DESIGN AND IMPLEMENTATION OF AN IMPROVED ELECTRONIC DOCUMENT MANAGEMENT SYSTEM (ENCODOC)

By

AJALA, OLUDAYO SAMUEL, M.Sc

Information Technology (NOUN 2015),

NOU133765867

A DISSERTATION SUBMITTED TO THE SCHOOL OF SCIENCE AND TECHNOLOGY,

NATIONAL OPEN UNIVERSITY OF NIGERIA,

LAGOS, NIGERIA

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

SCHOOL OF SCIENCE AND TECHNOLOGY

NATIONAL OPEN UNIVERSITY OF NIGERIA,

LAGOS, NIGERIA

JUNE, 2015

DECLARATION

I **AJALA, OLUDAYO SAMUEL** *humbly declare that this work entitled* **DESIGN AND IMPLEMETATION OF AN IMPROVED DOCUMENT MANAGEMENT SYSTEM (ENCODOC)** *is as a result of my research effort carried out in the School of Science and Technology National Open University of Nigeria under the supervision of Professor. O. A. Awodele. I further wish to declare that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any degree or diploma of any university or other institute of higher learning except where due acknowledgement has been made in the text.*

_____

"(signature /name/date)"

CERTIFICATION

This is to certify that this Dissertation entitled **DESIGN AND IMPLEMENTATION OF AN IMPROVED DOCUMENT MANAGEMENT SYSTEM (ENCODOC)** *was carried out by Ajala, Oludayo Samuel in the School of Science and Technology, National Open University of Nigeria, Lagos for the award of Master of Science Degree in Information Technology.*

_____	_____
Supervisor's Signature	Programme Leader's Signature
Date	Date

ACKNOWLEDGEMENTS

My profound gratitude to the Almighty God, the Alpha and Omega, Beginning and the End. I adore and will always adore Him for His absolute support throughout my sojourn in this citadel of learning and during the course of this work.

I acknowledge my ever supporting supervisor Professor O.A Awodele for his assistance, time, devotion, encouragement and advice throughout the project. I appreciate the effort of my co-supervisor Dr. F.K. Osisanwo who tirelessly helped me in the completion of this work. God bless you all.

I thank my wonderful parent Mr. and Pastor (Mrs.) Ajala, who always believe in me, support me financially, morally and spiritually. May you live long to eat the fruit of your labour. To my wonderful siblings Olubisi, Mary, Kemi and Odun, I appreciate you all for being there when I needed you the most. Also to my wonderful colleague Mr. Afikode, Mr. Illesanmi, Mr. Donald, Mr. Peter, Mr. Gbolagade. Thank you so much and I will surely miss you all.

ABSTRACT

Challenges Nigeria organizations are facing which include embezzlement, fraud, misappropriation and mismanagement can be checkmated if there is a proper and secure documentation of all documents. Hence this work focuses on the design and Implementation of an Improve Document Management System for Oyo state Housing Corporation with special emphasis on security and space management; Waterfall design model and three-tier architecture was used in the design of this work. HTML, CSS, Javascript and Jquery were used for the codes at the Client side, PHP was used at the Server-side scripting and MySQL was used at the Data Server side in the work. PHP AES encryption algorithm was used in encryption and decryption of all allowed document types in the system while Zzlib library was used for the compression module in the work. A two weeks user evaluation of the system showed interesting usage scenarios and future trends for improving user interaction. The work made it easier for organizations to find document, securely store document, share document without stress. Considering the immense benefit this work holds, it is important for all organizations to key into this work as a means of securely documenting its documents.

Keywords: Electronic Document Management System, Encryption, Compression, Improved EDMS, Security, Document Space Management, PHP AES Encryption.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

For many years, records management and the physical filing and retrieval of information were what companies did until the 80's when word processing sprung into life. Suddenly, documents were appearing in hardcopy and on computer. Over the next decade, document and document image management tried to find a footing in corporate and government offices but even though they worked to a degree, they were complex, extremely expensive, hard to manage and require a lot of effort for the users to index documents (Knowledgeone, 2005). In common language the word document usually means an information carrier containing written or drawn information for a particular purpose. Central to the idea of a document is usually that it can be easily transferred, stored and handled as a unit (Bjork, 2002). Organizing large volumes of physical records are difficult and there are instances where it is difficult to extract a record/document from the large volume of them, it is almost certain that management of document is prone to human error (Akashah, Syamsul, Jusoff & Christon, 2011).

Nowadays, storage systems are increasingly subject to attacks. So the security system is quickly becoming mandatory feature of the data storage systems. For the security purpose we are always dependent on the cryptography techniques (Kahanwal, Dua & Singh, 2012).

Encryption is the most effective way of computer science concerned with developing schemes and formula to achieve data and information security through the use of codes (Das, Lanjewar & Sharma, 2013). Today the privacy is the main issue to sending information from one point to another in data transmission. Encryption is the procedure that allows messages or information to be encoded in such a way that it is extremely difficult to read or understand where decryption is the procedure to transforming encoded text into the original message and

information (Das *et al.*, 2013). Why encryption? It is obvious there are developments in the field of information technology likewise have malware (a software that gives partial or full control of your computer to do whatever the malware creator wants) and Sniffer (a program or a piece of hardware that can intercept and log traffic passing over a digital network or part of a network) technology improved giving rise to unauthorized access and control of information, records and data. Security is a critical issue any organization can't joke about.

Data compression have a long history, it has been around since the beginning of electronics. Data compression can also be referred to as source coding or bit-rate reduction. It is the name for encoding information to minimize or at least reduce the number of bits used to encode information. Compression is usually used to minimize the space used on a hard disk or to minimize the amount of data transmitted through a transmitter that has limited bandwidth (Grajeda, Uribe & Parra, 2006). There are many positive effects from compressing data such as smaller size and less bandwidth usage; there are also some negative effects. The data must be decompressed to be read and this is an operation that takes recourses from the processor or hardware that needs the decompressed data. Data compression is often divided into lossy compression and lossless compression. Lossy compression is compression that is done with some losses in the original message. It is often used in audio and video applications and other situations where some data can be lost without the message being distorted beyond recognition (Grajeda *et al*., 2006).

The compression type that will be used in this work is the lossless form of data compression. Lossless data compression keeps the entire original message during the compression. The compressed data can be decompressed at any time and the entire message is kept down to the last bit. Lossless compression is used when no data can be lost, which is the case in this work. Data compression has been playing an important role in the areas of data transmission and data storage. Many great contributions have been made in this area, such as Huffman coding,

LZW algorithm, run length coding, and so on. These methods only focus on the data compression. On the other hand, it is very important for us to encrypt our data to against malicious theft and attack during transmission (Almelkar & Gandhe, 2014). Why Compression? File compression is a process of "packaging" a file (or files) to use less disk space. Compression works by minimizing redundancy in a file's code. Compression software allows you to take many files and compress them into one file, which is smaller than the combined size of the originals. Therefore, File compression allows you to store and back up significantly more data, faster which mean the transfer time and bandwidth needed is lessened when files are compressed. File compression is often a necessity for sending large documents over the Internet as email attachments since most email systems limit the size of each email message. Often it is easier to compress multiple documents into one document to attach to an email message rather than attaching them one-by-one. Also Files can become corrupted when they are transferred over the Internet in an uncompressed format.

Therefore, Enocodoc which for the purpose of this work is taken as the improved electronic document management system will be considered by using encryption, decryption and compression algorithm to strike balance into document management systems to improve it security and space management features, it can't be predicted when an attack targeted at rendering document useless or exposing vital information may occur.

1.2 Statement of the Problem

In an interview organized by Chinua Achebe Foundation: the Buhari/Idiagbon administration reported that Nigeria had already paid 27 billion of its external debt; why did Nigeria pay another 12 billion without the approval of the legislature (Umunnah, 2012). It is evident that improper documentation of files (records and documents) is the basis of unaccountability, embezzlement and fraud in any organization. The continuing existence of paper documents,

and their use alongside electronic documents, raises some key set of problems as they have paved ways for corruption, misappropriation, mismanagement, embezzlements and fraud as document concerning how money are being spent, how project are being carried out and how expenses used on project are either not properly kept or are easily tampered with without any trace of whom or when such change on such document were made. Therefore a system is needed that will end the dreadful life span of the virus called mismanagement, fraud and embezzlement that has sucked in deep into all our organizations.

1.3 Research Motivation

PUNCH Metro showed that many youths had developed into syndicates, using court premises as centres for issuing fake driver's licenses, number plate, Certificates of Occupancy, survey plans and Tax Identification Numbers, among others. It was learnt that these documents could be used to secure bail term, get loans from banks and to process documents for visas at embassies. Investigations also showed that it takes electronic verifications to discover that these documents were cloned (Hanafi, 2015).

There are few document management systems available but Encodoc has its own unique features which are ordering document retrieval, sorting of document, secure document processing (encryption and decryption) and document space management (compression, decompression and archiving) features which are not presently available in the existing EDMS. This work will expose PHP compression and encryption capacity in EDMS as this is needed in organizations to secure and maximize the available space of storage. In essence, the work will enhance document accountability and traceability, documents security, document storage and retrieval and ensure consistent business processes (workflows) for how documents are handled. It should be noted that the EDMS will also allow different stakeholders in the organization to view and access file from their convenience.

1.4 Aim and Objectives

The aim of this work is to design an improved E-Document management system while the specific objectives are to:

i. study existing works on DMS, encryption, decryption and compression algorithm.

ii. design a robust database to facilitate the EDMS and

iii. design an Improved electronic Document management system using the PHP version of AES encryption and decryption algorithm code and adding a compression module using PHP ZIP compression function.

1.5 Research Method Overview

Encodoc will be designed as a multi-tiers Web-based application, built using free and open source software. Tools to be used for the work will include: HTML, CSS, PHP, MYSQL, WAMP and Dreamweaver.

Three tiers architecture was adopted for this work:

1. The front-end tier: it is the user interface, making use of DREAMWEAVER which is the Integrated Development Environment where code is written, it includes all the web forms html page, PHP/Ajax enable.

2. The middle-end tier: is the web server which will be used for testing (loading, viewing and deploying) the project. To achieve encryption, a PHP based AES encryption will be developed which will act a module to be integrated into the main EDMS PHP code. To enable compression/archiving PHP ZIP function will be used and it will also be integrated into the main EDMS PHP Code and WAMP server will be used to this effect.

3. The back-end tier: This is the database. It stores information provided by the user through the user interface. For this work a robust database will be designed to ensure easy

access, retrieval, and manipulation of the information stored. The tool to use to this effect is PHPMYADMIN (MYSQL).

In the course of gathering data, the following activities shall be carried out:

(a) Research related work will be reviewed to understand in details the concept of encryption, decryption, compression and document management itself.
(b) Interviewing the stakeholder of Oyo state Housing Corporation to get first-hand information about the existing system.

1.6 Purpose of the Study

The purpose of this work is to bridge the gap between documentation and the security, integrity and tampering issues. This work also save space, money and time in the processing of documents compared to the current traditional paper documentation which requires large storage space and requires labour.

1.7 Significance of the Study

Encodoc gives you all the features you need to effectively manage your documents and that of your clients. It saves time, energy and expense on documentation to vastly improve the overall productivity of the organization and include multiple levels of security and version control to allow access to sensitive documents only with the proper permissions. For this work, Oyo State Housing Corporation was used as a case study as a large number of documents needed proper documentation and secure safe for a long period of time.

1.8 Organization of the Work

Chapter one introduces the research work; it presents the historical background of the study, the aim and objectives of the work. Chapter two shows a review of the literature about EDMS, advantages of using an EDMS, followed by discussion of research that focused on

identifying problems and challenges in implementing an EDMS likewise researching into literature of cryptography, its classification, encryption, decryption and compression, while it also introduces the waterfall model, it advantages, disadvantages and the modified waterfall model. Chapter three describes the methodology applied, tools used, design consideration and design architecture.

Chapter four shows how the work is being implemented. It highlighted the System Hardware Requirement, System Software Requirement and the procedure used in implementing the work while algorithms use and sample snapshots were also presented and its evaluation and discussion were also made. Chapter five gives a summary, conclusion and recommendation based on the results and findings of the work. Suggestion was also made for future research related to this work.

1.9 Definition of Terms

AES: The Advanced Encryption Standard is a symmetric-key block cipher algorithm and U.S. government standard for secure and classified data encryption and decryption.

Algorithms: An algorithm is a procedure or formula for solving a problem.

ASCII (American Standard Code for Information Interchange): is a character-encoding scheme. Originally based on the English alphabet, it encodes 128 specified characters into 7-bit binary integers as shown by the ASCII chart on the right.

Cryptanalysis: is the study of ciphers, ciphertext, or cryptosystems (that is, to secret code systems) with a view to finding weaknesses in them that will permit retrieval of the plaintext from the ciphertext, without necessarily knowing the key or the algorithm.

Cryptanalyst: is a specialist in cryptanalysis (i.e. A person expert in analyzing and breaking codes and ciphers).

Compression: is a reduction in the number of bits needed to represent data.

DES (Data encryption standard): is a common standard for data encryption and a form of secret key cryptography (SKC), which uses only one key for encryption and decryption.

DBMS (Database management systems): A computer software applications that interact with the user, other applications, and the database itself to capture and analyze data.

DMS (Document Management System): A document management system is a system (based on computer programs in the case of the management of digital documents) used to track, manage and store documents.

DSA (Digital Signature Algorithm): The Digital Signature Algorithm is a Federal Information Processing Standard for digital signatures.

EDMS (Electronic Document Management System): This is a collection of technologies that work together to provide a comprehensive solution for managing the creation, capture, indexing, storage, retrieval, and disposition of records and information assets of the organization.

GUI (Graphical user interface): GUI is a human-computer interface (i.e., a way for humans to interact with computers) that uses windows, icons and menus and which can be manipulated by a mouse.

Data integrity: This refers to maintaining and assuring the accuracy and consistency of data over its entire life-cycle, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data.

NIST (National Institute of Standards and Technology): this is a non-regulatory agency of the United States Department of Commerce whose mission is to Promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life.

NSA (National Security Agency): is an intelligence organization of the United States government, responsible for global monitoring, collection, and processing of information and data for foreign intelligence and counterintelligence purposes.

Privacy (Also called information privacy): is the aspect of information technology (IT) that deals with the ability an organization or individual has to determine what data in a computer system can be shared with third parties.

Unicode: is a computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems. Developed in conjunction with the Universal Character Set standard and published as The Unicode Standard, the latest version of Unicode contains a repertoire of more than 110,000 characters covering 100 scripts and multiple symbol sets.

CHAPTER TWO

LITERATURE REVIEW

In this chapter a review of the literature about Electronic Document Management Systems will be provided as motivation for this work. In the first section of this chapter, research about the evolution of document management will be discussed. This will be followed by a discussion of research that focused on cryptography and identifying the different type of encryption and their comparison. Discussion on compression will also be provided and related works in the field of work will be treated.

2.1 Document Management System

There are several definition for Document Management System, they are:

Document Management System is the automated control of electronic documents- page images, spreadsheets, word processing documents, and complex, compound documents - through their entire life cycle within an organization, from initial creation to final archiving. It allows organizations to exert greater control over the production, storage, and distribution of documents, yielding greater efficiencies in the ability to reuse information, to control a document through a workflow process, and to reduce product cycle times (Amir, 2007).

Document Management System: A proprietary electronic system that scans stores and retrieves documents received or created by an organization. There is a distinction between this and an Electronic Records Management System (Paperwise, 2015).

Document Management System: Originally, a document management system was a computer program (or set of programs) used to track and store images of paper documents. More recently, the term has been used to distinguish between imaging and records management systems that specialize in paper capture and records respectively. Document management systems commonly provide check-in, check-out, storage and retrieval of electronic documents often in the form of word processor files and the like (Keyes, 2012).

Document Management System: Handles documents by electronically storing, organizing, indexing and filing. They can be retrieved when required, without any loss of time. It uses imaging technology to enable access to the unstructured data, it brings all documents to your desktop and enables you to work with them, eliminating the need for paper-based documents and it is a powerful document archival system, which ensures safety of documents, faster access to them and huge cost savings (Delonti, 2014).

From all the definition above, we can simplify that a Document Management System is a process of documenting a document to a proper location and storage. Document management systems commonly provide storage, versioning, metadata, security, as well as indexing and retrieval capabilities. It is done so that it will be easier for the organization to retrieve and use the data efficiently and effectively. It reduces time, faster access and saves human cost. Because of this characteristic, Document Management System must also have a good security system and easy interface level so that it is safe and environment friendly to use.

2.1.1 Evolution of Document Management System

In the early days of document management, businesses and individuals had to keep all their files in filing cabinets. This allows for files to be stored in an understandable system. However, the systems became cumbersome, due to the huge amount of space that was used. Large companies would have entire rooms or whole storage areas full of filing cabinets for all of their paperwork. This was extremely inefficient; each file that came in needed to be filed, costing the business resources and time. In addition to that, if a file ever needed to be accessed, the company would have to spend more time, energy, and resources trying to find it. Sometimes this could take hours, and often ended in futility. The physical copies of the documents were difficult to deal with. There were a number of problems that companies faced in regards to the files they kept. These files could be stolen or misplaced. On top of

this, there was the threat of disaster such as a flood or fire. These tragedies would eliminate all documentation, making true document management impossible.

Beginning in the 1980s, a number of vendors began developing software systems to manage paper-based documents. These systems dealt with paper documents, which included not only printed and published documents, but also photographs, prints etc. (Wikipedia, 2015). Computers were becoming available to companies. Along with the computers, businesses could get document management storage, which would allow them to start keeping important files on the computer.

Later developers began to write a second type of system which could manage electronic documents, i.e., all those documents, or files, created on computers, and often stored on users' local file-systems. The earliest electronic document management (EDM) systems managed either proprietary file types, or a limited number of file formats. Many of these systems later became known as document imaging systems, because they focused on the capture, storage, indexing and retrieval of image file formats. EDM systems evolved to a point where systems could manage any type of file format that could be stored on the network. The applications grew to encompass electronic documents, collaboration tools, security, workflow, and auditing capabilities.

This decade, there are a number of options when it comes to document management; some of these systems are user-friendly while some are not.

2.2 Basic Aspect of Computer Security

There are three basic aspects in computer security: confidentiality, integrity and availability. Anderson (2001) argues that there are more aspects than these, but this work will use the view of Bishop (2003) on security with only three aspects.

2.2.1 Confidentiality

The confidentiality aspect concerns all access restrictions to information and resources. Information can be hidden or scrambled to limit the access to only concerned parties, or the information is protected by a virtual barrier enforced with for example the access control of an operating system. Existence of data may also apply to confidentiality, because the existence may reveal more than the data itself. For example if your organization normally never encrypts email, someone monitoring the trace could conclude that something is about to happen when suddenly all emails are encrypted. Hiding resources is also an important part of confidentiality, because the notion of system configuration and what operating systems are used can help an attacker to find the weakest link in your protection.

Contrary to Anderson (Anderson, 2001), secrecy and privacy is covered by the confidentiality aspect. With secrecy, it means to limit the amount of people having access to the information and privacy is the ability to protect personal secrets. Bishop makes no distinction between secrecy, privacy and confidentiality.

2.2.2 Integrity

How much data can be trusted or the trustworthiness of data and resources is the aspect of integrity. It can be divided into data integrity, that is the integrity of the contents, and origin integrity, the source of the data is correct, which is also called authentication. Integrity violations can either be prevented or detected. A prevention mechanism tries to prohibit any unauthorized operations that attempts to change the data. A user can try to change data for which he or she is not authorized for or an authorized user can try to change the data in other ways than allowed, i.e. authorized user performing an unauthorized operation. Bishop gives a good example to explain this difference (Bishop, 2003). Suppose an accounting system is running on a computer and someone hacks into the system and tries to modify the data that is an unauthorized user tries to violate the integrity of the system. But what if a hired accountant

tries to steal money by modifying the data and virtually transfer money to his own account that is an authorized user tries to perform an unauthorized operation. The detection mechanism tries not to prevent modification, rather making it possible to detect any modification violating the integrity. The data may be reported to be no longer trustworthy. Hence, Data integrity means maintaining and assuring the accuracy and consistency of data over its entire life-cycle, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data.

2.2.3 Availability

The ability to access information or a resource is referred to availability. A service can deliberately be blocked by an attacker making this aspect a part of computer security. These attempts are called denial of service attacks, or DOS-attacks for short, and can be difficult to identify and separated from increased normal-usage. For example, if you are running a popular web site and suddenly one day the site hits a peak in load. Has the site become very popular and you have not enough servers to handle this increased popularity, or is it an attacker performing a denial of service attack? By manually analyzing every request, you would be able to conclude whether it is an attack or not, but it is difficult for a mechanism to prevent and detect the attack, because it would require usage-pattern analysis.

2.3 Cryptography, Encryption and Decryption

From the word “cryptography", "crypto" stands for "hidden, secret", and "graphy" denotes "a process or form of drawing, writing, representing, recording, describing, etc., or an art or science concerned with such a process”. If the base word "encrypt" is broken into its root, you will see "en" and "crypt". The "en" part means "to make", and the "crypt" part (a variation of "crypto") means hidden or secret. Since "encrypt" is a verb, the base term then means "to make hidden or secret" (Forlanda, 2013).

Hence, Cryptography is the science of using mathematics to encrypt and decrypt information. Once the information has been encrypted, it can be stored on insecure media or transmitted on an insecure network (like the Internet) so that it cannot be read by anyone except the intended recipient and Encryption is the process of converting messages, information, or data into a form unreadable by anyone except the intended recipient. Encrypted data must be deciphered, or decrypted, before it can be read by the recipient (Aemer, 2005; Desai, 2014). Decryption is the process in which the ciphertext is converted back to plaintext.

Forlanda (2013) made an attempt to differentiate cryptography and encryption; he defined cryptography as the study or science of secret communication, while encryption is simply a component of that science. He further defined encryption as the process of hiding information, through the use of ciphers, from everybody except for the one who has the key and that encryption is a direct application of cryptography, and is something that websites use every day to protect information.

Human have been interested in protecting their messages and attempting to conceal certain information that they wanted to keep to their own possession by substituting parts of the information with symbols, numbers and pictures for different reason. The evolution of cryptography can be attributed to the need of protecting sensitive secrets (e.g. the Assyrians were interested in protecting their trade secret of manufacturing of the pottery, the Chinese were interested in protecting their trade secret of manufacturing silk, the Germans were interested in protecting their military secrets by using their famous Enigma machine).

