SECJ2203: Software Engineering

System Documentation (SD)

FYP Management System

2 July 2023

Faculty of Computing

Prepared by: Software Geeks

NO.	NAME	MATRIC NUMBER
1	Jeliza Justine A/P Sebastin (L)	A21EC0034
2	Nur Immal Hayati binti Hasmi Anuar	A21EC0111
3	Ain Batrisyia Binti Norazlan	A21EC0009
4	Nur Aisyah Fatihah Binti Mohamed Rozi	A21EC0107

Revision Page

a. Overview

This current version of System Documentation contains all materials related to this project. This version includes a Requirement Traceability Matrix (RTM) and test cases. The RTM helps track and manage the relationship between requirements and other project artifacts, ensuring that all requirements are addressed and properly implemented throughout the development process. It establishes traceable links between user requirements, system requirements, design documents, test cases, and defects, allowing stakeholders to monitor the progress of requirements from origin to implementation and testing phases. The test cases provide detailed conditions, inputs, and expected outcomes to assess the functionality and behavior of specific features, modules, or the entire system. They serve as specifications for the testing process and are organized based on the package, related use case, and sequence diagram.

b. Target Audience

This documentation targeted all the relevant stakeholders.

i. Students of Data Engineering Course (SECPH)
ii. Coordinator from the Faculty of Computing
iii. University Coach
iv. Industry Coach
v. Faculty of Computing Management Department

c. Project Team Members

List of the team member, roles, and the status for each assigned task for this SD version.

Member Name	Role	Task	Status
Nur Immal Hayati	Active Member	- TC001: Test <User Module> Subsystem - TC002: Test<Materials Module> Subsystem	Completed
Ain Batrisyia	Active Member	- TC004: Test<Submission> Subsystem - TC005: Test<View and Feedback> Subsystem	Completed
Jeliza Justine	Project Leader	- TC006: Test<Plagiarism Checker> Subsystem	Completed
Nur Aisyah Fatihah	Active Member	- TC007: Test<Grading Management> Subsystem	Completed

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Nur Immal Hayati binti Hasmi Anuar	Completed Chapters 1 to 5	20/5/2023
2.0	Ain Batrisyia binti Norazlan	Completed Chapters 6 to 7	12/6/2023
3.0	Jeliza Justine A/P Sebastin	Completed Chapter 8	2/7/2023

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this System Documentation (SD) is to give a detailed overview of the system development process, which includes the System Requirements Specification (SRS), System Design Document (SDD), and System Testing Document (STD). The SD serves as a reference for individuals involved in system development, such as developers, testers, project managers, and clients. It provides a clear understanding of the system requirements, design, and testing processes, ensuring all stakeholders are on the same page and working towards common objectives. Through its structured and comprehensive documentation, the SD contributes to the success of the system development process by minimizing errors, improving efficiency, and enhancing overall project outcomes.

1.2 Scope

This project's scope is to grab more FYP-i/PSM-i involved in the industry-integrated project using FYP Management System as a platform to complete their project-based learning activities. The user of this system will be fourth-year Data Engineering students in the Faculty of Computing, an industry coach, a university coach, and the examiners and coordinators. This FYP system allows students to submit their proposals to be reviewed and evaluated by the instructors assigned to them. Generally, the system should provide all necessary information and guidance for users to complete their work. Moreover, it should contain multiple functions for different types of users.

The scope of the FYP Management System web-based software includes the following:

- The software architecture, components, and modules would be designed, considering factors like compatibility with different browsers and operating systems.
- The design would also incorporate best practices in software engineering, including coding standards, testing procedures, and quality assurance processes.
- The software product will be developed following the waterfall approach, where coding is done based on the finalized design. The coding would adhere to established coding standards and best practices.

- Ongoing support, bug fixes, and updates would be provided to ensure that the software continues to meet the specified requirements and perform as expected.
- The system will be tested using a combination of manual and automated testing methods to ensure that it meets all specified requirements and performs as expected.

1.3 Definitions, Acronyms, and Abbreviations

Definitions of all terms, acronyms, and abbreviations used are to be defined here.

Term	Definition
Integrated Project Proposal System	The system is used to manage the project proposal of SECPH students.
FYP-i/PSM-i	Final Year Projects with Industry
SD	System Documentation
SDD	Software Design Description
SRS	Software Requirement Specification
Coordinators	The person who manages the whole system and has access to every platform.

Table 1.0: Definitions of all terms, acronyms, and abbreviations

1.4 References

- Thakur, D. (2013). *Software Design Documentation (SDD). Computer Notes.* https://ecomputernotes.com/software-engineering/softwaredesigndocumentation

- Rome,P. (2023), *What are Non Functional Requirements - With Examples* Retrieved May 18, 2023, from https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples

- Rajkumar. (2023), *What are Quality Attributes in Software Architecture* Retrieved May 18, 2023, from https://www.softwaretestingmaterial.com/quality-attributes-in-software-architecture/

1.5 Overview

This System Documentation contains all materials related to building this Industry Integrated Project Proposal System which is also known as Final Year Project Management System. This document has 3 main sections which are System Requirements Specification (SRS), System Design Document(SDD), and System Testing Document(SDS).

This System Documentation is organized as follows:

Purpose and Scope System - The objective of creating the system and all the tools used in implementing the system.

Specific Requirements - The user characteristics, and system features that have been interpreted in the form of a use case diagram and come along with an activity diagram and system sequence diagram.

Software System Attributes, Performance, and Other Requirements - Comprises all the relevant attributes for the system and performance requirements.

System Design Constraints - List of all the design constraints that need to pay more attention in developing the system.

This document is used to give all the stakeholders involved a clear view of this project. It provides all the information about the software development, design, and testing procedures throughout the project lifecycle.

2. Specific Requirements

2.1 **User characteristics**

The "FYP Management Systems" software will be used by five types of users: data engineering students, coordinators, examiners, university coaches, and industrial coaches.

2.1.1 Data engineering students

- Data engineering students who will use the software are expected to have basic computer skills, including familiarity with e-learning web-based applications.
- They need to submit their draft of the FYPi project proposal step by step before making the final submission to be graded by the examiners.
- Some FYP projects may be plagiarism from other resources so a plagiarism checker is a must for students to verify the authenticity of their project
- Sometimes students may not be aware if the project was successfully submitted or not, but the system will notify the students if they successfully submitted so they can be certain that their work has been received.

2.1.2 Coordinators

- The coordinator must undergo comprehensive training to understand the system's functions.
- All mandatory requirement forms such as rubric, e-evaluation, no-plagiarism endorsement form, and comment form for students should be uploaded in a single location with file tagging provided by the system.
- The system will also provide features for the coordinator to create a submission, and change the time and due date of an existing submission platform.
- Coordinator releases the official students' grades after verifying marks given by examiners, industrial coaches as well as university coaches.

2.1.3 Examiners

- Examiners are responsible for evaluating papers and are expected to uphold fairness, as the system will automatically reject submissions from students that are submitted after the deadline.
- Examiners are permitted to download and evaluate files uploaded by students, as the system grants them access to these files for grading purposes.
- The system requires examiners to submit an approval form for each student, indicating that all mandatory requirements have been fulfilled by the student, before allowing the examiners to allocate marks.

2.1.4 Industrial Coach

- Capable of delivering an accurate draft to the student within the anticipated time frame.
- System provides details and contact information of the industrial coach.

2.1.5 Universities Coach

- The system will provide a list of students assigned to a particular supervisor and set a limit on the number of students that the supervisor can access.
- The system sends reminders to the university coach to review the progress of their supervisees.

2.2 **System Features**

The Final Year Project Proposal Management system is a software that operates on a browser that the students from the Data Engineering course will use as their platform for updating the progress of their final year project proposal. This system is proposed as the need to enhance the process and management system efficiently compared to the previous system. In this project, a web-based software solution will be introduced to eliminate the current tedious way of managing students' final-year projects by developing a computerized system that assists in managing all the operations of the final-year project. It also provides a medium for easy communication between students and lecturers or coordinators during all the phases.

The system features are illustrated in Figure 1.1 below. The detailed description of each module and function is tabulated in Table 1.1.

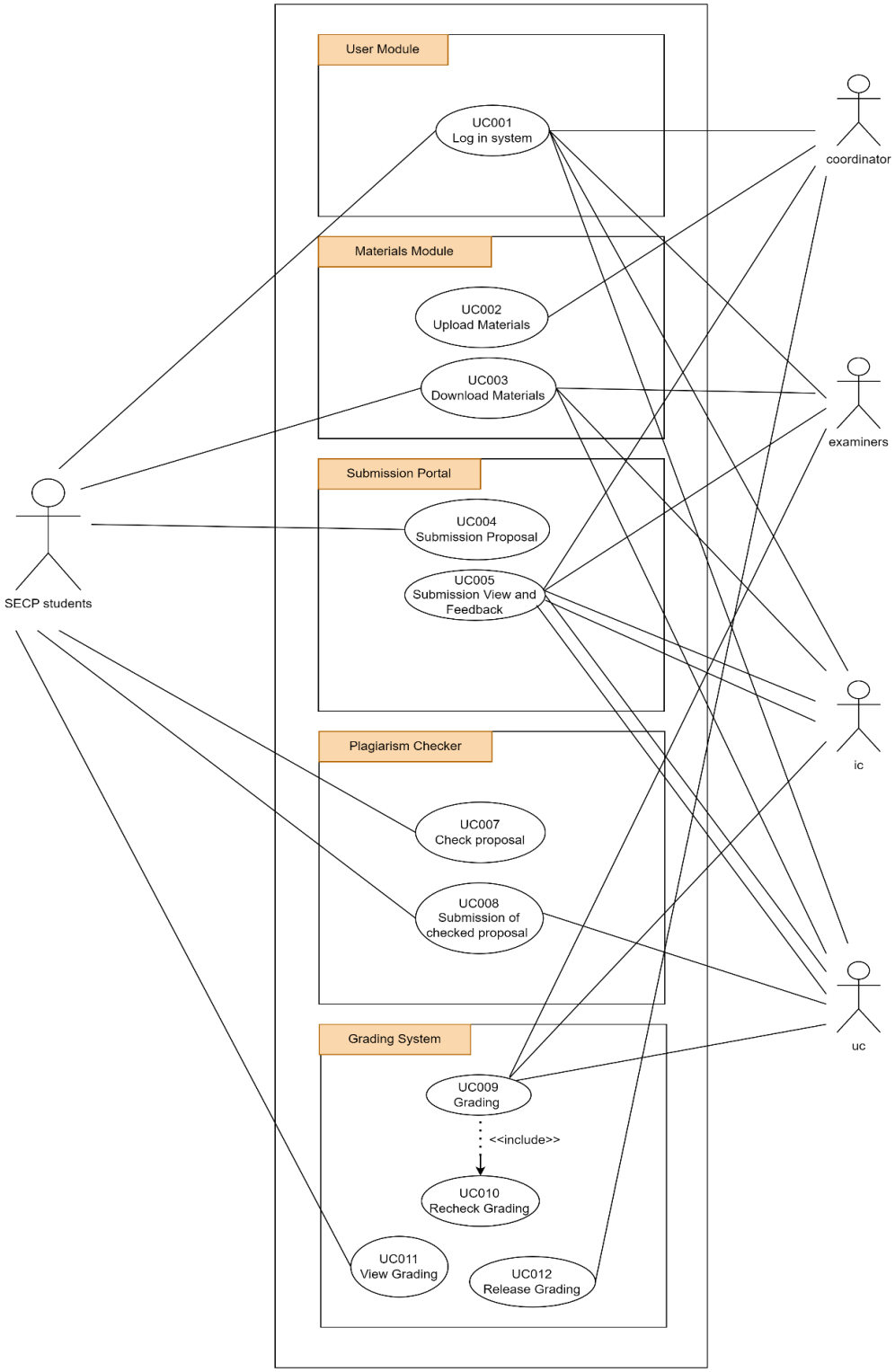


Figure 1.0: Use Case Diagram for The Final Year Project Proposal Management System

Table 1.1: Description of Module and Functions for Final Year Project Proposal Management System

Module	**Function**	**Description**
User Module	UC001 – Log in	This use case allows users to log in to the system.
Materials Module	UC002 - Upload Materials	This use case allows users to upload all the important materials for the proposal.
	UC003 - Download Materials	This use case allows users to download all the important materials for the proposal.
Submission Portal	UC004 - Submission Proposal	This use case allows users to submit the document for their project.
	UC005 - View and Feedback	This use case allows the user to view the submission and add feedback to give comments.
Plagiarism Checker	UC006 - Check Plagiarism	This use case allows users to check the percentage of similarity with the resources they used to refer.
	UC007 - Submission of Checked Proposal	This use case allows users to submit the checked proposal.
Grading System	UC008 - Grading	This use case allows examiners,ic and uc to do grading.
	UC009-Recheck Grading	This use case allows users to recheck the grading after they submit the student grade.
	UC010 - Release Grading	This use case allows coordinators to release the grade of the project for student view.
	UC011 - View Grading	This use case allows students to view the grade of their project submission.

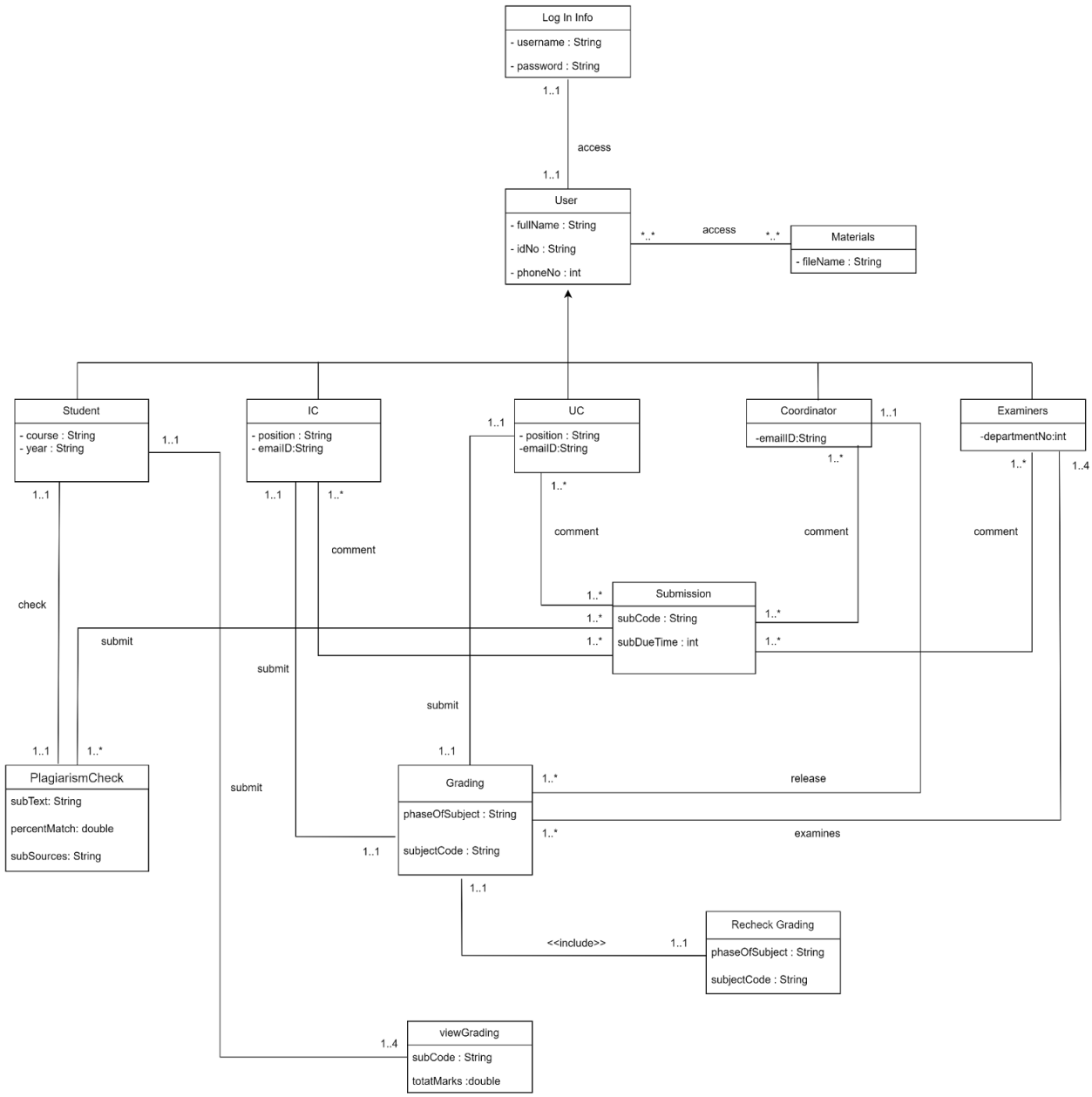


Figure 1.2: Domain Model Diagram for The Final Year Project Proposal Management System

2.2.1 UC001: Use Case - User Module

Use case: User Module
ID: UC0001
Actors: SECP Student, Coordinator, UC, IC, Examiner
Preconditions: Has stable internet connection to access the platform
Flow of events: 1. The use case starts when the user key in their username. 2. Then, the user will key in their authorized password. 3. The user then clicks the button Login. 4. The system will validate the username and password entered whether it is valid or not. 5. If the account is invalid, Exception 1 will follow. 6. If the account is valid, the user will be directed to the home page of the system. 7. The use case ends.
Exception flow: 1. Invalid password or username 1.1. Display error message

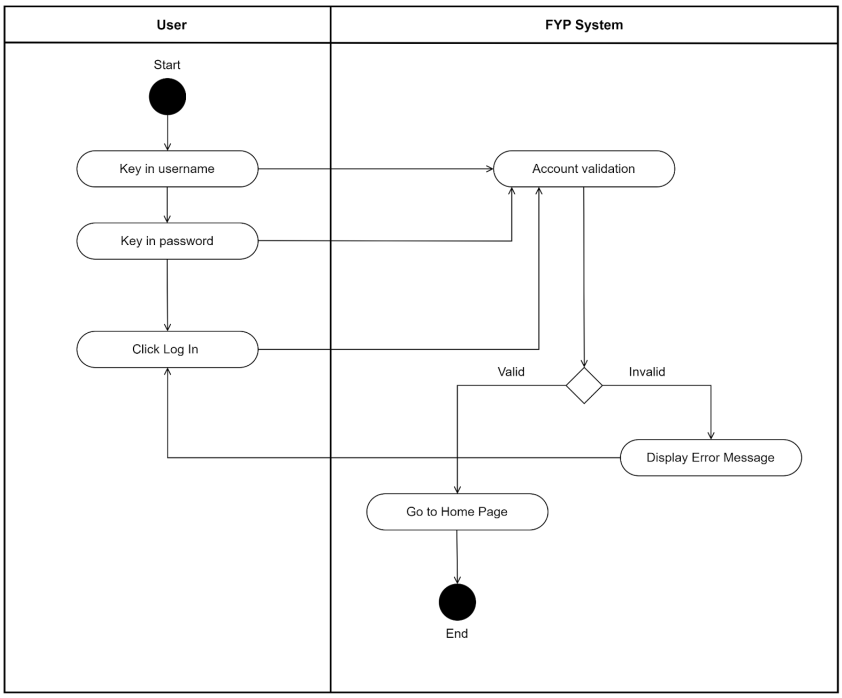


Figure 1.3: Activity Diagram for User Modules

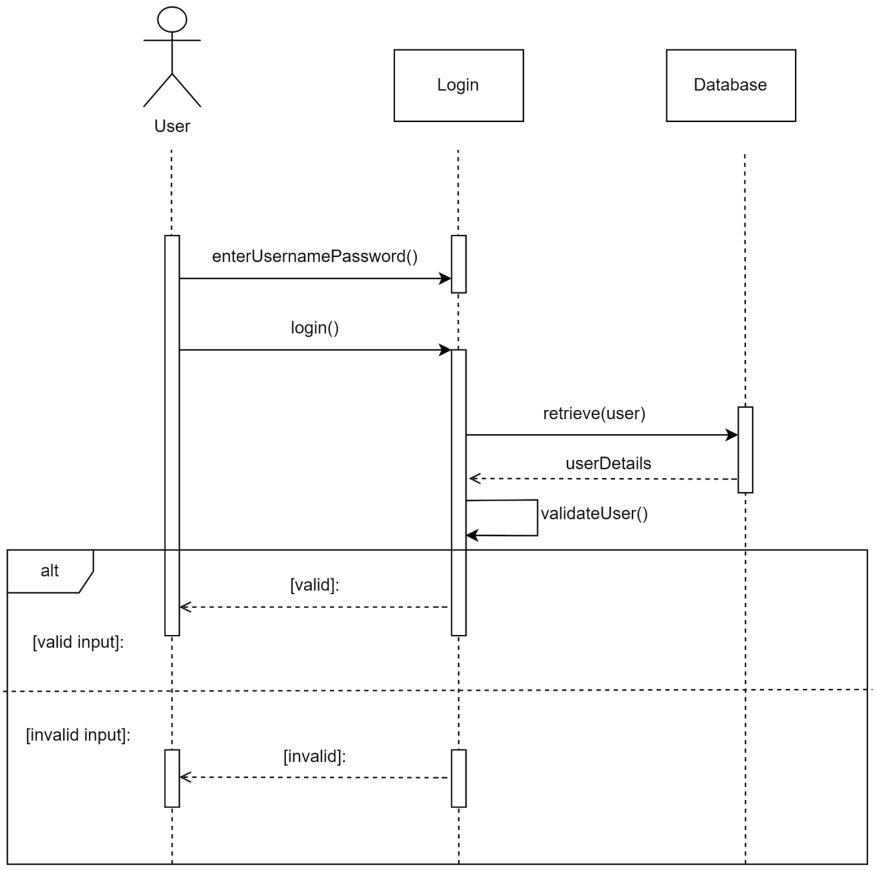


Figure 1.4: Sequence Diagram for User Modules

2.2.2 UC002 & UC003 : Use Case - Upload Materials & Download Materials

Use case: Upload Materials
ID: UC002
Actors: Coordinator
Preconditions: Users already login into the system.
Flow of events: 1. The use case starts when the user clicks the Materials button. 2. Then, the system will display the list of materials. 3. Coordinator will click the Add File button. 4. The system will display an upload file window pane. 5. Coordinator will then choose the file from the browser to upload into the system. 6. Then, the coordinator will click the Upload button. 7. The system will display a Upload Successful message. 8. The use case ends.
Postconditions:

Use case: Download Materials
ID: UC003
Actors: University Coach, Industry Coach, Examiner, SECPH Student
Preconditions: Users already log in into the system.
Flow of events: 1. The use case starts when the user clicks the Materials button. 2. Then, the system will display the list of materials. 3. The user can choose a file to download. 4. Then the user will download the file. 5. The system will display a Download Successful message. 6. The use case ends.
Postconditions:

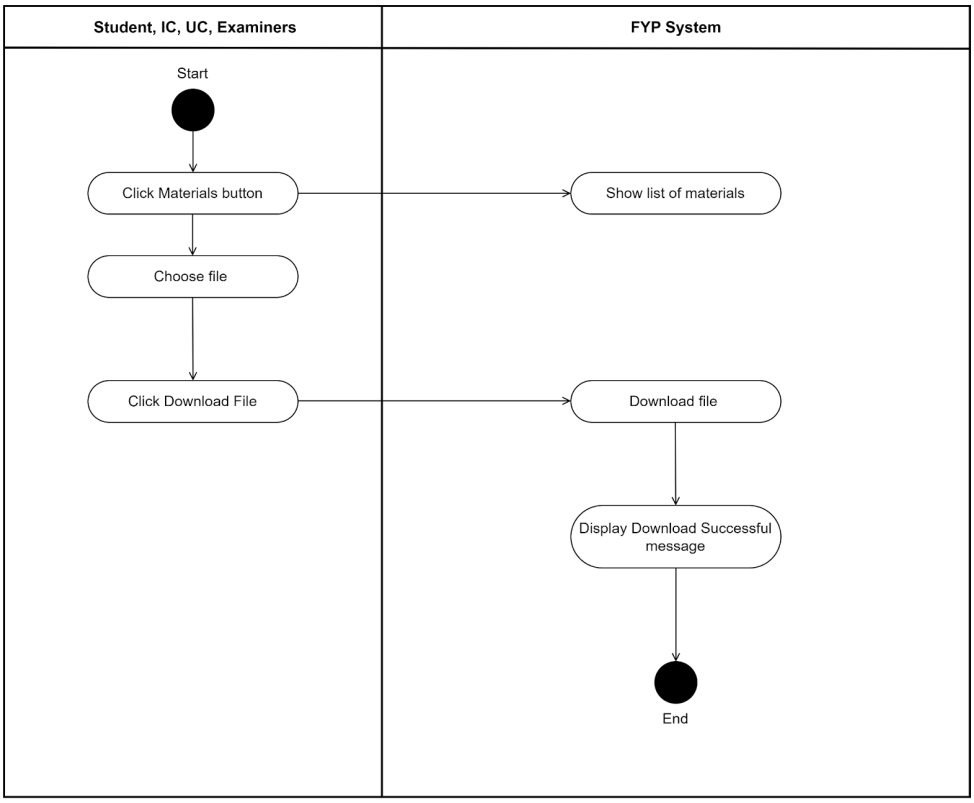


Figure 1.5: Activity Diagram for Materials Modules (UC003:Download Materials)

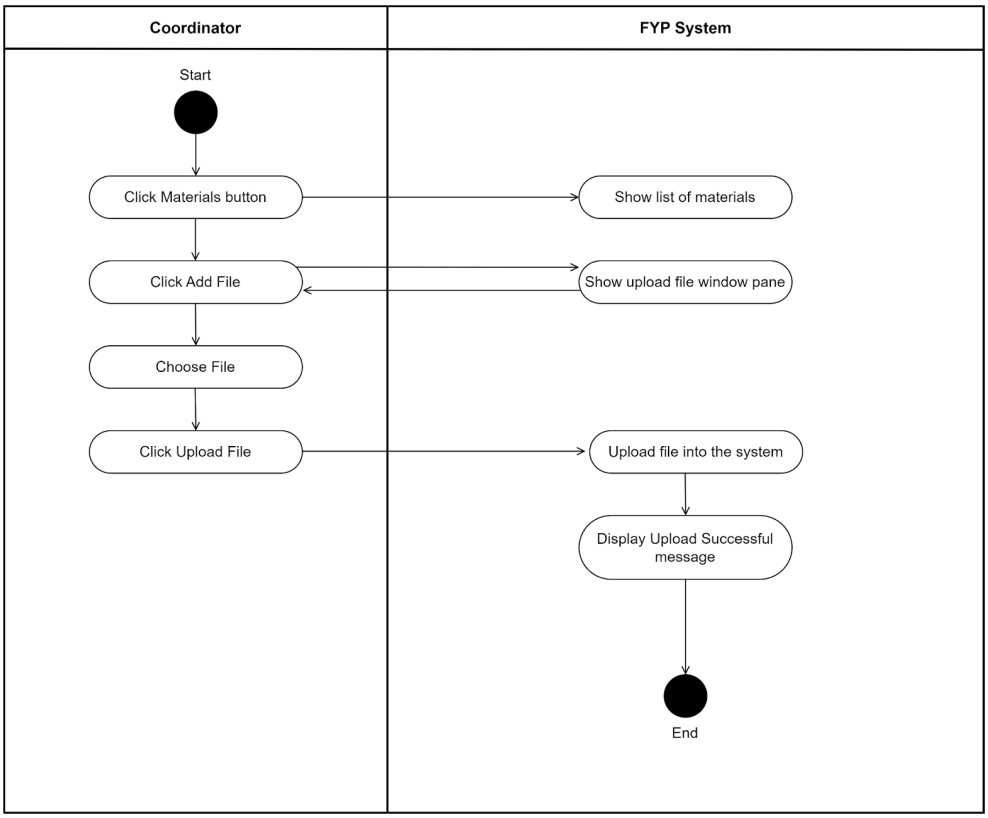


Figure 1.6:Activity Diagram for Materials Modules (UC002:Upload Materials)

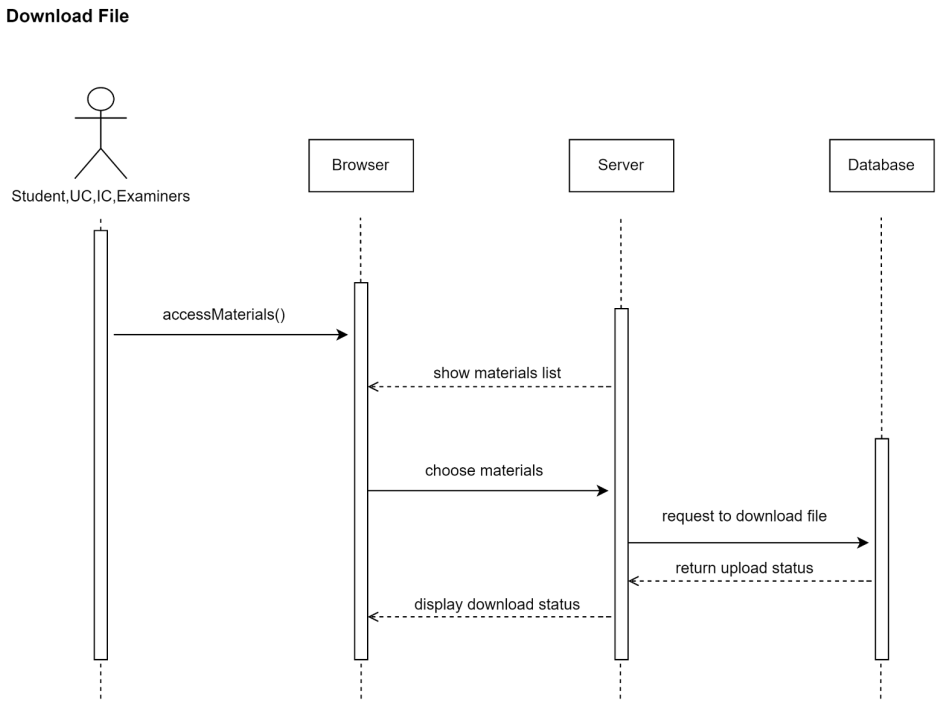


Figure 1.7: Sequence Diagram for Materials Modules (UC003:Download Materials)

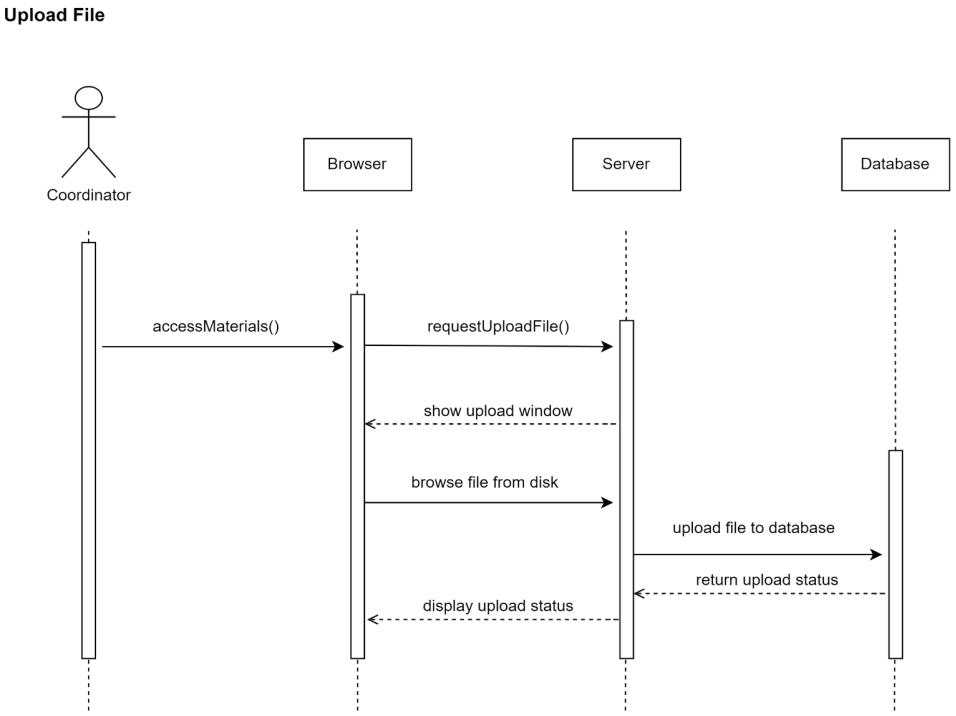


Figure 1.8: Sequence Diagram for Materials Modules (UC002:Upload Materials)

2.2.3 UC004: Use Case <Submission>

Use Case: <Submission>
ID: UC004
Actors: Student
Preconditions: 1. Has active internet connection. 2. Must submit in PDF format for report and pptx format for presentation slide.
Flow of events: 1. The use case starts when the user taps on the submission page. 2. System displays link submission for each chapter and clicks the chosen chapter. 3. System displays the submission page containing the submission form. 4. The user clicks on the ‘add submission’ button. 5. Choose documents from their folder. 6. System validates the document. If the document is invalid, Exception 1 is followed. 7. Click the submit button. 8. System display submission successful. 9. The use case ends.
Postconditions:

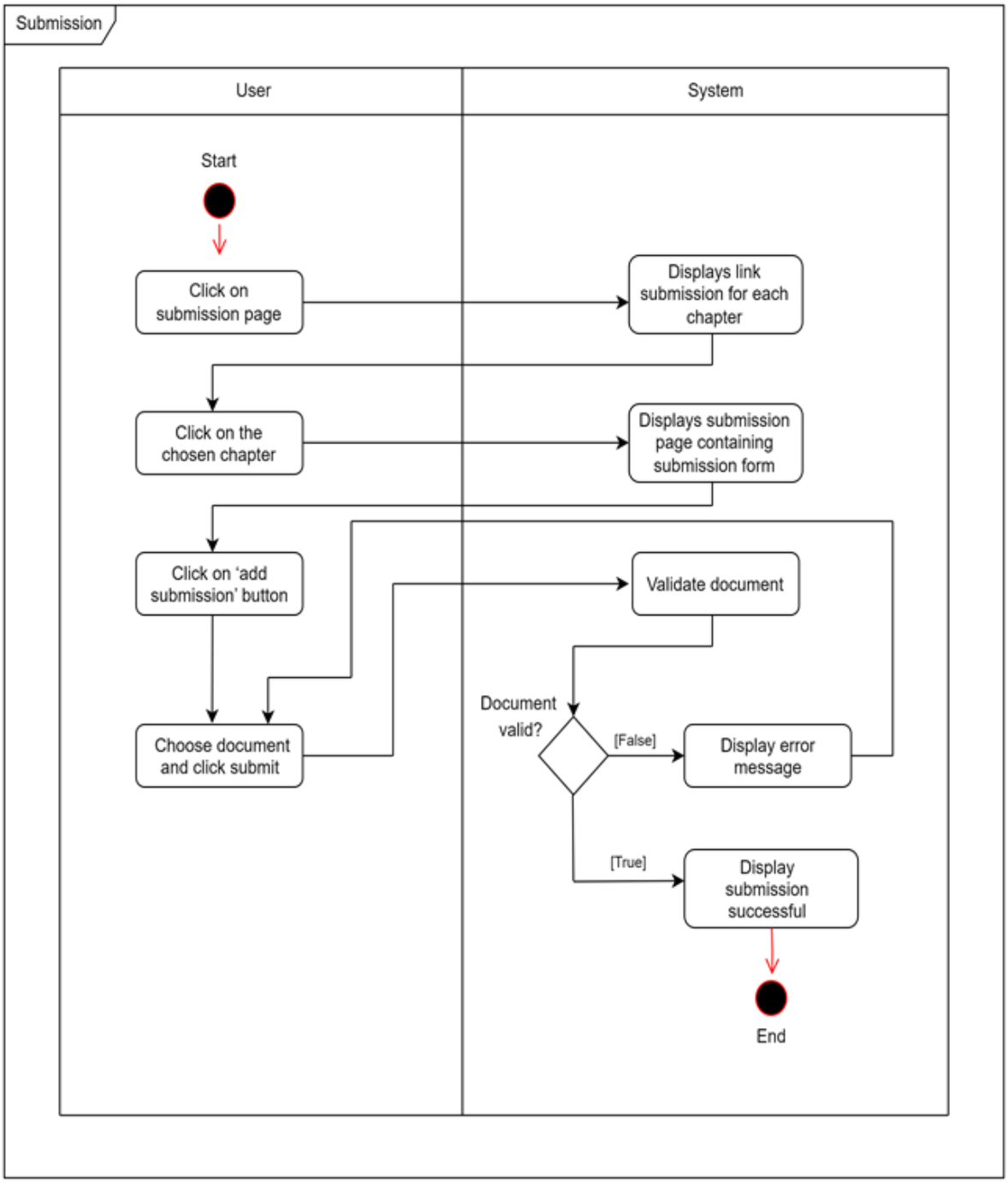


Figure 1.9: Activity Diagram for Submission Portal

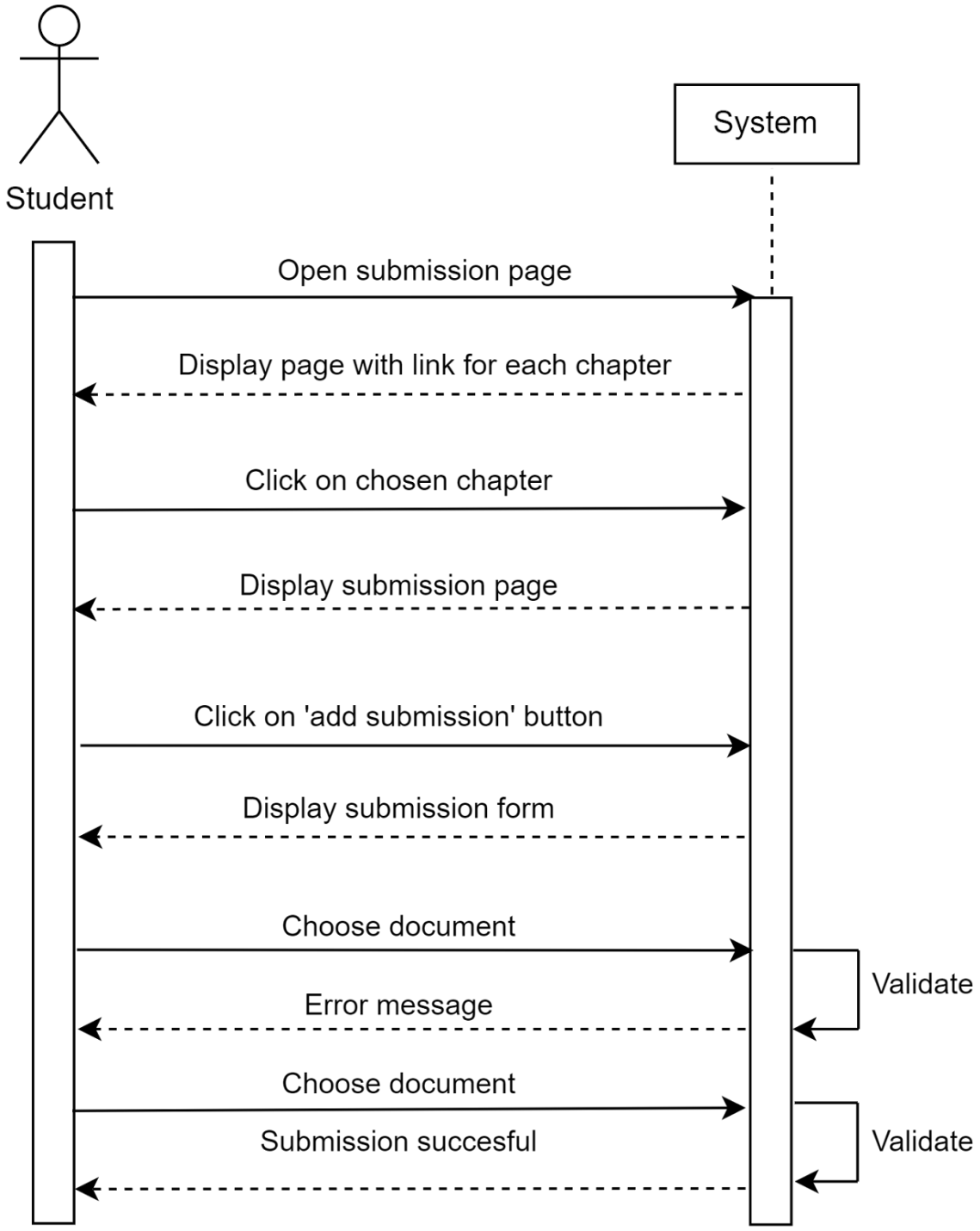


Figure 2.0: Sequence Diagram for Submission Portal

2.2.4 UC005: Use Case <View and Feedback>

Use Case: <View and Feedback>
ID: UC005
Actors: University Coach, Industry Coach, Examiner
Preconditions: 1. Has active internet connection. 2. There is a document to be reviewed.
Flow of events: 1. The use case starts when the user taps on the submission page. 2. System displays a list of student names assigned to each user. 3. System displays four gray buttons beside each student's name with each button representing each chapter. 4. If the button display green 4.1 Students have submitted the assignment. 5. Else 5.1 There has been no submission from the student yet. 6. The user clicks on the student name that they want to review. 7. System displays the overall submission from the chosen student. 8. The user clicks on the ‘add feedback’ button. 9. Write a feedback comment to help students improve their work 10. Submit the feedback. 11. System display feedback has been added successfully. 12. The use case ends.
Postconditions:

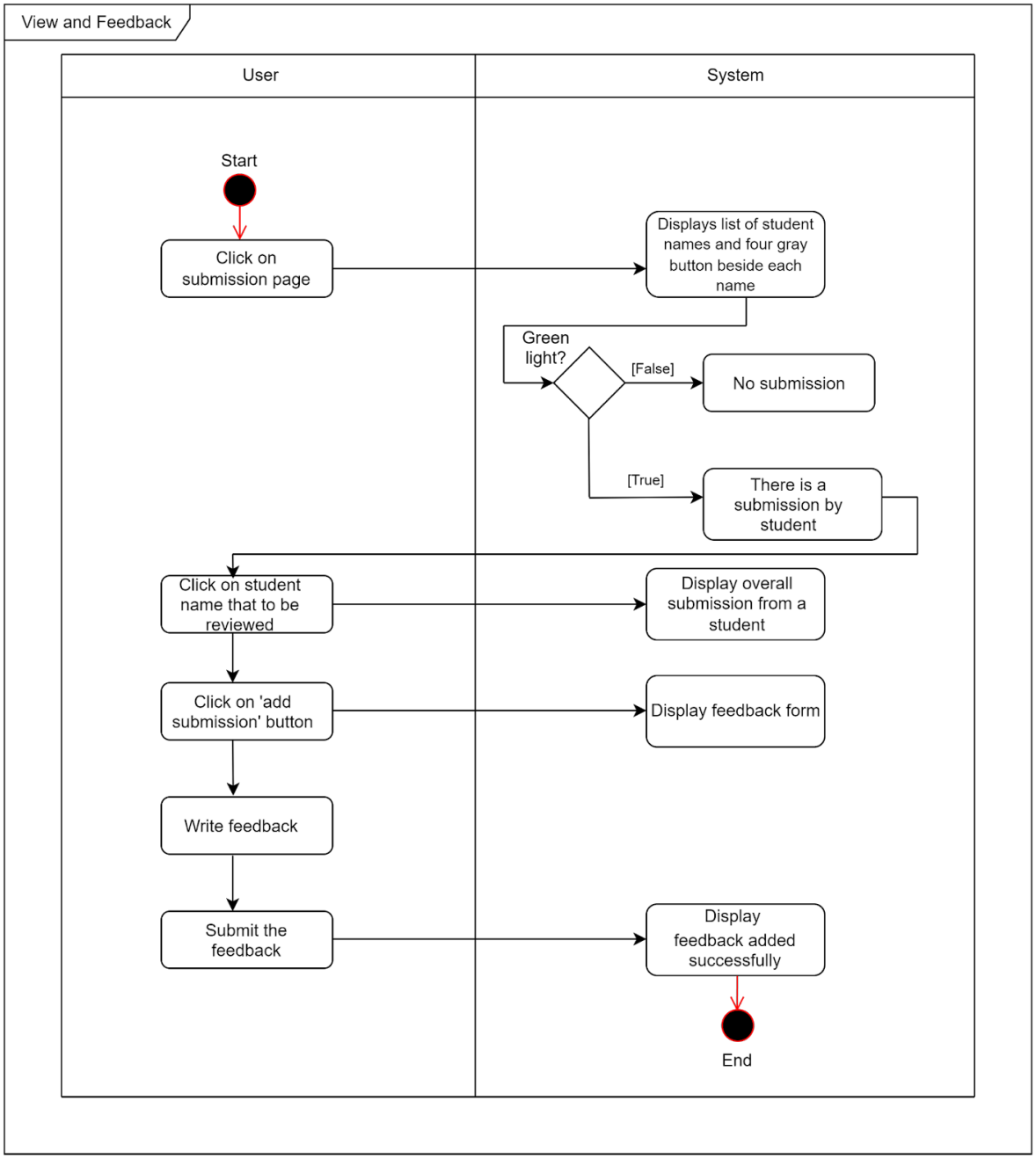


Figure 2.1: Activity Diagram for View and Feedback

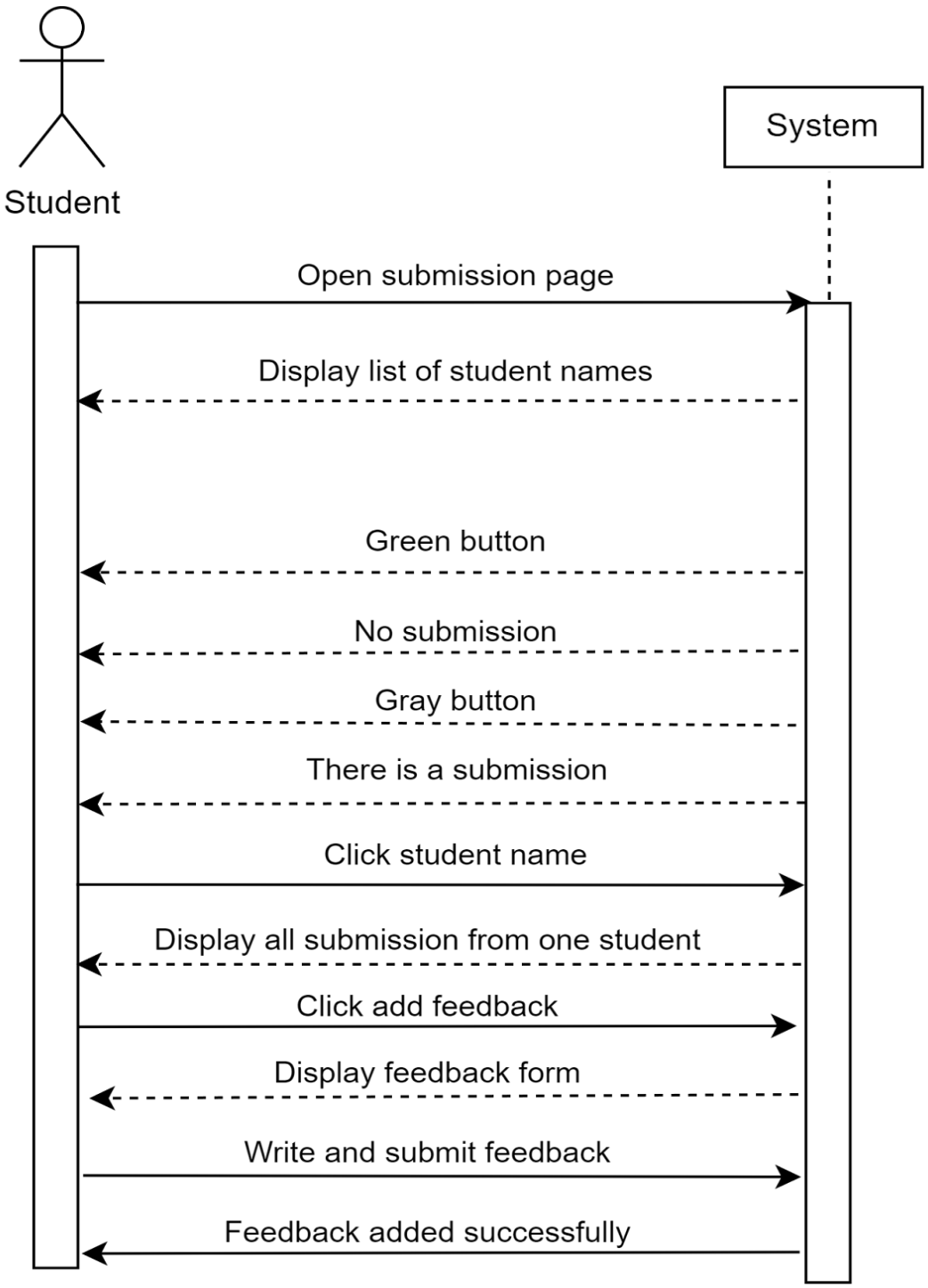


Figure 2.2: Sequence Diagram for View and Feedback

2.2.5 UC006 & UC007: Use Case <Plagiarism Checker> & <Submission of checked proposal>

Use Case: <Plagiarism Checker> & <Submission of checked proposal>
ID: UC006 & UC007
Actors: UTM Student
Preconditions: 1. The student is logged on to the system.
Flow of events: 1. The use case starts when the user taps on “Plagiarism Checker” in the menu after logging in. 2. System displays the plagiarism checker page containing the file upload section. 3. The user uploads his/her file. 4. The user taps on “Check Plagiarism”. 5. System shows the file uploaded with the percentage of matching detected and its sources. 6. System displays the download option at the bottom center of the page. 7. If the percentage of matching is more than the percentage set by the management system then 7.1 User modify or re-do the project phase. 7.2 Check again in the plagiarism checker. 8. Else 8.1 Users download the file and proceed to the submission portal. 9. User submits the checked proposal via the submission portal.
Exception Flow: 1. Percentage of matching exceeds the fixed value 1.1 System displays a warning message.

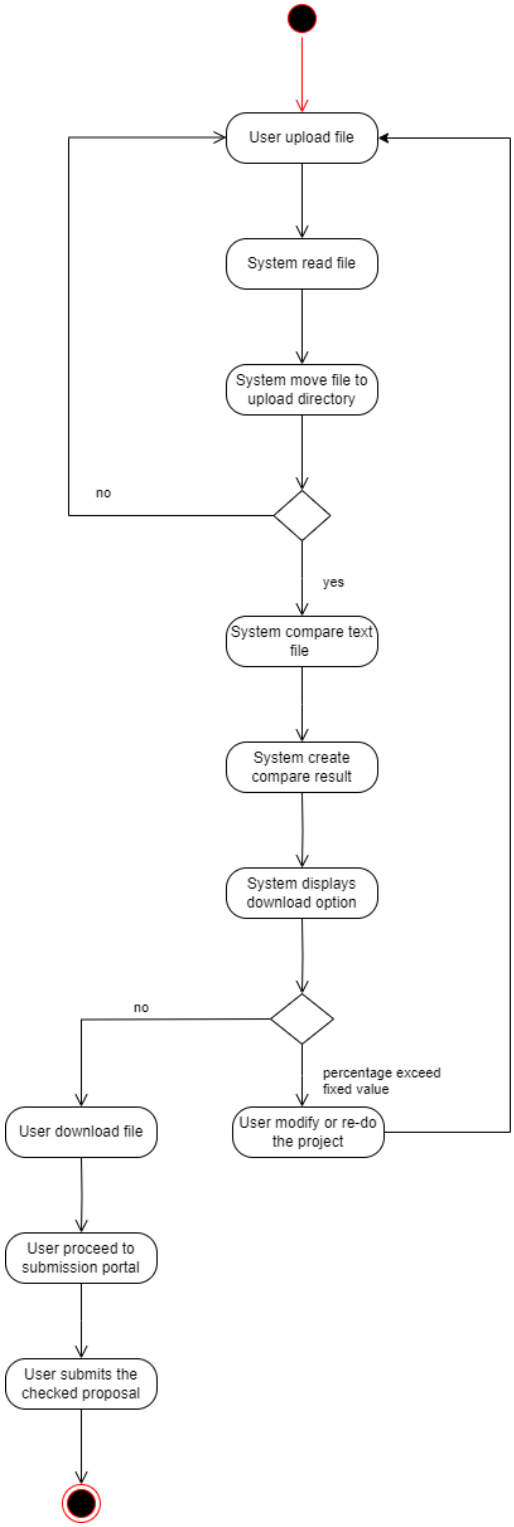


Figure 2.3: Activity Diagram for Plagiarism Checker

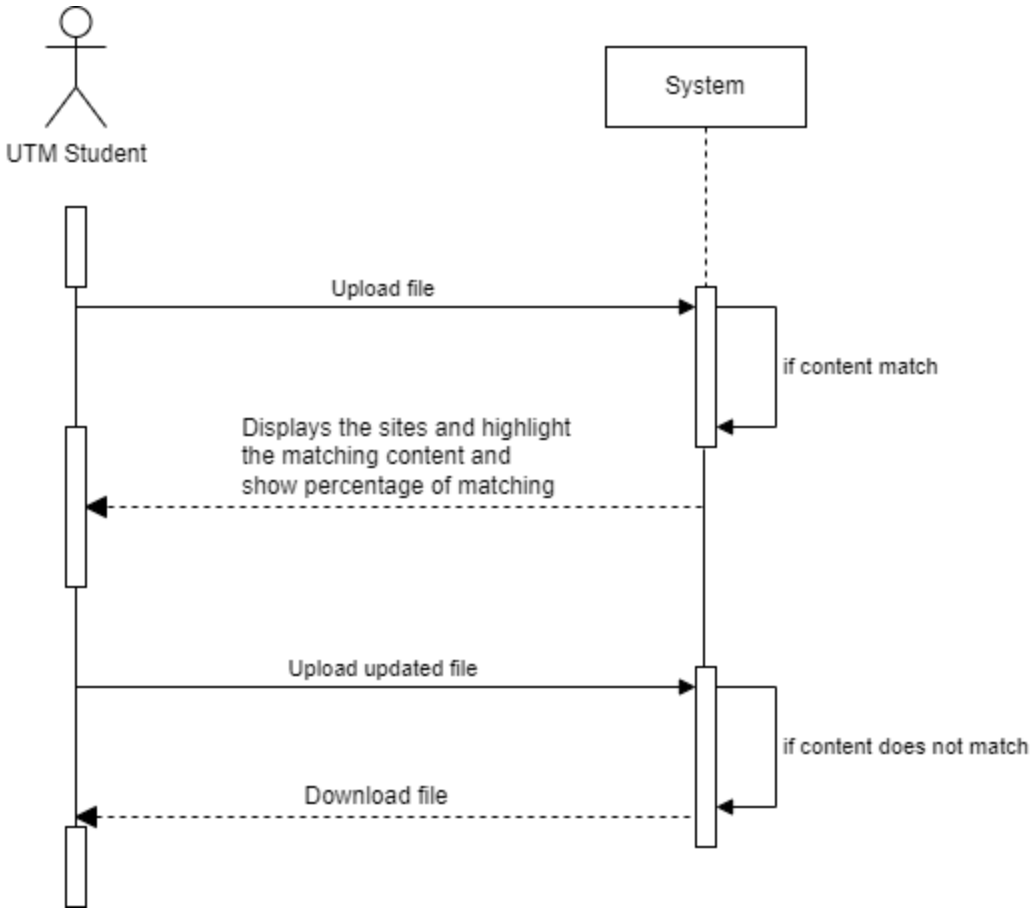


Figure 2.4: Sequence Diagram for Plagiarism Checker

2.2.6 UC008: Use Case <Grading>

Use Case: <Grading>
ID: UC008
Actors: Examiners, Industry Coach, University Coach
Preconditions: Examiners, Industry Coaches, University Coaches username and password to log into the system. A submission has been made by students.
Flow of events: 1. Examiner, Industry Coach, and University Coach login to the system. 2. If the user is examiner 2.1 While the student has few submission topics 2.1.1 Examiner open submission file 2.1.2 Examiner does the grading based on each chapter 2.1.3 If the examiner is done with the grading <<RecheckGrading>> 3. Else 3.1 Industry and University coach does the grading based on the student's skills 3.2 If the University and Industry Coach has done <<RecheckGrading>> 4. Examiner, UC, and IC submit the grading to system 5. The system records and saves the current marks given.
Postconditions: The coordinator receives the grades and needs their verification

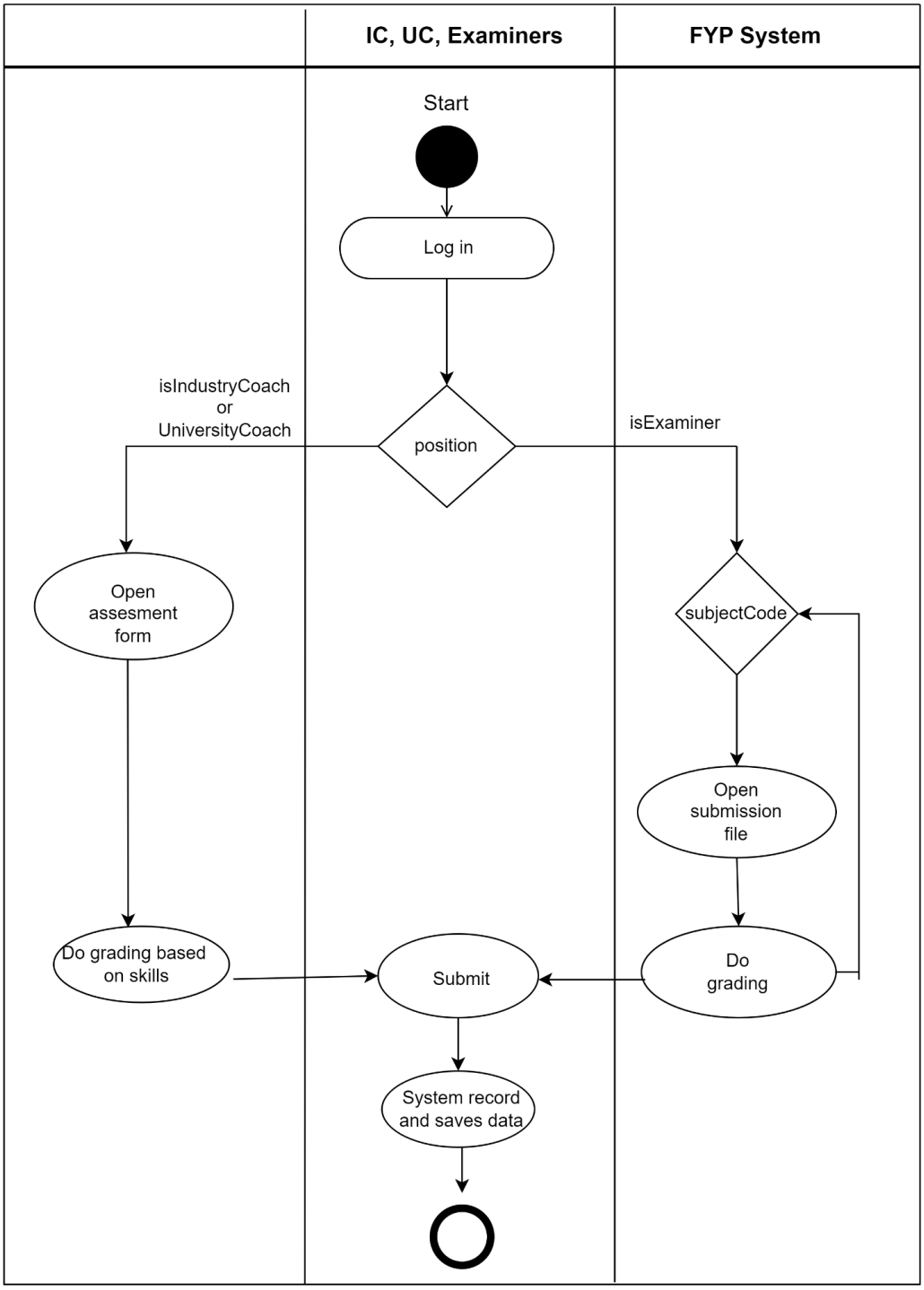


Figure 2.5: Activity Diagram for <Grading>

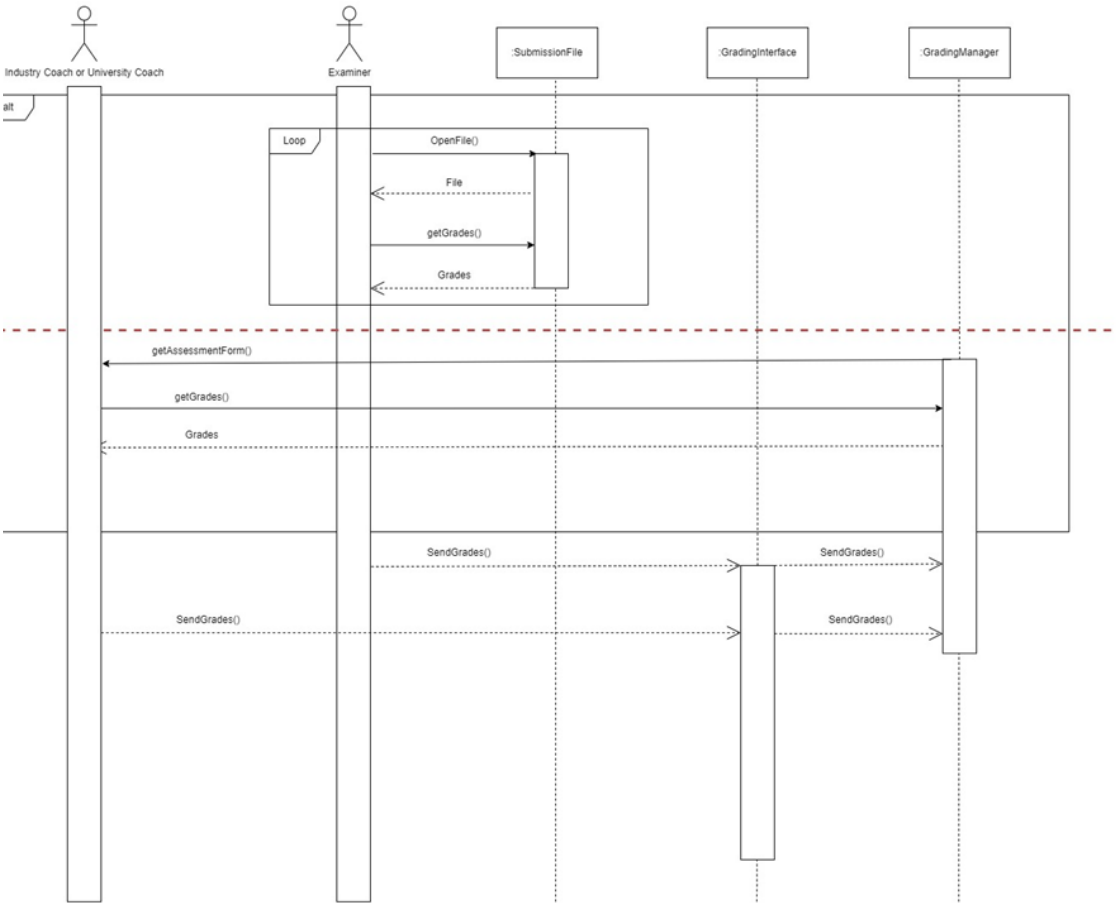


Figure 2.9: Sequence Diagram for <Grading>

2.2.7 UC009: Use Case <Recheck Grading>

Use case: Recheck Grading
ID: UC009
Actors: Examiner, Industry Coach, University Coach
Includes: UC006<< Grading>>
Preconditions: A valid Examiner's, Industry Coach's, University Coach's username and password to log into the system Students have made a submission. At least one grading has occurred.
Flow of events: 1. Examiner, Industry Coach, and University Coach login to the system. 2. If the user is Examiner 2.1 while the student has 4 courses 2.2 Examiner clicks on the recheck button. 2.3 System provides the graded course paper. 2.4 Examiner does the grading based on each course code. 3. Else 3.1 Industry Coach and University click the recheck button 3.2 System provides the graded assessment form. 3.3 Industry Coach and University Coach does the grading based on the skills of each student. 3.4 The Examiner, Industry Coach, and Examiner Coach will submit the grading in the System. 3.5 The system saves and records the grades of each student.

