

**DESIGN AND IMPLEMENTATION OF AN ONLINE
DOCUMENT MANAGEMENT SYSTEM**

BY

OLAYINKA, OLUSEGUN SOLOMON

16010301032

SUBMITTED TO

**THE DEPARTMENT OF COMPUTER SCIENCE AND
MATHEMATICS, COLLEGE OF BASIC AND APPLIED
SCIENCES, MOUNTAIN TOP UNIVERSITY,
OGUN, NIGERIA**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD DEGREE OF BACHELOR OF
SCIENCE (B.SC) IN COMPUTER SCIENCE.**

NOVEMBER, 2020

CERTIFICATION

This is to certify that this project, Document Management System was carried out by me Olayinka Olusegun Solomon (Matriculation Number: 16010301032) and duly supervised by Dr Omotunde Ayokunle.

.....

Dr. Omotunde Ayokunle

(Supervisor)

.....

Date

.....

Dr. I. O. Akinyemi

(Head of Department)

.....

Date

DEDICATION

This project is dedicated to the Almighty God.

ACKNOWLEDGEMENT

My sincere gratitude goes to the Almighty God for his everlasting grace and for giving me the wisdom, knowledge and understanding through the entirety of my programme.

I greatly appreciate the efforts of my project supervisor Dr Omotunde Ayokunle, who painstakingly took his time to ensure that this project becomes a successful one.

I would like to thank the H.O.D of Computer Science and Mathematics Dr I. O. Akinyemi, my lecturers: Dr Omotunde Ayokunle, Dr F. A. Kasali, Dr F. A. Oladeji, Dr P.A. Idowu, Mr O.J Falana, Dr M. O. Oyetunji, and Mrs O. Taiwo for their efforts and the knowledge they have imbibed in me and all of the academic and non-academic staff of the department of computer science and mathematics.

My appreciation goes to my Parents and Siblings, thank you all for your extraordinary support in every ramification. God bless you greatly.

Finally, I would like to thank all my colleagues in the department of computer science and mathematics, for the years we have been together and lessons we have learnt from one another.

ABSTRACT

For an average person, where and how to store files and records might be a simple and accomplishable activity but at the expense of your time, human effort and physical space depending on how bulky the documents might be. The major downside comes from the inefficient and unreliable handling of data by the person, an additional drawback is the loss of the important documents in the event of a disaster or an accident, as there are little or no backups of these stored documents, which are common problems in a lot of organizations with inadequate record management infrastructures.

The goal of this project research is to provide users with a web-based management system that allows easy access, organization and retrieval of files at any time and anywhere through an online browser on any device; the system offers the essential features of a web-based file system and additional document management options, such as a powerful search engine incorporated into the application, automated version control to monitor changes on the documents in the application.

TABLE OF CONTENT

Title	Page
CERTIFICATION	1
DEDICATION	2
ACKNOWLEDGEMENT	3
ABSTRACT	4
TABLE OF CONTENT	5
LIST OF FIGURES	8
LIST OF TABLES	11
CHAPTER ONE	12
INTRODUCTION	12
1.1 Background to the Study	12
1.2 Statement of the problem	13
1.3 Aim and objectives	13
1.4 Research Methodology	14
1.5 Scope of study	15
1.6 Significance of the study	15
1.7 Definition of terms	16
CHAPTER TWO	19
LITERATURE REVIEW	19
2.0 Introduction	19
2.1 Document Management Systems	19

2.1.1	Evolution of Document Management Systems	20
2.1.2	Survey report on the use of document management systems	24
2.2	Basic Aspects of Computer Security	30
2.3	Cryptography, Encryption and Decryption	30
2.4	Encryption Algorithm Classification	32
2.5	Review of related existing systems	40
2.6	Review of related works	41
2.7	Summary of the Literatures Reviewed	45
CHAPTER THREE		46
RESEARCH METHODOLOGY		46
3.1	Introduction	46
3.2	Software Development Life Cycle	46
3.2	System Analysis	47
3.3	System Design	49
CHAPTER FOUR		70
RESULT AND DISCUSSION		70
4.0	Introduction	70
4.1	System Hardware Requirements	70
4.2	System Software Requirements	70
4.3	Implementation Procedure	71
4.4	Developed System Images	71

CHAPTER FIVE	88
SUMMARY, CONCLUSION AND RECOMMENDATION	88
5.0 Introduction	88
5.1 Summary	88
5.2 Conclusion	88
5.3 Limitation to the study	88
5.4 Contribution to Knowledge	89
5.5 Recommendation for Further Studies	89
REFERENCES	90
APPENDIX	96
SOURCE CODE	96

LIST OF FIGURES

Figure	Page
2.1 Different models of online/other document management systems using patterns	12
2.2 DM Systems used by Project Managers	14
2.3 DM Systems used by Project Managers (by type of organization)	16
2.4 DM Systems used by Project Managers (by age groups)	17
2.5 What prevents effective usage of DM tools	18
2.6 Encryption and Decryption process	20
2.7 Comparison of Encryption Algorithms based on scalability	25
2.8 Comparison of Encryption Algorithms based on time taken to encrypt	27
3.1 Docaris' DFD Level 0	38
3.2a Docaris' DFD Level 1 (Administrator)	39
3.2b Docaris' DFD Level 1 (User)	40
3.3 Docaris' Use Case Diagram	42
3.4 Docaris' Conceptual Data Model	44
3.5 Docaris' Logical Data Model	45
3.6 Docaris' Physical Data Model	46
3.7 Docaris' Class Diagram	47
3.8 User/Administrator Login Activity Diagram	48

3.9	Document Search Activity Diagram	49
3.10	Document Upload Activity Diagram	50
3.11	Document Download Activity Diagram	51
3.12	Document Edit Activity Diagram	52
3.13	Document Delete Activity Diagram	53
3.14	Document Share Activity Diagram	54
4.1a	Docaris Login Page (Desktop View)	61
4.1b	Docaris Login Page (Mobile View)	62
4.2a	Docaris Dashboard Page (Desktop View)	63
4.2b	Docaris Dashboard Page (Desktop View)	64
4.3	Add Document View	65
4.4	View Documents	66
4,5	Document Workflow View	67
4.6	Document Permission View	68
4.7	Document Activity View	69
4.8	Document Tags View	70
4.9	Document Search with Text Highlighting	71
4.10	Workflow Configuration	72
4.11	System Configuration	73
4.12	User Management	74

4.13	Server Logs	75
4.14	File uploaded to Database after encryption	76

LIST OF TABLES

Table		Page
2.1	Comparison of Encryption Algorithms based on Architecture	26
2.2	A review of related systems	29

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

In the early stages of document management, documents were stored and arranged in filing cabinets which proved to be a reliable system. However, this method of storing files became cumbersome due to a large amount of space used costing the companies resources, time and energy whenever a document needed to be filed or accessed. The physical storage of documents also had several problems like files being untraceable, stolen, destroyed in accidents or natural disasters and even problems caused by users of the documents within the organization.

The era of software systems to manage paper-based documents that handled written and printed documents, images, prints and so on quickly sprung up in the Eighties and over the years developers started creating systems capable of managing electronic documents which resulted in systems capable of managing any sort of file format that could be stored on the network. Eventually, syncing and real-time sharing platforms such as Google Docs emerged with the development of Web 2.0 in 2004 (O'Reilly, 2005).

Recently, document management systems are commonly subject to attacks by cybercriminals and malicious users which makes a robust security system a compulsory feature of data storage systems. This security problem is usually solved with cryptography approaches (Kahanwal, Dua, & Singh, 2012).

Encryption works by taking plain text and converting it into cyphertext, which is formed from seemingly random characters. Only those who have a special key can decrypt it (ATP Electronics, Inc., 2019).

The major use of data encryption is to secure digital data secrecy because it is stored on computer systems and transmitted using the net or other computer networks. Modern encryption algorithms that are a critical part in the security of IT systems and communications have replaced the obsolete Data Encryption Standard (DES).

These algorithms offer mystery and drive key security activities including integrity, non-renouncement and authentication. Authentication allows for the verification of a message's origin, and integrity provides proof that a message's contents haven't changed since it was sent while non-repudiation ensures that a message's sender cannot deny sending the message (Lord, 2020).

Therefore, Docaris which is being developed for this research work will be using an encryption and decryption algorithm to guarantee the document management system is secure alongside advanced search features, file versioning and user-friendly features to make the software as efficient and usable as possible by individuals and businesses.

1.2 Statement of the problem

Documents are very important assets of any organization because they contain information about the organization and its activities. Storing and managing these organizations' assets efficiently has been a major challenge to a lot of organizations, therefore a system that will provide a reliable means of storage and management for the organizations' documents has to be developed.

1.3 Aim and objectives

This research work aims to design and develop a secure online document management system while the specific objectives are:

- I. To study existing works on document management systems, encryption and decryption algorithms.
- II. To design and develop a system that stores and delivers files of various formats in seconds and maintain an efficient workflow of documents using Java for the business logic.
- III. To design a system with AES encryption and automated version control for the documents.
- IV. To design an efficient database that stores reliable and up to date records of files.
- V. To design a user-friendly and responsive user interface and smooth user experience to improve the productivity of the users of the system.

1.4 Research Methodology

The Document Management System is going to be designed as a multi-tier web-based application. Tools to be used for the work include HTML, CSS, Sass, JavaScript, AngularJs, Java, Hibernate, H2 database and Postgresql.

The three tiers architecture adopted for this work:

- **The presentation tier:** The top-most level of the application is the user-interface which interacts with end-users and translates tasks and results in an understandable format for the users. This layer is built using: Html, CSS, JavaScript and AngularJs which will communicate with other tiers in the network through API calls.
- **The application tier (or business logic):** This is the middle layer which controls and manages the app's functionality and performs detailed processing. The layer is going to be built with the Java programming language

and Hibernate framework which will expose API endpoints for connection to the client.

- **The Data-tier:** Houses database servers where information is stored and retrieved. H2 database or PostgresSql will be used to manage this layer.

In the course of gathering data, the following activities are going to be carried out:

1. Research's related work will be reviewed to understand in details the concept of document management systems.
2. Existing systems will be analyzed to understand the limitations of the applications.

1.5 Scope of study

The focus of this research is to develop a system that can be used by individuals or businesses secure storage and retrieval of documents. The system will provide users with the facilities of quick access, easy storage, and security of the documents stored.

1.6 Significance of the study

Management of documents is an increasingly important need in today's society as people are actively using and collecting large amounts of files and documents daily, these documents are created from data being generated from organizations, markets, institutions, individuals, researches, database and transactional reports, and lots of other sources. Paper documents have proven to be reliable storage of information since earlier days. However, paper can become costly and, if used disproportionately, wasteful.

The online cloud-based document management systems have numerous advantages over exchanging files using e-mail or storing files traditionally. These benefits are reduced document duplication, a single point of document storage, the

proper(automatic) version management, better document visibility, improved collaborative work, reduced time wasted for local classification and storage, better information circulation, etc. (Course Hero, n.d.).

This software helps the businesses to manage their ever-increasing number of varied documents by storing them into a single hub which might be accessed by authorized users and managed accordingly.

1.7 Definition of terms

- **Advanced Encryption Standard (AES):** is a symmetric key block cypher algorithm and United States government standard for securing and classifying data encryption and decryption.
- **Application Programming Interface (API):** is a collection of procedures and methods allowing the creation of utilizations that access the features or data of an OS, application or other services.
- **Cloud computing:** Cloud computing is the accessibility of Computer system resources when requested, particularly data storage and processing power without direct dynamic administration by the client. The term is mostly used to describe data centres available to several users over the web.
- **Cross-platform:** In software development, cross-platform software is computer software that has an implementation on multiple computing platforms.
- **Database:** a large collection of data organized especially for rapid search and retrieval.
- **Document:** according to ISO 12651-2, a document is "recorded information or object which may be treated as a unit".

- **Encryption:** In cryptography, encryption is the act of converting information to an encoded format. This process transforms the initial representation of the information, known as plaintext, into an alternative form referred to as ciphertext.
- **File sharing:** File sharing is the act of disseminating or giving access to electronic media, for example, computer programs, documents or digital books, multimedia and so on.
- **Responsive design is a graphic user interface (GUI)** design approach that is utilized to make content that adjusts easily to several screen sizes and ratios. Designers size elements in relative units (%) and apply media queries, so their designs can automatically adapt to the browser space to ensure content consistency across devices.
- **RESTful Web Services** are REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are lightweight, very scalable and maintainable and are very commonly used to create APIs for a web-based application.
- **Retrieval:** the process of accessing information from memory or other storage devices.
- **Server:** A running instance of an application capable of accepting requests from the client and giving responses accordingly.
- **Software:** a collection of programs, procedures, and related documentation related to a system and particularly a system.
- **Storage:** The retention of retrievable data on a computer or other electronic system.

- **Version control system (VCS)** – is a type of software tool that helps record changes to files by keeping a track of changes done to the code (or document in this context). It contains all the edits and historical versions (snapshots) of the project.
- **Web Browser:** A software application used to find, retrieve and show contents on the internet, including videos, web pages and other files.
- **Web Application:** An application program stored on a remote server and delivered over the net through a browser interface.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

This chapter provides a review of the literature about Online Document Management Systems as a motivation for this work. In the first section of this chapter, research about the evolution and a survey on the use of document management systems are discussed, the next section entails a discussion on research that focuses on cryptography, different kinds of encryption and their comparison. the last section provides details about related works within the field of work discussed.

2.1 Document Management Systems

There are numerous definitions for Document Management System, some of which are:

- Document management, often referred to as Document Management Systems (DMS), is the use of a computing system and software to store, manage and track electronic documents and electronic images of paper-based information captured through the use of a document scanner (Aiiim, 2020).

A document management system is proposed to help the business in dealing with the creation, storage and flow of documents by offering a centralized repository (Zammit, 2020).

- Document management is the process of saving, localizing, editing, and sharing various kinds of electronic documents in a way that's ideal and productive for organizations (Zeeman, 2019).
- An online Document Management System (DMS) is a system for storing and tracking documents. It began as a way to convert paper documents to digital documents. As a result, a DMS is frequently called an electronic file cabinet but it's more than that today (Schmidt, 2019).
- Document Management software automates the process of document management from development to storage to delivery across a business, increasing productivity and decreasing the expense and burden of maintaining paper records. (Capterra, 2020).
- 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).

From the definitions above, we can define that a Document Management System is a software that stores and manages documents into a centralized hub. Document Management Systems usually provide security, versioning, storage, metadata, retrieval and indexing capabilities.

2.1.1 Evolution of Document Management Systems

Document management had existed for several years where files of any sort were primitively stored in filing cabinets and file storage rooms but this method of storage proved to be inefficient as documents quickly

increased daily and accessing the stored documents would cost businesses a great deal of time and energy combing through the massive stacks of documents.

The beginning of the development of software applications to manage paper-based documents started in the Eighties. These systems were able to manage paper documents, that concerned written and printed documents, images, prints so on. Throughout this era, merely proprietary file types, or a restricted range of file formats were being managed by the foremost primitive electronic document management systems. Gradually, developers started designing systems capable of managing electronic documents – documents or files created on computers, and principally stored on users’ native file-systems, that led these systems to evolve to a degree where they could manage any variety of file format that could be kept on the network (Wikipedia, 2020). Eventually, with the development of Web 2.0 in 2004 (O’Reilly, 2005), real-time sharing and syncing platforms emerged, like Google Docs.

The academic analysis in the field of contemporary, particularly cloud-based document management systems (Gilson, 2015) and its usage for cooperative project work are slowly growing. For example, a search using keywords ‘google docs’ in Mendeley returns around 400 results and Emerald Insight returns 190 results. A partial analysis of the utilization of Google Docs within a specific domain has been carried out by (Mansor, 2012) – a collaboration between academics, and (Blau & Caspi, 2009; Suwantarathip & Wichadee, 2014; Watson, 2006; Zhou, Simpson, & Domizi, 2012). Google Docs has been studied by (Tan & Kim, 2015) to

verify the acceptance of the software by users within an organizational setting. They detected that the perceived quality and satisfaction positively affect the intention to continue using such tools.

Today, there are various alternatives in the context of document management. These systems are substantially more user-friendly. Adding files to your document management system is quick and easy. The system allows you to search for files in a matter of moments by any means. Along these lines, an ever-increasing number of organizations are losing the filing cabinets and advancing toward electronic systems.

This is saving resources for these organizations. Not only do they no longer need to spend additional money on an extra room, but they also don't need to stress over workers sitting around idly attempting to find documents. The time and resources saved adds up to thousands of dollars each year, even for small companies. (eFileCabinet, 2014).

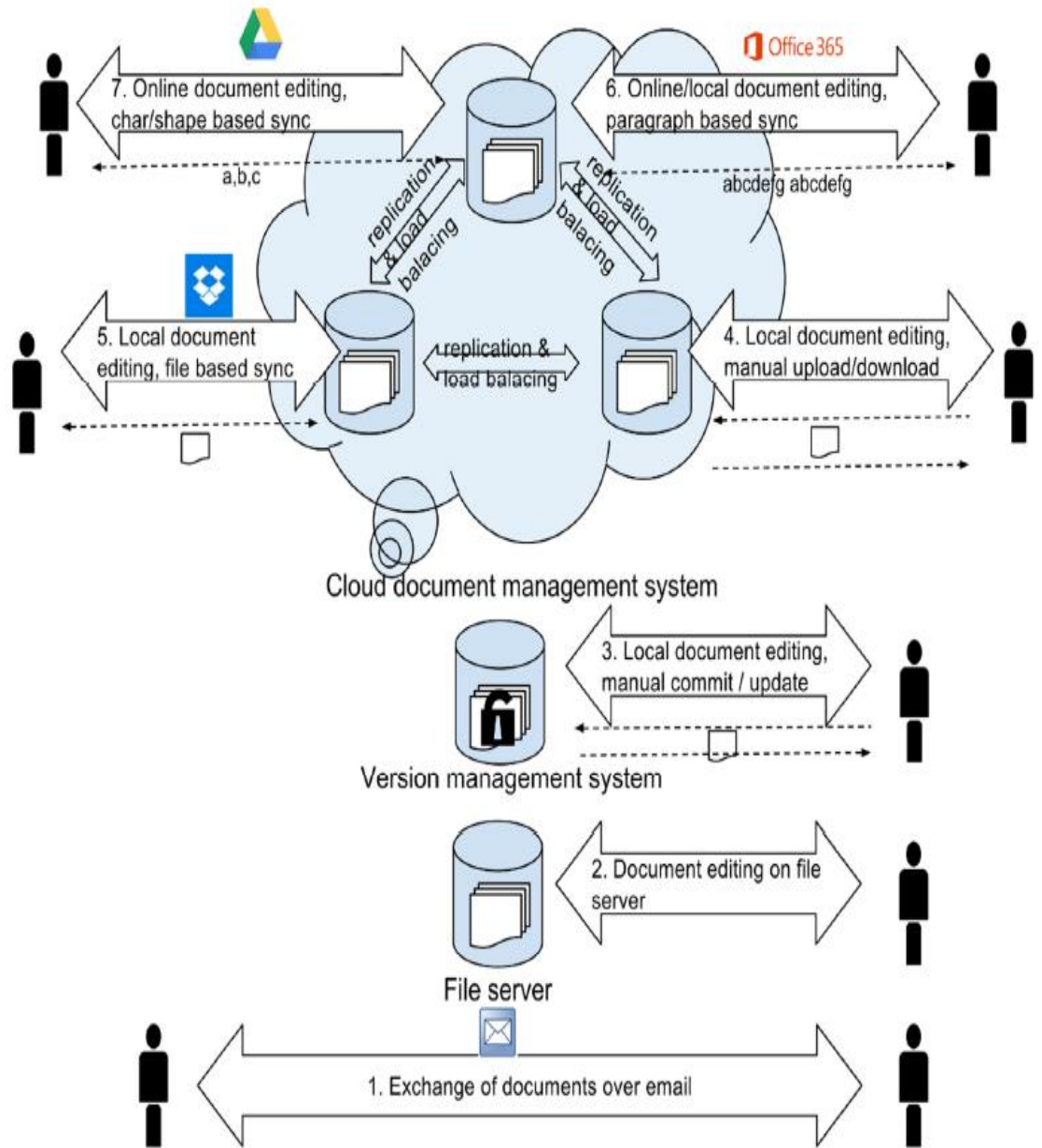


Fig. 2.1: Different models of online/other document management systems using patterns (1stwebdesigner, 2018)

2.1.2 Survey report on the use of document management systems

Using a web survey, (Tomislav Rozman, 2017) Collected data on document management and project participant demographics (country, age, the form of organization, educational background), experience in project management (proposal planning and scheduling, number of coordinated/collaborated/administrated projects), types of projects (source of funding, hierarchical structure), motivation (self-starting, being part of the organization, leadership, reason for participation).

The survey was constructed using Google Forms and the invites were sent via e-mail and posted in LinkedIn groups. The research included 44 individuals who supervised 244, collaborated in 544 and administered 484 EU programs. The data collected was analyzed using Google Sheets and IBM SPSS Tools. First, a univariate analysis was performed and then a bivariate analysis was used to explore the association between variables. Descriptive statistics were calculated using basic statistics (averages, etc.). Pivot tables were used for summarizing results. Correlations were discovered using Pearson's correlation coefficient. There were no missing data to be dealt with, so all the questions that used the 7-level Likert scale had to be answered. Answers to open questions were grouped into classes (coded).

Many projects are still being handled without a shared documentation framework (32%). Among other things [Figure 2.2], Google Drive and Dropbox are the most commonly used systems with an approximately equal share (GDrive-34.1% and Dropbox 33.5%).

DM system (by the number of projects managed)

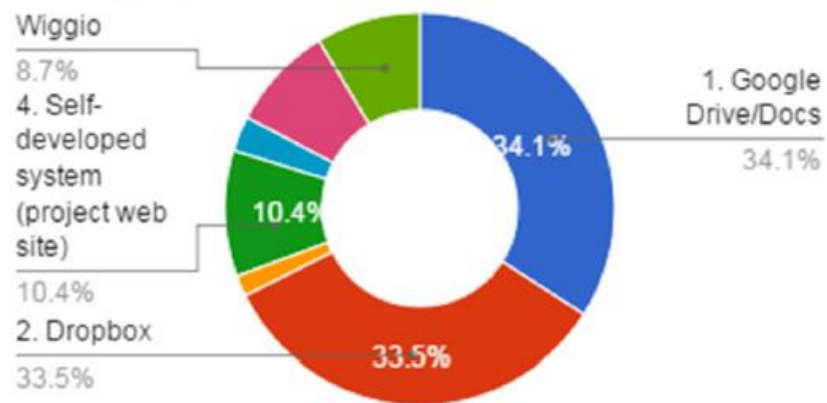


Fig. 2.2: Document Management (DM) systems used by project managers.

If only the answers “which kind of document management (DM) systems are used” are observed and grouped by user types [Fig. 2.3], we find out that Google Drive is mainly used by project managers, which work as consultants or come from private organizations. The reason could be in this product’s attractive pricing, quick learning curve and broad product coverage. Public, governmental and academic institutions mainly use Dropbox or a self-developed system.

