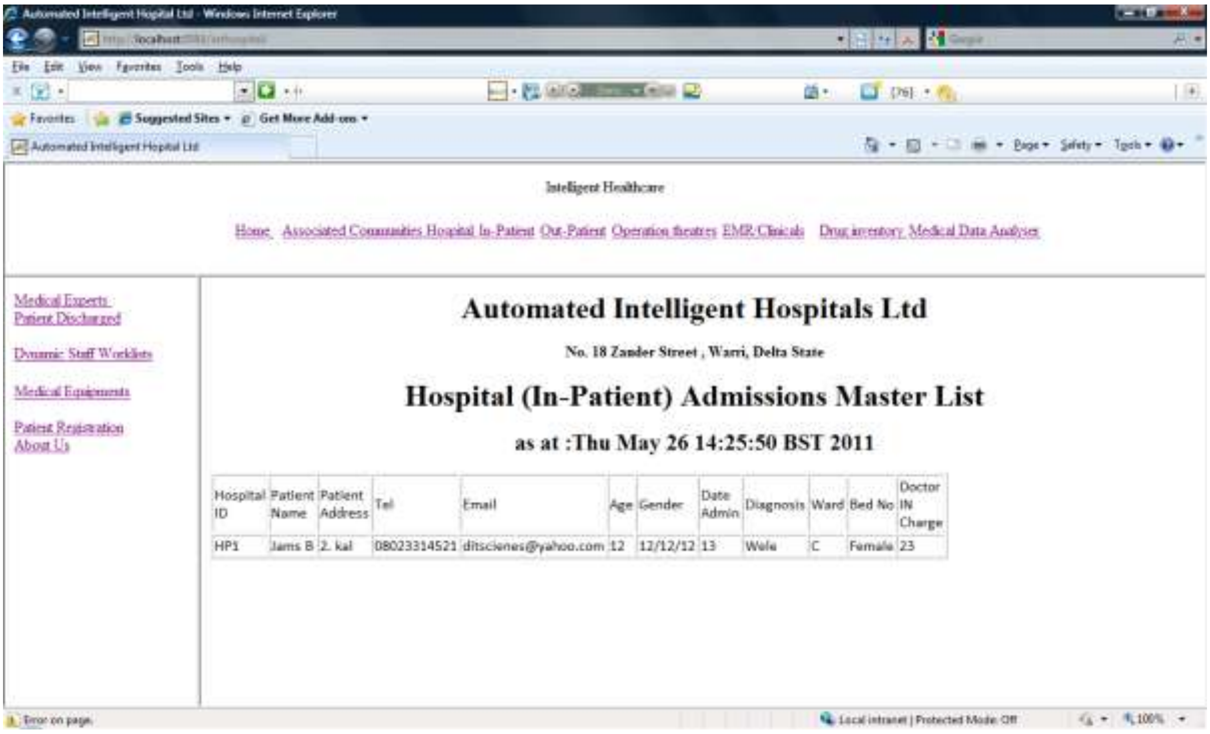
Date	Time	Sys	DIA	Irr. Heart	Pulse	Excess Move	MsCnt1	MsCnt 2	MsCnt3	Input Type	Meas. Type	Memo
4/6/2011	8:57:52 PM	127	85	n	76	n	0	0	0	0	2	n
4/6/2011	11:50:06 PM	134	90	n	71	n	0	0	0	0	2	n
4/6/2011	11:49:21 PM	131	90	n	70	n	0	0	0	0	2	n
4/6/2011	9:01:59 PM	118	82	n	78	n	0	0	0	0	2	n
4/6/2011	9:01:07 PM	121	89	n	75	n	0	0	0	0	2	n
4/6/2011	9:00:21 PM	129	82	n	77	n	0	0	0	0	2	n
4/6/2011	8:59:29 PM	118	82	n	77	n	0	0	0	0	2	n
4/6/2011	8:58:40 PM	131	85	n	76	n	0	0	0	0	2	n
4/6/2011	8:57:52 PM	127	85	n	76	n	0	0	0	0	2	n
4/6/2011	11:51:39 PM	134	85	n	77	n	0	0	0	0	2	n
4/6/2011	8:56:55 PM	134	87	n	70	n	0	0	0	0	2	n
4/6/2011	11:51:39 PM	134	85	n	77	n	0	0	0	0	2	n
4/6/2011	8:58:40 PM	131	85	n	76	n	0	0	0	0	2	n
4/6/2011	8:59:29 PM	118	82	n	77	n	0	0	0	0	2	n
4/6/2011	9:00:21 PM	129	82	n	77	n	0	0	0	0	2	n
4/6/2011	9:01:07 PM	121	89	n	75	n	0	0	0	0	2	n
4/6/2011	9:01:59 PM	118	82	n	78	n	0	0	0	0	2	n
4/6/2011	11:49:21 PM	131	90	n	70	n	0	0	0	0	2	n
4/6/2011	11:50:06 PM	134	90	n	71	n	0	0	0	0	2	n
4/6/2011	11:50:52 PM	129	87	n	73	n	0	0	0	0	2	n
4/6/2011	8:56:55 PM	134	87	n	70	n	0	0	0	0	2	n
4/6/2011	11:51:39 PM	134	85	n	77	n	0	0	0	0	2	n

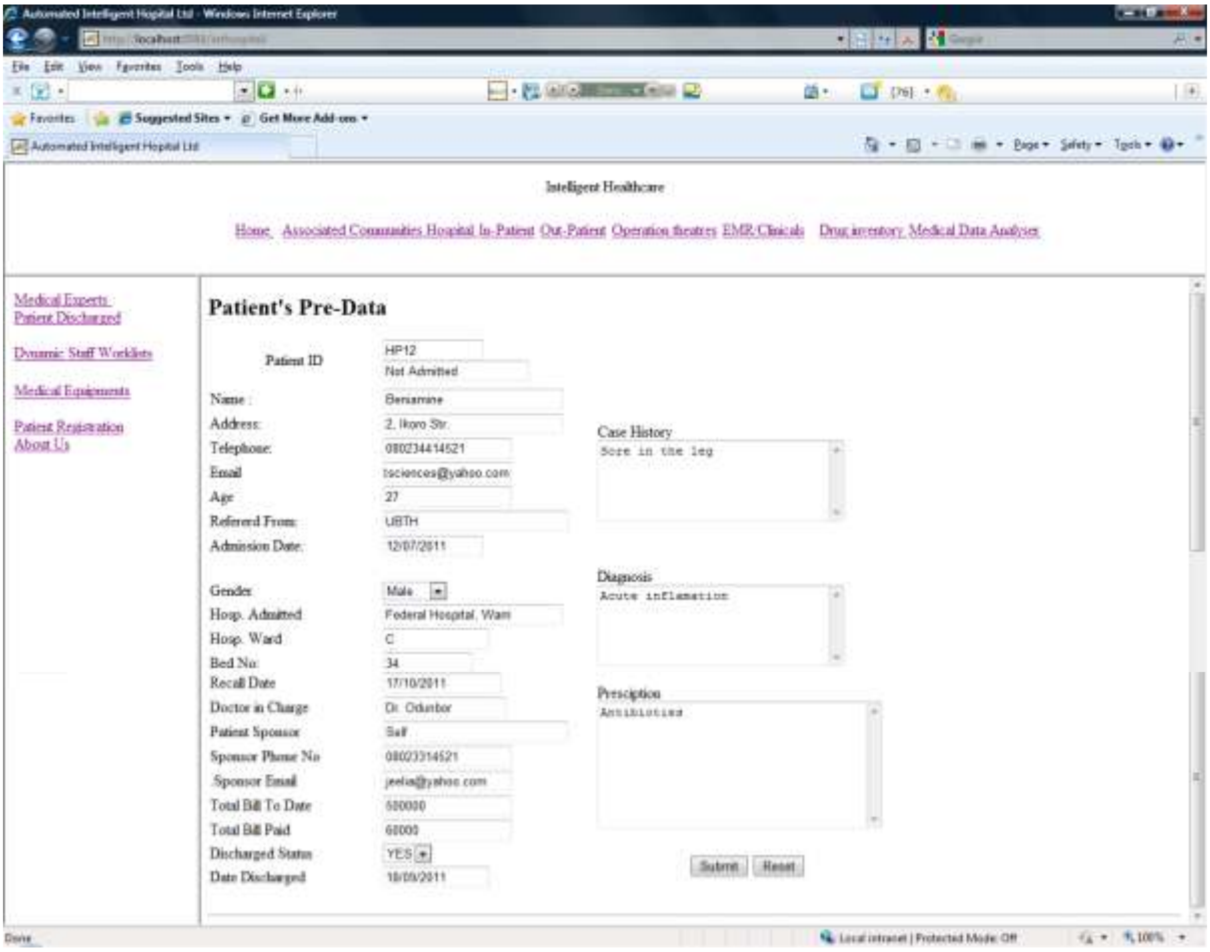
Pages: 1

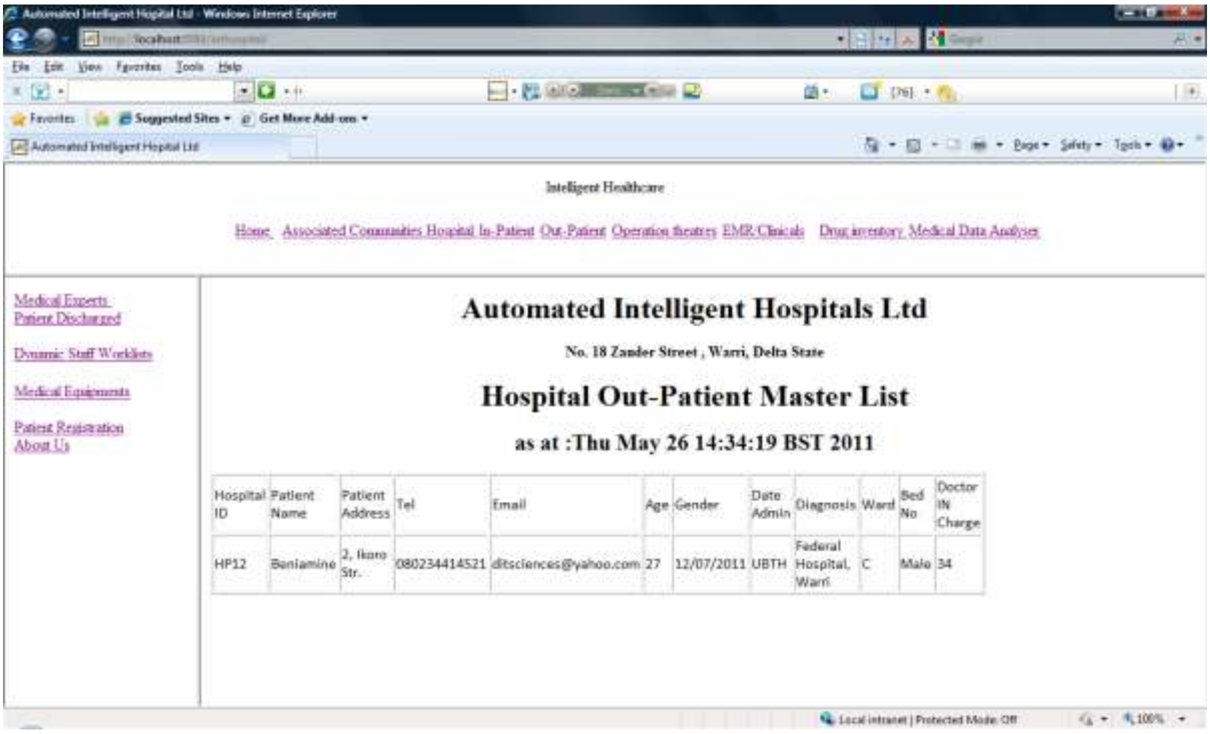
Program output 2: Java Application Interfaces