2.3.1 The Evolution of Cryptography, Encryption and Decryption

About 1900BC, An Egyptian scribe used non-standard hieroglyphs in an inscription (Kahn, 1967). Kahn (1967) lists this as the first documented example of written cryptography. 1500BC, ancient Assyrian merchants used intaglio, a piece of flat stone carved into a collage of images and some writing to identify themselves in trading transactions. Using this

mechanism, they are producing what today we know as 'digital signature.' The public knew that a particular 'signature' belonged to this trader, but only he had the intaglio to produce that signature. 500-600 BC, Hebrew scribes writing down the book of Jeremiah used a reversed-alphabet simple substitution cipher known as ATBASH. (Jeremiah started dictating to Baruch in 605 BC but the chapters containing these bits of cipher are attributed to a source labeled "C" (believed not to be Baruch) which could be an editor writing after the Babylonian exile in 587 BC, someone contemporaneous with Baruch or even Jeremiah himself).

ATBASH was one of a few Hebrew ciphers of the time (Dupuis, 1999). 487 BC, The Greeks used a device called the "skytale" – a staff around which a long, thin strip of leather was wrapped and written on. The leather was taken off and worn as a belt. Presumably, the recipient would have a matching staff and the encrypting staff would be left home. Julius Caesar (100-44 BC) used a simple substitution with the normal alphabet (just shifting the letters a fixed amount) in government communications. This cipher was less strong than ATBASH, by a small amount, but in a day when few people read in the first place, it was good enough. He also used transliteration of Latin into Greek letters and a number of other simple ciphers. When Julius Caesar sent messages to his trusted acquaintances, he didn't trust the messengers. So he replaced every A by a D, every B by an E, and so on through the alphabet. Only someone who knew the "shift by 3" rule could decipher his messages (Singh, 2000).

In 1379, Gabrieli di Lavinde at the request of Clement VII, compiled a combination substitution alphabet and small code -- the first example of the nomenclator Kahn has found. This class of code/cipher was to remain in general use among diplomats and some civilians for the next 450 years, in spite of the fact that there were stronger ciphers being invented in the meantime, possibly because of its relative convenience (Kahn, 1967). In 1553, Giovan Batista Belaso introduced the notion of using a passphrase as the key for a repeated

polyalphabetic cipher (Kahn, 1967). (This is the standard polyalphabetic cipher operation (Dupuis, 1999). In 1563, Giovanni Battista Porta wrote a text on ciphers, introducing the digraphic cipher. He classified ciphers as transposition, substitution and symbol substitution (use of a strange alphabet). He suggested use of synonyms and misspellings to confuse the cryptanalyst. He apparently introduced the notion of a mixed alphabet in a polyalphabetic tableau (Dupuis, 1999).

In 1585, Blaise de Vigenère wrote a book on ciphers, including the first authentic plaintext and ciphertext autokey systems (in which previous plaintext or ciphertext letters are used for the current letter's key). In 1623, Sir Francis Bacon described a cipher which now bears his name -- a biliteral cipher, known today as a 5-bit binary encoding (Singh, 1999). He advanced it as a steganographic device – by using variation in type face to carry each bit of the encoding.

In 1790, Thomas Jefferson, possibly aided by Dr. Robert Patterson (a mathematician at U. Penn.), invented his wheel cipher. This was re-invented in several forms later and used in WW-II by the US Navy as the Strip Cipher, M-138-A (Dupuis, 1999).

In 1917, William Frederick Friedman, later to be honored as the father of US cryptanalysis (and the man who coined that term), was employed as a civilian cryptanalyst (along with his wife Elizebeth) at Riverbank Laboratories and performed cryptanalysis for the US Government, which had no cryptanalytic expertise of its own. WFF went on to start a school for military cryptanalysts at Riverbank – later taking that work to Washington and leaving Riverbank (Dupuis, 1999).

1933-1945 The Enigma machine was not a commercial success but it was taken over and improved upon to become the cryptographic workhorse of Nazi Germany (Kahn, 1967). It was broken by the Polish mathematician, Marian Rejewski, based only on captured ciphertext and one list of three months worths of daily keys obtained through a spy. Continued breaks

were based on developments during the war by Alan Turing, Gordon Welchman and others at Bletchley Park in England (Dupuis, 1999).

Since the 1970s, a large number and variety of encryption, digital signature, key agreement, and other techniques have been developed in the field of public-key cryptography. The ElGamal cryptosystem (invented by Taher ElGamal) relies on the (similar, and related) difficulty of the discrete logarithm problem, as does the closely related DSA developed at the US National Security Agency (NSA) and published by NIST as a proposed standard (Kahn, 1967).

In 1976, a design by IBM based on the Lucifer cipher and with changes (including both S-box improvements and reduction of key size) by the US NSA, was chosen to be the U.S. Data Encryption Standard. It has since found worldwide acceptance, largely because it has shown itself strong against 20 years of attacks. Even some who believe it is past its useful life use it as a component e.g. 3- key triple-DES (Singh, 2000). In 1976, Diffie and Hellman (1976) published "New Directions in Cryptography", introducing the idea of public key cryptography. They also put forth the idea of authentication by powers of a one way function, now used in the S/Key challenge/response utility (Kahn, 1967).

The introduction of elliptic curve cryptography by Neal Koblitz and Victor Miller independently and simultaneously in the mid-1980s has yielded new public-key algorithms based on the discrete logarithm problem. Although mathematically more complex, elliptic curves provide smaller key sizes and faster operations for equivalent estimated security (Katz, 2007). In 1990, Xuejia Lai and James Massey in Switzerland published "A Proposal for a New Block Encryption Standard", a proposed International Data Encryption Algorithm (IDEA) -- to replace DES. IDEA uses a 128-bit key and employs operations which are convenient for general purpose computers, therefore making software implementations more efficient.

In 1991, Phil Zimmermann released his first version of PGP (Pretty Good Privacy) in response to the threat by the FBI to demand access to the clear text of the communications of citizens. PGP offered high security to the general citizen and as such could have been seen as a competitor to commercial products like Mailsafe from RSADSI (Singh, 2000). However, PGP is especially notable because it was released as freeware and has become a worldwide standard as a result while its competitors of the time remain effectively unknown.

In 1994, Professor Ron Rivest, author of the earlier RC2 and RC4 algorithms included in RSADSI's BSAFE cryptographic library, published a proposed algorithm, RC5, on the Internet. This algorithm uses data-dependent rotation as its non-linear operation and is parameterized so that the user can vary the block size, number of rounds and key length (Kahn, 1967). It is still too new to have been analyzed enough to enable one to know what parameters to use for a desired strength -- although an analysis by RSA Labs, reported at CRYPTO'95, suggests that w=32, r=12 gives strength superior to DES (Dupuis, 1999).

2.3.2 Encryption Algorithm Classification

Encryption technique can be classified into two categories (a) Symmetric Key and (b) Asymmetric Key (Singhal & Raina, 2011).

2.3.2.1 Symmetric Key

In the symmetric key encryption both for the encryption and decryption process the same key is used. Hence the secrecy of the key is maintained and it is kept private. Symmetric algorithms have the advantage of not consuming too much of computing power and it works with high speed in encryption. A block cipher is taken as the input, a key and input, and then the output block will be same in size in the symmetric key encryption. The symmetric key encryption takes place in two modes either as (i) the block ciphers or (ii) as the stream ciphers (Singhal & Raina, 2011; Jeeva *et al*., 2012). In the block cipher mode the whole data is divided into number of blocks and based on the block length the key is provided for

encryption. In the case of the stream ciphers the data is divided as small as single bits and randomized and then the encryption takes place. Symmetric key cryptosystems are much faster than the asymmetric key cryptosystems. The performance evaluation is taken place for the following symmetric key encryption techniques. DES, 3DES, IDEA, Blowfish, AES, TEA, MARS, RC6 and CAST-128 are the example of symmetric key encryption algorithm.

Types of Symmetric Key Encryption

i. DES (Data Encryption Standard): DES was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). It is based on the IBM proposed algorithm called Lucifer. DES was designed by IBM in 1972 and it was adopted by the U.S. Government as standard encryption technique in 1974 (Mandal, Parakash & Tiwari, 2012). It is a symmetric key block cipher encryption algorithm based on Feistel Network. DES uses 64 bit block of text and 56 bit key length, it performs total 16 rounds of processing to encrypt data (Shahi, 2001). In DES, the key was 64 bits but due to some restrictions from NSA (National Security Agency) IBM decided to use 56 bit key length for encryption and the remaining 8 bits is used as a parity bit for error detection, it also uses 8 boxes. DES divides the 64 bit block into two equal parts and then applies F - function on each part. F-function performs four different tasks - Expansion, Key Mixing, Substitution and Permutation. Decryption is the same process of encryption in DES.

ii. 3DES: 3DES was published in 1998 which is from DES. DES uses 56 bit key but 3DES uses 3 different keys total size of 168 bits (Halas, Bestak, Orgon & Kovac, 2012). All keys are identical or first key and third key may be same in 3DES. It also divides the text into 64 bit block and uses 8 S-boxes and performs 48 processing rounds. 3DES is more complicated and designed to protect data again different attacks. 3DES encrypts data by

applying DES encryption three times. 3DES is also approved by the U.S. Government to use because of its higher security (Mushtaque, Dhiman, Hussain & Maheshwari, 2014).

iii. CAST-128: CAST-128 (CAST 5) designed by Carlisle Adams and Stafford Tavares in 1996. It is a block cipher algorithm used in various applications. It is based on Feistel structure and performs 12 or 16 processing rounds. CAST uses 64 bit block, key length of 40-128 bit and contains the 4 S-boxes. CAST performs total 16 rounds if the key size is greater than 80 bits. To decrypt the encryption algorithm is used in reverse order (Mushtaque, 2014).

iv. MARS: MARS is a block cipher designed by IBM in 1998 and submitted to the AES and selected as one of the five finalists in AUGUST 1999. MARS is based on type-3 Feistel structure (heterogeneous structure), it uses 128 bits block, key size of 128, 192, 256 bits and a single S-box. The same algorithm is used for decryption in reverse (Mandal *et al*, 2014).

v. IDEA (International Data Encryption Algorithm): IDEA designed by Xuejia Lai and James Massey in 1991. It is also known as Improved Proposed Encryption Standard (IPES) because it is derived from Proposed Encryption Algorithm. It is symmetric key block cipher algorithm; it is based on substitution-permutation structure. It uses 64 bit block, 128 bit key and performs 8.5 rounds. In each round it performs three main operations XOR, Addition and Multiplication. Decryption is same as encryption only the key is reversed (Mushtaque *et al*., 2014).

vi. Blowfish: Blowfish is designed by Bruce Schneier in 1993. It is fast and simple block encryption algorithm used in the Secure Socket Layer and other program. Blowfish is based on Feistel Network supports 64 bit block and key size of 32- 448 bit. It contains 4 s-boxes and performs 16 processing rounds. Two main functions are performed in this

algorithm Key expansion and Data encryption. In blowfish the S-boxes are key dependent (Simar & Raman, 2011).

vii. RC6: RC6 is designed by Ron Rivest in 1998, it derived from its predecessor RC5. It is also based on Feistel structure and takes block size of 128 bits, key size 128, 192 or 256 bits and total number of processing rounds are 20. RC6 differ from RC5 because it uses 4 registers while RC5 uses 2 registers and it performs an extra multiplication operation (Mushtaque *et al*., 2014).

vii. AES (Advanced Encryption Standard): AES is a symmetric key block cipher encryption algorithm designed by Vincent Rijmen and Joan Daemen in 1998. It is based on Feistel network and support 128 bit block size and key length 128, 192 and 256 bits (Mandal *et al*, 2014). AES performs 10, 12 or 14 round and the number of rounds depends on the key. It means for 128 bit key length AES performs 10 rounds, for 192 bit key it performs 12 rounds and for 256 bit key it performs 14 rounds. In AES each round performs some steps. Key-expansion, Initial-round, Rounds and Final-rounds. In Rounds step, Sub-byte generation, Shift-rows, Mix-columns and Add-round key are performed whereas in Final-rounds step, same functions are performed except Mix-columns function (Simar & Raman, 2011).

Limitation of Symmetric Key Encryption

i. DES: DES does not provide strong security because of its key length 0f 56 bits. DES can be easily cracked by 2^56 imagination. Initially DES was accepted as the standard algorithm with strong security but after sometimes Brute force attack cracked DES.DES didn't design for software so it runs slowly. So, DES is not a secure encryption algorithm.

ii. 3DES: 3DES overcomes the problem of DES, but 3DES has also some disadvantages. 3DES performs DES operation three times (i.e. it uses 3 different keys of larger size

(3x56=168 bits)) to encrypt data so it requires almost 3 times more space than DES. Though it provides high level security in comparison to DES that's why 3DES is used by the U.S. Government.

iii. CAST 128: Using a known plain text attack Key of CAST 128 can be known by linear cryptanalysis. It can be broken by 2^{17} chosen plaintexts along with one related-key query in offline work of 2^{48} (Mushtaque *et al*., 2014).

iv. MARS: In MARS, no any significant limitation has been observed. Hardware implementation of MARS is some difficult and complex. Due to performing the function with Boolean complexity MARS is very complex to observe.

v. IDEA: Some possibilities of being attack were found in IDEA regarding minimum round version and different classes of weak keys (Mushtaque *et al*., 2014). First three rounds of IDEA algorithm is observed for related-key differential timing attacks and key-schedule attacks.

vi. Blowfish: Blowfish is a very secure algorithm but Initial 4 rounds of blowfish are observed unprotected from 2nd-order differential attack.

vii. RC6: For a class of weak keys, RC6 is analyzed that randomness is not achieved for up to 17 rounds. Otherwise it is observed that RC6 is a very secure algorithm (Mushtaque *et al*., 2014).

viii. AES: No any such kind of weakness has been observed in AES. Some initial rounds of AES are observed unprotected i.e. initial round can break by square method.

Table 1 show the result of comparison of encryption algorithms base on architecture and from figure 2.1, Mushtaque (2014) analyzed that AES is best among all these related algorithms. That the encryption performance of the AES is equal to blowfish but the memory required by

AES is less than blowfish. But on the basis of scalability it cannot be said that AES is best among all these algorithms. To become a better algorithm different parameters (architecture, scalability, security and flexibility) should be effective.

Comparison of all of this encryption algorithms base on their architecture is shown in table 2.1 while their comparison based on scalability (Memory usage and Performance) was shown in figure 2.2.

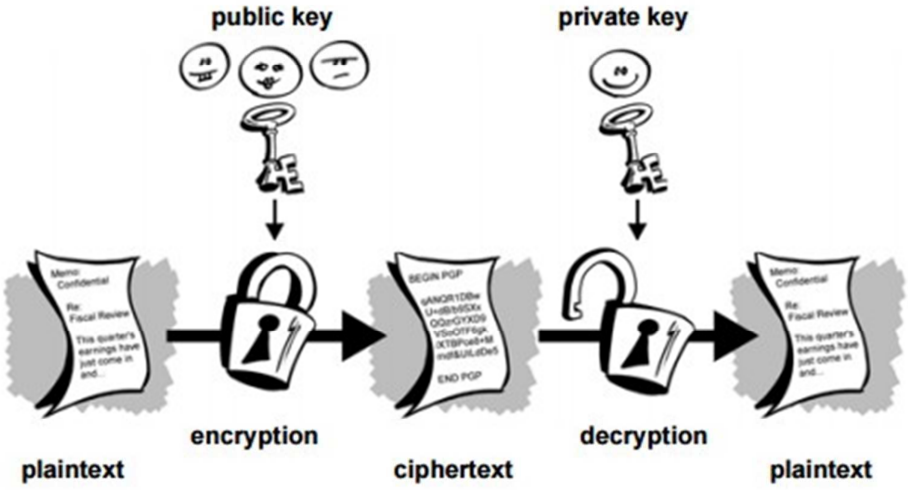

Figure 2.1: Encryption and Decryption (Mushtaque, 2014)

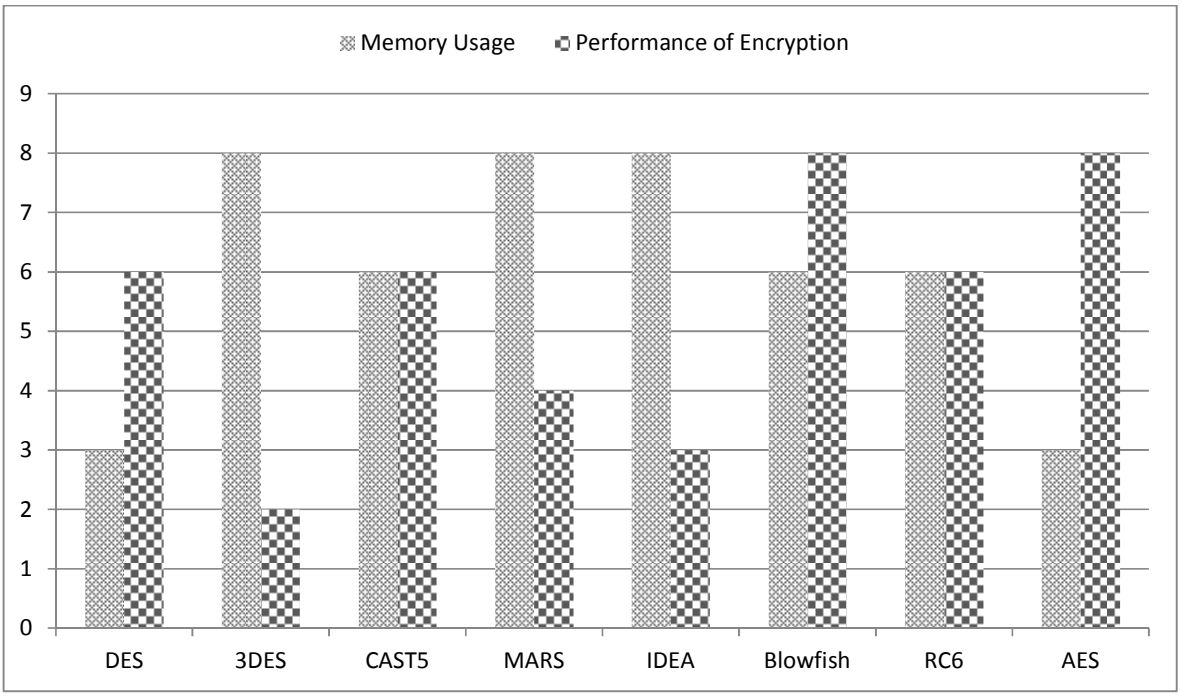

Figure 2.2: Comparison of Encryption Algorithm based on scalability [Memory usage and Performance] (Mushtaque, 2014)

Table 2.1: Comparison of Encryption Algorithms base on Architecture (Mushtaque, 2014)

Algorithms	Key Size	Block Size	Rounds	Structure	No. of S-boxes
DES	56 bits	64 bits	16	Feistel Network	8
3DES	168 bits	64 bits	48	Feistel Network	8
CAST 128/ CAST 5	40 - 128 bit	64 bits	12 or 16 (16 if key size > 80bit)	Feistel Network	4
MARS	128, 192 or 256 bits	128 bits	32	Feistel (Heterogenous Structure)	1
IDEA	128 bits	64 bits	8.5	Substitution-Permutation Structure	N/A
BLOWFISH	32 – 448 bits	64 bits	16	Feistel Network	4
RC6	128, 192 or 256 bits	128 bits	20	Feistel Network	N/A
AES	128, 192 or 256 bits	128 bits	10, 12 or 14 (depend on the size of key)	Feistel Network	1

2.3.2.2 Asymmetric Key

Asymmetric key encryption is the technique in which the keys are different for the encryption and the decryption process. They are also known as the public key encryption. One of these keys is published or public and the other is kept private. If the lock/encryption key is the one published then the system enables private communication from the public to the unlocking key's user (Yousuf & Sumer, 2011). If the unlock/decryption key is the one published then the system serves as a signature verifier of documents locked by the owner of the private key. Public key methods are important because they can be used for transmitting encryption keys or other data securely even when the both the users have no opportunity to agree on a secret key in private Algorithm (Jeeva *et al*., 2012).

The keys used in public-key encryption algorithms are usually much longer that improves the security of the data being transmitted. Asymmetric encryption algorithms need at least a 3,000-bit key to achieve the same level of security of a 128-bit symmetric algorithm. Public key encryption is biased on mathematical function, computationally intensive. RSA (Rivest-Shamir-Adleman) Algorithm and Diffie-Hellman Algorithm are types of asymmetric public key encryption (Diffie & Hellman, 1976).

2.3.3 Purpose of Encryption

In order to store or send sensitive data, the communication and the data must be secured. By using encryption, security and privacy of the data can be ensured. Even if an unauthorized user intercepts the communication channel or hack into the system and gets the sensitive data, he/she will not be able to read it without decrypting it using proper cipher and key.

2.3.4 Cipher

A cipher is an algorithm for performing encryption (and the reverse, decryption) — a series of well-defined steps that can be followed as a procedure. An alternative term is

encipherment. (Cakiroglu, 2010) The original information is known as plaintext, and the encrypted form as ciphertext. The ciphertext message contains all the information of the plaintext message, but is not in a format readable by a human or computer without the proper mechanism to decrypt it; it should resemble random gibberish to those not intended to read it. Ciphers are usually parameterized by a piece of auxiliary information, called a key. The encrypting procedure is varied depending on the key which changes the detailed operation of the algorithm. Without the key, the cipher cannot be used to encrypt, or more importantly, to decrypt. In non-technical usage, a "cipher" is the same thing as a "(secret) code" (Zotos & Litke, 2015).

2.4 Compression

Compression is the art of representing the information in a compact form rather than its original or uncompressed form (Pu, 2006) i.e. the size of a particular file can be reduced using data compression. This is very useful when processing, storing or transferring a huge file, which needs lots of resources. If the algorithms used to encrypt works properly, there should be a significant difference between the original file and the compressed file. When data compression is used in a data transmission application, speed is the primary goal. Speed of transmission depends upon the number of bits sent, the time required for the encoder to generate the coded message and the time required for the decoder to recover the original ensemble. In a data storage application, the primary concern is the degree of compression. Compression can be classified into lossy or lossless. In Lossless compression techniques the original data from the compressed file is reconstructed without any loss of data. Thus the information does not change during the compression and decompression processes. These kinds of compression algorithms are called reversible compressions since the original message is reconstructed by the decompression process. Lossless compression techniques are

used to compress medical images, text and images preserved for legal reasons, computer executable file etc. (Kodituwakku & Amarasinghe, 2011).

