SECJ2203: Software Engineering

System Documentation (SD)

Final Year Project (Maniacs System)

Version 2

Date: 16 JUN 2022

School of Computing, Faculty of Engineering

Prepared by: <Software Maniacs>

Revision Page

a. Overview

The current version of SD contains the System Architectural Design of Maniac System, a full description of the components with a complete package diagram and a detailed description of each subsystem with a complete package diagram. Aside from that, there's data design which comprises data descriptions and dictionaries for all of the entities in the database. The user interface design has also been described, with an overview of the user interface and screenshots.

b. Target Audience

Target Audience: Project Developer and Team

Stakeholders: Coordinators, Examiners, Industry Coaches, University Coaches and Students

c. Project Team Members

List the team members in a table by stating their roles and the status for each assigned task e.g. by sections for this SD version (complete, partially complete, incomplete). If the assigned tasks are not done and have been assigned to other team members, state accordingly.

Member Name	Role	Task	Status
MOHD FIRDAUS BIN ZAMRI	Group Leader	Details description of Communication Management Subsystem 1. Package diagram for <Communication Management> Subsystem 2. Class Diagram for <Communication Management> Subsystem 3. Sequence Diagram for <Contact Coaches scenario> 4. Interface Design 5. Check and suggest improvements to all parts, especially the use cases and graphics.	Completed
LUE GUO MING	Group Member	Details description for <Submission Management> Subsystem	Completed

		1. Class diagram for <Submission Management> Subsystem 2. Sequence diagram for <Create Submission Platform scenario> 3. Sequence diagram for <Change Time Date of Submission scenario> 4. Sequence diagram for <Make Submission scenario> 5. Sequence diagram for <View Submission scenario> 6. Sequence diagram for <Download Submission scenario> 7. Package Diagram for <Submission Management Subsystem> 8. Interface Design	
NUR HAZNIRAH BINTI HAZMAN	Group Member	Details description for <Notification Management> Subsystem 1. Class diagram for <Notification Management> Subsystem 2. Sequence diagram for <View Notification scenario> 3. Sequence diagram for <Delete Notification Scenario> 4. Package Diagram for <Notification Management> Subsystem 5. Overall Package Diagram 6. Component Model 7. Data Design	Completed
ERICA DESIRAE MAURITIUS	Group Member	Details Description for <Grading Management> Subsystem 1. Class Diagram for <Grading Management Subsystem> 2. Sequence diagram for <Grading Scenario> 3. Sequence diagram for <Recheck Grading Scenario> 4. Sequence diagram for <Release Grading Scenario>	Completed

		5. Sequence diagram for <View Grading Scenario> 6. Package Diagram for <Grading Management Subsystem> 7. Architecture Style and Rationale 8. Data Design	

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Team leader 1 (Mohd Firdaus Bin Zamri)	Completed version 1.0, until 2.5 Software System Attributes	05/06/2022
2.0	Team leader 1 (Mohd Firdaus Bin Zamri)	Completed version 2.0, until 6.0 Interface Design	16/06/2022

Note:
This System Documentation (SD) template is adapted from IEEE Recommended Practice for Software Requirements Specification (SRS) (IEEE Std. 830-1998), Software Design Descriptions (SDD) (IEEE Std. 1016-1998 1), and Software Test Documentation (IEEE Std. 829-2008) that are simplified and customized to meet the need of SECJ2203 course at School of Computing, UTM. Examples of models are from Arlow and Neustadt (2002) and other sources stated accordingly.

Table of Contents

1 Introduction 3-7

1.1 Purpose 3

1.2 Scope 4-5

1.3 Definitions, Acronyms and Abbreviations 6-7

1.4 References 7

1.5 Overview 7

2 Specific Requirements 8-55

2.1 External Interface Requirements 8-12

2.1.1 User Interfaces 8-9

2.1.2 Hardware Interfaces 10

2.1.3 Software Interfaces 10-11

2.1.4 Communication Interfaces 12

2.2 System Features 13-55

2.2.1 UC001: Use Case <View Notification> 16-18

2.2.2 UC002: Use Case <Delete Notification> 19-21

2.2.3 UC003: Use Case <Create Submission Platform> 22-24

2.2.4 UC004: Use Case <Change Time Date of Submission> 25-27

2.2.5 UC005: Use Case <Make Submission> 28-30

2.2.6 UC006: Use Case <ViewSubmission> 31-33

2.2.7 UC007: Use Case <DownloadSubmission> 34-36

2.2.8 UC008: Use Case <Grading> 37-39

2.2.9 UC009: Use Case <RecheckGrading> 40-42

2.2.10 UC010: Use Case <ReleaseGrading> 43-45

2.2.11 UC011: Use Case <ViewGrading> 46-48

2.2.12 UC012: Use Case <Contact Coaches> 49-51

2.3 Performance and Other Requirements 52

2.4 Design Constraints 53

2.5 Software System Attributes 54-55

3 System Architectural Design 56-59

3.1 Architectural Style and Rationale 56-57

3.2 Component Model 58-59

4 Detailed Description of Components 60-84

4.1 Complete Package Diagram 60

4.2 Detailed Description 61-84

4.2.1 P001: <Notification Management> Subsystem 61-65

4.2.2 P002: <Submission Management> Subsystem 66-74

4.2.3 P003: <Grading Management> Subsystem 75-82

4.2.4 P004:<Communication Management> Subsystem 83-84

5 Data Design 85-91

5.1 Data Description 85-87

5.2 Data Dictionary 88-91

6 Interface Design 92-110

6.1 Overview of Interface 92-110

1. Introduction

1.1. Purpose

This SD describes a combined documentation of the Software Requirement Specification (SRS), Software Design Document (SDD), and Software Test Description (STD). The primary purpose of this system and system documentation (SD) is to depict the system, as well as what it contains and how it interacts. The software requirement specifications (SRS), on the other hand, will demonstrate the functionality of the system, as well as what the system needs, as well as the tables that will be utilized and how it should be accomplished. Moreover, the SRS will provide a thorough understanding of what can be expected from the newly introduced system that will be built as a good understanding of the system and its features will allow the right software to be produced for the end user and will be utilized to build the project's future phases.[1]

The Software Test Description (STD) document provides the test preparations, test cases, and test methods that will be used to platform qualification testing the capabilities of a Computer Software Configuration Item (CSCI) defined in the Software Requirements Specification (SRS) and Software Design Document (SDD).[2]

Software Design Document (SDD) outlines the architecture and detailed design of the Maniac System to ease analysis, planning, implementation, and decision making. The design description is a method for expressing software design information and may be thought of as a blueprint or system model. An SDD is typically used in conjunction with an architectural diagram to provide pointers to the feature specifications of smaller elements of the design. In practice, the description is essential to manage a huge team around a unified vision, to be a reliable reference, and to explain all components of the programme and how they will function.[3]

The intended audience for this SD is the project developer and his/her team, whereas the stakeholders of the system are Coordinators, Examiners, Industry Coaches, University Coaches and Students. This SD can be utilized in any situation involving the project's needs and the solutions chosen. Finally, the document would offer a clear picture of the system that is being built.

1.2. Scope

The software product is named as Maniacs System. This software product is developed for the management and activities of students' final year projects with industry (FYP-i/PSM-i). The current FYP-i system is not mature enough as users conduct processes in several platforms, causing difficulties to communicate, and make mistakes in their work and progress.

Within the Maniacs System, the goal is to invent an interactive transaction-based software (web-based application) that is eligible to several users. Users who are able to use Maniacs System should involve the **PSM-i coordinators**, **students**, **industry and university coaches** and **examiners**. Generally, the system should provide all necessary information and guidance for users to complete their works. Moreover, it should contain multiple functions for different types of users. The information and guidance should be stated clearly in the form of announcement and notification by coordinators and be displayed in the notification interface. The system should not display irrelevant announcements for different users such as remind students to do a grading. The coordinator should have the authority to provide different submission links for different courses so students can submit and upload their FYP-i works to the system. The system should classify the course code and user ID for proper submission and database management. The system should not delete or remove a database data without confirmation of the user. After a submission is made, an examiner should be able to access the submitted file through the system submission interface. The system should not allow the examiner to delete or remove submitted files. The examiner, industry and university coaches can do grading and recheck grading to a student, the system should be able to differentiate the tasks and roles of examiners and coaches and provide a different grading interface. The system should provide a communication channel for coaches and students to make sure they can interact with each other throughout the FYP-i period.

System Scope include:

Make Announcement/Notification

- Coordinator write and display notification.
- Coordinator delete notification.
- Coordinator, student, industry and university coaches view notification.

Make submission

- Coordinator creates a submission platform with course code.
- Coordinator change submission date.
- Students make submissions based on course code.
- Students remove submissions based on course code.
- Examiner view submissions.

Do Grading

- Examiners do grading based on course submissions.
- Industry and university coaches do grading based on industrial training performance.
- Examiners, industry and university coaches recheck and remark for grading.
- Examiners release grading.
- Students view grading.

Communication

- Students send messages to industry and university coaches.
- Industry and university coaches send messages to students.

The benefits and objectives to propose such a system are:

- ➢ To improve the reliability of information.
- ➢ To provide formal, updated and detailed information to all users on time.
- ➢ To aid coordinator in lessening their burden during the final year project assessment.
- ➢ To reduce the storage of data due to redundancy of information.
- ➢ To ensure effective communication among different users.
- ➢ To ease the classification and management of several documents.

1.3. **Definitions, Acronyms and Abbreviation**

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

Term	**Definitions**
SD	System Documentation
SRS	Software Requirement Specification
SDD	Software Design Description
STD	Software Test Description
FYP-i/PSM-i	Final Year Projects with Industry
Maniac	The proposed system
Coordinators	The person who manages the whole system and has access to every platform.
Examiner	The person who does the grading of students' papers.
Industry Coach	The person who does the grading of students' skills.
University Coach	The person who does the grading of students' skills.
Students	The person who has the least ranking among stakeholders.
Domain Name System (DNS)	A critical component of the internet, serving as a means of matching names (the website you're looking for) to numbers (the address for the website).
File Transfer Protocol (FTP)	A standard communication protocol is used to deliver computer files from a server to a client on a computer network
Hypertext Transfer Protocol (HTTP)	Application-layer protocol used to send hypermedia documents like HTML.
Internet Protocol (IP)	A set of standards that governs the format of data transmitted via the internet or a local network.
Transmission Control Protocol/Internet Protocol (TCP/IP)	Protocols for Internet communication that allow digital computers to connect across long distances.

MQTT (Message Queuing Telemetry Transport)	An OASIS communications standard for the Internet of Things (IoT)

1.4. **References**

[1] "srs document for hotel management system." [Online]. Available: https://www.academia.edu/10313728/srs_document_for_hotel_management_system

[2] "1. SOFTWARE TEST DESCRIPTION (STD)." [Online]. Available: http://simonjwright.users.sourceforge.net/pushface.org/mil_498/std-did.htm#:~:text=3.1%20The%20Software%20Test%20Description%20%28STD%29%20describes%20the,adequacy%20of%20the%20qualification%20testing%20to%20be%20performed.

[3] "Software Design Documentation (SDD)." [Online]. Available: https://ecomputernotes.com/software-engineering/softwaredesigndocumentation

1.5. **Overview**

The system documentation will also include the Maniac system's Specific Requirement, as well as its external interface, hardware, and software interfaces. Moreover, the system documentation also contains functional requirements for the system, which include Use Case Diagram, Use Case Descriptions, Sequence Diagrams, Activity Diagrams and Domain Model that describe how each class interacts in the system.

The SD is organized by the team members of Maniac System, where each member has been assigned to their own responsibility, where some are responsible to create a new system, some are responsible to demonstrate how the combination of hardware and software may meet the requirements of the stakeholders when using the Maniac System.

2.Specific Requirements

2.1.External Interface Requirements

2.1.1.User Interfaces

Notification Interface

The types of user involved in the notification interface are coordinators, students, industry and university coaches. All users can view notifications but only coordinators have the authority to create, modify and delete contents of the notifications. The system should only display the bottoms and input boxes for users to create, modify and delete notifications when the type of user is coordinator. The system will ask for confirmation if a coordinator manages to delete a notification. Generally, all types of users can view the notification by entering the notification interface. To make the contents of notification be expressed clearly to all users, the system can classify the type of notification with user type so the user can notice which notifications are related and important to them.