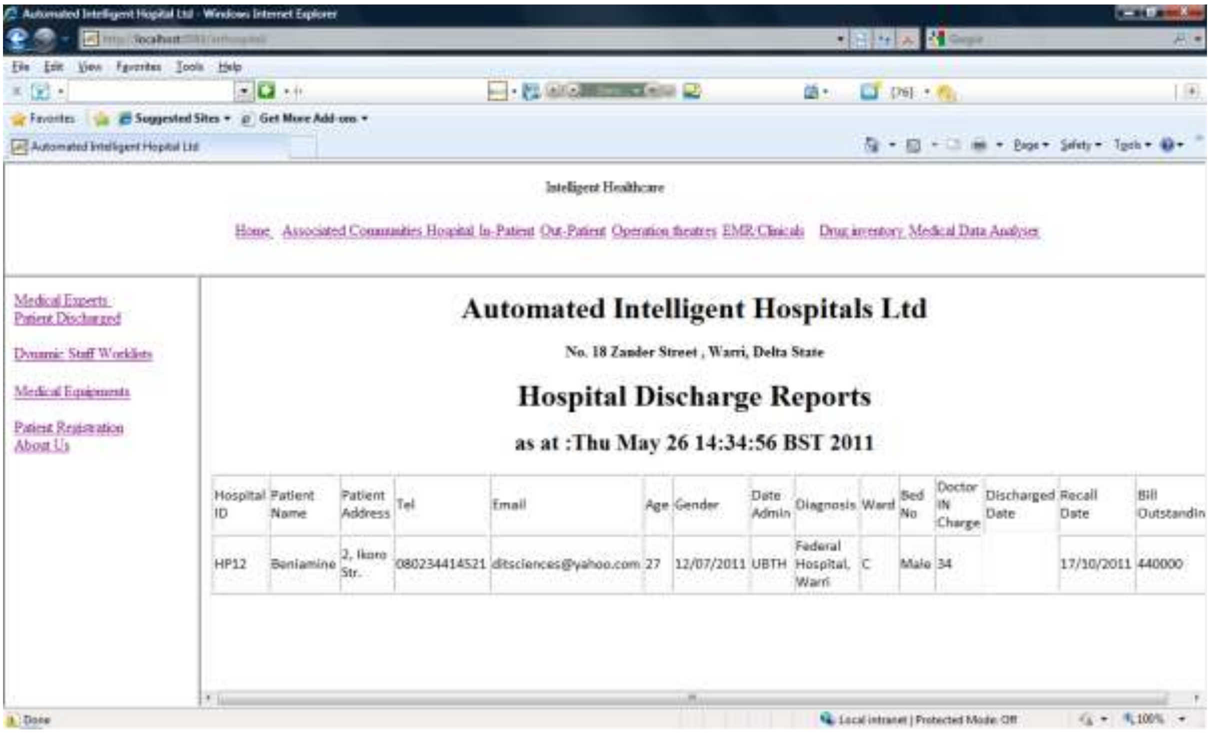


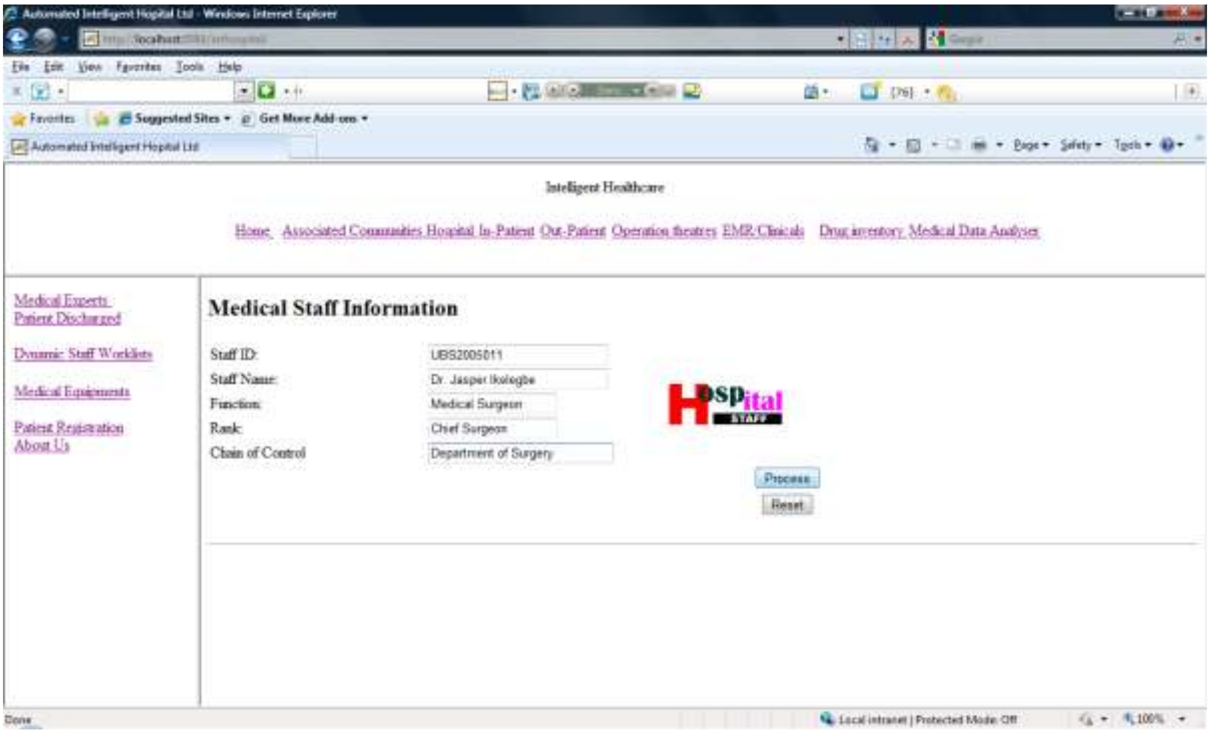

Intelligent Healthcare
Welcome to
Automated Intelligent Hospitals Ltd
RECENT NEWS

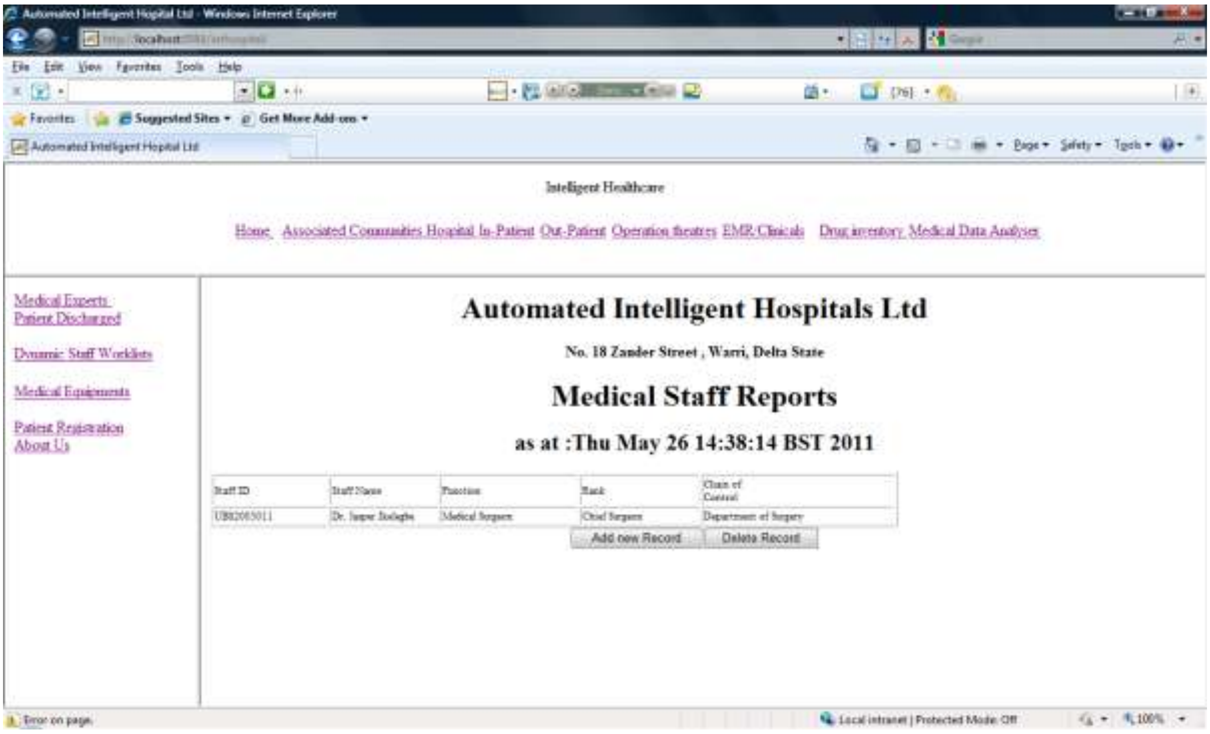

Automated Intelligent Hospital Ltd - Windows Internet Explorer
Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Automated Intelligent Hospitals Ltd
No. 18 Zander Street , Warri, Delta State
Hospital (In-Patient) Admissions Master List
as at :Thu May 26 14:25:50 BST 2011
Hospital ID
Patient Name
Patient Address
Tel
Email
Age
Gender
Date Admin
Diagnosis
Ward
Bed No
Doctor IN Charge
HP1
Jams B
2. kal
08023314521
ditscienes@yahoo.com
12
12/12/12
13
Wele
C
Female
23
Local intranet | Protected Mode: Off
100%

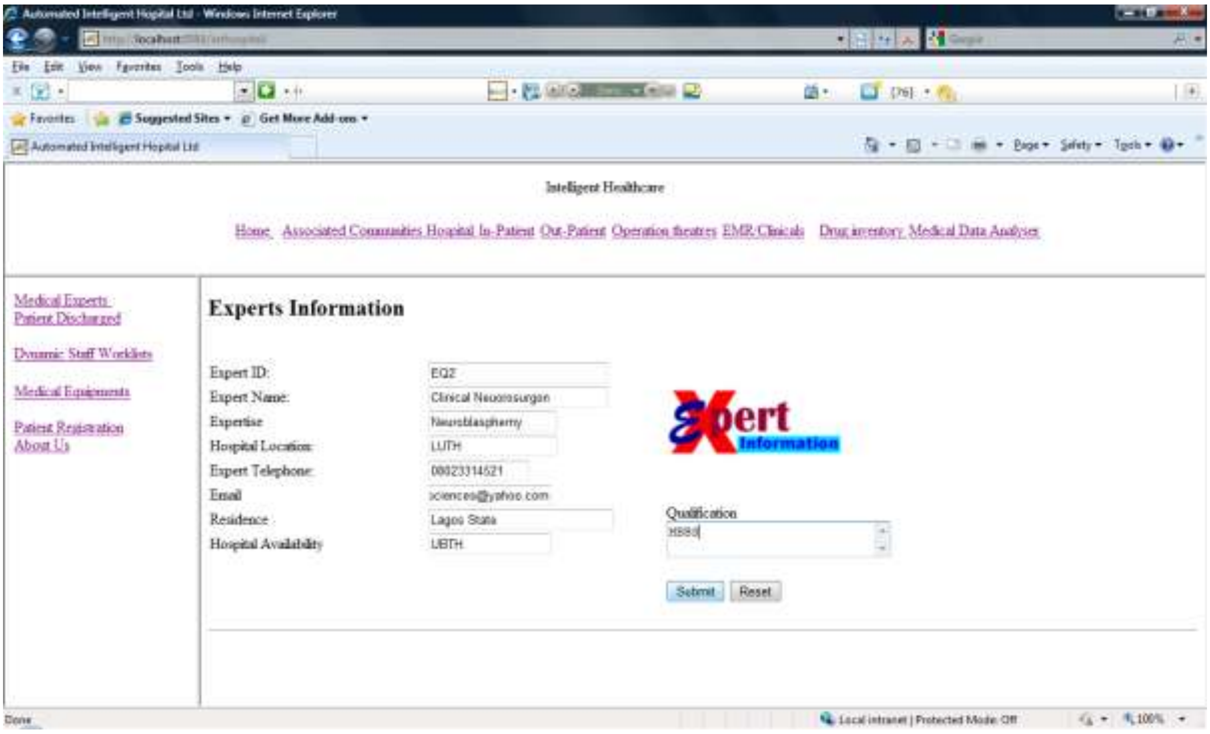

Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Patient's Pre-Data
Patient ID
HP12
Not Admitted
Name :
Beniamine
Address
2, Ikoro Str.
Telephone
080234414521
Email
Age
27
Referred From
UBTH
Admission Date
12/07/2011
Gender
Male
Hosp. Admitted
Federal Hospital, Warri
Hosp. Ward
C
Bed No
34
Recall Date
17/10/2011
Doctor in Charge
Dr. Odunbor
Patient Sponsor
Self
Sponsor Phone No
Sponsor Email
Total Bill To Date
Total Bill Paid
Discharged Status
YES
Date Discharged
Case History
Sore in the leg
Diagnosis
Acute inflamation
Presciption
Antibiotics
Submit
Reset

