DEVELOPMENT OF A LAN-BASED FILE SHARING SYSTEM

BY

OKONKWO AUGUSTINE CHIADIKAOBI
GOU/U22/CSC/848

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY, GODFREY OKOYE UNIVERSITY, ENUGU STATE.

IN PARTIAL FULFILLMENT OF THE AWARD OF BACHELOR OF SCIENCE (B.SC), DEGREE IN COMPUTER SCIENCE.

SUPERVISOR: PROF. N. M. OZOFOR

JUNE, 2026

APPROVAL PAGE

This research project, titled "Development of AI-Powered Digital Library System for Past University Course Materials," has been examined and approved by the Department of Computer Science and Mathematics, Faculty of Computing and Information Technology, Godfrey Okoye University, Enugu, Nigeria.

Prof. N. M. Ozofor ....

Supervisor Signature\Date

External Examiner ....

External Examiner Signature\Date

Dr. Frank Okebanama .....

HOD, Computer Science & Mathematics Signature\Date

Prof. I. A. Ajah .....

Dean, Faculty of Computing and Information Technology Signature\Date

CERTIFICATION

I certify that this project was carried out by me and that all sources of information used have been properly acknowledged. I understand that the university will take action if it is found that I submitted work that is not entirely mine.

OKONKWO AUGUSTINE CHIADIKAOBI
GOU/U22/CSC/848

DEDICATION

This project is lovingly dedicated to my parents, whose unwavering support, sacrifices, encouragement, and belief in my abilities have been the foundation of my academic journey.Their constant prayers, guidance, and commitment to my education have inspired me to persevere through every challenge and remain focused on my goals. I also dedicate this work to all students who strive for academic excellence despite the difficulties they encounter in accessing relevant study materials and examination resources. It is my sincere hope that this project will contribute, in its own way, to making learning more accessible, convenient, and effective for present and future students. Finally, this work is dedicated to everyone whose efforts and support, whether direct or indirect, have contributed to the successful completion of this project.

ACKNOWLEDGEMENTS

First and foremost, I give all glory and thanks to Almighty God for His grace, strength, wisdom, and guidance throughout the course of this study. Without His blessings and mercy, the successful completion of this project would not have been possible. My sincere appreciation goes to my supervisor, Prof. N. M. Ozofor, whose invaluable guidance, constructive criticisms, patience, and encouragement greatly contributed to the success of this work.

His willingness to provide direction and support whenever challenges arose was instrumental in shaping this project. I am deeply grateful to my parents and family members for their unwavering love, prayers, sacrifices, and moral support throughout my academic journey. Their belief in my abilities has been a constant source of motivation and inspiration. I also extend my heartfelt appreciation to my friends, classmates, and colleagues who contributed in various ways to the successful completion of this project. Their support, suggestions, and encouragement are sincerely appreciated.Finally, I acknowledge all those who, directly or indirectly, contributed to the realization of this work. May God richly bless you all.

ABSTRACT

In today's world, there is an increasing need for organisations and educational institutions to share information and collaborate effectively. However, the traditional methods of sharing files and communicating demonstrate several limitations, such as security risks, file duplication, limited access control, reliance on the Internet, and inefficient methods of collaboration. Therefore, this study has been undertaken with the aim of designing and developing a new method for sharing files securely over a local area network (LAN) using a File Sharing and Communication System (FSCS). To implement the aim and objectives of this study, the Object-Oriented Analysis and Design (OOAD) Methodology and Unified Modelling Language (UML) diagrams were used in order to provide model architecture and functionality for the CSFS. The CSFS was developed using software technologies such as Python, FastAPI, PostgreSQL, SQLAlchemy, HTML, CSS, JavaScript, and Bootstrap in a client/server environment. The primary features of the FSCS include: user authentication, role-based access control, user upload and download files, real-time messaging, activity logging, and management by administrators. To evaluate the performance of the FSCS, thirteen test cases were run on all functional modules of the FSCS, with all thirteen test cases having positive results. The results support the development of an FSCS that is an Internet-independent, cost-effective and secure method for sharing files and using other communication methods can lead to better collaboration and use of resources, as well as the enhancement of communication capabilities.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

The evolution of computer networks over recent decades and their continuous growth have, in a great way, transformed how organisations manage, share, and protect digital information. Network technology has evolved from the era of the mainframe in the 1960s to the age of minicomputers in the 1970s and microprocessors in the 1980s, to a world of interconnected ecosystems today, allowing seamless data exchange (Bhupathi, 2025). Local Area Networks (LANs) are designed to meet the need for low cost, high data-rate channels between machines operating within limited geographic location ranges. They began to be developed in the 1960s and 70s, but the majority of LANs were developed during the 1980s when computers became commonplace in the business and institutional worlds. Today, LANs are utilized as the fundamental element to link workstations, personal computers, and peripherals so as to share resources, share work processes, and communicate efficiently between individuals, which are all essential components of today's business practices (Goni & Shameem, 2021). For organisations operating such localised networks, sharing of resources, cooperation and good communication are essential to their productivity. They enable the sharing of files amongst teams, print from a single central printer, and have instant access to information at any workstation in the same building (Santos & Lopes, 2025).

Traditional file-sharing methods have been the main tools people have used to share digital material with each other for a long time, but they have had significant limitations that impact on effective and secure data transfer in the modern era, such as USB flash drives and external hard disks, which offer portable storage and at the same time have a weak ability to record who has received the data, and also they can be lost or stolen, even if the external hard drive is faster than the USB flash drive, the data can be lost or irretrievably damaged if the device is lost or stolen (Zhu et al., 2025). Sending large amounts of data via Bluetooth can be limited in speed and connectivity. There's still a great deal of businesses that are sending insecure Email attachments. Verizon's 2024 Data Breach Reports found that 94% of all malware transmitted to businesses and

organizations is received via email; almost half of the HTML embedded emails received are spam or contain a malware payload. Once an e-mail is created and sent with an attachment, it does not remain under management control and can be forwarded to other individuals, downloaded to personal devices, or uploaded to non-existent cloud services. This creates a path of data exposure that cannot be traced. Shared network folders typically result in version conflicts; when multiple email threads contain different document versions, it causes confusion and lost documents. Other than sharing files through shared drives, sharing files with physical drives only allows sharing data between one device at one time; this creates a collaborative bottleneck (Lonzetta et al., 2018).

As the demand for user-to-user, real-time communication continues to escalate in educational environments, commercial businesses, healthcare organisations, and any other organisation where timely communication between personnel will directly affect the efficiency of the organisation and the quality of decision making (Liu et al., 2025), real-time communications (RTC) or communications delivered within seconds of sending the message, such as instant messaging, video calling, tele/video conferencing and internet relay chat have become the mainstay of how employees in the modern workplace communicate with one another; however, without the benefit of a communications strategy to govern usage of these RTC channels, employees may become confused about which RTC channel they should use, leading to the dilution of messages sent via certain RTC channels, unproductively wasting time and energy in searching for messages, and exposing the organisation's systems to security breaches through the accidental transmission of message content through a different RTC channel than was intended (Mahmoud & Abozariba, 2025).

In the healthcare environment, poor communications that occur because of network latency between the sender and the recipient of a message results in delays in the transmission of messages that can lead to delays in making consult or care decisions, which in turn affects the expected patient care outcome because personnel are unable to communicate effectively. Delays associated with relying on RTC messaging via third-party service providers present an organisation with many challenges since many of these mobile communication applications do not protect organisations from the changing nature of sophisticated cyberattacks and are not compliant with recent regulatory requirements. RTC communications tools have become an integral part of doing business; however, a data security breach involving the use of RTC

communications platforms would have remarkably negative ramifications on an organisation, including significant leakages of confidential internal documents, and the unauthorised disclosure of very confidential, sensitive user information; both of which are clear indications that many third-party RTC platforms are not secure. Due to the absence of unified communication systems in many environmental networks, many are reliant upon third-party companies that are not bound by the client's control. As a result, this creates a security gap whereby external contractors, vendors, and customers can be invited into channels, third-party bots and apps can access message history and files, and cross-workspace messaging, which in turn can create huge blind spots for security teams (Howick et al., 2025). Communication inefficiencies significantly impact productivity and collaboration, as studies we reviewed show that 40% of businesses report reduced employee productivity due to inadequate internal collaboration, 38% report that ineffective collaboration costs them business, and 49% found that lack of communication and collaboration in a significant way impacted customer experience negatively (Subreman 2026).

Unauthorized access to files & resources is a major security problem faced by all organizations; it results from a lack of sufficient security controls for both the files & resources being accessed and the individuals and programs that misuse them. This access by unauthorized entities can occur as a result of misconfigured access policies, excessive permissiveness of access settings and ineffective authentication controls, and these factors create a security vulnerability that can be exploited. The weakest link in the security chain and the primary cause of security breaches is having no or poor user authentication. In order to effectively restrict access to networked devices only to authorised users/devices, it is therefore imperative to employ strong authentication methods, such as passwords, biometrics and two-factor authentication, which are necessary to prevent unauthorised access. To create a secure environment for sharing files & collaborating on projects, organizations must have secure methods for transmitting data and controlling file access over LANs; the methods they use must include end-to-end encryption, two-factor authentication for all users; role-based access controls; password protected sharing of files with expiration dates and a full audit log/history of use of files to keep sensitive data secure and support authorized users working together on shared files (Hasan et al., 2025).

Web access and usability offer both environmental benefits to providing quicker and less expensive sharing of documents, as compared to using hard-copy materials; do not have limitations on the size of files that can be attached and emailed; are available for anyone who has an internet connection from any location in the world; and allow for simple sharing of access to an individual or group of employees who do not work on-site, but are located anywhere in the world. Efficient data management requires secure and efficient methods of sharing data across multiple domains using data floating securely through attribute-based encryption in order to protect the privacy of Industrial data and prevent unauthorised access, while still assuring data confidentiality, integrity or availability across different networks with secure and robust mechanisms for encrypting, authenticating or controlling access to data (Silva et al., 2025). Overall, there is a need to develop a lan-based file sharing and communication web application, which will allow users to share files with one another over a LAN, providing users with a means to securely and reliably exchange information. It will also allow organisations to have an integrated solution that eliminates the critical gaps that exist with traditional file sharing methods, provides redundancy, and takes advantage of the high-speed, low-latency capabilities of a LAN infrastructure in order to effectively deliver the best performance possible in geographically bound environments, while at the same time, providing administrators with the ability to maintain control of the network administration functions within their organizations.

1.2 Statement of Problem

The fact that there are advanced communication technologies does not prevent the fact that in many academic institutions and organisations, a lack of efficiency exists in the sharing of internal files and communications. The key issues that are linked with the existing methods are:

i. Risk of viruses or data loss on external storage devices: Flash drives and other removable storage devices can be infected with viruses, lose data, or break.

ii. Internet Dependency: Internet-based system of file sharing needs a reliable Internet connection, but this may not be available or affordable.

ii. Security Risks: With online platforms your personal info can be accessed by others and can lead to data leaking or breaking, and breaking the law by having access to your private information.

iii. Time-consuming File Transfer: With big files transferring through email or the cloud can take a very long time, therefore you will want something other than those methods to send files.

iv. Absence of Centralised Control: In most cases your information is not stored or managed properly in a local environment; there is usually no way of knowing what is happening with your data.

The above issues highlight the need for an alternative file sharing and communication solution that is cost effective, secure, and does not rely on the internet. These issues can be fixed with a LAN based solution that creates a controlled environment for the transfer of files in a closed network setting.

1.3 Aims and Objectives of the Study

This study aims to design and implement a LAN-Based File Sharing and Communication System that enables efficient and secure data exchange within a local network.

The specific objectives of this study are to:

1. Examine the existing LAN-based file sharing and communication systems.
2. Identify the requirements needed in order to implement secure and efficient local network communication.
3. Design a web-based architecture for file sharing and communication.
4. Implement authentication and authorisation mechanisms for secure system access.
5. Integrate real-time communication features for enhanced user collaboration.
6. Evaluate and compare the system's performance, security, and usability.

1.4 Significance of the Study

There are several reasons why the LAN-based file sharing and Communication System is important:

Better Security: Decreases vulnerability to malware and unauthorised entry through the use of no external storage devices and no Internet-based platforms.

Better Efficiency: Provides quick and accurate transfer of files in the local network. Internet Usage: Saves on external storage costs and Internet usage.

Improved Collaboration: Users can communicate and share resources easily.

Educational Value: Offers hands-on experience with networking principles, client/server and system development.

This study will also help develop effective LAN solutions and show how LAN technology can be used to meet the requirements of a real-world communication and data sharing problem.

1.5 Scope of the Study

This study will focus on the creation of a file-sharing and communication system which works inside a Local Area Network (LAN) without connections to the internet. The system will allow the following features:

1. Security in the LAN (user authentication and access control).
2. Upload, download and manage shared files.
3. Full-featured chat feature for users who are connected.

A single server-based administration of resources that can be shared. Internet-based file sharing, cloud storage services, wide area network (WAN) communication and multimedia communication (voice and video calls) are not covered by the system. Only a controlled LAN environment is used for the implementation and testing of the system. As regards the file types it can process, it is developed to deal with academic file types typically used in institutional settings. These include document files (PDF, DOCX, PPTX, XLSX), plain text files (TXT), image files (JPEG, PNG), and compressed archive files (ZIP, RAR) that contain student assignments, project reports, course notes, lecture slides, and examination materials.

1.6 Limitations of the Study

The proposed LAN-based file sharing and Communication System has benefits, but this study has some limitations; the scope, implementation and performance of the system might be affected by these limitations.

First, the system is intended to be run as a Local Area Network (LAN) system. Therefore, it cannot be used to communicate or share files on larger networks like Wide Area Networks (WANs) or the Internet. This will restrict who can be able to use it remotely, as only those who are physically connected to the same network will be able to use it.

Secondly, the system relies mainly on text communication and basic functionality of file sharing. The scope of this study does not include advanced features such as voice communication, video conferencing, and multimedia streaming, which would be too time-consuming and complex to include. This restricts the system's capacity to be a full-fledged communication platform.

Another restriction is the implementation of security. The system may have access control and authentication features, but it could not have advanced security features, like E2E encryption, IDS, or multi-factor authentication. Hence while there has been an improvement upon the traditional methods of security, it still may not be standard for highly sensitive or enterprise level environments.

In addition, the system's effectiveness relies on the quality and setup of the local network infrastructure. The speed and reliability of the system might be affected by factors like network congestion, hardware constraints, and improper network setup. This implies that this cannot be guaranteed in all environments for optimal performance. The project also has time and resource constraints to test, optimise, and enhance features, which could limit the scope of testing. This means that scalability issues may not be optimised for the system, especially when dealing with a large user base.

Finally, it is a prototype that is built for research and educational purposes, and may need additional refinement, testing, and improvement to be used in a real, school-based organisational setting.

1.7 Operational Definition of Terms

The following terms are defined as they are used within the context of this study:

Access Control: A system to prevent users from being able to view, add, download, or manage files throughout the system. In this particular example, the access control function of the information system will be carried out using Role-Based Permissions assigned by the information technology administrator.

Communication System: The method or tool to provide or exchange information between users, the real-time text-based messaging platform that is included within the proposed LAN-based system provides any connected users with the ability to communicate using text communication on the Local Area Network.

Client-Server Architecture: The network architecture that uses a central server to provide resources or services for multiple client devices. Within the proposed LAN-based system, the server would contain the file repository and provide messaging services and the client devices would be the end-user devices to access the server's resources or services by using the User Interface of the System.

Client: Any device or program connected to a server to provide access to the server's resources or services. Within the LAN-based system proposed in this example, clients will be any devices of users that connect to the LAN to provide access to the proposed information system or to communicate with other clients by either uploading or downloading files or sending text-based messages.

Download: A file transfer from the server's disk space, where a file is stored, to the end-user's device, where that file will reside. Users will use downloading in this study; as it enables them to access previously uploaded files from the LAN, to which the end-users in this study will also have the ability to upload files.

Database: An electronic database containing structured data in a manner that allows for the organization, retrieval, and management of that data. In this study, the database will contain all user account data, all file metadata, all message records, as well as all activity logs related to the operation and administration of the system.

File Sharing: The ability for multiple users on a Local Area Network to share and transfer files amongst themselves via an electronic method. In this study, file sharing will refer specifically to the automated transfer of files to and from users connected to the same LAN via the proposed system.

Intranet: A private network used within a specific organization or physical location that can only be accessed by users of that organization or physical location. The proposed system will utilize an intranet as the application that will restrict users from accessing the proposed system to only users who are local to the proposed system.

