CHAPTER FOUR

SYSTEM DESIGN

4.1 Introduction

System design is the process of defining the architecture, components, modules, interfaces, and data flow of a system to meet specific requirements and ensure it functions efficiently and reliably (Eriksson et al., 2003). This chapter focuses on the design of the Student Project and Dissertation Management System, based on the data gathered and analyzed in the previous chapter. It outlines how the system is structured to meet user needs, detailing the planned interactions between users and system components. The design serves as a blueprint for development, ensuring that the system is functional, user-friendly, and scalable.

4.2 Requirement Analysis

Requirement analysis is the process of identifying, documenting, and understanding the needs and expectations of users to ensure that the system fulfills its intended purpose. It serves as a foundation for system design by translating user input into functional and non-functional requirements. In this study, the analysis focused on the key stakeholders students, supervisors, and administrative staff to gather detailed insights into their challenges and needs related to project and dissertation management. According to Olsson & Joelsson (2019), effective requirement analysis not only improves system accuracy but also reduces the risk of project failure by aligning technical solutions with real-world problems. The requirements collected through questionnaires, interviews, and focus group discussions were carefully reviewed and categorized to guide the development of features such as project submission, supervisor assignment, and progress tracking.

4.2.1 System Requirements

System requirements include all tool and data that were required for the development of the system. The requirements were data, hardware and software as explained below:

4.2.1.1 Data Requirements

Data requirements refer to the specific information needed for the development and functioning of the system. The required data included all relevant details exchanged between students, supervisors, and departmental coordinators, such as project titles, submission dates, supervision

records, and feedback. This information was essential for implementing key system features like project tracking and communication. The data was primarily gathered through questionnaires and interviews conducted during the data collection phase, ensuring that the system reflects actual user needs and real academic workflows.

4.2.1.2 Hardware requirements

Hardware requirements refer to the physical devices necessary to support the development, deployment, and use of the system. The system was developed and tested using a standard personal computer with 8GB RAM, a 2.81 GHz processor, and 500GB of hard disk space. On the user side, the system is web-based and can be accessed via any computer or mobile device with internet connectivity and a modern web browser. Servers used for hosting the application must support adequate storage, memory, and processing capabilities to ensure smooth performance and user access.

4.2.1.3 Software Requirements

Software requirements include all the tools and platforms necessary for developing, designing, testing, and running the system. For this study, the development was carried out using Visual Studio Code as the primary code editor, supported by XAMPP as a local server environment to run MySQL. The backend was developed using django, while react was used for the frontend. MySQL served as the database management system for storing and managing data. For system modeling and diagram creation, draw.io was used to design use case diagrams, class diagrams, and sequence diagrams. The system is web-based, so a modern web browser such as Google Chrome or Mozilla Firefox is required to access and test its functionalities. Additionally, a web server is needed for hosting the live version of the application.

Table 3.2 Software requirements

Software	Purpose of the software
Microsoft Word 2016 Microsoft Power Point 2016	Tools used to write, create and modify documents and prepared the presentation documents
XAMPP server	This is the server that was used to allow interaction of the application to interact with the database (MySQL)
Visual Studio Code	This is the software that was used for implementing the web portal. It was code editor
Draw.io	This is the software that was used for systems analysis and design
Google chrome and Mozilla Firefox	Web browser were used for finding articles and some of resources dealing with system Also for access and test the system functionalities

4.2.2 User Requirements

It refers to all things that user require from the system. These requirements may be functional or non-functional requirements, and explained as follows;

4.2.2.1 Functional Requirements

These are functionalities that the system is expected to perform. It may also be defined as a function of the system or its component (Madaus et al., 2019). The functional requirements of this system are:

i. The system shall allow user to register and login
ii. The system shall allow administrator to manage both supervisor and student
iii. The system shall allow administrator to post project related updates

iv. The system shall allow both student and supervisor to view all the project related updates posted by the administrator
v. The system shall allow student registering of title
vi. The system shall allow administrator to assign supervisor
vii. The system shall student to view the assigned supervisor
viii. The system shall allow supervisor to view the assigned student
ix. The system shall allow student to upload documents such as reports, SRS
x. The system shall allow student to book for consultation with supervisor
xi. The system shall allow supervisor to cross-check and accept consultation
xii. The system shall allow supervisor to provide feedback on uploaded document by student
xiii. The system shall allow student to view feedback provided by supervisor
xiv. The system shall generate progress reports for each student based on milestones and submitted documents.
xv. The system shall support role-based access control to ensure users access only authorized features and data.
xvi. The system shall allow the coordinator to set project submission deadlines.

