**DESIGN AND IMPLEMENTATION OF A WEB-BASED FINAL YEAR PROJECT MANAGEMENT SYSTEM (FYPMS)**

**(A CASE STUDY OF COMPUTER SCIENCE DEPARTMENT BABCOCK UNIVERSITY)**

**BY**

**IZEVBIZUA SAMUEL EDOSA 19/1084**

**ADEKOYA DAMILARE ANOINTING 19/0095**

**OMAGE OMOLE NATHANIEL 19/1200**

A PROJECT WORK SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCE

B.Sc. (HONS)

IN COMPUTER SCIENCE

**SUBMITTED TO:**

THE DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING AND ENGINEERING SCIENCES

BABCOCK UNIVERSITY, ILISHAN REMO

OGUN STATE, NIGERIA

**MARCH, 2023**

**CERTIFICATION**

We declare that the project work, Design and Implementation of a Final Year Project Management was carried out by the following people:

Student Name Matric Number

IZEVBIZUA SAMUEL EDOSA 19/1084

ADEKOYA DAMILARE ANOINTING 19/0095

OMAGE OMOLE NATHANIEL 19/1200

DR. RUTH C. AMANZE \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME OF SUPERVISOR SIGNATURE / DATE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME OF EXTERNAL EXAMINER SIGNATURE / DATE

­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NAME OF HOD SIGNATURE / DATE

**DEDICATION**

To the field of computer science, which has continuously pushed the boundaries of what is possible and transformed the way we live, work, and connect with each other. This dedication is a tribute to the tireless efforts of countless innovators, pioneers, educators, and researchers who have dedicated their lives to advancing this field and making it accessible to everyone. It is an honor to contribute to this community and to be part of a legacy that will continue to inspire future generations.

**ACKNOWLEDGEMENTS**

We would like to express our sincere gratitude to God Almighty for giving us the strength, knowledge, and opportunity to complete this project.

We would like to thank our supervisor, Dr. Ruth C. Amanze, for her valuable guidance and support throughout the project. Her constant feedback and encouragement helped us stay on track and meet our goals.

Finally, we would like to thank all the individuals who have provided us with their valuable insights, comments, and suggestions throughout the project. Your input has been invaluable to us. Thank you all for your support and encouragement.

**ABSTRACT**

This study presents the development of a final-year project online management system to replace the current manual approach. The manual process is tedious, time-consuming, and ineffective, leading to missed deadlines and poor project supervision. The project aims to systematically involve all participants in a single collaborative online system for exchanging information related to the final year project. The proposed web-based supervision management system enhances project monitoring and supervision, ensuring that all deadlines are met. The development process requires knowledge of web design and setup, database design and setup, and server configuration. The initial prototype comprises four sections, including user profiles, project submissions, performance evaluation, and a communication module. The system provides various interfaces and functionality for the two user groups, supervisors, and students, ensuring proper grading and approval of student's work before the specified deadline. User verification, including authentication and authorization, provides system security. JavaScript, React, Node.js, and Express.js will be used as both the frontend and backend programming language for the system's development. The backend database for the project is MongoDB, and the MERN stack approach will be used to construct this website. Overall, the study demonstrates the potential for an online management system in project supervision and management, with significant benefits to students and supervisors.

**KEYWORDS**

Collaboration, Deadline monitoring, Manual approach, Online management system, Project supervision.

**TABLE OF CONTENTS**

1. Cover Page
2. Title Page
3. Certification
4. Dedication
5. Acknowledgment
6. Abstract
7. Table of Contents
8. List of Figures
9. Abbreviations

**CHAPTER ONE (INTRODUCTION)** **1**

1.0 Background of the Study 1

1.1 Statement of the problem 2

1.2 Aim and specific objectives 3

1.3 Materials and Methods 3

1.4 Scope of the study 4

1.5 Significance of the study 5

1.6 Definition of terms 6

1.7 Organization of the project 8

**CHAPTER TWO (LITERATURE REVIEW)** **10**

2.0 Introduction 10

2.1 Historical background of the research 11

2.2 Overview of existing system 12

2.3 Review of related work 12

2.3.1 Staff to Student Interactive Platform 12

2.3.2 Student Tracking Systems 14

2.3.3 The Need for Final Year Projects Management System 16

2.3.4 Gaps to the study 18

2.4 Strength and weaknesses of the existing system 18

**CHAPTER THREE (SYSTEM ANALYSIS AND DESIGN)** **20**

3.0 Introduction 20

3.1 Research design 20

3.2 Functional and Non-functional requirements 21

3.2.1 Functional Requirements 21

3.2.2 Non-Functional Requirements 24

3.2.3 Proposed Model Diagram 25

3.3 System Architecture 26

3.3.1 UML Class Diagram 27

3.3.2 Database Enhanced Entity Relationship Diagram (EERD) 28

3.4 Flow Chart and UML Use Case Diagram 29

3.4.1 Flow Charts 29

3.4.2 UML Use Case Diagram 33

3.5 Development Tools 34

**CHAPTER FOUR (IMPLEMENTATION)** **36**

4.0 Introduction 36

4.1 System Testing 37

4.2 Graphical view of the project 37

**CHAPTER FIVE (SUMMARY, CONCLUSION AND RECOMMENDATION)**

5.0 Summary 49

5.1 Conclusion 49

5.2 Recommendation 50

5.3 Limitations of the study 51

5.4 Suggestions for further studies or research 52

**REFERENCES** **54**

**APPENDICES**……………………………………………………………………….**58**

**LIST OF FIGURES**

* Figure 3.1: Proposed Model Diagram (p. 25)
* Figure 3.2:Logical architecture and work flow of the proposed system respondents (p. 26)
* Figure 3.3: UML Class Diagram Showing the System Architecture of The FYPMS (p. 27)
* Figure 3.4: EERD of the Proposed FYPMS (p. 28)
* Figure 3.5: Signup Module Flowchart (p. 29)
* Figure 3.6: Login Module Flowchart (p. 30)
* Figure 3.7: Project Completion Module Flowchart (p. 31)
* Figure 3.8: Grading Module Flowchart (p. 22)
* Figure 3.9: UML Use Case Diagram (p. 33)
* Figure 4.1: Visual Interface of Signup Module (p. 38)
* Figure 4.2: Visual Interface of Login Module (p. 39)
* Figure 4.3: Visual Interface of Supervisor Dashboard Module (p. 40)
* Figure 4.4: Visual Interface of Student Dashboard Module (p. 41**)**
* Figure 4.5.1: Visual Interface 1 of the Student Project Module(p. 42)
* Figure 4.5.2: Visual Interface 2 of the Student Project Module (p. 43)
* Figure 4.6: Visual Interface of the Supervisor Project Module (p. 44, 45, 46)
* Figure 4.7.1: Supervisor Interface 1 of Communication Module (p. 47)
* Figure 4.7.2: Supervisor Interface 2 of Communication Module S (p. 48)
* Figure 4.8: Student Interface of Communication Module (p. 48)

**ABBREVIATIONS**

1. Web-based technologies – WBT
2. Higher Education Institutions - HEI
3. Software Development Life Cycle - SDLC
4. Final Year Project – FYP
5. Final Year Project Management System - FYPOMS
6. User-Friendly Online Platform - UFOP
7. Final project deliverables - FPD
8. MongoDB database system - MDB
9. JavaScript programming language - JS
10. Visual Studio Code - VSC
11. Lucid-Chart - LC
12. SQL – Structured Query Language
13. Microsoft Word - MSW
14. Enhanced Entity Relationship Diagram - EERD
15. Frontend - FE
16. Backend - BE
17. User interface - UI
18. Entity relationship diagrams - ERD
19. Unified Modeling Language - UML

**CHAPTER ONE**

**INTRODUCTION**

1. **Background of the Study**

The use of web-based technologies in educational settings worldwide is expanding rapidly, with benefits including the eradication of specific challenges such as the need for manual work, and the reduction of other challenges such as limited access to information and communication barriers. Online web-based technologies have made it simple to:

* Collaborate and share data
* Reduce development costs
* Centralize security and information availability
* Boost productivity.

These benefits are essential to ensure that final-year projects, which are typically completed in higher education institutions, are finished on time and to the criteria established by the institution. To ensure that final-year projects meet institutional standards and deadlines, it is crucial to have a seamless and effective exchange of information between students, supervisors, project coordinators, and other involved parties, including project updates, feedback, and progress reports. The traditional manual system used for managing and supervising Final Year Projects in the Department of Computer Science at Babcock University has several challenges, including being time-consuming, prone to errors, and lacking in efficiency. This can result in delays and may not always produce the desired outcomes, which can be problematic for all parties involved. This system can hinder the efficient administration and advancement of all parties involved, leading to little or no progress toward desired outcomes over time. A time-saving online management system that offers a better communication channel, streamlines project workflows, and provides a centralized platform for grading, communication, and project submission can eliminate these obstacles and improve project management. The proposed Final Year Project Online Management System (FYPMS) seeks to provide a web-based environment that supports improved and consistent management and supervision between students and their designated supervisors. The FYPMS offers appropriate solutions for the prior FYP management, and is suggested to increase the effectiveness of the management process.

**1.1 Statement of the Problem**

Final Year Project management in the Department of Computer Science at Babcock University is currently a manual process that involves group cooperation and regular meetings with supervisors. However, the existing management method has several drawbacks, including:

1. The occasional absence of supervisors or students from scheduled sessions can result in decreased efficiency and productivity.
2. There is no established method for all participants to connect and communicate effectively, which can lead to delays.
3. The manual system requires more labor and is time-consuming, making it less effective in terms of project management and supervision.
4. There is no reliable way to monitor the progress of individual students, which can result in the contribution of their subpar participation to the project's progress not being entirely accounted for.

**1.2 Aim and Specific Objectives**

The aim of the suggested system is to provide a comfortable, effective, dependable, and user-friendly platform that simplifies project supervision for Final Year Projects (FYPs) in the Department of Computer Science at Babcock University.

Objectives:

To achieve the aim stated above, the system will have the following specific objectives:

1. To provide a user-friendly online platform for project management and supervision that facilitates better communication and collaboration between students and supervisors.
2. To reduce the time and effort required for FYP management and supervision by automating various processes and workflows.
3. To ensure that FYPs are completed on time and meet the required standards by providing a structured and well-organized platform for project management and supervision.

**1.3 Materials and Methods**

This section outlines the materials and procedures used to complete the project, including the tools and system requirements.

System requirements for implementing this project include:

* Google Chrome web browser
* MongoDB database system
* JavaScript programming language
* Visual Studio Code for writing code
* Lucid-Chart for designing Flowcharts, UML and system flow diagrams
* MySQL workbench for designing the database Enhanced Entity Relationship Diagram (EERD)
* Microsoft Word for project documentation

This is a web-based system/website that uses JavaScript programming language for both its frontend and backend functionalities, with Google Chrome as the system's human interface. MongoDB serves as the database system, while Microsoft Word is utilized for project documentation. The two essential entities involved in the system overview are the Supervisor and the Student, and the system's primary objective is to build a channel for communication between them.

The system design process follows a series of steps, starting with the engineering of entity relationship diagrams using MySQL workbench to model the MongoDB database instance using a relational approach. UML diagrams are also employed to illustrate the system's workflow. The JavaScript code for the system's frontend and backend functionalities is written using Visual Studio Code. The system is tested to identify and address any issues or unmet user needs that may arise during development.

**1.4 Scope of the Study**

The primary objective of this project is to develop and deploy a full-stack website that addresses the following issues related to the management of final year projects:

* Inefficient communication between the student and the supervisor
* Creating a scoring system for individual participants based on completion of project phases within a certain time frame and other criteria
* Monitoring the progress of each project phase

The following are some research limitations:

* This system will not use algorithms to automatically pair supervisors with their pupils. Assigning of students will be done by the lecturers/supervisors.
* All of the school's departments will not be taken into account while designing and implementing the system as it is a prototype and various departments may have different methodologies in handling final year projects.
* It is only implemented for the benefit of the students and supervisors, and will not consider the project coordinator as a role player in this system.

This system's overall design and implementation will necessitate expertise and technical knowledge in the following areas:

* Web design, setup, and development
* Database design and management
* Server design and setup

**1.5 Significance of the Study**

The proposed Final Year Project Online Management System will benefit the following stakeholders:

* **The school:** The school will gain a lot from the study of the suggested Final Year Project Online Management System since an automated management system is practical from an economic standpoint and precise.
* **The students:** The project management system will be more adaptable and pleasant for students to learn crucial 21st-century skills, including communication, teamwork, and technological proficiency.
* **The Project Supervisors:** This will be used as a tool to manage student performance and project work easily.
* **Industry partners or employers:** who could potentially provide feedback or resources for the project or offer opportunities for collaboration.
* **Academic institutions or researchers:** who may be interested in studying the impact of the system on project management practices and student progress.
* **The Current Researcher:** The researcher will gain an understanding of the school's current project management system.
* **Future Researchers:** It will operate as their point of reference for their research activities and assist them in learning how to use newer models to achieve the desired aim and communicate their views in the study they have selected.