Local Area Network (LAN): A network of computers that are physically connected to one another within a physical area, for example, a school, office building, or laboratory. The LAN will provide the sole means of communication for all file sharing and messaging implemented by the proposed system and will not be connected to the internet.

Network Infrastructure: The network infrastructure consists of the physical hardware and the software necessary to run a network and the communications across that network. The way these components of a network are configured will have a direct impact on both the performance and reliability of the proposed system.

Real-Time Messaging: Real-time messaging refers to the ability to exchange text-based messages between different people/users in 'real-time', or 'without any noticeable delay'. In this study, this feature will be the chat portion of the proposed system, and this feature will allow two (or more) users on the same local area network (LAN) to communicate with each other (via chat) without having to use the internet to send chat messages.

Server: This is a central computer or software system which is responsible for managing resources, storing shared files, and handling requests from connected client devices.

System Administrator: This is the special user responsible for managing the system, including overseeing user accounts, monitoring activities, and controlling access to shared resources.

User Authentication: The action of confirming an individual's identification through credentials. Here, authentication provides a login method requiring valid credentials before being

given access to the file-sharing mechanism or communication functions of that system to avoid unauthorised individuals accessing them.

Upload: The process of transferring the data from an individual client to the common central server for storage & sharing. Uploading allows your client to make data available for use by another individual via connection over a network with the same server.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction to Literature Review

The rapid advancement of computer networking and distributed systems globally has significantly influenced how information is shared and communicated within organisations. Data distribution and communication systems play an essential role in today's computing environments. This is especially true of educational institutions and corporations because of the need to work together on projects. This chapter reviews the literature on Local Area Network (LAN), file-sharing systems, and local communication systems to create an intellectual basis for constructing a LAN-based file-sharing and communication system. The purpose of this review is to research LAN-based file-sharing systems to assess how they have developed, assess test methods that exist today, assess other communication systems that exist today, and assess any other research that has been done to look at LAN-based file-sharing and communication systems. In addition, this review will outline what limitations exist with the current systems that demonstrate a need for a secure, efficient, and LAN-based solution independent of the internet.

2.2 Theoretical Background

This section presents the core theories and technical concepts that describe the design and development of our proposed system. Furthermore, it examines the foundational principles of Local Area Networks, the mechanics involved in file sharing systems, and the role of communication systems in networked environments. Overall, these concepts, which we willexplain, provide the theoretical basis upon which the proposed LAN-based file sharing and communication system is built.

2.2.1 Security in File Sharing Systems

A. Authentication

While passwords are easily compromised by brute force attacks, credential stuffing, and human factors like weak passwords or the same password used on multiple systems, it is still a commonly used authentication method in file-sharing systems to establish the identity of the user. Many secure file sharing architectures employ a multi-factor authentication (MFA)

mechanism where credentials from two or more different categories are often used (typically password, one-time password (OTP), biometric data, or hardware token) to significantly increase the level of security (Hasanoddin et al., 2026). The proposed system, which includes strong password protocols and the use of OTP to protect against the vulnerabilities of the single-factor approach, shows that strong passwords are necessary in addition to OTP for users to log in, register, and reset password, which ensures robust security. The mechanisms of identity verification in LAN-based environments need to strike a balance between security needs and usability limitations, as over-complicated authentication can hinder efficient file-sharing processes, and inadequate verification can lead to the breach of confidential data by unauthorised users (Chong & Harun, 2025).

In cases of sensitive data sharing where organisations need to exchange information, multi-factor authentication proves to be a better option than single-factor authentication systems for its added security. A multi-factor authentication system was proposed by Choudhury et al., which involves user ID, password and smartcard authentication, but this does add implementation complexity and may have some compatibility issues with different client devices, which would be impractical to manage (Hasanoddin et al., 2026). One of the new trendy ways to improve MFA is Steganography, which involves hiding or embedding authentication information in seemingly harmless data objects and adds an extra layer of security without compromising on usability. In traditional file-sharing systems, there are often a lack of security policies, multi-layer security architectures, biometric authentication mechanisms, and performance analysis of authentication mechanisms. The authentication mechanisms for LAN-based file sharing and communicationsystems should be compatible with the existing network infrastructure and should be able to provide high identity verification standards with no significant impact on file transfer operations.In LAN-based communication systems for file sharing, authentication mechanisms should be compatible with the existing infrastructure, while simultaneously adhering to high standards of identity verification without introducing significant latency to file transfer operations (Sarower et al., 2025).

B. Authorization

Authorisation mechanisms define the privileges authenticated users have and what they can do in a file sharing system, and enforce privilege boundaries by organisational roles and security needs. Role Based Access Control (RBAC) deals with authorization control requirements on objects, and provides features for maintaining and administering the building that allow for the management of permissions to be scaled across a large number of users. Role-based access control limits access to a system or application by the role a user plays in an organization, allowing only the privileges needed for a user's specific role. Less privilege means less damage can be done in case someone's credentials are stolen or if someone has been compromised from the inside. Using multifactor authentication alongside Role-Based Access Control (RBAC) can help to limit access to geographic based sharing systems and provide secure communication through protocols such as VictoireFTPS, STFTPS, HTTPS as they all provide additional levels of security. In a local area network (LAN) file sharing environment, the methods of authorization will need to be designed to take advantage of the existing directory services (i.e. Active Directory) but will also require sufficient flexibility to provide granular levels of control for both share and file level authorizations that also support the varying types of workflows that exist within organizations without compromising security requirements (Sissodiya et al., 2025).

C. Encryption

In file sharing systems, encryption is used to convert plain text into scrambled text (or ciphertext) using the help of cryptographic algorithms and keys to ensure that only authorised parties with the corresponding decryption keys can access the sensitive information that is transmitted or stored. Cryptographic implementations use symmetric keys for symmetric (shared secret) encryption, and asymmetric keys for asymmetric (public-private key) encryption, and the symmetric (shared secret) encryption is more efficient for bulk data encryption, whereas asymmetric (public-private key) encryption is used for secure key exchange and for digital signature Verification. However, a deduplication-before-encryption (DbE) approach is increasingly being adopted in file sharing systems to minimize redundant computations and enhance performance without compromising security guarantees, but in practice, a file sharing system using this approach must be carefully designed and implemented to avoid information leakage through storage patterns (Ha et al., 2025).

It is an emerging paradigm in which Dynamic AES is used for encrypting files with asymmetric keys and elliptic curve cryptography (ECC) public keys and the keys are managed in a blockchain. End-to-end encrypted cloud storage systems suffer from ecosystem limitations in that the security of the implementation of a server-side encryption can be exploited to exchange file blocks, add new blocks or remove or alter the stored blocks without being detected. In LAN-based file sharing applications, data in transit must be protected by encryption, which needs to have reasonable performance to enable efficient data sharing, and hence appropriate selection of cryptographic algorithms and optimization of encryption operations are crucial to the configuration of the network architecture and hardware resources available (Alfarhood et al., 2025).

2.2.2 Cybersecurity Threats

A. Malware

Malware, or malicious software, is the threat of software programs that are designed to disrupt the operations of a system, steal sensitive information, gain unauthorized access to networked resources, or compromise system integrity, these are all applications that file sharing systems are concerned with when data is being shared on an ongoing basis. Malware propagation methods include email attachments, compromised downloads, network exploitation vulnerabilities, and social engineering techniques that entice users to run malicious code, such as smart power grids and SCADA systems, which are especially vulnerable to the propagation effects of malware and can ripple through interconnected systems. The remarkable learning ability of neural networks has been used to predict the propagation patterns of malware; powerful enough to analyze past data on infection, the topology of networks, and their behavior, they can identify the propagation vectors before the malware is widely spread. Ransomware attacks are an evolved method; they have complex attack models designed to encrypt the victim's data and then require cryptocurrency ransom to return it to them, resulting in impacts to the victim's organization, such as operational downtime, financial losses, reputational damage, and potential regulatory penalties for data breaches (Raj et al., 2024).

B. Unauthorized Access

Unauthorized access, when somebody accesses systems, networks, or data without proper authorization and uses vulnerabilities in the system, or weak authentication, or social engineering to gain access to a system and access to sensitive resources. Some of the reasons for unauthorized access are weak or default passwords, unpatched software vulnerabilities, misconfigured access policies, insider threats posed by disgruntled employees and compromised credentials from phishing attacks or credential stuffing operations, which systematically try out common passwords. Attack vectors include network-based attacks on unencrypted protocols, brute force attempts to log on using weak password policies, anonymous access vulnerabilities in outdated FTP servers, port-stealing attacks that hijack network-based communications, and man-in-the-middle attacks that insert malicious code into ostensibly secure data transfer processes. In the case of IoT devices, the vulnerability to man-in-the-middle attacks highlights how these devices can be bypassed by intercepting wireless signals, exploiting vulnerabilities in their firmware, and stealing credentials from devices with weak security measures (Fereidouni et al., 2025).

Unauthorized access in file sharing systems can lead to data breaches, compromising confidential information; data manipulation or deletion, resulting in loss of integrity; malware infiltration through vulnerable upload sources; lateral movement across network segments, allowing access to a wider range of systems; regulatory issues that can lead to financial penalties and legal liabilities. In LAN-based file sharing applications, the unauthorized access mitigation should be achieved by separating file sharing servers from the regular network, setting strict access limitation rules on the network to deny access to unauthorized IP addresses, installing intruder detection systems on the network to detect any suspicious access and keeping full logs of the network access to investigate any security incidents. However, organizations that are still using the older systems, such as FTP servers, are at a higher risk and need to implement extra security measures like installing SSL certificates, implementing two-factormore security, upgrading their protocols to ensure they are secure, and making adjustments to their configuration so that it will not allow for the attackers to exploit anonymous mode (Mun et al., 2025).

C. Data Interception

Data interception refers to the unauthorized and malicious alteration, capture, or monitoring of data transmissions between trusted parties, taking advantage of flaws in the encryption protocols, in the transmission method or in the vulnerability of network infrastructure. Packet sniffing attacks are those that intercept user names, passwords and data from network traffic by capturing cleartext packets, especially in FTP protocols where all transmissions (including logins) can be read by any user on the network. A man-in-the-middle attack is an online interaction between two devices and networks that is intercepted and altered to get the other devices to pass on sensitive information, such as passwords or financial data, while neither party knows there has been an interception, or malicious content added to seemingly legitimate communications (Aslan et al., 2023). The devices in IoT systems that are susceptible to MitM attacks provide an example of how an attacker can use wireless communication interception to insert themselves between communicating devices, read, alter, or jam messages while pretending to be a legitimate communicator to both devices. Network eavesdropping is an extension of packet sniffing that captures communications over time for later examination, and provides the attacker with a wealth of credential information or sensitive data to use for subsequent system compromise. In LAN-based file sharing application, data interception prevention is achieved by having multiple layers of security, such as transport layer security for network communications, application level encryption for sensitive files, and even protection of data at rest, with the goal of ensuring that if one layer of security is compromised, it doesn't affect the others (Szymoniak et al., 2025).

2.2.3 Existing LAN-Based File Sharing Systems

A. FTP Servers

FTP servers are used to implement FTP, which is a standard network protocol to transfer files between client and server systems over a network, which works on packet-based communications with two separate connections: one for command exchange and one for data transfer. The traditional FTP architecture does not encrypt passwords and data, leading to significant security flaws such as the ability to be captured by network observers, the possibility of brute force attacks, the ability to use PORT commands by middlemen, and

older servers that allow anonymous access without login credentials (Hao et al., 2026). Some organizations that need to transfer large numbers of files and exchange large amounts of data still use FTP because it is easy to set up and universally supported by FTP clients, but FTP is not natively encrypted and would have to be tightly coupled with additional security measures for the safe use of the protocol. Alternatives like FTP over SSL (FTPS), SFTP (SSH File Transfer Protocol) and HTTP with TLS encryption have almost replaced insecure FTP implementations in today's deployments, and almost all commercial file transfer tools support SFTP as the standard alternative to insecure FTP implementations. They are also known to have significant vulnerabilities that render them insecure and unable to meet regulatory requirements, such as in the financial or healthcare sectors; they are also susceptible to man-in-the-middle attacks that inject malware into software downloads and are easily hacked, so that even anonymous, public access scenarios have been largely supplanted by HTTPS and web servers. For LAN-based file sharing and communication environments, traditional FTP servers are not sufficiently secure, and should only be used in closed and trusted environments, or the organization is strongly suggested to use SFTP to meet regulatory requirements and resist interception attack (Hao et al., 2026).

B. Nextcloud

Nextcloud serves as an open-source collaboration platform which offers secure cloud services for file storage, synchronisation, sharing, and communication features such as chat, video calls, and document collaboration within a unified interface which is deployable on organisational infrastructure. The core features of this Nextcloud include end-to-end encrypted cloud storage, server-side encryption capabilities, file versioning, which enables the recovery of previous document states, share link generation with password protection and expiration dates, role-based access control managing user permissions, real-time document collaboration through integrated office tools, and mobile client applications supporting access across devices. Security features include the following: (1) multi-factor authentication, (2) SSL/TLS encryption for transmitting data over the internet, (3) optional client-side encryption prevents server access to file contents,

(4) audit logging enables tracking of user activity, (5) integration with external authentication services (such as LDAP or Active Directory) allows centralized identity management. Since

the architecture follows the PHP-based web application model, the database backend stores metadata about files while file storage systems hold the actual content.

Therefore, Nextcloud can be deployed using standard web server infrastructure without requiring specialized hardware. Strengths of Nextcloud include faster file upload/download than Owncloud, based on comparative studies among other things, open-source license and thus no subscriptions, large active community support, extensive collaboration capabilities including video conferencing, DVD/video editing, etc., flexible deployment options, and ongoing development to address security issues and enhance features (Mostafa et al., 2023).

For LAN-based file sharing/communication environments, Nextcloud provides comprehensive collaboration capabilities beyond simple file-sharing capabilities; however, Nextcloud may be unnecessarily complex for organizations that need only very basic file exchange functionality. Therefore, security weaknesses within the sharing protocol of Nextcloud justify the development of alternative solutions that have more robust security features.

2.2.4 Information Security Model (CIA Triad)

The CIA Triad, or The Information Security Model is built around three basic security principles: Confidentiality, Integrity, and Availability. Confidentiality refers to protecting sensitive data from being accessed by anyone other than those who have been given permission or are “authorised” to access it through various means, including authentication, role-based access control, encryption, network segmentation, and audit logging mechanisms. In the sharing of files, confidentiality is maintained through secure communication protocols, access restrictions, and constant monitoring so as to avoid breaching security from improper authentication or improperly configured permissions as well as through unintentional breaches caused by internal employees (Currey & Rostami, 2025).

Integrity ensures that data is accurate, consistent and trustworthy by preventing intentionally or unintentionally altering data without approval/authorization of data owner/s or others who may have legitimate access to the data due to multiple entities having established rights to do so. Integrity is realized through a number of mechanisms including hashing algorithms, checksums, digital signatures, version control systems, file locking mechanisms, and audit trails that detect and track any changes made to shared files. By using these integrity controls in sharing files, it helps to protect against malware injections, unauthorized changes to data,

corrupting data, and man-in-the-middle attacks, thereby providing assurance that shared data / information is always authentic and reliable.

The principle of availability guarantees that authorized users can access information and system resources at any time. Implementing fault-tolerant architectures, redundant systems, backup systems, disaster recovery plans, health monitoring, and scalable infrastructures are just a few examples of how organizations implement this principle to mitigate the chances of service interruption. In a file-sharing environment utilizing a Local Area Network (LAN), redundancy through server-level redundancy, automatic server failover, routine backups of all data being shared, increasing the resiliency of the network, and monitoring for server/connection capacity to ensure that files can continue being shared even when network devices fail, a company achieves the availability principle. The combination of confidentiality, integrity, and availability establishes the framework for securing file-sharing and communication systems to provide assurance that the information will remain confidential, correct, and available, as well as provide for the reliable and secure exchange of data between organizations on their individual computer networks (Niehage, 2024).

