DESIGN AND IMPLEMENTATION OF ELECTRONIC THESIS INFORMATION MANAGEMENT SYSTEM

CHAPTER ONE

# INTRODUCTION

* 1. **Background of the study**

Embracing a modern and efficient management system for university theses is an urgent and exciting opportunity waiting to be seized. The increasing number of students and research staff in universities has placed a heavy burden on the existing thesis management system, which is often characterized by manual processes, delays in document submission and retrieval, and difficulties in accessing and sharing information. This has resulted in an inefficient and cumbersome system, which has adversely affected the academic performance of the university.

The current system of thesis management is also characterized by a lack of transparency and accountability. In many universities, thesis submission and review processes are often done manually, with no centralized system in place to ensure accuracy and precision. This has led to discrepancies and errors in the submission and review of theses, resulting in delays in their completion and publication.

In view of these issues, the need for a comprehensive and effective system to manage thesis has become more pressing. This includes the need for an electronic thesis information management system (ETIMS) that would provide a centralized system for the submission, review, and publication of theses. An ETIMS would allow for the easy access and sharing of information, as well as the efficient tracking of thesis submitted by students and research staff. It would also provide a platform for the easy and accurate submission, review, and publication of thesis.

In light of the challenges posed by the current system of thesis management, the university is in urgent need of an efficient and effective ETIMS. This project therefore seeks to design and implementation of an ETIMS in the university. The study will consider the current system of thesis management, the challenges associated with it, a1n1d the potential benefits of an ETIMS. The study

will also analyze the various design considerations and implementation strategies that must be taken into account when developing an ETIMS.

# Problem Statement

Glisson and Chowdhury (2002) conducted a study that revealed that traditionally stored dissertation documents contain minimal information about the student's thesis. The document usually includes the topic, the student's name, and the supervisor's name. This information is entered as soon as the student submits a project title to the department and a supervisor is assigned. Afterward, the supervisor and student work independently: in most cases, the student sends draft chapters to their supervisor and receives feedback with corrections and comments. The documents are sent back and forth either through email attachments or in printed format. Printed copies of communication, such as emails between the student and supervisor, and notes from the supervisor, are usually kept in hard copy files for each student. After the student has defended their dissertation, a hard copy is kept in the library.

Tidline and Perry (2009) delve into the possibilities for change that Electronic Thesis management system (ETIM) offer. Their study emphasizes the role of an Electronic Thesis and Information Management System (ETIMS) in streamlining the submission and review process, enhancing document tracking, and fostering collaboration between students and supervisors. The study also emphasized that traditional methods of submitting and reviewing dissertation documents often involve time-consuming and cumbersome manual procedures. However, they recommended an ETIMS can automate these processes, allowing students to submit their work electronically and supervisors to provide feedback in a more efficient manner. This not only saves time but also ensures that the submission and review cycle is more streamlined and transparent.

The study will examine the current system of thesis management, the issues associated with it, and the potential advantages of an Electronic Thesis and Information Management System (ETIMS).

It will also analyze the various design considerations and implementation strategies that must be taken into account when creating an ETIMS.

* + 1. Research Questions
       - What is the best methodology used in the development of the current system
       - What is the workflow for thesis management
       - How is thesis information propagated to supervisors and students?
       - What are the management policies for thesis?

# Research objectives

* To create a standard system fitted with the essential functions for theses management
* To develop a system that can minimize the delays in thesis management
* To develop a system that can generate reports of students’ theses
* To develop a system that enables communication between supervisor and students

# Significance of the Study

The purpose of this project is to utilize a web base thesis submission to help students and supervisors monitor the progress of the thesis to complete their study. The system created will enhance the workflow of creating and overseeing theses. It will provide the research panel with the origin of the research thesis and other data that can be used to approve the student’s thesis. A web-based system will enable administrators and student supervisors to track the status of a proposal from the beginning to the end, making the procedure of proposal development, review and submission very transparent.

# Scope of study

Just as any researcher is bonded and restricted to encounter some limitations, this project is constrained by the following limitations:

1. The system developed will not be integrated with other school management system due to limited time available for the project.
2. Inadequate finance limited the information and timely completion of the work.
3. Head of department will able to assign student to supervisors
4. The system will have a chat feature for communication
5. Supervisor will approve student for defense.

# Organization of the study

The entire research project is divided into five chapters; a summary of the content contained in each chapter is briefly described below:

# Chapter One - Introduction

This chapter presents an outline of the research and it covers the background of the research, research problem, objectives of the study, research questions, significance of the research and the organization of the study.

# Chapter Two - Literature Review

Literature relevant to the study is reviewed in this chapter. Literature on electronic Thesis management systems. Related work to this research is also presented in this chapter.

# Chapter Three - Methodology

This chapter elaborates on the research methodological approach and highlights the research strategy and paradigms.

* **Chapter Four - System design and analysis:** Research/Survey Design; Data Collection and Analysis; Experimental Design and Analysis.
* **Chapter Five - System Implementation and Testing.** Implementing this system will involve testing (unit, system and user acceptance).

CHAPTER TWO

LITERATURE REVIEW

# Introduction

literature review is an essential part of any research project, it involves examining existing research materials related to the topic at hand, with a goal of identifying gaps in knowledge or areas that need further investigation.

# Definition of Information System

Electronic Thesis Information Management System, as an information system, is a crucial strategic resource that profoundly impacts key operations and determines the success of academic institutions.

Information Systems have extensive definitions, with Sauer (1993) arguing that some view it as an expense while others consider it as a solution or control mechanism. It can also be perceived as a technical problem or a threat to the quality of work-life.

Laudon & Laudon (2002) define Information Systems as a collection of interrelated components that gather, process, store, and disseminate information to facilitate decision-making, coordination, and control within an organization.

In contrast, Martin (1992) considers Information Systems as a valuable resource in organizations. Information Systems are present in all academic institutions, with varying degrees of sophistication. Some rely on manual paper-based systems, while others use spreadsheets and locally-run databases, such as Microsoft Access.

Carl Marnewick (2016) , argues that the use of an information management system will greatly improve the benefits for the organization and offer a number of advantages in managing business processes. An information system can alter business procedures to make them more effective and efficient while also encouraging improved management inside the organization.

Alternatively, centralized online databases are favored by some institutions to manage data related to their students. Understanding the broader concept of Information Systems is critical in the application of Electronic Thesis Information Management System in academic institutions.

# Related Works

Electronic thesis information management systems (ETIMS) have become increasingly popular in recent years due to the growth in digital technology and the need for improved access to scholarly resources. Advances in technology have made it easier to create and manage electronic resources, including theses and dissertations. With an ETIMS, universities can more efficiently manage and disseminate electronic theses, reducing the time and effort required for manual management. Additionally, ETIMS can also provide improved access to scholarly resources, including theses and dissertations. Researchers and the general public can benefit from online access to these resources, which can help to further enhance academic research and scholarship.

Several studies have examined the use and effectiveness of ETIMS. Chen and Chang (2017) conducted a comprehensive study to evaluate the effectiveness of a customized ETIMS in managing the submission and management of electronic theses. The authors found that the system was effective in improving the efficiency of the thesis submission and management process, reducing the workload for staff involved in thesis management, and improving data accuracy. By eliminating manual processes, the ETIMS reduced the chances of errors and inconsistencies, thus improving the overall quality of the data.

Similarly, In the study conducted by Liu and Liu (2019), the adoption and use of ETIMS in Chinese universities were examined. The authors found that the system was widely used in these universities and was perceived as being effective in facilitating the thesis submission and management process. ETIMS were found to be particularly useful in terms of reducing the workload of staff involved in thesis management and increasing user satisfaction. The system also

improved the quality of data and information by making it easier to access and manage electronic resources.

In another study, Rueda-Medina et. al (2019) evaluated the use of ETIMS at a Colombian university. They found that the system was effective in improving the visibility and accessibility of electronic theses to researchers and the general public. This was made possible by the system's ability to digitize and manage theses in a more efficient and reliable manner. The system also provided users with an improved user experience, making it easier for them to access and search for electronic resources.

Several studies have also examined the impact of ETIMS on institutional repositories. Zhou and Li (2018) conducted an evaluation of the use of ETIMS in a Chinese university repository. They found that the system was effective in improving the efficiency of the thesis submission and management process. This was accomplished through the features of the system, which simplified the submission process and provided automated management tools. The system also had a positive impact on the visibility and accessibility of the repository's electronic theses, making it easier for users to locate and access resources. Overall, ETIMS have been shown to be effective in improving the efficiency and effectiveness of the thesis submission and management process, as well as increasing the visibility and accessibility of electronic theses.

# Benefits of Using ETIMS

ETIMS offers universities and researchers a variety of benefits, such as increased efficiency in managing and disseminating electronic theses, as well as greater access to scholarly resources. According to a study by Alzahrani et al. (2020), using an ETIMS makes it easier for researchers, students, and faculty members to access and share research, and can lead to better citation rates for the institution and its researchers.

ETIMS can also help to improve the quality and visibility of electronic theses. By ensuring electronic theses are properly formatted, reviewed, and archived, institutions can provide valuable resources to the academic community. Better visibility can also lead to better citation rates for the institution and its researchers.

Another important benefit of ETIMS is that it helps to reduce the amount of physical space required to store traditional thesis copies. This leads to cost savings for institutions that would otherwise need to provide expensive storage facilities. As an example, a study by Suroso et al. (2019) found that implementing an ETIMS at an Indonesian university lead to a reduction in storage costs of 43%.

# Types of ETIMS

There are several types of electronic thesis information management systems (ETIMS) available, each with its own unique features and benefits

# Commercial ETIMS

These are commercial platforms designed to serve academic institutions, students, and researchers. Some examples include ProQuest ETD Administrator and Ebsco's ETD Service. These platforms feature advanced search and indexing capabilities, customized workflows, and compliance with industry standards. They also offer a range of value-added services such as plagiarism detection, document conversion, and archival storage. Commercial ETIMS are usually subscription-based, and institutions can choose from different pricing plans based on their usage requirements.

# Open-Source ETIMS

These are free and open-source platforms that offer a viable alternative to commercial ETIMS. Some examples of open-source ETIMS include DSpace, EPrints, and Open Repository. These platforms are highly customizable and offer a range of features, including metadata management, content discovery, and digital preservation. They are also compatible with several metadata

standards, making it easier to integrate with existing systems. Open-source ETIMS have a vibrant user community and offer a range of online resources and support forums.