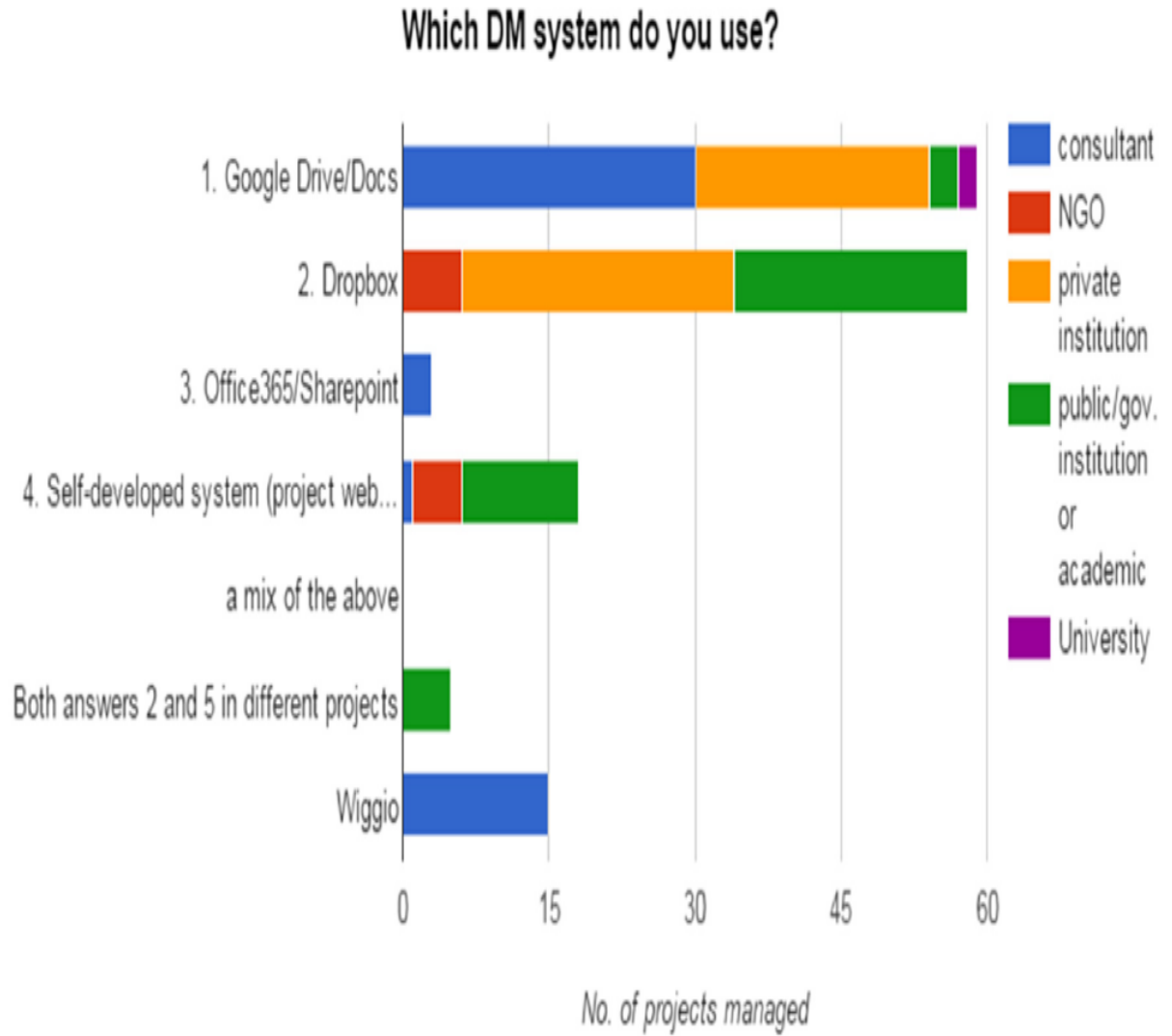


Fig. 2.3: DM systems used by project managers (by type of organizations)

An analysis of DM systems usage by age groups shows that generations 1961-1970 and 1944-1960 are less keen to use the cloud-based DM system [Fig. 2.4].

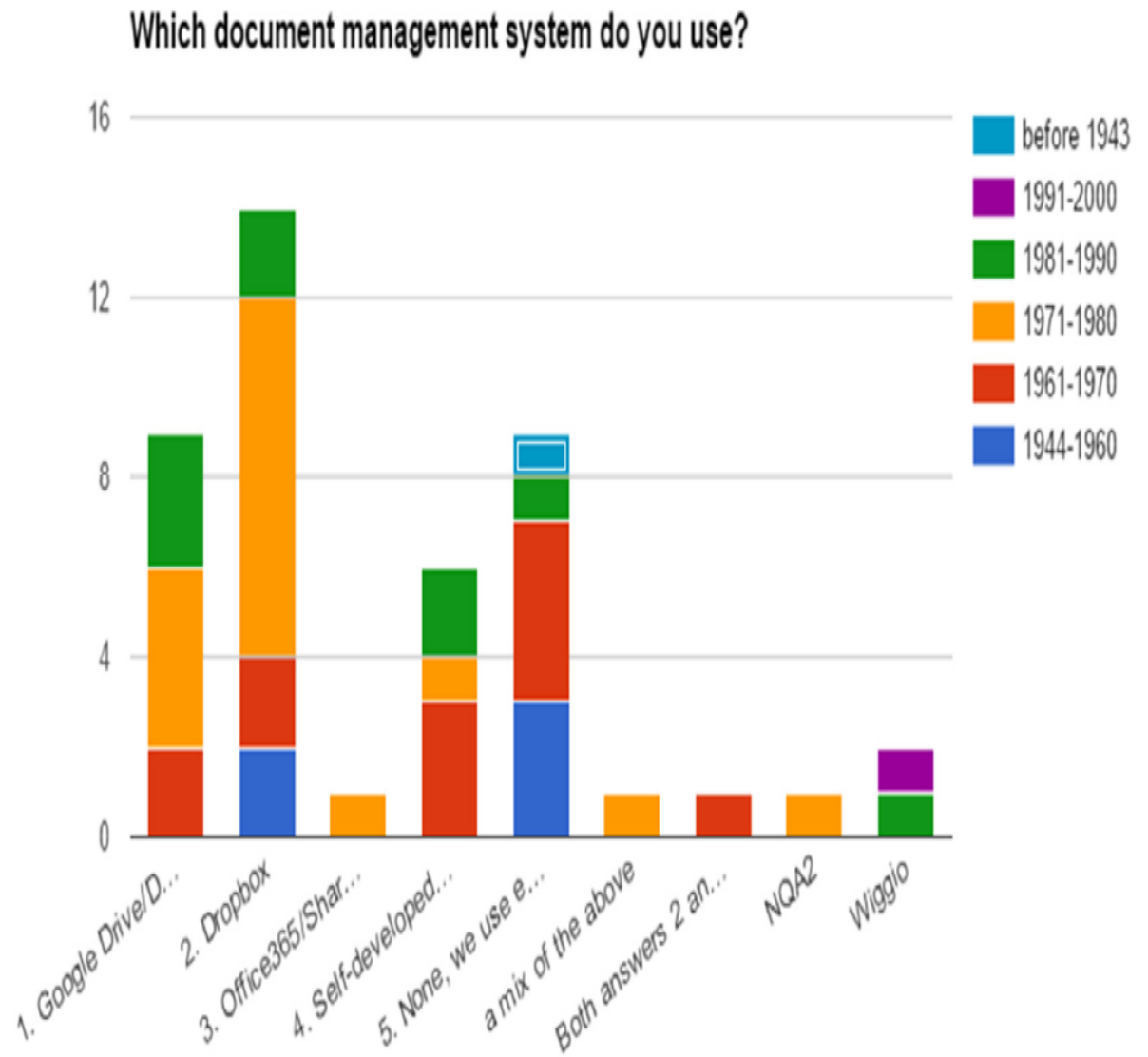


Fig. 2.4: DM systems used by project managers use (by age groups)

The two main obstacles for using cloud-based DM systems are:

1. The system is too cumbersome (it takes several clicks to perform a task) and
2. Security concerns. Other responses are presented in [Fig. 2.5].

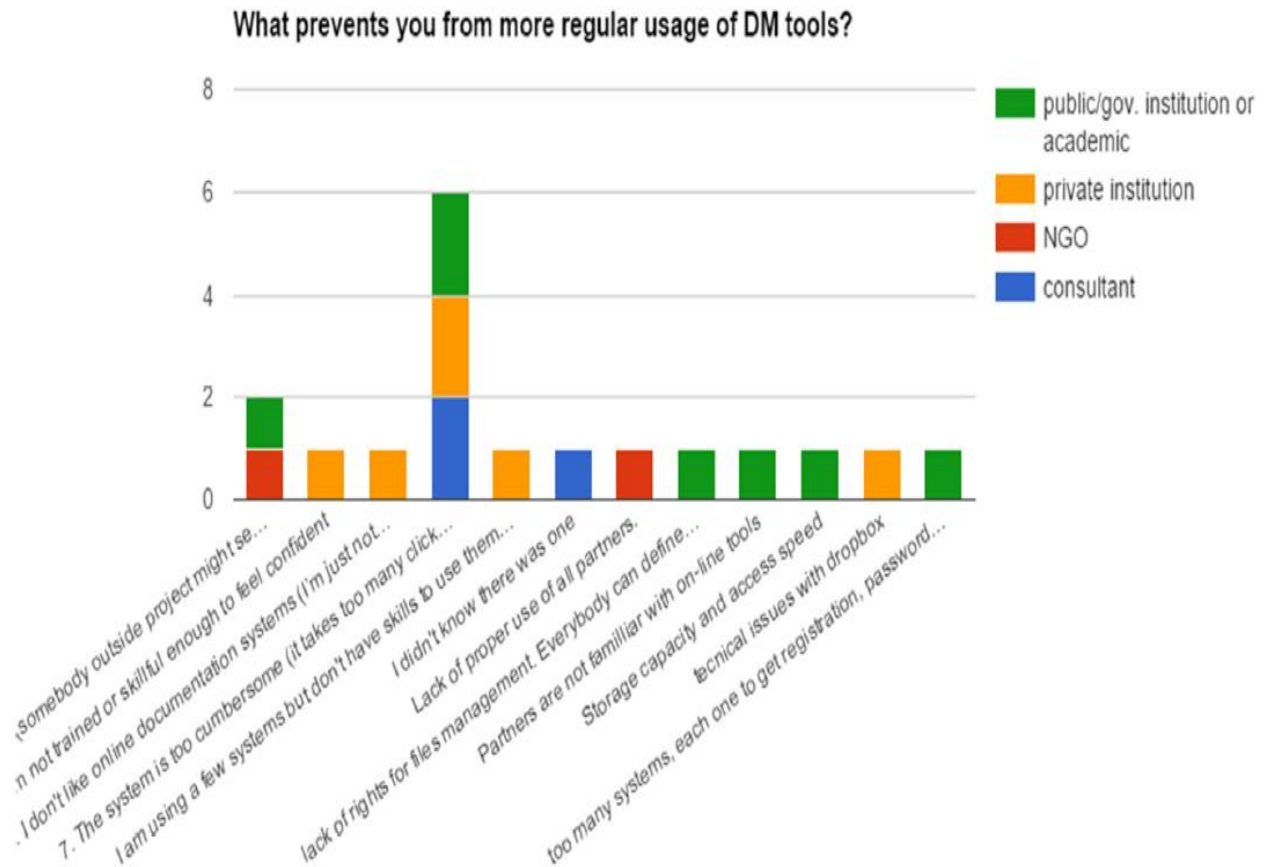


Fig. 2.5: What prevents effective usage of document management tools

2.2 Basic Aspects of Computer Security

Security refers to the preservation and protection of data in computer systems of an enterprise. Security is usually divided into safety resources, network security, security location where the data (server, etc.) and security services (Stosic, 2013). Security is based on four essential steps as follows:

- Evaluation: Estimate the likely risks and predictions for their removal.
- Protection: Avoid possible attacks to reduce the likelihood of compromising the system.
- Discovery: The process of recognizing the attack.
- Answer: A recovery with the possibility of further work or restoration of the system itself.

Three basic principles of information security make up the trinity of "great":

1. Confidentiality - an attempt to prevent the intentional, unauthorized disclosure,
2. Integrity - data is a system and as such must remain and must not be changed,
3. Availability - only certain staff can access the data.

2.3 Cryptography, Encryption and Decryption

Data Encryption is the process of converting human-readable plain text into an encoded format that only authorized persons or entities can access. Data security is an essential part of an Individual/organization which can be achieved through various methods. The encrypted data is secure for some time but it is still vulnerable to attack by a hacker trying to steal the data. There are numerous algorithms available in the market for encrypting the data. Encryption Key has a major role in the overall process of data encryption.

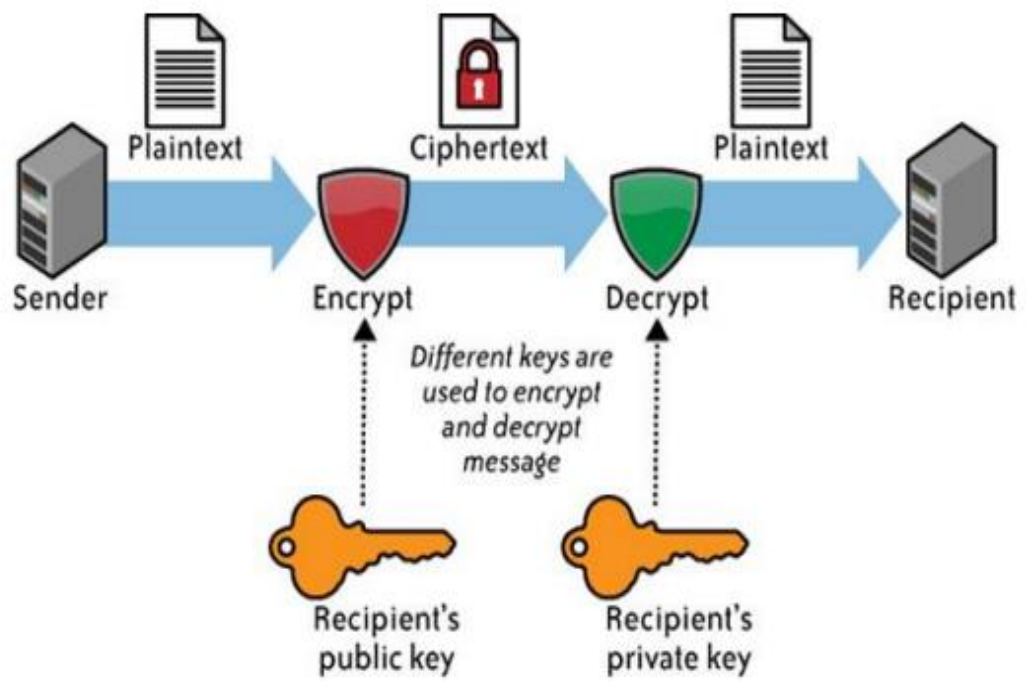


Fig. 2.6 Encryption and Decryption process

2.4 Encryption Algorithm Classification

Encryption methods can be classified into two categories:

- a) Symmetric Key and
- b) Asymmetric Key (Singhal & Raina, 2011).

1. Symmetric Key

In the design of symmetric key encryption, the same key is used for both the encryption and decryption process which ensures the key is kept private and the secrecy is maintained. Symmetric algorithms do not consume a lot of computing power and have high speed during the encryption process. During the encryption, a block cypher and a key are taken as the input which produces the output block that is the same size as the symmetric key encryption. Generally, symmetric-key cryptosystems are much faster than the asymmetric key cryptosystems. The performance evaluation of the following commonly used symmetric key encryption techniques will be taken into consideration: DES, 3DES, Blowfish, and AES.

Types of commonly used Symmetric Key Encryption

- **DES (Data Encryption Standard):** The first encryption standard that NIST (National Institute of Standards and Technology) recommended was DES. It is based on an algorithm called Lucifer, proposed by IBM. In 1972, IBM designed DES and it was adopted by the U.S. Government in 1974 as the standard encryption method (Mandal, Parakash, & Tiwari, 2012). It is a Feistel Network-based symmetric key block cypher encryption algorithm. DES uses a 64-bit text block and a 56-bit key length, performing a total of 16 rounds of data encryption processing. The key was 64 bits in DES, but IBM decided

to use 56-bit key length for encryption due to some restrictions from NSA (National Security Agency) and the remaining 8 bits are used as a parity bit for error detection, also using 8 boxes. The 64-bit block is split into two equal parts by DES and then the F-function is applied to each part. The F-function performs four distinct tasks: expansion, key mixing, replacement, and permutation. The same encryption process is used to decrypt in DES.

- **3DES:** 3DES, which comes from DES, was published in 1998. DES uses a 56-bit key, but 3DES uses a total size of 168 bits with 3 different keys (Halas, Bestak, Orgon, & Kovac, 2012). All keys are the same key or first key, and in 3DES, the third key may be the same. It also splits the text into 64-bit blocks, uses 8 S-boxes and performs 48 rounds of processing. 3DES is more complex and designed to re-protect data from various attacks. By applying DES encryption three times, 3DES encrypts data. The U.S. also authorizes 3DES. To be used by the government because of its higher security (Mushtaque, 2014).
- **Blowfish:** Blowfish was built in 1993 by Bruce Schneier. The Secure Socket Layer and other programs use a fast and simple block encryption algorithm. Blowfish is based on a 64-bit block and a 32-448-bit key size supported by the Feistel Network. It contains 4 s-boxes and conducts 16 rounds of processing. In this Key Expansion and Data Encryption algorithm, two main functions are carried out. The S-boxes in blowfish are key-dependent (Simar & Raman, 2011)
- **AES (Advanced Encryption Standard):** AES is a cypher encryption algorithm for symmetric key blocks designed in 1998 by Vincent

Rijmen and Joan Daemen. It is based on the Feistel network and supports 128-bit block sizes and 128, 192- and 256-bit key lengths (Lake, 2020). 10, 12 or 14 rounds are performed by AES and the number of rounds depends on the key. This means that AES performs 10 rounds for the 128-bit key length, it performs 12 rounds for the 192-bit key and 14 rounds for the 256-bit key. In AES, some steps are carried out in each round. Initial-round, Key-expansion, Rounds and Final-rounds. Sub-byte generation, Shift-rows, Mix-columns and Add-round keys are performed in Rounds, while the same functions are performed in Final rounds step except for the Mix-columns function (Simar & Raman, 2011).

Limitations of Symmetric Key Encryptions

- **DES:** Due to its key length of 56 bits, DES does not provide good protection. DES can easily be cracked by 2^{56} imagination. Initially, DES was recognized as a standard algorithm with good protection, but often when Brute force attack cracked DES. DES wasn't build run on slow software. So, DES isn't a stable encryption algorithm.
- **3DES:** 3DES overcomes the DES dilemma, but 3DES also has some disadvantages. 3DES conducts DES operation three times (i.e. uses 3 separate keys of a larger size ($3 \times 56 = 168$ bits)) to encrypt data, taking almost 3 times more space than DES. Since it offers a high degree of protection compared to DES, this is why 3DES is used by the U.S. Government of the world.

- **Blowfish:** Blowfish is a very stable algorithm, but the initial 4 rounds of blowfish are observed unprotected from the second-order differential attack.
- **AES:** No kind of weakness has been found in AES. Some initial rounds of AES are observed unprotected, i.e. the initial round can break by the square process.

Table 2.1 shows the result of the comparison of encryption algorithms base on architecture and from figure 2.6, (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 based on 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 these encryption algorithms based on their architecture is shown in table 2.1 while their comparison based on scalability (Memory usage and Performance) is shown in figure 2.7 and their comparison based on the time taken to encrypt various numbers of 16-byte blocks of data is shown in fig 2.8.