Lossy compression techniques reconstruct the original message with loss of some information. It is not possible to reconstruct the original message using the decoding process, and is called irreversible compression. The decompression process results an approximate reconstruction. It may be desirable, when data of some ranges which could not recognized by the human brain can be neglected. Such techniques could be used for multimedia images, video and audio to achieve more compact data compression (Kodituwakku & Amarasinghe, 2011).

2.4.1 History of Data Compression

Data compression has only played a significant role in computing since the 1970s, when the Internet was becoming more popular and the Lempel-Ziv algorithms were invented, but it has a much longer history outside of computing. Morse code, invented in 1838, is the earliest instance of data compression in that the most common letters in the English language such as "e" and "t" are given shorter Morse codes. Later, as mainframe computers were starting to take hold in 1949, Claude Shannon and Robert Fano invented Shannon-Fano coding. Their algorithm assigns codes to symbols in a given block of data based on the probability of the symbol occurring. The probability is of a symbol occurring is inversely proportional to the length of the code, resulting in a shorter way to represent the data (Almelkar & Gandhe, 2014).

Early implementations of Shannon-Fano and Huffman coding were done using hardware and hardcoded codes. It was not until the 1970s and the advent of the Internet and online storage that software compression was implemented that Huffman codes were dynamically generated based on the input data. Later, in 1977, Abraham Lempel and Jacob Ziv published their groundbreaking LZ77 algorithm, the first algorithm to use a dictionary to compress data.

More specifically, LZ77 used a dynamic dictionary oftentimes called a sliding window. In 1978, the same duo published their LZ78 algorithm which also uses a dictionary; unlike LZ77, this algorithm parses the input data and generates a static dictionary rather than generating it dynamically.

2.4.2 Compression Algorithms

i. Run Length Encoding Algorithm: Run Length Encoding (RLE) is the simplest of the data compression algorithms. The consecutive sequences of symbols are identified as runs and the others are identified as non runs in this algorithm. This algorithm deals with some sort of redundancy (Pai, Cheng, Lu & Ruan, 2012). It checks whether there are any repeating symbols or not, and is based on those redundancies and their lengths. Consecutive recurrent symbols are identified as runs and all the other sequences are considered as non-runs. For an example, the text "ABABBBBC" is considered as a source to compress, then the first 3 letters are considered as a non-run with length 3, and the next 4 letters are considered as a run with length 4 since there is a repetition of symbol B (Kodituwakku & Amarasinghe, 2011). The major task of this algorithm is to identify the runs of the source file, and to record the symbol and the length of each run. The Run Length Encoding algorithm uses those runs to compress the original source file while keeping all the non-runs without using for the compression process.

ii. Huffman Encoding Algorithm: Huffman Encoding Algorithms use the probability distribution of the alphabet of the source to develop the code words for symbols. The frequency distribution of all the characters of the source is calculated in order to calculate the probability distribution (Almelkar & Gandhe, 2014). According to the probabilities, the code words are assigned. Shorter code words for higher probabilities and longer code words for smaller probabilities are assigned. For this task a binary tree is created using the symbols as

leaves according to their probabilities and paths of those are taken as the code words. Two families of Huffman Encoding have been proposed: Static Huffman Algorithms and Adaptive Huffman Algorithms.

Static Huffman Algorithms calculate the frequencies first and then generate a common tree for both the compression and decompression processes. Details of this tree should be saved or transferred with the compressed file. The Adaptive Huffman algorithms develop the tree while calculating the frequencies and there will be two trees in both the processes. In this approach, a tree is generated with the flag symbol in the beginning and is updated as the next symbol is read.

iii. The Shannon Fano Algorithm: Shannon Fano coding named after Claude Elwood Shannon and Robert Fano, is a technique for constructing a prefix code based on a set of symbols and their probabilities (Almelkar & Gandhe, 2014). But it does not achieve the lowest possible expected code word length like Huffman coding. it is another variant of Static Huffman Coding algorithm (Kodituwakku & Amarasinghe, 2011). The only difference is in the creation of the code word. All the other processes are equivalent to Huffman Encoding Algorithm (Pai *et al.*, 2012).

iv. Arithmetic Encoding: In this method, a code word is not used to represent a symbol of the text. Instead it uses a fraction to represent the entire source message. The occurrence probabilities and the cumulative probabilities of a set of symbols in the source message are taken into account. The cumulative probability range is used in both compression and decompression processes (Almelkar & Gandhe, 2014).

In the encoding process, the cumulative probabilities are calculated and the range is created in the beginning. While reading the source character by character, the corresponding range of the character within the cumulative probability range is selected. Then the selected ranges are

divided into sub parts according to the probabilities of the alphabet (Pai *et al.*, 2012). Then the next character is read and the corresponding sub range is selected. In this way, characters are read repeatedly until the end of the message is encountered. Finally a number should be taken from the final sub range as the output of the encoding process. This will be a fraction in that sub range. Therefore, the entire source message can be represented using a fraction. To decode the encoded message, the number of characters of the source message and the probability/frequency distribution are needed.

v. The Lempel Zev Welch (LZW) Algorithm: Dictionary based compression algorithms are based on a dictionary instead of a statistical model (Pai *et al.*, 2012). A dictionary is a set of possible words of a language, and is stored in a table like structure and used the indexes of entries to represent larger and repeating dictionary words. LZW algorithm is one of such algorithms. In this method, a dictionary is used to store and index the previously seen string patterns. In the compression process, those index values are used instead of repeating string patterns (Kodituwakku & Amarasinghe, 2011). The dictionary is created dynamically in the compression process and no need to transfer it with the encoded message for decompressing. In the decompression process, the same dictionary is created dynamically. Therefore, this algorithm is an adaptive compression algorithm.

2.5 Waterfall model

The waterfall model was first presented by Winston W. Royce in 1970 as a flawed and non-working model. Despite this it became a very popular model in the world of software development because of its various advantages towards software designing and implementation. The defining aspect of the waterfall model is that none of the stages can be started with if the previous stage hasn't been completed. The original waterfall model consisted of the following seven stages:

i. Specification of Requirements

ii. Design

iii. Construction

iv. Integration

v. Testing and Debugging

vi. Installation

vii. Maintenance

2.5.1 Advantages of the Waterfall Model

The waterfall model is the oldest and most widely used model in the field of software development. This model has certain advantages that have made it so popular over the years. The advantages of this model are:

i. It is a linear model and linear models are the most simple to be implemented.

ii. The amount of resources required to implement this model is very minimal.

iii. After every stage of the waterfall model development, documentation is produced. This makes it easier to understand the product designing procedure.

iv. After every major stage of software coding, testing is done to check the correct running of the code.

2.5.2 Disadvantages of the Waterfall Model

i. The biggest disadvantage of the waterfall model just happens to be one of its greatest advantages. It is not possible to go back to previous phases. If the something in a previous phase has gone wrong, things can get very complicated in the present phase.

ii. It happens quite often that the client is not very clear of what he exactly wants from the software. Any changes that he mentions in between may cause a lot of confusion.

iii. Small changes or errors that arise in the completed software may cause a lot of problem.

iv. A working model of the software does not lie in the hands of the client before the final stage of the development cycle is completed.

The waterfall model is the most widely used model in software development projects. This model has many disadvantages but just as many, if not more, an advantage that ensures that it remains one of the most popular models used in the field of software development to date. There are many versions of this model that minimized the disadvantages of this model.

2.5.3 The Modified Waterfall Model

The modified waterfall model is closely based on the waterfall model. The reason for its existence is to minimize or erase the defects or disadvantages of the traditional waterfall model. The main change of this model, compared to the waterfall model, is that the phases in the modified waterfall model life cycle are permitted to overlap. This makes this model a lot more flexible to work with. It also makes it possible for a number of tasks to function concurrently, which ensures that the defects in the software are removed in the development stage itself and the added costs of making changes to the software before implementation is saved. Because there can be a number of phases active at one point of time, making changes to the design and rectifying errors introduced can be easily dealt with.

To every phase of the modified waterfall model diagram, a verification and validation step has been added. Another advantage of the modified waterfall model is that it takes a less formal approach to procedures, documents and reviews. Because of this, it reduces the huge bundle of documents. Due to this the developer has more time to devote to work on the code and does not have to bother about the procedures. This in turn helps to finish the product faster.

There are not only advantages to the modified waterfall model. This model also has a drawback. Because of its flexible nature, tracking the progress on all the stages becomes a difficult task. Also, this model still uses the stages from the traditional waterfall model and the dependency between stages still exists. This dependency can cause complications during the software development process. The development team may be tempted to move back and forth between the stages for fine tuning. This results in delay in project completion. This problem however, can be solved by setting up certain benchmarks for every phase. This helps to ensure the project is on schedule and does not go haywire. The modified waterfall model is extensively used in the software industries. This model still has all the advantages of the traditional waterfall model without the drawbacks and this has made it easier to work in the advanced stages.

2.6 Review of Related Existing Systems

a. HyperOffice: Whether it is SharePoint Server 2010 or its earlier versions, the costs and complexities of SharePoint are sometimes too much for a small to mid-sized company to bear. Even the terminology of SharePoint is enough to confuse users that do not have much previous knowledge of the platform. Some of these are: SharePoint Foundation, Server, Service e.t.c. That is why something like HyperOffice was developed, to provide features relevant to the company at hand, packaged as easy to use cloud based solution. Even though it is far from as complex and feature rich as SharePoint, it still provides quite a few features that most companies want, such as document management, intranet software, online project management, social collaboration etc. The selling point of their platform is: “There is no hardware to install, no software to download, no experts to hire. Just get online and get started (Hyperoffice, 2015).

b. Doccept: Doccept is a Document management application which offers functional depth in terms of document management as both a stand-alone program and one that integrates seamlessly through its extensive integration features with other applications within an organization.

It strength lies in its strong emphasis on storage and document life-cycle management including document expiration and fast document retrieval. It weakness lies in the absence of encryption and compression capacity in its documentation process. These weaknesses can be overcomes by upgrading the system and adding an encryption and compression mechanism to the present system (Doccept, 2015).

c. M-Files: M-Files are document management application that offers mobile apps for iPhones, iPads and Android and Windows smartphones. With the mobile apps you can view, review and approve stored documents, as well as make electronic signatures on them. The app also lets you take photos of a document or receipt, and save it to your vault automatically.

It can also be fully integrates with Microsoft Office. With a click of a button, you can save any Word, Excel or PowerPoint document directly to your M-Files vault. The strength of the work is that the system allows you to set restrictions based on individual users or groups of employees, this gives users the freedom to store any and everything in the system without worrying that prying eyes will see something they're not supposed to. The biggest downside of M-Files is the cost involved in using the system and another downside of M-Files is that it isn't compatible with Mac computers, which implies that if a company has invested in Mac desktops and laptops, this system is not a fit for such company (M-files, 2015).

d. PinPoint: PinPoint is a Document management system that uses a traditional file cabinet and folder approach. With PinPoint user can create as many cabinets, and folders within

them, as user likes. When users add documents into the system, you have the option to import them from either a computer or a scanner. You can also drag and drop files directly into the PinPoint system. The system allows you to set security rights at both the cabinet and document levels. You can also give different users access to certain folders and documents based on their roles. All of these features help to ensure that people who are not authorized to access certain documents cannot do so. A slight knock on the system is that the system doesn't currently have its own mobile application this could be a problem for those who prefer apps to mobile websites. Also, security emphasis was limited to privacy and level restriction which does not make document totally secure (Pin-point, 2015).

e. LogicalDOC: The system operates with an easy-to-understand document filing structure. You can create as many cabinets, and folders within them, as you like. When adding new documents into the system, you can upload them from your computer, drag and drop files or scan documents in. The important features of the system are the check-in and -out tool. This allows multiple users to work on a document without worrying that edits will be lost or overwritten. Users can view the status of documents and see if it's currently being worked on, and by whom. When a document is in the "checkout" status, it's always available for read-only operations like search and browsing. Like PinPoint security emphasis was limited to privacy and level restriction which does not make document totally secure (Logicaldoc, 2015).

2.7 Review of Related Research Works

Kodmelwar, Mayur, Ajinkya, Ashwini and Munmun (2012) in the journal Document Management System with Enhanced Security designed and implemented a document management system for a small to medium scale organization. The paper acknowledges the fact that merely having a simple storage/retrieval system was not enough, and hence we have stressed upon some features to enhance the basic DMS that are very useful in terms of

security, optimal disk usage, level of abstraction and productivity of the employees within an organization.

In the used Architecture, client machines are connected to the central server in order to perform different functionality for "EDMS" (Enhanced Document Management System). The client's machine may be connected through wireless connection or wired connection to the server. The central server contain different module such as web server, database, third party server and secondary storage. The requirements for the system were based on the findings of the literature review done as well as from the interview sessions done. After a careful analysis of the data collected, the findings of the analysis was used to derive the application requirements (i.e. The Client Application must be a desktop application that must have web browser integration, The Server Application would be a combination of a DBMS, a webserver and a third party server (to generate RSA keys) and The Client and the Server Machines must be connected via the 802.3 or 802.11 standards.

The strength of their work was that their work has an encrypted data transfer feature which allows the client machines to send their documents over to the main server for storage purposes in the organization. The technique of Digital enveloping was used for this feature. The Document(s) chosen for the transfer are themselves encrypted using a symmetric encryption and the symmetric key itself shall then be encrypted by RSA encryption technique which was an asymmetric key algorithm. The weakness of the work is that the encryption used in the work was designed for a plain text which can't be used for other file format i.e. pdf, excel etc.

Park and Kim (2010) in their work "Design and Implementation of E-Document Encryption System using Hash Algorithm" was aimed at making some observations on the current research knowledge about the introduction of EDM systems in the construction industry. The algorithm the study suggested first extends the original picture that was used in

electronic identification card into hash function using the encrypt key and then rearranges pixel of each image with the created value through scrambling algorithm. The suggested system separates the original image using block rearranging algorithm and unique key with scrambling method, rearranges it, and inserts the distorted image to the smart chip of identification card. Forge or falsification status can be confirmed by extracting the distorted image in the smart chip of identification card using an image extractor and comparing it to the original image using block inverse arrangement algorithm and unique key and check if it matches the original image. Hash algorithm has slower performance speed than AES algorithm but was simply only the part to create a key, and as other calculation conducts XOR so it obtains faster execution value in the aspect of speed. In case of encrypting the face area, it showed more improved speed as unnecessary parts were not encrypted. Moreover, the speed can be improved in cases of portrait pictures such as ID picture if only parts including information like face are encrypted, not the entire area including background

The strength of their work is that the Encryption method used in the work show improvement in the speed of the suggested method becomes improved and better than that of the symmetric-key algorithm in case only a certain part, like a face of a person, is encrypted. The data used in the test was an image including a face, and when about 40% of the face part was included in the encryption, the speed of the suggested method became better by roughly 40% than that of the AES algorithm The weakness of their work is that a little bit of noise condition occurred in loss compression like JPEG. This is because it is saved by using similarity level about adjacent color in JPEG compression process and more study about this is needed. Also the encryption techniques are targeted at image file format only.

Akashah, Rizal, Jusoff and Christon (2011) in their work "Electronic Document Management System" which was aimed at developing a framework of EDMS that was tailor-made to the SCM department that could also be used in developing the EDMS for

future research. Data of the current workflow of SCM department was gathered using interview and discussion approach. The purpose of the interview will be ton further understand the challenges the department faces in managing it documents. The interview was also aimed at finding the more detailed information that will translate into the form of the desired system. The main focus of the proposed EDMS was to facilitate the storage and extraction of documents from a database that relate to the processes of producing a contract between the organization and its vendors/suppliers. If put in a simplistic form, there are approximately four processes, tender plan preparation, sending bid invitations, bid evaluation and award recommendation.

Much emphasis was put on streamlining processes that the document management system was implement and the actual business processes of creating a typical SCM contract that the organization does on a regular basis. EDMS was definitely the best solution adopted by SCM department in this particular organization. Besides to facilitate the retrieval of the document in the department, it could provide a secure place to store the documents compared to the traditional filing system. Thorough research was carried out while gathering the system requirements to ensure that the developed framework was tally with the business requirements of the said department.

The strength of the work is that the proposed system could be enhanced to a Web-based database system allowing all input screens and forms of the system to be made online. While it weakness is that the work restrict it discussion only to certain important GUIs of the system focus on the GUIs through which an instructor interacts with the eCourse File Management System(it focus on the GUIs through which an instructor interacts with the eCourse File). Furthermore, the technical details of the database system are also not presented.

Groenewald (2004) in the paper "Symmetric Algorithm Survey: A Comparative Analysis" proposed and EDMS to manage and control all electronic documentation – whether word processing documents, spreadsheets, presentations, graphics or e-mail messages through their life cycle. For document security to be achieved he used version control, audit trails for each document there by controlling access to documents via various security levels. The work fails to include any encryption techniques to protect sensitive information and it fails to include a mean of managing the limited storage space. But the EDMS still was capable of controlling duplication.

Mansoor, Shujaat and Umer (2013) performed a detailed analysis on symmetric block encryption algorithms base on different parameters, in their work "**Symmetric Algorithm Survey: A Comparative Analysis**", analysis was done on most of the popular symmetric key algorithms (DES, Triple-DES, IDEA, Blowfish, TEA, CAST5, AES) in terms of Authentication, Flexibility, Reliability, Robustness, Scalability and Security which enable them to highlight the major weakness of the said algorithms, the paper also make each algorithm's strength and limitation transparent for application. During the analysis they observed that AES (Rijndael) was the best among all in terms of Security, Flexibility, Memory usage, and Encryption performance. Although the other algorithms were also competent but most of them have a tradeoff between memory usage and encryption performance with few algorithms been compromised.

JFSS (Java File Security System) is the design work Kahanwal, Dua & Singh (2012). The work "JFSS" is a java based file management system with enhanced security feature which uses cryptography as it form of encrypting files by providing a transparent UNIX file system interface to directory hierarchies that are automatically encrypted with a user supplied

keys. They achieved a high security by including the support of the Rijndeal Algorithm (AES) and saved the keys on the portable smart cards for the documents which are important. The strength of the JFSS is its ability to encrypt and process file in no time. It weaknesses is the lack of compression mechanism and it can't work on all platform except java enable device. This can be overcome by upgrading the system with compression mechanism using the java compression library.

Anwar and Naseer (2013) proposed **an eCourse file management system** to shift from the method of compiling paper-based course files to a more versatile method of compiling and maintaining electronic course files. The architecture of the work consists of three layers; database layer, system's module layer, GUI layer. The database layer stores the information of faculty, courses taught in a semester and the electronic versions of all course related documents such as syllabus, assignments students' worked sample. The systems module layer facilitates in storing and retrieving all data files stored in the database. The GUI allows various users to interact with the system and perform their specific task. The system offers many advantages such as ease of access, information retrieval, allaying the storage and disposal issues of paper-based files. The system also offers tremendous benefit in saving paper and printing costs, reducing the human and financial resources needed for compiling course file, minimizing the negative environmental impact, saving natural resources for future generations, and contributing towards a green sustainable environment. This system may be utilized for monitoring the progress of courses though out the duration of course offerings. The proposed system could be enhanced to a Web-based database system allowing all input screens and forms of the system to be made online. But it does not include any encryption and compression technique, which implies that all files are not secure while management of space is not achieved at the highest possible level.

Kodituwakku and Amarasinghe (2011) experimented on **the comparison of a number of different lossless compression algorithms for text data**. Several existing lossless compression methods are compared for their effectiveness. Test was carried out on different type of files; the main interest was on different test patterns. By considering the compression times, decompression times and saving percentages of all the algorithms, the Shannon Fano algorithm was considered as the most efficient algorithm among the selected ones since it shows better results for the large files and the values of the algorithm are in an acceptable range.

Mushtaque, Dhiman, Hussain and Maheshwari (2014) in the paper "Evaluation of DES, TDES, AES, Blowfish and Twofish Encryption Algorithm Based on Space Complexity" evaluate different types of encryption based on space complexity. The result of their work showed that TDES was better than all these algorithms. DES takes less space than TDES but DES is not a secure algorithm because 2^56 imagination brute force attack can crack this algorithm. TDES is strong algorithm but it also takes almost two times more space than DES. They concluded to design an algorithm with minimum space complexity will be a challenge.

In the Paper title "**Data encryption decryption using pentaoctagesimal SNS (strange number system)" Das, Lanjewar and Sharma (2013)** used their algorithm an encoding converter in text files. The most prominent feature of strange number system is its full fleshed Cryptography that provides techniques of encryption and decryption while hiding all the technical details. Data encryption with pentaoctagesimal SNS was a base conversion routine, symbol remapping, and a dynamic algorithm was the only encryption algorithm that is as secure as one-time pad. The strength of the work can be attributed to the fact that it provides an excellent data encryption and decryption technique to increases the data security and transfer rate during data communication.

Kattan (2006) researched on "Universal lossless compression technique with built in encryption". In his work he proposed a new universal lossless compression technique with a built-in encryption by combining each of the static Huffman tree algorithm which can be used for text compression and 4-variables k-map technique which is used for logic digital minimization. The proposed algorithm consists of four steps: Firstly compress a file, secondly encrypt it, thirdly decrypt it, and finally decompress it back identically to the original file. In the work, a new data set to measure the performance was proposed for the compression techniques called "Probability corps". The results show that the major factor which affects the performance of the proposed compression technique ratio is the density of the binary streams for the data. The work showed an acceptable compression ratio for both of the ASCII and the Unicode files; however, it was not as effective as the other existing techniques. The experiments showed encouraging results for the JPG files which was better than many other techniques.

2.8 Summary of Limitation of the Reviewed Research Works

The review of research works shows there are still gaps in the existence of document management system with respect to document security, privacy, space management and dynamism in accessibility. It shows that most encryption, decryption and compression technology used in the reviewed works are targeted at plaintext and image files. Also the review shows that java based encryption, decryption and compression algorithms was used which mean device to be used for the system must be java enable before it can communicate with the system.

Hence the needs for Encodoc that will bridge these gaps that exist in document management system because it will be able to work on different document types (e.g Ms word, Portable document format (PDF), spread sheets, Portable network graphics (PNG) etc.) and any device with a browser can access encodoc, which will make its accessibility dynamic.

CHAPTER THREE

METHODOLOGY

This chapter describes the method that was employed in the course of this work. The tools used, design consideration, data collection, analysis and design architecture are stated. How the database was designed, how encryption and compression were achieved was included.