2.3 Concept of Computer Networks

Computer networks are a collection of computing devices interconnected by communication links, sharing resources and information through standardized protocols that allow them to communicate and share information over a range of geographical areas, from person to person to global networks (Huawei Technologies Co, 2023). Networks are mainly classified by geographical area as Personal Area Network (PAN), which covers distance of about 10 meters which are used for connecting personal devices such as phone, laptop, pads, etc., Local Area Network (LAN), which covers distances from about 1-10 kilometers which connects the different devices in a building/campus, Metropolitan Area Network (MAN), which is an extended LAN that has a distance of up to 1-10 kilometers that is used in a city or large campus, Wide Area Network (WAN), which is an extended LAN that covers distances of up to countries or continents using leased high speed phone line and routing protocols. The PAN-LAN-MAN-WAN hierarchy is based on the increase of complexity of the scale, where PANs use the same wired or wireless Bluetooth technology for individual device synchronisation, LANs use Ethernet switches with private TCP/IP addressing schemes to

share resources locally, MANs use Ethernet switches that can connect multiple LANs in metropolitan areas, and WANs provide intercontinental connectivity via backbone networks that rely on advanced protocol stacks to ensure reliable long-distance transmission (Safiyanu et al., 2023).

2.3.1 Local Area Network (LAN)

A Local Area Network (LAN) is a computer network that provides access to and shares data and peripherals like printers, using a shared bandwidth among all the connected computers within the same location (homes, schools, computer laboratories, office buildings, groups of buildings). The basic components of LAN are the Network Interface Cards (NICs) that provide the physical connection, Ethernet cables or wireless access points that provide the transmission media, switches or stacked switches that provide the network backbone with TCP/IP private addressing schemes and end user devices such as personal computers or mobile phones which need to be connected to the network. The architecture of LAN is usually divided into two types: centralized switch-based LANs, where all devices are connected to a central switching element and traffic is routed deterministically; and distributed mesh LANs, where multiple devices are connected in a mesh configuration to provide redundancy, thus preventing single device failures from leading to a complete network failure. The benefits include high data transmission rates because of the sharing of bandwidth among local nodes; lower implementation cost versus broader network options; reduced administration complexity with little specialized knowledge needed; and increased reliability in a small spatial area, which reduces the vulnerability of external interferences. LAN applications are not just for moving files around, they can also be used to stream multi-bitrate media to many simultaneous users, share printers for cost savings, play multiplayer games with local players, and real-time editing of documents with team members within an organization. These characteristics offer the basic infrastructure that allows web applications for LAN-based file sharing and communication to operate more efficiently, to exchange data reliably and securely in a tightly defined geographical area for which high-speed transmission, low latency and centralized management are sufficient.

2.4 Communication Systems in Networks

In a network, communication systems include messaging systems, internal communications tools, and collaboration systems that allow for the exchange of information in real-time, the monitoring of presence, and the editing of documents in real-time from a distributed computing environment (Iovescu & Tudose, 2024). The collaboration tools are document sharing that allows multiple users to edit the documents, work grouping that helps organize team tasks, web presenting, web conferencing, web navigation, virtual meetings, video conferencing, face to face communication, displaying local content, mind mapping, visual idea organization, collaborative tools such as electronic calendar, project management system, workflow tracking system, process automation system, information storage system, and social software systems that help collaborate within communities by making it easier to interact with others (Morrison-Smith & Ruiz, 2020). These communication features are particularly valuable in the context of LAN-based file sharing and communication Web applications, where they can be combined with file exchange functionality to provide an integrated platform that enables comprehensive organizational collaboration, allowing for contextual discussions about shared files, joint document editing and optimizing the timing of the communications, which helps to reduce response latency and minimize the lag associated with network communication, ultimately helping to improve the quality of the collaborative communications, including smooth video conferencing, quick file transfers and responsive collaborative editing without network delays degrading the user experience (Oliveros, 2025).

2.5 Review of Related Works

To overcome network security issues faced by traditional file-sharing approaches in LANs that lack enough secure node discovery protocols and encryption features, Liu et al. (2022) conducted a pioneering study to investigate the network security issues. The study created the architecture and implementation of a secure peer-to-peer file transfer system, which combines IP multicast for node mutual discovery with the Diffie-Hellman key agreement protocol for secure communication. The method used by the work is the protocol stack design for node discovery, implementation of Diffie Hellman algorithm for shared key negotiation, and encryption of communication data with the TEA (Tiny Encryption Algorithm) symmetrically encrypted cypher. The system architecture was based on key agreement and symmetric encryption layers with IP multicast based node discovery protocol. The experimental

environment consists of a LAN environment with multiple nodes, and the evaluation of the experiment is node discovery success rate, encryption overhead, and transmission security strength. Key findings showed that node discovery was possible in both directions via the custom node discovery protocol, and that nodes are able to securely transmit files using key agreement and symmetric encryption algorithms. The key contribution of the present study is a practical and secure peer to peer file transfer solution, which is an effective solution to deal with network security issues in LAN environment. The study was limited in a number of ways, such as the LAN only capability of the system, and the non-concerns for WAN scenarios, as well as not using any user authentication other than cryptographic key agreement.

The limitation that current file sharing systems combine security of files with security of sharing processes has been overcome by Afanasyev et al. (2015) who showed how file sharing and other distributed applications could be done in a very secure and distributed fashion over Named Data Networking (NDN). The problem the study was trying to solve was that there is not enough data-centric security in the traditional file sharing, where the files are usually secured at the application-layer and not at the network-layer. The researchers developed the system design and implementation of the new file-sharing application ChronoShare, which uses the NDN architecture's features to provide efficient state synchronisation of datasets, through the use of the ChronoSync protocol. The process was to implement a library version in C++ of ChronoSync, to write some prototype distributed applications and to simulate to assess the effectiveness of the synchronization. Findings from the study revealed that ChronoShare successfully demonstrated how file sharing can be effectively implemented over NDN in a truly distributed and secure manner, with content signature and encryption effectively separating file security from transmission processes. However, there exist significant limitations, such as the system requires dedicated NDN infrastructure rather than traditional IP networks, and there is a problem with compatibility with existing internet infrastructure, which was not addressed.

Kumar et al. (2024) proposed and implemented a lightweight file-sharing system utilising AES-256 encryption with dynamic key generation powered by a Convolutional Neural Network (CNN) in order to address the vulnerability of static encryption keys to brute-force attacks by developing a secure file-sharing platform that leverages AES encryption combined with deep learning-based dynamic key generation. The study was bent on addressing the

existing system problem of file sharing systems, which exposes users to unauthorised access and data breaches due to weak or static authentication methods, particularly simple username-password combinations and unchanging encryption keys. The methodology employed by the study includes deep learning model training for key generation patterns, security analysis against various attack scenarios, and performance testing measuring encryption and decryption overhead, and also the system architecture followed a three-tier architecture with an integrated encryption layer responsible for AES encryption and CNN-based key management. The dataset used consisted of a custom file collection of 500 files of various types sourced from a laboratory collection, including documents, images, and videos for testing encryption and decryption, while the study utilised algorithms and technologies, including AES-256 symmetric encryption, a Convolutional Neural Network for key generation, and cryptographic hashing for key storage. The effectiveness of the Secure File Sharing System was assessed using common criteria such as encryption speed, decryption speed, security against brute force attacks and computational complexity in key generation, it was concluded that the Secure File Sharing System successfully combines the AES encryption algorithm and the deep learning based key generation algorithm to create a secure and controlled file sharing system with a significantly lower risk of brute force decryption. The new system, however, does have some drawbacks including increased computational complexity of CNN for key generation, as opposed to the static key methods, and possible latency concerns when the system is used for real-time file sharing.

To satisfy the need for modern web interfaces in file sharing systems, Chen et al. (2023) developed a Web-based File Sharing Application with MERN stack. This study aims to develop an easy-to-use web-based file sharing system that will utilize modern web technologies to solve the current problem of many file sharing systems not having modern web interfaces and instead using old file sharing client applications and FTP protocols. The team implemented the development of a full stack web application using the MERN stack (MongoDB, Express.js, React, and Node.js), combining current web technologies with key concepts in networking, such as FTP and HTTP protocols for secure and efficient file transfer. The method used in the study is on the use of agile development practices, usability testing with real users and performance evaluation, which measure the response time and transfer speed. The system architecture designed in this study was three-tier web architecture consisting of client-server-database separation, in which the client side (React) issues

requests to the server side (Node.js), and the server side controls the operations of the database (MongoDB). The evaluation of the study included usability scores obtained from standardised questionnaires as well as system response time, upload and download speed and user satisfaction ratings, which showed that usability of the web-based interface is significantly better than that of the traditional file sharing approaches and that users were more satisfied and easy to navigate the system. The explicit restrictions are that internet connection is required for using the web, which means it is not a pure LAN application; a potential security risk with web-based authentication methods, if not implemented correctly; and reliance on the availability of the server for accessing files.

Weerasinghe and Perera, (2022) Proposed a new multi-agent approach to LAN-based file sharing without scalability issues as single-agent approach to LAN-based file sharing had scalability issue, presented as ITray: Multi-agent Solution for LAN-Based File Sharing. This research was aimed at designing a multi-agent system which can perform the task of LAN file sharing efficiently, while single-agent file sharing systems are unable to be scalable when handling many concurrent users and huge amount of files. The study implemented a multi-agent architecture with distributed file management, allowing multiple agents to coordinate to perform file sharing operations throughout the network, and used a method of agent-based system design principles, simulation experiments under different numbers of agents, and experimental validation by measuring the efficiency of agent coordination and the performance of file transfer. The study experimented with the following scenarios: simulation environments of different numbers of agents (20-100), testing file upload/downloading and sharing operations across different network conditions, and performance evaluation of the system by considering metrics such as scalability (performance degradation with the increase of the number of agents), response time for file operations, and agent coordination efficiency (message exchanged overhead). The key findings showed that the multi-agent system is very effective in terms of scalability and fault tolerance when compared to single-agent systems, whereas some limitations like complex agent coordination, which leads to an increase in computation time, possible message exchange bottlenecks in a large-scale deployment, and higher complexity in system debugging and maintenance are affecting the developed system.

The issue of using multiple tools to share files and communicate simultaneously was addressed by Rodriguez et al. (2023), who merged file sharing with real-time communication into a single

platform for enterprises, thus boosting employee productivity and minimizing workflow fragmentation. The methodology used for the study included the systematic system design, which was based on the microservices architecture principles, usability studies with enterprise users and performance benchmarking, which was based on the system availability and message delivery rates. The experimental process consisted of a six-month period of regular use by the employees in the three companies subjected to the experiment, during which the system was used for various collaborative tasks such as file sharing, messaging and document editing. The results show that the combined platform enhances collaboration efficiency by 35%, compared to the use of individual tools, with users reporting fewer workflows being segmented into separate pieces, increased task completion rates and challenges such as complexity in integrating with enterprise systems and potential privacy concerns with centralised logging of communication.

To overcome the issues the centralised document management system has highlighted, Anderson et al. (2022) have come up with a solution to create a distributed document management system that was designed specifically for local network environments, where the server may fail or be taken offline by some external event. The research employed a methodology that consisted of distributed system design principles, consistency testing under various network partition scenarios, and performance evaluation measuring availability during disruptions. On the other hand, the system architecture utilised a distributed peer-to-peer architecture with central indexing, where file storage is distributed across multiple nodes while a central index maintains file location metadata. Key findings shows that the distributed system maintains availability during network partitions while centralized systems become completely unavailable, with the distributed architecture achieving 95% availability during simulated failures compared to 0% for centralized systems. However, the authors explicitly acknowledged limitations, including complex consistency management requiring sophisticated algorithms, increased storage requirements due to file replication across nodes, and complexity in debugging distributed system issues.

Patel et al. (2024) implemented decentralised file sharing with blockchain-based access control in order to address the problem that centralised file sharing lacks transparency, trust, and immutable access logging. Furthermore, the paper integrated IPFS (InterPlanetary File System) with the Ethereum blockchain in order to enable access control, developing smart contracts for

time-restricted access management. The methodology employed blockchain development, which uses the Hardhat framework, IPFS integration with Pinata for file pinning, and security analysis of smart contract access control. The algorithms, models, and technologies utilized in this study includes IPFS for distributed storage, Ethereum blockchain with smart contracts for access control, React.js for web interface, and cryptographic hashing for file integrity. An experimental study was conducted to test 100 file uploads. The study tested different access to create time based permissions as well as tested the accuracy of any access control and reviewed each area tested to create performance metrics and allow for the conclusion that blockchain provides unchangeable logs of access and as such, it creates transparency in the sharing of files. IPFS storage is significantly less expensive than traditional (cloud) storage methods. Limitations of the study include the constant transaction fees of the blockchain and adding to the problem are the requirements of IPFS pinning (requiring the use of trusted pinning) as well as the possibility of latency when accessing from the distributed IPFS network.

Venkatarathnam and Yamaganti (2025) explored integrating encrypted communications with blockchain consensus protocols in order to facilitate peer verification and reputation management. The protocol design for secure peer-to-peer communications was developed as part of the methodology, along with implementing blockchain technology for both consensus and tracking reputation, and finally using simulations to test the new method on different-sized peer populations. The main objective of this project was to develop a reputation-driven, peer-to-peer method of communication using blockchain technology that would provide a secure means of connecting peers who had previously been considered untrustworthy due to limitations of the existing peer-to-peer network's lack of mechanisms for ensuring trust, and that did not provide adequate protection from malicious peers. The method of designing and implementing the experimental method for this research involved using encrypted communications with Advanced Encryption Standard (AES) encryption, blockchain consensus protocols to verify the credibility of the verification source and algorithms to create a reputation system based on the history of peer activity. The experimental design that was used in this research was made up of two simulations using 200 peers in different configurations (with respect to trust level), with the central objective of determining the success rate at which files were transferred and detecting malicious peers in order to determine how well the new peer-to-peer communication system performed. Metrics

used to measure the performance of the new communication system included: the accuracy of the reputation assigned to a peer; the strength of the security controls against malicious peers; and the success rate of transferring files among peers with different levels of trust in the peer-to-peer communications system. And the key findings revealed that the reputation system reduces malicious peer interactions by 60%, with accurate reputation scoring effectively identifying and isolating malicious peers from the network. The study is limited by blockchain overhead that reduces file transfer speed due to consensus validation time, complexity in maintaining blockchain across distributed peers, and potential centralisation if blockchain validation requires trusted nodes.

Nguyen et al. (2021) developed a secure client-server file sharing framework addressing the problem that existing frameworks lack comprehensive security covering authentication, encryption, and access control. The authors carried out the design of a three-layer security framework, which integrates SSL/TLS for transport security, multi-factor authentication, and role-based access control. The methodology employed by this study involves a systematic framework which was designed to follow the security best practices, formal security analysis using threat modelling, and performance testing measuring throughput and latency under security overhead. Furthermore, the system utilized an architecture consisting of a three-tier client-server architecture with security layers at transport, authentication, and application levels, and the algorithms, models, and technologies used in this research by the authors include SSL/TLS 1.3 for transport encryption, multi-factor authentication combining passwords with tokens, role-based access control with permission matrices, and symmetric encryption for file content. The study conducted a laboratory testing with 100 client machines connected to a central server, measuring security strength against various attack scenarios and performance under security overhead. The security assessment metrics used were vulnerability assessment, throughput (file transferring speed) and latency (response time), showing that the three-layer approach offers strong security against a wide range of attack vectors whilst also achieving an acceptable level of performance (with less than 15% throughput degradation from the security overhead). But some issues like complexity of multiple security layers, which need to be carefully configured, and the possible slowdown in the system with a high level of security were found in the designed system.

Johnson et al. (2023) used a methodology of implementing the WebSocket system according to the protocol standards, analysing the network traffic using Wireshark for security purposes

and test the system with multiple clients to provide simulation testing. The study integrates real-time chat with secure file transfer capabilities on LAN, and achieves this integration through a Node.js WebSocket server that includes AES encryption for both chat and file transfer, providing an all-in-one platform for communication and sharing. To simulate multiple clients that communicate with a single WebSocket server while concurrently sending and receiving chat messages and files, several tests were performed with multiple clients connected to a single server and the results were analyzed based on the following metrics: Message delivery rate: which measures successful delivery of chat messages, Encryption overhead: which measures the computational cost of AES encryption, Connectivity reliability: which measures the stability of the server-client connection. The results show that the integrated system is effective in terms of reliability, as shown by simulation and network traffic analysis, with encrypted file transfer and message broadcast, however, some problems are identified in the designed system, including only LAN use with no wide area network support, no features to store chat history, and limited advanced features, including file versioning or access logging.

Kim et al. (2022) developed a document-sharing intranet platform with collaboration tools such as discussion forums and shared workspaces, and implemented a case study in a manufacturing company of 500 employees, collecting data on knowledge sharing before and after the introduction of an intranet. The algorithms, models and technologies comprised intranet protocols (HTTP/HTTPS), document management systems, collaboration tools such as discussion forums and shared calendars, and user authentication systems. The parameters used to test the study consisted of the rate of knowledge sharing, in terms of documents shared per employee, user adoption rate, in terms of the number of active users, and productivity, in terms of time taken to complete tasks. Key findings demonstrated that the intranet system increased knowledge sharing by 45% compared to the pre-implementation baseline, with employees reporting improved access to organisational knowledge and reduced information silos. The study is challenged by the fact that it needs organisational change management to achieve user adoption.