# Cloud-Based ETIMS

These are cloud-based platforms that offer flexible and scalable solutions for managing electronic theses and dissertations. They are usually designed to meet the needs of small to medium-sized academic institutions that do not have dedicated IT infrastructure for managing electronic theses. Examples include AlmaDMS, Ex Libris Cloudy, and Library Simplified. Cloud-based ETIMS are easy to set up and use, with minimal hardware and software requirements. They offer a range of features, including metadata management, document upload and retrieval, and user access control. They also provide data backup and disaster recovery services, ensuring the safety and security of electronic theses and dissertations.

# Existing Systems

Several ETIMS systems have been developed and implemented in various institutions. Some of the well-known systems include:

# ProQuest Dissertations and Theses (PQDT)

PQDT is a commercial ETIMS system that provides access to over six million graduate works. The platform allows authors to submit and publish their works, and users can access the documents through search capabilities. However, the system has been criticized for its high subscription fees, which limit access to quality research (ProQuest, 2021).

# Electronic Thesis and Dissertation Submission System (ETD)

ETD is a customizable system that allows institutions to manage the submission and approval processes of electronic theses and dissertations. The platform integrates with the institution's library system, providing access to the works through various channels. However, some users have

reported technical difficulties in the submission of their works, leading to delays in the approval process (University of Pittsburgh, 2021).

# OpenETD

OpenETD is an open-source ETIMS system that allows institutions to manage their electronic theses and dissertations. The platform is customizable and easy to use, providing various functionalities such as archiving, tracking, and search capabilities. However, the system lacks features such as plagiarism checkers and peer-reviewing, creating loopholes for the submission of poor-quality research works (University of North Texas, 2021).

# Gaps In Existing ETMS Systems

Despite the benefits of the ETIMS systems, some gaps have been identified. Firstly, most of the systems lack standardization, making it difficult for users to navigate the platforms. This can be attributed to the diverse needs of different institutions, leading to the development of customized systems that lack compatibility with other institutions (Ahmad *et al.,* 2019).

Secondly, plagiarism is a significant issue in the submission of electronic theses and dissertations. Most of the systems lack inbuilt plagiarism checkers, creating loopholes for the submission of plagiarized work, which undermines the quality of the research (Nguyen, 2014).

Thirdly, technical issues such as server downtime and system incompatibility can cause delays in the submission and approval processes, leading to inconveniences to the users. This can be attributed to the heavy traffic on the systems, leading to slow response times and system failures (Nguyen, 2014).

Finally, limited access to quality research has been identified as a major defect of the ETIMS systems. Most of the systems have high subscription fees, making it difficult for users to access quality research. This issue has been noted in the PQDT system, limiting access to research works that are crucial in shaping scholarly discussions (ProQuest, 2021).

ETIMS systems have revolutionized scholarly work management, providing various functionalities that enhance the ease of managing electronic theses and dissertations. However, several defects such as plagiarism, technical issues, limited access to quality research, and lack of system standardization have been identified. Institutions and developers need to address these shortcomings and improve the systems' functionalities to provide quality research work.

CHAPTER THREE

THE PROJECT METHODOLOGY

# Introduction

This section elaborates on the methodology applied in the research project.

Software development methodology, also known as a system development methodology or software development life cycle, refers to the process of dividing software development work into distinct phases or stages. The primary goal is to improve planning and management. Typically, it is considered a subset of the broader systems development life cycle. This methodology may involve pre-defining specific deliverables and artifacts that a project team creates and completes during the development or maintenance of an application.

There are several common methodologies, including waterfall, prototyping, iterative and incremental development, spiral development, rapid application development (RAD), extreme programming, and various types of agile methodologies. Some people use the term "life cycle model" as a more general term to describe a category of methodologies, while "software development process" refers to a specific process chosen by a specific organization. For instance, there are numerous software development processes that align with the spiral life cycle model.

In the context of this project, the chosen software development methodology is RAD (Rapid Application Development). RAD enables rapid development and delivery of high-quality systems with relatively greater ease of making changes during the development process. It also allows for quick system design changes in response to user demands and ensures a closer alignment between user requirements and system specifications. Additionally, RAD significantly saves time, money, and human effort.

# RAD (Rapid Application Development)

Rapid application development (RAD) is both a general term used to refer to alternatives to the conventional waterfall model of software development. In general, RAD approaches to software development put less emphasis on planning and more emphasis on process. In contrast to the waterfall model, which calls for rigorously defined specification to be established prior to entering the development phase, RAD approaches emphasize adaptability and the necessity of adjusting requirements in response to knowledge gained as the project progresses. Prototypes are often used in addition to or sometimes even in place of design specifications.

The RAD model would enable one to create this solution without the need to carry out prolonged planning activities since the specifications are well known.

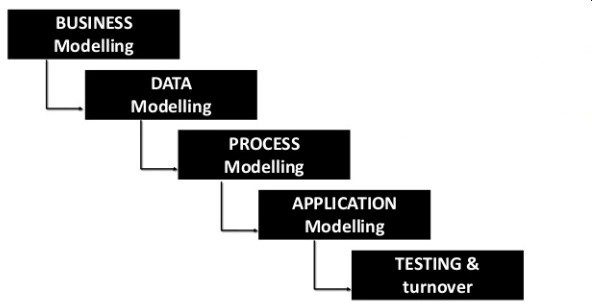


Figure 3.1 : RAD model

The phases in the rapid application development (RAD) model are:

# Business modeling:

This is the phase of the project in this model. Its primary objective is to comprehend the workflow requirements. This entails examining the current ETMs (Electronic Thesis Management System) to understand how it functions by observing workflows and conducting interviews with relevant entities. Additionally, it involves analyzing the crucial factors that contribute to the seamless flow of information.

# Data modeling:

The second phase would be to analyze the information gathered in the business modeling phase from a data point of view. Each workflow is carefully analyzed to determine the necessary data, its usage, and how and where it flows. The attributes that define the data objects are clearly outlined, and relationships between different data objects are established. This phase focuses on understanding the data requirements and structure to ensure effective data management and utilization throughout the project.

# Process modeling:

the data object model created is expanded to include the processes that are performed on each object. These processes may involve adding, retrieving, modifying, or even deleting data or specific attributes within the objects. Descriptions are identified and created for

each of these data object processes, commonly referred to as CRUD (Create, Read, Update, Delete). This provides a comprehensive understanding of how data objects are manipulated within the system.

# Application generation:

In the fourth phase of the RAD (Rapid Application Development) system development model, the process model and data model are transformed into a functional solution. This phase includes coding the software using established automation tools to create the application and database. Additionally, it involves integrating and configuring the ETMs (Electronic Thesis Management) system.

# Testing and turnover:

This phase is the last of the model, this entails conducting thorough testing to ensure the solution performs as expected and meets the requirements. After successful testing, the prototype is handed over to the relevant stakeholders for further evaluation and implementation.

# Pros And Cons Of Rapid Application Development

In modern Information Technology environments, many systems are now built using some degree of Rapid Application Development.

* + 1. The advantages of RAD
       - **Better quality:** involving users in the process of interacting with evolving prototypes often results in significantly improved business functionality compared to using a waterfall model. The software becomes more user-friendly and is better able to address the crucial business problems that end users face, rather than solely focusing on technical issues that may be of interest to developers.
       - **Risk control:** RAD allows for early identification and focus on key risk factors, which can then be adjusted based on empirical evidence gathered during the initial stages of the process.

For example, the complexity of prototyping the most complex parts of the system can be addressed early on. This approach contributes to a higher rate of projects being completed within their intended time frame and budget. By prioritizing the development of

incremental units, RAD reduces the likelihood of catastrophic failures that have often plagued large waterfall projects.

* + 1. The disadvantages of RAD

While Development (RAD) has numerous advantages, there are some potential disadvantages to consider:

* + - * Increased complexity: The iterative and rapid nature of RAD can lead to increased complexity in the development process. Managing multiple iterations and frequent changes can make it challenging to maintain a clear and consistent overall project structure.
      * scope creep: With RAD, there is a higher risk of scope creep, where additional requirements or changes are introduced throughout the development process. This can lead to scope expansion, increased development time, and potential budget issues if not carefully managed.
      * Dependency on user availability: RAD heavily continuous user involvement and feedback. If users are not available or do not actively participate, project progress can be hindered, leading to delays and decreased effectiveness of the RAD approach.
      * RAD Often demands higher skilled resources, including designers, developers, and domain experts. This can increase costs, both in terms of personnel and the use of automated code generating tools.
      * Limited suitability for large-scale projects: RAD is generally better suited for projects with a narrow focus. For large-scale, complex projects, the iterative nature of RAD may prove challenging to manage and integrate effectively.
      * Potential for quality trade-offs: The emphasis on speed and rapid development in RAD may in some compromises on quality. With shorter development cycles, thorough testing and quality assurance processes can sometimes be overlooked, potentially leading to a lower quality end product.

# When to Use RAD Model

RAD, or Rapid Application Development, is most suitable when there is a requirement to develop a system that can be modularized within a relatively short timeframe of 2-3 months. Additionally, it is recommended to use RAD when there is a sufficient availability of designers who can perform the necessary modeling tasks. It is also important to consider the budget, as RAD may require the utilization of automated code generating tools, which can incur additional costs.

Moreover, the RAD model is ideally chosen when there are resources with extensive business knowledge available. This is because RAD emphasizes the involvement of such users in the development process to ensure that the system effectively meets the specific business requirements. Lastly, RAD is advantageous when there is a pressing need to deliver the system within a short timeframe of 2-3 months.

# Project Plan

The following are a set of project management considerations that would serve as guide as to how to tackle the project from start to finish. The approaches chosen would allow for clear and easy identification of tasks, processes, requirements, categorizations and breakdowns of every step taken. This would offer a clear picture of what activities the project would entail as well offer an avenue of tracking the progress of work.