**1.6 Definition of Terms**

The terminology listed below were used by the researcher to clarify certain study concepts. The terms are operationally defined in the study's context.

1. **Web-based:** A web-based application is any program that is accessed via a network connection using HTTP rather than being stored in a device's memory. Web browsers are frequently used to run web-based applications.
2. **FYP:** This is an acronym for Final Year Project
3. **FYPMS:** This is an acronym for Final Year Project Management System.
4. **Workspace:** The use of this word in this chapter refers to the physical or virtual space/area/environment in which work will be carried out.
5. **HTTP:** This is an acronym for Hyper-Text Transmission Protocol. It is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers, but it can also be used for other purposes.
6. **Subpar:** This refers to performance that is below a usual or normal level.
7. **Client-side:** Also referred to as frontend, it refers to everything in a web application that is displayed or takes place on the client (end user device). This includes what the user sees, such as text, images, and the rest of the UI, along with any actions that an application performs within the user's browser.
8. **Server-side:** server-side refers to programs and operations that run on the server.
9. **Message-boards:** a platform where users can post comments about a particular issue or topic and reply to other users' postings.
10. **JavaScript:** JavaScript is a client-side and server-side scripting language inserted into HTML pages and is understood by web browsers. JavaScript is also an Object-based Programming language.
11. **Visual Studio Code:** Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide developer tools needed for quick code-build-debug cycle.
12. **ERD:** This is an acronym for Entity Relationship Diagram. An ERD, also known as an entity relationship model, is a graphical representation that depicts relationships among entities within an information technology (IT) system such as a database.
13. **Full-stack:** This refers to the development of both the client-side (frontend) and the server-side (backend) of a complete web application.
14. **Algorithms:** These are procedure or formula used for solving a problem.

**1.7 Organization of the Project**

This section describes the organization of the chapters in this project and provides a brief explanation of the scope of the project.

1. Chapter 1: (INTRODUCTION)
   * This chapter explains the project and serves as a request for approval of the project study.
2. Chapter 2: (LITERATURE REVIEW)
   * This chapter explores relevant publications and materials on related subjects in order to analyze current and existing systems and identify study gaps that can be filled in.
3. Chapter 3: (SYSTEM ANALYSIS AND DESIGN)
   * This chapter discusses the project's techniques, such as the design procedures and development plans.
4. Chapter 4: (IMPLEMENTATION)
   * This chapter examines all the requirements and techniques for implementation, including written code, system configuration, and testing.
5. Chapter 5: (SUMMARY, CONCLUSION, AND RECOMMENDATIONS)
   * This chapter provides a summary of the project, its conclusions, and recommendations for future research on related works.

The project aims to develop a web-based Final Year Project Management System (FYPMS) that will improve project management practices and facilitate the monitoring of students' progress by project supervisors. The FYPMS will provide a centralized platform for students and supervisors to collaborate and communicate on project-related activities. The scope of the project includes the design, development, and implementation of the FYPMS.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.0 Introduction**

The World Wide Web (WWW) is an information system that has revolutionized how information is accessed and distributed through the Internet. With the widespread adoption of information technology (IT), the web has become a popular tool in various institutions for management and administrative tasks. The idea of creating online systems that facilitate collaboration and data sharing is based largely on this technology.

Web servers are programs that handle users' network requests and provide them with the files needed to build web pages. Hypertext Transfer Protocol (HTTP) is the standard protocol used for this interaction. Web pages are digital files linked to the World Wide Web that can be viewed by anyone with an internet connection and a web browser. Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet, i.e., websites.

The part of a website that the user interacts with directly is known as the front end or the client-side of the application. In contrast, the server-side or the back end is the part of the website that users cannot see or interact with. It is used to store and manage data.

In this project, we aim to enhance the management and supervision of final year projects in higher institutions by implementing a web-based final year project management system. This system will make it easier for students and their project supervisors to interact in a comfortable, efficient, dependable, and user-friendly manner. This chapter provides a review of relevant literature that was conducted to support this research project.

* 1. **Historical Background of the Research**

The educational system at Babcock University Nigeria has continued to evolve with the adoption of new technological advancements. However, with the growing number of students, manual and conventional methods of managing and overseeing final year project administration and supervision have become time-consuming, labor-intensive, and fairly inefficient. The lack of a defined monitoring procedure results in missed deadlines and project timelines not being appropriately recognized.

Furthermore, the COVID-19 pandemic has affected the educational process by limiting physical interactions between students and lecturers. This has further exacerbated the challenges faced in the manual method of project supervision, making it difficult to provide adequate support and guidance to students.

The foundation of this study is the implementation of a web-based and online system that can enhance project monitoring and supervision in the department of Computer Science at Babcock University Nigeria. The web-based system is expected to provide a seamless and efficient method for managing and supervising final year projects, particularly in light of the limitations presented by the pandemic. It will also help to overcome the challenges of missed deadlines, inadequate support, and the inability to appropriately recognize student achievements.

* 1. **Overview of Existing System**

The existing system used for managing and supervising Final Year Projects at Babcock University's Department of Computer Science is a manual process that involves group cooperation and regular meetings with supervisors. However, this approach has several drawbacks that include the occasional absence of supervisors or students from scheduled sessions leading to decreased efficiency and productivity. There is also no established method for all participants to connect and communicate effectively, leading to delays. Additionally, the manual system requires more labor and is time-consuming, making it less effective in terms of project management and supervision. Lastly, there is no reliable way to monitor the progress of individual students, which can result in their subpar participation not being entirely accounted for in the project's progress. The proposed Final Year Project Online Management System (FYPMS) seeks to address these challenges by providing a web-based environment that supports improved and consistent management and supervision between students and their designated supervisors.

* 1. **Review of Related Works**
     1. **Staff to Student Interactive Platform**

Kambo (2017) authored an article on the use of the Internet for interactive learning, teaching, and research based on his experiences in Nigeria. The article provides in-depth explanations of the value of the internet as a learning tool and its effects on Nigerian education. It examines how the internet has improved research, learning and teaching capacities, as well as the methodologies employed by students, teachers, and researchers. According to his survey, 50% of teachers use the internet for interactive research while 40% of students use it for interactive learning. However, none of the teachers utilize it for interactive teaching. Jung (2019) developed a web-based project-based learning support system that was implemented successfully. Jung recognized that the use of the web had significantly improved the educational system by getting around constraints on time and space in conventional and contemporary schools. Teachers and students can now access a wide variety of materials and information online within cyberspace. Furthermore, learning via the web allows for both synchronous and asynchronous communication at any time. However, a lack of face-to-face communication may reduce students' motivation. In his paper, Jung proposes a learning model based on constructivist principles, called Web Project Learning, to provide motivation and collaborative learning for students in the web environment. The model is based on project-based learning and encourages parent and student participation, and can be applied to any subject. He put the model into action and demonstrated how it can be used in environmental education. Gallian (2016) created a system to keep students engaged in an online tutoring system at all times. Gallian noted that online tutoring systems have grown in popularity in recent years, with several of them being sponsored by top Universities. This popularity, made possible by a combination of accessibility, appealing content, and the absence of restrictions to participate, is not being truly successful due to low levels of student retention rate. Most studies of online tutoring systems focus solely on technological aspects or examples of implementations, with little attention paid to learner engagement and retention. Gallian attempted to gather articles by conducting a systematic review of existing work in several academic databases.

* + 1. **The Role and Benefits of Student Tracking Systems in Higher Education.**

Student tracking systems have become increasingly important in higher education due to the demand for measurable outcomes of student progress and institutional success. These systems allow educational institutions to monitor student progress, intervene when necessary, and improve learning outcomes. This review will explore the role of student tracking systems in higher education and their impact on student engagement and learning. According to Bers (1989), the development of tracking systems has been driven by increasing emphasis on marketing, accountability, communication with students, and internal student competition. Similarly, Helic (2000) argues that good online tutoring necessitates regular monitoring of a learner's progress with the material and testing of the acquired knowledge and skills. Li and Li (2019) suggest that assessment tracking systems can be used to enhance student engagement and learning. These systems provide students with immediate feedback on their performance, which can motivate them to learn and improve their performance. Additionally, the systems allow educators to identify struggling students and provide them with targeted interventions to improve their learning outcomes. Galusha (1997) suggests that tracking and progress systems have been significantly applied to online teaching, assessment, and measurement, allowing teachers to gauge student response, feedback, and progress toward goals. These systems have been crucial in distance education. Similarly, Ragan (1998) argues that tracking systems solve the problem of learners who lack casual contact with the teacher and other students by allowing them to be easily reached through the system. Oppaga (2006) notes that many state institutions have adopted automated systems that track students' progress toward degrees and put registration holds on them if they veer off course until they meet an academic advisor. These systems assist students in graduating on time and allow Universities to allocate their academic advising resources more effectively. In addition to the aforementioned benefits, student tracking systems can enhance student engagement and learning outcomes. Li and Li (2019) highlight the importance of using assessment tracking systems to promote student engagement, and suggest that such systems can improve the effectiveness of formative assessment. The authors note that when students are given frequent, timely feedback on their progress, they are more likely to engage in their learning and take responsibility for their own academic success. Furthermore, tracking systems can provide instructors with valuable insights into their students' learning needs and progress, allowing them to make data-informed adjustments to their teaching and support. Research, has shown that tracking systems can have a positive impact on student engagement and learning outcomes. For example, a study by Kim and Lang (2016) found that an online learning platform that provided personalized feedback and progress tracking led to increased student engagement and motivation. Similarly, a study by (Kim et al, 2017) found that a learning analytics system that provided feedback on students' learning behaviors and progress led to improved learning outcomes. While student tracking systems have many benefits, it is important to note that they must be used in a responsible and ethical manner. As Li and Li (2019) note, tracking systems should not be used to solely evaluate student performance, but should instead be used to support student learning and growth. Additionally, tracking systems should be transparent, and students should be informed about how their data is being used.

In conclusion, student tracking systems play an increasingly important role in measuring student progress and institutional success in higher education. These systems can be used to enhance student engagement and learning by providing immediate feedback and targeted interventions to struggling students. Additionally, these systems aid students in completing their studies within the expected time frame while also helping Universities optimize their allocation of academic advising resources. As higher education continues to evolve, it is likely that student tracking systems will become even more important for ensuring student success.

* + 1. **The Need for A Final Year Projects Management System**

Final year projects are an important part of undergraduate education, providing students with the opportunity to apply the knowledge and skills they have acquired throughout their degree program. These projects require a significant amount of planning and management, both for students and faculty, and can often be a source of stress and frustration for all involved. In recent years, there has been an increasing need for a Final Year Projects Management System (FYPMS) to help streamline the process and improve outcomes. In this literature review, we will examine three recent articles that address this need and explore the benefits and challenges of implementing a FYPMS. Several authors have explored the development of FYPMS to address the challenges faced by both students and faculty in managing final year projects. Gomaa and Mohamed (2020) describe the development of an online FYPMS for engineering students. The system was designed to address the lack of communication and coordination, insufficient guidance, and difficulty in tracking progress. The authors conducted a survey of students and faculty to identify the key features and requirements for the system and used these findings to develop a web-based platform. The system includes features such as task scheduling, progress tracking, and communication tools to facilitate collaboration between students and supervisors. The authors report positive feedback from users and suggest that the system has improved the efficiency and effectiveness of project management. Iruka, Akinyemi, and Oyewunmi (2019) explore the challenges and opportunities of implementing a FYPMS in a Nigerian university. The authors conducted a survey of final year students, project supervisors, and academic staff to identify the key issues and concerns related to project management. They found that the lack of a centralized management system was a significant challenge, leading to issues such as delayed feedback, poor communication, and difficulty in tracking progress. The authors also identified opportunities for improving the process, such as the use of technology and increased collaboration between students and supervisors. They suggest that the implementation of a FYPMS could lead to improved outcomes and increased student satisfaction. Olajubu, Olakanmi, and Oyewole (2017) describe the design and implementation of a FYPMS for computer science students at a Nigerian university. The authors used an iterative design process to develop the system, incorporating feedback from students and faculty at each stage. The system includes features such as progress tracking, task scheduling, and communication tools to facilitate collaboration between students and supervisors. The authors report positive feedback from users and suggest that the system has improved the efficiency and effectiveness of project management. They also noted the potential for the system to be adapted for use in other departments and institutions. In conclusion, the need for a Final Year Projects Management System has become increasingly evident in recent years, with many Universities and institutions seeking to streamline the process and improve outcomes. The articles reviewed here highlight the benefits and challenges of implementing such a system, as well as opportunities for improvement. The use of technology, collaboration between students and supervisors, and centralized management are key factors in the success of a FYPMS. While there are still challenges to be addressed, such as resistance to change and lack of resources, the development of FYPMS holds great promise for improving the final year project experience for all involved.

* + 1. **Gaps To the Study**

The reviewed literature has highlighted several areas in which the current systems for project-based learning can be improved. In particular, three key gaps have been identified that are ripe for further investigation and development.