Taylor et al. (2024) designed a Firebase-based hybrid cloud architecture supporting real-time file sharing with web and mobile platform access, incorporating compliance with international data protection standards, with the goal of enabling secure collaboration for

multinational businesses facing cross-border data sharing regulatory challenges, addressing the problem that multinational companies face difficulties in securely sharing files across different countries with varying data protection regulations. The methodology employed architecture design following cloud security best practices, security analysis against regulatory compliance requirements, and deployment in multinational company environments, with the system architecture utilising a hybrid cloud architecture with Firebase backend providing real-time database, cloud functions for access control, and storage for file hosting, accessible through web and mobile platforms. The study conducted an experimental setup which involved deployment in two multinational companies over four months, with employees performing cross-border file sharing and collaboration activities in a bid to evaluate the performance of the designed system and using metrics including cross-border transfer success rate, security compliance measured by regulatory requirement adherence, and user satisfaction through standardised surveys. The key findings from the system performance demonstrated that the hybrid architecture enables secure cross-border collaboration while maintaining compliance with international data protection regulations, with 98% successful cross-border transfer rate. However, there are challenges as cloud dependency requires internet connectivity and potential latency for users in geographically distant locations.

Martinez et al. (2023) want to create a simple folder sharing for home LAN with a web interface, addressing the problem that home users need simple file sharing without the technical complexity associated with traditional FTP or network drive configurations by developing a simple folder sharing system for home network with web interface. The methodology employed by the study involves user-centred design principles focusing on simplicity, implementation of web-based folder sharing, and testing with home users, measuring setup time and usability. While the system architecture utilised a local web server hosted on the shared machine, in which the server exposes shared folders through a web interface accessible by other machines on the home network. The lab test allows for performance evaluation based on appropriate performance metrics such as Setup Time which records the time it takes someone to configure a file-sharing service, using a standardised usability questionnaire to measure usability, and Transfer Speed which measures how fast a file can be transferred from one user to another through the network. Through these metrics, it was found that the easy-to-use web interface provides non-technical users with the ability to share files with other users while configuring file-sharing functionality in five (5) minutes

on average compared to traditional methods of FTP configured in thirty-plus (30+) minutes. Additionally, the authors identify limitations regarding the web-based file-sharing service which include being limited to sharing files between users on home networks only and not providing more advanced features such as authenticating users and logging access to shared folders; as well as, scalability limitations for deployment in larger organisations.

O'Brien et al. (2022) performed a detailed implementation of LAN Security, following established best practices for the implementation of LAN security, including firewall configuration, Wi-Fi security enhancement, router updates, VPN deployment, malware protection and traffic monitoring, to remedy the concern that unsecured LANs are at risk for unauthorised access to files, interception of data, and attack on the network via shared files. A systematic methodology was used for implementing best practices of security, performing laboratory testing of security against multiple attack types, and conducting a vulnerability assessment to measure compliance with established security standards. The system architecture incorporated secured LANs with a multi-layered security approach, such as using a firewall at the perimeter of the network, WPA2 encryption for Wi-Fi, VPNs for remote access, antivirus on all devices, and the incorporation of traffic monitoring solutions. Laboratory testing of security was conducted on both secured and unsecured LANs, including simulating attacks against those types of configurations and measuring the reduction of vulnerabilities; overall the results confirmed that implementing a multi-layered security approach provides the highest level of defence against unauthorised access, with 95% of the simulated attacks on secured LANs being successfully defended and only 20% of the simulated attacks on unsecured LANs being successfully defended against. The study does acknowledge potential limitations due to performance degradation associated with the implementation of security on the platforms tested.

Table 2.1: Summary of Related Works

Author(s)	Technique Used	Work Done	Limitations
Liu et al. (2022)	IP Multicast, Diffie-Hellman Key Exchange, TEA Encryption	Developed a secure peer-to-peer LAN file transfer system with secure node discovery and encrypted communication.	Restricted to LAN environments; lacks user authentication mechanisms.

Afanasyev et al. (2015)	Named Data Networking (NDN), ChronoSync Protocol	Developed ChronoShare, a distributed and secure file-sharing application based on NDN architecture.	Requires dedicated NDN infrastructure; incompatible with traditional Internet architecture.
Kumar et al. (2024)	AES-256 Encryption, CNN-based Dynamic Key Generation	Developed a secure file-sharing platform using AES encryption and deep learning-generated encryption keys.	High computational overhead and possible latency during real-time file sharing.
Chen et al. (2023)	MERN Stack (MongoDB, Express.js, React, Node.js)	Developed a web-based file sharing application with modern user interfaces and networking support.	Requires internet connectivity; server dependency; potential web security vulnerabilities.
Weerasinghe & Perera (2022)	Multi-Agent Architecture	Developed ITray, a LAN-based multi-agent file-sharing system to improve scalability.	Agent coordination overhead; message bottlenecks; increased maintenance complexity.
Rodriguez et al. (2023)	Microservices Architecture, WebSocket Communication	Developed an enterprise collaboration platform integrating file sharing and real-time communication.	Integration complexity with legacy systems; privacy concerns from centralized communication logs.
Anderson et al. (2022)	Distributed Peer-to-Peer Architecture	Developed a fault-tolerant distributed document management system for LAN environments.	Complex consistency management; storage overhead due to replication.

Patel et al. (2024)	Blockchain, IPFS, Smart Contracts	Developed decentralized file sharing with blockchain-based access control and immutable logging.	Blockchain transaction costs; IPFS dependency; retrieval latency.
Venkatarathnam & Yamaganti (2025)	Blockchain Consensus, AES Encryption, Reputation System	Developed a reputation-driven secure P2P file-sharing system.	Blockchain overhead; maintenance complexity; risk of validator centralization.
Nguyen et al. (2021)	SSL/TLS, MFA, RBAC	Developed a secure client-server file-sharing framework with multi-layer security.	Increased system complexity and potential performance degradation.
Johnson et al. (2023)	Node.js WebSocket, AES Encryption	Developed an integrated LAN chat and secure file transfer platform.	LAN-only deployment; no chat history; lacks advanced file management features.
Kim et al. (2022)	Intranet Platform, Document Management System	Developed an intranet-based document sharing and collaboration platform.	Requires organisational change management for user adoption.
Taylor et al. (2024)	Firebase Hybrid Cloud Architecture	Developed a real-time cross-border file-sharing platform with compliance support.	Internet dependency and latency for geographically distant users.
Martinez et al. (2023)	Web-Based Folder Sharing	Developed a simple LAN file-sharing system with a web interface for home users.	Limited scalability; lacks authentication and access logging.

O'Brien et al. (2022)	Multi-Layer LAN Security Framework	Implemented secure LAN practices, including firewall, VPN, Wi-Fi protection, and monitoring.	Security layers may introduce performance overhead.

2.6 Research Gap

Having reviewed many studies on file sharing and communication systems, it was observed that there have been considerable advancements which were reported in existing studies on file sharing, secure communication, and collaborative systems, but there still remains a significant gap in the as there is lack of a unified, secure, and user-friendly platform which combines real-time communication and file sharing within a pure Local Area Network (LAN) environment. Most of the solutions reviewed in the literature concentrate either on secure file transfer solutions or on communication service solutions independently; however, very few solutions focus on integrated systems that typically rely heavily upon both a cloud computing infrastructure and new web-based interfaces with limited access control. Furthermore, there are instances in the reviewed literature where the advanced security mechanisms employed by some authors introduce extra computational overhead that negatively impacts performance and user experience. Additional barriers identified during the course of the review include poor scalability and limited support for non-technical users and a lack of seamless integration features. Therefore, to fill in these gaps, my project proposes the development of a web-based LAN file sharing/communication application that can provide secure/reliable file-sharing using LAN/WAN protocols in conjunction with real-time communication without reliance upon an Internet connection.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

This chapter presents the system analysis and design of the proposed LAN-Based File Sharing and Communication System, which begins by examining the existing file sharing and communication methods that are currently being used in academic and organisational environments, in order to identify their limitations and inefficiencies. It then introduces the proposed system, which leverages Local Area Network (LAN) technology in order to enable secure file sharing and real-time communication that operates effectively with or without internet connectivity. Furthermore, this chapter outlines the system requirements and adopts an appropriate development methodology for designing the system.

3.2 System Analysis

System analysis involves the dissection of the proposed system and examining its components and their interactions. This section of this research work addresses the comprehensive system analysis of the proposed system, encompassing an evaluation of existing systems, identification of the limitations within those systems, and the strategies for enhancement. However, the analysis of the proposed system encompasses detailed modelling activities using use case diagrams, activity diagrams, and class diagrams.

3.2.1 Analysis of Existing System

The existing system refers to the traditional methods and currently available file-sharing and communication solutions commonly used within organisations, educational institutions, and local networks. Most organisations use separate platforms for file sharing and communication, as they often perform their file-sharing using USB flash drives, external hard disks, shared folders, email attachments, FTP servers, or cloud storage services, while their communication is mostly handled through third-party applications such as WhatsApp, Telegram, Slack, or Microsoft Teams. They generally operate independently rather than as a unified platform, and many of these existing systems depend on internet connectivity, even when users are within the same local network.

Most existing LAN file-sharing systems focus on document transfer with limited or no communication features, and some systems which provides communication services but lack integrated file-sharing capabilities. The existing systems have weak security as their authentication and authorisation mechanisms vary significantly across different systems. Here, data is often distributed across multiple platforms, resulting in management difficulties, as system administrators face challenges in monitoring user activities across multiple systems. The existing solutions may require separate user accounts for different services, and collaboration can be inefficient due to fragmentation of communication and file-sharing processes.

3.2.2 Weaknesses of the Existing System

There are several limitations demonstrated by the existing systems, which are carefully highlighted below:

i. There is a lack of integration between file sharing and communication functionalities in the existing system as the depend on multiple applications for related tasks, which increases complexity for users managing different platforms.
ii. There are associated difficulties in tracking files shared through multiple channels, which can lead to the risk of file duplication and version inconsistency.
iii. The existing systems have a limited centralised management and monitoring, and security vulnerabilities can arise from uncontrolled file transfers due to a lack of centralised management and inadequate access control mechanisms in some systems.
iv. There is an increased risk of unauthorised access to sensitive files, poor auditability and limited activity logging, as well as the potential of having malware transmission through removable storage devices.
v. The current system depends heavily on internet services for communication despite LAN availability.
vi. Administrative overhead associated with managing multiple platforms and reduced efficiency, which is caused by switching between communication and file-sharing tools.

3.2.3 Analysis of the Proposed System

The proposed system is a LAN-Based File Sharing and Communication Web Application which will be designed to provide a unified platform for secure data exchange and

communication within a local area network. The proposed system will integrate file sharing and communication into a single web application, which is a centralised platform accessible through web browsers. The new system will support user registration, authentication, and authorisation, as well as design a role-based access control for managing permissions. It presents a secure method for file upload, download, storage, and sharing functionalities, and integrates real-time messaging and communication between users.

A centralised database for storing user information and file metadata, likewise, activity logging and monitoring capabilities will be implemented in the proposed system. These intended features of the new system will significantly improve collaboration at work or in different organisations through integrated communication features, thereby eliminating the dependence on external communication platforms. The proposed system will operate within a local network environment, which will create faster data transfer due to LAN infrastructure, provide enhanced security through controlled access mechanisms, and improve resource utilisation and management. There is a simplified administration through centralised control and a scalable architecture that can support future enhancements.

3.2.4 Advantages of the Proposed System

The proposed system presents a unified environment for file sharing and communication, which improves collaboration among users.

i. Faster file transfer within the local network and a reduction in the reliance on internet connectivity.
ii. There is a centralised management of users and resources, which will, in turn, guarantee secure authentication and authorisation mechanisms.
iii. Enhanced protection against unauthorised access.
iv. The proposed system improves accountability through user activity tracking, and it has better file organisation and management, which will reduce the risk of file duplication and version conflicts.
v. The new system incorporates real-time communication support, increased operational efficiency, and a user-friendly web-based interface.

vi. They are easier to maintain and manage, and the proposed system is scalable for future system expansion, as well as having a cost-effective deployment within organisations and institutions.

3.2.5 System Requirements

The requirements for the system outline the essential functional and non-functional requirements for the successful implementation and operation of a LAN-Based File Sharing and Communication System.

A. Functional Requirements

i. Secure registration and login via the system are necessary for all users.

ii. All users must be authenticated before being given access to resources on the system.

iii. Role-based authorisation is required to provide proper access control.

iv. All users shall have the ability to upload files to the shared repository.

v. All users shall have the ability to download files from the SharePoint on the system, provided they have access to that file.

vi. All users shall have the ability to view/manage their own files that have been shared on the SharePoint.

vii. All users shall have the ability to communicate in real-time (Via IM) with others on the Network therein.

viii. All users will receive notifications (emails/desktop alerts) that they have messages or new files.

ix. All users shall have the ability to view their file-sharing and communication activity log.

x. Administrative functions will be implemented to allow for managing users' information and activities, files, and the system.

B. Non-functional Requirements

i. Secure data transmission and storage to the local area network.

ii. Quick response times for both file sharing and message operations.

iii. Must use dependable authentication and authorisation methods.

iv. Should support a number of simultaneous users without appreciable performance degradation.

v. Should safeguard user IDs, passwords, and any other private data that contain sensitive information by means of securing them with encryption methods.

vi. Must possess a user-friendly and high-performance web interface.

vii. Must have excellent availability and reliability while working.

viii. Will allow for scalability and flexibility regarding future enhancements and/or integration.

3.3 System Design Methodology

The Object-Oriented Analysis and Design (OOAD) methodology was chosen for this research as it offers a structured approach to analysing user needs, designing components that will be used in the system, modelling how the components will work together, and constructing web-based applications that meet these requirements. Using this methodology allows for creating modular, scalable, and maintainable software systems through the use of object-oriented principles and modelling methodologies.

The development process includes requirements analysis (the process of identifying what the system should do), system design (how the various parts of the computer work), database design; user interface design, implementation, testing, and deploying the proposed Local Area Network (LAN) based File Sharing and Chat System (FSC). OOAD also allows for representing system structure, functionality, and reasoning about relationships using Unified Modelling Language (UML) diagrams.

3.3.1 Unified Modelling Language (UML) Diagrams

UML diagrams are tools that help visualize, create, design and document software based on their function; this means that UML diagrams will give users insight into how the system will work when completed including how people will use it, what will be built, how components relate to one another and what function they have in the final software product. These diagrams can be used as a means of communicating between developers, systems analysts and project stakeholders during the entire development process. The proposed LAN Based File Sharing and Communication System will use UML diagrams to represent system requirements, interactions between users, the communication process for sending and receiving files/messages, how to access documents/files as well as what administrative function will exist in the proposed system. The specific structural models developed for this research include a Use Case Diagram, Activity Diagram, Sequence Diagram and Class Diagram.

A. Use Case Diagram for the Proposed System

This diagram visually illustrates the functional requirements identified for the proposed LAN-Based File Sharing and Communication System and demonstrates various types of user/system interactions. This document identifies actors that are relevant in the system environment as well as all services provided by the proposed system. There are two main actors within this system, which are the Users and Administrators. For the Users, interaction occurs with the following functional requirements: registration, logging in, uploading/downloading, sharing of files, sending and receiving messages, viewing notifications and managing personal accounts. For the Administrators, the included functional requirement would be managing Users, monitoring activity of the entire system, restricting/controlling access permissions, managing resources, and maintaining the system itself. The use case diagram of the proposed system provides a high-level overview of how users interact with the system and how the system responds to user requests. Figure 3.1 presents the use case diagram for the proposed LAN-Based File Sharing and Communication System.