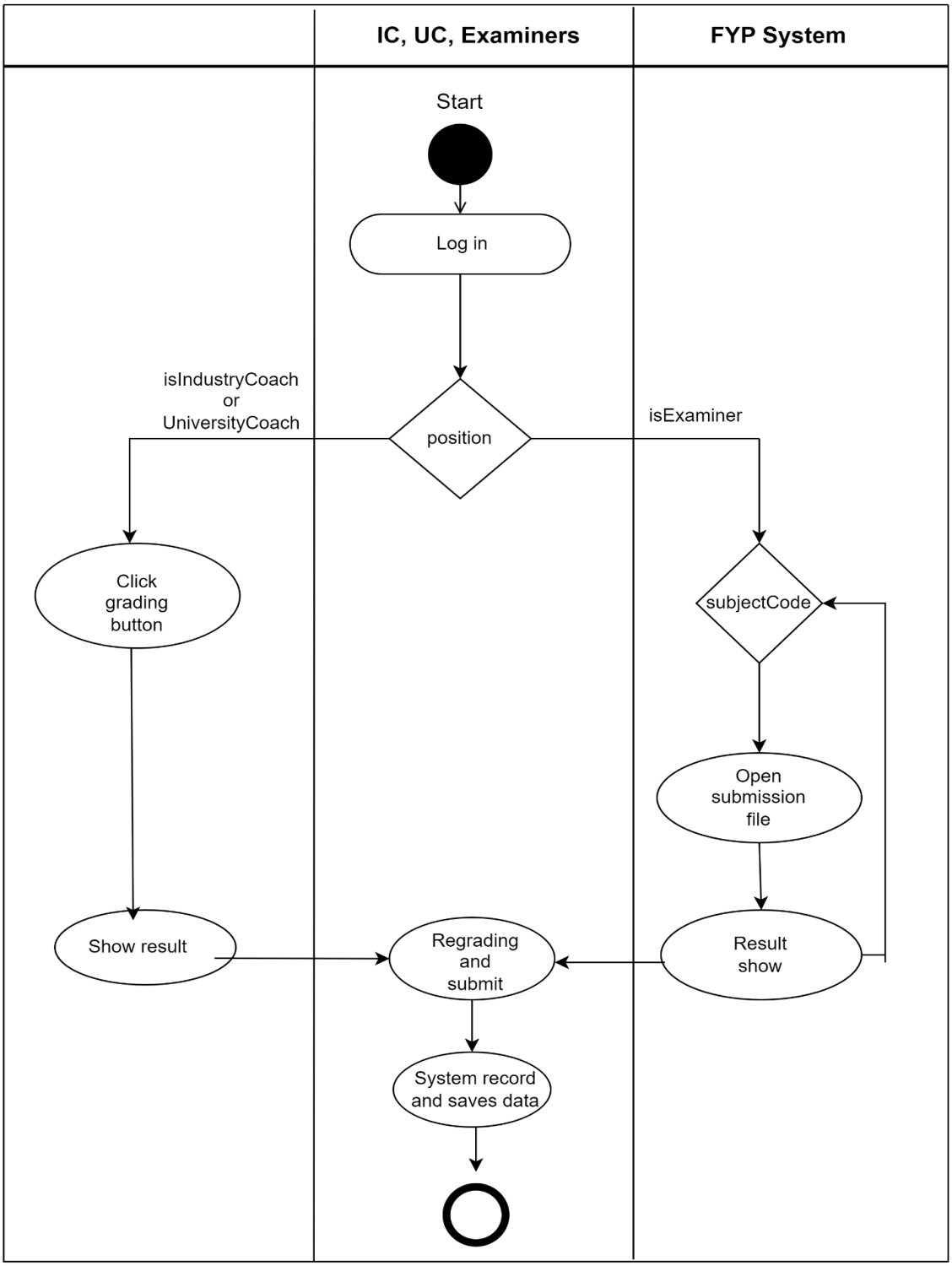


Figure 2.6: Activity Diagram for <Recheck Grading>

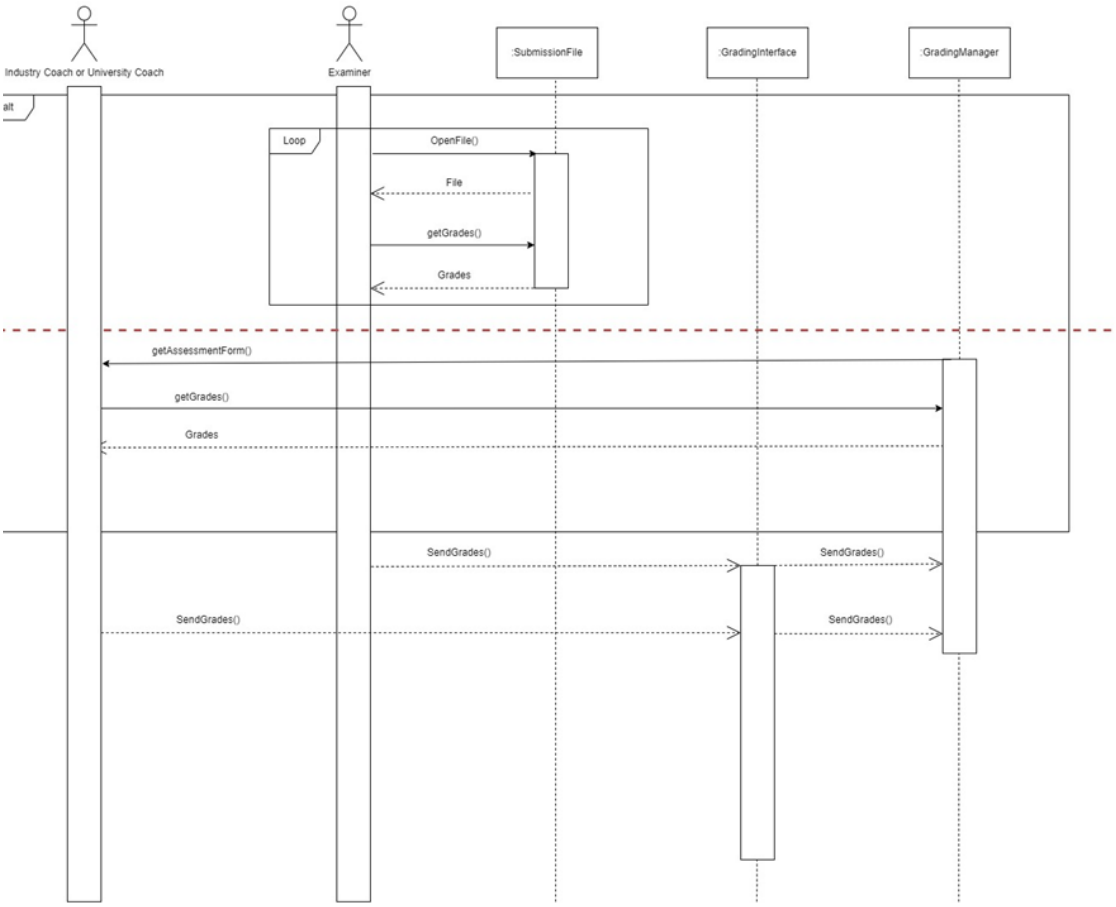
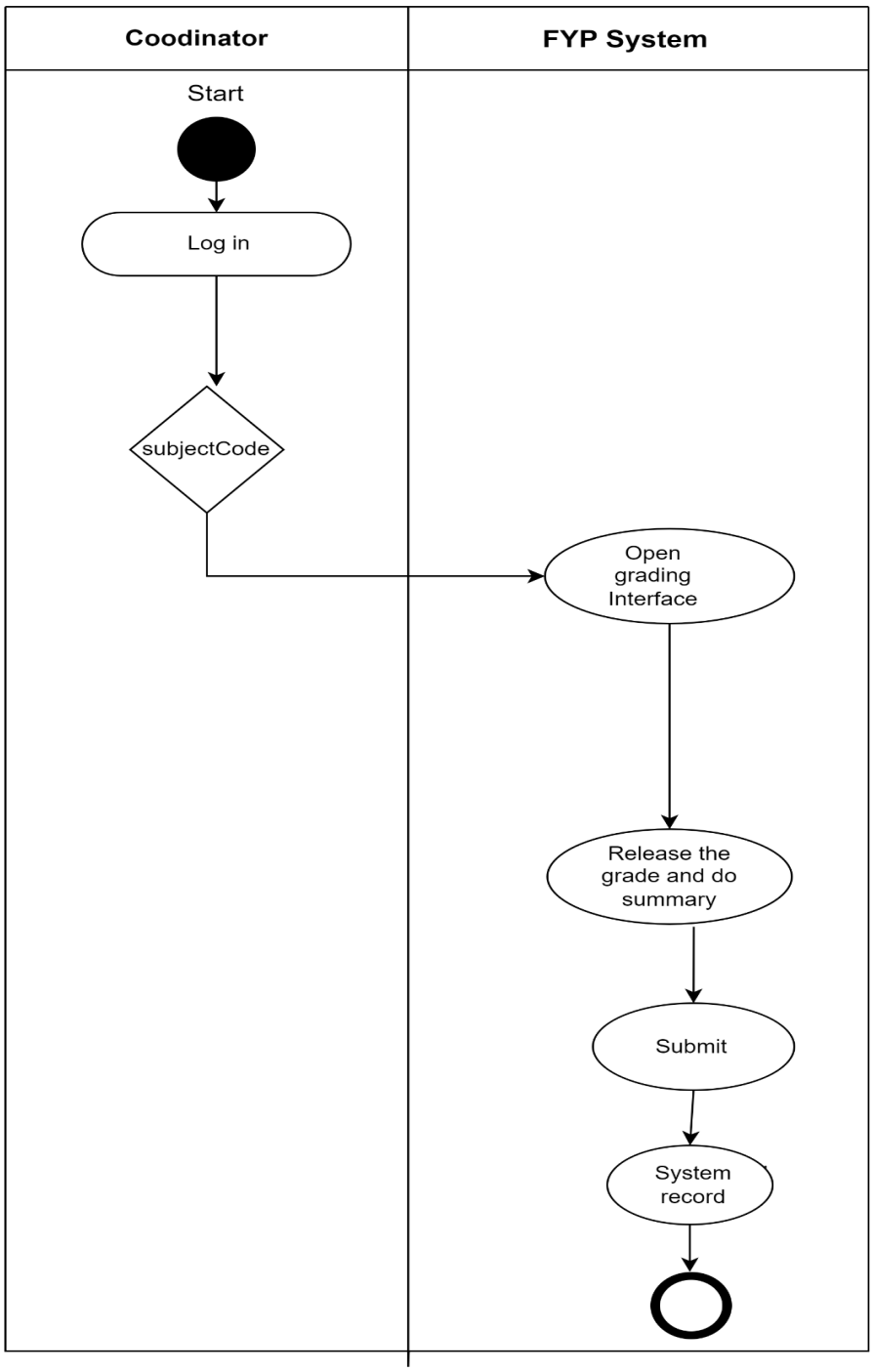


Figure 2.9: Sequence Diagram for <Recheck Grading>

2.2.8 UC011: Use Case <Release Grading>

Use case: Release Grading
ID: UC011
Actors: Coordinator
Preconditions: A legitimate Coordinator login and password, as well as grades, have been supplied. Each course has received a grade by examiners and UC and IC.
Flow of events: 1.The Coordinator login to the system. 2.The Coordinator navigates to the Grading interface. 3.The Coordinator compares the recheck grades for each course. 4.The Coordinator will compile and write a report about the grades of the students. 5.The Coordinator release their grading when clicking submit button 6.The system displays a confirmation message. 7.If the Coordinator press confirms. 7.1. The System displays the grades. 8.Else 8.1. The System returns back to the grading interface.
Postconditions: The grades are displayed with students' information.


Coodinator
FYP System
Start
Log in
subjectCode
Open grading Interface
Release the grade and do summary
Submit
System record

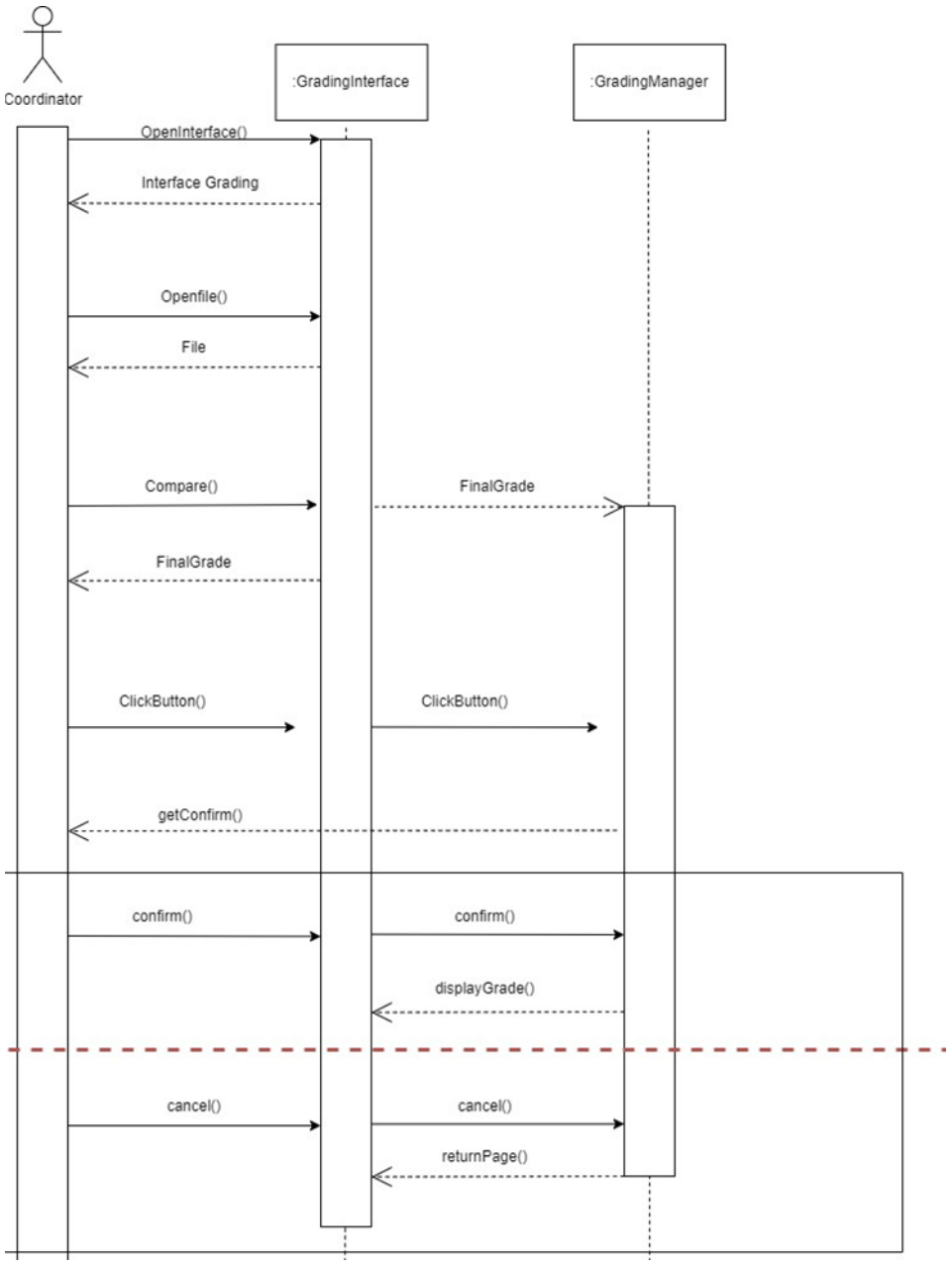


Figure 3.0: Sequence Diagram for <ReleaseGrading>Figure 2.7: Activity Diagram for <Release Grading>

2.2.9 UC010: Use Case <View Grading>

Use case: View Grading
ID: UC010
Actors: Students
Preconditions: A valid student's username and password to log into the system. Submissions have been made by the students. Grading has been made by the examiner, industry coach, and university coach.
Flow of events: 1. The Student login to the system. 2. The Student opens the Grading Interface. 3. System asks students for the course code. 4. The Student enters the code. 5. If the code entered is valid 5.1. The system will display the grade. 6. Else 6.1. The system will display an invalid code.
Postconditions: The student receives the grades.

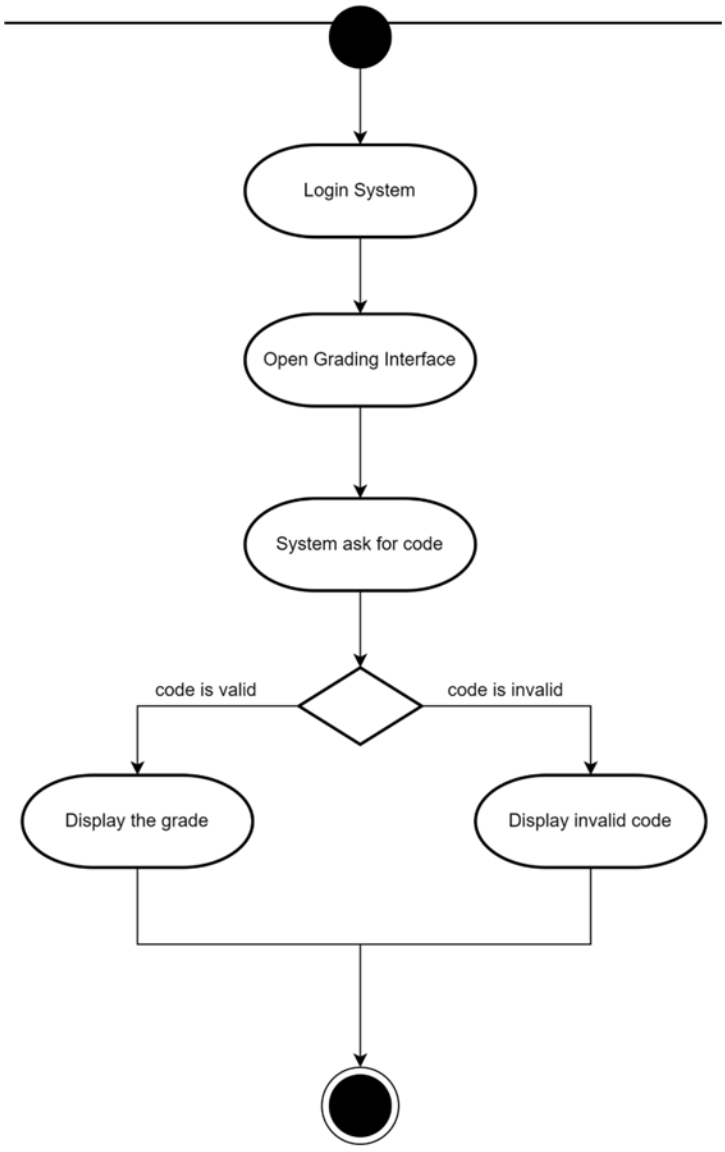


Figure 2.8: Activity Diagram for <ViewGrading>

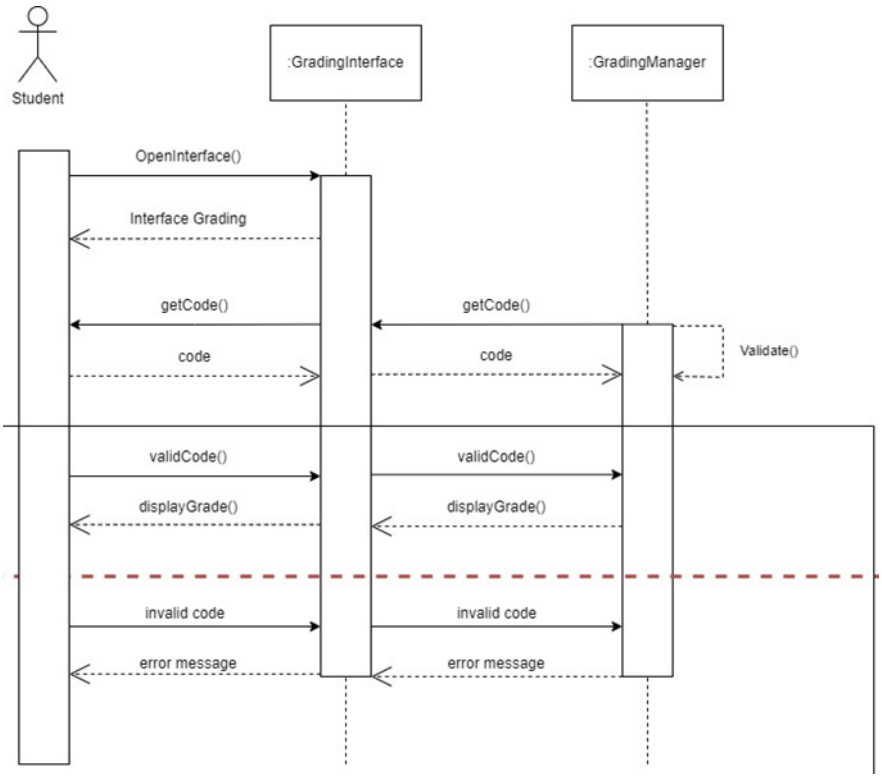


Figure 3.1: Sequence Diagram for <ViewGrading>

2.3 Software System Attributes, Performance, and Other Requirements

Non-functional requirements are the requirements that describe how the system should behave or operate, rather than what it should do. They address the characteristics or qualities of the system, such as its usability, reliability, performance, security, maintainability, and compatibility. Non-functional requirements are important to write in the SD because they help ensure that the system meets the stakeholders' needs and expectations. The ISO/IEC/IEEE 29148 standard suggests that non-functional requirements should be specified under three main categories: Software System Attributes, Performance, and Other Requirements.

When designing a final-year project management system, it's important to consider various software system attributes to ensure its effectiveness, efficiency, and usability. Examples of some key attributes of the Software System are as follows:

- **Functionality:** The system should provide essential features for managing final-year projects, such as modules to access materials, a submission portal to submit documents, a plagiarism checker, and a grading system.
- **Usability:** The system should have an intuitive and user-friendly interface, making it easy for students, coordinators, examiners, industry coaches, and university coaches to navigate and use the system effectively.
- **Reliability:** The system should be dependable, ensuring that it performs its intended functions consistently and accurately. It should have minimal downtime and handle errors gracefully. Data integrity and backup mechanisms should be in place to prevent data loss.
- **Maintainability:** The system should be designed with maintainability in mind, making it easier to modify, enhance and fix issues in the future.
- **Compatibility:** The system should be compatible with various platforms and devices, such as desktop computers, tablets, and mobile devices. It should be responsive and adapt well to different screen sizes and resolutions.

Performance requirements define the system's capability to respond to user requests and handle data in a timely manner. Performance is a critical attribute for a final-year project management system as it directly impacts user experience, productivity, and the system's ability to handle increasing workloads. these requirements include the following:

- **Response Time:** The probability of the system will return results within 1 second and it varies if the internet connection is slow.
- **Scalability:** The increase in the system's workload that the system should be able to process.
- **Capacity:** The system should handle a maximum number of users and a maximum amount of data.

- **Throughput:** The system should handle a maximum number of requests in a given period of time.

Other requirements refer to any non-functional requirements that do not fit under the categories of software system attributes or performance. These requirements include the following:

- **Security:** Security requirements are to protect sensitive project information and user data. The system should include authentication mechanisms, access controls, encryption requirements, and secure communication protocols to ensure the system's security and prevent unauthorized access or data breaches.
- **Data Management:** The system should have data backup and retention policies, data validation rules, data archiving requirements, and data privacy regulations to ensure proper management and protection of project-related data.
- **Environmental:** The system should give a positive impact on the environment. This may include reducing energy consumption, minimizing paper usage through digital workflows, or adhering to environmentally friendly practices in system design and operations.
- **Legal and Regulatory:** The system should ensure compliance with data protection and privacy laws. This includes obtaining appropriate consent for data collection and processing, implementing security measures to protect personal information, and providing users with control over their data.

2.4 Design Constraints

Design constraints refer to limitations or specifications that influence the design process of a software, product, or system. It sets boundaries and limitations within which the design must operate and guides the design process by providing a framework for making decisions.

There are lots of constraints that may need to be considered to ensure that the software is effective and appropriate for its intended use. The system will have hardware, security, compatibility, and performance constraints. Hardware constraint Security constraints must address authentication, authorization, data encryption, secure communication, and other relevant security measures. Compatibility constraints

Constraint	Description
Hardware constraints Include considerations such as available memory, processing power, storage capacity, and network connectivity.	The system is designed to accept any accepted format (.pdf or .pptx) with a maximum of 8GB file size and must have an active internet connection.
Security constraints must address authentication, authorization, data encryption, secure communication, and other relevant security measures.	The system is designed to ensure that only authorized students can access and submit documents.
	The system will implement regular data backups to prevent data loss in case of system failures, disasters, or accidental deletion. Establish reliable recovery mechanisms to restore the system and the submitted documents to a functional state.
Compatibility constraints Refer to a requirement or limitation that ensures the compatibility of a software system with other systems, components, or environments	The system will be designed to run on specific operating systems, such as Windows, macOS, Linux, or mobile platforms like Android or iOS. Also compatible with different browsers (e.g., Chrome, Firefox, Safari, Internet Explorer)
Performance constraints To ensure that the system performs optimally across different web browsers, devices, and network conditions	The system must be designed to support a maximum of 1000 concurrent users with a response time of fewer than 5 seconds across various devices, including desktops, laptops, tablets, and mobile phones

3. System Architectural Design

3.1 Architecture Pattern and Rationale

We chose the Layered Architecture as the style for our FYP Management System because it divides the system into different layers, each with its own specific tasks. The lower layers provide important services that are used by the layers above them. In our system, we have three layers:

- User Interface Layer: This layer includes the actual interface that users see and interact with. It allows users to view information and perform actions in the FYP Management System.

- Domain Layer (Business Logic): The Domain Layer contains the logic and rules that represent the core functions of our system. It handles things like managing FYP projects, students, and supervisors. It ensures that our system operates according to the business requirements.

- Data Access Layer: The Data Access Layer helps us work with the database where we store our data. It makes it easier to retrieve and save information from our entity-relational database.

We choose the Layered Architecture because it allows us to enhance our existing manual system by adding new features without disrupting the existing functionality. If we need to replace a whole layer in the future, we can do so as long as the interfaces between layers remain the same. This reduces the cost of replacing or upgrading specific components, such as the user interface. Overall, the Layered Architecture provides us with a structured and modular approach to develop our FYP Management System, ensuring that each layer focuses on its specific tasks and allowing for future growth and flexibility.

