



Wolkite University

College of Computing and Informatics

Department of Information System

**Final Project on web based continuing and Distance Education
Management System for wolkite university**

Group members

| <u>Name</u> | <u>ID Number</u> |
|------------------------|------------------|
| Sultan dina----- | 2229/14 |
| Halima mustefa ----- | 2822/14 |
| Natnael workineh ----- | 1879/14 |

Advisor Name: Mrs Hana M.

January 2017

Wolkite, Ethiopia

DECLARATION

This is to declare that this project which is done under the supervision of Mrs. Hana.M and having the title web based continuing and Distance Education Management System for wolkite university is the sole contribution of: Sultan dina, Halima mustefa and Natnael workineh.

No part of this project has been reproduced illegally (copy and paste) which can be considered as plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if a violation of this declaration is proven.

Date: _____

Group members:

Full name

Signature

Sultan dina

Halima mustefa

Natnael workineh

APPROVAL FORM

This is to confirm that the project entitled web based continuing and Distance Education Management System for wolkite university, College of Computing and Informatics, Department of Information Systems.

| | | |
|-------------------------------|--------------------|---------------|
| ----- Advisor name | ----- Signature | ----- Date |
| ----- Department head name | ----- Signature | ----- Date |
| ----- Examiner 1 name | ----- Signature | ----- Date |
| ----- Examiner 2 name | ----- Signature | ----- Date |
| ----- Examiner 3 name | ----- Signature | ----- Date |

Acknowledgment

We extend our deepest gratitude to all individuals and institutions who have contributed to the successful completion of this project.

First and foremost, we thank **Wolkite University** for providing us with the opportunity and resources to undertake this project and for their continuous support and encouragement throughout the process.

Abstract

Wolkite University is one of the higher institutions involved in continuing and distance education. This project is conducted in Wolkite University's continuing and distance education program.

The Web-Based Continuing and Distance Education Management System for Wolkite University is designed to address the limitations of the current semi-automated system by transitioning to a fully automated online platform. The project aims to streamline critical processes such as student registration, learning material distribution, grading, and communication.

Leveraging modern web technologies, the system provides students, instructors, and administrators with efficient, user-friendly, and centralized access to educational resources and administrative functionalities. By automating these processes, the Web-Based Continuing and Distance Education Management System reduces resource wastage, enhances accessibility, and fosters inclusive, scalable, and modern education delivery.

Table of content

| | |
|--|-------------------------------------|
| Acknowledgment | iii |
| Abstract | iv |
| List of Acronyms..... | vii |
| List of Figures | viii |
| List of Tables | Error! Bookmark not defined. |
| Chapter One Introduction..... | 1 |
| 1.1 Introduction | 1 |
| 1.2 Background of organization..... | 1 |
| 1.3 Background of the Project | Error! Bookmark not defined. |
| 1.4 Existing System Study | Error! Bookmark not defined. |
| 1.5 Statement of the Problem..... | 2 |
| 1.6 Proposed System..... | Error! Bookmark not defined. |
| 1.7 General Objective | 2 |
| 1.8 Specific Objectives | 2 |
| 1.9 Scopes and Limitations of the Project | 3 |
| 1.9.2 Limitations..... | Error! Bookmark not defined. |
| 1.10 Methodology of the Project | 8 |
| 1.11 System Analysis and Design Methodologies..... | Error! Bookmark not defined. |
| 1.12 Software Tools..... | Error! Bookmark not defined. |
| 1.13 Hardware tools used | 11 |
| 1.14 Feasibility Study | Error! Bookmark not defined. |
| 1.17 Technical feasibility..... | Error! Bookmark not defined. |
| 1.18 Operational feasibility..... | Error! Bookmark not defined. |
| 1.19 Significance of the Project..... | Error! Bookmark not defined. |
| Chapter 2 | Error! Bookmark not defined. |
| Alleviation of existing system..... | Error! Bookmark not defined. |
| 2.1 Introduction to existing system..... | Error! Bookmark not defined. |
| 2.2 Drawback of the existing system | 17 |
| The current semi-automated manual system for Wolkite University's Continuing and Distance Education Program has several key drawbacks: | 17 |
| Physical distribution of materials, printing, and manual record-keeping lead to increased costs and resource wastage..... | 17 |
| Delays in delivering course materials and assignments due to manual distribution. | 17 |
| Students must visit the university in person for registration, and administrative processes are time-consuming. | 17 |
| Students must visit the center to check grades, which is inconvenient and inefficient. | 17 |
| 2.2 Performance (Response Time)..... | Error! Bookmark not defined. |

| | |
|---|-------------------------------------|
| 2.3 Efficiency of the Existing System..... | Error! Bookmark not defined. |
| 2.4 Users of the Existing System | 15 |
| 2.5 Major Functions/Activities in the Existing System | 16 |
| Chapter 3 | 19 |
| Proposed system..... | 19 |
| 3.1 Introduction | 19 |
| 3.2 Functional Requirements | 19 |
| 3.3 Non-Functional Requirements | Error! Bookmark not defined. |
| CHAPTER FOUR..... | 30 |
| SYSTEM FEATURE..... | 30 |
| 4. Introduction | Error! Bookmark not defined. |
| 4.2 System use case | 30 |
| 4.3. Business rule..... | Error! Bookmark not defined. |
| 4.4. User interface prototype | Error! Bookmark not defined. |
| 4.5 Activity diagrams..... | 55 |
| 4.6. Sequence diagram model..... | 65 |
| | 72 |
| Figure 4.22 exam management..... | Error! Bookmark not defined. |
| 4.7 Logic model..... | Error! Bookmark not defined. |
| CHAPTER FIVE | Error! Bookmark not defined. |
| 5. SYSTEM DESIGN | 75 |
| 5.1. Introduction | 75 |
| 5.2. Architectural Design..... | 77 |
| 5.3. Class Design model | 81 |
| 5.4. Persistent model..... | 83 |
| 5.5. User Interface Design | 87 |
| CHAPTER SIX | Error! Bookmark not defined. |
| 6. CONCLUSION | Error! Bookmark not defined. |
| 6.1. Conclusion | Error! Bookmark not defined. |
| 5. REFERENCE..... | Error! Bookmark not defined. |

List of Acronyms

List of abbreviation

| | |
|---------------------------|--|
| Cd-Rom Drives----- | Compact Disc Read-Only Memory |
| Css----- | Cascading Style Sheets |
| Dbms ----- | Database Management System |
| Gb----- | Gigabyte |
| Ghz ----- | Gigahertz |
| Hd----- | Hard Disk |
| Html ----- | Hypertext Markup Language |
| Ict----- | Information And Communications Technology. |
| Mb ----- | Megabyte |
| Mysql ----- | My Structured Query Language |
| Ram----- | Random Access Memory |
| Pc ----- | Personal Computer |
| Php ----- | Hypertext Preprocessor |
| Ram: 8 Gb ----- | Random Access Memory |
| Uml ----- | Unified Modeling Language: |
| Wku----- | Wolkite University |
| Xampp ----- | Cross-Platform, Apache, Mysql, Php, And Perl: |

List of Figures

| | |
|--|----|
| FIGURE <u>4.1</u> SYSTEM USE CASE DIAGRAM | 32 |
| FIGURE <u>4.2</u> USER INTERFACE PROTOTYPE | 52 |
| FIGURE <u>4.3</u> LOGIN AND CREATE ACCOUNT ACTIVITY DIAGRAM | 55 |
| FIGURE <u>4.4</u> ACTIVATE, DEACTIVATE AND UPDATE ACTIVITY DIAGRAM STUDENT REGISTRATION AND ASSIGN INSTRUCTOR ACTIVITY DIAGRAM..... | 56 |
| FIGURE <u>4.5</u> STUDENT REGISTRATION AND ASSIGN INSTRUCTOR ACTIVITY DIAGRAM..... | 57 |
| FIGURE <u>4.6</u> DOWNLOAD ASSIGNMENT QUESTION AND SOLUTIONS ACTIVITY DIAGRAM..... | 58 |
| FIGURE <u>4.7</u> DOWNLOAD ASSIGNMENT QUESTION AND SOLUTIONS ACTIVITY DIAGRAM..... | 60 |
| FIGURE <u>4.8</u> UPLOAD AND DOWNLOAD MODULE AND LOGOUT ACTIVITY DIAGRAM..... | 60 |
| FIGURE <u>4.9</u> GENERATE REPORT AND VIEW REPORT ACTIVITY DIAGRAM..... | 61 |
| FIGURE <u>4.10</u> POST ANNOUNCEMENT AND GIVE COMMENT ACTIVITY DIAGRAM | 62 |
| FIGURE <u>4.10</u> SUBMIT AND VIEW RESULT ACTIVITY DIAGRAM | 63 |
| FIGURE <u>4.11</u> ONLINE PAYMENTS AND TRANSACTIONS | 64 |
| FIGURE <u>4.12</u> ACTIVITY DIAGRAM FOR CONDUCTING AND TAKING EXAMS ERROR! BOOKMARK NOT DEFINED. | |
| FIGURE <u>4.13</u> SCHEDULING AND JOINING..... | 65 |
| FIGURE <u>4.14</u> SEQUENCE DIAGRAM FOR STAFF MEMBER LOGIN | 66 |
| FIGURE <u>4.15</u> SEQUENCE DIAGRAM FOR UPLOAD MODULE | 67 |
| FIGURE <u>4.16</u> SEQUENCE DIAGRAM FOR DOWNLOAD MODULE | 67 |
| FIGURE <u>4.17</u> SEQUENCE DIAGRAM FOR UPLOAD ASSIGNMENT..... | 67 |
| FIGURE <u>4.18</u> SEQUENCE DIAGRAM FOR ASSIGN INSTRUCTOR..... | 68 |
| FIGURE <u>4.19</u> SEQUENCE DIAGRAM FOR GENERATE REPORT | 69 |
| FIGURE <u>4.20</u> SEQUENCE DIAGRAM FOR POST ANNOUNCEMENT | 71 |
| FIGURE <u>4.21</u> SEQUENCE DIAGRAM FOR SUBMIT RESULT | 71 |
| FIGURE <u>4.22</u> EXAM MANAGEMENT FIGURE AND 4.22 PAYMENT MANAGEMENT | 72 |
| FIGURE <u>4.23</u> TUTORIAL AND VIDEO LEARNING..... | 72 |
| FIGURE <u>4.1</u> COMPONENT MODEL DIAGRAM | 79 |
| FIGURE <u>5.2</u> DEPLOYMENT DIAGRAM..... | 81 |
| FIGURE <u>5.3</u> CLASS DIAGRAM..... | 82 |
| FIGURE <u>5.4</u> PERSISTENT MODEL | 83 |
| FIGURE <u>5.4</u> HOME PAGE USER INTERFACE | 88 |
| FIGURE <u>5.6</u> LOGIN USER INTERFACE | 89 |

List of Tables

| | |
|--|----|
| TABLE 1-1: ESTIMATED BUDGET REQUIRED FOR THE PROJECT..... | 12 |
| TABLE 1-2: TIME SCHEDULE OF THE PROJECT | 12 |
| TABLE 1-3: TEAM COMPOSITION..... | 13 |
| | |
| TABLE 4 1: SELECTION OF ACTORS AND USE CASE | 30 |
| TABLE 4 3:MANAGE ACCOUNT USE CASE DESCRIPTION | 34 |
| TABLE 4 4:DESCRIPTION OF UPLOAD MODULE USE CASE | 34 |
| TABLE 4 5: DOWNLOAD MODULE USE CASE DESCRIPTION..... | 35 |
| TABLE 4 6:DOWNLOAD ASSIGNMENT USE CASE DESCRIPTION | 36 |
| TABLE 4 7: UPLOAD ASSIGNMENT USE CASE DESCRIPTION..... | 36 |
| TABLE 4 8: REGISTER STUDENT USE CASE DESCRIPTION..... | 37 |
| TABLE 4 9: REGISTER COURSE USE CASE DESCRIPTION..... | 38 |
| TABLE 4 10:VIEW COMMENT USE CASE DESCRIPTION | 39 |
| TABLE 4 11: CHANGE PASSWORD USE CASE DESCRIPTION | 39 |
| | |
| TABLE 4 12:ASSIGN INSTRUCTOR USE CASE DESCRIPTION | 41 |
| TABLE 4 13:POST ANNOUNCEMENT USE CASE DESCRIPTION..... | 41 |
| TABLE 4 14:VIEW REPORT USE CASE DESCRIPTION | 43 |
| TABLE 4 15:GENERATE REPORT USE CASE DESCRIPTION..... | 43 |
| TABLE 4 16: VIEW GRADE USE CASE DESCRIPTION | 44 |
| TABLE 4 17:GIVE COMMENT USE CASE DESCRIPTIONS | 45 |
| | |
| TABLE 4 18: ADD CURRICULUM USE CASE DESCRIPTION..... | 46 |
| TABLE 4 19:VIEW CURRICULUM USE CASE DESCRIPTION | 48 |
| TABLE 4 20:VIEW RESULT USE CASE DESCRIPTION VIEW RESULT USE CASE DESCRIPTION | 49 |
| TABLE 4 21:UPLOAD VIDEO | 50 |
| TABLE 4 22:VALIDATE TRANSACTIONS..... | 50 |
| TABLE 4 23: JOIN TUTORIAL | 52 |

CHAPTER ONE

INTRODUCTION

In today's rapidly evolving world, continuing and distance education has become a cornerstone for providing flexible and accessible learning opportunities to individuals seeking personal and professional growth. However, traditional systems for managing distance education often rely on manual processes, which are time-consuming, inefficient, and prone to errors. [1] These limitations hinder the delivery of quality education and the ability to meet the growing demand for lifelong learning.

A Web-Based Continuing and Distance Education Management System addresses these challenges by leveraging technology to automate and streamline educational processes. This system ensures efficient management of student registration, learning materials, communication, and administrative tasks, creating a centralized platform that improves accessibility, reduces resource wastage, and enhances the overall learning experience. By transitioning to a web-based solution, institutions can deliver education that is more inclusive, scalable, and aligned with modern needs.

1.1 Background of organization

Wolkite University, established in 2011, is a public higher education institution located in Wolkite Town, in the Gurage Zone of the Southern Nations, Nationalities, and Peoples' Region (SNNPR), Ethiopia. [2] The mission is to produce skilled, innovative, and responsible graduates, deliver research-based solutions to societal and developmental challenges, and provide training and consultancy services to the community and government. The university is committed to enhancing education quality, addressing national issues through research and technology, and fostering sustainable development, guided by its core values of excellence, inclusiveness, integrity, continuous learning, accountability, and academic freedom. The institution offers a wide range of undergraduate and postgraduate programs across various fields, including science, technology, social sciences, business, and education. It has become a hub for knowledge creation and skill development, addressing the educational and professional needs of the region.

1.2 Statement of the Problem

The continuing and distance education system at Wolkite University currently faces significant challenges, despite being semi-automated. The system relies heavily on manual processes, which result in inefficiencies, increased costs, and reduced accessibility for students.

As a result of our analysis, the existing system has the following key problems:

High Cost of Duplicating Learning Materials: The production of physical copies of modules and materials requires substantial financial resources.

Expensive Distribution of Learning Materials: The distribution of these materials to various centers incurs additional logistical and operational costs.

Time-Consuming and Costly Processes for Registration and Tutorials: Activities such as student registration, tutorial scheduling, and grade reporting are labor-intensive and costly, leading to delays.

Inconvenient Access to Grade Reports: Students are required to travel to designated centers to view their results, which is both time-consuming and inconvenient.

1.3. Objectives of the Project

The primary objective of the is to modernize and streamline the operations of Wolkite University's continuing and distance education programs by transforming them into a web-based platform.

1.3.1 General Objective

The general objective of this project is to develop a web-based Continuing and Distance Education Management System for Wolkite University to enhance the efficiency and accessibility of managing Continuing and Distance education programs.

1.3 Specific Objectives

To achieve the general objective of developing the Web-Based Continuing and Distance Education Management System for University are:

- Analyze the current system to identify inefficiencies and problems in the manual processes.
- Define functional and non-functional requirements based on user needs and system performance goals.
- Design an intuitive and accessible interface that enhances the user experience for students, instructors, and administrators.
- Develop and implement the system using appropriate web technologies to ensure smooth integration and performance. Test the system thoroughly to ensure it is secure, efficient, and fully functional.
- Train users (students, staff, and instructors) to ensure they can effectively use the new system. Evaluate the system's performance through user feedback and system monitoring to assess its impact. Optimize the system based on feedback and performance data for improved functionality and scalability.

1.4 Feasibility Analysis

This feasibility analysis assesses the proposed Web-Based Continuing and Distance Education Management System for Wolkite University in terms of technical, operational, economic, and other relevant feasibility aspects. It examines whether the project can be successfully developed and implemented, considering resources, user acceptance, and potential benefits.

1.4.1 Technical Feasibility

Availability of Required Technology: The technologies required for the development of the system, including HTML, CSS, JavaScript, MySQL, and modern frameworks (React, Node.js), are widely available and suitable for the project's needs.

Resources and Infrastructure: Wolkite University's existing infrastructure, such as internet connectivity and hardware, is sufficient for the system's requirements. The project can be built using readily available software development tools.

Manpower Availability: Skilled developers, testers, and system administrators can be sourced either within the university or externally, ensuring the availability of necessary human resources.

Direction for Improvement: Ensure the university's infrastructure is assessed and upgraded if necessary, especially in remote locations.

1.4.2 Operational Feasibility

User Acceptance: There is a clear need for the automation of the current semi-automated system, and users (students, instructors, and administrative staff) will benefit from a more efficient, reliable, and user-friendly system.

Management Support: The university management supports the project as it aligns with their goal of improving education accessibility and operational efficiency.

Efficiency Gains: The new system will automate manual processes like registration, grading, and material distribution, significantly reducing the time spent on administrative tasks.

Direction for Improvement: Conduct training sessions and provide continuous support to ensure a smooth transition to the new system.

1.4.3 Economic Feasibility

Cost Savings: The system will reduce costs associated with paper-based materials, manual data entry, and administrative overheads. It will also reduce delays and inefficiencies.

Long-Term Benefits: The system's efficiency will lead to better resource management, improved service delivery, and long-term savings for the university.

Direction for Improvement: Conduct a detailed cost-benefit analysis to justify the investment in the system by highlighting long-term savings and efficiency gains.

1.4.4 Other Feasibility

Political Feasibility: Government Support The project aligns with national educational goals of improving access to education and supporting distance learning. Political support may be available in the form of grants or policies encouraging digital transformation in higher education.

Compliance with Regulations: The project will adhere to local regulations regarding data protection, accessibility, and security standards, which is essential for smooth implementation.

Direction for Improvement: Promote digital literacy through training programs to increase user acceptance. Ensure the system complies with government regulations to avoid any legal or political challenges.

1.5 Scope and Limitation of the Project

The scope and limitation of the **Web-Based Continuing and Distance Education Management System** for Wolkite University define the boundaries of what the system aims to accomplish and identify areas beyond its intended functionality.

1.5.1 Scope of the Project

The project focuses on automating and streamlining Wolkite University's continuing and distance education operations. Specifically, it includes:

Student Management: Online registration for courses and tutorials. Digital access to personal profiles, course enrollment, and academic records.

Learning Materials: Uploading and accessing course modules, tutorials, and lecture materials online.

Grade Management: Viewing grades and performance reports in real time. Submission and review of assignments and exam results.

Communication Tools: Sending timely updates and notifications regarding schedules, deadlines, and announcements.

Administrative Functions: Managing user accounts (students, instructors, and staff). Generating reports and overseeing system operations through an administrative dashboard.

Scalability and Security: The system is designed for future growth, with features that allow additional functionalities to be added. Data protection with user authentication and encryption. Support for Multiple Courses: Management of various programs offered..

1.5.2 Limitation of the Project

While the system aims to address key challenges in distance and continuing education, the following limitations are acknowledged:

Internet Dependency: The system requires a stable internet connection, limiting accessibility in areas with poor or no connectivity.

User Training Requirements: Some users may face difficulty using the system due to a lack of digital literacy, requiring extensive training sessions.

Limited Offline Functionality: The system does not support offline access to materials or services, which could hinder usability in remote regions.

1.6 Significance of the Project

The Web-Based Continuing and Distance Education Management System holds both societal and technological importance:

1.6.1 Societal Importance

Improved Education Access: The system provides greater accessibility to education for students in remote areas, overcoming geographic and logistical barriers.

Time-Saving and Convenience: By digitizing processes like registration and grade access, the system reduces the time and effort required for administrative tasks, benefiting both students and staff.

Resource Conservation: It reduces reliance on paper-based processes, contributing to environmental sustainability.

Equity in Learning: Students gain equal access to resources such as course materials and announcements, fostering inclusiveness in education.

1.6.2 Technological Importance

Modernization of Processes: The system leverages technology to transform outdated manual systems into efficient, automated workflows, enhancing the overall functionality of the university.

Scalability: Designed with scalability in mind, the system can accommodate future growth in student numbers and features, ensuring longevity.

Data Management and Security: Advanced database systems ensure secure, reliable storage and retrieval of data, safeguarding sensitive student and administrative information.

Integration of Innovations: The system paves the way for incorporating emerging technologies like analytics and e-learning tools to improve the education experience further.

1.7 Beneficiary of the Project

The beneficiaries of the Web-Based Continuing and Distance Education Management System are:

Students: Gain convenient access to course materials, grades, and timely updates, saving time and effort.

Instructors: Simplify course management, grading, and student progress tracking.

Administrative Staff: Automate tasks like registration and reporting, reducing workload and errors.

University Management: Enhance oversight and operational efficiency while improving the institution's reputation.

Society: Provide accessible education, especially in remote areas, supporting community development.

1.8 Methodology of the Project

This section explains the approach taken to develop the Web-Based Continuing and Distance Education Management System detailing how requirements were gathered, tools for analysis and design, the development model, testing methodology, and the tools and technologies used.

1.8.1 Data Collection Tools/Techniques

To gather comprehensive requirements, the following techniques were employed:

There are different data gathering techniques which help us to understand the present systems general activities and its procedures. We used three methodologies to gather the needed information. The methods we use for data collection are:

Interviewing: As a method for the collection of data about the activities of University of continuing and distance education we use interviewing methods to understand people who belong to the current system. Also we raised questions that helped us to develop the new system.

Direct Observation: We observe the current system and identify the problems regarding the working process .So it helps us as an easy way to understand the system and to develop the project. By observing their system structure we design a project to solve the problems.

Document Analysis: we analyze some documents and forms found in the office to get information about the current system.