Table 3.1 Project Dependency

|  |  |  |
| --- | --- | --- |
| Tasks, Duration and Dependencies | | |
| **Task** | **Duration(weeks)** | **Dependencies** |
|  |  |  |
| Business Modeling | 3 |  |
|  |  |  |
| Data Modeling | 3 | BM |
|  |  |  |
| Process Modeling | 4 | BM,DM |
|  |  |  |
| Application Generation | 6 | PM |
|  |  |  |
| Testing | 3 | AM |
|  |  |  |
| Deployment And Rollout | 2 | TE |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Project Phase Schedule | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| PROJECT PHASE | WEEKS | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| **Business modeling** |  | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Data modeling** |  |  | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Process modeling** |  |  |  |  | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Application generation** |  |  |  |  | | | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |  | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Deployment And Rollout** |  |  |  |  |  |  |  |  |  |  |  | |

Figure 3.2 Project schedule

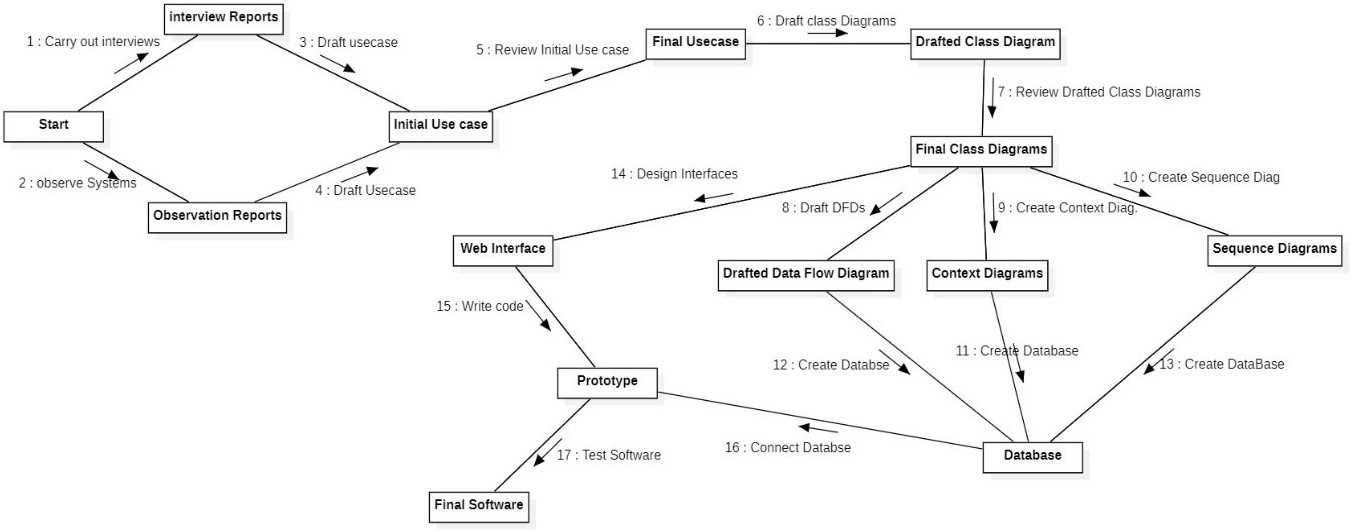


Figure 3.3 Project flow

CHAPTER FOUR

# SYSTEM DESIGN, IMPLEMENTATION, TESTING, RESULTS AND ANALYSIS

* 1. **Introduction**

This chapter deals with the design and analysis of the proposed Electronic Theses Management System. System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well running system.

# Analysis of the Current System

Effective thesis management requires good organizational and time-management skills, as well as clear communication with supervisors and other stakeholders. It's essential to stay on track with deadlines and milestones to ensure the successful completion of the thesis project. The existing method of thesis management tend to be time-consuming.

Three main parties are involved in managing a final-year thesis: students, supervisors, and the head of department. The department head assigns a Supervisor to guide each student, and after which it becomes the joint responsibility of the student and the supervisor to oversee the progress of the thesis and report writing. The procedures for monitoring can differ among supervisors, who also serve as intermediaries between students and the department head's office.

Currently, logbooks are used to monitor the progress of student thesis and to record all activities during supervision. Students are required to meet with their supervisors at least five times throughout the semester, with each meeting recorded in writing and facilitated by the supervisors. This could be done electronically on one platform which will ensure real time monitoring

The center of dissertation management is the department head, responsible for hiring supervisors for students, compiling dissertation proposal documents, editing dissertation submissions, compiling a final dissertation document, and inputting student grades into a student management system referred to as ICampus.

The users interviewed recommended that the proposed system should be user friendly, multipurpose enough to handle a number of users at a go, could generate feedback when request is submitted and a use of passwords which could deny access to unauthorized users of system which ensured security. Context diagrams, Data flow diagrams and Entity Relationship Diagram (ERD) were used in the analysis and design of the system.

# Requirement Specification

After analyzing the data collected, a number of requirements are formulated namely user requirement, system hardware and software attribute. These were grouped as user, functional, nonfunctional and systems requirements.

# User Requirement

During data collection, an investigation found out how the current system operates, not only that but also tried out which problems are faced and how best they can be settled. The users described some of the basic requirements of the system this includes:

# User Authentication and Authorization:

* + - * + Users should be able to register and create accounts.
        + The system should support different user roles (students, faculty, administrators) with varying levels of access and permissions.
        + Users should log in securely using authentication methods.

# Thesis Submission:

* + - * + Students should be able to submit their theses electronically.
        + The system should allow uploading of thesis documents in various formats (PDF, Word, etc.).

# Approval Workflow:

* + - * + Faculty and administrators should be able to review and approve submitted theses.
        + Notifications should be sent to relevant parties when a thesis is submitted, reviewed, or approved.
        + The system should support approval workflow based on the institution's requirements.

# Plagiarism Detection:

* + - * + The system should integrate with plagiarism detection tools to check submitted theses for originality.
        + Plagiarism reports should be generated and made available to reviewers.

# Document Management:

* + - * + Uploaded theses should be stored securely and organized for easy retrieval.
        + Different versions of the same thesis should be maintained.
        + Archiving functionality should be available for completed theses.

# Search and Retrieval:

* + - * + Users should be able to search for theses based on various criteria (author, title)
        + Search results should be displayed in a user-friendly format.

# Functional And Non-functional Requirements

**Functional Requirements:**

The following is the desired functionality of the new system.

# User Authentication and Authorization:

* + - * + Users must be able to register and log in securely using their credentials.
        + Different user roles (student, faculty, administrator) must have distinct permissions and access levels.

# Thesis Submission:

* + - * + Students should be able to upload their theses in PDF format.
        + The system should validate the file format and size during upload.

# Approval Workflow:

* + - * + Faculty and administrators must have the ability to review, comment on, and approve or reject submitted theses.
        + The system should support sequential or parallel approval workflows.

# Plagiarism Detection:

* + - * + The system must integrate with a third-party plagiarism detection tool.
        + Plagiarism reports should be generated and presented to reviewers during the approval process.

# Document Management:

* + - * + Uploaded theses must be stored securely and accessible only to authorized users.
        + The system should allow version control for thesis documents.

# Search and Retrieval:

* + - * + Users must be able to search for theses using keywords, author names, or titles.
        + Search results should be displayed in a clear and organized manner.

# User Profile Management:

* + - * + Users should be able to edit their profiles, update contact information, and provide areas of expertise (faculty).

# Reporting and Analytics:

* + - * + Administrators should have access to reports on thesis submission statistics, approval rates, and user engagement.
        + Data visualization tools must be used to create informative graphs and charts.

# Non-Functional Requirements:

1. **Usability:**
   * The user interface should be intuitive and user-friendly to accommodate users with varying technical backgrounds.
   * Response times for actions and page loads should be within an acceptable range to maintain user engagement.

# Security and Privacy:

* + The system must implement strong encryption for data storage and transmission.
  + User data and thesis documents must be protected from unauthorized access.

# Performance:

* + The system should support a large number of concurrent users without significant performance degradation.
  + Upload and download speeds for thesis documents should be optimal.

# Availability:

* + The system should have a high uptime percentage to ensure continuous access.
  + Regular maintenance and updates should be performed during off-peak hours to minimize disruption.

# Scalability:

* + The system should be designed to handle increased user and data loads in the future.
  + Scalability measures should be in place to accommodate potential growth.

# Compatibility:

* + The system should be compatible with modern web browsers and devices, including desktops, tablets, and mobile phones.

# Regulatory Compliance:

* + The system must adhere to relevant data protection and privacy regulations such as Data Protection Act, 2012 (Act 843)

# System Requirement Software Requirement

Table 4 .1 Software Requirement

|  |  |
| --- | --- |
| Operating System: | Ms Windows, Linux, macOS, Android |
| Browser: | Chrome, Mozilla Firefox, Edge browser |

# Hardware Requirement

Table 4 .2 Hardware Requirement

|  |  |
| --- | --- |
| Processor: | Intel care i3( 2.0 GHz) |
| RAM | 1012 MB Ram (minimum main  memory space recommended |
| Storage | 250GB minimum Hard Disk Space |

# System Design

After interpretation of the data, tables were drawn and process of data determined to guide in the implementation stage of the project. The tools, which were employed during this methodology stage, where use case diagrams ,sequence diagrams, context diagrams, activity diagrams, class and object diagrams. The design ensures that only allows authorized users to access the systems information.

# Context Diagram

A context diagram is a type of diagram used in system analysis and design to provide a high-level view of a system's boundaries and the interactions between the system and its external entities. It helps in understanding the scope of a system and the context in which it operates.

