

**DEVELOPMENT OF A MODEL FOR USER-CENTRIC
CYBER DISASTER RECOVERY**

By

**KARIM USMAN
2011517004P**

**A PhD DISSERTATION SUBMITTED TO THE
DEPARTMENT OF COMPUTER SCIENCE,
IN PARTIAL FULFILLMENT FOR THE AWARD OF
DOCTOR OF PHILOSOPHY (PhD) IN SOFTWARE
ENGINEERING IN COMPUTER SCIENCE**

**FACULTY OF PHYSICAL SCIENCES,
NNAMDI AZIKIWE UNIVERSITY, AWKA.**

OCTOBER, 2019

CERTIFICATION

This is to certify that I am responsible for this research work, that the original work submitted is mine except as specified in the references, and acknowledgements and neither the dissertation, nor the original work contained therein has been submitted to this University or any institution for the award of a degree.

Karim Usman
2011517004P

Date

APPROVAL PAGE

This dissertation entitled Development of a Model for User-Centric Cyber Disaster Recovery carried out by Karim Usman, with Registration Number 2011517004P has been read and approved for the Department of Computer Science, Faculty of Physical Sciences, Nnamdi Azikiwe University, Awka in partial fulfillment for the award of Doctor of Philosophy (PhD) in Computer Science.

By

Prof. H.C. Inyama
Supervisor

Date

Prof. O. R. Okonkwo
Head, Department of Computer Science

Date

Prof. Stella Chiemeka
External Examiner

Date

Prof. S. O. Anigbogu
Dean, Faculty of Physical Sciences

Date

Prof. P. K. Igbokwe
Dean, SPGS

Date

DEDICATION

To Almighty God for His strength that is always made perfect in our weaknesses.

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my Supervisor Prof. Inyama H.C. for his patience, guidance, and encouragement during the course of this research. Your mentoring and tutoring truly saw me to this level. My sincere acknowledgment also goes to the Head, Department of Computer Science, Prof. O. R. Okonkwo, The Dean, Faculty of Physical Sciences Prof. S. O. Anigbogu, and to my amiable Lecturers Prof. (Mrs) V. E. Ejiofor, Prof. B. C. Ekechukwu, Dr. M. O. Onyesolu, Dr. D. C. Edebeatu and Dr. S. O. Okide for their positive contributions.

I am also indebted to my beloved wife Mrs Karim Risikatu and our beloved children Victory, Confidence, Courage and Michelle for their support and understanding during this stressful period.

I owe a great deal to my parents Mr and Mrs Usman Salami of blessed memory, my elder sister, Mrs Muhammed Barikisu and my nephew Hassan Ibrahim without whose support and encouragement, the completion of this dissertation would have been very much impossible.

I remain ever grateful to my spiritual Parent and Mentors, Pst. Dr. (Mrs) Adah Olabode and Pst Dr. S. T. Olabode for their prayers and spiritual guidance throughout this research.

Finally, I wish to acknowledge my friends who offered their moral support during this stressful period. May God bless you all!

TABLE OF CONTENTS

CONTENT	PAGES
TITLE PAGE	i
CERTIFICATION	II
APPROVAL PAGE	iii
DEDICATION	iv
ACKNOWLEDGMENTS	V
TABLE OF CONTENTS	VI
LIST OF TABLES	viii
LIST OF FIGURES	IX
ABSTRACT	XI
CHAPTER ONE: INTRODUCTION	
1.1 Background of the Study	1
1.2 Statement of the Problem	4
1.3 Aim and Objectives of the Study	5
1.4 Significance of the Study	6
1.5 Scope of the Study	7
1.6 Limitations of the Study	7
1.7 Definition of Terms	7
CHAPTER TWO: LITERATURE REVIEW	
2.1 Overview	10
2.2 Conceptual Review	10
2.3 Cloud Computing Model	23
2.4 Cloud Computing Vendors	27
2.5 Cloud Security Solutions	30
2.6 Cloud Computing Management Platforms	30
2.7 Disaster Recovery (DR)	35
2.8 Security Issues in Service Delivery Models of Cloud Computing	38

2.9	Related Works on Disaster Recovery Systems	53
2.10	Summary of Related Works	65
2.11	Summary of Literature Review and Knowledge Gap	71
CHAPTER THREE: SYSTEM ANALYSIS AND METHODOLOGY		
3.1	System Analysis	73
3.1.1	Analysis of the Existing System	73
3.2	Analysis of the Proposed System	75
3.2.2	High-Level Model of the Proposed System	79
3.3	Methodology Adopted	80
CHAPTER FOUR: SYSTEM DESIGN AND IMPLEMENTATION		
4.1	Objectives of the Design	82
4.2	Main Menu/Control Centre	82
4.3	Submenu/Subsystem	84
4.4	Specifications	88
4.5	Object Diagrams	103
4.6	System Implementation	109
4.7	Results and Discussion	119
CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATION		
5.1	Summary	129
5.2	Conclusion	129
5.3	Recommendations	130
5.4	Contribution to Knowledge	131
REFERENCES		133
APPENDIX A: PROGRAM LISTINGS		144
APPENDIX B: SAMPLE OUTPUTS		212

LIST OF TABLES

2.1	Deployment model description	27
2.2	Eucalyptus and OpenNebula compared	35
2.3	Security challenges in identity management [IdM] and sign-on process.	52
2.4	Literature Review Table	66
4.1	User Table	90
4.2	Application Table	91
4.3	Backup Configuration Table	91
4.4	Backup History Table	92
4.5	Subscription History Table	92
4.6	List of Identifiers and their Meaning	103
4.7	Actual Result versus Expected Test Result	112
4.8	Comparison between DES, 3DES, BF, AES, and the Hybrid Encryption Time(s)	114
4.9	Comparison between LZW, Huffman Coding, Shannon Coding and the M-Huffman	115

LIST OF FIGURES

1.1	Various issues/challenges to the cloud model. (Jangra and Bala, 2012)	3
2.1	Four Major Elements of a Cyberspace (Stytz and Banks, 2014)	13
2.2	Five Layers of the Cyberspace (Lehto and Neittaanmäki, 2015)	14
2.3	Two Streams Model of Cybersecurity Norm Emergence Process at the United Nations (Maurer, 2011)	16
2.4	Example of Active and Passive Attacks (Sobh, 2013)	20
2.5	Eucalyptus structure (Endo et al., 2010; Peng et al., 2009)	32
2.6	OpenNebula Platform (Haji, Letaifa, and Tabbane, 2010)	33
2.7	Overviews of RUBiS system architecture (Wood et al., 2010)	58
2.8	The architecture of the HS-DRT system (Ueno et al., 2010)	60
2.9	Disaster-CDM Framework (Grolinger et al., 2013)	61
2.10	Distributed Cloud System Disaster Recovery Architecture (Ousterhout et al., 2010)	62
3.1	Data Flow Diagram of the Existing System	75
3.2	Model Diagram for File Backup	77
3.3	Model Diagram for the Billing System (Pay as you go)	78
3.4	Model Diagram for the Recovery System	79
3.5	High-Level Model of the New System	81
3.6	DSDM Process Diagram (Stapleton, 1997)	82
4.1	Main Menu	84
4.2	Registration Sub System	85
4.3	Configuration Sub System	86
4.4	Payment Sub System	87
4.5	Security Sub System	88
4.6	Report Sub System	89
4.7	Architectural Diagram of the New System	95
4.8	User Registration Form	96
4.9	App Registration Form	97
4.10	Subscription Form	97
4.11	Backup Configuration Form	98
4.12	E-R Diagram of the System	104
4.13	Use Case Diagram of the System	105
4.14	Class Diagram of the System	106

4.15	Deployment Diagram of the System with a focus on Download Module	107
4.16	Sequence Diagram of the System	108
4.17	Input File Size versus Encryption Time for some Encryption Algorithms	114
4.18	Input File Size versus Compressed Output File Size for some Compression Algorithms	116
4.19	User Authentication Page	120
4.20	User Registration Page	121
4.21	Application Registration Page	122
4.22	Application Configuration Page	123
4.23	Backup File Download Page	124
4.24	File Downloading Dialogue Box	125
4.25	List of Downloaded Files	126
4.26	Decompression and Decryption Interface	127
4.27	Sample of Decompressed and Decrypted Backup File	128

ABSTRACT

In a world of interdependent economies and online transactions, a large volume of data is hosted on the cyberspace on daily basis. Cyber threats and attacks are steadily increasing. Most time, these threats and attacks are targeted at service providers but service users are greatly affected by the attacks due to their vulnerability level. When disasters knockdown the infrastructures of a single service provider, it will have ripple effects on thousands of innocent service users. Therefore, service users need more than ever to prepare for major crises targeted at their service providers. To cope with this trend, every service user requires an independent business continuity plan (IBCP) or disaster recovery plan (DRP) and data backup policy which falls within their cost constraints while achieving the target recovery requirements in terms of recovery time objective (RTO) and recovery point objective (RPO). The aim of this research work is to develop a user-centric cyber disaster recovery mechanism to enable service users to independently develop and implement their independent data backup policies that best suits their remote databases. The system developed is highly compatible with MYSQL, MSSQL and Oracle databases. With this system, service users have the liberty to independently define and implement their private backup plans and disaster recovery policies and also to configure their remote databases by selecting the entities to be backed up and for each entity selected, the backup frequency is also selected. The system creates backup files for the remote databases in accordance with their configuration settings. The backup files are encrypted to prevent its contents from Man in the Middle attacks (MITM). The encrypted backup files are also compressed to enhance its transmission across networks. A combination of Dynamic System Development Methodology (DSDM) and Object-Oriented Analysis and Design Methodology (OOADM) are used to design the system while Java Enterprise Edition (JEE) is used to develop the system. The system is well tested and the results obtained are compared with some well-known systems and outputs are relatively good.

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

In the world of technology, change is the only constant factor. The change is so dynamic that what was considered at one era to be an obsolete technology can become the current trend in another era. Computing in the past three decades witnessed a tremendous change in the way and manner data were processed which resulted in the emergence of distributed systems against the pre-existing centralized systems. It is interesting to see that presently, computing is going back to some sort of centralization with a brand name called Cloud computing (Westerlund and Kratzke, 2018).

The term *Cloud* first appeared in the early 1990s in the communication world. It was a time when telecom providers introduced the use of Virtual Private Network (VPN) services for data communication. Virtual Private Network services could maintain the same bandwidth as fixed networks could at significantly less cost. These networks (VPN and the fixed network) supported dynamic routing which enhanced balanced utilization across the network and an increase in bandwidth efficiency. This led to the coining of the term *telecom cloud*. Cloud computing has a similar premise in that it provides a virtual computing environment which is dynamically allocated to meet the need of the user.

Cloud computing is a computing paradigm that employs internet technologies to provide scalable and elastic computing infrastructure (hardware, software, processing, and storage) as a service to the external customer (Mishra, Mohapatra, Mishra, and Sahoo, 2018). It relies basically on both internet and virtualization technologies while the former provides client access to the cloud, the later offers each subscriber one or more individual virtual instances. With virtualization technology, several virtual servers can be hosted by each physical server (Zhao, Amagasaki, Iida, Kuga, and Sueyoshi, 2017).

Prior to the emergence of cloud computing, supercomputers were used in specific areas like the military, government agencies, universities and research laboratories to handle enormous complex calculations. Cloud computing therefore aims at further diversifying

the use of supercomputers by applying their power to solving problems that require complex computational resources (Buyya and Son, 2018). With the Internet connection, users are granted immediate access to a large number of the world's most sophisticated supercomputers together with their corresponding processing power, interconnected at diverse locations around the world.

In the past, Information Technology (IT) resources and applications were provided as products which were sold or licensed from vendors to users and then exploited locally on local computers. Cloud Computing brings about a shift in the paradigm in that instead of purchasing hardware or software, a user purchases remote access to them via the Internet. Service Providers (SPs) only issue an invoice to Service Users (SUs) on a utility basis that is pay-as-you-go as done with electricity, water, and telecommunication or on a subscription basis. That is why it is viewed as a business model which delivers IT resources and applications as services rather than products and accessible remotely rather than locally (Shovon, Roy, Sharma, and Whaiduzzaman, 2018).

The evolution of cloud services has enabled entities to do more with less few resources and better operating efficiency. This has many tangible benefits for business, however, there are inherent security risks that must be evaluated, addressed, and resolved before business owners will have confidence in completely outsourcing their IT requirements to service providers. IT companies take security, performance, data availability and difficulty in bringing back data in-house (Data Backup) as top challenging factors inhibiting them from adopting cloud services (Jangra and Bala, 2012). Their finding is as presented in Figure 1.1.

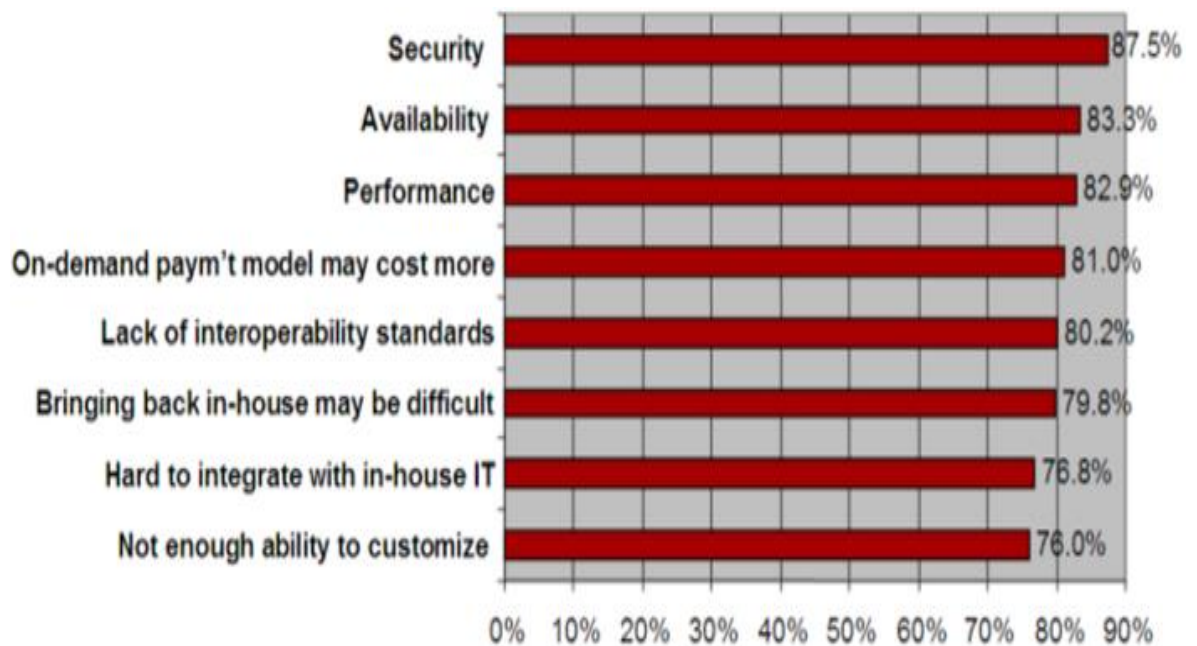


Figure 1.1: Various issues/challenges to the cloud model. (Jangra and Bala, 2012)

Figure 1.1 presents security, performance, data availability and difficulty in bringing back data in-house (i.e. Data Backup) are the most challenging issues in cloud computing. These hiccups explain why many business owners and some government agencies are yet to trust and utilize the immense benefits of cloud computing. Many enterprises which have planned to migrate to cloud prefer using the cloud for less sensitive data and store important data within enterprise boundary (Jangra and Bala, 2012). It is important to note however, that no matter how careful you are with your personal data, by subscribing to cloud services, you will be giving up control to an external source. This distance between you and the physical location of your data creates a barrier. It may also create more space for a third party to access your information without your knowledge or approval. With this, regular back up of your private data becomes very difficult. This Inadequate data backups and improper data syncing are what has made many businesses vulnerable to ransomware, a specific type of cloud security threat (Stergiou, Psannis, Kim and Gupta, 2018). Ransomware operates by locking away company's data in encrypted files, only allowing them to access the data once a ransom has been paid. With appropriate data backup solutions, companies need no longer fall prey to these threats.

Thus, despite the overwhelming benefits available on cloud computing, two major challenges that are yet to be completely handled are the issues of security flexibility in

disaster recovery policies (Jangra and Bala, 2012). The cloud is often seen as valuable to individuals with malicious intent like terrorists and hackers due to the large volume of information hosted on it. There is a lot of personal information and potentially secure data that people store on their computers, and this information is now being transferred to the cloud. This makes it critical for cloud users to understand the security measures that cloud service providers have in place. Disaster recovery as an aspect of the security package is one of the major factors service users take into consideration when subscribing for online hosting. What encryption methods do the providers have in place? What methods of protection do they have in place for the actual hardware that your data will be stored on?

Presently, most cloud service users depend on the security and the backup policies provided for them by their service providers (Rachana and Guruprasad, 2014). With the emergence of cloud computing as a preferred technology for outsourcing IT operations, security issues inherent in the hosting model have assumed greater significance. Inherent in the concept of cloud computing are the risks associated with entrusting confidential and sensitive data to third parties or service providers (SPs). In spite of the several advantages that cloud computing offers, there are several concerns and issues which need to be solved before the ubiquitous adoption of this computing paradigm. First, in cloud computing, the user may not have the kind of control over his/her data. Secondly, the cloud users may risk losing data by having them locked into proprietary formats and may lose control over their data since the tools for monitoring who is using them or who can view them are not always provided to the service users (Garg, Thakral, Nalwa and Choudhury, 2018). Data loss is, therefore, a potentially real risk in some specific deployments. Thirdly, it may not be easy to tailor service-level agreements (SLAs) to the specific needs of a business. Compensation for downtime may be inadequate and SLAs are unlikely to cover the concomitant damages.

1.2 Statement of the Problem

Studies in the literature have shown that a lot of disaster recovery models and data backup policies are currently available. However, service users do not have control over these models and policies since they are always on the premises of service providers (Thomas, 2018). This explains why service users constantly depend on service

providers for backup policies and disaster recovery plans. This issue of dependency inherent in available disaster recovery solutions makes service users to face the following challenges;

- i. Difficulty in bringing back their data to a location of their choice once hosted (Jangra and Bala, 2012).
- ii. Inability to configure their backup policy
- iii. Storage services provided by one service provider may be incompatible with another service provider as such it becomes very difficult for service users to transfer their applications from one service provider to other in the phase of disaster without losing chunk of their sensitive data (e.g. Microsoft cloud is incompatible with Google cloud) (Basu et al., 2018; Popović and Hocenski, 2010).

This work therefore, seeks to assess these problems and offer a simple, concise and direct solution by developing a User-Centric Cyber Disaster Recovery Model that will place critical backup and disaster recovery decision making into the hands of Service Users. Affordable, efficient, and scalable, cloud computing is still the best solution for most businesses, but it leaves Service Users vulnerable if the proper precautions are not taken (Shovon et al., 2018).

1.3 Aim and Objectives of the Study

The aim of this research is to develop a Model for User-Centric Cyber Disaster Recovery. The objectives of the research are to;

- i. develop a prototype to implement the model
- ii. grant Service Users the privilege to define and implement their independent Disaster Recovery Plan (DRP) and data backup policy
- iii. automatically generate compressed and encrypted backup files in accordance to the backup policy
- iv. grant Service Users the privilege to download the compressed and encrypted backup files to any location of their choice within their premises
- v. provide an interface to decompressed and decrypted the backup file to generate pure Structural Query Language (SQL) scripts to facilitate recovery process in an event of cyber disaster

1.4 Significance of the Study

This study is important in a number of ways. ICT units of both government, private sectors and public institutions will find the outcome of this research work useful because it will aid them to easily configure the backup policies of their remote databases in accordance with their specific needs. With this, database administrators of any organisation will be very sure that very important entities in their remote databases are being backup at their chosen frequencies and their backup files can be downloaded to any location or device of their choice at any time even when their service provider is suffering from any form of service down time as a result of cyber disaster.

The implementation of this system will grant flexibility to Service Users to migrate from one Service Provider to another without losing their sensitive data as copies of their remote database instances are always within their premises. In event of disaster hitting their Service Providers or even a prolonged disruption of services, Service Users can easily change to other Service Providers in less time without loss of sensitivity

The work when completed shall be very useful in many governmental and non-government institutions around the globe where application hosting on cyberspace has become eminent. In Nigeria for instance, it will be useful for Military and all the Para-Military to deploy this solution to enable them to bring their operation data which are currently hosted by Service Providers in other countries back to their premises. All Universities, Polytechnics and Colleges of Education in Nigeria have one portal or the other with which the institutions are being managed. Currently, all of these portals have their operational data in the custody of Service Providers. The findings of this research shall be useful to such institutions as it will help them to have access to their operational data and have a base to run back to in the event of a disaster. The findings of this research shall also very useful to all the commercial banks, examination bodies and small and medium scale businesses all over the globe.

The works shall also be very useful to Data Center Operators all over the globe as it will provide a gateway for backup files from one Service Provider to be ported in the infrastructures of other Service providers thereby providing handy solution to the issue of incompatibility in storage services deployed by various Service Provider.

1.5 Scope of the Study

This study is limited to quality and timely backup of databases which is a critical aspect of a disaster recovery system. The scope of the study is therefore summarized as follows:

The system so modeled can work perfectly with the three most commonly adopted Relational Database Management Systems (RDBMS) in cyberspace which are; Oracle, MySql and MSSql.

Due to the fact that the model hopes to grant absolute freedom for service users in terms of Disaster Recovery Policy, the new system generates encrypted backup files for users in accordance with their configurations which they can log in and download such backup files into their local system at any point in time.

Also, due to the fact that it would be difficult to get sensitive login details of corporate websites of most organisations, the system is mostly tested on local systems and an online portal (<https://www.naitesmkd.org>) which is an educational portal design and hosted by the researcher.

1.6 Limitations of the Study

There is no 100% guarantee that all online applications can be backed up with this system since some online applications are running on different Database Management Systems (DMBS) other than the once covered by the system. Also, Databases with pictures and video clips cannot be backed up and encrypted by the system.

1.7 Definition of Terms

The following terms are used throughout this research work.

Backup Replication: Backup replication is the frequent electronic copying of databases schema and data from a database in one computer or server to another location for storage and possibly recovery in event of a disaster (Zou and Jahanian, 1999).

Catastrophe: This can occur as a result of the occurrence of a disaster. Catastrophes may be avoided by using disaster avoidance mechanisms (Sánchez and Goldberg, 2003).

Cold Backup Site: In a cold backup site, data is often only replicated on a periodic basis, leading to a high Recovery Point Objective (RPO) of hours or days. In addition, servers to run the application after failure are not readily available, and there may be a delay of hours or days as hardware is brought out of storage or re-purposed from test and development systems, resulting in a high Recovery Time Objective (RTO). It can be difficult to support business continuity with cold backup sites, but they are a very low-cost option for applications that do not require strong protection or availability guarantees (Wood et al., 2010).

Cyber: This is used in referring to the Internet (Kumar, Raghavan, Rajagopalan, and Tomkins, 1999).

Cyberspace: The entire region of Internet coverage in the world

Disaster: A disaster is an event that creates an inability for an organization to provide essential services (Raphael, 1986).

Disaster Recovery (DR): This is the area of security planning that deals with protecting an organization from the effects of significant negative events of disasters

Disaster Recovery Plan (DRP): This is sometimes referred to as a Business Continuity Plan (BCP) or Business Process Contingency Plan (BPCP) it describes how an organization is to deal with potential disasters. Just as a disaster is an event that makes the continuation of normal functions impossible, a disaster recovery plan consists of the precautions taken so that the effects of a disaster will be minimized and the organization will be able to either maintain or quickly resume mission-critical functions (Wallace and Webber, 2017). Typically, disaster recovery planning involves an analysis of business processes and continuity needs; it may also include a significant focus on disaster prevention.

(DRPM): The individual or individuals assigned to oversee the creation, implementation, testing, periodic review and distribution of the DRP (Smith, Martin, and Wenger, 2018).

Hot Backup Site: A hot backup site typically provides a set of mirrored standby servers that are always available to run the application once a disaster occurs, providing minimal RTO and RPO. Hot standby typically uses synchronous replication to prevent any data loss due to a disaster (Wood et al., 2010).

Recovery: Recovery pertains to the immediate reinstatement of an organization's essential services after a natural or man-made disaster or other emergency situations.

Recovery Point Objective (RPO): This is the time (relative to the disaster) to which you plan to recover your data (Chatterjee, Mahalingam, Jayaraman, and Maliakal, 2016).

Recovery Time Objective (RTO): This is the maximum tolerable length of time that a computer, system, network, or application can be down after a failure or disaster occurs (Chatterjee et al., 2016).

Service Users (SU): These are businesses or individuals that subscribe to the services on the internet (Subashini and Kavitha, 2011).

Service Provider (SP): A service provider is a company that offers some component of cloud computing typically Infrastructure as a Service (IaaS), Software as a Service (SaaS) or Platform as a Service (PaaS) to other businesses or individuals (Subashini and Kavitha, 2011).

Threat: A threat is a potential attack that may lead to a misuse of information or resources (Mirkovic and Reiher, 2004).

Traditional Security: This can be defined as the measure taken to ensure the safety and material existence of data and personnel against theft, espionage, sabotage, or harm (Tow, Thakur, and Hyun, 2000).

Transmission Control Protocol/Internet Protocol (TCP/IP): This is an Internet protocol that operates as a thin layer. It controls data transmission in packets and routes the data to its destination (Feit, 1998).

Unified Modeling Language (UML): It is a system design tool or language that utilizes design models (drawings) independent of any programming language that can be implemented developing information systems (Booch, 2005).

Universal Resources Locator (URL): It is a unique website or web page address (Bahrs, Lillie, and Van Horn, 2006).

Vulnerability: Vulnerability refers to the flaws in a system that allows an attack to be successful (Mirkovic and Reiher, 2004).

Warm Backup Site: A warm backup site may keep state up to date with either synchronous or asynchronous replication schemes depending on the necessary RPO. Stand-by servers to run the application after failure are available, but are only kept in a *warm* state where it may take minutes to bring them online (Wood et al., 2010).

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

This chapter reviews literature relevant to this work under three sections; Conceptual review: This discusses the concept of cyberspace, forms of cyber threats and cyber-attacks. Secondly, the concept of Cloud Computing Model: which discusses the available Cloud Security Solutions, and Cloud Computing Management Platforms. Lastly, the concept of Cyber Disaster Recovery which discusses which discusses the requirements, mechanism and policy of available Cyber Disaster Recovery solutions.

2.2 Conceptual Review

This sub section discusses the general concepts of cyberspace, forms of cyber threats and cyber-attacks

2.2.1 The Concept of Cyberspace

The term cyberspace originated from Alvin Toffler's book titled future shock (Toffler, 1970). However, Williams Gibson a science fiction writer was the first to adopt it to write science fiction stories in a book Neuromancer (Gibson, 1995). Gibson gave credit to John Brunner author of shock wave rider 1975 as the inventor but Brunner refers the origin to Alvin Toffler.

In Toffler's work, a section titled "The Cyborgs among Us" which described the possibilities of humans integrating with the machine and human brains functioning independently out of their skulls. Gibson then adapted this concept in a science fiction called "Cyberpunk". Cyberpunks are people who explore the digital landscapes of electronic space and a term often used to describe outlaws and hackers from the computing point of view. In this fiction, everyone even punks and street gangs had access to technology, while huge multinational corporations combat each other illegally because each of them held more power and wealth than world governments (Gibson, 1995).

Furthermore, wars, as well as normal criminal acts were executed through purely electronic means, often using programs like artificial intelligence viruses that are so smart and like humans that some even receive a real human citizenship.

Today, this story has become a reality as “cyber-war is no longer a science fiction and the debate among policymakers on what norms should guide behavior in cyberspace has been in full swing” (Maurer, 2011). Cyberspace has gained so much attention from several researchers. Cyberspace has been described as a constituent of four major elements by (Stytz and Banks, 2014) as illustrated in Figure 2.1. The elements are: Data, computing technologies (like computer hardware, software, network infrastructure, network protocols, virtualization and cloud computing), information analysis/comprehension technologies (such as information virtualization, collaboration and data mining technologies) and information interaction/management technologies (such as human-computer interaction, intelligent agent technologies, human Internet interference, personalization technologies and database technologies). In the same vein, (Lehto and Neittaanmäki, 2015) classified the structure of the cyberspace into a five-layer model (illustrated in Figure 2.2) as the physical, syntactic, semantic, service and cognitive layers. The physical layer consists of physical hardware elements such as processors, storage devices, switches, routers, wired and wireless conduits and the signals that travel along these hardware; the syntactic layer consist of programs, conventions (such as protocols, encryption, error correction and compression) and addressing modes (TCP/IP) that controls computing systems and networks, while the semantic layer is made up of information stored and manipulated by systems and the rules that guide them; while the service layer contains public and commercial services used by users of a network and the cognitive layer is the user’s information awareness environment referred to as “the mental layer, including the user’s cognitive as well as emotional awareness”.

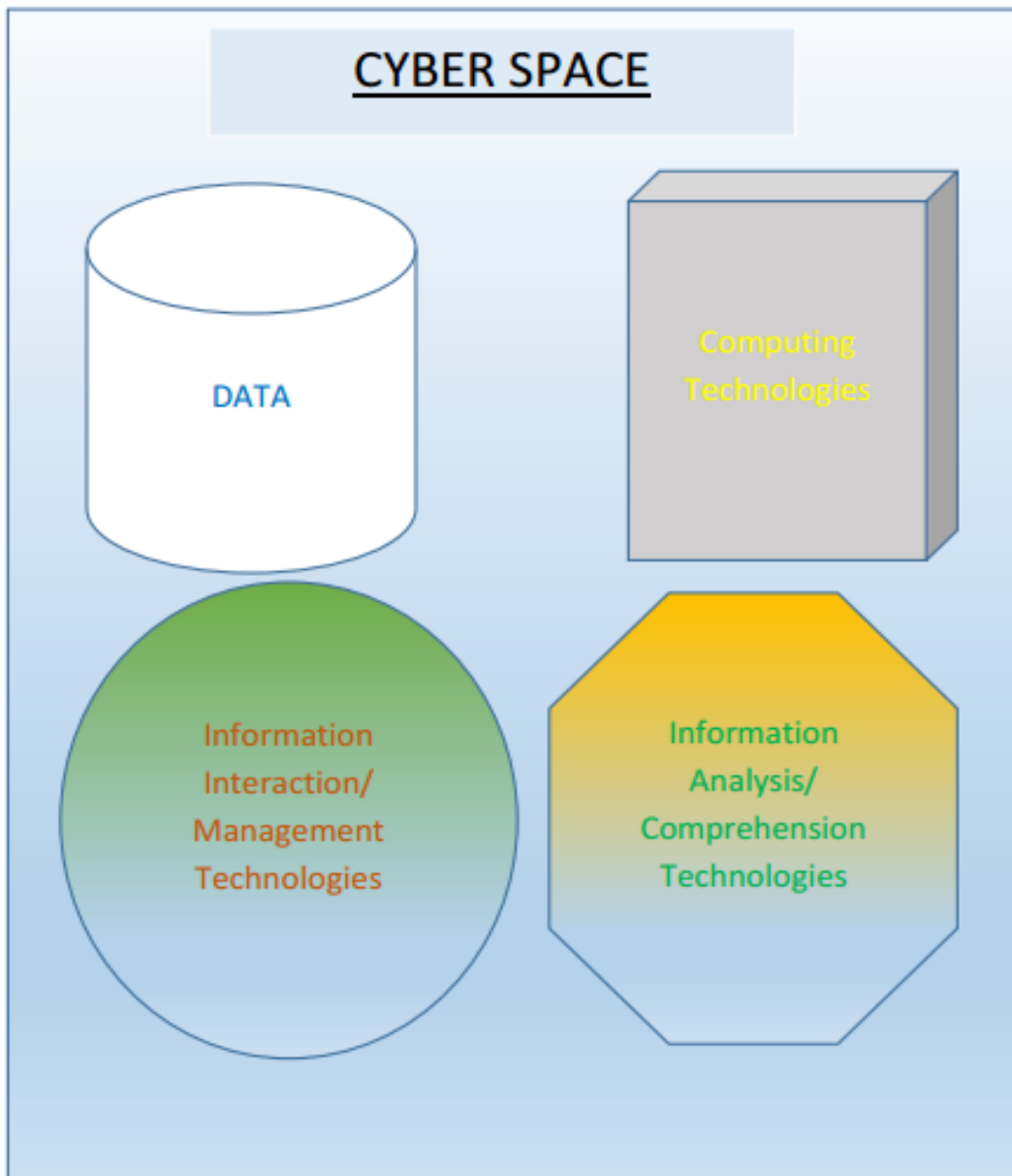


Figure 2.1: Four Major Elements of a Cyberspace (Stytz and Banks, 2014)

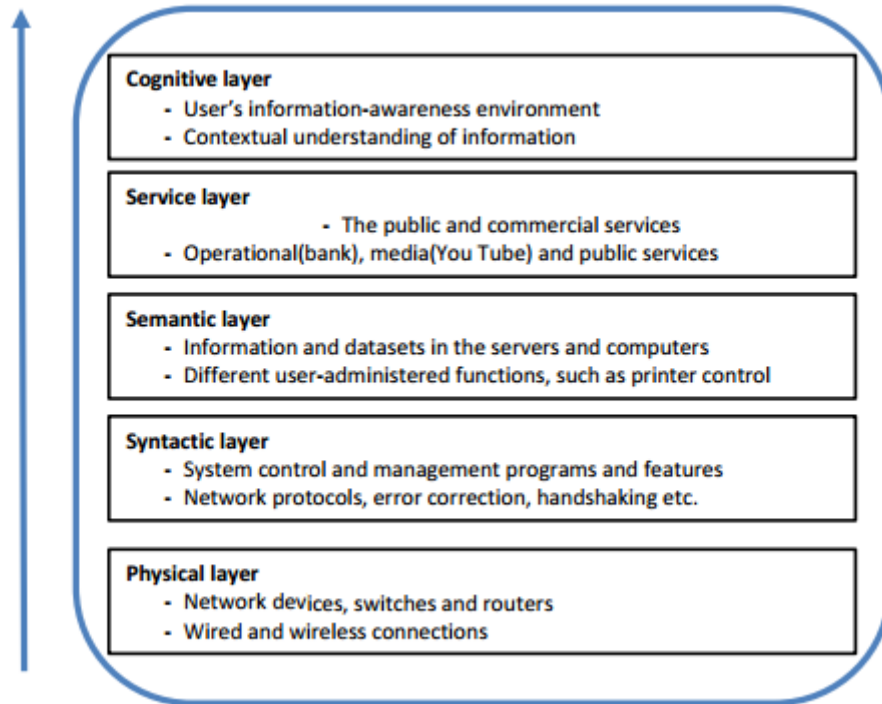


Figure 2.2: Five Layers of the Cyberspace (Lehto and Neittaanmäki, 2015)

Furthermore, (Schmitt, 2015) asserts that the definition of cyberspace should encompass four aspects of cyberspace: firstly it is an operational space which involves the use of cyberspace by people and organisations to act and create effects either solely in cyberspace or across other domains, secondly, a natural domain which is made up of electromagnetic activity that can be entered into using electronic technology, thirdly, information-based were activities like storage, modification, exchange and exploitation of information is carried out and lastly, interconnected networks which create room for electromagnetic activity to carry information. Based on this Kuehl's definition of cyberspace as "A global domain within the information environment whose distinctive and unique character is framed by the use of electronics and the electromagnetic spectrum to create, store, modify, exchange and exploit information via interdependent and interconnected networks using information-communication technologies" was adopted to be the most encompassing and communicates the unique aspect of cyberspace (Schmitt, 2015). The cyberspace is not owned by anybody or organisation, however, today, the control of cyberspace has become very important not only because of the actions of individual participants but because of the following:

- i. The infrastructure of cyberspace which is important to the functioning of national and international security systems, trade networks, emergency services, basic communications, and other public and private activities.

- ii. National governments now see potential threats to the security of their citizens and to the stability of their administrations arising within cyberspace.
- iii. A cyber-attack is an attack targeted at one or more of the following four elements Data, computing technologies, information analysis/comprehension technologies and information interaction/management technologies which the world today thrives on for decision making.

2.2.2 Forms of Cyber Threats

The general concern of security has to do with the disruption of Confidentiality, Integrity (which includes authenticity and non-repudiation) and Availability (CIA). However, cybersecurity involves threats that go beyond the disruption of CIA. According to (Gunneriusson and Ottis, 2013), “cyber threats come in the form of state actors, criminal groups, terrorist organisations, hacktivists, professional hackers for hire (i.e. mercenaries) that operate in a manmade environment, whereas, no man-made system is completely perfect”. Therefore, there are four areas of security concern regarding the cyberspace. They include Espionage, Crime, Cyberwar and Cyberterrorism. These four areas are categorized into two streams: Cyberwarfare and Cybercrime. These two streams were the focus in the deliberations between policy-makers in the United States (US) and international bodies on what rules should guide the behaviour of participants in cyberspace. The deliberation resulted into two streams of negotiations by the United Nations as; a politico-military stream which focused on cyber warfare and an economic stream which focused on cybercrime. The contents of each stream are as illustrated in Figure 2.3.

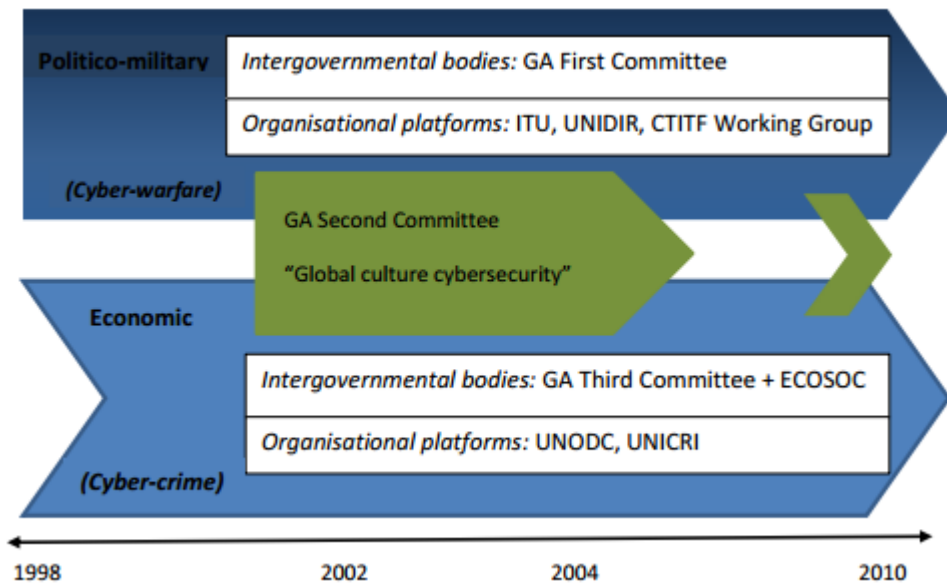


Figure 2.3: Two Streams Model of Cybersecurity Norm Emergence Process at the United Nations (Maurer, 2011)

Key

GA: General Assembly

ITU: International Telecommunication Union

UNIDIR: United Nations Institute of Disarmament Research

CTITF: Counter-Terrorism Implementation Task Force

ECOSOC: Economic and Social Council

UNODC: United Nations Office on Drugs and Crime

UNICRI: United Nations Interregional Crime and Justice Research Institute

i. Cyber Espionage

According to (Brumback, 2015), “Spying is said to be the world’s second oldest profession because it enhances gaining information advantage over competitors to ensures competitiveness and increases the likelihood of survival”. Cyber espionage also knew as cyberspying is the process of obtaining confidential information from individuals and organisations without permission using cyber means (especially Internet platform) for personal, economic, political advantage or for malicious intents. It can occur in different forms such as; traditional espionage which involves a government’s determinations to acquire secret information from another government, Economic espionage which involves a state’s attempts to acquire without permission trade secrets of foreign private enterprises and cooperate or industrial espionage which

involves a company acquiring illegally another company's trade secrets with no government involvement (Fidler, 2018).

There are three major reasons why states engage in cyber espionage (Hjortdal, 2011): to discourage other states by penetrating its critical infrastructure, to gain increased knowledge so as to advance more quickly in military development and to make economic gains. For instance, the Chinese government is believed to spy into U.S information to steal their trade secrets for economic advancement (Iasiello, 2014) and gain military Knowledge (Hjortdal, 2011). The Chinese contemplating having offensive and defensive cyberwar capacity created hacker groups of Chinese natives took several steps to defend its cyberspace and established cyber war military units. The U.S-China economic and security review commission estimates that there are up to 250 groups of hackers in China that are sophisticated enough to threaten U.S cyberspace and engage in extensive cyber espionage by establishing network spy stations not far from the U.S in Cuba that monitors U.S Internet traffic. This has led to one of the worst episodes of cyber espionage experienced by the U.S to date known as "*Titan Rain*". The incident involved the extraction of between 10 to 20 terabytes of data off the Pentagon's unclassified network and infiltrating U.S infrastructure with logic bombs (Hjortdal, 2011).

ii. Cyber Crime

Cybercrime "is an economic stream focused on the criminal misuse of information technologies" (Lagazio, Sherif, and Cushman, 2014). Cybercrime in its earliest stage involved physical damage to computers such as; crashes due to an electric power surge that might be caused by discontented and fraudulent employees and destabilization of telephone networks for personal gain or revenge. This later graduated to impersonation where Jerry Neal Schneider and Kevin Mitnick as teenagers were fully engaged in. Jerry used Dumpster diving to retrieve information about telephone and telegraphs which he later use to impersonate company personnel on the phone and make orders for the company while Kevin was breaking into computers using social engineering (that is, the use of deception to manipulate individuals into divulging confidential or personal information that may be used for fraudulent purposes) (Bachrach and Rzeszut, 2014)

Now cybercrime has become as sophisticated as information technology itself if not more. Criminals now use malicious codes to gain unauthorized access to computers for financial gain, political actions, and economic prowess.

iii. Cyber War

Cyberwarfare is “the unauthorized penetration by, on behalf of, or in support of, a government into another nations computer or network, or any other activity affecting a computer system, in which the purpose is to add, alter, or falsify data, cause the disruption of or damage to computers, or network devices, or the objects a computer system controls”. It is a virtual fight carried out in the cyberspace (the battlefield) using cyber weapons or techniques to exploit a rival in cyberspace to one’s advantage and cause damage to the exploited rival. China listed such weapon and techniques (Clark and Knake, 2010) as Planting information mines, conducting information reconnaissance, changing network data, releasing information bombs, dumping information garbage, disseminating propaganda, applying information deception, releasing cloned information, organizing information defence and establishing network spy stations. Debates have been on if actually there is a war called cyberwar, however, according to (Clark and Knake, 2010), “if a network is taken over, cyber warriors could steal all of its information or send out instructions that move money, spill oil, vent gas, blow up generators, derail trains, crash airplanes, send a platoon into an ambush, or cause a missile to detonate in the wrong place”. Therefore, cyberwar is possible because cyberwarriors can take over and crash the cyberspace in a flash bypassing the physical war front. Crashing the cyberspace can lead to data been wiped out, financial system collapse, supply chain halt, spinning satellite out of orbit into space and grounding of airlines.