4.2.2.2 Non-functional requirements

Non-functional requirements are those that specify how a system operates and the values that it should have such as usability, scalability and performance. Functional requirements are concerned with how the system will carry out a certain task (Madaus et al., 2019). Non-functional requirements of Student Project and dissertation management system are as follows;

i. Performance: The system should be responsive and scalable to handle a large number of users and listing.
ii. Security: Implementation of the secure authentication and essential data encryption
iii. Reliability: The system should be available and reliable, minimizing downtime and ensure data integrity.
iv. Usability: The user interface should be intuitive and easy to navigate for both student, supervisor and administrator
v. Compatibility: The system should be compatible with different device and browsers

4.3 System Design

Is the way in which the flow of the system organized to enable smooth development progress. The whole system functionality drawn in the diagrams to make clear understanding on each process of the system and facilitating the interactions of the interfaces of users. The following were the designs used in this project;

4.3.1 Unified Modelling Language

Unified Modeling Language (UML) is a standardized visual language used to represent the structure and behavior of a system through various types of diagrams (Eriksson et al., 2003). The UML was used to design and communicate how different components of the Student Project and Dissertation Management System interact with each other. It helped in visualizing system processes, user roles, and data flow before actual development began. The diagrams created such as use case diagrams, class diagrams, and sequence diagrams served as blueprints that guided the development team in building a system that meets user requirements. UML was chosen for its clarity, flexibility, and wide acceptance in software engineering practices.

4.3.1.1 Use-case diagrams

Use-case diagrams are used to represent the interactions between users (actors) and the system, showing what the system is expected to do from a user's perspective. In this study, use-case diagrams were developed to identify and illustrate the main functionalities of the Student Project and Dissertation Management System, such as project submission, supervisor assignment, and progress tracking. Each use-case diagram outlines the roles of different users including students, supervisors, and administrators and their interactions with the system features. According to Eriksson et al. (2003), use-case diagrams are essential for capturing functional requirements and providing a high-level overview of system behavior, which helps both developers and stakeholders understand how the system should operate. These diagrams served as the foundation for further detailed system design and development.