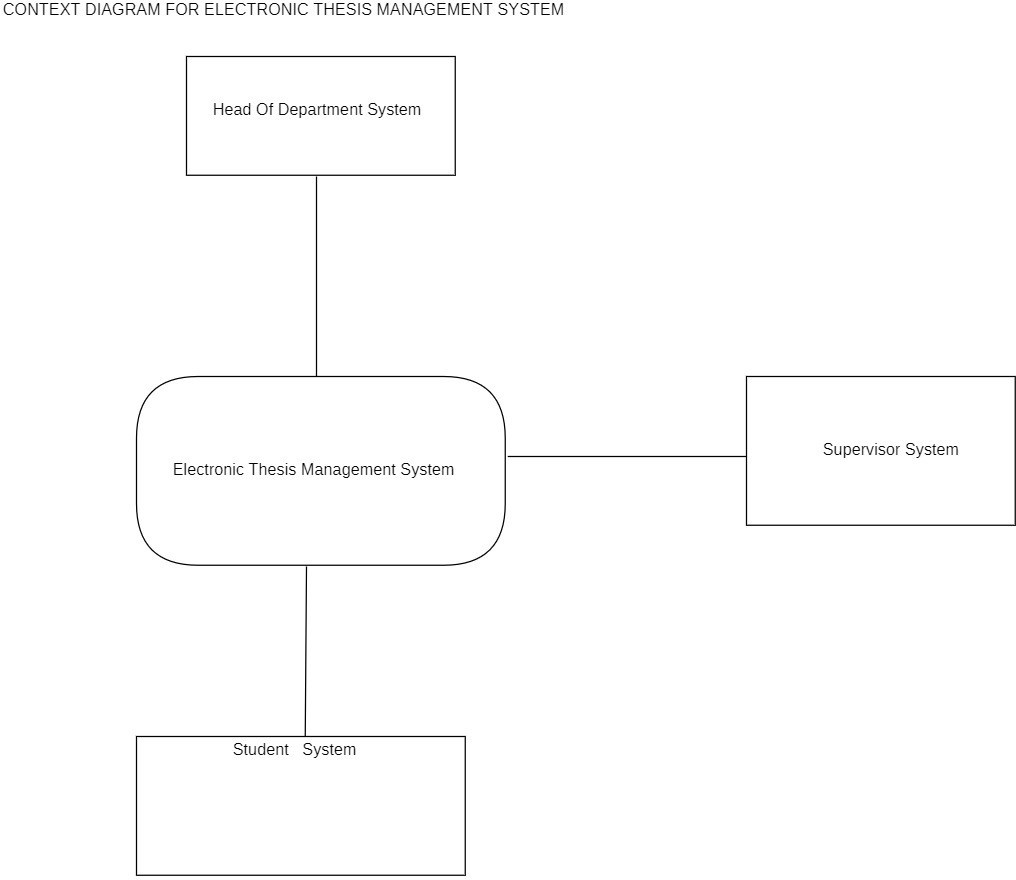


Figure 4.1 Context Diagram

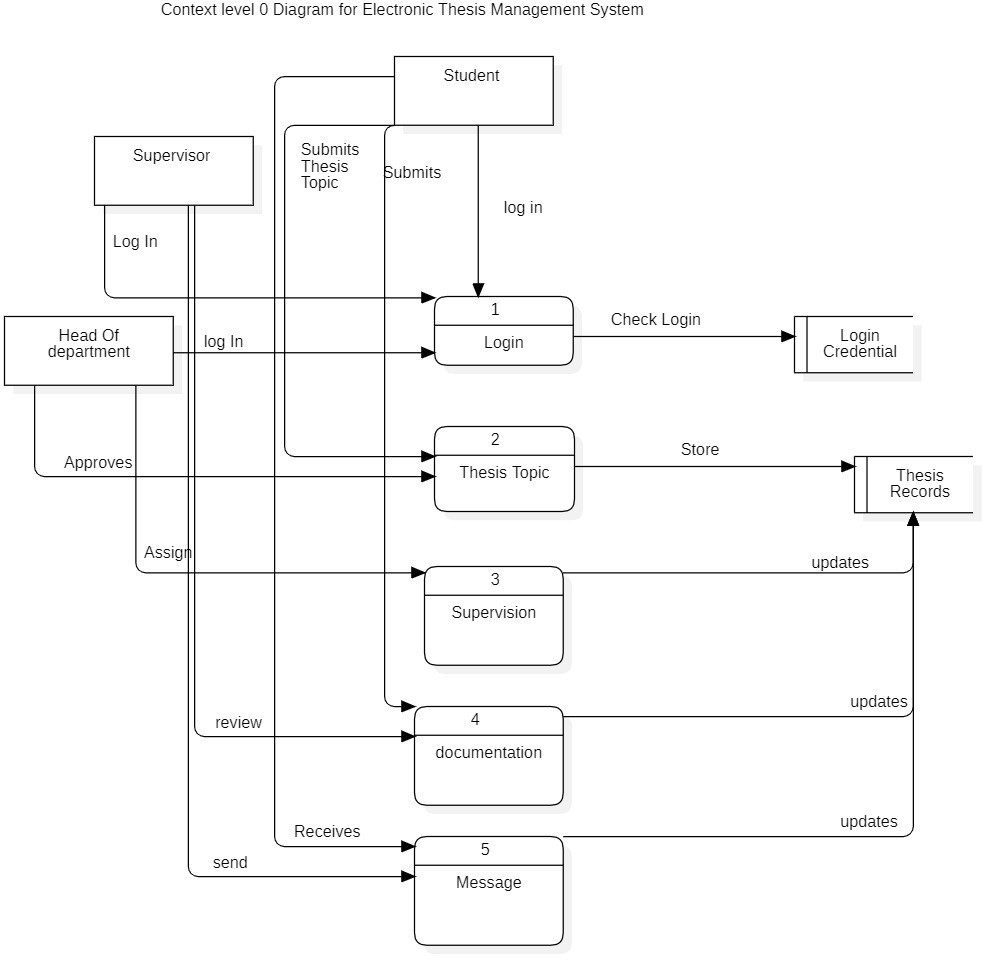


Figure 4.2 context level 0 Diagram

# Use case Diagram

Fowler (2018), defines use case as a diagram that visually represents the functional requirements of a system or application from a user's perspective. It's a high-level diagram used to illustrate how users interact with a system and what functionalities the system provides in response to those interactions.

In a use case diagram, actors (representing users, external systems, or entities) and use cases (representing specific interactions or tasks) are depicted as elements, and their relationships are shown through connectors. The use case diagram helps to outline the system's functionalities, the users who interact with those functionalities, and the specific scenarios or situations in which these interactions occur.

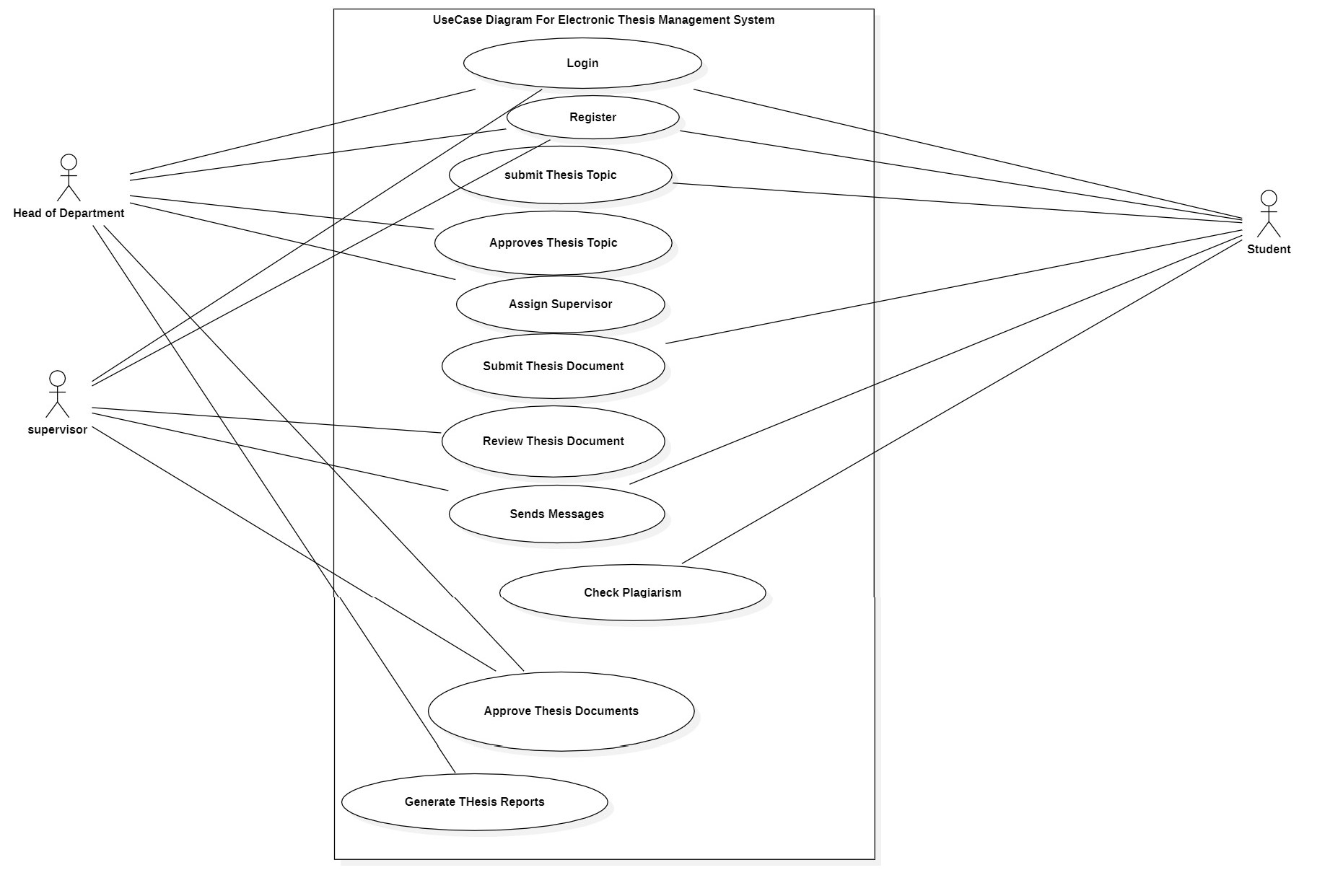


Figure 4.3 Use Case Diagram

# Sequence Diagram

sequence diagram provides a visual representation of the interactions between objects or components in the system.