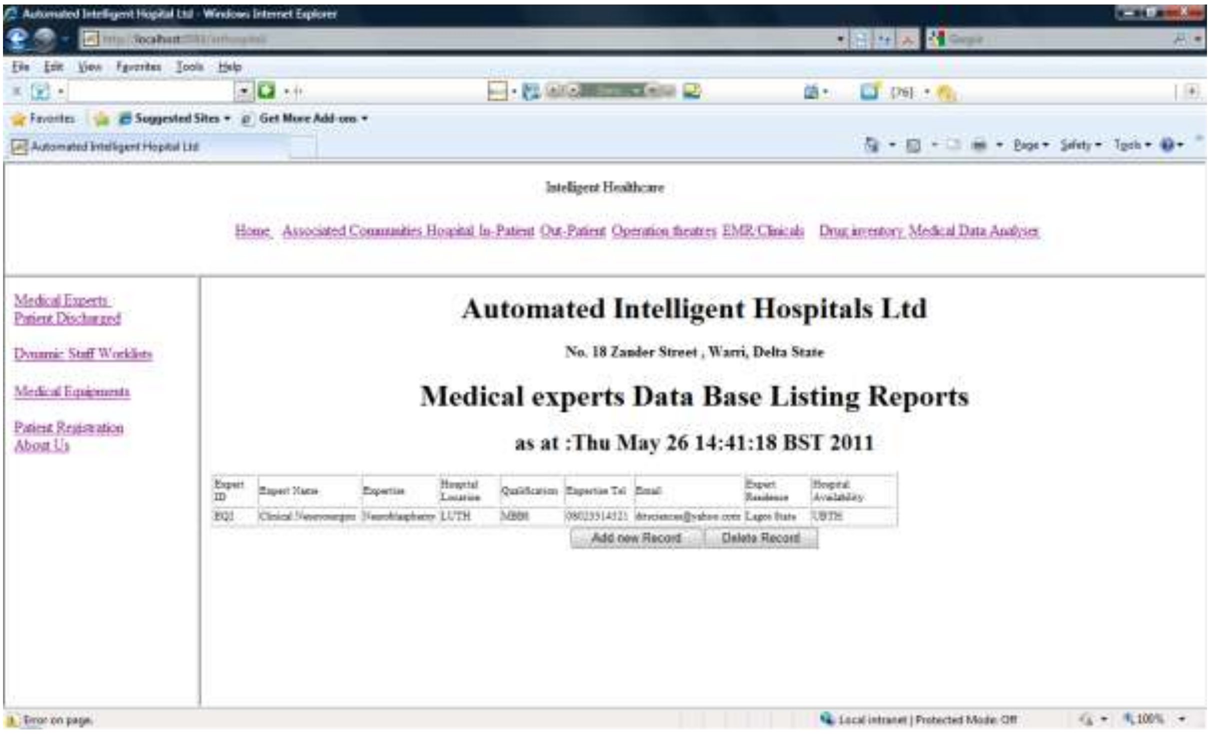

Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Automated Intelligent Hospitals Ltd
No. 18 Zander Street , Warri, Delta State
Hospital Out-Patient Master List
as at :Thu May 26 14:34:19 BST 2011
Hospital ID
Patient Name
Patient Address
Tel
Email
Age
Gender
Date Admin
Diagnosis
Ward
Bed No
Doctor IN Charge
HP12
Beniamine
2, Ikoro Str.
080234414521
ditsciences@yahoo.com
27
12/07/2011
UBTH
Federal Hospital, Warri
C
Male
34

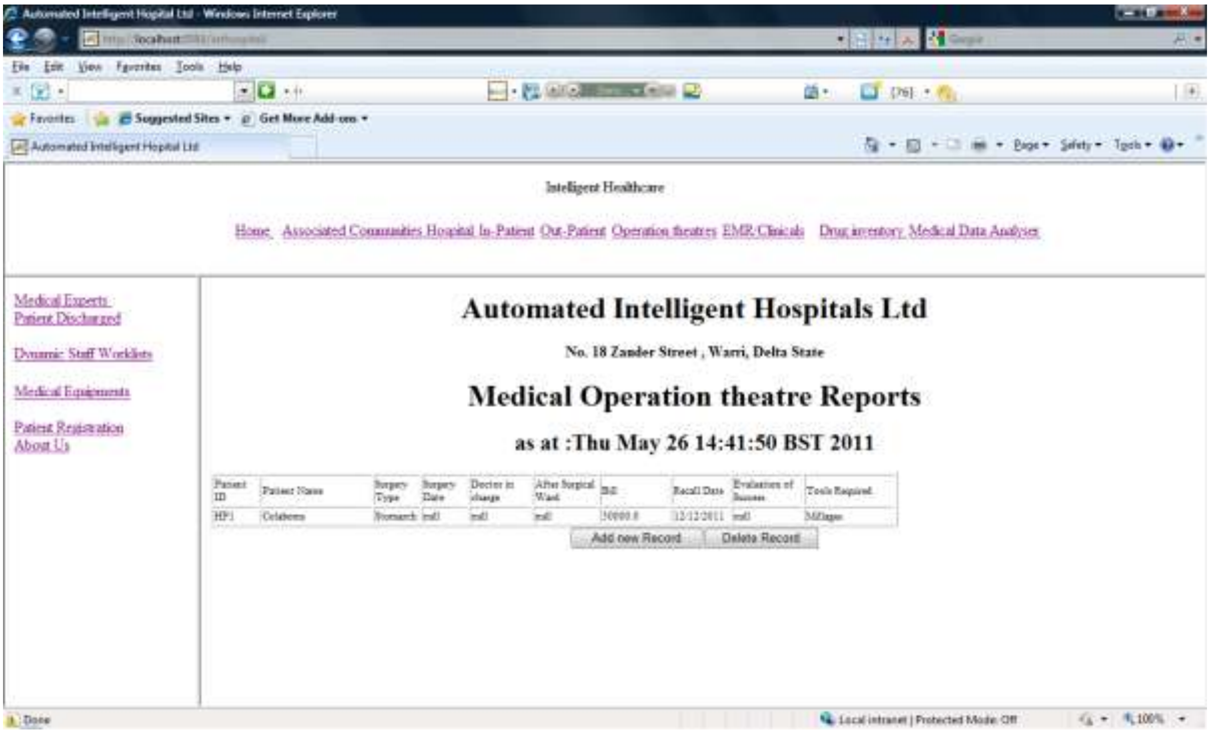

Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Automated Intelligent Hospitals Ltd
No. 18 Zander Street , Warri, Delta State
Hospital Discharge Reports
as at :Thu May 26 14:34:56 BST 2011
Hospital ID | Patient Name | Patient Address | Tel | Email | Age | Gender | Date Admin | Diagnosis | Ward | Bed No | Doctor IN Charge | Discharged Date | Recall Date | Bill Outstandin
HP12 | Beniamine | 2, Ikoro Str. | 080234414521 | ditsciences@yahoo.com | 27 | 12/07/2011 | UBTH | Federal Hospital, Warri | C | Male | 34 | | 17/10/2011 | 440000
Local intranet | Protected Mode: Off
100%

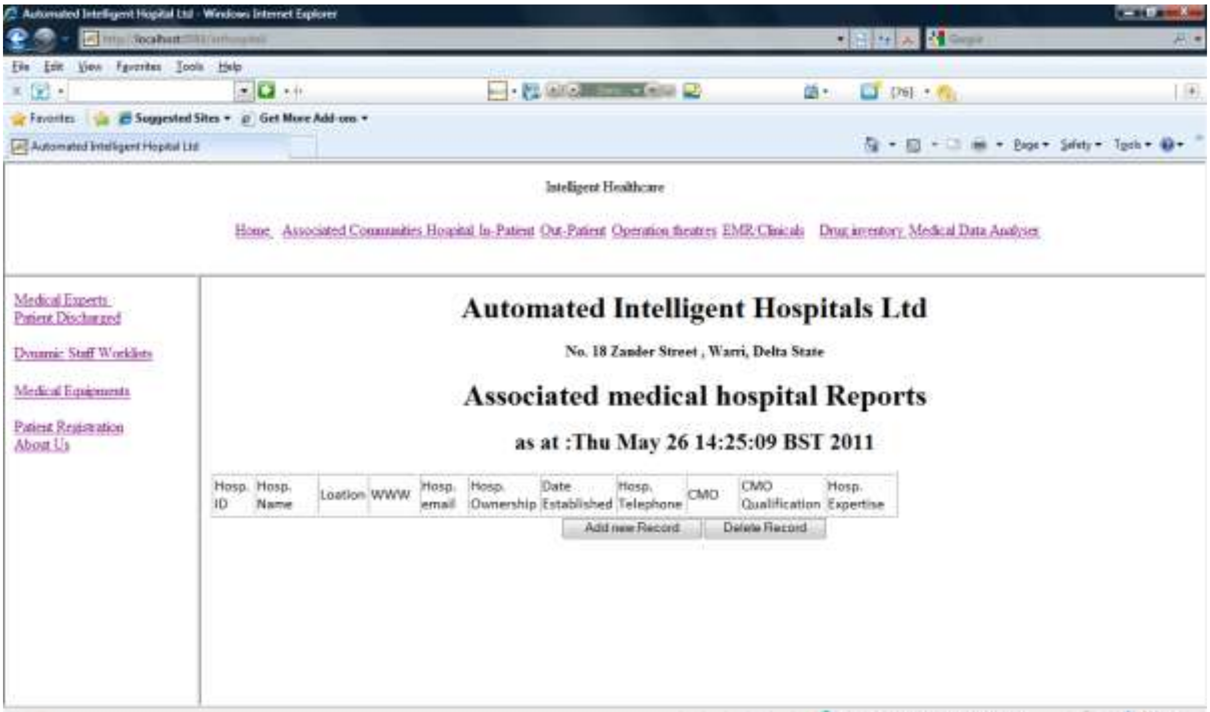

Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Medical Staff Information
Staff ID: UBS2005011
Staff Name: Dr. Jasper Ikelegbe
Function: Medical Surgeon
Rank: Chief Surgeon
Chain of Control: Department of Surgery
Hospital
STAFF
Process
Reset
Local intranet | Protected Mode: Off
100%