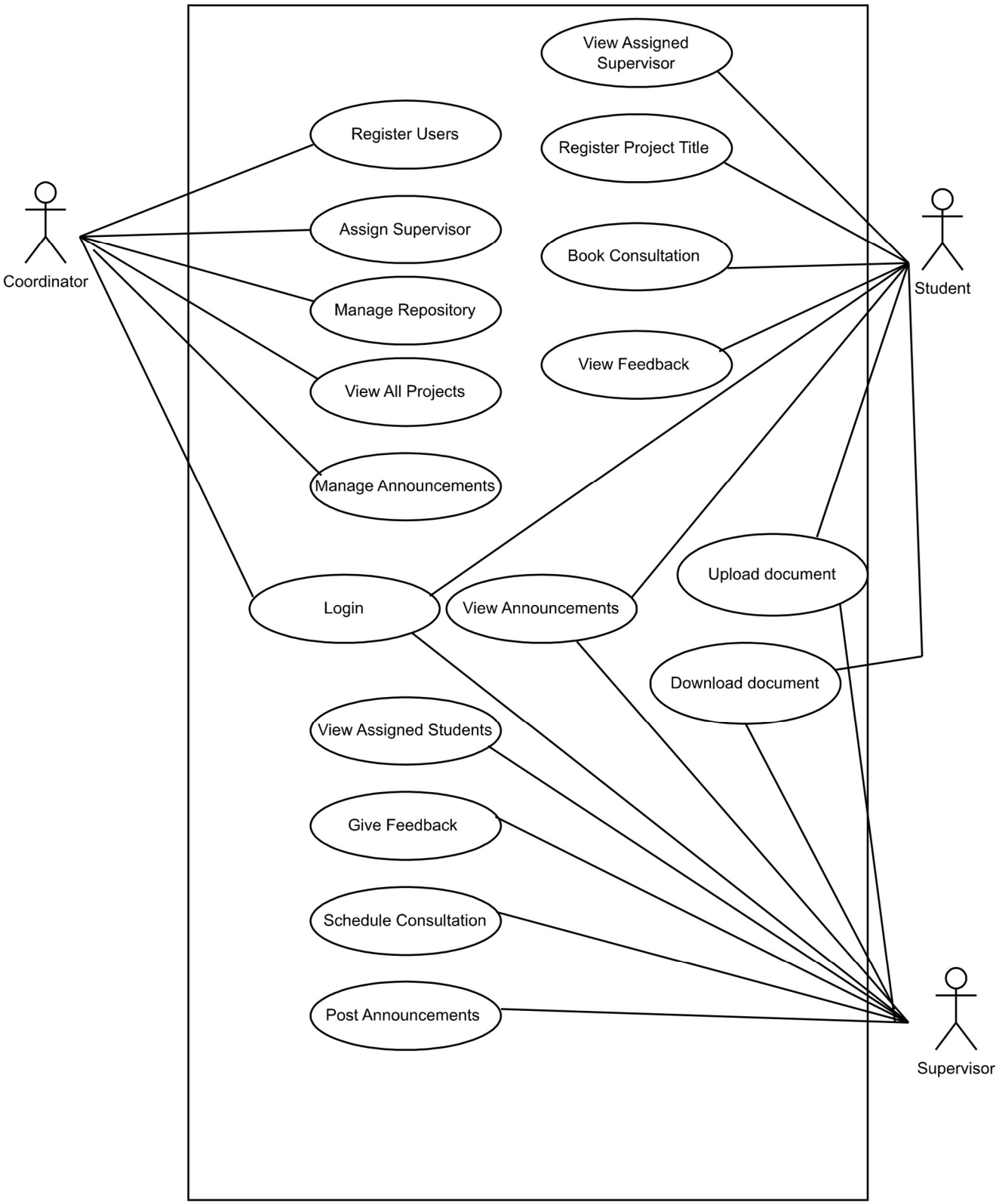

Figure 4.2 Use-case diagrams

4.3.1.2 Class diagrams

Class diagrams are used to represent the static structure of a system by showing its classes, attributes, methods, and the relationships among them. In this study, class diagrams were used to model the key entities of the Student Project and Dissertation Management System, such as Student, Supervisor, Project, and Administrator. These diagrams helped define how data is organized and how different parts of the system interact with one another. According to (Eriksson et al., 2003), class diagrams are vital in object-oriented design because they provide a clear blueprint of the system's data structure, which supports consistency throughout the development process. By using class diagrams, the project team ensured that each component of the system was well-defined and aligned with the requirements identified during the analysis phase.

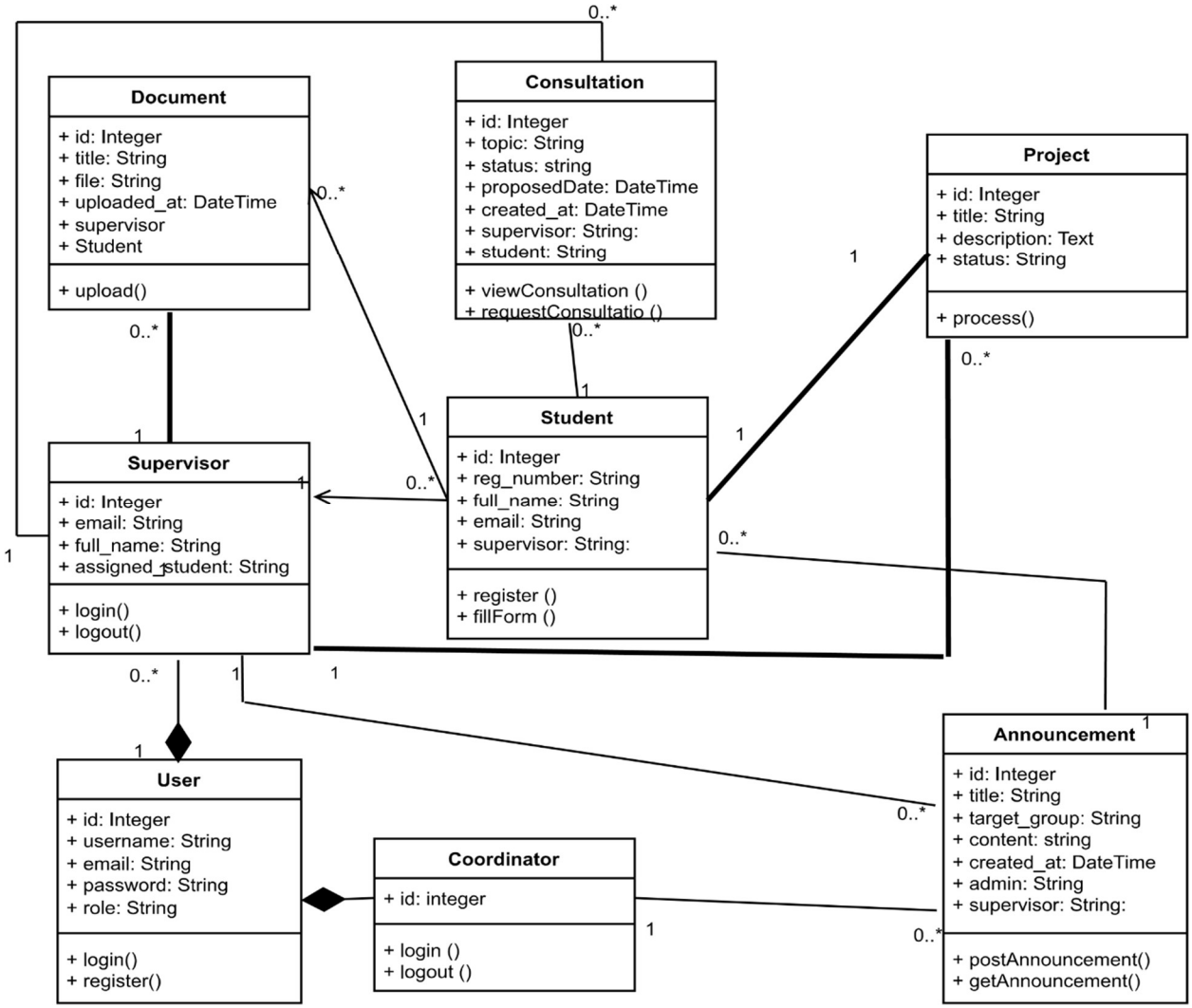

Figure 4.3 Class diagrams