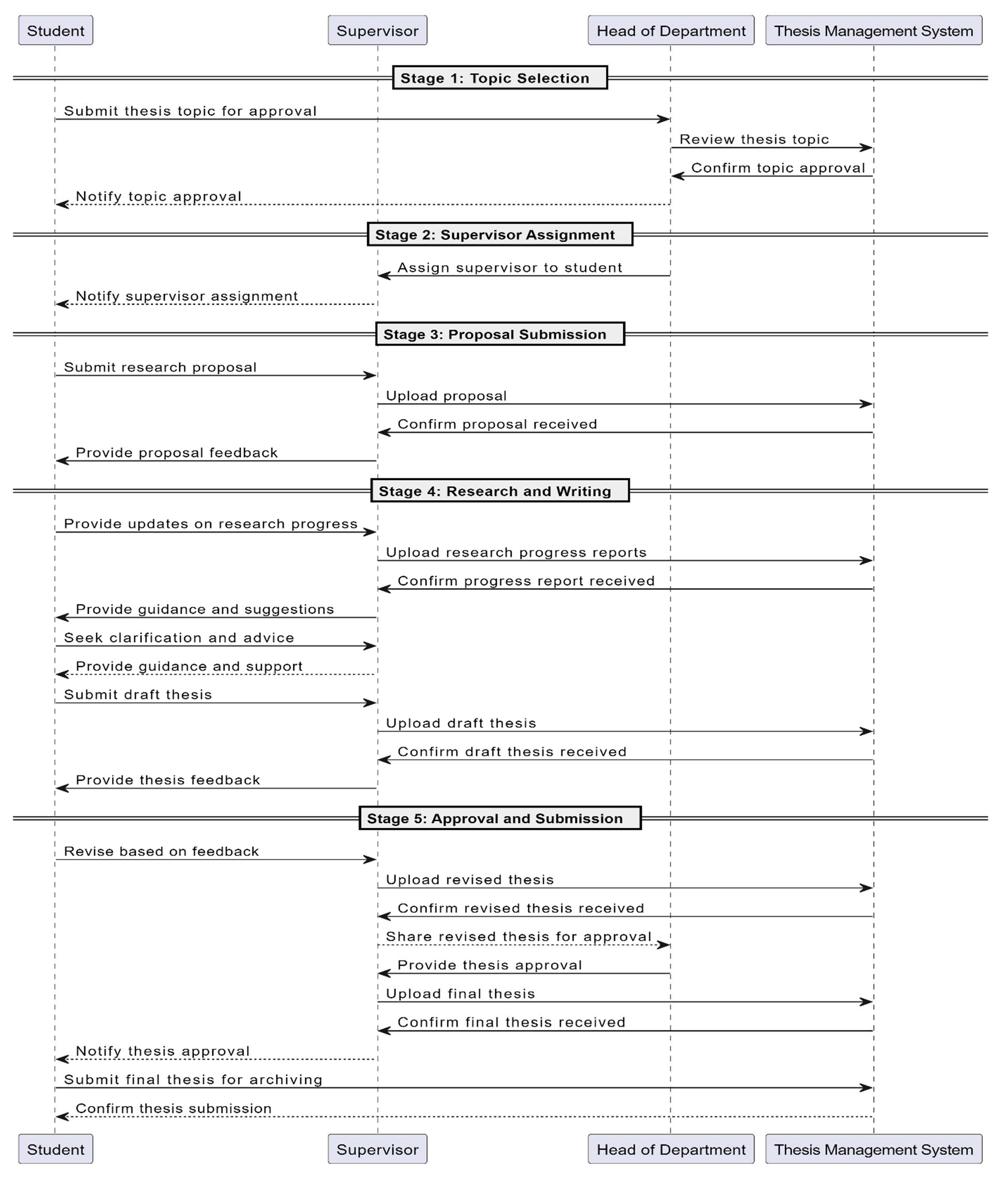


Figure 4.4 Sequence Diagram 1

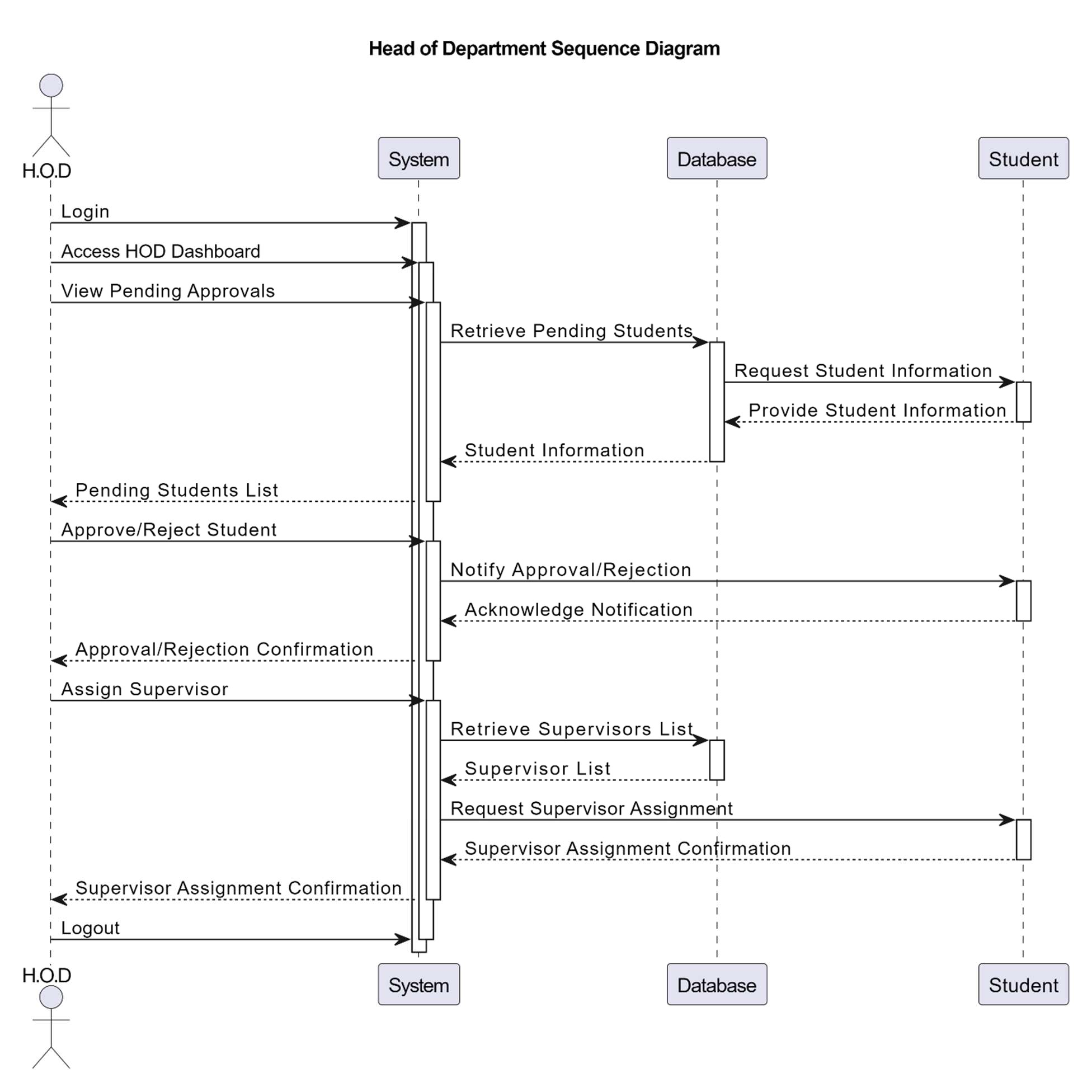


Figure 4.5 Head of Department Sequence Diagram

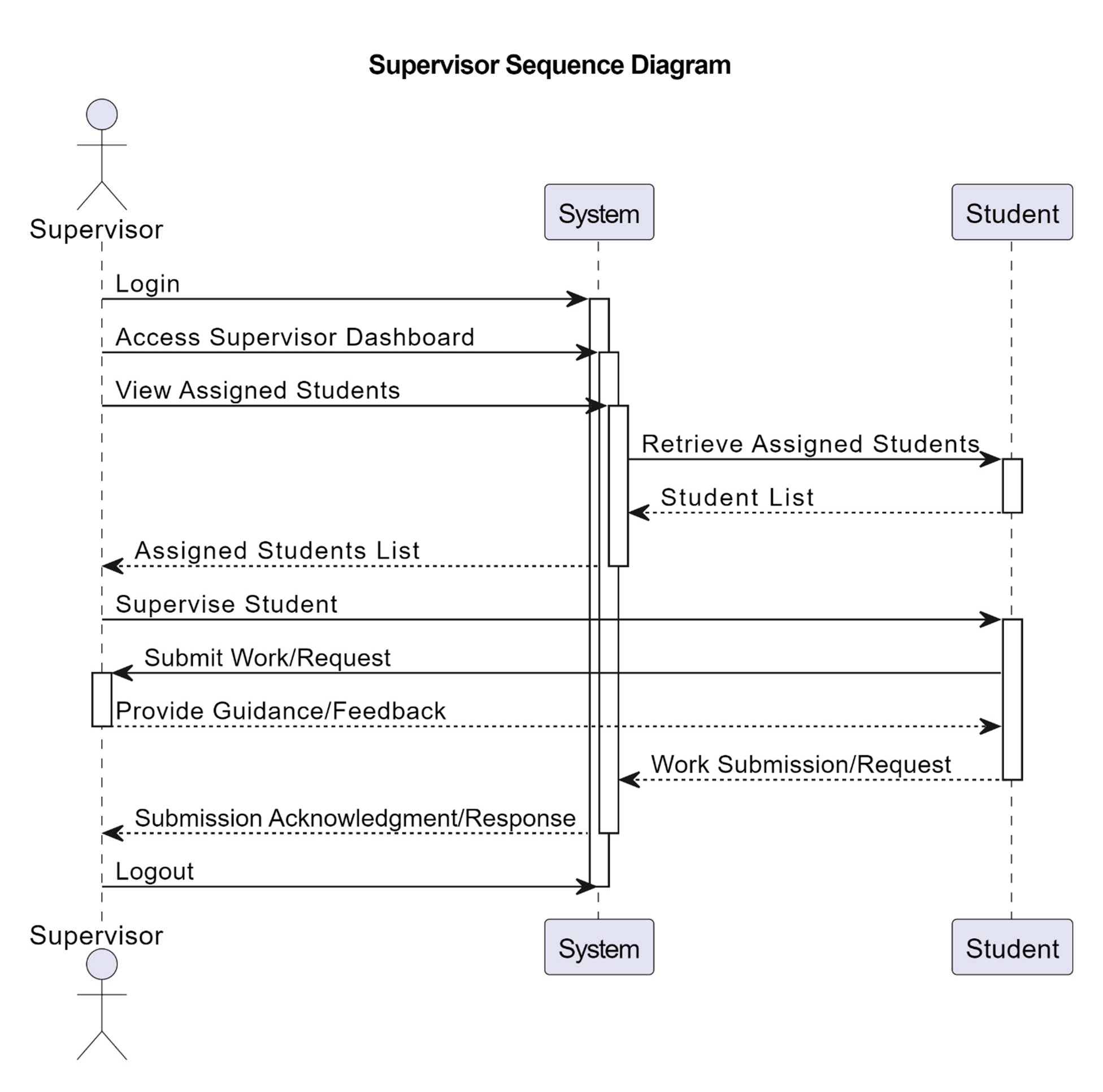


Figure 4.6 Supervisor Sequence Diagram

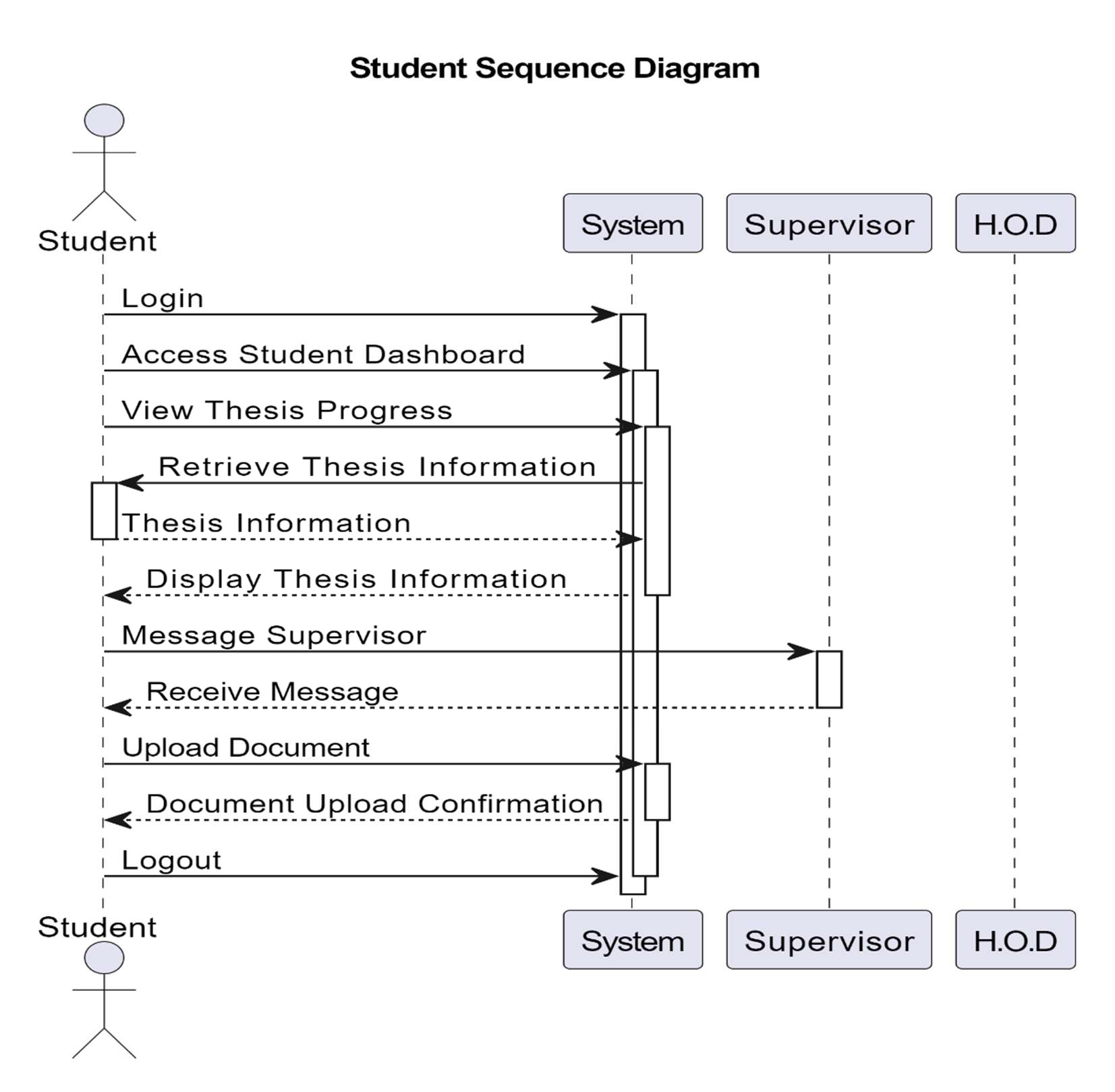


Figure 4.7 Student Sequence Diagram

# Class Diagram

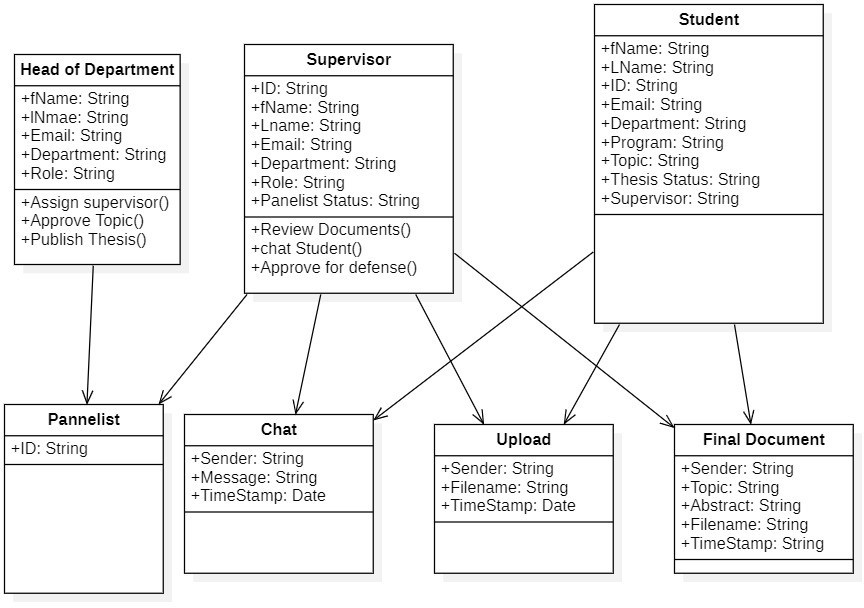


Figure 4.8 Class Diagram

# Object Diagram

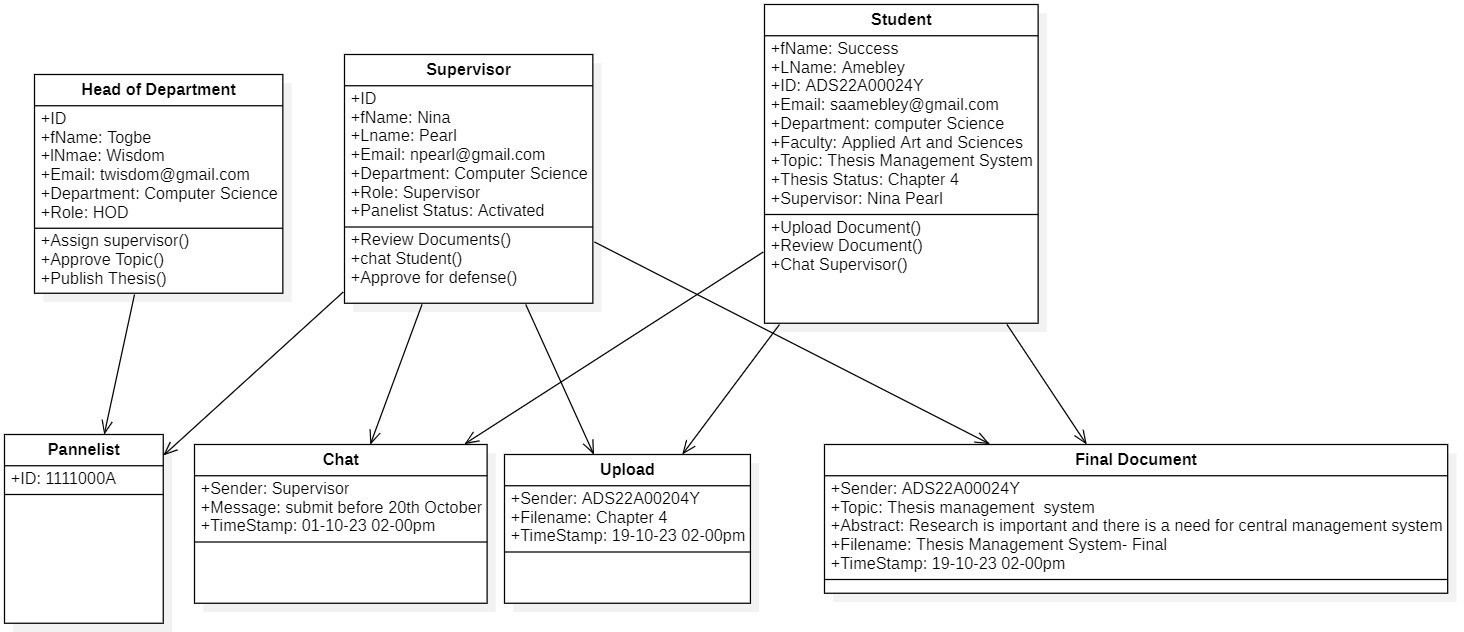


Figure 4.9 Object Diagram

# Entity Relationship Diagram

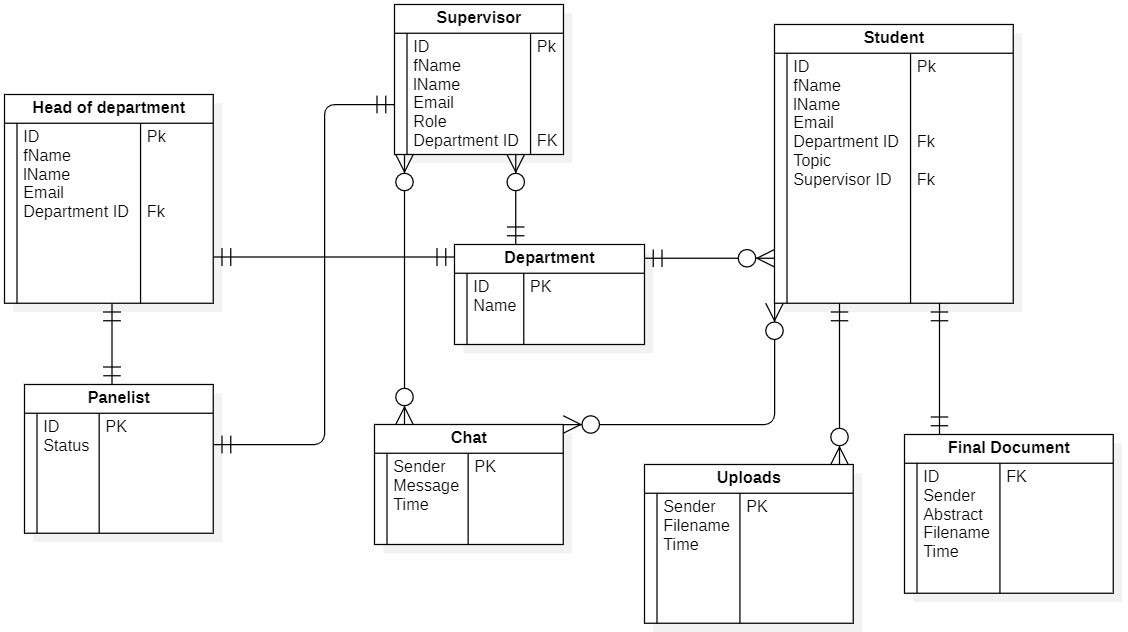


Figure 4.10 Entity Relationship Diagram

# System Implementation

The project use JavaScript scripting language for the development and adopted its Model View approach. The User Interface(UI); is enhanced using bootstrap framework with open source components and frameworks.

# Development Architectural Pattern

The architectural pattern that was used was the Model View Controller. It is an architectural pattern that categorizes an application into three logic components. These components are Models, View and Controller.