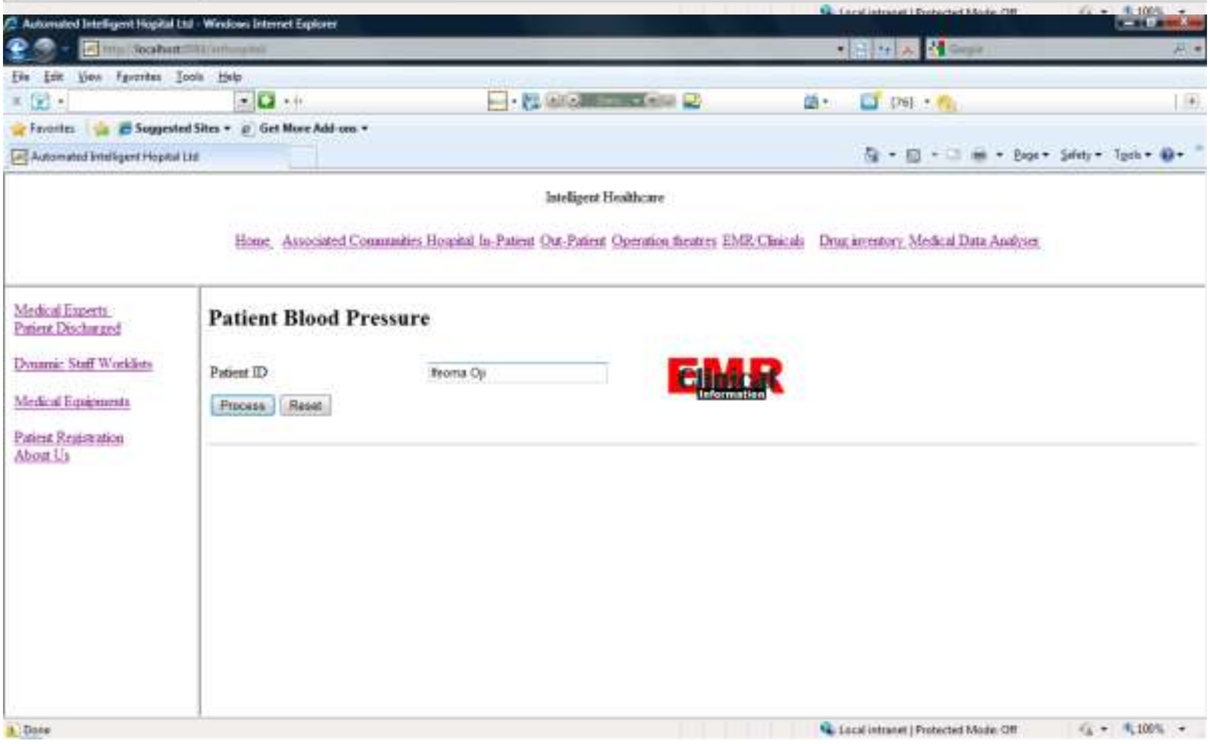

Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Automated Intelligent Hospitals Ltd
No. 18 Zander Street , Warri, Delta State
Medical Staff Reports
as at :Thu May 26 14:38:14 BST 2011
Add new Record
Delete Record

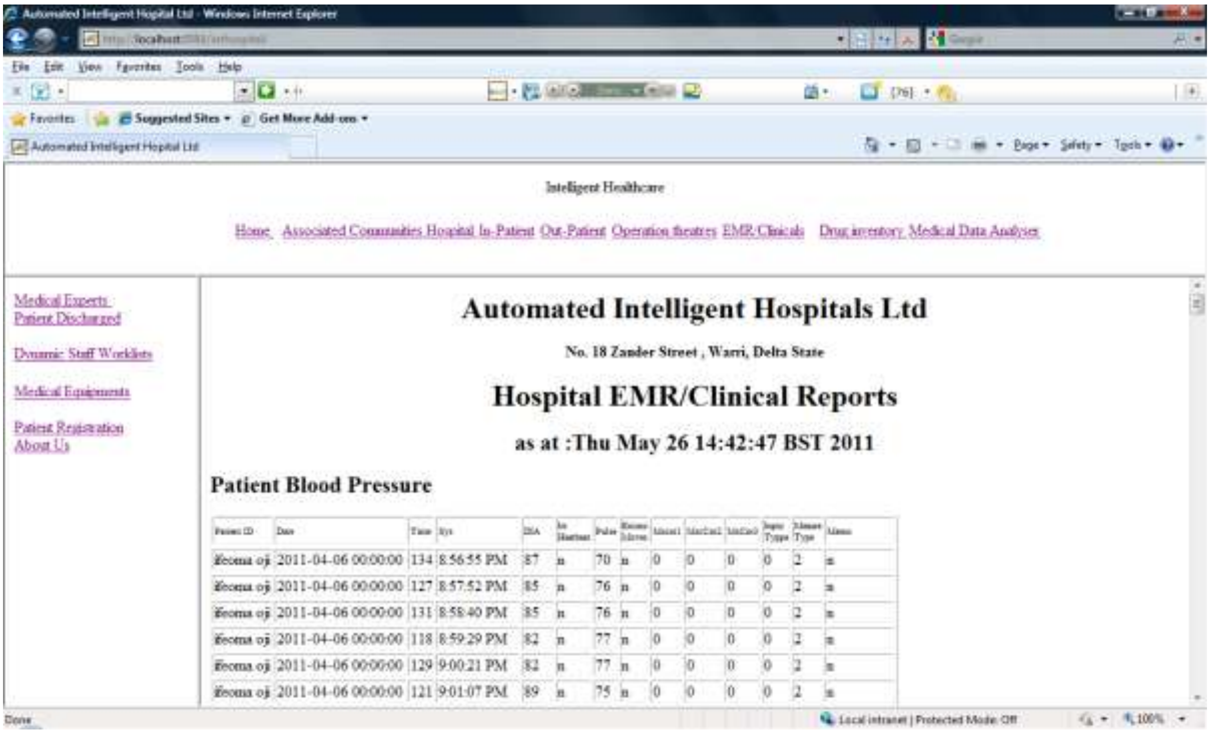

Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Experts Information
Expert ID:
Expert Name:
Expertise
Hospital Location
Expert Telephone
Email
Residence
Hospital Availability
Qualification
Submit
Reset


Automated Intelligent Hopital Ltd - Windows Internet Explorer
Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Automated Intelligent Hospitals Ltd
No. 18 Zander Street , Warri, Delta State
Medical experts Data Base Listing Reports
as at :Thu May 26 14:41:18 BST 2011
Expert ID
Expert Name
Expertise
Hospital Location
Qualification
Expertise Tel
Email
Expert Residence
Hospital Availability
LUTH
MBBS
Lagos State
UBTH
Add new Record
Delete Record
Error on page.
Local intranet | Protected Mode: Off
100%


Automated Intelligent Hopital Ltd - Windows Internet Explorer
Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Automated Intelligent Hospitals Ltd
No. 18 Zander Street , Warri, Delta State
Medical Operation theatre Reports
as at :Thu May 26 14:41:50 BST 2011
Patient ID
Patient Name
Surgery Type
Surgery Date
Doctor in charge
After Surgical Ward
Bill
Recall Date
Evaluation of Success
Tools Required
HP1
Celebrere
Stomach
null
12/12/2011
Add new Record
Delete Record
Done
Local intranet | Protected Mode: Off
100%


Automated Intelligent Hopital Ltd - Windows Internet Explorer
Inteligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Automated Intelligent Hospitals Ltd
No. 18 Zander Street , Warri, Delta State
Associated medical hospital Reports
as at :Thu May 26 14:25:09 BST 2011
Hosp. ID
Hosp. Name
Loation
WWW
Hosp. email
Hosp. Ownership
Date Established
Hosp. Telephone
CMO
CMO Qualification
Hosp. Expertise
Add new Record
Delete Record


Automated Intelligent Hopital Ltd - Windows Internet Explorer
Inteligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Patient Blood Pressure
Patient ID
Process
Reset
EMR
Done
Local intranet | Protected Mode: Off



Intelligent Healthcare

Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser

Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us

Automated Intelligent Hospitals Ltd

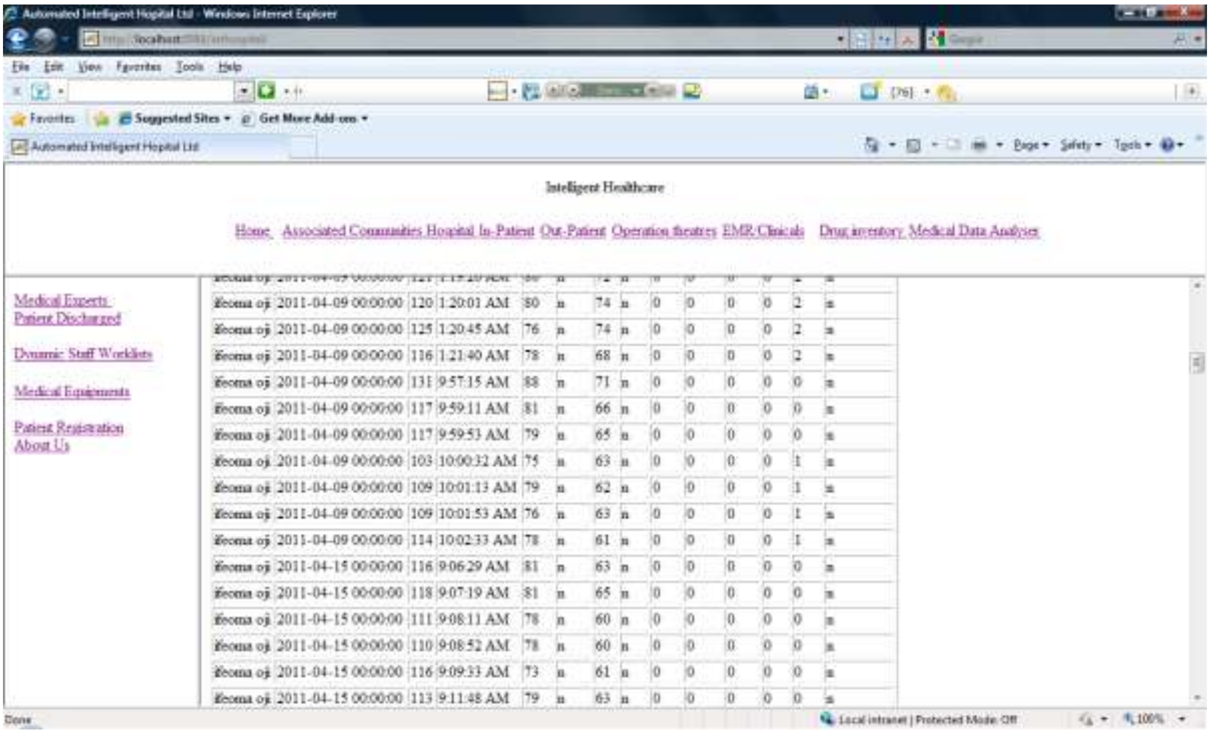
No. 18 Zander Street , Warri, Delta State

Hospital EMR/Clinical Reports

as at :Thu May 26 14:42:47 BST 2011

Patient Blood Pressure

Patient ID	Date	Time	Sys	DIA	[illegible]	Pulse	[illegible]	[illegible]	[illegible]	[illegible]	Input Type	Measure Type	Memo
ifeoma oji	2011-04-06 00:00:00	134	8:56:55 PM	87	n	70	n	0	0	0	0	2	n
ifeoma oji	2011-04-06 00:00:00	127	8:57:52 PM	85	n	76	n	0	0	0	0	2	n
ifeoma oji	2011-04-06 00:00:00	131	8:58:40 PM	85	n	76	n	0	0	0	0	2	n
ifeoma oji	2011-04-06 00:00:00	118	8:59:29 PM	82	n	77	n	0	0	0	0	2	n
ifeoma oji	2011-04-06 00:00:00	129	9:00:21 PM	82	n	77	n	0	0	0	0	2	n
ifeoma oji	2011-04-06 00:00:00	121	9:01:07 PM	89	n	75	n	0	0	0	0	2	n