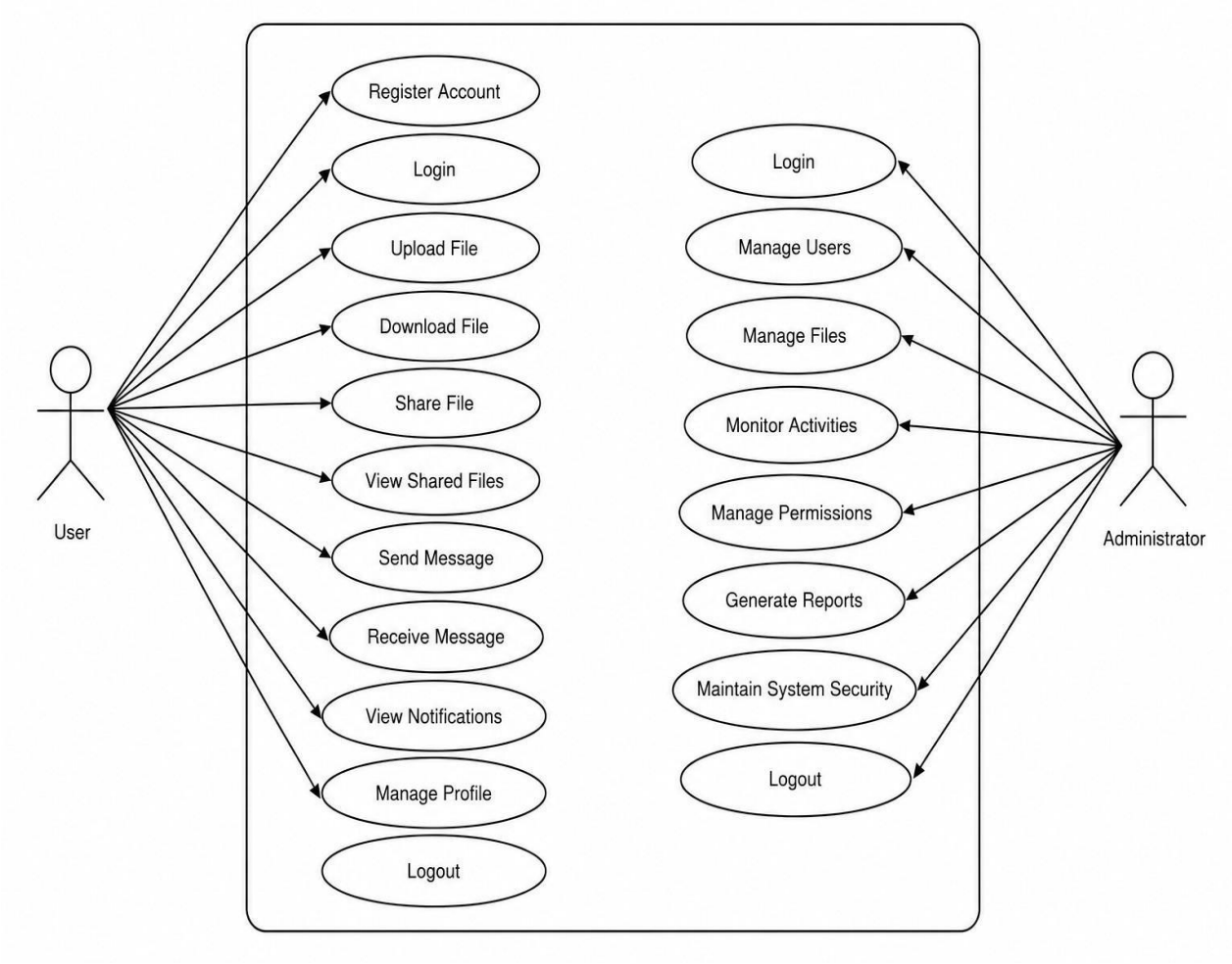

Figure 3.1: Use case diagram of Proposed System

The Use Case Diagram is used to represent the functional requirements and interactions of the proposed LAN-Based File Sharing and Communication System. It defines 2 main actors: the User and the Administrator. User interacts with the system to carry out activity: Registering in, authentication, uploading/downloading files, sharing files, view shared files, sending/receiving messages, view personal profile, view notifications, or log out. The Administrator performs to system management functions such as user management, file management, activity monitoring, permission administration, report generation, system security maintenance, and authentication operation. The diagram illustrates the direct interaction between the actors and the system through the specific use cases that are located within the system boundary, providing a clear representation of the system's functionalities, user responsibilities, and administrative controls

needed to ensure secure and efficient sharing and communication of files within a local area network environment.

B. Class Diagram

The UML class diagram illustrates the system's static structure, showing the blueprint of the objects, their data, and how they interact. Figure 3.2 presents the class diagram of the proposed LAN-Based File Sharing and Communication System.

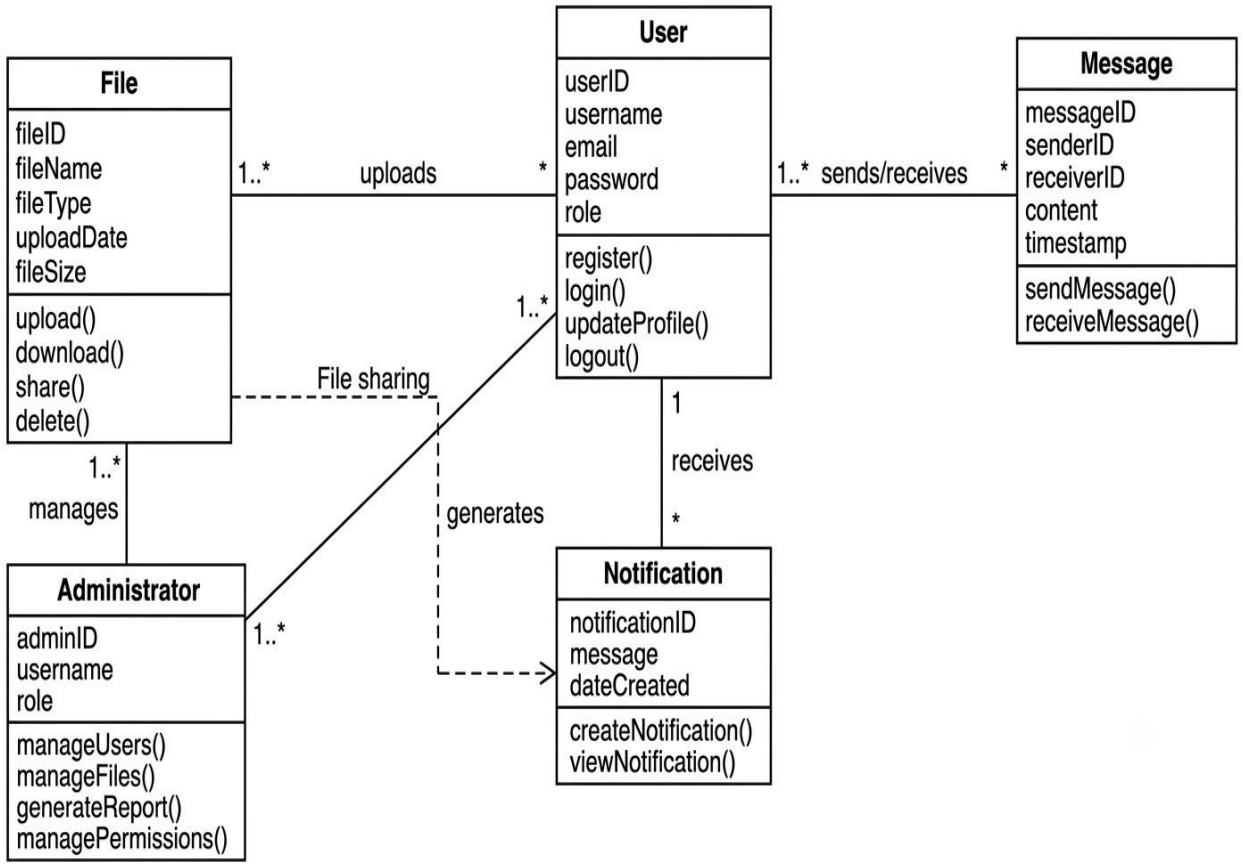

Figure 3.2 Class Diagram of the Proposed System

The UML class diagram presents a clean and academic representation of the structural architecture for the LAN-Based File Sharing and Communication System, detailing its fundamental data elements, functionalities, and relationships. The User class is at the heart of the system, responsible for some of the most critical authentication and profile management tasks

and used as the main communicator and file operator. Users can upload more than one File entity that contains data assets that are shared throughout the local network, or send messages using the Message class, including sender and receiver identity and content timestamp. Structural associations are used to ensure operational accountability and system health: an asynchronous dependency exists as file-sharing operations automatically create Notification logs for designated recipients and a dedicated Administrator class provides administrative control and special privileges for system files and user permissions, as well as for system activity and monitoring.

C. Activity Diagram

The UML activity diagram in Figure 3.3 lays out a series of operations (workflow) that comprise the LAN-Based File Sharing & Communication System, and functions as an architectural blueprint for a Computer Science Project.

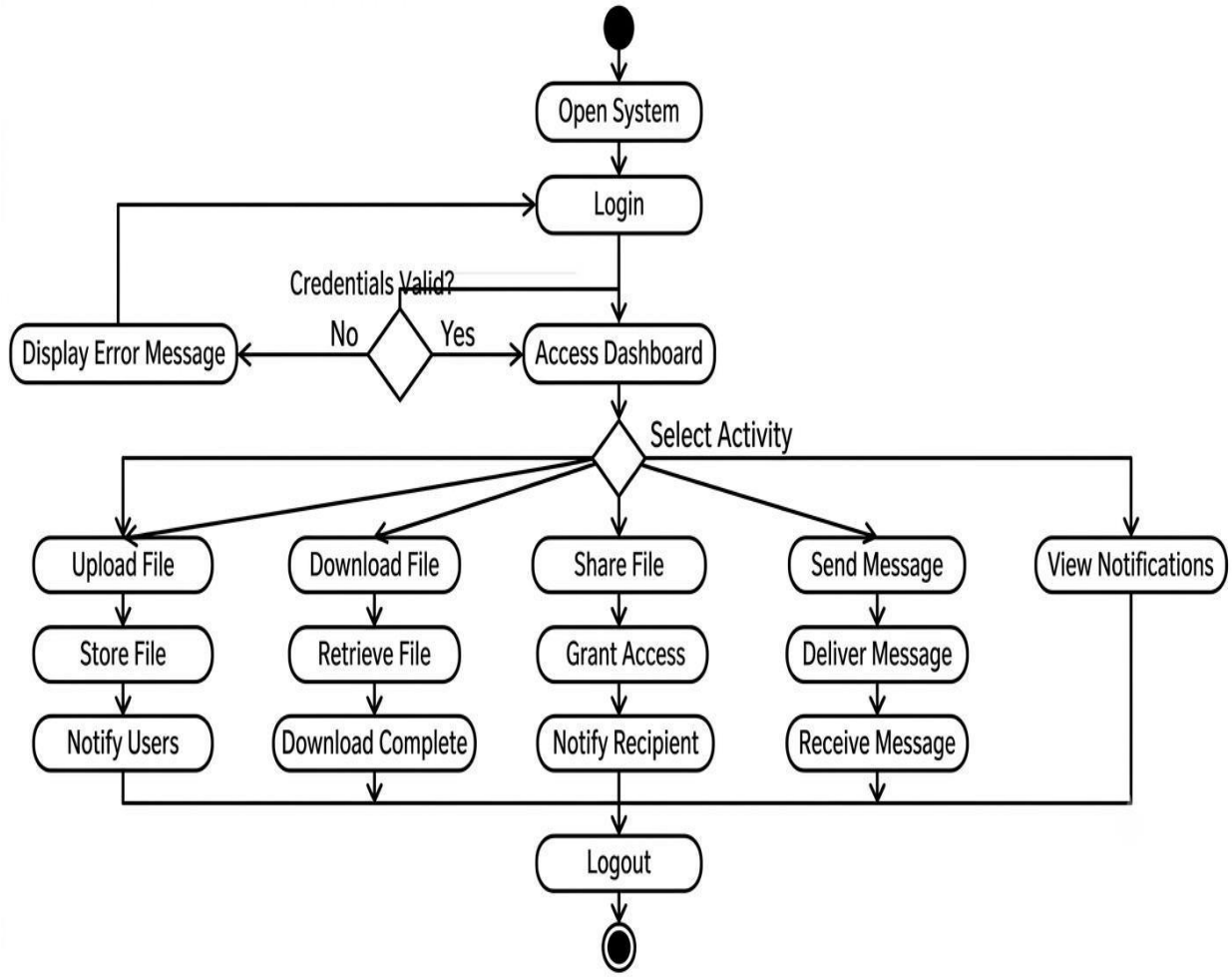

Figure 3.3 Activity Diagram of Proposed System

The first node in the diagram is the Initial Node, which represents the user opening the system and proceeding to log in. After logging into the system, the user will go through a decision diamond where their credentials will be validated; if any of the inputs are invalid, the user will return to the Log On Screen with an error message. If the user's login credentials were successfully authenticated, the user will then proceed to the Main Dashboard. The Main Dashboard has one Central Decision Node that branches out into five parallel, mutually exclusive Functional Pathways: Uploading a File (i.e., stores the file on the server and letting users know), Downloading a File (i.e., retrieves the downloaded file and completes the download), Sharing a File (i.e., grants file access and notifies the intended recipient), Sending a Message (i.e., handles the sending and receipt of messages), and/or Viewing Notifications. After completing any of the Functional Pathways, the flow of control will converge to a single Logout Process. The activity diagram will terminate with the Final Node that represents the closure of the user's session with the system.

D. Sequence Diagram

The UML sequence diagram shows how 7 key entities in a LAN-Based File Sharing and Communication System interact over time with each other during their regular performance. Figure 3.4 presents the sequence diagram of the proposed LAN-Based File Sharing and Communication System, with the first event of the process taking place when a User starts the application and enters the User's login information through the user's Web Interface. The User Web Interface submits the User's login information to the Authentication Module so that it can verify the user's credentials against the Database. An Alternative conditional block is used to process the outcome of the authentication attempt. If the User credentials are invalid, the Authentication Module will respond by calling displayErrorMessage(). If the User credentials are valid, the Authentication Module will send the User to the User's system dashboard.

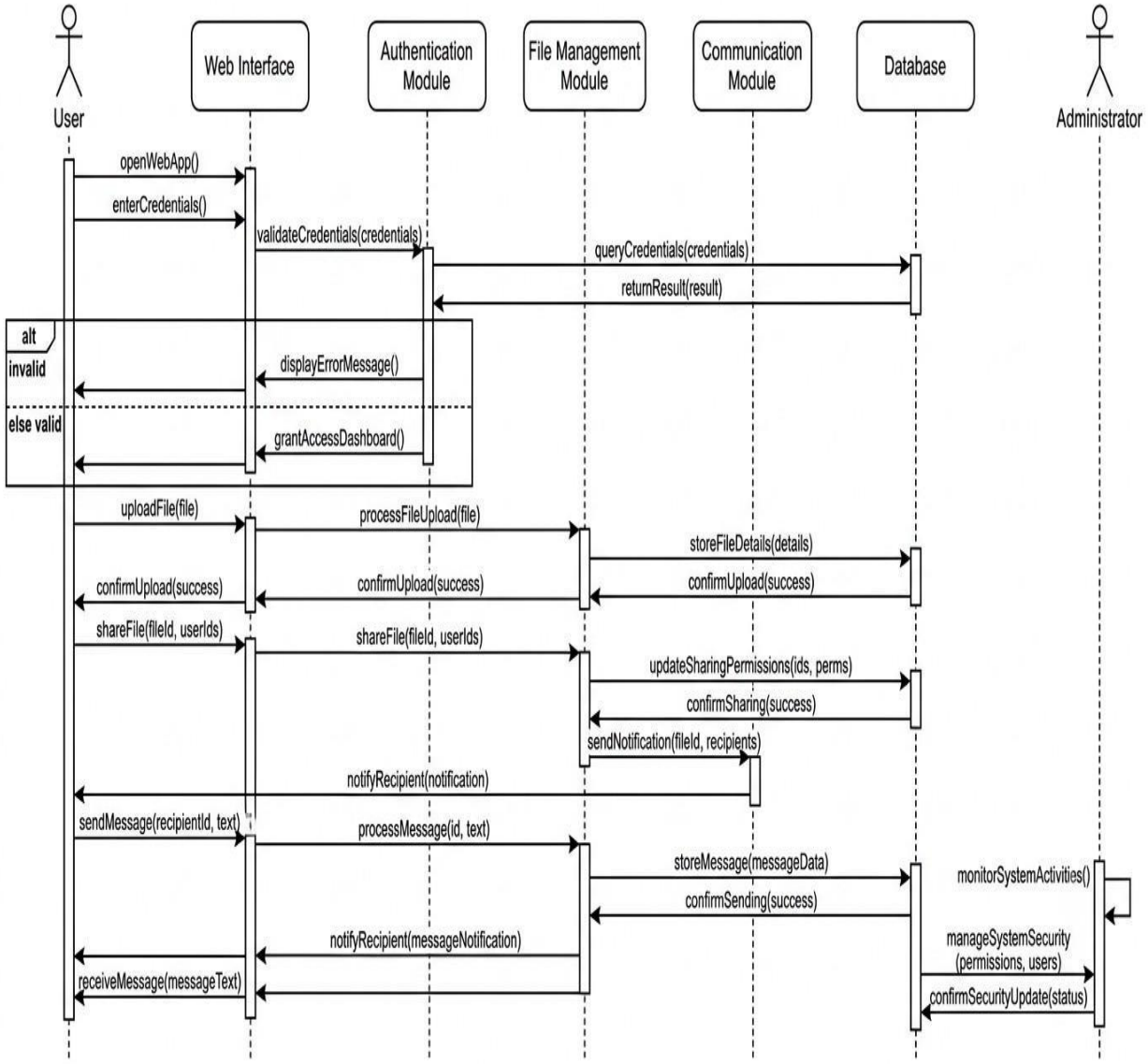

Figure 3.4 Sequence Diagram of Proposed System

Once the User is authenticated, the User can perform the following three main operations in synchronous request/response loops: upload a file, the File Management Module will write the file information to the Database; share a file, the Database will update the permissions associated with the file; and send messages, the Communications Module will route text to the and deliver it to the recipient(s). Simultaneously, the Administrator has an independent lifeline to monitor the global activity of the system and to directly manage the Database to maintain security for the overall system configuration and users' permissions within the local network.