# Models

The Model category handles all data-related logic that the application uses. The data can be the type that is being transferred between the view and Controller component or other business logic- related data. For example, a supervisor object will retrieve the student’s information from the database, manipulate it and update it back to the database or use it to render data. This component performs Create, Read, Update and Delete (CRUD) to the database.

# Views

The View category is the User Interface part of the application. For example, the “registration” will include all the User Interface components such as text boxes, dropdowns, etc that the user interacts with. The View component is usually implemented using front-end technologies (HTML and CSS).

# Controllers

The Controller components serve as an interface between Model and View components to be able to process all the business logic and incoming requests, manipulate data using Model Component and interact with the Views to take input as well as render final output. For example, the passport controller will handle all the authentication and inputs from the user and update the database.

# Programing the system

Node.js and Express.js are used for the development of this system. Since Express.js is a popular web application framework for Node.js, our design and implementation also follow the same pattern. The system is built using the principles of Node.js and Express.js, which allows for efficient and scalable development.

In this project, various routes, controllers, models, and views are implemented. Express.js provides a routing mechanism that allows us to define different endpoints and handle HTTP requests accordingly. Each route is associated with a specific controller that handles the logic and data manipulation for that particular endpoint.

Models in this system are implemented using classes and are responsible for interacting with the database or any other data source. They encapsulate the queries and data manipulation operations. Views in this system represent the user interface and are responsible for rendering data to be presented to the user. Views are implemented using EJS templating engine, which allow for dynamic content rendering.

Overall, the system follows the Object-Oriented Programming paradigm and leverages the functionalities of Node.js and Express.js to provide a robust and scalable solution.

# System Testing

System testing, also referred to as end-to-end testing, is conducted to ensure that a fully integrated system meets its specified requirements. In this type of testing, various aspects of the system are examined together as a whole. For instance, a system test may involve evaluating the functionality of a login interface, performing tasks like creating and modifying entries, transmitting or printing results, carrying out summary operations or removing/archiving entries, and finally logging off. System testing can be categorized into two approaches: white-box testing and black-box testing. These approaches describe the perspective adopted by a test engineer while designing test cases.

* White-box testing: The tester has access to the source code and technical knowledge of the system being tested. The goal of white-box testing is to ensure that all paths and conditions within the system code are tested thoroughly. This includes testing individual functions,

branches, loops, and statements to validate their correctness and identify any potential errors or vulnerabilities.

* Black-box testing: The tester focuses solely on the inputs and outputs of the system without any knowledge of its internal code or implementation. The goal of black-box testing is to evaluate the system's functionality, its response to various inputs, and its compliance with specified requirements or specifications.

# Testing Strategy

Testing involves the following:

* Unit testing

The first level of testing is known as unit testing. Unit testing aims to thoroughly test each individual program to guarantee its complete functionality.

* Integration testing

The second stage is integration testing. In this units components are integrated and tested as a complete system to ensure that the requirements are met.

* Acceptance testing

Acceptance testing is a crucial stage in the software development process. It involves the planning and execution of different types of tests to demonstrate that the implemented software system meets the specified requirements. By conducting acceptance testing, I aim to validate that the software system functions as intended and satisfies the needs of the end-users.

**CHAPTER FIVE**

**FINDINGS, CONCLUSION AND RECOMMENDATIONS**

* 1. **Introduction**

This chapter will review and summarize the study of the project and introduce system contribution, maintenance problem, future works and recommendations for the system.

# Summary

The Electronic Thesis Management System is designed to streamline and simplify all stages of the thesis writing process. This system enables students to easily submit their thesis topics for approval from the head of the department and subsequently assigns a supervisor to provide guidance throughout the writing process. Students can also submit research proposals, receive feedback, and track their progress. Supervisors can review and provide feedback on drafts, ensuring the quality of the final thesis. With this system, students can efficiently complete their theses and receive timely approvals, while supervisors and the head of the department have a centralized platform to manage and track the entire process. Overall, this system optimizes the workflow for all participants involved in the thesis writing process

# Maintenance

System maintenance involves modifying and updating the Electronic Thesis Management System after its initial delivery. These changes can range from simple code corrections to more extensive modifications to address design errors, enhance specifications, or accommodate new requirements. There are three main types of system maintenance:

* + - **Fault Repairs:** This type of maintenance focuses on addressing errors or issues in the system. Repairing a requirements error can be costly as it may require substantial redesign, while coding errors are usually less expensive to correct. For instance, users are not able to login with their credentials, with this one has to recheck for bugs and fix them.
    - **Environmental Adaptation:** Maintenance of this nature becomes necessary when changes occur in the system's environment, such as hardware or operating system upgrades. In these cases, the system is adjusted or modified to adapt and perform seamlessly within the updated environment. Instances when the server breaks down, a new server is setup, the ETMs need to be modified to adapt the new server configurations.
    - **Functionality Addition:** Changes in system requirements due to organizational or business needs may require functionality additions. This type of maintenance often involves significant modifications to the software to meet the expanded or altered requirements. For instance, when users want to use 2-Factor Authentication, this will require modification to the Authentication module.

Overall, system maintenance ensures the continued effectiveness and relevance of the Electronic Thesis Management System by addressing errors, adapting to environmental changes, and accommodating evolving requirements.

# Future Works and Recommendations

The future works and recommendations for the Electronic Thesis Management System, include the following suggestions:

* + - Integration of Plagiarism Checker API: To enhance the integrity of the thesis writing process, it is recommended to integrate a plagiarism checker API. This integration will enable automatic plagiarism detection and ensure the originality of the submitted theses.
    - Integration with Library Management System: Integrating the Electronic Thesis Management System with the library management system can provide seamless access to relevant research materials for the students. This integration will enable students to easily

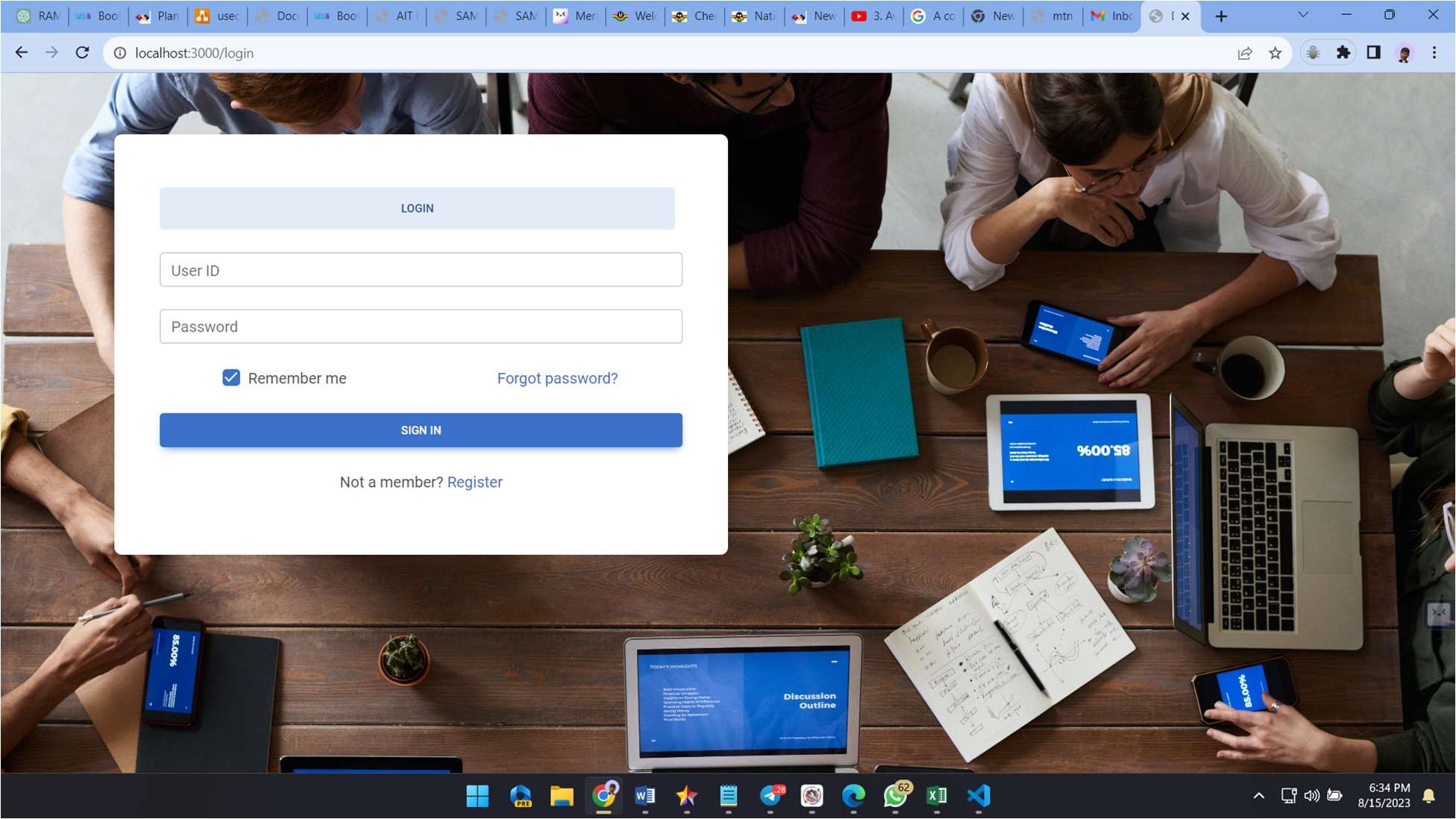
access resources required for their research, enhancing the overall efficiency of the thesis writing process.

* + - Integration of Student Database: Enabling the integration of the student database into the system will facilitate easy enrollment of students into the Electronic Thesis Management System. This integration will streamline the process of adding and managing student information, ensuring accurate and up-to-date records.

Additionally, in terms of future directions, it is recommended to make improvements in the student module. This can include features such as setting reminders for thesis-related deadlines and saving important dates to the calendar. These enhancements will help students stay organized and meet their thesis milestones effectively.