Submission Interface

The submission interface should be accessed by coordinators, students and examiners. In this interface, different types of users have different roles and tasks. Based on the types of users, the system will provide different functions for users to complete their tasks. At first, the system allows the coordinators to create new submission platforms. The coordinator needs to identify the course code, time and due date, submission file type of a submission platform. Every submission platform is independent and has its own database. The system will also provide a feature for the coordinator to change time and due date of an existing submission platform. After a submission platform is created, the system should display the submission platform to students. The system should not allow students to create, modify and delete a submission platform. The task of students in the submission interface is to upload and submit one or more files to a submission platform from valid drives. The system will validate the submitted files and save them to databases based on its course code. After a student successfully completes a submission, the student's submission status will also change to "Submitted" by the system.

After a file is submitted and saved to the databases, the system will allow examiner users to access the file. The examiners can view and download the submitted files from the submission interface, and do further grading based on the files in the grading interface.

Grading Interface

The end users involved in this Grading Interface are Examiners, University Coaches and Industry Coaches, Coordinators as well as Students. The system should allow Examiners as well as University Coaches and Industry Coaches to grade the students' works and students' respectively. For Examiner, they had to open the downloaded submission file of the students, whereas for University Coaches and Industry Coaches, the system will provide an assessment form for them. Moreover, they have to do a recheck grading after they have graded each submission file and assessment form to have a fair and square grade for each student. After the rechecking of grading, Examiner and University Coaches as well as Industry Coaches will submit the grading to the system. Not to forget about Coordinator, the system only allows them to release the official Grade to the Grading Interface. Moreover, the Coordinator has to compare the recheck grades of each submission file of the students and come out with a final grade. Also, they have to write a report about the grades of students. On the other hand, for Students, the system allows them to only view their Grades only if they have entered the correct subject code for that particular subject that they wish to see.

Communication Interface

The Communication Interface involves Students and University Coaches as well Industry Coaches. Students are allowed to exchange messages with University Coaches as well Industry Coaches especially when Students have problems with their Final Project assessment. The end users involved can type their message in the chat box and click the send button. The system will save the messages that have been sent by the end users so for the end users to refer back to the messages especially for students when they asked the coaches about the details or criteria of the assessments.

2.1.2. **Hardware Interfaces**

The hardware that supports this system aids in optimizing its use. The system will be adaptable for the user to use in a variety of devices such as smartphones, laptops and desktops. The students can answer their assessment both offline and online. When the students are online, they can download as well as submit their assessment to the respected interface of the system. For Examiner and Industry Coaches as well as Industry Coaches, they can do the grading both online and offline as long as they have the material needed. On the other hand, for Coordinators, they can do the report both online and offline. However, they can only release the grading when they are online. Google Chrome, Firefox, Opera, Microsoft Edge can support or run the system.

2.1.3. **Software Interfaces**

1. Software Product: Data management system

Name: Oracle Big Data SQL Cloud Service
Mnemonic: -
Specification number: N/A
Version number: Oracle Database 21c 21.1
Source: https://docs.oracle.com/en-us/iaas/Content/bigdata/cloud-sql.htm
Interface: Notification, submission, grading and communication database
Description: Database management and storage either on traditional local drive or cloud platform. Databases include the notification content, submission and grading document, communication message and other information details. Databases can be well protected and stored in the software system.

2. Software Product: Operating system

Name: Windows 10 Internet of Thing Core
Mnemonic: Windows 10 IoT Core
Specification number: SKU 6F6-00037
Version number: Build 19042.572
Source: https://docs.microsoft.com/en-us/windows/iot/iot-enterprise/features
Interface: Drive

Description: Windows 10 Internet of Thing Core is the replacement of the embedded Windows OS. Operating system is the basic component and platform to manage system hardware, software resources, and common services for computer programs. It also used to download (input), upload (output) files and act as a local drive.

3. Software Product: Web browser

Name: Google Chrome /Firefox /Opera /Microsoft Edge

Mnemonic: -

Specification number: N/A

Version number: Latest Version

Source: N/A

Interface: Cloud service

Description: Web browser is the necessary platform for web-based software systems. It can access the system and data from the online server through the Internet. By using a browser, the system does not allocate much storage and memory of the local drive.

2.1.4.**Communication Interfaces**

MQTT and HTTP are two communication protocols that the cloud allows for communication and connectivity. These devices can communicate via Cloud IoT Core over a bridge that uses either HTTP or MQTT. These bridges are essential components of Cloud IoT Core. When you build a device registry, you can select protocols such as HTTP, MQTT, and others. MQTT is a popular standard protocol that is supported by embedded devices. This protocol is widely used in machine-to-machine interactions.

Since HTTP is a connectionless protocol, devices do not keep in touch with the cloud IoT core. As an alternative, people can send requests and get answers. Cloud computing protocols are a collection of rules that allow two electronic elements to connect and share data with one another. It is mostly used for communication, storage, encryption, networks, decryption, security, and user login management. Moreover, HTTP is also used to communicate between the system service and the web servers. When we execute it in browsers, it will connect with web servers over HTTP.

Furthermore, the Domain Name System (DNS) is a critical component of the internet, serving as a means of matching names (the website you're looking for) to numbers (the address for the website). Anything linked to the internet - computers, tablets, mobile phones, and websites - has a numerical Internet Protocol (IP) address. When searching our system, this will provide more reliable communication. In the TCP/IP protocol, FTP is used to distribute files across networks. This is utilized when the user wishes to upload and download a file.