1.8.2 System Analysis and Design

System Analysis: is the process of breakdown an entire system into modules, analyzing each module separately, and determining the relationship between them. In the system development process we use system modeling i.e. creating models of the system. System modeling is the process of creating a model of system by analyzing and organizing the system elements. This is the phase where we deeply understand the existing system problems and find alternative solutions. Models are:

Use case diagram: Use cases define interactions between external actors and the system to attain particular goals. There are three basic elements that make up a use case:

Actors: Actors are the type of users that interact with the system.

System: Use cases capture functional requirements that specify the intended behavior of the system.

Goals: Use cases are typically initiated by a user to fulfill goals describing the activities and variants involved in attaining the goal

Class diagram: is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Activity diagram: An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.

Sequence diagram: Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

Model high-level interaction between active objects in a system. Model the interaction between object instances within a collaboration that realizes a use case

Model the interaction between objects within a collaboration that realizes an operation

Either model generic interactions (showing all possible paths through the interaction) or specific instances of interaction (showing just one path through the interaction)

System Design: - to design the system the project team has chosen Object Oriented Modeling techniques and Unified modeling language tools. Design models are:

Deployment diagram: are used to visualize the topology of the physical components of a system, where the software components are deployed.

Component diagram: are describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

1.8.3 System Development Model

The **Development Model** was selected for its flexibility and user-centric approach:

Reason for Selection: Incremental development allows delivering working modules for feedback after each iteration. Accommodates changes based on feedback during the development cycle. Mitigates risks by addressing issues early in smaller iterations.

Development Stages: Requirement Gathering and Analysis Collected requirements in collaboration with stakeholders.

Design: Created modular designs using UML diagrams.

Implementation: Developed system components iteratively.

Testing: Validated each module before integration into the system.

Deployment: Delivered the final product incrementally for end-user evaluation.

1.8.4 System Testing Methodology

Testing ensures the system meets functional, performance, and user requirements:

Unit Testing: Verified individual modules like login, registration, and grade management for functionality.

Integration Testing: Ensured smooth interaction between frontend, backend, and database components.

System Testing: Tested the complete system to validate all features and performance under various conditions.

Acceptance Testing: Conducted with end-users to ensure the system meets their expectations and requirements.

1.8.5 Development Tools and Technologies

The following tools and technologies were used for efficient system development:

Frontend Technologies: HTML, CSS, and JavaScript: To build responsive and visually appealing user interfaces.

React.js: Used for creating dynamic and reusable components for the frontend.

Backend Technologies: Node.js: For handling server-side operations, ensuring high performance.

MySQL: Used for creating and managing the relational database to store student, course, and administrative data securely.

Documentation and Modeling Tools: Unified Modeling Language (UML) Tools like Star UML were used for creating class, sequence, and activity diagrams.

Documentation Tools: Microsoft Word

Deployment Environment: We use Wamp server to configure a MySQL database and to use node js applications for easy configuration.

1.6 Hardware tools used

PC (laptop): to perform our tasks like writing documentation and codes.

Flash disk: to store file temporarily and used to file transfer computer to computer

CD (compact disk 700M): to make back up files or store files.

Mobile camera: to capture images that we need for the project.

1.9 Budget and Time Schedule of the Project

The Web-Based Continuing and Distance Education Management System project proposal includes an estimated budget and timeline to guide its development and implementation.

1.9.1 Budget Estimation

Table 0-1: estimated budget required for the project:

| Category | Description | Estimated Cost (ETB) |
|------------------------|--|----------------------|
| Personnel Costs | Developers, testers, and system administrators | 10,000 |
| Stationery | Paper, pens, and other materials | 5,00 |
| Communication | Mobile cards for communication with stakeholders | 300 |
| Hardware | Laptop and desktop purchases for development | 50,000 |
| Hosting and Deployment | Cloud server setup or local hosting infrastructure | 7000 |
| Training | Workshops and materials for user training | 3000 |
| Miscellaneous | Other unforeseen expenses | 1000 |
| Transport | Travel for meetings, data collection, and training | 1,000 |
| Total | | 72800 ETB |

1.9.2 Time Schedule of the Project

Table 0-2: Time Schedule of the Project

| Task | 25/02/17 - 09/03/17 | 10/03/17- 26/03/17 | 27/03/17- 26/04/17 | 16/05/17- 16/08/17 | 17/08/17- 27/08/17 | 28/08/17 |
|-----------------------|---------------------|--------------------|--------------------|--------------------|--------------------|----------|
| Requirement Gathering | [REDACTED] | | | | | |
| System Analysis | | [REDACTED] | | | | |
| System Design | | | [REDACTED] | | | |
| Implementation | | | | [REDACTED] | | |
| Testing | | | | | [REDACTED] | |

| | | | | | | | |
|-------------------------|--|--|--|--|--|--|--|
| Deployment and Training | | | | | | | |
|-------------------------|--|--|--|--|--|--|--|

1.10. Team Composition

Table 0-3: Team Composition

| Team Member | Role | Responsibilities and Tasks |
|------------------|----------------|--|
| Sultan Dina | Project Lead | <ul style="list-style-type: none"> - Oversee the overall project progress - Coordinate tasks and activities among team members - Finalize the project deliverables - Prepare and write the project documentation |
| Halima Mustafa | Developer | <ul style="list-style-type: none"> - Develop and implement system functionalities - Code both front-end and back-end features |
| Natnael workinah | UI/UX Designer | <ul style="list-style-type: none"> - Design the user interface - Create wireframes and prototypes - Conduct testing to identify and resolve bugs |

1.11. Document Organization

Chapter 1: Introduction – Project overview, objectives, and scope. Chapter 2: Literature Review – Review of related research and technologies.

Chapter 3: System Design – System architecture and design details. Chapter 4: Implementation – Development process, tools, and technologies used.

Chapter 5: Testing and Evaluation – Testing procedures and results. Chapter 6: Results and Discussion – Project outcomes and analysis. Chapter 7: Conclusion and Future Work – Summary and future improvements.

References – Cited sources. Appendices – Supplementary materials.

CHAPTER TWO

DESCRIPTION OF THE EXISTING SYSTEM

2.1. Introduction of Existing System

The existing system at Wolkite University for managing the Continuing and Distance Education Program is a semi-automated manual system that involves a combination of paper-based processes and basic digital tools. While some aspects of the system, such as student registration and grade management, have been digitized, much of the core functionality remains manual.

The system primarily relies on traditional methods for distributing learning materials, registering students, and conducting tutorials. Instructors prepare learning materials and distribute them physically at various locations, including Wolkite University and its branches. [2] Students are required to attend face-to-face tutorial sessions and visit the university in person to register for courses, view their grades, and receive academic updates.

2.2 Users of the Existing System

The existing system at Wolkite University is used by:

Students: Students are the primary beneficiaries of the system, utilizing it for tasks such as registration, course enrollment, accessing grades, and collecting learning materials.

Physically visit the university to complete registration, course enrollment, and access grades.

Instructors: Instructors are responsible for delivering course content, grading assignments, and managing student progress.

Distribute course materials and lecture notes in physical format.

Record and manage grades manually in grade books or spreadsheets.

Submit finalized grades to the administrative office for consolidation.

Administrative Staff: Administrative staff manage all backend operations of the system, including processing student registration, managing course enrollments, consolidating grades, and maintaining student records.

Manage general administrative activities, such as filing, reporting, and correspondence

University Management: University management oversees the system's overall operation, ensuring compliance with policies and regulations. They use the system for decision-making and reviewing reports on the academic and administrative processes.

Supervise the overall administration of the continuing and distance education programs.

Ensure compliance with university policies and regulations.

Review reports and provide approvals for administrative processes and decisions.

2.3 Major Functions/Activities in the Existing System

The existing semi-automated system at Wolkite University performs several key functions and activities, though many are still manual:

Student Registration: Students register for courses in person, which involves manual data entry and record-keeping by administrative staff.

Distribution of Learning Materials: Learning materials (e.g., modules, textbooks) are printed and physically distributed to students, leading to delays and high costs.

Tutorial Sessions: Tutorial sessions are scheduled and conducted in person at various stations. These sessions are planned manually, and instructors are responsible for organizing them.

Grading and Assessments: Instructors grade assignments and exams manually, and students must visit the university to check their results.

Communication of Updates: Important announcements (e.g., exam schedules, registration deadlines) are communicated, often causing delays and confusion.

Student Record Management: Student records, such as course registrations and grades, are managed by administrative staff.

Reporting and Data Collection: Administrative staff compile and report data on student progress, course completion, and other metrics,

Assignment Submission and Feedback: Students submit assignments in person, and instructors provide feedback manually,

2.4 Drawback of the existing system

The current semi-automated manual system for Wolkite University's continuing and Distance Education Program has several key drawbacks:

High Operational Costs: Physical distribution of materials, printing, and manual record-keeping lead to increased costs and resource wastage.

Inefficient Material Distribution: Delays in delivering course materials and assignments to distribution.

Students must visit the university in person for registration, and administrative processes are time-consuming.

Limited Access to Grades: Students must visit the center to check grades, which is inconvenient and inefficient.

Lack of Real-Time Communication: Delays in disseminating important information such as schedule changes or updates.

Overburdened Administrative Staff: Manual data entry and material preparation lead to increased workload and errors.

Limited Scalability: The system struggles to handle increasing numbers of students and courses.

Inconsistent Data Management: Manual entry of student records increases the risk of errors and inconsistencies.

2.5 Business Rules of the Existing System

The existing semi-automated system for managing Wolkite University's continuing and distance education programs operates under several principles and rules. These rules govern how tasks are performed and by whom, influencing the functional requirements of the system. [3]

Registration Rules: Only administrative staff are authorized to process student registration forms and approve new student enrollments. Students must submit all required documents (e.g., ID, prior academic records) during registration.

Course Enrollment Rules: Administrative staff must verify and record each student's course enrollment.. **Grade Submission Rules:** Instructors are responsible for calculating and submitting grades for their assigned courses. Final grade submissions must be approved and recorded by administrative staff in the official grade register.

4. Learning Material Distribution Rules: Only authorized administrative staff can distribute course materials to students.

5. Communication Rules: Announcements and updates must be approved by university management before being shared with students.

6. Reporting and Data Management Rules: Only administrative staff and university management are authorized to access and compile student records and administrative reports. Sensitive information, such as grades and personal student data, must be handled confidentially and stored securely in physical files.

Implications for Functional Requirements

These business rules imply the need for specific functional requirements in an automated system, such as:

Role-based access control to enforce user permissions. Automated workflows for registration, course enrollment, and grade management. Secure communication channels for announcements and updates. Centralized and secure data management with audit trails for sensitive information.

CHAPTER 3

PROPOSED SYSTEM

3.1 Introduction

During the analysis of the existing system through observations and interviews with users, the project team identified significant challenges associated with Wolkite University's semi-automated Continuing and Distance Education Management System. These challenges stem from the reliance on manual processes, which result in inefficiencies, delays, and resource wastage in critical areas such as student registration, learning material distribution, tutorial scheduling, grading, and communication.

To address these issues, the project team has proposed a Web-Based Continuing and Distance Education Management System. The new system will centralize and automate these activities, allowing students, instructors, and administrative staff to access services through a single, user-friendly online platform.

3.2 Functional Requirements

The **Functional Requirements** define the specific functions that the **Web-Based Continuing and Distance Education Management System** must perform to meet the needs of users and ensure smooth operations. These requirements describe the core features and operations that must be implemented for the system to function effectively

3.3 Functional Requirements

User Registration & Authentication: Students, instructors, and administrators can create accounts and log in securely Role-based access control is implemented.

Course Enrollment: Students can enroll in courses, and instructors can approve enrollments. Manages course capacity and waiting lists.

Grade Management: Instructors can enter and update grades. Students can view grades and generate grade reports.

Learning Material Distribution: Instructors upload course materials, which students can access and download.

Communication & Notifications: Announcements and real-time messaging for students and instructors. Calendar for important academic dates.

Reporting & Analytics: Generate performance, enrollment, and custom reports for students and courses.

System Administration & Maintenance: Admins manage user roles, data backup, and security.

Scalability & Flexibility: New courses can be added, and the system can handle increased users and data.

The proposed Web-Based Continuing and Distance Education Management System for Wolkite University will include the following functional requirements:

Student Functions:

Students should have individual accounts.

Students should be able to register for courses online.

Students should access learning materials, grades, and academic progress.

Students should receive real-time notifications (e.g., schedules, deadlines, updates).

Students should request tutorial session enrollment.

Students should update their personal account information.

Students should access and download course completion certificates.

Students should be able to print/view their academic records on-screen.

Instructor Functions:

Instructors should have individual accounts.

Instructors should manage and upload learning materials.

Instructors should input grades and provide feedback for assignments and exams.

Instructors should schedule and manage tutorial sessions.

Instructors should view and update their account information.

Instructors should track student progress within their assigned courses.

Administrative Functions:

The admin should be able to register new users (students, instructors, and staff).

The admin should manage and update user accounts and roles.

The admin should oversee course creation and enrollment.

The admin should generate and view detailed system usage and academic reports.

The admin should manage notifications for students and staff.

The admin should ensure secure data backup and recovery.

Officer Functions:

Officers should approve or reject student tutorial session requests.

Officers should manage clearance and certification requests.

Officers should register and track borrowed university materials (e.g., books, equipment).

Officers should access reports and manage student records.

System Features:

The system should allow secure user authentication.

The system should enable real-time updates for notifications, grades, and materials.

The system should allow automated grading for specific assignments.

The system should generate printable reports for students, instructors, and administrators.

The system should provide a user-friendly dashboard for personalized access to resources.

The system should enable data analytics for administrative purposes.

3.3 Non-functional Requirements

Non-functional requirements specify the quality attributes, constraints, and standards that the system must meet. These requirements focus on **how** the system will perform rather than **what** the system will do. Below are the relevant non-functional requirements

Performance: Response Time: The system must respond to user actions. Concurrency and Availability

Scalability: Data Storage: The system should be able to manage large amounts of student, course, and grade data with minimal impact on performance.

Security: Authentication, Data Encryption, Authorization.

Usability: User Interface, Accessibility Multi, device Support.

Maintainability: Code Quality, Error Handling, System Monitoring.

Compliance: Data Protection Laws University Policies

3.3.1 User Interface and Human Factors

The proposed Web-Based Continuing and Distance Education Management System is designed with the user experience (UX) and user interface (UI) principles tailored to the primary stakeholders—university students, instructors, and administrative officers. Recognizing that many users may have limited experience with web applications, the system will adhere to the following design principles: [4]

Easy to Use: The system will provide a straightforward navigation structure and user-friendly design, ensuring that all users can easily perform tasks without extensive technical knowledge.

Responsive Design: The application will adapt seamlessly to various devices, including desktops, tablets, and smartphones, providing an optimal experience across all platforms.

Easily Adaptable: The system will include intuitive guidance (tooltips, help icons, and tutorials) to make it easy for users to learn and adapt to its features quickly.

Attractive and Interactive: The interface will feature a visually appealing design with engaging elements like buttons, animations, and color schemes that enhance user satisfaction and encourage engagement.

Simple Design: Intuitive, role-based dashboards for students, instructors, and administrators. **Responsive Design:** Compatible with desktops, laptops, tablets, and smartphones.

User Expertise Levels

Students: Basic computer skills; guided interface for ease of use.

Instructors: Moderate expertise; simple tools for uploading materials and managing grades.

Administrators: Familiar with office tools; advanced options for record and report management.

3.4 Hardware Consideration

The proposed Web-Based Continuing and Distance Education Management System will be designed to function seamlessly across various hardware platforms. The system will ensure compatibility and responsiveness

Key hardware considerations include:

Cross-Device Compatibility

The system will work efficiently on desktops, laptops.

Minimal Resource Requirements: The system will be lightweight, requiring minimal processing power and storage, ensuring it functions well on older or less powerful devices.

Browser Compatibility: The system will be fully functional across popular web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, without malfunctions.

University Infrastructure: Integrates with existing servers and networking equipment.

These considerations will ensure that users can access the system reliably, regardless of the device or platform they are using

3.4 Security Issues

The proposed Web-Based Continuing and Distance Education Management System will implement robust security mechanisms to ensure the safety and integrity of data and prevent unauthorized access. Key considerations include:

User Authentication: Users will be required to log in using unique credentials (username and password) to access the system.

User Authorization: Role-based access control (RBAC) will be employed to grant specific privileges based on user roles (e.g., student, instructor, admin).

Data Protection: Sensitive data, such as user information and grades, will be encrypted during storage and transmission.

System Monitoring: Security logs will track login attempts, data changes, and unauthorized access attempts to ensure transparency and detect anomalies.

These measures will ensure the system's resilience against attacks and unauthorized usage.

3.4.1 Performance Consideration

The Web-Based Continuing and Distance Education Management System is designed to deliver reliable performance under varying loads while ensuring responsiveness for all users.

System Responsiveness: The system should respond to user actions (e.g., logging in, accessing materials, and submitting forms) within **2-3** seconds under normal conditions.

Critical operations, such as grade entry or report generation, should complete within **5** seconds during peak load.

Concurrent Users: The system must support concurrent users (students, instructors, and administrators) without performance degradation.

Typical Load: Average daily usage of active users, including students accessing materials and instructors entering grades.

. 3.4.2 Error Handling and Validation

The proposed Web-Based Continuing and Distance Education Management System will include robust error handling and validation mechanisms to ensure smooth and reliable operation. Key considerations are as follows:

User Error Handling: Users will be prompted to review their input before any creation, processing, or update operation.

If invalid data is entered, the system will display a clear error message and request reentry in the correct format.

System Error Management: Run-time errors will be managed using exception handling techniques to prevent system crashes.

Errors occurring during database interactions or system processes will trigger specific error messages to guide users or administrators toward corrective actions.

Input Validation: The system will validate all user input fields to ensure data consistency and accuracy.

The system ensures robust error handling, validation, and reliability under adverse conditions.

3.4.3 Feedback and Notifications

Whenever an issue arises, the system will notify users with descriptive messages, enabling them to address problems without confusion. By implementing these mechanisms, the system will maintain stability and usability while effectively managing and communicating errors.

3.5 Quality Issues

The proposed Web-Based Continuing and Distance Education Management System will prioritize precision, reliability, and robustness to ensure it supports the university's resource control effectively. Key quality considerations include:

Reliability: The system will retrieve and display only the data requested by users accurately. Users can trust that all information provided by the system is consistent and up-to-date.

Robustness: The system will handle errors gracefully and guide users to take appropriate corrective actions. It will maintain its stability even under unexpected conditions or user errors.

Security: Access to the system will be strictly controlled through login mechanisms, allowing only authorized users to perform operations based on their roles. All users, including students, staff, and administrators, will be granted specific access rights to maintain data integrity and security.

User Authentication: Users must register and login to access key functionalities such as clearance forms.

Role-based access will ensure users can only perform tasks permitted for their role.

3.6 Backup and Recovery

The Web-Based Continuing and Distance Education Management System will include a comprehensive backup and recovery mechanism to ensure data integrity and availability in case of system failures or unforeseen events. Key considerations for backup and recovery include:

Data Backup: The system will automatically back up critical data (e.g., student records, course materials, clearance requests) on a regular basis to prevent data loss.

Backup files will be stored securely in multiple locations (e.g., cloud storage, on-premise servers) to ensure redundancy.

Recovery Process: In the event of data loss or system malfunction, the system will allow for quick data recovery from the most recent backup.

The recovery process will be automated to minimize downtime and ensure the system is restored to its most current state.

Backup Scheduling: Backups will be scheduled at regular intervals depending on the volume of data and the system's usage patterns.

Critical operations like student registration and exam results will be backed up more frequently to ensure real-time protection.

Data Integrity: The backup and recovery mechanism will include checks to verify that data is not corrupted during the backup process and that it can be restored accurately.

These measures will ensure that the system can recover quickly from any data loss or failure, maintaining continuity of operations and minimizing service disruptions.

3.7 Physical Environment

For the Web-Based Continuing and Distance Education Management System to function optimally and securely, the physical environment of both the server and client devices must be carefully considered. [5] Key considerations include:

Server Location: The server hosting the system must be placed in a secure, climate-controlled data center to prevent physical damage and unauthorized access.

Access to the server room should be restricted to authorized personnel only, with proper security measures such as surveillance, access logs, and biometric authentication.

Backup power systems (e.g., UPS or generators) should be in place to ensure continuous operation during power outages.

Client Location: Client devices (e.g., computers, tablets) should be within the local area or network that is required to connect to the server.

The client devices should be configured with the necessary security measures, such as firewalls and antivirus software, to protect against potential threats.

Network Connectivity: Reliable internet connections should be ensured for both the server and client devices to guarantee consistent access to the system.

The server's network infrastructure should support high availability, with redundant connections to ensure seamless communication between the server and client devices.

External Factors:

Weather Conditions: The cloud infrastructure is protected from weather-related issues, while on-premises servers are housed in climate-controlled data centers.

Power Supply: Cloud providers ensure power redundancy; on-premises servers will have UPS and backup generators to prevent downtime.

3.8 Resource Issues

Storage Resources: Efficient management of database size and incremental backups to avoid excessive storage usage.

Bandwidth Requirements: Stable, high-speed internet connections with potential network load balancing during peak periods.

Power Consumption: Cloud hosting dynamically scales to optimize resource usage, and on-premises servers must be energy-efficient.

3.9 Documentation

The Web-Based Continuing and Distance Education Management System will include comprehensive documentation to ensure smooth operation, maintainability, and user support. Key aspects of the documentation include:

User Documentation: Clear and detailed user manuals will be provided, explaining how to navigate the system, perform tasks (e.g., registration, clearance request), and troubleshoot common issues.

Help contents and tooltips will be integrated within the system interface to offer immediate guidance to users while interacting with the platform.

Administrator Documentation: Specific documentation for system administrators will outline how to manage user accounts, monitor system performance, and configure system settings.

Technical Documentation for Maintainers: Details on system architecture, installation, backup and recovery procedures, and security management for administrators and technical staff.

Development Process Documentation: Includes requirements specification, system design, testing processes, and version control for future updates and maintenance.