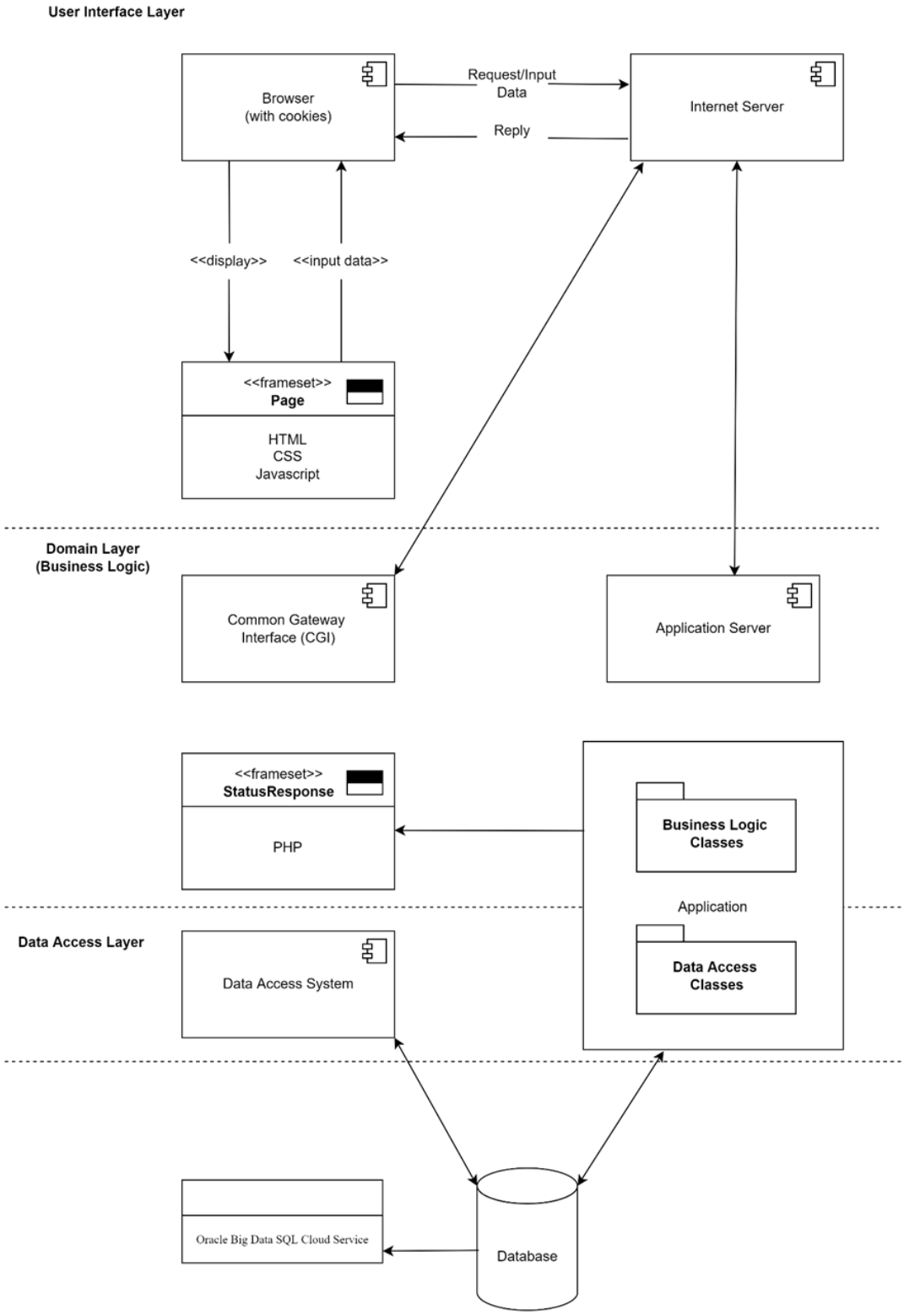


Figure 3.1: Complete Architectural Diagram of <FYP Management System>

3.2 Component Model

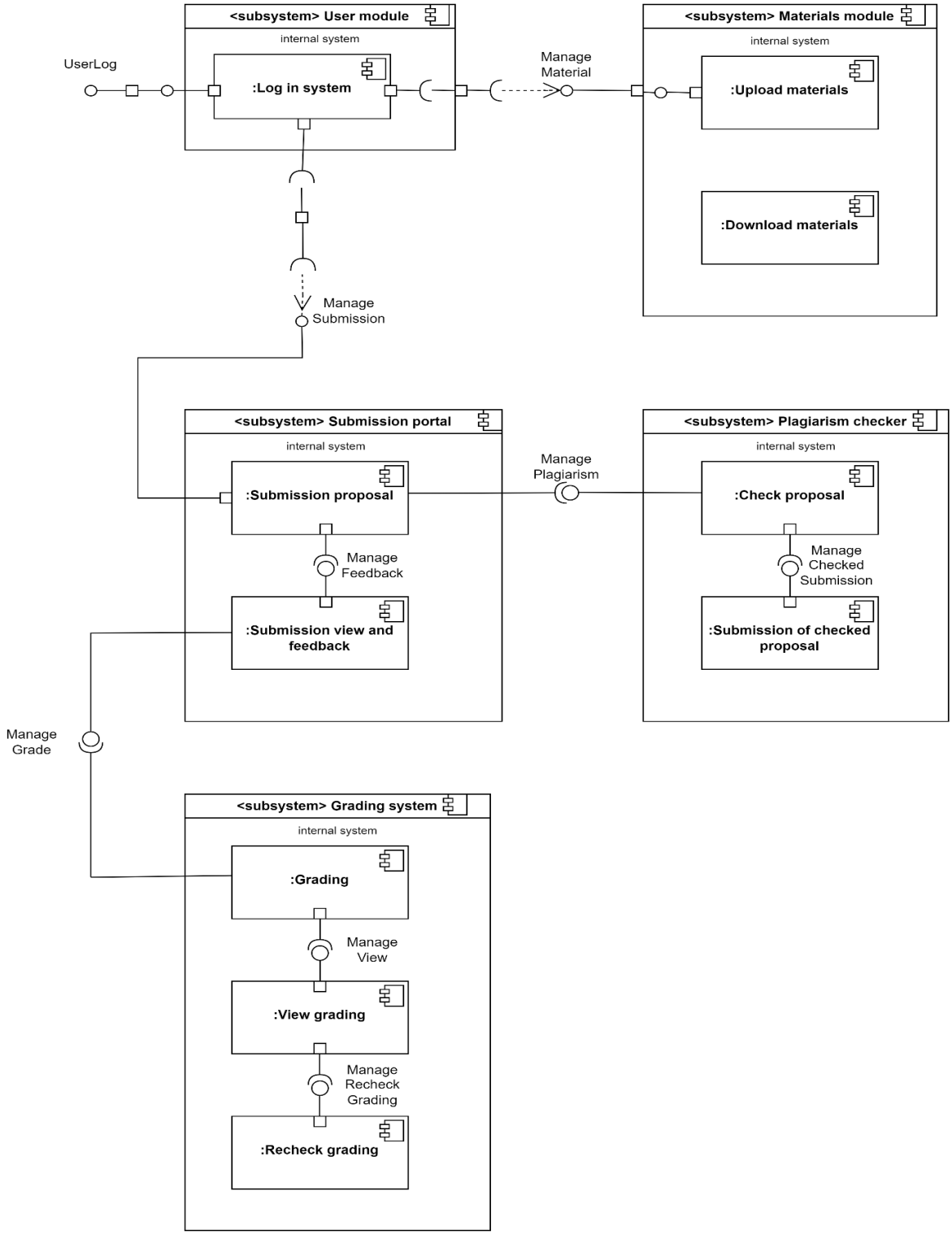


Figure 3.2: Component Model for Final Year Project Management System

4. Detailed Description of Components

4.1 Complete Package Diagram

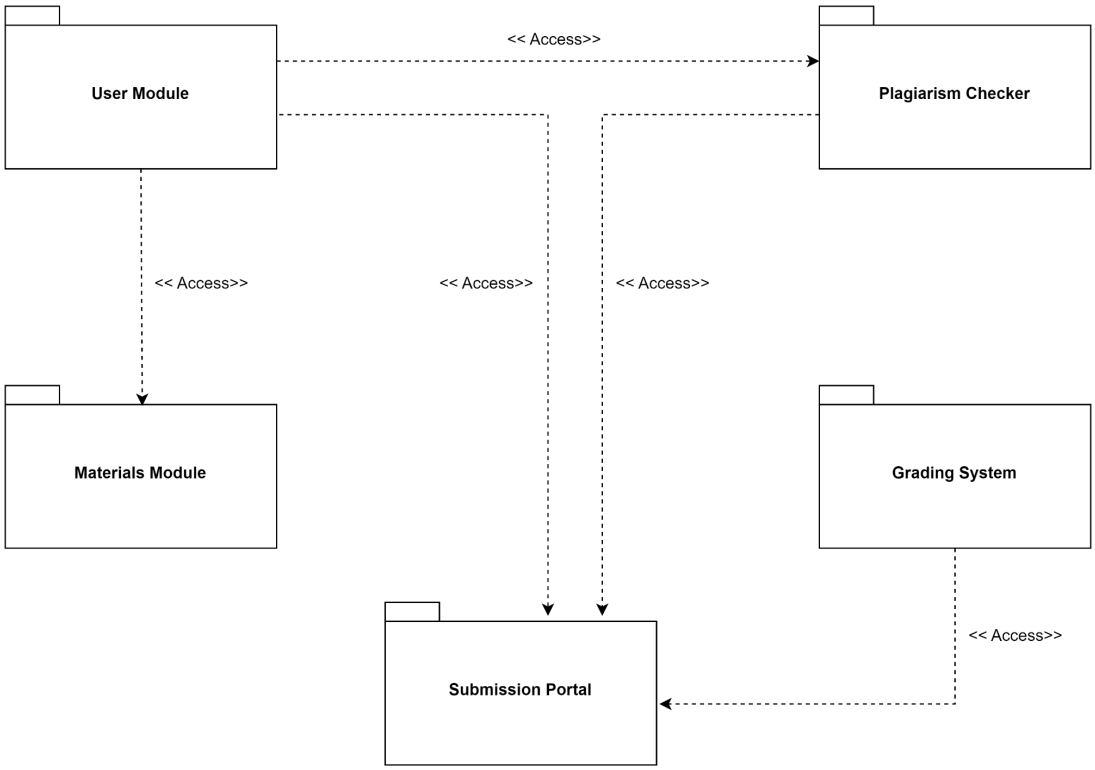


Figure 4.1: Package Diagram for Final Year Project Proposal Management System

4.2 Detailed Description

4.2.1 P001: <User Module> Subsystem

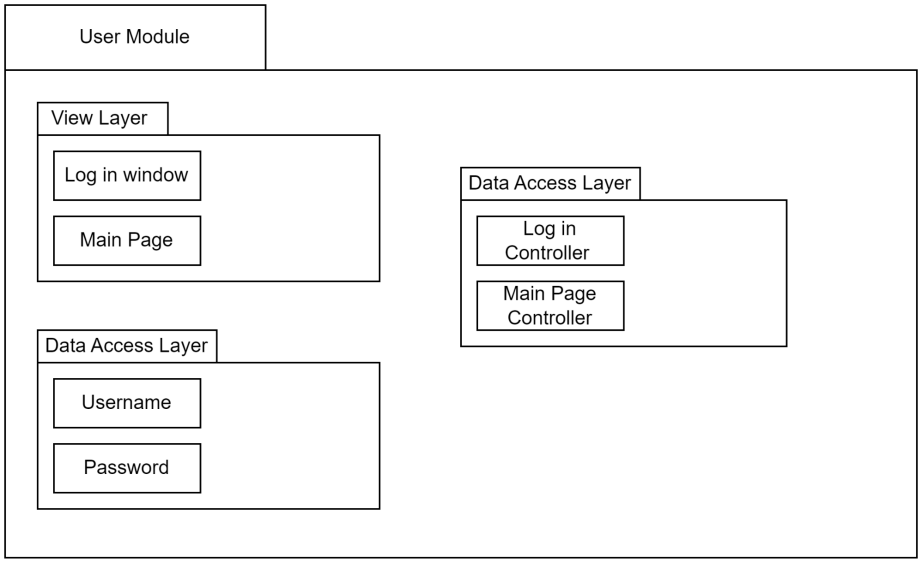


Figure 4.2: Package Diagram for User Module Subsystem

4.2.1.1 Class Diagram

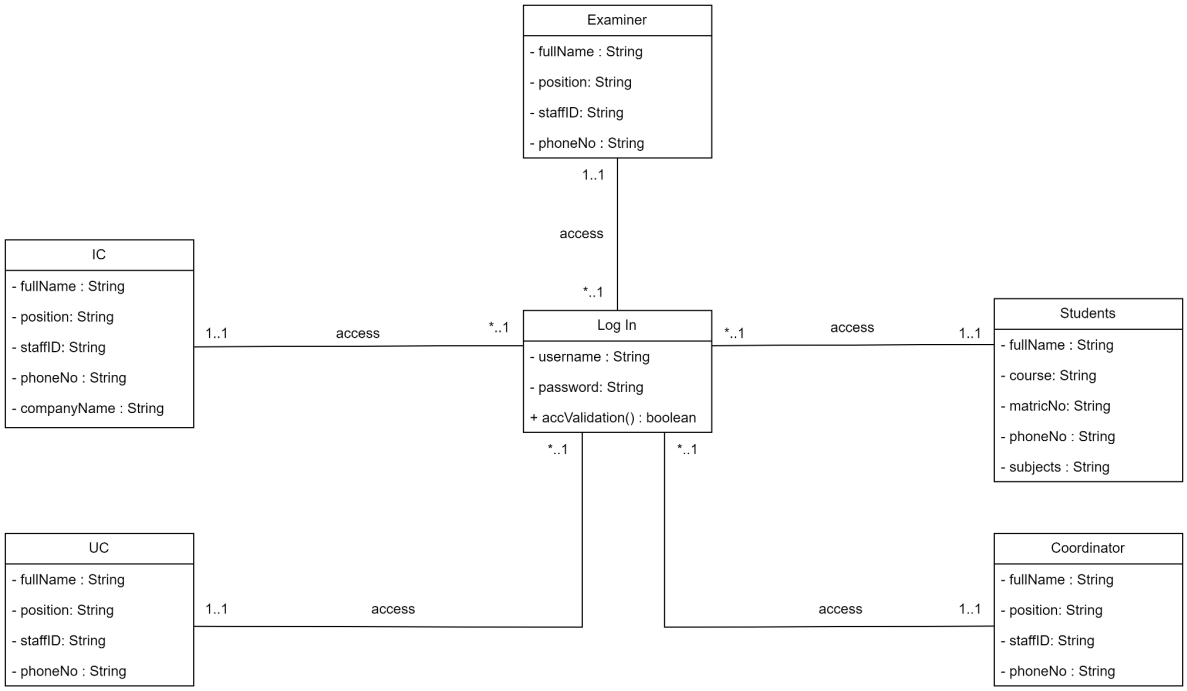


Figure 4.3.: Class Diagram for User Module Subsystem

Entity Name	Log In
Method Name	accValidation()
Input	- username - password
Output	- 1 represents true (valid input) - 0 represents false (invalid input)
Algorithm	1. Start 2. Read username and password. 3. If username is valid return 1 else return 0 4. If bool == 1 Log in successful else Login unsuccessful Re-login 5. End

4.2.1.2 Sequence Diagram

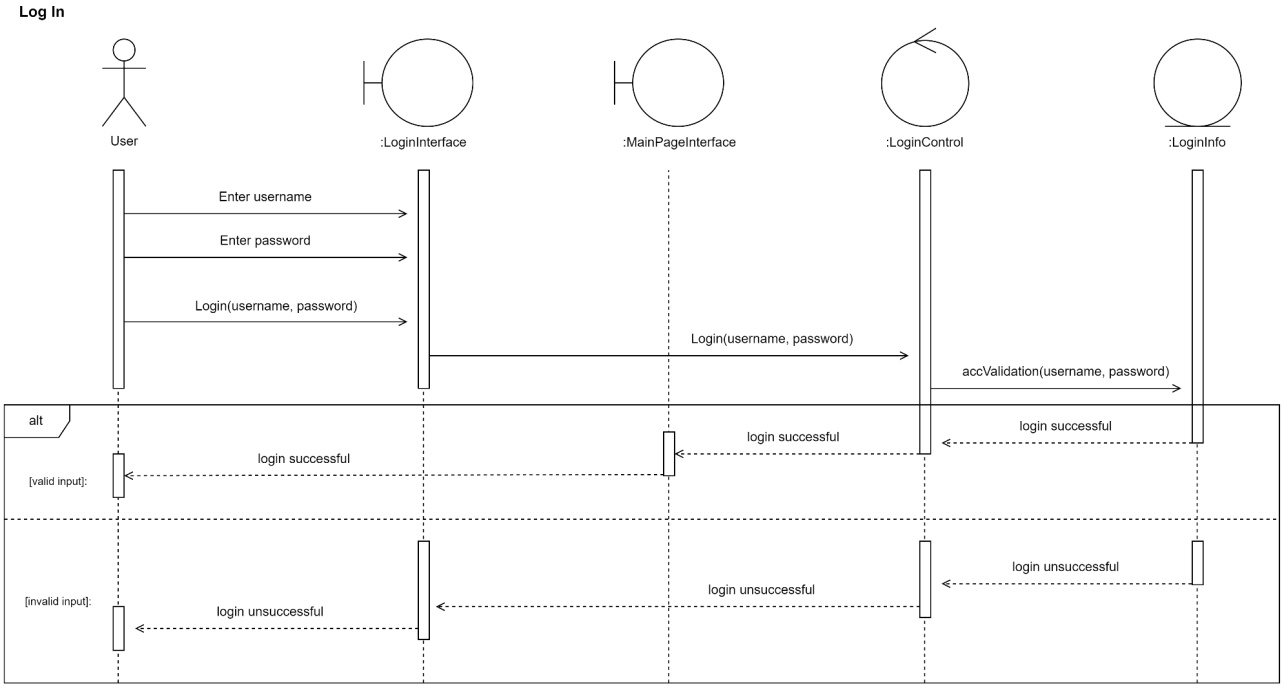


Figure 4.4: Sequence Diagram for User Module Subsystem

4.2.2 P002: <Materials Module> Subsystem

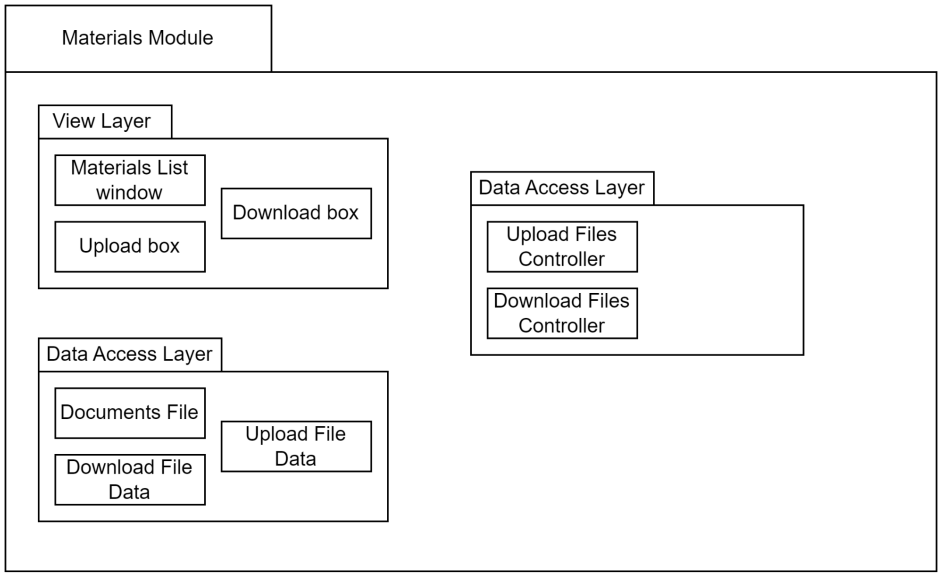


Figure 4.5: Package Diagram for Materials Module Subsystem

4.2.2.1 Class Diagram

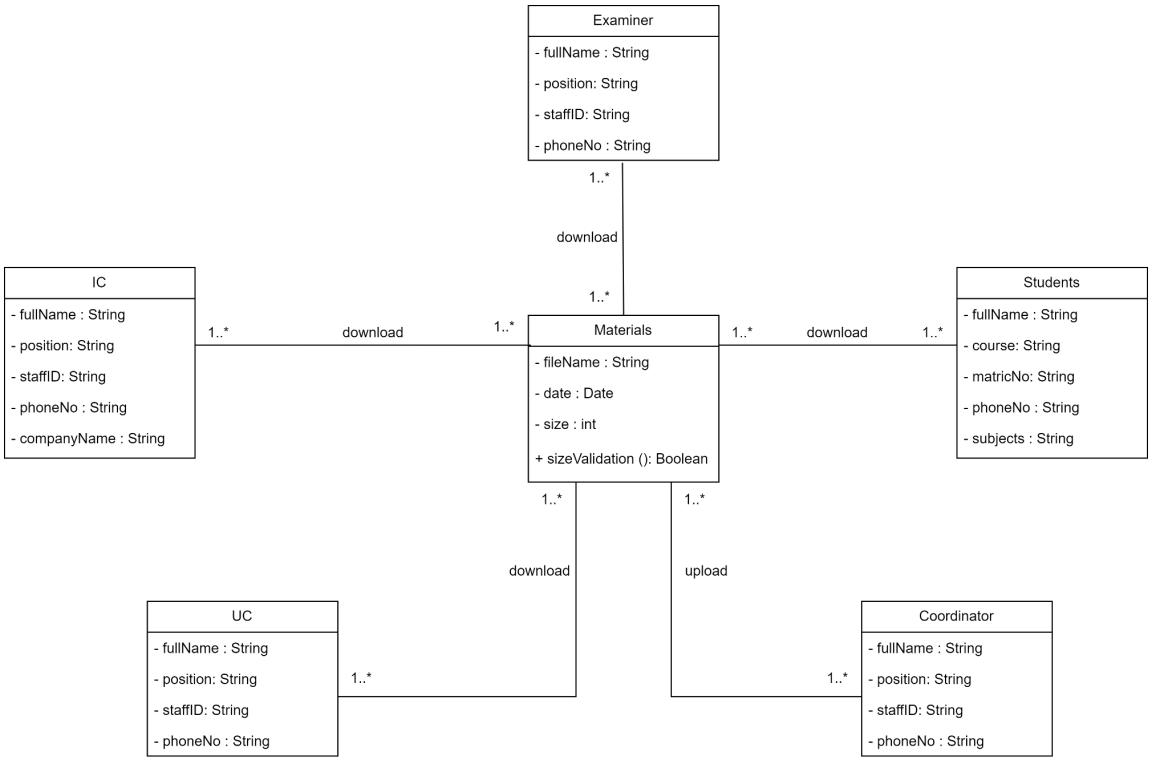


Figure 4.6: Class Diagram for Materials Module Subsystem

Entity Name	Materials
Method Name	sizeValidation()
Input	- size
Output	- 1 represent true (valid input) - 0 represent false (invalid input)
Algorithm	1. Start 2. Read file size. 3. If size < 100Mb return 1 else return 0 4. If bool == 1 Upload successful else Upload unsuccessful Re-upload 5. End

4.2.2.2 Sequence Diagram

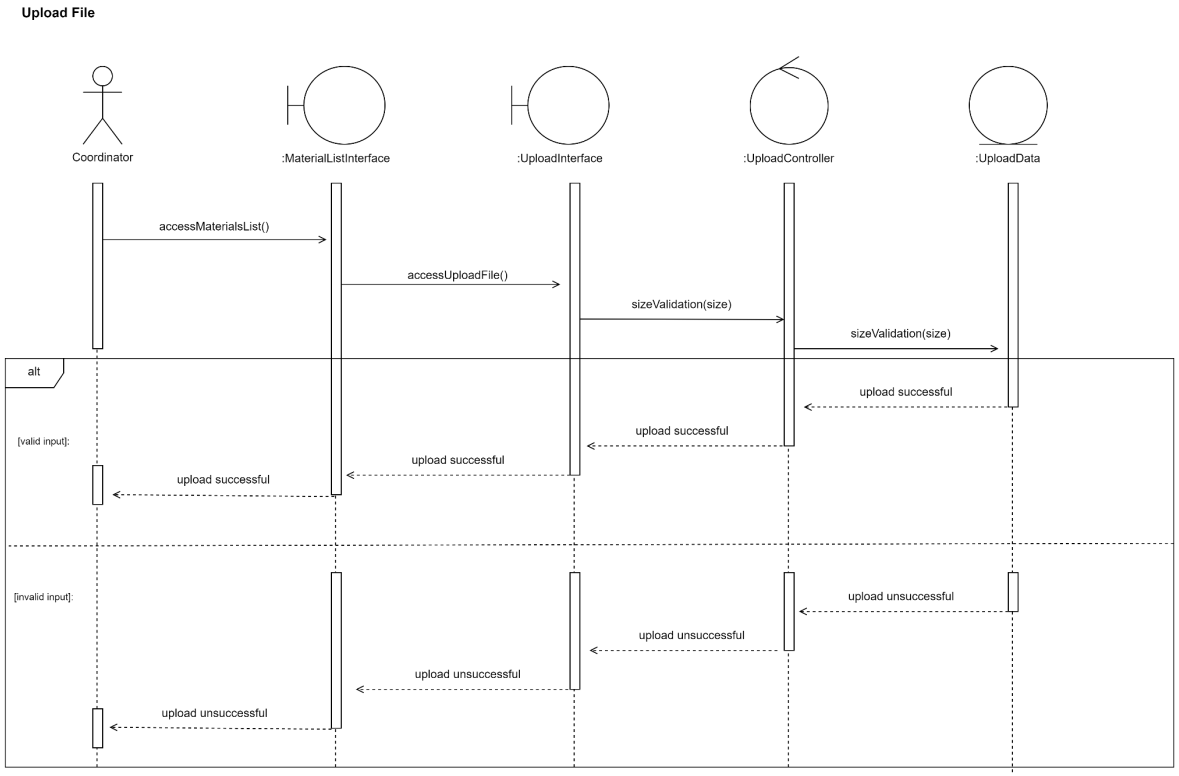


Figure 4.7: Sequence Diagram for Materials Subsystem (Upload File)