2.2.**System Features**

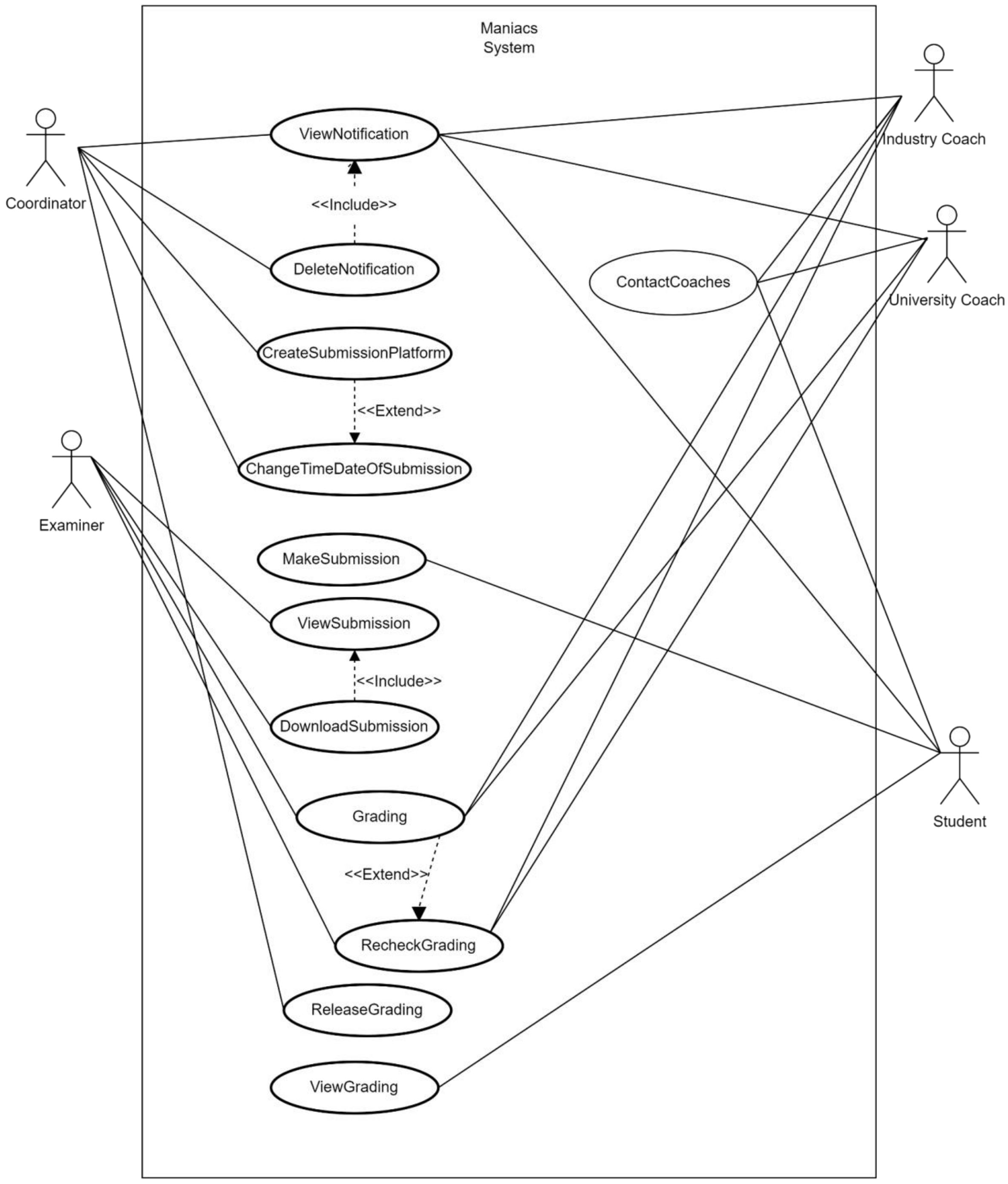


Figure 2.1: Use Case Diagram for <Maniac System>

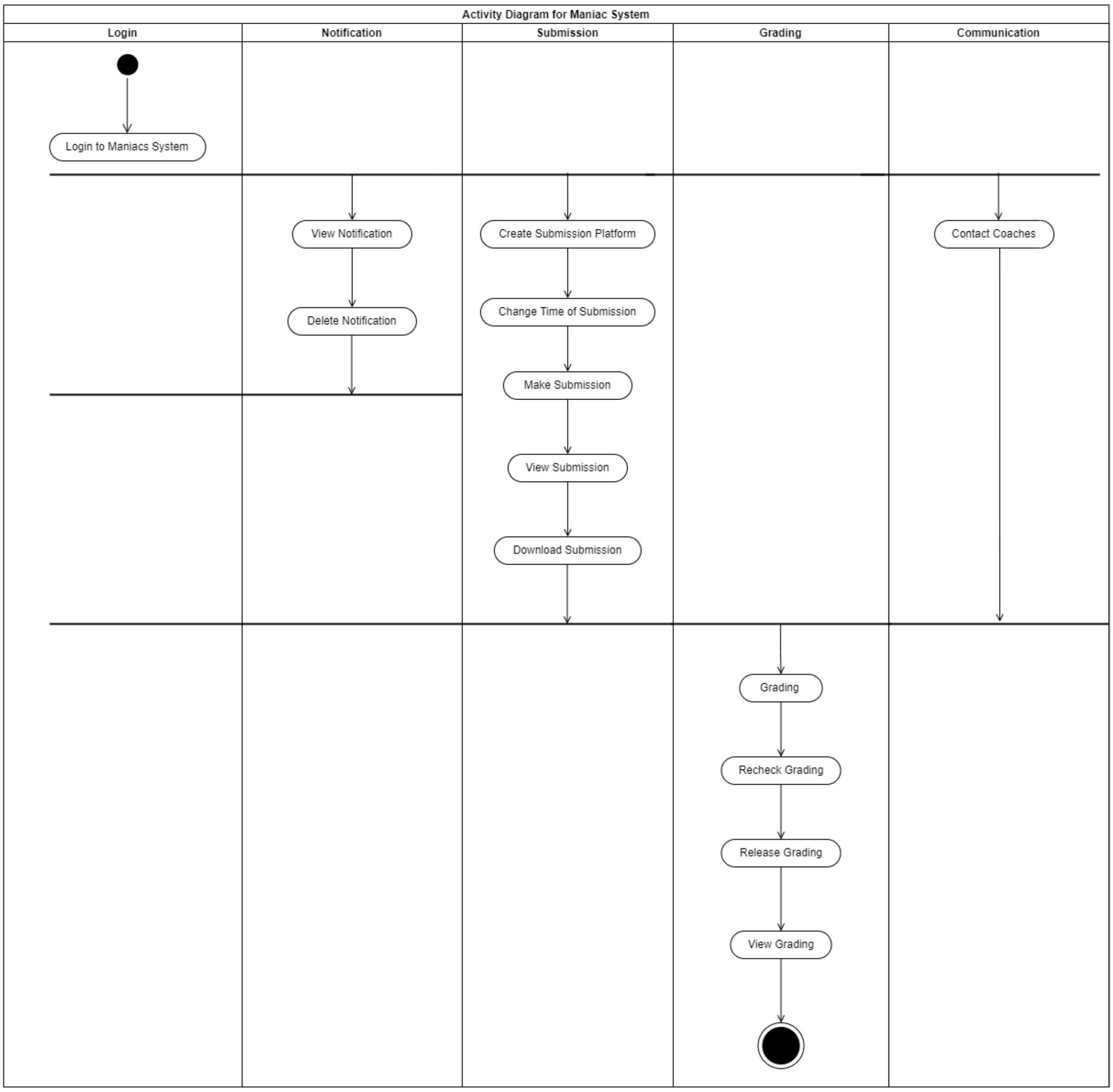


Figure 2.2: Activity Diagram for <Maniac System>

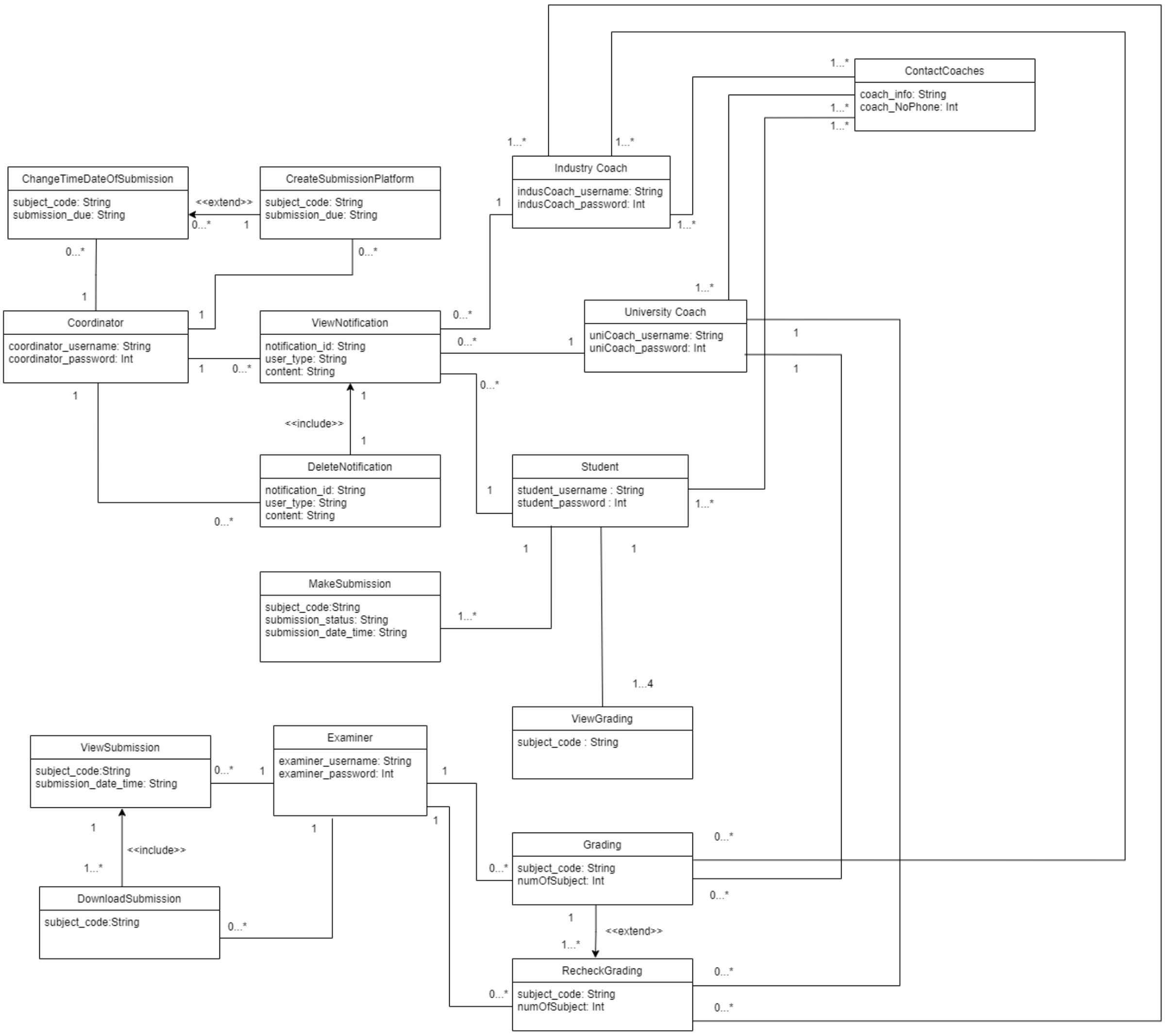


Figure 2.3: Domain Model for <Maniac System>

2.2.1.**UC001: Use Case <View Notification>**

Use case: View Notification
ID: UC1
Actors: Coordinator, University coaches, Industry coaches, Students
Preconditions: A valid Coordinator, University Coach, Industry Coach or Student username and password in order to log in to the system.
Flow of events: 1. The Coordinator, University coach, Industry coach or Student log into the system 2. The Coordinator, University Coach, Industry Coach or Student opens the Notification interface. 3. The notification is displayed in list form whereby the latest notification will be on the top of list.

Table 2.1: Use Case Description for <View Notification>

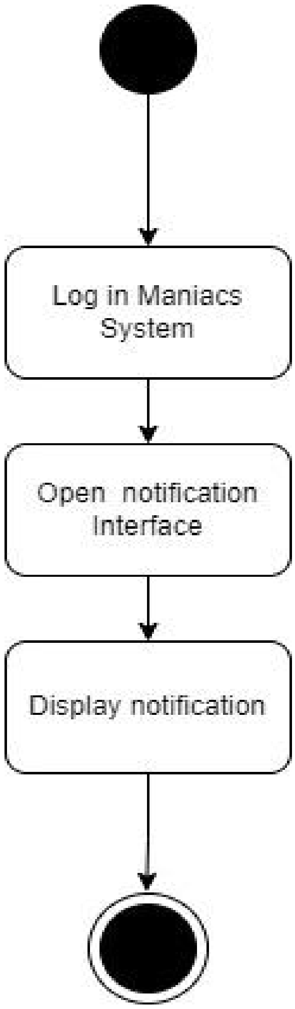


Figure 2.4: Activity Diagram for <View Notification>

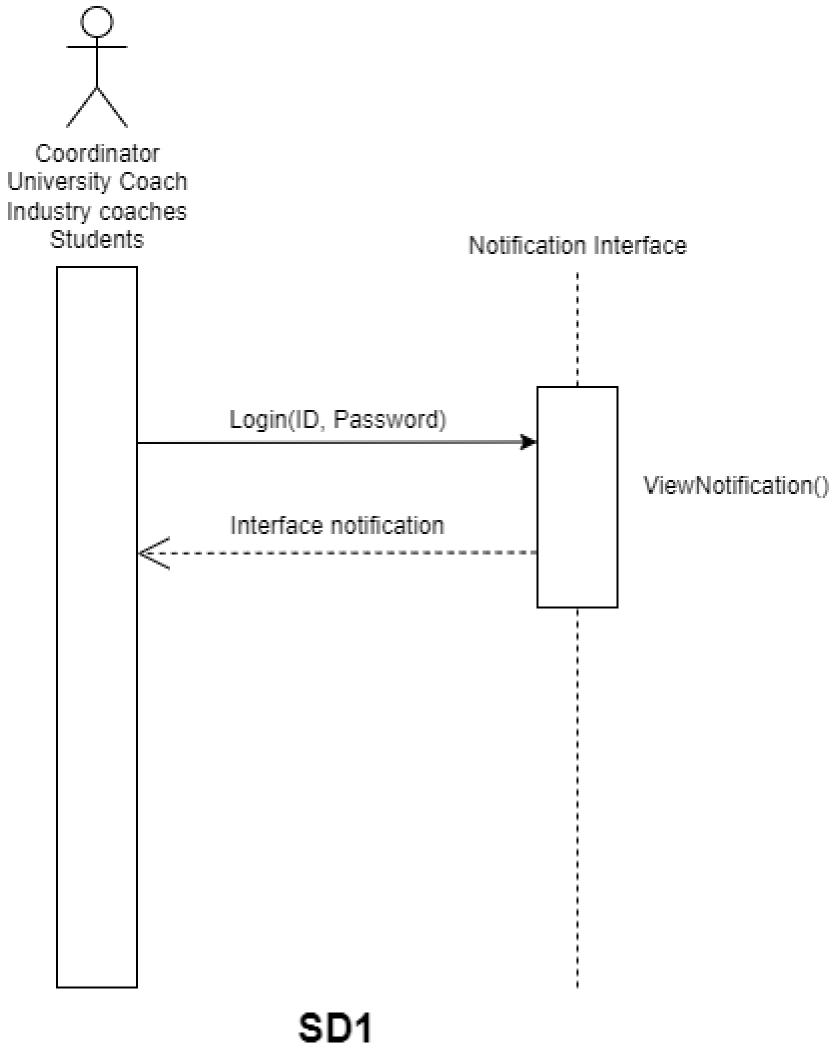


Figure 2.5: Sequence Diagram for <View Notification>

2.2.2.UC002: Use Case <Delete Notification>

Use case: Delete Notification
ID: UC2
Actors: Coordinator
Includes: UC1 View Notification
Preconditions: A valid Coordinator username and password to log into the system
Flow of events: 1. The Coordinator, University coach, Industry coach or Student log into the system 2. The Coordinator, University Coach, Industry Coach or Student opens the Notification interface. 3. include (View Notification) 4. The coordinator clicks on the select icon and clicks on the notifications that they want to delete. 5. The coordinator clicks on the delete button to delete the notifications from the notification list. 6. The system displays a confirmation message. 7. If Coordinator confirms the deletion or click yes ..The system will delete the notification. 8. Else ..The system will cancel the selection and deletion
Postconditions: The notification is being deleted and the notification list is being updated.

Table 2.2: Use Case Description for <Delete Notification>

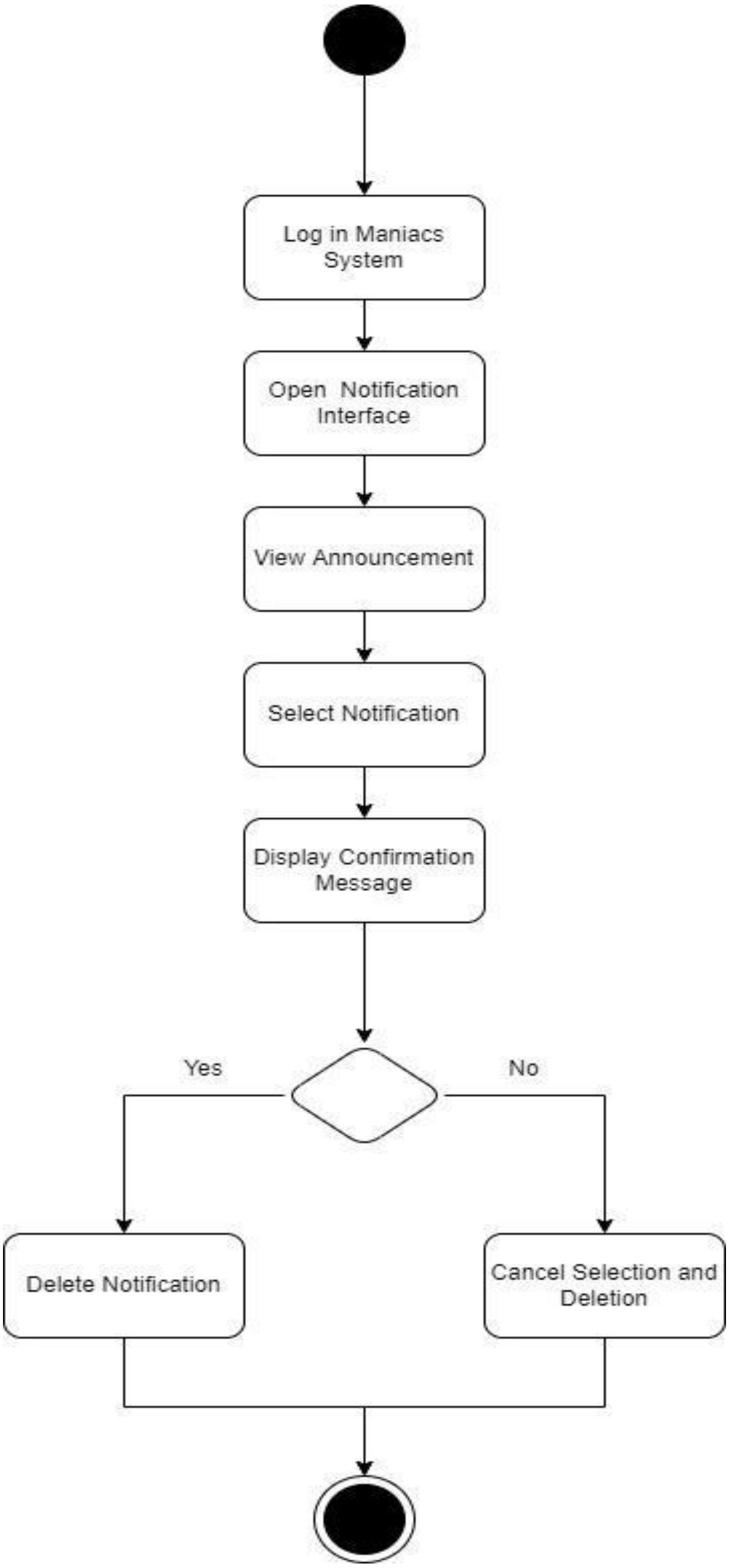

Log in Maniacs System
Open Notification Interface
View Announcement
Select Notification
Display Confirmation Message
Yes
No
Delete Notification
Cancel Selection and Deletion