CHAPTER FOUR

SYSTEM ANALYSIS

4.1. System Model

This section of the Chapter describes the modeling of the proposed system by using object-oriented methodology and tools. All the activities performed by the actors and all the processes involved are analyzed by using different modeling diagrams. [6]

4.2 System use case

A use case diagram is a summary of who uses the system and what they can do with it. It describes the relationship among the requirements, users and the major components of the system. Use case diagram shows the relationships between users (actors) and use cases within a system or applicant. They provide an overall view of how the systems are used and the various roles and actions that take place within the system

Table 4 1: Selection of actors and use case

| Actors | Use case |
|--------|----------|
| | |

| | | |
|---------------|-----------------------|--|
| Administrator | Register | <ul style="list-style-type: none"> • Apply to register |
| Instructor | Login | <ul style="list-style-type: none"> • Upload/Download assignment |
| Student | Logout | <ul style="list-style-type: none"> • question/Answer |
| Registrar | View Grade | <ul style="list-style-type: none"> • Download/Upload module |
| Dep-Head | View Comment | <ul style="list-style-type: none"> • Sublimit Result |
| Coordinator | Submit Result | <ul style="list-style-type: none"> • Approve Result |
| Bank | Update Account | <ul style="list-style-type: none"> • View announcement |
| | Create Account | <ul style="list-style-type: none"> • View Comment |
| | Deactivate Account | <ul style="list-style-type: none"> • Give Comment |
| | Activate Account | <ul style="list-style-type: none"> • Register Student |
| | Add Curriculum | Register Department |
| | Post Announcement | Register Course |
| | Add/drop course | Assign Instructor |
| | View profile | View Report |
| | Update password | Generate report |
| | Manage Payments | Schedule Tutorial |
| | Upload Video | Conduct Exam |
| | Validate Transactions | Pay Online |
| | | Take Exam |
| | | Schedule Exam |

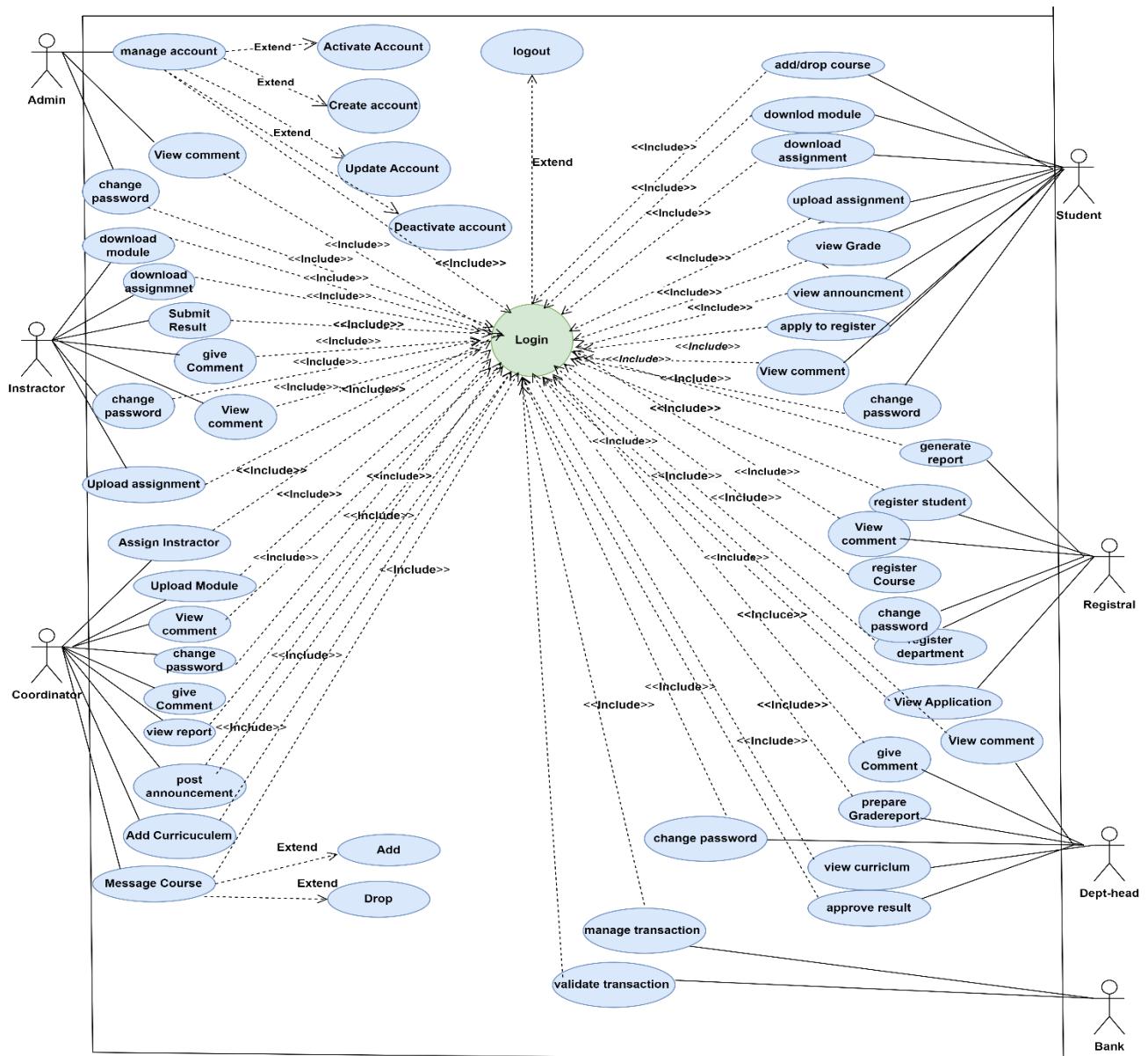


Figure 4. 1: System use case diagram

4.2.1 Use Case Documentation

Table 4 2: Login Use Case Description

| | |
|---------------------|--|
| Use Case Name | Login |
| Identifier | Uc1 |
| Description | All the actors must be login first before using the system by users own account in order to perform their action. |
| Actor | System admin, Coordinator, Registrar-officer, Instructor, student and department head are actors of this System |
| Pre-condition | The user who wants to login must have user name and password. |
| Post-condition | <p>The authenticated user gets the appropriate page.</p> <p>User gets access to the system according to their predefined system privilege and</p> <p>Finally the user can logout or turn off the page.</p> |
| Extends | Logout |
| Include | _____ |
| Flow of events step | <p>Select the login link</p> <p>The system displays the login form</p> <p>Fill user name and password</p> <p>System validates user name and password.</p> <p>The system displays the appropriate page.</p> |
| Alternative flow | If user name and password are incorrect return back to step 2 in flow event |

Table 4 3: Manage account Use Case Description

| | |
|--------------------|---|
| Use case name | Manage account |
| Identifier | Uc2 |
| Description | <ol style="list-style-type: none"> 1. System admin manages users' account that extends update, create, deactivate and activate accounts. 2. Admin Updates user account like changing user name and password. 3. Admin Creates new account for instructors, student, registrar, Coordinator and Department head 4. Admin Deactivate User account 5. Admin Activate User account |
| Actor | Admin |
| Pre-condition | <p>Account is created for all, Students to have account they must be registered first</p> <p>Admin should have to enter a valid user name and password in order to create, update, activate and deactivate user account.</p> |
| Post-condition | <ol style="list-style-type: none"> 1. You update, create, deactivate and activate user account successfully message will be displayed. 2. Finally logout from the page. |
| Extends | Update, create, activate, deactivate account |
| Include | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. Admin should have to choose account item 2. Then click on the selected account item. 3. Then Admin should have to update, create, deactivate and activate. The user's account. |
| Alternative flow | Confirmation/rejection message will be displayed and the admin try again to manage user account. |

Table 4 4: Description of Upload module use case

| | |
|---------------|---------------|
| Use-case name | Upload module |
| Identifier | Uc3 |

| | |
|---------------------|--|
| Description | The process will start by activating the system or the window or the home page and click login link then form will be displayed. |
| Actor | Coordinator |
| Pre-condition | The Coordinator should prepare file to upload And Coordinator should have to enter a valid user name and password. |
| Post-condition | Upload module successfully. Finally logout from the page |
| Extends | _____ |
| Include | Login |
| Flow of event steps | <ol style="list-style-type: none"> 1. The Coordinator should clicks on upload Module link. 2. Coordinator should have to click on browse the file button 3. browses where the file to be uploaded 4. Then click on upload button. 5. Display success full message |
| Alternative flow | If the size or type of file to be uploaded is out of size, the Coordinator should try to upload the file. |

Table 4 5: Download module use case Description

| | |
|----------------|--|
| Use-case name | Download module |
| Identifier | Uc4 |
| Description | The process will start the Student and Instructor by activating the system or the window and the home page and click login link then form will be displayed. |
| Actor | Student, Instructor |
| Pre-condition | <ol style="list-style-type: none"> 1. There should be download module. 2. Student must insert his/her ID number as user name and password. |
| Post-condition | <ol style="list-style-type: none"> 1. Download module material successfully 2. Finally logout from the page |
| Extends | _____ |
| Include | Login |

| | |
|--------------------|---|
| Flow of event step | <ol style="list-style-type: none"> 1. The Student should click on download module link. 2. He/she enter module code and module name 3. Student click on download file button |
| Alternative flow | if file not downloaded it shows No file message and the Student try to download the file |

Table 4 6: download Assignment use case Description

| | |
|--------------------|--|
| Use-case name | Download assignment |
| Identifier | Uc5 |
| Description | <p>This process can be performed by Instructor as well as student.</p> <ol style="list-style-type: none"> 1. An instructor can download assignment answers submitted by Student. 2. Student can download assignment questions that uploaded by Instructor. |
| Actor | Instructor, Student |
| Pre-condition | There should be uploaded assignment questions to Students and There should be submitted assignment answer for Instructor. |
| Post-condition | <ol style="list-style-type: none"> 1. The assignment was downloaded successfully. 2. Finally logout from the page |
| Extends | _____ |
| Include | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. The user click on download assignment link 2. User enters course code and course name. 3. Then user press on download button. |
| Alternative flow | If no uploaded assignment no file message will be displayed |

Table 4 7: Upload assignment use case Description

| | |
|---------------|---|
| Use-case name | Upload assignment |
| Identifier | Uc6 |
| Description | <p>This process can be performed by instructor as well as Student.</p> <ol style="list-style-type: none"> 1. An Instructor uploads assignment questions for Student. 2. Student uploads assignment solutions submit for Instructor. |

| | |
|--------------------|--|
| Actors | Instructor ,Student |
| Pre-condition | 1: An Instructor must prepare assignment questions to upload. 2: Students must do their assignment to submit |
| Post-condition | 1. The assignment was uploaded successfully. 2. Finally logout from the page |
| Extends | _____ |
| Include | Login |
| Flow of event step | 1. The user click on upload assignment link 2. User fills the form. 3. Then he/she press on upload button 4. The system should have to check the submission date. |
| Alternative flow | If submission date has been passed the Student cannot upload the assignment so, Student must submit on the provided date. |

Table 4 8: Register student use case Description

Register student use case Description

| | |
|--------------------|--|
| Use case name | Register student |
| Identifier | Uc7 |
| Description | Registrar office registers Students who fulfil necessary criteria. Send information of Student for Instructor, Academic dean |
| Actor | Registrar officer |
| Pre-condition | 1 Registrar officer enters valid user name and password to get Student registration form. |
| Post-condition | 1. Instructor can access the registered Student list 2. Finally logout from the page |
| Extends | _____ |
| Include | Login |
| Flow of event step | 1. The registrar officer fills the basic information of the Student. 2. After completion of filling the form he/she click on register button. 3. If ID repetition occurs the error message will be displayed |

| | |
|------------------|---|
| Alternative flow | If the same id is present on the same academic year the system displays this Student already registered message. So the registrar officer must give unique ID for each Student on the same academic year. |
|------------------|---|

Table 4-1: Register department use case Description

| | |
|--------------------|---|
| Use case name | Register Department |
| Identifier | Uc8 |
| Description | Registrar office registers Department which is add to the system by Coordinator |
| Actor | Registrar officer |
| Pre-condition | 1 Registrar officer enters valid user name and password to get Department registration form. |
| Post-condition | <ol style="list-style-type: none"> 1. Registrar officer can access the registered Department list 2. Finally logout from the page |
| Extends | _____ |
| Include | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. The registrar officer fills the basic information of the Department. 2. After completion of filling the form he/she click on register button. 3. If Department ID repetition the error message will be displayed |
| Alternative flow | If the same id is present on the same academic year the system displays this course already registered message. So the registrar officer must give unique ID for each course on the same academic year. |

Table 4 9: Register course use case Description

| | |
|----------------|---|
| Use case name | Register Course |
| Identifier | Uc9 |
| Description | Registrar office registers course which is add to the system by Coordinator |
| Actor | Registrar officer |
| Pre-condition | 1 Registrar officer enters valid user name and password to get course registration form. |
| Post-condition | <ol style="list-style-type: none"> 1. Registrar officer can access the registered course list 2. Finally logout from the page |
| Extends | _____ |

| | |
|--------------------|---|
| Include | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. The registrar officer fills the basic information of the course. 2. After completion of filling the form he/she click on register button. 3. If occurs ID repetition the error message will be displayed |
| Alternative flow | If the same id is present on the same academic year the system displays this course already registered message. So the registrar officer must give unique ID for each course on the same academic year. |

Table 4 10: View comment use case Description

| | |
|--------------------|---|
| Use case name | View Comment |
| Identifier | Uc10 |
| Description | Student, Admin, Coordinator, Registrar, Instructor and Department head can view comment. |
| Actor | Student, Admin, Coordinator, Registrar, Instructor and Department head |
| Pre-condition | <ol style="list-style-type: none"> 1. There should be posted comment by Student, Admin, Coordinator, Registrar, Instructor and Department head. 2. They should have to enter a valid user name and password in order to view comment. |
| Post-condition | <ol style="list-style-type: none"> 1. The system displays the comment of the staff. 2. Finally logout from the page. |
| Extends | _____ |
| Include | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. They click on view comment link. 2. The system display comment 3. Then staff clicks on view button. |
| Alternative flow | If the comment giver fills incorrect user name and password try again. |

Table 4 11: Change Password use case description

| | |
|---------------|---------------------------------|
| Use Case Name | Change password |
| Identifier | Uc11 |
| Description | To change the current password. |

| | |
|---------------------|--|
| Actor | System admin, Coordinator, Registrar-officer, Instructor, Department head and Student. |
| Pre-condition | The user should have an account and the users should know the current password to change them. |
| Post-condition | The system display password is changed successfully message. |
| Extends | _____ |
| Include | Login |
| Flow of events step | <ol style="list-style-type: none"> 1. Select the change password link 2. The system displays the change password form 3. Fill current password and new password confirm password. 4. The system displays password is changed successful. |
| Alternative flow | If current password and new password confirm password. Incorrect Tri again |

Table 4.13 Apply to register use case description

| | |
|--------------------|---|
| Use case name | Apply to register |
| Identifier | Uc12 |
| Description | The Students send request to Register |
| Actor | Student |
| Pre-condition | 1 The system display home page and contains the apply links. |
| Post-condition | 1. The system display request is sent successfully |
| Extends | _____ |
| Include | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. Student clicks on apply link. 2. Then he/she fills the form of the apply link. 3. Then he/she click on apply button. |
| Alternative flow | If the Student fills incorrect apply the system display try again the apply request. |

Table 4 12: Assign Instructor use case description

| | |
|--------------------|---|
| Use case name | Assign instructor |
| Identifier | Uc13 |
| Description | The Coordinator assign Instructor to the students |
| Actor | Coordinator |
| Pre-condition | To assign the instructor First the instructor present in the Collage |
| Post-condition | 1. The Instructor assign successfully 2. Finally logout from the page. |
| Extends | _____ |
| Include | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. Coordinator clicks on assign Instructor link. 2. Then Coordinator fills the form. 3. Then Coordinator clicks on assign instructor. |
| Alternative flow | If the Coordinator fills incorrect form must check it. |

Table 4 13: post announcement use case description

| | |
|---------------|-------------------|
| Use case Name | post announcement |
|---------------|-------------------|

Table2.

| | |
|--------------------|--|
| Identifier | Uc14 |
| Description | The announcement should be posted by the Coordinator |
| Actor | Coordinator |
| Pre-condition | The announcement upload by Coordinator |
| Post condition | Successfully add, successfully update and successfully delete |
| Extends | _____ |
| Include | Login |
| Flow of events | <ol style="list-style-type: none"> 1. Execute browser 2. open the home page 3. click on user menu for login 4. select user type as dean which is displayed in the page 5. Fill the login information 6. click on Submit button 7. hover on access menu and click on announcement 8. Access and manage the event and news module like add, update and delete the event and news information |
| Alternative action | Successfully add, successfully update and successfully delete |

Table 4 14: view report use case description

| | |
|--------------------|--|
| Use case name | View report |
| Identifier | Uc15 |
| Description | The report should be send by the registrar |
| Actor | Coordinator |
| Pre-condition | 1 The report sends by the registrar. 2 The Coordinator should insert valid user name and password to view the report. |
| Post-condition | 1. Coordinator views the report. 2. Finally logout from the page. |
| Extends | _____ |
| Include | Login |
| Flow of event step | 1. Coordinator clicks on view report link. 2. Then he/she fills View report code. 3. Then he/she click on view button. |
| Alternative flow | If the registrar fills incorrect Report he/she must check it. |

Table 4 15: Generate report use case description

| | |
|----------------|--|
| Use case name | Generate report |
| Identifier | Uc16 |
| Description | From this use case the registrar generate report |
| Actor | Registrar |
| Pre-condition | 1. There should be generating report. 2. Registrar should have to enter a valid user name and password in order to generate report. |
| Post-condition | 1. The report is generate successfully 2. Finally logout from the page. |
| Extends | _____ |
| Include | Login |

| | |
|--------------------|--|
| Flow of event step | <ol style="list-style-type: none"> 1. Registrar clicks on generate report link. 2. Then he/she fills generate report code. 3. Then he/she click on view button. |
| Alternative flow | If the Registrar fills incorrect Report he/she must check it. |

Table 0-2: View Announcement use case description

| | |
|--------------------|---|
| Use case name | View Announcement |
| Identifier | Uc17 |
| Description | The Student View announcement posted by Coordinator |
| Actor | Student |
| Pre-condition | <ol style="list-style-type: none"> 1. There should be posted announcement by Coordinator. 2. The Student should have to enter a valid user name and password in order to View announcement. |
| Post-condition | <ol style="list-style-type: none"> 1. The system displays the announcement of the Student. 2. Finally logout from the page. |
| Extends | _____ |
| Include | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. Student clicks on view announcement link. 2. The system display announcement 3. Then Student clicks on view button. |
| Alternative flow | If the Student fills incorrect view announcement system admin must check it. |

Table 4 16: view grade use case description

| | |
|----------------|--------------------------------------|
| Use case name | View Grade |
| Identifier | Uc18 |
| Description | Students need to view their grades |
| Actors | Student |
| Pre-condition | Enter Username and Password to login |
| Post-condition | Grades are available for students. |
| Extends | _____ |
| Include | Login |

| | |
|-----------------------------|---|
| Flow of events | <ol style="list-style-type: none"> 1. A user selects the view grade option. 2. System request what the user needs. 3. The user selects the course grade to be displayed. 4. The system displays the request grade automatically. 5. The user views the grade. 6. Use case ends. |
| Alternative Flow of events: | <ol style="list-style-type: none"> 1. The requested course grade is not complete to be viewed. 2. The system prompts the student the request grade is not complete. 3. The system returns to step 2 of normal course. 4. Use case ends. |

Table 4 17: Give Comment use case descriptions

| | |
|--------------------|---|
| Use case name | Give Comment |
| Identifier | Uc19 |
| Description | Student, Department Head, Instructor, Coordinator, Registrar Give Comment to the staff members. |
| Actor | Student, Department Head, Instructor, Coordinator, Registrar |
| Pre-condition | <ol style="list-style-type: none"> 1. There should be posted Comment by Student, Department Head, Instructor, Coordinator and Registrar. 2. The Staff should have to enter a valid user name and password in order to give Comment. |
| Post-condition | <ol style="list-style-type: none"> 1. The system displays the Comment of the Staff. 2. Finally logout from the page. |
| Extends | _____ |
| Include | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. Instructor clicks on Give Comment link. 2. The system display Comment 3. Then Staff clicks on Give button. |
| Alternative flow | If the Staff fills incorrect Give Comment system the system must display pleas try again message. |

Table 4. 21 Submit Result use case description

| | |
|----------------------------|--|
| Use case name | Submit Result |
| Identifier | Uc20 |
| Description | The Instructor insert the result of Students |
| Actors | Instructor |
| Pre-condition | 1. First Enter Username and password to login |
| Post-condition | The Instructor the student results in to registrar. |
| Extends | _____ |
| Include | Login |
| Flow of events | 1. Instructor selects the submit option. 2. System displays submit result page. 3. Instructor selects which course result to be submit. 4. The system submit result Use case ends. |
| Alternative Flow of events | 1. System check the data entered. 2. System inform to the Instructor, you enter incorrect data or empty data and re-enter. 3. The system returns to step 2 of normal course. 3. Use case ends. |

Table 4 18: Add curriculum use case description

| | |
|----------------|--|
| Use case name | Add curriculum |
| Identifier | Uc21 |
| Description | To describe how the Coordinator add course curriculum in to the system |
| Actors | Coordinator |
| Pre-condition | First Enter Username and password to login |
| Post-condition | The Coordinator Add course curriculum into the system |
| Extends | _____ |

| | |
|----------------------------|--|
| Include | Login |
| Flow of events | <ol style="list-style-type: none"> 1. The Coordinator selects the add course curriculum option. 2. System displays add course curriculum page. 3. The Coordinator selects which course curriculum to be posted in to the system. |
| Alternative Flow of events | <ol style="list-style-type: none"> 1 System check the data entered. 2 System inform to the Coordinator, you enter incorrect data or empty data and re-enter. 4. The system returns to step 2 of normal course. 5. Use case ends. |

Table 4.19 Add/Drop Course use case description

| | |
|----------------------------|--|
| Use case name | add/drop Course |
| Identifier | Uc22 |
| Actor | Student |
| Description Actors | To describe how the Student add/drop Course |
| Pre-condition | First Enter Username and password to login in order to access the system |
| Post-condition | The Student performs the activities Add or Drop Courses. |
| Extends | _____ |
| Include | Manage Course |
| Flow of events | <ol style="list-style-type: none"> 1. The Student selects the add/drop form. 2. System displays add /drop course form. 3. Then Students fill the appropriate information to add/drop course. |
| Alternative Flow of events | <ol style="list-style-type: none"> 1 System check the data entered. 2 System inform to the student, you enter incorrect data or empty data and re-enter. 3 The system returns to step 2 of normal course. 4 Use case ends. |

Table 4 19: View Curriculum use case description

| | |
|--------------------|--|
| Use case name | View Curriculum |
| Identifier | Uc23 |
| Description | To describe how the Department Head, Student and Registrar office view the course Curriculum |
| Actor | Department Head, Student and Registrar office |
| Pre-condition | 1 The Curriculum view by them. 2 They should insert valid user name and password to view the Curriculum |
| Post-condition | 1. Department Head, Student and Registrar office views the Curriculum. 2. Finally logout from the page. |
| Extends | _____ |
| Include | Login |
| Flow of event step | 1. They clicks on view Curriculum link. 2. Then they fill view Curriculum. 3. Then they click on view report button. |
| Alternative flow | If they fills incorrect View Curriculum system admin must check it. |