* The lack of a messaging system for effective communication between students and their supervisor.
* The absence of platforms that allow for the sharing of software development repositories, hindering collaborative work among students and faculty.
* The need for submission modules that provide a structured way for students to submit their work, increasing efficiency and reducing errors in the process.
  1. **Strengths And Weaknesses of The Existing System**

Some strengths of the existing system include:

* The major components of the system are managed manually, so power and internet disruptions do not significantly obstruct the flow of project completion.
* The use of logbooks to record meetings and other information is convenient and easily accessible, which somewhat eliminates the risk of data redundancy.
* The current use of physical meetings ensures the availability of the student to an extent.

Some weaknesses of the existing system include:

* The use of logbooks focuses on the progress of report writing. There is no standard medium for monitoring the progress of software development and implementation.
* There is a tendency to introduce inconsistent data entries because of these manual methods of recording logbooks.
* Limited opportunities for feedback: Students may not receive timely or adequate feedback from their supervisor, which can impact the quality of their work and their overall learning experience.
* Inefficiency in communication: The use of manual methods for communication, such as logbooks and physical meetings, can be time-consuming and lead to miscommunication or delays in response times.

**CHAPTER THREE**

**SYSTEM ANALYSIS AND DESIGN**

**3.0 Introduction**

The process of system analysis and design involves the organization and development of information systems through the use of more efficient methods. This process is achieved by defining and comprehending what the system should accomplish, as well as how its various components should be put into action and collaborate. The primary focus of system analysis and design is to examine how a system behaves, its connections to other subsystems, and its capacity to achieve a specific objective. This often entails an examination of the system's output quality and performance.

In this chapter, we will discuss the system analysis and design process for the final year project management system. Through this process, we aim to develop an efficient system that meets the requirements of all stakeholders involved in the management of final year projects. We will explore various methods and tools that can be used to analyze and design the system, including the use of modeling and prototyping. The ultimate goal of this process is to improve the existing system and create a more effective and efficient solution for managing final year projects.

**3.1 Research Design**

In the research design, the iterative Software Development Life Cycle (SDLC) model will be used for this project after a thorough examination of different SDLC models because of its repetitive nature and ability to revert to earlier phases, it has a less expensive effect on incorporating modifications at this early stage of software development/implementation and other advantages. The iterative process involves a continuous cycle of planning, analysis, implementation, testing and evaluation.

* **Planning**: This is the first stage of the iterative model. In this stage, an outline of the basic requirements is generated, related documents are gathered, a plan and a timeline for the first iterative cycle is created.
* **Analysis and design:** This is the second stage of the iterative model. After finalizing system needs and doing some feasibility analysis, database models and technical requirements based on the plan will be created together with a functioning architecture, schematic, or algorithm that satisfies the requirements. This chapter will emphasize on this stage of development.
* **Implementation:** This is the third and most crucial stage of the iterative model. In this stage, the necessary design and system functionalities are created to meet the requirements through code execution.
* **Testing:** In this stage, defects are detected and recognized in order to be fixed and more precisely meet system requirements.
* **Evaluation:** This is the final stage of the iterative model. In this stage, project stakeholders evaluate the project's state and compare this iteration with the requirements and expectations during the evaluation stage.

**3.2 Functional and Non-Functional Requirements**

**3.2.1 Functional Requirements**

The functional requirements for the Final Year Project Management System are listed below. These requirements have been divided into various modules to improve system organization and efficiency.

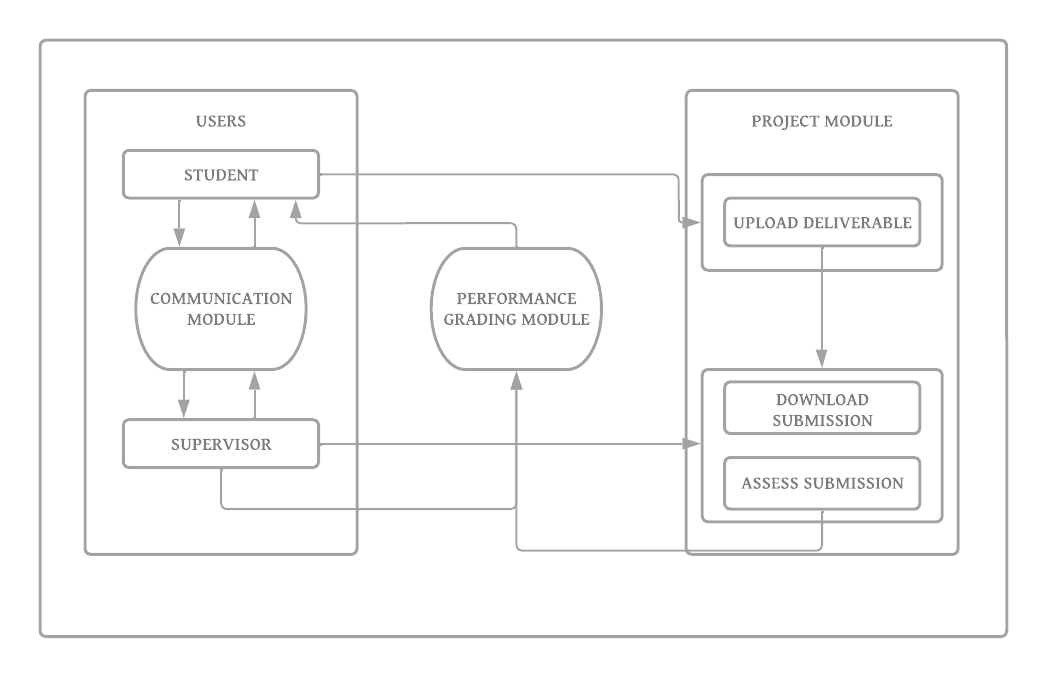
1. **Signup module**
   1. The system shall allow users to create new profiles by entering personal details such as name, email address, and phone number.
   2. The system shall generate a unique identification code for each user upon successful registration.
   3. The system shall store user data securely and prevent unauthorized access.
2. **Login module**
   1. The system shall provide a login page for already registered users to enter their email and password.
   2. The system shall authenticate user credentials and authorize access to the system based on the user's role.
   3. The system shall restrict unauthorized access to sensitive user data.
3. **Student dashboard/view module**
   1. The system shall provide a dashboard for students to view and manage their project progress.
   2. The dashboard shall display the student's basic information, including name, email address, and phone number.
4. **Supervisor dashboard/view module**
   1. This module shall display the supervisor's basic information, including name, email address, and phone number.
   2. The system shall allow supervisors to view a list of students they are currently supervising.
   3. Each list item shall serve as a link to detailed information about each student's project progress and development where supervisors will assess students' progress through uploaded documents and deliverables by downloading and approving or refusing them.
5. **Project progress module**
   1. The system shall divide each project into chapters that require a submission of an upload file.
   2. The system shall allow students to track their progress by viewing the state of their uploaded documents and deliverables (either it is approved or not).
   3. The system shall allow supervisors to track students' progress by assessing uploaded documents and deliverables by downloading and approving them
6. **Grading module**
   1. The system shall allow supervisors to assign points to students based on their overall performance at any point in time.
   2. The grading system shall be transparent and easily accessible to both supervisors and students.
7. **Software sharing module**
   1. The system shall allow students to share their software development progress with supervisors and other students.
   2. The module shall provide input platforms for links containing repositories such as GitHub or Bitbucket for sharing code or even links to live previews of their implementation development.
8. **Communication module**
   1. The system shall provide a messaging system for supervisors and students to communicate and send messages online which may include meeting links.
   2. The messaging system shall be secure and reliable.

**3.2.2 Non-Functional Requirements**

In addition to the functional requirements, the following non-functional requirements and constraints have been identified for the project:

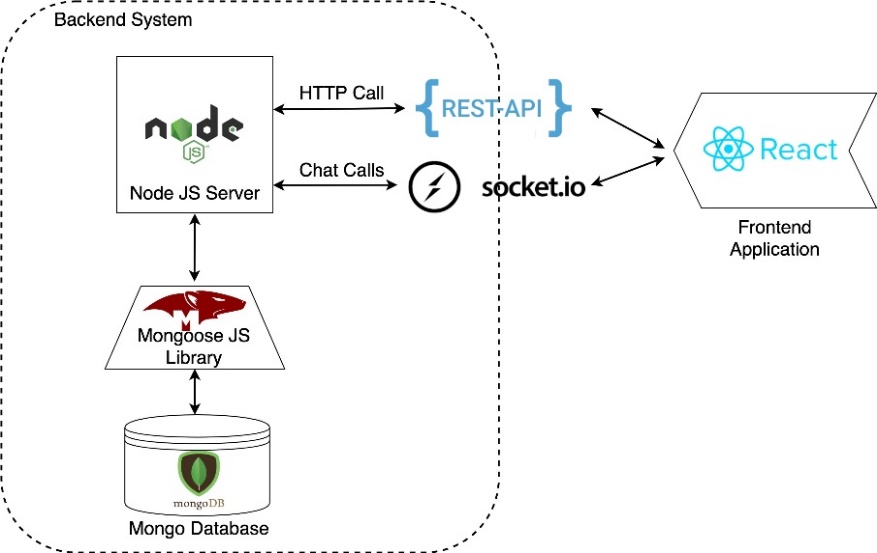
* Time constraints: Each stage of a project must be completed within a specified time span, as designated by the supervisor. Failure to meet the set timelines may result in penalties or affect the overall grade of the student.
* Grading and Assessment: The grading module will be managed by the supervisor who will assign points to the student based on their overall performance, including factors such as meeting deadlines, quality of work, and project completion. The total points can later be converted to a form of Continuous Assessment (CA) as the university department may decide.
* Code sharing: The software sharing module will only provide a platform for students to share links containing code repositories such as GitHub, or even live previews of their development results. Students are required to provide clear and concise comments in their code to aid understanding and enhance collaboration among team members.
* Security: The system will ensure data privacy and confidentiality by implementing security features such as user authentication, access controls, and encryption of sensitive data.
* Scalability: The system should be designed to handle a large number of users, projects, and data with minimal effect on its performance.
* Usability: The system should be user-friendly, with clear and concise instructions, error messages, and help sections to aid users in navigating the system.

**3.2.3 Proposed Model Diagram**

****

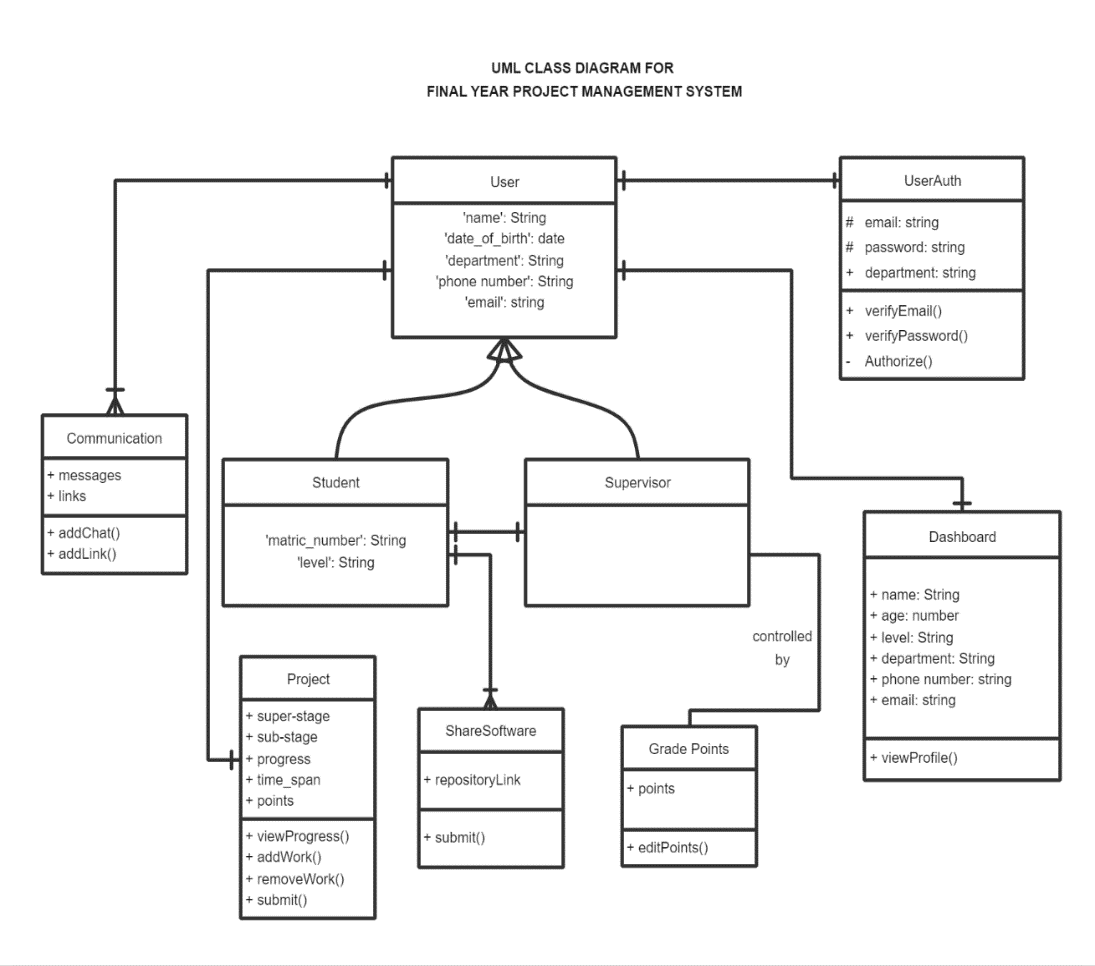
**Fig 3.1** Proposed Model Diagram of the FYPMS