3.1 Design Model Used

A Modified Waterfall model was used in this work. Instead of the six major phases of the waterfall model, it was reduced into 5 different phases.

These phases are:

a. Analysis

b. Tool Evaluation & Selection

c. Design

d. Implementation

e. Post-Implementation

This approach is a more straight forward waterfall model. It starts with an analysis. Once this phase is completed the input can be used in the next phase, the tool evaluation and selection phase. Here is where this model differs from the waterfall model because the design phase will be implemented differently than is custom and there is no need for a develop phase, since the product will be an existing one. The design phase will also be more about the integration of the product with existing infrastructure and how the system will be filled in, instead of the design of the system itself.

a. Analysis phase

During the analysis phase, the high level needs, goals and objectives of the improved EDMS will be determined and the requirements of the improved EDMS for Oyo state Housing Corporation will be gathered. In this phase the needs for the creation of the improved EDMS

and management of Oyo state Housing Corporation will be made clear. During this phase, interviews, Observations and collection of relevant documents were done. At the end of this phase a complete functional requirements specification, which outlines all facets of the system were produced. A series of profiles, which will document the information gathered, which supports not only the functional requirements specification but also the tool selection and design phase will also be produced. The profiles to be created are:

a. Organization profile

b. Document profile(s)

c. User profile(s)

b. Tool evaluation & Selection phase

During this phase an evaluation will be made of the available tools currently on the market against the requirements acquired during the analysis phase. Here technical capabilities, platform support, open architecture, cost and the level of integration required between individual applications will be considered. This will then be used to aid in the decision on what will be done to fill the gaps. This could be, finding of add-on applications, designing custom ones, or leaving the requirement unfulfilled.

c. Design phase

In the design phase, requirements for the hardware needed to support the applications will be made. In this phase a lay out of the overall architecture and design of the system will be made. The design lays out the major pieces or functions of the system and how they will work together to meet the objectives of the system. This includes how information or documents will flow through the system, and where processes will be automated or manually accomplished. It will also lay out the details of how each piece of the system or task will be accomplished.

d. Implementation phase

In this phase the achievement lies with a completely installed and tested infrastructure and a document that details the installation configuration. It will also include a set of tools that make up the system. This phase is where the system becomes real. The users will be trained on the system. Document loading and conversion begins here and will probably take a couple of days or even weeks.

e. Post-implementation phase

In this phase a look was taken into the success of the implemented system. This was done by measuring results, including user perceptions and business metrics. From this phase important lessons were learnt and can be used as future reference for other projects.

3.2 Tools Used

The tool used for this work are discussed below

3.2.1 PHP

(Recursive backronym for "Hypertext Preprocessor"): a server-side scripting language designed for web development but also used as a general-purpose programming language. It serves as the back-end which will interact with the database, server and applets. PHP is at the forefront of Web 2.0 and Service Oriented Architectures enabler technologies along with other open source projects MySQL, Apache or JBoss. With ten years of development behind it, PHP is a relatively young programming language. Nevertheless, millions of developers world-wide use PHP to develop systems that power over 20 million websites.

Today, PHP is a full featured comprehensive programming language with solid object orientation support. While it was called a scripting language in the past, today it is more referred to as a dynamic programming language. Unlike traditional programming languages such as C/C++, PHP does not have to be compiled. Instead the source code is interpreted at

runtime. Another benefit of PHP is flexibility. Since no compilation is needed, it is easy to make changes or bug fixes within minutes and to deploy new versions of the program frequently. Additionally, it is easy to prototype new applications and concepts; typically compared to C++ or Java, PHP application development takes 50% of the time.

3.2.2 JavaScript

A dynamic computer programming language used as part of Web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

JavaScript has become one of the most popular programming languages on the Web. Initially, however, many professional programmers denigrated the language because its target audience consisted of Web authors and other such "amateurs", among other reasons. The advent of Ajax returned JavaScript to the spotlight and brought more professional programming attention. The result was a proliferation of comprehensive frameworks and libraries, improved JavaScript programming practices, and increased usage of JavaScript outside Web browsers, as seen by the proliferation of server-side JavaScript platforms.

3.2.3 HTML

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. It is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). It acts as the front-end interface which directly interacts with the user. Web browsers can read HTML files and compose them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to

interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML.

3.2.4 MYSQL

Named MYSQL officially but also called "My Sequel" as of July 2013, the world's second most widely used relational database management system (RDBMS) and most widely used open source RDBMS. It a relational database management system (RDBMS) with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. It serves as the store house for all information concerning the EDMS. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP (Linux, Apache, MySQL, and Perl/PHP/Python) or WAMP (Windows, Apache, MySQL, Perl/PHP/Python) open source web application software stack.

3.2.5 Apache HTTP Server

Apache HTTP Server, colloquially called Apache, is the world's most widely used web server software. Apache supports a variety of features, many implemented as compiled modules

which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP. Popular authentication modules include mod_access, mod_auth, mod_digest, and mod_auth_digest, the successor to mod_digest. Popular compression methods on Apache include the external extension module, mod_gzip, implemented to help with reduction of the size (weight) of Web pages served over HTTP. ModSecurity is an open source intrusion detection and prevention engine for Web applications. Apache logs can be analyzed through a Web browser using free scripts, such as AWStats/W3Perl or Visitors. Virtual hosting allows one Apache installation to serve many different Web sites. For example, one machine with one Apache installation could simultaneously serve www.example.com, www.example.org, test47.test-server.example.edu, etc.

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs). It supports password authentication and digital certificate authentication. Anyone can adapt the server for specific needs because the source code is freely available, and there is a large public library of Apache add-ons. Instead of implementing a single architecture, Apache provides a variety of Multiprocessing Modules (MPMs), which allow Apache to run in a process-based, hybrid (process and thread) or event-hybrid mode, to better match the demands of each particular infrastructure. This implies that the choice of correct MPM and the correct configuration is important. Where compromises in performance need to be made, the design of Apache is to reduce latency and increase throughput, relative to simply handling more requests, thus ensuring consistent and reliable processing of requests within reasonable time-frames.

3.2.6 WAMP

WAMP is an acronym for Windows/Apache/MySQL/PHP, Python, (and/or) PERL. WAMP refers to a set of free (open source) applications, combined with Microsoft Windows, which are commonly used in Web server environments. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is called a server stack. In this stack, Microsoft Windows is the operating system (OS), Apache is the Web server, MySQL handles the database components, while PHP, Python, or PERL represents the dynamic scripting languages.

3.2.7 Dreamweaver

Adobe Dreamweaver is a web design and development application that provides a visual WYSIWYG editor (colloquially referred to as the Design view) and a code editor with standard features such as syntax highlighting, code completion, and code collapsing as well as more sophisticated features such as real-time syntax checking and code introspection for generating code hints to assist the user in writing code. The Design view facilitates rapid layout design and code generation as it allows users to create and manipulate the layout of HTML elements. Dreamweaver features an integrated browser for previewing developed webpages in the program's own preview pane in addition to allowing content to be open in locally installed web browsers. It provides transfer and synchronization features, the ability to find and replace lines of text or code by search terms or regular expressions across the entire site, and a templating feature that allows single-source update of shared code and layout across entire sites without server-side includes or scripting.

For the research work Dreamweaver will be used because of it real-time syntax checking, code introspection for generating code and source formatting. Also since version 5, Dreamweaver supports syntax highlighting for most languages (ActionScript, Active Server

Pages (ASP), C#, Cascading Style Sheets (CSS), ColdFusion, EDML, Extensible HyperText Markup Language (XHTML), Extensible Markup Language (XML), Extensible Stylesheet Language Transformations (XSLT), HyperText Markup Language (HTML), Java, Javascript, Hypertext Preprocessor (PHP), Visual Basic (VB), Visual Basic Script Edition (VBScript), Wireless Markup Language (WML) etc.).

3.3 Design Architecture

3-tier system architecture as shown on figure 3.1 was used for this work

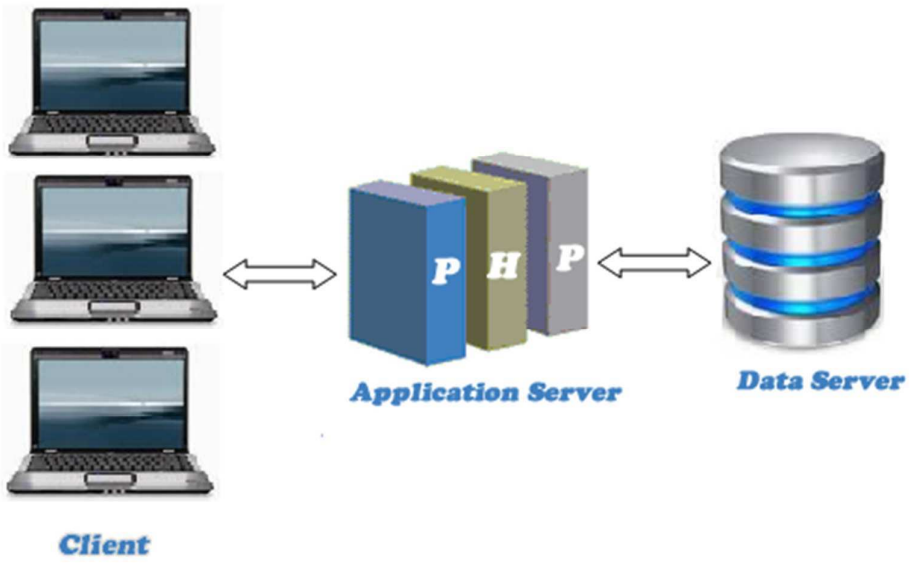

Figure 3.1: Three Tier Architecture (Alessandro, 2004).

3.3.1 The 3-tier Architecture

In this architecture, there are 3 main elements:

a. The client tier, that is responsible for the presentation of data, receiving user elements and controlling the user interface. Static or dynamically generated content rendered by the browser (front-end).

b. The data server tier, that is responsible for data storage. A database, comprising both data sets and the database management system or RDBMS software that

manages and provides access to the data (back-end). For this work, we used MySQL.

c. The application server tier, that is responsible for the business logic of the system. In fact, business-objects that implements the business rules "live" here, and are available to the client-tier. Can also be refer to a dynamic content processing and generation level e.g., Java EE, ASP.NET, PHP, ColdFusion platform (middleware). This tier protects the data from direct access by the clients. For this work, we used PHP.

Advantages of Three Tier Architecture

a. High performance, lightweight persistent objects
b. Scalability: Each tier can scale horizontally.
c. Performance – Because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers.
d. High degree of flexibility in deployment platform and configuration.
e. Better Re-use.
f. Improve Data Integrity.
g. Improved Security – Client is not direct access to database.
h. Easy to maintain and modification is bit easy, won't affect other modules.
i. In three tier architecture application performance is good.

3.3.1.1 The Client Tier

The client tier was design using HTML, CSS, JavaScript, JQuery XHTML generated from PHP. This is shown in the appendix.

3.3.1.2 The Data Server Tier

For the robust database for this work we used MySQL that is an open-source relational database. A relational database is a type of database management system (DBMS) that stores data in the form of related tables. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. The most common options when choosing a DBMS are Oracle, MySQL and PostgreSQL. Table 3.1 showed the summary of the pros and cons of these database systems: The main difference between these three systems is that MySQL and PostgreSQL are open source while Oracle is not. Oracle offers more advanced functionality than the other two systems: it is the fastest, supports transactions (sets of basic operations considered as single operations) and has enterprise-level data protection and distribution capabilities, such as full-scale clustered replication. On the other hand, of all open-source database solutions available, MySQL is the most impressive as it requires low machine requirement and easy to setup.

Table 3.1: Table summarizing the pros and cons of the major database system (Alessandro, 2004).

	Oracle	**MySQL**	**PostreSQL**
Pros	- Fastest commercial DBMS -Writers never blocks readers - Transactions, rollbacks and subselects support.	- Open source - Low machine requirements - Easy to setup	- Open source - Easy to administer -Transactions and rollbacks support
Cons	- Not open source (Oracle licenses are expensive)	- No transactions, rollbacks and subselects	- No support to fault tolerant installations.

MySQL is also used because of the following features ; Cross-platform support, Stored procedures, using a procedural language that closely adheres to SQL/PSM, Triggers, Cursors, Updatable views, Online DDL when using the InnoDB Storage Engine, Information schema, Performance Schema, Built-in Replication support, SSL support, Query caching, Full-text indexing and searching, Embedded database library, Unicode support, Partitioned tables with pruning of partitions in optimizer, Shared-nothing clustering through MySQL Cluster, Multiple storage engines, allowing one to choose the one that is most effective for each table in the application, Native storage engines InnoDB, MyISAM, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, NDB cluster and it also has a set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.

The database of the system "encodoc_db" will have the following table;

a. users table: this table was used to store information about the user of the system as shown in Table 3.2.

b. documents table: this table was used to store meta-data and encryption and decryption information about different types of documents in the system as shown in Table 3.4.
c. userlog: this table will store information of all activity perform by users as shown in Table 3.5.
d. newusers table: this table was used to store information about first time users and will be used to hold an OTP (One-Time Password) as shown in Table 3.6.
e. Config Table: this table was used to store information about the general configuration of the system.
f. allowdocumenttype table: this table will be used to store information about the main page (e.g. logo, images, title, favicon etc.) as shown in Table 3.3

The purpose of the database is to store the required information about documents and their references and to keep record of paper activities. Information stored in the database includes documents' reference information (authors, type, upload date and time, size etc.)

Table 3.2: Table showing Structure of users table.

Column	Type	Null	Default	Column
Id	int(15)	No		Primary
uName	varchar(30)	No		
uPass	varchar(30)	No		
uEncrypt	varchar(100)	No		
uLname	varchar(30)	No		
uOname	varchar(50)	No		
uPhone	varchar(15)	No		
uStatus	varchar(30)	No		
uLevel	varchar(30)	No		
uDept	varchar(30)	No		

Table 3.3: Table showing the structure of Allowdocumenttype table

Column	Type	Null	Default	Column
Id	int(8)	No		Primary
Extension	varchar(5)	No		
description	varchar(150)	No		
Status	varchar(15)	No		
Image	varchar(150)	No		

Table 3.4: Table showing the structure of Documents

Column	Type	Null	Default	Column
Id	int(15)	No		Primary
dName	varchar(30)	No		
dTitle	varchar(80)	No		
dOwner	int(15)	No		
dSize	varchar(15)	No		
dType	varchar(10)	No		
dAccess	varchar(15)	No		
dDate	date	No		
dTime	time	No		
dDept	varchar(30)	No		

Table 3.5: Table showing the structure of userlog table

Column	Type	Null	Default	Column
Id	int(30)	No		primary
usId	int(15)	No		
usAction	varchar(300)	No		
usDate	Date	No		
usTime	time	No		

Table 3.6: Table showing the structure of newusers table

Column	Type	Null	Default	Column
Id	int(15)	No		primary
Otp	varchar(50)	No		
Used	varchar(1)	No		
nDept	varchar(30)	No		
nName	varchar(30)	No		

Advantages and Limitations of Database Management

Information may be stored in databases that contain either elements of data or entire documents stored in digital form. Significant potential advantages of database management systems for records management are:

i. Faster access to information.
ii. Centralization of information.
iii. Flexibility of information retrieval.
iv. Reduction in miss-filing.

Some limitations of database management systems are:

i. Cost of the necessary equipment and software.
ii. Need for additional expertise to administer and operate the electronic system.
iii. Cost of maintaining duplicate systems (in many situations) when electronic files, because of legal or historical requirements, cannot replace paper or microform documents.

For this work the limitation of data management has being overcomes by using a widely available open source RDBMS and fully automating the activity of the database.

3.3.1.3 Application Server Design

The work is designed in two stages making use of appropriate modeling tools to describe the operation of the Improve Document Management System in details. The stages are: Administrative panel and User end, the function of each part is well defined using the User cases and activity diagram.

3.4 Use Cases

Use cases are "a description of set of sequences of actions, including variants, that a system performs that yield an observable result of value to an actor". They are used in order to: design system from user's perspective, communicate system behavior in user's term and enumerate all externally visible behavior. Figure 3.2 and figure 3.3 showed the use cases for this work (there are two actors for the system: a normal user and an administrator).

As shown in the figure 3.2, a normal user can register/login to the system, download available document, upload a document, edit a document, delete document, share document and send document. The administrator, on the other hand, can also login, manage users account (e.g. generating an OTP for a first time user, disabling a user and deleting a user account), edit system configurations, backup systems information and database, manage user log (i.e. view user logs and view documents activity logs) and perform all other function a normal user can perform.

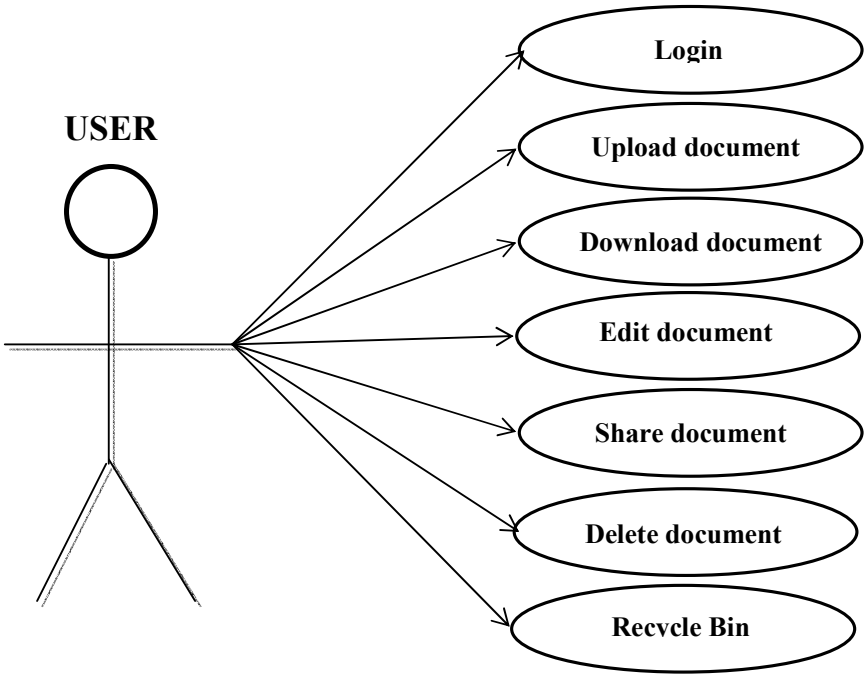

Figure 3.2: A description of set of sequences of actions by users

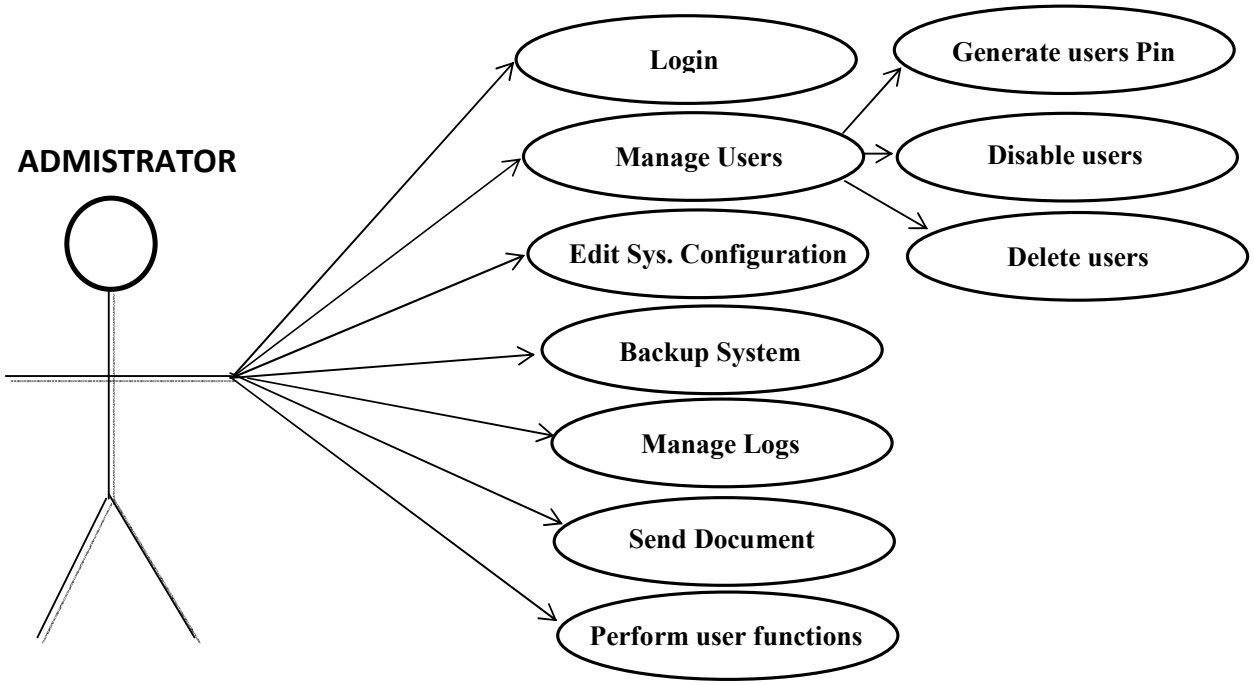

Figure 3.3: A description of set of sequences of actions by Administrator

3.4.1 User Stories

After collecting all use cases, user stories can be written. A user story is the smallest amount of information (a step) necessary to allow the customer to define (and steer) a path through the system. The user stories are divided into 2 main categories: user side (stories for the general users) and administration side (the stories for the administrators of the system).