Table 4. 20 Approves result use case description

| | |
|----------------|--|
| Use case name | Approve result |
| Identifier | Uc24 |
| Description | To Describe how the Department Head approves the result which generated by the registrar |
| Actor | Department Head |
| Pre-condition | Department Head enters valid user name and password to approve the result. |
| Post-condition | 1. The Department Head can access the approve result form to perform its action 2. Finally logout from the page |
| Extends | _____ |
| Login | Login |

| | |
|--------------------|---|
| Flow of event step | <ol style="list-style-type: none"> 1. The Department Head fills the basic information to approve result. 2. After completion of filling the form he/she click on approve result button. 3. If ID repetition occurs the error message will be displayed |
| Alternative flow | If the Department Head fills incorrect approve result system admin must check it. |

Table 4 20: view result use case description view result use case description

| | |
|--------------------|--|
| Use case name | View result |
| Identifier | Uc25 |
| Description | The Students can view the result of the course including assignments and final exam posted by the Instructor. |
| Actor | Student |
| Pre-condition | <ol style="list-style-type: none"> 1. There should be posted result by instructor. 2. Instructor should have to enter a valid user name and password in order to post course result. |
| Post-condition | <ol style="list-style-type: none"> 1. The result of the course will be displayed successfully 2. . Finally logout from the page. |
| Extends | _____ |
| Login | Login |
| Flow of event step | <ol style="list-style-type: none"> 1. Student clicks on view course result link. 2. Then he/she fills course code. 3. Then he/she click on view button. |
| Alternative flow | If the Student fills incorrect course code he must check it. |

Table 4.21: Manage Payments

Manage Payments

| Field | Description |
|---------------|---------------------|
| Use Case Name | Manage Payments |
| Identifier | UC1 |
| Actors | Administrator, Bank |

| | |
|------------------|--|
| Pre-condition | Payment records exist in the system. |
| Post-condition | Payments are updated, and reports are generated. |
| Includes | Validate Transactions, Generate Receipt |
| Flow of Events | <ol style="list-style-type: none"> 1. Admin selects "Manage Payments." 2. System displays payment records. 3. Admin validates or updates records. 4. System confirms changes and generates reports |
| Alternative Flow | If validation fails, display an error and log the issue. |

Table 4 21: Upload Video

| Field | Description |
|------------------|--|
| Use Case Name | Upload Video |
| Identifier | UC2 |
| Actors | Instructor |
| Pre-condition | Instructor is logged in and has video content prepared. |
| Post-condition | Video is uploaded and accessible to students. |
| Includes | Validate File Format |
| Flow of Events | <ol style="list-style-type: none"> 1. Instructor selects "Upload Video." 2. System displays upload form. 3. Instructor uploads a file and submits. 4. System validates file format and saves the video |
| Alternative Flow | If file format is invalid, display an error message. |

Table 4 22: Validate Transactions

| Field | Description |
|---------------|-----------------------|
| Use Case Name | Validate Transactions |

| | |
|------------------|--|
| Identifier | UC3 |
| Actors | Bank |
| Pre-condition | Transaction details are submitted by the system. |
| Post-condition | Verify Payment Details |
| Flow of Events | <ol style="list-style-type: none"> 1. System sends transaction details to the bank. 2. Bank validates details. 3. Bank responds with approval or error. |
| Alternative Flow | If validation fails, notify the user and request corrections. |

Table 4. 30 Schedule Tutorial

| Field | Description |
|------------------|---|
| Use Case Name | Schedule Tutorial |
| Identifier | UC4 |
| Actors | instructor |
| Pre-condition | Instructor is logged in and selects a course. |
| Post-condition | Tutorial is scheduled, and notifications are sent to students. |
| Includes | Notify Students |
| Flow of Events | <ol style="list-style-type: none"> 1. Instructor selects "Schedule Tutorial." 2. System displays scheduling options. 3. Instructor selects a time and confirms. 4. System updates the schedule and sends notifications. |
| Alternative Flow | If the selected time is unavailable, suggest alternative slots. |

Table 4. 32 Pay Online

| Field | Description |
|----------------|--|
| Use Case Name | Pay Online |
| Identifier | UC6 |
| Actors | Student |
| Pre-condition | Student is logged in and selects payment functionality. |
| Post-condition | Payment is processed successfully, and receipt is generated. |
| Includes | Validate Transactions |

| | |
|------------------|---|
| Flow of Events | <ol style="list-style-type: none"> 1. Student selects "Pay Online." 2. System displays payment options. 3. Student enters payment details. 4. System validates and processes the payment. 5. System confirms and displays a receipt. |
| Alternative Flow | If payment fails, display an error and allow retry. |

Table 4. 33 Take Exam

| Field | Field |
|------------------|---|
| Use Case Name | Take Exam |
| Identifier | UC7 |
| Actors | Student, Instructor |
| Pre-condition | Student is registered for the course and exam. |
| Post-condition | Exam is completed and submitted for grading. |
| Includes | Validate Student Registration |
| Flow of Events | <ol style="list-style-type: none"> 1. Student selects "Take Exam." 2. System displays exam questions. 3. Student answers and submits the exam. 4. System saves responses for grading. |
| Alternative Flow | If session times out, save answers automatically. |

Table 4 23: Join Tutorial

| Field | Description |
|------------------|--|
| Use Case Name | Join Tutorial |
| Identifier | UC8 |
| Actors | Student |
| Pre-condition | Tutorial is scheduled, and student is notified |
| Post-condition | Student attends the tutorial. |
| Includes | View Tutorial Details |
| Flow of Events | <ol style="list-style-type: none"> 1. Student selects "Join Tutorial." 2. System verifies schedule and allows access. 3. Tutorial is conducted. |
| Alternative Flow | If tutorial is canceled, notify the student. |

Figure 4. 2: user interface prototype

4.4.1 Class Diagram

The **Class Diagram** illustrates the conceptual relationships among objects/classes. This diagram provides a high-level overview of the system's structure, focusing on the relationships between key classes without including specific details like visibility, cardinality, or methods.

Data Dictionary

Below is a combined table that includes the attributes, data types, sizes, and constraints for all identified entities in the Web-Based Continuing and Distance Education Management System

| Entity | Attribute | Data Type | Key/Constraints |
|-------------------|------------------|------------------|------------------------|
| Student | StudentID | INT | Primary Key |
| | Name | VARCHAR | Not Null |
| | Email | VARCHAR | Unique, Not Null |
| | Password | VARCHAR | Not Null, Encrypted |
| Instructor | InstructorID | INT | Primary Key |
| | Name | VARCHAR | Not Null |
| | Email | VARCHAR | Unique, Not Null |
| | Password | VARCHAR | Not Null, Encrypted |
| Course | CourseID | INT | Primary Key |
| | Name | VARCHAR | Not Null |
| | Description | TEXT | Null |
| Material | MaterialID | INT | Primary Key |
| | Title | VARCHAR | Not Null |
| | File | BLOB | Not Null |
| | UploadDate | DATE | Not Null |
| Grade | GradeID | INT | Primary Key |
| | StudentID | INT | Foreign Key |
| | Marks | FLOAT | Not Null |

4.4 Activity diagrams

Activity diagrams, which are related to program flow plans (flowcharts), are used to illustrate activities. In the external view, we use activity diagrams for the description of those business processes that describe the functionality of the business system. [7]

Contrary to use case diagrams, in activity diagrams it is obvious whether actors can perform business use cases together or independently from one another. Activity diagrams allow you to think functionally. Purists of the object-oriented approach probably dislike this fact.

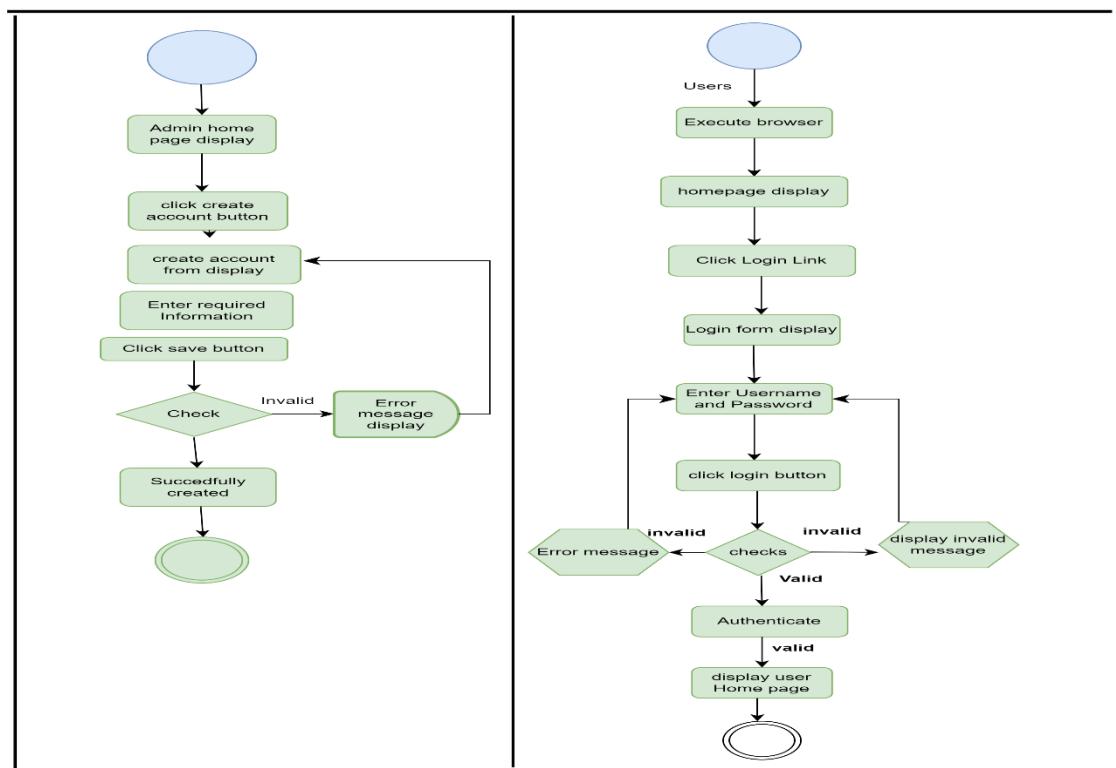


Figure 4. 3: Login and Create account activity diagram

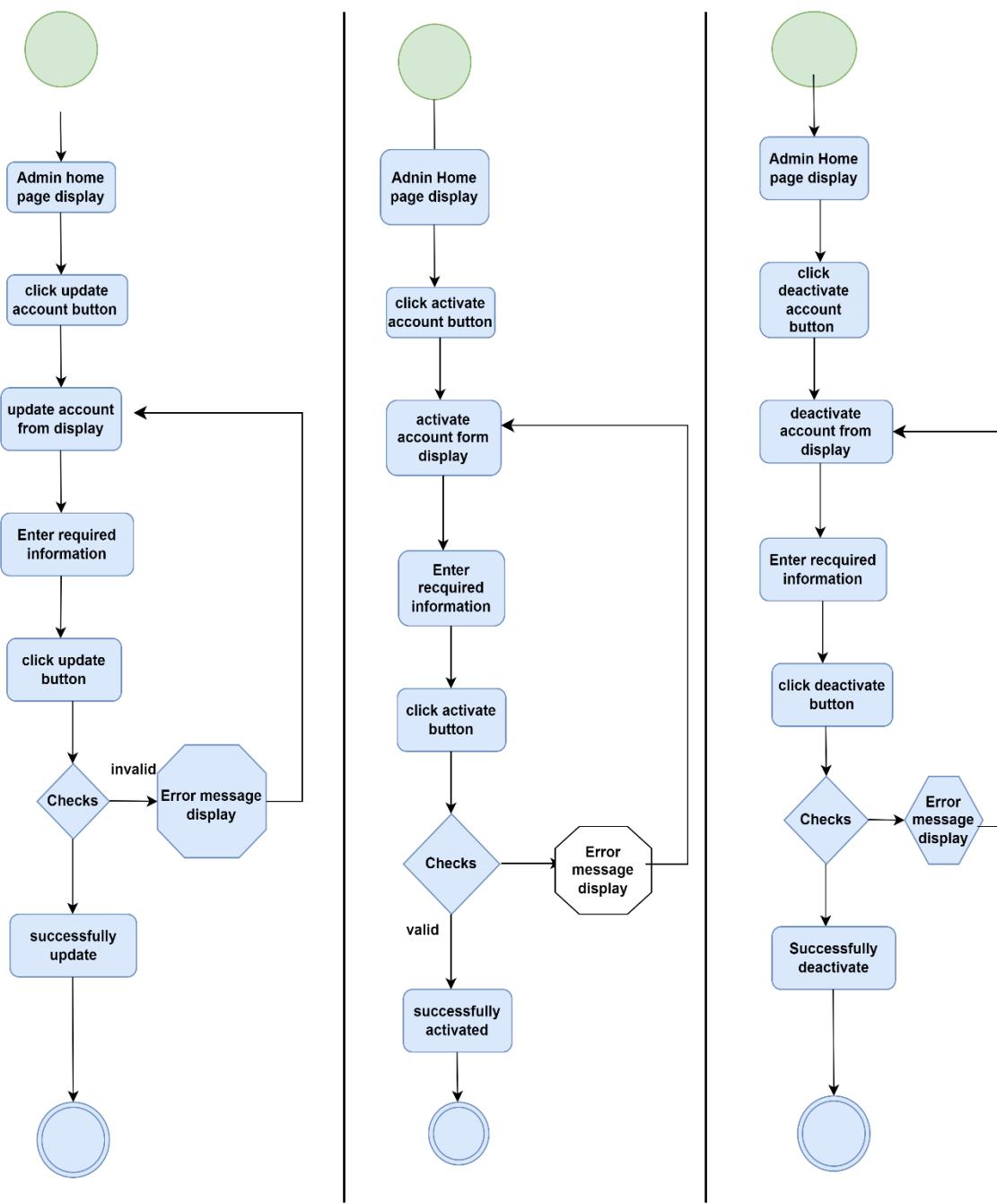
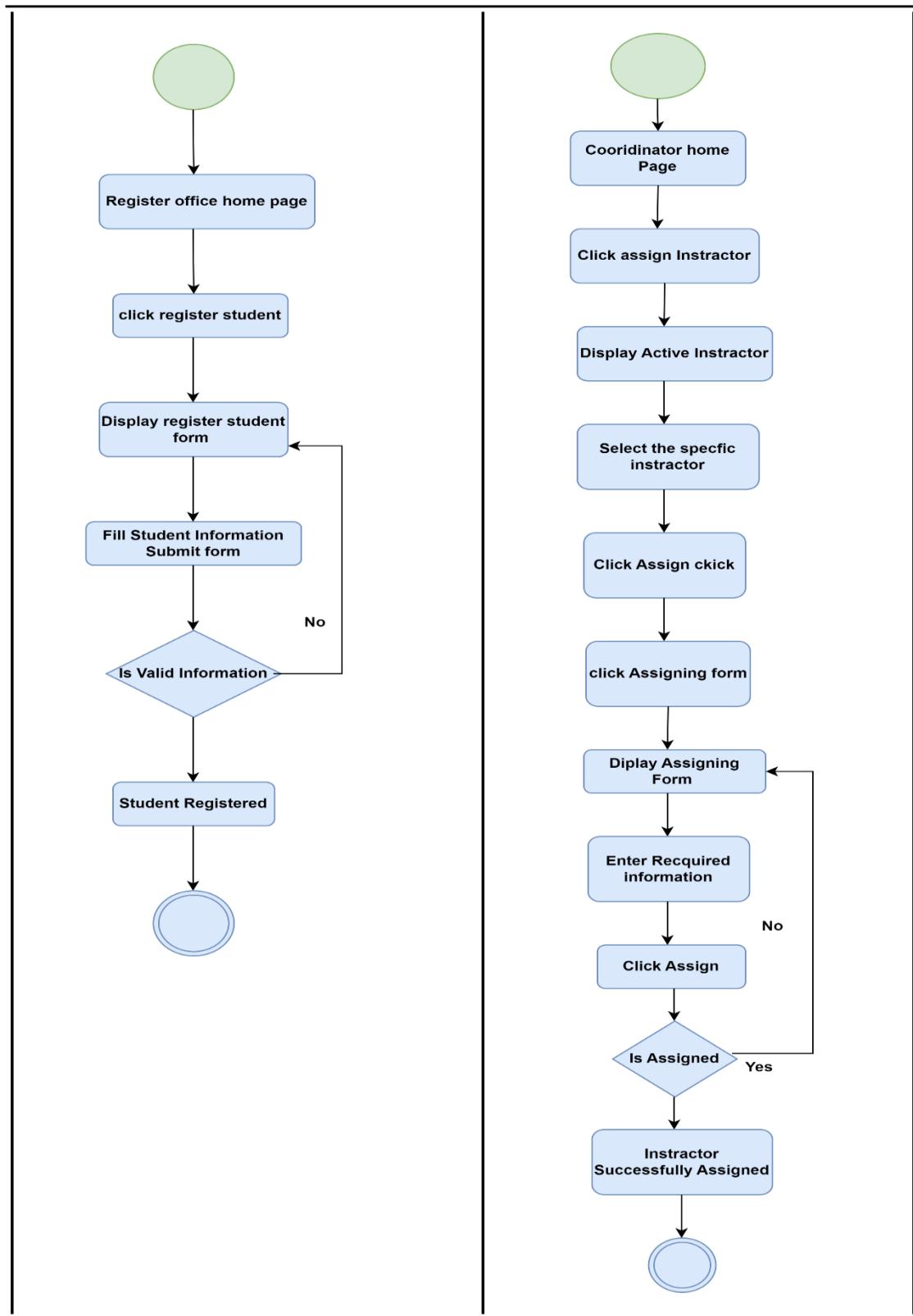


Figure 4. 4: Activate, Deactivate and Update activity diagram Student registration and Assign Instructor activity diagram