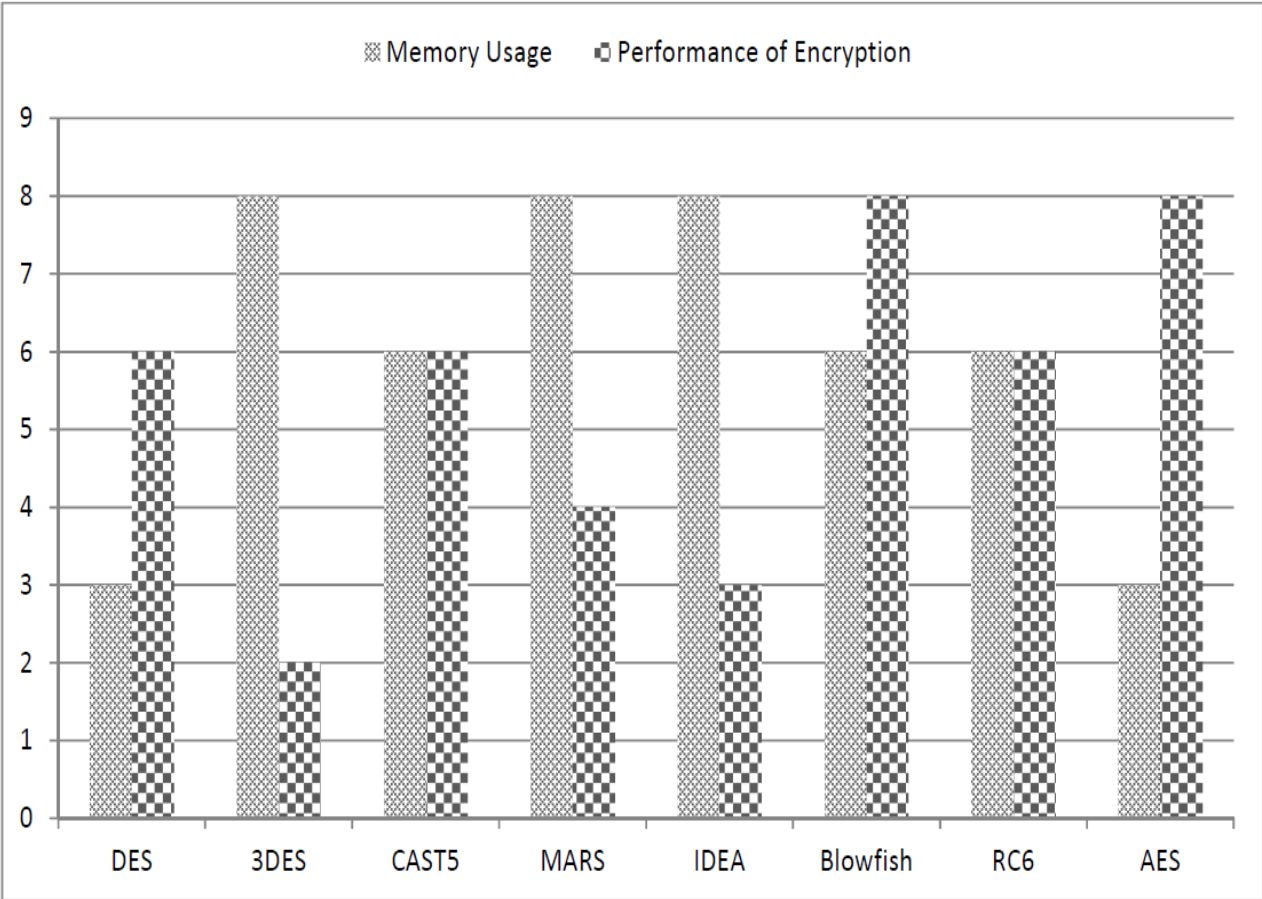


Fig. 2.7 Comparison of Encryption Algorithms based on scalability (Memory usage and Performance)

Characteristics	DES	3DES	Blowfish	AES
Key Length	56 bits	168bit	32 – 448 bits	128, 1982 or 256 bits
Rounds	16	48	16	10, 12 or 14 (depends on the size of the key)
Block Size	64 bits	64 bits	64 bits	128 bits
Speed	Slow	Very Slow	Faster	Fast
Security	Not Secure Enough	Adequate Security	Adequate Security	Excellent Security
Structure	Feistel Network	Feistel Network	Feistel Network	Feistel Network
No. of S-boxes	8	8	4	1

Table 2.1 Comparison of Encryption Algorithms based on Architecture

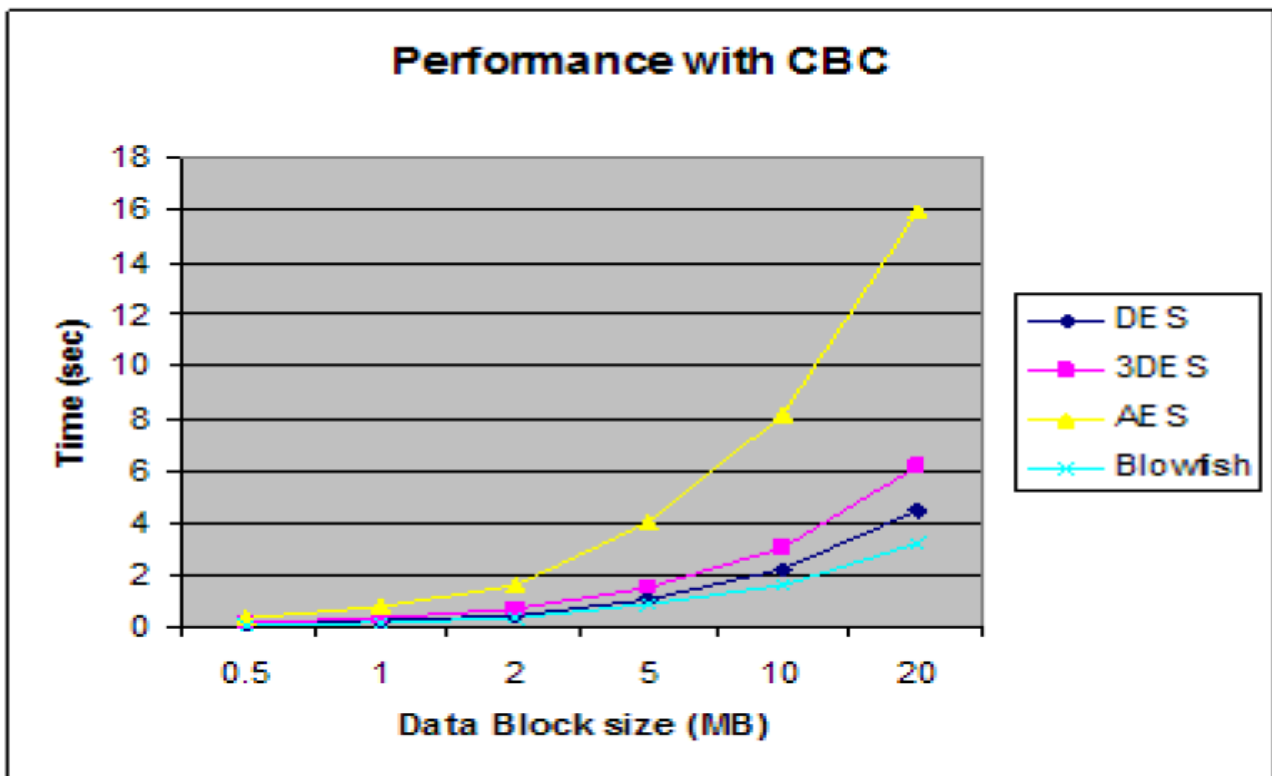


Fig. 2.8 Comparison of Encryption algorithms based on the time taken to encrypt various numbers of 16-byte blocks

2. Asymmetric Key

Asymmetric key encryption is a method in which keys are different for encryption and decryption. They're also classified as public-key encryption. One of these keys is written or made public and the other is kept secret. If the locking/encryption key is released, the device allows private contact from the public to the owner of the unlocking key (Yousuf & Summer, 2011). If the unlock/decrypt key is published, the device can act as a signature verifier for documents locked by the owner of the private key.

Public key methods are vital because they can be used to securely transfer encryption keys or other data even when both users have no opportunity to agree on a private algorithm secret key (Jeeva, V., & Kanagaram, 2012).

The keys used in public-key encryption algorithms are typically much longer, which increases the protection of the data being transmitted. Asymmetric encryption algorithms require at least a 3,000-bit key to achieve the same level of security as a 128-bit symmetric algorithm. Public key encryption has a mathematical function bias, computationally intensive. RSA (Rivest-Shamir-Adleman) Algorithm and Diffie-Hellman Algorithm are forms of public-key asymmetric encryption (Diffie & Hellman, 1976).

2.5 Review of related existing systems

Product Features	Zoho Docs	Logical Doc	Sharepoint Online	PDFelement	Docaris (Proposed System)
File Support					
System Support	Desktop & Web	Web-based	Web-based, Android, iOS & Windows mobile	Windows, Mac, Android & iOS	Web-based
Cloud Storage	Yes	Yes	Yes	Yes	Yes
OCR Feature	No	Yes	No	Yes	Yes
Sync	Yes	Yes	Yes	No	Yes
Files Encryption	Yes	Yes	Yes	Yes	Yes
Co-author in real-time	Yes	Yes	Yes	No	No
Users and groups	Yes	Yes	Yes	Yes	Yes
Version control	Yes	Yes	Yes	No	Yes

Table 2.2: A review of related systems

2.6 Review of related works

1. (Kahanwal, Dua, & Singh, 2012) **designed a Java File Security System (JFFS) which** is a Java-based file management system with an enhanced security function that uses cryptography as a method of file encryption by offering a transparent UNIX file system interface to directory hierarchies that are automatically encrypted with the user-supplied key. They achieved high security by providing support for the Rijndael Algorithm (AES) and saved keys on portable smart cards for documents that are significant. The power of the JFSS is its ability to encrypt and process a file quickly. Its drawback is the lack of a compression mechanism and it can't operate on all platforms except a Java-enabled device. This can be solved by updating the compression mechanism framework using the Java compression library.
2. **The work "Electronic Document Management System"** aimed at designing an EDMS application tailored to the SCM department that could also be used in the production of the EDMS for future study. Data from the existing SCM workflow was obtained using the interview and discussion method. The interview would aim to better understand the problems facing the department in managing its records. The purpose of the interview was also to find more specific details that would be converted into the form of the requested framework. The main aim of the proposed EDMS was to facilitate the storage and extraction of documents from a database relating to the contract-making process between the organization and its vendors/suppliers. Approximately four processes, the preparation of the tendering plan, the sending of tender invitations, the evaluation of bids and

the recommendation on awards, are put in a simplified form. Much focus has been placed on streamlining procedures that have been introduced by the document management system and on the actual business processes of developing a standard SCM contract that the company undertakes daily.

3. In this particular company, EDMS was the best approach implemented by the SCM department. In addition to enabling the retrieval of the document inside the agency, it could provide a safe place to store documents compared to the conventional filing system. Strong research was carried out while meeting the system requirements to ensure that the structure established was compatible with the business requirements of the said department. The strength of the work is that the proposed system could be improved by a web-based database system that would allow all input screens and system forms to be rendered online but its drawback is that the work limits its discussion only to those important system GUIs focusing on the GUIs through which the instructor interacts with the eCourse File Management System (focused on the GUIs through which the instructor interacts with the eCourse File). Furthermore, the technical details of the database system are also not given.
4. (Anwar & Naseer, 2013) suggested an eCourse file management system that would move from the paper-based compilation files to a more flexible method of compiling and maintaining electronic course files. The work's architecture consists of three layers; a database layer, a device module layer, and a GUI layer. The database layer stores faculty records, courses taught in a semester, and electronic copies of all documents relevant to the course, such as the syllabus, assignments of the student sample. The

Framework Module Layer makes it easier to store and retrieve all data files contained in the database. The GUI allows different users to interact with the system and perform their specific tasks. The system provides many advantages, such as information retrieval, ease of access, storage and disposal problems for paper-based files. The system also provides immense benefits in saving paper and printing costs, reducing the human and financial resources required to compile a course file, mitigating negative environmental effects, saving natural resources for future generations, and contributing to a green sustainable climate. This device can be used to track the progress of courses in the course offerings. The proposed system could be upgraded to a web-based database system that allows all system input screens and forms to be rendered online, but does not provide any encryption and compression techniques, which means that all files are not protected while space management is not at the highest possible level.

5. (Park & Kim, 2010) **in their work “Design and Implementation of E-Document Encryption System using Hash Algorithm”** which was aimed at making some findings on existing research information on the implementation of EDM systems in the construction industry. The algorithm suggested in the study first extends the original image used in the electronic identification card to the hash function using the encrypted key and then rearranges the pixels of each image with the value generated by the scrambling algorithm. The suggested framework extracts the original image using a block rearrange algorithm then re-arranges the unique key with the scrambling process and integrates the distorted image

into the smart identification card chip. Forge or falsification status can be checked by extracting the distorted image from the smart chip of the ID card using an image extractor and comparing it to the original image using a block inverse arrangement algorithm and a unique key and verifying whether it matches the original image.

The hash algorithm has a slower output speed than the AES algorithm, but it was just a part to construct a key, and as the other calculation does XOR, it gets faster execution value when it comes to speed. In the case of encrypting the face field, the speed was improved as unnecessary sections were not encrypted. Also, the speed can be improved in cases of portrait images, such as an ID image, if only sections, including face information, are encrypted, not the entire region, including the background. The strength of their work is that the Encryption method used in the work shows an increase in the pace of the suggested method is improved and better than that of the symmetric key algorithm if only a certain portion, like a person's face, is encrypted. A picture containing a face was the data used in the test, and when about 40% of the face part was used in the encryption, the speed of the suggested method was approximately 40% better than that of the AES algorithm. The downside of their work is that there was a little bit of noise in the loss compression like JPEG. This is because it is saved in the JPEG compression process by using the similarity level of the adjacent colour and further research is required. The encryption methods are often aimed solely at the image file format.

6. (Groenewald, 2004) in the paper “**Symmetric Algorithm Survey: A Comparative Analysis**” proposed an 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 thereby 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.

2.7 Summary of the Literatures Reviewed

The review of research works shows there are still gaps in the existence of a document management system concerning document security, privacy, space management and accessibility. It shows that most encryption, decryption and compression technology used in the reviewed works are targeted at plain text and image files.

Hence the needs for Docaris that will bridge these gaps that exist in document management system because it will be able to work on different document types (e.g. Microsoft Word, Portable Document Format (PDF), spreadsheets, Portable Network Graphics (PNG) etc.) and any device with a browser can access Docaris, which will make its accessibility dynamic.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

This chapter describes the methods, tools, design consideration, data collection, analysis and design architecture used in this research work. It also includes the design of the database and the business logic in the software.

3.2 Software Development Life Cycle

- **The waterfall model:** This takes into account the core process activities of specification, development, validation and evolution and describes them as separate process phases such as requirements specification, software design, implementation, testing, and maintenance.
- **Stages in the model**
 1. **Requirements analysis and definition:** During the requirements analysis phase, the services, constraints and objectives of Docaris are decided by consulting system users through interviews, observations and research. They are then specified in detail and serve as a specification for the system.
 2. **System and software design:** The system design phase allocates specifications to either hardware or software systems by setting up an overall device architecture. Software design requires defining and explaining the abstractions and relationships of the fundamental software system.

3. **Implementation and unit testing:** At this level, the design of the software is carried out as a series of programs or program units. Unit testing requires ensuring that each unit meets its requirements.
4. **Integration and system testing:** individual application units or systems are implemented and evaluated as a complete system to ensure that the specifications of the software are met. The software system is shipped to the customer after review.
5. **Operation and maintenance Normally (though not necessarily),** this is the longest life cycle phase. The system is built and put into operation. Maintenance involves fixing errors that have not been detected in earlier phases of the life cycle, improving the implementation of system units and enhancing the services of the system as new requirements are identified.

3.2 System Analysis

System analysis is the method of collecting and analyzing facts, diagnosing issues and using the knowledge to make the recommended system changes.

- **Problems of the existing system**

The conventional method of storing documents which are using file storage cabinets is still widely used in businesses and by individuals most especially in this region of the World (Nigeria/Africa). This method poses a lot of challenges as discussed in Chapter 1 of this research work, some of the problems are:

- Unable to locate the right document.
- Working on the wrong version of a file.
- Forced to manually merge documents.

- Low levels of Productivity.
- Employee Frustration.
- Multiple copies of a document.
- No Disaster Recovery/Business Recovery.
- Less secured documents.

• **Capabilities of the new system**

The proposed system is a lightweight document management system with the following features:

- Web-based.
- Responsive graphical user interface.
- Supports PDF, DOCX, PPTX, ODT, image and video files.
- File versioning (Version Control).
- Optical Character Recognition (OCR).
- Powerful search engine with recommendations and text highlights.
- Custom User-defined metadata.
- 256-bit AES encryption of stored data and files.
- Tag system.
- Comments.
- User/group permissions.
- Document sharing by URL.
- Audit Log.
- Storage quota for each user.
- RESTful Web APIs.

- Hierarchical groups.
- 2-factor authentication.
- Workflow system.

3.3 System Design

System design is the process of defining system architecture, modules, interfaces and data to meet the defined requirements. System design is also the application of system theory to product development (Wikipedia, Systems design, 2020).

- **Process Modelling**

Software Process Modeling is a coherent collection of activities for the description, design, implementation and testing of software systems. The software process model is an abstract representation of a process that provides a process definition from a specific perspective.

- **Data Flow Diagram (DFD)**

Data flow diagrams are used to graphically represent data flow in a business information system. DFD defines the processes involved in the transfer of data from input to file storage and the generation of reports on a device.

- **DFD Level 0 (Context Level)**

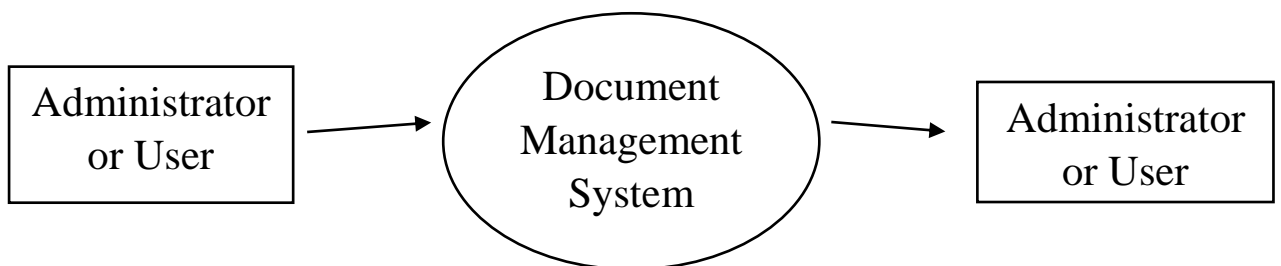


Fig. 3.1 Docaris' DFD Level 0

- **DFD Level 1 (Administrator)**

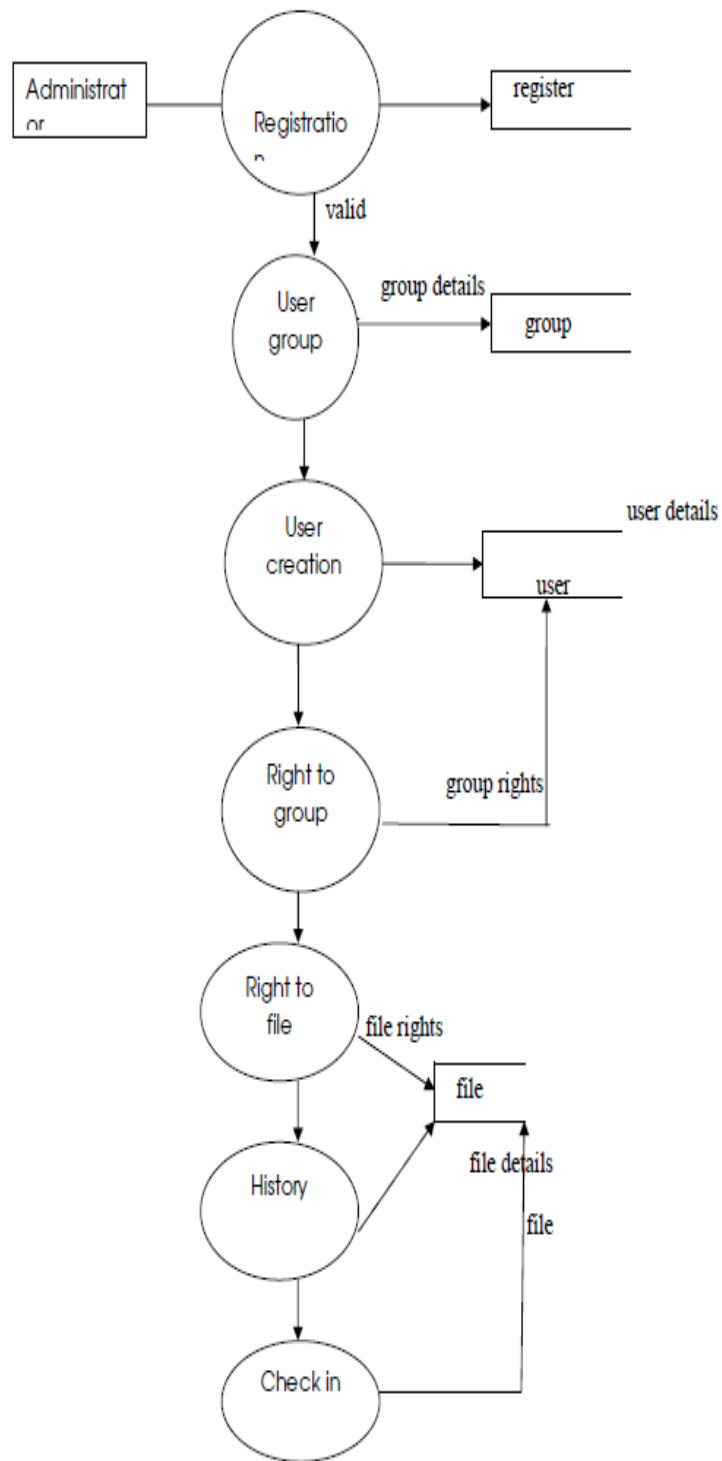


Fig. 3.2a Docaris' DFD Level 1 (Administrator)

- **DFD Level 1 (User)**

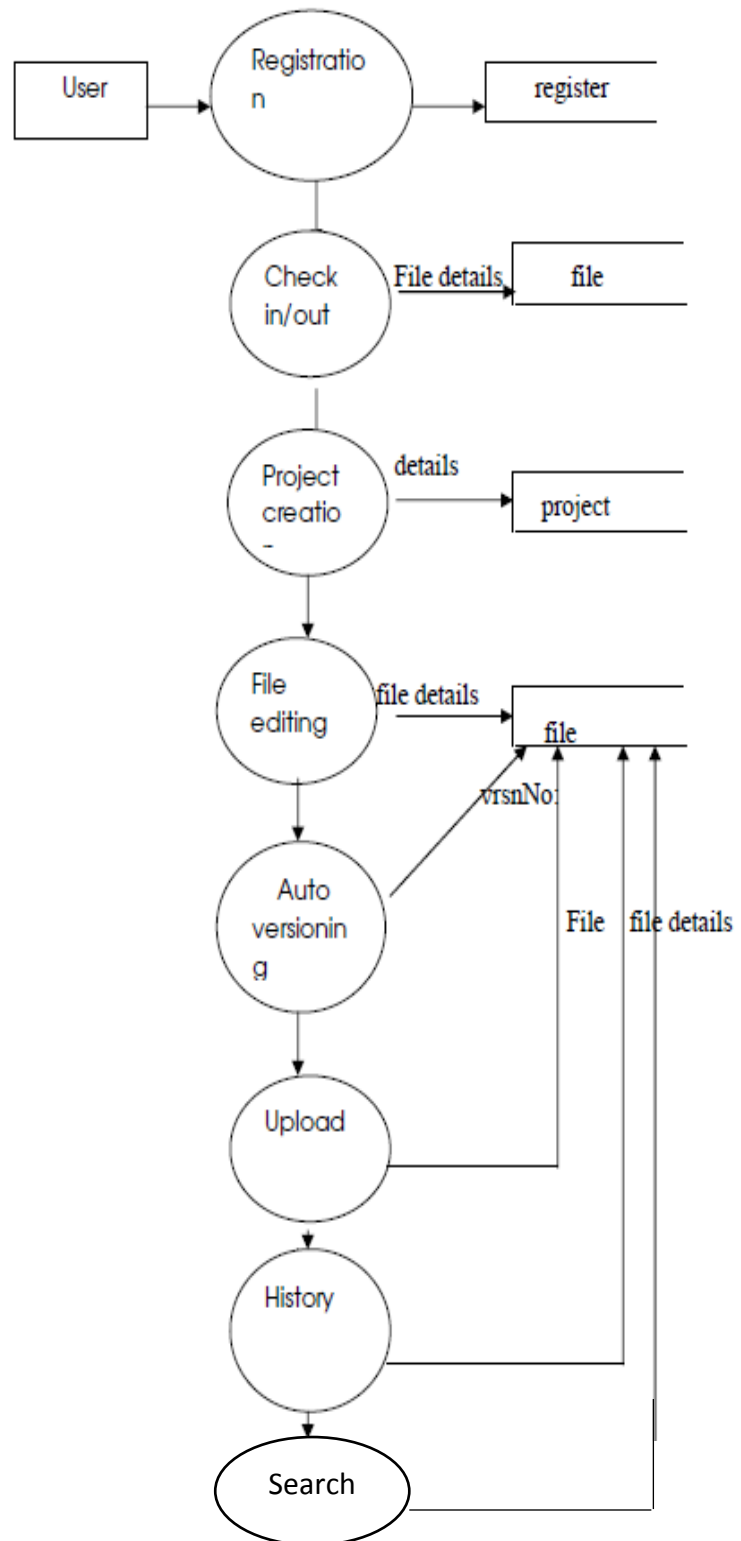


Fig. 3.2b Docaris' DFD Level 1 (User)

- **Use Case Diagram**

A use case explains how a user uses a system to achieve a specific purpose. The use case diagram consists of the system, the associated use cases and the actors and links them to each other for visualization.