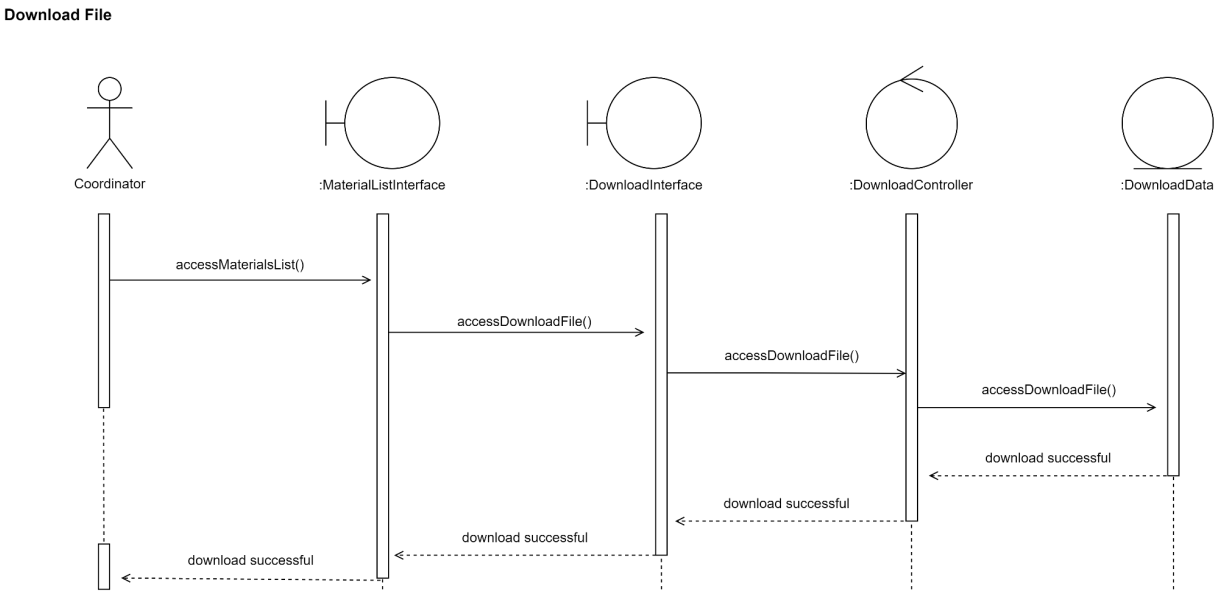


Figure 4.8: Sequence Diagram for Materials Subsystem (Download File)

4.2.3 P003: <Submission> Subsystem

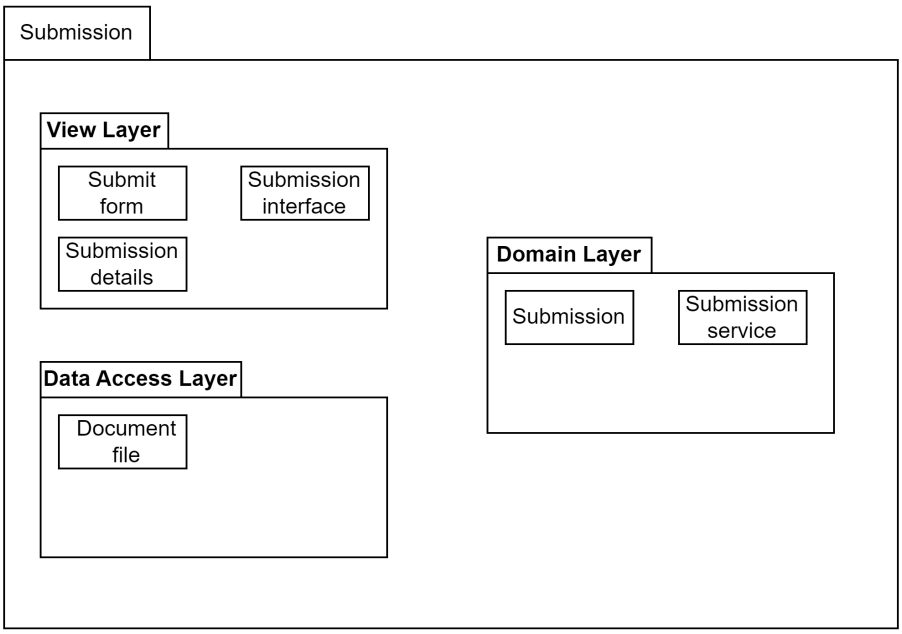


Figure 4.9: Package Diagram for Submission Subsystem

4.2.3.1 Class Diagram

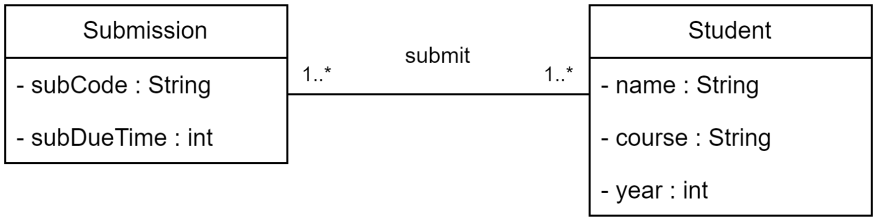


Figure 4.10: Class Diagram for Materials Submission Subsystem

Entity Name	Add submission
Method Name	get_submission()
Input	Document: The submission content.
Output	- 1 represents true (submission successful) - 0 represents false (submission failed)
Algorithm	1. Start 2. Add submission 3. Validate document 4. If the document is valid return 1 else return 0 5. If bool == 1 Submission successful else Submission unsuccessful Re-submit 6. End

4.2.3.2 Sequence Diagram

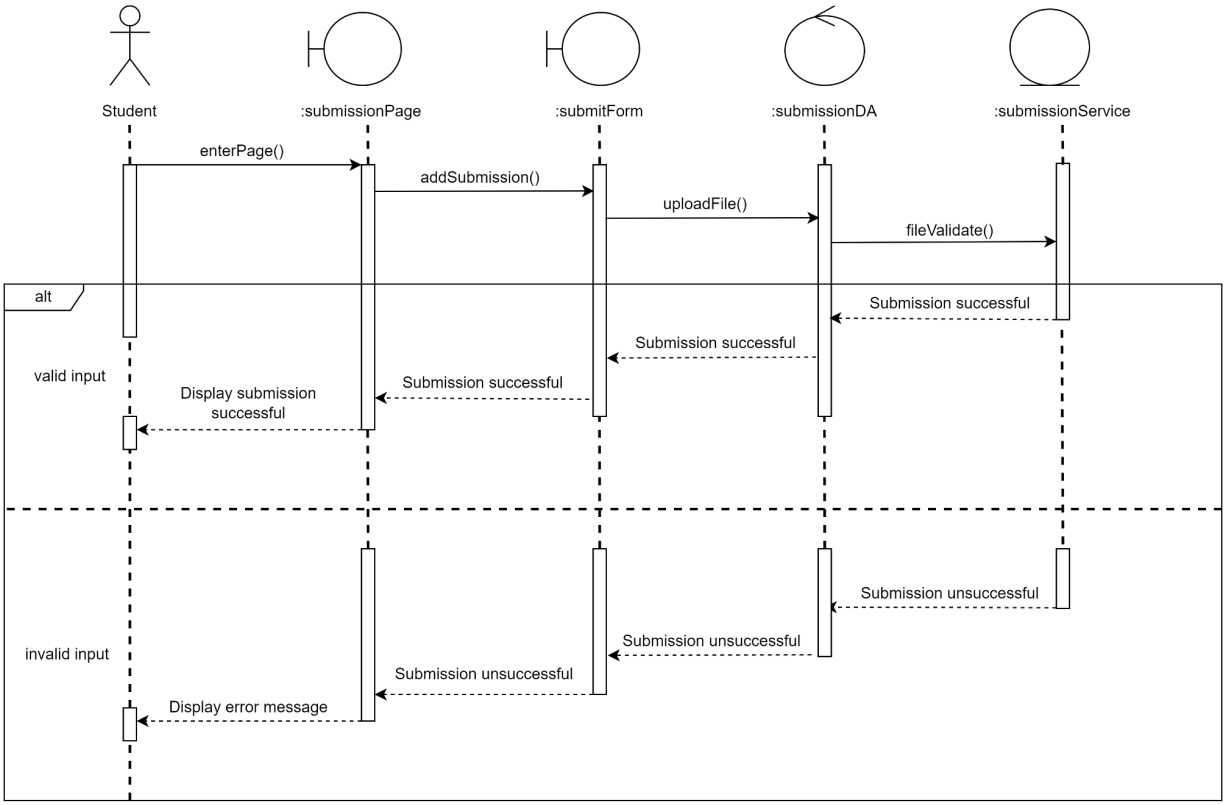


Figure 4.11: Sequence Diagram for Submission Subsystem

4.2.4 P004: <View and Feedback> Subsystem

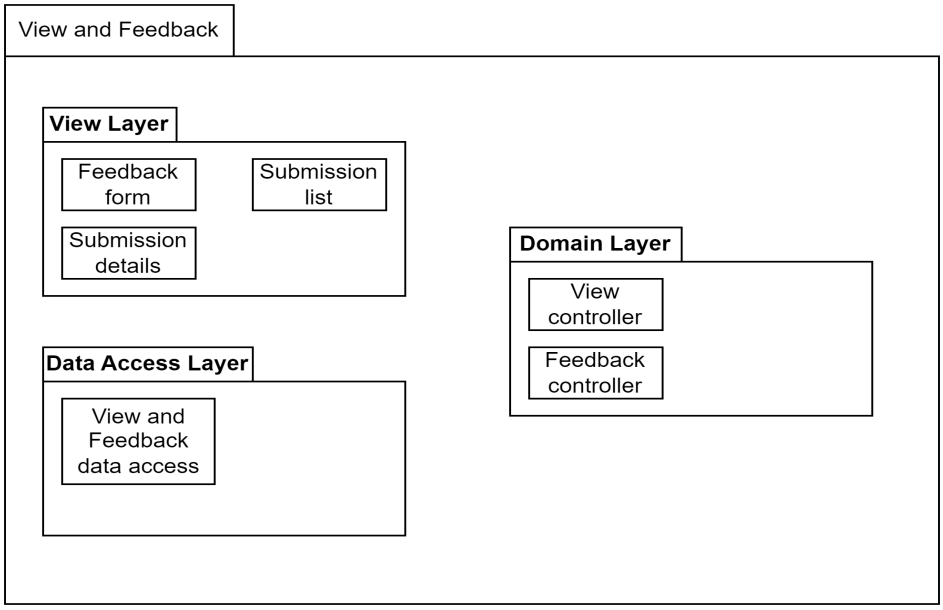


Figure 4.12: Package Diagram for View and Feedback Subsystem

4.2.4.1 Class Diagram

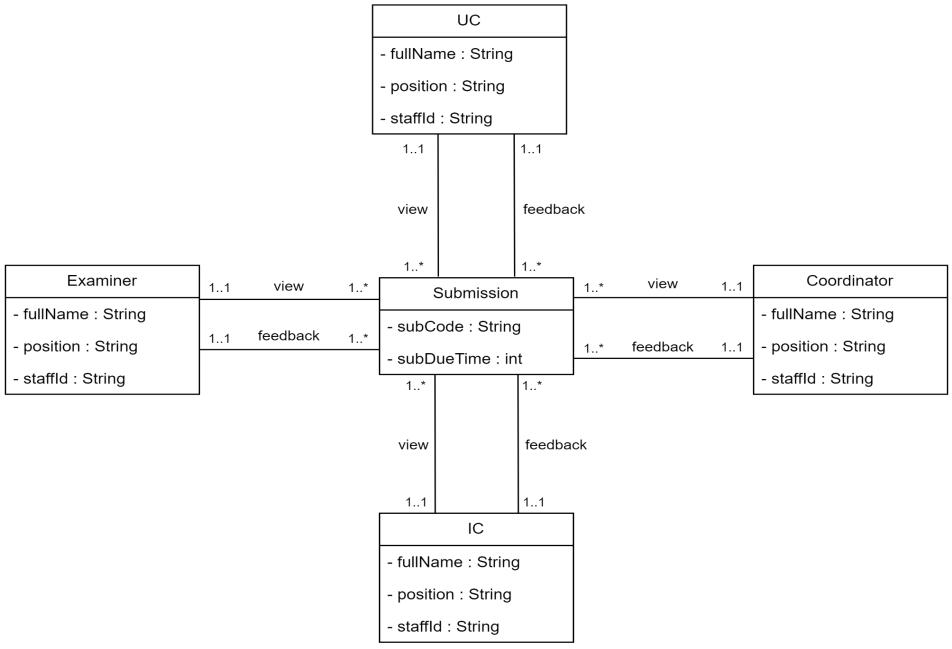


Figure 4.13: Class Diagram for Materials Module Subsystem

Entity Name	View and give feedback
Method Name	give_feedback()
Input	Feedback: The feedback provided for the submission
Output	- Submission Details: Information about the submission, such as user ID, content, timestamp, and approval status. - Feedback status: Comment about the current state of the submission.
Algorithm	1. Start 2. Read submission 3. View submission 4. Add feedback 5. End

4.2.4.2 Sequence Diagram

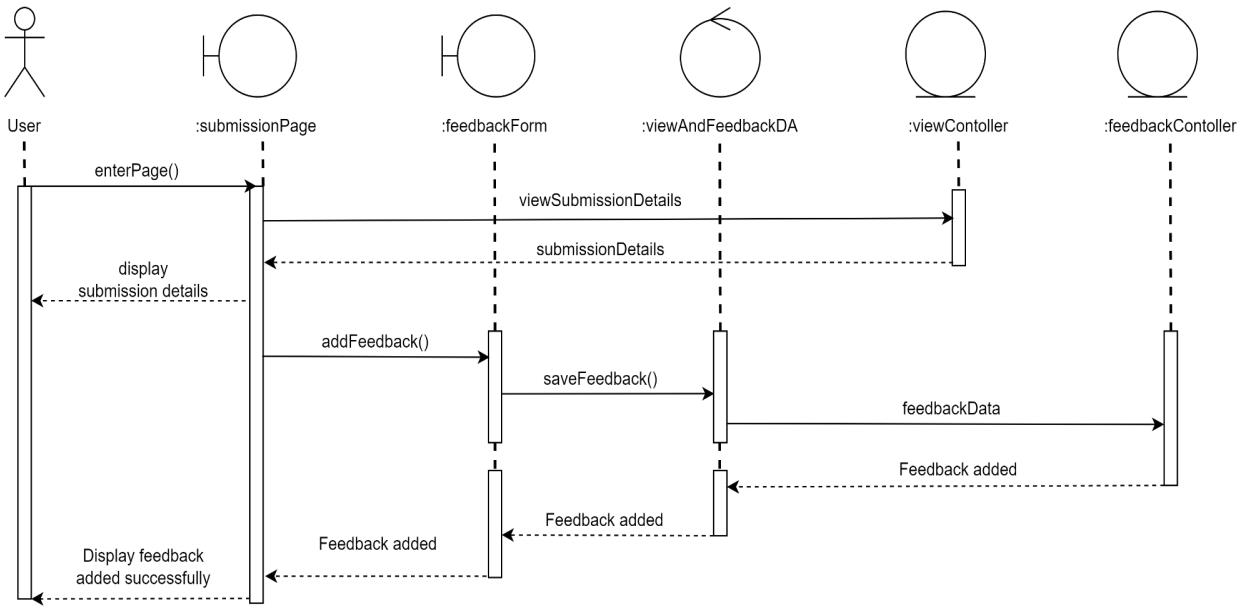


Figure 4.14: Sequence Diagram for View and Feedback Subsystem

4.2.5 P005: <Plagiarism Checker> Subsystem

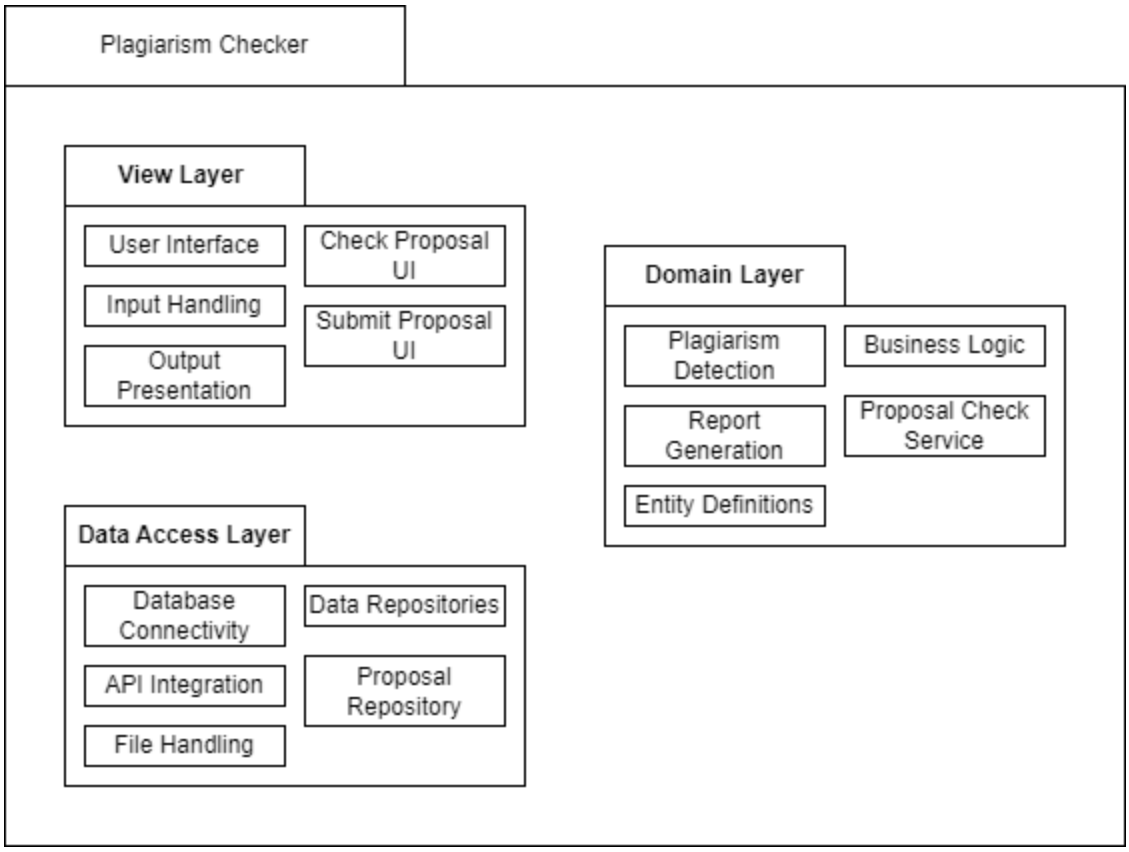


Figure 4.15: Package Diagram for <Plagiarism Checker> Subsystem

4.2.5.1 Class Diagram

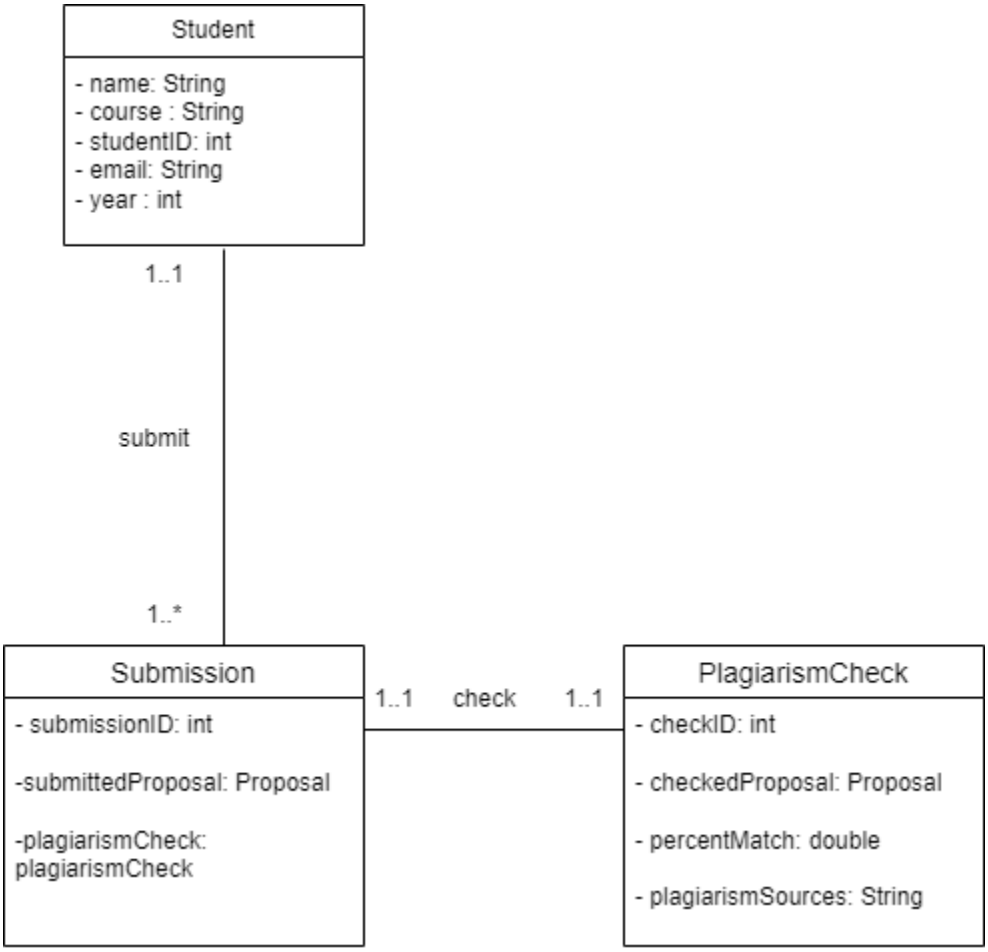


Figure 4.16: Class Diagram for <Plagiarism Checker> Subsystem

Entity Name	PlagiarismChecker
Method Name	checkProposal
Input	- proposal: Proposal
Output	- percentMatch: double - plagiarismSources: String
Algorithm	1. Preprocess the proposal by removing irrelevant content and normalizing text. 2. Compare the proposal against a database of existing sources. 3. Calculate the plagiarism score based on the matched content. 4. Identify the sources of plagiarism and generate a list of plagiarism sources. 5. Return the plagiarism score and plagiarism sources.