Figure 2.6: Activity Diagram for <Delete Notification>

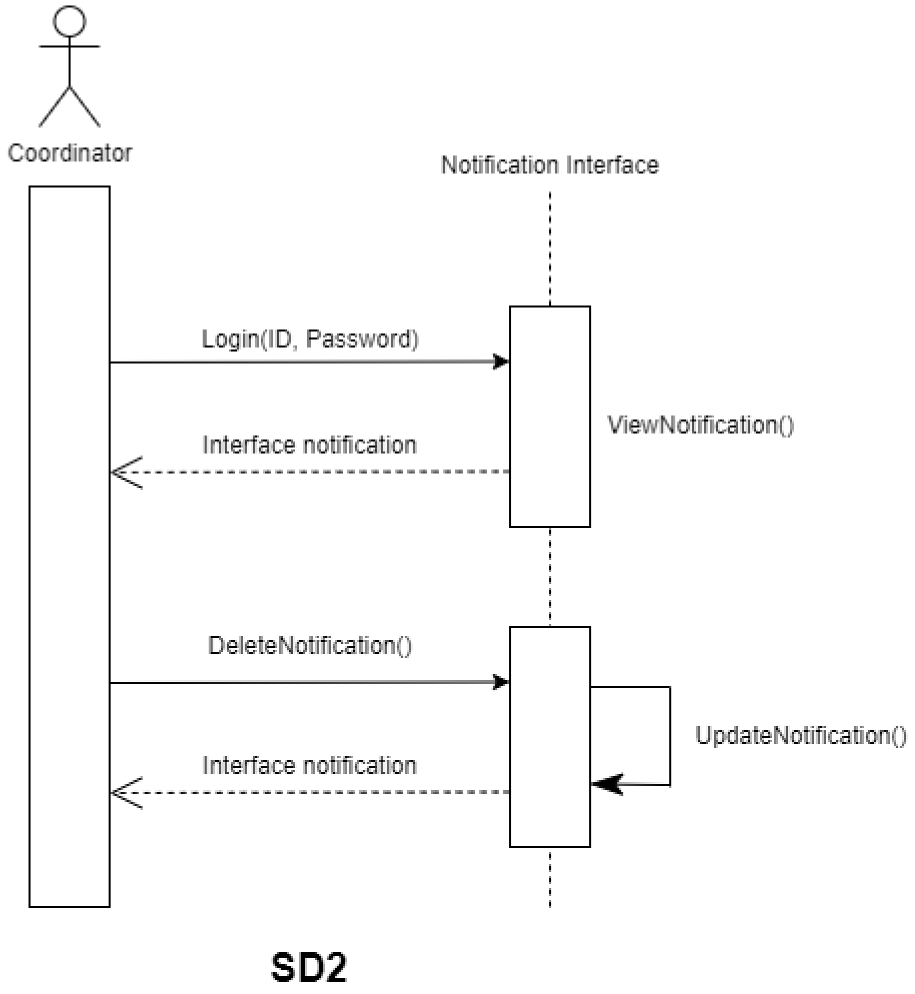


Figure 2.7: Sequence Diagram for <Delete Notification>

2.2.3.**UC003: Use Case <Create Submission Platform>**

Use case: Create Submission Platform
ID: UC3
Actors: Coordinator
Extension Point: <ChangeTimeDateOfSubmission>
Preconditions: A valid Coordinator username and password to log into the system.
Flow of events: 1.The Coordinator logs into the system. 2.The Coordinator opens the Submission interface. 3.The Coordinator selects to create a new submission platform. 4.The Coordinator fills in submission details/description. 5.The Coordinator selects the submission file type and number. 6.The Coordinator selects the date and time of submission. 7.The Coordinator confirms to create the submission platform.
Postconditions: Students can access a submission platform.

Table 2.3: Use Case Description for <Create Submission Platform>

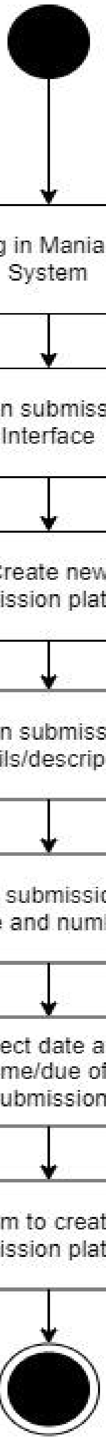
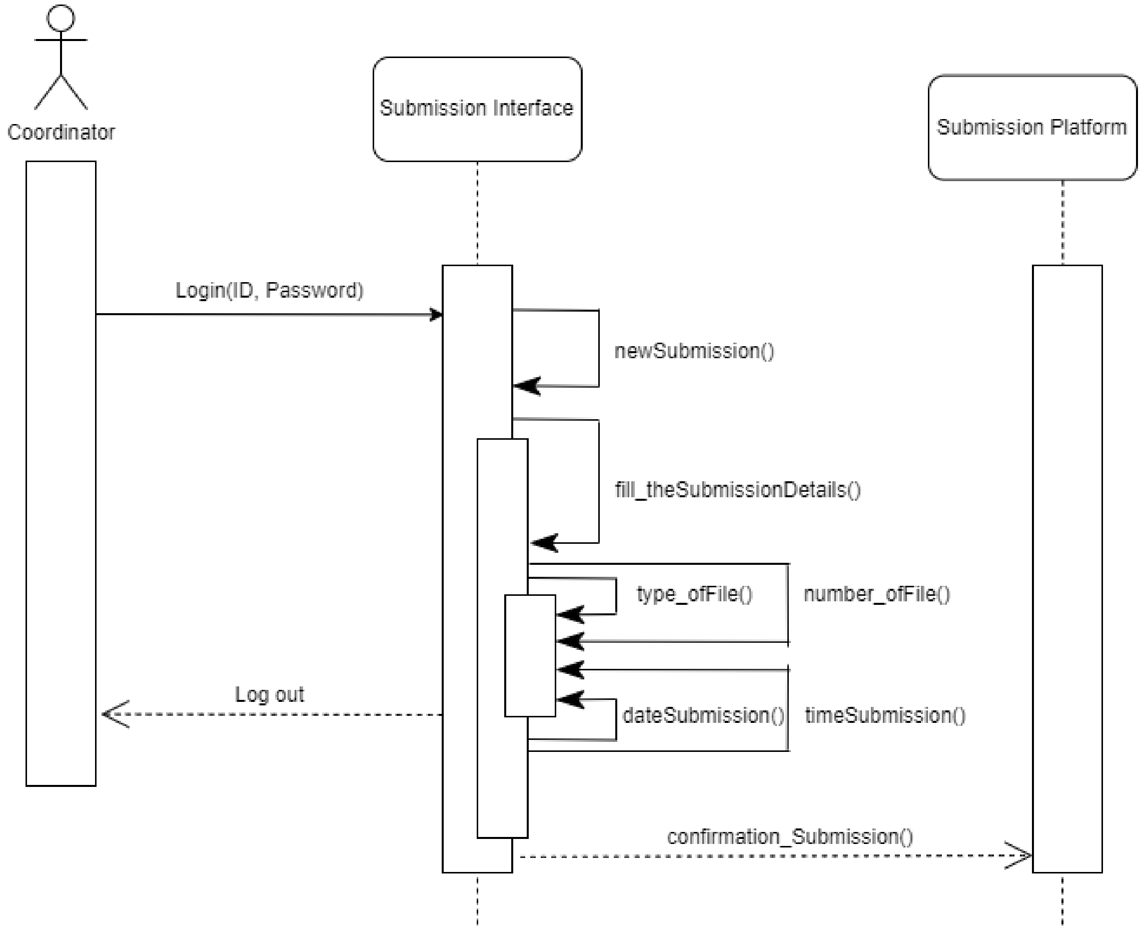

Log in Maniacs System
Open submission Interface
Create new submission platform
Fill in submission details/description
Select submission file type and number
Select date and time/due of submission
Confirm to create the submission platform

Figure 2.8: Activity Diagram for <Create Submission Platform>



SD3

Figure 2.9: Sequence Diagram for <Create Submission Platform>

2.2.4.UC004: Use Case <Change Time Date of Submission>

Use case: Change Time Date of Submission
ID: UC4
Actors: Coordinator
Extends: UC3 CreateSubmissionPlatform
Preconditions: A valid Coordinator username and password to log into the system. A valid submission platform is created.
Flow of events: 1.The Coordinator selects to edit the submission platform. 2.The Coordinator set a new date and time for submission. 3.The system display the confirmation message 4.If the Coordinator select “Save and make change” 1.System sets a new date and time for the submission platform. 5.If the Coordinator select “Cancel” 1.System remains the original date and time for the submission platform.
Postconditions: The time and date of a submission platform is updated.