Other reasons why Cyberwar is also possible include:

1. The vulnerability of the Internet: There are at least five major vulnerabilities in the design of the Internet itself (Clark and Knake, 2010). They include:
 - i. The addressing system: The domain name system which translates the address to a form the computer can understand (0’s and 1’s) can be hacked and information changed to misdirect to a phony webpage.

- ii. Routing among Internet service providers (ISPs): A system known as Border Gateway Protocol (BGP) is the main system used to route packets across the Internet. These packets carry the destination and source addresses of packets being transmitted. BGP decides the station the packet goes to next and it also establishes peer relationship between routers on different networks, however, "there are no mechanisms internal to BGP that protect against attacks that modify, delete, forge, or replay data, any of which has the potential to disrupt overall network routing behaviour" (Clark and Knake, 2010).
 - iii. Encryption: Almost everything that makes it work is unencrypted and most of the encrypted connections that transmits information such as password due to cost and speed are dropped back into insecure mode after password transmission.
 - iv. Secure traffic management: The Internet's ability to allow intentionally malicious traffic designed to attack computers, malware with a minimum if any check also makes it vulnerable. Most times the users are responsible for their own protection, most ISPs do not take responsibilities for keeping bad traffic from your computer because it is expensive, slows down the traffic and privacy concern.
 - v. Decentralized Design: The Internet is a network with a decentralized design. That is, it's intended not to be controlled singly or collectively by the government, therefore priority is placed on decentralization rather than security.
2. Flaws in hardware and software design: vendors in an attempt to build compatible and easy to use hardware and software-built software and hardware with flaws in design which make them vulnerable. This vulnerability can be exploited by cyber warriors.

iv. *Cyber Terrorism*

This is a terrorist attack carried out by capable offensive Internet Technology (IT) available through new computing technologies and network using cyber weapons such as malware to take down individual computers or computing devices or a nation's critical infrastructures without physical risk to the criminals. In summary, cybercriminal strives for financial gain, while the cyber warrior fights for military objectives and the cyber terrorist pursues personal agenda.

2.2.3 Cyber Attacks

Cyber-attack is an act in cyberspace that is expected to cause economic, psychological, physical, reputational and strategic harm (Kenney, 2015). Threats to cyberspace come in form of attacks, these attacks are clearly classified and illustrated in Figure 2.4.

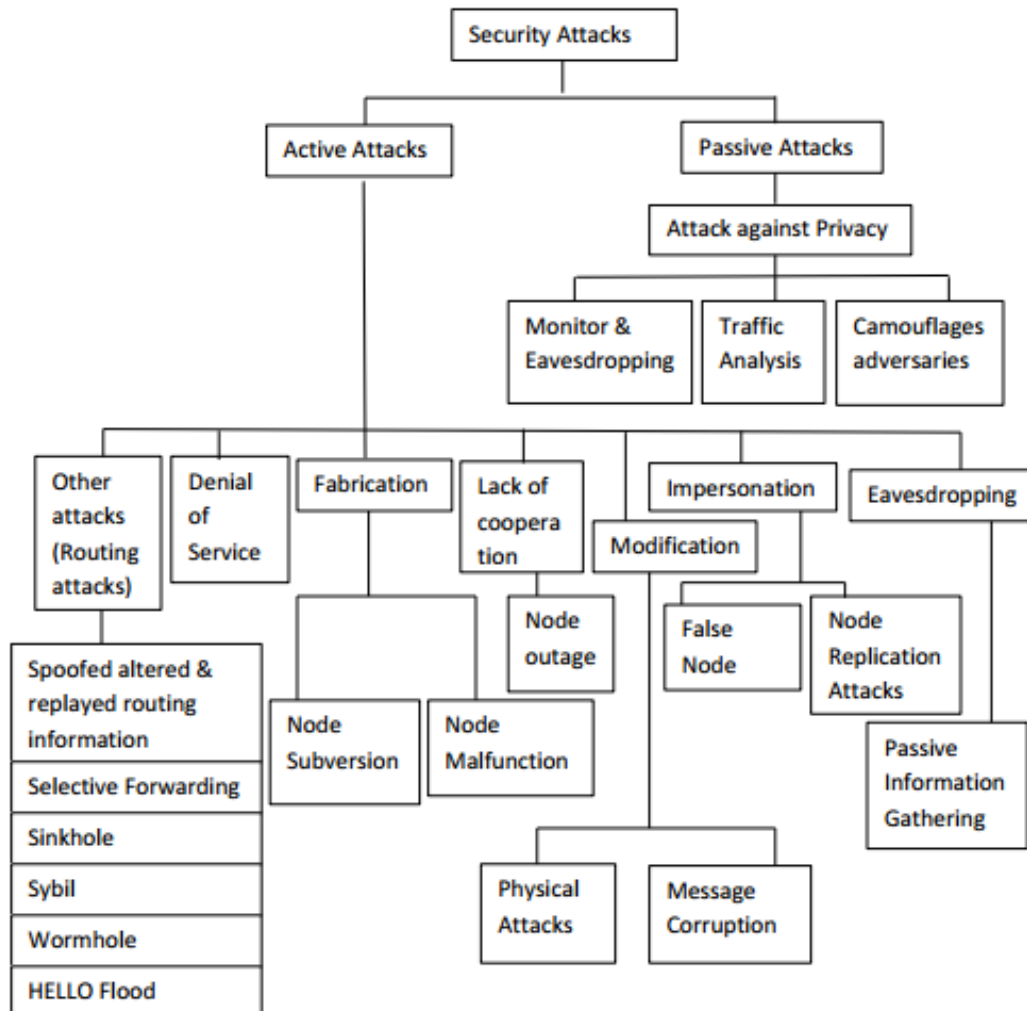


Figure 2.4: Example of Active and Passive Attacks (Sobh, 2013)

a. Active Attacks

An active attack destroys the data exchanged in the network and disrupts the regular functioning of the network (Indirani and Selvakumar, 2012). They are of two categories: internal and external attacks. Compromised nodes within the network execute internal attacks while remote nodes which are away from the network execute external attacks. Examples include Wormhole Attack, Black Hole Attack, Byzantine Attack, Information Disclosure and Resource Consumption Attack.

- i. Wormhole attack: It occurs when two malicious nodes share a private communication link between them. One node captures the traffic information of the network and sends them directly to another node. The link that exists between these two nodes is referred to a wormhole. The worm hole can eavesdrop traffics, maliciously drop packets, and perform man-in-the-middle attacks against the network protocols.
- ii. Blackhole attacks: This is achieved through the routing protocols when a route request is sent out, a particular intermediate node that wants to perform the blackhole attack send an intermediate node replay as having the route to the destination so as to attract the packet to it.
- iii. Byzantine attack: In this attack, individual or group of compromised nodes function together to create routing loops, forwarding packet through non-optimal paths and dropping of the packet using a selective approach which can result to a deprivation of some legitimate nodes of routing services. It is very complicated to identify these types of attack hence the name Byzantine attack.
- iv. The information disclosure attack: This attack is targeted at the privacy requirements of the network. Confidential information's like route location, node status or secret keys and password are captured and leaked out by the malicious node to the unauthorized nodes.
- v. Resource consumption attacks: Floods the bandwidth with meaningless data thereby overwhelming the network with resources and preventing legitimate users from having access to the network. They are also referred to as Denial of Service (Dos) attacks. DoS attacks may be as sophisticated as spoofing 802.11 disassociation management frames to the wireless terminals, or as simple as using a Radio Frequency (RF) generator in the 2.4 GHz band to jam the RF channel (Gharge, Halse, and Jagtap, 2013).
- vi. Replay attack: In this type of attack, the intruder passively monitors and captures transmitted packets between a wireless terminal and an access point using a utility such as a sniffer. Example of sniffers includes; air snort available as freeware on the Internet.

b. Passive Attacks

This type of attack intrudes the data exchange within the network without modifying it such that the regular behaviour of the network is not affected. They are very complicated to identify as the normal operation of the network is not affected. Examples are Snooping, Wi-Fi mooching, joyriding, Rogue Access Points (RAP), wardriving, warchalking etc.

- i. Snooping refers to the illicit use of another person's data. This may refer to watching e-mail informally that is displayed on another's computer screen or observing other people typing confidential information such as passwords. More complicated snooping involves using a software program to observe the process of a computer or network device (Indirani and Selvakumar, 2012).
- ii. Wi-Fi mooching is another passive attack, a mooch user connect through unsecured Wi-Fi.
- iii. Joyriders, joyriding is when a Wi-Fi belonging to subscribers are opened without their consent.
- iv. Rogue AP, this is an AP in the network that was not originally deployed by a network staff.
- v. Wardriving, this term originates from a phone hacking technique used in the 1980s known as "war dialing". It involves dialing every phone number in a specific sequence in search of modems (Sobh, 2013). In the same vein, wardriving attacks the RF signal of 802.11 since Cyber-networks extend beyond the confines of a building, with a wireless laptop or terminal, a hacker simply drives through business premises passively listening to get a strong RF signal. Without good security, little effort is then required to penetrate the network.
- vi. Warchalking is a practice of marking a series of symbols on walls along sidewalks to indicate nearby wireless access to users. Figure 2.4 illustrates more of these attacks.

Most cyber threats and attacks are enabled via malware (malicious software). Malware includes viruses, worms, Trojans, logic bombs, rootkits and spyware.

- i. Logic bomb: It is a malicious program deliberately written to produce results when certain conditions unexpected and unauthorized by legitimate

users of the time bomb which explodes at a set time by its designer. Example includes “the infamous Jerusalem virus (also known as the Friday the 13th virus) of 1988, it duplicated itself every Friday and on the thirteenth of the month, causing system slowdown on every Friday the 13th after May 13, 1988, it also corrupts all available disks on the infected system” (Bachrach and Rzeszut, 2014).

- ii. Trojan horses: These are malicious programs that pretend to be useful yet contain harmful codes or are outrightly harmful. Examples include Keylogger Trojans designed to capture keystroke logs, Haephtrati Trojan named after Michael Haephtrati who used it to steal money from bank accounts.
- iii. Worms and viruses: These are malicious codes that replicate themselves in the system they infect like biological worms and viruses. They can be spread through spam emails. The example includes; the Christmas tree worm, a spam mail represented by characters like the Christmas tree, Morris worm which was released into the Internet by a student called Robert T. Morris, Melissa and MS-Word macro virus. Once a document infected by this worm is loaded, it accesses the victims' emails address book and sends copies of itself to the first 50 people on the mailing list of its victim. The virus attaches an infected document to the email message with subject line “Subject: Important Message From<name> “the name is the name of the sender. The email message reads: “Here is that document you asked for...don’t show anyone else ;-)” and includes an MS-Word file as an infected attachment” (Bachrach and Rzeszut, 2014). I love you is another worm propagated through email from familiar contacts with the subject I LOVE YOU with the attachment LOVELETTER-FOR-YOU.txt.vbs which also infects the address book and mails itself to all contacts in the mailing list. Stuxnet is another worm designed for industrial espionage targeted at SCADA systems.

Cyber-attacks obviously have come to stay despite many preventive measures put in place to mitigate them.

c. Ransomware

Ransomware is a growing cyber threat in recent years that encrypts a user's files and holds the decryption key until a ransom is paid by the victim. Ransomware is a type of malware that uses malicious code that infects a computer and spreads rapidly to encrypt the data or to lock the machine (Jinal and Ashish, 2017). This malware makes the data inaccessible to the users and the attackers demand payment from the user to have their files unencrypted and accessible. The payment is often requested in Bitcoin (is a cryptocurrency and a payment system) or other invisible currency. Businesses and individuals worldwide are currently under attack by Ransomware (Zavarsky and Lindskog, 2016). Ransomware victimizes internet users by hijacking user files, encrypting them, and then demanding payment in exchange for the decryption key (Nolen, Henry, Patrick, Kevin, and 2016). Some most common methods used by cybercriminals to spread ransomware are Spam email campaigns that contain malicious links or attachments; Internet traffic redirects to malicious websites; Drive-by downloads, etc. Some security applications detect ransomware based on its activity such as File System Activities, Registry Activities, Device control Communications, Network Activity, and Locking mechanism (Zavarsky and Lindskog, 2016). Security firms are consistently developing and releasing anti-ransomware application and decryption tools in response to the threat. It locks the system or encrypts the data leaving victims unable to help to make a payment and sometimes it also threatens the user to expose sensitive information to the public if payment is not done. However, solutions may not always be present because some encryption is too difficult to break without the decryption key (Jinal and Ashish, 2017). In the event of an attack, organizations can minimize damage if they can detect the malware early. Business and individuals worldwide are currently under attack by ransomware. The main purpose of ransomware is to maximize the monetization using malware(Zavarsky and Lindskog, 2016). It has started doing more than just displaying advertisements, blocking service, disable keyboard or spying on user activities.

2.3 Cloud Computing Model

Cloud Computing Model is primarily made up of three service models, four deployment models and five essential characteristics (Gadia, 2009). Based on each model, the overall risks and benefits varies, hence it is important that enterprises consider the risks

that are associated with them when considering different types of service and deployment models.

2.3.1 Cloud Computing Service Model

The service model defines the three types of cloud computing or cloud services (Shovon et al., 2018). They include: Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a Service (PaaS)

i. Infrastructure as a Service (IaaS): This can also be called utility computing (Adamov and Erguvan, 2009). It has to do with delivering computing infrastructures as a service. The infrastructures can include IT resources like servers, storage spaces, network equipment and system software such as operating system and database systems. The major rationale behind it is to supply users with on-demand access to these IT resources and charge fees for their usage (De Assunção, Di Costanzo, and Buyya, 2009). Users only pay for the resources they utilize. The provisioning of the infrastructure is done using virtualization technology (a technology of abstracting computer resources) hence clients look and operate the infrastructure exactly like standard one, while in reality, it is one of many virtual environments hosted simultaneously on the same physical infrastructure resources. Amazon Inc., Sun, and IBM are key providers of this type of computing paradigm (Adamov and Erguvan, 2009).

ii. Software as a Service (SaaS): It is a software delivering paradigm where software is hosted off-premise and delivered to a large number of clients (users) through the web using subscription model as payment mode, SaaS presents some advantages to both users and the providers. Firstly, to the user, it means no upfront expenditures on server or software licensing; to the provider, it means low cost of maintenance compared to the conventional hosting since just one application is to be maintained (Adamov and Erguvan, 2009). Secondly, the SaaS business model help customers to avoid expensive, time wasting and laborious upgrade processes and the associated investment in training specialized personnel. Thirdly, deployment time, subscription and service revenue are accelerated with the ease with which customers can configure a solution that meets their needs.

iii. Platform as a Service (PaaS): Platform as a Service is an advancement that results from SaaS (Lawton, 2008). It is a type of cloud computing that delivers application development environment as a service. PaaS system supports the full software development life cycle: designing, implementation, testing, and deployment. As a Web-based application platform, developers, project managers, and testers are not required to go through the pains of downloading or installing any development software on their local computers. With PaaS, users can build their applications running on the provider's infrastructure and subsequently deliver them to their clients via the Internet from the provider's server. PaaS help programmers to increase their productivity thereby enabling companies to build and release products even more quickly with reduced cost of development. PaaS vendors are Google, Salesforce.com, Bungee Labs, Cog head among others (Lawton, 2008)

2.3.2 Cloud Computing Deployment Models

The cloud deployment model defines the following as possible deployment models (Gadia, 2009): Public cloud, Community cloud, Private cloud and Hybrid cloud . Table 2.1 gives a brief description of each of the cloud computing deployment models;

Table 2.1: Deployment model description

Deployment model	Description
Public cloud	<ol style="list-style-type: none">1. This is a cloud for the general public or a large industry group.2. It is owned by an organization selling cloud services or cloud vendors
Community cloud	<ol style="list-style-type: none">1. Allows sharing by several organizations2. Intended to support a specific community with a shared mission or interest3. Maybe managed by either the organization or a third party
Private cloud	<ol style="list-style-type: none">1. Solely operated by/for an organization.2. Maybe managed by either the organization or a third party
Hybrid cloud	<p>It is a composition of two or more cloud (public, community or private) that remain unique entities but bound together by standardized or proprietary technology that enables portability or data and application. An example is a cloud bursting for load balancing between clouds</p>

Source: (Gadia, 2009)

2.3.3 Cloud Computing Characteristics

The general characteristics of cloud computing include On-demand self-service, resource pooling, rapid elasticity, broad network access and measured service.

i. On-demand self-service: This characteristic avail cloud providers the ability to automatically provide computing capabilities or resources, such as server and network storage as needed without requiring human interaction with each service provider (Gadia, 2009).

ii. Resource Pooling: In order to serve multiple customers using a multi-tenant model with different physical and virtual resources dynamically assigned and reassigned based on demand, cloud vendor's resources are pooled. Generally, the customer/client has no control or knowledge over the very location of the provided resources (Gadia, 2009). Nevertheless, he/she may be able to specify location at a higher level of abstraction (e.g., country, region or data center). Examples of such resources include processing, storage, memory, network bandwidth, and virtual machines.

iii. Rapid Elasticity: With cloud computing, the provisioning of capabilities is made rapidly and elastically and, in many cases, automatic so as to scale out quickly and rapidly released to scale in quickly. The capabilities available for provisioning often appear to be unlimited to the customer/client and any quantity could be purchased at any time he/she wants.

iv. Broad Network Access: The National Institute of Standards and Technologies (NIST) asserts that the cloud network should be accessible anywhere, by almost any device (e.g. smartphone, laptop, mobile devices, PDA).

v. Measured Service: Cloud systems control and optimize resource usage automatically by leveraging a metering capability (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled and reported, providing transparency for both the provider and customer of the utilized service.

2.4 Cloud Computing Vendors

Cloud Computing began in earnest with the advent of Amazon's Web-based services (AWS), EC2, followed by Yahoo, Google, International Business Machines (IBM) and Microsoft. The most recent entries in the list of Cloud Computing vendors are Sun,

Intel, Oracle, SAS and Adobe (Buyya and Son, 2018). All these companies invested heavily in cloud computing infrastructure in order to provide vendor-based cloud services to the targeted market.

2.4.1 Benefits of Cloud Computing

Cloud computing presents a lot of benefits to businesses today especially with the increasing cuts on IT budgets resulting from the economic recession. With cloud computing today, enterprises can focus just on their core businesses rather than on scalability of infrastructures. Below are some of the benefits of cloud computing:

i. Dynamic Resource Loading: Cloud Computing allows a customer to subscribe to the number of IT resources (hardware and software) that is needed at any given time and dynamically increase or decrease this amount based on the changing requirements. The Cloud Computing model as said earlier actually treats hardware and software resources in a similar way as traditional utility resources such as electricity, gas, water, etc are treated. All the customers need to do is to define their resource requirement as the need arises and the cloud computing center satisfies them dynamically (Shovon et al., 2018).

ii. Higher Performance: Supercomputers are known to be in use in the Cloud Computing environment. No wonder then that Cloud Computing offers higher performance that is scalable to the users' changing needs. It really makes no much sense to purchase high-performance hardware for a single unit if the requirement for such hardware is periodical. This is unlike the Cloud Computing environment where such high computing power is sold to many clients, to each one for a specific time frame (Shovon et al., 2018).

iii. Professional Maintenance and Administration: Highly-trained staff is required for software and hardware maintenance and administration. It is always costly employing such people especially small organizations which may not be able to afford them. To make it worse, often, such highly-skilled IT personnel are not disposed to working for small countryside business owners. Such a problem is not obtainable in the world of Cloud Computing as hardware and software are maintained and managed by the highly-skilled IT personnel hired by Cloud Computing providers.

iv. Timely Software Updates: Cloud Computing is dedicated to providing clients with the latest software. This is done by continuously watching out for possible new

software updates and patches and applying them immediately as they are released. This actually saves customers the stress of having to buy, uninstall and install (as the case might be) of software on their local systems or servers. A typical scenario is in a SaaS model where the provider instantly updates the software to adjust it to changes in legal regulations.

v. Higher Level of Security: Cloud computing providers employ very professional and research-oriented staff in the area of security, for instance, to ensure higher security of hardware and software. This is rarely obtainable if not unobtainable, in multiple small business units.

vi. Good Practices Dissemination: Dissemination of good organizational and managerial practices is one major contribution of Cloud Computing apart from making modern and latest software available. There is diversification in offices and agencies in the public sector. The rich local governments are found in the large cities while the poor ones in the small countryside towns. Purchasing and deploying e-government solutions are usually not affordable for small local offices. They also lack highly-trained staff in areas such as maintaining and administering complex Internet-based solutions. Cloud computing helps to overcome these problems by ensuring a uniform level of e-government practices where electronic public services are uniformly deployed throughout the whole state and not for the rich local government alone (Shovon et al., 2018).

vii. Investment Cost to Operational Cost Shift: Organizations (public and private) spend much money buying hardware and software; when they do, they invest a large sum from the beginning (Shovon et al., 2018). With Cloud Computing, they need only purchase service which is paid for based on an agreed period (typically on monthly basis). There is, therefore, a shift from single high investment cost towards periodical operational costs which are stretched over a long time period.

viii. Low Startup Cost: Cloud Computing is very helpful in making 'infant' companies stand to their feet by offering them significant computing capability than they can otherwise afford (Kaufman, 2009). They do this using the concept of economies of scale (Foster, Zhao, Raicu, and Lu, 2008). For instance, at the beginning of a small and medium scale business, the owner may not have the resources to purchase in-house computers or even the relevant security, the cloud offers an alternative that is cost effective.

2.5 Cloud Security Solutions

One of the major worries expressed by users about the cloud is the one that borders on security. Apart from the application of encryption technology for secured communication between the cloud and the clients, the following are the emerging security measures in place in a cloud environment. They are very necessary for confidentiality, integrity, and availability of data (Gadia, 2009):

- i. Non-private data only should be stored in the cloud. Sensitive data like medical, financial, mortgage, insurance and military or intelligence records are best not processed or stored on public clouds. They are better off processed in-house so as to avoid the chances of compromising its integrity and confidentiality.
- ii. Administrators' actions and cloud key entry points should be audited and logged. This is to ensure accountability as actions can be traced to a role or an administrator.
- iii. Use of a service provider with the multi-location / multi-country presence or better still different cloud providers. This will go a long way alleviating the problem of locking up in a location.
- iv. Highly customized and transaction-heavy applications such as legacy applications should be retained in-house.
- v. Secure network connections for cloud administration.

2.6 Cloud Computing Management Platforms

There are several Cloud Computing Management Platforms in literature, however this section reviews only the platforms that are available in the marketplace.

2.6.1 Eucalyptus

This is an acronym for Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems. Eucalyptus is an open source software infrastructure for implementing on-premise clouds on existing Enterprise IT and service provider infrastructure (Endo, Gonçalves, Kelner, and Sadok, 2010). It provides support for the deployment of private and hybrid clouds for enterprise data centers. It is compatible with EC2 from Amazon since it is an open-source implementation of it (Nurmi et al., 2009).

a. Eucalyptus structure

Basically, Eucalyptus is made up of three components which define its topology structure. These include Cloud Controller (CLC), Cluster Controller (CC), and Node Controller (NC). Each of these has some sort of well-defined API in the form of WSDL API and also in the form of WSDL documents specifying the operations that the service can perform and the input/output data structures. Figure 2.5 is an illustration of the Eucalyptus structure components.

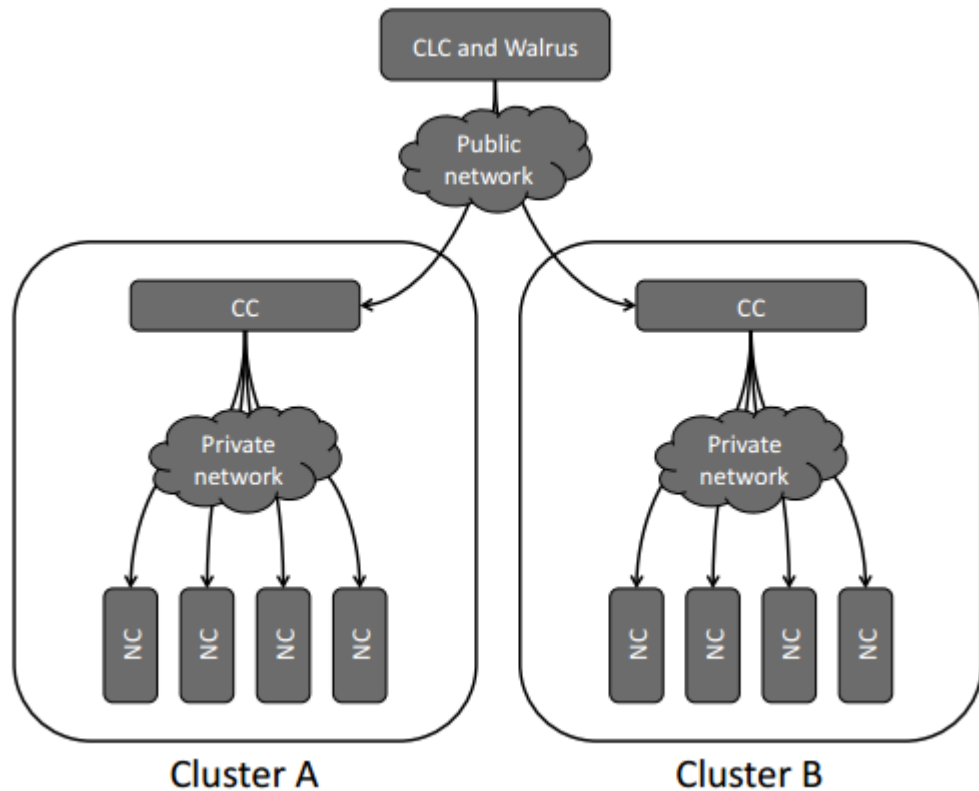


Figure 2.5: Eucalyptus structure (Endo et al., 2010; Peng et al., 2009)

i. Cloud Controller (CLC): It is the point through which cloud administrators, developers, project managers, and end users' interface with the cloud and its resources. It is used for querying the node managers for information about resources, high-level scheduling decisions making and implement them by sending requests to cluster controllers. In fact, the CLC is responsible for the exposure and management of the underlying virtualized resources (servers, network, and storage)

ii. Cluster Controller (CC): The Cluster Controllers (CCs) make up the front- end for each cluster defined in the cloud (Endo et al., 2010; Peng et al., 2009). They are generally either executed on cluster front-end machines or on any machine that has network connectivity to both the nodes running NCs and to the machine running the

CLC. CCs are responsible for information gathering about a set of VMs. They help schedules VM execution on specific NCs.

iii. Node Controller (NC): It is executed on any node earmarked for hosting VM instances (Endo et al., 2010). Hence, it is referred to as the machines on which virtual machine instances run. NCs are responsible for controlling the execution, inspection, and termination of VM instances on the host where it runs (Peng et al., 2009)

2.6.2 OpenNebula

OpenNebula is an open-source toolkit developed for the purpose of building any type of cloud with ease. Be it private, public or hybrid. It is designed in such a way that it can be integrated with any networking and storage solution and to fit into any existing data center (Milojčić, Llorente, and Montero, 2011). With OpenNebula, transforming data center into a flexible and agile virtual infrastructure such that it dynamically adapts to the changing demands of the service workload can be less tasking. Figure 2.6 is an illustration of the platform structure of OpenNebula.

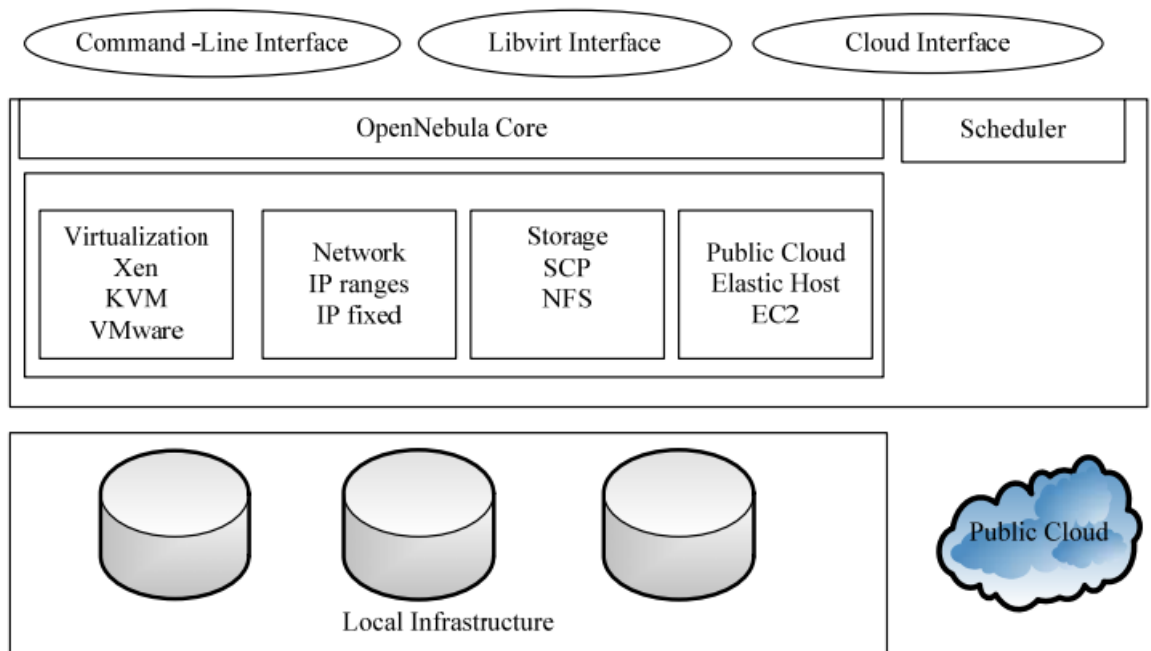


Figure 2.6: OpenNebula Platform (Haji, Letaifa, and Tabbane, 2010).

OpenNebula as an open and flexible virtual infrastructure (storage, server, and network) management tool can synchronize storage, network, and virtualization technologies

such that users can deploy services on the distributed infrastructure dynamically in accordance with the allocation policies at the data center and remote cloud resources.

a. *Benefits of OpenNebula*

OpenNebula has several benefits in terms of infrastructure management, usage and system integrations (Haji et al., 2013).

Infrastructure management: From the infrastructure point of view, OpenNebula enhances:

- i. Dynamic resizing of the physical infrastructure by adding new hosts, and dynamic cluster partitioning to meet capacity requirements of services.
- ii. Management of the entire virtual and physical distributed infrastructure from a centralized point.
- iii. Maximum utilization of existing resources by creating an infrastructure that incorporates the heterogeneous resources in the data center. This is evident in the structure diagram of Figure 2.6
- iv. Operational saving with the consolidation of the server to reduce the number of physical systems, thereby reducing space, administration effort, power and cooling requirements
- v. Lower expenses on infrastructure since it enables the combination of both local and remote Cloud resources, thereby eliminating the over-purchase of systems to meet peaks demands.

Infrastructure Usage: OpenNebula presents the following benefits to infrastructure users (clients):

- i. Services that are scalable and faster in delivery so as to cope with the spontaneous and dynamic nature of end-user's demand.
- ii. Complete control of the lifecycle of virtualized services management.

System Integration: OpenNebula presents the following benefits to system integrators:

- i. It fits into any existing data center because of its open, flexible and extensible interfaces, architecture and components
- ii. Builds any type of Cloud deployment (public, private or hybrid)

- iii. It is open source software, Apache license. This makes it quite easy and cheap to get
- iv. It supports and enhances seamless integration with any product and service in the virtualization/cloud environment (ecosystem) and management tool in the data center. Example of which include cloud providers, VM managers, virtual image managers, service managers, and management tools.

c. *Eucalyptus versus OpenNebula*

For a better understanding of these cloud management platforms or middleware for an informed choice of which to use, Table 2.2 gives a detailed analysis and comparison based on different aspects of implementation.

Table 2.2: Eucalyptus and OpenNebula compared

	Eucalyptus	OpenNebula
Cloud character	Public, private, hybrid	Public, private, hybrid
Scalability	Scalable	Dynamical, Scalable
Service type	IaaS	IaaS
Compatibility	EC2 and S3	Open, multi-platform
Development approach	Command line	Command line
VM support	VMWare, Xen, KVM, vBox	VMWare, Xen
Web interface	Web service	Libvirt, EC2, OCCI API
OS support	Linux	Linux
Development Language	Java	Java

Source (Haji et al., 2013)

2.6.3 *Nimrod*

The emergence of computational grids (cluster/couple of geographically distributed computer clusters as a single unified resource) has made popular the concept of the computational economy today. This provides an environment where users can hire or rent computing resources on the fly and hence pay for just what is used. With the computational grid, in order to get the best value for money, users at runtime can also

make a bid and negotiate for the best possible resources at low cost with service providers. Computational grids are very suited for large-scale parameter experiments/applications (e.g. simulations in Medicine and Engineering) due to the fact that it provides high-throughput computing needed by such experiments and the computational economy benefits (Abramson, Giddy, and Kotler, 2000). To manage the complexities associated with large-scale parametric computing on clusters of the computational grid, Nimrod is therefore needed (Buyya and Son, 2018).

Nimrod is an architecture for resource management and scheduling in the geographically distributed clusters of computing systems called computational grid. It provides a simple declarative and parametric model in which parametric experiments can be expressed (Buyya and Son, 2018). For instance, Nimrod is used in making the use of computational economy simpler by providing users with an interface that allows them to set constraints/parameters such as the deadline for application execution and price for completing the execution within the stipulated timeframe. It then collects each computing resource to meet the demand as specified in the user requirements/constraints by using the method of resource reservation and bidding.

2.7 Disaster Recovery (DR)

A typical disaster recovery service works by replicating application state between two data centers; if the primary data center becomes unavailable, then the backup site can take over and will activate a new copy of the application using the most recently replicated data. The primary goal of every DR system is to provide business continuity by allowing applications to swap over to a backup site while minimizing service disruptions.

2.7.1 Disaster Recovery Requirements

There are some fundamental requirements necessary for an effective disaster recovery service. Some of these requirements may be based on business decisions such as the monetary cost of system downtime or data loss, while others are directly tied to application performance and correctness. Keeton et al., (2006) listed these fundamental disaster recovery requirements as follows;

- i. Recovery Point Objective (RPO):* The RPO of a DR system represents the point in time of the most recent backup prior to any failure. The necessary RPO is generally a business decision. For some applications, absolutely no data can be lost (RPO = 0), requiring continuous synchronous replication to be used, while for other applications, the acceptable data loss could range from a few seconds to hours or even days.
- ii. Recovery Time Objective (RTO):* The RTO is an orthogonal business decision that specifies a bound on how long it can take for an application to come back online after a failure occurs. This includes the time to detect the failure, prepare any required servers in the backup site (virtual or physical), initialize the failed application, and perform the network reconfiguration required to reroute requests from the original site to the backup site so the application can be used. Depending on the application type and backup technique, this may involve additional manual steps such as verifying the integrity of the state or performing application specific data restore operations, and can require careful scheduling of recovery tasks to be done efficiently. Having a very low RTO can enable business continuity, allowing an application to seamlessly continue operating despite a site-wide disaster
- iii. Performance:* For a disaster recovery service to be useful, it must have a minimal impact on the performance of each application being protected under failure-free operation. DR can impact performance either directly such as in the synchronous replication case where an application writes will not return until it is committed remotely or indirectly by simply consuming disk and network bandwidth resources which otherwise the application could use.
- iv. Consistency:* The disaster recovery service must ensure that after a failure occurs the application can be restored to a consistent state. This may require the DR mechanism to be applied specifically to ensure that all relevant states are properly replicated to the backup site. In other cases, the DR system may assume that the application will keep a consistent copy of its important state on disk, and use a disk replication scheme to create consistent copies at the backup site.
- v. Geographic Separation:* It is important that the primary and backup sites are geographically separated in order to ensure that a single disaster will not impact both sites. This geographic separation adds its own challenges since increased distance leads to higher WAN bandwidth costs and will incur greater network latency. Increased round-trip latency directly impacts application response time when using synchronous

replication. As round-trip delays are limited by the speed of light, synchronous replication is feasible only when the backup site is within 10s of kilometers of the primary. Asynchronous techniques can improve performance over longer distances but can lead to greater data loss during a disaster. Distance can especially be a challenge in cloud-based DR services as a business might have only coarse control over where resources will be physically located.

2.7.2 Disaster Recovery Mechanisms

Disaster Recovery is primarily a form of long-distance state replication combined with the ability to start up applications at the backup site after a failure is detected. The amount and type of state that is sent to the backup site can vary depending on the application's needs. State replication can be done at one of these layers: (i) within an application, (ii) per disk or within a file system, or (iii) for the full system context. Replication at the application layer can be the most optimized, only transferring the crucial state of a specific application. For example, some high-end database systems replicate state by transferring only the database transaction logs, which can be more efficient than sending the full state modified by each query (Lahiri, Ganesh, Weiss, and Joshi, 2001). Backup mechanisms operating at the file system or disk layer replicate all or a portion of the file system tree to the remote site without requiring specific application of knowledge (Keeton et al., 2006). The use of virtualization makes it possible to not only transparently replicate the complete disk but also the memory context of a virtual machine, allowing it to seamlessly resume operation after a failure; however, such techniques are typically designed only for LAN environments due to significant bandwidth and latency requirements (Cully et al., 2008).

The level of data protection and speed of recovery depends on the type of backup mechanism used and the nature of resources available at the backup site. In general, DR services fall under one of the following categories:

- a. Hot Backup Site:** A hot backup site typically provides a set of mirrored standby servers that are always available to run the application once a disaster occurs, providing minimal RTO and RPO. Hot standbys typically use synchronous replication

to prevent any data loss due to a disaster. This form of backup is the most expensive since fully powered servers must be available at all times to run the application, plus extra licensing fees may apply for some applications (Wood et al., 2010). It can also have the largest impact on normal application performance since network latency between the two sites increases response time.

b. Warm Backup Site: A warm backup site may keep state up to date with either synchronous or asynchronous replication schemes depending on the necessary RPO. Standby servers to run the application after failure are available, but are only kept in a *warm* state where it may take minutes to bring them online. This slows recovery, but also reduces cost; the server resources to run the application needs to be available at all times, but active costs such as electricity and network bandwidth are lower during normal operation (Wood et al., 2010).

c. Cold Backup Site: In a cold backup site, data is often only replicated on a periodic basis, leading to an RPO of hours or days. In addition, servers to run the application after failure are not readily available, and there may be a delay of hours or days as hardware is brought out of storage or repurposed from test and development systems, resulting in a high RTO. It can be difficult to support business continuity with cold backup sites, but they are a very low-cost option for applications that do not require strong protection or availability guarantees. The on-demand nature of cloud computing means that it provides the greatest cost benefit when peak resource demands are much higher than average case demands. This suggests that cloud platforms can provide the greatest benefit to DR services that require warm stand-by replicas. In this case, the cloud can be used to cheaply maintain the state of an application using low-cost resources under ordinary operating conditions (Wood et al., 2010).

2.8 Security Issues in Service Delivery Models of Cloud Computing

Cloud computing utilizes three delivery models by which different types of services are delivered to the end user. The three delivery models are the SaaS, PaaS, and IaaS which provide infrastructure resources, application platform and software as services to the consumer (Gadia, 2009). These service models also place a different level of security required in the cloud environment. IaaS is the foundation of all cloud services, with

PaaS built upon it and SaaS in turn built upon it. Just as capabilities are inherited, so are the information security issues and risks. There are significant trade-offs to each model in terms of integrated features, complexity vs. extensibility and security. If the cloud service provider takes care of only the security at the lower part of the security architecture, the consumers become more responsible for implementing and managing the security capabilities. A recent survey by Cloud Security Alliance (CSA) and IEEE indicates that enterprises across sectors are eager to adopt cloud computing but that securities are needed both to accelerate cloud adoption on a wide scale and to respond to regulatory drivers. It also details that cloud computing is shaping the future of IT but the absence of a compliance environment is having a dramatic impact on its growth. Organizations using cloud computing as a service infrastructure, critically like to examine the security and confidentiality issues for their business-critical insensitive applications. Yet, guaranteeing the security of corporate data in the cloud is difficult, if not impossible, as they provide different services like SaaS, PaaS, and IaaS. Each service has its own security issues (Kandukuri and Rakshit, 2009).

SaaS is a software deployment model where applications are remotely hosted by the application or service provider and made available to customers on demand, over the Internet (Adamov and Erguvan, 2009). The SaaS model offers customers significant benefits, such as improved operational efficiency and reduced costs. SaaS is rapidly emerging as the dominant delivery model for meeting the needs of enterprise IT services. However, most enterprises are still uncomfortable with the SaaS model due to the lack of visibility about the way their data is stored and secured.

Security and Data backup policy concerns are the most commonly cited reason why enterprises are not interested in SaaS (Jacobs, 2005). Consequently, addressing enterprise security concerns has emerged as the biggest challenge for the adoption of SaaS applications in the cloud (Subashini and Kavitha, 2011). However, to overcome the customer concerns about application and data security, vendors must address these issues head-on. There is a strong apprehension about insider breaches, along with vulnerabilities in the applications and systems' availability that could lead to loss of sensitive data and money (Jacobs, 2005). Such challenges can dissuade enterprises from adopting SaaS applications within the cloud.

IaaS completely changes the way developers deploy their applications. Instead of spending big with their own data centers or managed hosting companies or collocation services and then hiring operations staff to get it going, they can just go to Amazon Web Services or one of the other IaaS providers, get a virtual server running in minutes and pay only for the resources they use. With cloud brokers like RightScale, enStratus, etc., they could easily grow big without worrying about things like scaling and additional security. In short, IaaS and other associated services have enabled startups and other businesses focus on their core competencies without worrying much about the provisioning and management of infrastructure. IaaS completely abstracted the hardware beneath it and allowed users to consume infrastructure as a service without bothering anything about the underlying complexities. The cloud has a compelling value proposition in terms of cost, IaaS only provides basic security (perimeter firewall, load balancing, etc.) and applications moving into the cloud will need higher levels of security provided at the host (Adamov and Erguvan, 2009).

PaaS is one layer above IaaS on the stack and abstracts away everything up to OS, middleware, etc. This offers an integrated set of developer environment that a developer can tap to build their applications without having any clue about what is going on underneath the service. It offers developers a service that provides a complete software development and lifecycle management from planning to design to building applications to deployment to testing and to maintenance. Everything else is abstracted away from the view of the developers. The dark side of PaaS is that these advantages can be very helpful for hackers to leverage the PaaS cloud infrastructure in deploying malware to command, control and even to go behind IaaS applications (Lawton, 2008).