X

Figure 4. 5: Student registration and Assign Instructor activity diagram



Figure 4. 6: Download assignment Question and Solutions activity diagram

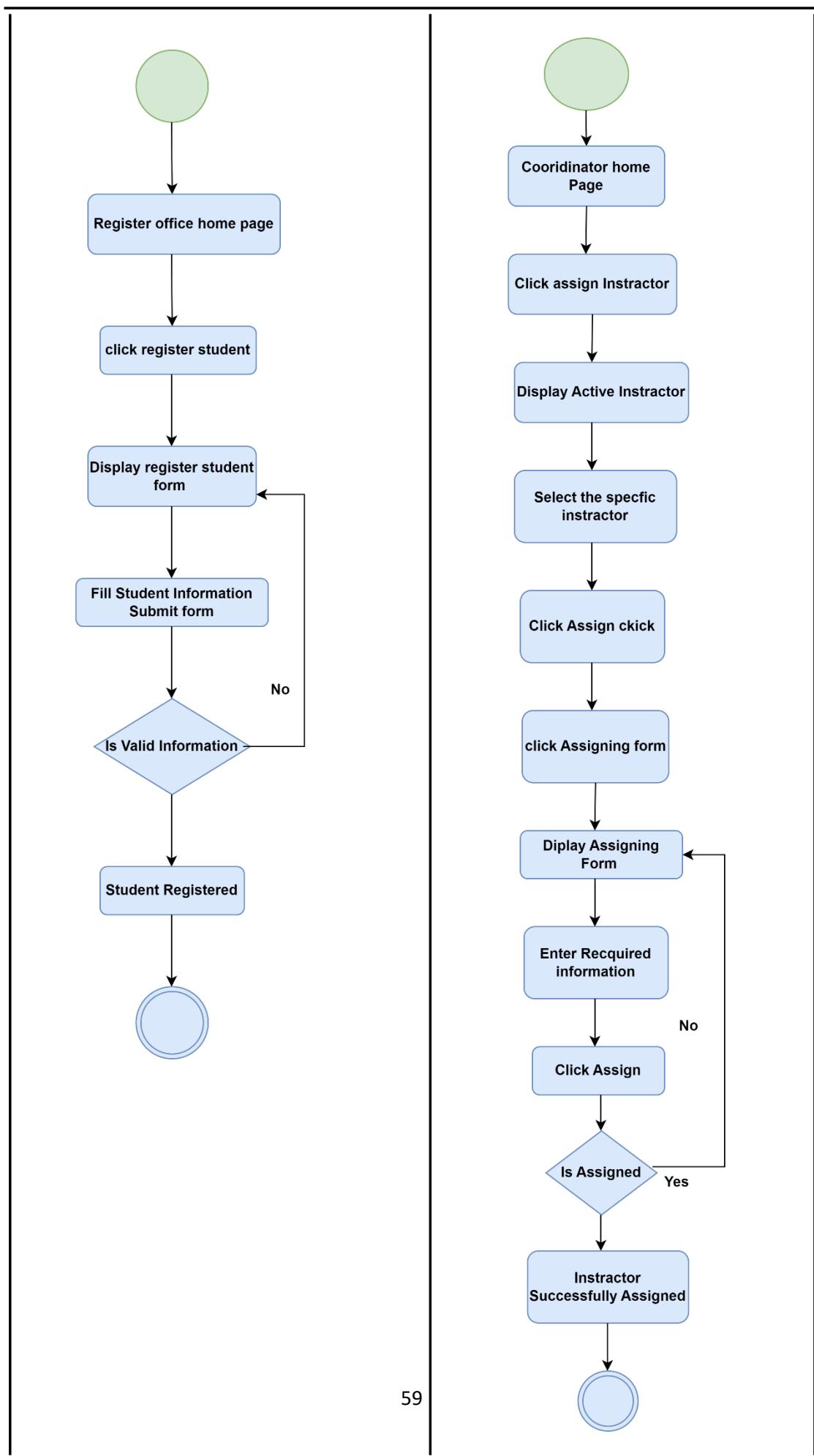


Figure 4. 7: Download assignment Question and Solutions activity diagram

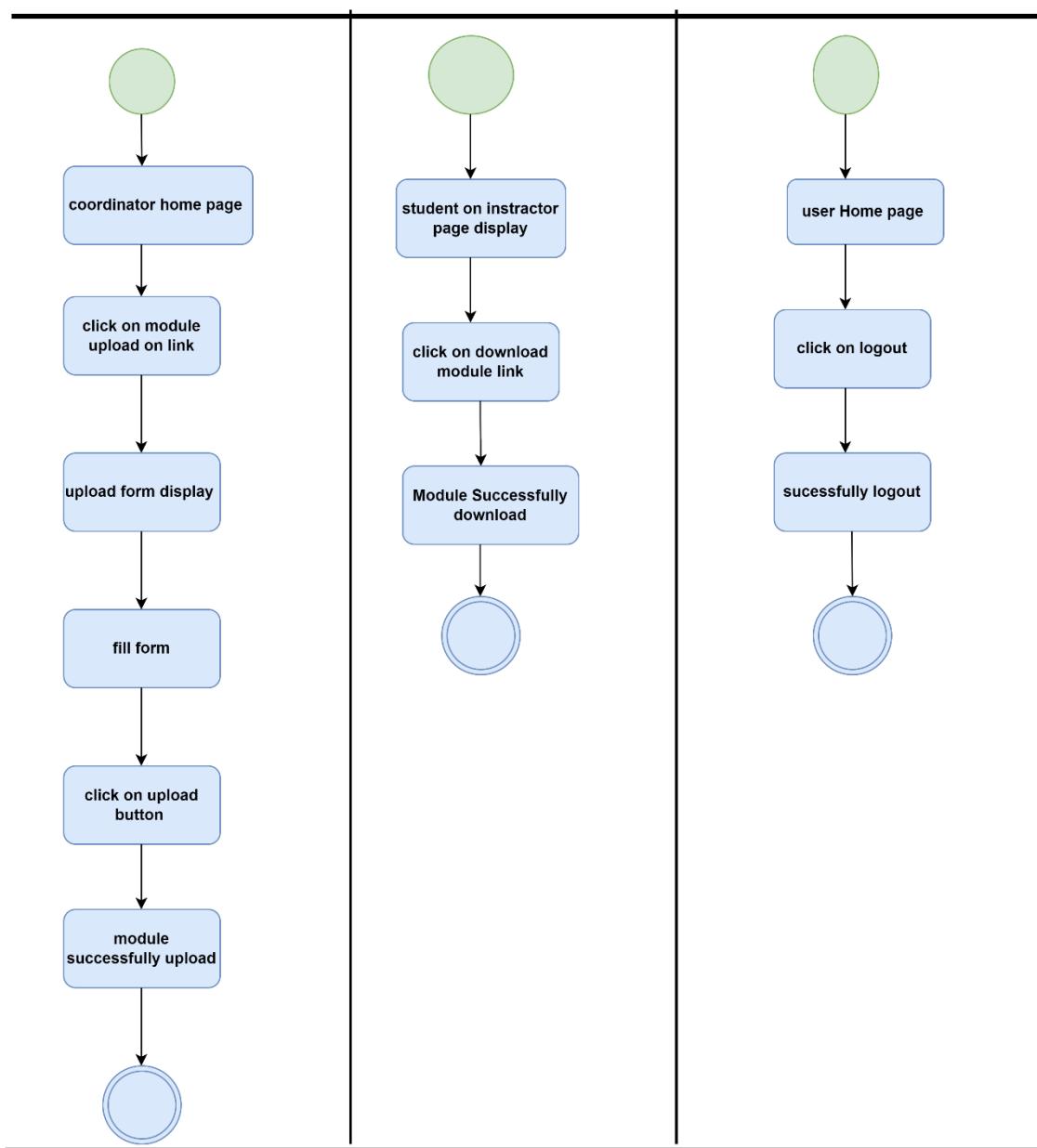


Figure 4. 8: Upload and Download Module and Logout Activity Diagram

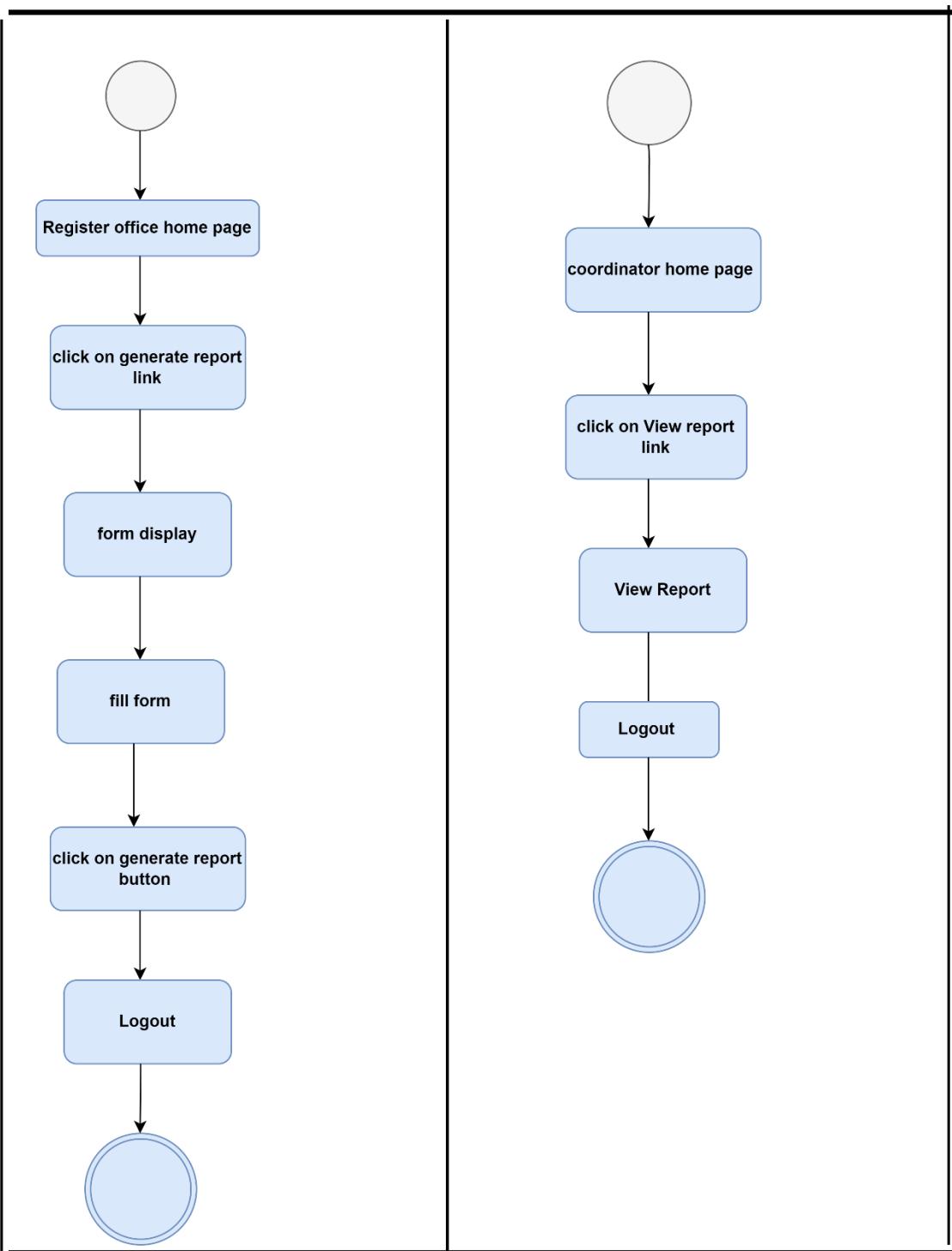


Figure 4. 9: Generate report and View report Activity Diagram

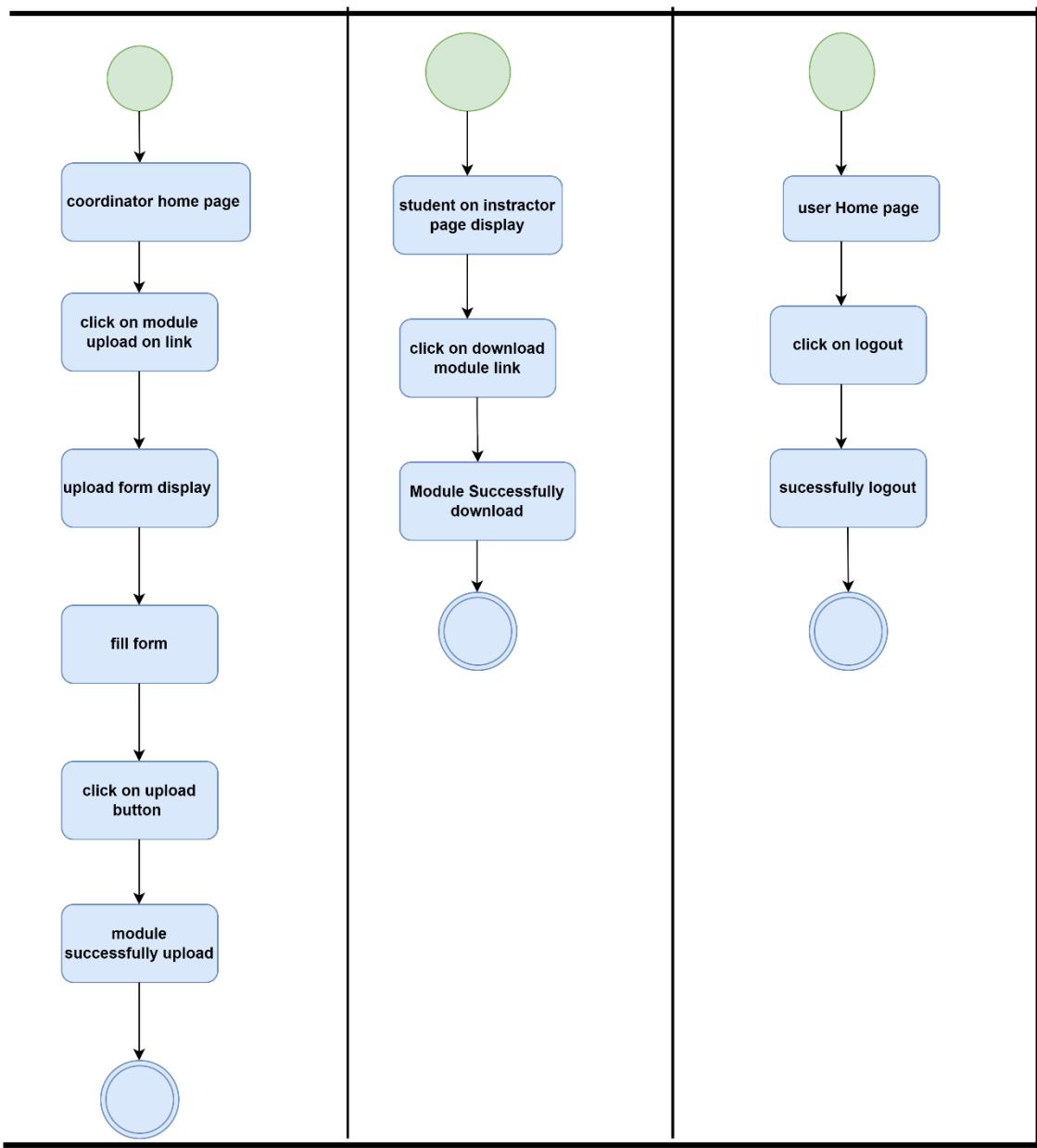


Figure 4. 10: Post Announcement and Give Comment activity diagram

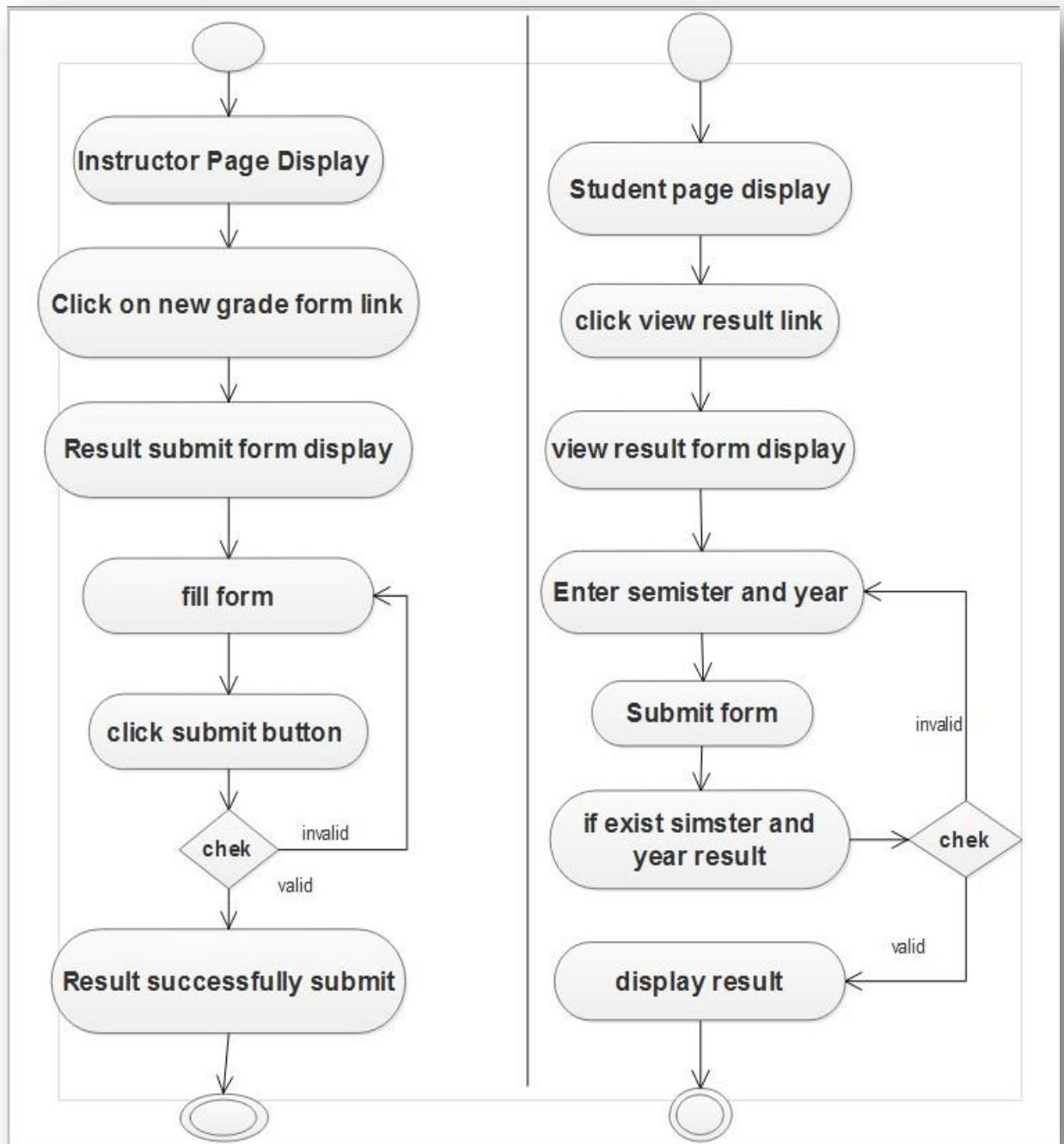


Figure 4. 11: Submit and View result activity diagram

Online Payments

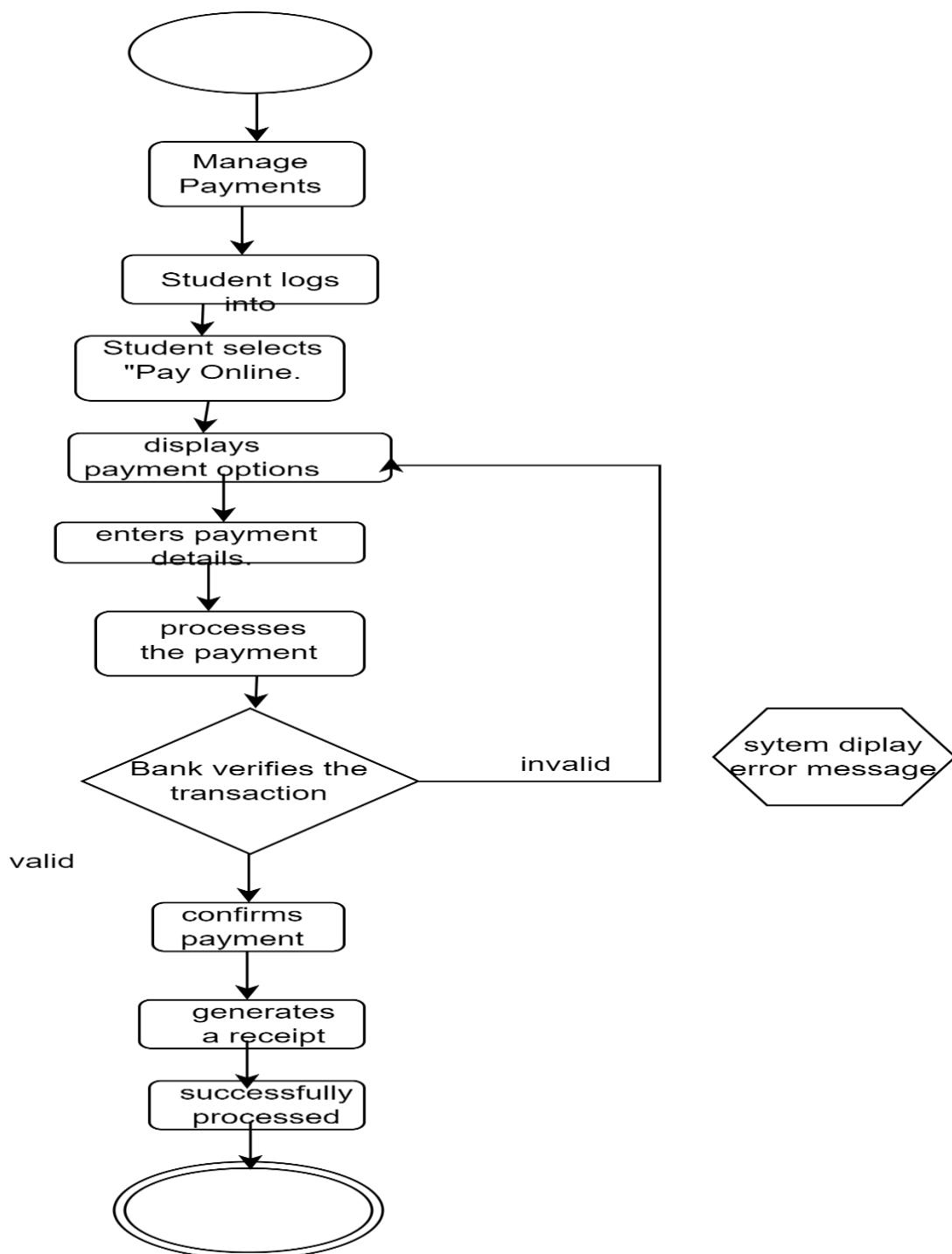


Figure 4. 12: Online Payments and Transactions

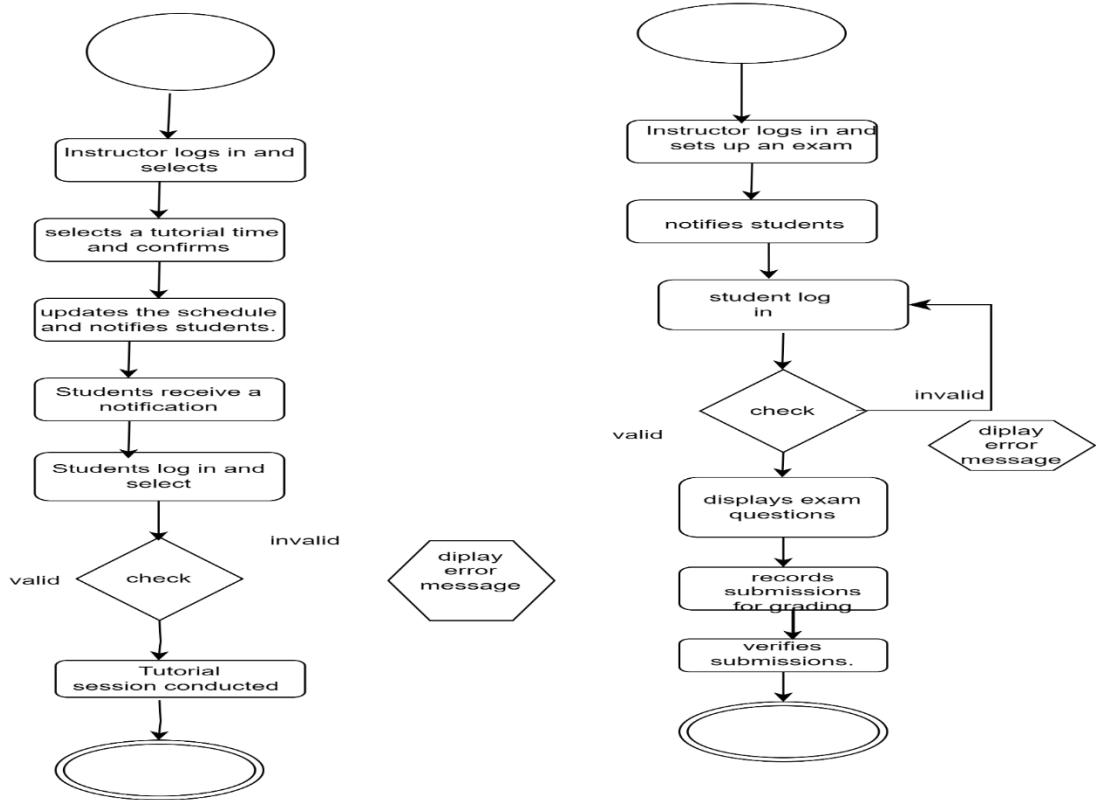


Figure 4. 13: Scheduling and Joining

4.6. Sequence diagram model

A sequence diagram links use case with objects. It shows the interaction between participating objects in a given use case.

It is helpful to identify the missing objects that are not identified in the analysis object model.