3.4.1.1 User Side Stories:

The Login page is the entry point to access the main services:

Registration: The registration page allows user to provide his/her personal data (name, email address, phone number, and password) and receive a username. User Name and password allow the user to access to his/her dashboard and use the system. It performs basic checks on entered data and provides user registration or an error message if information was inputted wrongfully. It activity is shown in figure 3.4

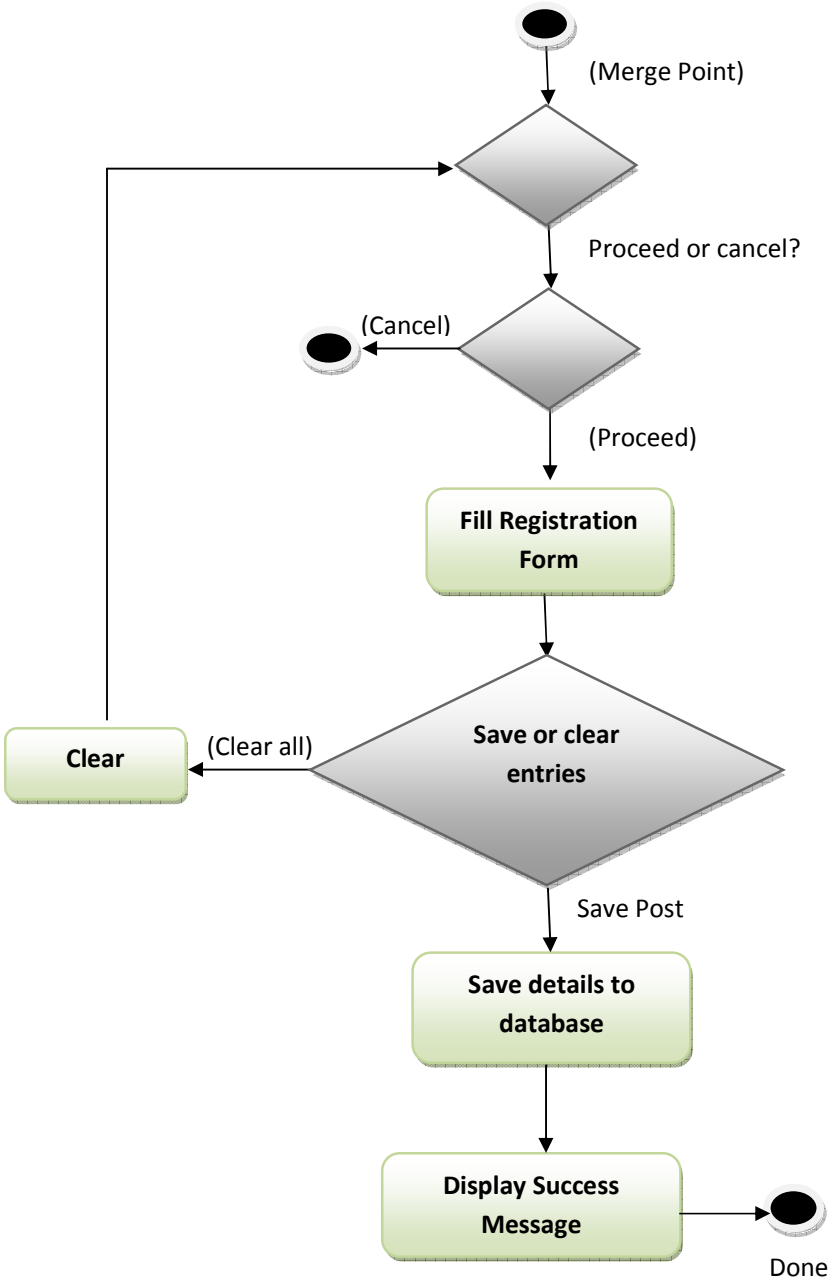

Figure 3.4: Activity Diagram Registration

Login: Every time the user tries to access to non-public areas (Download, download page, upload page etc.), he/she is asked to provide his/her User Name and password. These are entered through a form. If the username and password are correct, the user is logged in and not asked to login throughout the session. Otherwise an error message is raised. It activity is shown in figure 3.5.

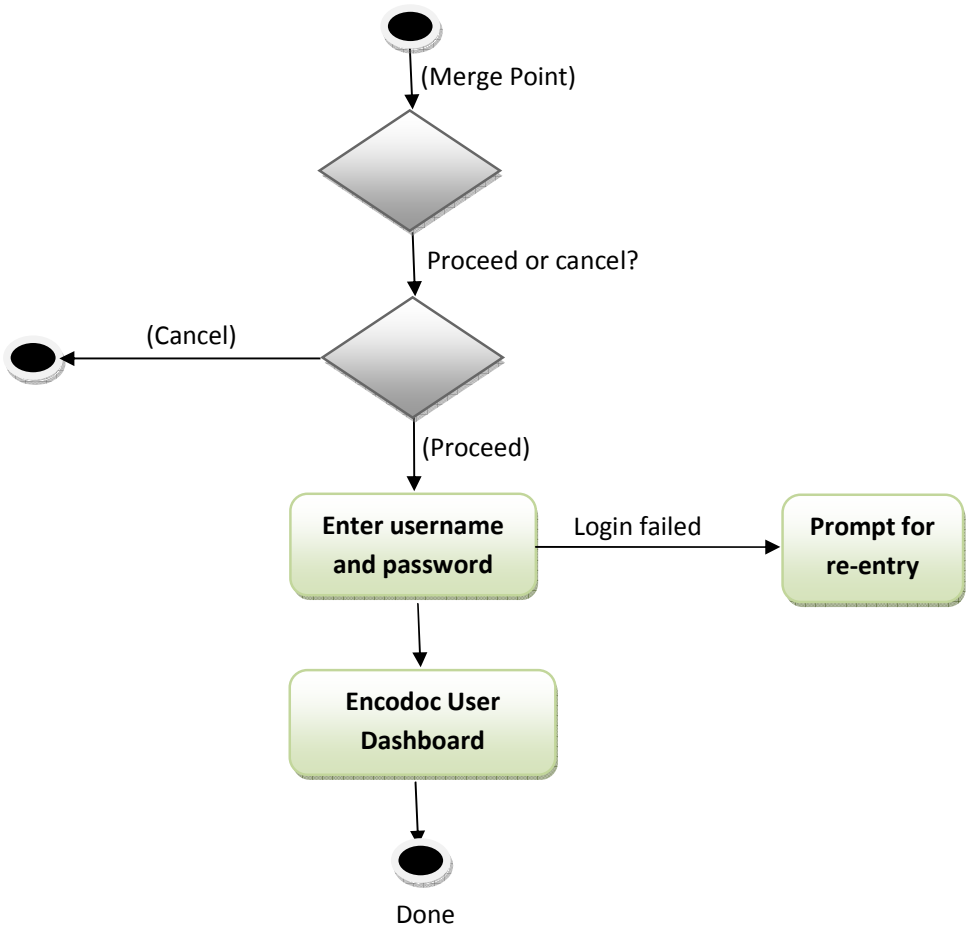

Figure 3.5: User Login Activity Diagram

Search: The user can search for document providing a key word by different criterions:

• In a given category (username, document name, document title, document size etc.)

• In a given department

• In a giving document file type/format

It activity is shown in figure 3.6.

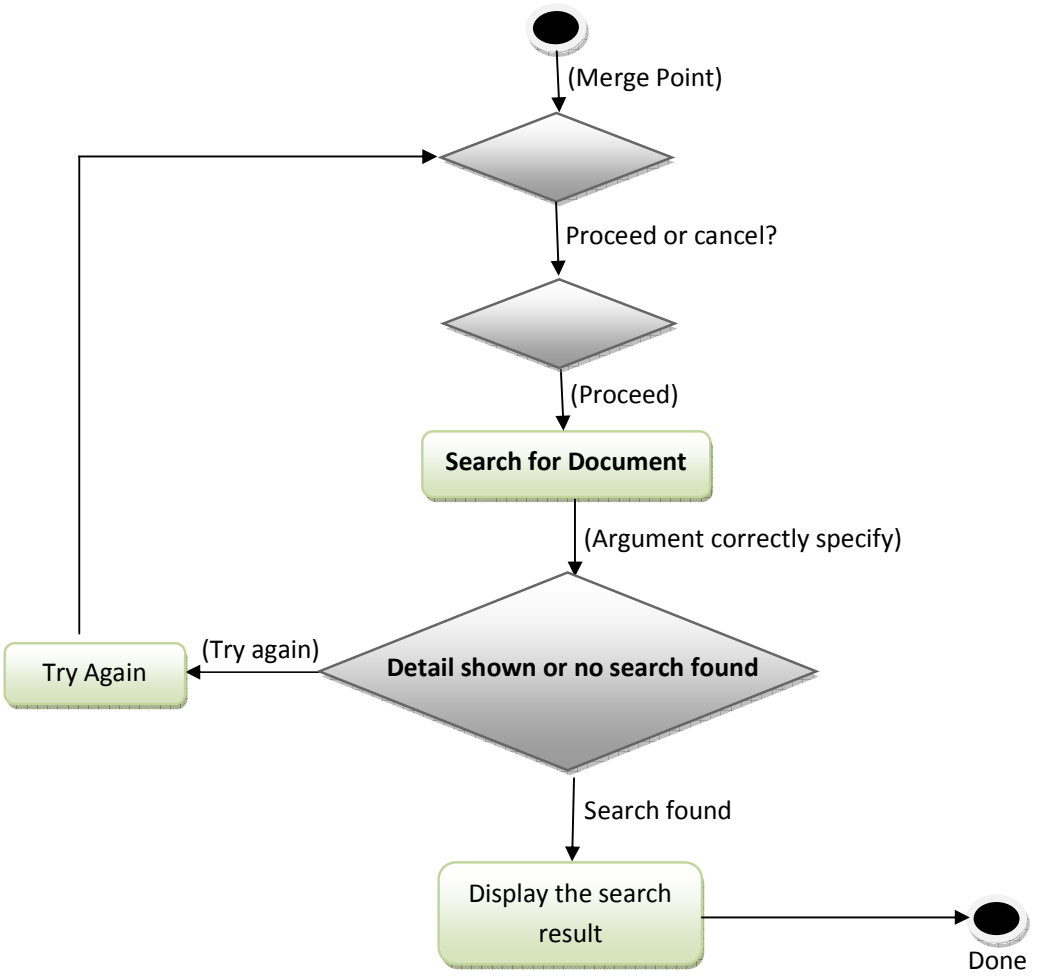

Figure 3.6: Activity Diagram Search Document

Upload: Upload requirement are shown on the upload page. From this page the user can upload a document by pushing the button "UPLOAD" after providing relevant information about the document about to be uploaded. It activity is shown in figure 3.7.

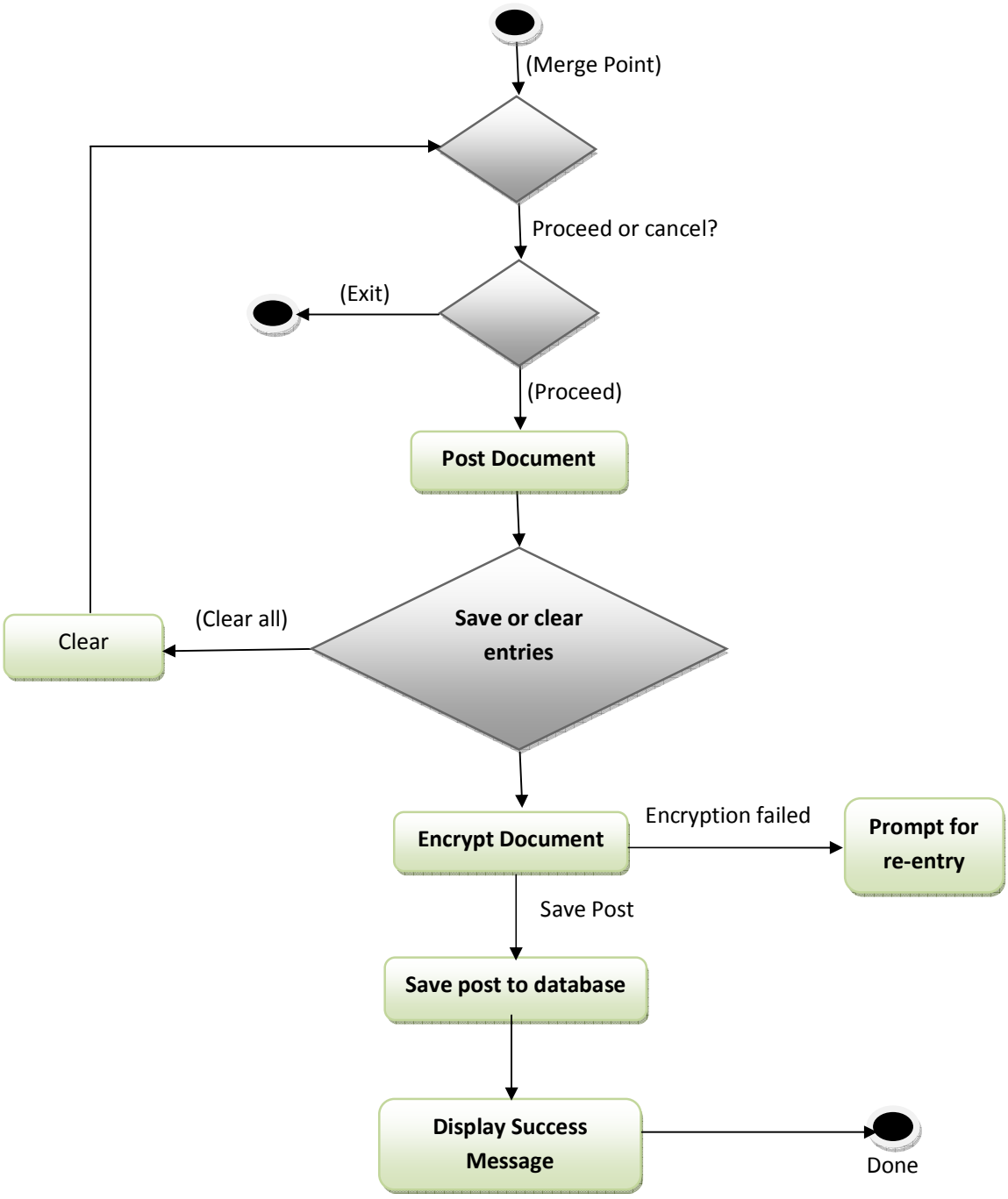

Figure 3.7: Document Upload Activity Diagram

Download: The user can download document uploaded by him or document made public by the creator of the document. It activity is shown in figure 3.8.

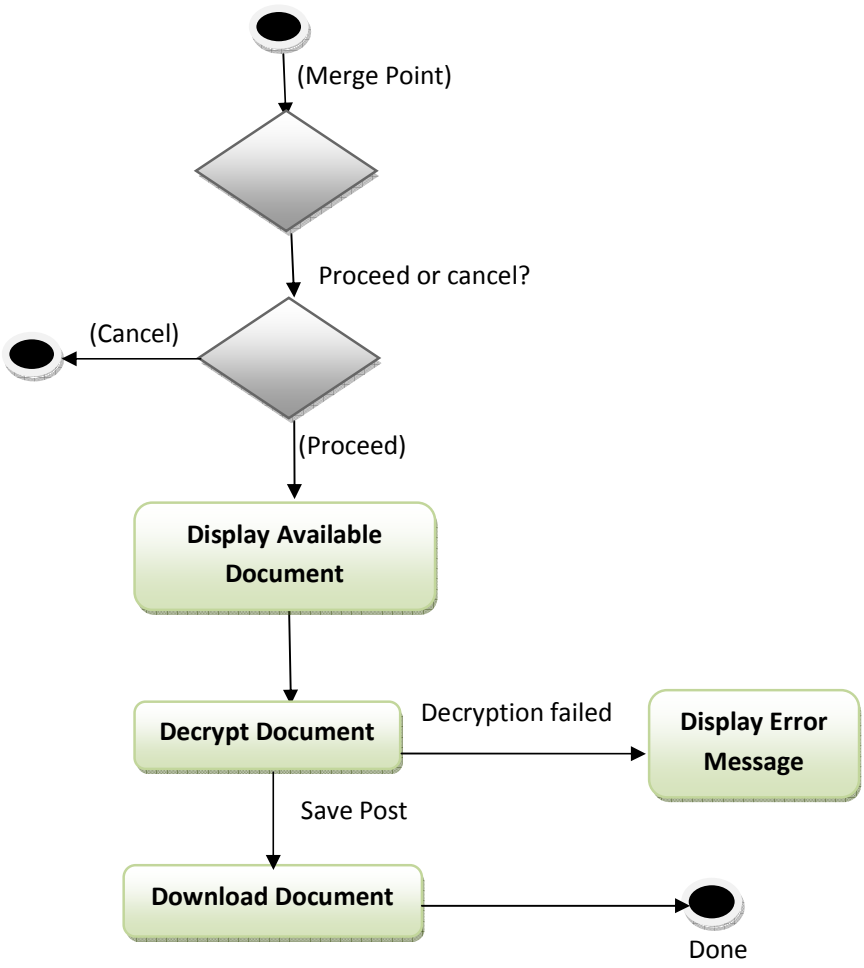

Figure 3.8: Document Download Activity Diagram

Edit: The user can makes changes to information about a document from the edit page. It activity is shown in figure 3.9.

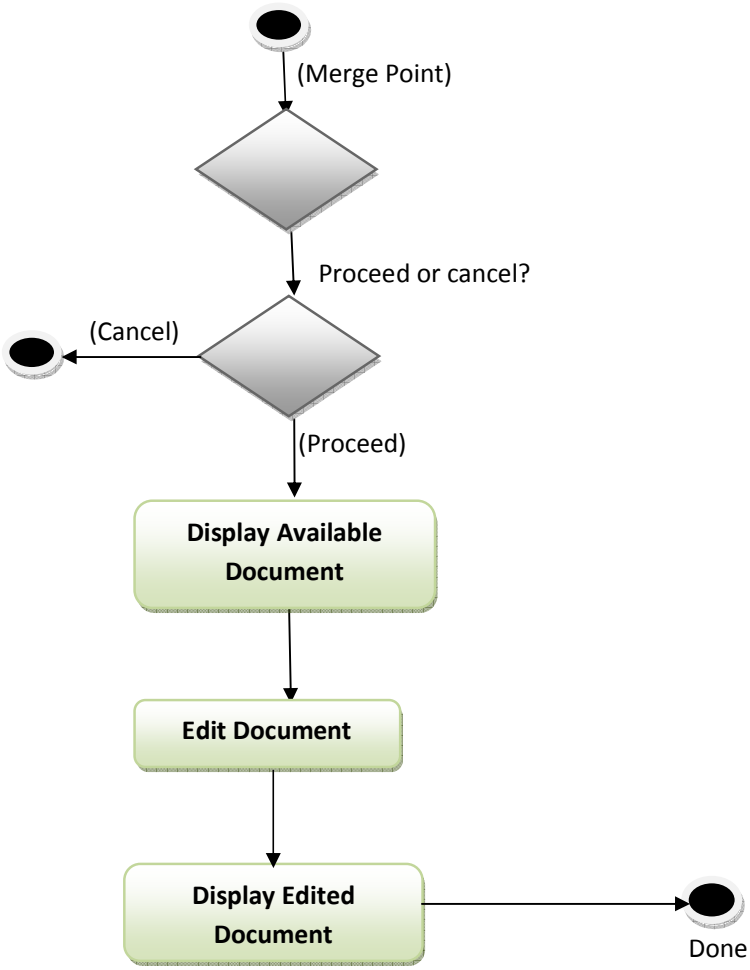

Figure 3.9: Document Editing Activity Diagram

Delete: From this page, the user can delete a document created by him/her and the document will be place in the recycle bin. It activity is shown in figure 3.10.

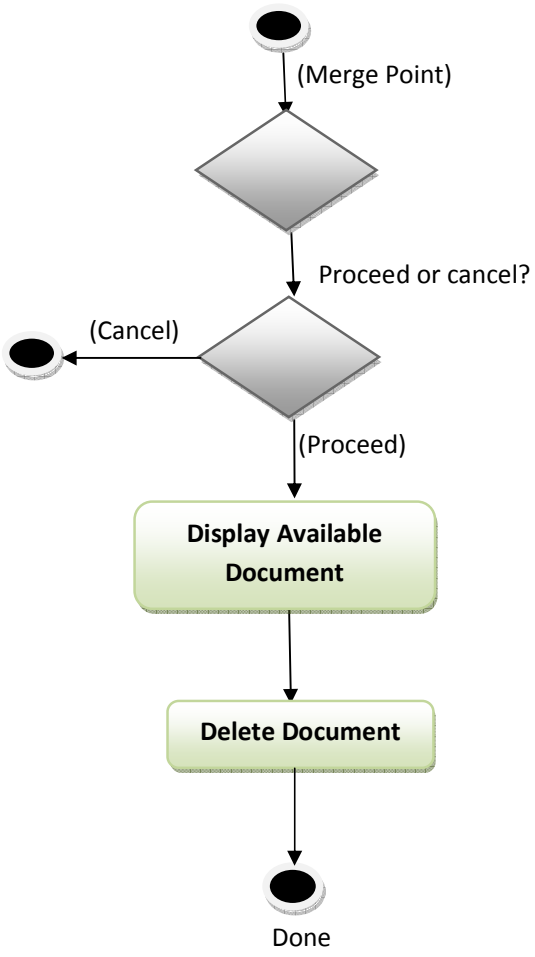

Figure 3.10: Document Delete Activity Diagram

Recycle Bin: from the page user can restore deleted document and delete a document permanently. It activity is shown in figure 3.11.