3.3.2 Data Collection

The data collection methods for system modelling and design use direct observation and published literature (both scholarly and technical). The primary function of these methods, through direct observation only, is to gather enough data regarding real-world file sharing and communication practices, tools (manual methods), and processes currently in operation within the academic field, which will provide the necessary operational knowledge to define the end-user's needs to support the design of the proposed end-user system. Additionally, by conducting a review of relevant research literature, technical reports, or previously completed projects related to the proposed system, we can include this data to help inform design decisions concerning the proposed system's architecture, data architecture, and user interface layout. This combined method provided the necessary information for accurately modelling the proposed end-user system, as well as providing sufficient data for defining the proposed end-user system's functional and non-functional requirements.

3.4 System Architecture

The system will utilise a Client/Server Architecture where the server will be the central point of control for all resources and services available on the system, while clients will access resources through the Local Area Network (LAN). The advantage of the Client/Server architecture is that it will provide centralised control, simplify the management of the system, and provide a consistent level of performance for all users connected to the system.

Figure 3.5 presents the system architecture of the proposed LAN-Based File Sharing and Communication System, which consists of three main tiers:

i. Presentation Tier (Client Side): This is the front-end user interface and includes the Registration/Login Interface, File Sharing Interface, Messaging Interface, and Admin Dashboard. User input will be received at the Presentation Tier and sent to the Application Tier for processing.

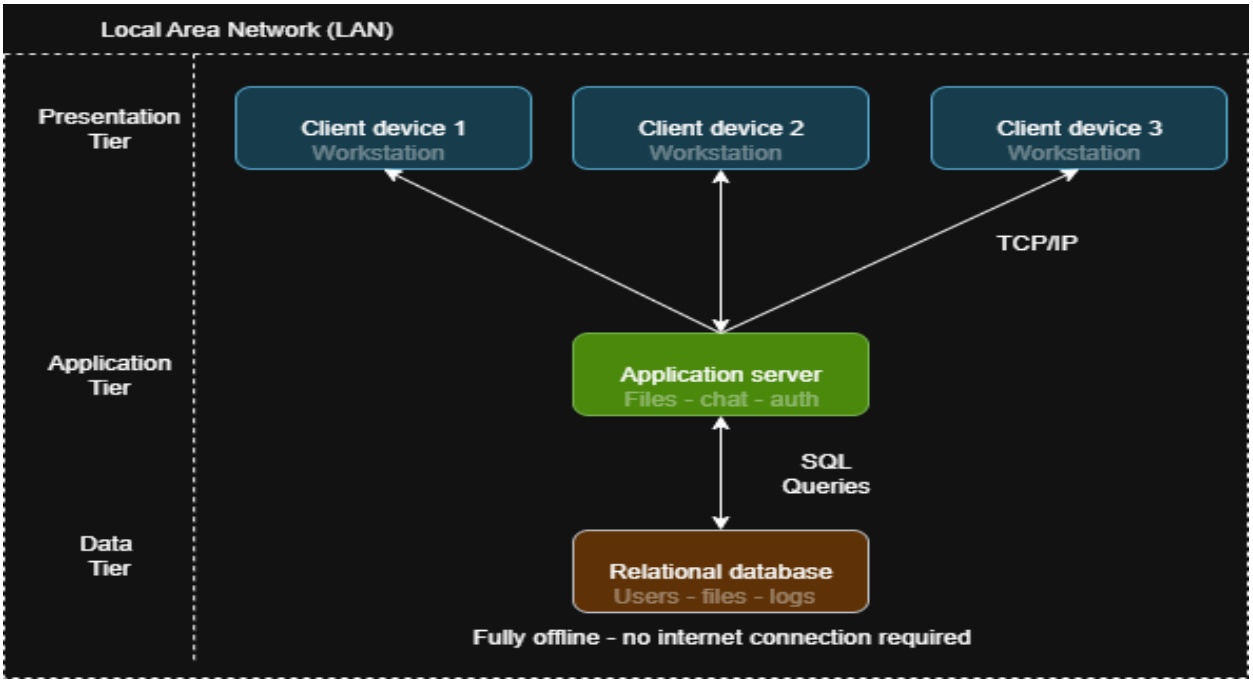

Figure 3.5 Proposed System Architecture Diagram

ii. Application Tier (Server Side): This is the main core processing layer of the system, and it will be located on the server. The Application Tier will process requests for user Authentication, file uploads/downloads, route real-time messages, enforce access controls, and log all activity. All Business Logic will reside in the Application Tier.

iii. Data Tier (Database Layer): This layer will be responsible for managing all persistent data storage for the system via a Relational Database. The Data Tier will contain user records, file metadata, message history, and activity logs. The Data Tier will respond to data queries that originate from the Application Tier.

All clients will connect to the server through the LAN using standard TCP. No internet connection is required for the system to function, making it fully self-contained within the local network environment.

3.5 Database Design Schema

Database design defines the structure of the database tables, which we will use in the design and implementation of this proposed system to store, organise, manage, and retrieve system information. The proposed LAN-Based File Sharing and Communication System uses a relational database structure in order to effectively maintain user records, file information, sharing permissions, messages, and notifications. It ensures that data is organised very well and stored securely, and in the same vein, making it easily accessible during file sharing, communication, and administrative operations.

Table 3.1: User Table for LAN-Based File Sharing and Communication System

Attribute	Data Type	Description
user_id	Int (11)	Unique identifier for each system user.
full_name	Varchar(100)	Full name of the registered user.
username	Varchar(50)	Username used for system login.
email	Varchar(100)	User's email address.
password	Varchar(255)	Hashed password used for authentication.
role	Enum	Defines user role, such as User or Administrator.
phone_number	Varchar(15)	Contact number of the user.
status	Varchar(20)	Indicates whether the user account is active or inactive.
date_joined	Timestamp	Date and time the user account was created.
updated_at	Timestamp	Date and time the user information was last updated.

Table 3.2: File Table

Attribute	Data Type	Description
file_id	Int (11)	Unique identifier for each uploaded file.
user_id	Int (11)	Foreign key linking the file to the user who uploaded it.
file_name	Varchar(150)	Original name of the uploaded file.

file_type	Varchar(150)	Format or extension of the file, such as PDF, DOCX, PNG, or ZIP.
file_size	Float	Size of the uploaded file.
file_path	Varchar(255)	Storage path or location of the file on the server.
upload_date	Timestamp	Date and time the file was uploaded.
access_level	Varchar	Defines whether the file is private, shared, or public within the LAN.
description	Text	Brief description or details about the uploaded file.
updated_at	Timestamp	Date and time the file record was last updated.

Table 3.3: Message Table

Attribute	**Data Type**	**Description**
message_id	Int (11)	Unique identifier for each message.
sender_id	Int (11)	Foreign key linking the message to the user who sent it.
receiver_id	Int (11)	Foreign key linking the message to the message recipient.
message_content	Text	Text content of the message.
message_status	Varchar(20)	Indicates whether the message is sent, delivered, or read.
sent_at	Timestamp	Date and time the message was sent.
read_at	Timestamp	Date and time the message was read by the recipient.
created_at	Timestamp	Date and time the message record was created.
updated_at	Timestamp	Date and time the message record was last updated.

Table 3.4: File Sharing Table

Attribute	**Data Type**	**Description**
share_id	Int (11)	Unique identifier for each file-sharing record.
file_id	Int (11)	Foreign key linking the shared file to the File table.

owner_id	Int (11)	Foreign key identifying the user who owns or uploaded the file.
receiver_id	Int (11)	Foreign key identifying the user receiving access to the file.
permission_type	Varchar(20)	Defines access permission, such as view, download, or edit.
shared_at	Timestamp	Date and time the file was shared.
expiry_date	Date	Optional date when access to the shared file expires.
sharing_status	Varchar(20)	Indicates whether the sharing permission is active or revoked.

Table 3.5: Notification Table

Attribute	Data Type	Description
notification_id	Int (11)	Unique identifier for each notification.
user_id	Int (11)	Foreign key linking the notification to the receiving user.
notification_type	Varchar(30)	Type of notification, such as file share, message, or system alert.
notification_message	Text	Content of the notification displayed to the user.
status	Varchar(10)	Indicates whether the notification has been read or unread.
created_at	Timestamp	Date and time the notification was generated.

Table 3.6: Activity Log Table

Attribute	Data Type	Description
log_id	Int (11)	Unique identifier for each system activity record.
user_id	Int (11)	Foreign key linking the activity to a system user.
activity_type	Varchar(30)	Type of activity performed, such as login, upload, download, share, or message.
activity_description	Text	Detailed description of the activity performed.

ip_address	Varchar(45)	IP address of the device used within the local network.
activity_time	Timestamp	Date and time the activity occurred.

CHAPTER FOUR

SYSTEM IMPLEMENTATION, RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the implementation, evaluation, and discussion of the LAN-Based File Sharing and Communication System, which was designed and built in order to decrease vulnerability to malware and unauthorised entry through the use of no external storage devices, provides quick and accurate transfer of files in the local network and an improved collaboration where users can communicate and share resources easily. The study further discusses the development environment, system testing, and system interface components in this chapter.

4.2 Development Environment and Tools

The system was developed within a Windows Subsystem for Linux (WSL) environment, using Ubuntu under WSL to provide a stable Linux-based development platform. Visual Studio Code was utilised as the integrated development environment in addition to the Jupyter Notebook in order to carry out the data preparation, preprocessing and exploratory data analysis, as well as model training, testing and validation. We employed PostgreSQL for database implementation, and FastAPI was adopted for backend API development due to its speed and scalability.

4.2.1 Programming Languages and Libraries

Python's ease of use, flexibility, and web app support were chosen for the study. For developing the proposed LAN-Based File Sharing and Communication System, several libraries and frameworks are installed and integrated. Our API (a back-end application programming interface) was developed using FastAPI, which facilitates communication between the userinterface, database, file management modules and communication services.

The user records, file information, messages, notifications and activity logs were stored in a relational database system called PostgreSQL. For handling efficient communication between the application and the database, SQLAlchemy, an Object Relational Mapper (ORM) tool, was used. The FastAPI backend was deployed and run using Uvicorn. Uvicorn was used as the application server for running and deploying the FastAPI backend. Furthermore, HTML, CSS, JavaScript, and Bootstrap were used to create a responsive and user-friendly web application. Overall, these tools helped to create a secure, efficient, and scalable file-sharing and communication platform.

Table 4.1 presents the tools and libraries we used in the design and implementation of the proposed system.

Table 4.1 Tools and their purpose

Tool/Library	**Purpose**
Python	Main programming language
FastAPI	Backend API development
Uvicorn	Backend server execution
PostgreSQL	Database management
SQLAlchemy	Database communication and ORM
HTML	Web page structure design
CSS	User interface styling
JavaScript	Client-side interactivity
Bootstrap	Responsive user interface development
Git	Version control and source code management
Visual Studio Code	Source code development and editing

4.2.2 Development Platform and Hardware Requirements

The study developed a Local Area Networking (LAN) based File/Communication system using a development environment that includes FastAPI, PostgreSQL, and Python virtual environment for dependency management. FastAPI applications ran on Uvicorn server, and PostgreSQL had to be used as the back-end DBMS; Visual Studio Code has also been used as the Integrated Development Environment (IDE) for writing, debugging, and verifying that the system works as intended. The LAN-based system setup supports deployment in a Local Area Network (LAN) but could also be extended to consider larger-sized end users' businesses. Development platforms and the hardware requirements used during system implementation are provided in Table 4.2.

Table 4.2: Development Platform and Hardware Requirements

Component	**Specification**
Operating System	Windows 10/11 or Ubuntu Linux
Code Editor	Visual Studio Code
Backend Framework	FastAPI
Database	PostgreSQL
Web Technologies	HTML, CSS, JavaScript, Bootstrap
Programming Language	Python 3
Application Server	Uvicorn
Version Control	Git
RAM Requirement	Minimum 8 GB
Storage Requirement	Minimum 20 GB Free Space
Network Requirement	Local Area Network (LAN) Connectivity

Processor Requirement	Intel Core i3 or Higher

4.3 System Implementation and Modules

The proposed system was set up as an application on a local area network to be accessed online, comprising four core modules. The modules correspond one-to-one with functional requirements outlined in Chapter Three.

Module 1: User Authentication Module:

This module is used for user registration and login. New users register by filling out the registration form with their full name, username and password. Werkzeug's utilities for security hashing the passwords before storing them in the database, so plain-text passwords are never stored. When the user logs in, the user credentials are checked against the stored hash, and a user session is started if the credentials are correct. Unauthorised users are forced to log on and denied access to all the features of the system.

Module 2: File Sharing Module:

This module handles all activities related to uploading and downloading files. If the user has authenticated accounts, then he/she can select a file on his/her device and upload it to the server, which stores it in a particular server directory. The file name, type, size, identity of the person who uploaded it and the time of upload are stored in the database. Other logged-on users have a view of the file repository, can see what files are available, and can start a download to their local machine. The system imposes format restrictions, only accepting file formats of document, image and archive defined by the functional requirements.

Module 3: The Real-Time Messaging Module:

This module allows for real-time text-based communication between connected users. With Flask-SocketIO, the server will wait for messages to be sent and broadcast them to all the connected browsers in real time. Messages are stored in the database with the identity of the sender, the content of the message and the time of the message, so that the history of messages

can be viewed during the session. The messaging interface is dynamic, meaning no page reloading which means a seamless messaging experience.

Module 4: Administrative Module:

This module is available only to users who are administrators. It has a dedicated dashboard that allows the administrator to see all users registered, to manage the users' accounts, to track uploaded files, to delete files from the repository and to view the logs of system activity. Access is controlled at the route level, which means that normal users are not able to access administration functions even if they try to take a direct link to the admin URL.

4.4 Testing and Evaluation

After implementation of the proposed LAN-Based File Sharing and Communication System, system testing was conducted in order to verify that the modules which were implemented functioned correctly and met the requirements that were specified in Chapter Three. And for effective usage and in order to ensure the efficiency of the system, two levels of testing were performed, including system testing and user interface testing.

4.4.1 System Testing

System testing was carried out by executing a series of test cases covering each module of the system. The table below presents the test cases, the expected outcomes, and the actual results recorded during testing.