Figure 3.3 shows Docaris' (Proposed System) use case diagram, a regular user can register/login to the system, search documents by metadata, tags, title, and other fields, view document versions, browse documents, upload documents, add files to documents, add tags to documents, include additional metadata to documents, add comments on documents, delete comments on documents, add a document workflow browse through documents, and share documents while an administrator can perform all of the user functions and also manage users: add users, edit users, disable users, delete users; manage user groups: add new groups, edit existing groups, add users to a group; edit system's configuration, customize system theme (UI theme), manage server's log.

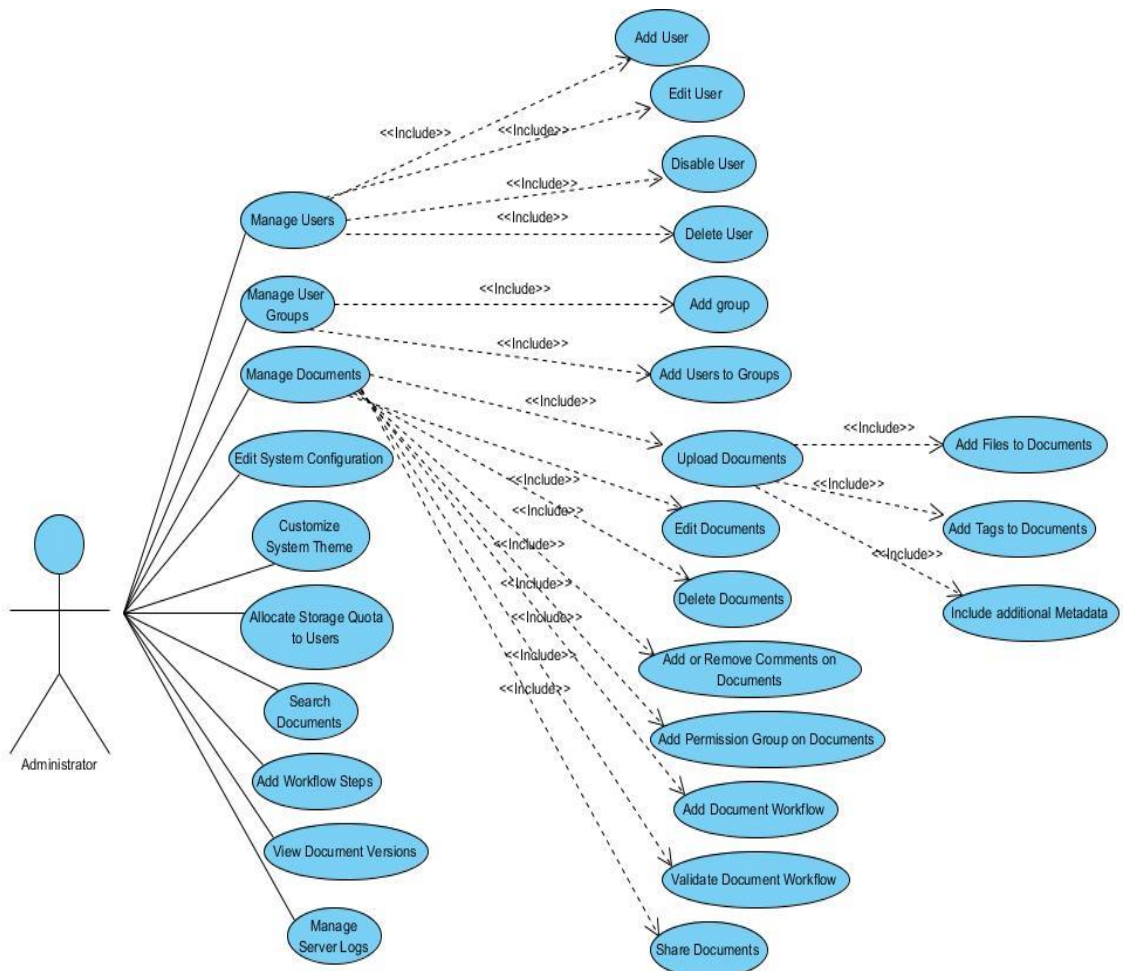
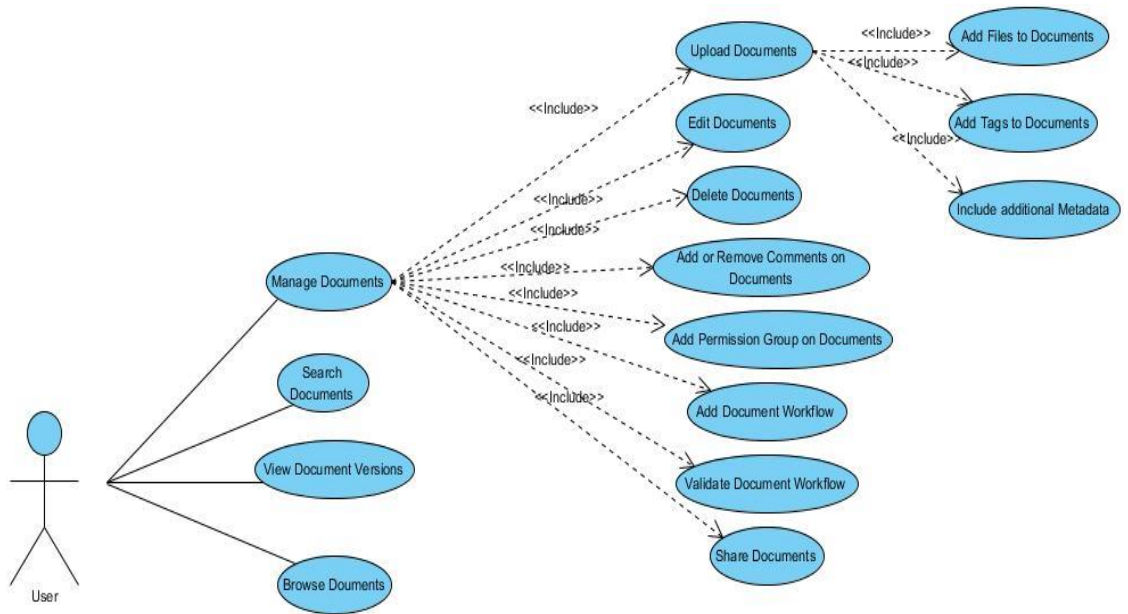


Fig. 3.3 Docaris' Use case diagram

- **Data Modelling**

Data modelling is a method used to identify and evaluate the data specifications required to support business processes within the framework of applicable information systems in organizations. Therefore, the data modelling process includes trained data modellers working closely with business partners as well as potential users of the information system (Wikipedia, Data Modelling, 2020).

- **Entity-Relationship Diagram (ERD)**

An entity-relationship diagram (ERD) is a pictorial representation of entities and their relationships with each other, usually used to model the structure of data within databases or information systems. This "graphical representation" serves two purposes. It helps database professionals to explain the overall design succinctly and accurately. The ER diagram can be quickly converted into a relational diagram. There are three components to the ERD: persons, attributes and relationships.

- **Conceptual ERD / Conceptual Data Model**

The Conceptual Model describes the Entities and their Relationships.

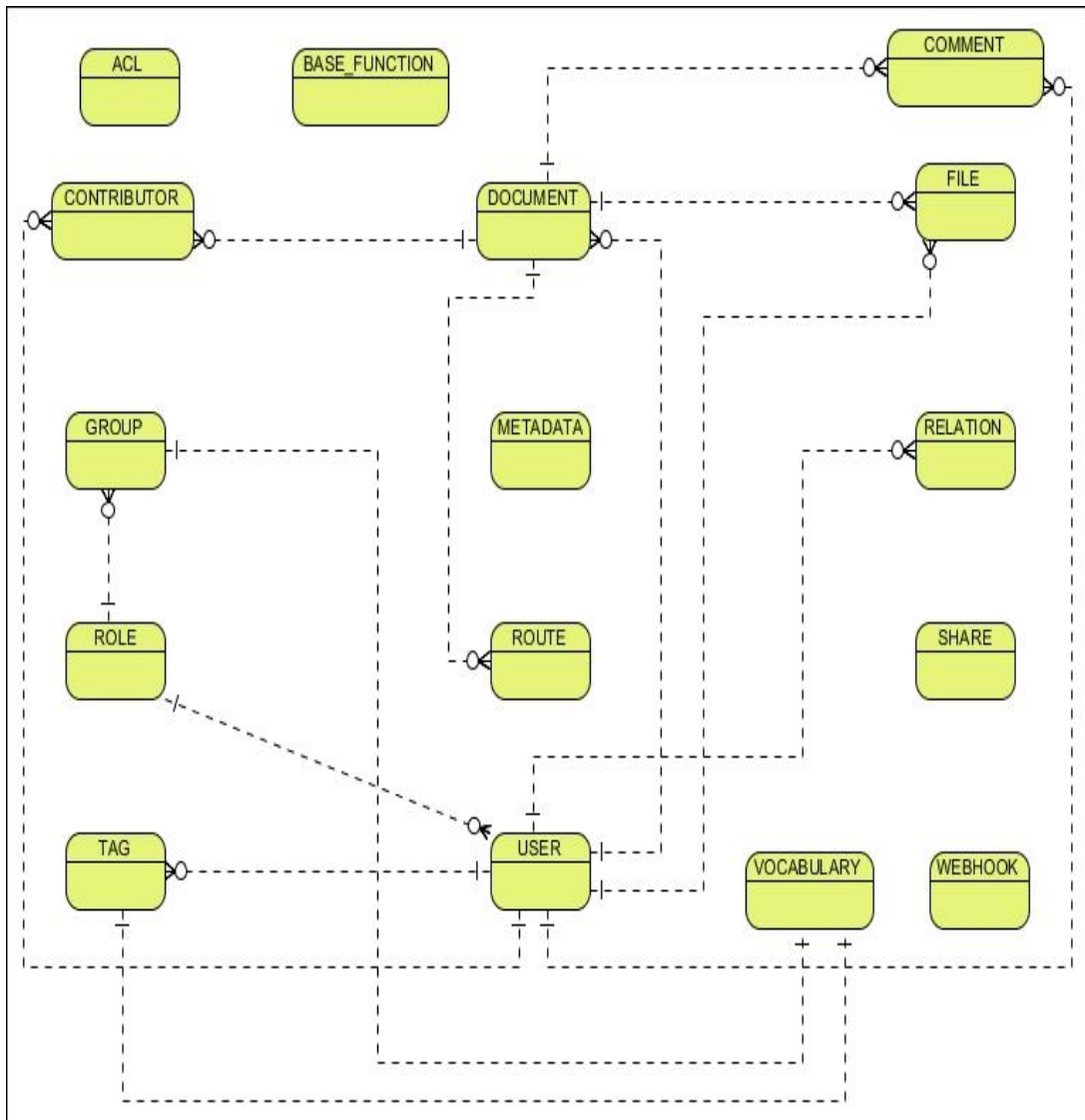


Fig. 3.4 Docaris' Conceptual Data Model

- **Logical ERD / Logical Data Model**

The Logical ERD defines the structure of the data elements and their relationships.

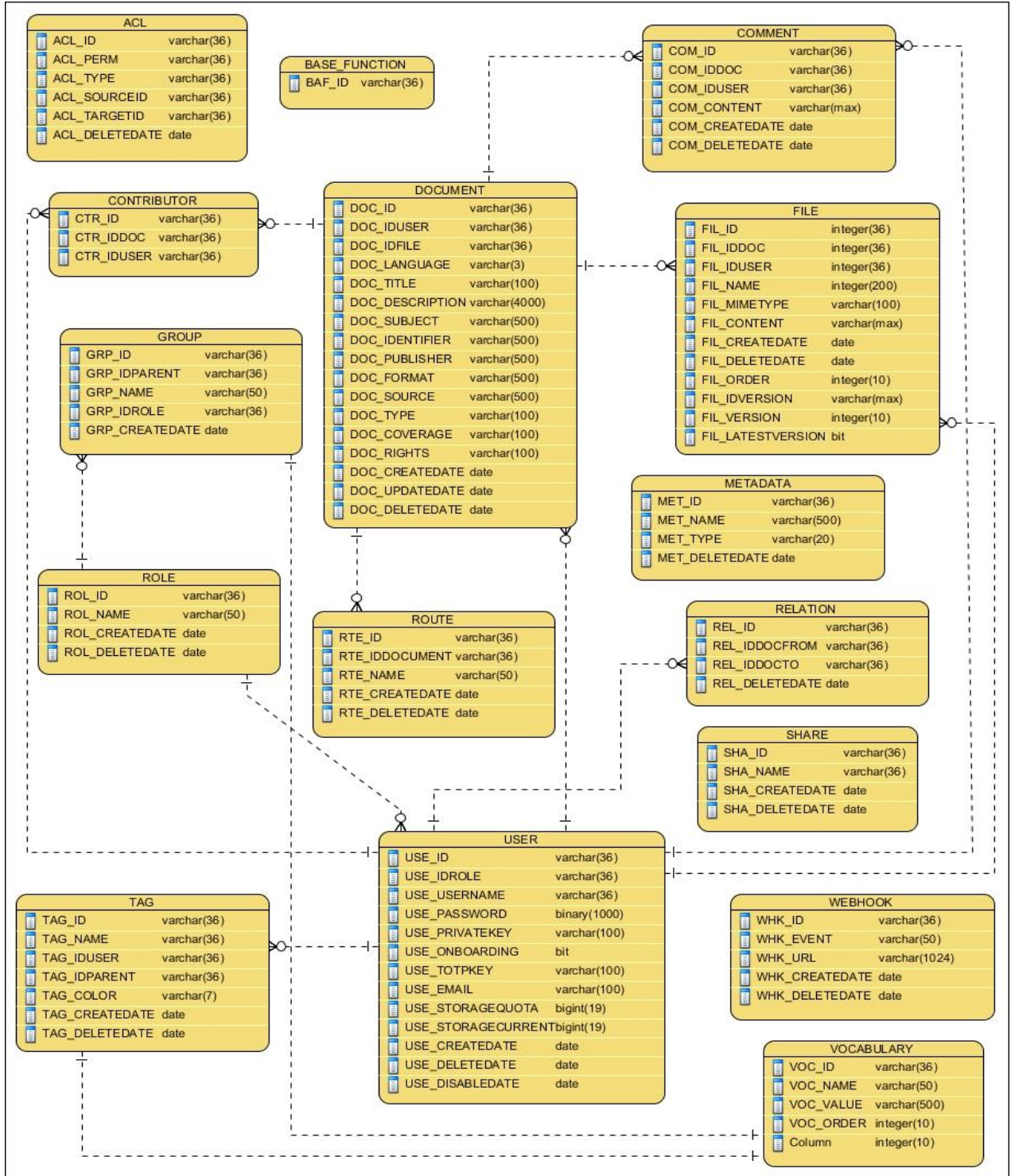


Fig. 3.5 Docaris' Logical Data Model

- Physical ERD / Physical Data Model

The physical ERD defines the database-specific implementation of the data model.

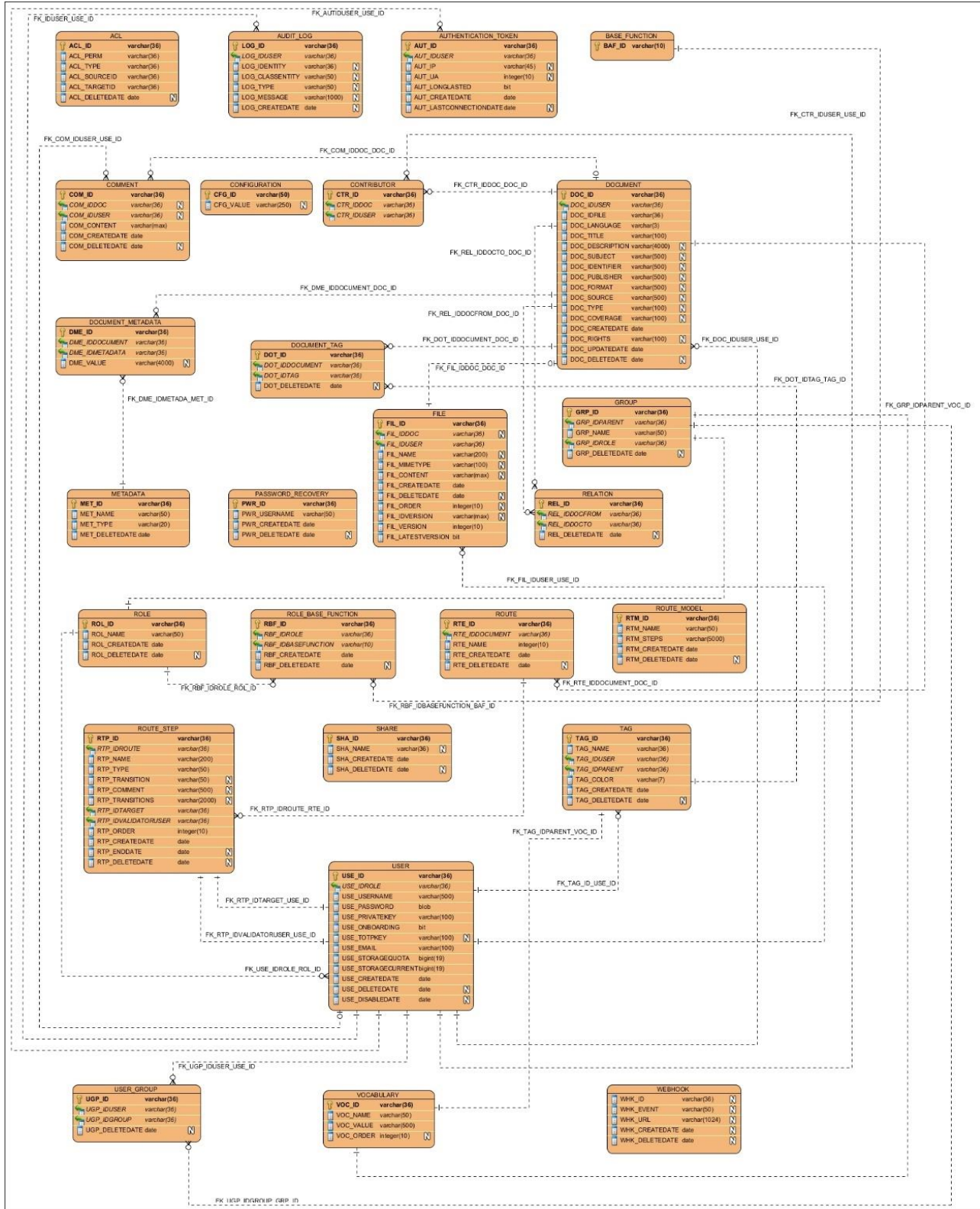


Fig. 3.6 Docaris' Physical Data Model

- **Class Model**

A Class diagram is a type of static structure diagram that explains the structure of the system by showing the classes, attributes, operations (or methods) of the system, and the relationship between objects.

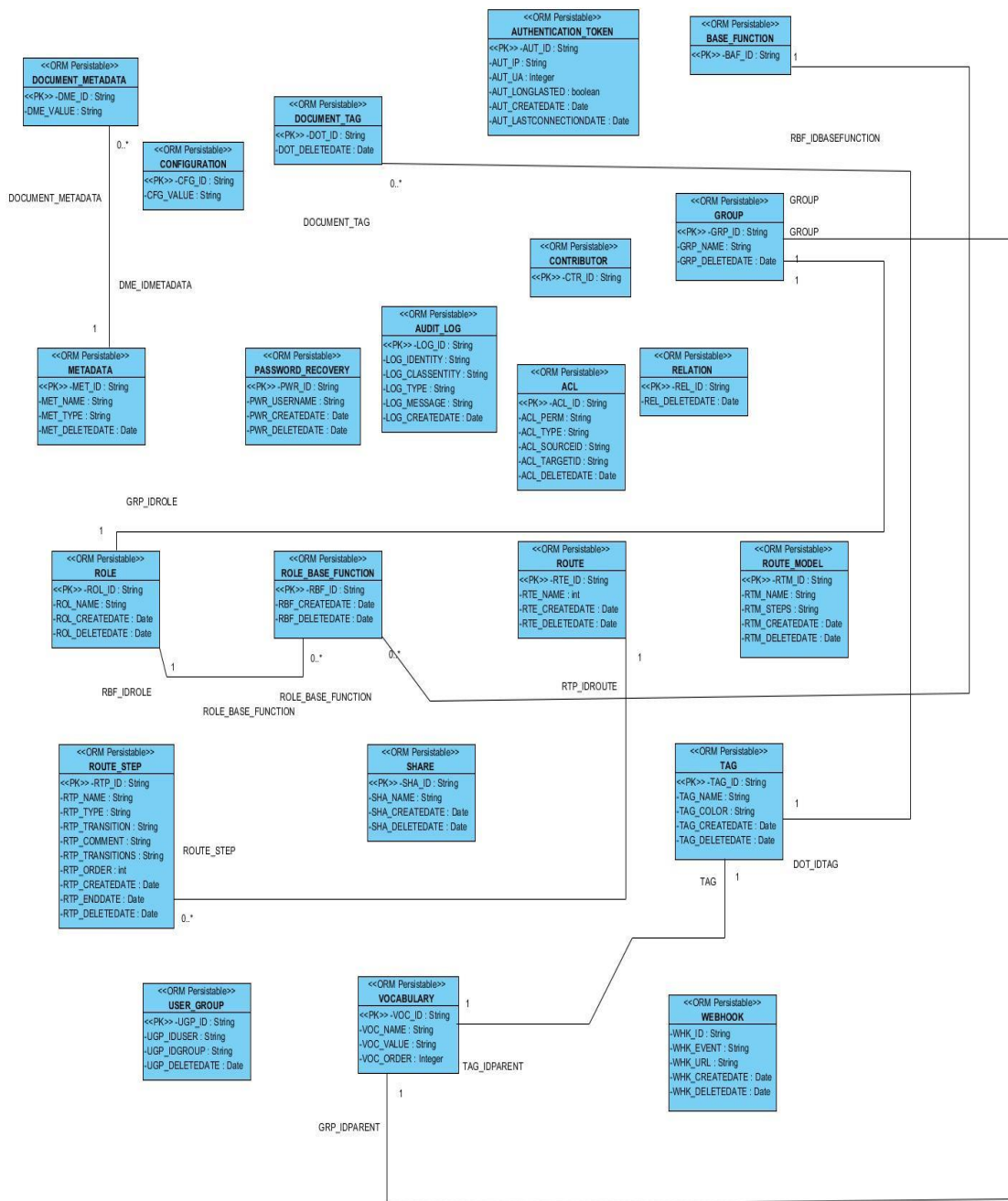


Fig. 3.7 Docaris' Class Diagram

- **Activity Diagrams**

An activity diagram visually represents a sequence of acts or control flow in a system similar to a flowchart or data flow diagram. Activity diagrams are also used in the simulation of business processes.