2.8.1 Security Issues in SaaS

In SaaS, the client has to depend on the provider for proper security measures. The provider must do the work to keep multiple users' from seeing each other's data. So it becomes difficult for the user to ensure that the right security measures are in place and also difficult to get the assurance that the application will be available when needed (Choudhary, 2007). With SaaS, the cloud customer will, by definition, be substituting new software applications for old ones. Therefore, the focus is not upon portability of

applications, but on preserving or enhancing the security functionality provided by the legacy application and achieving a successful data migration (Seccombe et al., 2009). The SaaS software vendor may host the application on its own private server or deploy it on a cloud computing infrastructure service provided by a third-party provider (e.g. Amazon, Google, etc.). The use of cloud computing coupled with the pay-as-you-go approach helps the application service provider reduce the investment in infrastructure services and enables it to concentrate on providing better services to customers. Over the past decades, computers have become widespread within enterprises, while IT services and computing have become a commodity (Choudhary, 2007).

Enterprises today view data and business processes as strategic and guard them with access control and compliance policies (Jacobs, 2005). However, in the SaaS model, enterprise data is stored at the SaaS provider's data center, along with the data of other enterprises. Moreover, if the SaaS provider is leveraging a public cloud computing service, the enterprise data might be stored along with the data of other unrelated SaaS applications. The cloud provider might, additionally, replicate the data at multiple locations across countries for the purposes of maintaining high availability. Most enterprises are familiar with the traditional on-premise model, where the data continues to reside within the enterprise boundary, subject to their policies (Subashini and Kavitha, 2011).

Consequently, there is a great deal of discomfort with the lack of control and knowledge of how their data is stored and secured in the SaaS model. There are strong concerns about data breaches, application vulnerabilities, and availability that can lead to financial and legal liabilities. The layered stack for a typical SaaS vendor is critical aspects that must be covered across layers in order to ensure the security of the enterprise data is as highlighted below;

i. Data Security

In a traditional on-premise application deployment model, the sensitive data of each enterprise continues to reside within the enterprise boundary and is subject to its physical, logical and personnel security and access control policies. However, in the SaaS model, the enterprise data is stored outside the enterprise boundary, at the SaaS

vendor end. Consequently, the SaaS vendor must adopt additional security checks to ensure data security and prevent breaches due to security vulnerabilities in the application or through malicious employees. This involves the use of strong encryption techniques for data security and fine-grained authorization to control access to data (Jacobs, 2005). In cloud vendors such as Amazon, the Elastic Compute Cloud (EC2), Service providers do not have access to customer instances login details and cannot log in to their sessions. Every EC2 Administrator with business needs are required to use their individual cryptographically strong Secure Shell (SSH) keys to gain access to a host. All such accesses are logged and routinely audited. While the data at rest in Simple Storage Service (S3) is not encrypted by default, users can encrypt their data before it is uploaded to Amazon S3, so that it is not accessed or tampered with by any unauthorized party. Malicious users can exploit weaknesses in the data security model to gain unauthorized access to data. The following assessments test can be used to validate the security of the enterprise data stored at the SaaS vendor: Cross-site scripting [XSS], Access control weaknesses, SQL injection flaws, Cross-site request forgery [CSRF], Cookie manipulation, Hidden field manipulation, Insecure storage, and Insecure configuration. Any vulnerability detected during these tests can be exploited to gain access to sensitive enterprise data and lead to a financial loss (Jacobs, 2005).

ii. Network security

In a SaaS deployment model, sensitive data is obtained from the enterprises, processed by the SaaS application and stored at the SaaS vendor end. All data flow over the network needs to be secured in order to prevent leakage of sensitive information. This involves the use of strong network traffic encryption techniques such as Secure Socket Layer (SSL) and the Transport Layer Security (TLS) for security. In case of Amazon Web Services (AWS), the network layer provides significant protection against traditional network security issues, such as MITM (Man-In-The-Middle) attacks, IP spoofing, port scanning, packet sniffing, etc. For maximum security, Amazon S3 is accessible via SSL encrypted endpoints. The encrypted endpoints are accessible from both the Internet and from within Amazon EC2, ensuring that data is transferred securely both within AWS and to or from sources outside of AWS. However, malicious users can exploit weaknesses in network security configuration to sniff network

packets. The following assessments test and validate the network security of the SaaS vendor:

1. Network penetration and packet analysis
2. Session management weaknesses
3. Insecure SSL trust configuration.

Any vulnerability detected during these tests can be exploited to hijack active sessions, gain access to user credentials and sensitive data.

iii. Data Locality

In a SaaS model of a cloud environment, the consumers use the applications provided by the SaaS and process their business data. But in this scenario, the customer does not know where the data is getting stored. In many cases, this can be an issue. Due to compliance and data privacy laws in various countries, locality of data is of utmost importance in many enterprise architectures (Subashini and Kavitha, 2011). For example, in many EU and South America countries, certain types of data cannot leave the country because of potentially sensitive information. In addition to the issue of local laws, there is also the question of whose jurisdiction the data falls under when an investigation occurs. A secure SaaS model must be capable of providing reliability to the customer on the location of the data of the consumer.

iv. Data Integrity

Data integrity is one of the most critical elements in any system. Data integrity is easily achieved in a standalone system with a single database. Data integrity in such a system is maintained via database constraints and transactions. Transactions should follow ACID (atomicity, consistency, isolation, and durability) properties to ensure data integrity (Blaze, Feigenbaum, Ioannidis, and Keromytis, 1999). Most databases support ACID transactions and can preserve data integrity. Next, in the complexity chain are distributed systems. In a distributed system, there are multiple databases and multiple applications. In order to maintain data integrity in a distributed system, transactions across multiple data sources need to be handled correctly (Jacobs, 2005).

v. Data Segregation

Multi-tenancy is one of the major characteristics of enterprise application hosting. As a result of multi-tenancy, multiple users can store their data using the applications provided by service providers. In such a situation, data of various users will reside at the same location. In such, intrusion of data of one user by another becomes possible in his environment. This intrusion can be done either by hacking through the loopholes in the application or by injecting client code into the application server. A client can write a masked code and inject into the application. If the application executes this code without verification, then there is a high potential of intrusion into other's data. An enterprise application servers should, therefore, ensure a clear boundary for each user's data (Jacobs, 2005). The boundary must be ensured not only at the physical level but also at the application level. The service should be intelligent enough to segregate the data from different users.

A malicious user can use application vulnerabilities to handcraft parameters that bypass security checks and access sensitive data of other tenants. The following assessments test can be used to validate the data segregation of the SaaS vendor in a multi-tenant deployment: SQL injection flaws, Data validation and insecure storage (Jacobs, 2005). Any vulnerability detected during these tests can be exploited to gain access to sensitive enterprise data of other tenants.

iv. Data Access

Data access issue is mainly related to security policies provided to the users while accessing the data. In a typical scenario, a small business organization can use a cloud provided by some other provider for carrying out its business processes. This organization will have its own security policies based on which each employee can have access to a particular set of data. The security policies may entitle some considerations wherein some of the employees are not given access to a certain amount of data. These security policies must be adhered to by the cloud to avoid intrusion of data by unauthorized users (Blaze et al., 1999). The SaaS model must be flexible enough to incorporate the specific policies put forward by the organization. The model must also be able to provide organizational boundary within the cloud because multiple

organization will be deploying their business processes within a single cloud environment.

vii. *Authentication and Authorization*

Most companies, if not all, store their employee information in some type of Lightweight Directory Access Protocol (LDAP) servers. In the case of SMB companies, a segment that has the highest SaaS adoption rate, Active Directory (AD) seems to be the most popular tool for managing users (Paquin and Thompson, 2010). With SaaS, the software is hosted outside of the corporate firewall. Many a time, user credentials are stored in the SaaS providers' databases and not as part of the corporate IT infrastructure. This means SaaS customers must remember to remove/disable accounts as employees leave the company and create/enable accounts as employees come on board. In essence, having multiple SaaS products will increase IT management overhead. For example, SaaS providers can provide and delegate the authentication process to the customer's internal LDAP/AD server, so that companies can retain control over the management of users.

viii. *Data Confidentiality Issue*

The definitional borders of cloud computing are much debated today. Cloud computing involves the sharing of storage by users for their own information on remote servers owned or operated by others and accessed through the Internet or other connections (CAI and WANG, 2009). Cloud computing services exist in many variations, including data storage sites, video sites, tax preparation sites, personal health record websites and many more. The entire contents of a user's storage device may be stored with a single cloud provider or with many cloud providers. Whenever an individual, a business, a government agency, or any other entity shares information in the cloud, privacy or confidentiality must be properly managed.

ix. *Web application security*

SaaS is software deployed over the internet and/or is deployed to run behind a firewall in local area network or personal computer. The key characteristics include Network-based access to, and management of, commercially available software and managing

activities from central locations rather than at each customer's site, enabling customers to access application remotely via the Web.

SaaS application development may use various types of software components and frameworks. These tools can reduce time-to-market and the cost of converting a traditional on-premise software product or building and deploying a new SaaS solution. Examples include components for subscription management, grid computing software, web application frameworks, and complete SaaS platform products. One of the must-have requirements for a SaaS application is that it has to be used and managed over the web. The software which is provided as a service resides in the cloud without tying up with the actual users. This allows improvising the software without inconveniencing the user. Security holes in the web applications thus create a vulnerability to the SaaS application. In this scenario, the vulnerability can potentially have a detrimental impact on all of the customers using the cloud. The challenge with SaaS security is not any different than with any other web application technology, however, one of the problems is that traditional network security solutions such as network firewalls, network intrusion detection, and prevention systems (IDS and IPS), do not adequately address the problem. Web applications introduce new security risks that cannot effectively be defended against at the network level and do require application-level defenses.

Verizon Business in their Verizon Business 2008 Data Breach Investigation Report (Wade, Hylender, and Valentine, 2008) reported 59% of the breaches to involve hacking with the following breakdown:

1. Application/service layer-39%
2. OS/platform layer-23%
3. Exploit known vulnerability-18%
4. Exploit unknown vulnerability-5%
5. Use of backdoor-5%.

Attacks targeting applications, software, and services were by far the most common techniques, representing 39% of all hacking activity leading to data compromise. This followed a trend in recent years of attacks moving up the stack. Far from past, operating system, platform, and server level attacks accounted for a sizable portion of breaches. Eighteen percent of hacks exploited a specific known vulnerability while 5% exploited

unknown vulnerabilities for which a patch was not available at the time of the attack. Evidence of re-entry via backdoors, which enable prolonged access and control of compromised systems, was found in 15% of hacking-related breaches. The attractiveness of this to criminals desiring large quantities of information is obvious. SQL injection is one type of attack which makes the web application more vulnerable (Lee, Jeong, Yeo, and Moon, 2012). If the application is vulnerable to such type of attacks, the entire data behind the application is at risk. The data can either belong to the organization from where the attack is launched or it can also be private data of some other organization hosted in the same cloud. Since the web applications and SaaS are tightly coupled in providing services to the cloud users, most of the security threats of the web application are also posed by the SaaS model of the cloud (Lee et al., 2012). The Open Web Application Security Project has identified Top 10 security risks faced by web applications (Boberski, 2010). Those threats are:

1. Injection flaws like SQL, OS and LDAP injection
2. Cross-site scripting
3. Broken authentication and session management
4. Insecure direct object references
5. Cross-site request forgery
6. Security misconfiguration
7. Insecure cryptographic storage
8. Failure to restrict URL access
9. Insufficient transport layer protection.
10. Unvalidated redirects and forwards.

x. Data Breaches

Since data from various users and business organizations lie together in a cloud environment, breaching into the cloud environment will potentially attack the data of all the users. Thus the cloud becomes a high-value target (Kaufman, 2009). In the Verizon Business breach report blog (Cooper, 2008) has it that external criminals pose the greatest threat (73%), but achieve the least impact when compared with insiders' threat which poses less than (18%), but achieves the greatest impact. Though SaaS advocates claim that SaaS providers can provide better security to customers' data than

conventional means, insiders do not have direct access to databases, but it does not reduce the risk of insider breaches which can be a massive impact on the security. The SaaS providers' employees have access to a lot more information and a single incident could expose information from many customers. SaaS providers must be compliant with PCIDSS (Payment Card Industry Data Security Standards) (PCIDSS, 2009) in order to host merchants that must comply with PCIDSS.

xi. Vulnerability in Virtualization

Virtualization is one of the main components of cloud computing but it poses major security risks. Ensuring that different instances running on the same physical machine are isolated from each other is a major task of virtualization which is not met completely in today's scenario (Attanasio, 1973). The other issue is the control of administrator on the host and guest operating systems. Current VMMs (Virtual Machine Monitors) do not offer perfect isolation. Many bugs have been found in all popular VMMs. Virtual machine monitor should be root secured, meaning that no privilege within the virtualized guest environment permits interference with the host system. Some vulnerability has been found in all virtualization software which can be exploited by malicious, local users to bypass certain security restrictions or gain unauthorized privileges. For example, the vulnerability of Microsoft Virtual PC and Microsoft Virtual Server could allow a guest operating system user to run code on the host or another guest operating system.

xii. Availability

The SaaS application needs to ensure that enterprises are provided with service around the clock (Jacobs, 2005). This involves making architectural changes at the application and infrastructural levels to add scalability and high availability. A multi-tier architecture needs to be adopted, supported by a load-balanced application server. Resiliency to hardware/software failures, as well as to the denial of service attacks, needs to be built from the ground up within the application (Cully et al., 2008). At the same time, an appropriate action plan for business continuity (BC) and disaster recovery (DR) needs to be considered for any unplanned emergencies. This is essential to ensure the safety of the enterprise data and minimal downtime for enterprise applications (Jacobs, 2005).

With Amazon, for instance, the WS API endpoints are hosted on the same Internet-scale, world-class infrastructure that supports the Amazon.com retail site. Standard Distributed Denial of Service (DDoS) mitigation techniques such as synchronous cookies and connection limiting are used (Shovon et al., 2018). To further mitigate the effect of potential DDoS attacks, Amazon maintains internal bandwidth that exceeds its provider-supplied Internet bandwidth. These assessments test and validate the availability of the SaaS vendor.

1. Authentication weaknesses
2. Session management weaknesses.

Many applications provide safeguards to automatically lock user accounts after successive incorrect credentials. However, incorrect configuration and implementation of such features can be used by malicious users to mount a denial of service attacks.

xiii. Backup

The SaaS vendor needs to ensure that all sensitive enterprise data is regularly backed up to facilitate quick recovery in case of disasters (Jacobs, 2005). Also, the use of strong encryption schemes to protect the backup file is recommended to prevent accidental leakage of sensitive information and also to prevent the backup file from Man in the Middle (MITM) attack (Callegati, Cerroni, and Ramilli, 2009). In the case of cloud vendors such as Amazon, the data at rest in S3 is not encrypted by default. The users need to separately encrypt their data and backups so that it cannot be accessed or tampered with by unauthorized parties. The following assessment tests are used to validate the security of the data backup and recovery services provided by the SaaS vendor: Insecure storage and insecure configuration. Any vulnerability detected during these tests can be exploited to gain access to sensitive enterprise data stored in backups.

xiv. Identity Management and Sign-on Process

Identity management (IdM) or ID management is a broad administrative area that deals with identifying individuals in a system (such as a country, a network or an organization) and controlling the access to the resources in that system by placing restrictions on the established identities. The SaaS vendor can support identity management and sign on services using any of the following models (Kaufman, 2009).

Independent IdM stack: The SaaS vendor provides the complete stack of identity management and sign-on services. All information related to user accounts, passwords, etc. is completely maintained at the SaaS vendor end.

Credential synchronization: The SaaS vendor supports replication of user account information and credentials between enterprise and SaaS application. The user account information creation is done separately by each tenant within the enterprise boundary to comply with its regulatory needs. Relevant portions of user account information are replicated to the SaaS vendor to provide sign-on and access control capabilities. The authentication happens at the SaaS vendor end using the replicate credentials.

Federated IdM: The entire user account information including credentials is managed and stored independently by each tenant. The user's authentication occurs within the enterprise boundary. The identities of this user as well as certain user attributes are propagated on-demand to the SaaS vendor using federation to allow sign-on and access control.

The security challenges for adopting these models and the relative advantage: and disadvantages are listed in Table 2.3. The following assessments test can be used to validate the security of the identity management and sign-on process of the SaaS vendor: Authentication weakness analysis and Insecure trust configuration. Any vulnerability detected during these tests can be exploited to take over user accounts and compromise sensitive data.

Table 2.3: Security challenges in identity management [IdM] and sign-on process.

IdM and SSO model	Advantages	Disadvantages	Security challenges
Independent IdM stack	Easy to implement. No separate integration with enterprise directory.	The users need to remember separate credentials for each SaaS application.	The IdM stack should be highly configurable to facilitate compliance with enterprise policies: e.g. password strength, etc.
Credential Synchronization	Users do not need to remember multiple passwords.	Requires integration with enterprise directory. Has higher security risk due to transmissions of user credentials outside the enterprise perimeter.	The SaaS vendors need to ensure the security of the credentials during transit and storage and prevent their leakage.
Federated IdM	Users do not need to remember multiple passwords No separate integration with enterprise directory Low-security risk value as compared to credential synchronization	Relatively more complex to implement	The SaaS vendor and tenants need to ensure that proper trust relationships and validations are established to ensure secure federation of user identities

Source (Kaufman, 2009)

2.8.2 Security Issues in PaaS

In PaaS, the provider might give some control to the people to build applications on top of the platform. But any security below the application level such as host and network intrusion prevention, disaster recovery will still be in the scope of the provider and the provider has to offer strong assurances that the data remains inaccessible between applications (Blaze et al., 1999). PaaS is intended to enable developers to build their own applications on top of the platform. As a result, it tends to be more extensible than SaaS. This trade-off extends to security features of PaaS (Adamov and Erguvan, 2009).

2.8.3 Security Issues in IaaS

With IaaS, the developer has better control over the security as long as there is no security hole in the virtualization manager. Though in theory, virtual machines might be able to address these issues in practice, there are plenty of security problems (Attanasio, 1973; Gajek, Liao, and Schwenk, 2007). The other factor is the reliability of the data that is stored within the provider's hardware. Due to the growing virtualization of everything in the information society, retaining the ultimate control over data to the owner of data regardless of its physical location has become a topic of utmost interest. To achieve maximum trust and security on a cloud resource, several techniques would have to be applied (Descher, Masser, Feilhauer, Tjoa, and Huemer, 2009). The security responsibilities of both the provider and the consumer greatly differ between cloud service models. Amazon's Elastic Compute Cloud (EC2) infrastructure as a service offering, as an example, includes vendor responsibility for security up to the hypervisor, meaning they can only address security controls such as physical security, environmental security and virtualization security (Loganayagi and Sujatha, 2012). The consumer, in turn, is responsible for the security controls that relate to the IT system including the OS, applications, and data (Seccombe et al., 2009).

2.8.4 Impact of Deployment Model

IaaS is prone to various degrees of security issues based on the cloud deployment model through which it is being delivered. Public cloud poses a major risk whereas private cloud seems to have a lesser impact. Physical security of infrastructure and disaster management if any damage is incurred to the infrastructure (either naturally or intentionally), is of utmost importance. Infrastructure not only pertains to the hardware

where data is processed and stored but also the path where it is getting transmitted. In a typical cloud environment, data will be transmitted from source to destination through umpteen number of third-party infrastructure devices (Brooks et al., 2003). There is a high possibility that data can be routed through an intruder's infrastructure.

Although cloud architecture is an improvised technology, the underlying technologies remain the same. The cloud is just built over the internet and all the concerns related to security on the internet are also posed by the cloud. The basis of the cloud technology makes the consumer and provider reside at a different location and virtually access their sources over the Internet. Even if an enormous amount of security is put in place in the cloud, the data is transmitted through the normal underlying Internet technology. So, the security concerns which are threatening the Internet also threaten the cloud. But, in a cloud, the risks are overwhelmingly high. This is because of its vulnerability and the asset value of the resources the cloud. Cloud systems still use normal protocols and security measures that are used on the Internet but the requirements are at a higher extent. Encryption and secure protocols cater to the needs to a certain extent but they are not context oriented. A robust set of policies and protocols are required to help secure transmission of data within the cloud. Concerns regarding intrusion of data by external non-users of the cloud through the internet should also be considered. Measures should be set in place to make the cloud environment secure, private and isolated on the Internet to avoid cyber criminals attacking the cloud.

2.9 Related Works on Disaster Recovery Systems

Since the emergence of the cloud computing paradigm, researchers have conducted studies that are related to disaster recovery. In this section, the researcher takes a critical look at these different Disaster Recovery systems with special emphasis on the various algorithms they used, benefits and weaknesses of each system.

2.9.1 SecondSite: Disaster Tolerance as a Service

The SecondSite is a disaster tolerance as a service deployed over the cloud (Rajagopalan, Cully, O'Connor, and Warfield, 2012). This platform is intended to handle three challenges:

1. Reducing RPO

2. Failure detection
3. Service restoration.

For this reason, it uses three techniques:

1. Using storage to keep writes between two checkpoints: Checkpoints move between sites in a specific period. However, if a failure happens in this time, some data will be lost. For this reason, a Distributed Replicated Block Device (DRBD) is used to store replications in both synchronous and asynchronous modes
2. Using a quorum node to detect and distinguish a real failure: A quorum node has been designed to monitor primary and backup server. If replications have not been received by the backup site in the waiting time, the backup site sends a message to quorum node. In this case, if the quorum node receives a heartbeat from a primary node, it means the primary server is active and the replication link has a problem; otherwise the backup site will be activated.
3. Using a backup site: There is a geographically separated backup site which allows replicating groups of virtual machines through wide-area Internet links.

SecondSite increases ability to fast failure detection and also differentiate between network failures and host failures. Using DRDB, resynchronize storage can be done for recovering primary site without VMs interruption in the backup site. Although SecondSite is not suitable for stateless services, however, it increases availability for small and medium businesses.

2.9.2 Remus: High availability via asynchronous virtual machine replication

Remus is based on Xen hypervisor (Barham et al., 2003). It is a high availability cloud service to tolerate disaster using storage replication combined with live VM migration (Cully et al., 2008). In this system, protected software is encapsulated in the virtual machines to asynchronously replicate whole-system checkpoints in a backup site with a high frequency. It is assumed that both replicas are in the same local area network (LAN). Remus is aimed at three main goals:

1. Providing low-level service to gain generality
2. Transparency
3. Seamless failure recovery.

Remus uses an active primary host and a passive backup host to replicate checkpoints. All writes have to be stored in backup RAM until a checkpoint completes. Migrated Virtual machines execute on the backup only if a failure is detected. Remus usage consists of four stages:

1. Stop running VMs and propagate only changed states into a buffer
2. Transmission of buffered states into backup RAM
3. Send an ACK message to the primary host after checkpoint completion
4. Release the network buffer to external clients.

This system integrates simple failure detection into the checkpoint process. If checkpoints are not received by the backup site in an epoch, the backup site will be activated. On the other hand, if the backup response is not received during a specific period, the primary site will suppose a failure at the backup host. However, Remus increases performance overhead which leads to some latency, because it requires ensuring consistent replication. In addition, this system needs a significant bandwidth. Similarly, this platform focuses on Service Provider and their infrastructures.

2.9.3 Romulus: Disaster tolerant system based on kernel virtual machines

Romulus as a Disaster Recovery Solution is an extension of the Remus system (Caraman, Moraru, Dan, and Kristaly, 2009). It is based on the KVM hypervisor (Kivity, Kamay, Laor, Lublin, and Liguori, 2007). Romulus provides an accurate algorithm for disaster tolerant in seven detailed stages which are:

1. Disk replication and network protection
2. VM checkpoint
3. Checkpoint synchronization
4. Additional disk replication and network protection
5. VM replication
6. Replication synchronization
7. Failure detection and failover.

The flaw of Remus is that it uses one buffer to replicate writes between primary host and backup. If a failure occurs in this buffer before transferring checkpoint, it causes an inconsistency between the disk and VM state; and it can break fault tolerance of Remus. For this reason, Romulus uses a new buffer to replicate disk writes after any checkpoint. The second flaw is that network egress traffic cannot be released until

completely transferring checkpoint to storage backup host which can decrease system performance. However, Romulus uses a new egress traffic buffer to solve this problem. Romulus can tolerate failure in two situations:

1. On the fly: it consists disk and VM state replication into a new writes buffer during VM running.
2. Failover: the ability of service recovery after a disaster.

2.9.4 *Myriad: Cost-Effective Disaster Tolerance*

This work proposed a new approach for achieving disaster tolerance in large, geographically-distributed storage systems. The system, called *Myriad*, can achieve the same level of disaster tolerance as a typical single mirrored solution, but uses considerably fewer physical resources, by employing cross-site checksums (via erasure codes) instead of direct replication (Chang et al., 2002). It provides a disaster tolerant service with respect to resource allocation issue which is a challenge in DT services. Host and backup clusters are monitored by high availability controllers. Each cluster has three different controllers:

1. Storage controller: To control and manage the cluster storage.
2. Cluster controller: To manage IPs, centralized memory and CPU availability.
3. Node controller: To load, start and stop the VM.

Different nodes and also different clusters can communicate with each other for better resource allocation. For this purpose, backup cluster controller allocates a VM to a node. Then, node controller loads and starts the VM and allocates it to the primary host. Finally, primary node controller loads and starts the VM.

In this system, VM failover consists of two scenarios. The first scenario is cluster failure. In this situation, the backup cluster will be activated. Node failure is another scenario in which cluster controller releases VMs' IP and allocates a backup node to compose required VMs. This system is most useful for extended distance and metropolitan clusters because of low latency requirements.

2.9.5 *Kemari: Virtual machine synchronization for fault tolerance*

Kemari is a cluster system which tries to keep VMs transparently running in the event of hardware failures (Tamura, Sato, Kihara, and Moriai, 2008). Kemari uses the

primary-backup approach so that any storage or network event that changes the state of the primary VM must be synchronized in backup VM. This system has gained the benefits of the Lock stepping and the Checkpointing (Bressoud and Schneider, 1996). Two main approaches for synchronizing VM state include:

1. Less complexity compared to lock stepping approach.
2. It does not need any external buffering mechanisms which can affect the output latency.

2.9.6 RUBiS: Disaster Recovery as a Cloud Service

RUBiS, according to (Wood et al., 2010) is a cloud architecture aims at both Disaster Recovery and also minimizing costs with respect to Service Level Agreement. As shown in Figure 2.7, in ordinary operation, a primary data center including some servers and a database accomplish normal traffics. A cloud is in charge of disaster recovery with two types of resources;

1. Replication mode resources for getting backup files before a disaster which is active
2. Failover mode resources that will be activated only after a disaster

It is notable that service providers can rent inactive resources to other customers for revenue maximization. In the case of a disaster, leased resources must be released and allocated to the failover procedure.

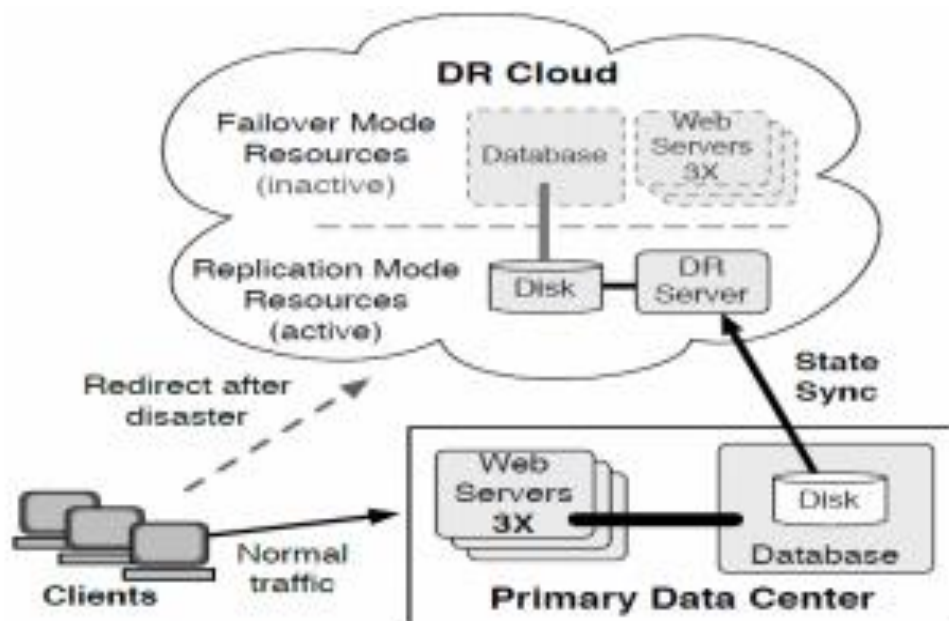


Figure 2.7: Overviews of RUBiS system architecture (Wood et al., 2010)

2.9.7 Hypervisor-Based Fault Tolerance (HBFT) System

This system is a Hypervisor-Based Fault Tolerant (HBFT) prototype which uses a mechanism similar to Remus (Zhu, Jiang, Xiao, and Li, 2011). However, instead of Remus which uses a separate local disk for replication, HBFT uses a Network Attach Storage (NAS). Shared storage may become a single point and cause a weakness of this method, so RAID or commercial NAS solution should be deployed (Patterson, Gibson, and Katz, 1988). On the other hand, because of using shared storage, the need for synchronizing is decreased and also file system state is maintained in the event of a disaster. Another major setback of this system is that it is only operational as service provider premises.

2.9.8 High Security Distribution and Rake Technology (HS-DRT) System

The goal of the HS-DRT system is protecting important data from natural or subversive disasters (Ueno, Miyaho, Suzuki, and Ichihara, 2010). As illustrated in Figure 2.8, this system uses an HS-DRT processor with a cloud computing system. Clients serve as terminals which request some web applications. The HS-DRT processor has functioned as a web application and also encryption, spatial scrambling, fragmentation of data. In the end, data is sent and stored in a private or public cloud. The system architecture is as shown in Figure 2.7. This system severely increases the security of data before and after the disaster in cloud environments. However, it has some weaknesses;

1. The performance of the web application will be decreased if the number of duplicated copies increases.
2. This system cannot guarantee consistency between different copies of file data.
3. Also it is only operational within service provider premises.

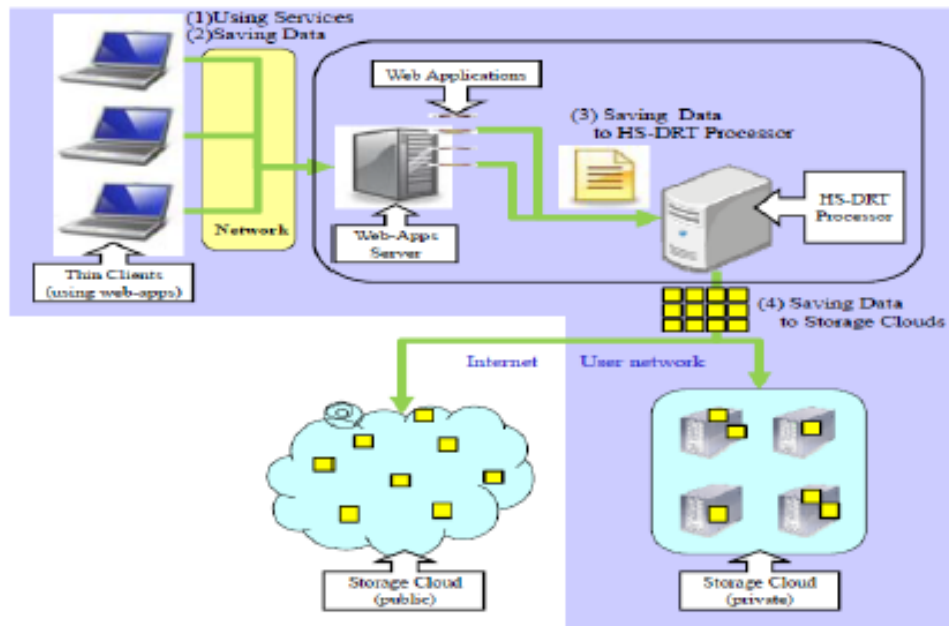


Figure 2.8: The architecture of the HS-DRT system (Ueno et al., 2010)

2.9.9 PipeCloud: using causality to overcome speed-of-light delays in cloud-based disaster recovery

This cloud-based multi-tier application system uses the Pipelined replication technique as a Disaster Recovery solution (Wood, Lagar-Cavilla, Ramakrishnan, Shenoy, and Van der Merwe, 2011). PipeCloud architecture is composed of a cloud backup site and a primary data center. The goal of this system is mirroring storage to the backup site and minimizing RPO. The main tasks of PipeCloud are;

1. Replicating all disk writes to a backup site by the replication technique
2. Tracking the order and dependencies of the disk writes
3. Releasing network packets only after storing the disk writes on the backup site.

This system results in a higher throughput and lower response time by decreasing the impact of WAN latency on the performance. For this purpose, the system overlaps replication with application processing. Also, it guarantees zero data loss consistency. However, unlike of Remus, PipeCloud cannot protect the memory states because it leads to large overhead on WAN. Also, it is only operational within service provider premises.

2.9.10 Knowledge as a service framework for disaster data management

A huge amount of disaster-related data has been generated by government, organization, automation systems, and even social media. This aims at providing

Knowledge as a Service KaaS framework for disaster data management which can lead to better preparation, response and recovery of disasters (Grolinger, Capretz, Mezghani, and Exposito, 2013). As shown in Figure 2.9, this system uses both relational and NoSQL databases to store data (Schram and Anderson, 2012). Disaster-CDM consists of two parts;

1. Knowledge acquisition: Obtaining knowledge from a variety of sources, processing and storing in data centers.
2. Knowledge delivery service: Merging information from diverse databases and delivering knowledge to users.

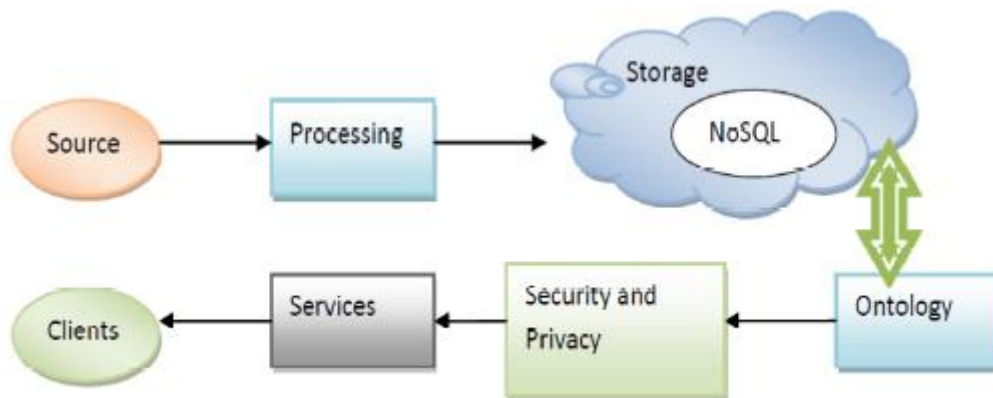


Figure 2.9: Disaster-CDM Framework (Grolinger et al., 2013)

2.9.11 Distributed Cloud System Architecture

Distributed Cloud System Disaster Recovery Architecture provides high dependability of the system based on severe redundancy (Ousterhout et al., 2010). The system has multiple data centers which are geographically separated from each other. Each data center includes both hot and warm physical nodes. VMs are active in both warm and hot physical nodes but only running in the hot nodes. In order for a Distributed Cloud System Architecture to be effective, there must be a backup server which stores a copy of each VM. When a physical node failure occurs, the VMs migrate to a warm physical node. In the case of a disaster which makes a data center unavailable, backup site transmits VM copies to another data center. Although this system architecture is expensive, it highly increases the dependability which can be adequate for Infrastructure as a Service (IaaS) clouds. Figure 2.10 shows the architecture of this DR system.

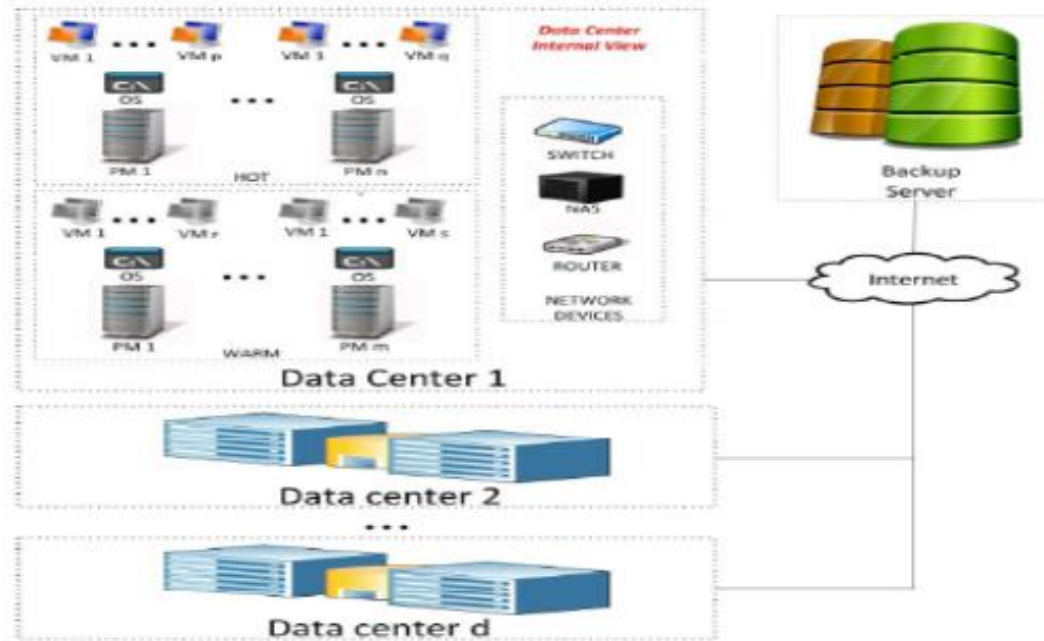


Figure 2.10: Distributed Cloud System Disaster Recovery Architecture (Ousterhout et al., 2010)

2.9.12 Oracle Recovery Manager (RMAN)

Oracle developed and integrated Recovery Manager (RMAN) into Oracle DBMS 12c (Kuhn, Alapati, and Nanda, 2013). With RMAN, one can perform the following types of backups;

1. Datafiles and control files
2. Server parameter file
3. Archived redo logs
4. RMAN backups

RMAN can store backup data in a logical structure called a backup set, which is the smallest unit of an RMAN backup. A backup set contains the data from one or more data files, archived redo logs, control files or server parameter file. Backup sets which are only created and accessed through RMAN, are the only form in which RMAN can write backups to media managers such as tape drives and tape libraries. RMAN is a very powerful and users' friendly tools; however, it is not compatible with other DBMSs and it does not perform a checksum on backup files. (Kuhn et al., 2013)

2.9.13 Backup as a Service from MTN – Cloud Storage Packages

MTN Backup as a Service (BaaS) is a prepaid hosting service that offers MTN subscribers the ability to back up their SIM and Phone securely onto backup servers in the MTN datacenter via the internet contacts (Paschal, 2016). The MTN BaaS is a service that provides MTN subscribers the ability to initiate and manage backups and restores on their phones virtually from anywhere in the world via a secure connection. The service requires subscribers to buy a new MTN 128k SIM Card with MTN Backup enabled, in order to back up the contacts stored on their SIM. The service is not compatible with other networks or MTN SIM that are not MTN Backup enabled. It is ideal for small-size data backup from mobile devices. It does not use the pay-as-you-go billing system.

2.9.14 Disaster Recovery as a Service

This is a Disaster Recovery Plan in the Cloud for SMEs (Rebah and Sta, 2016). The solution is based on the use of the virtual servers in the cloud which are started in case of disaster. Therefore, users are working on cloud mode with the saved data (i.e. replicated and installed on these virtual servers at the last automatic backup operation). When the problem that caused the accident is technically solved, data can be relocated to the company's servers and users can connect as usual. On the other hand, the solution proposed a new solution called "*DRaaS*" which benefits from the main advantages of Cloud and offers SME more beneficial solutions through improved quality recovery service and a significant decrease of costs due to low prices of data storage and pay per use. The effectiveness of this solution was also justified by listing several researches offering DR models in the cloud and the feedback from several companies.

Cloud computing is an economic and technological revolution that is being increasingly imposed on SMEs especially in terms of data storage and disaster recovery (Rebah and Sta, 2016).

2.9.15 Embedded Backup System

The work focused on the design and implementation of an embedded backup system which uses a hard disk and an embedded development board to integrate a backup-

server and a backup-client. It also uses increment backup in non-fixed block-level, and provides a friendly Web UI (user interface) to the end-user (Liu, Liu, and Yang, 2009). The uses checksums algorithm to compare backup files. However, it does not encrypt backup file and this could expose the files to Man in the Middle (MITM) attack.

2.9.16 Cross Platform Backup System

This work as proposed by (Chen and Zheng, 2008) is a design of XML-based GUI for cross platform backup system, in which the interface definitions are based on XML storage format. The implementation of the work achieves the plug-in management for centrally backup/restore processing of heterogeneous systems. The system is compatible with most DBMS. It also provides friendly user interfaces. However, it lacks system security since it does not encrypt data. Data leakages are very possible in multi tenant's environment.

2.9.17 Auto-Data Recovery System on Cloud Scheme

This scheme offers data storage and sharing services to users (Dhane and Joshi, 2015). The system uses a Trusted Third-Party Auditor (TTPA) to publicly audit the integrity of shared data in the cloud for users. In a group, there is one original user and a number of group users. The original user is the original owner of data. This original user creates and shares data with other users in the group through the cloud. Both the original user and group users are able to access, download and modify shared data. Shared data is further divided into a number of blocks. A user can modify a block in shared data by performing an insert, delete or update operation on the block. However, the system lack adequate data security and prevention from Man in the Middle (MITM) attack.

2.9.18 Server Virtualization Data Storage and Backup System

(Mayur and Vani, 2016) integrated server using private cloud model for backup and data storage. They developed an application for the implementation in such way that automatically synchronizes all information backed up or stored by the user in the virtual folder to the cloud. They used Parallel Virtual File System (PVFS) for data storage in order to increase the performance of applications that requires high I/O data demands. PVFS is an open source file system. The system was able to reduce access time to data

by allowing both input and output operations to be carried out simultaneously. On the client-side, an application is developed that allows data to be transferred much faster. The advantages of this implementation are that it can reuse existing infrastructure (servers, cluster, and other devices) that reduces the cost and increases the throughput

2.9.19 HDFS Backup and Disaster Recovery System

In this work, a system for Hadoop Distributed File System (HDFS) was introduced to backup and protect data from all kind of damages (Luo, Wang, Huang, and Yu, 2016). This system backups not only the metadata but also the real data of file system to a remote backup server, and can be applied to any other file systems which implements HDFS. The system achieved its primary aim by implementing a backup system built for HDFS with little effect to normal service, low cost for backup generation, high speed for backup transmission, and user-friendly interaction. Many practical features are also imbedded in the system.

2.9.20 Dependability models for designing disaster tolerant cloud computing systems.

In this work, models are presented for dependability evaluation of cloud computing systems deployed into geographically distributed data centers as well as taking into account disaster occurrence (Silva, Maciel, Tavares, and Zimmermann, 2013). The approach is based on a hybrid modelling technique, which considers combinatorial and state-based models. The proposed technique allows the impact assessment of disaster occurrence, VM migration and data center distance on system dependability. With this system, it is possible to run multiple VMs in the same host. The system aims at reducing VM migration times and distances between data centers. However, it does not encrypt or compress backup files. Also, the system is only operational as service provider premises.