Table 2.4: Use Case Description for <Change Time Date of Submission>

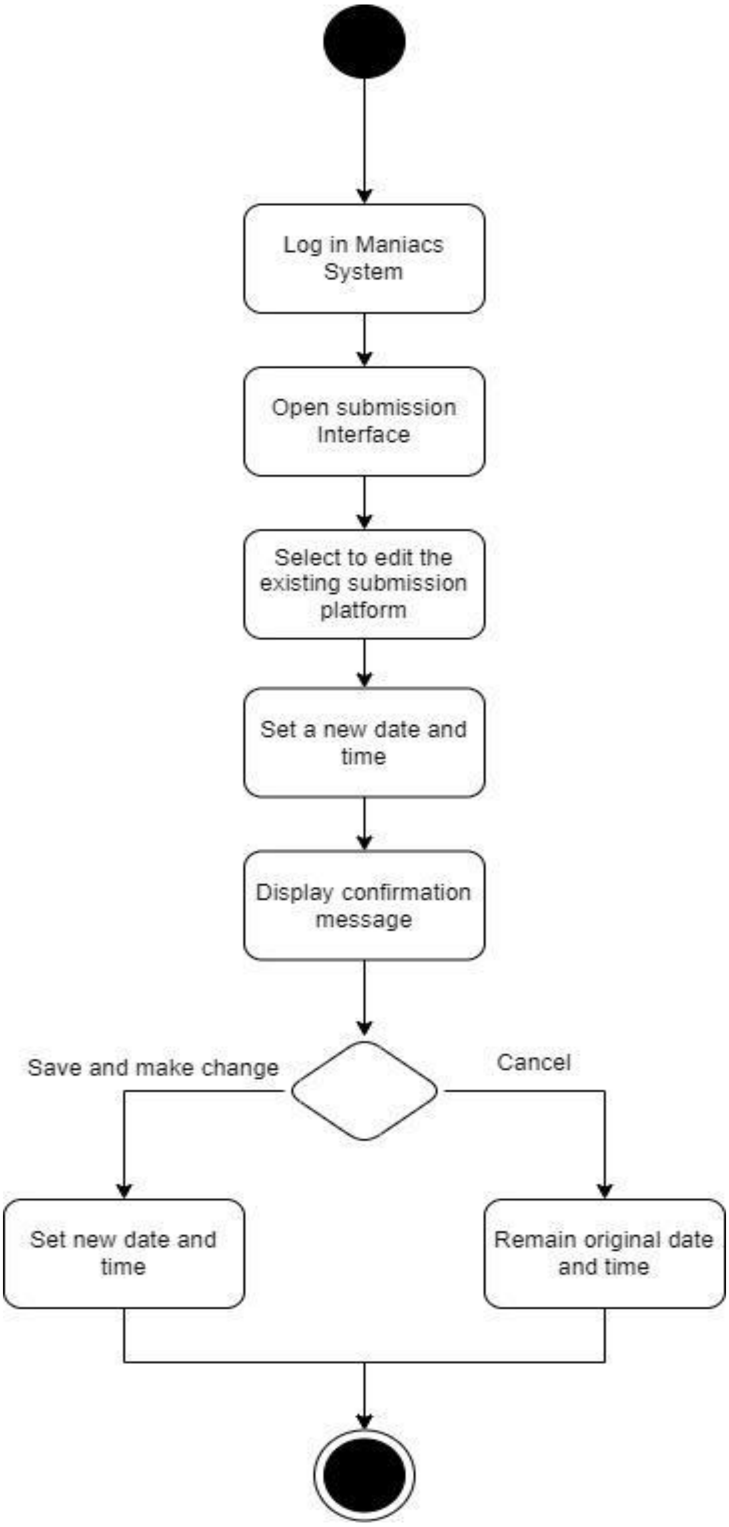
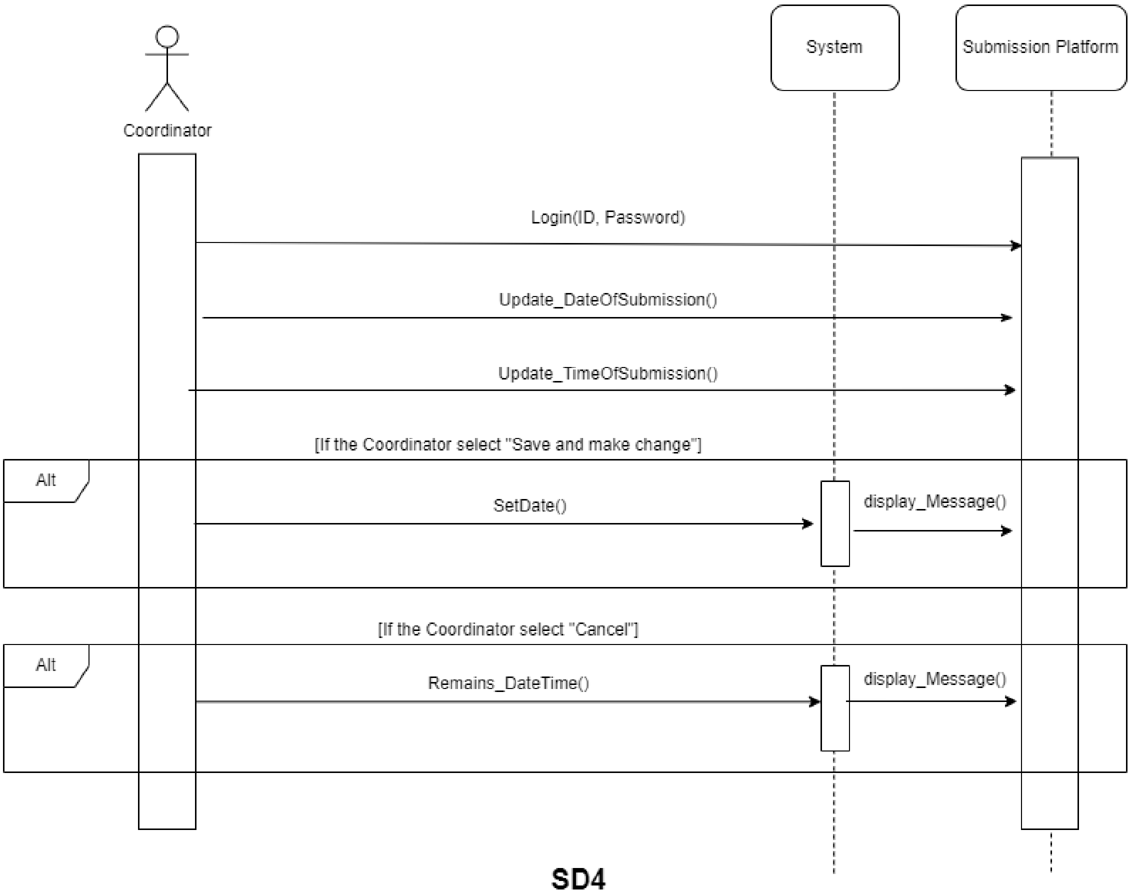

Log in Maniacs System
Open submission Interface
Select to edit the existing submission platform
Set a new date and time
Display confirmation message
Save and make change
Cancel
Set new date and time
Remain original date and time

Figure 2.10: Activity Diagram for <Change Time Date of Submission>



Coordinator
System
Submission Platform
Login(ID, Password)
Update_DateOfSubmission()
Update_TimeOfSubmission()
[If the Coordinator select "Save and make change"]
Alt
SetDate()
display_Message()
[If the Coordinator select "Cancel"]
Alt
Remains_DateTime()
display_Message()
SD4

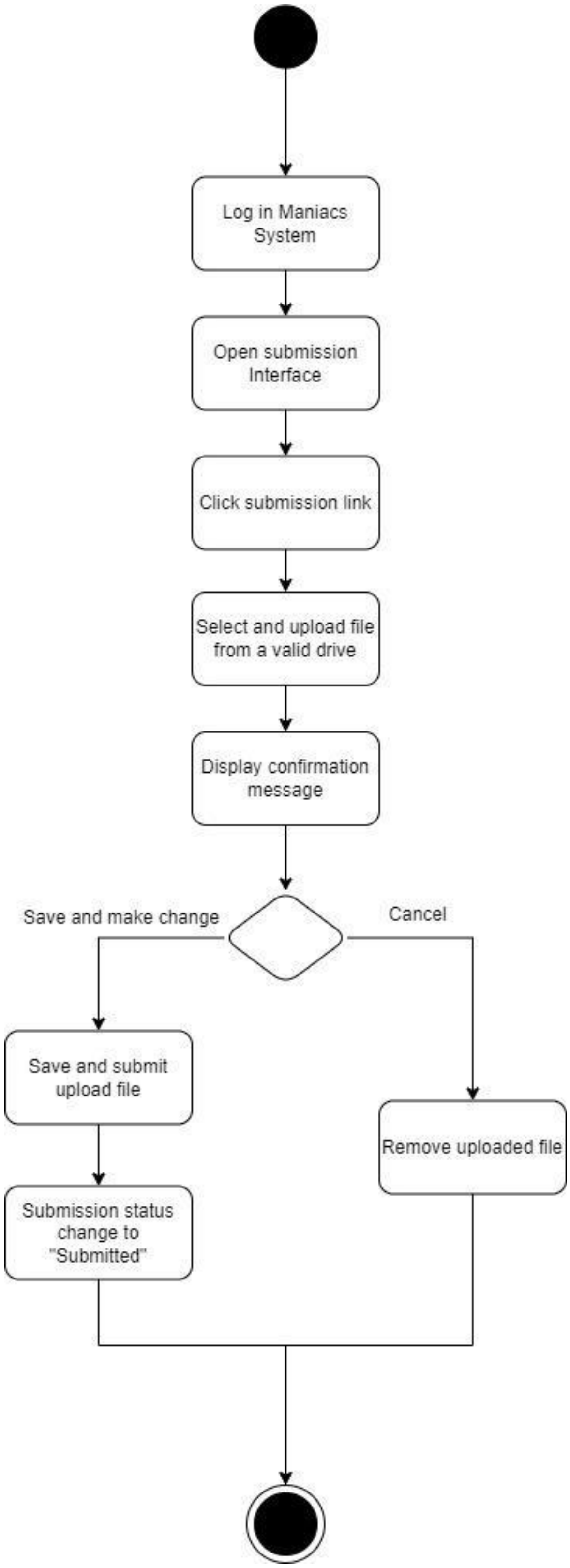
Figure 2.11: Sequence Diagram for <Change Time Date of Submission>

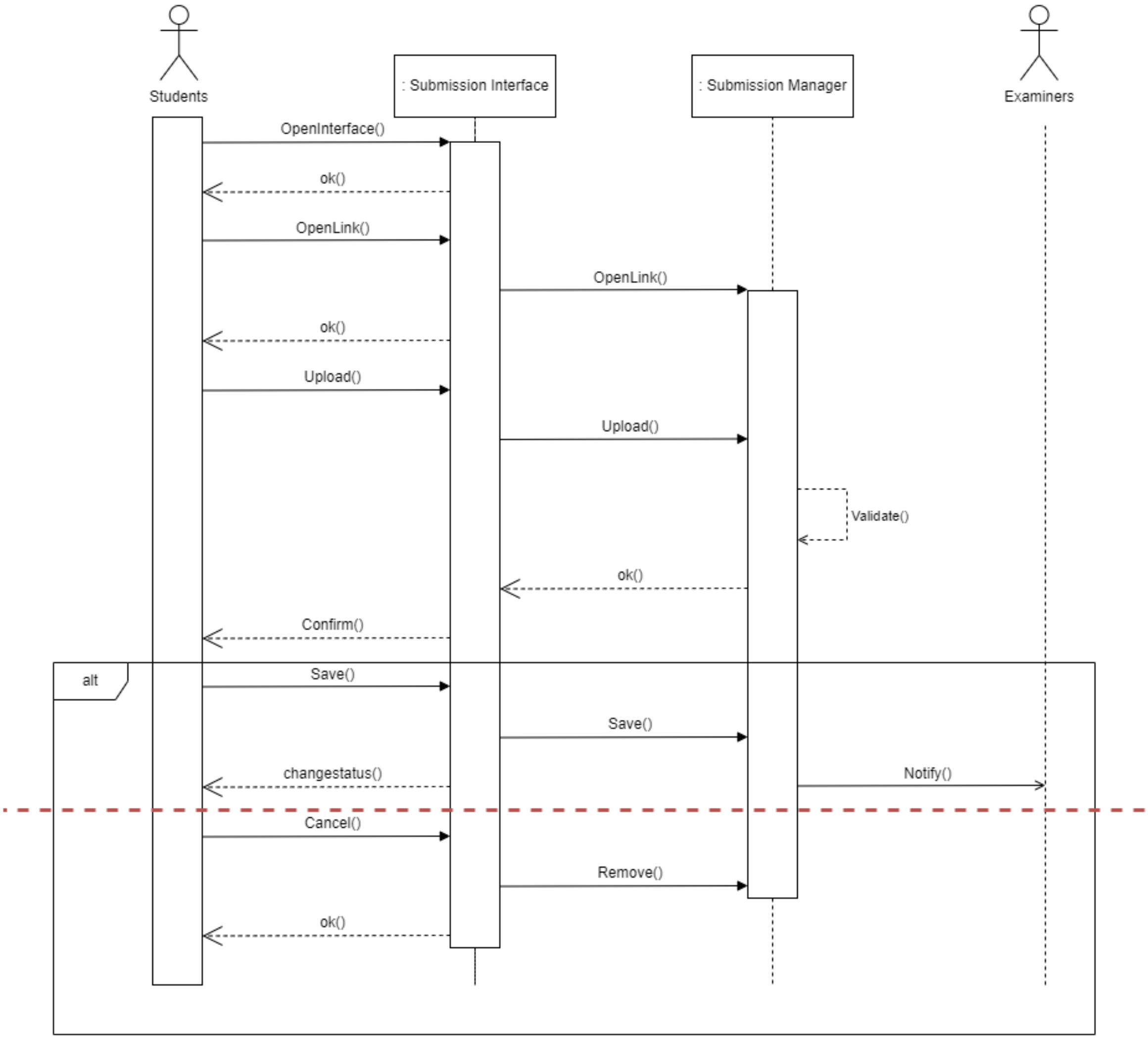
2.2.5.UC*005*: Use Case <Make Submission>

Use Case: Make Submission
ID: UC5
Actors: Student
Precondition: A valid student username and password in order to log in to the system. A valid submission platform is created.
Flow of Events: 1.The student logins to the system. 2.The student opens the submission interface. 3.The student clicks on the submission link. 4.The student selects and uploads his work/file from a valid drive. 5.The system displays confirmation messages. 6.If the student select “Save and make change” 1.System save and submit the uploaded file. 2.System change submission status to “Submitted”. 7.If the student select “Cancel” 1.System remove the uploaded file.
Postcondition: The system will notify the examiner. A submitted file can be viewed by an examiner.

Table 2.5: Use Case Description for <Make Submission>

Figure 2.12: Activity Diagram for <Make Submission>


Log in Maniacs System
Open submission Interface
Click submission link
Select and upload file from a valid drive
Display confirmation message
Save and make change
Cancel
Save and submit upload file
Remove uploaded file
Submission status change to "Submitted"



SD5

Figure 2.13: Sequence Diagram for <Make Submission>

2.2.6.UC*006*: Use Case <View Submission>

Use case: View Submission
ID: UC6
Actors: Examiner
Preconditions: A valid examiner username and password in order to log in to the system. A valid submission platform is created.
Flow of events: 1. The examiner logins to the system. 2. The examiner opens the submission interface. 3. A list of submission files displayed with its author(student), submission date and time. 4. The examiner clicks and selects one file to view its content.