Test No.	Module	Test Case Description	Expected Result	Actual Result	Status
TC-01	Authentication	User registers with valid credentials	Account created and user redirected to login page	Account created successfully	✓ Pass
TC-02	Authentication	User registers with an already existing username	Error message displayed: "Username already exists"	Error message displayed correctly	✓ Pass
TC-03	Authentication	User logs in with correct credentials	User redirected to dashboard	Login successful	✓ Pass

TC-04	Authentication	User logs in with incorrect password	Error message displayed: "Invalid credentials"	Error message displayed correctly	✓ Pass
TC-05	File Sharing	User uploads a supported file type (PDF)	File saved to server and listed in file repository	File uploaded and visible in repository	✓ Pass
TC-06	File Sharing	User uploads an unsupported file type (.exe)	Upload rejected with error message	Error message displayed; file not saved	✓ Pass
TC-07	File Sharing	User downloads a file from the repository	File downloaded to user's local machine	File downloaded successfully	✓ Pass
TC-08	Messaging	User sends a message to the chat	Message appears instantly in the chat window for all connected users	Message delivered in real time	✓ Pass
TC-09	Messaging	Message history is visible after page reload	Previous messages displayed upon reconnection	Message history loaded correctly	✓ Pass
TC-10	Admin	Administrator accesses admin dashboard	Dashboard displays user list, file list, and activity log	Dashboard loaded with correct data	✓ Pass
TC-11	Admin	Administrator deletes a file	File removed from repository and database	File deleted successfully	✓ Pass
TC-12	Admin	Regular user attempts to access admin dashboard	Access denied; user redirected to dashboard	Access denied correctly	✓ Pass
TC-13	General	System operates with no internet connection	All features function normally over LAN	Full functionality confirmed offline	✓ Pass

All thirteen test cases performed for our proposed system passed successfully, confirming that the system behaves as expected across all modules.

4.4.2 User Interface Testing and Walkthrough

The system screens were tested for their usability, clarity and responsiveness in user interface testing. The key interfaces were walked through to ensure ease of navigation between modules and the correct functionality of all interactive components. The following interfaces were tested:

A. Login and Registration Page: Users could easily get to the Registration Form and fill it with their information without getting lost and go back to the Login Page. Form validation messages were informative and concise.

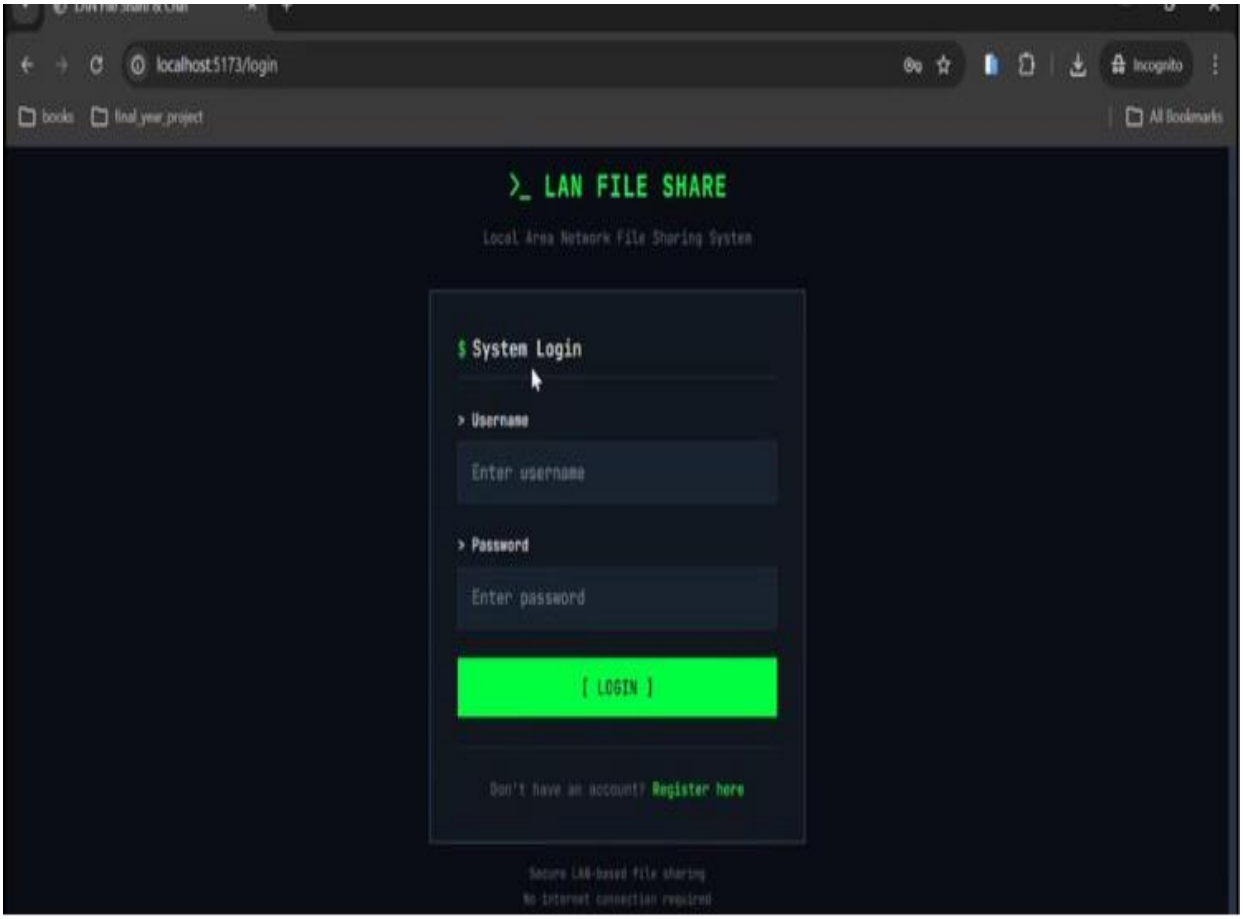

Figure 4.1 Login Page

B. User Dashboard: Upon logging in, the user would be shown a clean dashboard with navigation options for accessing the file repository and the messaging interface. There was a consistent and accessible layout.

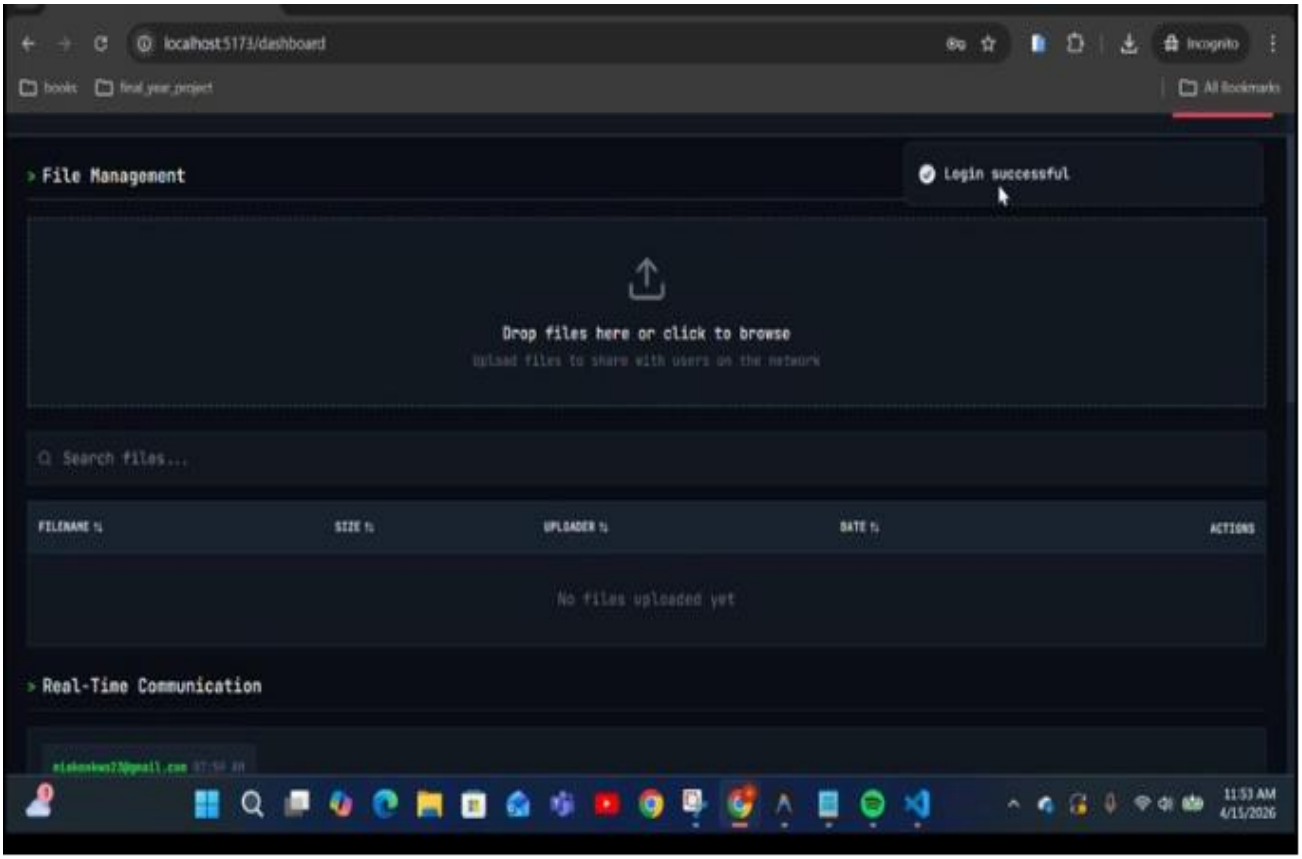

Figure 4.2 User Dashboard page

C. File Upload Interface: File selection control and upload button worked as intended. A success message was displayed when the upload was complete, and the file list was automatically updated.

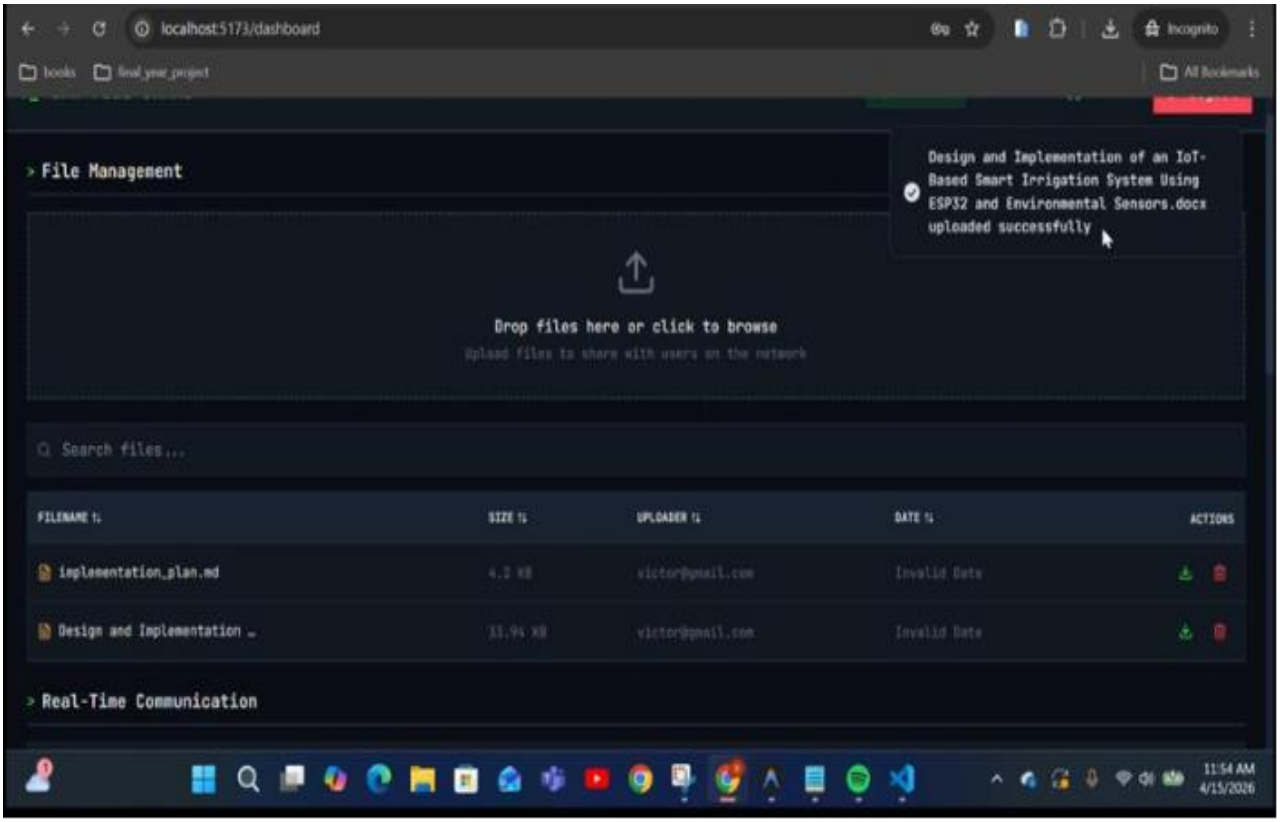

Figure 4.3 File Upload interface

D. Messaging Interface: The real-time chat window displayed the messages with the sender's name and time of display. Other users' messages were displayed without having to refresh the page, thus verifying that Socket was working.

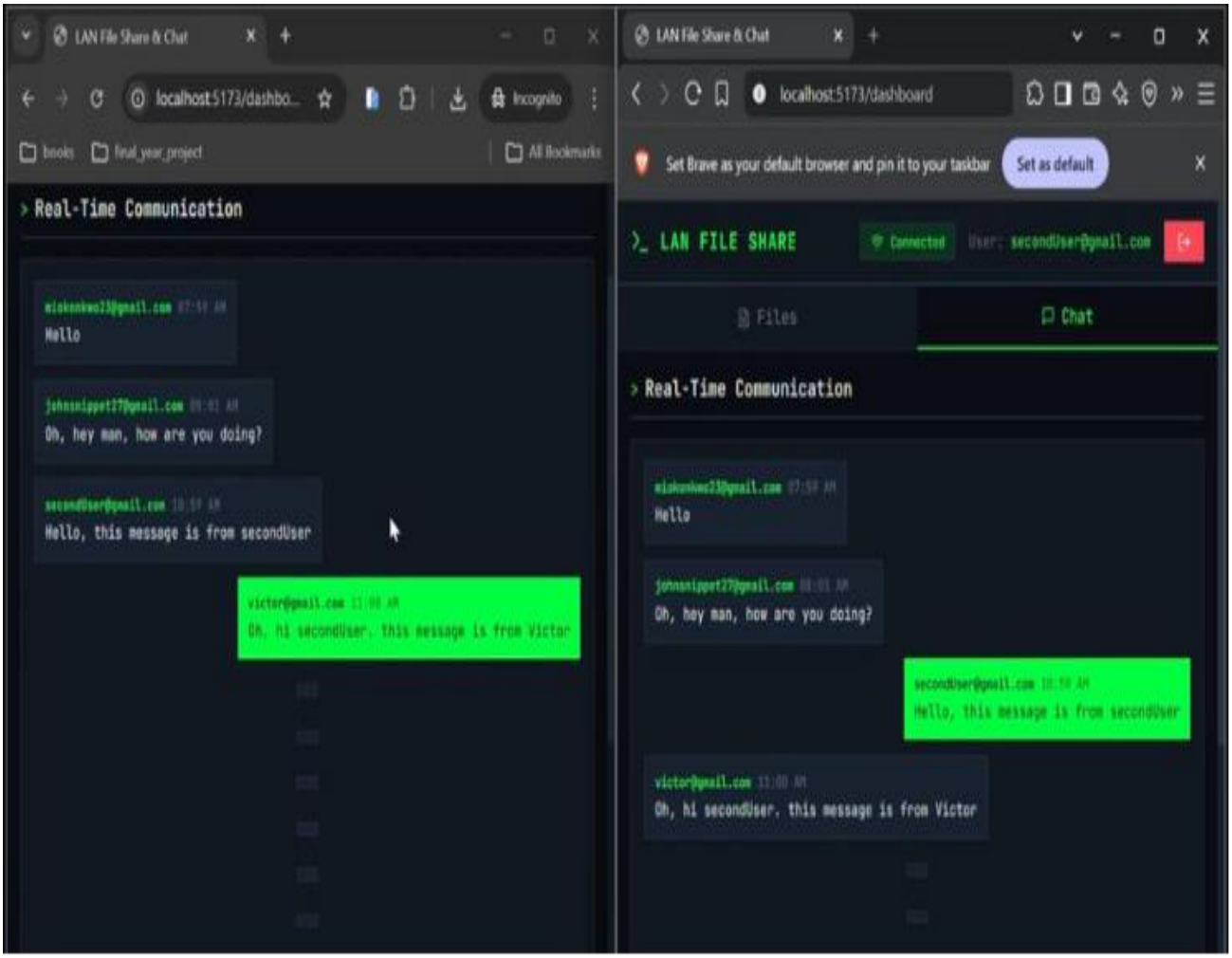

Figure 4.4: Messaging Interface

4.5 Discussion of Results and System Performance

Both system testing and user interface testing confirmed that the LAN-Based File Sharing and Communication System met all five objectives outlined in Chapter 1 of this paper; that is, all thirteen test cases passed with no errors. The results indicate that all modules were correct and stable. No Internet connectivity was provided to the system during all phases of testing; therefore, all of the major hypotheses presented in this research were validated. Because there was not a single file upload or download that failed, and all real-time messaging occurred with little-to-no delay between connected client systems on the test LAN, the system's core premise was validated. Access control was correctly enforced at every privilege level; regular users could

not access any administrative functions, and unauthenticated users could not access any system resources.