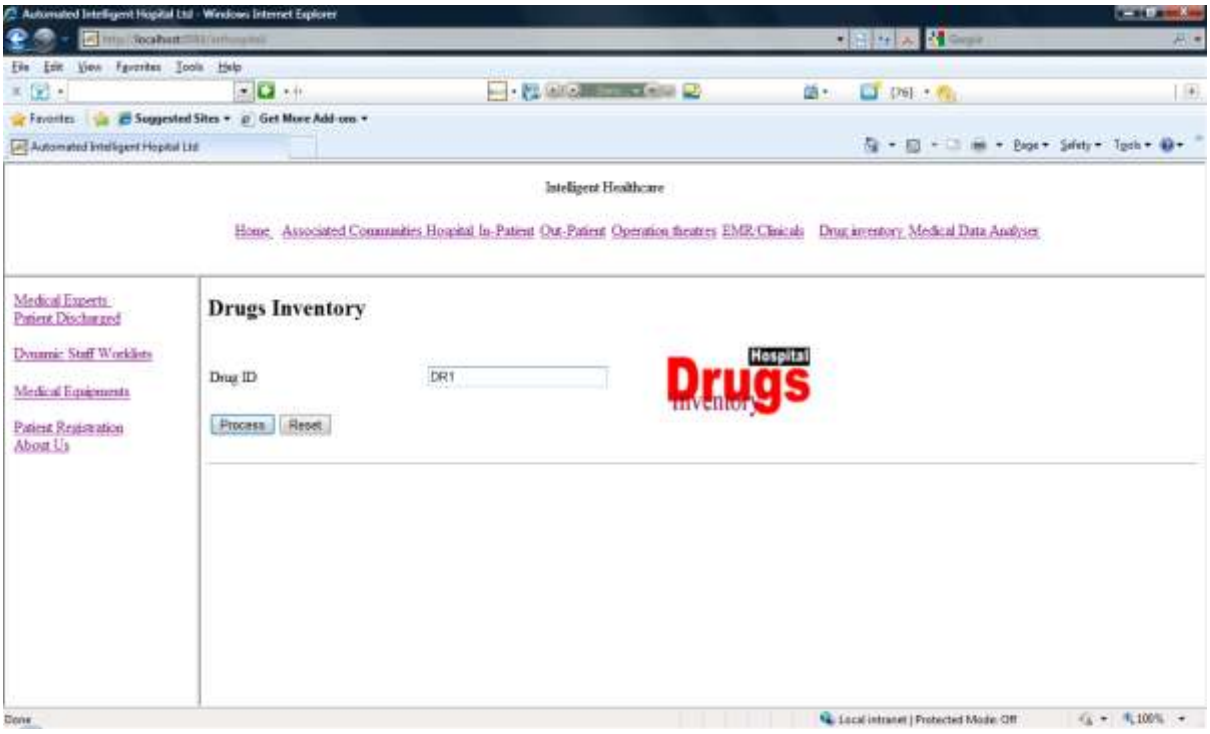


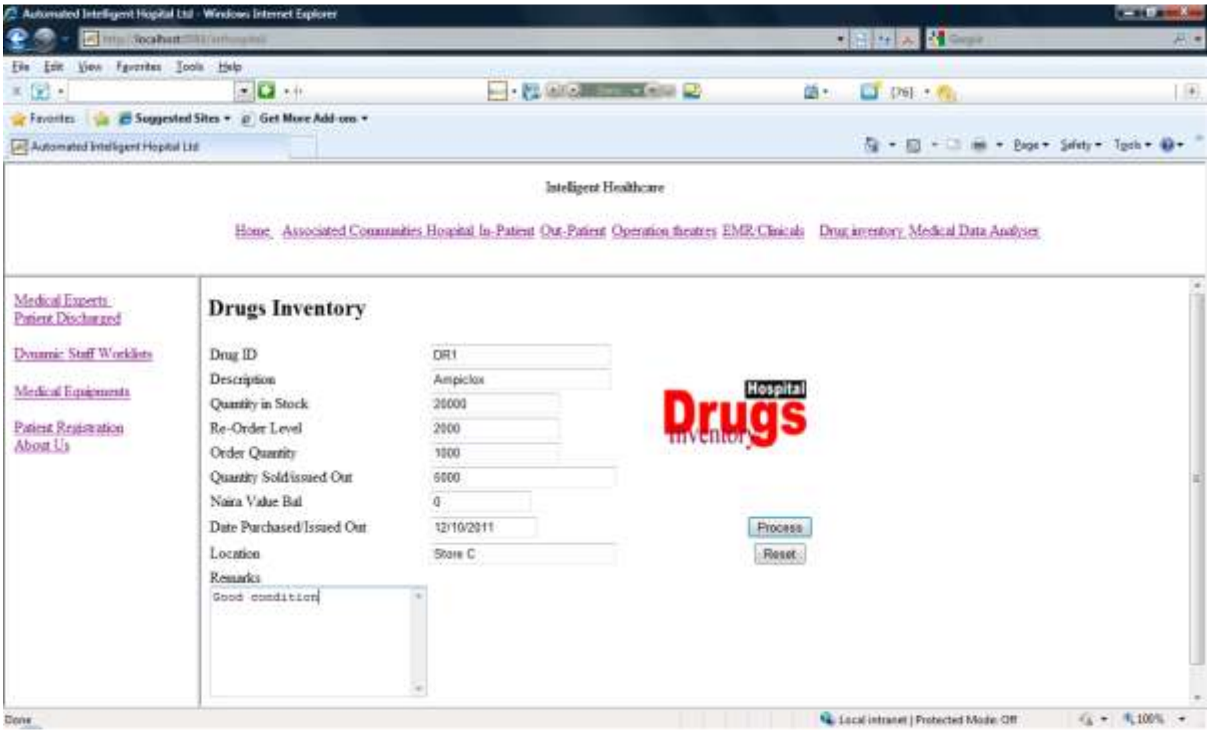
Intelligent Healthcare

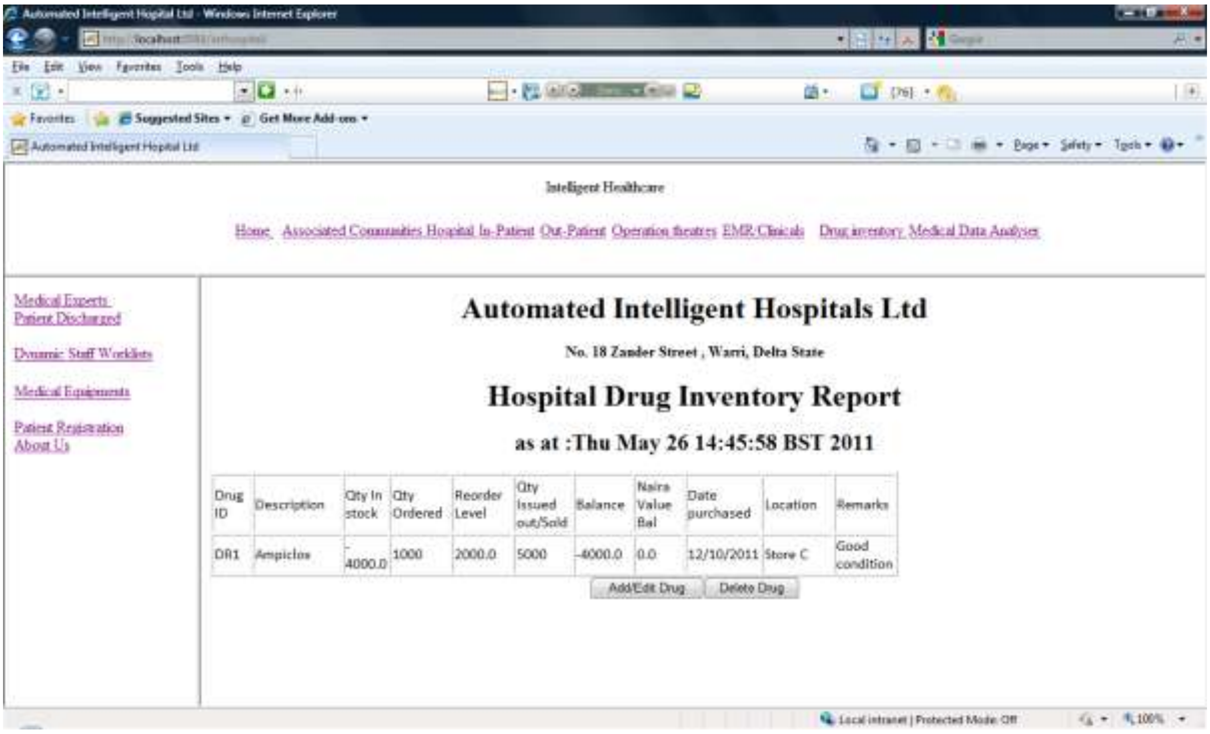
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser

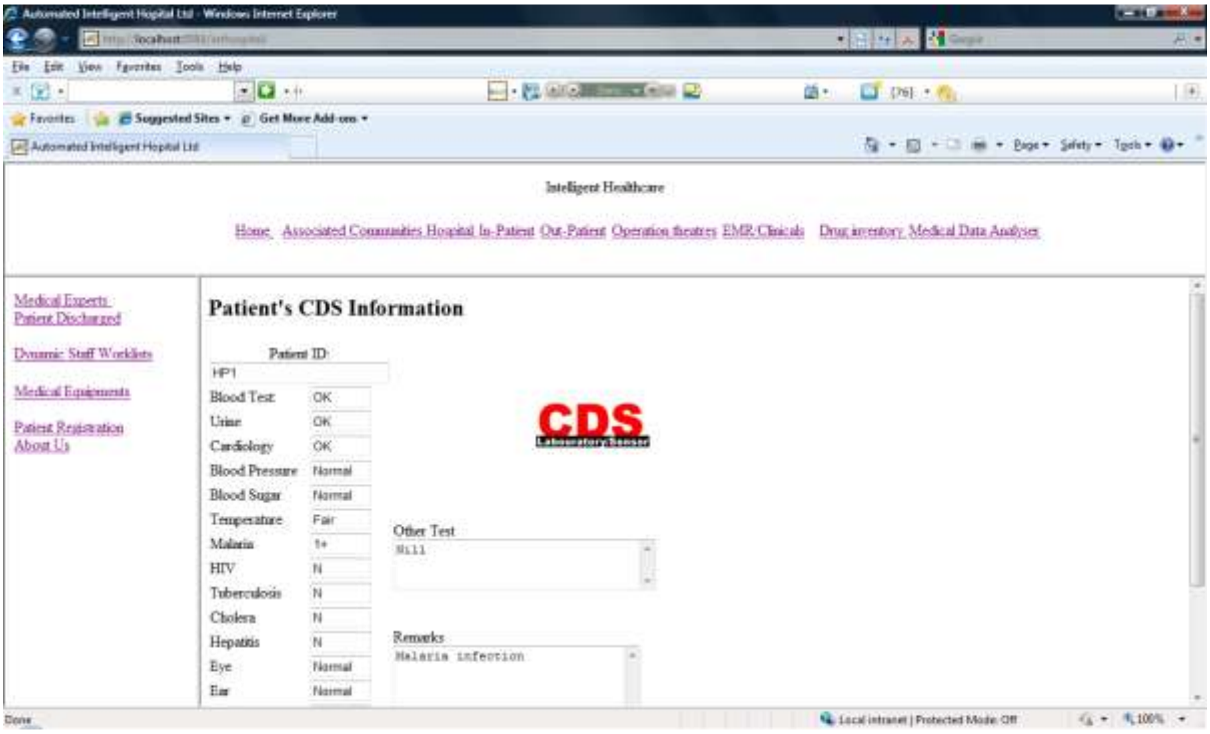
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us