**3.3 System Architecture**

****

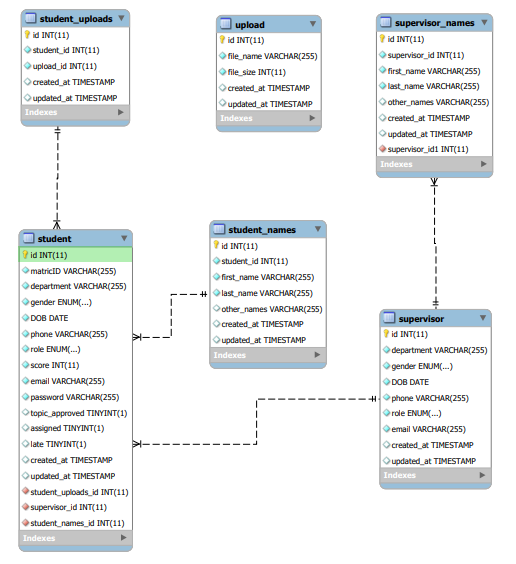
**Fig 3.2** Logical architecture and work flow of the proposed system

**3.3.1 UML Class Diagram**

****

**Fig 3.3** UML Class Diagram Showing the System Architecture Of The FYPMS

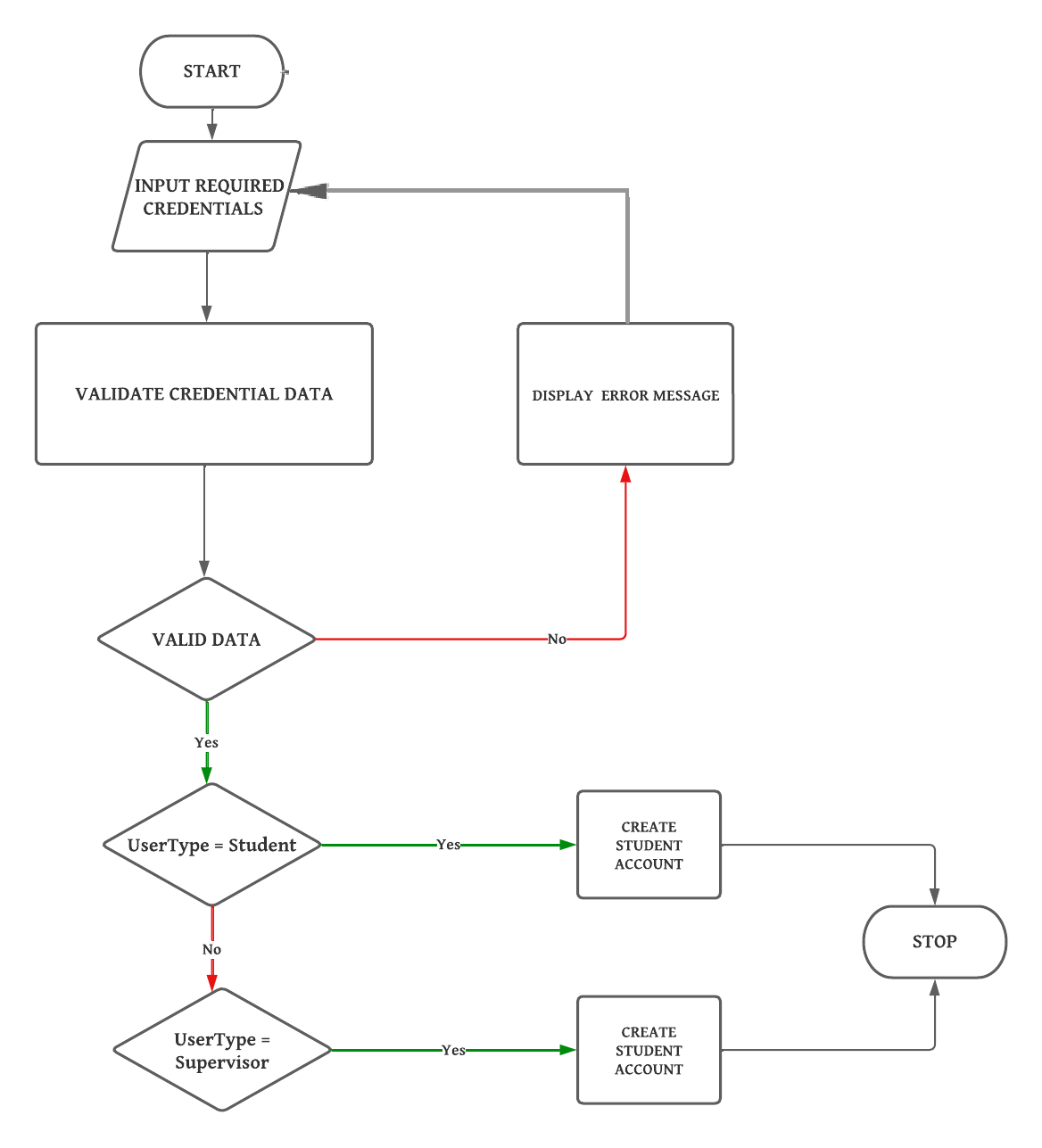
**3.3.2 Database EERD**

****

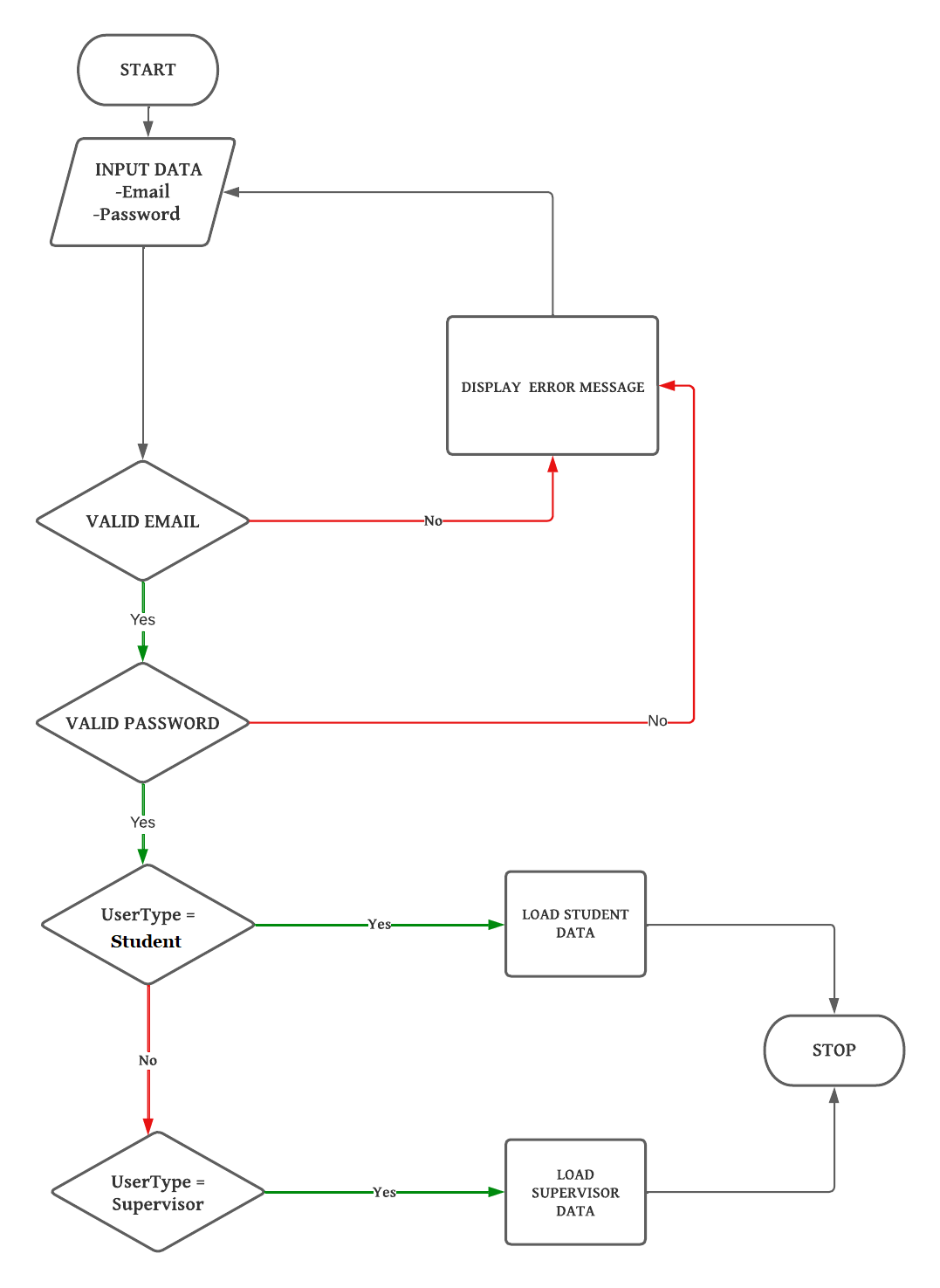
**Fig 3.4** EERD of the Proposed FYPMS

**3.4 Flow Chart and UML Use Case Diagram**

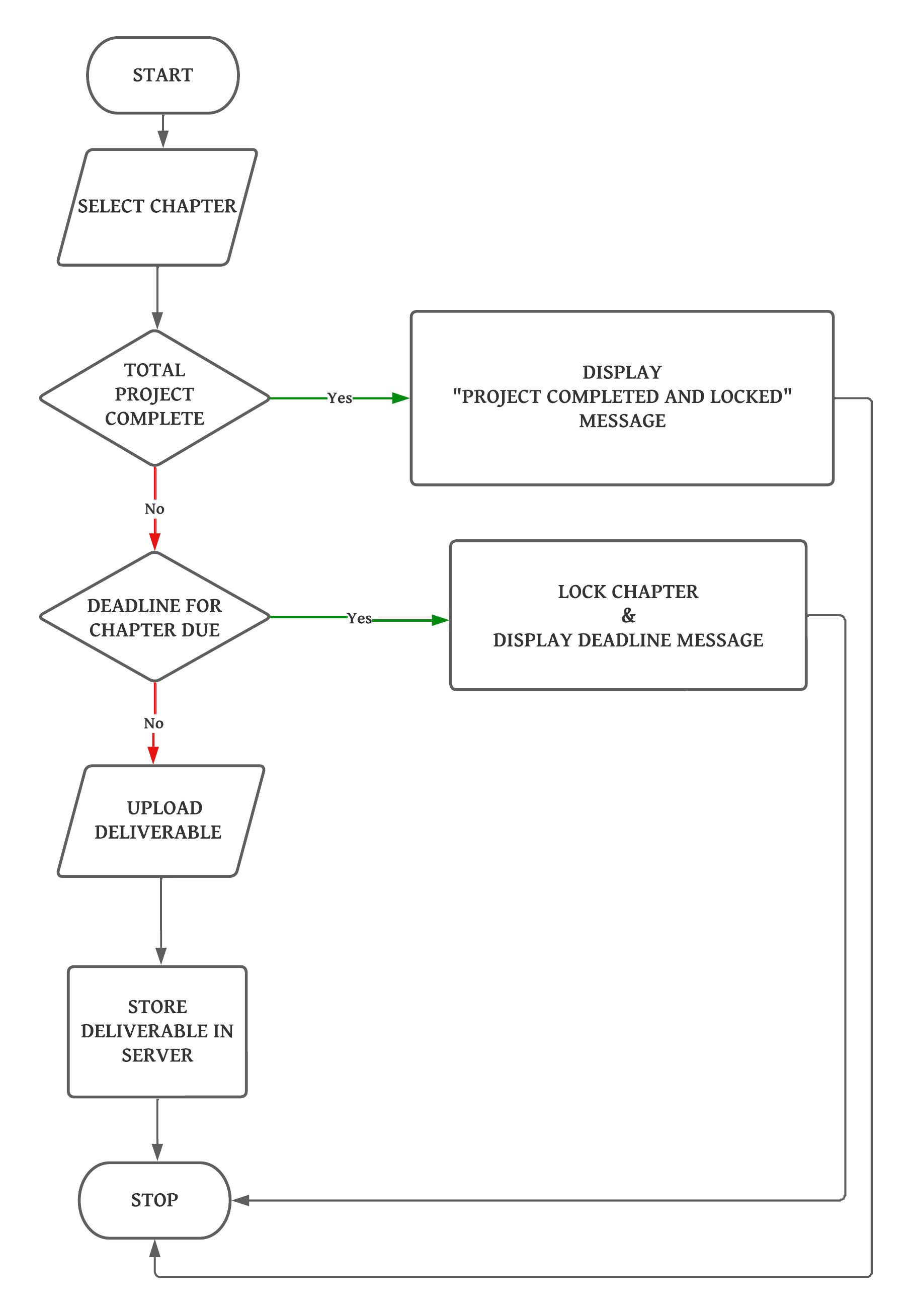
**3.4.1 Flow Charts**

****

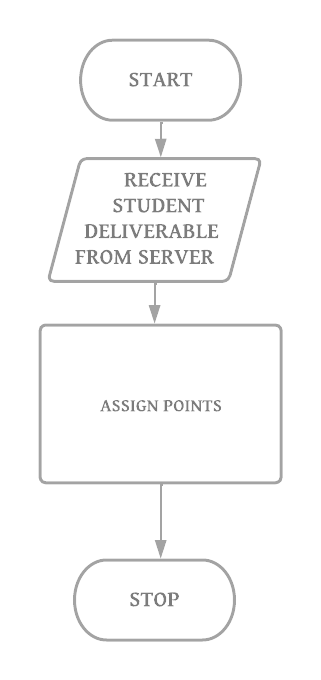
**Fig 3.5** Signup Module Flowchart

****

**Fig 3.6** Login Module Flowchart

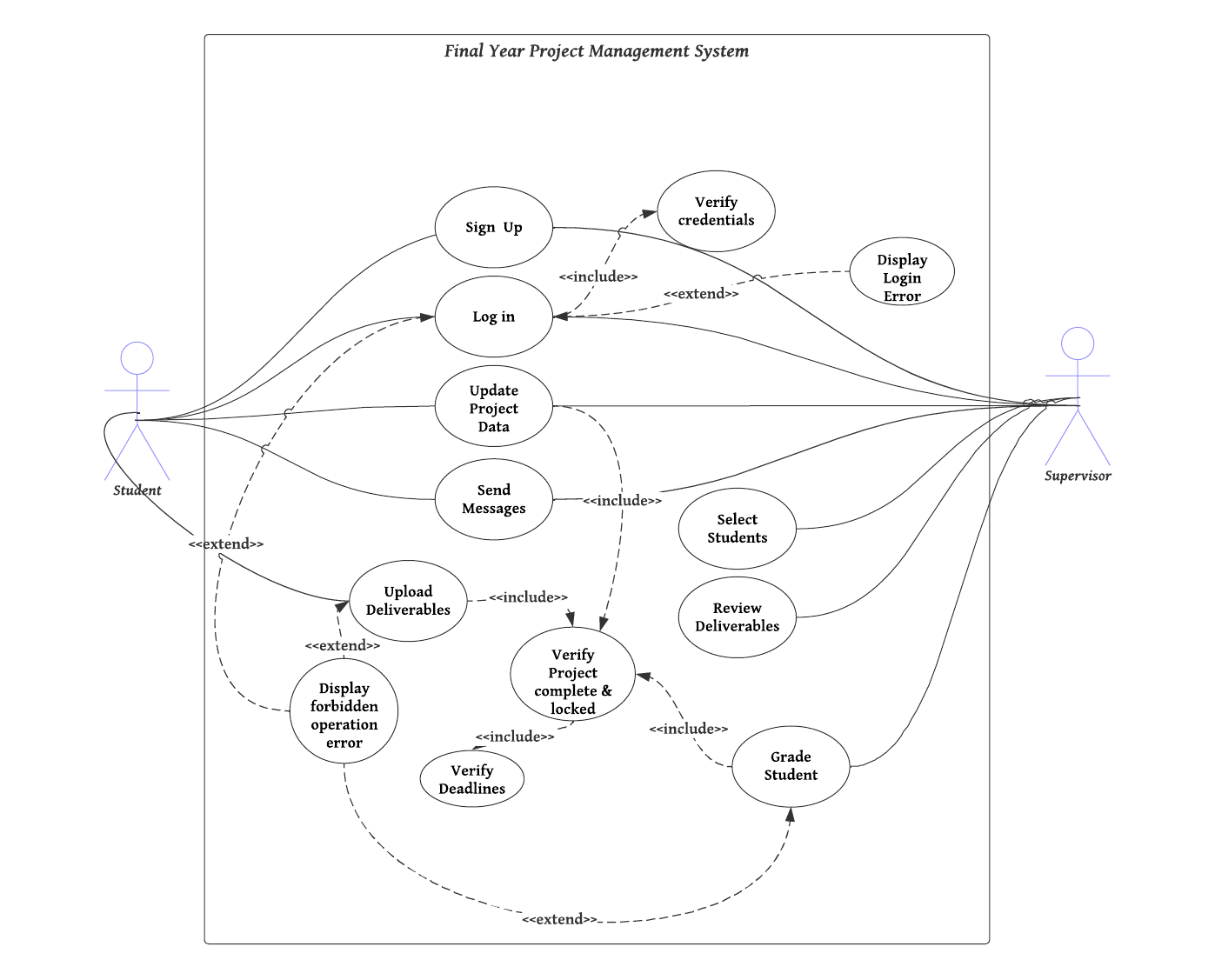
****

**Fig 3.7** Project Completion Module Flowchart



**Fig 3.8** Grading Module Flowchart

**3.4.2 UML USE CASE DIAGRAM**



**Fig 3.9** UML Use Case Diagram

**3.5 Development Tools**

In this section, we will discuss the tools that were used to implement the final year project management system. The choice of development tools is critical to the success of any software development project. The following tools were selected based on their compatibility with the requirements of the project:

* Programming language: The project was developed using JavaScript programming language. JavaScript is a versatile language that can be used for both frontend and backend development, it has a large community of users, and has a wide range of libraries and frameworks available that can simplify software development.
* Frontend framework: The frontend of the project was developed using React. React is a popular JavaScript library for building user interfaces and has a large community of users and a wide range of libraries and tools that can simplify frontend development.
* Backend framework: The backend of the project was developed using Node.js and Express.js. Node.js is a JavaScript runtime environment that allows developers to run JavaScript on the server side. Express.js is a popular Node.js framework that makes it easy to build web applications and Application Programming Interfaces (APIs).
* Integrated Development Environment (IDE): Visual Studio Code IDE was used for development. Visual Studio Code is a powerful IDE that offers features such as intelligent code completion, code highlighting, and debugging that help to speed up development and ensure code quality.
* Version Control System (VCS): Git was used for version control. Git is a widely used VCS that offers features such as branch management, merging, and conflict resolution, which are essential for collaborative software development.

These tools were instrumental in the successful development of the final year project management system.