2.10 Summary of Related Works

All of the reviewed related works are as summarized in Table 2.3;

Table 2.3: Literature Review Table

S/N	SOURCE	TITLE	PROBLEM ADDRESSED	METHODOLOGY/EVALUATION PROPERTIES	STRENGTH/RESULTS	WEAKNESS/COMMENT
1.	(Cully et al., 2008)	Remus: High availability via asynchronous virtual machine replication	To provide high availability by presenting machine replication as a service at the virtualization platform layer	In this system, protected software is encapsulated in the virtual machines to asynchronously replicate whole-system checkpoints in a backup site with a high frequency	Providing low-level service to gain generality, Transparency and Seamless failure recovery	It is only operational within service provider premises
2.	(Rajagopalan et al., 2012)	SecondSite: Disaster tolerance as a service	To Reduce Replication Overhead	In this system, Distributed Replicated Block Device (DRBD) is used to store replications in both synchronous and asynchronous modes	Reducing RPO, Failure detection and Service restoration	It is not a good choice for such stateless services. Also, it is only operational within service provider premises
3.	(Caraman et al., 2009)	ROMULUS: Disaster tolerant system based on kernel virtual machines	Romulus can tolerate failure in two situations: On the fly and Failover	Romulus uses a new egress traffic buffer to replicate disk writes after any checkpoint	Romulus provides an accurate algorithm for disaster tolerant	It is only operational within service provider premises

4	(Chang et al., 2002)	Myriad: Cost-Effective Disaster Tolerance	This work proposed a new approach for achieving disaster tolerance in large, geographically-distributed storage systems	The system employed cross-site checksums algorithms (via erasure codes) instead of direct replication	Myriad used considerably fewer physical resources	It does not compress backup files. And it is only operational at service provider premises
5.	(Tamura et al., 2008)	Kemari: Virtual machine synchronization for fault tolerance	To Provide a cluster system that synchronizes VMs for fault tolerance	Kemari uses the primary-backup approach so that any storage or network event that changes the state of the primary VM must be synchronized in backup VM.	Kemari offers a feasible approach to fault tolerance that does not require the use of specific hardware or modification of applications	. Also, it is only operational within service provider premises
6.	(Wood et al., 2010)	RUBiS: Disaster Recovery as a Cloud Service	It aims at minimizing costs with respect to Service Level Agreement	It uses a primary data center including some servers and a database to accomplish normal traffics. While a cloud-based system is in charge of disaster recovery	It permits service providers to rent inactive resources to other customers for revenue maximization.	Sometime, leased resources may not be available to resolve disaster-related challenges. Also, it is only operational within service provider premises
7.	(Zhu et al., 2011)	Optimizing the performance of virtual machine	Optimizing the performance of the virtual machine	Network Attach Storage (NAS)	It reduces the need for synchronizing and thereby increases the	The use of Shared storage is a major issue in this work. Also, it is only

		synchronization for fault tolerance			performance of the virtual machine.	operational within service provider premises
8.	(Ueno et al., 2010)	Performance evaluation of a disaster recovery system and practical network system applications.	To increases the security of data before and after the disaster in cloud environments	This system uses an HS-DRT processor with a cloud computing system. Clients serve as terminals which request some web applications.	Its system was able to achieve fragmentation and encryption of backup files	This system cannot guarantee consistency between different copies of backup files. Also, it is only operational within service provider premises
9.	(Wood et al., 2011)	PipeCloud: using causality to overcome speed-of-light delays in cloud-based disaster recovery	The goal of this system is mirroring storage to the backup site and minimizing RPO	This cloud-based multi-tier application system uses the Pipelined replication technique as a Disaster Recovery solution	This system results in a higher throughput and lower response time by decreasing the impact of WAN latency on the performance	PipeCloud cannot protect the memory states. Also, it is only operational within service provider premises
10.	(Grolinger et al., 2013)	Knowledge as a service framework for disaster data management	This aims at providing Knowledge as a Service (KaaS) framework for disaster data management	It uses both relational and NoSQL databases to store the huge amount of disaster-related data have been generated by government, organization, automation systems, and even social media	It gave information that can lead to better preparation, response and recovery of disasters	It only provides the necessary information on disasters. It lacks technical implementation of disaster recovery

11.	(Ousterhout et al., 2010)	The case for RAMClouds: scalable high-performance storage entirely in DRAM	To increase dependability system backups by using Infrastructure as a Service (IaaS) model on the clouds	The system uses multiple data centers which are geographically separated from each other. Each data center includes both hot and warm physical nodes.	It highly increases the dependability of backup servers in event of a disaster	It is quite expensive and not affordable by service users
12.	(Kuhn et al., 2013)	RMAN Recipes for Oracle Database 12c: A Problem-Solution Approach	To provide a user-friendly environment for Oracle database back up.	RMAN store backup data in a logical structure called a backup set, which can contain data from one or more data files and server parameter file.	RMAN is a very powerful and users' friendly tools	It is not compatible with other DBMSs. It does not perform a checksum on backup files. It does not encrypt backup files.
13.	(Paschal, 2016)	Backup as a Service from MTN – Cloud Storage Packages	To enable SIM and Phone to be securely backed up on MTN datacenter	The system creates backup files, encrypts and stores them in MTN datacenter. The backup files are decrypted whenever the restoration module is invoked.	It is a very powerful and users' friendly tool for MTN subscribers	It is ideal for small-size data backup from mobile devices. It does not use the pay-as-you-go billing system. It also has a compatibility issue
14.	(Rebah and Sta, 2016)	Disaster Recovery as a Service: A Disaster Recovery Plan in the Cloud for SMEs	To deliver a cost-effective disaster recovery system for Small and Medium Scale Enterprises	The system created backup files and store them on them on virtual cloud servers. It falls back on these back files in event of a disaster.	It uses a pay-as-you-go billing model which makes it very affordable to SMEs	It does not encrypt backup files. Also, it is only operational within service provider premise

15.	(Liu et al., 2009)	Design and implementation of an embedded backup system.	The system aimed at reducing the duplication of backup files by using checksums algorithm to compare backup files.	The system used a hard disk and an embedded development board to integrate a backup-server and a backup-client. It also uses increment backup in non-fixed block-level	It uses checksums algorithm to compare backup files. It provides a friendly Web UI (user interface) to the end-user	It does not encrypt backup files.
16.	(Chen and Zheng, 2008)	Design and implementation of XML-based GUI for a cross-platform backup system	The system aimed at designing a general plug-in management system for backup/restore	The system defines system backup/restore tasks and saved them in tables of a database in terms of XML codes.	It is compatible with most DBMS. It also provides friendly web user interfaces	Data security and prevention from Man in the Middle attack. Data leakages are very possible in multi-tenants' environment
17.	(Dhane and Joshi, 2015)	Public Auditing System with Auto-Data Recovery System on Cloud Scheme	The system proposed a public auditing scheme for a regenerating-code-based cloud storage system, where the data owners are privileged to delegate TTPA for their data validity checking	The system uses a Trusted Third-Party Auditor (TTPA) to publicly audit the integrity of shared data in the cloud for users	The performance evaluation shows that our scheme is highly efficient and can be feasibly integrated into a regenerating code-based cloud storage system.	Data security and prevention from Man in the Middle attack

18.	(Mayur and Vani, 2016)	Server Virtualization using Cloud Environment for Data Storage and Backup	The system aims to increase the performance of applications that require high I/O data demands	The system used the Parallel Virtual File System (PVFS) and private cloud model for backup and data storage	The adoption of Parallel Virtual File System (PVFS) increases the performance of the system	It is only operational within service provider premise
19.	(Luo et al., 2016)	Backup and Disaster Recovery System for HDFS.	This system aims at creating backups for both metadata and the real data	The system used the Hadoop Distributed File System (HDFS) to backup and protect data from all kind of damages.	The backing up of both metadata and real data is a great achievement	It is not compatible with most files system and DBMS.
20.	(Silva, Maciel, Tavares, and Zimmermann, 2013)	Dependability models for designing disaster tolerant cloud computing systems.	The system aims at reducing VM migration times and distances between data centers	The work an approach which is based on a hybrid modelling technique, which considers combinatorial and state-based models	The system permits the execution of multiple VMs in the same host.	It does not encrypt or compress backup files It is only operational within service provider premise

2.11 Summary of Literature Review and Knowledge Gap

Most of the related works used the principle of server replication combined with live VMs migration to create replicates of the primary servers from time to time and store them in backup servers. These replicates contain data and applications of many users whose applications are hosted on these servers. This explains why such replicates cannot be handed over to any service user.

Service providers set a single server replication frequency which is then applied to be the backup frequency of all the applications running on such server. This again is a challenge. Some mission-critical applications may be running on such a server and the owners of such applications may require that their databases be backed up more frequently than the default settings. On the other hand, some applications may even consider the default replication frequency to be too high.

The researcher also considers servers replication as a waste of resources as most time entities that have not being updated after the last backup any change is always replicated along with the other entities at every replication process. However, if users are allowed to configure their backup settings, they are in a better position to know the entities to back up without wasting system resources.

Again, hidden and irregular charges for backup services is one of the knowledge gaps identified. Most of the service providers offers flat charges system. Users are charged flat rate with little or no attention paid on the size of data been backed up for the users. For instance, Backup as a Service from MTN Cloud Storage Package charges as much as N 150.00 per month just to backup ordinary phone contacts which in most cases are far less than 10Kb of data (Paschal, 2016). Also, the same amount of money is charged even for customers with bigger size of data.

The unavailability of [checksumming](#)^[H1] mechanism in most of disaster recovery solutions is one of the knowledge gaps identified. The researcher is of the view that checksumming algorithm should be adopted in all backup process to avoid the situation of backing up the same instances of databases at every predetermined backup frequency. Most of the current solutions normally backup databases without

checksumming to know whether there had been any appreciable updates on the database between the previous backup process and the next backup process. Involving checksumming algorithms in backup process does not just reduced resource usage, it also reduces operation cost on the side of service users.

Finally, the storage services provided by one service provider are not incompatible with another service provider. For instance, Microsoft cloud storage service is incompatible with Google cloud storage service (Basu et al., 2018; Popović and Hocenski, 2010). As a result of this incompatibility in storage services systems, it is very difficult for service users to transfer their applications from one service provider to other in the phase of disaster without losing a chunk of their sensitive data. Thus, a User-Centric Cyber Disaster Recovery Model that permits to define and implement their private disaster recovery policy is required; hence this study.

CHAPTER THREE

SYSTEM ANALYSIS AND METHODOLOGY

3.1 System Analysis

In this section, a clear and details analysis shall be carryout on both existing and proposed disaster recovery solutions.

3.1.1 Analysis of the Existing System

Several disaster recoveries models have been developed over the years and series of research are still going on to improve on the functionalities of these models or even to develop new and better models. Most of these disaster recovery models employ server replication and mirroring methods. With these methods, the entire server hosting many applications is replicated at a preset interval of time. Such replicates are always in the custody of service providers since it contains the data of many users. Service users do not have access to these replicates nor have any control over them. They rely completely on the backup policy provided form them by their service providers. This approach is like one putting all of his eggs on a single basket. In the phase of minor disasters, service providers may recover within a reasonable amount of time since they have backup sites in several places.

However, in the phase of major disasters like Ransomware attack, the possibility of service providers recovering all of their data is very slim since all of the service provider backup sites can be brought down within a fraction of second by a well-coordinated Ransomware or terrorist attack. It is in the light of this understanding that the researcher proposed a User-Centric Cyber Disaster Recovery Model where service users will have total control of their database backup stored on their premises so that when their service provider is brought down completely by any form of disaster, service users can simply swap to another service provider and continue their business without losing any of their sensitive data.

3.1.2 Data Flow Diagram of the Existing System

The data flow diagram of the existing system is as shown in Figure 3.1. There are two distinct premises, one for Service Users and another for Service Providers.

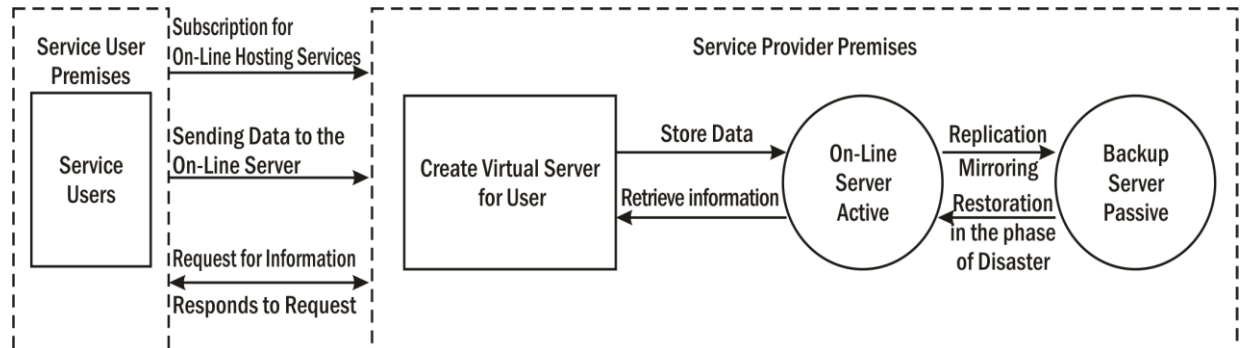


Figure 3.1: Data Flow Diagram of the Existing System

3.1.3 Advantages of the Existing System

1. It requires fewer technicalities for service users
2. It reduces data linkages as third party software are not involved in the data replications and recovery processes
3. It may have a higher level of data consistency as the entire system is always being backed up

3.1.4 Disadvantages of the Existing System

1. Service users do not have any knowledge of what is being backed up on their behalf and where the backup files are been stored. All that service users need to do is to simply subscribe to the disaster recovery policy made available to them by their service providers.
2. In an event of disaster hitting service providers, Service users will be completely helpless and will have to wait until service providers recover from the disaster, and if the disaster is so fatal that they cannot recover, then users may completely lose their data.
3. Service users may also suffer from unnecessary charges if they are charged by the size of their backup files. This is so because some entities that were backed up by the default backup frequency of the server may really not deserve such frequent backups if service users are opportune to configure their backup policies.

3.2 Analysis of the Proposed System

The proposed User Centric Cyber Disaster Recovery System will automatically generate data backup files at the preset interval of time (T). A checksum algorithm is used to compare the backup file (F_n) generated at time (T_n) with the previous backup (F_{n-1}) generated at time (T_{n-1}). If the checksum calculator shows any difference in the files, then the new backup file will be generated and stored on the system else it will be discarded. Once backup files are generated for storage at any given time, the system automatically encrypts and place it on a private secured compartment created for the users. The system maintains privately secured compartments for every registered user to enhance data segregation in a Multi-tenancy Architecture. The users log onto the system and download encrypted backup files to their local system. The primary reason for encrypting the backup file is to prevent it from Man in the Middle (MITM) attack. In the event of any disaster or service interruption, users will then retrieve their encrypted backup files, decrypt and restore their services with another service provider and their businesses will continue to run with very little or no delay. With the proposed system, service users can now have their own private organized disaster recovery plan instead of depending on service providers aimlessly and helplessly for their disaster recovery processes. All of the functionalities in the proposed system are modularized and all the modules are represented by classes. Objects of these classes shall intelligently interact with one another to logically deliver the aim of the proposed system. Where necessary, all forms of object-oriented programming relationships such as inheritance, association, composition and aggregation will be implemented.

Graphical models are used to clearly illustrate the operations of some key modules. Figure 3.2 is the model diagram for the file backup and replication module, Figure 3.3 is the model diagram of the billing module while Figure 3.4 is the model diagram for the recovery process.

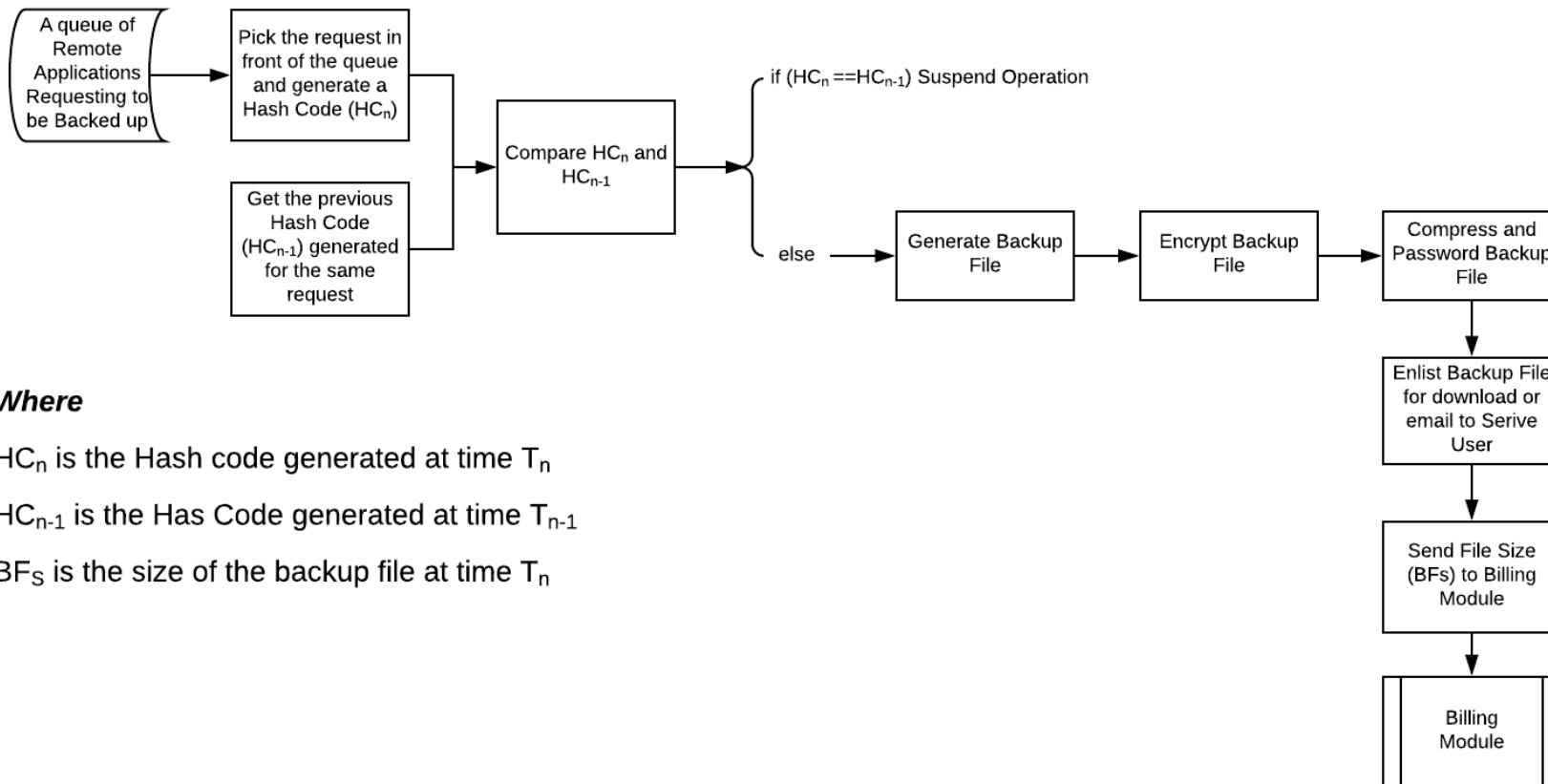
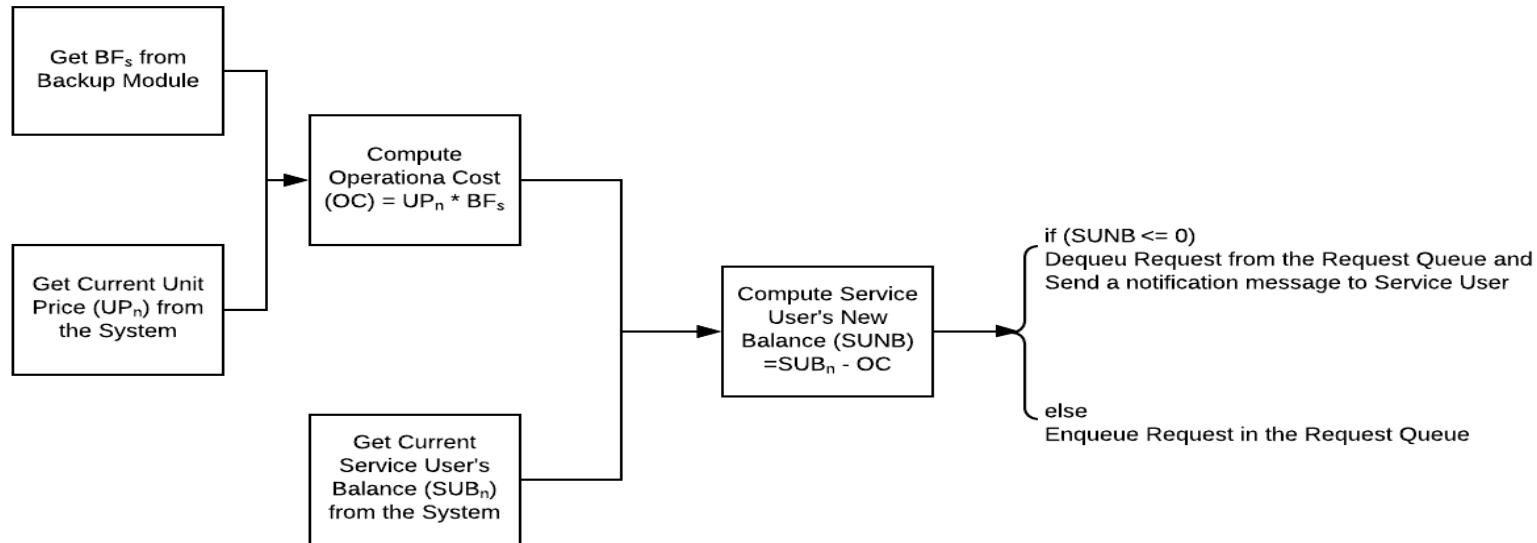


Figure 3.2: Model Diagram for File Backup



Where

BF_s is the size of the backup file at time T_n

UP_n is the backup unit price per Kilobyte of data at time T_n

OC is the Operational Cost of Backup processes

SUB_n is the Service User's Balance at time T_n

$SUNB$ is the Service User's New Balance after OC has been deducted from his/her previous Balance

Figure 3.3: Model Diagram for the Billing System (Pay as you go)

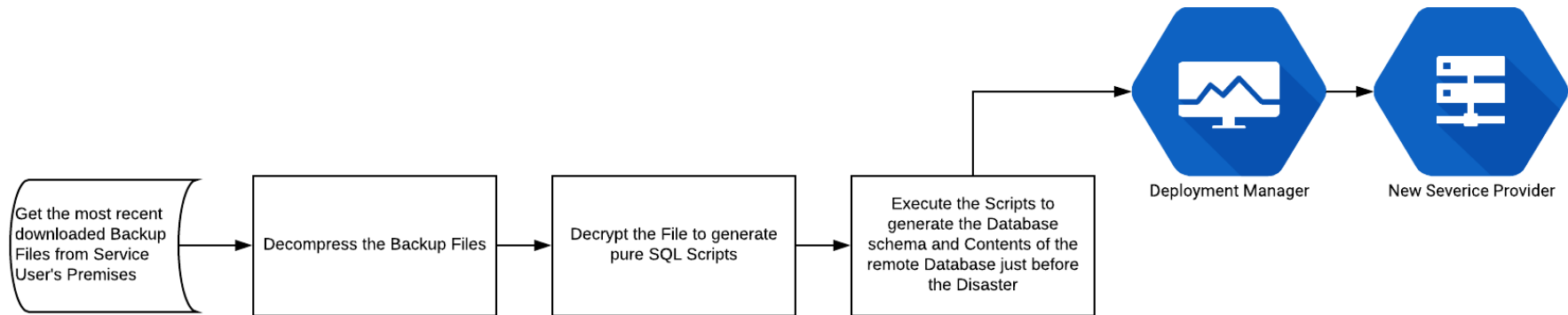


Figure 3.4: Model Diagram for the Recovery System

3.2.1 Advantages of the Proposed System

Some of the advantages of the proposed system are as listed;

1. The system will grant service users the opportunity to define and configure their private Disaster Recovery Plans
2. With the system, service users will have their backup files stored in their premises. In an event of disaster targeted to their service provider, they can easily and quickly swap to another service provider without having to wait helplessly for their service providers to restore services.
3. The system will completely free users from any disaster that may be targeted on their service providers as they will be having access to the most recent consistent state of their remote databases even when their service provider is knocked down by a disaster.

3.2.2 High-Level Model of the Proposed System

The High-Level Model of the proposed system is as shown Figure 3.5. The system is designed to enable Service Users to have access to their remote databases and backup files. The users will be completely responsible for making critical decisions on their disaster recovery policies. It also grants users the absolute liberty to swap from one service provider to another without the fear of losing very vital parts of the data or thinking of replicates compatibility between different service providers. The system will be generating the backup file at preset intervals of time of some selected entities of the entire database schema and store same to the location chosen by service users. Backup files will be highly secured by strong encryption mechanism to prevent it from Man in the Middle (MITM) attack. At the point of usage, the backup file will be decrypted by the corresponding decryption mechanism.

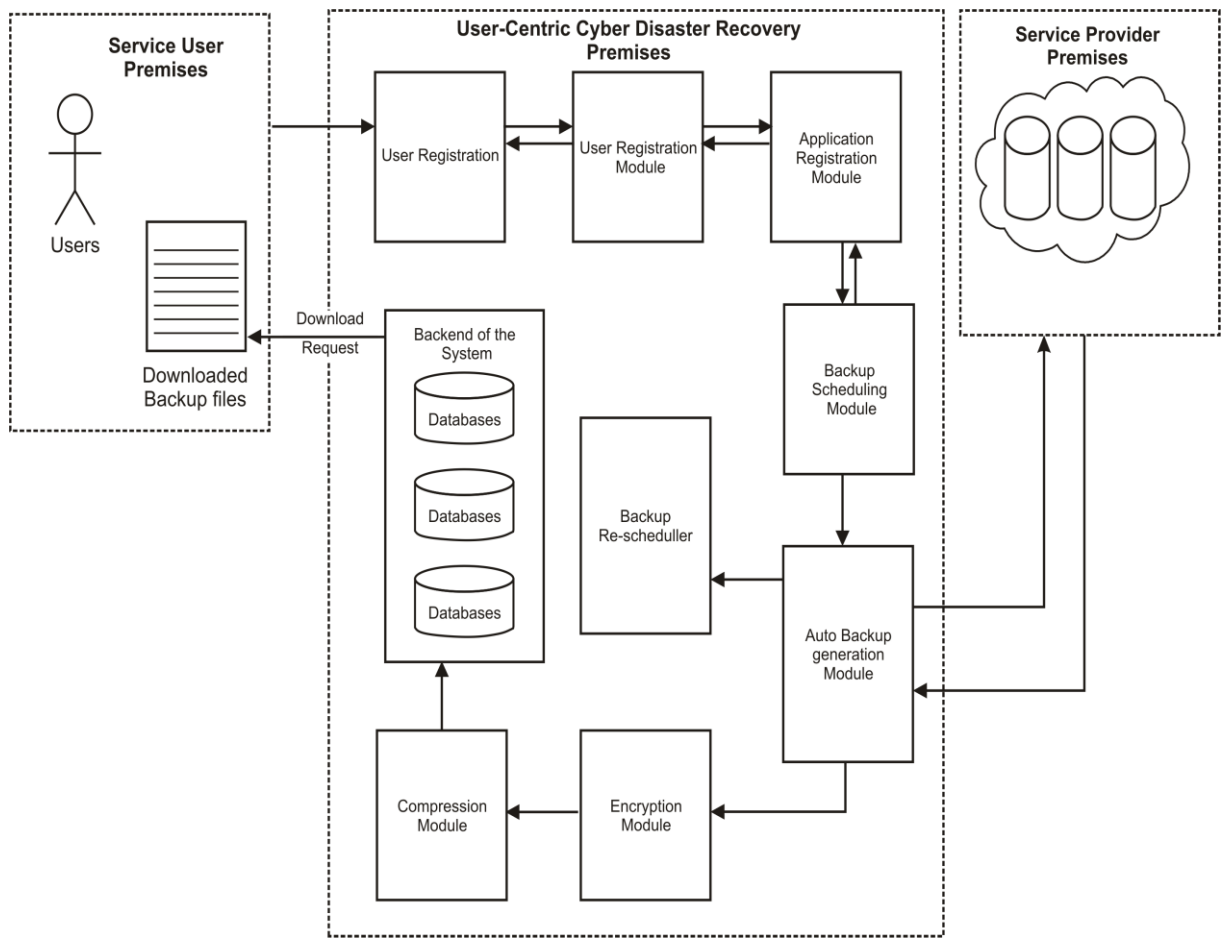


Figure 3.5: High-Level Model of the Proposed System

3.3 Methodology Adopted

A combination of the iterative method of Dynamic System Development Methodology (DSDM) and Object-Oriented Analysis and Design Methodology (OOADM) is used in this work. DSDM assumes that all previous steps may be revisited as part of its iterative approach. Therefore, the current step need be completed only enough to move to the next step since it can be finished in a later iteration. The premise is that the business requirements may probably change as understanding increases such that any further work would probably be a waste. According to this approach, the time is taken as a constraint i.e. the time is fixed; resources are fixed while the requirements are allowed to change. This does not follow the fundamental assumption of making a perfect system the first time but provides a usable and useful 80% of the desired system in 20% of the total development time. This approach has proved to be very useful under time constraints and varying requirements. DSDM processes are as presented in Figure 3.6.

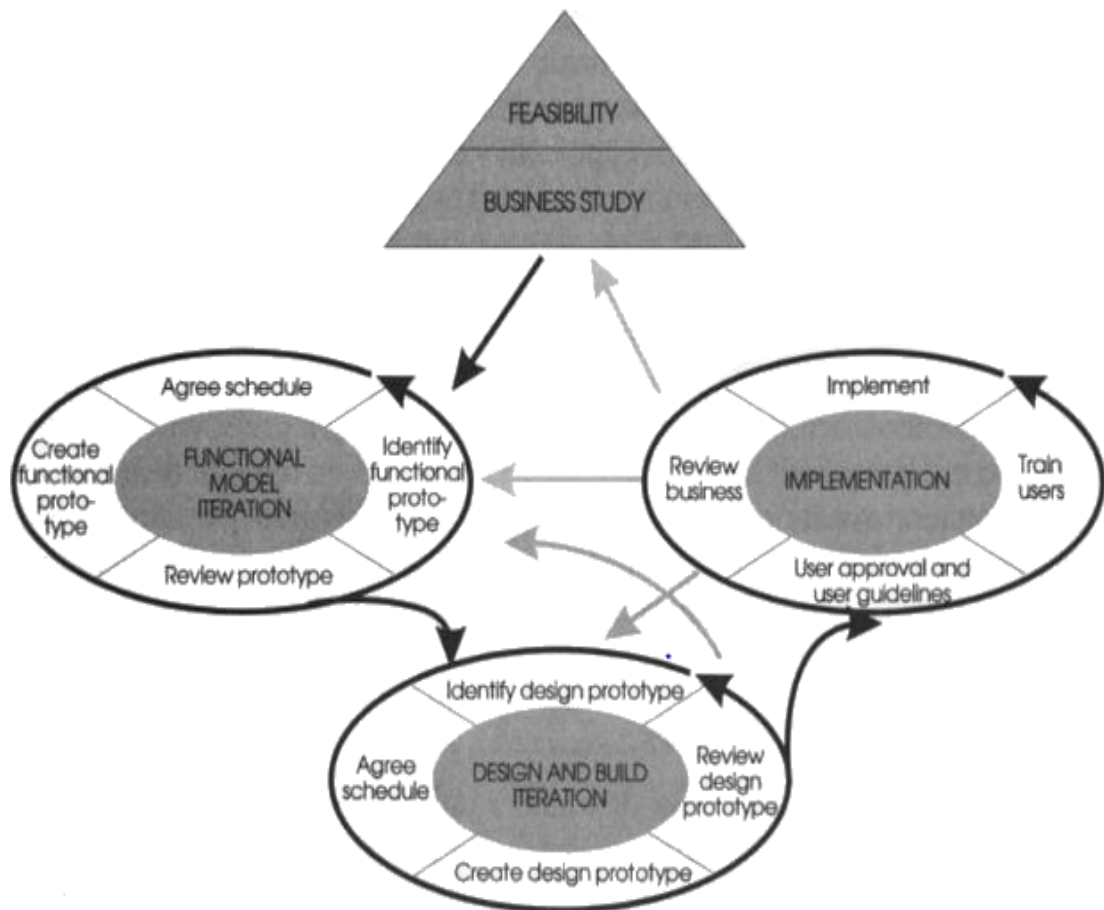


Figure 3.6: DSDM Process Diagram (Stapleton, 1997)

On the other hand, the OOADM methodology is used to identify the objects needed in the system and their interrelationships. Adequate and relevant UML diagrams such as class diagram, use case diagram, activity diagram and deployment diagrams were generated which makes the coding process quite easy and straightforward. Most qualities of object-oriented programming such as polymorphism, inheritance, encapsulation and code reusability were employed in the development of the new user-centric disaster recovery system which was able to identify, authenticate and assign functionalities to different categories of users based on their login detail.

CHAPTER FOUR

SYSTEM DESIGN AND IMPLEMENTATION

4.1 Objectives of the Design

The design objectives of the new system are to

- i. grant service users the privilege to independently configure backup policy for their remote databases hosted with various service providers.
- ii. automatically generate backup files in accordance with the configuration patterns of service users.
- iii. ensure that backup files are encrypted to protect from Man in the Middle attack while on transit from service providers' premises to service users' premises.
- iv. ensure that the encrypted backup files are compressed to enhance its transmission over the internet.
- v. ensure that the compressed files are password protected to further enhance the security nature of the system.
- vi. ensure that the backup files can be decompressed, decrypted at recovery time only with the right keys

4.2 Main Menu/Control Centre

The functionalities of the system are logically grouped together in menu and submenu. Figure 4.1 shows the structural arrangement of the main menu.

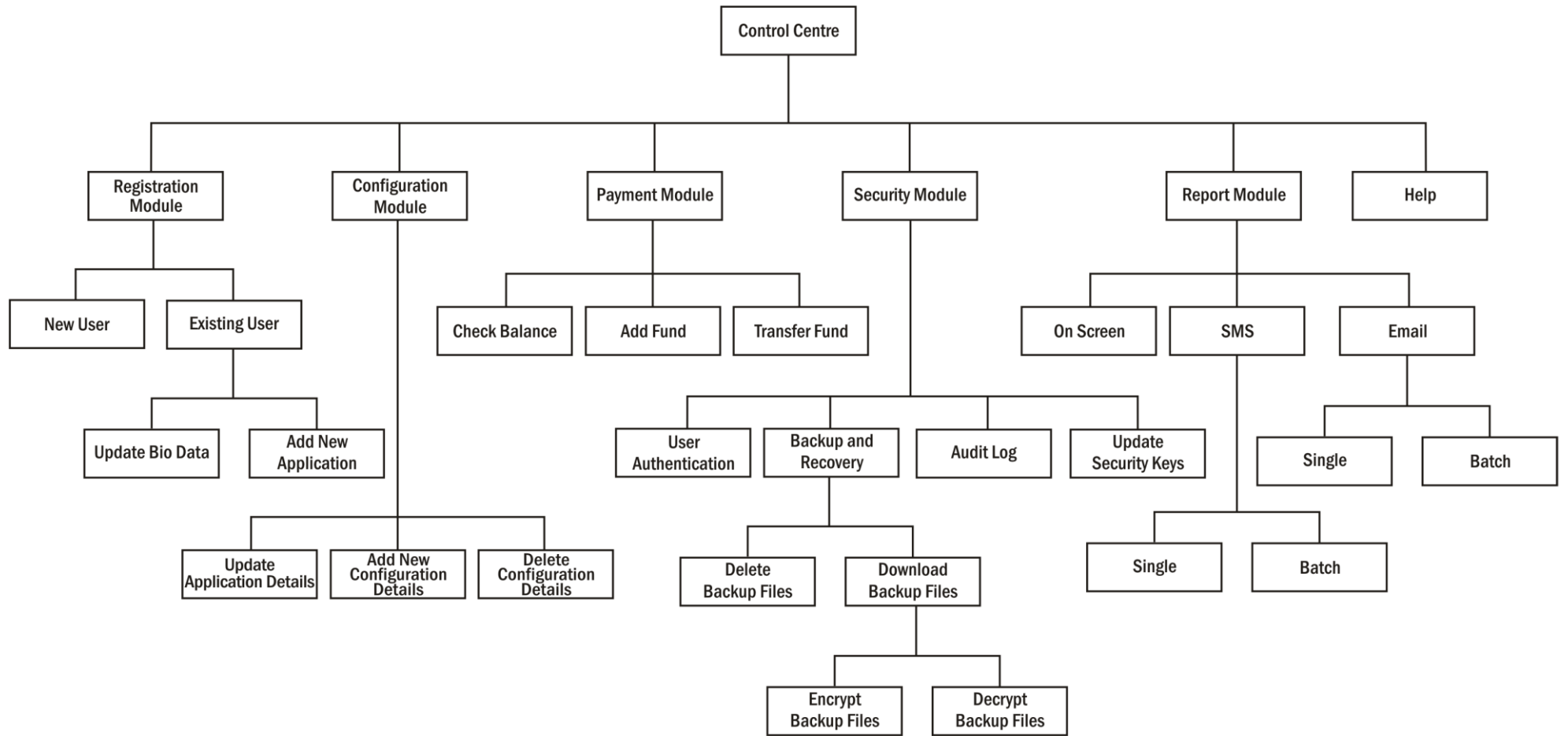


Figure 4.1: Main Menu

4.3 Submenu/Subsystem

The menu arrangement is further broken down to submenu and subsystem in accordance to the complexity of the functionalities under these menus. This section discusses in details some of these sub systems.

4.3.1 Registration Subsystem

This subsystem is used to register new users into the system. At this registration point, new users carefully fill and submit the registration form. This form implements CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) to test if this form is being filled by human or by a robot. This mechanism is used to checkmate the Denial of Service (DoS) attack. Once the registration is completed, an account will be created for such users. Decompression and decryption keys will be sent to them via Short Message Service (SMS). Figure 4.2 is a structural diagram of the Registration Subsystem.

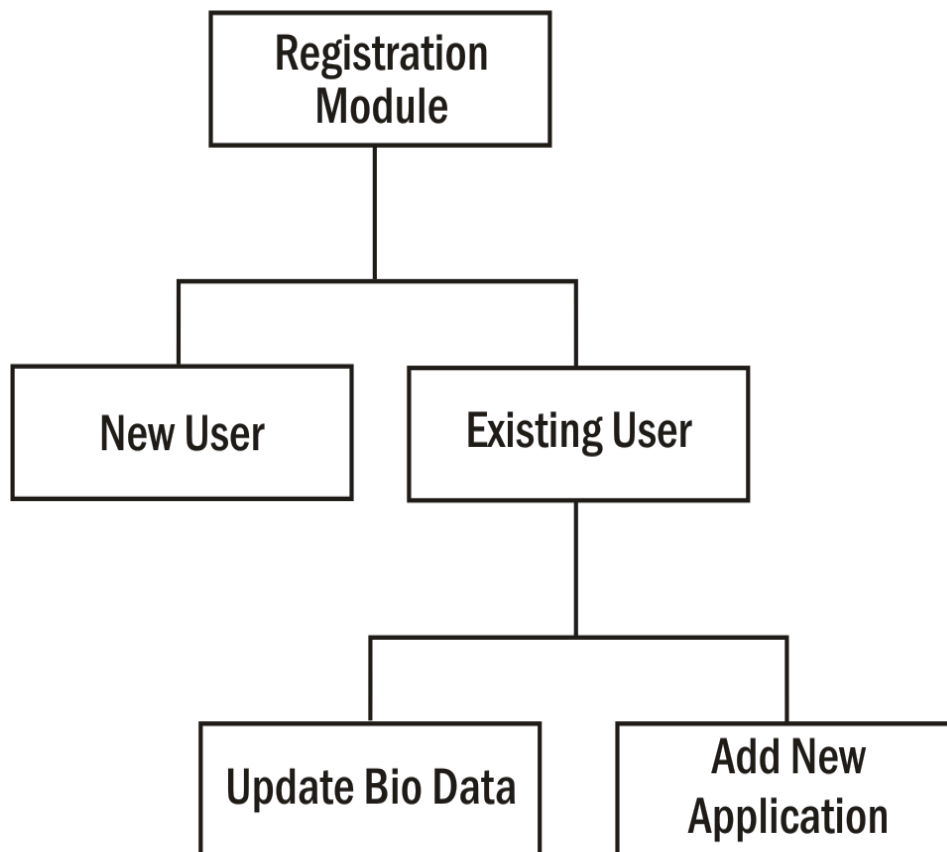


Figure 4.2: Registration Subsystem

4.3.2 Configuration Subsystem

This subsystem is used by registered users to configure their Independent Business Continuity Plan (IBCP) on the new system. With this system, users supply the details of their remote databases connection strings. These details are then captured and saved on the new system. The system then used this detail as a handle to connect to login into service providers' premises and get the entire schema of the remote database and present same for users to comfortably define their backup policies by selecting the entities from the database schema to be backed up and for every selected entity their corresponding backup frequency is also selected. Submenus that are under this subsystem are New Configuration, Update Configuration and Delete Configuration. Figure 4.3 is a structural diagram of the Configuration Subsystem.

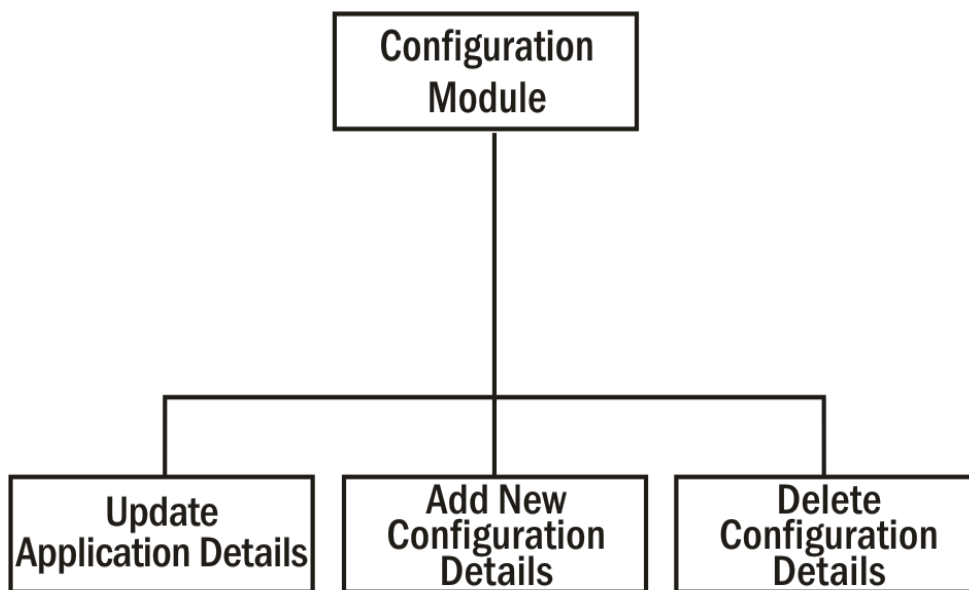


Figure 4.3: Configuration Subsystem

4.3.3 Payment Sub System

This subsystem is used to manage users' wallet on the new system. The new system uses a pay-as-you-go billing model. Submenus that are under this

subsystem are Add Fund, Transfer Fund and Check Balance. Figure 4.4 is a structural diagram of the Payment Subsystem.