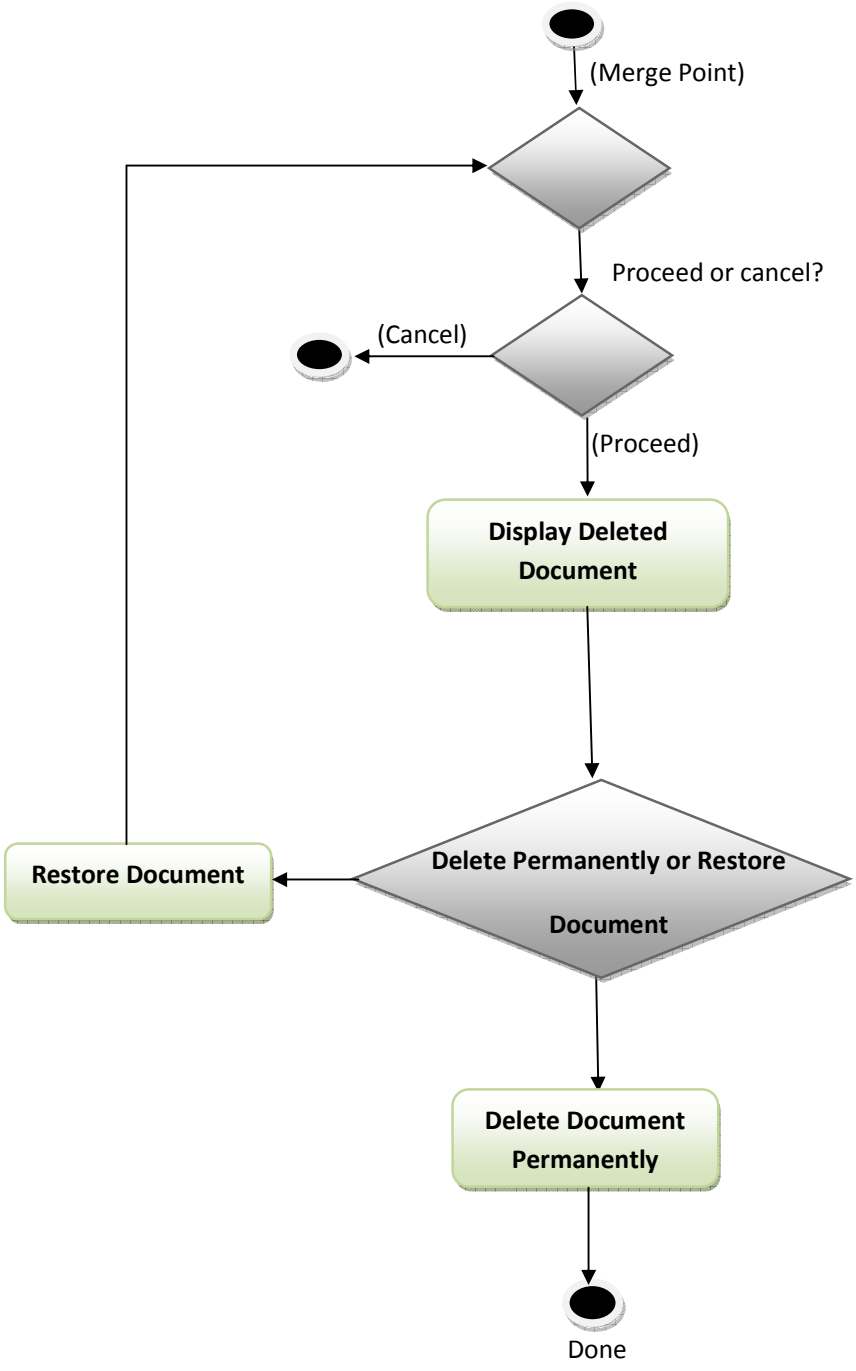

Figure 3.11: Document Recycle Bin Activity Diagram

Share: From this page user can share document uploaded by him/her to a user or to a department. It activity is shown in figure 3.12.

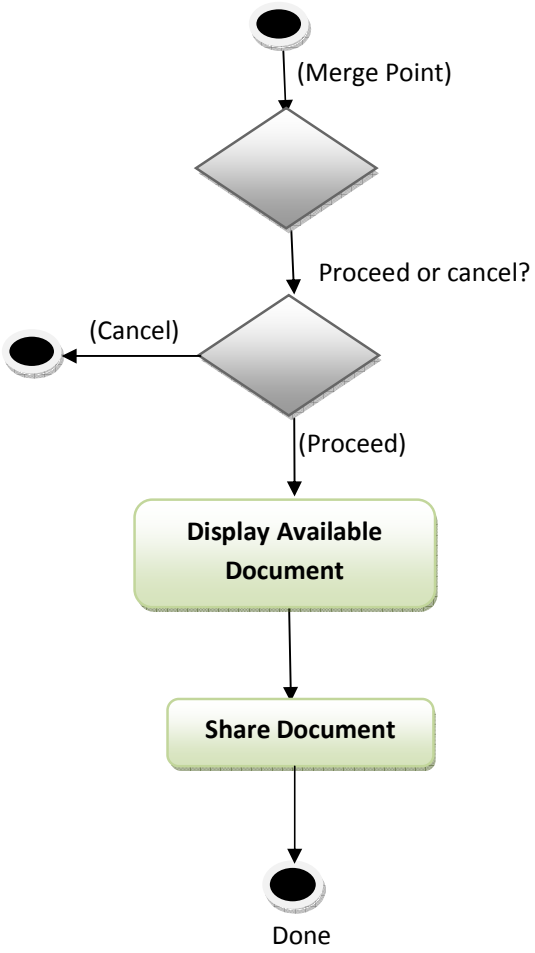

Figure 3.12: Document Share Activity Diagram

3.4.1.2 Administrator Side Stories:

Administrator Dashboard: From the login page, providing his/her username and password, the system checks and if the user is an administrator, he/she can access the administrator page that shows the administrator menu to access all the administration activities (Edit System

configuration, manage logs, manage users and backup system). It activity is shown in figure 3.13.

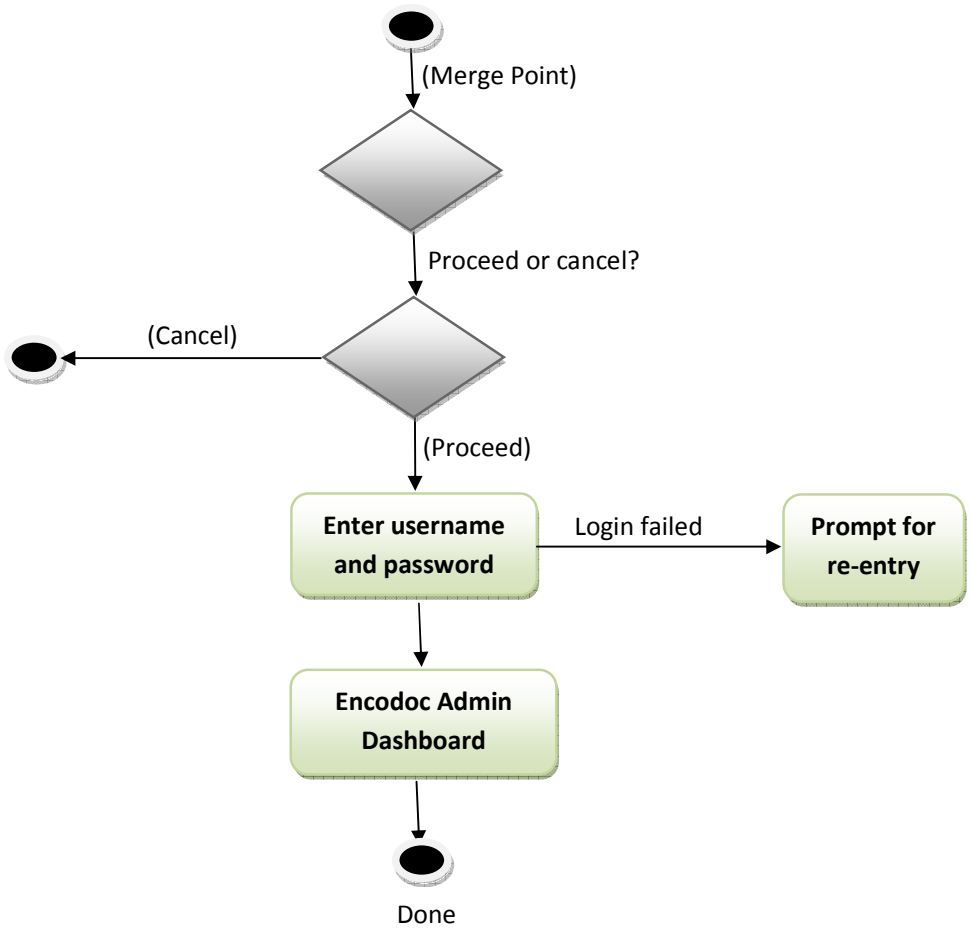

Figure 3.13: Admin Login Activity Diagram

Edit System Configuration: the administrator can edit the mode of operation (allowed document types etc.) of the system likewise look and feel of the system, logo, images, banners, etc.

Manage Log: The administrator can view and delete the logs of activities performed by a user. It activity is shown in figure 3.14.

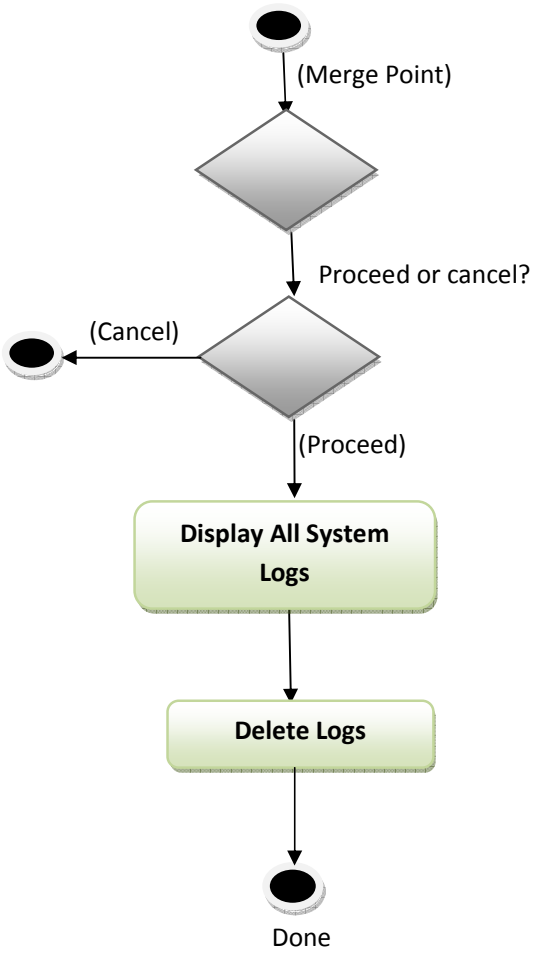

Figure 3.14: Manage Log Activity Diagram

Manage User: The administrator can access and modify all data of users stored in the database and also disable or delete a user. It activity is shown in figure 3.15.

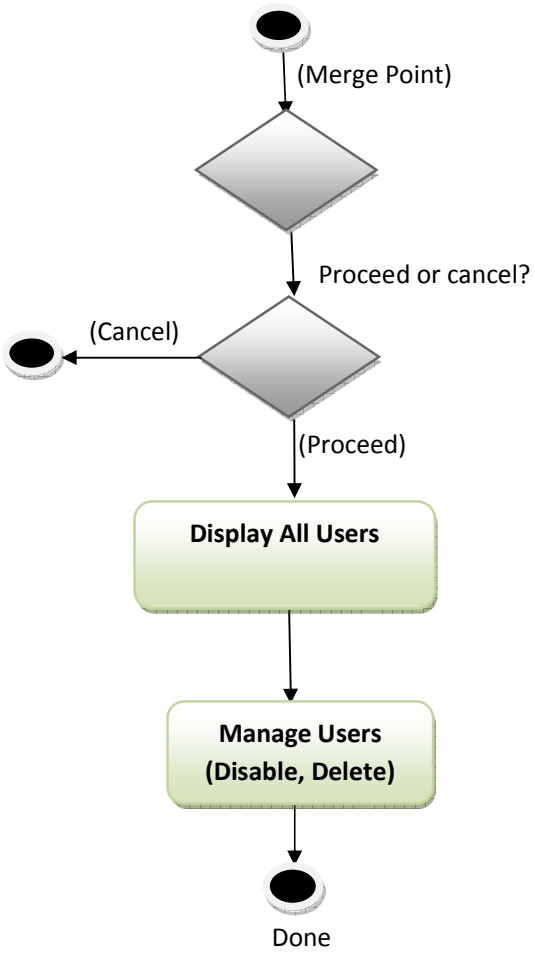

Figure 3.15: Manage User Activity Diagram

Send Document: This page give the administrator access to send a document as an email to any email address. It activity is shown in figure 3.16.

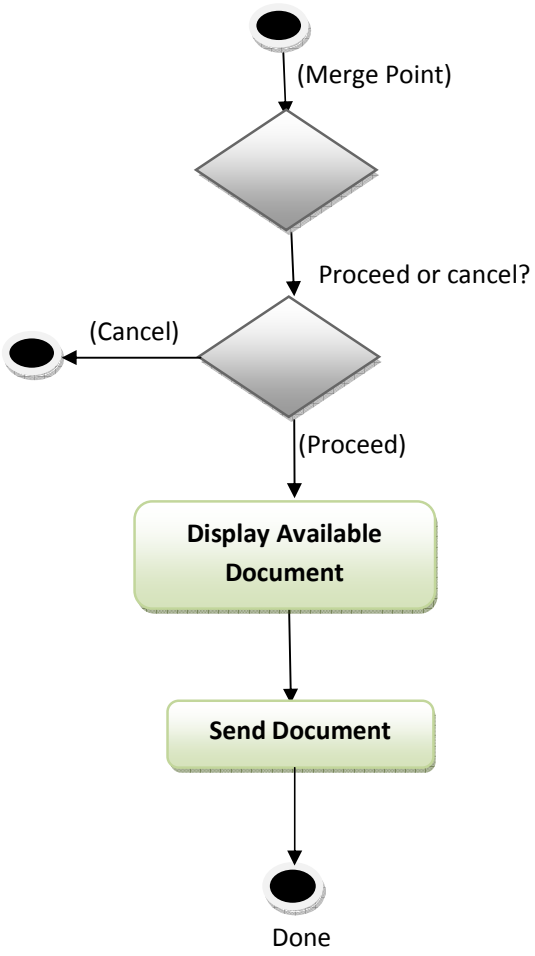

Figure 3.16: Send Document Activity Diagram

Backup System: The administrator can back-up the database by clicking the "BACK-UP" button on the backup page.

3.5 Designing the AES Algorithm

AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits. The key size used for an AES cipher specifies the number of repetitions of

transformation rounds that convert the input, called the plaintext, into the final output, called the ciphertext. The numbers of cycles of repetition are as follows:

a. 10 cycles of repetition for 128-bit keys.

b. 12 cycles of repetition for 192-bit keys.

c. 14 cycles of repetition for 256-bit keys.

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

AES High-level description

i. KeyExpansions - round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.

ii. InitialRound

AddRoundKey—each byte of the state is combined with a block of the round key using bitwise xor.

iii. Rounds

a. SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

b. ShiftRows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

c. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.

d. AddRoundKey

iv. Final Round (no MixColumns)

a. SubBytes

b. ShiftRows

c. AddRoundKey.

a. SubByte step: In the SubBytes step, each byte $a_{i,j}$ in the state matrix is replaced with a SubByte $S(a_{i,j})$ using an 8-bit substitution box, the Rijndael S-box as shown in figure 3.17. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over GF (2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid

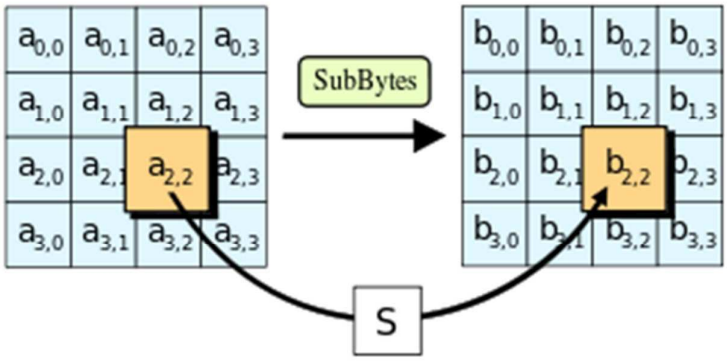

Figure 3.17: SubBytes operation for AES

any fixed points i.e., $S(a_{i,j}) \neq a_{i,j}$, and also any opposite fixed points, i.e., $S(a_{i,j}) \oplus a_{i,j} \neq 0xFF$. While performing the decryption, Inverse SubBytes step is used, this requires first taking the affine transformation and then finding the multiplicative inverse.

b. ShiftRow step: The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset as shown if figure 3.18. For AES, the first row is left unchanged.

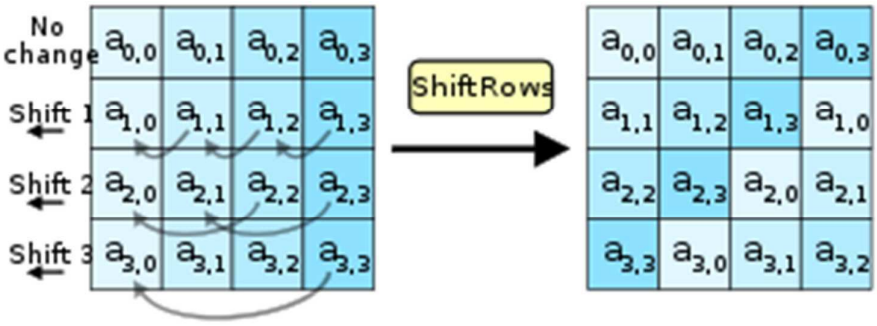

Figure 3.18: ShiftRows operation for AES.

Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. For blocks of sizes 128 bits and 192 bits, the shifting pattern is the same. Row n is shifted left circular by n-1 bytes. In this way, each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). For the 256-bit block, the first row is unchanged and the shifting for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks. The importance of this step is to avoid the columns being linearly independent, in which case, AES degenerates into four independent block ciphers.

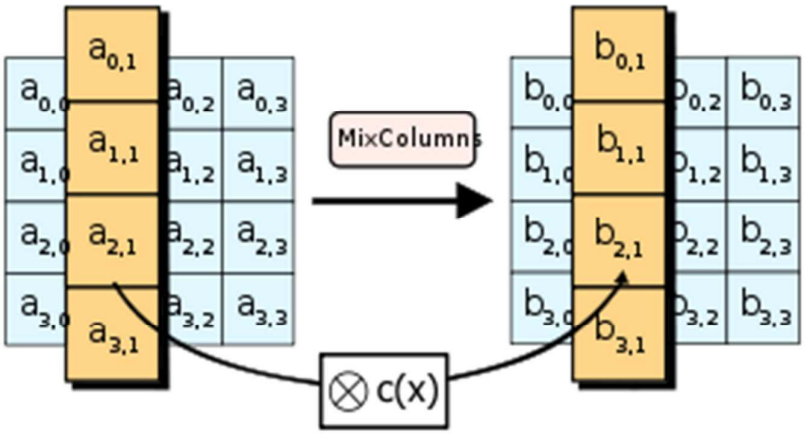

Figure 3.19: MixColumns operation for AES

c. MixColumns step: from figure 3.19, the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation. The MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in the cipher. During this operation, each column is multiplied by a fixed matrix:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Matrix multiplication is composed of multiplication and addition of the entries, and here the multiplication operation can be defined as this: multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing XOR with the initial unshifted value. After shifting, a conditional XOR with 0x1B should be performed if the shifted value is larger than 0xFF. (These are special cases of the usual multiplication in $GF(2^8)$.) Addition is simply XOR.

In the general sense, each column is treated as a polynomial over $GF(2^8)$ and is then multiplied modulo x^4+1 with a fixed polynomial $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$. The coefficients are displayed in their hexadecimal equivalent of the binary representation of bit polynomials from GF(2)[x]. The MixColumns step can also be viewed as a multiplication by the shown particular MDS matrix in the finite field $GF(2^8)$. This process is described further in the article Rijndael mix columns.

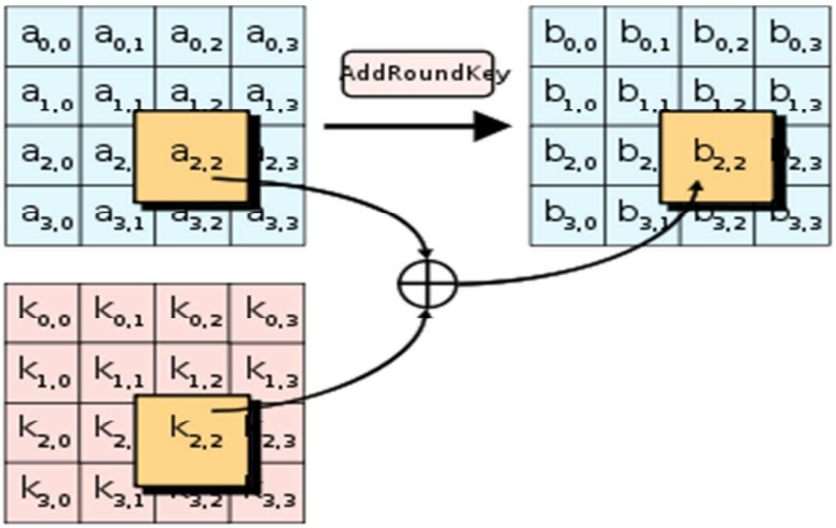

Figure 3.20: AddRoundKey operation for AES

d**.** The AddRoundKey step: In the AddRoundKey step, each byte of the state is combined with a byte of the round subkey using the XOR operation (⊕) as show in figure 3.20. In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

Optimization of the cipher: On systems with 32-bit or larger words, it is possible to speed up execution of this cipher by combining the SubBytes and ShiftRows steps with the MixColumns step by transforming them into a sequence of table lookups. This requires four 256-entry 32-bit tables, and utilizes a total of four kilobytes (4096 bytes) of memory — one kilobyte for each table. A round can then be done with 16 table lookups and 12 32-bit exclusive-or operations, followed by four 32-bit exclusive-or operations in the AddRoundKey step.

If the resulting four-kilobyte table size is too large for a given target platform, the table lookup operation can be performed with a single 256-entry 32-bit (i.e. 1 kilobyte) table by the use of circular rotates.

CHAPTER FOUR

IMPLEMENTATION AND EVALUATION

This chapter discuss in detail the processes involved in putting the new system into full operation. The stages are system hardware requirement, system software requirement, and implementation procedure, sample of implementation snapshot and evaluation of result.

4.1 System Hardware Requirements

The following hardware resources are needed to be put in place in order to power the application: A host for the system and clients to connect to it via intranet.

i. Pentium IV Computer (2.6 GHz processor's speed)
ii. 40GB or more hard disk space requirement
iii. 1GB or more of RAM
iv. A high speed intranet connection
v. 1024*768 pixels screen resolution
vi. Mouse
vii. Keyboard

4.2 System Software Requirement

i. Any operating system that support GUI e.g. Windows 7, Windows 8.
ii. WAMP Server
iii. Zziplib Library.
iv. Macromedia Dreamweaver 8, Fireworks.
v. Browser such as (Firefox, internet explorer, chrome)

4.3 Data Source

For this work data was source from the following:

a. Oyo state Housing Corporation.
b. The Internet to research closely related works
c. Some Libraries
d. PHP Manual

4.4 Implementation Procedure

i. Install WAMP Server
ii. Include the Zzip Library to WAMP
iii. Configure the server
iv. Start the application server from start menu
v. Open your favourite browser .i.e. Firefox, Chrome and type http://localhost/encodoc

The login page appears but without menus and the user will have to login or use an OTP (one-time password) if he is a first time user to create his username and password, because the user has to register before he can login with username and password supplied at the point of registration. Having duly registered and login with the username and password, the services of the system can be accessed.