**CHAPTER FOUR**

**IMPLEMENTATION**

**4.0 Introduction**

This chapter presents the implementation phase of the final year project management system. The previous chapter focused on the design of the system, which involved creating a detailed system specification, system architecture, and user interface design. The implementation phase involves turning the design into a functioning system, ready for testing and deployment.

The implementation phase is a critical stage in the software development life cycle. It involves writing code, integrating system components, and conducting various tests to ensure that the system meets its functional and non-functional requirements. It is important to note that the design phase lays the foundation for a successful implementation. Therefore, it is crucial to ensure that the design is accurate and complete before embarking on the implementation phase.

This chapter will provide a detailed description of the implementation process, including the tools and technologies used, and the challenges faced during the implementation. The chapter is organized into two main sections: system testing and graphical view of the project. The system testing section will describe the testing process, including the types of tests performed and the results obtained. The graphical view of the project section will provide an overview of the user interface and system components of the final year project management system.

* 1. **System Testing**

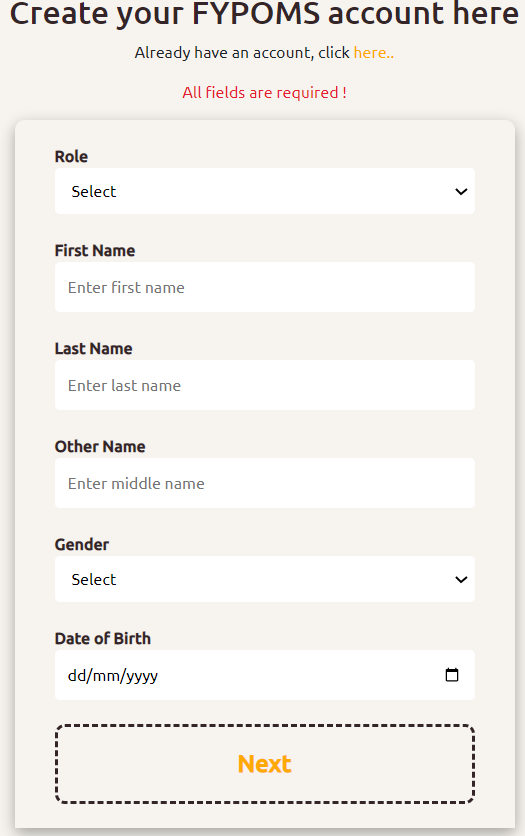
The testing phase is crucial in software development as it helps identify and fix issues in the system before it is deployed for use. In the testing phase of this project, the system was put through a rigorous testing process to ensure it met the stakeholders' requirements.

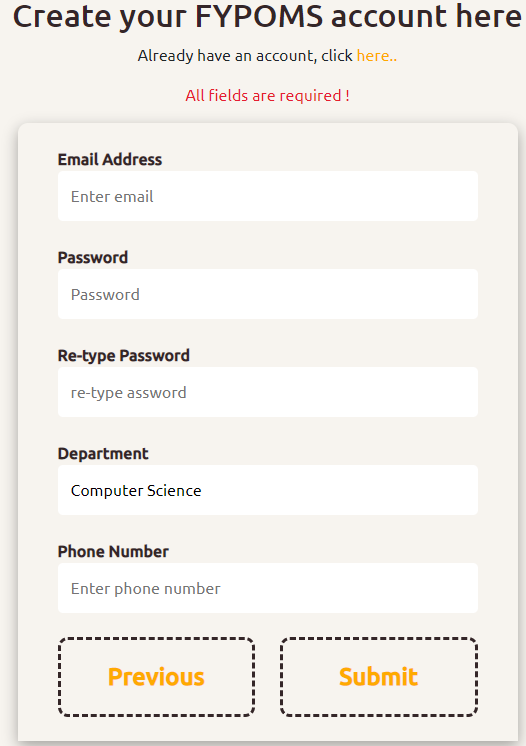
The testing process involved making students sign up first before supervisors could start selecting students to supervise. This was done to ensure that the system's user registration and login functionalities were working as intended.

During the testing phase, several issues were identified that were taken note of by the development team. These issues included problems with the user interface, system responsiveness, and data validation. The development team took note of these issues and addressed them to ensure that the system met the stakeholders' requirements.

The system was retested to ensure that the identified issues have been fixed and the system is fully functional. The system passed the testing phase and was deployed for use by the stakeholders.

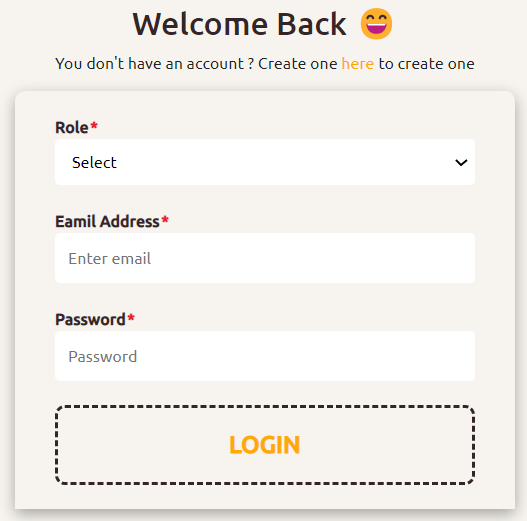
* 1. **Graphical View of The Project**
* **Signup module** - This module is responsible for allowing users to create new profiles by entering their personal details, such as their name, email address, and phone number, matric number, etc. Upon successful registration, the system generates a unique token for each user and stores their data securely to prevent unauthorized access.