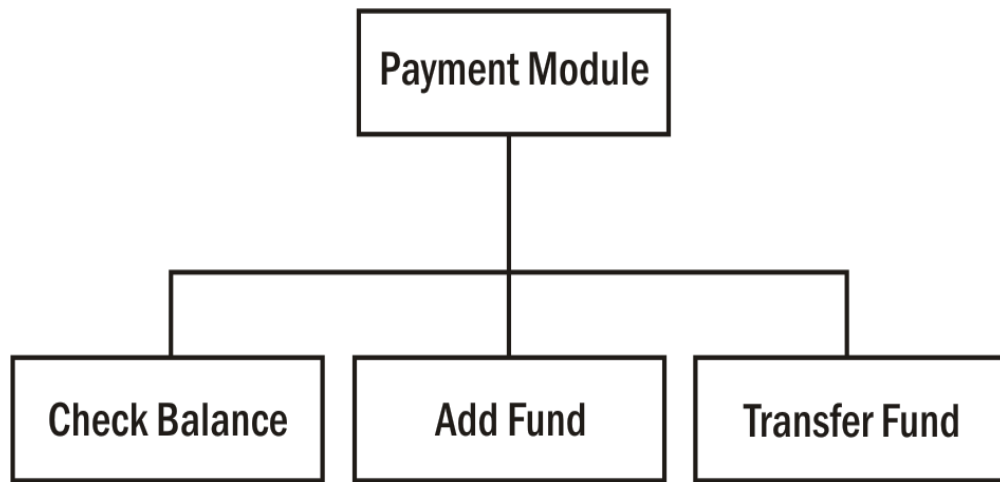


Figure 4.4: Payment Subsystem

4.3.4 Security Subsystem

This subsystem is used to manage all the security mechanism of the new system ranging from Password management, Users' Authentication, Decryption and Decompression mechanisms. Submenus that are under this subsystem are User Authentication, Backup and Recovery, Audit Log, Update Security Keys, Download Backup Files, Decompress Backup Files and Decrypt Backup Files. Figure 4.5 is a structural diagram of the Security Subsystem.

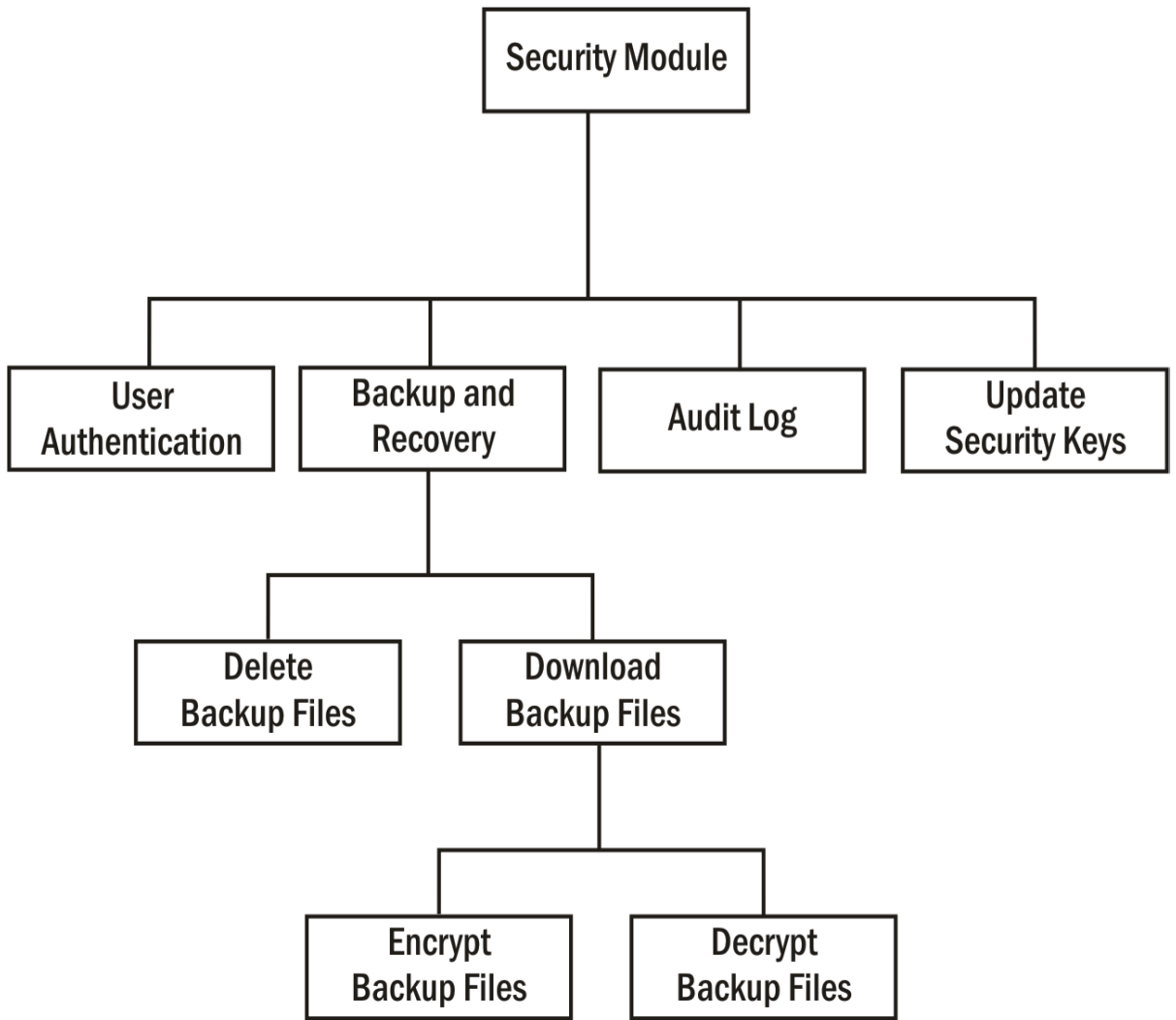


Figure 4.5: Security Subsystem

4.3.5 Report Subsystem

This subsystem is used to manage all the report tools used in the new system. Submenus that are under this subsystem are On Screen, SMS and Emails. Figure 4.6 is a structural diagram of the Report Subsystem.

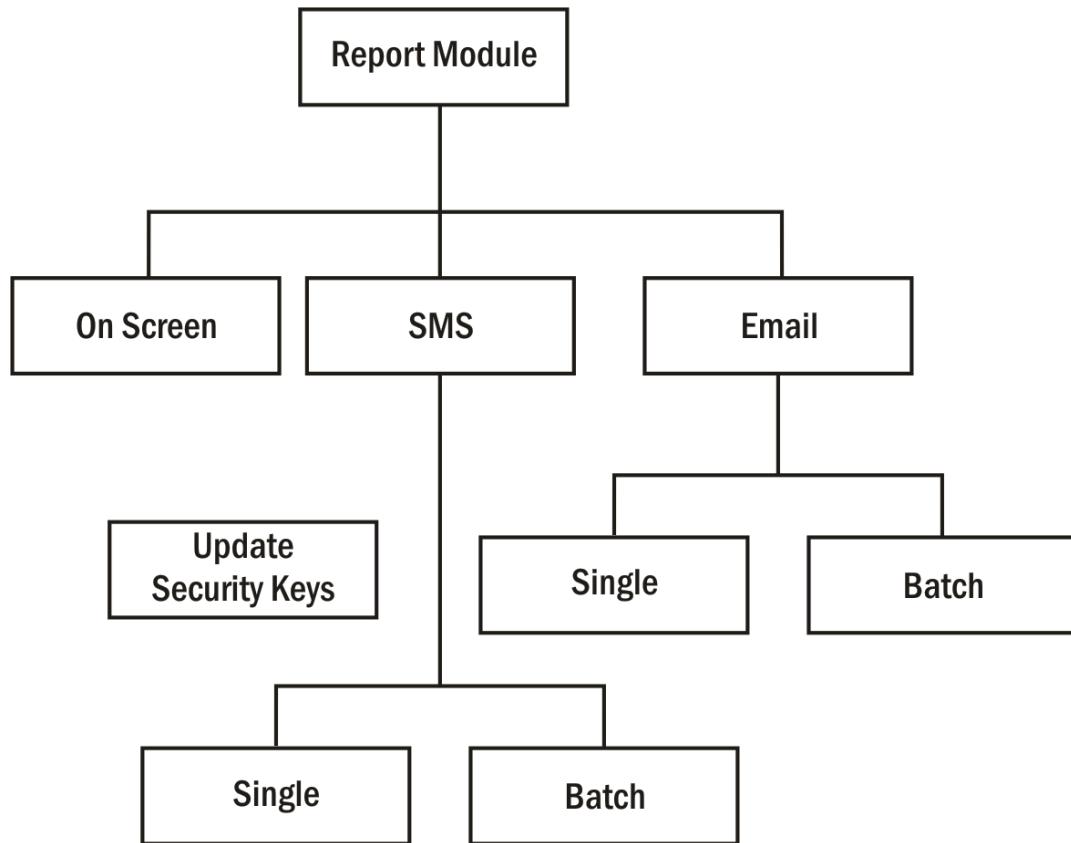


Figure 4.6: Report Sub System

4.4 Specifications

This section outlines the general Software Requirement Specifications (SRS) adopted for this work. Database tools, database schema, Program Module Specifications, Input/Out Specifications and Data Dictionary (DD) shall all be discussed in this section.

4.4.1 Database Development Tool

MySQL was used as the database development tool because it is an open source multithreaded, multi-user Database Management System (DBMS). It is suitable for the research due to the dynamism of the cyberspace. MySQL, in addition, supports standard Structured Query Language (SQL). With the MySQL server, several users can work

concurrently, in a secure environment and it offers fast data access to only authorized users.

4.4.2 Database Design and Structure

The database design is made up of tables. some of the tables and their attributes are as shown in Tables 4.1 to 4.5;

Table 4.1: User Table

SN	Field Name	Data Type	Length	No of Decimal Places	Key
1	UserId	Varchar	100		Primary Key
2	FirstName	Varchar	100		
3	MiddleName	Varchar	100		
4	LastName	Varchar	100		
5	Sex	Varchar	45		
6	Email	Varchar	200		
7	PhoneNo	Varchar	45		
8	DateofBirth	Varchar	45		
9	DateofRegistration	Varchar	45		
10	CreditBalance	Double	10	2	
11	UserPassportId	Varchar	100		Foreign Key

Table 4.2: Application Table

SN	Field Name	Data Type	Length	Key
1	ApplicationId	Varchar	100	Primary Key
2	UserId	Varchar	100	Foreign Key
3	URL	Varchar	200	
4	LastName	Varchar	100	
5	DatabaseName	Varchar	100	
6	DBMSType	Varchar	45	
7	DatabaseUserName	Varchar	100	
8	DatabasePassword	Varchar	100	
9	DatabasePortNumber	Varchar	45	

Table 4.3: Backup Configuration Table

SN	Field Name	Data Type	Length	Key
1	TransactionId	Varchar	100	Primary Key
2	ApplicationId	Varchar	100	Foreign Key
3	EntityName	Varchar	100	
4	DateofConfiguration	Varchar	100	
5	DateofNextBackup	Varchar	45	

Table 4.4: Backup History Table

SN	Field Name	Data Type	Length	No of Decimal Places	Key
1	TransactionId	Varchar	100		Primary Key
2	ApplicationId	Varchar	100		Foreign Key
3	VolumeofBackupFile	Double	10	2	
4	DateofBackup	Varchar	45		
5	StorageLinks	Varchar	100		

Table 4.5: Subscription History Table

SN	Field Name	Data Type	Length	No of Decimal Places	Key
1	TransactionId	Varchar	100		Primary Key
2	UserId	Varchar	100		Foreign Key
3	Amount	Double	10	2	
4	DateofSubscription	Varchar	45		

4.4.3 Program Module Specification

The entire application is broken into six major modules. Each module is responsible for a particular task. The modules and their functionalities are as articulated and shown hereafter;

i. Registration Module

This module is used to create an account for every user of this application. Before anyone can make tangible use of this application, he/she must first and foremost, register with the application. At registration, some vital details such as login (details, encryption and decryption key) about the user will be captured and stored in the database. After a successful registration, users will then be given the privilege to

register their applications which they intend to be monitored and backup by the new system at regular intervals of time in accordance with their configuration details.

ii. Configuration Module

This module is used for the configuration of applications to be monitored. The module gives users the privilege to select the entities in their remote database to be monitored. The module also gives users the opportunity to set up the frequency of backup process they desire to be carried out per entity.

iii. Payment Module

This module is used for an online wallet for the users. Users will be given various modes of crediting their wallets and they shall make their payments using any of the modes most suitable to them. It also gives users the ability to transfer fund to one another on the platform.

iv. Backup and Scheduling Module

This module is strictly managed by the system. It runs at the background like a daemon thread without any human intervention. It constantly refreshes the entire system and checks for applications and entities that are due for backup. It backs up such entities and set the next backup date and time for them in accordance with their configuration parameters. The module is also responsible for encrypting the backup files using strong encryption mechanism. This is very important to prevent the backup files from Man in the Middle (MITM) attack.

v. File Compression Module

This module uses a hybrid of Huffman Coding and Shannon-Fan Coding compression algorithms to compress backup files. This is to reduce the size of backup files so that they can be easily encrypted and also increase the performance of the system. With this hybrid compression algorithm, big files in the range of 15 to 20 MB can be compressed in milliseconds and no information would be lost.

vi. Download Module

This module provides for users to download their backup files. For every successful backup and file compression operations carried out by the system, an encrypted backup file will be generated for users. Upon logging into the system, users have the privilege to download these files into their local systems. With these files in their possession, users can extract, decrypt and execute the scripts to generate the most consistent state of their databases. With this, service users will no longer lose their vital data in the event of any form of disaster hitting their service provider.

vii. Decryption Package

The new system will also make a decryption Package available for service users to download and install on their local system. This package is what service users will use to decrypt the encrypted backup file they download from the system. At the point of decrypting, this package will require users to supply their decryption key which is another level of security. This decryption key will be very difficult for Man in the Middle (MITM) to hijack at this point because this whole process of decryption will take place at the users' local system.

The general architectural diagram of the new system is as presented in Figure 4.7.

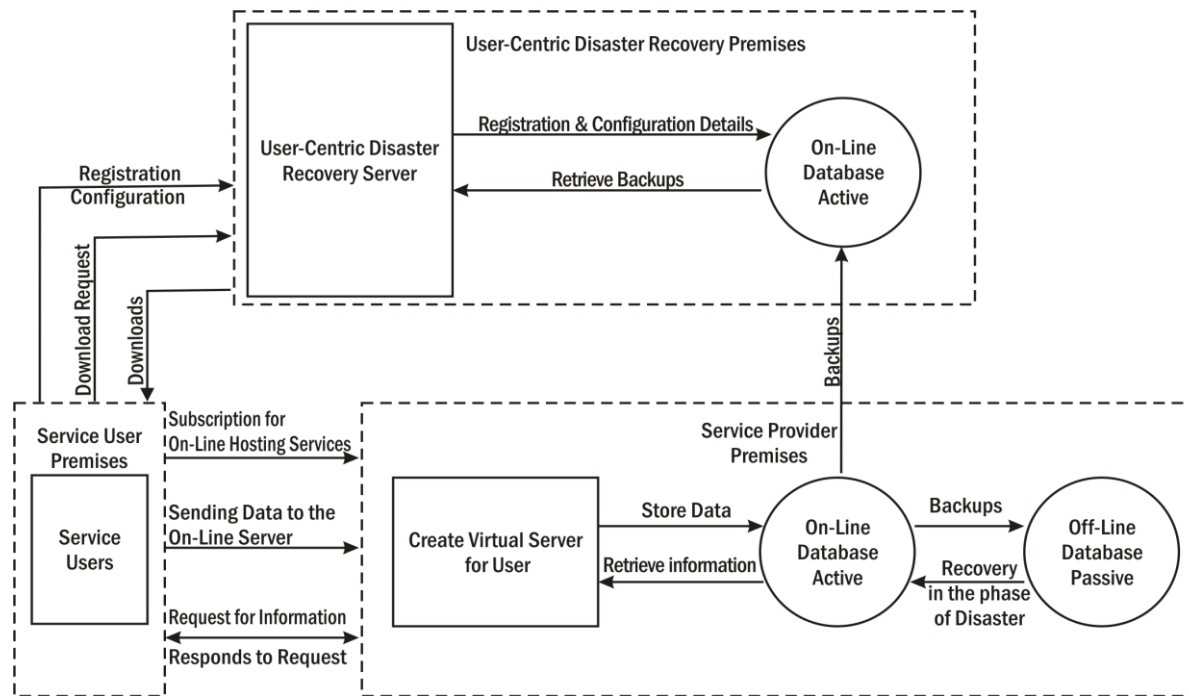


Figure 4.7: Architectural Diagram of the New System

4.4.4 Input/Output Format

The system does not depend on sophisticated input devices like the card reader, light pen or voice actuated input devices. Simple devices like keyboard, mouse, scanner and digital camera for image capturing are just enough to operate the system to its full capacity. Data will be captured into the system via the use of web-enabled forms. Typical examples of these data capturing forms are as shown in Figures 4.8 to 4.11;

User Registration Form

First Name

Middle Name

Last Name

email

Phone Number

Sex Male Female

Password

Confirm Password

Date of Birth

Passport

Figure 4.8: User Registration Form

App Registration Interface

Url

Database Name

Database Password

DBMS

Port Number

Submit

Figure 4.9: App Registration Form

Subscription Form Interface

Pin

Card No

User ID

Submit

Figure 4.10: Subscription Form

Backup Configuration Interface

App ID

Entity	Backup	Frequency
Ent1	<input type="text"/>	<input type="text" value="▼"/>
Ent2	<input type="text"/>	<input type="text" value="▼"/>
Ent3	<input type="text"/>	<input type="text" value="▼"/>
•	•	•
•	•	•
•	•	•
EntN	<input type="text"/>	<input type="text" value="▼"/>

Figure 4.11: Backup Configuration Form

4.4.5 Algorithm

An algorithm is an ordered set of unambiguous, executable steps that defines a terminating process. Algorithms that are meant to be invoked by other algorithms are called sub-algorithms while algorithms that invoked themselves are referred to as recursive-algorithms (Usman, Ge, Karim, and Agber, 2014). Most of these concepts shall be fully employed in the development of the functional algorithm for this research work. In this work, the entire algorithm is divided into sub-algorithms. The sub-algorithms are integrated together with the main algorithm which invokes the sub-algorithms one after the other until the entire work is completely handled.

i. Main algorithm

Pre: Users interactively identify themselves as either new or existing users, and also indicate the category of transactions they are interested in carrying out.

Post: Users are directed by the system to the right sub-algorithm(s)

1. Start
2. If (new user) then invoke (User registration sub-algorithm)
3. Input login details
4. If (login details are correct) then go to step 5.1 else go to step 3
- 5.1 To register new applications invoke (App registration sub-algorithm)
- 5.2 To configure registered apps invoke (App configuration sub-algorithm)
- 5.3 To Credit your account invoke (Subscription sub-algorithm)
- 5.4 To download backup files invoke (Download sub-algorithm)
6. Exit

ii. User registration sub-algorithm

Pre: New users are directed to this sub-algorithm by the main algorithm

Post: This sub-algorithm enables new users to register into the system and a unique user identification number (UIN) will be generated for the user

1. Start
2. Input bio-data
3. Input phone number and email address
4. Input password
5. Confirm password
6. If (password is confirmed) then go to step 7 else go to step 4
7. The system sends a verification code to the phone number and email address
8. If(Verification code is confirmed) go to step 9 else go to step 12
9. Generate an encryption and decryption key using Deffie Hellman algorithm
10. Sent his/her login details and to his/her phone number
11. Create the user by persisting his/her records in the Database
12. Return

iii. Apps registration sub-algorithm

Pre: App registration requests are directed to this sub-algorithm by the main algorithm

Post: This sub-algorithm registers new applications and generates a unique app identification number (AIN) for the application

1. Start
2. Input application's *Uniform Resource Locator (URL)*
3. *Input* application's *DBMS type*
4. *Input* application's *Database name*
5. *Input* application's *Database password*
6. *Input* application's *Database port number*
7. Whitelist our IP addresses
8. Generate and send the AIN to the user's phone number and email address
9. Create a private compartment for the application using the AIN
10. Create the Apps by persisting the record in the Database
11. Return

iv. App configuration sub-algorithm

Pre: App configuration requests are directed to this sub-algorithm by the main algorithm

Post: This sub-algorithm configure applications

1. Start
2. Choose the Application to be configured from the list of all your registered Apps
3. Display all the entities contained in the Database of the application
4. Select the entities that are required to be backed up
5. For all the selected entities, set the frequency of the backup operation
6. Persist the settings on the Database
7. *Return*

v. Subscription sub-algorithm

Pre: Subscription requests are directed to this sub-algorithm by the main algorithm

Post: This sub-algorithm credits the account balance of the user

1. Start
2. Choose payment mode (pm)
3. If (pm = web pay) invoke (web pay sub-algorithm)
4. If (pm = scratch card) then go to step 5
5. Input PIN
6. If (PIN is Valid) then go to step 7 else go to step 5
7. Determine scratch card value (card_value)
8. Determine the user's old balance (old_balance)
9. Determine users new balance (new_balance = old_balance+ card_value)
10. Set user's account balance to new balance
11. Set PIN to Invalid
12. Return

vi. Web Pay sub-algorithm

Pre: Web pay requests are directed to this sub-algorithm by subscription sub-algorithm

Post: This sub-algorithm credits the account balance of the user

1. Start
2. Input web pay details or ATM card details
3. Input amount of top-up (top_up_value)
4. If (Details are Valid) go to step 5 else go to step 2
5. Determine users old balance (old_balance)
6. Determine users new balance (new_balance = old_balance+ top_up_value)
7. Set user's account balance to new balance
8. Debit ATM card with top_up_value
9. Return

vii. Auto backup algorithm

Pre: Auto backup is a recursive algorithm that constantly runs at the background of the system as a daemon thread.

Post: This sub-algorithm queues up all the applications that are due for backup at the time of system refreshment and backed up one after the other

1. Start
2. Retrieve service charge per Megabyte (rate)
3. Pick up all apps that are due for back up and queue them up for processing
4. Pick the app in front of the queue
5. Check user's credit balance (cr_balance)
6. If (cr_balance \geq 0.00) then go to step 7 else go to step 14
7. Create a backup file for the app
8. Determine the size of the backup file (size) in Megabyte
9. Determine backup fee (backup_fee = rate * size)
10. Determine user's new balance (new_balance = cr_balance - backup_fee)
11. If (new_balance \leq 0.00) then send top up reminder message to user
12. Encrypt and store the backup file in the application private compartment
13. If(queue \neq empty) go to step 4 else Invoke (auto backup algorithm)
14. Exit

viii. Download sub-algorithm

Pre: Download requests are directed to this sub-algorithm by the main algorithm

Post: This sub-algorithm downloads backup files to the user's local system

1. Start
2. Chose the Application to backup file from the list of all your registered Apps
3. Display the list of backup files in descending order of date of the backup
4. Select the file to be downloaded (Recommended most recent)
5. Download the file to your local system
6. Return

4.4.6 Data Dictionary

A Data Dictionary, also called a Data Definition Matrix, provides detailed information about the data elements, such as standard definitions of data elements. In this work, data elements are identifiers which are user-defined names given to program entities such as variables, constants, classes and methods. Some of these data elements and their corresponding definition are as presented in Table 4.6

Table 4.6: List of Identifiers and their Meaning

SN	Variable	Meaning
1	Application	The name was given to the class that represents Application table in the entity beans of the project
2	Backupconfiguration	The name was given to the class that represents the Backupconfiguration table in the entity beans of the project
3	Backuphistory	The name was given to the class that represents the Backuphistory table in the entity beans of the project
4	Subscriptionhistory	The name was given to the class that represents the Subscriptionhistory table in the entity beans of the project
5	Userpassport	The name was given to the class that represents the Userpassport table in the entity beans of the project
6	Usersdata	The name was given to the class that represents the Usersdata table in the entity beans of the project
7	MainSession	The name was given to the Session Bean class that integrate every other class in the system
8	MainSessionLocal	The name was given to the interface where every abstract method are declared
9	getApplication (String id)	This is the name given to the method that retrieves a single application record from the database. The return type of this method is an entity set of Application i.e. Application.java

10	getBackupconfiguration (String id)	This is the name given to the method that retrieves a single backupconfiguration record from the database. The return type of this method is an entity set of Backupconfiguration i.e. Backupconfiguration.java
11	getBackuphistory(String id)	This is the name given to the method that retrieves a single backuphistory record from the database. The return type of this method is an entity set of Backuphistory i.e. Backuphistory.java
12	getSubscriptionhistory (String id)	This is the name given to the method that retrieves a single subscriptionhistory record from the database. The return type of this method is an entity set of Subscriptionhistory i.e. Subscriptionhistory.java
13	getUsersdata(String id)	This is the name given to the method that retrieves a single user record from the database. The return type of this method is an entity set of Usersdata i.e. Usersdata.java
14	getDatabaseTables(String dbtype, String domain, String databasename, String portno, String username, String password)	This is the name given to the method that checks clients' databases and returns the list of entities in the database. The return type of this method is a list of String List<String>
15	newUsersdata (Usersdata param)	This is a void method that creates a new user in the system
16	getAllBackupconfigurati on()	This is the name given to the method that retrieves all backupconfiguration record from the database. The return type of this method is a list of String List<String>

4.5 Object Diagrams

This section discusses all the design diagram to be used in the development of the new system. Details of these diagrams are as presented hereafter;

i. Entity Relationship (ER) Diagram

The ER diagram for the system is as presented below. It is created with MySQL workbench from the database schema presented in section 3.9.2. The ER diagram of the new system is as presented in Figure 4.12.

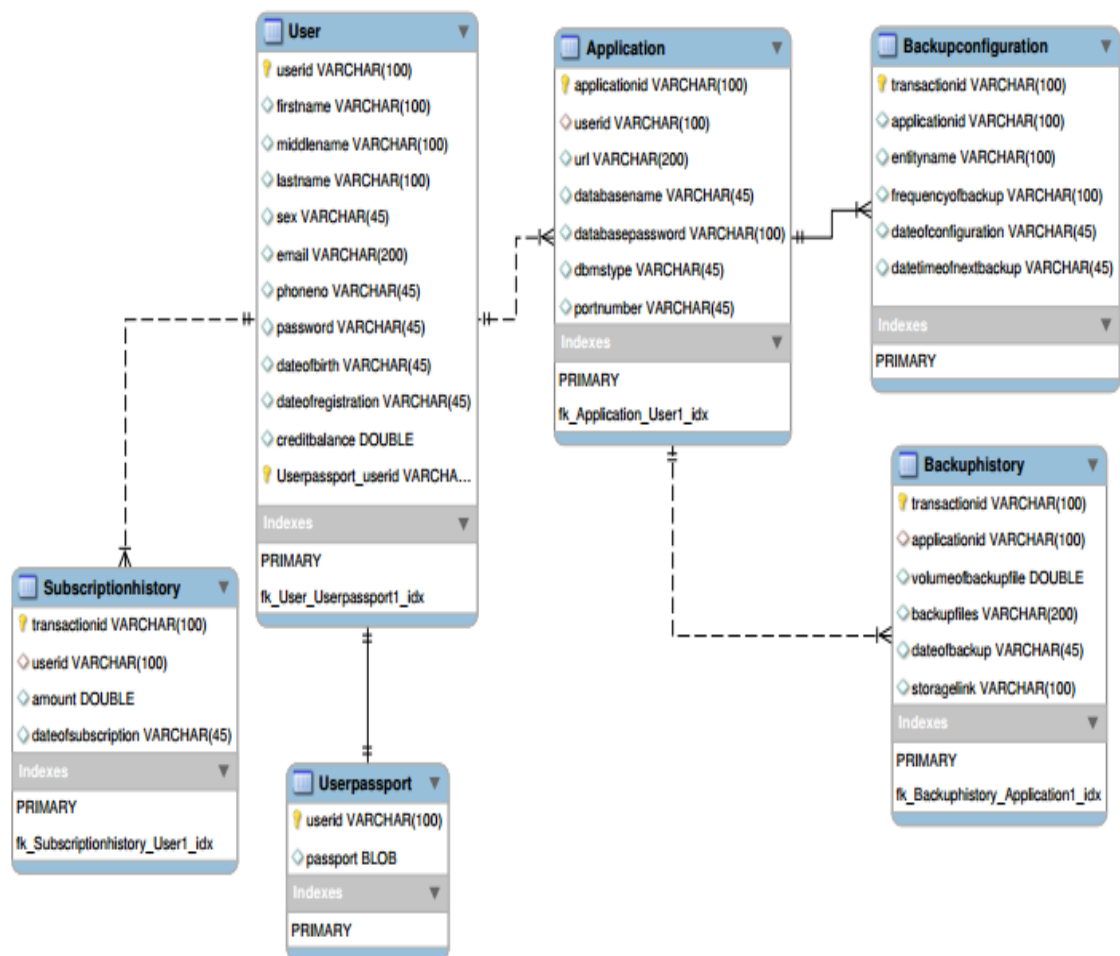


Figure 4.12: E-R Diagram of the System

ii. Use Case Diagram

Use case diagram is a behavioural UML diagram type and is frequently used to analyze various systems. Use case diagram helps to visualize the different types of roles in a system and how those roles interact with the system. The use case diagram of the new system is designed with Poseidon for UML version 8.0 and is as shown in Figure 4.13.

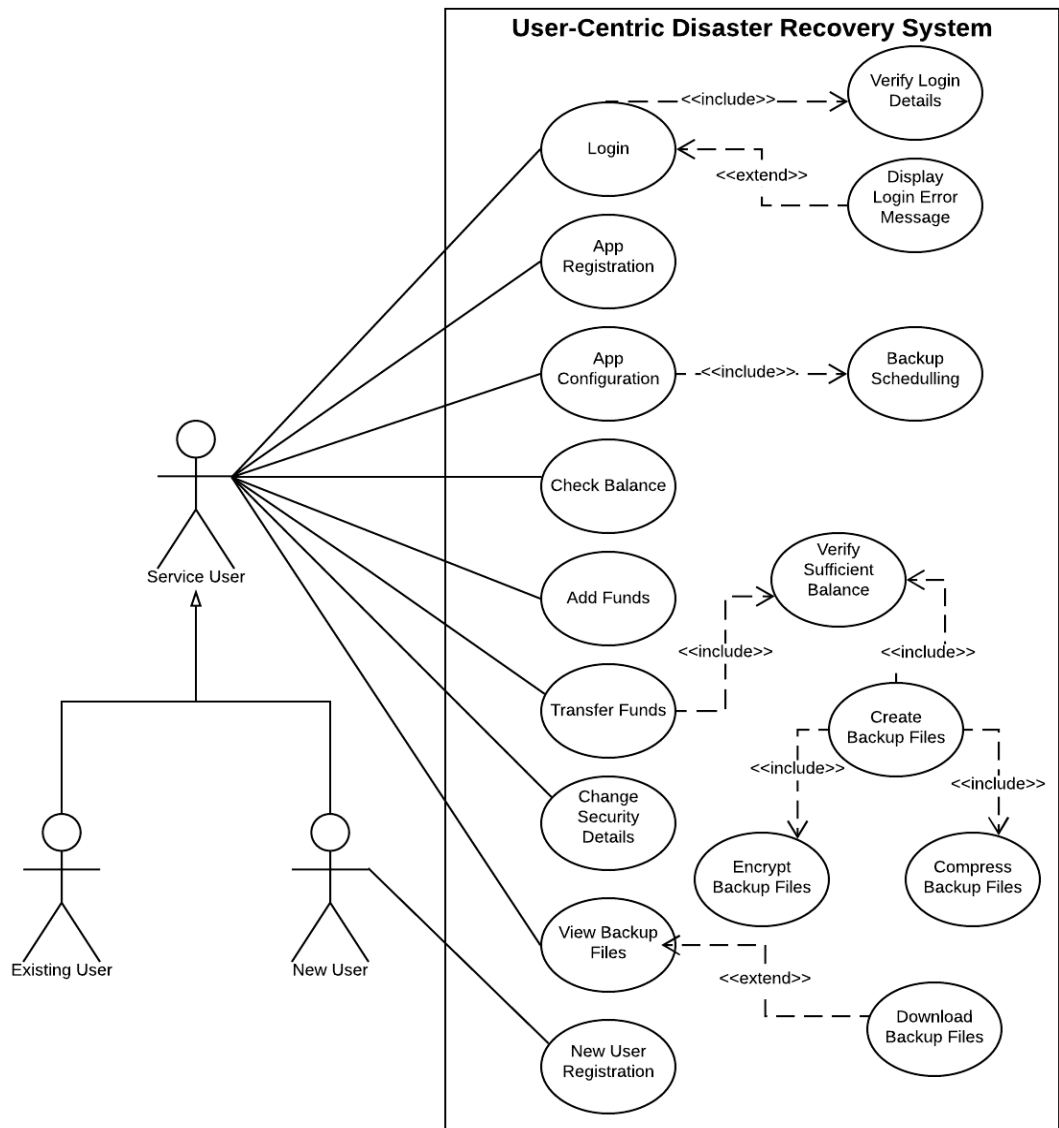


Figure 4.13: Use Case Diagram of the System

iii. *Class Diagram*

The class diagram of the new system is as presented in Figure 4.14.

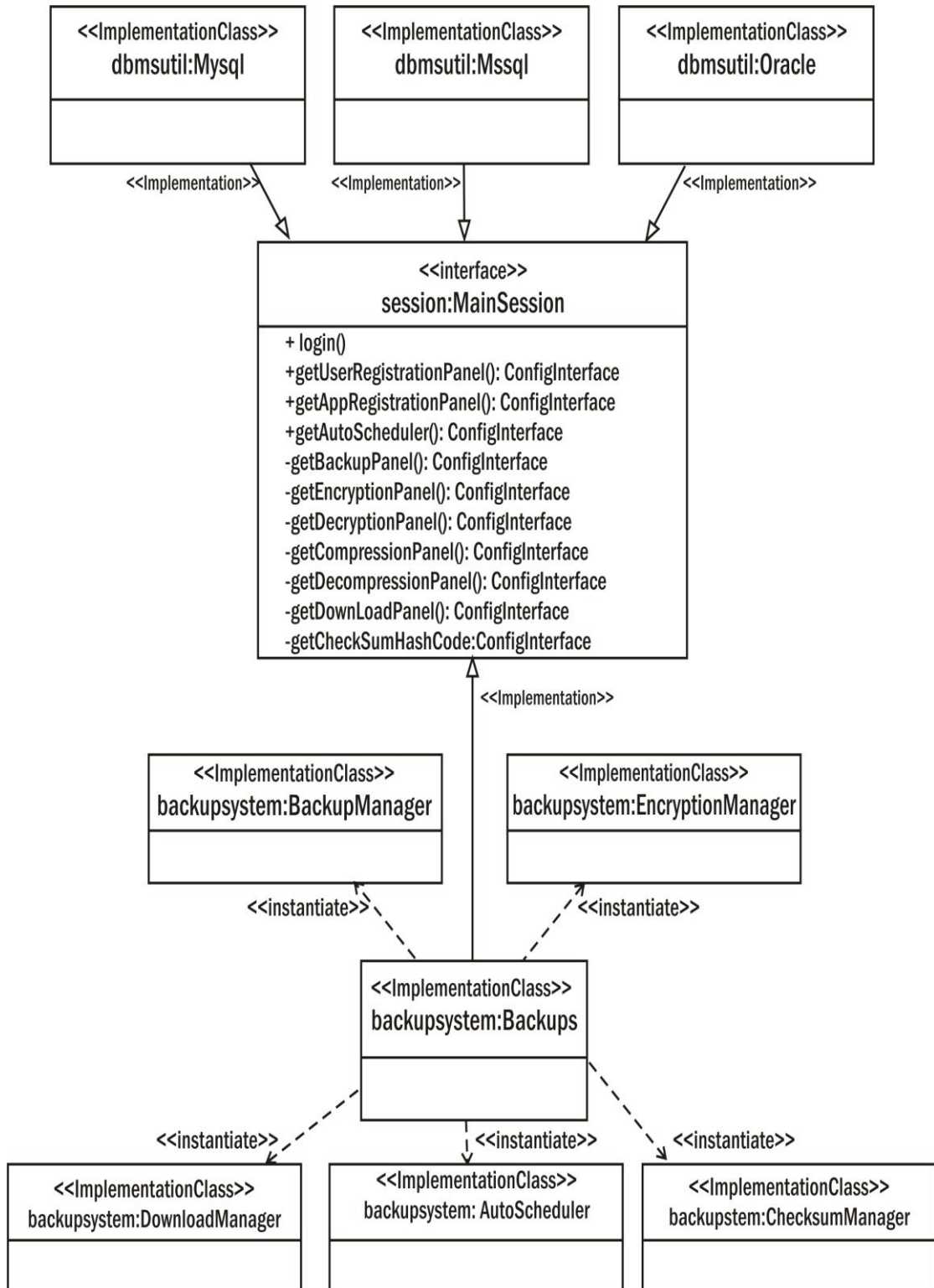


Figure 4.14: Class Diagram of the System

iv. Deployment Diagram

Deployment diagrams are used to document the distribution of Web application components. The main elements in UML deployment diagrams are nodes which are rendered graphically as cubes. The deployment diagram of the download subsystem of the new system is as presented in Figure 4.15.

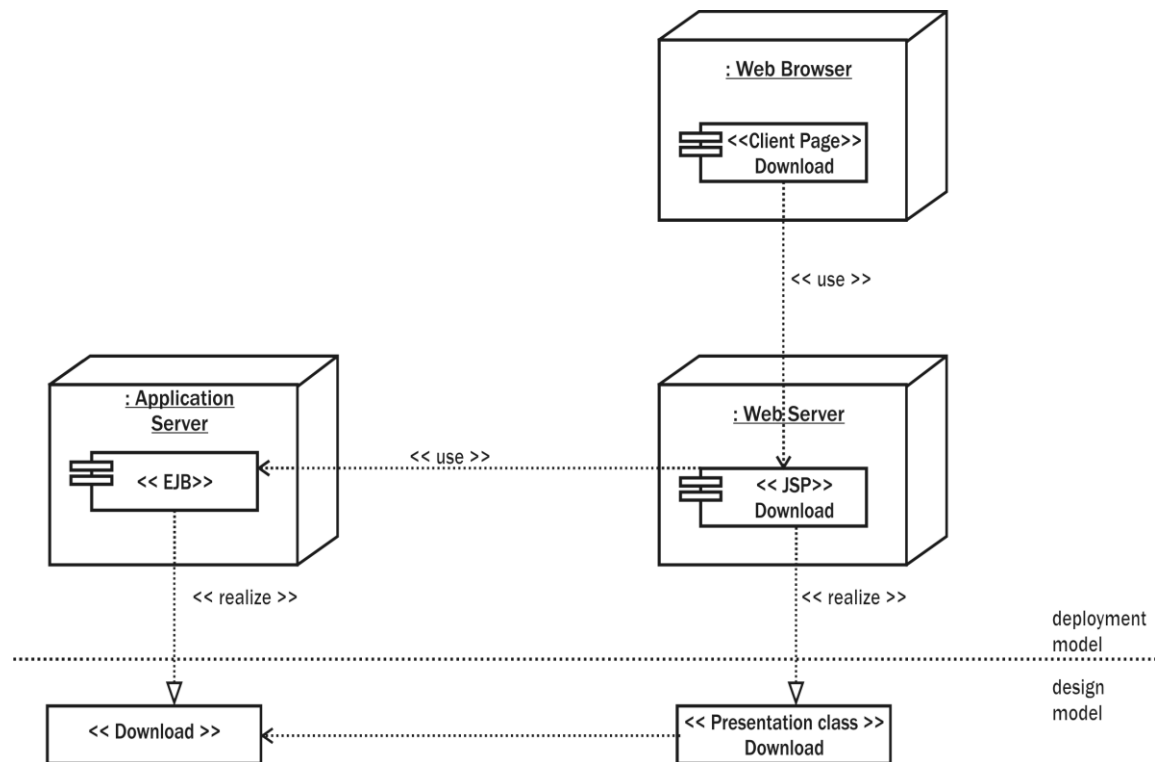


Figure 4.15: Deployment Diagram of the System with a focus on Download Module

v. Sequence Diagram

Sequence diagrams describe how and in what order the objects in a system function. The Sequence diagram of the new system is as presented in Figure 4.16.