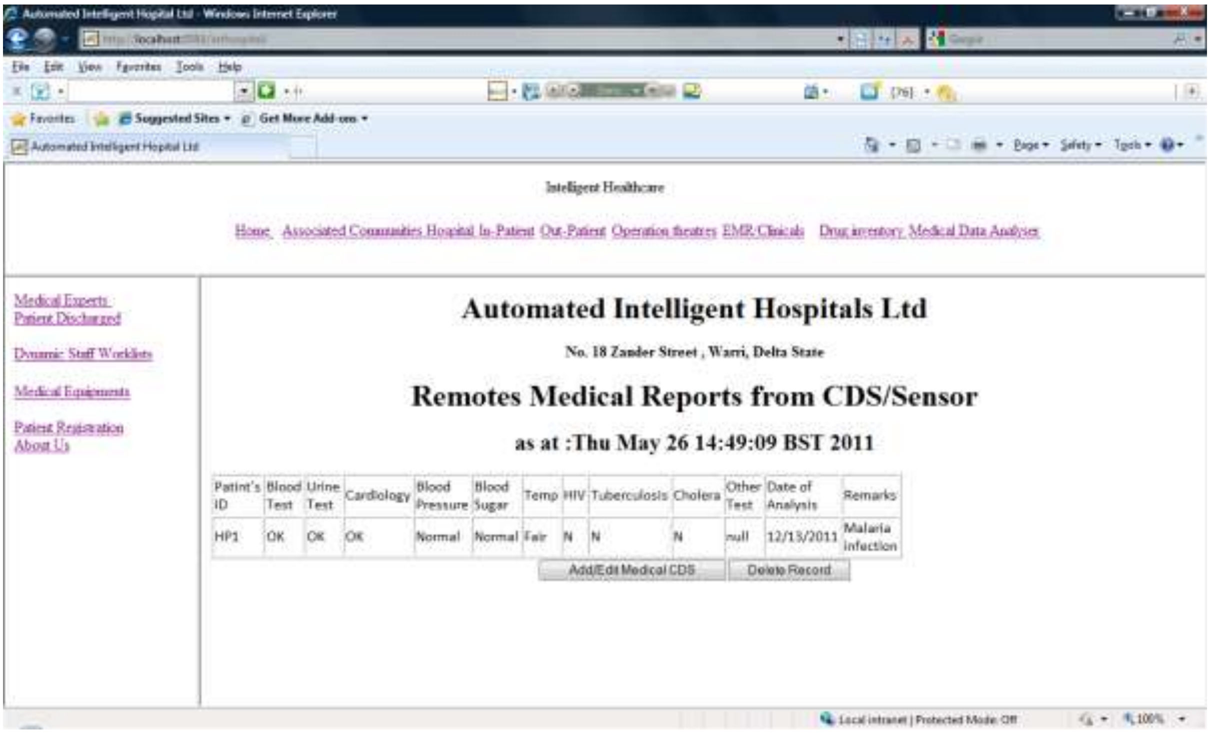
ifeoma oji	2011-04-09 00:00:00	120	1:20:01 AM	80	n	74	n	0	0	0	0	2	n
ifeoma oji	2011-04-09 00:00:00	125	1:20:45 AM	76	n	74	n	0	0	0	0	2	n
ifeoma oji	2011-04-09 00:00:00	116	1:21:40 AM	78	n	68	n	0	0	0	0	2	n
ifeoma oji	2011-04-09 00:00:00	131	9:57:15 AM	88	n	71	n	0	0	0	0	0	n
ifeoma oji	2011-04-09 00:00:00	117	9:59:11 AM	81	n	66	n	0	0	0	0	0	n
ifeoma oji	2011-04-09 00:00:00	117	9:59:53 AM	79	n	65	n	0	0	0	0	0	n
ifeoma oji	2011-04-09 00:00:00	103	10:00:32 AM	75	n	63	n	0	0	0	0	1	n
ifeoma oji	2011-04-09 00:00:00	109	10:01:13 AM	79	n	62	n	0	0	0	0	1	n
ifeoma oji	2011-04-09 00:00:00	109	10:01:53 AM	76	n	63	n	0	0	0	0	1	n
ifeoma oji	2011-04-09 00:00:00	114	10:02:33 AM	78	n	61	n	0	0	0	0	1	n
ifeoma oji	2011-04-15 00:00:00	116	9:06:29 AM	81	n	63	n	0	0	0	0	0	n
ifeoma oji	2011-04-15 00:00:00	118	9:07:19 AM	81	n	65	n	0	0	0	0	0	n
ifeoma oji	2011-04-15 00:00:00	111	9:08:11 AM	78	n	60	n	0	0	0	0	0	n
ifeoma oji	2011-04-15 00:00:00	110	9:08:52 AM	78	n	60	n	0	0	0	0	0	n
ifeoma oji	2011-04-15 00:00:00	116	9:09:33 AM	73	n	61	n	0	0	0	0	0	n
ifeoma oji	2011-04-15 00:00:00	113	9:11:48 AM	79	n	63	n	0	0	0	0	0	n


Automated Intelligent Hospital Ltd - Windows Internet Explorer
Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Drugs Inventory
Drug ID
DR1
Process
Reset
Hospital
Drugs
Inventory
Local intranet | Protected Mode: Off


Automated Intelligent Hospital Ltd - Windows Internet Explorer
Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Drugs Inventory
Drug ID
DR1
Description
Ampiclox
Quantity in Stock
20000
Re-Order Level
2000
Order Quantity
1000
Quantity Sold/issued Out
6000
Naira Value Bal
0
Date Purchased/Issued Out
12/10/2011
Location
Store C
Remarks
Good condition
Process
Reset
Hospital
Drugs
Inventory
Local intranet | Protected Mode: Off

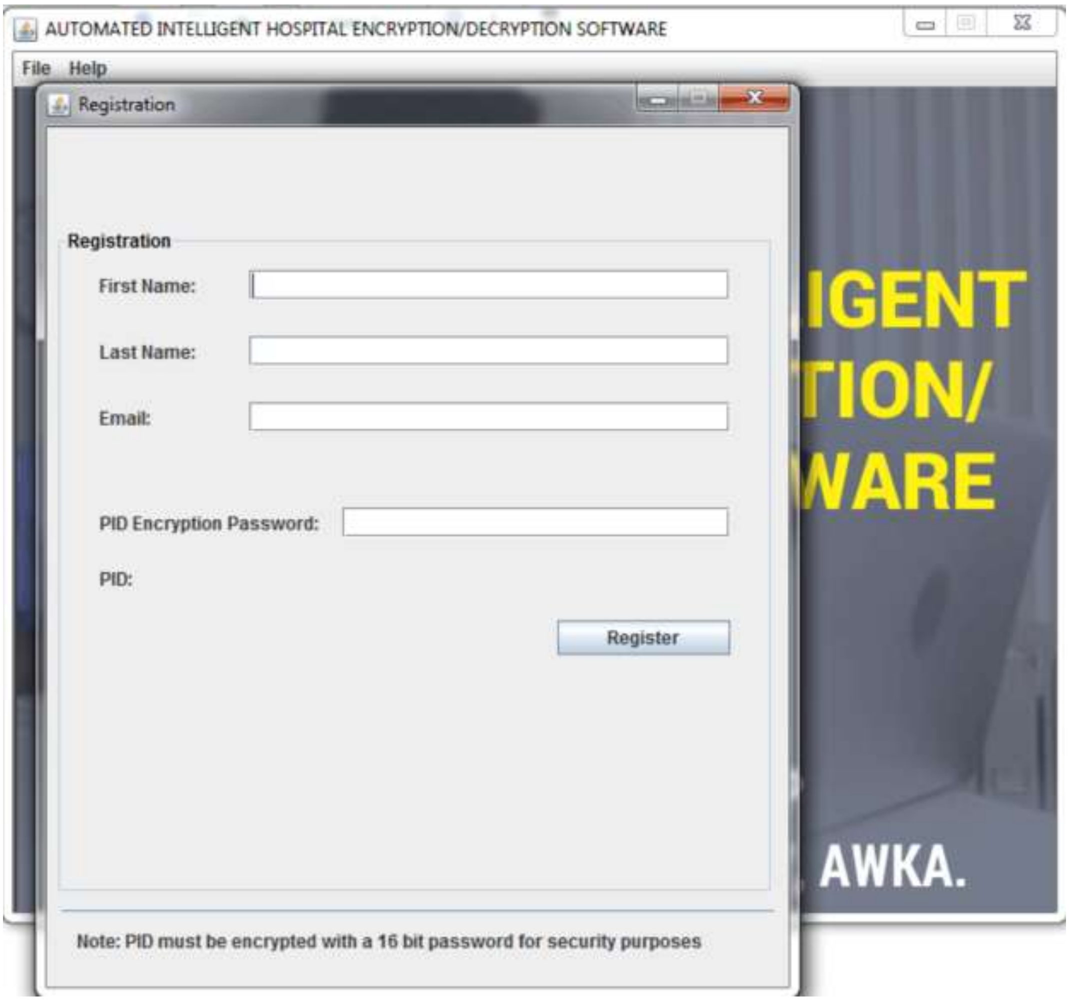

Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Automated Intelligent Hospitals Ltd
No. 18 Zander Street , Warri, Delta State
Hospital Drug Inventory Report
as at :Thu May 26 14:45:58 BST 2011
Drug ID
Description
Qty In stock
Qty Ordered
Reorder Level
Qty Issued out/Sold
Balance
Naira Value Bal
Date purchased
Location
Remarks
DR1
Ampiclox
-4000.0
1000
2000.0
5000
-4000.0
0.0
12/10/2011
Store C
Good condition
Add/Edit Drug
Delete Drug
Local intranet | Protected Mode: Off

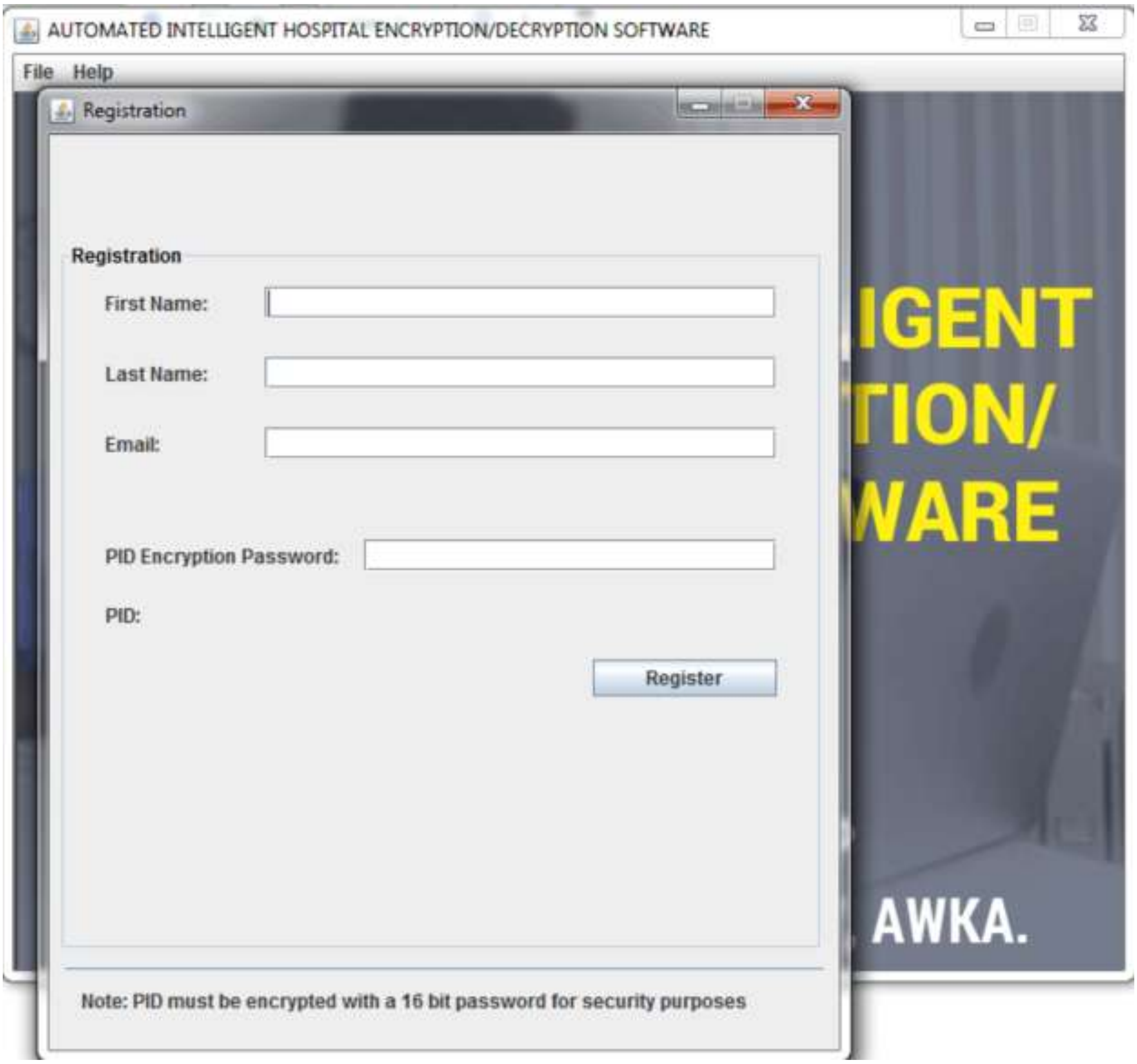

Intelligent Healthcare
Home Associated Communities Hospital In-Patient Out-Patient Operation theatres EMR/Clinicals Drug inventory Medical Data Analyser
Medical Experts
Patient Discharged
Dynamic Staff Worklists
Medical Equipments
Patient Registration
About Us
Patient's CDS Information
Patient ID:
HP1
Blood Test OK
Urine OK
Cardiology OK
Blood Pressure Normal
Blood Sugar Normal
Temperature Fair
Malaria 1+
HIV N
Tuberculosis N
Cholera N
Hepatitis N
Eye Normal
Ear Normal
CDS
Other Test
Nill
Remarks
Malaria infection
Done
Local intranet | Protected Mode: Off