Table 2.6: Use Case Description for <View Submission>

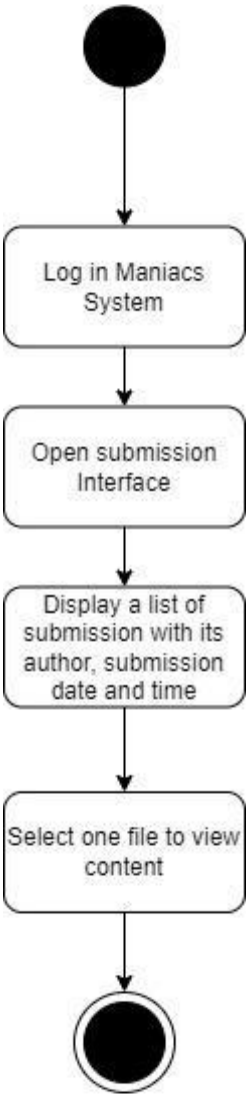

Log in Maniacs System
Open submission Interface
Display a list of submission with its author, submission date and time
Select one file to view content

Figure 2.14: Activity Diagram for <View Submission>

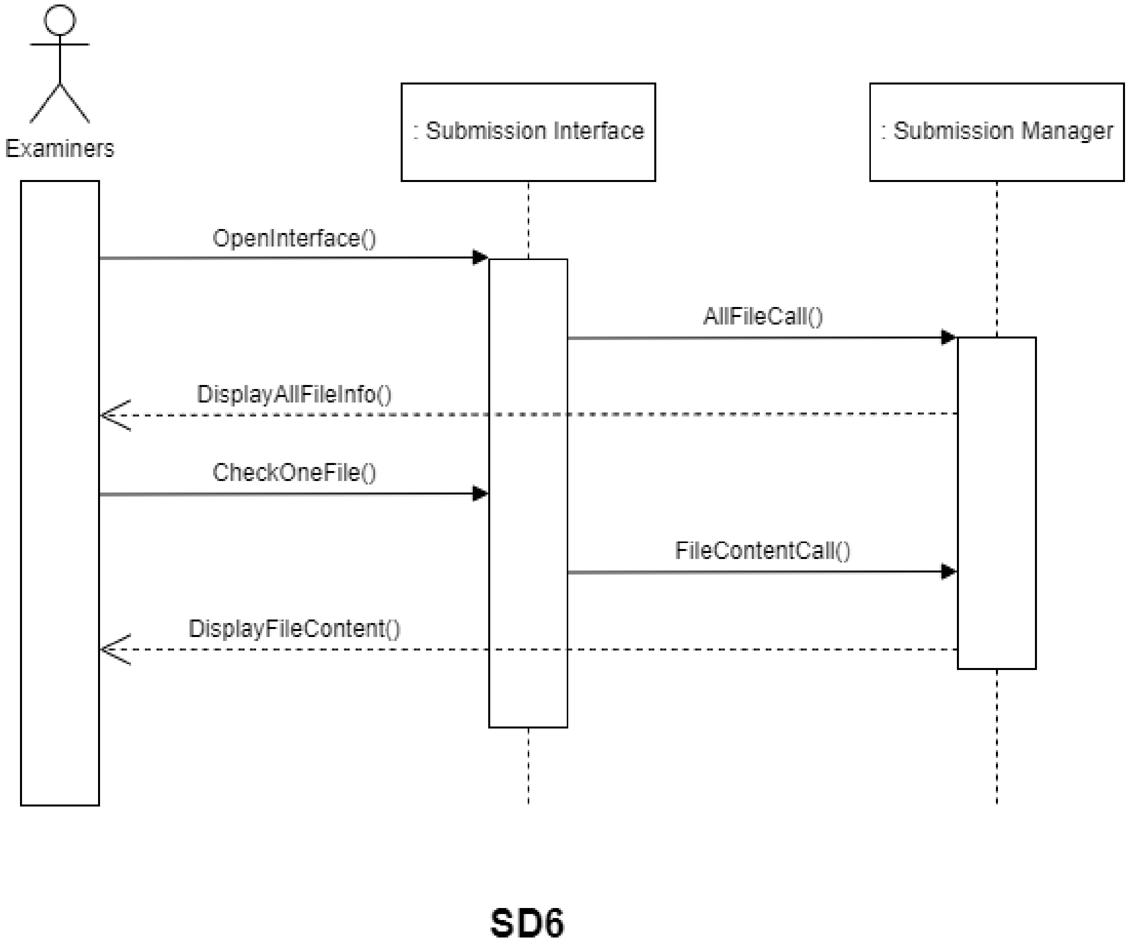


Figure 2.15: Sequence Diagram for <View Submission>

2.2.7.UC*007*: Use Case <Download Submission>

Use case: Download Submission
ID: UC7
Actors: Examiner
Includes: UC6 View submission
Preconditions: A valid examiner username and password in order to log in to the system. A valid submission platform is created. A valid file is submitted.
Flow of events: 1. The examiner logins to the system. 2. The examiner opens the submission interface. 3. Include (View submission) 4. The examiner clicks the "Download" button to download the file to the device.
Postcondition: A file can be accessed in local drive.

Table 2.7: Use Case Description for <Download Submission>

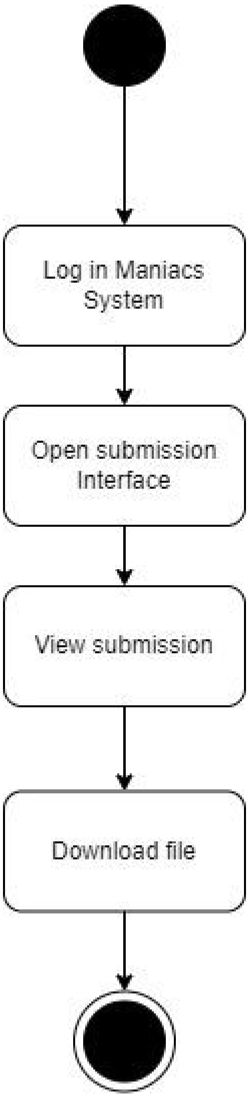
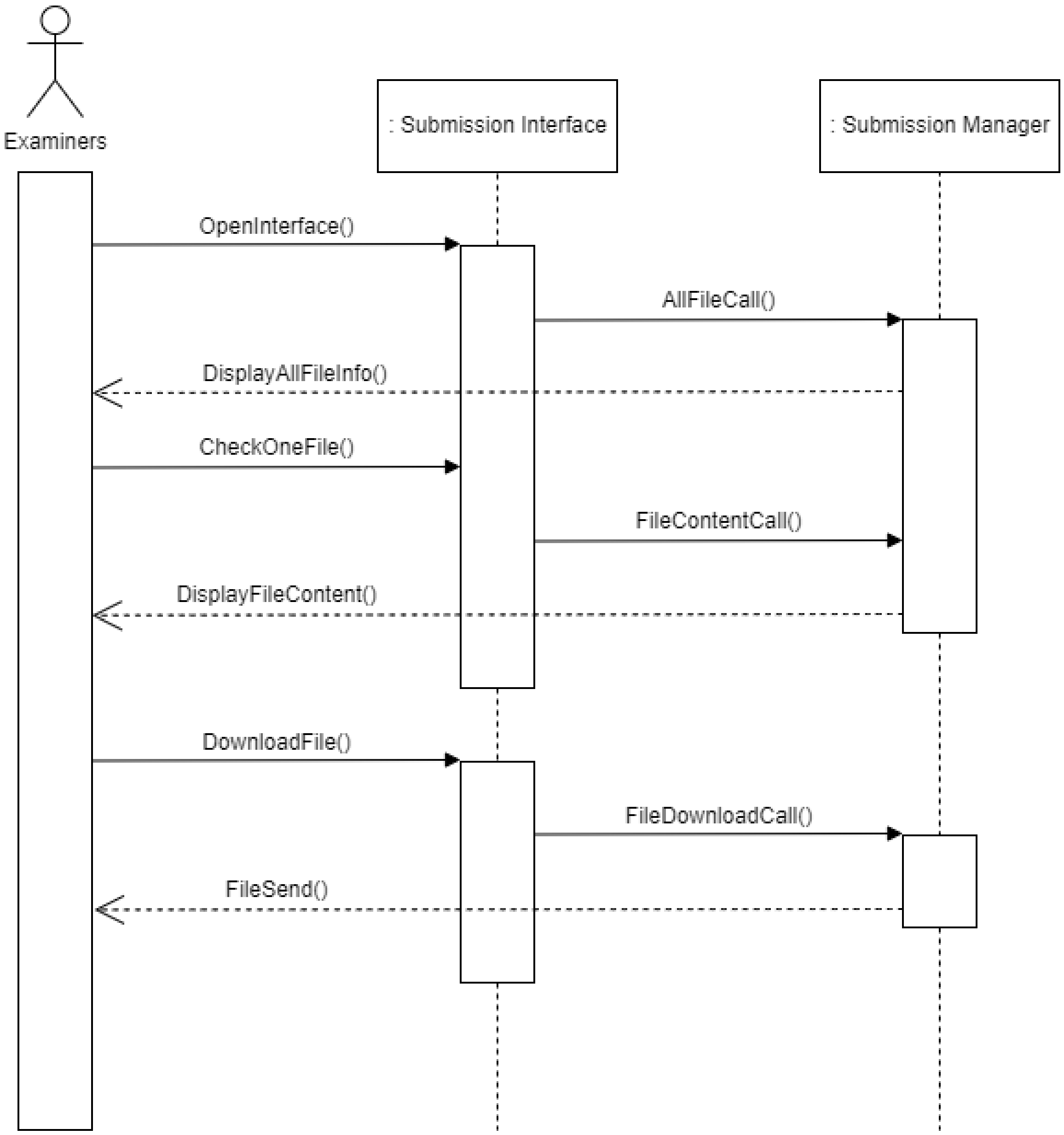


Figure 2.16: Activity Diagram for <Download Submission>



SD7

Figure 2.17: Sequence Diagram for <Download Submission>

2.2.8.UC*008*: Use Case <Grading>

Use case: Grading
ID: UC8
Actors: Examiner, Industry Coach, University Coach
Extension Points: <RecheckGrading>
Preconditions: A valid Examiner's, Industry Coach's, University Coach's username and password to log into the system. A submission has been made by students.
Flow of events: 1. Examiner, Industry Coach, University Coach login to the system. 2. If user is Examiner 2.1 While the student have 4 courses 2.2.1 Examiner selects "open" submission file. 2.2.2 Examiner does the grading based on each course code. 2.2.3 If the examiner have done the grading <<RecheckGrading>> 3. Else 3.1 System will provide an assessment form about each student. 3.2 The Industry Coach and University Coach does the grading based on the skills of each student. 3.3 If the Industry Coach and University Coach have done the grading <<RecheckGrading>> 4. The Examiner, Industry Coach and Examiner Coach will submit the grading in the System. 5. The system will save and record the grades of each student.

Postconditions:

The coordinator receives the grades.