- **User / Administrator Login Activity**

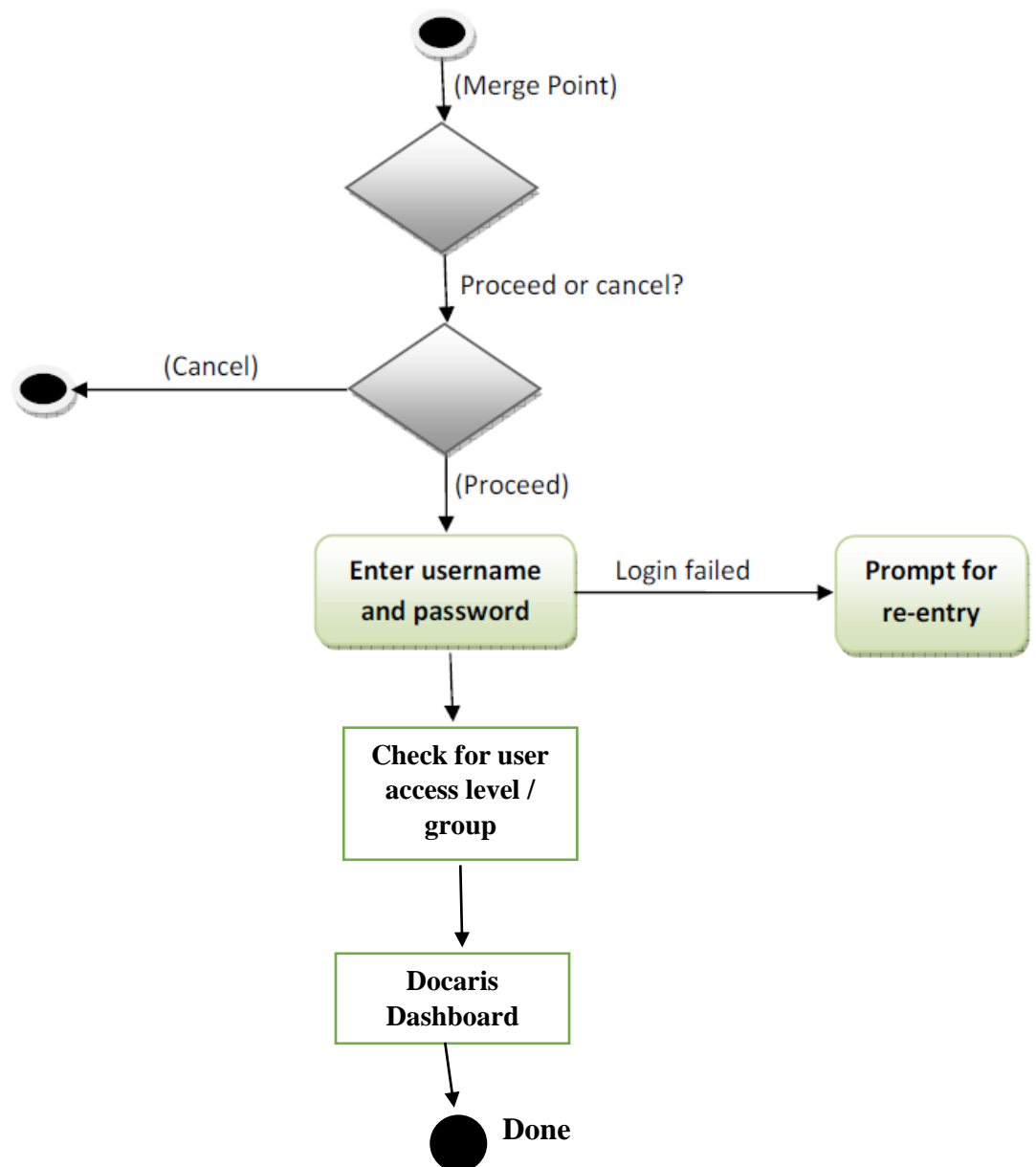


Fig. 3.8 User / Administrator Login Activity Diagram

- **Document Search Activity**

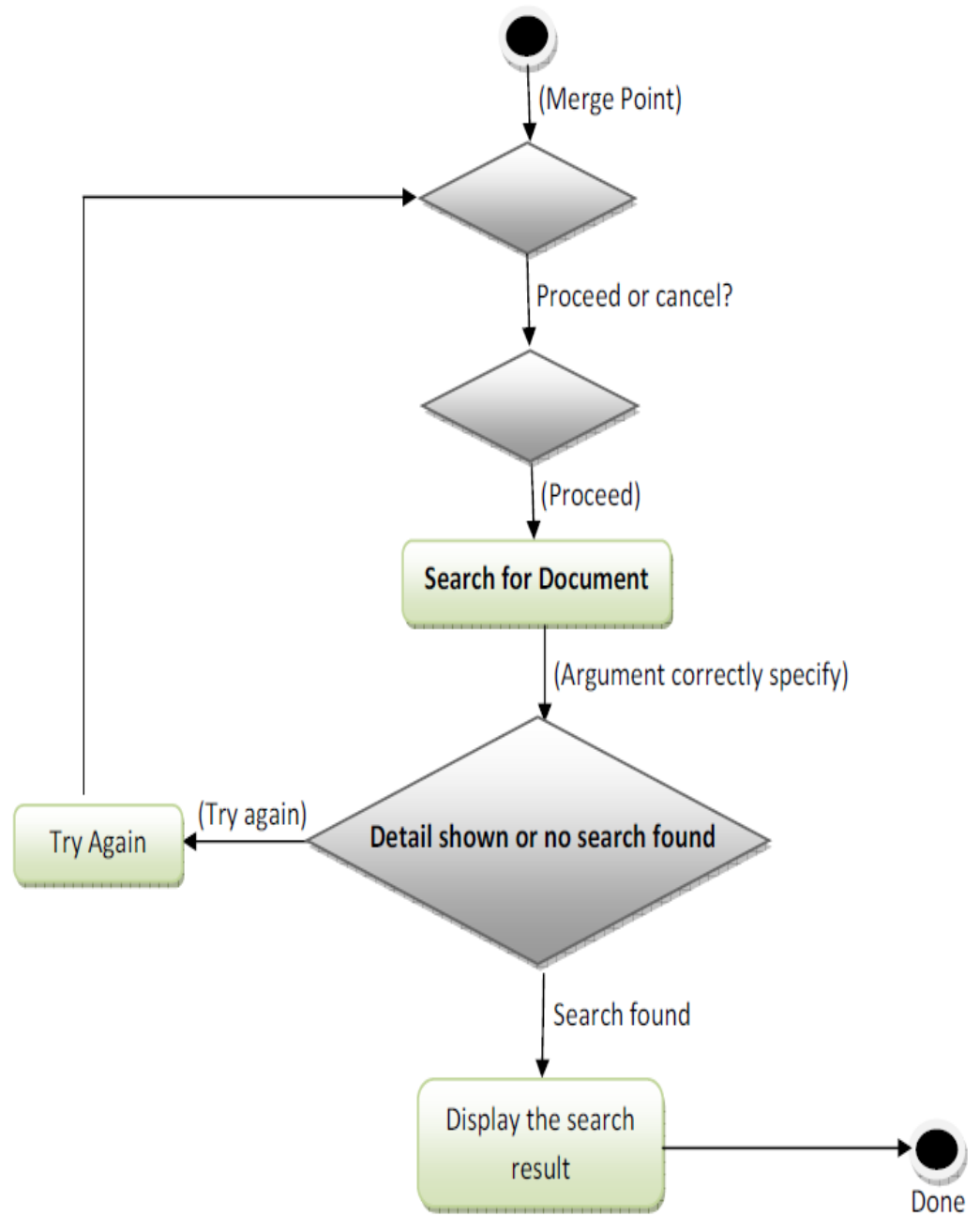


Fig. 3.9 Document Search Activity Diagram

- **Document Upload Activity**

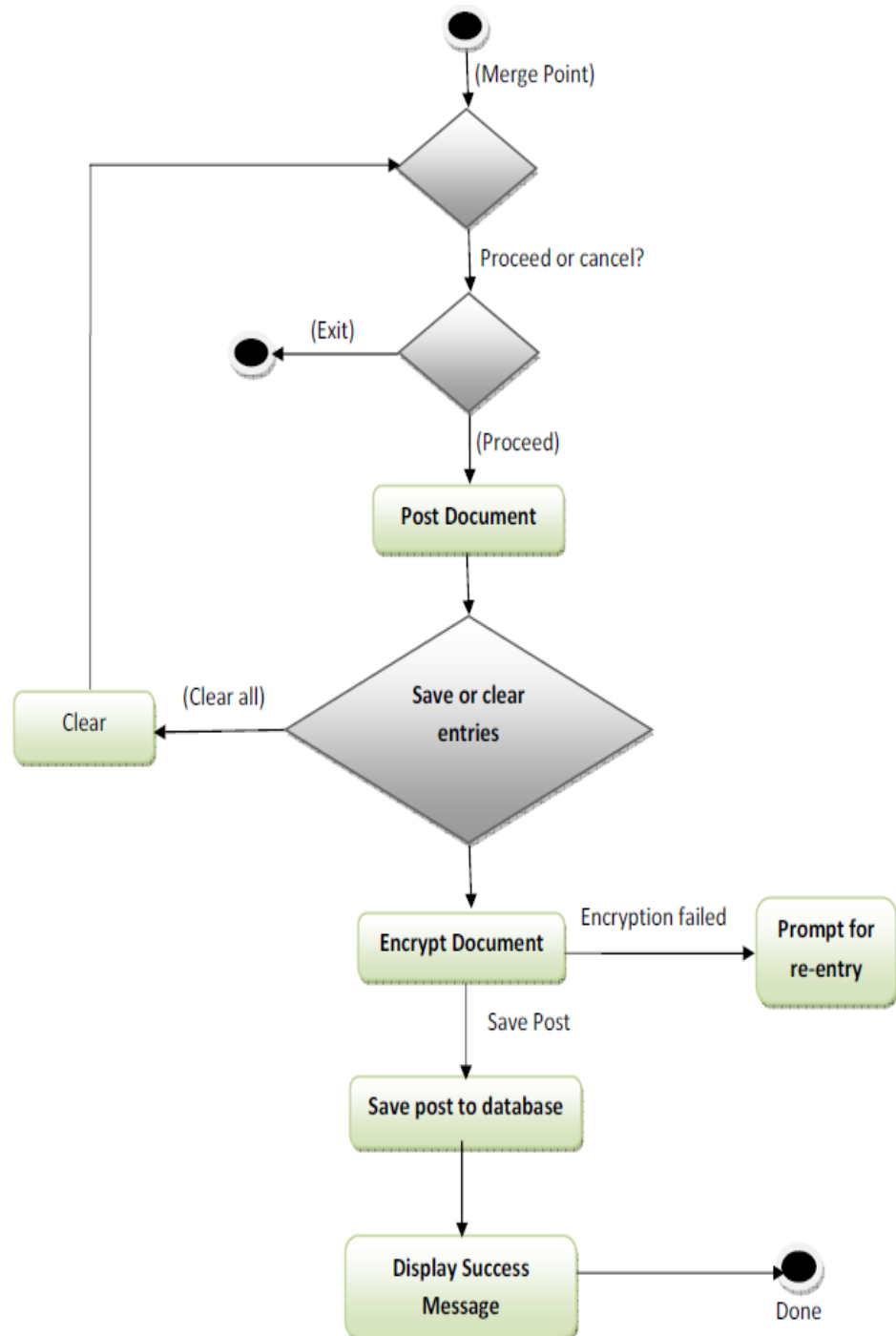


Fig. 3.10 Document Upload Activity Diagram

- **Document Download Activity**

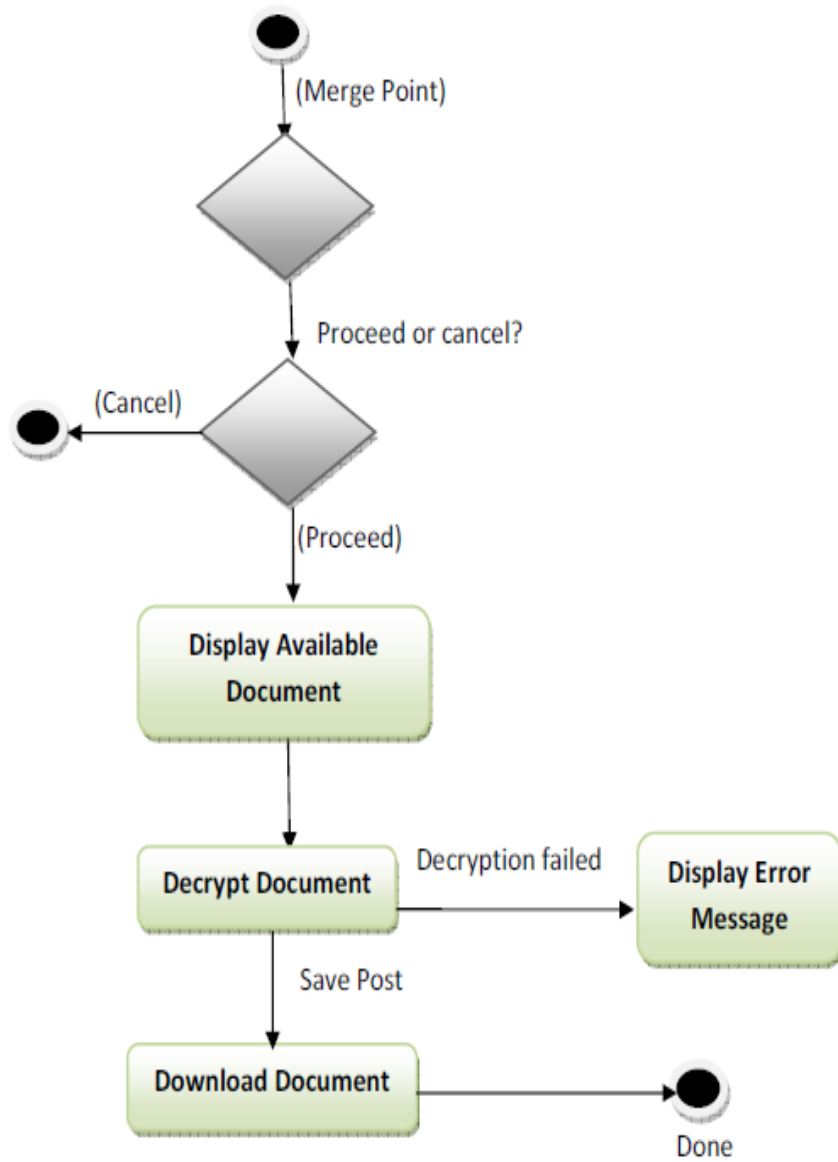


Fig. 3.11 Document Download Activity Diagram

- **Document Edit Activity**

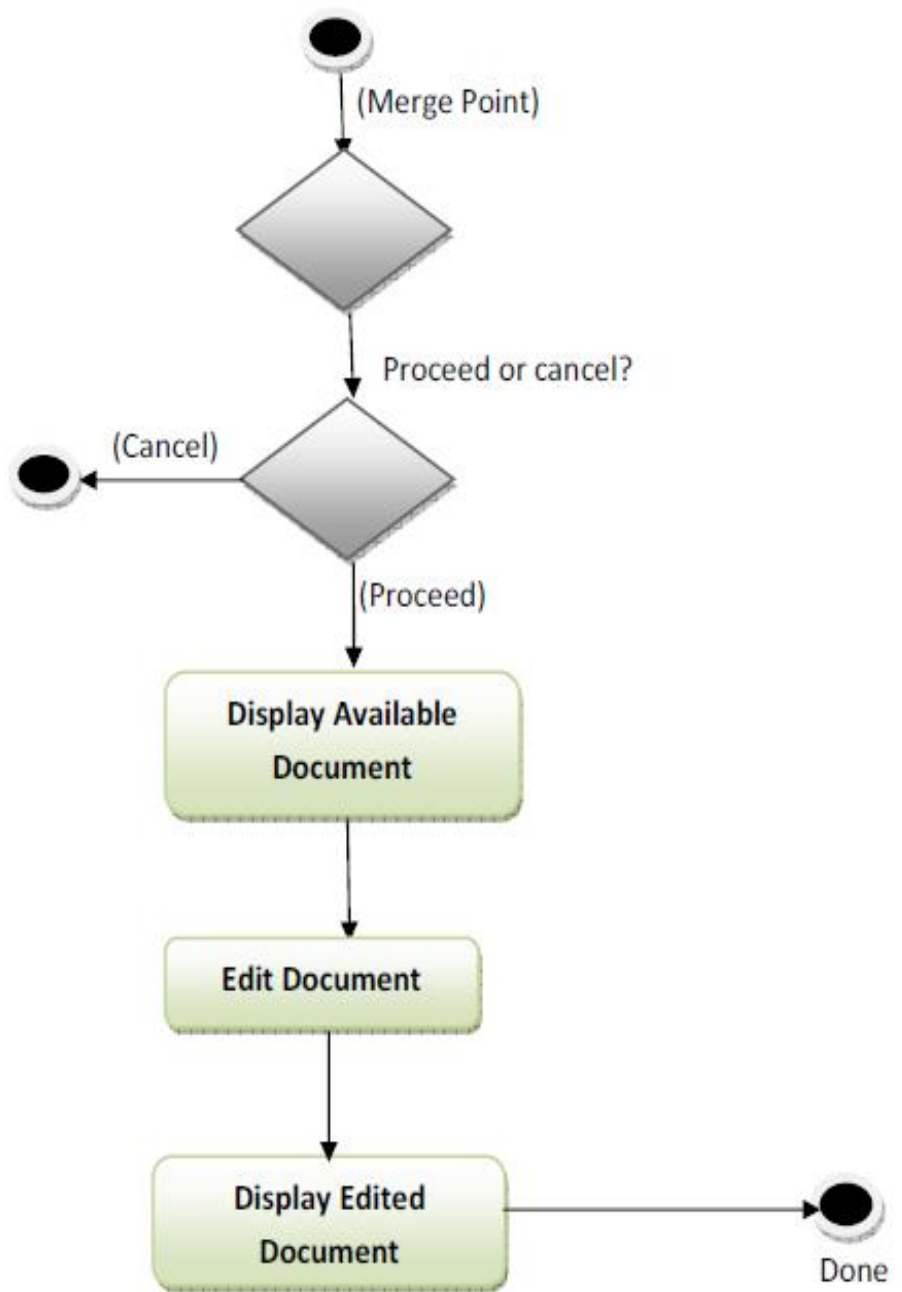


Fig. 3.12 Document Edit Activity Diagram

- **Document Delete Activity**

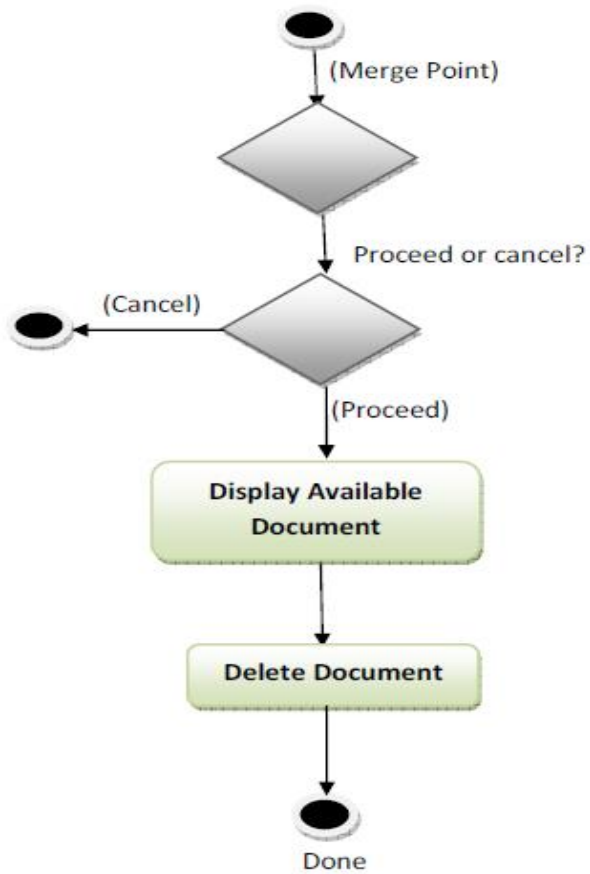


Fig. 3.13 Document Delete Activity Diagram

- **Document Share Activity**

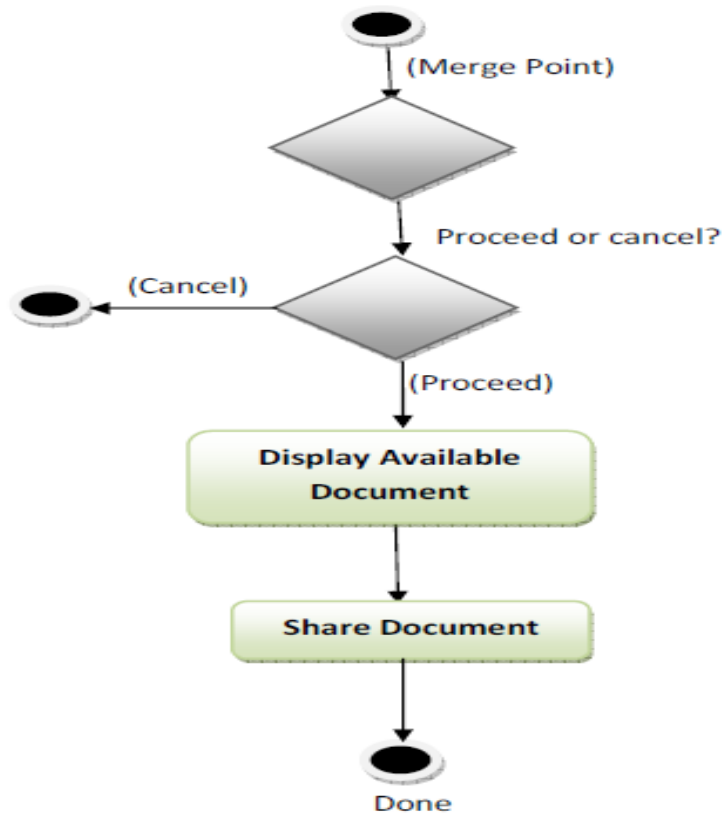


Fig. 3.14 Document Share Activity

3.4 Languages and Tools Used

The languages and tools used in completing this project are:

- **Hypertext Markup Language (HTML)**

HTML stands for Hypertext Markup Language. Html defines the structure of a web page and the content flows in such a page.

- **Cascading Style Sheets (CSS)**

Cascading Style Sheets is a style sheet language used for beautifying and defining the presentation of contents written in a markup language like HTML.

- **Bootstrap**

Bootstrap is a CSS framework which is free and open-source. It was built to be a responsive, mobile-first web development framework.