From the use case and the class diagrams shown in the previous section the sequence diagrams of the system is shown as follows:

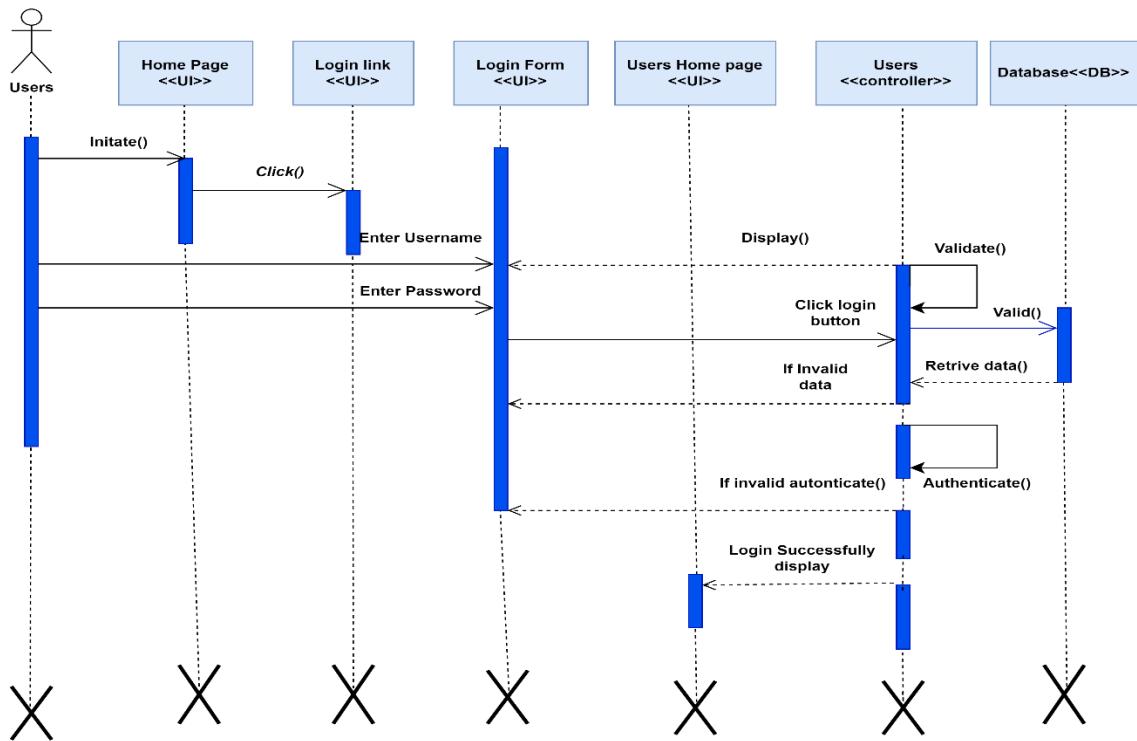


Figure 4. 14: Sequence diagram for Staff Member login

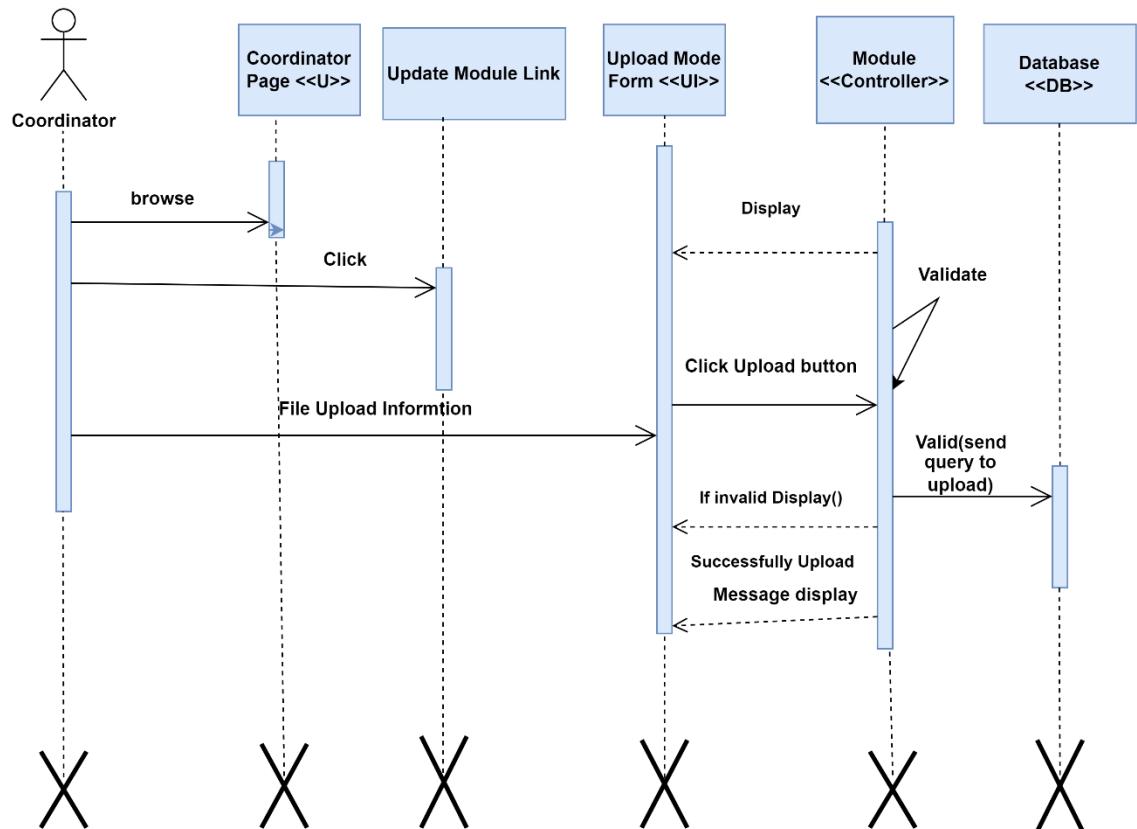


Figure 4. 15: Sequence diagram for upload module

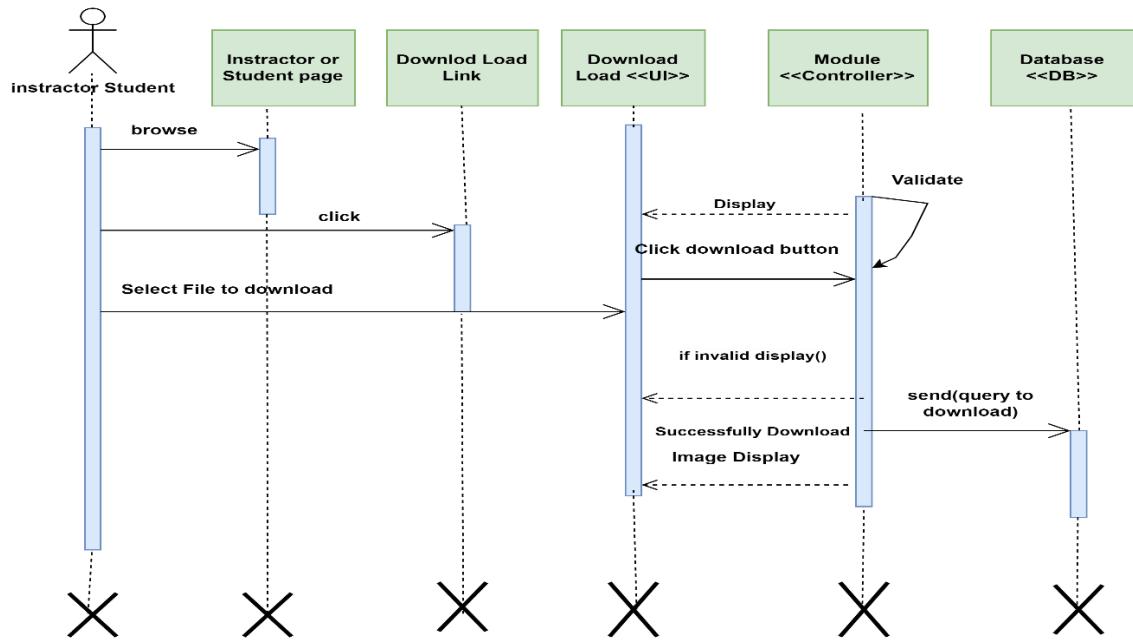


Figure 4. 16: Sequence diagram for Download module

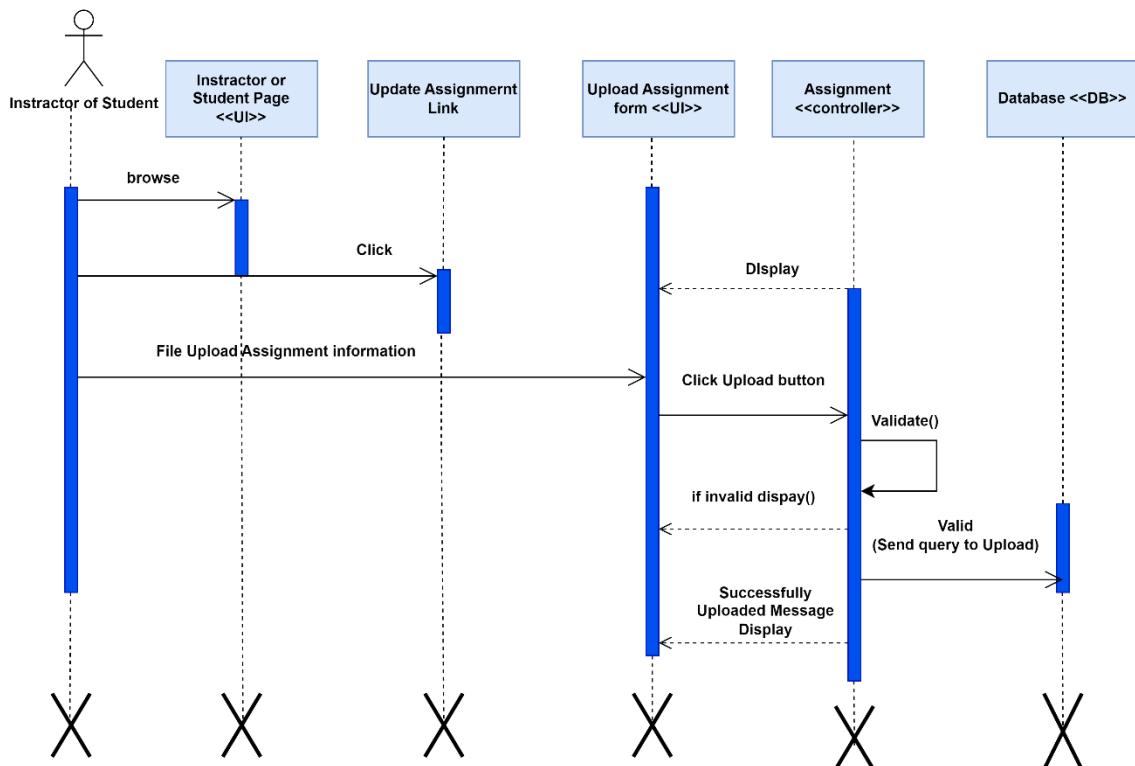


Figure 4. 17: 16 Sequence diagram for upload assignment

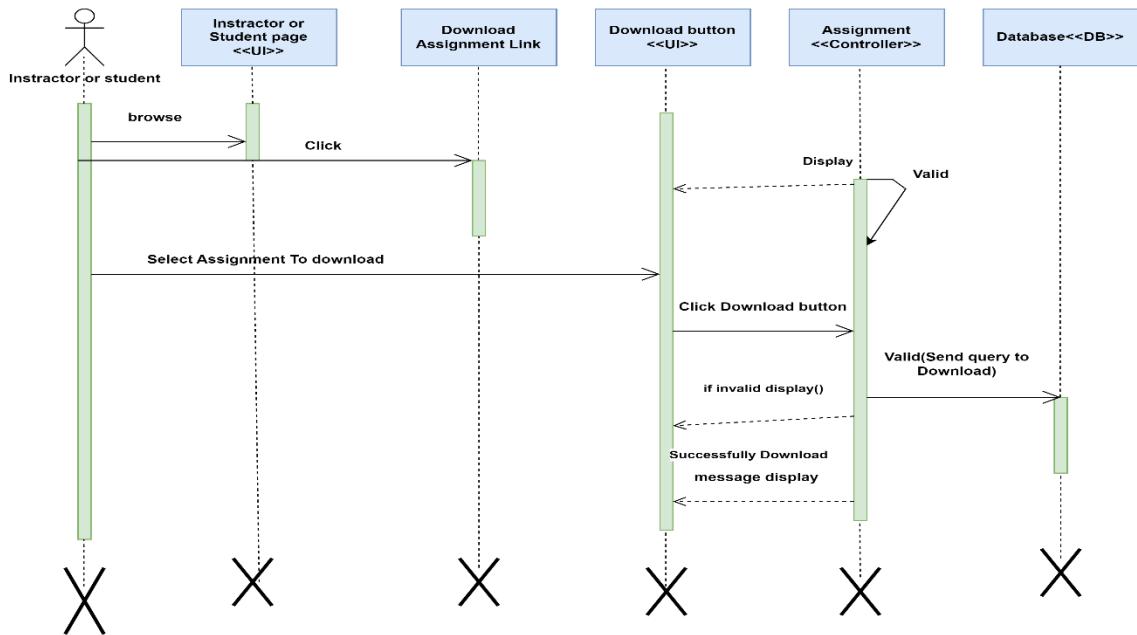


Figure 4.17 Sequence diagram for download assignment

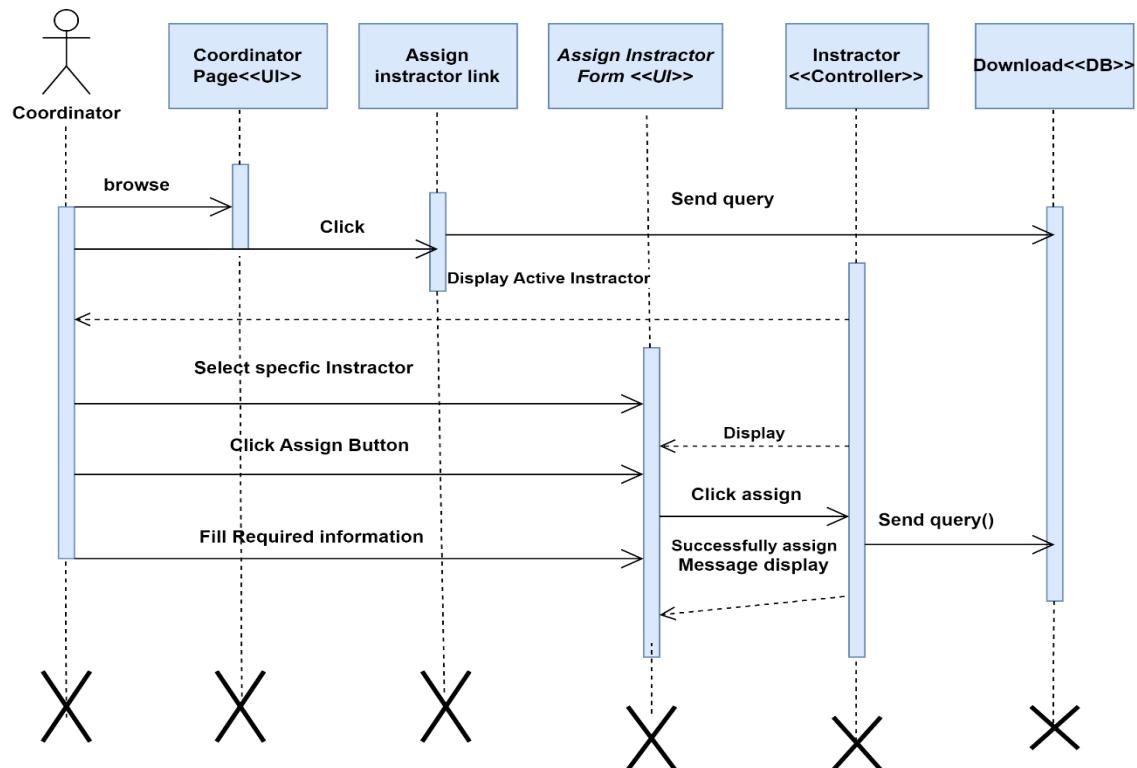


Figure 4. 18: Sequence diagram for Assign Instructor

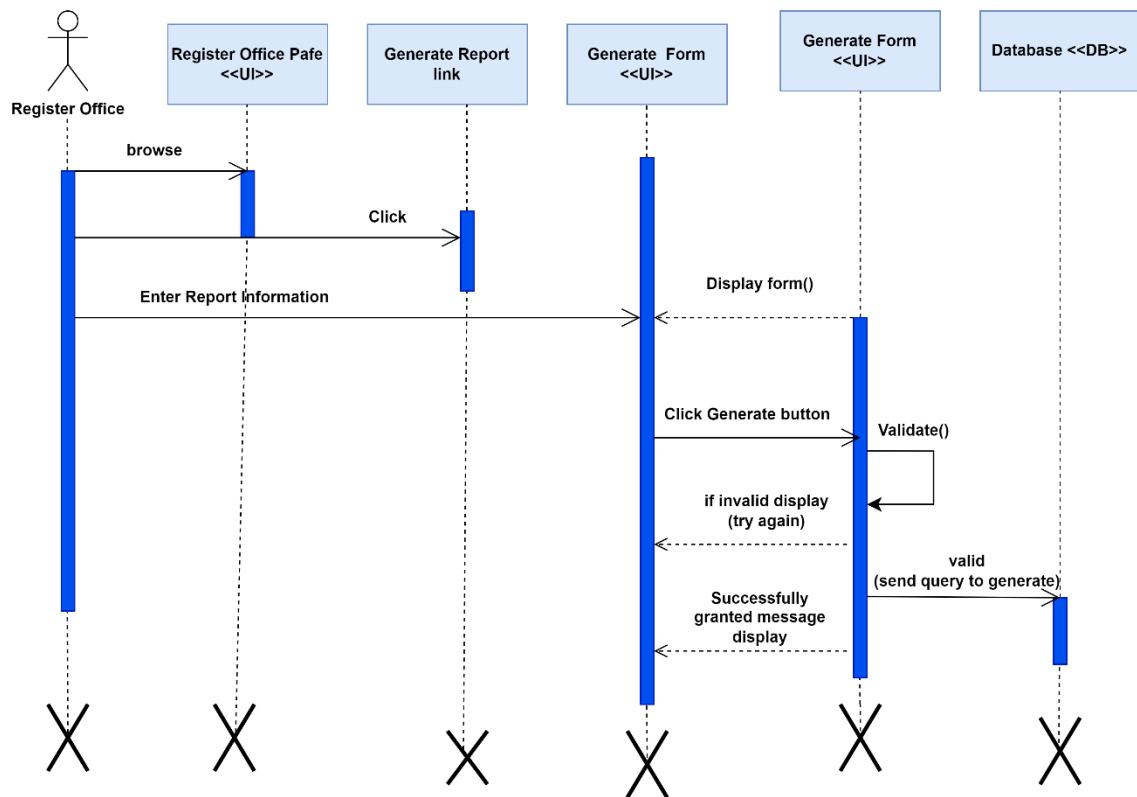


Figure 4. 19: Sequence diagram for generate report

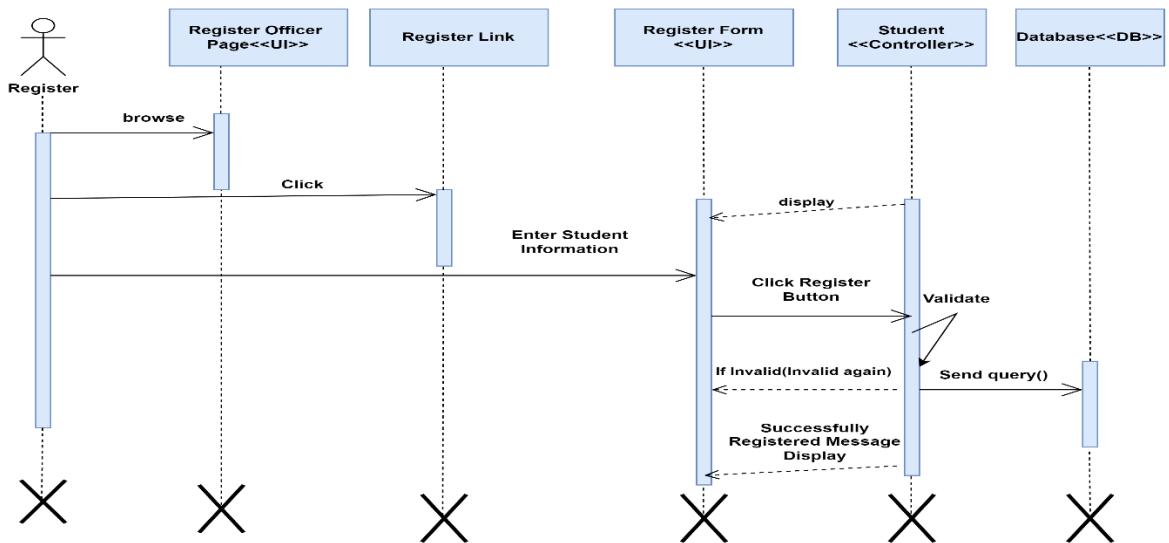


Figure 4.18 Sequence diagram for student registration

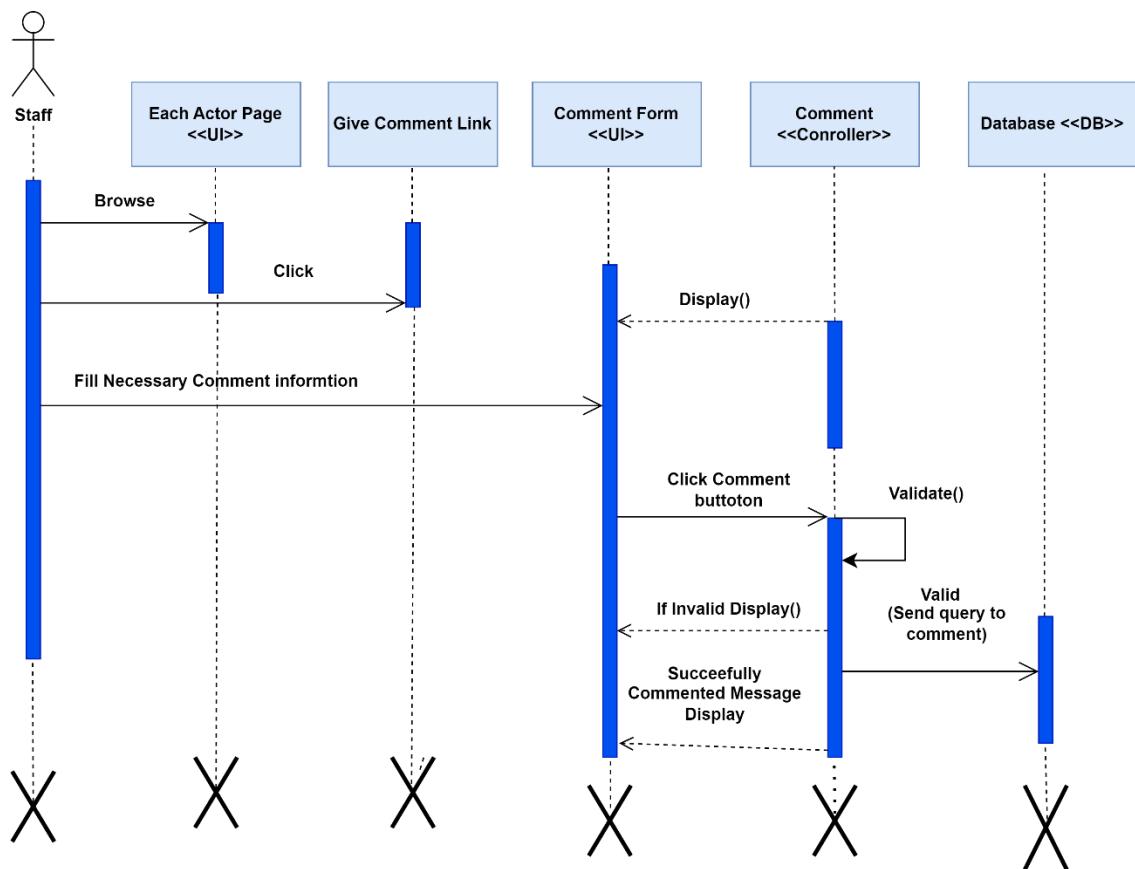


Figure 4.20 Sequence diagram for give comment

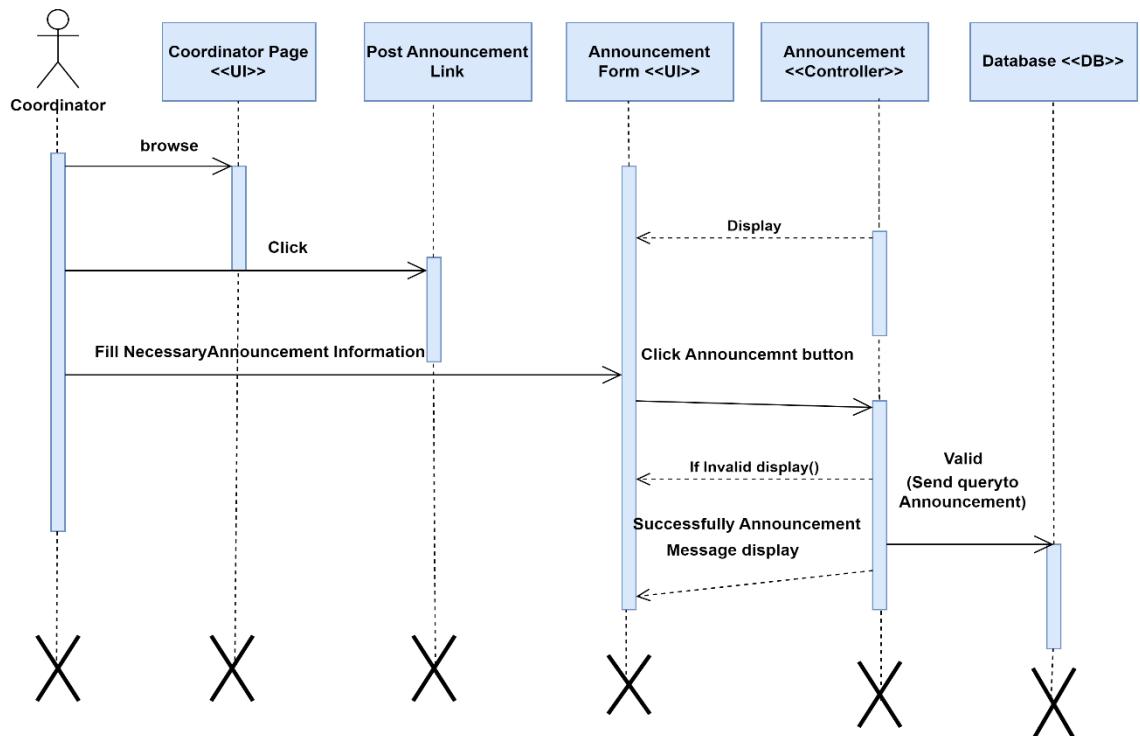


Figure 4. 20: Sequence diagram for post announcement

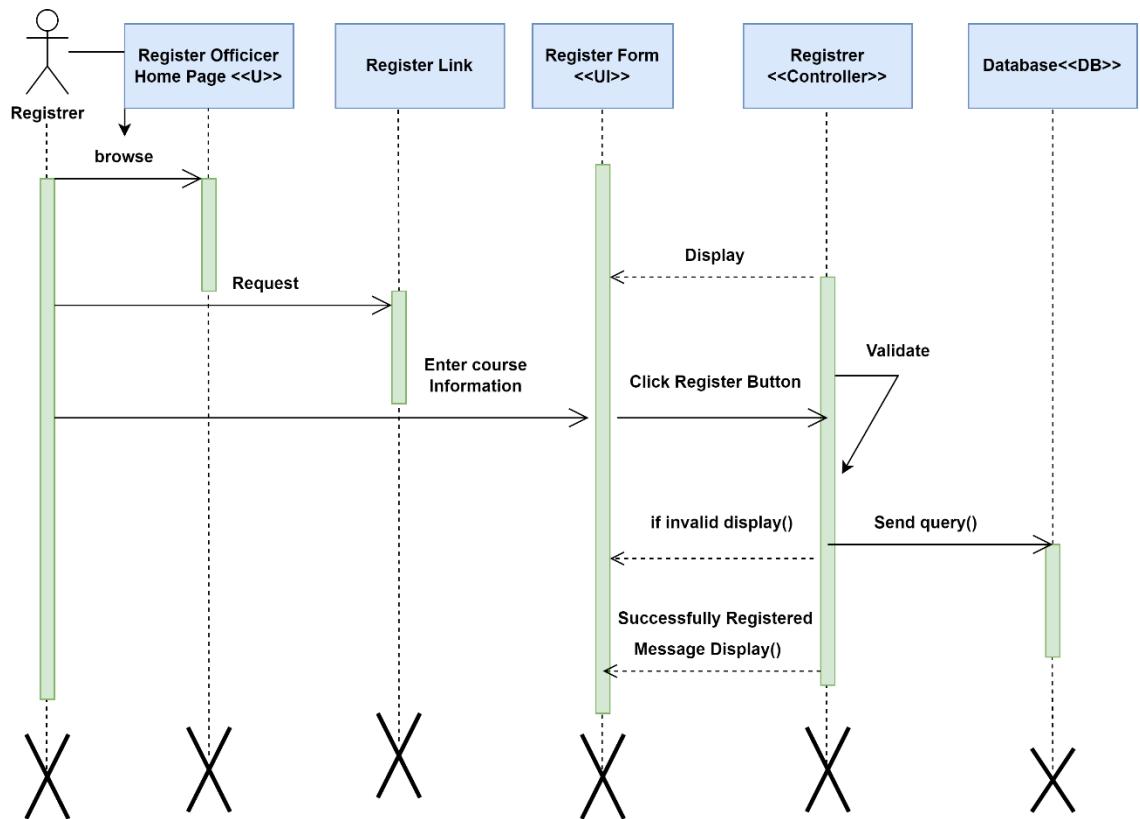


Figure 4.22 Sequence diagram for register course

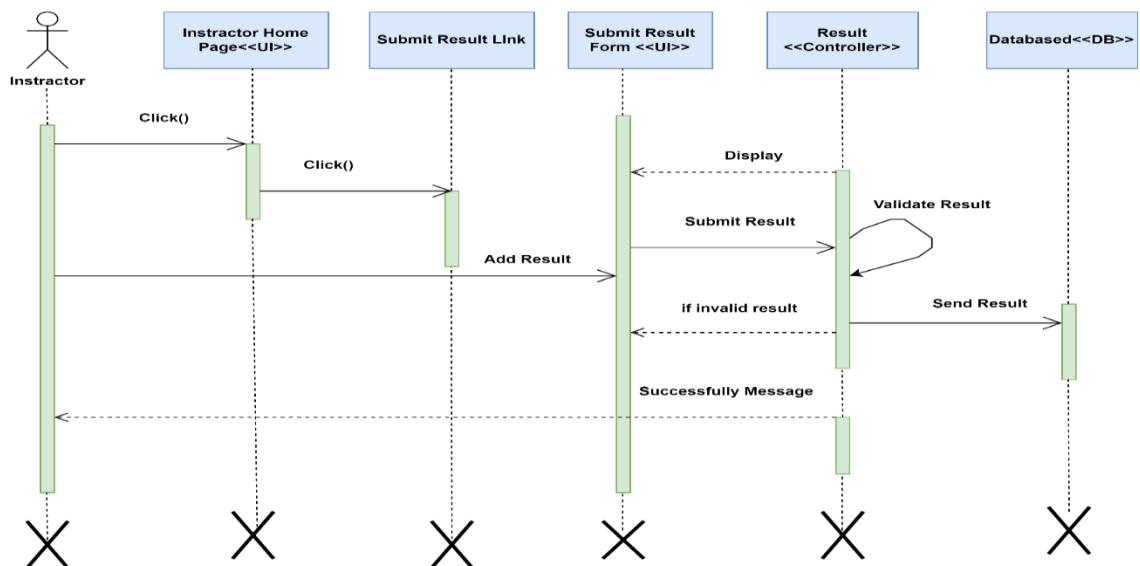


Figure 4. 21: Sequence diagram for Submit result

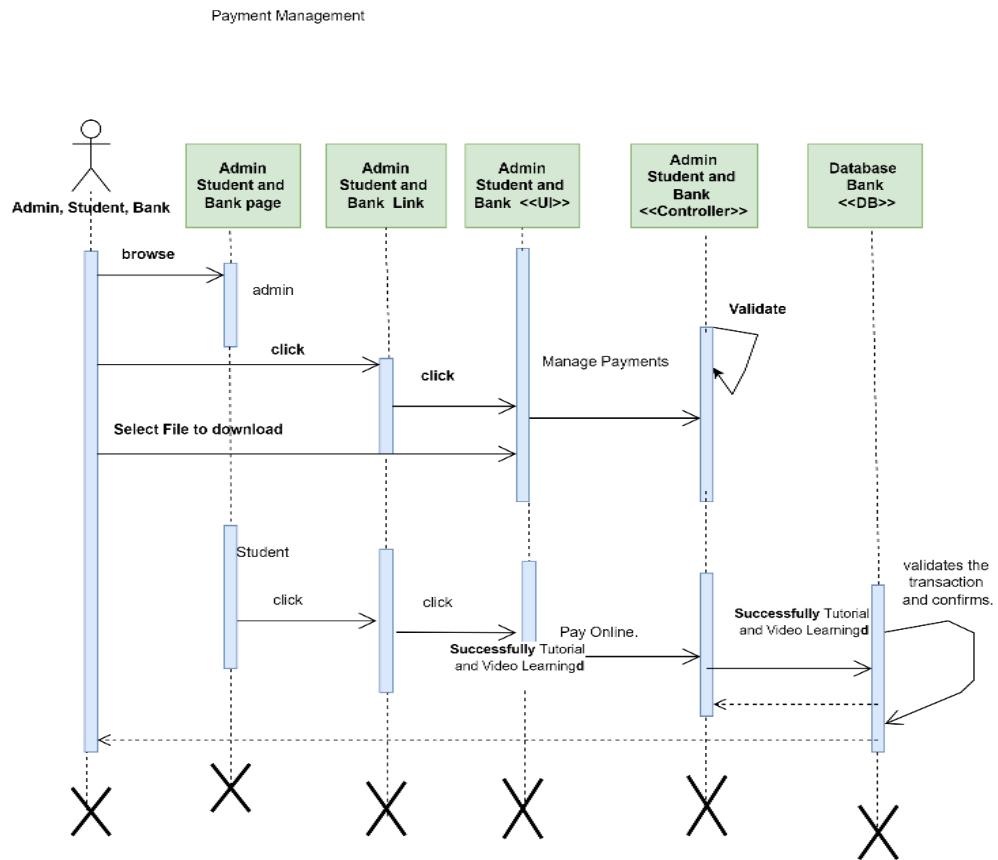


Figure 4.22 payment management

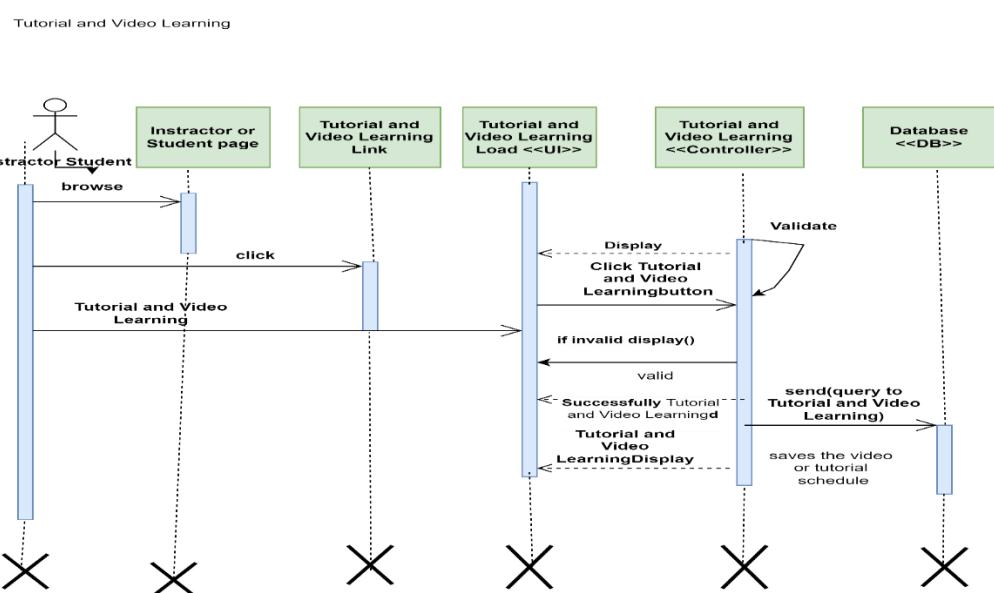


Figure 4. 22: tutorial and video learning

4.5. State Chart Diagram

A State Chart Diagram represents the lifecycle of an object, defining its various states and the events that trigger transitions between these states. Below is an example for a Student Account in the Web-Based Continuing and Distance Education Management System:

States of the Student Account

Inactive: The account is created but not activated.

Initial state.

Active: The student logs in and gains access to the system.

Pending Enrollment:

The student submits a course enrollment request, waiting for approval.

Enrolled:

The student's enrollment request is approved, and they are officially part of the course.

Graduated: The student completes all course requirements and exits the system.

Blocked: The account is restricted due to policy violations or system issues.

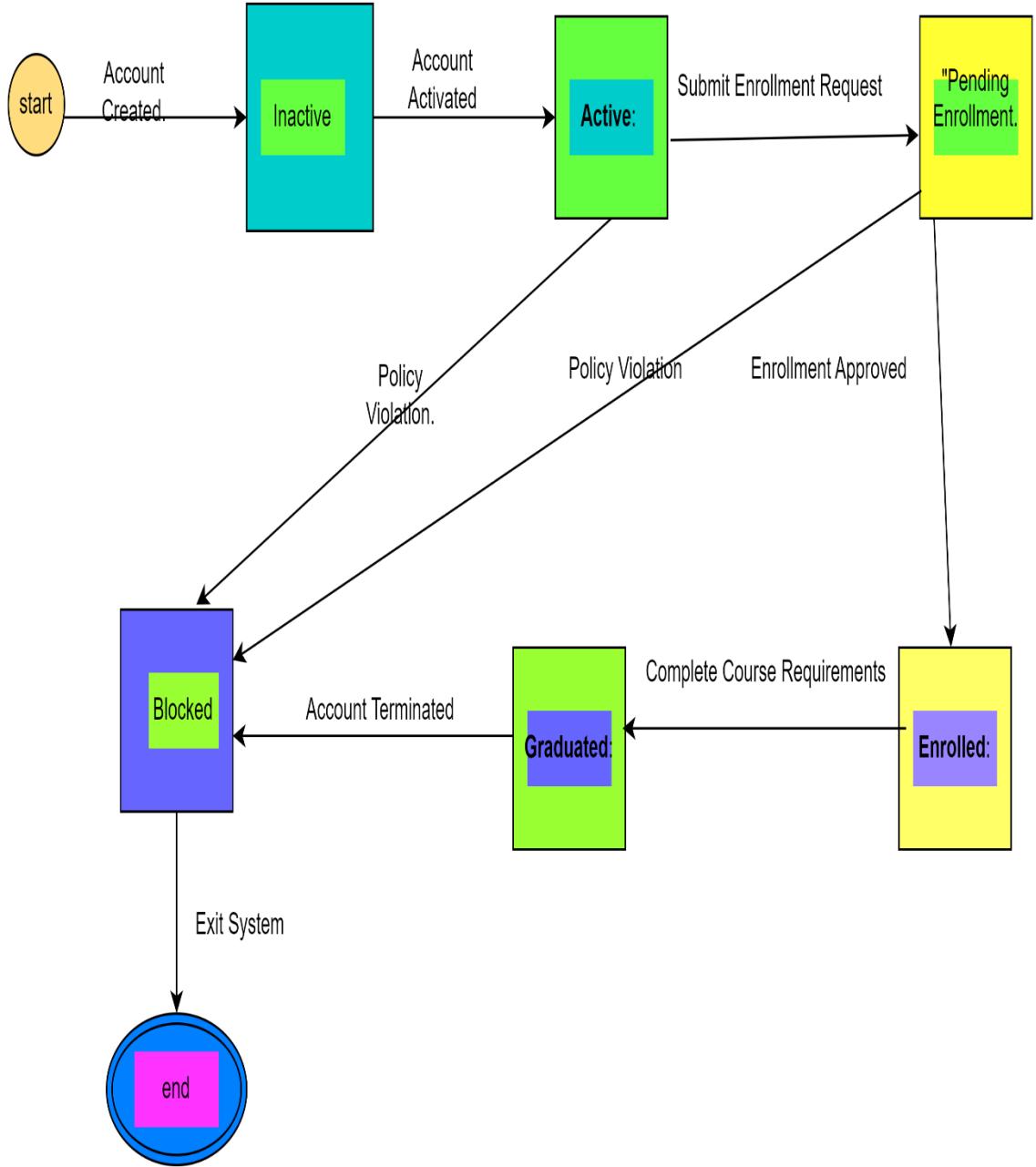


Figure 4. 25 state diagram

CHAPTER FIFE

5. SYSTEM DESIGN

5.1. Introduction

System design is the transformation of the analysis model into a system design model.

System design is the first part to get into the solution domain in a software development.

This chapter provides the design part of the project is discussed. [8]

5.3 Design Goals

The design goals outline the key qualities the system aims to achieve during development.

These goals align with the non-functional requirements and ensure the system meets user expectations and operational needs.

User Interface and Human Factors: Provide an intuitive, accessible, and responsive interface that accommodates users with varying technical expertise. Role-based dashboards for students, instructors, and administrators