Table 2.8: Use Case Description for <Grading>

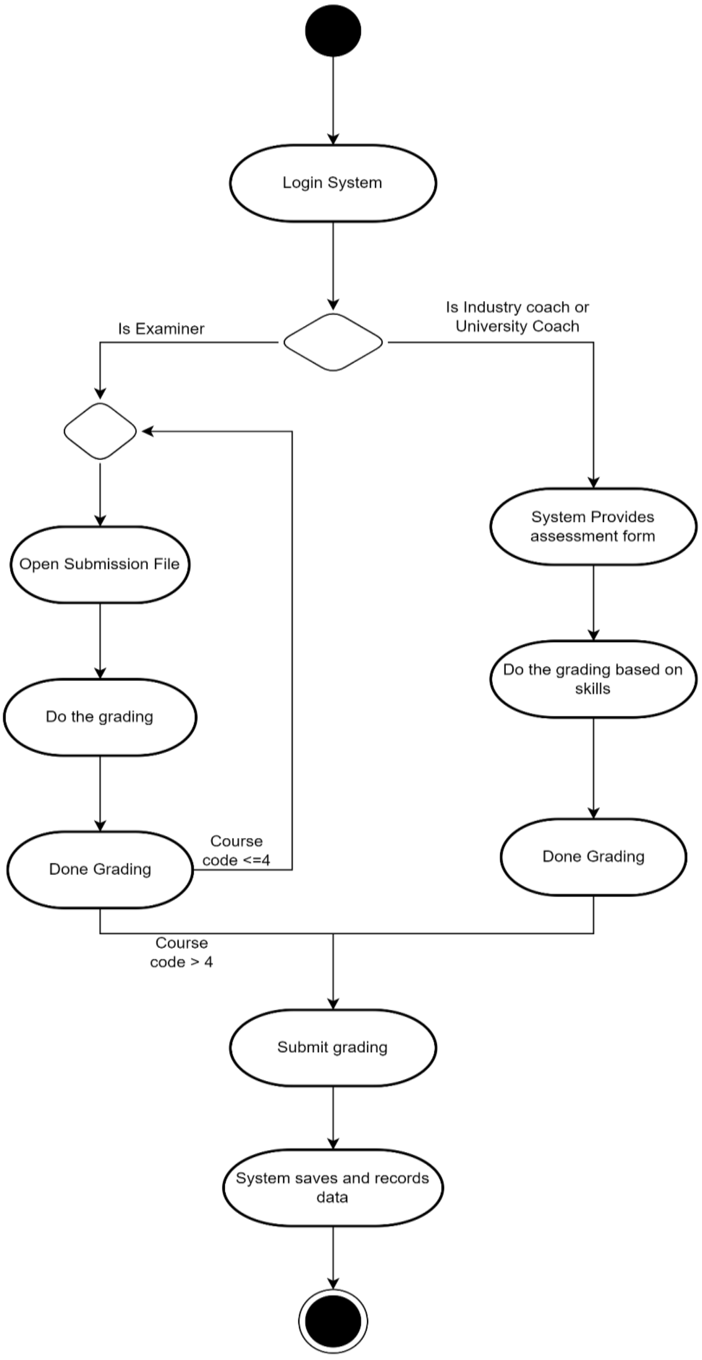

Login System
Is Examiner
Is Industry coach or University Coach
Open Submission File
Do the grading
Done Grading
Course code <=4
Course code > 4
System Provides assessment form
Do the grading based on skills
Done Grading
Submit grading
System saves and records data

Figure 2.18: Activity Diagram for <Grading>

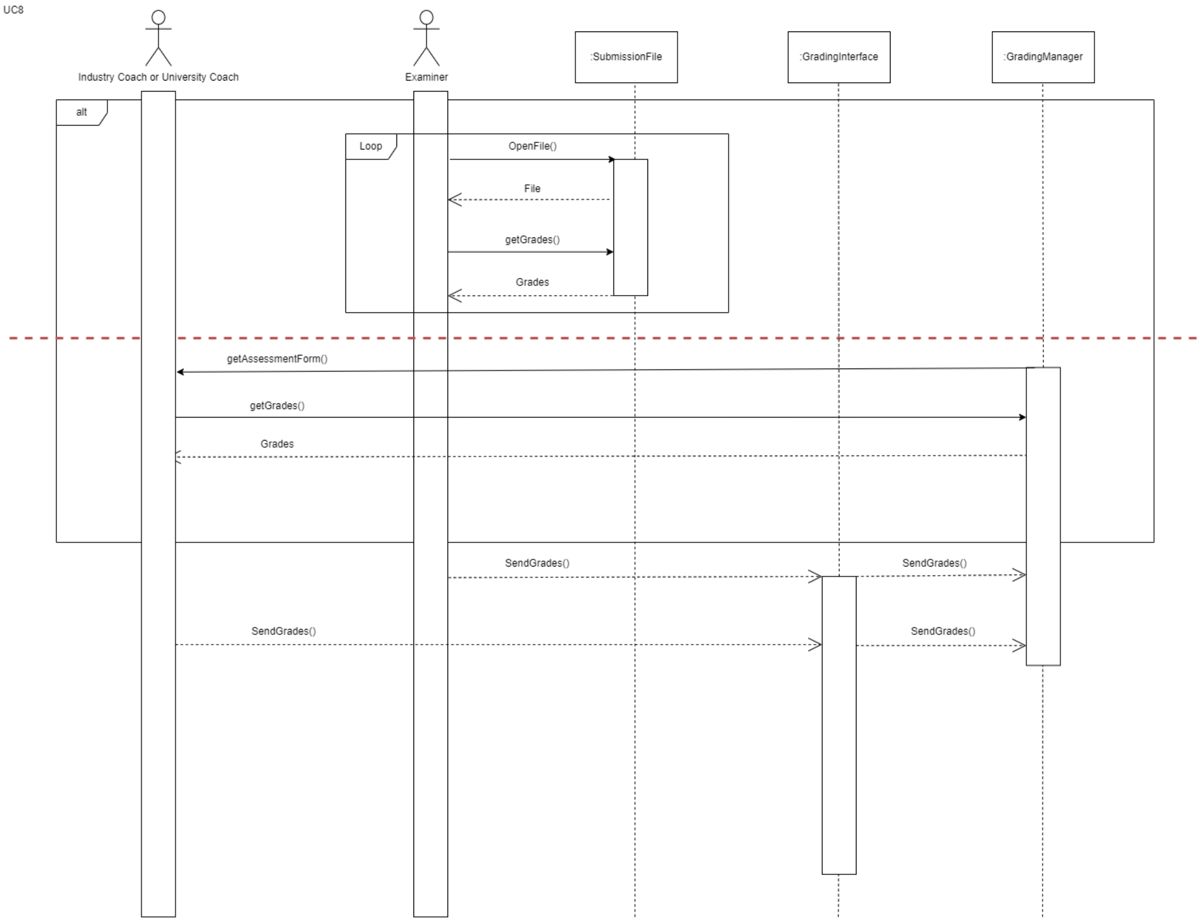


Figure 2.19: Sequence Diagram for <Grading>

2.2.9.UC*009*: Use Case < Recheck Grading>

Use case: Recheck Grading
ID: UC9
Actors: Examiner, Industry Coach, University Coach
Extends: UC8 Grading
Preconditions: A valid Examiner's, Industry Coach's, University Coach's username and password to log into the system A submission has been made by students. A grading has been done at least once.
Flow of events: 1. Examiner, Industry Coach, University Coach login to the system. 2. If user is Examiner 2.1 While the student have 4 courses 2.2.1 Examiner clicks on the recheck button. 2.2.2 System provides the graded course paper. 2.2.3 Examiner does the grading based on each course code. 3. Else 3.1 Industry Coach and University click on 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. 4. The Examiner, Industry Coach and Examiner Coach will submit the grading in the System. 5. The system will save and record the grades of each student.

Table 2.9: Use Case Description for <Recheck Grading>

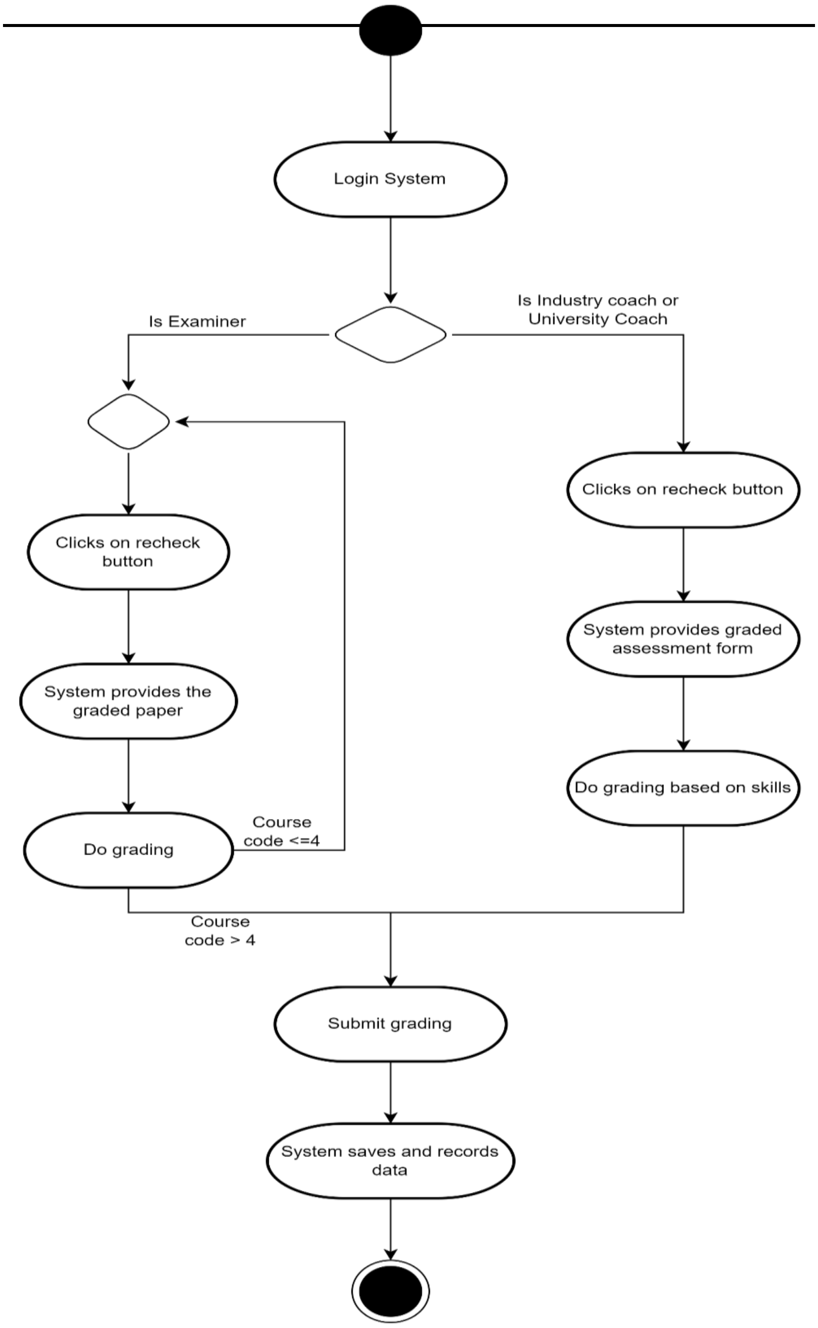

Login System
Is Examiner
Is Industry coach or University Coach
Clicks on recheck button
System provides the graded paper
Do grading
Course code <=4
Course code > 4
Clicks on recheck button
System provides graded assessment form
Do grading based on skills
Submit grading
System saves and records data