- **JavaScript (JS)**

JavaScript, often abbreviated as JS, is dynamically typed programming language that conforms to the ECMAScript specification. It is high-level, often just-in-time compiled, and multi-paradigm.

- **AngularJS**

AngularJS is a structural framework for dynamic web applications. It lets you use HTML as your template language and lets you extend the HTML's syntax to articulate the components of your application clearly and succinctly.

- **Java**

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

- **Hibernate Framework**

Hibernate ORM is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database

- **H2 Database**

H2 is a relational database management system written in Java. It can be embedded in Java applications or run in client-server mode.

- **Jetty Web Server**

Eclipse Jetty is a Java web server and Java Servlet container. Web servers are typically meant for serving documents or resources to users but Jetty is now often used for machine to machine communications especially within a large software framework.

- **Postgresql**

PostgreSQL or Postgres, is a free and open-source relational database management system highlighting extensibility and SQL compliance.

CHAPTER FOUR

RESULT AND DISCUSSION

4.0 Introduction

This section addresses the implementation of this project, it also discusses the design and analysis of the system, including screenshots from the application and the interfaces used in the creation of the application as well as the methods used in the development of the system.

4.1 System Hardware Requirements

The following hardware resources are needed to run (host) the application:

- i. Intel Core i3 or higher (2.6 GHz processor's speed).
- ii. 256GB or more, hard disk space requirement.
- iii. 4GB RAM or more.
- iv. A high-speed intranet connection.
- v. An LCD Monitor.
- vi. Mouse.
- vii. Keyboard.

4.2 System Software Requirements

The following prerequisites are needed to be installed on the system

- i. Operating system with support for GUI
- ii. Java 8 with the Java Cryptography Extension
- iii. Tesseract 3 or 4 for OCR
- iv. Maven 3
- v. NPM
- vi. Grunt
- vii. ffmpeg for video thumbnails

- viii. mediainfo for video metadata extraction
- ix. A web app server like Jetty or Tomcat

4.3 Implementation Procedure

Docaris is organized into three maven modules namely:

- docaris-core
- docaris-web
- docaris-web-common

To run Docaris, you need to follow these steps:

i. Launch the build

From the root directory: *mvn clean -DskipTests install*

ii. Run a stand-alone version

From the docaris-web directory: *mvn jetty:run*

iii. Build a .war to deploy to your servlet container

From the docaris-web directory: *mvn -Pprod -DskipTests clean install*

You will get your deployable WAR in the docs-web/target directory.

4.4 Developed System Images

a. Login Page

The login page is a fully responsive landing page that adjusts according to different screen sizes where the users input their username and password, and the system validates the data entered is correct from the database as shown in fig. 4.1a.

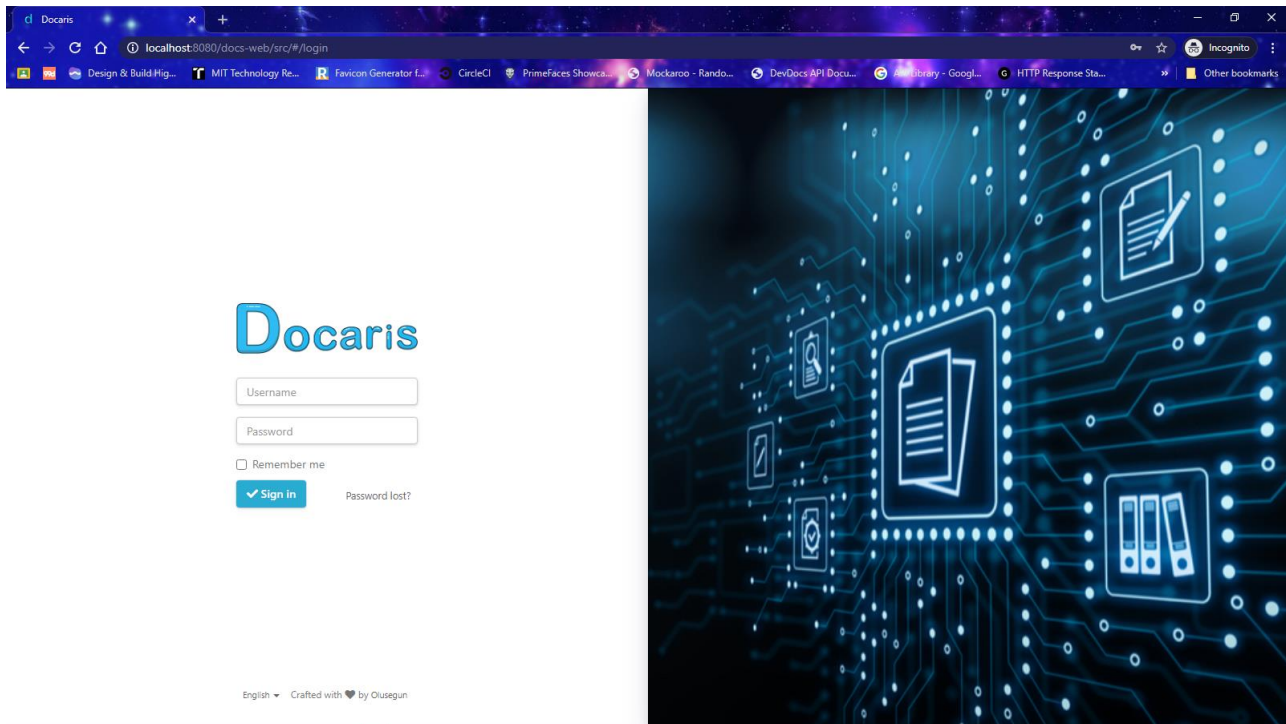


Fig. 4.1a Docaris' Login page (Desktop view)

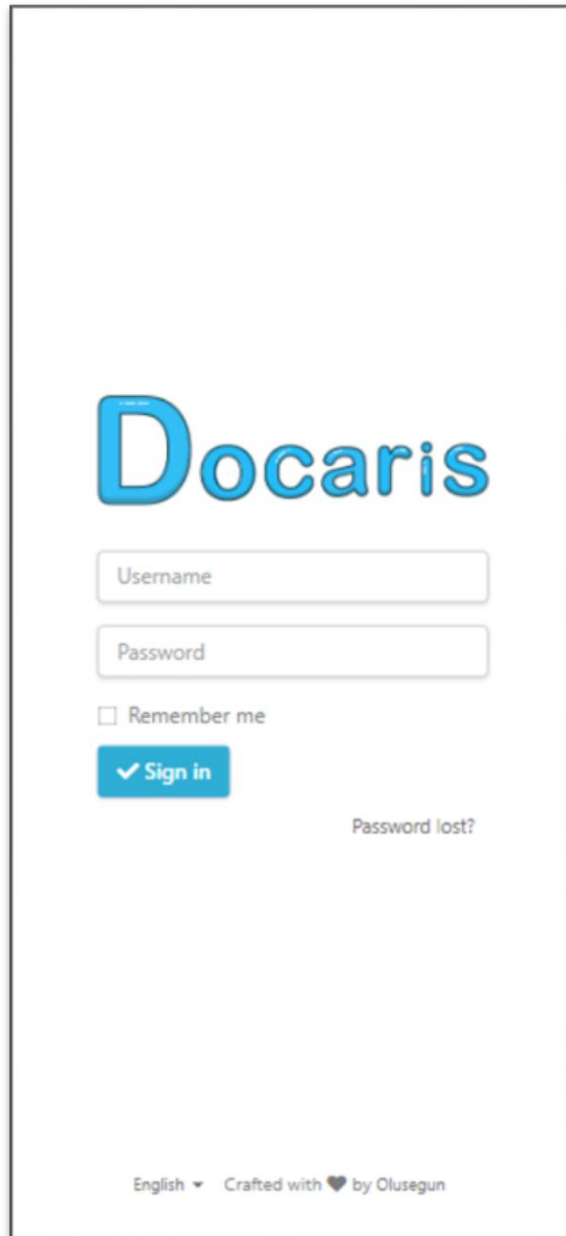


Fig. 4.1b Docaris' Login page (Mobile view)

b. Dashboard page

The dashboard page is rendered when correct login credentials are provided to the system (Username and Password). The dashboard shows the user a list of documents which the user has permission to browse through.

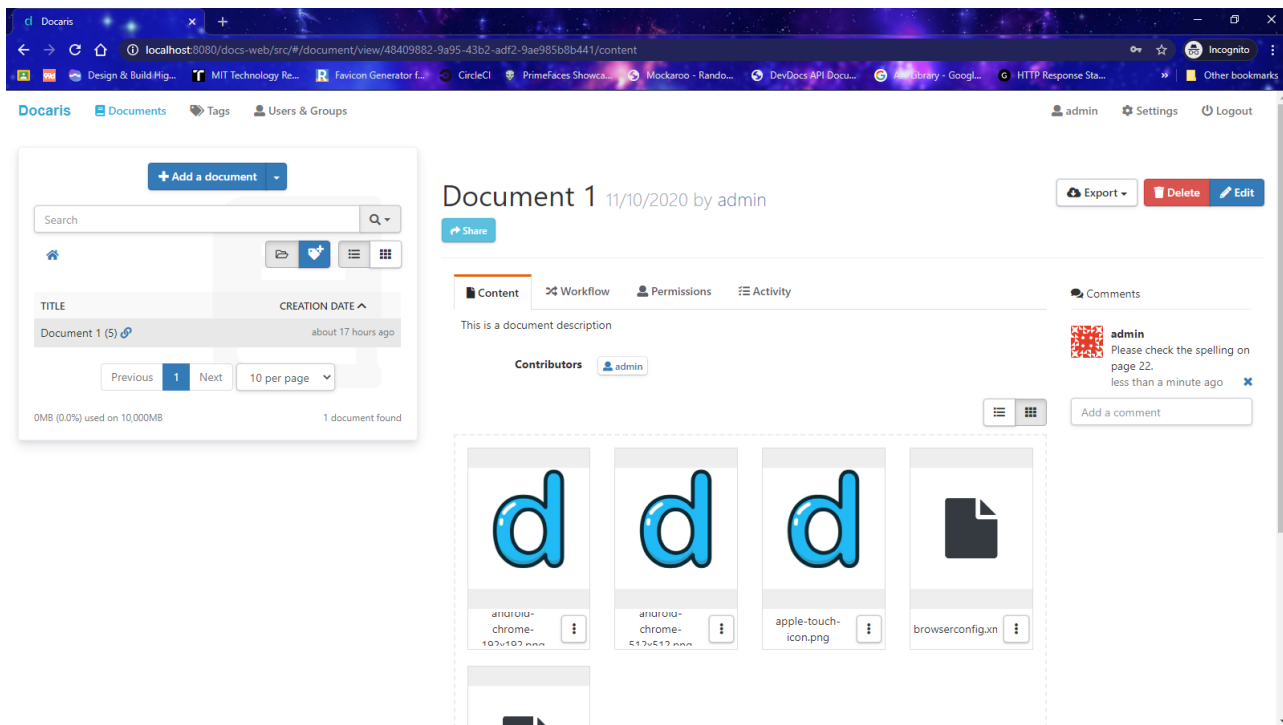


Fig.4.2a. Docaris' Dashboard page (Desktop View)

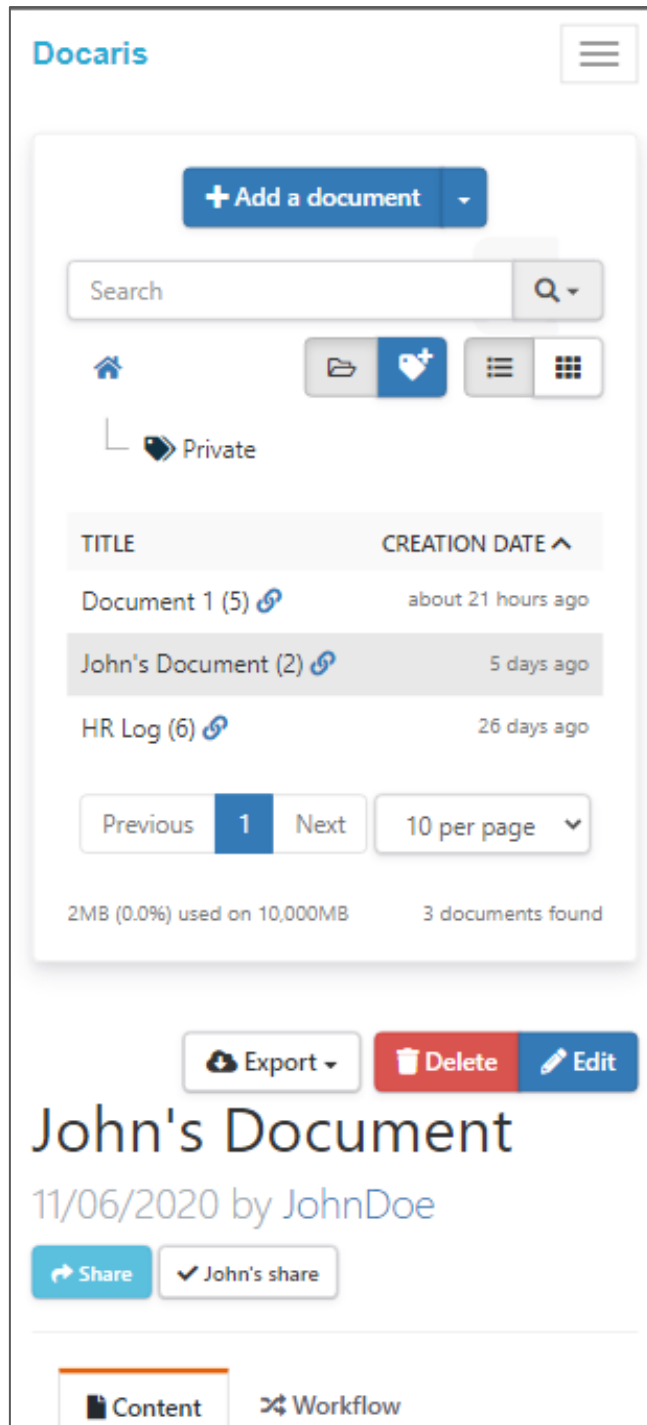


Fig.4.2b. Docaris' Dashboard page (Mobile View)

c. Add Document

Users can add documents, with title, description, tags, files and custom metadata.

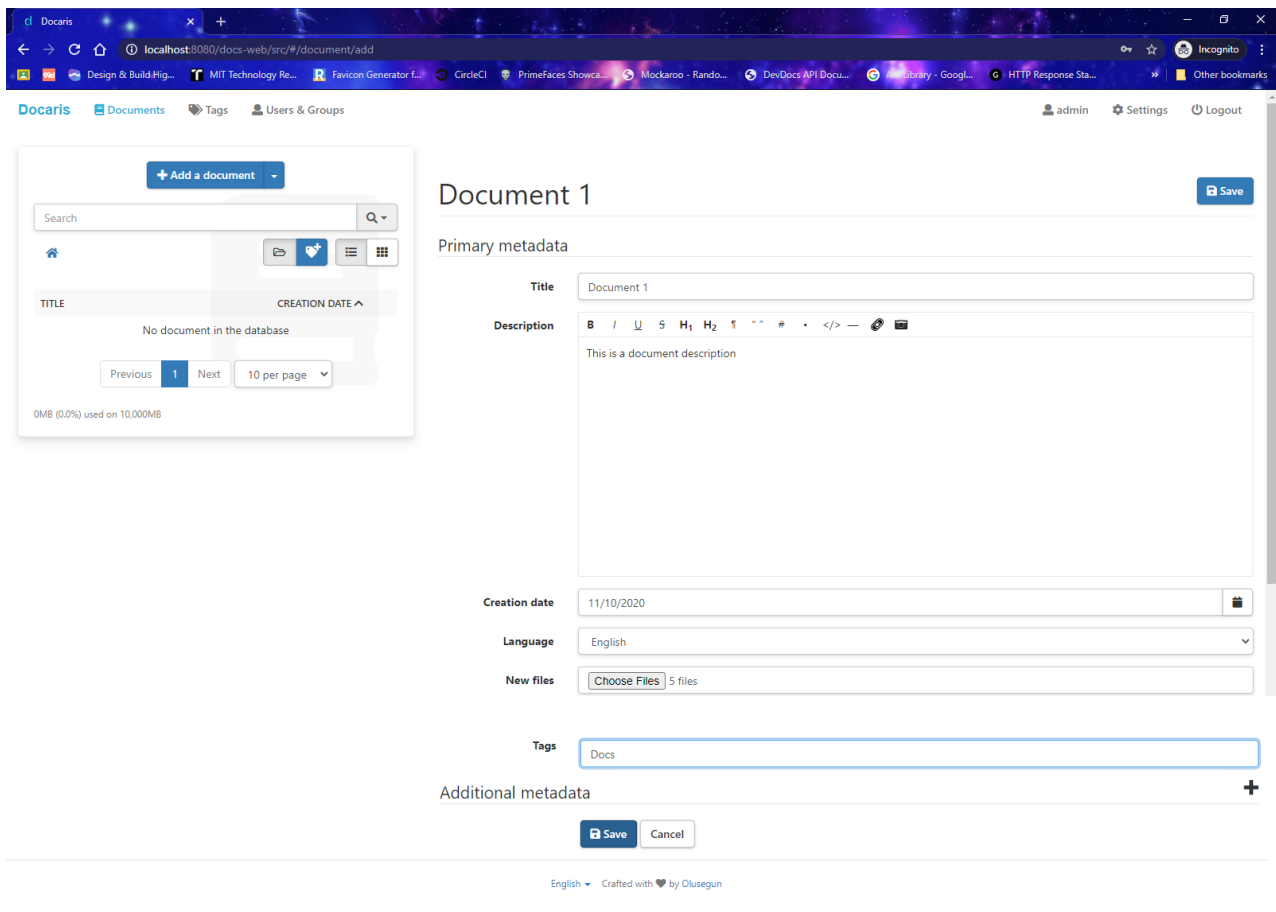


Fig. 4.3 Add Document View

d. View Documents

Users can select documents to browse through if they have permission to read or write the document.

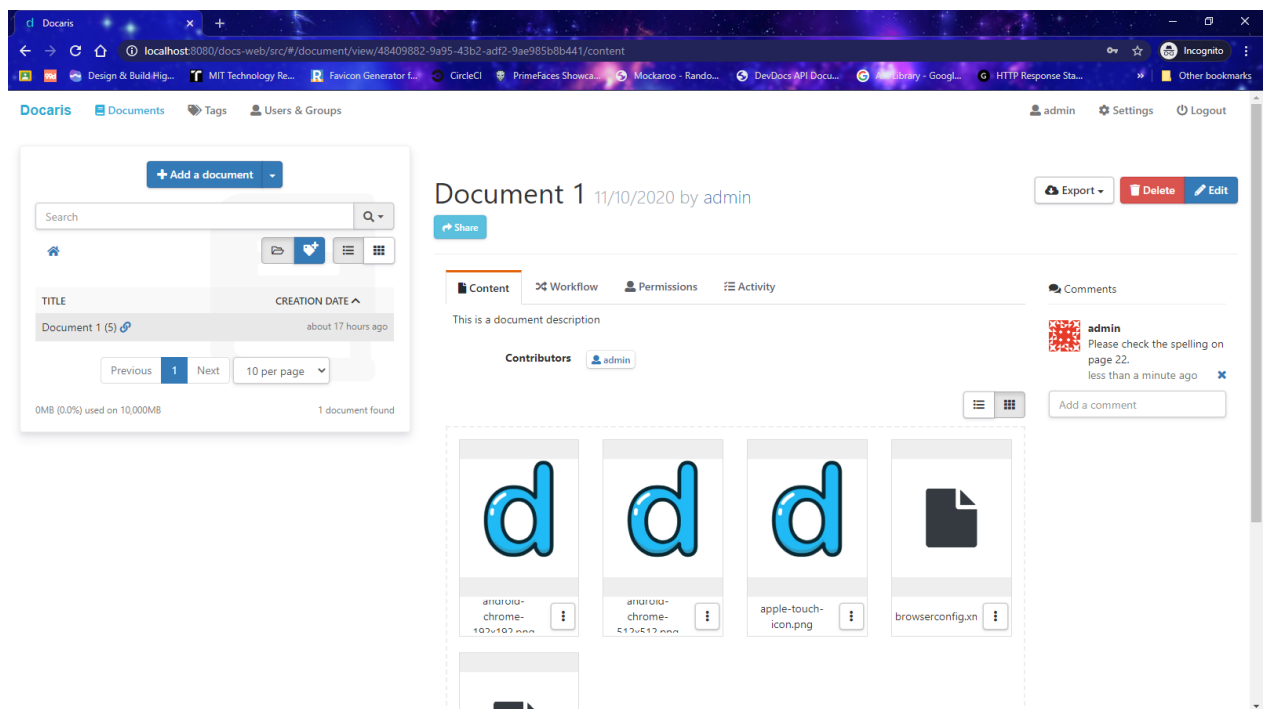


Fig. 4.4 View Documents

e. Document Workflow

Document workflows are processes created for certain user groups by the administrator to validate a document.

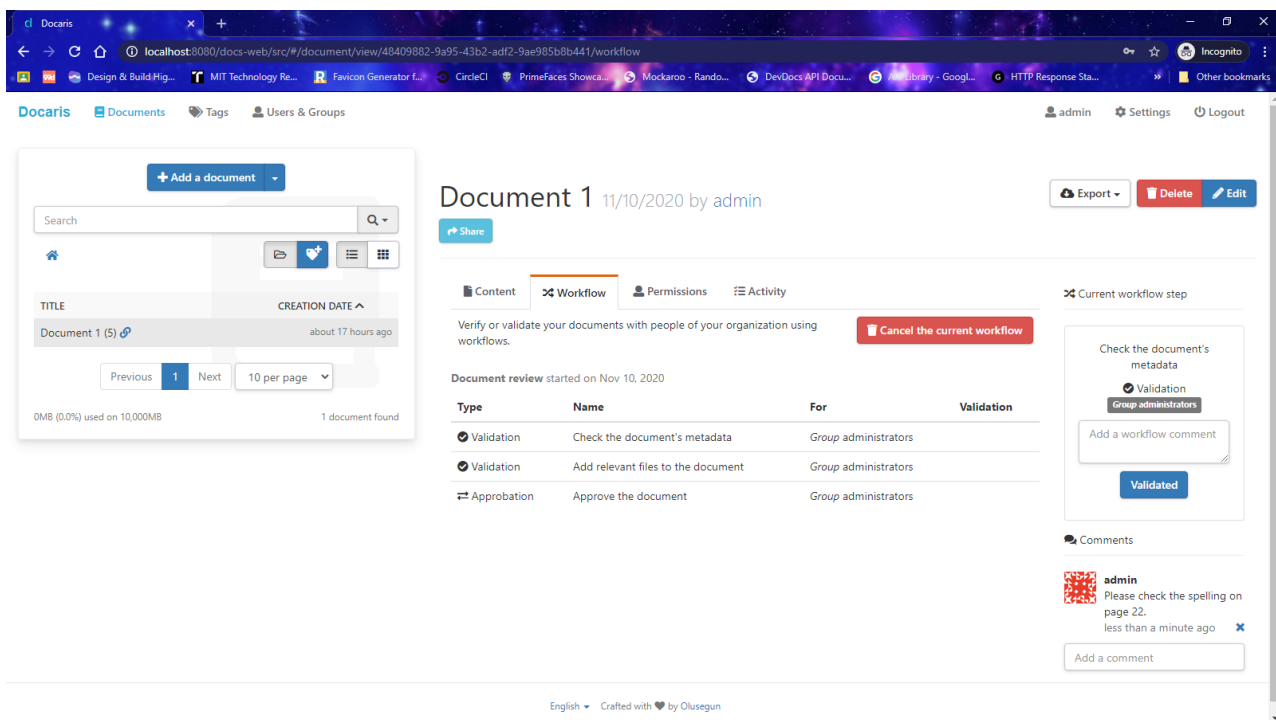


Fig. 4.5 Document workflow view

f. Document Permission

Permissions “can read” and “can write” can be set for individual users or a group of users by the author of a document.