Hardware Consideration: Ensure compatibility with modern devices and efficient utilization of server resources.

Security Issues: Protect the system from unauthorized access and secure sensitive data. Multi-factor authentication, role-based access control RBAC.

Performance Consideration: Optimize response times and support high concurrency.

Error Handling and Validation: Handle exceptions gracefully and validate all user inputs to ensure system reliability. Provide actionable error messages, log critical errors, and sanitize user inputs to prevent vulnerabilities.

Quality Issues: Deliver a reliable, robust, and maintainable system.

Backup and Recovery: Ensure minimal data loss and quick recovery in case of failures. Daily incremental backups, weekly full backups, and automated recovery processes.

Physical Environment: Ensure system deployment in a secure, controlled, and stable environment.

Resource Issues: Optimize computational, storage, and bandwidth resources for scalability and cost efficiency. Efficient database indexing, cloud-based scalability, and load balancing during peak usage.

Documentation: Provide comprehensive user, technical, and development documentation. User manuals for students and staff, system architecture diagrams for maintainers, and process logs for future development.

5.3 Proposed System Architecture

The **DCEMS** uses a three-tier architecture for modularity and scalability:

Presentation Layer: User interface for students, instructors, and administrators.

Technologies: HTML, CSS, JavaScript, React.js.

Application Layer: Processes business logic like enrollment, grade management, and notifications. Technologies: Node.js or Django with REST APIs.

Data Layer: Stores and manages user, course, grade, and material data. Technologies: MySQL or PostgreSQL.

Functionality Assignment:

User Management: Authentication, role-based access, and session management.

Course Management: Enrollment handling and course data storage.

Grade Management: Input, validation, and storage of grades.

Material Distribution: Upload and secure access to materials.

Communication: Notifications and announcements delivery.

Reporting: Generating performance and system usage reports.

5.3.1 Subsystem Decomposition and Description

The **DCEMS** is divided into the following key subsystems, each with specific responsibilities:

User Management: Handles user registration, login, and role-based access control.

Course Management: Manages course creation, enrollment, and listings.

Grade Management: Manages grade input and viewing.

Material Distribution: Facilitates uploading and accessing course materials.

Communication: Manages notifications and messages between users.

Reporting: Generates reports on student performance and other system activities.

These subsystems are interconnected:

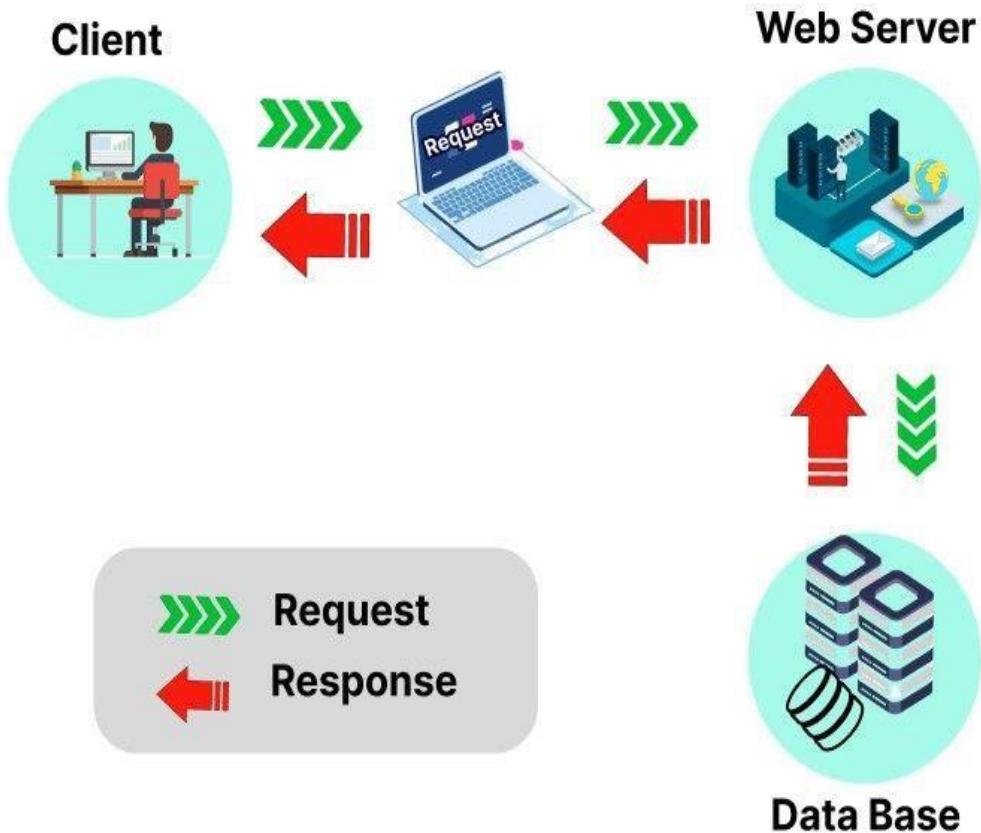
User Management connects with all subsystems for authentication and access control.