4.5 Algorithms

This works' algorithms are divided into: Encryption algorithm, Decryption Algorithm and Compression algorithm.

Encryption Algorithm: the encryption algorithm used in this work is show in figure 4.1.

```php
<?php
//Include the library
require_once 'aes256/AESCryptFileLib.php';
//Include an AES256 Implementation
require_once 'aes256/MCryptAES256Implementation.php';
$password="password";
$filename = $_POST['fname'];
 move_uploaded_file($_FILES["file"]["tmp_name"], "image/".$filename.".".
$extension);
//Construct the implementation
$mcrypt = new MCryptAES256Implementation();
//Use this to instantiate the encryption library class
$lib = new AESCryptFileLib($mcrypt);
//This example encrypts and decrypts the README.md file
$file_to_encrypt = "image/".$filename.".".$extension;
$encrypted_file = "image/".$filename.".aes";
//Ensure target file does not exist
@unlink($encrypted_file);
//Encrypt a file
$lib->encryptFile($file_to_encrypt, $password, $encrypted_file);
// ensure file doesnt exist
@unlink($file_to_encrypt);
echo "Encrypted";
?>
```

Figure 4.1: Encryption Algorithm

Decryption Algorithm: the encryption algorithm used in this work is show in figure 4.2.

```php
<?php
$zip = new ZipArchive();
$zip_name = "zip/marker2.zip";
if (file_exists($zip_name)){
    echo "File Exist, No Archiving Done";
}
else {
if($zip->open($zip_name, ZIPARCHIVE::CREATE)!==TRUE){
}
else{
$error = "* Sorry ZIP creation failed at this time";
}
$files=array("chemmy.aes", "chemmy_Ds.docx", "diode.aes");
foreach ($files as $file) {
  $zip->addFile(getcwd()."/image/".$file, $file);
}
$zip->close();
}
?>
```

Figure 4.2: Decryption Algorithm

Compression Algorithm: the compression algorithm used in this work is show in figure 4.3.

```php
<?php
//Include the library
require_once 'aes256/AESCryptFileLib.php';
//Include an AES256 Implementation
require_once 'aes256/MCryptAES256Implementation.php';
    $ext = $_POST["ext"];
    $filename= $_POST['fname'];
    $password="password";
    //Construct the implementation
$mcrypt = new MCryptAES256Implementation();
//Use this to instantiate the encryption library class
$lib = new AESCryptFileLib($mcrypt);
//This example encrypts and decrypts the README.md file
$encrypted_file = "image/".$filename.".aes";
$decrypted_file = "image/".$filename."_Ds.".$ext;
    //Ensure file does not exist
@unlink($decrypted_file);
//Decrypt using same password
$lib->decryptFile($encrypted_file,$password, $decrypted_file);
echo "Decrypted";
?>
```

Figure 4.3: Compression Algorithm

4.6 Sampled Snapshot

Login page: This serves as the first interface that welcomes the user to the site, this page has no menus but from the page, user will be able to access the login form to login into the system as shown in figure 4.4, if the user is a first time user then he has to register as he is require to login with the username and password.

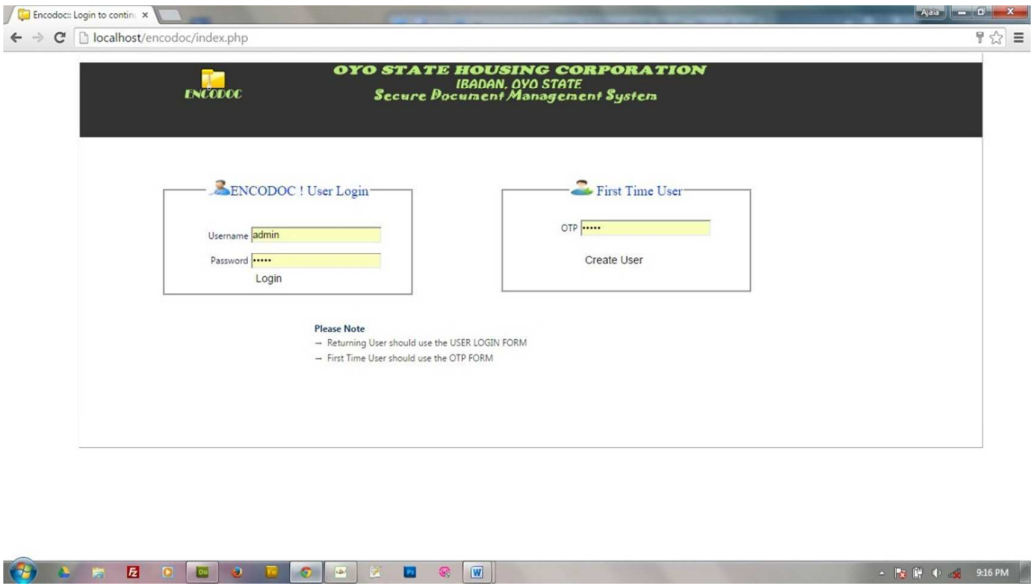

Figure 4.4: Login Page

A Dashboard: This is a page that is generated after a successful login by the client or Admin having duly registered; it has all the menus that can be used to access all the systems functions as shown in figure 4.5. On this page, the client is given the chance to logout his account to avoid unauthorized user from accessing his/her account.

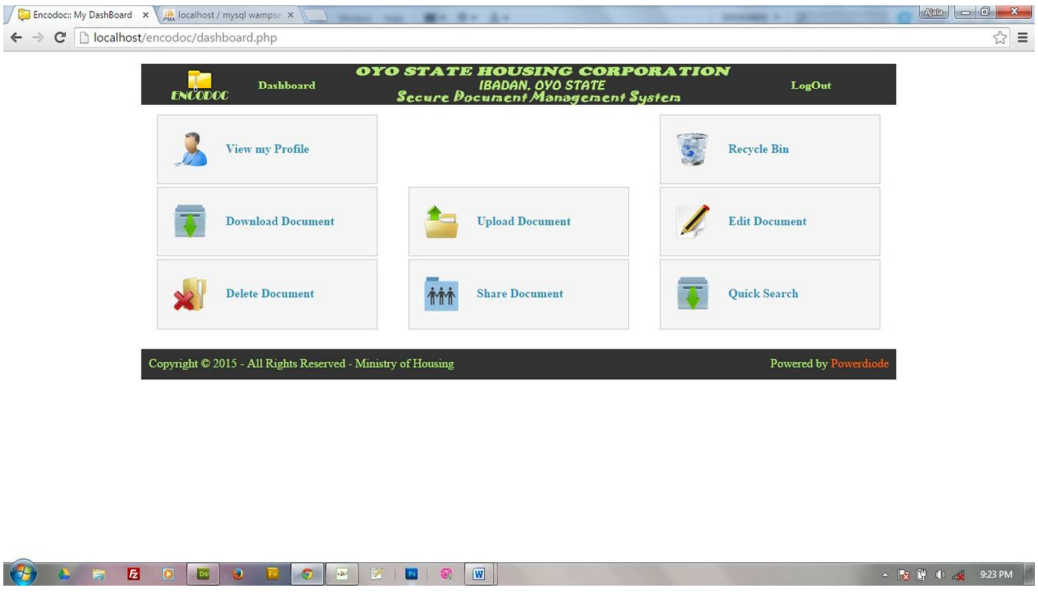

Figure 4.5: A Successful Login Page (Dashboard)

Registration Page: This is where the new user has to register before accessing the system. A critical look at this form, it has no menu as shown in figure 4.6.

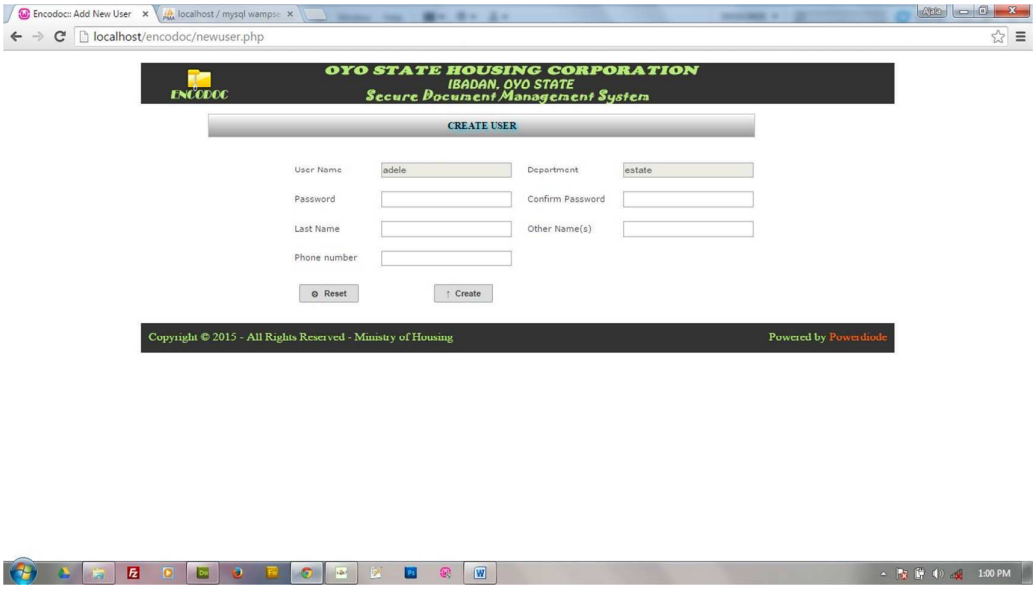

Figure 4.6: Registration Page

Upload Document: This is a form that affords the user the opportunity to upload his document into the system as shown in figure 4.7.

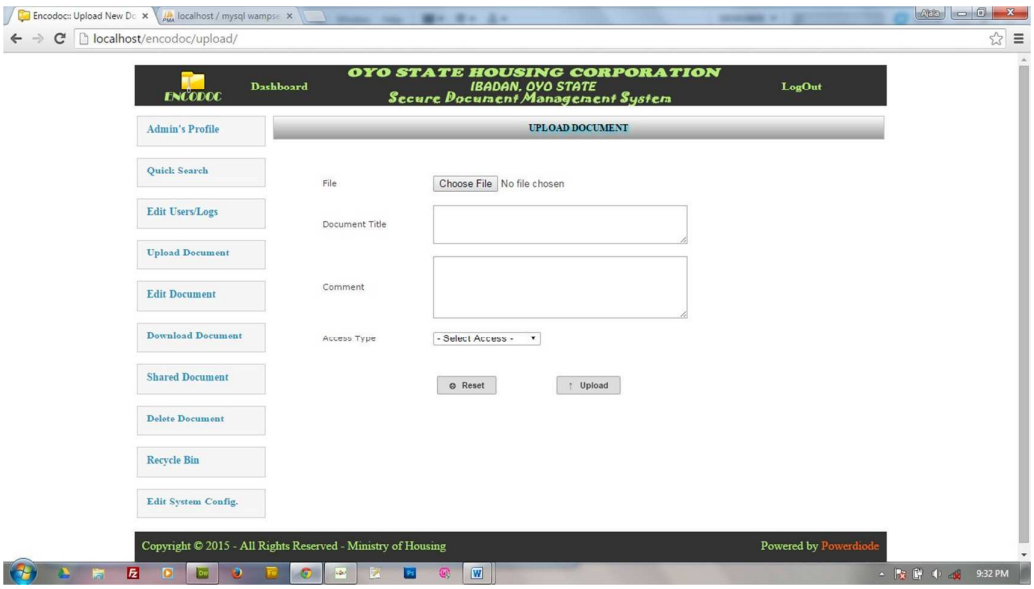

Figure 4.7: Upload Document

Edit Document: This page gives the user opportunity to edit information about already uploaded document by the user either to correct error in upload, increase search chances or to aid sharing as shown in figure 4.8.

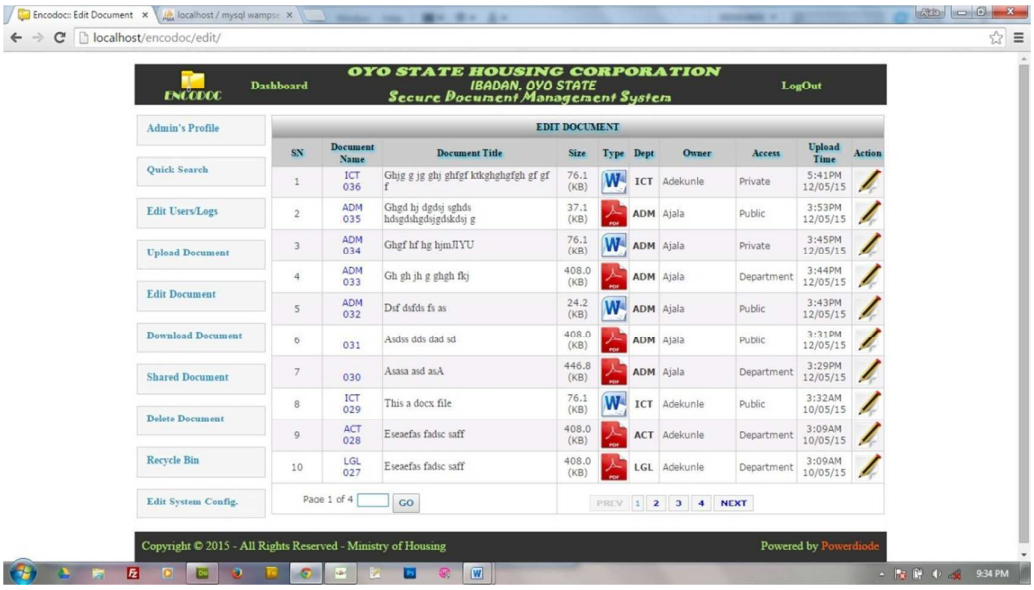

Figure 4.8: Edit Document

Delete Document: from this page user can delete document uploaded by him and the document will be moved to the recycle bin as shown in figure 4.9. But once the document is deleted it can't be accessed by the user or by user which the document has being shared with.

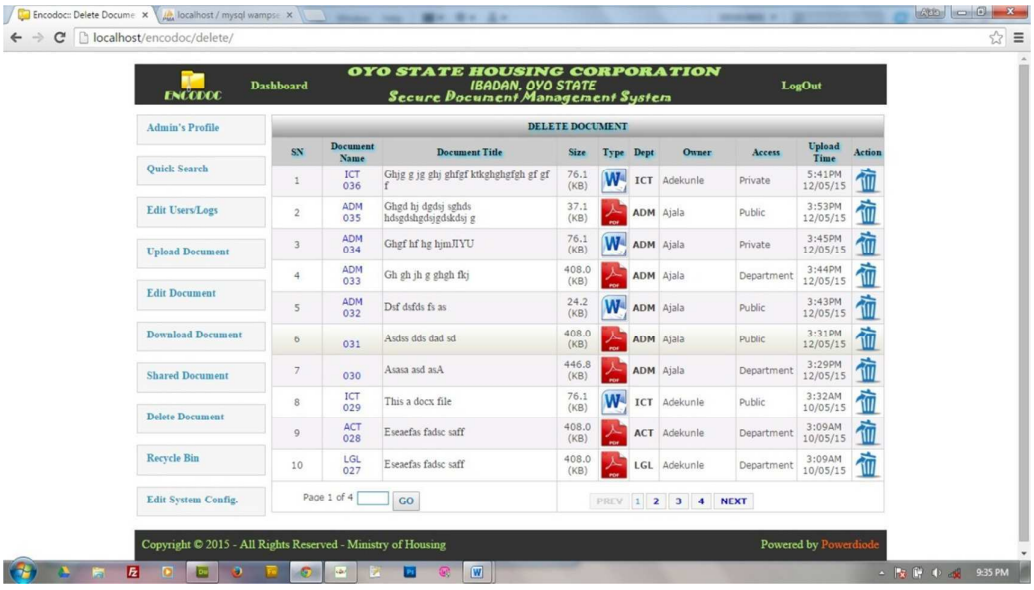

Figure 4.9: Delete Document

Recycle Bin: Document deleted will be moved here; therefore the user can choose to delete the document permanently or restore the document as shown in figure 4.10.

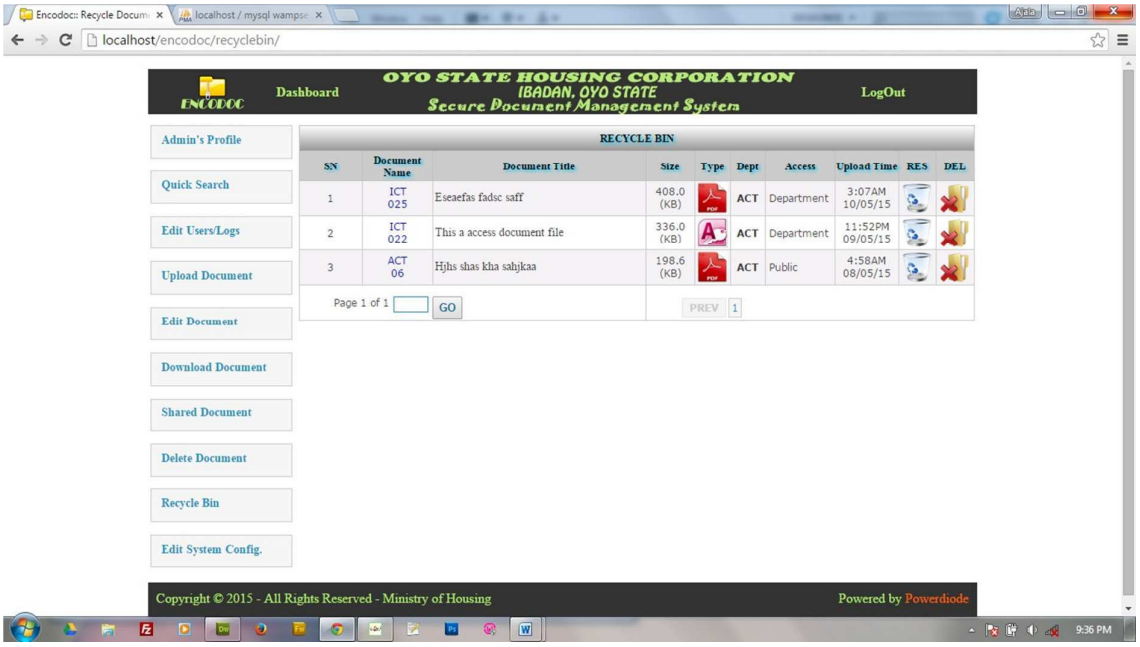

Figure 4.10: Recycle Bin

Share Document: This page allows user view document that was shared with him and to share document he upload with other users or a department as shown in 4.11.

Figure 4.11: Share Document

Admin Login: This is the administrative login with username = admin and password = admin to keep check on the administrative end to avoid unauthorized personnel as shown in figure 4.12.

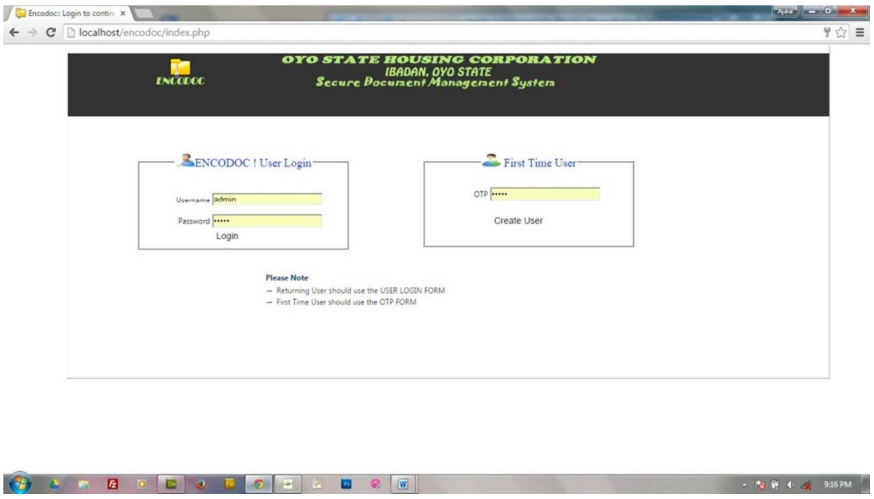

Figure 4.12: Admin Login

Admin Home: This is the admin home for the administrator, where he can manage users, manage system log and edit other system configuration, the side menu of this page is higher in number than that of a user as shown in figure 4.13.

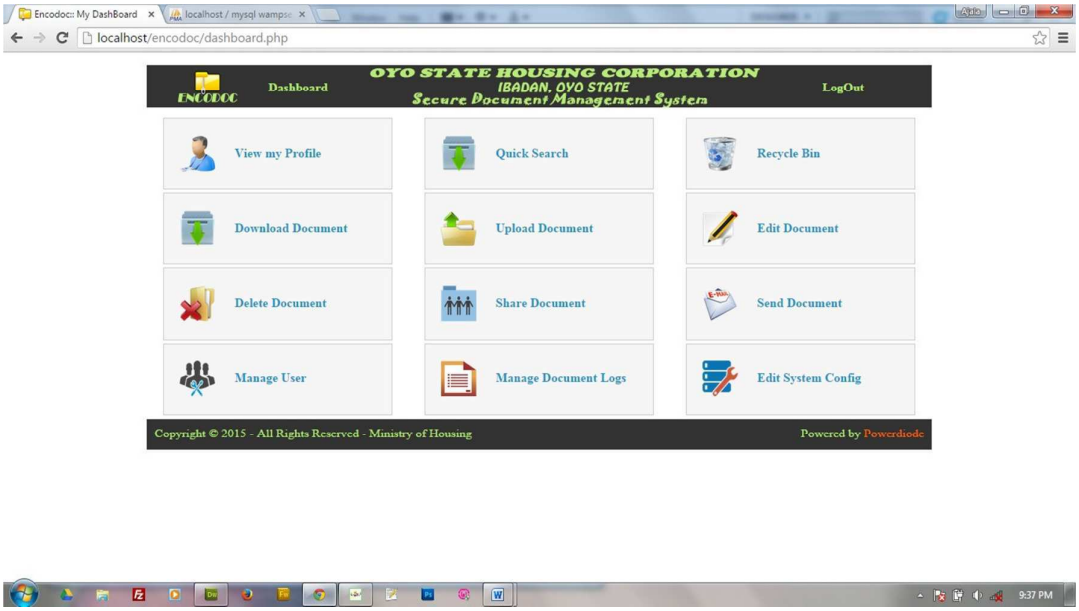

Figure 4.13: Admin Home

Send Document: This is where documents can be send as an email to client that desired any services of the organization as shown in figure 4.14.