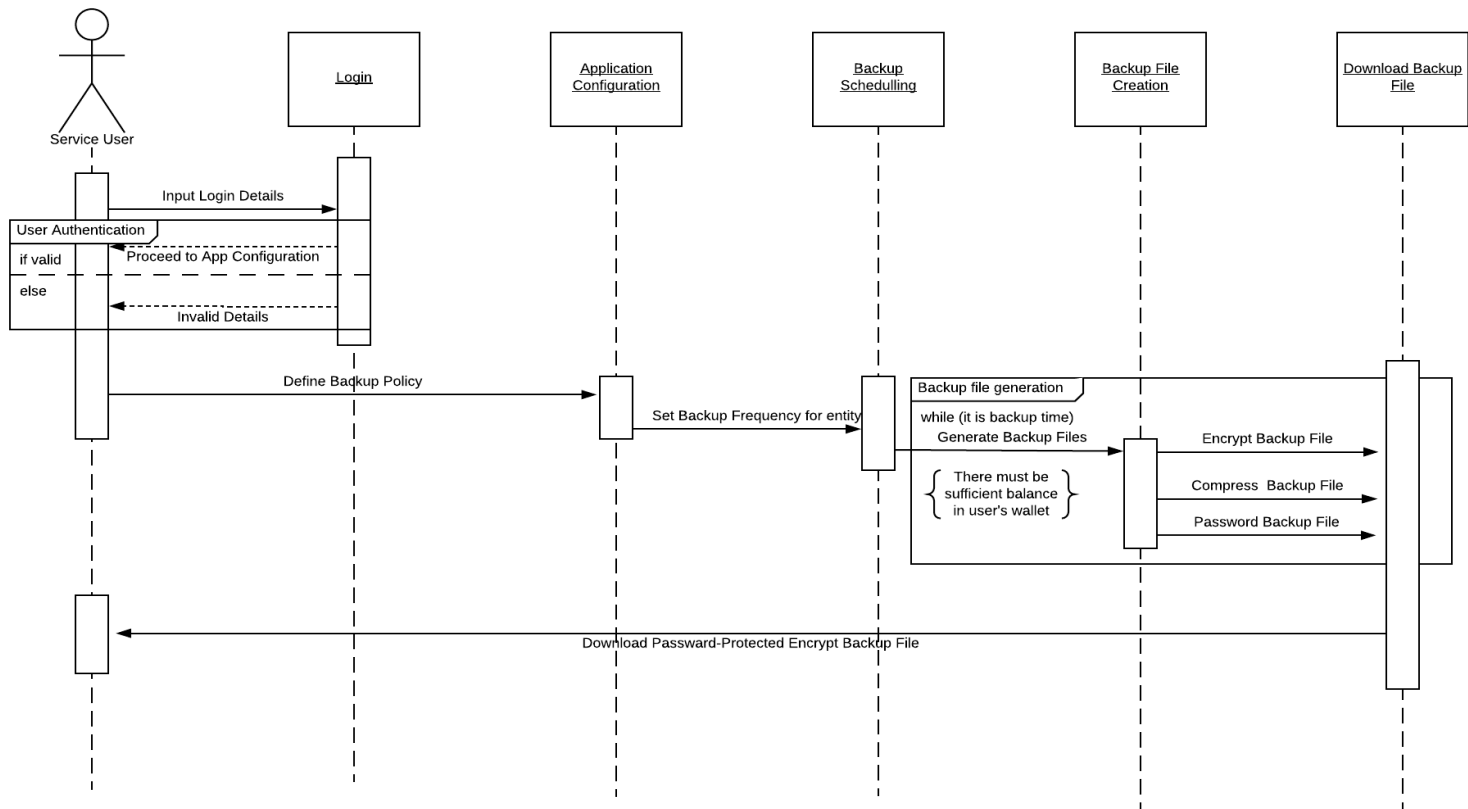


Figure 4.16: Sequence Diagram of the System

4.6 System Implementation

The actual implementation of the new system begins in this section. The section starts by looking at the various system requirements of both the client and server systems.

4.7.1 New System Requirements

The new system is based on client/server architecture. Therefore, the minimum system requirements of the server machine in terms of software and hardware are quite different from that of the client system. In the light of this, the requirement of the client system is stated separately from that of the server machine.

4.6.1.1 Hardware Requirements

i. Hardware requirements of client systems: The minimum hardware requirements for the client systems to effectively execute the new solution are as follows;

- a. 750MHZ processor speed
- b. 128MB of RAM
- c. 10 GB of Hard Disk Drive

ii. Hardware requirements of server systems: The minimum hardware requirements for the server machine to effectively execute the new solution are as follows;

- a. 2.00GHZ processor speed
- b. 2GB RAM size
- c. 500GB Hard Disk Drive

4.6.1.2 Software Requirements

i. Software requirements of client systems: The client systems do not require any special software to execute the new system. This is due to the fact that the new system is platform independent. All that is required of the client system is any graphical user interface (GUI) operating system and any web browser. However, for clients to execute the recovery module which is a standalone program written in Java, Java Development Kit (Java 1.8) or higher version is recommended. The system is compatible with

virtually all operating systems and all web browsers. As a matter of fact, the new system can even be executed on tablets and smartphones.

ii. Software requirements of server systems: The design system is to be hosted on the internet to give global access to our esteemed target users. However, it can be hosted on a local server for testing purpose. The minimal requirements of the local server in addition to GUI operating system and good browser are as follows;

- a. WildFly 10 Application Server
- b. MySql Server
- c. Java Development Kit (Java 1.8) or higher versions.
- d. Any Good GUI Server Operating System
- e. Web browser (Mozilla Firefox, Internet Explorer, or any other one)

4.6.2 Program Development

The software development task begins in this section. After the design process, all of the processes were at this point converted to executable codes.

i. Choice of Programming Environment

The new system is developed with Java Enterprise Edition (JEE) Technology in conjunction with Netbeans 8.2 Integrated Development Environment (IDE). MySql server was used to design and implement the back end (Database) portion of the system while Java Server Pages (JSP) was used as the front end. JavaScript was also used as the client-side scripting. Wildfly 8.0 was the application server that was employed for the deployment of the system.

ii. Language Justification

The entire business logic (Middleware) of the new system is completely developed with Java. This is because the combined Dynamic System Development Methodology (DSDM) and Object-Oriented Analysis and Design Methodology (OOADM) can fully be implemented with pure Object-Oriented Programming Language. Java is one of those Languages. Most Object-Oriented features such as polymorphism, inheritance, encapsulation and code reusability that were employed in the development of the new

system are easily programmable with Java. Above all, the researcher is an experienced Java developer as such, he is very comfortable with the Language.

4.6.3 System Testing

The new system was thoroughly tested in this section. Testing was done for all the modules one after another. After integration, the entire system was also tested.

4.6.3.1 Test Plan

Every unit of the software was tested to ascertain its functionality in accordance with the objectives of the system, errors were debugged and the final working system was then integrated. Testing of the system was done in phases as follows:

- i. The Login Procedure of Users
- ii. Application Registration Procedure
- iii. Application Configuration Procedure
- iv. Card Payment Procedure
- v. Data Backup Procedure
- vi. File Compression Procedure
- vii. Backup Files Encryption Procedure
- viii. File Decompression Procedure
- ix. File Decryption Procedure

4.6.3.2 Test Data

Some of the data that were used for the system testing are as follows:

i. Login

Admin Login

1. USER NAME: Admin
2. PASSWORD: Admin

User Login

1. USERNAME: KIT
2. PASSWORD: oiza4mein2018
3. DECRYPTION KEY: 4781

ii. Card payment

Users are credited with Five Thousand Naira (N 5,000.00) only to enable them to enjoy the backup module for a while on trial basis before they can subscribe. As the default amount in their wallet reduces, they are expected to credit their wallets using various payment options. Card payment is one of these options, the data shown hereafter were used to test this module;

- a. **Test Successful Transactions**
CARD NO: 6280511000000095
EXP DATE: Dec 2026
PIN: 0000
CVV: 123

- b. **Insufficient Fund Transactions**
CARD NO: 506102000000009996
EXP DATE: Jan 2026
PIN: 0000
CVV: 123

- c. **Fund Incorrect PIN**
CARD NO: 506102000000009997
EXP DATE: Jan 2026
PIN: 0000
CVV: 123

4.6.3.3 Actual Test Result versus Expected Test Result

The actual test result was compared with the expected result and the findings are as presented in Table 4.7.

Table 4.7: Actual Result versus Expected Test Result

Expected Results	Actual Results
Should retrieve and display the entire schema of remote databases	Was able to retrieve and display the entire schema of remote databases
Should allow users to configure their backup policy and function in accordance with the configuration	Was able to work in accordance with the backup policy defined by users in their configuration entity
To compress backup file so as to reduce the size and prevent it from virus attack	Was able to compress the file
To encrypt the backup file to prevent it from Man in the Middle (MITM) attack	Was able to generate an encrypted file

To decrypt the encrypted file into SQL script to enable users to restore their remote databases into the local system in the phase of a disaster. Was able to decrypt the files to SQL format when the right decryption keys are supplied. If the wrong keys are supplied, the files will not be decrypted.

4.6.3.4 Performance Evaluation

The researcher used a laptop with the following processor configuration: Intel(R) Core (TM) i5-2540M CPU @ 2.60GHz, RAM of 6GB and Hard disk of 500GB to conduct experiments in a view of collecting some performance data from the system. The Operating System used in the experiments was Windows 10 64-bit. In the experiments, various sizes of remote databases are used and the backup files were encrypted and compressed. The sizes of the backup files range from 64Kb to 20Mb.

Several performance metrics such as Encryption time, Compression size, CPU clock cycles and battery power were collected. The encryption time is considered the time that an encryption algorithm takes to produce a cypher text from a plaintext. Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as the total size of the plaintext in bytes encrypted divided by the encryption time.

The CPU process time is the time that a CPU is committed only to the particular process of calculations. It reflects the load of the CPU. The more CPU time used in the encryption process, the higher is the load of the CPU. The CPU clock cycles are a metric, reflecting the energy consumption of the CPU while operating on encryption operations. Each cycle of CPU will consume a small amount of energy.

The following comparison analysis tasks performed were as shown below;

1. A comparison is conducted between the results of the selected different encryption schemes in terms of the encryption time of five different encryption

algorithm with ten different sizes of backup files in .sql format. The result is as shown in Table 4.8 and Figure 4.17;

Table 4.8: Comparison between DES, 3DES, BF, AES, and the Hybrid Encryption Time(s)

Input File Size (Kb)	DES (s)	3DES (s)	BF (s)	AES (s)	M-AES(s)
64.00	0.01	0.02	0.01	0.01	0.01
128.00	0.02	0.05	0.01	0.02	0.03
512.00	0.06	0.19	0.05	0.10	0.10
1,024.00	0.13	0.38	0.10	0.19	0.20
5,120.00	0.64	1.91	0.51	0.96	1.00
8,192.00	2.02	3.06	1.82	1.53	1.60
10,240.00	3.27	3.83	2.03	1.91	2.01
15,360.00	3.91	5.74	3.54	2.87	3.01
18,432.00	4.29	6.89	3.85	3.44	3.61
20,480.00	4.55	7.66	4.05	3.83	4.01

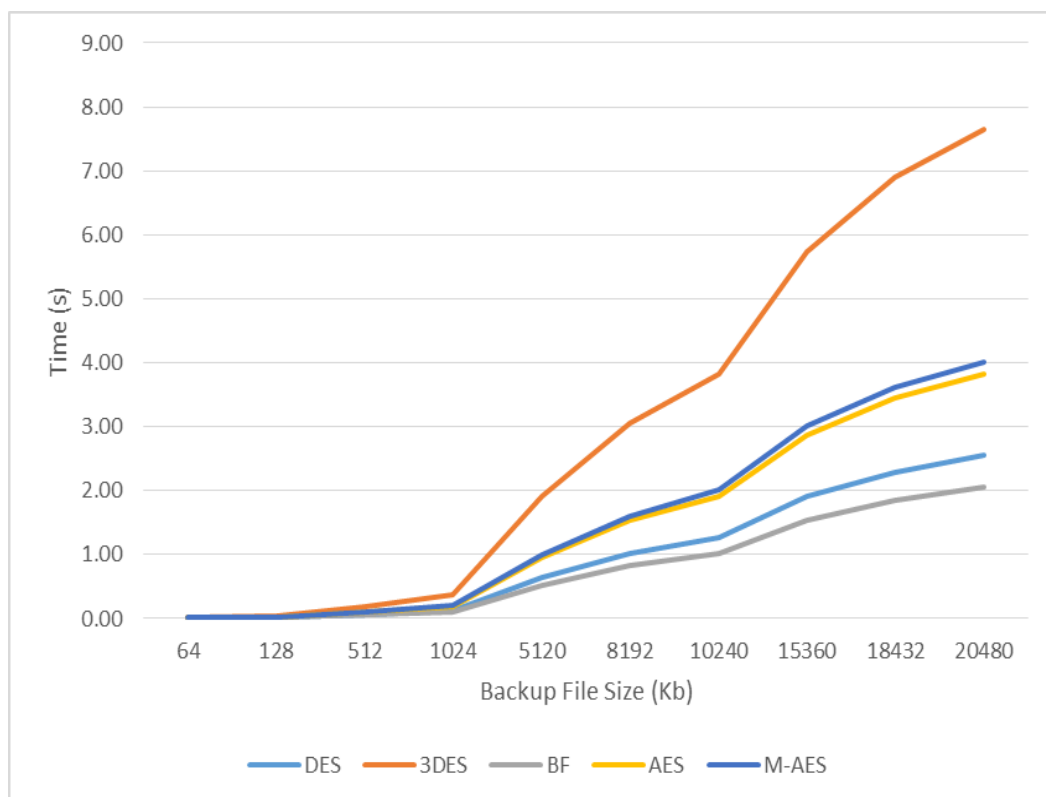


Figure 4.17: Input File Size versus Encryption Time for some Encryption Algorithms

The results are as expected. The Modified AES (M-AES) requires more processing time than the Advanced Encryption Standard (AES) because of its key-chaining nature. The results are shown in Table 4.8 and Figure 4.17 indicates also that the extra time added is not significant for many applications, knowing that the new system is an enhancement over AES and that it better in terms of file protection from Man in the Middle (MITM) attacks.

2. Also, in this work, the encrypted backup files are always compressed to enhanced easy transmission of the files over network facilities. A modified version of Huffman Coding compression algorithm is used to compress backup files. A comparison is also conducted between the results of the selected different compression schemes in terms of the sizes of the compressed files of four different compression algorithm with ten different sizes of backup files ranging from 64 Kb to 20Mb. The result is as shown in Table 4.9 and Figure 4.18;

Table 4.9: Comparison between LZW, Huffman Coding, Shannon Coding and the M-Huffman

Input File		Huffman	Shannon-Fan	M-Huffman
Size (Kb)	LZW (Kb)	Coding (Kb)	Coding (Kb)	Coding (Kb)
64.00	36.57	18.29	18.82	16.00
128.00	73.14	36.57	37.65	32.00
512.00	292.57	146.29	150.59	128.00
1,024.00	585.14	292.57	301.18	256.00
5,120.00	2,925.71	1,462.86	1,505.88	1,280.00
8,192.00	4,681.14	2,340.57	2,409.41	2,048.00
10,240.00	5,851.43	2,925.71	3,011.76	2,560.00
15,360.00	8,777.14	4,388.57	4,517.65	2,800.00
18,432.00	10,532.57	5,266.29	5,421.18	2,800.00
20,480.00	11,702.86	5,851.43	6,023.53	2,800.00

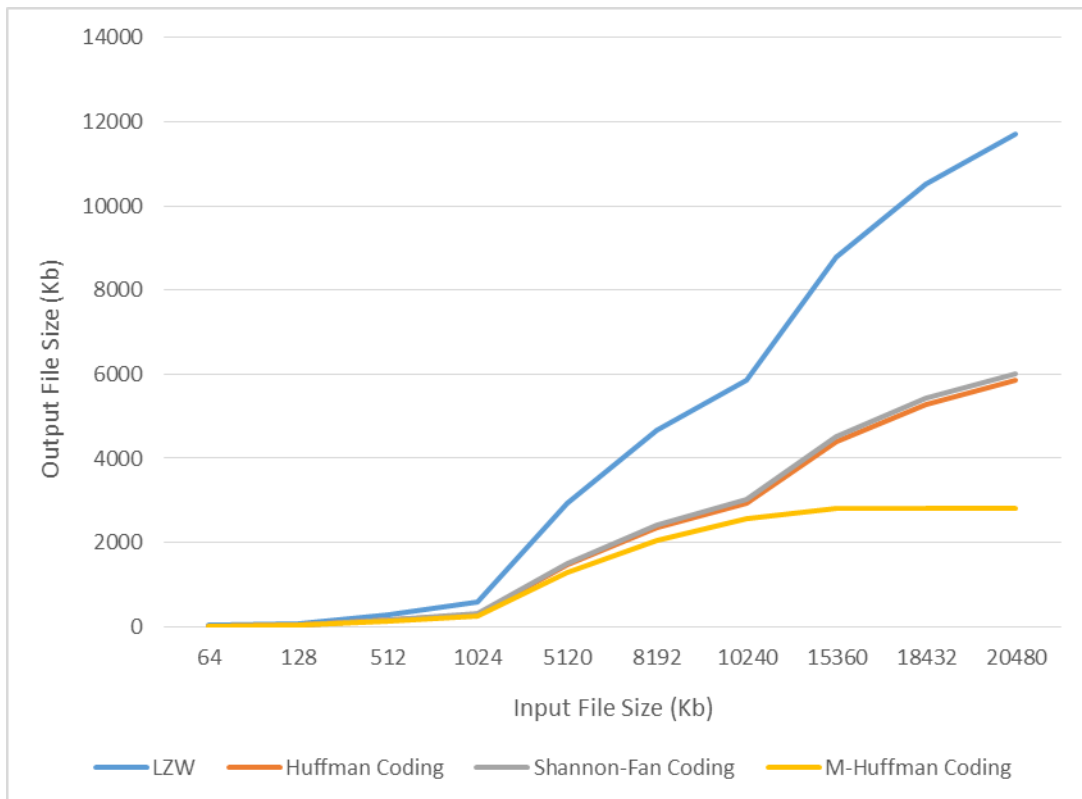


Figure 4.18: Input File Size versus Compressed Output File Size for some Compression Algorithms

The results from Table 4.9 and Figure 4.18 show the superiority of the hybrid compression algorithm used in the work over other algorithms. Figure 4.13 also shows that as the size of input file increases, the output file size of the hybrid algorithm converges.

4.6.3.5 Limitations of the System

The system is limited in the following ways;

- i. It works perfectly with only MySQL, Oracle and MSSql databases
- ii. Databases with pictures and video clips cannot be backed up and encrypted by the system.

4.6.4 System Security

The new system is highly secured as various aspects of software security are implemented in the system.

4.6.4.1 Password Protection

Despite the growing number of innovative ways to validating users, password-based validation is still one of the most popular methods of all (Barkadehi, Nilashi, Ibrahim, Fardi, and Samad, 2018). In this system, passwords are used to validate users. Upon keying in the right user name and its corresponding password, users are logged into their private dash board from where they can then navigate through all of the functionalities of the system. Another striking security feature of the new system is its stateless session management principle. When users' session are left unattended to for a period of 2minutes, the system automatically shuts down such sessions. Such users are forced to re-login to activate their sessions. This is done to avoid session hijacking by hackers.

4.6.4.2 Authentication

Security considerations of the new system form part of the maintenance. To ensure the integrity of a user's session, a three-tier authentication system was designed (login details validation, SMS confirmation, captcha verification). With all of these security policies, it will be very difficult for unauthorised users to gain access into the system.

4.6.4.3 Digital Signature

It is hoped that when the new system is finally hosted for active operation, the online portal will subscribe to third-party Secured Socket Layers (SSL) so that it will be running on https instead of the normal http. This digital certification is very necessary for the online payment module to be implementable.

4.6.4.4 Cryptography

To prevent the backup files from Man in the Middle (MITM) attack while on transit from service providers' premises to service users' premises, encrypted versions of the files are transmitted over the networks. Also, to enhance transmission processes, the

encrypted backup files are further compressed to reduce the size of the files. As an additional security mechanism, the compressed files are password protected.

4.6.5 Training

For the new system to function effectively and efficiently, educating and training of staff is necessary. Training is conducted for the staff selected to do the job of keying in data and running of the system. The members of staff selected, are trained for a period of time on how to manipulate and operate the system so as to be acquainted with the computer and the system designed. The staff members are also given procedural manuals to assist them in operating the system. They are also educated on how to safeguard files in the system to avoid unauthorized users from gaining access to the system files. They are also cautioned to be very sensitive and watchful for people who would want to get critical information from them via social engineering.

However, training is a continuous exercise. Whenever new functionality is added to the new system, there will be need to retrain the users of the system.

4.6.6 Documentation

This section explains how the new system can be put into use. For the users to successfully operate the system, they must have some level of knowledge in database design or software development. The system is hosted on a virtual server with www.eapps.com accessible through www.kitnija.com over the Internet. After starting up (booting) their computer systems, users can launch any web browser available on their system and visit the universal resources locator (URL) www.kitnija.com to commence their registration process. Users must complete all the login requirements before access is granted to them. Once access is granted, users can utilize the resources available on their dashboard. The portal is very interactive and user-friendly. Users of the system must be trained on how to operate the system. Since the system is not multilingual, users must be able to read and write in English Language.

4.6.7 System Conversion

Conversion from one information system (IS) to another is common in all organizations, regardless of type, size or location. On the information technology (IT)

side, conversion can involve hardware, operating system (OS), database management system (DBMS) and the database it supports, and/or application portfolio.

4.6.7.1 Changeover Procedures

System changeover is concerned with the smooth shift from one way of doing things to another with the view of reducing business disruption during the changeover. There are three main methods used: phased changeover, direct changeover and parallel changeover.

a. Phased Changeover: In this approach, one part of the overall system is changed. If problems arise, they are limited in scope and therefore non-critical. Once the system has been successfully changed in one area, the other areas can follow suit. The lessons learned during the initial phase changeover are used to ensure the successful changeover of the whole system.

b. Parallel running: In this approach, both the old and the new systems run side-by-side, using live data, so that project managers can compare the efficiency and reliability of the new system. Once they are satisfied, the old system is taken offline and the new system becomes fully active and utilized across the organisation.

c. Direct changeover: There is a single fixed point where one system stops being used and the new one becomes live. This is the cheapest, quickest and easiest form of system changeover but is also the most dangerous. With this method of changeover, if there is any system breakdown or inefficiency, the whole organisation will be thrown into serious problems.

4.6.7.2 Recommended Procedure

The parallel changeover is recommended for the new system. In this procedure, the new System will run side-by-side the old system using live data. Gradually, the non-compatible components of the old system will be eliminated while integrating the compatible components into the new system until the old system is eventually phased out.

4.7 Results and Discussion

This section discusses some of the output obtained from the new system. The results are displayed as screenshots with a brief explanation. A comprehensive sample outputs are presented at Appendix B.

i. Authentication Page

Figure 4.19 shows the authentication screen that appears as soon as the program is launched from the web browser. The new system is compatible with all known web browsers. On this page, existing users simply key in their login details. Upon confirming that the login details are correct, the system then triggers a secret key generated with Diffie Hellman encryption/decryption algorithms to the users' phone number as SMS. At the second level of authentication, the user must correctly provide the number before he/she will be granted access to the main functionalities of the system.

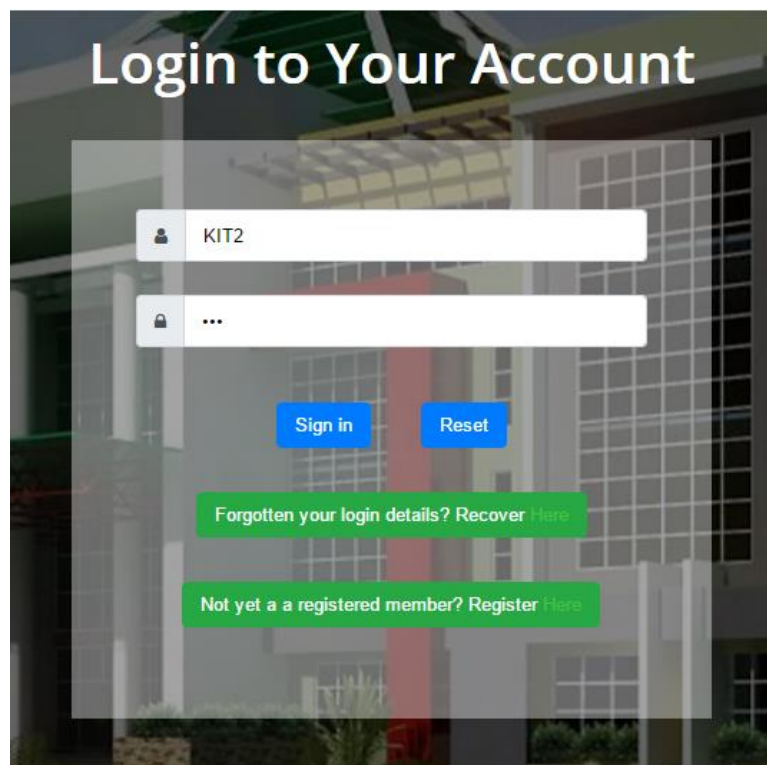
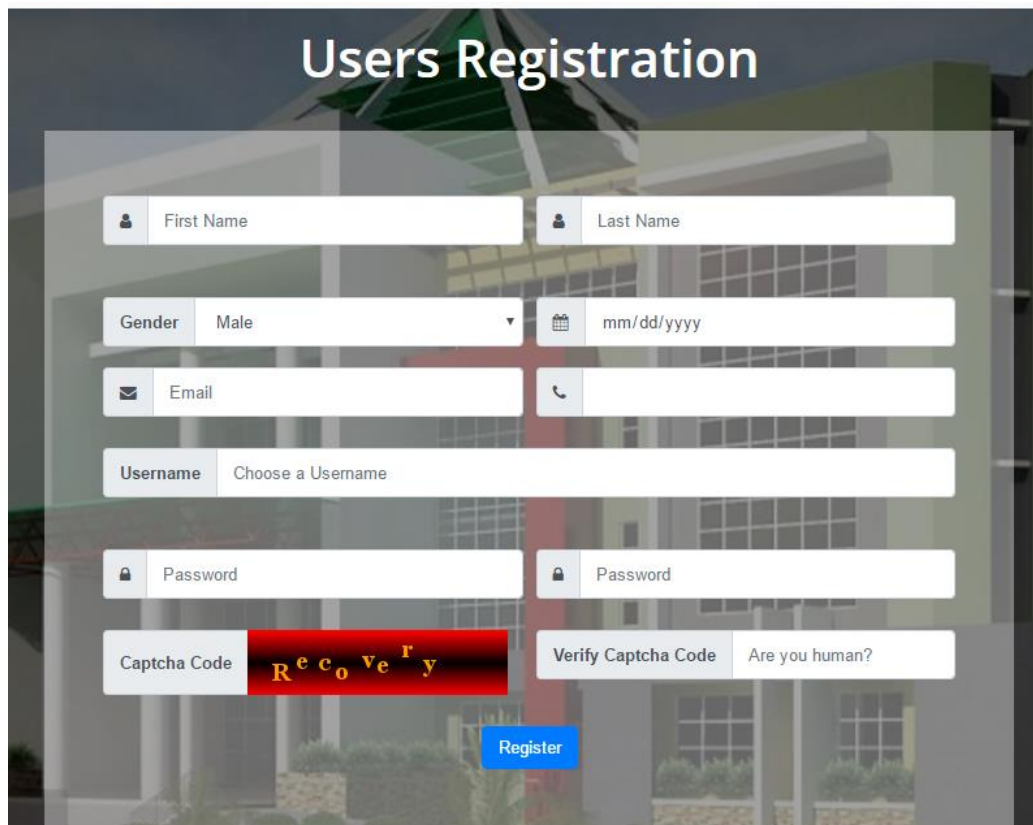


Figure 4.19: User Authentication Page

The page provides a link for existing users who have forgotten their login details to recover it. After providing correct answers to some security questions, the system will generate a new password and send it to the email address of the user contained in the database of the system. Also contained on the page is a link for new users to start a fresh registration process.

ii. New User Registration Page

Figure 4.20 shows the registration page. On this page, all the necessary data of new users are captured into the system. Upon submission of the form, a secret key generated with Diffie Hellman encryption/decryption algorithm is sent to the phone number of the user. He/she must confirm the number before the registration process is completed. This is to prevent robots and unserious users from registering with the system. The page also equipped with Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) which must be verified by users before their registration can be completed. All of these security features is to guide against Distributed Denial of Service (DDoS) attack.



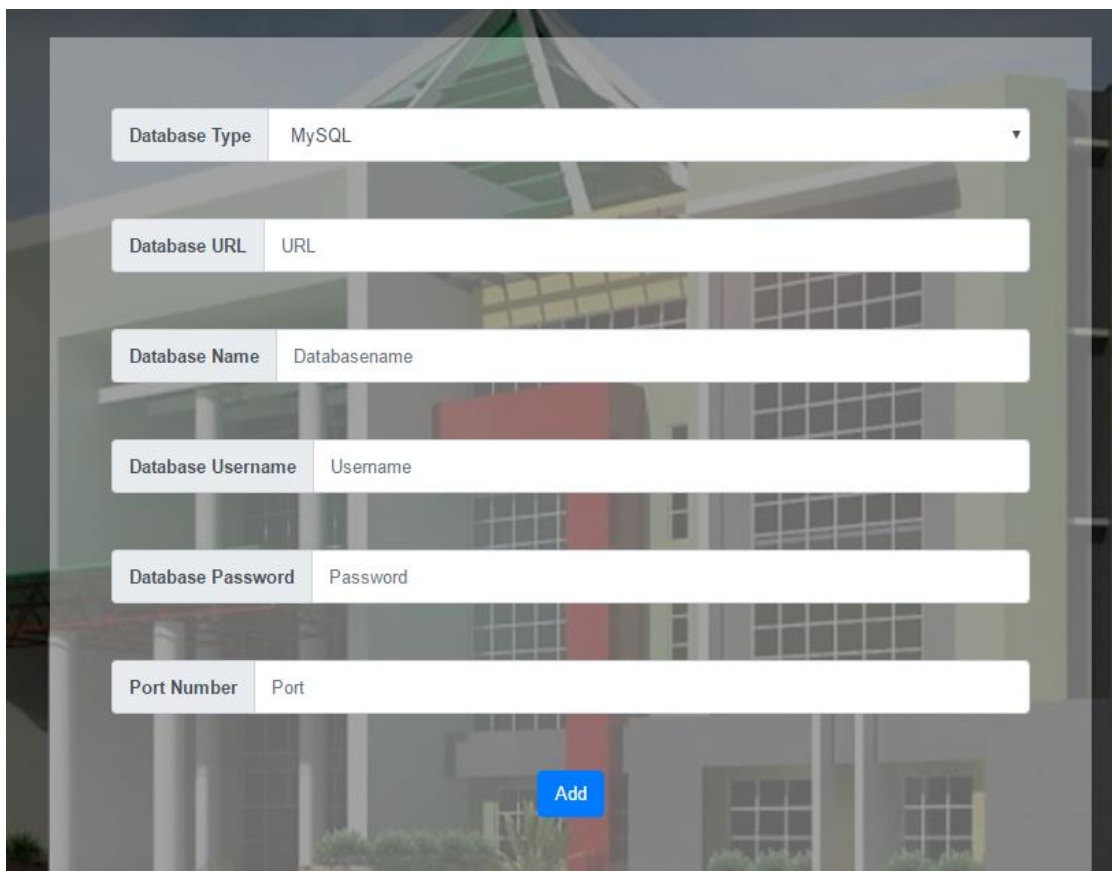
The image shows a web form titled "Users Registration" overlaid on a background image of a building. The form contains the following fields and elements:

- First Name** and **Last Name**: Text input fields.
- Gender**: A dropdown menu currently showing "Male".
- Date of Birth**: A date picker field showing "mm/dd/yyyy".
- Email**: A text input field with an envelope icon.
- Phone**: A text input field with a telephone icon.
- Username**: A text input field with the placeholder text "Choose a Username".
- Password**: Two text input fields, each with a lock icon, for password and confirmation.
- Captcha Code**: A text input field with a red background and the word "Recovery" in a stylized font.
- Verify Captcha Code**: A text input field with the text "Are you human?".
- Register**: A blue button at the bottom center of the form.

Figure 4.20: Registration Page

iii. Remote Database Registration Page

Figure 4.21 shows the remote database registration page. This module is only available to successfully authenticated users. The page permit users to register their remote databases to be monitored and backed up by the new system. The page captures all the necessary details to establish a connection to remote databases. After a successful registration of the database, the system creates a private compartment where the backup files of the database will be kept. A user can register more than one remote database, therefore the relationship between users and remote databases is one to many.



The image shows a web form for registering a remote database. The form consists of six input fields, each with a label on the left and a value on the right. The fields are: Database Type (MySQL), Database URL (URL), Database Name (Databasename), Database Username (Username), Database Password (Password), and Port Number (Port). Below the form is a blue button labeled 'Add'. The background of the form is a blurred image of a building.

Figure 4.21: Application Registration Page

iv. Application Configuration Page

Once remote databases have been successfully registered, users are then permitted to configure their backup policy for the databases. The system will establish a connection with the remote database, display the entire schema of the database showing all the entities contained in the database as shown in Figure 4.22. At this point, users are given the privilege to select the entities they want to back up and for every entity, they also select the corresponding backup frequency. Once this configuration is done, the system will continually be generating backups in accordance with the configuration pattern. Users are at liberty to modify their configuration at any point in time.

Backup Frequency

None

#	Entity Name	Check to Backup
1	alumni	<input type="checkbox"/>
2	application	<input type="checkbox"/>
3	applicationpin	<input type="checkbox"/>
4	appnumbering	<input type="checkbox"/>
5	bank	<input type="checkbox"/>
6	carryovernumbering	<input type="checkbox"/>
7	carryovers	<input type="checkbox"/>
8	counter	<input type="checkbox"/>
9	courseeregnumbering	<input type="checkbox"/>
10	department	<input type="checkbox"/>

Figure 4.22: Application Configuration Page

v. **Backup File Download Page**

This page displays all the backup files generated for a remote database. The files are presented in a chronological order as shown in Figure 4.23. The most recent files are at the top of the table while the older files are pushed down the table. For the purpose of memory space management, the system does not keep more than ten backup files per database. For security reason, the files are always in their encrypted form to prevent them from Man in the Middle (MITM) attack during download. A hybrid of Deffie Hellman Encryption and Advanced Encryption Standard (AES) Algorithm is used in the encryption mechanism of this system.

#	File Name	Entity	File Size (KB)	Date of Backup	Download
1	2019-04-100733161970.zip	courseregistrations paymenthistory userroles users	4,636.73	2019-06-13 11:40:01	Download
2	2019-04-100703161274.zip	courseregistrations paymenthistory userroles users	4,637.79	2019-04-10 07:57:54	Download
3	2019-04-100633164605.zip	courseregistrations paymenthistory userroles users	4,637.79	2019-04-10 06:35:00	Download
4	2019-03-301723432355.zip	courseregistrations paymenthistory userroles users	5,632.58	2019-04-10 06:05:01	Download

Figure 4.23: Backup File Download Page

vi. File Downloading Dialogue Box

File Downloading Dialogue Box given service user the privilege to download their backup files to their premises so that they will always have access to their files at all times. Compressed encrypted backup files are what is only available to be downloaded as shown in Figure 4.24. This is also for the purpose of security such that even if the file is intercepted and copied by Man-In-The-Middle (MITM) attack, they will still not be useful to them as they will not be able to decompress and decrypt the file to its useful formats.

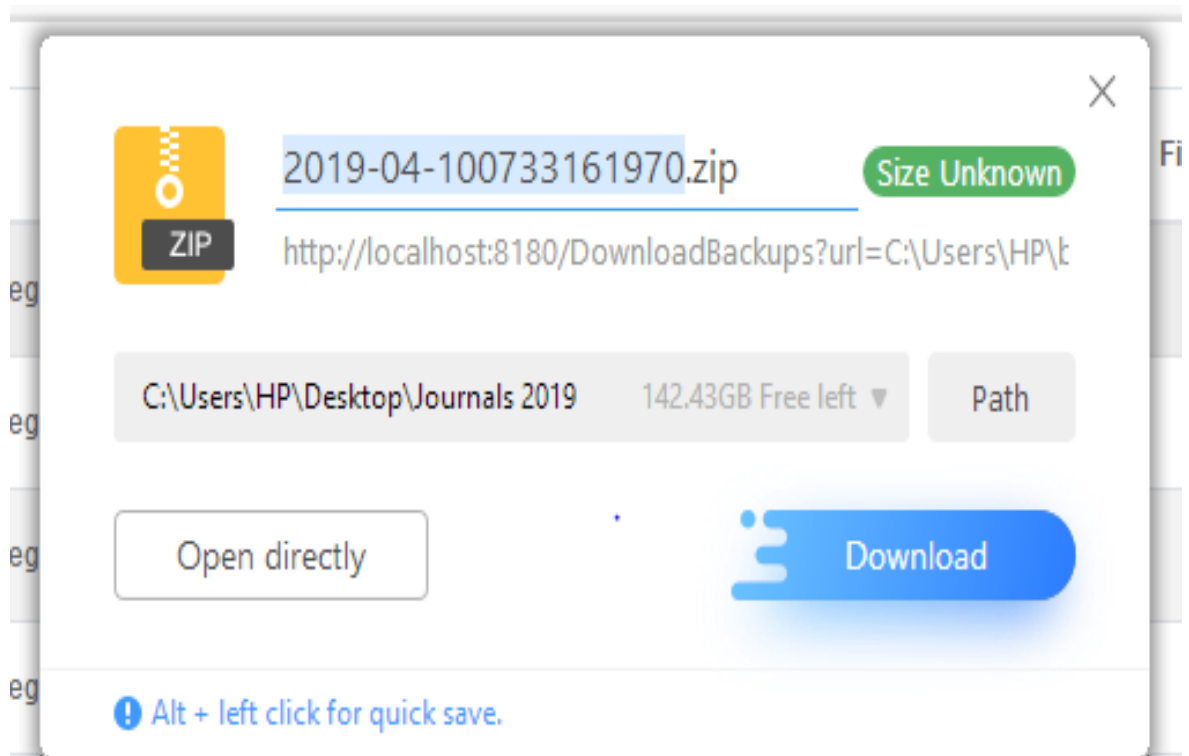
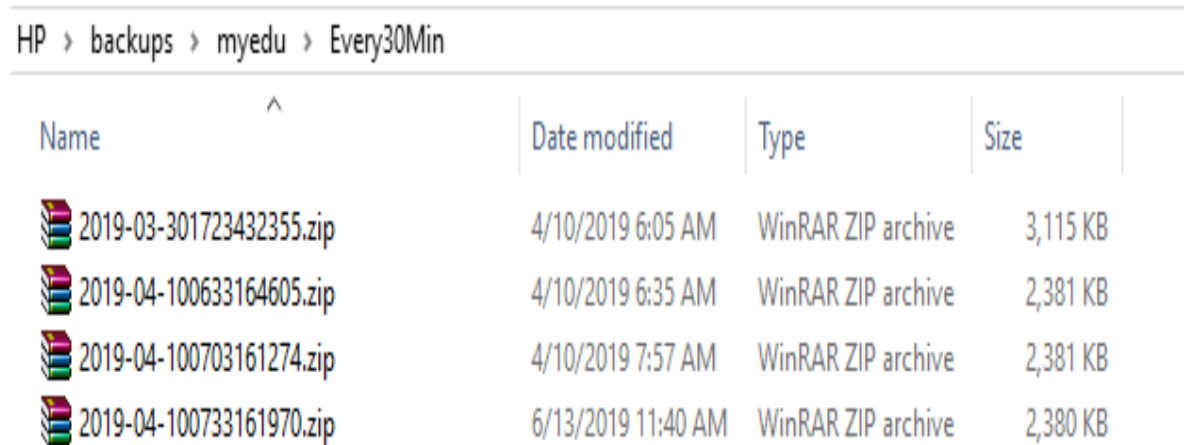


Figure 4.24: File Downloading Dialogue Box

vii. List of Downloaded Files

Backup Files are downloaded from the remote server to users' premises, they are always arranged in a chosen directory on their local system as shown in Figure 4.25; The files are always delivered as compressed file with the .zip extension.