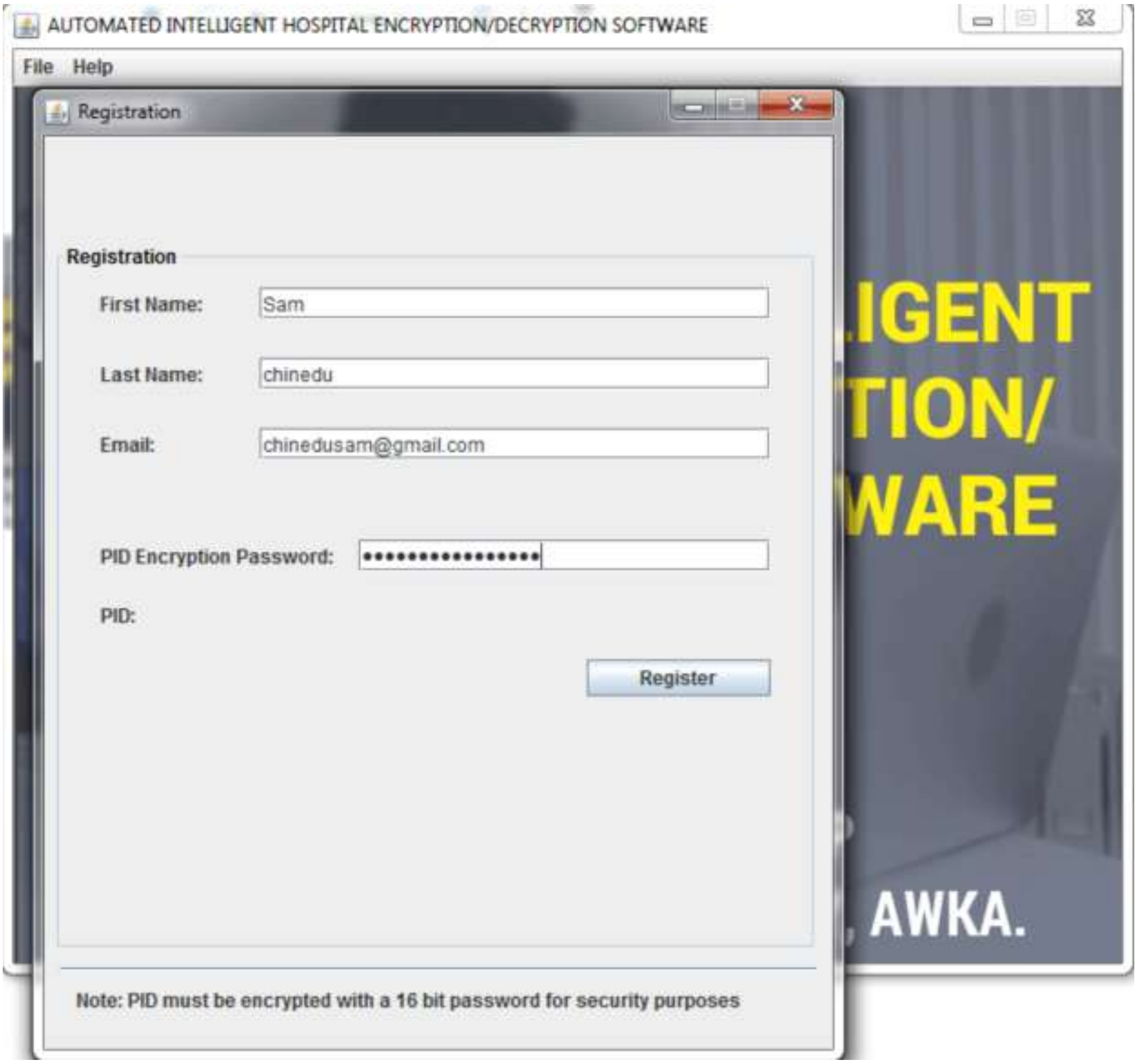


The Encryption Output

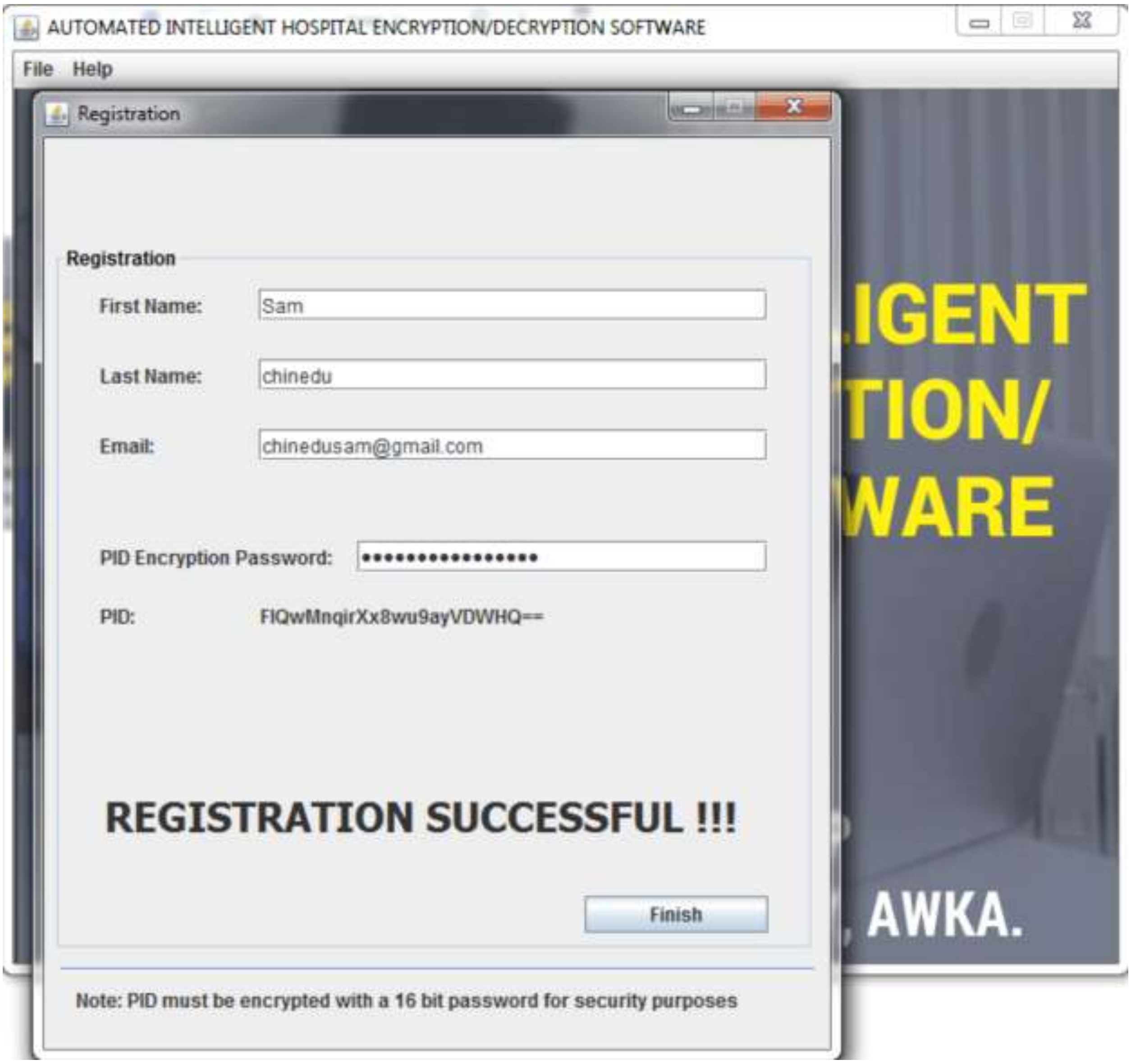

AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/DECRYPTION SOFTWARE
File
Help
Register New User
Check Medical Status
Exit
AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/ DECRYPTION SOFTWARE
BY
IFEOMA VIVIAN OJI
REG. NO: 2007517008P
NNAMDI AZIKIWE UNIVERSITY, AWKA.