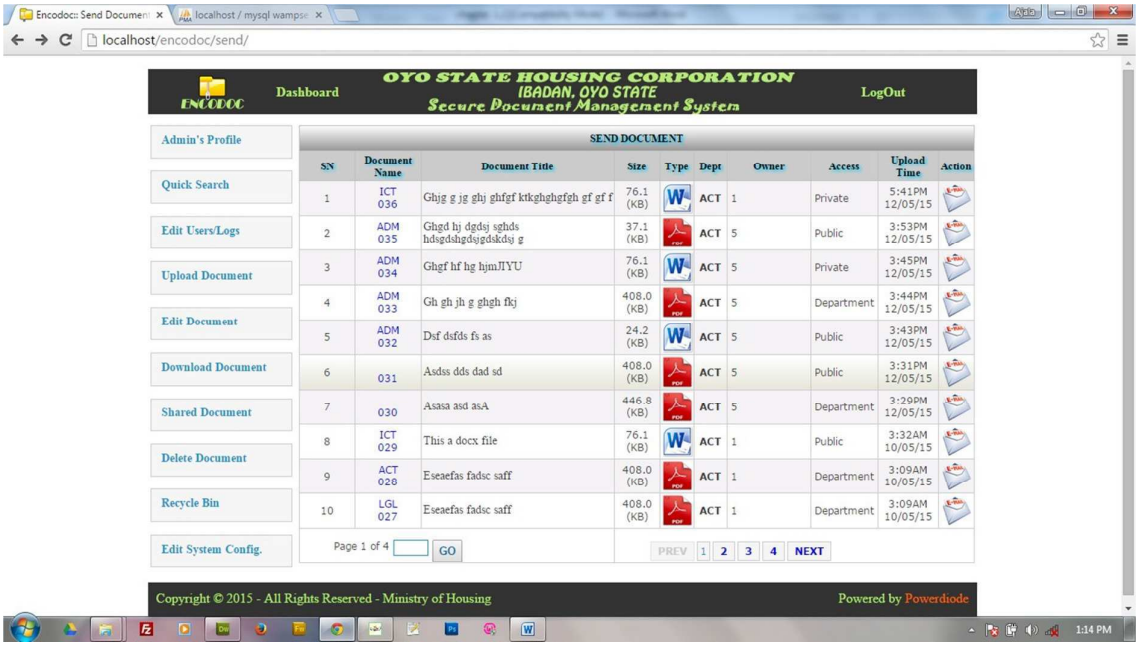

Figure 4.14: Send Document

Manage Users: this page allow admin access to user information and power to force a change to any users' information and power over any user access to the system as shown in figure 4.15.

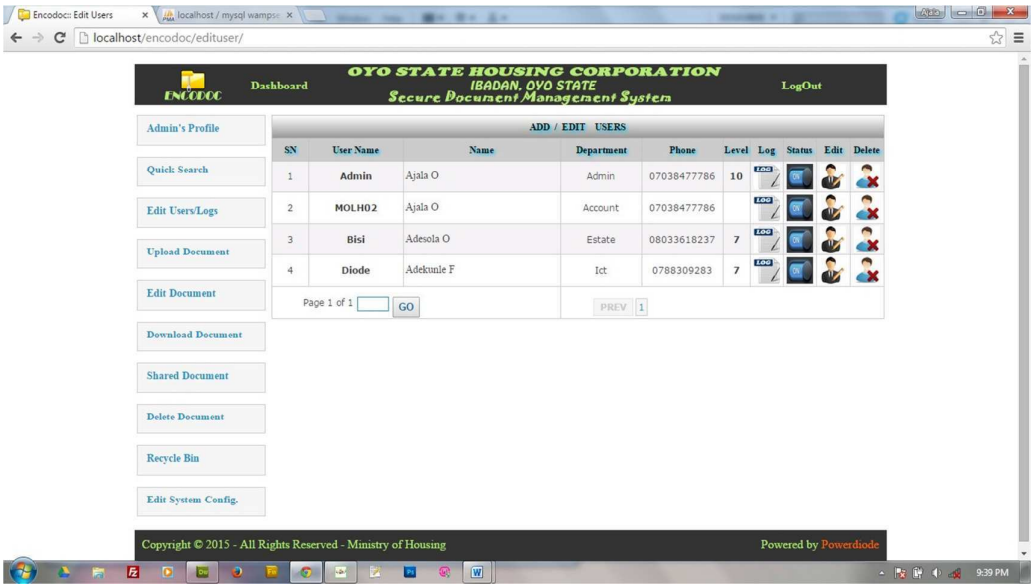

Figure 4.15: Manage User

Manage Logs: this page display all logs of user activities as shown in figure 4.16.

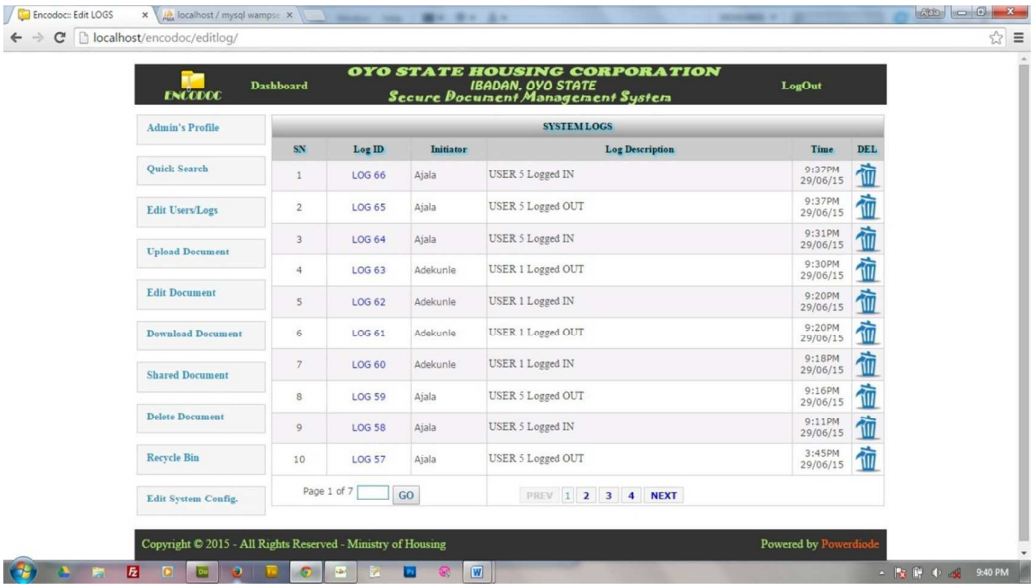

Figure 4.16: Manage Logs

Database Implementation Snapshot

The database name is encodoc_db and it consists of seven tables viz: users, documents, userlog, newusers, share, config, allowdocumenttype.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(15)			No	None	AUTO_INCREMENT
2	uName	varchar(30)	latin1_swedish_ci		No	None	
3	uPass	varchar(30)	latin1_swedish_ci		No	None	
4	uEncrypt	varchar(100)	latin1_swedish_ci		No	None	
5	uLname	varchar(30)	latin1_swedish_ci		No	None	
6	uOname	varchar(50)	latin1_swedish_ci		No	None	
7	uPhone	varchar(15)	latin1_swedish_ci		No	None	
8	uStatus	varchar(30)	latin1_swedish_ci		No	None	
9	uLevel	varchar(30)	latin1_swedish_ci		No	None	
10	uDept	varchar(30)	latin1_swedish_ci		No	None	
11	uemail	varchar(50)	latin1_swedish_ci		No	None	

Figure 4.17: Users table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(15)			No	None	AUTO_INCREMENT
2	otp	varchar(50)	latin1_swedish_ci		No	None	
3	used	varchar(1)	latin1_swedish_ci		No	None	
4	nDept	varchar(30)	latin1_swedish_ci		No	None	
5	nName	varchar(30)	latin1_swedish_ci		No	None	
6	nlevel	int(2)			No	None	

Figure 4.18: Newusers table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(15)			No	None	AUTO_INCREMENT
2	dName	varchar(30)	latin1_swedish_ci		No	None	
3	dTitle	varchar(80)	latin1_swedish_ci		No	None	
4	dOwner	int(15)			No	None	
5	dSize1	varchar(15)	latin1_swedish_ci		No	None	
6	dSize2	varchar(10)	latin1_swedish_ci		No	None	
7	dType	varchar(10)	latin1_swedish_ci		No	None	
8	dAccess	varchar(15)	latin1_swedish_ci		No	None	
9	dDate	date			No	None	
10	dTime	time			No	None	
11	dDept	varchar(30)	latin1_swedish_ci		No	None	
12	dDept2	varchar(4)	latin1_swedish_ci		No	None	
13	dStatus	varchar(10)	latin1_swedish_ci		No	None	
14	dEncrypt	varchar(60)	latin1_swedish_ci		No	None	
15	dPath	varchar(150)	latin1_swedish_ci		No	None	
16	dEditDate	date			No	None	
17	dEditTime	date			No	None	
18	dComment	varchar(300)	latin1_swedish_ci		No	None	

Figure 4.19: documents table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(30)			No	None	AUTO_INCREMENT
2	usId	int(15)			No	None	
3	usAction	varchar(300)	latin1_swedish_ci		No	None	
4	usDate	date			No	None	
5	usTime	time			No	None	
6	usDept	varchar(30)	latin1_swedish_ci		No	None	
7	usStatus	varchar(15)	latin1_swedish_ci		No	None	

Figure 4.20: Userlog table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(30)			No	None	AUTO_INCREMENT
2	sSender	int(15)			No	None	
3	sReceiver	varchar(30)	latin1_swedish_ci		No	None	
4	sDoc	int(30)			No	None	
5	sDate	date			No	None	
6	sTime	time			No	None	

Figure 4.21: share table

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(8)			No	None	AUTO_INCREMENT
2	extension	varchar(5)	latin1_swedish_ci		No	None	
3	description	varchar(150)	latin1_swedish_ci		No	None	
4	status	varchar(15)	latin1_swedish_ci		No	None	
5	image	varchar(150)	latin1_swedish_ci		No	None	

Figure 4.22: Allowdocumenttypes table

4.7 Evaluation of Result

The purpose of this work is to design an improved electronic document management system which will enhance security and reduce space used by documentation of the organization. There a study was ran to evaluate the technical aspect of the work and user interaction with the system

4.7.1 Technical Evaluation

A sample Microsoft word document, PDF and PNG file was uploaded into the server and was opened in the server file system to check the effectiveness of the encryption function of the system, as it is believed that only an active authenticated user can decrypt a document in the system, therefore the file system was open and all document has “AES” file extension format and when opened, what was displayed was different symbols. An accredited user now logged in into the system decrypt it and we have our decrypted document back as it was in the original document that was encrypted.

4.7.2 Users Evaluation

Because this system is interactive in nature, technical evaluation and performance measures are not sufficient for the evaluation process. Therefore, a user study was conducted to evaluate the system in action. The goal of this study is to confirm the feasibility of using the tool and to highlight points of failure that need to be addressed in future implementations.

4.7.2.1 Method

Prototype of the system was setup on three different desks and computers in separate offices. All three users were IT officer, a secretary and one Admin staff. Each was allowed to use the system for a week. An interview was now conducted with the three to collect a qualitative data based on the experience that each user had while using the system. Several interviews were held at the end of the users' working day to further obtain deeper understanding of their experiences and concerns about the system.

4.7.2.2 Results

All users were excited about the idea of automatically documenting their documents and also ensuring higher security of their document using this system. The system made it easier for them to find document, securely store document, share document and with less stress. One user pointed out that they would spend few minutes trying to find document from the large pile of available documents. Users were able to add comment on any document in case of any important note that need to be added for intended user to note before he/she uses the document. Users stressed the advantage of having the system embedded into daily operation as participants realized the benefits of this system in keeping track of their documents and ensuring the document is securely safe from unauthorized personnel.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

This chapter gives a summary of results, conclusion and recommendation based on the results and findings of the work. This chapter gives suggestion for future research related to this work.

5.1 Summary of Result

This work "An Improved electronic document Management System" has being designed taken Oyo state Housing Corporation as a Case study. The work sought to eliminate the problem related to documentation, improve security and credibility of document by automated encryption system as the ministry is expected to handle/possess quite a number of documents which need to be kept safe and credible. The project gave the organization the opportunity to uniquely secure document and yet ensure that limited space is used. An accredited user can login using the organizations URL to access or Upload files i.e. the user must be duly registered to access the system. The encryption decryption and archiving functions of the system is fully automated as all files uploaded into the system will be automatically encrypted and any file needed to be downloaded will be automatically decrypted while a user can also download a compressed version of their file or an archived version (as the case may be) uploaded into the system.

5.2 Conclusion

According to the document management needs at Oyo state Housing Corporation, this work designs and implements a system that facilitates secure documentation, fast retrieval, space management and longtime storage assurance. This work developed ausable, maintainable web application (Encodoc) which work fine with the documentation requirements of Oyo state Housing Corporation. This study emphasizes the practices used to build a usable and

maintainable secure document management system. The practices of this thesis work show that carefully designed usability evaluation is an effective way to locate the usability problems of an application (if any) and could consequently improve the application's usability.

Considering the various advantages of the improved EDMS, it will not be out of place to say that the improved electronic document management system will be of immense advantage to the teeming clients, users and the organization as a whole.

5.3 Recommendation

Considering the geometrical advancement in the information technology world and the large number of documents that are being handled by organizations daily combine with the need for accountability and proper document management, it is therefore highly recommended that the organization implement Encodoc. There are high economic advantages attached to this software on the part of the organization and users. Above all, it is user-friendly and documentation should be checked in case of future modification.

5.4 Contribution to the Knowledge

This work will lay a sound background for further research related to PHP Version of Document Management System with improve security and space management features. This work will enhance the understanding and need for users to ensure security and integrity of document used by them. It will allow user the choice of picking from either the java based EDMS or the new PHP based improved EDMS. Also, this work will protect document from unauthorized access and malicious attack thereby creating room for accountability (since document will secure, untampered and can be accessed by only authorized personnel over time) and give confidence and reliability to individuals and organization using the EDMS.

5.5 Suggestion for Future Research

The suggestion made here is that fingerprint verification can be researched to be added as authentication mode of accessing the system such that the user authentication medium will be by fingerprint scanning, because user might be careless with the password and should a user who is an admin be careless with his password he will void the effectiveness of the system.

REFERENCES

Akashah, P. A., Syamsul, R., Jusoff, K. & Christon, E. (2011). Electronic Document Management System. *World Applied Sciences Journal (Special Issue on Computer Applications & Knowledge Management)*, **12**: 55-58.

Alessandro, A. (2004). Online Auction System, Unpublished bachelor thesis, University of Bolzano/Bozen.

Almelkar, M. & Gandhe, S. T. (2014). Implementation of Lossless Image Compression Using FPGA. International Journal of Emerging Technology and Advanced Engineering. Vol 4: 2250-2459

Amir, M. B. S. (2007). Document Management System Portal (E-Tanah).

Anderson, R. J. (2001): Security Engineering: a guide to building dependable distributed systems. First edn. Wiley Computer Publishing.

Anwar, M. A. & Naseer, A. (2013). An e-Course file management system: A green campus initiative. Vol.3, No.1.

Bishop, M. (2003): Computer Security - Art and Science. First edn. Addison Wesley.

Björk, B. C. (2002). Electronic Document Management System in Construction. *ITcon Journal*, **8**: 105-118.

Cakiroglu, M. (2010).Software Implementation and Performance Comparison of Popular Block Ciphers on 8-bit Low-Cost Microcontroller. *International Journal of the Physical Sciences*. Vol. 5: 1338-1343.

Das, D., Lanjewar, U. A. & Sharma, S. J. (2013). Design an Algorithm for Data Encryption and Decryption Using Pentaoctagesimal SNS. *International Journal of Computer Trends and Technology (IJCTT),* **6**: 84-88.

Delonti (2014). Retrieved from http://www.delontiuniverse.com:8080/Intranet/ DocMgtHealthcare.aspx.

Diffie, W. & Hellman, M. (1976). New Directions in Cryptography, IEEE Transactions on Information Theory.

Doccept (2015). Retrieved March 10, 2015 from http://www.doccept.com/features

Dupuis, C. (1999). A Short History of Crypto. Retrieved from http://jproc.ca/crypto/crypto_hist.html.

eFileCabinet Ltd. (2015). History of Document Management [Online]. Retrieved March 10, 2015 from http://www.efilecabinet.com/document-management.

Forlanda, J. (2013). What is the Difference between Encryption and Cryptography? Retrieved from http://www.brighthub.com/computing/enterprise-security/articles/65254.aspx.

Grajeda, V. Z., Uribe, C. F. & Parra, R.C. (2006).Parallel Hardware/Software Architecture for the BWT and LZ77 Lossless Data Compression Algorithms.

Groenewald, T. (2004).Electronic Document Management: A Human Resource Management Case Study. *SA Journal of Human Resource Management,* **2**: 54-62.

Groenewald, T. (2004). Electronic Document Management: A Human Resource Management Case Study. *SA Journal of Human Resource Management.*Vol.2, No.1:54-62.

Halas, M., Bestak, I., Orgon, M & Kovac, A. (2012).Performance Measurement of Encryption Algorithms and Their Effect on Real Running in PLC Networks.

Hanafi, A. (2015). Lagos courts where syndicates issue fake CsofO, tax certificates. Retrieved from http://www.punchng.com/metro-plus/lagos-courts-where-syndicates-issue-fake-csofo-tax-certificates/.

Hyperoffice (2015). Retrieved from http://www.hyperoffice.com/intranet-software-solution.

Jeeva, A. L., Palanisamy V. & Kanagaram, K. (2012*)*.Comparative Analysis of Performance Efficiency and Security Measures of Some Encryption Algorithms. *International Journal of Engineering Research and Applications (IJERA).* Vol. 2, Issue 3: 3033-3037.

Kahanwal, B., Dua, K. & Singh, G. P. (2012).Java File Security System (JFSS).*Global Journals Inc.*, **12**: Version 1.0.

Kahn, D. (1967).The Codebreakers - The Story of Secret, Macmillian.

Kattan, A. (2006). Universal Lossless Compression Technique with Built-in Encryption.

Katz, J. (2007). Introduction to Modern Cryptography. CRC Press.

Keyes, J. (2012).Social Networking Tools to Transform Your Organization.

Knowledgeone Corporation (2005). Implementing Electronic Document Management with Knowledgeone Corporation.Pg 1-19.

Kodituwakku, S. R. & Amarasinghe, U. S. (2011). Comparison of Lossless Data Compression Algorithms for Text Data. *Indian Journal of Computer Science and Engineering.* Vol 1, No 4: 416-425.

Kodmelwar, M. K., Agarkar, M., Borle, A., Deshmukh, A. & Bhagat, M. (2012). Document Management System with Enhanced Security. *IOSR Journal of Computer Engineering (IOSRJCE),* **1**:18-23.

Logicaldoc (2015). Retrieved March 10, 2015 from http://www.logicaldoc.com/product/features.html

M-Files (2015). Retrieved March 10, 2015 from https://www.m-files.com/en/latest-ecm-features

Mandal, A. K., Parakash, C. & Tiwari, M. A. (2012). Performance Evaluation of Cryptographic Algorithms: DES and AES.*IEEE Student's Conference of Electrical, Electronics and Computer science.* Vol 41:1-5.

Mansoor, E., Shujaat, K. & Umer, B.K. (2013). Symmetric Algorithm Survey: A Comparative Analysis. *International Journal of Computer Applications* (0975 – 8887) Vol 61, No.20.

Maximum Compression (2011). Summary of the multiple file compression benchmark tests.

Mushtaque, M. A. (2014). Comparative Analysis on Different Parameters of Encryption Algorithms for Information Security. *International Journal of Computer Sciences and Engineering Open Access Research Paper.* Vol. 2: 2347-2693.

Mushtaque, M. A., Dhiman, H., Hussain, S. & Maheshwari, S. (2014). Evaluation of DES, TDES, AES, Blowfish and Twofish Encryption Algorithm Based on Space Complexity. *International Journal of Engineering Research & Technology (IJERT)*, Vol. 3 Issue 4.

Pai, B. T., Cheng, F., Lu, S. & Ruan, J. S. (2012). Sub-Trees Modification of Huffman Coding for Stuffing Bits Reduction and Efficient NRZI Data Transmission, IEEE Transactions on Broadcasting, vol. 58, No. 2.

Paperwise (2015).Retrieved fromhttp://www.paperwise.com/glossary-of-terms/.

Park, J. & Kim, S. (2010). Design and Implementation of E-Document Encryption System using Hash Algorithm. *International Journal of Database Theory and Application,* **3**: No 3.

Pinpoint (2015). Retrieved March 10, 2015 from http://www.lsspdocs.com/pinpoint-electronic-document-management/

Porter-Roth, B. & Porter-Roth Associates (2006). Applying Electronic Records Management in the Document Management Environment.*White paper*, 1-16.

Pu, I. M. (2006). Fundamental data compression. Elsevier, Britain.

Salleh, H., Zainon, N., Alshawi, M. & Sabli, A.M. (2011). The Implementation of Document Management System (DMS) In Managing Sub-Contracts Tenders: A Contractor's Perspectives. *International Journal of the Physical Sciences,* **6**: 3302- 3309.

SANS Institute (2001).History of Encryption.

Simar, P. S. & Raman M. (2011).Comparison of Data Encryption Algorithms. *International Journal of Computer Science and Communication*, Vol. 2: 125-127.

Singh, S. (2000). The Code Book: The Evolution of Secrecy from Ancient Egypt to Quantum Cryptography.

Singhal, N. & Raina, J. P. S. (2011).Comparative Analysis of AES and RC4 Algorithms for Better Utilization. *International Journal of Computer Trends and Technology.*

Umunnah, U.O. (2012). Lack of Accountants And Accountability In Nigeria Financial District[s]: Is Sheer Irresponsibility. Retrieved from http://www.gamji.com/article5000/NEWS5499.htm

Wikipedia: http://en.wikipedia.org/wiki/Document_management_system. (Accessed 2015, March).

Yousuf, M. & Sumer, M. T (2011). Secure Emails: An Intergrity Assured Email Systems Using PKI.

Zotos, K. & Litke, A. (2015). Cryptography and Encryption. Retrieved May 11, 2015 from http://arxiv.org/ftp/math/papers/0510/0510057.pdf.