Name	Date modified	Type	Size
 2019-03-301723432355.zip	4/10/2019 6:05 AM	WinRAR ZIP archive	3,115 KB
 2019-04-100633164605.zip	4/10/2019 6:35 AM	WinRAR ZIP archive	2,381 KB
 2019-04-100703161274.zip	4/10/2019 7:57 AM	WinRAR ZIP archive	2,381 KB
 2019-04-100733161970.zip	6/13/2019 11:40 AM	WinRAR ZIP archive	2,380 KB

Figure 4.25: List of Downloaded Files

viii. Decompressing and Decrypting Backup Files

Download Backup Files are also in their compressed and encrypted forms, they files have to be decompressed and decrypted to generate a pure sql scripts which contain the database instance that can be used to restore the applications to a consistence state in the phase of disaster. This Decompression and Decryption process required a secured password to be supplied by service users before the operation can be successful. Figure 4.26 shows the Decompression and Decryption interface;

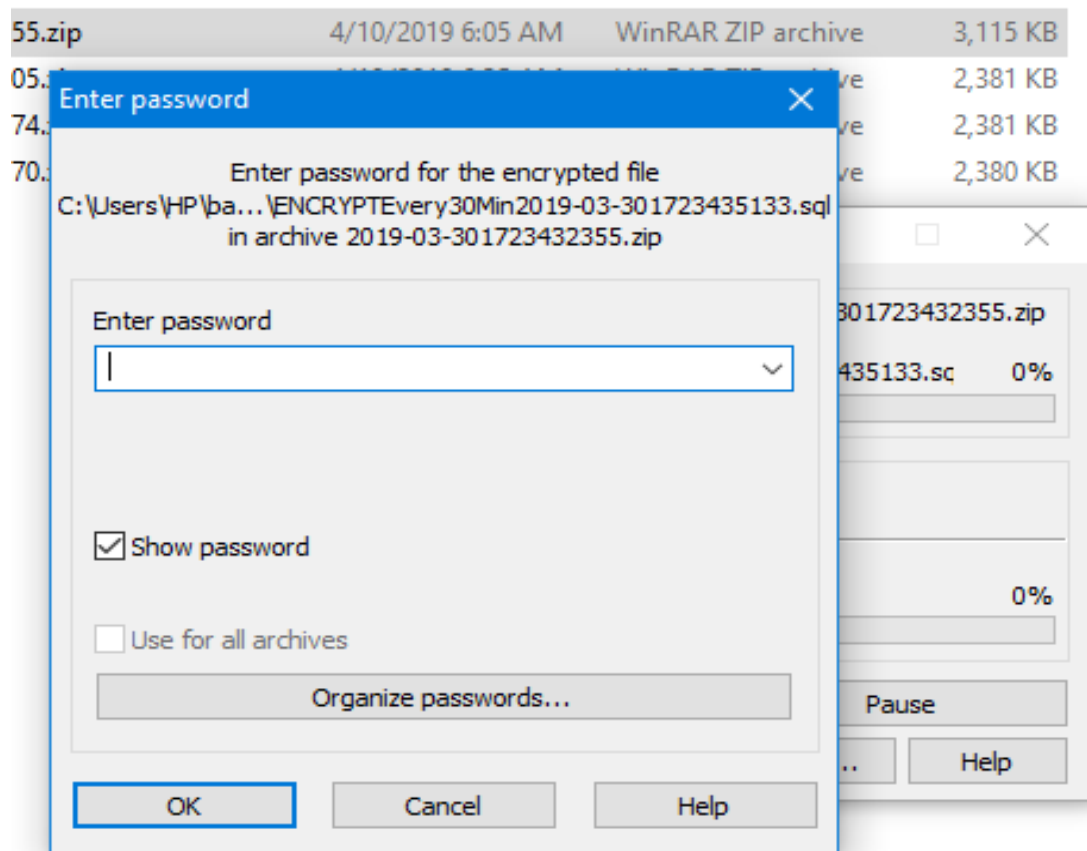
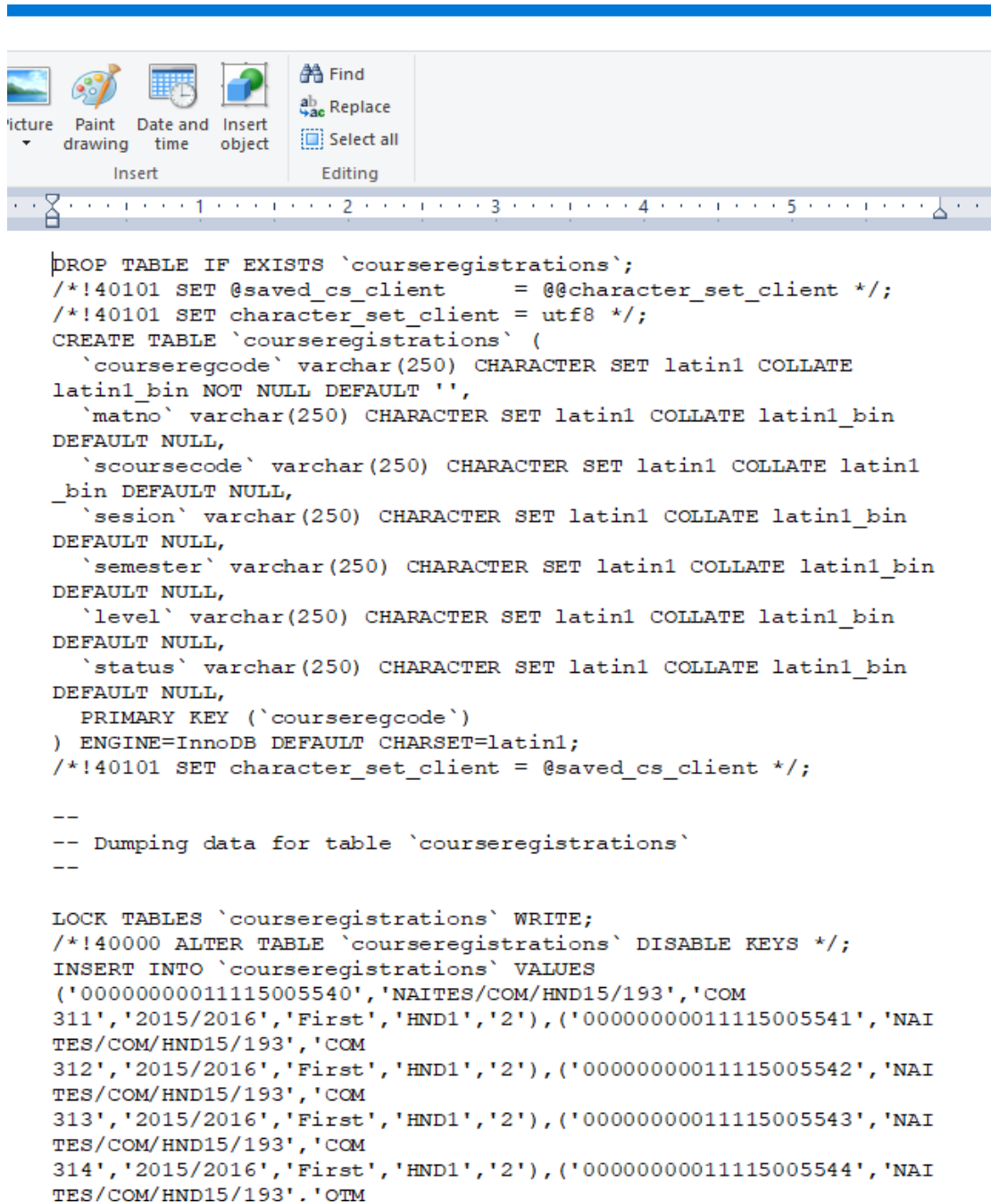


Figure 4.26: Decompression and Decryption interface

ix. Sample of Decompressed and Decrypted Backup Files

A successful decompressed and decrypted process generates a pure sql scripts as shown in Figure 4.27; with these types of remote database instances in the possession of Service Users, they are very much ready to recover from any form of Cyber disaster hitting their Service Providers.



```
PROP TABLE IF EXISTS `courseregistrations`;  
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client  = utf8 */;  
CREATE TABLE `courseregistrations` (  
  `courseregcode` varchar(250) CHARACTER SET latin1 COLLATE  
latin1_bin NOT NULL DEFAULT '',  
  `matno` varchar(250) CHARACTER SET latin1 COLLATE latin1_bin  
DEFAULT NULL,  
  `scoursecode` varchar(250) CHARACTER SET latin1 COLLATE latin1  
_bin DEFAULT NULL,  
  `sesion` varchar(250) CHARACTER SET latin1 COLLATE latin1_bin  
DEFAULT NULL,  
  `semester` varchar(250) CHARACTER SET latin1 COLLATE latin1_bin  
DEFAULT NULL,  
  `level` varchar(250) CHARACTER SET latin1 COLLATE latin1_bin  
DEFAULT NULL,  
  `status` varchar(250) CHARACTER SET latin1 COLLATE latin1_bin  
DEFAULT NULL,  
  PRIMARY KEY (`courseregcode`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
/*!40101 SET character_set_client  = @saved_cs_client */;  
  
--  
-- Dumping data for table `courseregistrations`  
--  
  
LOCK TABLES `courseregistrations` WRITE;  
/*!40000 ALTER TABLE `courseregistrations` DISABLE KEYS */;  
INSERT INTO `courseregistrations` VALUES  
(  
'00000000011115005540', 'NAITES/COM/HND15/193', 'COM  
311', '2015/2016', 'First', 'HND1', '2'), ('00000000011115005541', 'NAI  
TES/COM/HND15/193', 'COM  
312', '2015/2016', 'First', 'HND1', '2'), ('00000000011115005542', 'NAI  
TES/COM/HND15/193', 'COM  
313', '2015/2016', 'First', 'HND1', '2'), ('00000000011115005543', 'NAI  
TES/COM/HND15/193', 'COM  
314', '2015/2016', 'First', 'HND1', '2'), ('00000000011115005544', 'NAI  
TES/COM/HND15/193', 'OTM
```

Figure 4.27: Sample of Decompressed and Decrypted Backup Files

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

The research focused on Service Users being able to define and implement their private Disaster Recovery Policies. To achieve this goal, a set of objectives were drawn up and using the philosophy of software development, the objectives were interpreted into a third-party platform independent software solution. The system acts as a middleware between Service Providers and Service Users. It acquires Disaster Recovery plans from Service Users and uses same to generate backup files of their remote databases hosted in the premises of Service Providers. These backup files are generated in an interval of times as configured by Service Users. The files are then encrypted using a strong encryption mechanism to prevent them from being accessed by Man in the Middle (MITM) or unauthorized users. A hybrid of Diffie Hellman Encryption and Advanced Encryption Standard (AES) Algorithms is used in the encryption mechanism. Also, the backup files are compressed and password protected using a hybrid of Huffman Coding and Shannon-Fan Coding compression algorithms to reduce their sizes so as to make their transportation of the cyberspace much easier. These password-protected, compressed and encrypted files are then delivered to the premises of Service Users.

5.2 Conclusion

Disaster Recovery Models are heavily managed by Service Providers. Service Users do not have control over these models and policies since they are always on the premises of Service Providers. The issues of incompatibility of storage services used by various Service Providers is also one of the challenges this research hopes to address. Most of the available Disaster Recovery Solutions uses server replication method where the entire active server is replicated and stored in backup servers of the Service Providers thereby making it very difficult for Service Users to access since the backup servers are still within the premises of Service Providers. It is also not a good decision to handover server replicates to Service Users since it comprises data of multiple users. It is in the light of this understanding that the researcher advocates for a User-Centric Disaster Model. This model grants Service Users the privilege to define and deploy their private Disaster Recovery Policy. With the new model, Service Users define their Disaster Recovery Policy and register same with the system which then visits Service Providers

at an interval of time, create private backup files and drop them in Service User premises.

5.3 Recommendations

The following are the recommendations made based on the findings of this study.

- i. Governments and non-governmental institutions should create and deploy their private Disaster Recovery Policy instead of depending on Service Providers, hence the new system is useful as it provides Service Users with the opportunity to create and deploy their Disaster Recovery Policies instead of depending on Service Providers.
- ii. Service Providers should work together to provide a common and compatible storage service for Service Users to enable them to migrate easily from one provider to another in the phase of disaster
- iii. Disaster Recovery Models deployed by Service Providers should seek for the opinion of Service Users as per which entity in their applications they desire to be backed up and at which frequency they desire the backup.

5.3.1 Application Areas

The research is very useful in many governmental and non-government institutions around the globe where application hosting on cyberspace has become eminent. In Nigeria for instance, it will be useful for Military and all the Para-Military to deploy this solution to enable them to bring their operation data which are currently hosted by Service Providers back to their premises. All Universities, Polytechnics and Colleges of Education in Nigeria have one portal or the other with which the institutions are being managed. Currently, all of these portals have their operational data in the custody of Service Providers. The findings of this research are useful to such institutions as it will help them to have access to their operational data and have a base to run back to in the event of a disaster. The findings of this research are also very useful to all the commercial banks, examination bodies and small and medium scale businesses all over the globe.

5.3.2 Suggestions for Further Research

User Centric Model for Cyber Disaster Recovery System has addressed the issues of incompatibility of storage services used by various Service Providers and the inaccessibility of Backup files stored in the premises of Service Providers by Service Users. However, the contribution to knowledge is not completely exhaustive. The model can be extended to be compatible with all Relational Database Management Systems (RDBMS). Also, further research work should be done to investigate the possibility of coordinating all Service Providers in a view to deploying a common ground for their storage management policy.

5.4 Contribution to Knowledge

- i. The study has contributed to research on the area of Cyberspace Disaster Recovery Models by developing a paradigm shift model which focuses more on Service Users rather than Service Providers.
- ii. The research also creates a gateway to address the issues of incompatibility of storage services used by various Service Providers
- iii. The research is also a very handy solution to the threat and havoc posted on cyberspace by ransomware.

REFERENCES

- Abramson, D., Giddy, J., and Kotler, L. (2000, May). High performance parametric modeling with Nimrod/G: Killer application for the global grid?. In *Proceedings 14th International Parallel and Distributed Processing Symposium. IPDPS 2000* (pp. 520-528). IEEE.
- Adamov, A., and Erguvan, M. (2009, October). The truth about cloud computing as new paradigm in IT. In *2009 International Conference on Application of Information and Communication Technologies* (pp. 1-3). IEEE.
- Attanasio, C. R. (1973, March). Virtual machines and data security. In *Proceedings of the workshop on virtual computer systems* (pp. 206-209). ACM.
- Bachrach, D. G., and Rzeszut, E. J. (2014). Don't trust anyone over... Anything. In *10 don'ts on your digital devices* (pp. 107-120): Springer.
- Bahrs, P., Lillie, B. T., and Van Horn, I. B. (2006). Method and apparatus for portable universal resource locator and coding across runtime environments. In: Google Patents. U.S. Patent No. 7,134,076. Washington, DC: U.S. Patent and Trademark Office.
- Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., ... and Warfield, A. (2003, October). Xen and the art of virtualization. In *ACM SIGOPS operating systems review* (Vol. 37, No. 5, pp. 164-177). ACM.
- Barkadehi, M. H., Nilashi, M., Ibrahim, O., Fardi, A. Z., and Samad, S. (2018). Authentication systems: A literature review and classification. *Telematics and Informatics*. 35(5), 1491-1511.
- Basu, S., Bardhan, A., Gupta, K., Saha, P., Pal, M., Bose, M., ... and Sarkar, P. (2018, January). Cloud computing security challenges and solutions-a survey. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 347-356). IEEE.

- Blaze, M., Feigenbaum, J., Ioannidis, J., and Keromytis, A. D. (1999). The role of trust management in distributed systems security. In *Secure internet programming* (pp. 185-210): Springer.
- Boberski, M. (2010). *The ten most critical Web application security risks*. Tech. rep., OWASP.
- Booch, G. (2005). *The unified modeling language user guide*: Pearson Education India.
- Bressoud, T. C., and Schneider, F. B. (1996). Hypervisor-based fault tolerance. *ACM Transactions on Computer Systems (TOCS)*, 14(1), 80-107.
- Brooks, V. Z., Terman, C., Agarwal, A., Berners-Lee, T., Knight Jr, T., Chow, A., and Natkin, L. (2003). Computer Science and Artificial Intelligence Laboratory Manual (CSAIL).
- Brumback, G. (2015). America's oldest professions: Warring and spying. *North Charleston, SC: Create Space Independent Publishing Platform*.
- Buyya, R., and Son, J. (2018, May). Software-defined multi-cloud computing: a vision, architectural elements, and future directions. In *International Conference on Computational Science and Its Applications* (pp. 3-18). Springer, Cham.
- CAI, J., and WANG, S.-m. (2009). Cloud computing system instances based on google. *Computer Knowledge and Technology*, 5(25), 7093-7095.
- Callegati, F., Cerroni, W., and Ramilli, M. (2009). Man-in-the-Middle Attack to the HTTPS Protocol. *IEEE Security and Privacy*, 7(1), 78-81.
- Caraman, M. C., Moraru, S. A., Dan, S., and Kristaly, D. M. (2009). Romulus: Disaster tolerant system based on kernel virtual machines. *Annals of DAAAM and Proceedings*. 1671-1673.

- Chang, F. W., Ji, M., Leung, S. T., MacCormick, J., Perl, S. E., and Zhang, L. (2002, January). Myriad: Cost-Effective Disaster Tolerance. In *File and Storage Technologies (FAST)* (Vol. 2, p. 8).
- Chatterjee, P., Mahalingam, A., Jayaraman, R., and Maliakal, J. (2016). Data recovery point review in a continuous data protection system. *U.S. Patent No. 9,495,370*. Washington, DC: U.S. Patent and Trademark Office.
- Chen, H., and Zheng, Z. (2008, December). Design and implementation of XML-based GUI for cross platform backup system. In *2008 International Symposium on Computer Science and Computational Technology* (Vol. 2, pp. 380-382). IEEE.
- Choudhary, V. (2007, January). Software as a service: Implications for investment in software development. In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)* (pp. 209a-209a). IEEE.
- Clark, R. A., and Knake, R. K. (2010). Cyber war: The next threat to national security and what to do about it *New York: Ecco*.
- Cully, B., Lefebvre, G., Meyer, D., Feeley, M., Hutchinson, N., and Warfield, A. (2008, April). Remus: High availability via asynchronous virtual machine replication. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation* (pp. 161-174).
- De Assunção, M. D., Di Costanzo, A., and Buyya, R. (2009, June). Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters. In *Proceedings of the 18th ACM international symposium on High performance distributed computing* (pp. 141-150). ACM.

- Descher, M., Masser, P., Feilhauer, T., Tjoa, A. M., and Huemer, D. (2009, March). Retaining data control to the client in infrastructure clouds. In *2009 International Conference on Availability, Reliability and Security* (pp. 9-16). IEEE.
- Dhane, S. V., and Joshi, P. (2015). Public auditing system with auto-data recovery system on cloud scheme. *International Journal of Science and Research (IJSR)*, 4(11) 81-93.
- Endo, P. T., Gonçalves, G. E., Kelner, J., and Sadok, D. (2010, May). A survey on open-source cloud computing solutions. In *Brazilian Symposium on Computer Networks and Distributed Systems* (Vol. 71).
- Feit, S. (1998). *TCP/IP: Architecture, Protocols, and Implementation with IPv6 and IP Security*. McGraw-Hill, Inc..
- Fidler, D. P. (2018). Cybersecurity and the New Era of Space Activities. *Digital and Cyberspace Policy Program, April 2018*.
- Foster, I., Zhao, Y., Raicu, I., and Lu, S. (2008). Cloud computing and grid computing 360-degree compared. *arXiv preprint arXiv:0901.0131*.
- Gadia, S. (2009). Cloud computing: An auditor's perspective. *ISACA Journal*, 6, 24.
- Gajek, S., Liao, L., and Schwenk, J. (2007, November). Breaking and fixing the inline approach. In *Proceedings of the 2007 ACM workshop on Secure web services* (pp. 37-43). ACM.
- Garg, D., Thakral, A., Nalwa, T., and Choudhury, T. (2018, June). A Past Examination and Future Expectation: Ransomware. In *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)* (pp. 243-247). IEEE.

- Gharge, D. M., Halse, S. V., and Jagtap, S. B. (2013). Forward Selection Call Admission Control with Intrusion Detection System in IEEE 802.16E WiMAX Network. *vol, 2*, 1-9.
- Gibson, W. (1995). *Neuromancer*. 1984. *New York: Ace*.
- Grolinger, K., Capretz, M. A., Mezghani, E., and Exposito, E. (2013, June). Knowledge as a service framework for disaster data management. In *2013 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp. 313-318). IEEE.
- Gunneriusson, H., and Ottis, R. (2013). Cyberspace from the hybrid threat perspective. *Journal of Information Warfare, 12*(3), 67-77.
- Haji, A., Letaifa, A. B., and Tabbane, S. (2010). Cloud Computing: Several Cloud-oriented Solutions. In *4th International Conference On Advanced Engineering Computing and Applications in Sciences,(ADVCOMP2010)*(Vol. 25).
- Hjortdal, M. (2011). China's use of cyber warfare: Espionage meets strategic deterrence. *Journal of Strategic Security, 4*(2), 2.
- Iasiello, E. (2014). Is cyber deterrence an illusory course of action? *Journal of Strategic Security, 7*(1), 6.
- Indirani, G., and Selvakumar, K. (2012). Swarm based detection and defense technique for malicious attacks in mobile ad hoc networks. *International Journal of Computer Applications, 50*(19).
- Jacobs, D. (2005). Enterprise software as service. *Queue, 3*(6), 36-42.
- Jangra, A., and Bala, R. (2012). A survey on various possible vulnerabilities and attacks in cloud computing environment. *International Journal of Computing and Business Research, 3*(1), 1-13.

- Jinal, P. T., and Ashish, D. P. (2017). A comprehensive survey: Ransomware attacks prevention, monitoring and damage control. *International Journal of Research and Scientific Innovation (IJRSI)*, IV(VIS), 116 - 121.
- Kandukuri, B. R., and Rakshit, A. (2009, September). Cloud security issues. In *2009 IEEE International Conference on Services Computing* (pp. 517-520). IEEE.
- Kaufman, L. M. (2009). Data security in the world of cloud computing. *IEEE Security and Privacy*, 7(4), 61-64.
- Keeton, K., Beyer, D., Brau, E., Merchant, A., Santos, C., and Zhang, A. (2006, April). On the road to recovery: restoring data after disasters. In *ACM SIGOPS Operating Systems Review* (Vol. 40, No. 4, pp. 235-248). ACM.
- Kenney, M. (2015). Cyber-terrorism in a post-stuxnet world. *Orbis*, 59(1), 111-128.
- Kivity, A., Kamay, Y., Laor, D., Lublin, U., and Liguori, A. (2007). Kvm: The linux virtual machine monitor In: *Proceedings of the Linux Symposium*, 225-230.
- Kuhn, D., Alapati, S., and Nanda, A. (2013). *Rman recipes for oracle database 12c: A problem-solution approach*: Apress.
- Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999). Trawling the web for emerging cyber-communities. *Computer networks*, 31(11-16), 1481-1493.
- Lagazio, M., Sherif, N., and Cushman, M. (2014). A multi-level approach to understanding the impact of cyber crime on the financial sector. *Computers and Security*, 45, 58-74.
- Lahiri, T., Ganesh, A., Weiss, R., and Joshi, A. (2001, May). Fast-Start: quick fault recovery in oracle. In *ACM SIGMOD Record* (Vol. 30, No. 2, pp. 593-598). ACM.

- Lawton, G. (2008). Developing software online with platform-as-a-service technology. *Computer*, 41(6).
- Lee, I., Jeong, S., Yeo, S., and Moon, J. (2012). A novel method for sql injection attack detection based on removing sql query attribute values. *Mathematical and Computer Modelling*, 55(1-2), 58-68.
- Lehto, M., and Neittaanmäki, P. (2015). *Cyber security: Analytics, technology and automation* (Vol. 78): Springer.
- Liu, J. N., Liu, G., and Yang, T. (2009, March). Design and implementation of an embedded backup system. In *Eighth International Symposium on Optical Storage and 2008 International Workshop on Information Data Storage* (Vol. 7125, p. 71251O). International Society for Optics and Photonics.
- Loganayagi, B., and Sujatha, S. (2012). Enhanced cloud security by combining virtualization and policy monitoring techniques. *Procedia Engineering*, 30, 654-661.
- Luo, S., Wang, Y., Huang, W., and Yu, H. (2016, December). Backup and Disaster Recovery System for HDFS. In *2016 International Conference on Information Science and Security (ICISS)* (pp. 1-4). IEEE.
- Maurer, T. (2011). Cyber norm emergence at the United Nations—an analysis of the UN's activities regarding cyber-security. *Belfer Center for Science and International Affairs*, 6668.
- Mayur, S. A., and Vani, N. (2016). Server virtualization using cloud environment for data storage and backup *International Journal of Science and Research (IJSR)*, 5(6), 23-31.

- Milojčić, D., Llorente, I. M., and Montero, R. S. (2011). Opennebula: A cloud management tool. *IEEE Internet Computing*, 15(2), 11-14.
- Mirkovic, J., and Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2), 39-53.
- Mishra, S., Mohapatra, S. K., Mishra, B. K., and Sahoo, S. (2018). Analysis of Mobile Cloud Computing: Architecture, Applications, Challenges, and Future Perspectives. In *Applications of Security, Mobile, Analytic, and Cloud (SMAC) Technologies for Effective Information Processing and Management* (pp. 81-104). IGI Global.
- Nolen, S., Henry, C., Patrick, T., Kevin, R. B. (2016, June). Cryptolock (and drop it): stopping ransomware attacks on user data. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)* (pp. 303-312). IEEE.
- Nurmi, D., Wolski, R., Grzegorzcyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D. (2009, May). The eucalyptus open-source cloud-computing system. In *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid* (pp. 124-131). IEEE Computer Society.
- Ousterhout, J., Agrawal, P., Erickson, D., Kozyrakis, C., Leverich, J., Mazières, D., ... and Rumble, S. M. (2010). The case for RAMClouds: scalable high-performance storage entirely in DRAM. *ACM SIGOPS Operating Systems Review*, 43(4), 92-105.
- Paquin, C., and Thompson, G. (2010). U-prove ctp white paper. *Microsoft Corporation*.
- Paschal, O. (2016). Backup as a service from mtn – cloud storage packages. Retrieved from <https://www.naijatechguide.com/2015/08/backup-as-service-from-mtn-cloud.html>

- Patterson, D. A., Gibson, G., and Katz, R. H. (1988). *A case for redundant arrays of inexpensive disks (raid)* (Vol. 17, No. 3, pp. 109-116). ACM.
- Peng, J., Zhang, X., Lei, Z., Zhang, B., Zhang, W., and Li, Q. (2009, December). Comparison of several cloud computing platforms. In *2009 Second international symposium on information science and engineering* (pp. 23-27). IEEE.
- Popović, K., and Hocenski, Ž. (2010, May). Cloud computing security issues and challenges. In *The 33rd International Convention MIPRO* (pp. 344-349). IEEE.
- Rachana, S., and Guruprasad, H. (2014). Emerging security issues and challenges in cloud computing. *International Journal of Engineering Science and Innovative Technology*, 3(2), 485-490.
- Rajagopalan, S., Cully, B., O'Connor, R., and Warfield, A. (2012). Secondsite: Disaster tolerance as a service. *Acm Sigplan Notices*, 47(7), 97-108.
- Raphael, B. (1986). *When disaster strikes: A handbook for the caring professions*: Hutchinson London.
- Rebah, H. B., and Sta, H. B. (2016, July). Disaster Recovery as a Service: A Disaster Recovery Plan in the Cloud for SMEs. In *2016 Global Summit on Computer and Information Technology (GSCIT)* (pp. 32-37). IEEE.
- Sánchez, C., and Goldberg, S. R. (2003). How to handle the threat of catastrophe. *Journal of Corporate Accounting and Finance*, 14(6), 35-40.
- Schmitt, M. N. (2015). In defense of due diligence in cyberspace. *Yale LJF*, 125, 68.
- Schram, A., and Anderson, K. M. (2012, October). MySQL to NoSQL: data modeling challenges in supporting scalability. In *Proceedings of the 3rd annual*

conference on Systems, programming, and applications: software for humanity (pp. 191-202). ACM.

Seccombe, A., Hutton, A., Meisel, A., Windel, A., Mohammed, A., and Licciardi, A. (2009). Security guidance for critical areas of focus in cloud computing. *Cloud Security Alliance*, 2(1), 2-70.

Shovon, A. R., Roy, S., Sharma, T., and Whaiduzzaman, M. (2018, June). A restful e-governance application framework for people identity verification in cloud. In *International Conference on Cloud Computing* (pp. 281-294). Springer, Cham.

Silva, B., Maciel, P., Tavares, E., and Zimmermann, A. (2013, June). Dependability models for designing disaster tolerant cloud computing systems. In *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (pp. 1-6). IEEE.

Smith, G., Martin, A., and Wenger, D. E. (2018). Disaster recovery in an era of climate change: the unrealized promise of institutional resilience. In *Handbook of Disaster Research* (pp. 595-619). Springer, Cham.

Sobh, T. S. (2013). Wi-fi networks security and accessing control. *International Journal of Computer Network and Information Security*, 5(7), 9.

Stapleton, J. (1997). *Dsdm, dynamic systems development method: The method in practice*: Cambridge University Press.

Stergiou, C., Psannis, K. E., Kim, B.-G., and Gupta, B. (2018). Secure integration of IoT and cloud computing. *Future Generation Computer Systems*, 78, 964-975.

Stytz, M. R., and Banks, S. B. (2014). Toward attaining cyber dominance. *Strategic Studies Quarterly*, 8(1), 55-87.

- Subashini, S., and Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1), 1-11.
- Tamura, Y., Sato, K., Kihara, S., and Moriai, S. (2008, June). Kemari: Virtual machine synchronization for fault tolerance. In *Proc. USENIX Annu. Tech. Conf. (Poster Session)*.
- Thomas, J. (2018). Individual cyber security: Empowering employees to resist spear phishing to prevent identity theft and ransomware attacks. *Thomas, JE (2018). Individual cyber security: Empowering employees to resist spear phishing to prevent identity theft and ransomware attacks. International Journal of Business Management*, 12(3), 1-23.
- Tow, W. T., Thakur, R., and Hyun, I.-T. (2000). *Asia's emerging regional order: Reconciling traditional and human security*: United Nations University Press.
- Ueno, Y., Miyaho, N., Suzuki, S., and Ichihara, K. (2010, August). Performance evaluation of a disaster recovery system and practical network system applications. In *2010 Fifth International Conference on Systems and Networks Communications* (pp. 195-200). IEEE.
- Usman, K., Ge, T., Karim, R., and Agber, S. (2014). *A practical guide to computer programming with basic* (Second ed. ISBN. 978-978-50713-7-5). Makurdi: Shekinah Publisher.
- Wade, H., Hylender, C., and Valentine, A. (2008). Verizon business 2008 data breach investigation report. In: Verizon Enterprises.
- Wallace, M., and Webber, L. (2017). *The disaster recovery handbook: A step-by-step plan to ensure business continuity and protect vital operations, facilities, and assets*: Amacom.

- Westerlund, M., and Kratzke, N. (2018, July). Towards Distributed Clouds: A Review About the Evolution of Centralized Cloud Computing, Distributed Ledger Technologies, and A Foresight on Unifying Opportunities and Security Implications. In *2018 International Conference on High Performance Computing and Simulation (HPCS)* (pp. 655-663). IEEE.
- Wood, T., Cecchet, E., Ramakrishnan, K. K., Shenoy, P. J., van der Merwe, J. E., and Venkataramani, A. (2010). Disaster recovery as a cloud service: Economic benefits and deployment challenges. *HotCloud, 10*, 8-15.
- Wood, T., Lagar-Cavilla, H. A., Ramakrishnan, K. K., Shenoy, P., and Van der Merwe, J. (2011, October). PipeCloud: using causality to overcome speed-of-light delays in cloud-based disaster recovery. In *Proceedings of the 2nd ACM Symposium on Cloud Computing* (p. 17). ACM.
- Zavarsky, P., and Lindskog, D. (2016). Experimental analysis of ransomware on windows and android platforms : Evolution and characterization *Evolution and Characterization, 94*, 465 - 472.
- Zhao, Q., Amagasaki, M., Iida, M., Kuga, M., and Sueyoshi, T. (2017). A study of heterogeneous computing design method based on virtualization technology. *ACM SIGARCH Computer Architecture News, 44*(4), 86-91.
- Zhu, J., Jiang, Z., Xiao, Z., and Li, X. (2011). Optimizing the performance of virtual machine synchronization for fault tolerance. *IEEE Transactions on Computers, 60*(12), 1718-1729.
- Zou, H., and Jahanian, F. (1999). A real-time primary-backup replication service. *IEEE Transactions on Parallel and Distributed Systems, 10*(6), 533-548.

APPENDIX A PROGRAM LISTINGS

```
package project.sessions;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import project.entities.*;
import project.util.Settings;
@Stateless
public class MainSession implements MainSessionLocal {
    @PersistenceContext(unitName = "DatabackupPU")
    private EntityManager em;
    Settings settings = new Settings();
    public void persist(Object object) {
        em.persist(object);
    }

    @Override
    public Application getApplication(String id) {
        return em.find(Application.class, id);
    }

    @Override
    public Backupconfiguration getBackupconfiguration(String id) {
        return em.find(Backupconfiguration.class, id);
    }

    @Override
    public Backuphistory getBackuphistory(String id) {
        return em.find(Backuphistory.class, id);
    }
}
```

```

@Override
public Subscriptionhistory getSubscriptionhistory(String id) {
    return em.find(Subscriptionhistory.class, id);
}

@Override
public Usersdata getUsersdata(String id) {
    return em.find(Usersdata.class, id);
}

@Override
public Usersdata getUsersdataByEmail(String email) {
    Usersdata us = null;
    try {
        List<Usersdata> usli = (List<Usersdata>) em.createNamedQuery(
            "Usersdata.findByEmail").setParameter("email",email).getResultList(
        );
        for (Usersdata usd : usli) {
            us = usd;
            break;
        }
    } catch (Exception j) {
    }
    return us;
}

@Override
public Userpassport getUserpassport(String id) {
    return em.find(Userpassport.class, id);
}

@Override
public void newApplication(Application param) {
    em.persist(param);
}

```

```

@Override
public void editApplication(Application param) {
    try {
        Application app = em.find(Application.class,
            param.getApplicationid());
        if (app != null) {
            em.merge(param);
        }
    } catch (Exception js) {
    }
}

@Override
public void newBackupconfiguration(Backupconfiguration param) {
    em.persist(param);
}

@Override
public void newBackuphistory(Backuphistory param) {
    em.persist(param);
}

@Override
public void newSubscriptionhistory(Subscriptionhistory param) {
    em.persist(param);
}

@Override
public void newUsersdata(Usersdata param) {
    em.persist(param);
}

@Override
public void newUserpassport(Userpassport param) {
    em.persist(param);
}

@Override

```

```

public List<Application> getAllApplication() {
    return (List<Application>)
        em.createNamedQuery("Application.findAll").getResultList();
}

@Override
public List<Application> getApplicationByUser(String userid) {
    return (List<Application>)
        em.createNamedQuery("Application.findByUserId").setParameter("userid",
            userid).getResultList();
}

@Override
public List<Backupconfiguration> getAllBackupconfiguration() {
    return (List<Backupconfiguration>)
        em.createNamedQuery("Backupconfiguration.findAll").getResultList();
}

@Override
public List<Backupconfiguration> getBackupconfigurationByApp
    (String username) {
    return (List<Backupconfiguration>)
        em.createNamedQuery("Backupconfiguration.findByApplicationid").setParameter("a
            pplicationid", username).getResultList();
}

@Override
public List<Backuphistory> getAllBackuphistory() {
    return (List<Backuphistory>)
        em.createNamedQuery("Backuphistory.findAll").getResultList();
}

@Override
public List<Subscriptionhistory> getAllSubscriptionhistory() {

```

```

        return (List<Subscriptionhistory>)
em.createNamedQuery("Subscriptionhistory.findAll").getResultList();
    }
    @Override
    public List<Usersdata> getAllUsersdata() {
        return (List<Usersdata>) em.createNamedQuery("User.findAll").getResultList();
    }
    @Override
    public List<Userpassport> getAllUserpassport() {
        return (List<Userpassport>)
em.createNamedQuery("Userpassport.findAll").getResultList();
    }

    @Override
    public void removeApplication(String id) {
        Application idx = em.find(Application.class, id);
        if (idx != null) {
            em.remove(idx);
        }
    }
    @Override
    public void removeBackupconfiguration(String id) {
        Backupconfiguration idx = em.find(Backupconfiguration.class, id);
        if (idx != null) {
            em.remove(idx);
        }
    }
    @Override
    public void removeBackuphistory(String id) {
        Backuphistory idx = em.find(Backuphistory.class, id);
        if (idx != null) {
            em.remove(idx);
        }
    }

```

```

}
@Override
public void removeSubscriptionhistory(String id) {
    Subscriptionhistory idx = em.find(Subscriptionhistory.class, id);
    if (idx != null) {
        em.remove(idx);
    }
}
@Override
public void removeUsersdata(String id) {
    Subscriptionhistory idx = em.find(Subscriptionhistory.class, id);
    if (idx != null) {
        em.remove(idx);
    }
}

@Override
public void removeUserpassport(String id) {
    Subscriptionhistory idx = em.find(Subscriptionhistory.class, id);
    if (idx != null) {
        em.remove(idx);
    }
}

@Override
public Usersdata login(String username, String password) {
    Usersdata us = null;
    try {
        if (username.equals(settings.adminUsername) andand
password.equals(settings.adminPassword)) {
            us = new Usersdata(username, "ADMIN", "ADMIN", "ADMIN", "", "", "",
password, "", "", new Double(0));
        } else {

```



```

        Usersdata uss = this.getUsersdata(username);
        if (uss != null) {
            if (uss.getPassword().equals(password)) {
                us = uss;
            }
        }
    } catch (Exception js) {
    }
    return us;
}

```

@Override

```

public Application checkDuplicateApplication(String url, String dbname) {
    Application app = null;
    try {
        List<Application> li = (List<Application>) em.createQuery("SELECT a
FROM Application a WHERE a.url = :url AND a.databasename= :dbname")
            .setParameter("url", url).setParameter("dbname",
dbname).getResultList();
        for (Application ap : li) {
            app = ap;
            break;
        }
    } catch (Exception js) {
    }
    return app;
}

```

@Override

```

public Backupconfiguration checkDuplicateBackupconfiguration(String
entityname) {
    Backupconfiguration app = null;

```

```

try {
    List<Backupconfiguration> li = (List <Backupconfiguration>)
    em.createQuery("SELECT a FROM Backupconfiguration a WHERE
    a.entityname = :entityname").setParameter("entityname",
    entityname).getResultList();
    for (Backupconfiguration ap : li)
        {
            app = ap;
            break;
        }
    } catch (Exception js) {
    }
    return app;
}
@Override
public List<String> getDatabaseTables(String dbtype, String domain, String
databasename, String portno, String username, String password) {
    List<String> tables = new ArrayList();
    try {
        Class.forName(settings.getDriver(dbtype));
    } catch (ClassNotFoundException e) {
        System.out.println("Unable to load driver class");
    }
    System.out.println("ssss "+settings.getDriver(dbtype));
    ResultSet rs = null;

    try {
        Connection con =
        DriverManager.getConnection(settings.getConnectionUrl(dbtype, domain, portno,
databasename), username, password);
        DatabaseMetaData dmd = con.getMetaData();
        rs = dmd.getTables(null, null, null, null);
        ResultSetMetaData rsmd = rs.getMetaData();

```

```

        int numCols = rsmd.getColumnCount();
        while (rs.next()) {
            for (int i = 3; i <= numCols; i+=3) {
                tables.add(rs.getString(i));
            }
        }
    } catch (Exception j) { }
    return tables;
}
}

```

```

@Local
public interface MainSessionLocal {
    Application getApplication(String id)
    Backupconfiguration getBackupconfiguration(String id);
    Backuphistory getBackuphistory(String id);
    Subscriptionhistory getSubscriptionhistory(String id);
    Usersdata getUsersdata(String id);
    Userpassport getUserpassport(String id);
    void newApplication(Application param);
    void newBackupconfiguration(Backupconfiguration param);
    void newBackuphistory(Backuphistory param);
    void newSubscriptionhistory(Subscriptionhistory param);
    void newUsersdata(Usersdata param);
    void newUserpassport(Userpassport param);
    List<Application> getAllApplication();
    List<Backupconfiguration> getAllBackupconfiguration();
    List<Backuphistory> getAllBackuphistory();
    List<Subscriptionhistory> getAllSubscriptionhistory();
    List<Usersdata> getAllUsersdata();
    List<Userpassport> getAllUserpassport();
}

```

```

void removeApplication(String id);
void removeBackupconfiguration(String id);
void removeBackuphistory(String id);
void removeSubscriptionhistory(String id);
void removeUsersdata(String id);
void removeUserpassport(String id);
public Usersdata login(String username, String password);
public Usersdata getUsersdataByEmail(String email);
public List<Backupconfiguration> getBackupconfigurationByApp(String username);
public List<Application> getApplicationByUser(String userid);
public Application checkDuplicateApplication(String url, String dbname);
public void editApplication(Application param);
public Backupconfiguration checkDuplicateBackupconfiguration(String entityname);
public List<String> getDatabaseTables(String dbtype, String domain, String
databasename, String portno, String username, String password);
}

```

Captcha Servlet

```

package servlets;
import java.awt.Color;
import java.awt.Font;
import java.awt.GradientPaint;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.image.BufferedImage;
import java.io.*;
import java.net.*;

import java.util.Random;
import javax.imageio.ImageIO;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

```

```

@WebServlet(name = "CaptchaServlet", urlPatterns = {"/CaptchaServlet"})
public class CaptchaServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
                                HttpServletResponse response)
        throws ServletException, IOException {

        int width = 200;
        int height = 50;

//NNAMDI AZIKIWE Inyiamama Okonkwo ᠠᠨᠢᠭᠪᠠᠭᠠ Anigbogu Ejiofor Boniface

        char data[][] = {
            { 'T', 'n', 'y', 'i', 'a', 'm', 'a', },
            { 'O', 'k', 'o', 'n', 'k', 'w', 'o', },
            { 'K', 'a', 'r', 'i', 'm', },
            { 'U', 's', 'm', 'a', 'n', },
            { 'A', 'n', 'i', 'g', 'b', 'o', 'g', 'u', },
            { 'E', 'j', 'i', 'o', 'f', 'o', 'r', },
            { 'N', 'n', 'a', 'm', 'd', 'i', },
            { 'A', 'z', 'i', 'k', 'i', 'w', 'e', },
            { 'O', 'i', 'z', 'a' },
            { 'B', 'o', 'n', 'i', 'f', 'a', 'c', 'e', },
            { 'S', 'o', 'l', 'u', 't', 'i', 'o', 'n', },
            { 'K', 'r', 'e', 'a', 't', 'i', 'v', 'e' },
            { 'D', 'i', 's', 'a', 's', 't', 'e', 'r', },
            { 'R', 'e', 'c', 'o', 'v', 'e', 'r', 'y', },
            { 'O', 'n', 'i', 'z', 'e', }
        };

        BufferedImage bufferedImage = new BufferedImage(width, height,
            BufferedImage.TYPE_INT_RGB);

```

```

Graphics2D g2d = bufferedImage.createGraphics();

Font font = new Font("Georgia", Font.BOLD, 18);
g2d.setFont(font);

RenderingHints rh = new RenderingHints(
    RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);

rh.put(RenderingHints.KEY_RENDERING,
    RenderingHints.VALUE_RENDER_QUALITY);

g2d.setRenderingHints(rh);

GradientPaint gp = new GradientPaint(0, 0,
    Color.red, 0, height/2, Color.black, true);

g2d.setPaint(gp);
g2d.fillRect(0, 0, width, height);

g2d.setColor(new Color(255, 153, 0));

Random r = new Random();
int index = Math.abs(r.nextInt()) % 15; //Chk here

String captcha = String.valueOf(data[index]);
request.getSession().setAttribute("captcha", captcha );

int x = 0;
int y = 0;

for (int i=0; i<data[index].length; i++) {

```

```
x += 10 + (Math.abs(r.nextInt()) % 15);
y = 20 + Math.abs(r.nextInt()) % 20;
g2d.drawChars(data[index], i, 1, x, y);
}
```

```
g2d.dispose();
```

```
response.setContentType("image/png");
OutputStream os = response.getOutputStream();
ImageIO.write(bufferedImage, "png", os);
os.close();
}
```

```
protected void doGet(HttpServletRequest request,
                    HttpServletResponse response)
                    throws ServletException, IOException {
    processRequest(request, response);
}
```

```
protected void doPost(HttpServletRequest request,
                    HttpServletResponse response)
                    throws ServletException, IOException {
    processRequest(request, response);
}
}
```

DownloadBackups Servlet

```
package servlets;
import java.io.BufferedOutputStream;
import java.io.FileInputStream;
```

```

import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.net.URLEncoder;
import javax.mail.internet.MimeUtility;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name = "DownloadBackups", urlPatterns = {"/DownloadBackups"})
public class DownloadBackups extends HttpServlet {
    String url = null;
    /**
     * Processes requests for both HTTP GET and
     POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
        url = request.getParameter("url");
        InputStream in =new FileInputStream (url);

        String filename = url.substring(url.lastIndexOf("\\"),url.length());
        String agent = request.getHeader("USER-AGENT");
        if (agent != null andand agent.indexOf("MSIE") != -1)

```



```

    {
        filename = URLEncoder.encode(filename, "UTF8");
        response.setContentType("application/x-download");
        response.setHeader("Content-Disposition","attachment;filename=" +
filename);
    }
    else if ( agent != null andand agent.indexOf("Mozilla") != -1)
    {
        response.setCharacterEncoding("UTF-8");
        filename = MimeUtility.encodeText(filename, "UTF8", "B");
        response.setContentType("application/force-download");
        response.addHeader("Content-Disposition", "attachment; filename=\"\" +
filename + "\"");
    }

```

```

        BufferedOutputStream out = new
BufferedOutputStream(response.getOutputStream());
        byte by[] = new byte[32768];
        int index = in.read(by, 0, 32768);
        while (index != -1) {
            out.write(by, 0, index);
            index = in.read(by, 0, 32768);
        }
        out.flush();

```

```

}

```

```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">

```

```

/**

```

```

    * Handles the HTTP <code>GET</code> method.