Course Management interacts with Grade Management and Material Distribution.

Grade Management works with Course Management to assign grades to students.

Material Distribution provides materials specific to courses.

Reporting aggregates data from User Management, Course Management, and Grade Management.



5. 2.1. Component model

Components are generally units of computation or operation in the system. A component has a name, which is generally chosen to represent the role of the component or the functions Web-Based Continuing and Distance Education Management System.

The different components of a system are likely to interact while the system is in operation to provide the services expected of the system. After all, components exist to provide parts of the services and features of the system, and these must be combined to deliver the overall system functionality.

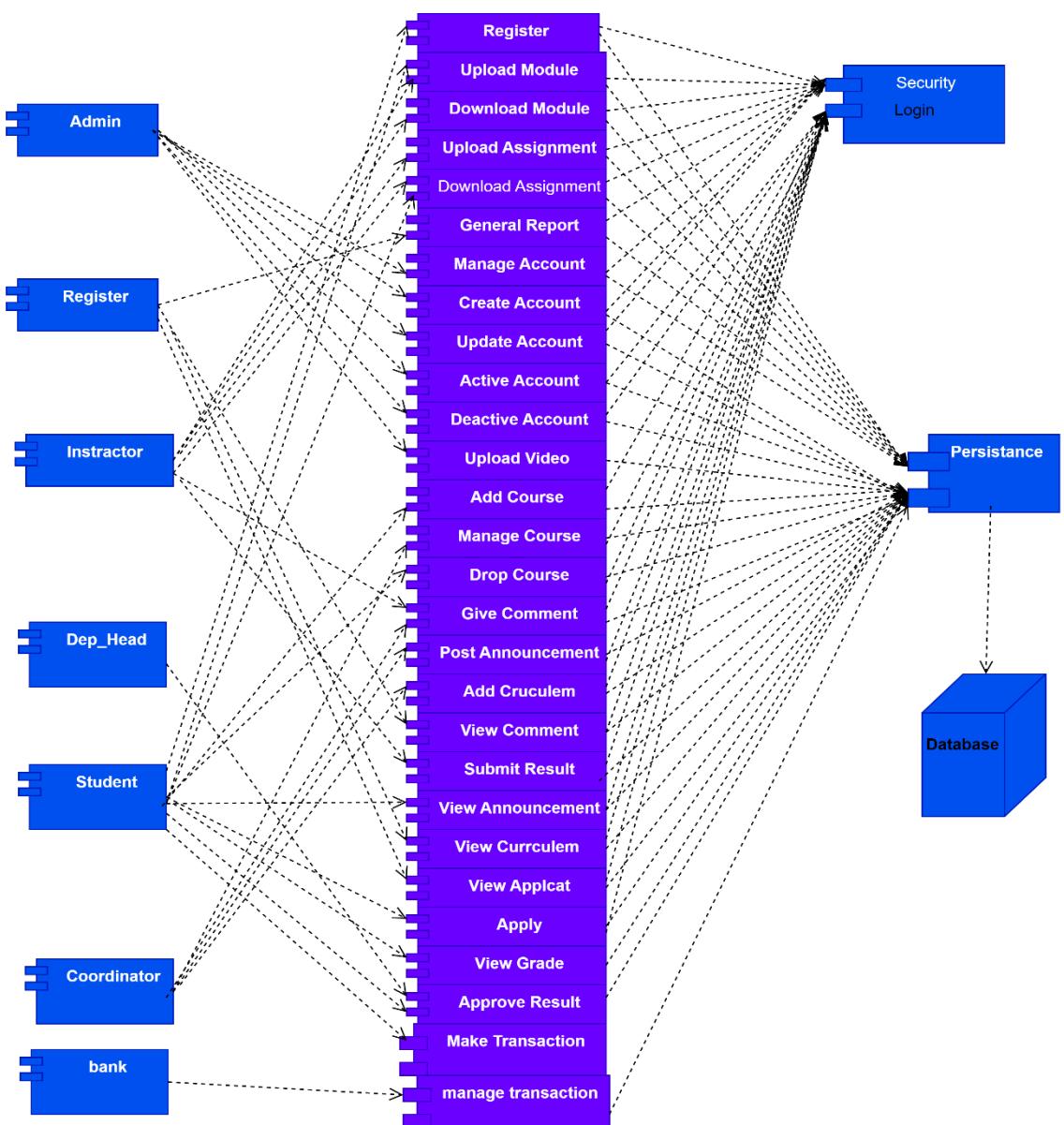


Figure 4. 23: component model diagram

5.3.2 Hardware/Software Mapping

Hardware/Software mapping illustrates how the subsystems of the Web-Based Continuing and Distance Education Management System are mapped to the hardware resources and software components. This mapping ensures that each subsystem runs on the appropriate infrastructure for optimal performance.

User Management Subsystem:

Hardware: Web Server

Software: Node.js/ React.js for frontend.

Course Management Subsystem:

Hardware: Web Server/Database Server.

Software: MySQL/ Node.js/

Grade Management Subsystem:

Hardware: Web Server/Database Server.

Software: MySQL/PostgreSQL, Node.js/Django.

Material Distribution Subsystem:

Hardware: Cloud Storage.

Software: React.js.

UML Deployment Diagram: The UML Deployment Diagram would illustrate the physical components (hardware) and how they map to software components. It will depict nodes such as:

Web Server with components like the User Management Subsystem, Course Management Subsystem, and Frontend Interface.

Database Server hosting MySQL/ for Grade Management and Course Data.

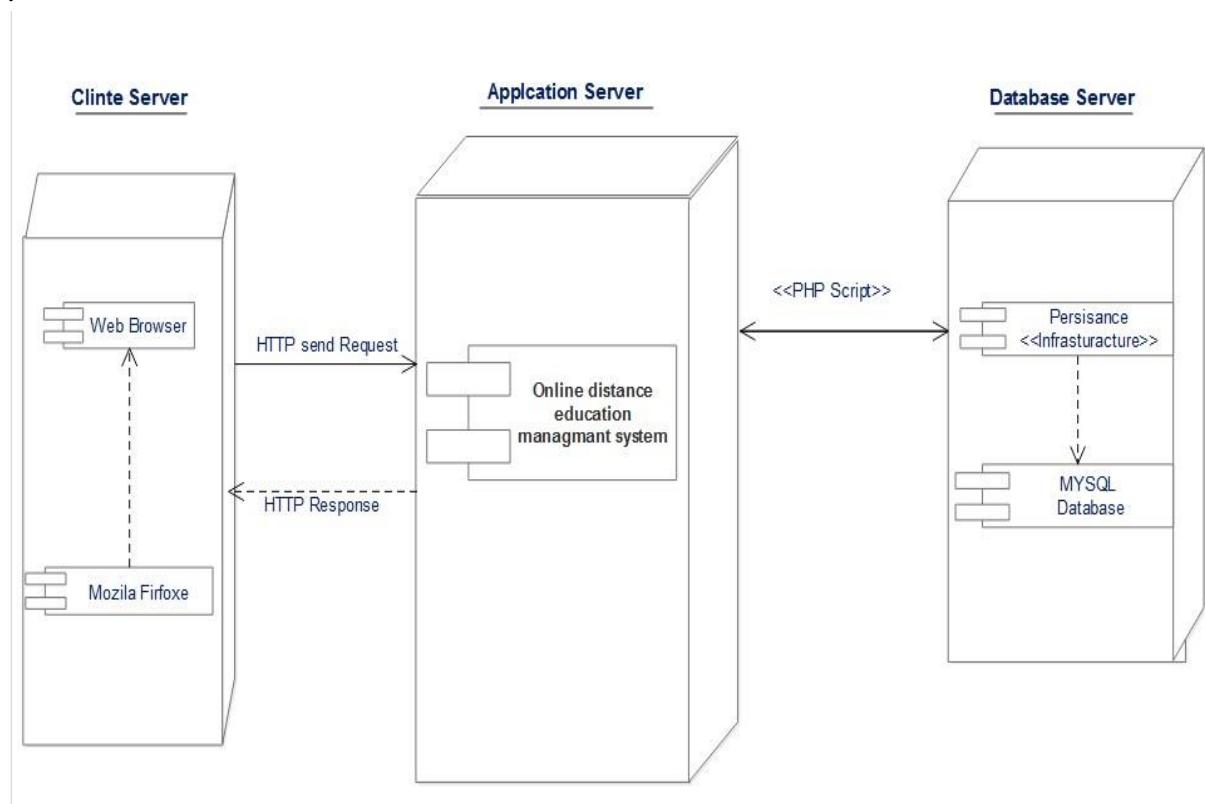


Figure 4. 24: Deployment Diagram

5.3. Class Design model

Class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. Graphically, a class diagram is drawn as a rectangle with three compartments holding the class name, attributes, and operation. [9]

Class Names: Every class must have a name that distinguishes it from other classes. A name is a textual string. That name alone is known as a simple name; a qualified name is the class name prefixed by the name of the package in which that class lives.

Attributes: An attribute is a named property of a class that describes a range of values that instances of the property may hold. A class may have any number of attributes or no attributes at all. An attribute represents some property of the thing you are modelling that is shared by all objects of that class.

Operations: An operation is the implementation of a service that can be requested from any object of the class to affect behavior. In other words, an operation is an abstraction of something you can do to an object that is shared by all objects of that class. A class may have any number of operations or no operations at all.

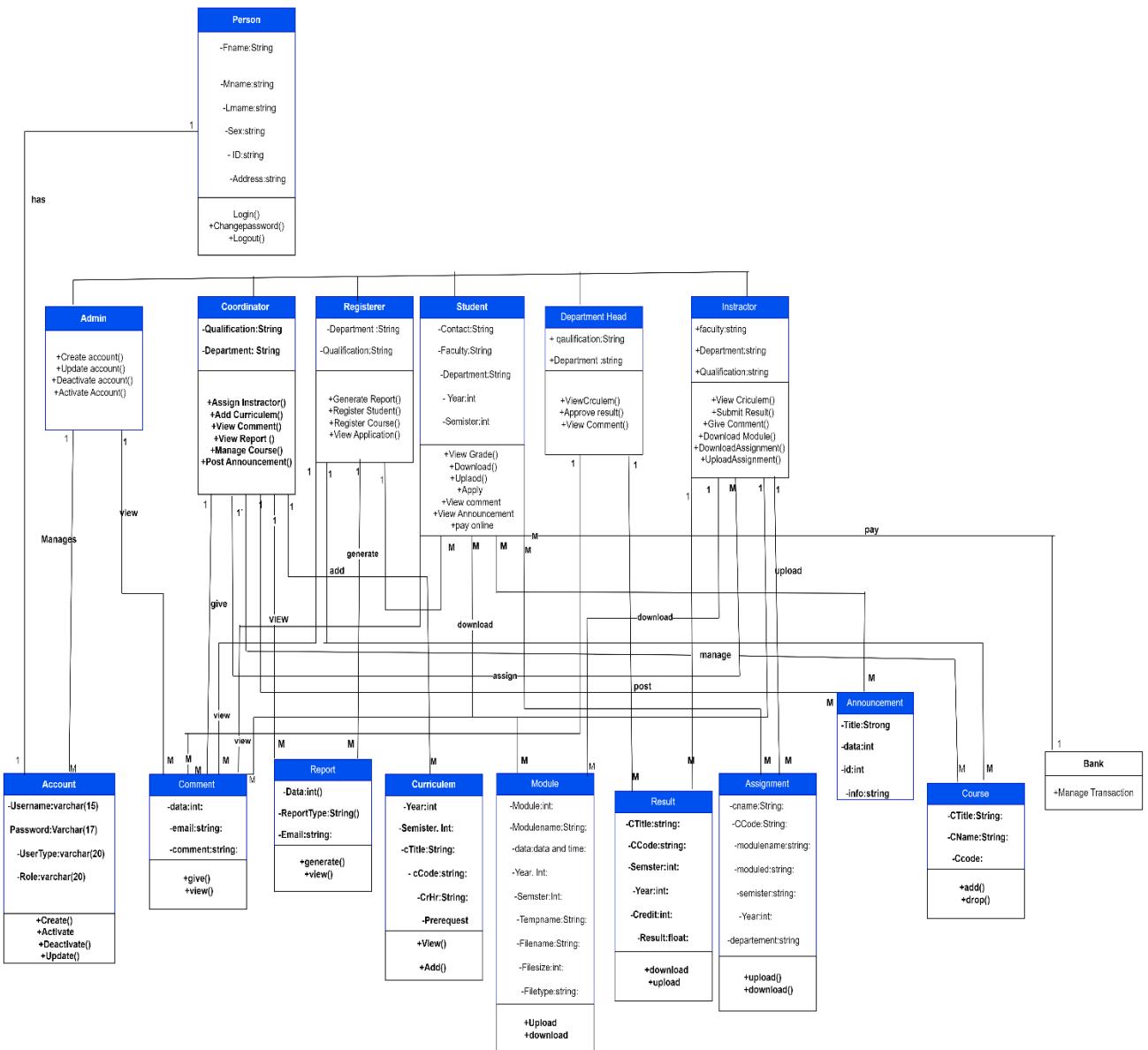


Figure 4. 25: Class diagram

5.4. Persistent model

Persistent data management deals with how the system is going to handle the actual data need to be stored on the database of the system. The purpose of persistence modelling is which objects in the system design are required to be stored persistently.

Clearly, in a database driven application like this one, almost all system interactions have deal with persistent data. In order to store information persistently we map objects into tables and the attributes of the object into fields to the specific table based on the objects found on the system.

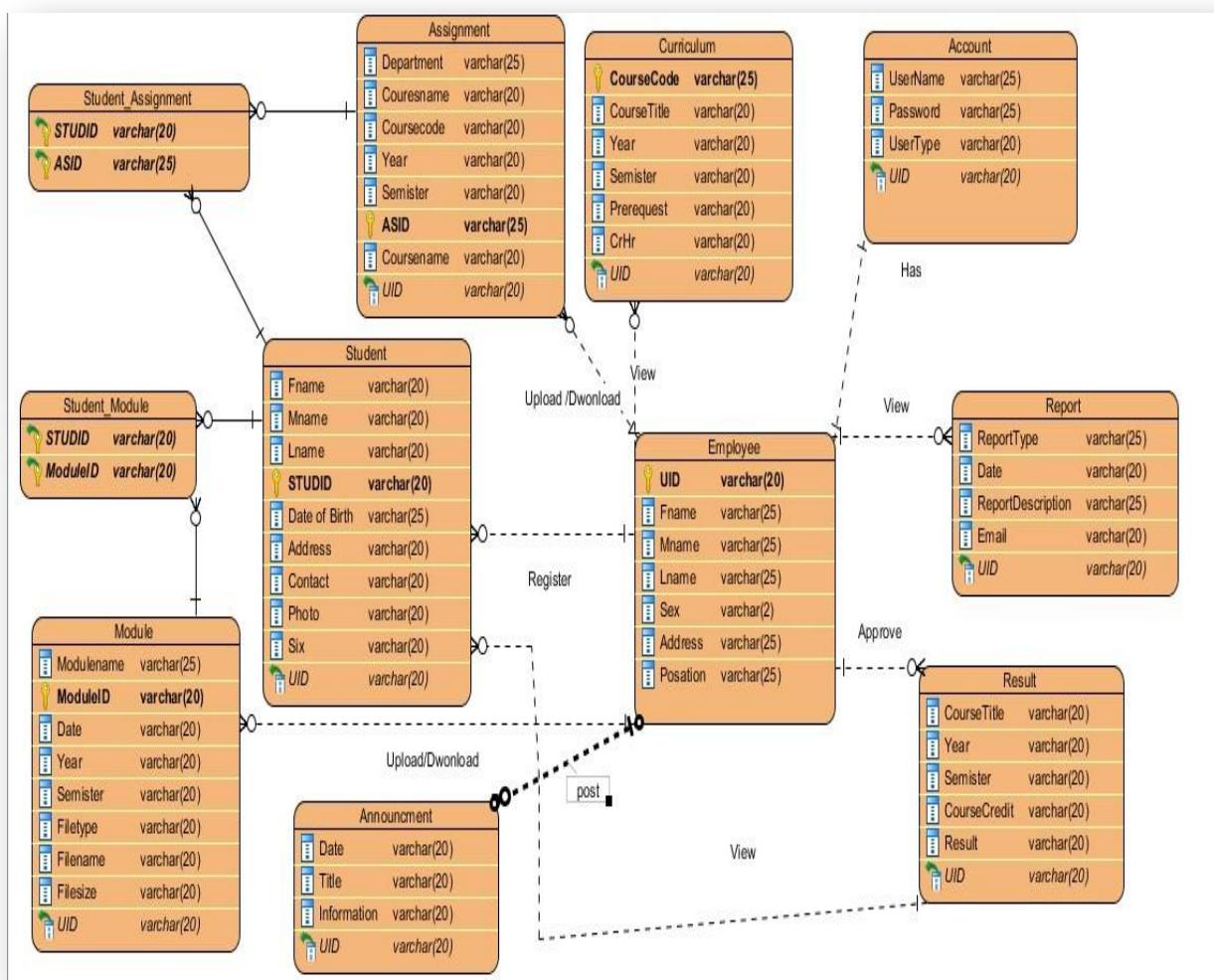


Figure 4. 26: 4 Persistent model

5.3.6 Access Control and Security

Access control and security ensure that only authorized users can access specific functionalities and data within the Web-Based Continuing and Distance Education Management System. This section describes the user model, access privileges, authentication mechanisms, encryption, and key management.

User Model and Access Privileges:

| Actor | Objects/Modules | Operations |
|----------------------|--|--|
| Student | User Management, Course, Grade, Material, Communication | Register, Enroll, View Grades, Download Materials, Receive Notifications |
| Instructor | User Management, Course, Grade, Material, Communication | Manage Courses, Enter Grades, Upload Materials, Post Announcements |
| Administrator | User Management, Course, Grade, Manage Users, Approve Material, Communication, Reporting | Enrollments, Manage Courses and Grades, Generate Reports |

Access Control:

Student: Limited to course browsing, enrollment, and grade viewing.

Instructor: Manages courses, grades, and materials.

Administrator: Full control over all system functions, including user and course management.

Security Mechanisms: Authentication Multi-factor authentication (MFA) with OTP and secure password policies. Additional Security Practices:

Audit Trails: All actions logged for security monitoring.

Regular Audits: Penetration testing and security reviews

5.4 Packages

In this section, we describe the decomposition of the **Web-Based Continuing and Distance Education Management System** subsystems into **packages**.

Each package groups related functionalities and code components. We also define the dependencies between packages and their expected usage in the system.

Divided into the following packages, each with specific responsibilities:

User Management Package: Manages user registration, authentication, and roles.

Depends on the **Database Package** for user data.

Course Management Package: Handles course creation, enrollment, and updates.

Depends on **User Management** for role verification and **Database** for course data.

Grade Management Package: Manages grade input, updates, and viewing.

Relies on Course Management for course details and **Database** for grade storage.

Material Distribution Package: Facilitates uploading, storing, and accessing course materials. Depends on Course Management and Database.

Communication Package: Sends notifications, announcements, and messages. Relies on User Management and Database.

Reporting Package: Generates academic and administrative reports. Aggregates data from other packages via Database.

Core Dependency: The Database Package supports all other packages for data storage and retrieval.

UML Package Diagram: Centralize the Database Package and connect it to all others.

Show dependencies like User Management → Database, Course Management → User Management, etc.

5.5 Algorithm Design

1. Student Registration

Task: Registers a new student.

Pseudocode:

IF all fields are filled AND email is valid THEN

 Save details in database

 RETURN "Registration Successful"

ELSE

 RETURN "Error: Missing/Invalid Information"

2. Login Authentication

Task: Validates user credentials.

Pseudocode:

IF username EXISTS AND password matches THEN

 RETURN "Access Granted"

ELSE

 RETURN "Access Denied"

3. Enroll in Course

Task: Adds a student to a course.

Pseudocode:

IF course EXISTS AND student NOT enrolled THEN

 Save enrollment in database

 RETURN "Enrollment Successful"

ELSE

 RETURN "Error: Already Enrolled or Course Not Found"

4. Upload Learning Material

Task: Allows instructors to upload course materials.

Pseudocode:

IF instructor authorized for course THEN

 Save material in database

 RETURN "Upload Successful"

ELSE

 RETURN "Unauthorized Access"

5. Generate Report

Task: Compiles a student's performance data.

Pseudocode:

Retrieve student details, courses, and grades

Compile into a report

RETURN performanceReport

6.5. User Interface Design

User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate that Automation of Project on web based continuing and Distance Education Management System for wolkite university.

The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals.

Home page User interface of the system (UI-1)

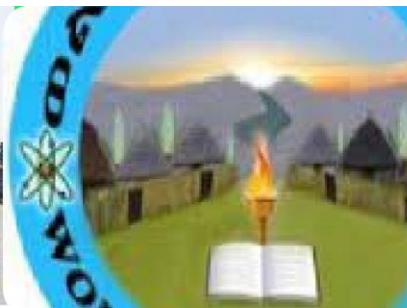
[home](#)[about us](#)[Apply to register](#)[Department](#)[Contact us](#)[Feed back](#)[Help](#)[Gallery](#)[Events](#)

Figure 4. 27: Home Page User Interface

Login page Interface of the system (UI-2)



Figure 4. 28: login User Interface

References

- [1] D. Keegan, Foundations of Distance Education (3rd ed.). Routledge., 1996.
- [2] W. University:, "About Wolkite University," [Online]. Available: <https://www.wku.edu.et..>
- [3] M. G. & K. Moore, Distance Education: A Systems View of Online Learning (3rd ed.), Belmont: Cengage Learning, 2011.
- [4] J. J. Garrett, The Elements of User Experience: User-Centered Design for the Web, Berkeley, 2010.
- [5] W. (. Stallings, Computer Security: Principles and Practice, Boston: MA: Pearson., 2017.
- [6] G. R. J. & J. I. Booch, The Unified Modeling Language User Guide., Addison-Wesley.: Reading, MA, 1999.
- [7] M. Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.), Addison-Wesley.: Boston, MA: , 2003.
- [8] R. S. Pressman, Software Engineering: A Practitioner's Approach (8th ed.), New York,: NY: McGraw-Hill., 2014.
- [9] J. J. I. & B. Rumbaugh, The Unified Modeling Language Reference Manual (2nd ed.), Reading, MA: Addison-Wesley.: Reading, MA: , 2004.

Thank You