4.2.5.2 Sequence Diagram

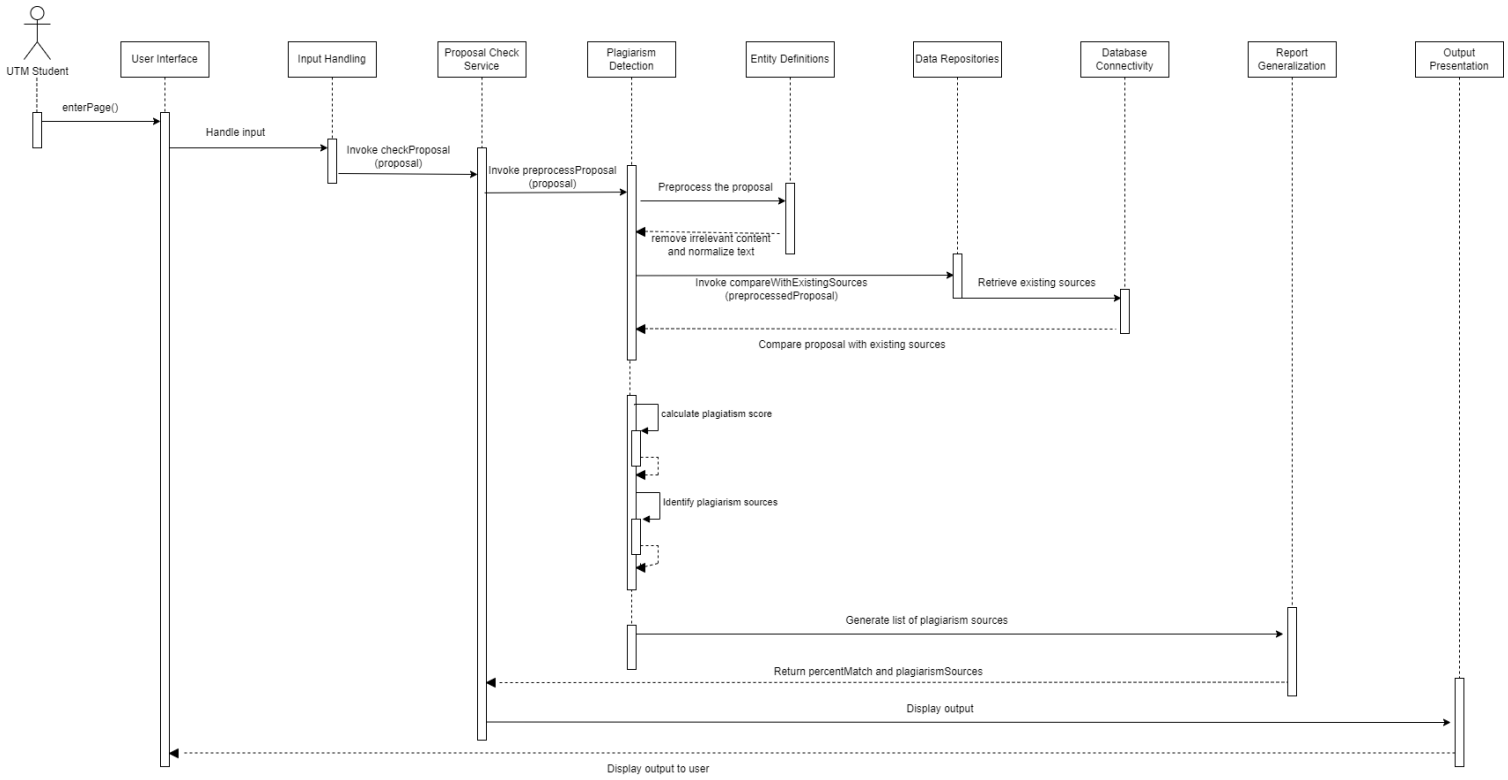


Figure 4.17: Sequence Diagram for Plagiarism Checker Subsystem

4.2.6 P006: <Grading Management> Subsystem

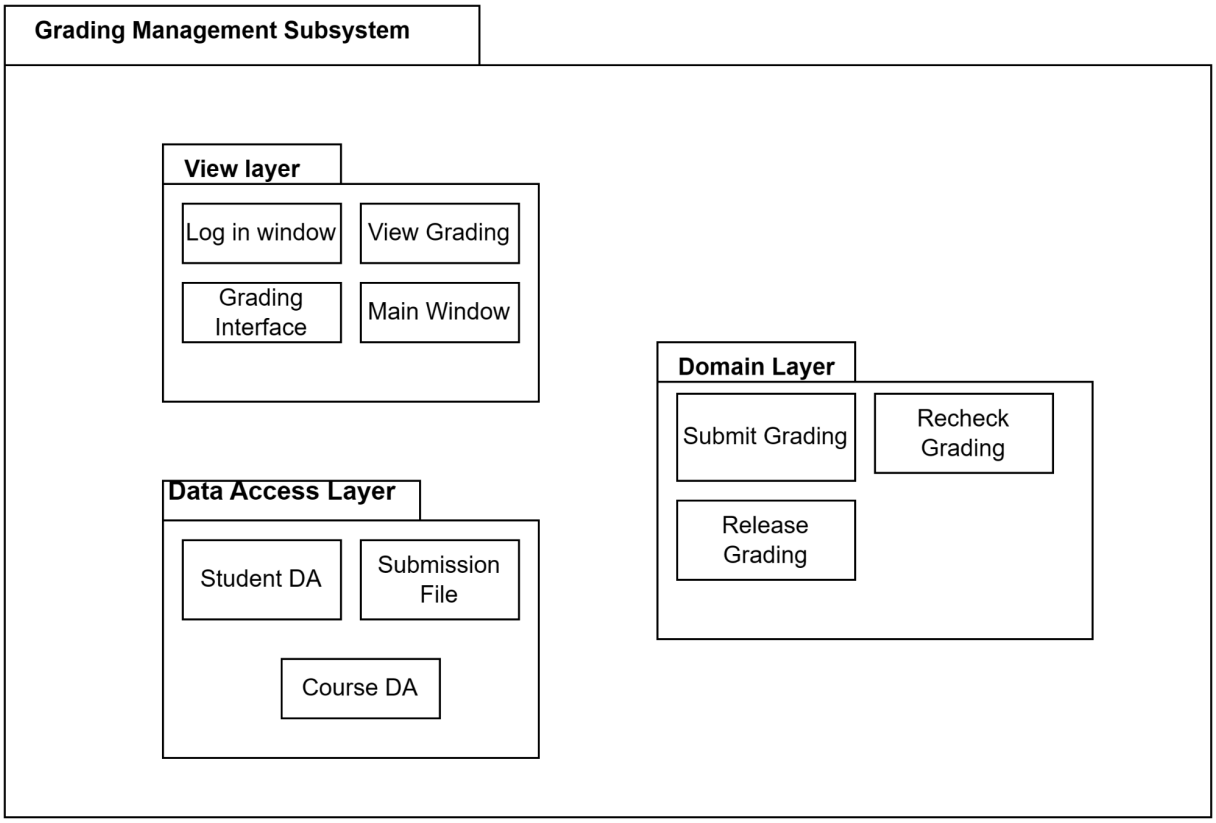


Figure 4.18: Package Diagram for Grading Module Subsystem

4.2.6.1 Class Diagram

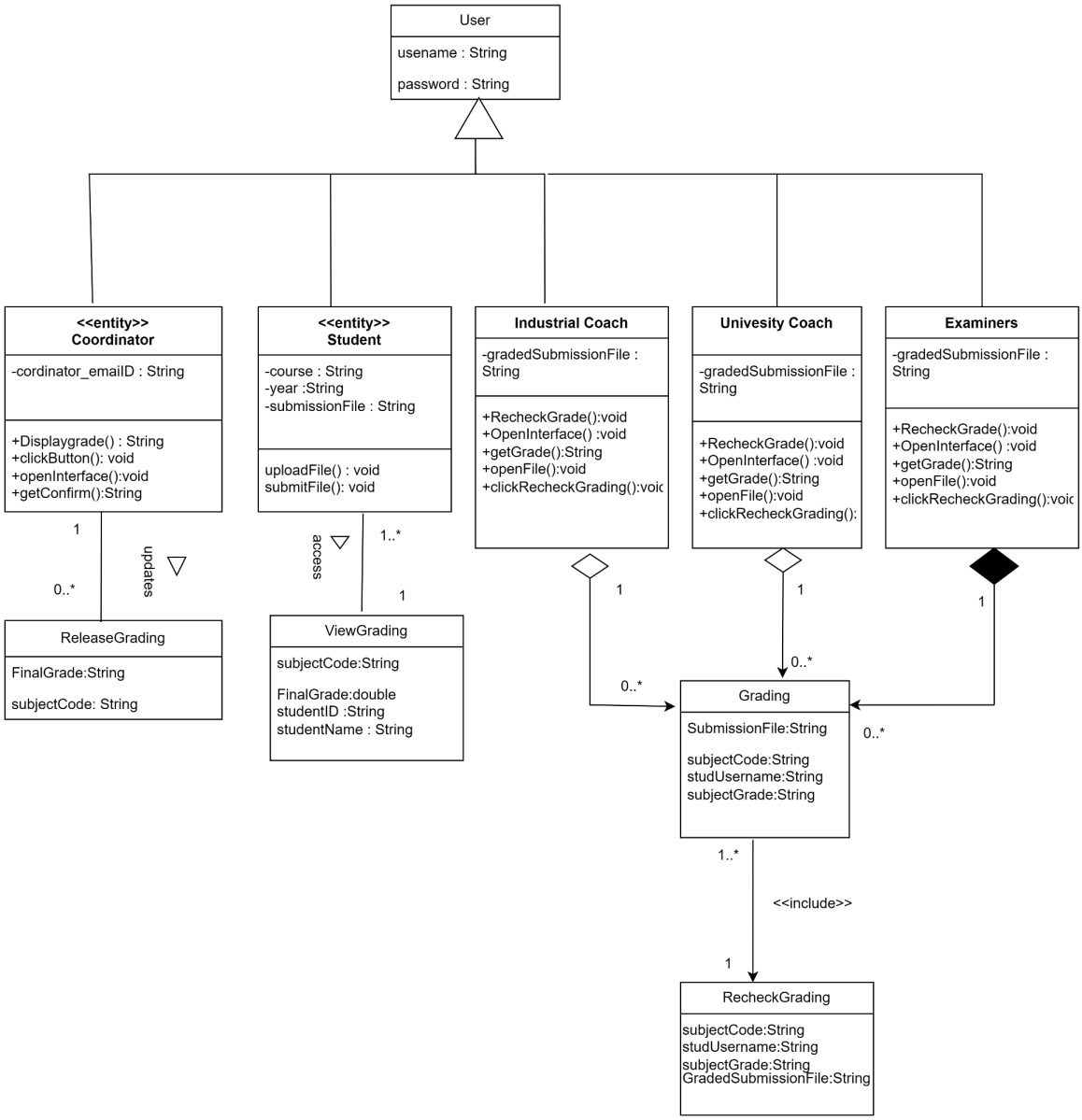


Figure 4.19: Class Diagram for Grading Module Subsystem

UC008: Grading

Entity Name	Examiner, Industry Coach, University Coach
Method Name	getGrades() OpenFile() OpenInterface()
Input	Submission_File, subject_code
Output	Graded_SubmissionFile
Algorithm	1. Start 2. Login System 2.1 IF user is Examiner 2.1.1 Start Loop 2.1.1.1 Select ‘Open’ Submission_File 2.1.1.2 Do grading based on Course Code 2.1.1.3 End Loop 2.2 ELSE 2.2.1 Get Assessment Form 2.2.2 Do grading based on skill 3. The user submits Graded_SubmissionFile to Grading Interface 4. Save and record the Grades 5. End

UC0009: Recheck Grading

Entity Name	Examiner, Industry Coach, University Coach
Method Name	RecheckGrade() OpenInterface() getGrades() OpenFile() ClickRecheckButton()
Input	Graded_SubmissionFile, phaseOfSubject
Output	Final_Grade
Algorithm	1. Start 2. Login System 2.1 IF user is Examiner 2.1.1 Start Loop 2.1.1.1 Click 'Recheck' Button 2.1.1.2 Do grading based on Course Code 2.1.1.3 End Loop 2.2 ELSE 2.2.1 Click 'Recheck' Button 2.2.2 Do grading based on skill 3. User submit Final_Grade to Grading Interface
	4 Save and record the Final_Grade 5 End

UC010: Release Grading

EntityName	Coordinator
Method Name	DisplayGrade(): String OpenInterface() Compare() ClickButton() getConfirm() Cancel() returnpage()
Input	subject_code
Output	Final Grades have been decided.
Algorithm	1. Start 2. Select Grading Interface. 3. Select File 4. Click Open File 5. Click 'Public' Button to release Grades 6. Display Confirmation Message 6.1 IF user press confirm 6.1.1 Display Grade 6.2 ELSE 6.2.1 Return Grading Interface 7. END

UC011: View Grading

Entity Name	Student
Method Name	OpenInterface() ViewGrade(): String displayGrade(): String getCode()
Input	phaseOfSubject
Output	Grades of a subject have been displayed
Algorithm	1. Start 2. Select Grading Interface 3. Select Enter Subject Code 4. Student enters Subject Code 4.1 IF code is valid 4.1.1 Display Grade 4.2 ELSE 4.2.1 Return Grading Interface 5. End

4.2.6.2 Sequence Diagram

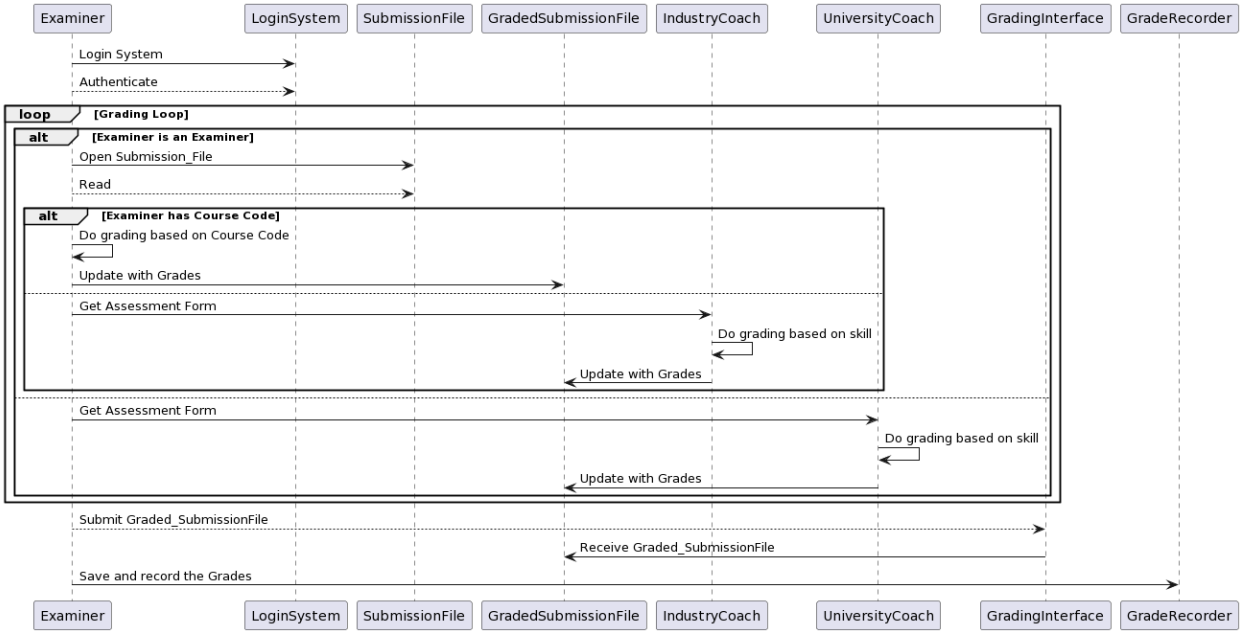


Figure 4.20: Sequence Diagram for Grading Module Subsystem

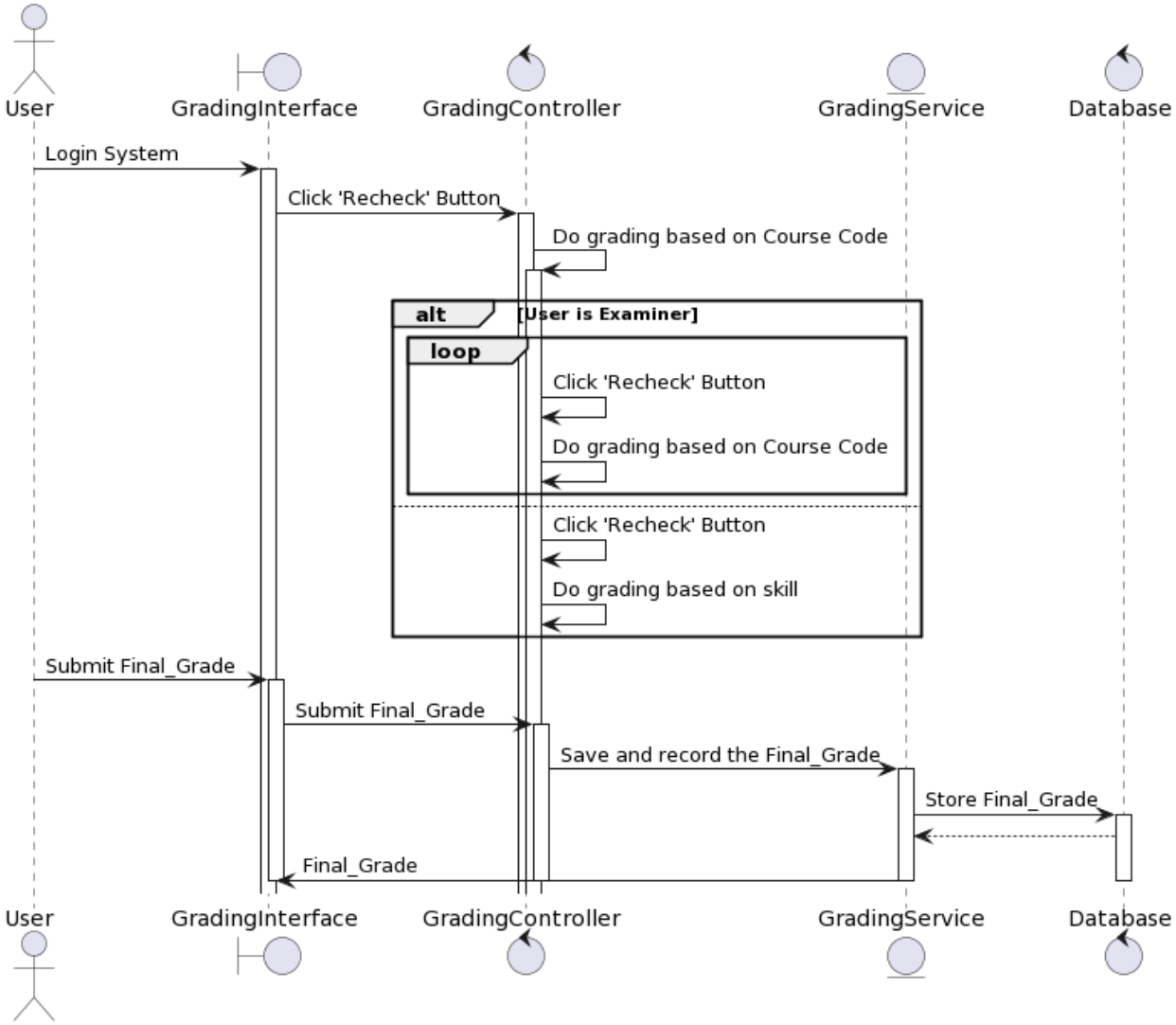


Figure 4.21: Sequence Diagram for Recheck Grading Subsystem

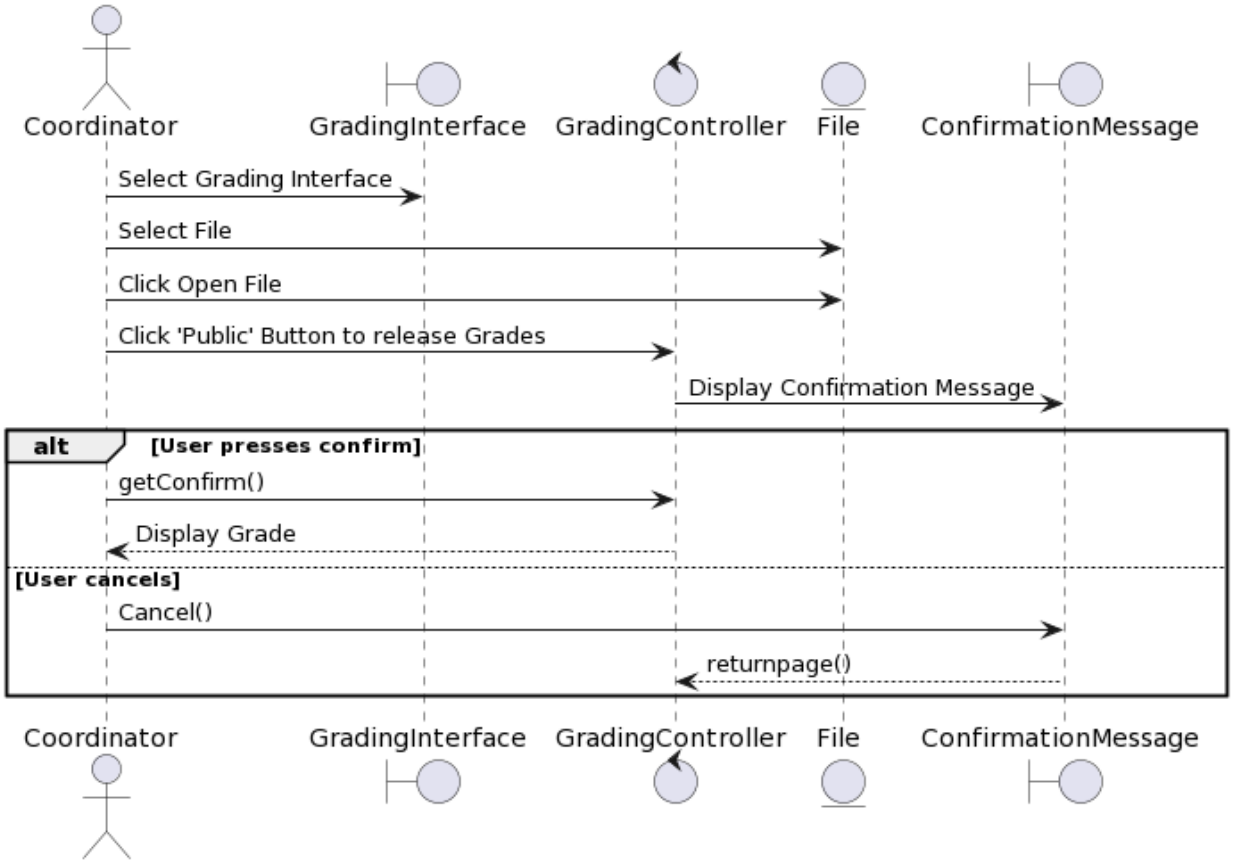


Figure 4.22: Sequence Diagram for Release Grading Subsystem

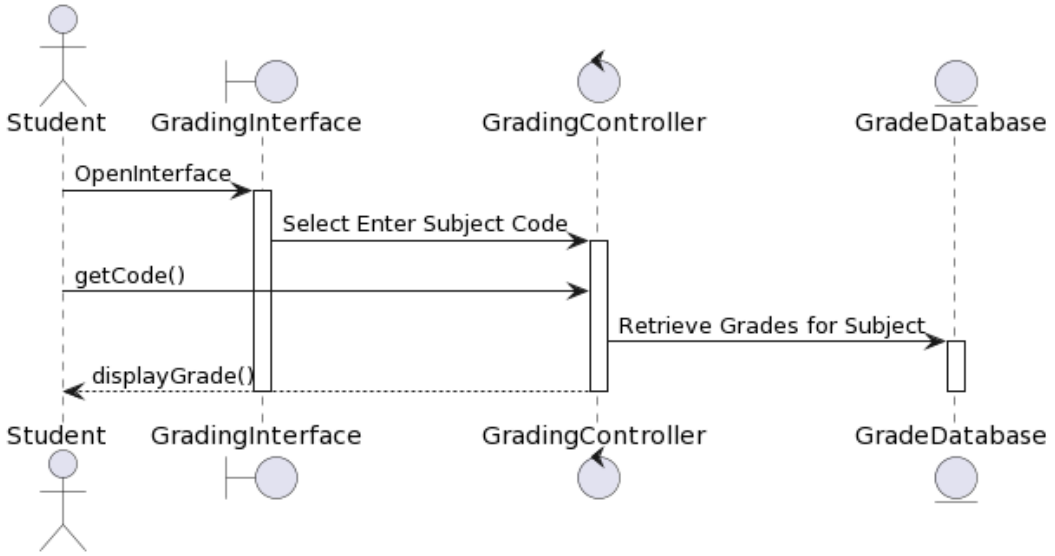


Figure 4.23: Sequence Diagram for View Grading Subsystem

5. Data Design

5.1 Data Description

The major data or systems entities are stored in a relational database named as…, processed, and organized into *n* entities as listed in Table 5.1.