In addition to validating that all testing met all requirements of the system, the performance of the system was examined to determine whether three different client devices can connect concurrently. The results indicated that all three clients were able to connect, and no degradation was experienced in messaging response time or file transfer speed due to the addition of additional clients. The limitations associated with this limitation were previously discussed in Chapter 1 and represent potential areas for future research.

In conclusion, the findings of this study demonstrate that a fully functional, secure, offline-capable file-sharing and communication system can be successfully implemented within a LAN environment using lightweight, open-source tools. Thus, the proposed solution is viable and cost-effective.

CHAPTER FIVE

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.1 Introduction

This chapter summarises, concludes and recommends for the research project Design and Implementation of a LAN-Based File Sharing and Communication System. The purpose of the chapter is to provide a brief overview of the research process utilised throughout all four chapters preceding it, establish conclusions based on results achieved during implementation and testing of the system, and provide recommendations for future enhancement and use of the system. This chapter will be a final review as to whether or not the research met its research objective(s) and what impact it has on the field of computer networking and institutional information management.

5.2 Summary of the Study

In academic and organisational settings where there is poor or expensive internet access, or no internet access, the ongoing challenges of file sharing and internal communications were the motivation for this research.

Chapter One has presented the background of the study, the problem of dependence on physical storage mediums and internet-based tools as the main mediums to share information and the aim and objectives of the study. The importance of the proposed system was discussed, and the scope, limitations and operational definition of the terms used in this study were discussed. A detailed review of the literature related to the research was given in Chapter Two. Theories included the basics of Local Area Network and file sharing systems and communication models in networked environments. Fifteen related studies were reviewed and their contributions, methodology and limitations are summarised. The research gap identified is the absence of a single LAN based unified platform that supports file sharing and real-time online communication in a single platform in an academic environment.

The analysis and design of the proposed system were a major concern of Chapter Three. The current system was found to have several drawbacks, such as dependency on physical media, limited real-time communication capabilities, security issues, and inadequate file management. The limitations were overcome by the design of the proposed system in the client server architecture with three layer presentation, application and data. The system requirements were defined, the Object-Oriented Analysis and Design methodology was presented, and UML diagrams such as use case, activity and class diagrams were created to model the structure and behaviour of the system. Designs for input, output, database and architecture were also discussed. In Chapter Four, we implemented the system using Python, Flask, Flask-SocketIO, SQLite and conventional web technologies. The modules were user authentication, file sharing, real-time message and administration. Thirteen test cases were run on all the modules for system testing—all of which passed successfully. User interface testing proved the system was navigable, responsive and functioned correctly. The results showed that the system could be used completely offline over a LAN, could support multiple users at the same time and had proper access control, which was respected at all privilege levels.

5.3 Conclusion

The purpose of this study, which is to create and develop a LAN-based file sharing and communication tool that can be used to share files and communicate in real-time among users in the same LAN without depending on an internet connection or physical storage media, has been successfully achieved. The results of implementation and testing can be seen that this objective has been achieved. The first goal was to identify the shortcomings of current file sharing and communication techniques, which was accomplished by the system analysis carried out in Chapter Three, where it was found that there are several limitations, such as security, efficiency and lack of integrated communication capabilities. The three-tier client-server architecture, the database schema and UML models shown in Chapters Three and Four satisfied the second objective, which was the design of a LAN-based file sharing system. To achieve the third goal of having real-time communication in the system, the Flask-SocketIO messaging module was used to pass on messages to all users connected to the system during the testing phase.

The fourth objective, implementation of user authentication and role-based access control, was successfully completed and all test cases were executed successfully in which the system limited

access according to user's privilege levels. The fifth objective of finding the performance of the proposed system was realized with the thirteen test cases that were carried out in chapter four, and all of them were found to pass. The system showed that it is possible to create a light, full offline, cost-efficient platform for institutional communication and file sharing based on open source tools and installed in a standard LAN environment. However, it is reported that its execution performance was not evaluated when it is used by many users and no data encryption was used, and it is admitted that its improvement is also needed in the future.

5.4 Recommendations

Based on this research, the following recommendations are made for future development and adoption of the system:

i. End-to-end encryption in transfers and messages: Future updates of the system should include end-to-end encryption for transfers and messages. Protocols like TLS (Transport Layer Security) should be implemented to ensure data is safe during transmission within the LAN, especially in educational or organisational settings where sensitive information is being shared.

ii. Tested with limited concurrency: The current system is tested with a small number of concurrent users. Future development is recommended to work on scalability, implementing a more robust database system (PostgreSQL/MySQL) and optimising the server architecture to handle more concurrent connections without any impact on performance.

iii. Mobile Application Development: An application should be created that can run on a mobile device to enable the user to use the file repository and messaging functions from a mobile device on the same LAN. This would greatly enhance the accessibility and usability in a larger institutional setting.

iv. File Versioning and Recovery: An improvement of the file versioning functionality would be helpful, allowing users to view previous versions of uploaded files and recover accidentally deleted files. This would mean the system is more robust and would be more readily adaptable to use in collaborative academic activity.

v. Notification System: Push notification functionality should be included to notify users of new files uploaded or received new messages, even when the application is in the background. This would enhance the involvement of users and guarantee the timely access to shared resources.

vi. Institutional Adoption: It is proposed that academic departments/Institutions, which have poor internet facilities, should consider adopting and customising this system in their internal communication and handling of files. It is a practical, easily implemented solution due to its low cost of deployment, being offline, and ease of use.

References

Afanasyev, A., Zhu, Z., Yu, Y., Wang, L., & Zhang, L. (2015). The story of ChronoShare, or how NDN brought distributed secure file sharing back. In *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)* (pp. 525–530). IEEE. https://doi.org/10.1109/MASS.2015.59

Alfarhood, S., Khaleel, M. I., Safran, M., Shakor, M. Y., & Zhu, M. (2024). Dynamic AES encryption and blockchain key management: A novel solution for cloud data security. *IEEE Access, 12*, 21558–21574. https://doi.org/10.1109/ACCESS.2024.3351119

Anderson, P., Mitchell, R., & Brown, S. (2022). Distributed document management system for local network environments. *Information Systems, 108*, 234–251. https://doi.org/10.1016/j.infostr.2022.101876

Anthal, J., Choudhary, S., & Shettiyar, R. (2023). Decentralizing file sharing: The potential of blockchain and IPFS. In *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)* (pp. 773–777). IEEE. https://doi.org/10.1109/InCACCT57535.2023.10141817

Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics, 12*(6), 1333. https://doi.org/10.3390/electronics12061333

Banga, P., Nirottam, Kumar Singh, S., & Singh, P. (2023). A secure file encryption and decryption system using AES for text and images. *International Journal of Research–Granthaalayah, 11*(12). https://doi.org/10.29121/granthaalayah.v11.i12.2023.6125

Bhupathi, K. K. (2025). The digital fabric of society: A critical analysis of computer networking's transformative impact on social structures and institutions. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *11*(1), 1376–1386. https://doi.org/10.32628/CSEIT251112244

Chen, L., Wang, M., & Zhang, H. (2023). Design and implementation of web-based file sharing application using MERN stack. *Journal of Web Engineering*, *18*(4), 412–429. https://doi.org/10.11591/jwe.v18i4.412

Chong, C. L., & Harun, N. Z. (2025). Secure file sharing system with strong password and one-time password authentication. *Journal of Computing Research and Innovation*, *10*(1). https://doi.org/10.24191/jcrinn.v10i1.500

Currey, J., & Rostami, M. (2025). Formal verification for preventing misconfigured access policies in cloud systems. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.11122676

Fereidouni, H., Fadeitcheva, O., & Zalai, M. (2025). IoT and man-in-the-middle attacks. *Security and Privacy*, *8*(3), e70016. https://doi.org/10.1002/spy2.70016

Goni, O., & Shameem, A. (2021). Implementation of local area network (LAN) and build a secure LAN system for BAEC head quarter. *Research Journal of Computer Systems Engineering*, *1*(1). https://doi.org/10.36811/rjcse.2021.110003

Ha, G., Ge, X., Jia, C., Chen, Y., & Su, Z. (2025). Revisiting SGX-based encrypted deduplication via PoW-before-encryption and eliminating redundant computations. *IEEE Transactions on Dependable and Secure Computing*, *22*(3), 2037–2053. https://doi.org/10.1109/TDSC.2024.3476288

Hao, J., Xiao, Y., Zhou, W., Wang, P., & Huawei Technologies Co., Ltd. (2026). Network basics and management. In *Operating System Basics and Practice*. Springer. https://doi.org/10.1007/978-981-95-2185-2_7

Hasan, S. S. U., Ghani, A., Daud, A., Akbar, H., & Khan, M. F. (2025). A review on secure authentication mechanisms for mobile security. *Sensors*, *25*(3), 700. https://doi.org/10.3390/s25030700

Hasanoddin, S., Mannan, M. A., Varma, M. R., Shiva, M., & Abhishek, J. (2026). A hybrid identity protection and multi-factor authentication framework for cloud security. *American Journal of AI Cyber Computing Management, 6*(1), 335–347.

Howick, J., Bennett-Weston, A., Solomon, J., Nockels, K., Bostock, J., & Keshtkar, L. (2024). How does communication affect patient safety? Protocol for a systematic review and logic model. *BMJ Open, 14*(5), e085312. https://doi.org/10.1136/bmjopen-2024-085312

Huawei Technologies Co., Ltd. (2023). Network basics in cloud computing. In *Cloud Computing Technology*. Springer. https://doi.org/10.1007/978-981-19-3026-3_4

Iovescu, D., & Tudose, C. (2024). Real-time document collaboration—System architecture and design. *Applied Sciences, 14*(18), 8356. https://doi.org/10.3390/app14188356

Jindal, K., Kharb, H., & Nirala, R. K. (2026). P2P-SFS: Secured peer-to-peer file sharing with end-to-end encryption and authentication. In *Proceedings of the International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*. IEEE. https://doi.org/10.1109/IITCEE67948.2026.11394461

Johnson, M., Davis, A., & Wilson, R. (2023). Secure LAN chat and file transfer system with AES encryption. *International Journal of Advanced Computer Science and Applications, 14*(6), 567–578. https://doi.org/10.14569/IJACSA.2023.0140656

Kim, S., Park, J., & Choi, Y. (2022). Intranet-based knowledge sharing system for enterprise collaboration. *Journal of Knowledge Management, 26*(4), 892–910. https://doi.org/10.1108/JKM-03-2021-0234

Kirti, M., Maurya, A. K., & Yadav, R. S. (2024). Fault-tolerance approaches for distributed and cloud computing environments: A systematic review, taxonomy and future directions. *Concurrency and Computation: Practice and Experience, 36*(13), e8081. https://doi.org/10.1002/cpe.8081

Kumar, A., Singh, R., & Sharma, P. (2024). AES encryption-based secure file sharing system with dynamic key generation using deep learning. *International Journal of Information Security, 12*(3), 234–248. https://doi.org/10.1007/s10207-024-00789-x

Lonzetta, A. M., Cope, P., Campbell, J., Mohd, B. J., & Hayajneh, T. (2018). Security vulnerabilities in Bluetooth technology as used in IoT. *Journal of Sensor and Actuator Networks, 7*(3), 28. https://doi.org/10.3390/jsan7030028

Liu, T., Luo, Y. T., Pang, P. C.-I., & Kan, H. Y. (2025). Exploring the impact of information and communication technology on educational administration: A systematic scoping review. *Education Sciences, 15*(9), 1114. https://doi.org/10.3390/educsci15091114

Mahmoud, H., & Abozariba, R. (2025). A systematic review on WebRTC for potential applications and challenges beyond audio video streaming. *Multimedia Tools and Applications, 84*, 2909–2946. https://doi.org/10.1007/s11042-024-20448-9

Michael, N. B., & Usman, K. (2024). Enhancing file transfer security and efficiency through compression, fragmentation and reassembling techniques. *International Journal of Computer Applications, 186*(39). https://doi.org/10.5120/ijca2024923973

Mostafa, A. M., Ezz, M., Elbashir, M. K., Alruily, M., Hamouda, E., Alsarhani, M., & Said, W. (2023). Strengthening cloud security: An innovative multi-factor multi-layer authentication framework for cloud user authentication. *Applied Sciences, 13*(19), 10871. https://doi.org/10.3390/app131910871

Morrison-Smith, S., & Ruiz, J. (2020). Challenges and barriers in virtual teams: A literature review. *SN Applied Sciences, 2*, 1096. https://doi.org/10.1007/s42452-020-2801-5

Mun, H., Han, K., Damiani, E., Yeun, H. K., Kim, T.-Y., Martino, L., & Yeun, C. Y. (2025). A comprehensive survey on digital twin: Focusing on security threats and requirements. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3563621

Niehage, J. (2024). Vulnerabilities in Nextcloud's server-side encryption. *End-to-End Encrypted Cloud Storage in the Wild: A Broken Ecosystem*. ACM. https://doi.org/10.1145/3658644.3690309

Oliveros, A. G. G. (2025). An interactive file sharing system to support collaborative learning and resource management among university faculty and students. *International Journal of Research and Scientific Innovation (IJRSI), 12*(7). https://doi.org/10.51244/IJRSI.2025.120700014

Raj, A., Narayan, V., Muskan, V., Sani, A., Sharma, P., & Sarma, S. S. (2024). Modern ransomware: Evolution, methodology, attack model, prevention and mitigation using multi-tiered approach. *Security and Privacy*, *7*(6), e436. https://doi.org/10.1002/spy2.436

Safiyanu, A. M., Usman, O. M., & Jibrin, Y. A. (2023). Exploring the landscape of computer networking: An in-depth survey. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*. https://doi.org/10.22214/ijraset.2023.55599

Santos, F. O., & Lopes, N. L. (2025). The effectiveness of collaborative centralized information systems in project management: A pilot demonstration of a multifunctional building in Africa. *Buildings, 15*(6), 860. https://doi.org/10.3390/buildings15060860

Sarower, A. H., Bhuiyan, T., Hassan, M., Arefin, M. S., & Hossain, G. (2025). SMFA: Strengthening multi-factor authentication with steganography for enhanced security. *IEEE Access, 13*, 38915–38932. https://doi.org/10.1109/ACCESS.2025.3545769

Saxena, N., Hayes, E., Bertino, E., Ojo, P., Choo, K.-K. R., & Burnap, P. (2020). Impact and key challenges of insider threats on organizations and critical businesses. *Electronics, 9*(9), 1460. https://doi.org/10.3390/electronics9091460

Silva, L., Pimentel, B., Duarte, B., Escarpini, R., Sousa, L., Cruz, N., & Silva, R. (2025). Accessibility by design: A systematic review of inclusive e-book standards, tools, and practices. *Sustainability, 17*(24), 11173. https://doi.org/10.3390/su172411173

Sissodiya, A., Chiquito, E., Bodin, U., & Kristiansson, J. (2025). Formal verification for preventing misconfigured access policies in Kubernetes clusters. *IEEE Access, 13*, 141698–141718. https://doi.org/10.1109/ACCESS.2025.3597504

Stoumpos, A. I., Kitsios, F., & Talias, M. A. (2023). Digital transformation in healthcare: Technology acceptance and its applications. *International Journal of Environmental Research and Public Health, 20*(4), 3407. https://doi.org/10.3390/ijerph20043407

Subreman, N. S. M. (2026). The impact of effective organisational communication on employees' productivity. *Electronic Journal of Business and Management, 8*(4). https://doi.org/10.65136/ejbm.v8i4.67

Szymoniak, S., Piątkowski, J., & Kurkowski, M. (2025). Defense and security mechanisms in the Internet of Things: A review. *Applied Sciences, 15*(2), 499. https://doi.org/10.3390/app15020499

Weerasinghe, L. D. S. B., & Perera, I. (2022). Evaluating the inter-service communication on microservice architecture. In *2022 7th International Conference on Information Technology Research (ICITR)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICITR54757.2022.9992945

Zhu, D., Zhou, Z., Li, Y., Zhang, H., Chen, Y., Zhao, Z., & Zheng, J. (2025). A survey of data security sharing. *Symmetry, 17*(8), 1259. https://doi.org/10.3390/sym17081259

Appendix