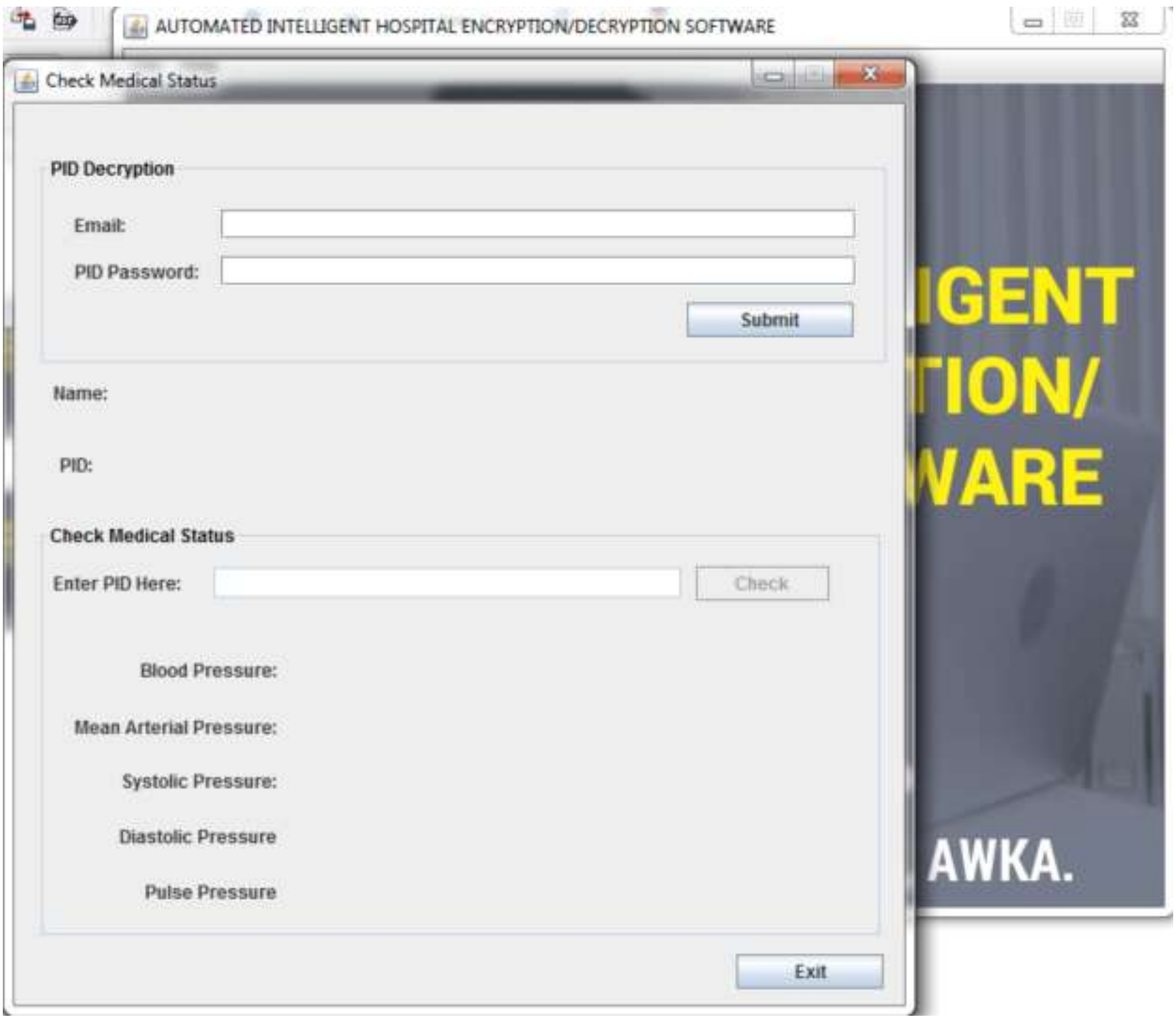

AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/DECRYPTION SOFTWARE
File Help
Registration
Registration
First Name:
Last Name:
Email:
PID Encryption Password:
PID:
Register
IGENT
TION/
WARE
AWKA.
Note: PID must be encrypted with a 16 bit password for security purposes

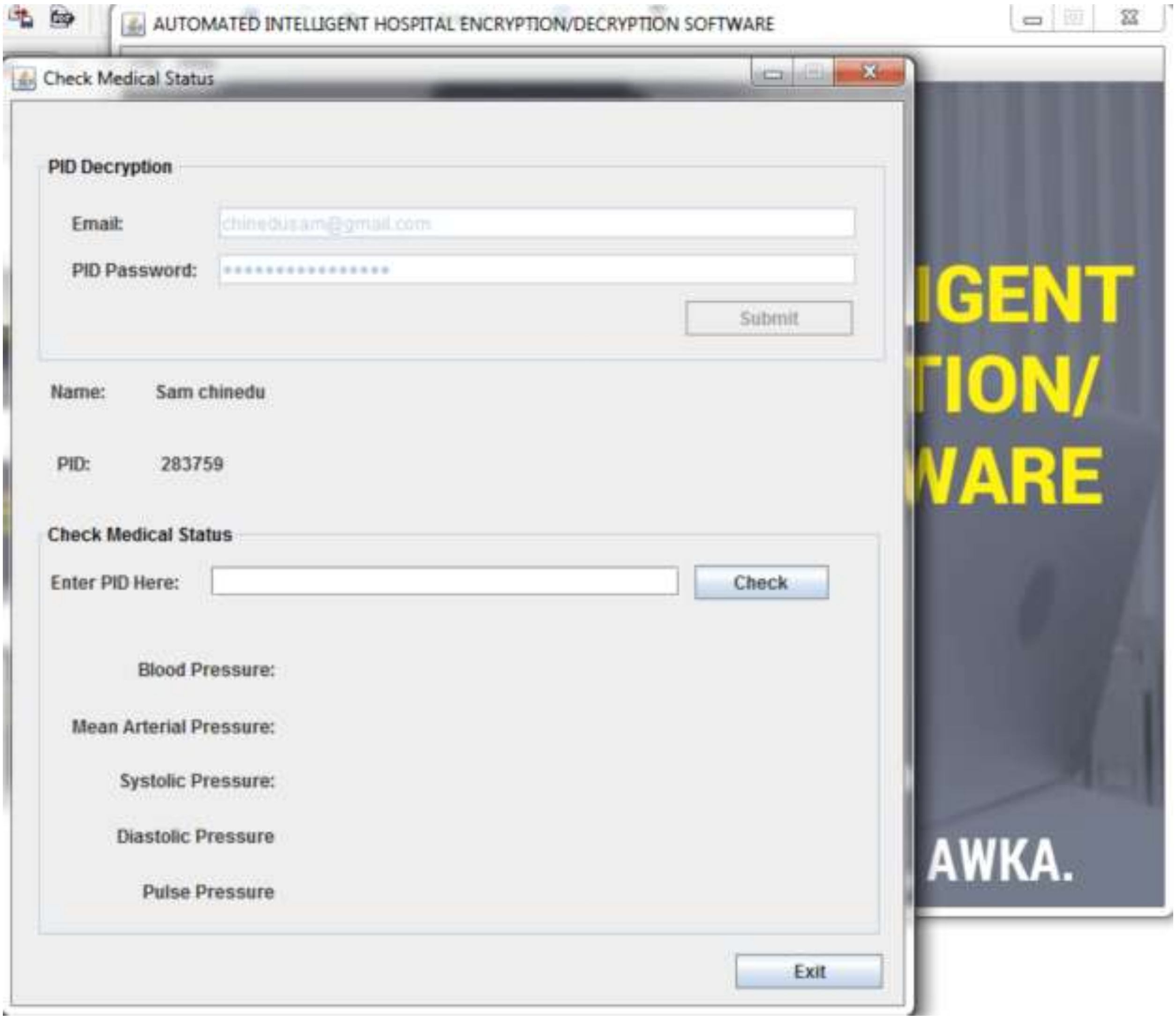

AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/DECRYPTION SOFTWARE
File Help
Registration
Registration
First Name:
Last Name:
Email:
PID Encryption Password:
PID:
Register
Note: PID must be encrypted with a 16 bit password for security purposes
IGENT
TION/
WARE
AWKA.



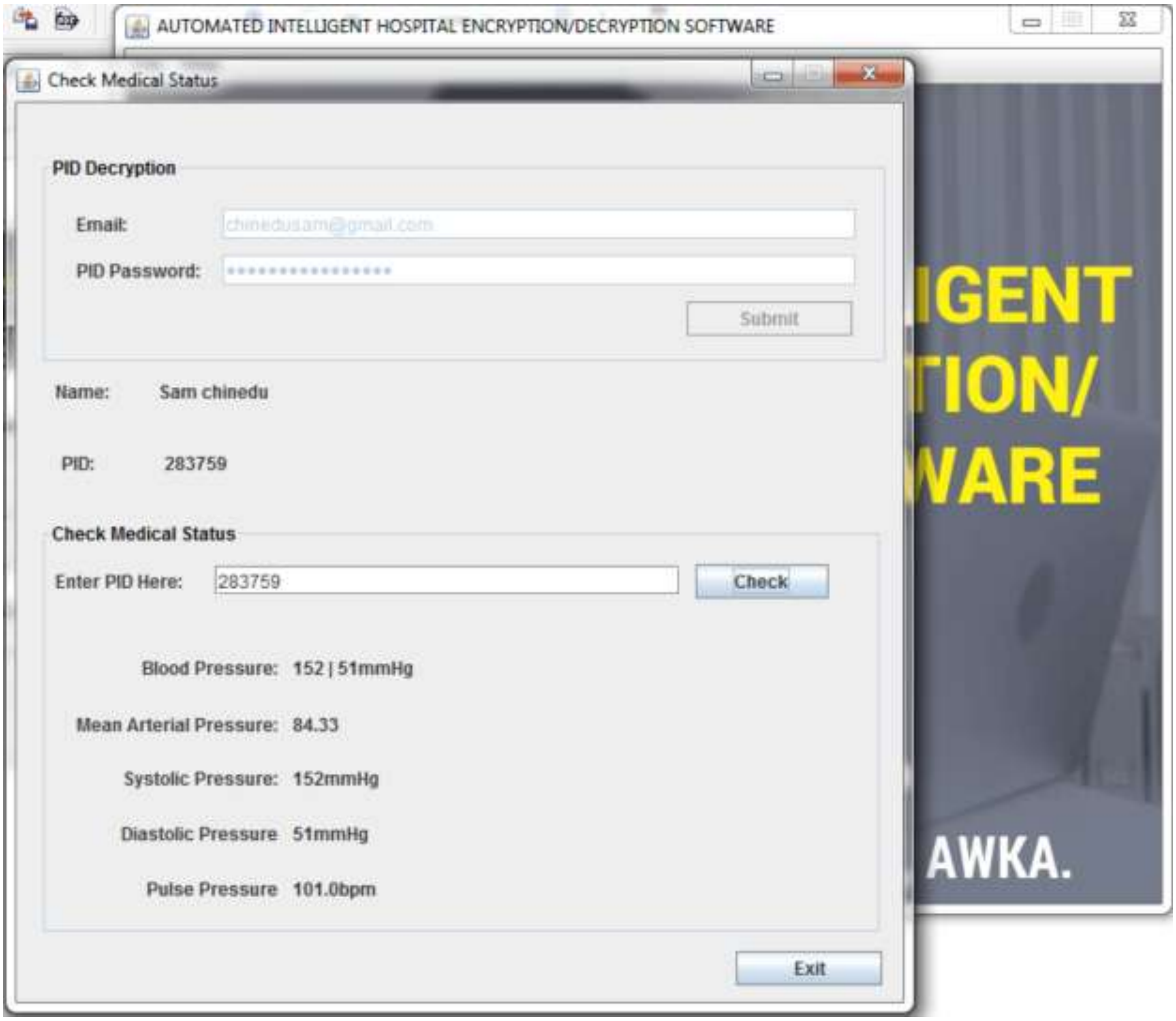
AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/DECRYPTION SOFTWARE
File Help
Registration
Registration
First Name:
Sam
Last Name:
chinedu
Email:
chinedusam@gmail.com
PID Encryption Password:
PID:
Register
Note: PID must be encrypted with a 16 bit password for security purposes
LIGENT
TION/
WARE
AWKA.


AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/DECRYPTION SOFTWARE
File Help
Registration
Registration
First Name: Sam
Last Name: chinedu
Email: chinedusam@gmail.com
PID Encryption Password:
PID: FIQwMnqirXx8wu9ayVDWHQ==
REGISTRATION SUCCESSFUL !!!
Finish
Note: PID must be encrypted with a 16 bit password for security purposes
.IGENT
TION/
WARE
, AWKA.


AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/DECRYPTION SOFTWARE
Check Medical Status
PID Decryption
Email:
PID Password:
Submit
Name:
PID:
Check Medical Status
Enter PID Here:
Check
Blood Pressure:
Mean Arterial Pressure:
Systolic Pressure:
Diastolic Pressure
Pulse Pressure
Exit
IGENT
TION/
WARE
AWKA.



AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/DECRYPTION SOFTWARE
Check Medical Status
PID Decryption
Email:
chinedusam@gmail.com
PID Password:
Submit
Name:
Sam chinedu
PID:
283759
Check Medical Status
Enter PID Here:
Check
Blood Pressure:
Mean Arterial Pressure:
Systolic Pressure:
Diastolic Pressure
Pulse Pressure
Exit
IGENT
TION/
WARE
AWKA.


AUTOMATED INTELLIGENT HOSPITAL ENCRYPTION/DECRYPTION SOFTWARE
Check Medical Status
PID Decryption
Email:
chinedusam@gmail.com
PID Password:
Submit
Name: Sam chinedu
PID: 283759
Check Medical Status
Enter PID Here: 283759
Check
Blood Pressure: 152 | 51mmHg
Mean Arterial Pressure: 84.33
Systolic Pressure: 152mmHg
Diastolic Pressure 51mmHg
Pulse Pressure 101.0bpm
Exit
IGENT
TION/
WARE
AWKA.