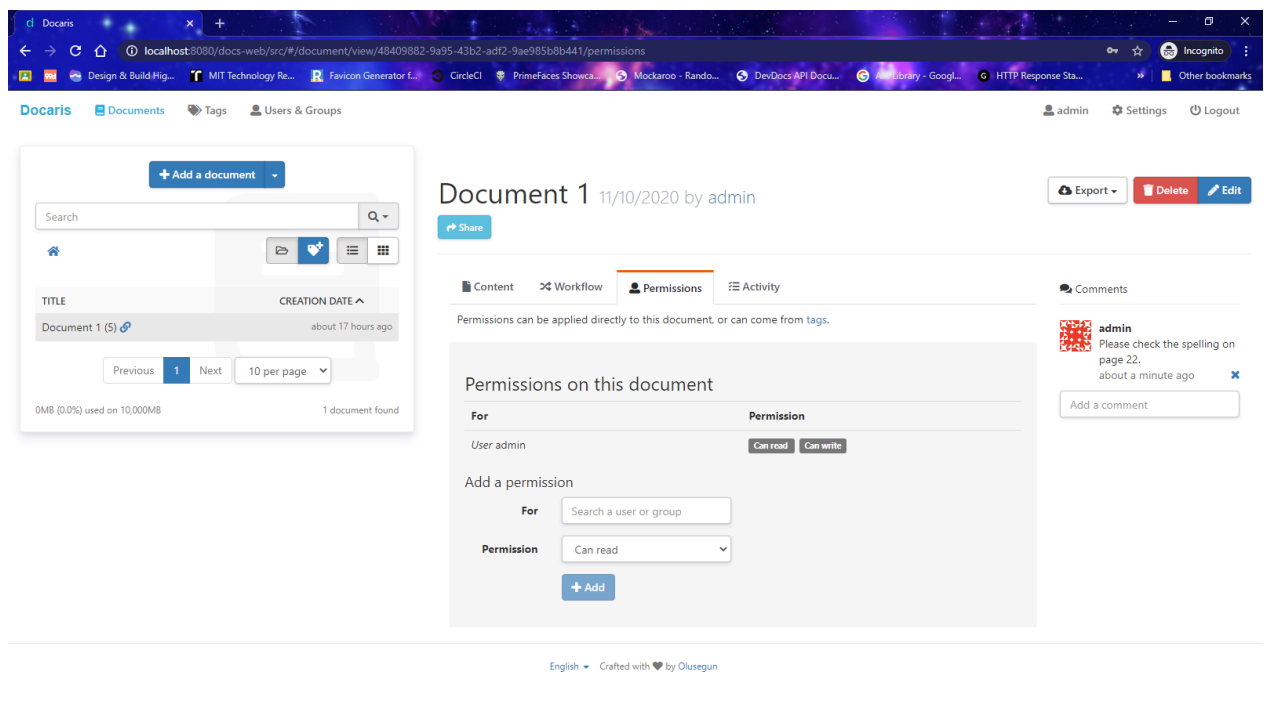


Fig. 4.6 Document Permission View

g. Document Activity

The author of a document can view the activities carried out on individual documents.

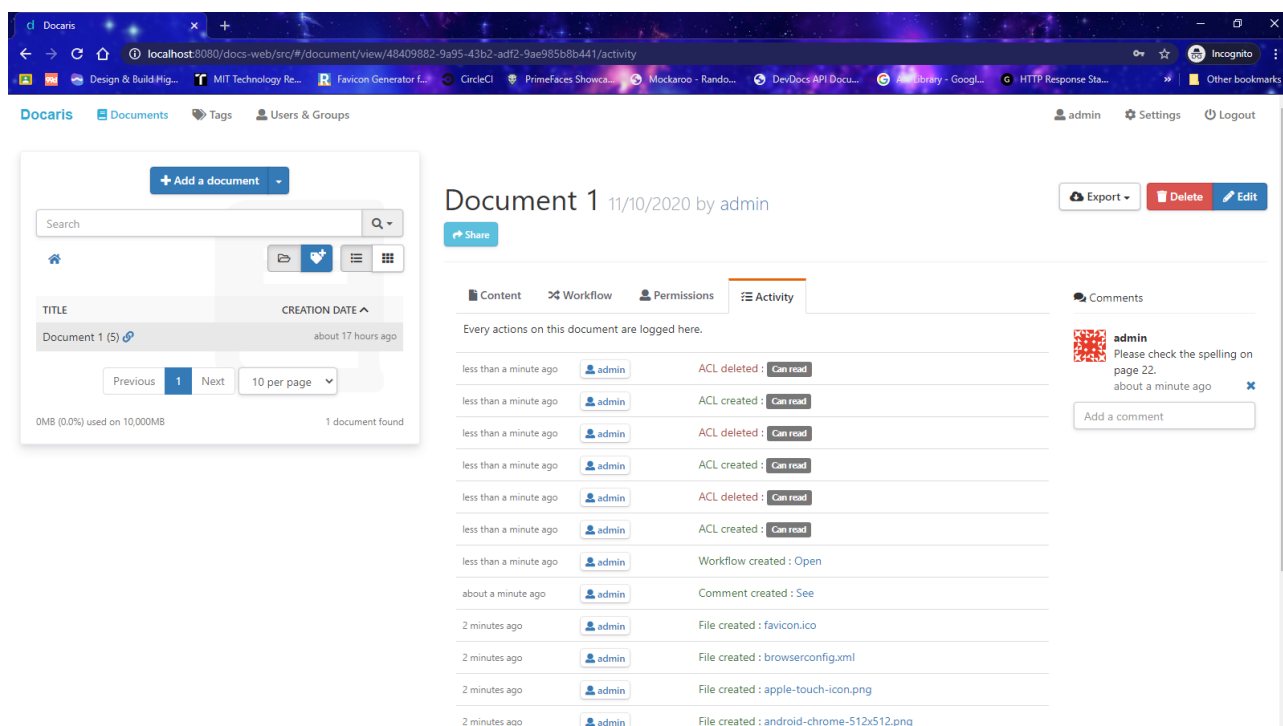


Fig. 4.7 Document Activity View

h. Tags

Tags are used on a document and can be used to search for documents fast, and assign permissions on a document based on the tag.

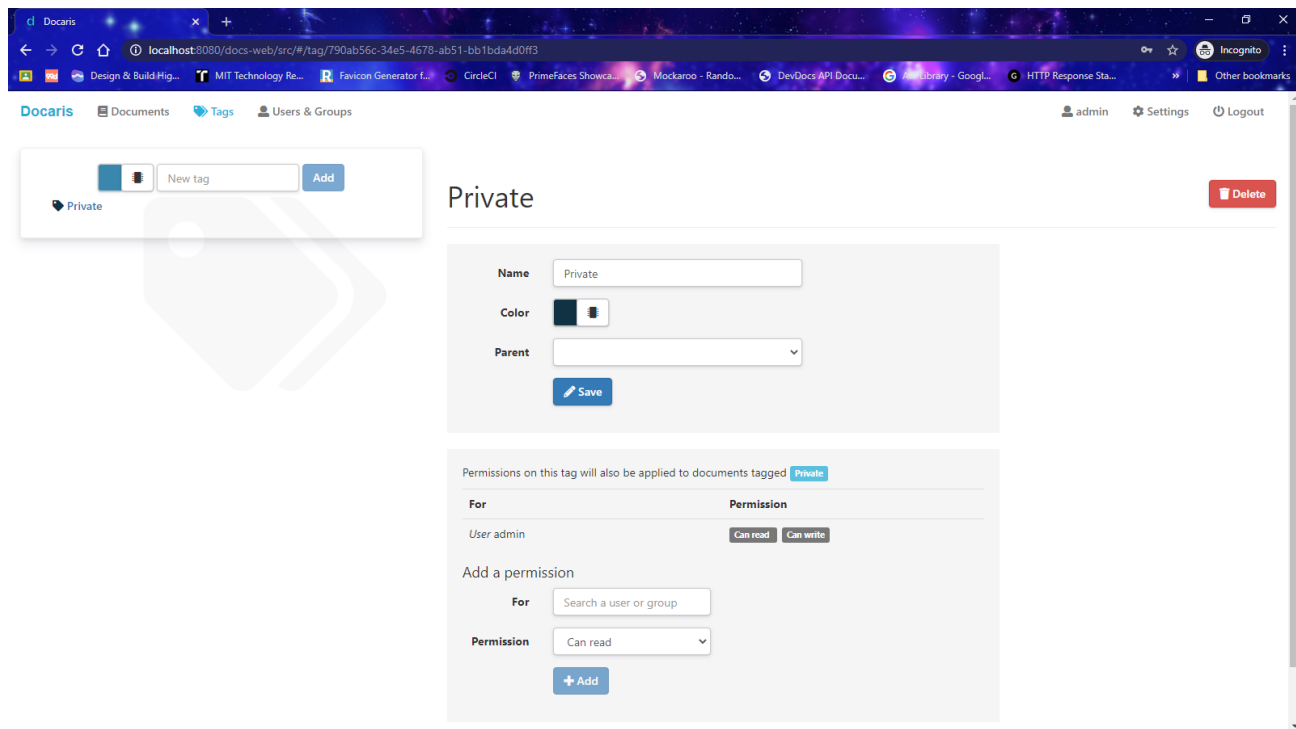


Fig. 4.8 Document Tags

i. Document Search

Documents can be searched for with text highlighting by the flexible search engine of the system.

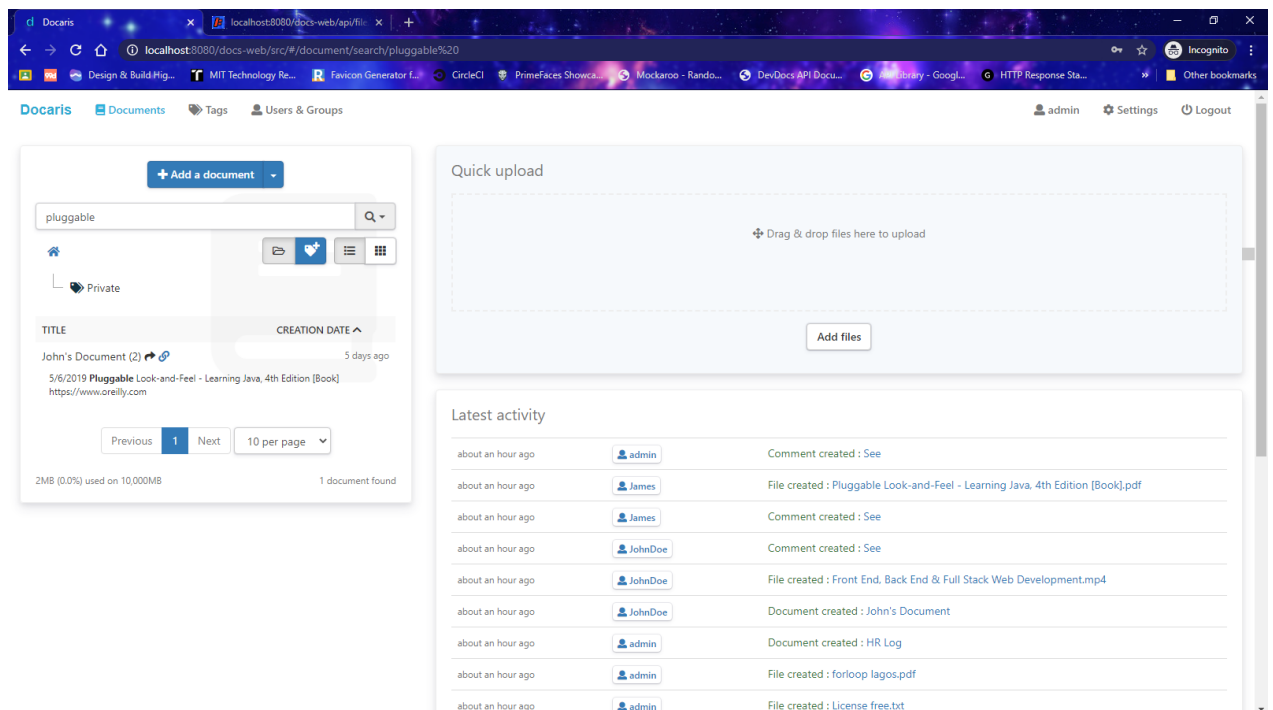


Fig. 4.9 Document Search with text highlighting

j. Workflow configuration

An administrator can create a new type of workflow to be used on documents.

The screenshot shows the Docaris web application interface for workflow configuration. The browser address bar indicates the URL is localhost:8080/docs-web/src/#/settings/workflow/add. The page title is "Workflow configuration" with an "Add a workflow" button. On the left, there are two sidebar menus: "Personal settings" (User account, Two-factor authentication, Opened sessions) and "General settings" (Workflow, Users, Groups, Vocabularies, Configuration, Custom metadata, Monitoring). The main content area features a table with one entry: "Document review" created on "11/10/2020". To the right is the "Add a workflow" form, which includes a "Name" field, a "Step type" dropdown set to "Validate", an "Assigned to" search field, a "Step name or description" text area, and a "Validated" section with an "Add a tag" dropdown. At the bottom of the form are "Add a workflow step" and "Save" buttons. The footer of the page reads "English - Crafted with ❤️ by Olusegun".

Fig. 4.10 Workflow Configuration

k. Edit System Configuration and Customize Theme

Administrators can edit the system's configuration and customize the theme.

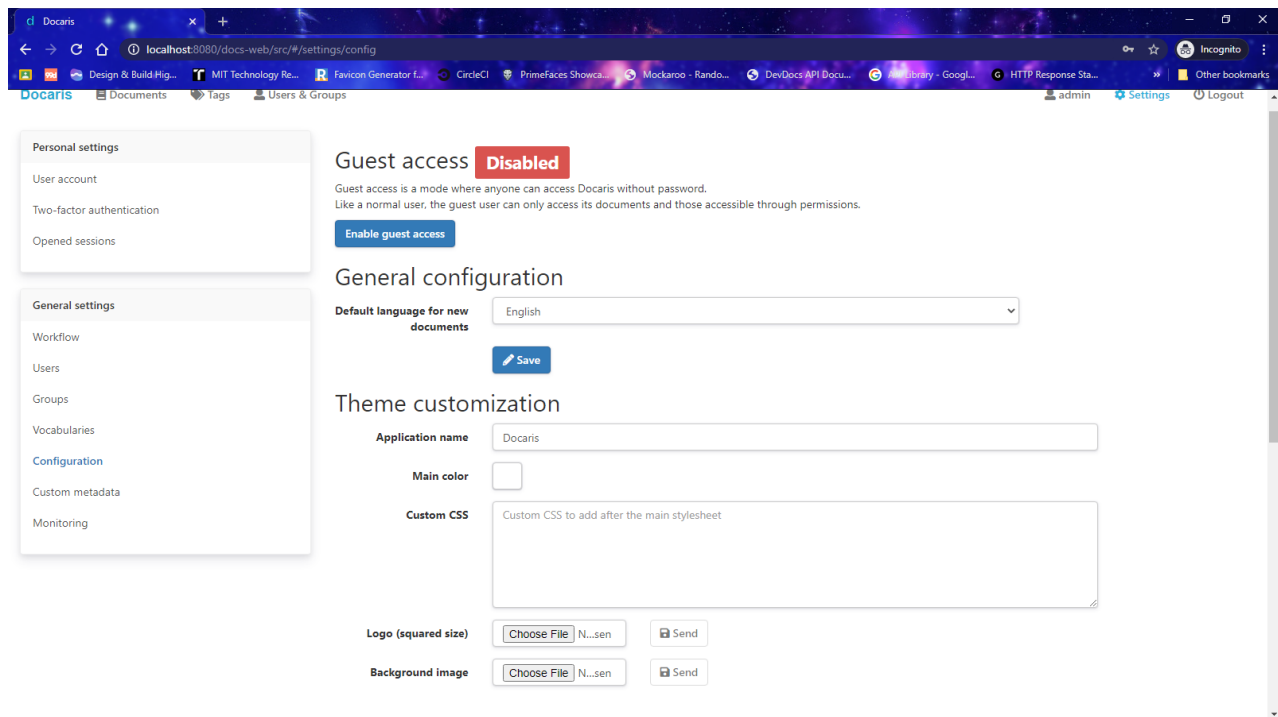


Fig. 4.11 System Configuration

I. Manage Users and User Groups

Administrators can manage the users and user groups of the system.

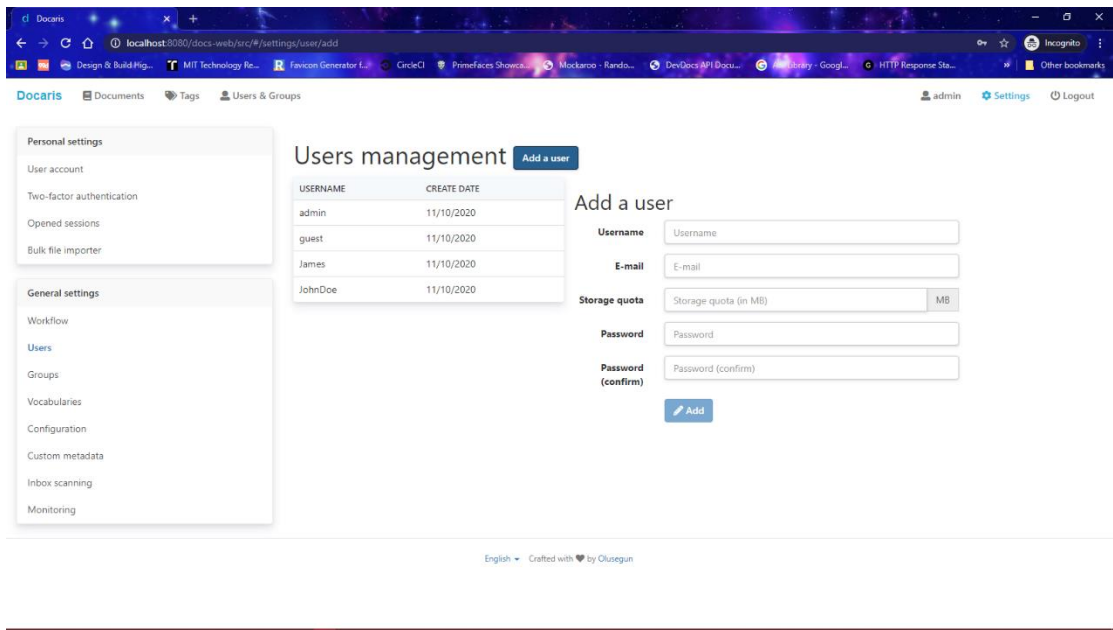
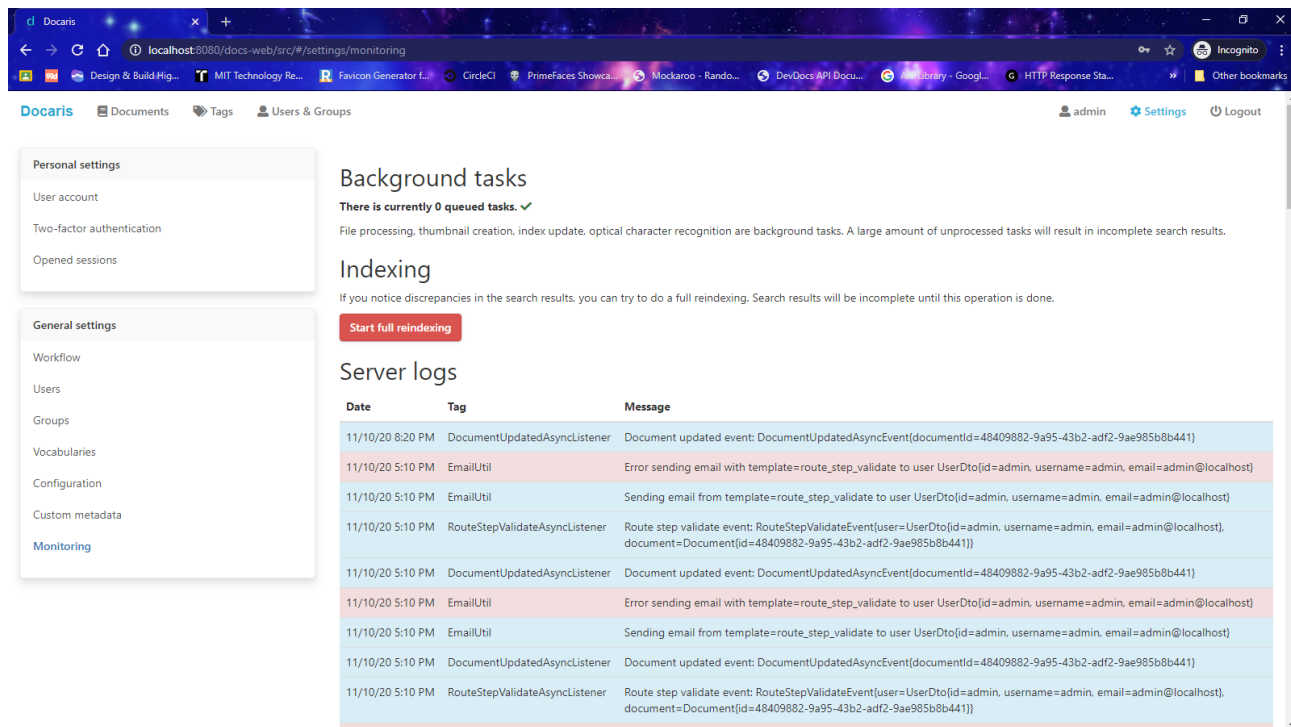


Fig. 4.12 User Management

m. Server Logs

An administrator can monitor and view server logs.