****

****

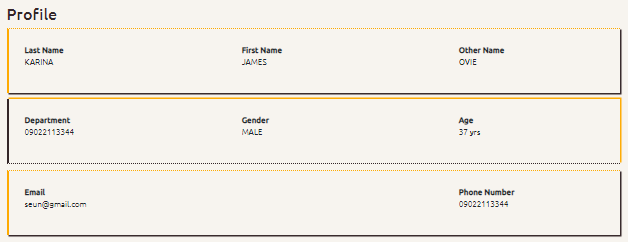
**Fig 4.1** Visual Interface of Signup Module

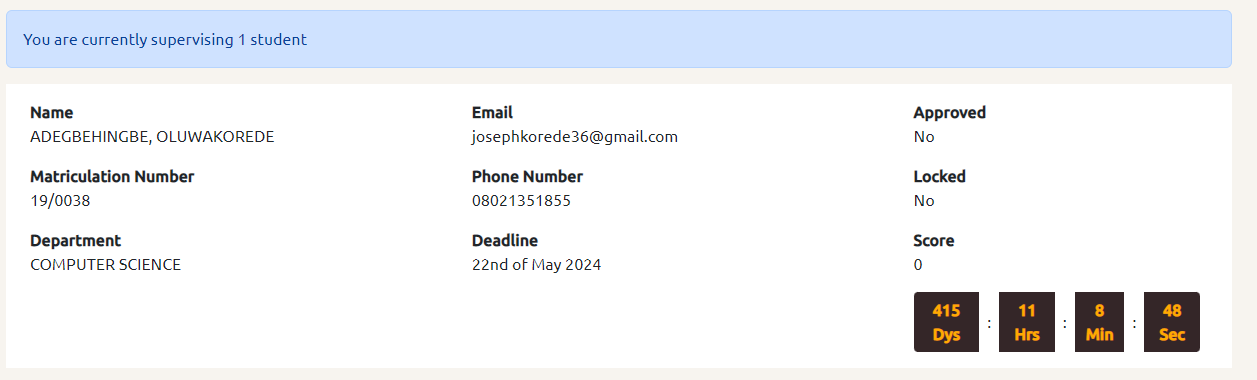
**Login module** - This module provides a login page for already registered users to enter their email and password. The system then authenticates their credentials and authorizes access to the system based on the user's role. It also restricts unauthorized access to sensitive user data.

****

**Fig 4.2** Visual Interface of Login Module

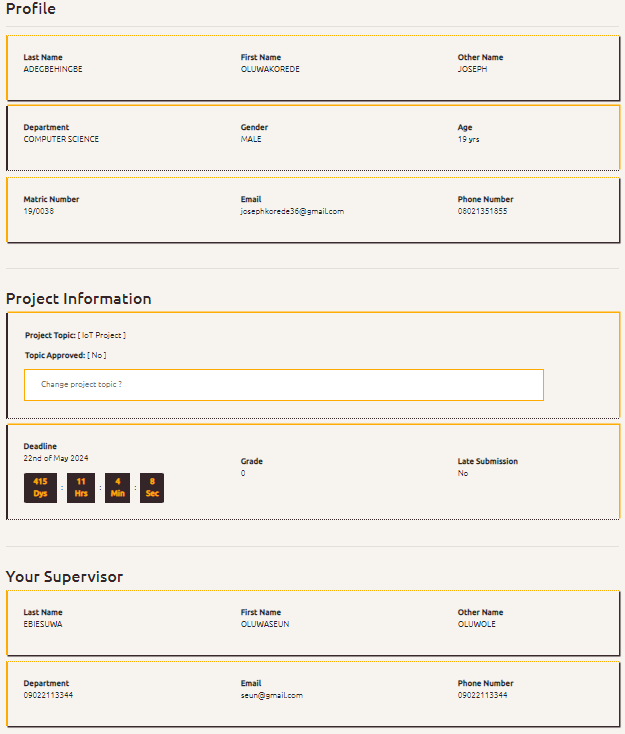
* **Supervisor dashboard/view module** - This module allows supervisors to view a list of students they are currently managing. Each list item serves as a link to detailed information about each student's project progress and development.





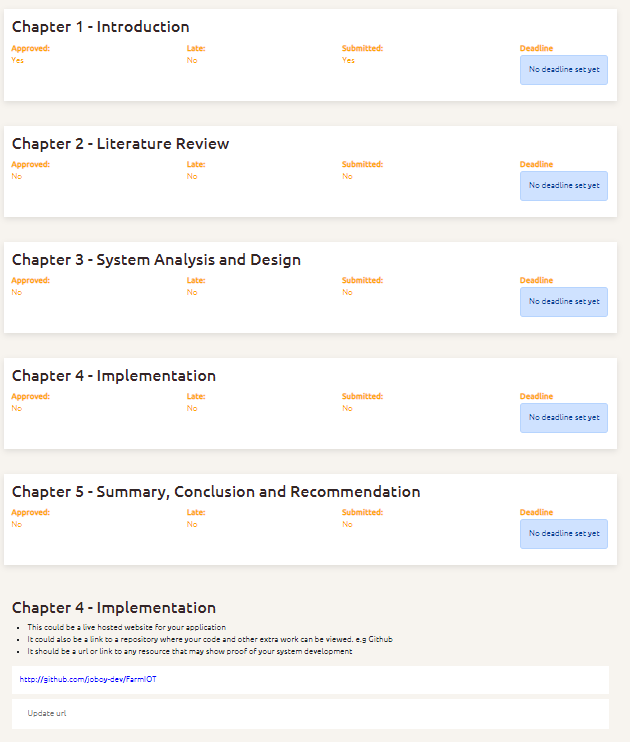
**Fig 4.3** Visual Interface of Supervisor Dashboard Module

* **Student dashboard/view module** - This module provides a dashboard for students to view and manage their project progress. It displays the student's basic information, including their name, email address, and phone number along with their project progress and basic project information such as grade, project deadline set by supervisor, late submission status and their assigned Supervisors’ details.

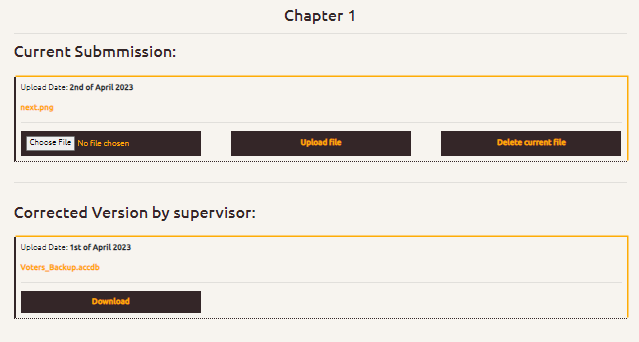


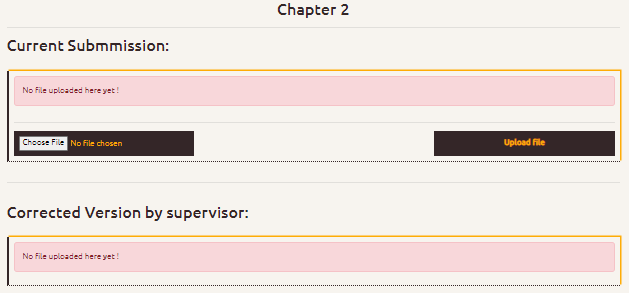
**Fig 4.4** Visual Interface of Student Dashboard Module

* **Project progress module -** This module divides each project into chapters that contain tasks to be completed. The system allows supervisors to track their students' progress by assessing uploaded documents and deliverables by downloading and approving them along with monitoring their live software development process. Embedded in this module is the grading module and software development module.

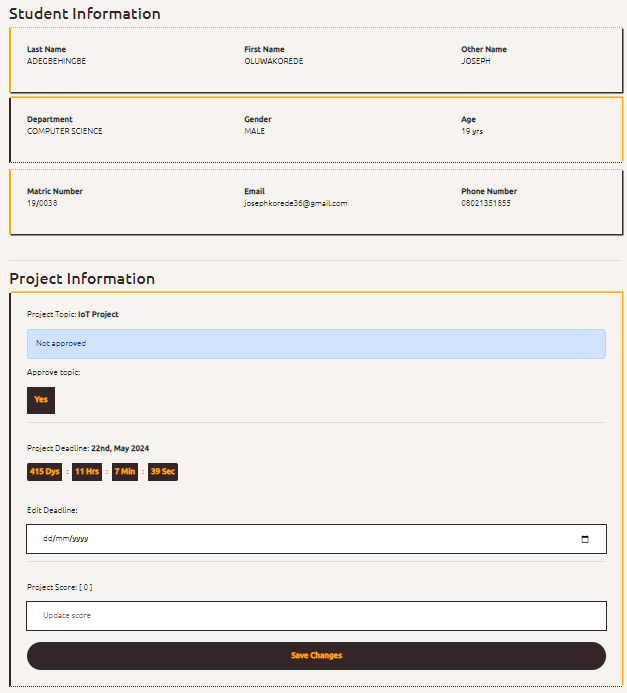
****

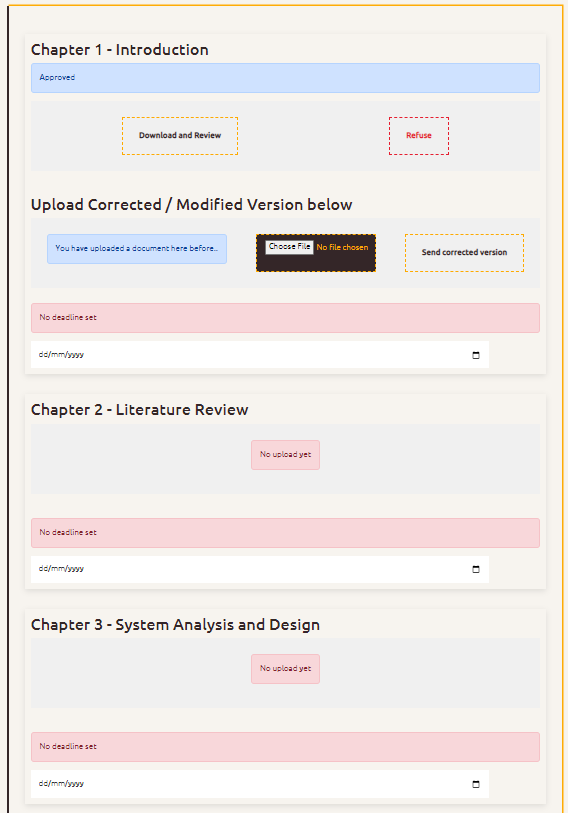
**Fig 4.5.1** Visual Interface 1 of the Student Project Module