Table 5.1: Description of Entities in the Database

No.	Entity Name	Description
1	Students	Data stores the details about students including their full name, matric number, phone number, course and subjects taken.
2	Coordinator	Data stores the details about coordinators including their full name, position, staff ID, and phone number.
3	IC	Data stores the details about IC including their full name, position, staff ID, company name, and phone number.
4	UC	Data stores the details about UC including their full name, position, staff ID, and phone number.
5	Examiners	Data stores the details about the examiner including their full name, position, staff ID, and phone number.
6	Log In	Stores usernames and passwords for each log-in into the system.
7	Materials	Stores all details about important materials uploaded into the system including file name, date uploaded, and size of the file.

8	Proposal	Represents a project proposal submitted by a student.
9	PlagiarismCheckResult	Represents the result of a plagiarism check performed on a proposal.
10	Submission	Represents the submission of a checked proposal.
11	PlagiarismSource	Represents a potential source or reference identified as plagiarism.
12	Grading	Allow the user to do grading on the downloaded submission file. (SubmissionFile, subjectCode, StudentUsername, SubjectGrade)
13	RecheckGrading	Ensure the grades of the graded Submission_File. are graded fair and square, (GradedSubmissionFile, subjectCode, studentUsername, subjectGrade)
14	Release Grading	Release the final grades of the students. (subjectCode, subjectGrade).
15	View Grading	Allow user to view the grade of their project submission(subjectCode, subjectGrade)

5.2 Data Dictionary

5.2.1 Entity: <Students>

Attribute Name	Type	Description
fullName	String	Student's full name
course	String	Course taken by students
matricNo	String	Student's matric number
phoneNo	String	Student's phone number
subjects	String	Subject's taken throughout all semesters.
email	String	Student's email address
year	int	Student's academic year

5.2.2 Entity: <Coordinator>

Attribute Name	Type	Description
fullName	String	Coordinator's full name
position	String	Coordinator position in university
staffID	String	Coordinator's staff id number
phoneNo	String	Coordinator's phone number

5.2.3 Entity: <IC>

Attribute Name	Type	Description
fullName	String	IC's full name
position	String	IC's position in company
staffID	String	IC's staff id number
phoneNo	String	IC's phone number
companyName	String	Name of the company

5.2.4 Entity: <UC>

Attribute Name	Type	Description
fullName	String	UC's full name
position	String	UC's position in company
staffID	String	UC's staff id number
phoneNo	String	UC's phone number

5.2.5 Entity: <Examiner>

Attribute Name	Type	Description
fullName	String	Examiner's full name
position	String	Examiner's position in company
staffID	String	Examiner's staff id number
phoneNo	String	Examiner's phone number

5.2.6 Entity: <Log In>

Attribute Name	Type	Description
username	String	Username of the user
password	String	Password of the user account

5.2.7 Entity: <Materials>

Attribute Name	Type	Description
fileName	String	Name of the file uploaded
date	Date	Date of uploading and downloading file
size	int	Size of file uploaded

5.2.8 Entity: <Submission>

Attribute Name	Type	Description
subCode	String	Subject Code
subDueTime	int	Time due for submission
submissionID	int	Unique identifier for the submission.

5.2.9 Entity: <Plagiarism Checker>

Attribute Name	Type	Description
checkID	int	Unique identifier for the check result.
percentMatch	double	The percentage of matched content indicating the level of plagiarism.
plagiarismSources	String	A list of sources or references identified as potential plagiarism sources.

5.2.10 Entity: <Grading>

Attribute Name	Type	Description
submissionFile	String	Submission File for each subject of the Student
subjectCode	String	Uniquely identifies the phase of subjects that have been submitted by the student
studUsername	String	Username of the Student
subjectGrade	String	The grades for each subject of the Student

5.2.11 Entity: <RecheckGrading>

Attribute Name	Type	Description
GradedSubmissionFile	String	Graded Submission File for each subject of the Student
subjectCode	String	Uniquely identifies the code of subjects
studUsername	String	Username of the Student
subjectGrade	String	The grades for each subject of the Student

3.2.12 Entity: <ReleaseGrading>

Attribute Name	Type	Description
subjectCode	String	Uniquely identifies the code of subjects
FinalGrade	String	The final grade of the subject for each student

5.2.13 Entity: <ViewGrading>

Attribute Name	Type	Description
studentName	String	Name of Student
studentID	Int	Uniquely identifies the Student
subjectCode	String	Uniquely identifies the code of subjects
FinalGrade	String	The final grade of the subject for each student

6. Requirements Traceability Matrix

Table 6.1: RTM for <FYP Management System>

Package Item	Use Case ID	Use Case Description	Sequence Diagram ID	Sequence Diagram Description	Test Case ID
Package 1	UC001	Log In	SD001	User Log In	TC001
Package 2	UC002	Upload Materials	SD002	User Upload Materials	TC002
	UC003	Download Materials	SD003	User Download Materials	TC003
Package 3	UC004	Submission	SD004	User Submit File	TC004
Package 4	UC005	View and Feedback	SD005	User View Submission and Give Feedback	TC005
Package 5	UC006 & UC007	Check Plagiarism & Submission of Checked Proposal	SD006	User check the percentage of similarity with the resources used to refer and submit the checked proposal	TC006
Package 6	UC008	Grading	SD007	Grading Module Subsystem	TC008
	UC009	Recheck Grading	SD008	Recheck Grading Subsystem	TC009
	UC010	Release Grading	SD009	Release Grading Subsystem	TC010
	UC011	View Grading	SD010	View Grading Subsystem	TC011

7. Test Cases

A test case is a thorough set of requirements, assumptions, and anticipated results used to evaluate the performance and behavior of a particular feature, module, or system. It acts as a specification for the actions to be taken and the outcomes that should be seen throughout the testing process.

7.1 TC001: Test <User Module> Subsystem: < Log In (UC001)>

This test contains the following test cases:

(a) TC001_01: Test <Scenario of sequence diagram1 (SD001)>

7.1.1 TC001_01: Test <state scenario of sequence diagram1 (SD001)>

This test contains the following scenarios:

(a) TC001_01_01: Test <Normal scenario of sequence diagram1 (SD001)>

Table 7.1 : TC001_01_01-<Normal Scenario of Sequence diagram1 (SD001)>

Test Case ID	T001	**Test Case Discription**	Log In System
Created By	Nur Immal Hayati binti Hasmi Anuar		

QA Tester's Log

Date Tested	28/6/2023	**Test Case (Pass/Fail/Not)**	

S#	Prerequisites:
1	User currently access internet browser
2	
3	
4	

S#	Test Data
1	Username = aliabu12
2	Password = ali@12abu
3	
4	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Access the website using link given	Can access the website		
2	Enter username & password	Key in successfully		
3	Click Log In	Log in successfully		
4				

(b) TC001_01_02: Test <Exception scenario of sequence diagram1 (SD001)>

Table 7.1 : TC001_01_02-<Exception Scenario of Sequence diagram1 (SD001)>

Test Case ID	T001	**Test Case Discription**	Log In System
Created By	Nur Immal Hayati binti Hasmi Anuar		

QA Tester's Log

Date Tested	28/6/2023	**Test Case (Pass/Fail/Not)**	

S#	Prerequisites:
1	User currently access internet browser
2	
3	
4	

S#	Test Data
1	Username = ali12
2	Password = ali@12
3	
4	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Access the website using link given	Can access the website		
2	Enter username & password	Key in successfully		
3	Click Log In	Failed to log in		
4				

7.2 TC002: Test <Materials Module> Subsystem: < Upload File (UC002)>

This test contains the following test cases:

(a) TC002_01: Test <Scenario of sequence diagram2 (SD002)>

(a) TC002_02: Test <Scenario of sequence diagram3 (SD003)>

7.2.1 TC002_01: Test <state scenario of sequence diagram 1 (SD001)>

This test contains the following scenarios:

(a) TC002_01_01: Test <Normal scenario of sequence diagram2 (SD002)>

Table 7.2 : TC002_01_01-<Normal Scenario of Sequence diagram 2 (SD002)>

Test Case ID	T002	**Test Case Discription**	Upload file into the system
Created By	Nur Immal Hayati binti Hasmi Anuar		

QA Tester's Log

Date Tested	28/6/2023	**Test Case (Pass/Fail/Not)**	

S#	Prerequisites:
1	User currently in the system
2	
3	
4	

S#	Test Data
1	A file with size 20Mb
2	
3	
4	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Click Materials button	Can click button		
2	Display all the list of materials	Can display list of materials		
3	Click upload materials	Can click upload button		
4	Choose file to upload	Can choose file to upload		
5	Upload file	Upload successfully		

(b) TC002_01_02: Test <Exception scenario of sequence diagram2 (SD002)>

Table 7.2 : TC002_01_02-<Exception Scenario of Sequence diagram 2 (SD002)>

Test Case ID	T002	Test Case Discription	Upload file into the system
Created By	Nur Immal Hayati binti Hasmi Anuar		

QA Tester's Log

Date Tested	28/6/2023	Test Case (Pass/Fail/Not)	

S#	Prerequisites:
1	User currently in the system
2	
3	
4	

S#	Test Data
1	A file with size 121Mb
2	
3	
4	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Click Materials button	Can click button		
2	Display all the list of materials	Can display list of materials		
3	Click upload materials	Can click upload button		
4	Choose file to upload	Can choose file to upload		
5	Upload file	Upload failed		

(c) TC003_02_01: Test <Normal scenario of sequence diagram3 (SD003)>

Table 7.2 : TC002_02_01-<Normal Scenario of Sequence diagram 3 (SD003)>

Test Case ID	T003	Test Case Discription	Download file into the system
Created By	Nur Immal Hayati binti Hasmi Anuar		

QA Tester's Log

Date Tested	28/6/2023	Test Case (Pass/Fail/Not)	

S#	Prerequisites:
1	User currently in the system
2	
3	
4	

S#	Test Data
1	
2	
3	
4	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Click Materials button	Can click the button		
2	Display all the list of materials	Can display materials		
3	Click download materials	Can click the button		
4	Download file	Download successfully		

7.3 TC004: Test <Submission> Subsystem: < Submit File (UC004)>

This test contains the following test cases:

(a) TC004_01: Test <Scenario of sequence diagram (SD004)>

7.3.1 TC004_01: Test <Submission (SD004)>

This test contains the following scenarios:

(a) TC004_01_01: Test <Normal scenario of sequence diagram (SD004)>

Table 7.3 : TC004_01_01-<Normal Scenario of Sequence diagram (SD004)>

Test Case ID	TC004	**Test Case Discription**	Submit File
Created By	Ain Batrisyia Binti Norazlan		
QA Tester's Log			
Date Tested	30/6/2023	**Test Case (Pass/Fail/Not)**	

S#	Prerequisites:
1	Have internet connection
2	Must submit in PDF format
3	
4	

S#	Test Data
1	Valid document format
2	
3	
4	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Enter submission page	Display submission page		
2	Click link for chosen chapter	Can click all the link		
3	Click add submission	Can click the add submission button		
4	Upload file	Upload file successfully		
5	Click submit button	Can click submit button		

(b) TC004_01_02: Test <Exception scenario of sequence diagram1 (SD004)>

Table 7.3 : TC004_01_02-<Exception Scenario of Sequence diagram (SD004)>

Test Case ID	TC004	**Test Case Discription**	Submit File
Created By	Ain Batrisyia Binti Norazlan		
QA Tester's Log			
Date Tested	30/6/2023	**Test Case (Pass/Fail/Not)**	

S#	Prerequisites:
1	Have internet connection
2	Must submit in PDF format
3	
4	

S#	Test Data
1	Valid document format
2	
3	
4	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Enter submission page	Display submission page		
2	Click link for chosen chapter	Can click all the link		
3	Click add submission	Can click the add submission button		
4	Upload file	Upload file unsuccessful		
5	Click submit button	Unable to click submit button		

7.4 TC005: Test <View and Feedback> Subsystem: < View Submission and Give Feedback (UC004)>

This test contains the following test cases:

(a) TC005_01: Test <Scenario of sequence diagram (SD005)>

7.4.1 TC005_01: Test <Submission (SD004)>

This test contains the following scenarios:

(a) TC005_01_01: Test <Normal scenario of sequence diagram (SD005)>

Table 7.4 : TC005_01_01-<Normal Scenario of Sequence diagram (SD005)>

Test Case ID	TC005	**Test Case Discription**	View Submission and Give Feedback
Created By	Ain Batrisyia Binti Norazlan		
QA Tester's Log			
Date Tested	30/6/2023	**Test Case (Pass/Fail/**	

S#	Prerequisites:
1	Have internet connection
2	Must have submission from student
3	
4	

S#	Test Data
1	Valid there is submission from student
2	
3	
4	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Enter submission page	Display submission page		
2	Click on student name	Can click all the names		
3	View submission	Can view submission		
4	Click add feedback	Can add feedback		
5	Click submit button	Can click submit button		

(b) TC005_01_02: Test <Exception scenario of sequence diagram (SD005)>

Table 7.4 : TC005_01_02-<Exception Scenario of Sequence diagram (SD005)>

Test Case ID	TC005	**Test Case Discription**	View Submission and Give Feedback
Created By	Ain Batrisyia Binti Norazlan		
QA Tester's Log			
Date Tested	30/6/2023	**Test Case (Pass/Fail/**	

S#	Prerequisites:
1	Have internet connection
2	Must have submission from student
3	
4	

S#	Test Data
1	Valid there is submission from student
2	
3	
4	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Enter submission page	Display submission page		
2	Click on student name	Can click all the names		
3	View submission	Cannot view submission		
4	Click add feedback	Unable to add feedback		
5	Click submit button	Unable click submit button		

7.5 TC006: Test <Plagiarism Checker> Subsystem: <Check Plagiarism (UC006) & Submission of Checked Proposal (UC007)>

This test contains the following test cases:

(a) TC006_01: Test <Scenario of sequence diagram 6 (SD006)>

7.5.1 TC006_01: Test <state scenario of sequence diagram 6 (SD006)>

This test contains the following scenarios:

(a) TC006_01_01: Test <Normal scenario of sequence diagram 6 (SD006)>

Table 7.8 : TC006_01_01-<Normal Scenario of Sequence diagram 6 (SD006)>

Test Case ID	TC006	Test Case Discription	Check Plagiarism
Created By	Jeliza Justine A/P Sebastin		

QA Tester's Log

Date Tested	28/6/2023	Test Case (Pass/Fail/Not)	Pass

S#	Prerequisites:
1	The user is logged into the system
2	The user has a file to upload
	for plagiarism checking
3	

S#	Test Data
1	A document with percentage of matching is within the
	acceptable range
2	
3	

Test Scenario

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Click Plagiarism	Can click the menu		
	Checker menu			
2	Display file upload	Can display the section		
	section			
3	Click upload file	Can click the button		
4	Click "Check Plagiarism"	Can click the button		
5	Display percentage of	Can display the		
	matching detected	percentage and its sources		
	and its sources			
6	Display download option	Can display the option		
7	Enter submission page	Can enter the page		
8	Click add submission	Can click the button		
9	Upload file	Upload successfully		
10	Click submit button	Can click submit button		

(b) TC006_01_02: Test <Exception scenario of sequence diagram 6 (SD006)>

Table 7.9 : TC006_01_02-<Exception Scenario of Sequence diagram 6 (SD006)>

Test Case ID	TC006	Test Case Discription	Check Plagiarism
Created By	Jeliza Justine A/P Sebastin		

QA Tester's Log

Date Tested	28/6/2023	Test Case (Pass/Fail/Not)	Fail

Sil	Prerequisites:
1	The user is logged into the system
2	The user has a file to upload
	for plagiarism checking
3	

Sil	Test Data
1	A document with percentage of matching is higher than the set
	threshold
2	
3	

Test Scenario

Step Sil	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Click Plagiarism	Can click the menu		
	Checker menu			
2	Display file upload	Can display the section		
	section			
3	Click upload file	Can click the button		
4	Click "Check Plagiarism"	Can click the button		
5	Display percentage of	Can display the		
	matching detected	percentage and its sources		
	and its sources			
6	Display download option	Display failed		
7	Enter submission page	Cannot enter the page		
8	Click add submission	Cannot click the button		
9	Upload file	Upload failed		
10	Click submit button	Cannot click submit butto		

7.6 TC008: Test <Grading Module> Subsystem: < Grading(UC008) >

This test contains the following test cases:

(a) TC008_01: Test <Scenario of sequence diagram8 (SD008)>

(b) TC009_01: Test <Scenario of sequence diagram9 (SD009)>

(c) TC010_01: Test <Scenario of sequence diagram10 (SD010)>

(d) TC011_01: Test <Scenario of sequence diagram11 (SD011)>

This test contains the following scenarios:

(a) TC008_01: Test <Normal scenario of sequence diagram8 (SD008)>

TC008_01<Normal Scenario of Sequence Diagram (SD-008)>

Test Case ID	TC008_01	**Test Case Discription**	Grading
Created By	Aisyah Fatihah		

QA Tester's Log

Date Tested	28/6/2023	**Test Case (Pass/Fail/Not)**	Pass

S#	Prerequisites:
1	Log in to Fyp System
2	Student already made submission
3	
4	

S#	Test Data
1	Valid examiners id and password
2	Valid industrial coach id and password
3	Valid universities coach id and password
4	

Test Scenario Examiners,Industrial and University Coach do grading on students

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Access to grading interface	Grading interface display		
2	Choose subject code	Each subject code display		
3	Click view submission button	Documents submission display		
4	Click button for grading	Students details appear		
5	Enter grade for each subject	Student details and grade display		

(b) TC008_01_02: Test <Exception scenario of sequence diagram8(SD008)>

TC008_01_02<Exception Scenario of Sequence Diagram (SD-008)>

Test Case ID	TC008_01_02	Test Case Discription	Grading
Created By	Aisyah Fatihah		

QA Tester's Log

Date Tested	28/6/2023	Test Case (Pass/Fail/Not)	Fail

S#	Prerequisites:
1	Log in to Fyp System
2	Student do not made any submission
3	
4	

S#	Test Data
1	Valid examiners id and password
2	Valid industrial coach id and password
3	Valid universities coach id and password
4	

Test Scenario: Examiners,Industrial and University Coach do grading on students

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Access to grading interface	Grading interface display		
2	Choose subject code	Each subject code display		
3	Click view submission button	Documents submission displa		
4	Click button for grading	Students details do not appea		
5	Grading declined	Back to homepage		

(c) TC009_01: Test <Normal scenario of sequence diagram9 (SD009)>

TC009_01<Normal Scenario of Sequence Diagram (SD-009)>

Test Case ID	TC009_01	Test Case Discription	Recheck Grading
Created By	Aisyah Fatihah		

QA Tester's Log

Date Tested	28/6/2023	Test Case (Pass/Fail/Not)	Pass

S#	Prerequisites:
1	Log in to Fyp System
2	Student already made submission
3	Grading have been made
4	

S#	Test Data
1	Valid examiners id and password
2	Valid industrial coach id and password
3	Valid university id and password
4	

Test Scenario: Examiners,Industrial and University Coach do grading on students

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Access to grading interface	Grading interface display		
2	Choose subject code	Fyp subject code display		
3	Click grading button	Subject grade display		
4	Click regrading button	Students grade appear		
5	Change to new grade	New grade recorded		

(d) TC010_01: Test <Normal scenario of sequence diagram10 (SD010)>

TC010_01<Normal Scenario of Sequence Diagram (SD-010)>

Test Case ID	TC010_01	Test Case Discription	Release Grading
Created By	Aisyah Fatihah		

QA Tester's Log

Date Tested	28/6/2023	Test Case (Pass/Fail/Not)	Pass

S#	Prerequisites:
1	Log in to Fyp System
2	Student already made submission
3	Grading have been made
4	

S#	Test Data
1	Valid coordinators id and password
2	
3	
4	

Test Scenario: Coordinators release students' official grade result

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Access to grading interface	Grading interface display		
2	Choose subject code	Each subject code displa		
3	Click grading button	Subject grade display		
4	Cross checking the grad	Grade display by each rol		
5	Click release button	Official grade appear		

(e) TC011_01: Test <Normal scenario of sequence diagram11(SD011)>

TC011_01<Normal Scenario of Sequence Diagram (SD-011)>

Test Case ID	TC011_01	Test Case Discription	View Grading
Created By	Aisyah Fatihah		

QA Tester's Log

Date Tested	28/6/2023	Test Case (Pass/Fail/Not)	Pass

S#	Prerequisites:
1	Log in to Fyp System
2	Student already made submission
3	Grading have been made
4	

S#	Test Data
1	Valid student id and password
2	
3	
4	

Test Scenario: Student view grade result

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Access to grading interface	Grading interface display		
2	Choose subject code	Each subject code display		
3	Click grading button	Subject grade display		

(f) TC011_01_02: Test <Exception scenario of sequence diagram11(SD011)>

TC011_01_02<Exception Scenario of Sequence Diagram (SD-011)>

Test Case ID	TC011_01_02	Test Case Discription	View Grading
Created By	Aisyah Fatihah		

QA Tester's Log

Date Tested	28/6/2023	Test Case (Pass/Fail/Not)	Fail

S#	Prerequisites:
1	Log in to Fyp System
2	Student already made submission
3	Coordinator do not realise any grading
4	

S#	Test Data
1	Valid student id and password
2	
3	
4	

Test Scenario: Student view grade result

Step S#	Step Details	Expected Results	Actual Results	Pass/Fail/Not executed/Suspended
1	Access to grading interface	Grading interface display		
2	Choose subject code	Each subject code display		
3	Click grading button	Subject grade do not display		