The screenshot shows the Docaris web application interface. On the left, there are navigation menus for 'Personal settings' (User account, Two-factor authentication, Opened sessions) and 'General settings' (Workflow, Users, Groups, Vocabularies, Configuration, Custom metadata, Monitoring). The main content area is titled 'Background tasks' and indicates 'There is currently 0 queued tasks.' Below this is the 'Indexing' section with a 'Start full reindexing' button. The 'Server logs' section displays a table of log entries.

Date	Tag	Message
11/10/20 8:20 PM	DocumentUpdatedAsyncListener	Document updated event: DocumentUpdatedAsyncEvent(documentId=48409882-9a95-43b2-adf2-9ae985b8b441)
11/10/20 5:10 PM	EmailUtil	Error sending email with template=route_step_validate to user UserDto(id=admin, username=admin, email=admin@localhost)
11/10/20 5:10 PM	EmailUtil	Sending email from template=route_step_validate to user UserDto(id=admin, username=admin, email=admin@localhost)
11/10/20 5:10 PM	RouteStepValidateAsyncListener	Route step validate event: RouteStepValidateEvent[user=UserDto(id=admin, username=admin, email=admin@localhost), document=Document(id=48409882-9a95-43b2-adf2-9ae985b8b441)]
11/10/20 5:10 PM	DocumentUpdatedAsyncListener	Document updated event: DocumentUpdatedAsyncEvent(documentId=48409882-9a95-43b2-adf2-9ae985b8b441)
11/10/20 5:10 PM	EmailUtil	Error sending email with template=route_step_validate to user UserDto(id=admin, username=admin, email=admin@localhost)
11/10/20 5:10 PM	EmailUtil	Sending email from template=route_step_validate to user UserDto(id=admin, username=admin, email=admin@localhost)
11/10/20 5:10 PM	DocumentUpdatedAsyncListener	Document updated event: DocumentUpdatedAsyncEvent(documentId=48409882-9a95-43b2-adf2-9ae985b8b441)
11/10/20 5:10 PM	RouteStepValidateAsyncListener	Route step validate event: RouteStepValidateEvent[user=UserDto(id=admin, username=admin, email=admin@localhost), document=Document(id=48409882-9a95-43b2-adf2-9ae985b8b441)]

Fig. 4.14 Server Logs

4.5 Result of Encrypted Files

The 256-bit encryption algorithm works efficiently and changes files uploaded to an encoded format called cyphertext which can be decrypted by the Document Management system as seen below in Fig. 4.15.

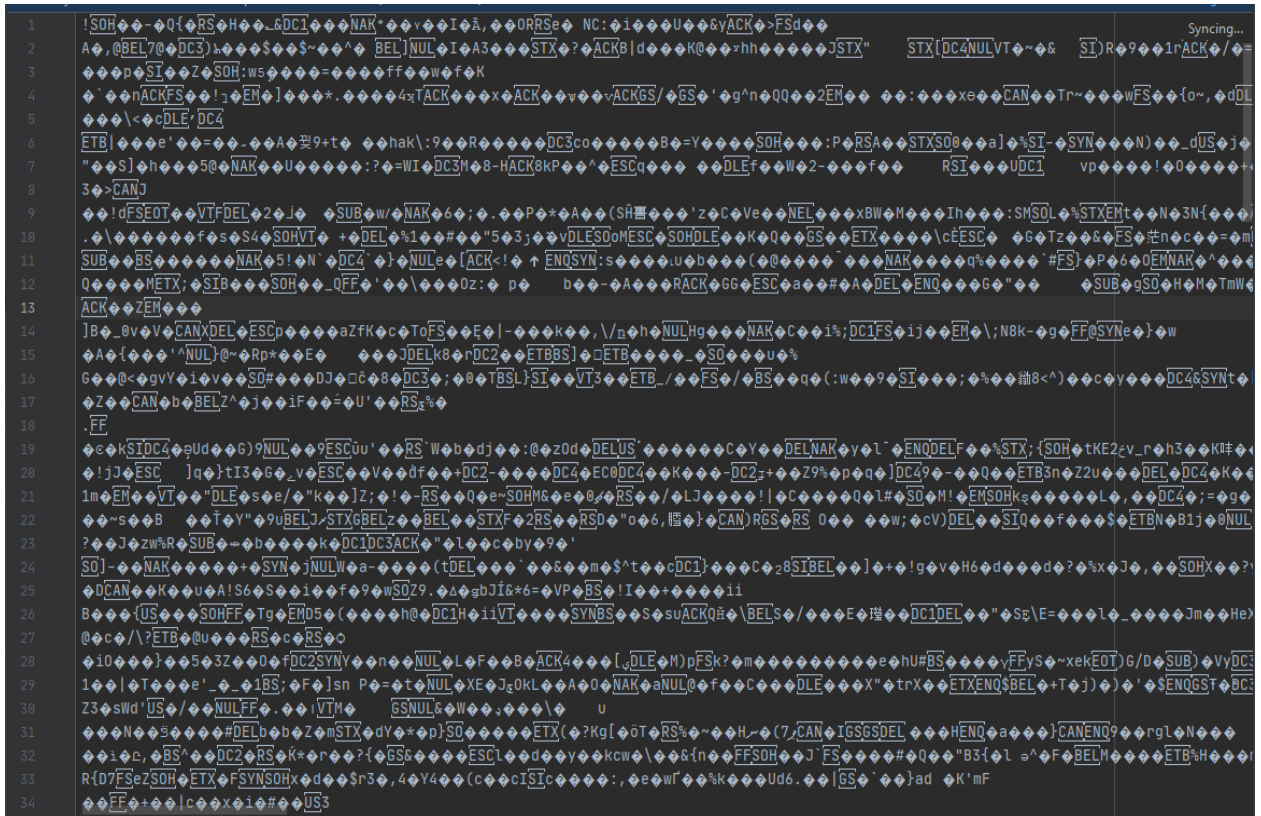


Fig. 4.14 File uploaded to Database after encryption

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.0 Introduction

This chapter describes a summary of results, conclusion, limitation to the study, contribution to knowledge and recommendation based on the results and discoveries of the work.

5.1 Summary

The designed system has handled the issues associated with the traditional means of storing documents in storage rooms and filing cabinets, the system is also very secure due to the use of the 256-bit AES encryption algorithm. The Document Management System is an efficient, time-saving and easy way to store, view and control multiple versions of a file and manage different formats of files. It is now an easy task and managing it is much easier.

5.2 Conclusion

Time is important, and time saved is a definite advantage of DMS, which also translates directly into improved productivity. Faster and more effective document retrieval will improve the morale of workers and increase customer satisfaction. Document management systems are also flexible to meet the changing needs of every organization.

5.3 Limitation to the study

- i.** The system does not support real-time co-authoring of documents.
- ii.** The system does not use any data compression algorithms.

5.4 Contribution to Knowledge

This research work will serve as a detailed resource for further research in designing and implementing a document management system using Java and its Cryptography extension.

Document Management System with improved security and space management features.

The system has a lot of features which ensures a smooth and secure operation in managing documents.

5.5 Recommendation for Further Studies

For future research, the feature to collaborate with other users in real-time while working on documents should be implemented and data compression algorithms should be used to achieve less usage of digital storage space.

REFERENCES

- 1stwebdesigner. (2018, March 14). *project-management-collaboration-tools*. Retrieved from 1stwebdesigner: <http://www.1stwebdesigner.com/project-management-collaboration-tools/>
- Aiim. (2020). *What is Document Management (DMS)?* Retrieved March 12, 2020, from Aiim.org: <https://www.aiim.org/What-Is-Document-Imaging#>
- 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)*, 55-58.
- Anwar, M. A., & Naseer, A. (2013). *An e-Course file management system: A green campus. Vol. 3, No. 1.*
- ATP Electronics,Inc. (2019, June 26). *Secure your data with AES-256 encryption*. Retrieved from ATP Electronics,Inc.: <https://www.atpinc.com/blog/what-is-aes-256-encryption>
- Blau, I., & Caspi, A. (2009). What Type of Collaboration Helps? *Psychological Ownership, Perceived Learning and Outcome Quality of Collaboration Using Google Docs.*, 48-55.
- Capterra. (2020, April 14). *Document Management Software*. Retrieved from Capterra: <https://www.capterra.com/document-management-software/>

- Course Hero. (n.d.). *Within this article, we will mostly tackle collaboration and communication*. Retrieved March 13, 2020, from Course Hero:
<https://www.coursehero.com/file/p4ndfas/Within-this-article-we-will-mostly-tackle-the-collaboration-and-communication/>
- Diffie, W., & Hellman, M. (1976). *New Directions in Cryptography, IEEE Transactions on Information Theory*.
- eFileCabinet. (2014, August 27). *Short History of Document Management*. Retrieved from www.efilecabinet.com: <https://www.efilecabinet.com/document-management/>
- Gilson, L. L. (2015). Virtual Teams Research: 10 years, 10 Themes, and 10 Opportunities. *Journal of Management*, 1313-1337. Retrieved from <http://doi.org/10.1177/0149206314559946>
- Groenewald, T. (2004). Electronic Document Management: Human Resource Management. *SA Journal of Human Resource Management*, 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*.
- Jeeva, A. L., V., P., & 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., 3033-3037.
- Kahanwal, B., Dua, K., & Singh, G. P. (2012). Java File Security System (JFSS). *Global Journals Inc., 12: Version 1.0*.

- Katz, R. &. (1979). Communication patterns, project performance, and task characteristics: An empirical evaluation and integration in an R&D setting. *Organizational Behaviour and Human Performance*, 139-162. Retrieved from [http://doi.org/10.1016/0030-5073\(79\)90053-](http://doi.org/10.1016/0030-5073(79)90053-)
- Lake, J. (2020, February 17). *What is AES encryption and how does it work?* Retrieved from Comparitech: <https://www.comparitech.com/blog/information-security/what-is-aes-encryption/>
- Lord, N. (2020, September 30). *Data Protection 101: What Is Data Encryption? Definition, Best Practices & More.* Retrieved from DigitalGuardian: <https://digitalguardian.com/blog/what-data-encryption>
- 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.
- Mansor, A. (2012). Google Docs as a Collaborating Tool for Academicians. *Procedia - Social and Behavioral Sciences*, 59, 411-419. Retrieved from <http://doi.org/10.1016/j.sbspro.2012.09.295>
- Mushtaque, M. A. (2014). Comparative Analysis of Different Parameters of Encryption Algorithms for Information Security. *International Journal of Computer Sciences and Engineering Open Access Research Paper. Vol. 2.*, 2347-2693.
- Nicoletti, B. (2012). Project Management and Cloud Computing. *PM World Today*, 1-11. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=74028642&site=ehost-live>

- O'Reilly, T. (2005). *What is Web 2.0?* Retrieved December 18, 2019, from O'Reilly Media.: <http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=1>
- Paperwise. (2015). *Glossary of terms*. Retrieved from Paperwise: <https://downloads.paperwise.com/resources/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*.
- Richard Schoen, G. S. (2016). *8 Reasons It's Time to Implement*. Retrieved from <https://www.slideshare.net/HelpSystems/8-reasons-you-need-an-electronic-document-management-system>
- Schmidt, G. J. (2019, March 15). *How Does a Document Management System Work?* Retrieved from HelpSystems Blog: <https://www.helpsystems.com/blog/how-does-document-management-system-work>
- Simar, P. S., & Raman, M. (2011). Comparison of Data Encryption Algorithms. *International Journal of Computer Science and Communication, Vol 2.*, 125-127.
- Singhal, N., & Raina, J. P. (2011). Comparative Analysis of AES and RC4 Algorithms for Better Utilization. *International Journal of Computer Trends and Technology*.
- Stosic, L. (2013). Computer Security and Security Technologies. (*JPMNT*) *Journal of Process Management – New Technologies, Vol. 1, No.1*.

- Suwantarathip, O. &. (2014). The Effects of Collaborative Writing Activity Using Google Docs on Students's Writing Abilities. *Turkish Online Journal of Educational Technology - TOJET*, 13(2005), 148-156. Retrieved from http://eric.ed.gov/?q=Google+Docs&ff1=dySince_2010&id=EJ1022935
- Tan, X. &. (2015). User acceptance of SaaS-based collaboration tools: a case of Google Docs. *Journal of Enterprise Information Management*, 423-442. Retrieved from <http://doi.org/10.1108/JEIM-04-2014-0039>
- Tomislav Rozman, A. R. (2017, January 1). *An Analysis of Web-based Document Management and Communication Tools*. Retrieved from ResearchGate: <http://www.igi-global.com/article/an-analysis-of-web-based-document-management-andcommunication->
- Vienna Advantage. (n.d.). *7 Characteristics of the Ideal Document Management System*. Retrieved January 8, 2020, from Vienna Advantage: <http://www.viennaadvantage.com/dms-overview.php>
- Ward, S. (2020, January 7). *How to Create a Document Management System*. Retrieved from The balance small business: <https://www.thebalancesmb.com/creating-a-document-management-system-2948084>
- Watson, R. (2006). Extending Google Docs to Collaborate on Research Papers. *Toowoomba Queensland AU The University of Southern Queensland Australia*, 23, 1-11. Retrieved from [Retrieved from http://www.sci.usq.edu.au/staff/dekeyser/googledocs.pdf](http://www.sci.usq.edu.au/staff/dekeyser/googledocs.pdf)
- Wikipedia. (2020). *Data Modelling*. Retrieved from Wikipedia the free Encyclopaedia.

- Wikipedia. (2020, March 11). *Document Management System*. Retrieved from Wikipedia, the free encyclopedia:
https://en.wikipedia.org/wiki/Document_management_system
- Wikipedia. (2020). *Systems design*. Retrieved April 17, 2020, from Wikipedia, the free encyclopedia: https://en.wikipedia.org/wiki/Systems_design
- Yousuf, M., & Summer, M. T. (2011). *Secure Emails: An Integrity Assured Email Systems Using PKI*.
- Zammit, I. (2020, March). *Document Management System Definition*. Retrieved from Folderit web site: <https://www.folderit.com/blog/document-management-system-definition/>
- Zeeman, A. (2019, November 27). *Document Management*. Retrieved from Toolshero: <https://www.toolshero.com/information-technology/document-management/>
- Zhou, W. S. (2012). Google Docs in an Out-of-Class Collaborative Writing Activity. *International Journal of Teaching and Learning in Higher Education*, 359-375. Retrieved from <http://www.isetl.org/ijtlhe/>

APPENDIX

SOURCE CODE

```
public class DocumentDao {  
    /**  
     * Creates a new document.  
     * @param document Document  
     * @param userId User ID  
     * @return New ID  
     */  
    public String create(Document document, String userId) {  
        // Create the UUID  
        document.setId(UUID.randomUUID().toString());  
        document.setUpdateDate(new Date());  
        // Create the document  
        EntityManager em = ThreadLocalContext.get().getEntityManager();  
        em.persist(document);  
        // Create audit log  
        AuditLogUtil.create(document, AuditLogType.CREATE, userId);  
  
        return document.getId();  
    }  
    /**  
     * Returns the list of all active documents.  
     *  
     * @param offset Offset
```

```

* @param limit Limit
* @return List of documents
*/
@SuppressWarnings("unchecked")
public List<Document> findAll(int offset, int limit) {
    EntityManager em = ThreadLocalContext.get().getEntityManager();
    Query q = em.createQuery("select d from Document d where d.deleteDate is null");
    q.setFirstResult(offset);
    q.setMaxResults(limit);
    return q.getResultList();
}
/**
* Returns the list of all active documents from a user.
*
* @param userId User ID
* @return List of documents
*/
@SuppressWarnings("unchecked")
public List<Document> findByUserId(String userId) {
    EntityManager em = ThreadLocalContext.get().getEntityManager();
    Query q = em.createQuery("select d from Document d where d.userId = :userId and
d.deleteDate is null");
    q.setParameter("userId", userId);
    return q.getResultList();
}
/**
* Returns an active document with permission checking.
*

```

```

* @param id Document ID
* @param perm Permission needed
* @param targetIdList List of targets
* @return Document
*/

public DocumentDto getDocument(String id, PermType perm, List<String> targetIdList) {

    AclDao aclDao = new AclDao();

    if (!aclDao.checkPermission(id, perm, targetIdList)) {

        return null;

    }

    EntityManager em = ThreadLocalContext.get().getEntityManager();

    StringBuilder sb = new StringBuilder("select distinct d.DOC_ID_C, d.DOC_TITLE_C,
d.DOC_DESCRIPTION_C, d.DOC_SUBJECT_C, d.DOC_IDENTIFIER_C,
d.DOC_PUBLISHER_C, d.DOC_FORMAT_C, d.DOC_SOURCE_C, d.DOC_TYPE_C,
d.DOC_COVERAGE_C, d.DOC_RIGHTS_C, d.DOC_CREATEDATE_D,
d.DOC_UPDATEDATE_D, d.DOC_LANGUAGE_C, ");

    sb.append(" (select count(s.SHA_ID_C) from T_SHARE s, T_ACL ac where
ac.ACL_SOURCEID_C = d.DOC_ID_C and ac.ACL_TARGETID_C = s.SHA_ID_C and
ac.ACL_DELETEDATE_D is null and s.SHA_DELETEDATE_D is null) shareCount, ");

    sb.append(" (select count(f.FIL_ID_C) from T_FILE f where f.FIL_DELETEDATE_D
is null and f.FIL_IDDOC_C = d.DOC_ID_C) fileCount, ");

    sb.append(" u.USE_USERNAME_C ");

    sb.append(" from T_DOCUMENT d ");

    sb.append(" join T_USER u on d.DOC_IDUSER_C = u.USE_ID_C ");

    sb.append(" where d.DOC_ID_C = :id and d.DOC_DELETEDATE_D is null ");

    Query q = em.createNativeQuery(sb.toString());

    q.setParameter("id", id);

    Object[] o;

```

```

try {
    o = (Object[]) q.getSingleResult();
} catch (NoResultException e) {
    return null;
}

DocumentDto documentDto = new DocumentDto();

int i = 0;

documentDto.setId((String) o[i++]);

documentDto.setTitle((String) o[i++]);

documentDto.setDescription((String) o[i++]);

documentDto.setSubject((String) o[i++]);

documentDto.setIdentifier((String) o[i++]);

documentDto.setPublisher((String) o[i++]);

documentDto.setFormat((String) o[i++]);

documentDto.setSource((String) o[i++]);

documentDto.setType((String) o[i++]);

documentDto.setCoverage((String) o[i++]);

documentDto.setRights((String) o[i++]);

documentDto.setCreateTimestamp(((Timestamp) o[i++]).getTime());

documentDto.setUpdateTimestamp(((Timestamp) o[i++]).getTime());

documentDto.setLanguage((String) o[i++]);

documentDto.setShared(((Number) o[i++]).intValue() > 0);

documentDto.setFileCount(((Number) o[i++]).intValue());

documentDto.setCreator((String) o[i]);

return documentDto;
}

/**
 * Deletes a document.

```

```

*
* @param id Document ID
* @param userId User ID
*/
public void delete(String id, String userId) {
    EntityManager em = ThreadLocalContext.get().getEntityManager();

    // Get the document
    Query q = em.createQuery("select d from Document d where d.id = :id and d.deleteDate
is null");

    q.setParameter("id", id);

    Document documentDb = (Document) q.getSingleResult();

    // Delete the document
    Date dateNow = new Date();
    documentDb.setDeleteDate(dateNow);

    // Delete linked data
    q = em.createQuery("update File f set f.deleteDate = :dateNow where f.documentId =
:documentId and f.deleteDate is null");

    q.setParameter("documentId", id);
    q.setParameter("dateNow", dateNow);
    q.executeUpdate();

    q = em.createQuery("update Acl a set a.deleteDate = :dateNow where a.sourceId =
:documentId and a.deleteDate is null");

    q.setParameter("documentId", id);
    q.setParameter("dateNow", dateNow);
    q.executeUpdate();

    q = em.createQuery("update DocumentTag dt set dt.deleteDate = :dateNow where
dt.documentId = :documentId and dt.deleteDate is not null");

    q.setParameter("documentId", id);

```

```

    q.setParameter("dateNow", dateNow);

    q.executeUpdate();

    q = em.createQuery("update Relation r set r.deleteDate = :dateNow where
(r.fromDocumentId = :documentId or r.toDocumentId = :documentId) and r.deleteDate is not
null");

    q.setParameter("documentId", id);

    q.setParameter("dateNow", dateNow);

    q.executeUpdate();

    // Create audit log

    AuditLogUtil.create(documentDb, AuditLogType.DELETE, userId);

}

/**
 * Gets an active document by its ID.
 *
 * @param id Document ID
 * @return Document
 */

public Document getById(String id) {

    EntityManager em = ThreadLocalContext.get().getEntityManager();

    Query q = em.createQuery("select d from Document d where d.id = :id and d.deleteDate
is null");

    q.setParameter("id", id);

    try {

        return (Document) q.getSingleResult();

    } catch (NoResultException e) {

        return null;

    }

}

```

```

/**
 * Update a document and log the action.
 *
 * @param document Document to update
 * @param userId User ID
 * @return Updated document
 */
public Document update(Document document, String userId) {
    EntityManager em = ThreadLocalContext.get().getEntityManager();

    // Get the document

    Query q = em.createQuery("select d from Document d where d.id = :id and d.deleteDate
is null");

    q.setParameter("id", document.getId());

    Document documentDb = (Document) q.getSingleResult();

    // Update the document

    documentDb.setTitle(document.getTitle());

    documentDb.setDescription(document.getDescription());

    documentDb.setSubject(document.getSubject());

    documentDb.setIdentifier(document.getIdentifier());

    documentDb.setPublisher(document.getPublisher());

    documentDb.setFormat(document.getFormat());

    documentDb.setSource(document.getSource());

    documentDb.setType(document.getType());

    documentDb.setCoverage(document.getCoverage());

    documentDb.setRights(document.getRights());

    documentDb.setCreateDate(document.getCreateDate());

    documentDb.setLanguage(document.getLanguage());

    documentDb.setFileId(document.getFileId());

```

```

        documentDb.setUpdateDate(new Date());

        // Create audit log
        AuditLogUtil.create(documentDb, AuditLogType.UPDATE, userId);

        return documentDb;
    }

    /**
     * Update the file ID on a document.
     *
     * @param document Document
     */
    public void updateFileId(Document document) {
        EntityManager em = ThreadLocalContext.get().getEntityManager();

        Query query = em.createNativeQuery("update T_DOCUMENT d set DOC_IDFILE_C =
:fileId, DOC_UPDATEDATE_D = :updateDate where d.DOC_ID_C = :id");

        query.setParameter("updateDate", new Date());
        query.setParameter("fileId", document.getFileId());
        query.setParameter("id", document.getId());

        query.executeUpdate();
    }

    /**
     * Returns the number of documents.
     *
     * @return Number of documents
     */
    public long getDocumentCount() {
        EntityManager em = ThreadLocalContext.get().getEntityManager();

        Query query = em.createNativeQuery("select count(d.DOC_ID_C) from
T_DOCUMENT d where d.DOC_DELETEDATE_D is null");

        return ((Number) query.getSingleResult()).longValue();
    }

```


}
}