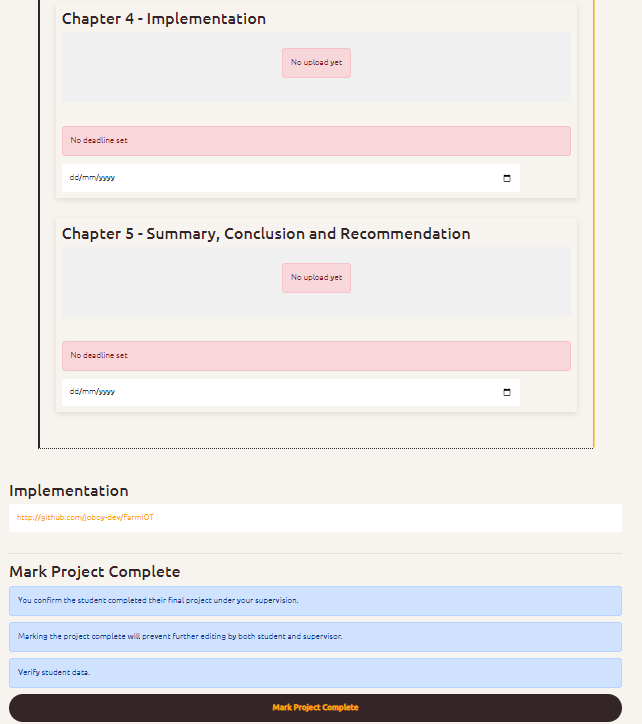




**Fig 4.5.2** Visual Interface 2 of the Student Project Module

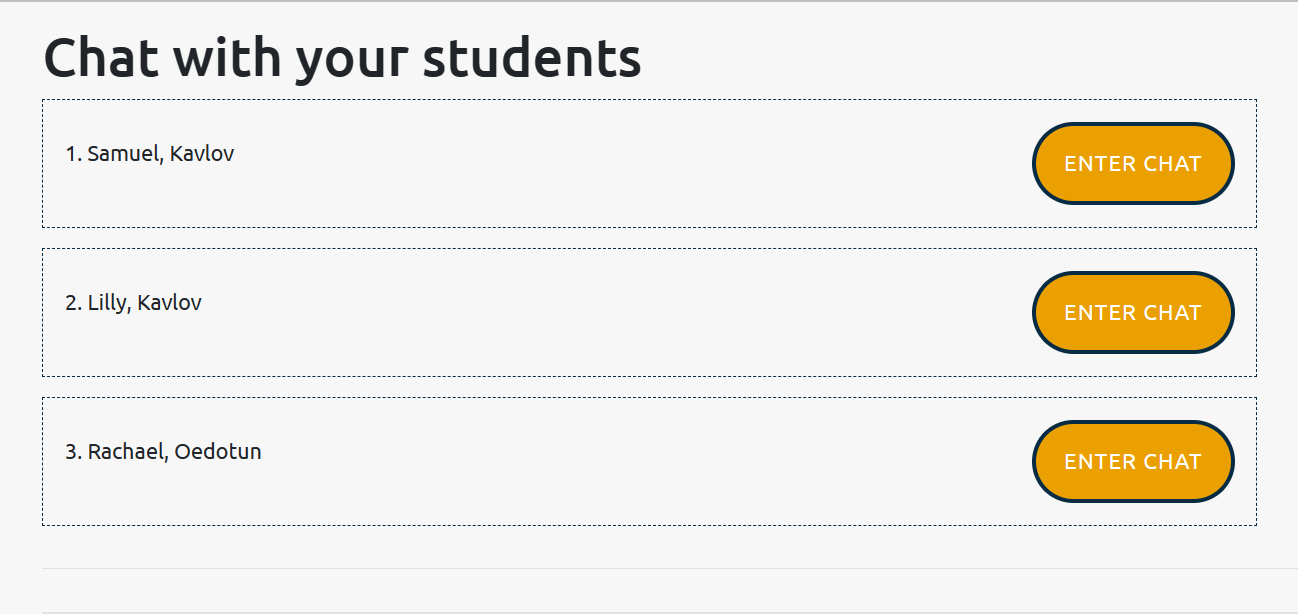


****

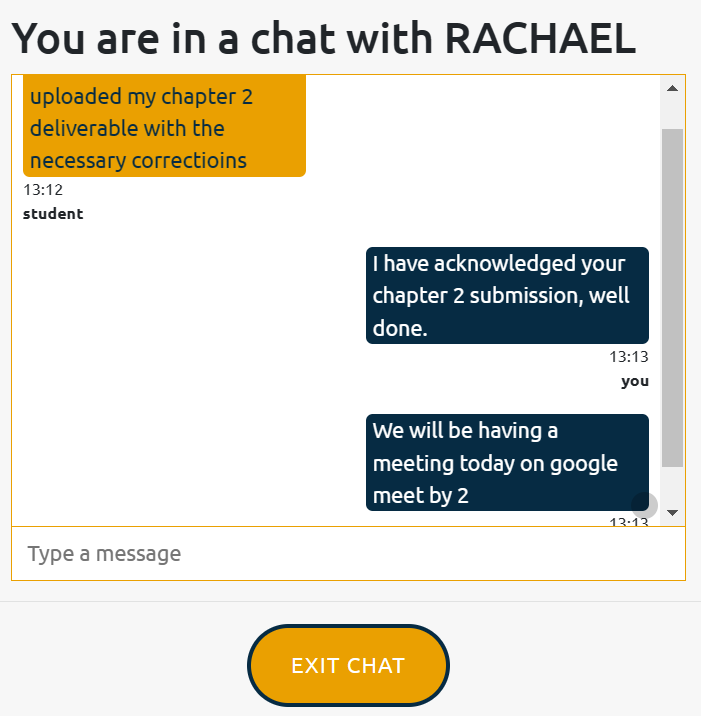
****

**Fig 4.6** Visual Interface of the Supervisor Project Module

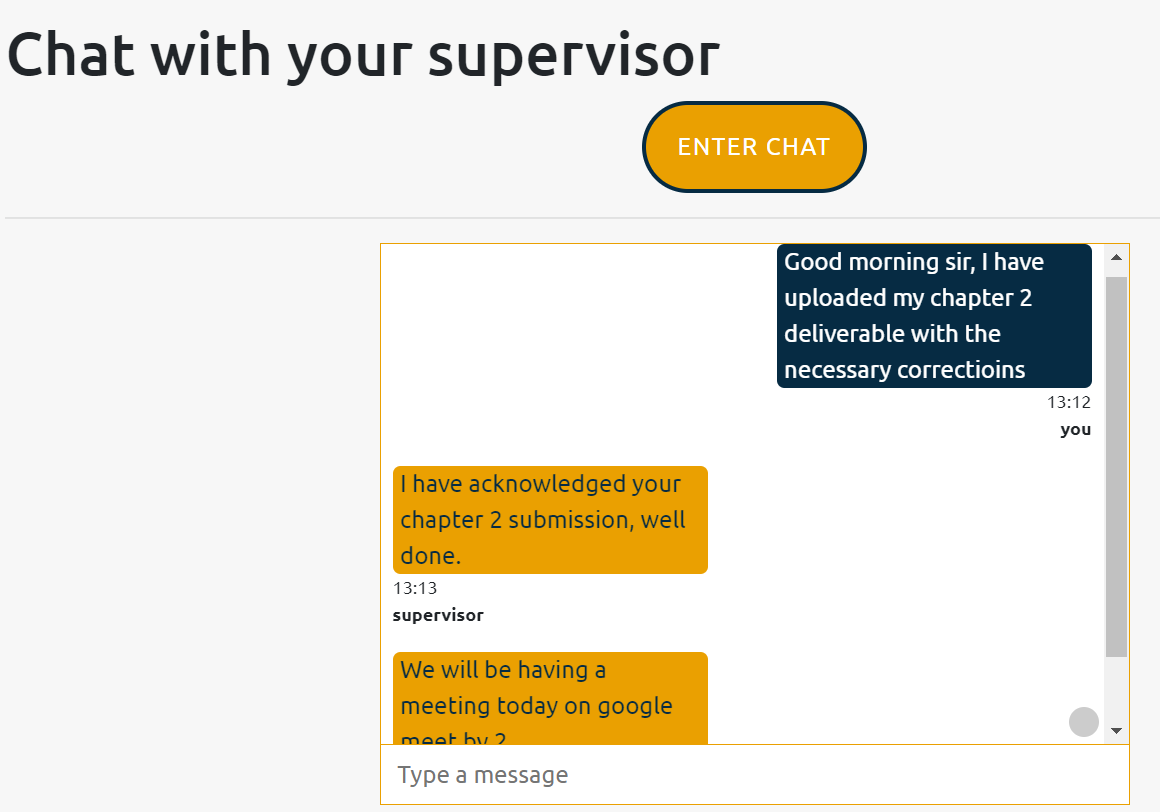
* **Communication module -** This module provides a messaging system for supervisors and students to communicate and send online meeting links and chats



**Fig 4.7.1** Supervisor Interface 1 of Communication Module



**Fig 4.7.2** Supervisor Interface 2 of Communication Module



**Fig 4.8** Student Interface of Communication Module

**CHAPTER FIVE**

**SUMMARY, CONCLUSION AND RECOMMENDATIONS**

**5.0 Summary**

This final chapter summarizes the project and provides a conclusion based on the results of the research conducted. The study developed a web-based final year project management system to simplify the process of managing final year projects for students and supervisors. To achieve this, the development team utilized a range of tools, including JavaScript, React, Node.js, and Express.js, to create a system that is user-friendly, efficient, and effective. The project was developed in several stages, beginning with a detailed analysis of the requirements and design of the system. The project has met its objectives and has been shown to be effective in managing final year projects. The system simplifies the process of managing final year projects for students and supervisors, which will save time and effort. However, further testing and refinement may be required to ensure that the system is fully functional and meets all user requirements. Overall, the development team has successfully designed and implemented a web-based final year project management system using JavaScript, React, Node.js, and Express.js. The system is user-friendly, efficient, and effective and has the potential to make a significant contribution to the field of project management in higher education.

**5.1 Conclusion**

In conclusion, the final year project management system is an essential tool for the effective management of final year projects. The system is a web-based system/website that provides a channel for communication and collaboration between students and supervisors. The implementation of the system used a combination of several technologies, including React, Node.js, Express.js, and JavaScript. These technologies provided a robust and flexible platform for developing the system, with its frontend and backend functionalities.

The project has met its objectives of providing a solution to solve the problems associated with the manual management of final year projects. The system provides a platform for students to register, upload their project topics and deliverables. The supervisors, on the other hand, can view and select the students they wish to supervise. Additionally, the system provides a communication channel between the students and their supervisors, allowing for seamless collaboration throughout the project's duration.

The system's success is evident in the positive feedback received from its users, which attests to its effectiveness in facilitating the management of final year projects. The implementation of the final year project management system is a significant achievement, and it is recommended that it be adopted by institutions of higher learning to simplify and enhance the management of final year projects.

**5.2 Recommendation**

Based on the findings of the study and the challenges encountered during the implementation, the following recommendations are proposed for future development:

* **Grouping of Students:** It is recommended that future implementation of the system should include grouping students in group of threes’ to be managed by a supervisor. This will ease the workload on supervisors and ensure that students receive equal attention and guidance.
* **Use of Google Cloud Storage API:** To improve file storage in the backend, it is recommended that the implementation should include the use of Google Cloud Storage API. This will provide a more secure and scalable storage option, with features such as automatic data redundancy and geo-redundancy.
* **Multi-departmental implementation:** It is recommended that the implementation should cut across borders, and not just a department, such that the entire institution can generically benefit from this system. This will improve collaboration and communication across different departments and ensure that resources are maximized.

By implementing these recommendations, the final year project management system will be more efficient, more user-friendly and have a wider scope of use, ultimately contributing to the academic success of students and the productivity of supervisors.

**5.3 Limitations to The Study**

Although the final year project management system was developed to meet the requirements of the project stakeholders, there were some limitations that were encountered during the implementation process. The following are some of the limitations that should be taken into consideration:

* The system was designed to be used within the department of computer science as a form of testing this prototype, and therefore, it may not be suitable for other departments in the institution without modifications to meet their unique requirements and standards for handling final year projects.
* The system is not designed to handle a large volume of data. It may become slow when used by a large number of students and supervisors.
* The system is dependent on Internet connection, and as such, the functionality of the system is subject to network availability and stability.
* The system does not have an automatic backup system. In case of a system failure or any other data loss, it will be challenging to retrieve lost data.

These limitations may have an impact on the effectiveness of the system, and as such, measures should be put in place to address these limitations.

**5.4 Suggestions for Further Research**

This project has developed a functional final year project management system that has the potential to improve the management of final year projects in the Department of Computer Science. However, there are areas that require further studies or research that can improve the system even more. Some of these areas include:

* **Integration with cloud-based storage solutions:** While the system provides a basic file storage feature, further studies can be conducted to integrate cloud-based storage solutions such as Google Cloud Storage API to enhance the system's data storage and accessibility.
* **Implementation across departments:** The system has been designed to meet the needs of the Department of Computer Science, but similar systems can be implemented in other departments. Further studies can be conducted to develop similar systems for other departments to improve the management of final year projects across the institution.
* **User experience testing:** Although the system was tested by potential users, further studies can be conducted to obtain feedback on the user experience and interface design. This can help identify areas that need improvement and lead to the development of more intuitive and user-friendly interfaces.
* **Security testing:** While security was considered in the development of the system, further studies can be conducted to test the system's resilience against potential security threats and attacks.

In conclusion, these are just a few areas where further studies or research can be conducted to improve the final year project management system. It is important to continue to research and develop the system to keep up with the changing needs and requirements of the academic institution and its stakeholders.

**REFERENCES**

Abdul Wahab, R. B. (2012). Final Year Project Online Management System

(FYPOMS): Client and Server Design.

*All about the Iterative Design Process | Smartsheet*. (2019, January 2). All About the

Iterative Design Process | Smartsheet. <https://www.smartsheet.com/iterative-process-guide>

Auksorncherdchoo, S., & Sukstrienwong, A. (2018). *Software development for*

*student behavior tracking system*. International Journal of Interactive Digital Media, 6(1), 32-37. https://doi.org/10.12928/ijidm.6.1.04

Bakar, M. A., Jailani, N., Shukur, Z., & Yatim, N. F. M. (2011). Final year

supervision management system as a tool for monitoring Computer Science projects. *Procedia-Social and Behavioural Sciences*, *18*, 273-281.

Bello, M., Asogbon, M. G., Oyelade, O. J., & Akinwale, A. T. (2020). *Design and*

*implementation of an online final year project management system*. International Journal of Emerging Technologies in Learning, 15(9), 106-118. doi: 10.3991/ijet.v15i09.12290.

*Design and implementation of a web-based final year students research project*

*supervision management system*. (n.d.). Design and Implementation of a Web-based Final Year Students Research Project Supervision Management System. https://www.classgist.com/projectdetails.aspx?id=745

Gao, S., Wang, Q., & Yang, Y. (2019). *Design and implementation of a final year*

*project management system based on C/S architecture*. Journal of Physics: Conference Series, 1237, 042006. doi: 10.1088/1742-6596/1237/4/042006

Gomaa, M. S., & Mohamed, S. M. (2020). *The Development of an Online Final Year*

*Projects Management System*. Journal of Engineering Science and Technology, 15(3), 1705-1720.

Graf, Del Monte, Allendorff, Mendonca-McCoy, & R. VanLandingham. (n.d.).

*Student Tracking Systems Can Be Used to Enhance Graduation and Retention Rates*. <https://oppaga.fl.gov/Documents/Reports/06-48.pdf>

H. Bers, E. T. (n.d.). *ERIC - ED307925 - Using Student Tracking Systems Effectively.*

*New Directions for Community Colleges, Number 66., New Directions for Community Colleges, 1989*. ERIC - ED307925 - Using Student Tracking Systems Effectively. New Directions for Community Colleges, Number 66., New Directions for Community Colleges, 1989. <https://eric.ed.gov/?id=ED307925>

Iruka, J. O., Akinyemi, O. J., & Oyewunmi, O. T. (2019). *Challenges and*

*Opportunities for Implementing a Final Year Project Management System.* International Journal of Advanced Computer Science and Applications, 10(10), 99-104.