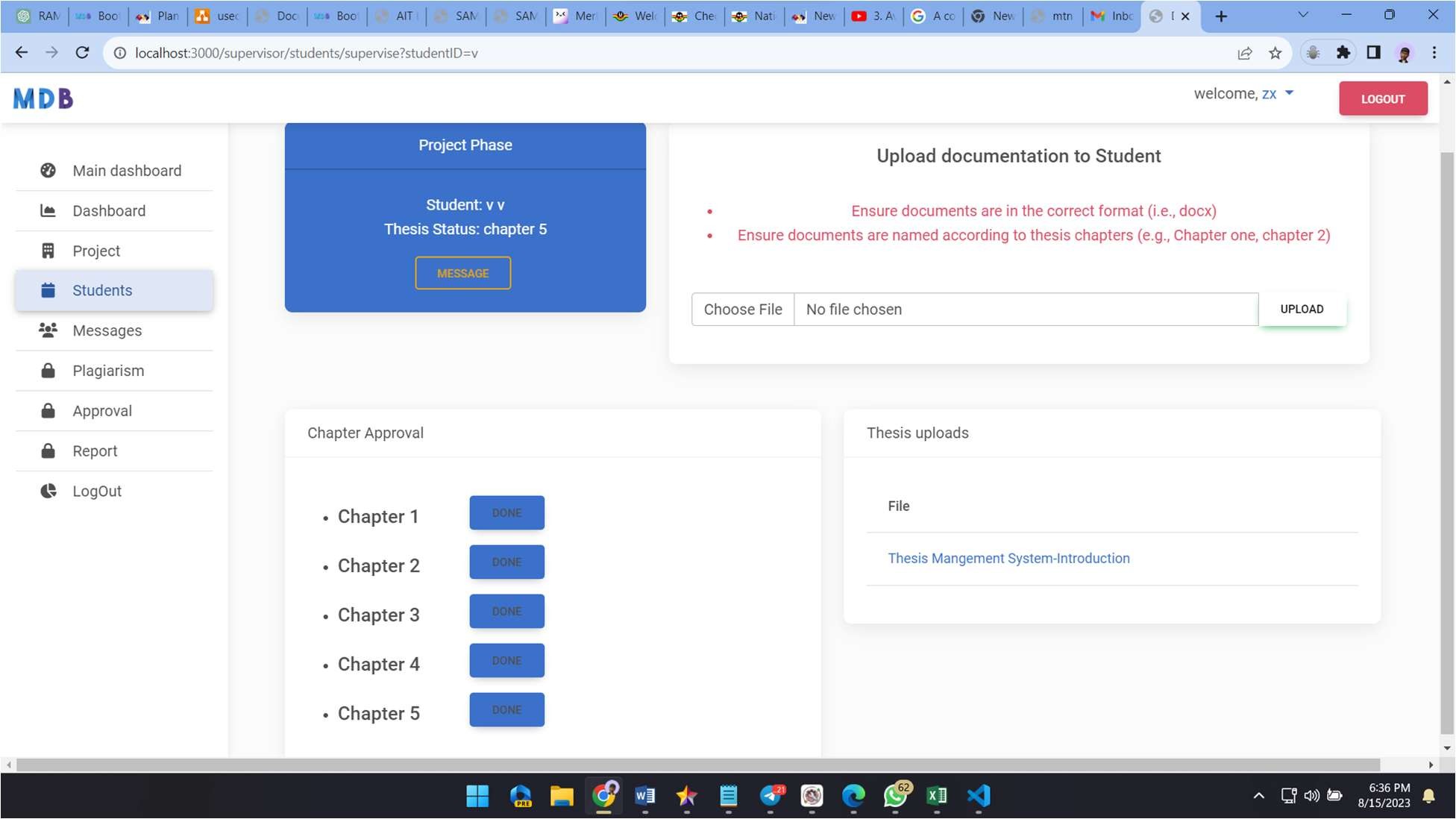
Overall, these recommendations and future works aim to enhance the functionality, efficiency, and user experience of the Electronic Thesis Management System, making it a more comprehensive platform for managing all aspects of the thesis writing process.

In figure 5.1 shows a preview of the login page. This page authentic users before they again access to main system.



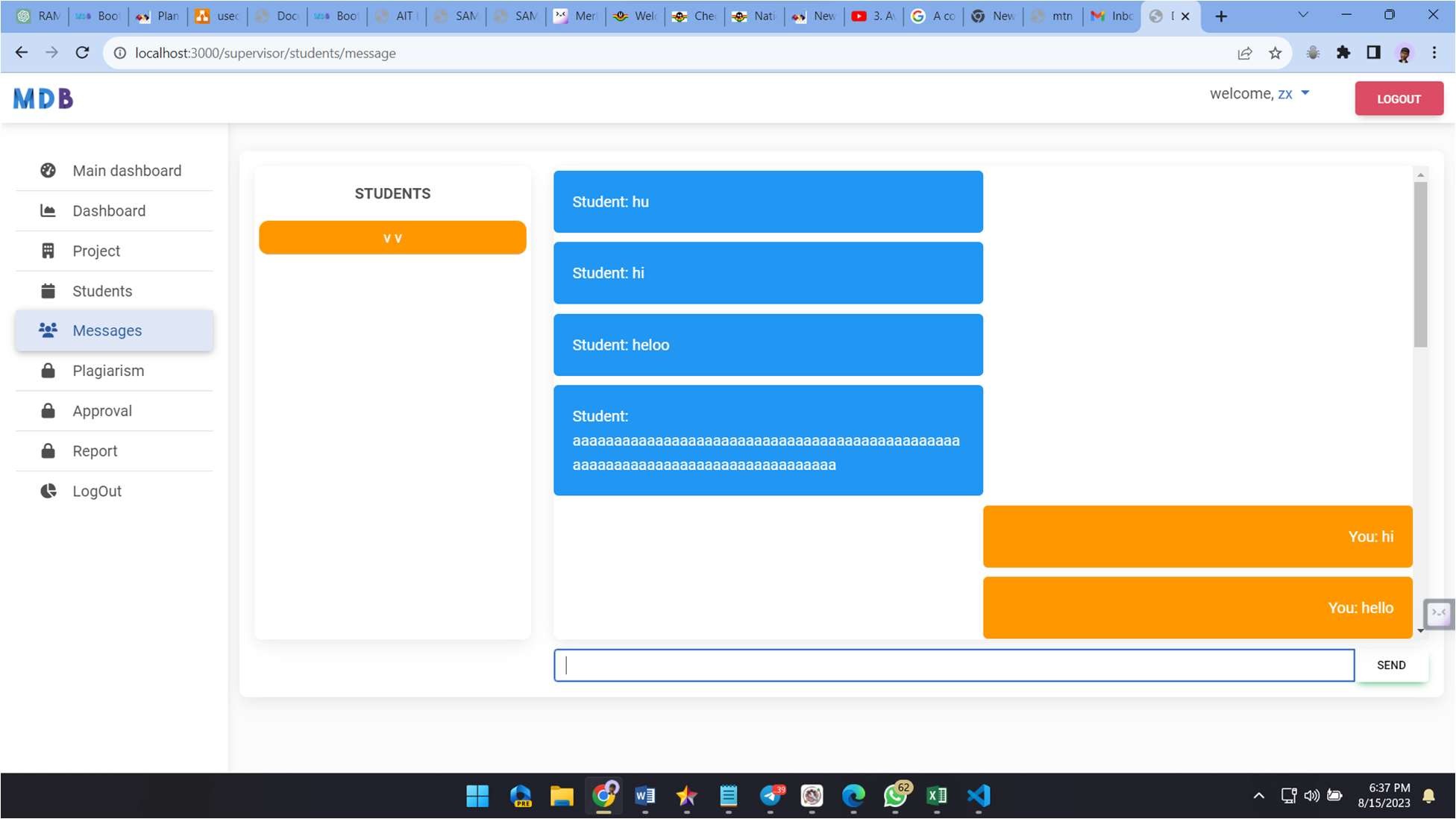
**Figure 5.1** Login Page

Figure 5.2 shows a preview of the supervision page. This page enables supervisors to monitor and supervise students project.



**Figure 5.2** Supervising Page

Figure 5.3 shows a preview of the chat feature page. This page enables supervisors to communicate with students.



**Figure 5.3** Chat page

REFERENCES

Ahmad, S., Khan, B., & Ahmad, M. Z. (2019). An overview of electronic thesis information management system. International Journal of Advanced Science and Technology, 28(15), 1101- 1110.

Alzahrani, E., Alqahtani, M., & Qashqari, M. (2020). Evaluation of electronic thesis and dissertation system. Journal of King Saud University - Engineering Sciences, 32(1), 64-70.

Chen, Y. C., & Chang, Y. L. (2017). Customized electronic thesis information management system: Implementation, benefits, and challenges. Journal of Educational Technology Development and Exchange, 10(2), 117-131.

Fowler, M. (2018). *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.

Glisson, W. B., & Chowdhury, G. G. (2002). Design of a digital dissertation information management system. *Program*, *36*(3), 152-165. https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=5221&context=libphilprac

https://[www.proquest.com/products-services/dissertations/](http://www.proquest.com/products-services/dissertations/)

Laudon, K. C. and Laudon, J. P. (2002) “Management Information System: managing the digital firm”, 7th Edition, Prentice Hall.

Liu, J., & Liu, J. (2019). A study on electronic thesis and dissertation (ETD) in the context of China. Library Management, 40(3/4), 174-189.

Marnewick, C. (2016). Benefits of information system projects: The tale of two countries. *International journal of project management*, *34*(4), 748-760.

Martin, C. & Powell, P., (1992). “Information System: a management perspective”, McGraw Hill, New York.

Nguyen, T. (2014). Electronic theses and dissertations (ETDs) management, academic institutions and governance: Retrospect and prospect. The Electronic Journal of Information Systems in Developing Countries, 63(6), 1-22.

Olatokun, W. M., & Adekunle, T. O. (2018). Evaluation of electronic theses and dissertations systems in Nigeria. Library Philosophy and Practice, 1-20.

ProQuest. (2021). ProQuest Dissertations and Theses. Retrieved from

Rueda-Medina, D. M., Morales-Zambrano, G. H., & Torres-Dávila, E. (2019). Benefits of an electronic thesis and dissertation information management system in a Colombian university. Journal of Education and Practice, 10(15), 48-55.

Sauer, C. (1993). *Why information systems fail: A case study approach*. Alfred Waller Ltd., Publishers.

Suroso, S., Sigit, D. V., Arnott, D., & Mustamu, E. (2019). Electronic thesis and dissertation system: A case study of its implementation in a university in Indonesia. Journal of Physics: Conference Series, 1363, 012058.

University of North Texas. (2021). OpenETD. Retrieved from https://open.library.unt.edu/openetd/

University of Pittsburgh. (2021). Electronic Thesis and Dissertation Submission System. Retrieved from https://[www.etd.pitt.edu/](http://www.etd.pitt.edu/)

Zhou, Q., & Li, C. (2018). Exploring the role of electronic thesis and dissertation (ETD) system in institutional repository: A case study in Chinese universities. Library Hi Tech, 36(2), 349-358.