```

```

*
* @param request servlet request
* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {

```

```
        return "Short description";
    } // </editor-fold>

}
```

User Registration Page

```
<% @page contentType="text/html" pageEncoding="UTF-8"% >
<% @include file="WEB-INF/jspf/initialize.jspf" %>
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title><%= settings.schoolName%></title>
    <meta name="description" content="<%= settings.schoolName%>">
    <meta name="keywords" content="<%= settings.keywords%>">
    <meta name="author" content="<%= settings.schoolName%>">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="css2/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="css/settings.css" media="screen" />
    <link href="css/prettyPhoto.css" rel="stylesheet">
    <link href="css/flexslider.css" rel="stylesheet">
    <link rel="stylesheet" href="css/slider.css">
    <link rel="stylesheet" href="css/owl.carousel.css">
```

```
<link href="css2/font-awesome/css/font-awesome.min.css" rel="stylesheet">
```

```
<link href="css/style.css" rel="stylesheet">
```

```
<link href="css2/style.css" rel="stylesheet">
```

```
<link href="css/green.css" rel="stylesheet">
```

```
<link rel="shortcut icon" href="img/favicon/favicon.png">
```

```
<!-- custom style for form -->
```

```
<style type="text/css">
```

```
body {
```

```
padding: 0;
```

```
}
```

```
.apply-section {
```

```
background-image: linear-gradient(rgba(0, 0, 0, 0.6), rgba(0, 0, 0, 0.6)),  
url('img2/portfolio_item_8.jpg');
```

```
height: 100%;
```

```
background-size: cover;
```

```
background-position: center;
```

```
background-repeat: no-repeat;
```

```
}
```

```
.apply-page {  
  
    margin: auto;  
}  
  
.form-div {  
    position: relative;  
    z-index: 1;  
    background: rgba(255, 255, 255, 0.4);  
    width: 50%;  
    margin: 0 auto 100px;  
    padding: 45px;  
    text-align: center;  
}  
</style>  
  
</head>  
  
<%    String errorMsg = request.getParameter("errorMsg");  
  
    String msg = "";  
    String sms = "";
```

```

String uname = request.getParameter("username");

String password = request.getParameter("password");

String password2 = request.getParameter("password2");

String firstname = request.getParameter("firstname");

String secretKey = "";

String lastname = request.getParameter("lastname");

String sex = request.getParameter("sex");

String email = request.getParameter("email");

String phone = request.getParameter("phone");

String dob = request.getParameter("dob");

String but = request.getParameter("submit");

String captcha = (String) session.getAttribute("captcha");

String code = (String) request.getParameter("code");

if (errorMsg != null) {

    msg = "<div class=\"alert alert-warning\">" + errorMsg + "</div>";

}

if (uname != null andand uname.trim().length() > 0 andand
password.trim().length() > 0 andand password2 != null andand email != null) {

    try {

        uname = uname.trim().toLowerCase();

        email = email.toLowerCase();

        password = settings.getMD5(password);

        password2 = settings.getMD5(password2);

```

```

secretKey = settings.getDecryptionCode();

Usersdata com = mainLocal.getUsersdata(uname);

if (com != null || uname.equalsIgnoreCase("admin")) {

    msg = "<div class=\"alert alert-warning\">Username " + uname + "
already taken.</div>";

}

else if (!password.equals(password2)) {

    msg = "<div class=\"alert alert-warning\">Password does not
match.</div>";

}

if (!captcha.equals(code)) {

    msg = "<div class=\"alert alert-warning\">Captcha does not
match.</div>";

}

else {

    Usersdata com2 = mainLocal.getUsersdataByEmail(email);

    if (com2 != null) {

        msg = "<div class=\"alert alert-warning\">Email " + email + " already
registered, if you have forgotten your password <a
href=\"recoverpassword.jsp\">Recover here.</a></div>";

    }

    else {

        try {

```

```

        Usersdata ud = new Usersdata(uname, firstname, secretKey,
lastname, sex, email, phone, password, dob, settings.getTodaysdate(), new
Double(5000));

        mainLocal.newUsersdata(ud);

        session.setAttribute("UNAME", uname);

        session.setAttribute("ROLE", "USER");

        sms= "Your Account has been created successfully. Your Secret
Decryption Key is "+secretKey+ ". It has been sent to your phone number "+phone+".
Do not share with anybody for security reason";

        //send email

        SmsAlertHelper sendSMS = new SmsAlertHelper(sms, phone);

        msg = "<div class=\"alert alert-success\">"+sms+"</div>";

    } catch (Exception j) {

    }

    //response.sendRedirect("userDefault.jsp");

}

}

} catch (Exception js) {

}

}

%>

<body>

<header>

<% @include file="WEB-INF/jspf/publicheader.jspf" %>

```



```
<%  
    if (roles.equals("ADMIN")) {  
%>  
<% @include file="WEB-INF/jspf/adminmenu.jspf" %>  
<%  
    } else if (roles.equals("USER")) {  
%>  
<% @include file="WEB-INF/jspf/usermenu.jspf" %>  
<%  
    } else {  
%>  
<% @include file="WEB-INF/jspf/publicmenu.jspf" %>  
<%  
    }  
%>  
  
</header>  
  
<!-- Page Heading ends -->  
  
<!-- Page content starts -->  
  
<div class="content">
```

```

<div class="container">
  <div class="row">

    <!-- Login form -->
    <div class="col-md-12">
      <div class="formy well">
        <!-- Title -->

        <!-- new form -->
        <section class="apply-section pt-5">

          <h1 class="text-white p-3 my-3 text-center text-bold">Users
Registration</h1>

```

```

<div class="container">
  <div class="row justify-content-around">

    <div class="col-md-12">
      <div class="apply-page">
        <div class="form-div">

        <%
          if (msg.length() > 0) {

```

```
%>
<%= msg%>
<%
}
%>
```

```
<form action="#" method="POST" class="apply-form" name="applyForm"
id="apply">
<!-- first name -->
<div class="form-row mb-4">
    <div class="form-group col-md-6">
        <div class="input-group">
            <div class="input-group-prepend">
                <span class="input-group-text">
                    <i class="fa fa-user"></i>
                </span>
            </div>
            <input type="text" required=""
name="firstname" class="form-control" id="firstname" placeholder="First Name">
        </div>
    </div>
    <!-- last name -->
    <div class="form-group col-md-6">
        <div class="input-group">
            <div class="input-group-prepend">
```

```
        <span class="input-group-text">
            <i class="fa fa-user"></i>
        </span>
    </div>

    <input type="text" name="lastname"
class="form-control" id="lastname" placeholder="Last Name">
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- gender -->
```

```
<div class="form-row">
```

```
    <div class="form-group col-md-6">
```

```
        <div class="input-group">
```

```
            <div class="input-group-prepend">
```

```
                <span class="input-group-text">
```

```
                    <strong>Gender</strong>
```

```
                </span>
```

```
            </div>
```

```
            <select class="form-control" name="sex" id="">
```

```
                <option value="Male">Male</option>
```

```
                <option value="Female">Female</option>
```

```

        </select>

    </div>

</div>

<!-- date of birth -->

<div class="form-group col-md-6">
    <div class="input-group">
        <div class="input-group-prepend">
            <span class="input-group-text">
                <i class="fa fa-calendar"></i>
            </span>
        </div>
        <input type="date" required="" name="dob"
class="form-control" id="dob" >
    </div>
</div>
</div>

```

```

<!-- EMAIL -->

```

```

<div class="form-row mb-2">
    <div class="form-group col-md-6">
        <div class="input-group">
            <div class="input-group-prepend">

```

```
        <span class="input-group-text">
            <i class="fa fa-envelope"></i>
        </span>
    </div>

    <input type="email" required=""
name="email" class="form-control" id="email" placeholder="Email"
Pattern="<%=settings.EMAIL_PATTERN%>" title="Invalid email">

</div>
</div>
```

```
<!-- PHONE-->
```

```
<div class="form-group col-md-6">
    <div class="input-group">
        <div class="input-group-prepend">
            <span class="input-group-text">
                <i class="fa fa-phone"></i>
            </span>
        </div>
        <input type="text" required=""
name="phone" class="form-control" id="phone" pattern="[0-9]{11}" title="Invalid
phone number" required>
    </div>
</div>

</div>
```

```
<!-- username -->
<div class="form-row mb-4">
    <div class="form-group col-md-12">
        <div class="input-group">
            <div class="input-group-prepend">
                <span class="input-group-text">
                    <strong>Username</strong>
                </span>
            </div>
            <input type="text" class="form-control"
id="username" required="" name="username" placeholder="Choose a Username">
        </div>
    </div>
</div>
```

```
<!-- password -->
<div class="form-row mb-2">
    <div class="form-group col-md-6">
        <div class="input-group">
            <div class="input-group-prepend">
                <span class="input-group-text">
                    <i class="fa fa-lock"></i>
                </span>
            </div>
        </div>
    </div>
```

```

        <input type="password" required=""
name="password" class="form-control" id="password" placeholder="Password">
    </div>

    </div>

    <!--
confirm password -->

    <div class="form-group col-md-6">
        <div class="input-group">
            <div class="input-group-prepend">
                <span class="input-group-text">
                    <i class="fa fa-lock"></i>
                </span>
            </div>
            <input type="password" required=""
name="password2" class="form-control" id="password2" placeholder="Password">
        </div>
    </div>

    </div>

    </div>

    <!-- Captcha Code -->

    <div class="form-row mb-2">
        <div class="form-group col-md-6">
            <div class="input-group">
                <div class="input-group-prepend">
                    <span class="input-group-text">
                        <strong>Captcha Code</strong>
                    </span>
                </div>
            </div>
        </div>
    </div>

```



```

        </span>
    </div>
    

</div>
</div>
<!--
confirm password -->

<div class="form-group col-md-6">
    <div class="input-group">
        <div class="input-group-prepend">
            <span class="input-group-text">
                <strong>Verify Captcha Code</strong>
            </span>
        </div>
        <input type="text" required="" name="code"
class="form-control" id="code" placeholder="Are you human?">
    </div>
</div>
</div>

    <button type="submit" name="submit" class="btn btn-primary text-
white">Register</button>
</form>

</div>

```

</div>

</div>

</div>

</div>

</section>

</div>

</div>

</div>

</div>

</div>

<%@include file="WEB-INF/jspf/publicfooter.jspf" %>

<!-- Javascript files -->

<script src="js/jquery.js"></script>

<script src="js2/bootstrap.min.js"></script>

<script src="js/jquery.themepunch.tools.min.js"></script>

<script src="js/jquery.themepunch.revolution.min.js"></script>

<script src="js/jquery.isotope.js"></script>

<script src="js/jquery.prettyPhoto.js"></script>

<script src="js/filter.js"></script>

<script src="js/jquery.flexslider-min.js"></script>

<script src="js/jquery.cslider.js"></script>

```

<script src="js/modernizr.custom.28468.js"></script>

<script src="js/owl.carousel.min.js"></script>

<script src="js/respond.min.js"></script>

<script src="js/html5shiv.js"></script>

<script src="js/custom.js"></script>

<script type="text/javascript" src="js/jquery.vticker.js"></script>

<script type="text/javascript">

    $(function () {

        $('#news-container').vTicker({

            speed: 500,

            pause: 5000,

            animation: 'fade',

            mousePause: true,

            showItems: 3

        });

    });

</script>

<script>

    // Revolution Slider

    var revapi;

    jQuery(document).ready(function () {

        revapi = jQuery('.tp-banner').revolution(

```

```
    {
      delay: 9000,
      startwidth: 1170,
      startheight: 450,
      hideThumbs: 200,
      shadow: 0,
      navigationType: "none",
      hideThumbsOnMobile: "on",
      hideArrowsOnMobile: "on",
      hideThumbsUnderResolution: 0,
      touchenabled: "on",
      fullWidth: "on"
    });
  });

</script>
</body>
</html>
```

Application Registration Page

<%--

Document : index

Created on : Jan 8, 2017, 12:48:48 PM

Author : doc

--%>

<% @page contentType="text/html" pageEncoding="UTF-8"% >

<% @include file="WEB-INF/jspf/initialize.jspf" % >

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title><%= settings.schoolName% ></title>

<meta name="description" content="<%= settings.schoolName% >">

<meta name="keywords" content="<%= settings.keywords% >">

<meta name="author" content="<%= settings.schoolName% >">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="css2/bootstrap.min.css" rel="stylesheet">

<link href="css2/font-awesome/css/font-awesome.min.css" rel="stylesheet">

<link href="css2/style.css" rel="stylesheet">

<link rel="stylesheet" href="css/settings.css" media="screen" />

<link href="css/prettyPhoto.css" rel="stylesheet">

<link href="css/flexslider.css" rel="stylesheet">

<link rel="stylesheet" href="css/slider.css">

<link rel="stylesheet" href="css/owl.carousel.css">

```
<link href="css/font-awesome.min.css" rel="stylesheet">
```

```
<link href="css/style.css" rel="stylesheet">
```

```
<link href="css/green.css" rel="stylesheet">
```

```
<link rel="shortcut icon" href="img/favicon/favicon.png">
```

```
<style type="text/css">
```

```
body {
```

```
padding: 0;
```

```
}
```

```
.apply-section {
```

```
background-image: linear-gradient(rgba(0, 0, 0, 0.6), rgba(0, 0, 0, 0.6)),  
url('img2/portfolio_item_8.jpg');
```

```
height: 100%;
```

```
background-size: cover;
```

```
background-position: center;
```

```
background-repeat: no-repeat;
```

```
}
```

```
.apply-page {
```

```
margin: auto;
```

```

}

.form-div {
    position: relative;
    z-index: 1;
    background: rgba(255, 255, 255, 0.4);
    width: 50%;
    margin: 0 auto 100px;
    padding: 45px;
    text-align: center;
}
</style>
</head>
<%
    String id = null;
    Usersdata app = null;
    try {
        id = (String) session.getAttribute("USERID");
    } catch (Exception j) {
    }
    if (id != null) {
        app = mainLocal.getUsersdata(id);
        if (app != null) {

```

```

    } else {
        response.sendRedirect("managedatabases.jsp");
    }
} else {
    response.sendRedirect("managedatabases.jsp");
}
%>

<%
    String msg = "";
    String dbtype = request.getParameter("dbtype");
    String dburl = request.getParameter("dburl");
    String dbname = request.getParameter("dbname");
    String portno = request.getParameter("portno");
    String dbusername = request.getParameter("dbusername");
    String pword = request.getParameter("password");
    String but = request.getParameter("submit");

    if (dburl != null andand dbname.trim().length() > 0) {
        try {
            dburl = dburl.trim().toLowerCase();
            Application ap = mainLocal.checkDuplicateApplication(dburl, dbname);
            if (ap != null) {

```



```
        msg = "<div class=\"alert alert-warning\">This App is already added. <a  
href=\"managedatabases.jsp\">Back to Apps</a></div>";
```

```
    } else {
```

```
        try {
```

```
            Application appx = new Application(settings.generateId(), id, dburl,  
dbname, pword, dbtype, portno, dbusername);
```

```
            mainLocal.newApplication(appx);
```

```
        } catch (Exception j) {
```

```
        }
```

```
response.sendRedirect("managedatabases.jsp");
```

```
    }
```

```
    } catch (Exception js) {
```

```
    }
```

```
    }
```

```
%>
```

```
<body>
```

```
    <header>
```

```
        <% @include file="WEB-INF/jspf/publicheader.jspf" %>
```

```
        <%
```

```
            if (roles.equals("ADMIN")) {
```

```
        %>
```

```
        <% @include file="WEB-INF/jspf/adminmenu.jspf" %>
```

```
        <%
```

```
            } else if (roles.equals("USER")) {
```

```

%>

<%@include file="WEB-INF/jspf/usermenu.jspf" %>

<%

} else {

%>

<%@include file="WEB-INF/jspf/publicmenu.jspf" %>

<%

}

%>

</header>

<div class="headsep"></div>

<div class="page-head">

<div class="container">

<div class="row">

<div class="col-md-12">

<h2>Add Applications to your Account</h2>

</div>

</div>

</div>

</div>

</div>

<!-- Page Heading ends -->

```

```
<!-- Page content starts -->
```

```
<div class="container">
```

```
<section class="apply-section pt-5">
```

```
<div class="container">
```

```
<div class="row justify-content-around">
```

```
<div class="col-md-12">
```

```
<div class="apply-page">
```

```
<div class="form-div">
```

```
<%
```

```
if (msg.length() > 0) {
```

```
%>
```

```
<%= msg%>
```

```
<%
```

```
}
```

```
%>
```

```
<form class="apply-form" role="form" action=""  
method="post">
```

```
<div class="form-row mb-4">
```

```
<div class="form-group col-md-12">
```

```
<div class="input-group">
```

```

        <div class="input-group-prepend">
            <span class="input-group-text">
                <strong>Database Type</strong>
            </span>
        </div>

        <select class="form-control" id="dbtype"
name="dbtype">

            <%
                try {
                    List<String> st =
settings.getDatabaseTypes();

                    for (String sts : st) {
                        %>
                        <option><%=sts%></option>
                    }
                } catch (Exception j) {
                }
            %>
        </select>

    </div>

</div>

```

```

<div class="form-row mb-4">
    <div class="form-group col-md-12">
        <div class="input-group">
            <div class="input-group-prepend">
                <span class="input-group-text">
                    <strong>Database URL</strong>
                </span>
            </div>
            <input type="text" class="form-control"
id="dburl" required="" name="dburl" placeholder="URL">
        </div>
    </div>
</div>
</div>
<div class="form-row mb-4">
    <div class="form-group col-md-12">
        <div class="input-group">
            <div class="input-group-prepend">
                <span class="input-group-text">
                    <strong>Database Name</strong>
                </span>
            </div>
            <input type="text" class="form-control"
id="dbname" required="" name="dbname" placeholder="Databasename">
        </div>
    </div>
</div>

```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="form-row mb-4">
```

```
<div class="form-group col-md-12">
```

```
<div class="input-group">
```

```
<div class="input-group-prepend">
```

```
<span class="input-group-text">
```

```
<strong>Database Username</strong>
```

```
</span>
```

```
</div>
```

```
<input type="text" class="form-control"
```

```
id="dbusername" name="dbusername" placeholder="Username">
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="form-row mb-4">
```

```
<div class="form-group col-md-12">
```

```
<div class="input-group">
```

```
<div class="input-group-prepend">
```

```
<span class="input-group-text">
```

```
<strong>Database Password</strong>
```

```

        </span>
    </div>

    <input type="text" class="form-control"
id="password" name="password" placeholder="Password">

        </div>
    </div>
</div>

<div class="form-row mb-4">

    <div class="form-group col-md-12">
        <div class="input-group">
            <div class="input-group-prepend">
                <span class="input-group-text">
                    <strong>Port Number</strong>
                </span>
            </div>
            <input type="number" step="1" min="1024"
max="9999" required="" name="portno" class="form-control" id="portno"
placeholder="Port">

                </div>
            </div>
        </div>
    </div>

```

```
<button type="submit" name="submit" class="btn
btn-primary text-white">Add</button>
```

```
<!-- first name -->
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
</div>
```

```
<%@include file="WEB-INF/jspf/publicfooter.jspf" %>
```

```
<!-- Javascript files -->
```

```
<script src="js/jquery.js"></script>
```

```
<script src="js2/bootstrap.min.js"></script>
```

```
<script src="js/jquery.themepunch.tools.min.js"></script>
```

```
<script src="js/jquery.themepunch.revolution.min.js"></script>
```

```
<script src="js/jquery.isotope.js"></script>
```

```
<script src="js/jquery.prettyPhoto.js"></script>
```



```

<script src="js/filter.js"></script>

<script src="js/jquery.flexslider-min.js"></script>

<script src="js/jquery.cslider.js"></script>

<script src="js/modernizr.custom.28468.js"></script>

<script src="js/owl.carousel.min.js"></script>

<script src="js/respond.min.js"></script>

<script src="js/html5shiv.js"></script>

<script src="js/custom.js"></script>

<script type="text/javascript" src="js/jquery.vticker.js"></script>

<script type="text/javascript">

    $(function () {

        $('#news-container').vTicker({

            speed: 500,

            pause: 5000,

            animation: 'fade',

            mousePause: true,

            showItems: 3

        });

    });

</script>

<script>

    // Revolution Slider

```

```

var revapi;

jQuery(document).ready(function () {

    revapi = jQuery('.tp-banner').revolution(

        {

            delay: 9000,

            startwidth: 1170,

            startheight: 450,

            hideThumbs: 200,

            shadow: 0,

            navigationType: "none",

            hideThumbsOnMobile: "on",

            hideArrowsOnMobile: "on",

            hideThumbsUnderResolution: 0,

            touchenabled: "on",

            fullWidth: "on"

        });

    });

</script>

</body>

</html>

```

Settings Class

```

package util;

import java.awt.Color;

import java.awt.Image;

```

```
import java.awt.image.BufferedImage;

import java.io.ByteArrayInputStream;

import java.io.ByteArrayOutputStream;

import java.io.IOException;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import java.sql.Timestamp;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.GregorianCalendar;

import java.util.List;

import java.util.Random;

import java.util.StringTokenizer;

import java.util.UUID;

import javax.imageio.ImageIO;

public class Settings {

    public String lookuplink = "java:app/TSPRS-ejb/MainSession";

    public String schoolName = "";
```

```

public String schoolBox = "";

public String schoolMoto = "";

public String schoolAbbr = "UCCDRS";

public String keywords = "";

public String adminUsername = "admin";

public String adminPassword = "21232f297a57a5a743894ae4a801fc3";

public double ratePerKb = 0.1;

public String pay_item_id = "101";//101

public String product_id = "6207";//6207

public String currency_code = "566";

public String leadbank_code = "7";

public String leadbank_acc_no = "1006948866";

public String interswitch_url = "https://sandbox.interswitchng.com/webpay/pay";

public String interswitch_query_url =
"https://sandbox.interswitchng.com/webpay/api/v1/gettransaction.json";

public String mac_key =
"CEF793CBBE838AA0CBB29B74D571113B4EA6586D3BA77E7CFA0B95E2783
64EFC4526ED7BD255A366CDDE11F1F607F0F844B09D93B16F7CFE87563B227
2007AB3";

public java.text.NumberFormat formatNoComma = new
java.text.DecimalFormat("#####00");

public java.text.NumberFormat formater = new
java.text.DecimalFormat("###,###,###,###,###,###,###,##0.00");

public String project_url = "http://localhost:8180";

public String HOME_DIR = System.getProperty("user.home");

public String DIR_NAME = "RECOVER";

```

```

public String PATH_MAKER = "/";

public final String EMAIL_PATTERN

    = "^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@"
    + "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$";

public String getPincodes(int size) {

    String pin;

    Random ra = new Random();

    long l1 = ra.nextLong() % 100000000;

    if (l1 < 0) {

        l1 = (-1) * l1;

    }

    long l2 = ra.nextLong() % 100000000;

    if (l2 < 0) {

        l2 = (-1) * l2;

    }

    long l3 = ra.nextLong() % 100000000;

    if (l3 < 0) {

        l3 = (-1) * l3;

    }

    long l4 = ra.nextLong() % 100000000;

    if (l4 < 0) {

        l4 = (-1) * l4;

    }
}

```

```
String f = Long.toString(11) + Long.toString(12) + Long.toString(13) +  
Long.toString(14);
```

```
String cor = "1234567890987654321123456789";
```

```
pin = (f + cor).substring(0, size);
```

```
return pin;
```

```
}
```

```
public String getMonthName(String month1) {
```

```
    //TODO implement getMonthName
```

```
    String monthName = null;
```

```
    int month = new Integer(month1);
```

```
    switch (month) {
```

```
        case 1:
```

```
            monthName = "JANUARY";
```

```
            break;
```

```
        case 2:
```

```
            monthName = "FEBRUARY";
```

```
            break;
```

```
        case 3:
```

```
            monthName = "MARCH";
```

```
            break;
```

```
        case 4:
```

```
            monthName = "APRIL";
```

```
            break;
```

```
case 5:
    monthName = "MAY";
    break;
case 6:
    monthName = "JUNE";
    break;
case 7:
    monthName = "JULY";
    break;
case 8:
    monthName = "AUGUST";
    break;
case 9:
    monthName = "SEPTEMBER";
    break;
case 10:
    monthName = "OCTOBER";
    break;
case 11:
    monthName = "NOVEMBER";
    break;
case 12:
    monthName = "DECEMBER";
    break;
```

```
    }  
  
    return monthName;  
}  
  
public String getMonthNameShort(String month1) {  
  
    //TODO implement getMonthName  
  
    String monthName = null;  
  
    int month = new Integer(month1);  
  
    switch (month) {  
  
        case 1:  
  
            monthName = "JAN";  
  
            break;  
  
        case 2:  
  
            monthName = "FEB";  
  
            break;  
  
        case 3:  
  
            monthName = "MAR";  
  
            break;  
  
        case 4:  
  
            monthName = "APR";  
  
            break;  
  
        case 5:  
  
            monthName = "MAY";  
  
            break;  
  
        case 6:
```



```
        monthName = "JUN";
        break;
    case 7:
        monthName = "JUL";
        break;
    case 8:
        monthName = "AUG";
        break;
    case 9:
        monthName = "SEP";
        break;
    case 10:
        monthName = "OCT";
        break;
    case 11:
        monthName = "NOV";
        break;
    case 12:
        monthName = "DEC";
        break;
    }
    return monthName;
}

public double Round(double number, int decimalPlaces) {
```

```

    double modifier = Math.pow(10.0, decimalPlaces);

    return Math.round(number * modifier) / modifier;
}

public String getCurrentTime() {

    String hr1;

    String min1;

    String sec1;

    java.util.Calendar cal = new java.util.GregorianCalendar();

    int hr = cal.get(GregorianCalendar.HOUR_OF_DAY);

    int min = cal.get(GregorianCalendar.MINUTE);

    int sec = cal.get(GregorianCalendar.SECOND);

    if (Integer.toString(hr).length() < 2) {

        hr1 = "0" + hr;

    } else {

        hr1 = hr + "";

    }

    if (Integer.toString(min).length() < 2) {

        min1 = "0" + min;

    } else {

        min1 = min + "";

    }

    if (Integer.toString(sec).length() < 2) {

        sec1 = "0" + sec;

    } else {

```

```

        sec1 = sec + "";
    }

    String TIME = hr1 + ":" + min1 + ":" + sec1;

    return TIME;
}

public String getTodaysdate() {

    String month1;

    String day1;

    java.util.Calendar cal = new java.util.GregorianCalendar();

    int year = cal.get(GregorianCalendar.YEAR);

    int month = (cal.get(GregorianCalendar.MONTH) + 1);

    int day = cal.get(GregorianCalendar.DATE);

    if (Integer.toString(month).length() < 2) {

        month1 = "0" + month;

    } else {

        month1 = month + "";

    }

    if (Integer.toString(day).length() < 2) {

        day1 = "0" + day;

    } else {

        day1 = day + "";

    }

    String TODAY = year + "-" + month1 + "-" + day1;

    return TODAY;
}

```

```

}

public String getPortalCopyright() {

    String rt = "Copyright 2019";

    return rt;

}

public String getReceiptFooter(String paycat) {

    String loginPage = "";

    return loginPage;

}

public String[] getStatesInNigeria() {

    String[] st = {"Abia", "Adamawa", "Akwa Ibom", "Anambra", "Bauchi",
"Bayelsa", "Benue", "Borno",

    "Cross River", "Delta", "Ebonyi", "Edo", "Ekiti", "Enugu", "FCT", "Gombe",

    "Imo", "Jigawa", "Kaduna", "Kano", "Katsina", "Kebbi", "Kogi", "Kwara",
"Lagos",

    "Nasarawa", "Niger", "Ogun", "Ondo", "Osun", "Oyo", "Plateau", "Rivers",

    "Sokoto", "Taraba", "Yobe", "Zamfara"

    };

    return st;

}

public byte[] scale(byte[] fileData, int width, int height) {

    ByteArrayInputStream in = new ByteArrayInputStream(fileData);

    ByteArrayOutputStream buffer = new ByteArrayOutputStream();

```

```

try {
    BufferedImage img = ImageIO.read(in);

    if (height == 0) {
        height = (width * img.getHeight()) / img.getWidth();
    }

    if (width == 0) {
        width = (height * img.getWidth()) / img.getHeight();
    }

    Image scaledImage = img.getScaledInstance(width, height,
Image.SCALE_SMOOTH);

    BufferedImage imageBuff = new BufferedImage(width, height,
BufferedImage.TYPE_INT_RGB);

    imageBuff.getGraphics().drawImage(scaledImage, 0, 0, new Color(0, 0, 0),
null);

    ImageIO.write(imageBuff, "jpg", buffer);
} catch (IOException e) {
}

return buffer.toByteArray();
}

public String getMD5(String str) {

    String co = str;

    MessageDigest md;

    try {

        md = MessageDigest.getInstance("MD5");

```

```

byte[] pb = str.getBytes();

md.reset();

byte[] db = md.digest(pb);

StringBuilder sb = new StringBuilder();

for (int i = 0; i < db.length; i++) {

    sb.append(Integer.toHexString(0xff & db[i]));

}

co = sb.toString();

} catch (NoSuchAlgorithmException j) {

}

return co;

}

public static String toCamelCase(String inputString) {

String result = "";

if (inputString.length() == 0) {

    return result;

}

char firstChar = inputString.charAt(0);

char firstCharToUpperCase = Character.toUpperCase(firstChar);

result = result + firstCharToUpperCase;

for (int i = 1; i < inputString.length(); i++) {

    char currentChar = inputString.charAt(i);

    char previousChar = inputString.charAt(i - 1);

    if (previousChar == ' ') {

```

```

        char currentCharToUpperCase = Character.toUpperCase(currentChar);

        result = result + currentCharToUpperCase;

    } else {

        char currentCharToLowerCase = Character.toLowerCase(currentChar);

        result = result + currentCharToLowerCase;

    }

}

return result;

}

public String generatUUID() {

    String uniqueID = UUID.randomUUID().toString();

    return uniqueID;

}

public String generateHash512(String target) {

    try {

        MessageDigest md = MessageDigest.getInstance("SHA-512");

        md.update(target.getBytes());

        byte byteData[] = md.digest();

        //convert the byte to hex format method 1

        StringBuilder sb = new StringBuilder();

        for (int i = 0; i < byteData.length; i++) {

            sb.append(Integer.toString((byteData[i] and 0xff) + 0x100,

16).substring(1));

```

```

    }

    return sb.toString().toUpperCase();

} catch (NoSuchAlgorithmException e) {

    throw new RuntimeException(e);

}

}

```

```

public String generateId() {

    UUID uid = UUID.randomUUID();

    String id = String.valueOf(uid).replaceAll("-", "");

    return id;

}

```

```

public String getPreviousDate() {

    DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

    Calendar cal = Calendar.getInstance();

    cal.add(Calendar.DATE, -1);

    return dateFormat.format(cal.getTime());

}

```

```

public String getCurrentdatetime() {

    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm:ss");

    LocalDateTime datetime = LocalDateTime.now();

    String curr = datetime.format(formatter);

    return curr;
}

```



```

    }

    public String createEmailMessage(String name, String en, String amount, String link)
    {
        String msg = "<img src=\"\" + this.project_url + "/img/logo.png\"><br/>"
            + "Dear <em>" + name + "</em><br/>"
            + "Your scheduled backup with entity name<b>" + en + "</b> was
Successful<br/>"
            + "Your current credit balance is: <b>" + amount + "</b><br/><br/>"
            + "Click <a href=link>here</a> to download the backup file to your local
system.<br/>";

        return msg;
    }

    public String createEmailMessage2(String name, String en, String amount, String
link) {
        String msg = "<img src=\"\" + "/img/logo2.png\"><br/>"
            + "Dear <em>" + name + "</em><br/>"
            + "A new backup for <b>" + en + "</b> has been created<br/>"
            + "Your current credit balance is: <b>" + amount + "</b><br/><br/>"
            + "Click <a href=\"\" + this.project_url + "/DownloadBackups?url=" + link
+ "\">here</a> to download your backup file.<br/>"
            + "Or copy the link below and paste on the browser to continue<br/><br/>"
            + link;

        return msg;
    }
}

```

```

public String createEmailMessage3(String name, String en, String amount, String
link) {

    String msg = "<img src=\"\" + project_url + \"/img/logo2.png\"><br/>"

        + "Dear <em>" + name + "</em><br/>"

        + "A new backup for <b>" + en + "</b> has been created<br/>"

        + "Your current credit balance is: <b>" + amount + "</b><br/><br/>"

        + "Click <a href=\"\" + project_url + "/DownloadBackups?url=" + link +
"\>here</a> to download your backup file.<br/>"

        + "Or copy the link below and paste on the browser to continue<br/><br/>"

        + link;

    return msg;

}

```

```

public String createErrorEmailMessage(String name, String refno, String transid,
String reason) {

    String msg = "<img src=\"\" + \"/img/logo.png\"><br/>"

        + "Dear <em>" + name + "</em><br/>"

        + "Your transaction with reference number <b>" + refno + "</b> was
<b>Unsuccessful</b><br/>"

        + "Transaction ID: <b>" + transid + "</b><br/>"

        + "Reason: <b>" + reason + "</b><br/>"

        + "Click <a href=\"\" + this.project_url + "/paymentChannels.jsp?payref=" +
refno + "\">here</a> to try again.<br/>"

        + "Or copy the link below and paste on the browser to continue<br/><br/>"

```

```
        + this.project_url + "/paymentChannels.jsp?payref=" + refno;

    return msg;
}
```

```
public String returnMonth(String date) {

    String[] array = new String[3];

    StringTokenizer st = new StringTokenizer(date, "/-:");

    int i = 0;

    while (st.hasMoreElements()) {

        array[i] = st.nextToken();

        i += 1;

    }

    return array[1];

}
```

```
public String returnDay(String date) {

    String[] array = new String[3];

    StringTokenizer st = new StringTokenizer(date, "/-:");

    int i = 0;

    while (st.hasMoreElements()) {

        array[i] = st.nextToken();

        i += 1;

    }

}
```

```

    return array[2];
}

public String returnDay2(String date) {
    String[] array = new String[3];
    StringTokenizer st = new StringTokenizer(date, "/-:");
    int i = 0;
    while (st.hasMoreElements()) {
        array[i] = st.nextToken();
        i += 1;
    }
    int d = Integer.parseInt(array[2]);
    return d + "";
}

public String returnYear(String date) {
    String[] array = new String[3];
    StringTokenizer st = new StringTokenizer(date, "/-:");
    int i = 0;
    while (st.hasMoreElements()) {
        array[i] = st.nextToken();
        i += 1;
    }
    return array[0];
}

```

```

    }

    public String[][] getMonth() {

        String[][] month = {

            {"01", "January"},

            {"02", "Febuary"},

            {"03", "March"},

            {"04", "April"},

            {"05", "May"},

            {"06", "June"},

            {"07", "July"},

            {"08", "August"},

            {"09", "September"},

            {"10", "October"},

            {"11", "November"},

            {"12", "December"}

        };

        return month;

    }

    public long convertStringToTimestamp(String str_date) {

        try {

            DateFormat formatter;

            formatter = new SimpleDateFormat("yyyy-mm-dd");

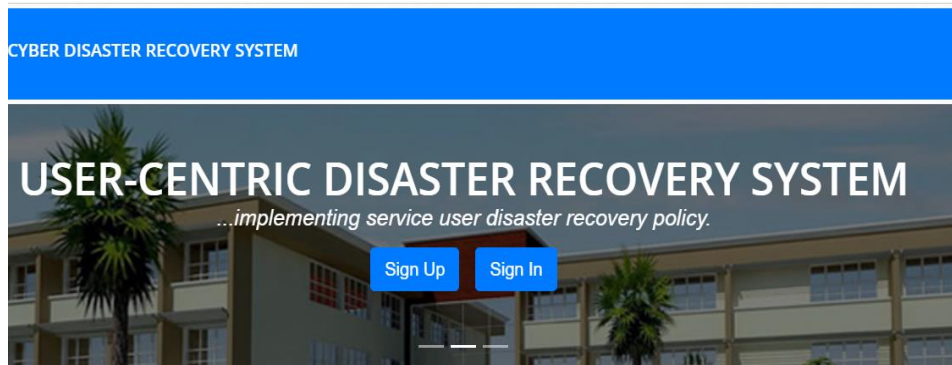
            Date date = (Date) formatter.parse(str_date);

            java.sql.Timestamp timeStampDate = new Timestamp(date.getTime());

```

```
System.out.println(str_date+"\t"+timeStampDate.getTime()*1000);  
  
return timeStampDate.getTime()*1000;  
  
} catch (ParseException e) {  
  
System.out.println("Exception :" + e);  
  
return 00;  
  
}  
  
}  
  
}
```

APPENDIX B SAMPLE OUTPUTS



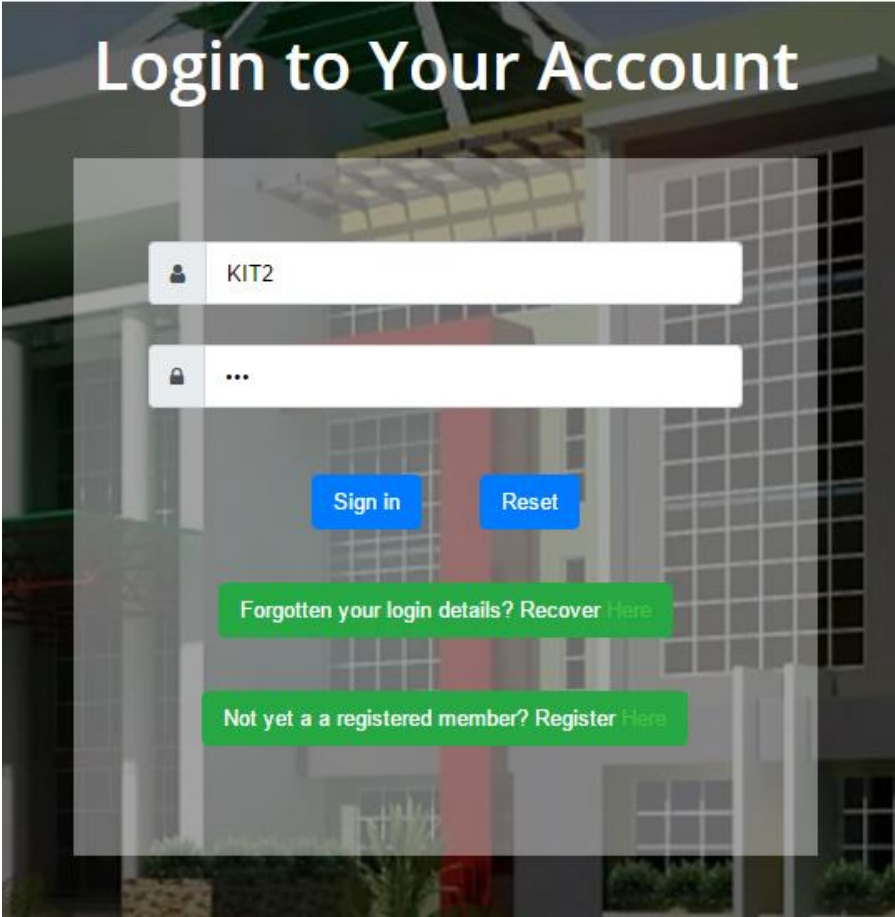
ABSTRACT

In a world of interdependent economies and online transactions, a large volume of data hosted on the cyberspace daily bases. Cyber threats and attacks are steadily increasing. Most time, these threats and attacks are targeted at service providers but service users are greatly affected by the attacks due to their vulnerability level. When disasters knockdown the infrastructures of a single service provider, it will have ripple effects on thousands of innocent service users. Therefore, service users need more than ever to prepare for major crises targeted at their service providers. To cope with this trends, every service user requires an independent business continuity plan (BCP) or disaster recovery plan (DRP) and data backup policy which falls within their cost constraints while achieving the target recovery requirements in terms of recovery time objective (RTO) and recovery point objective (RPO). The aim of this research work is to develop a model for a user-centric disaster recovery system to enable service users to independently develop their data backup policies that best suits their remote databases. The system developed is highly compatible with MYSQL, MSSQL and Oracle databases. With this system, service users have the liberty to independently define and implement their private backup plans and disaster

Home Page

The image shows the 'Users Registration' page. The title 'Users Registration' is at the top. The form contains several input fields: 'First Name' and 'Last Name' (with person icons), 'Gender' (a dropdown menu set to 'Male'), 'Date of Birth' (a calendar icon and 'mm/dd/yyyy' format), 'Email' (with an envelope icon), 'Phone Number' (with a phone icon), 'Username' (with a dropdown menu set to 'Choose a Username'), 'Password' (two fields, each with a lock icon), 'Captcha Code' (with a red 'R e c o v e r y' image), 'Verify Captcha Code', and 'Are you human?'. A blue 'Register' button is at the bottom center.

Sign Up Page



Sing In Page

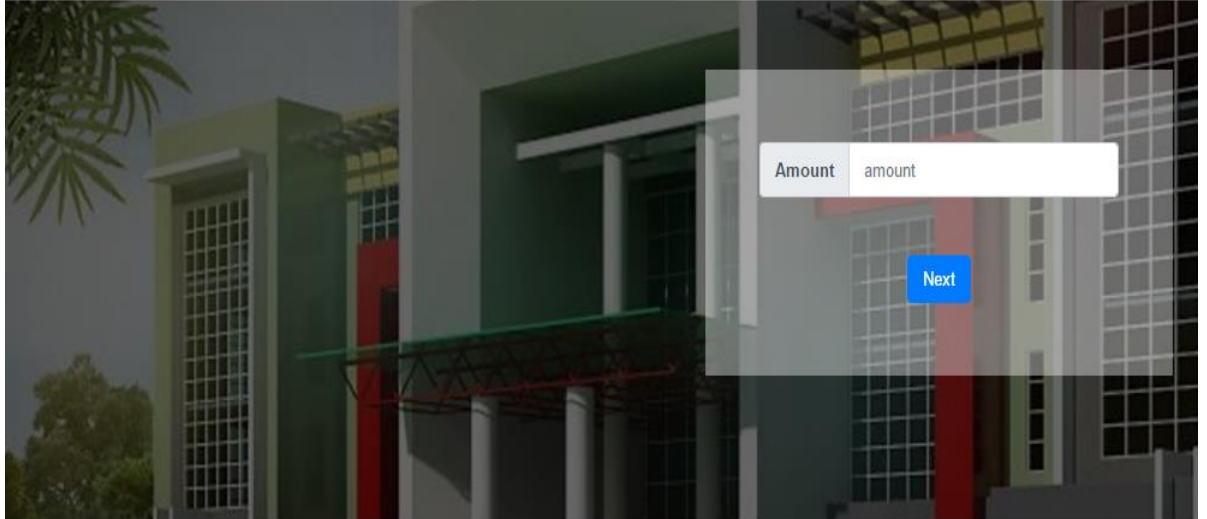
New App

#	Database Type	Database URL	Database Name	Port No	Username	Password	App Configurations	Edit	Delete
1	MySQL	localhost	blinks	3306	root	mysql	1	Edit	Delete
2	MySQL	localhost	myedu	3306	root	mysql	1	Edit	Delete
3	MySQL	localhost	minkisuites	3306	root	mysql	1	Edit	Delete

List of Registered Application

Add Funds

Currently, your wallet balance is: (NGN 9,535.62)




Add Funds Page

Payment Options

ONLINE PAYMENT OPTION

You have selected to add N1,000.00

Click the 'Pay Online' Button below to proceed.

 This payment method makes use of the payer's ATM Card details. Payment is processed instantly and will grant you access to the requested service.

[Pay Online](#)

Please note that our application does not store your ATM card details. Your card details are entered on the Interswitch platform

BANK PAYMENT OPTION

You have selected to add N1,000.00

You can equally make payment through the Bank details below;

1. UBA

Account Name: Kreative Information Technologies Nigeria Ltd

Account Number: 2005368812

Sort Code: 2349872468

2. GTB

Account Name: Kreative Information Technologies Nigeria Ltd

Account Number: 8812200536

Sort Code: 7246823498

Payment Options Page