Ismail, S. I., Abdullah, R., Kar, S. A. C., Fadzal, N., Husni, H., & Omar, H. M. (2017,

December). Online project evaluation and supervision system (opens) for final year project proposal development process. In *2017 IEEE 15th Student Conference on Research and Development (SCOReD)* (pp. 210-214). IEEE.

Jung, H., Jun, W., & Gruenwald, L. (2001, July). A design and implementation of

web-based project-based learning support systems. In *International Conference Human Society@ Internet* (pp. 354-367). Springer, Berlin, Heidelberg.

Leung, C. H., Lai, C. L., Yuan, T. K., Pang, W. M., T. Tang, J. K., Ho, W. S., &

Wong, T. L. (n.d.). *The Development of a Final Year Project Management System for Information Technology Programmes*. The Development of a Final Year Project Management System for Information Technology Programmes | SpringerLink. https://link.springer.com/chapter/10.1007/978-3-662-46158-7\_9

Li, W., & Li, X. (2019). *Enhancing student engagement and learning through*

*assessment tracking systems*. Journal of Educational Technology Development and Exchange, 12(1), 1-14.

Mark, K. (n.d.). *The Need For A Final Year Project Management System*. (PDF) the

Need for a Final Year Project Management System | Kajubi Mark - Academia.edu. <https://www.academia.edu/73230906/The_Need_For_A_Final_Year_Project_Management_System>

Mohamed, M. A. (2017). An implementation of final year project management

system: A case study at Universiti Sultan Zainal Abidin. *World Applied Sciences Journal*, *35*(8), 1246-1251.

Ndupu, Ogege, & Ojoboorun. (2020). *A Final Year Project Management System*.

Ogbaudu, O. (2021). *Design and Implementation of Staff to Student Interactive*

*Platform*.

Olajubu, E. O., Olakanmi, O. E., & Oyewole, S. O. (2017). *Design and*

*Implementation of a Final Year Project Management System for Computer Science Students.* International Journal of Advanced Computer Science and Applications, 8(4), 60-66.

R. L. (n.d.). *Barriers to Learning in Distance Education: Jill M. Galusha*. Barriers to

Learning in Distance Education: Jill M. Galusha. <http://jscarcella.academic.csusb.edu/E-Text_Shea/She504/Assignments/WR3_DL_barriers.htm>

Ragan, L. C. Good teaching is good teaching: An emerging set of guiding principles

and practices for the design and development of distance education. *Cause Effect*, *22*, 20-24.

Rahman, M. R., Hasan, M. K., Muntasir, M. K., & Chowdhury, M. F. I. (2017). *Final*

*year project management system: A case study in Bangladesh. International Journal of Information and Education Technology*, 7(1), 66-69. doi: 10.18178/ijiet.2017.7.1.822

S. W. (2022, July 28). *The Benefits of Web-Based Systems for Business*. Aezion.

https://www.aezion.com/blogs/the-benefits-of-web-based-systems-for-business/

Scerbakov, N. (n.d.). *Web based training: What do we expect from the system*. (PDF)

Web Based Training: What Do We Expect From the System | Nick Scerbakov - Academia.edu. <https://www.academia.edu/28931376/Web_based_training_What_do_we_expect_from_the_system>

*The Structure of the Internet - EnioF.com*. (n.d.). The Structure of the Internet –

EnioF.com. http://www.eniof.com/comp/ascomp/comp2/the-structure-of-the-internet

*Web Hosting Services, VPS Servers and Domain Names by NTC Hosting*. (n.d.). What

Is a website - a Collection of Web Pages. <https://www.ntchosting.com/encyclopedia/internet/website/>

**APPENDIX**

1. **Signup module source code**

// Student Signup

const create\_student = async (req, res) => {

const {names,department, gender, DOB, phone, role, score, email, password, matricID } = req.body;

try {

await Student.create({names,department, gender, DOB, phone, role, score, email, password, matricID, topic\_approved: false, assigned: false, late: false}

res.status(200).json({ message: "Student profile created" });

} catch (error) {

const errors = handleSignupErrors(error);

res.status(400).json(errors);

}

};

// Supervisor Signup

const create\_supervisor = async (req, res) => {

const { names, department, gender, DOB, phone, role, email, password } =

req.body;

try {

await Supervisor.create({names, department, gender, DOB, phone, role,

email, password, studentsSUP: []});

res.status(200).json({ message: "Supervisor profile created"});

} catch (error) {

const errors = handleSignupErrors(error);

res.status(400).json(errors);

}

};

1. **Login module source code**

const login = async (req, res) => {

const { email, password } = req.body;

try {

const user = await Student.login(email, password);

const token = createToken(user.\_id);

res.status(200).json({email: user.email, token, id: user.\_id, role: user.role, DOB: user.DOB});

} catch (error) {

const errors = handleLoginErrors(error);

res.status(400).json(errors);

}};

1. **Source code for supervisor dashboard module**

// Get students assigned to a supervisor

const get\_my\_students = async (req, res) => {

const { supervisorId } = req.params;

try {

const supervisor = await Supervisor.findById(supervisorId);

if (!supervisor) {

throw new Error("this supervisor data does not exist");

}

const studentsArr = supervisor.studentsSUP;

if (studentsArr.length) {

const myStudents = await Promise.all(

studentsArr.map(async (studentId) => {

return await Student.findById(studentId);

})

);

res.status(200).json({

message: "Student data retrieved successfully", myStudents

});

} else

res.status(200).json({

message: "You currently do not have any students under supervision",

});

} catch (error) {

res.status(400).json({ error: error.message });

}};

1. **Source code for student dashboard module**

// Get a single student’s data

const get\_student = async (req, res) => {

const { studentId } = req.params;

try {

const student = await Student.findById(studentId);

res.status(200).json({ student });

} catch (error) {

res.status(404).json({ error: error.message });

}};

1. **Source code for project progress module**
   1. **Student Upload Document to Chapter for Submission**

const upload\_file\_to\_chapter = async (req, res) => {

const { chapterId, studentId } = req.params;

const label = `p${chapterId}.file`;

const submission = `p${chapterId}.submitted`;

try {

const student = await Student.findById(studentId);

if (!student) {

throw new Error("this student data does not exist");

}

const newFile = req.file;

const File = await Upload.create({

filename: newFile.filename,

originalname: newFile.originalname,

mimetype: newFile.mimetype,

path: newFile.path,

size: newFile.size,

uploader: studentId,});

await Student.findByIdAndUpdate(

studentId,

{ $push: { uploads: File.\_id } },

{ new: true }

);

await Project.findOneAndUpdate(

{ studentID: studentId },

{ $set: { [label]: File.\_id, [submission]: true } }

);

res.status(200).json({

message: "File has been uploaded successfully",

file: req.file,

});

} catch (error) {

return res.status(404).json({ error: error.message });

}};

* 1. **Student Delete Uploaded Document from Chapter**

const delete\_file = async (req, res) => {

const { studentId, fileId, chapterId } = req.params;

const label = `p${chapterId}.file`;

const submission = `p${chapterId}.submitted`;

if (!mongoose.Types.ObjectId.isValid(fileId)) {

return res

.status(404)

.json({ "Error Message": "this file does not exist" });

}

try {

const student = await Student.findById(studentId);

const foundFile = await Upload.findById(fileId);

if (!student) {

throw new Error("this student data does not exist");

}

if (!foundFile) {

throw new Error("this file data does not exist");

}

// Check for file ownership

if (JSON.stringify(foundFile.uploader).includes(studentId)) {

// DELETE RECORDS OF FILE FROM DATABASE SCHEMAS (STUDENT&FILE)

await Upload.findByIdAndDelete(foundFile.\_id);

// DELETE FILE ID FROM STUDENT DATA

await Student.findByIdAndUpdate(

studentId,

{ $pull: { uploads: foundFile.\_id } },

{ new: true });

// Delete file from project schema

await Project.findOneAndUpdate(

{ studentID: studentId },

{ $set: { [label]: null, [submission]: false } }

);

res.status(200).json({

message: "File has been deleted successfully",

});

// DELETE FILE FROM SERVER

fs.unlink("./uploadedFiles/" + foundFile.filename, (err) => {

if (err) {

throw new Error(err.message);

}

});

} else {

throw new Error("Unauthorised access to foreign file");

}

} catch (error) {

return res.status(404).json({ error: error.message });

}};

* 1. **Student Update Project Topic and Software Link**

const update\_details = async (req, res) => {

const { studentId } = req.params;

const { topic, softwareLink } = req.body;

try {

const student = await Student.findById(studentId);

if (!student) {

throw new Error("this student data does not exist");

}

if (topic) {

await Topic.findOneAndUpdate(

{ studentID: studentId },

{ $set: { title: topic } },

{ new: true }

);

await Student.findByIdAndUpdate(

studentId,

{ $set: { project\_topic: topic } },

{ new: true }

);

res.status(202).json({ updatedTopic: topic });

}

if (softwareLink) {

await Project.findOneAndUpdate(

{ studentID: studentId },

{ $set: { softwareLink: softwareLink } },

{ new: true }

);

res.status(202).json({ updatedLink: softwareLink });

}

} catch (error) {

res.status(400).json({ error: error.message });

}};

* 1. **Supervisor Update Student Project Data**

const update\_student\_student\_status = async (req, res) => {

const { studentId, supervisorId } = req.params;

const { queryScore, queryDeadline, queryTopicApprove } = req.body;

try {

// Find the student and supervisor

const student = await Student.findById(studentId);

const supervisor = await Supervisor.findById(supervisorId);

// Check if the student and supervisor exist

if (!student) {

throw new Error("this student data does not exist");

}

if (!supervisor) {

throw new Error("this supervisor data does not exist");

}

// Update data in Student Schema

const updateStudentQuery = await handleStudentUpdateData(

queryScore, queryDeadline, queryTopicApprove);

const updatedStudent = await Student.findOneAndUpdate(

{id: studentId},{$set: updateStudentQuery },{new: true}

);

res.status(202).json({

message: "Student data updated successfully",

updatedStudent,

});

} catch (error) {

res.status(400).json({ error: error.message });

}};

1. **Middleware functions**

**// Supervisor Student Project Deadline, Chapter Deadlines and Mark Project Complete**

const update\_student\_project\_status = async (req, res) => {

const { studentId, supervisorId } = req.params;

const { chapterID, queryComplete, chapterDeadline } = req.body;

try {

// Find the student and supervisor

const student = await Student.findById(studentId);

const supervisor = await Supervisor.findById(supervisorId);

// Check if the student and supervisor exist

if (!student) {

throw new Error("this student data does not exist");

}

if (!supervisor) {

throw new Error("this supervisor data does not exist");

}

// Update data in Project Schema

const current\_overall\_deadline = student.deadline;

const updateProjectQuery = await handleProjectUpdateData(

queryComplete,

chapterDeadline,

current\_overall\_deadline,

chapterID

);

let updatedProject;

if (chapterID) {

const label\_deadline = `p${chapterID}.deadline`;

updatedProject = await Project.findOneAndUpdate(

{ studentID: studentId },

{

$set: {

[label\_deadline]: updateProjectQuery.deadline,

projectComplete: updateProjectQuery.projectComplete,

},

},

{ new: true }

);

}

if (chapterID === undefined) {

updatedProject = await Project.findOneAndUpdate(

{ studentID: studentId },

{

$set: {

projectComplete: updateProjectQuery.projectComplete,

},

},

{ new: true }

);

}

res.status(202).json({

message: "Student project data updated successfully",

updatedProject,

});

} catch (error) {

res.status(400).json({ error: error.message });

}};

1. **Source code for communication module**

const express = require("express");

const mongoose = require("mongoose");

const http = require("http");

const { Server } = require("socket.io");

const cors = require(“cors”)

require("dotenv").config();

const Room = require("./models/Room");

const Chat = require("./models/Chat");

const app = express();

app.use(cors())

mongoose.connect(process.env.MONGO\_URI, () => {

const server = http.createServer(app);

const io = new Server(server, {

cors: {

methods: ["GET", "POST"],

origin: `http://localhost:3000`,

},

});

**// handle socket.io events here**

io.on("connection", async (socket) => {

let chat\_results;

socket.on("join\_room", async (data) => {

console.log(data)

await socket.join(data);

// Get chats from the database

chat\_results = await Chat.find({ roomID: data })

.limit(100)

.sort({ \_id: 1 })

.exec();

socket.emit("room\_chats", chat\_results);

});

**// Receive message from client**

socket.on("send\_message", async ({ roomId, author, message, time }) => {

console.log(roomId, author, message, time);

await Chat.create({

roomID: roomId,

author,

message,

time,

});

io.in(roomId).emit("receive\_message", { roomId, author, message, time });

});

**// Disconnect Socket**

socket.on("disconnect", () => {

console.log("client disconnected");

});

});

server.listen(process.env.PORT, () => {

console.log(

`Connected to database and listening to port ${process.env.PORT}`

);

});

});