Figure 2.20: Activity Diagram for <Recheck Grading>

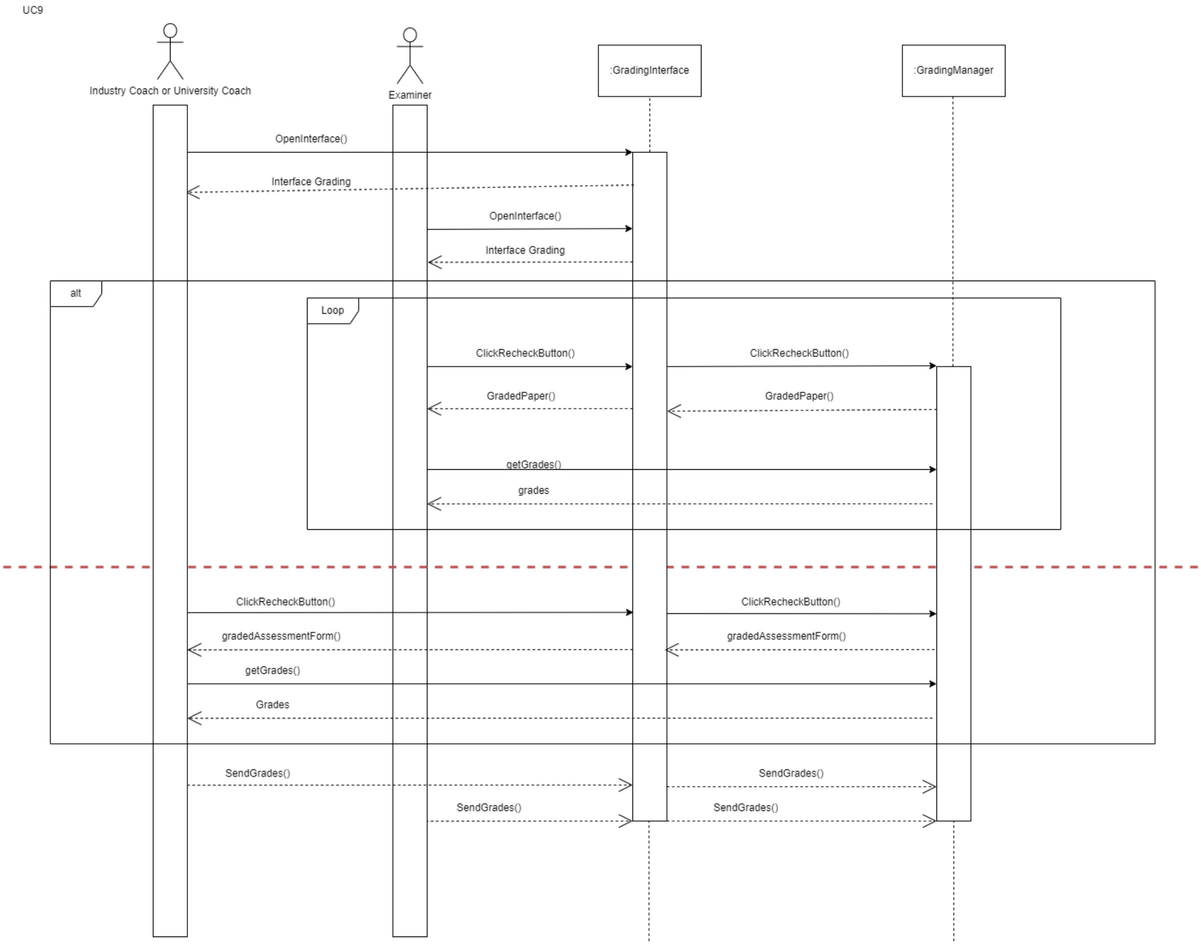


Figure 2.21: Sequence Diagram for <Recheck Grading>

2.2.10.**UC*010*: Use Case <Release Grading>**

Use case: Release Grading
ID: UC10
Actors: Coordinator
Preconditions: A valid Coordinator username and password to log into the system, grades have been provided. A grading has been made for each course.
Flow of events: 1. The Coordinator login to the system. 2.The Coordinator opens the Grading interface. 3.The Coordinator compares the recheck graded for each course. 4.The Coordinator will write a report about the grades of the students. 5.The Coordinator will release the grades to the students by clicking on the public button. 6.The system will display a confirmation message. 7.If the Coordinator press confirms. 7.1. The System will display the grades. 8.Else 8.1. The System will return back to the grading interface.
Postconditions: The grades are displayed with students' information.

Table 2.10: Use Case Description for <Release Grading>

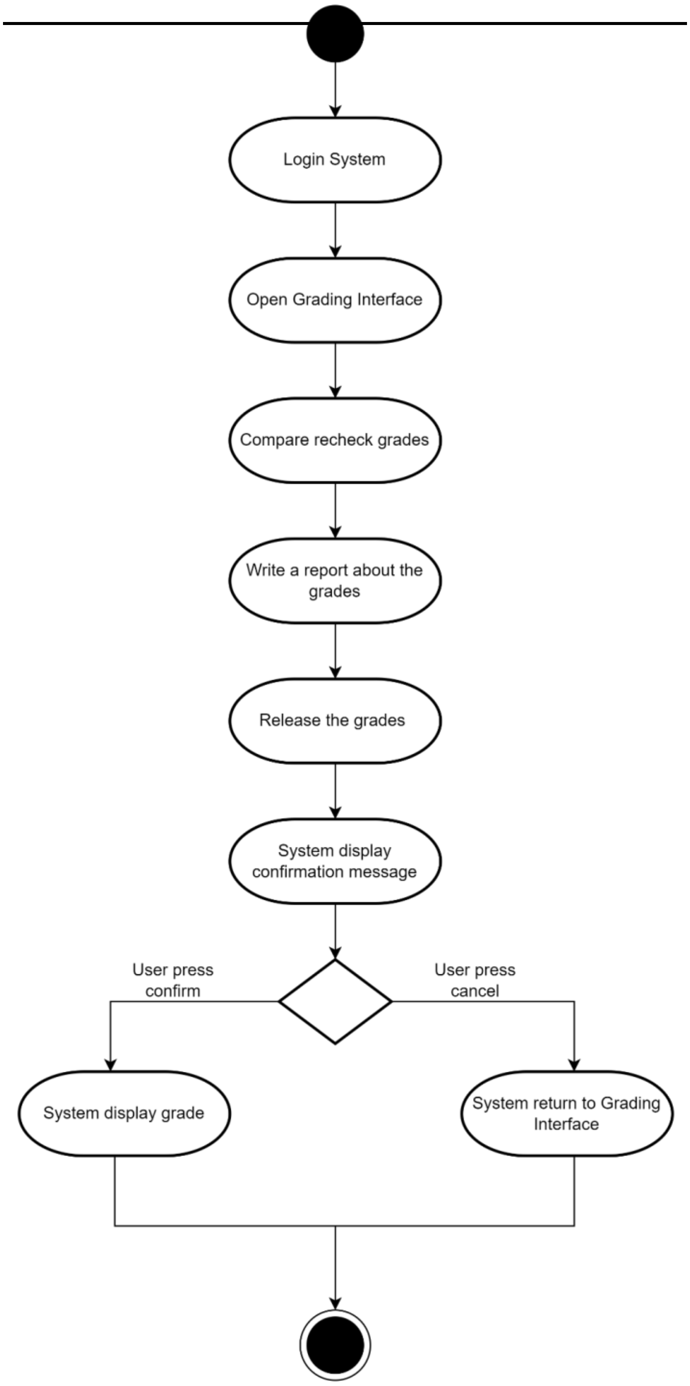


Figure 2.22: Activity Diagram for <Release Grading>

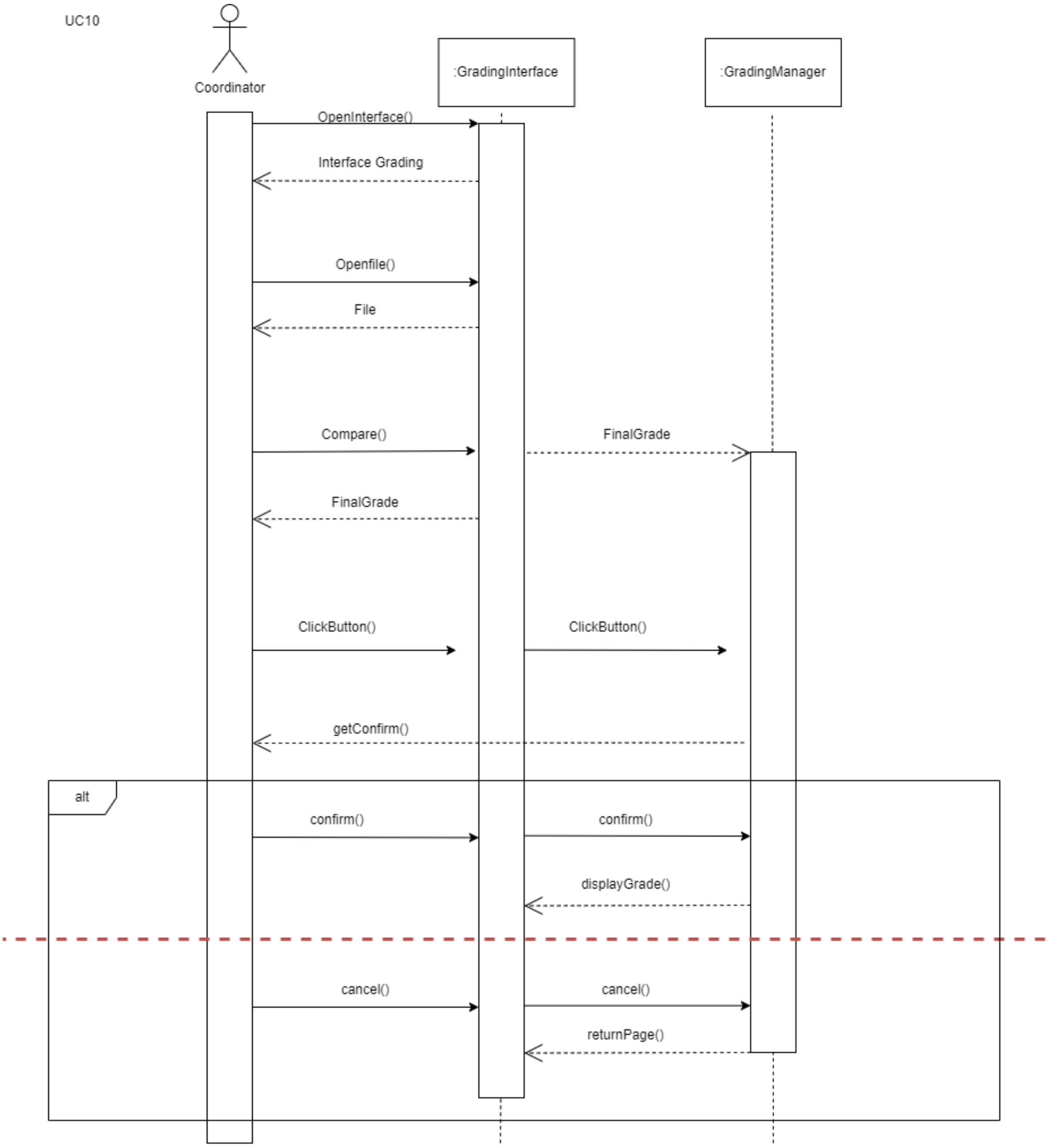


Figure 2.23: Sequence Diagram for <Release Grading>

2.2.11.**UC*011*: Use Case <View Grading>**

Use case: View Grading
ID: UC11
Actors: Students
Preconditions: A valid student’s username and password to log into the system. Students have made a submission. A 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 will ask students for course code. 4. The Student enters the code. 5. If code entered is valid 5.1. System will display the grade. 6. Else ..System will display invalid code.
Postconditions: The student receives the grades.

Table 2.11: Use Case Description for <View Grading>

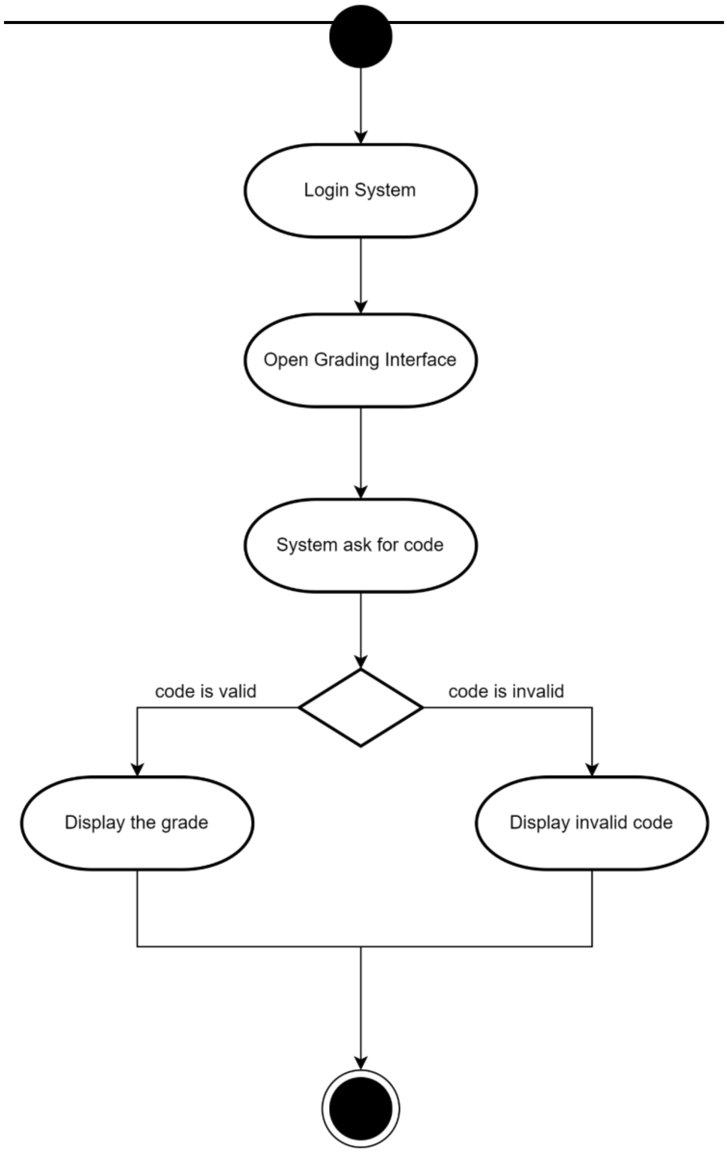


Figure 2.24: Activity Diagram for <View Grading>

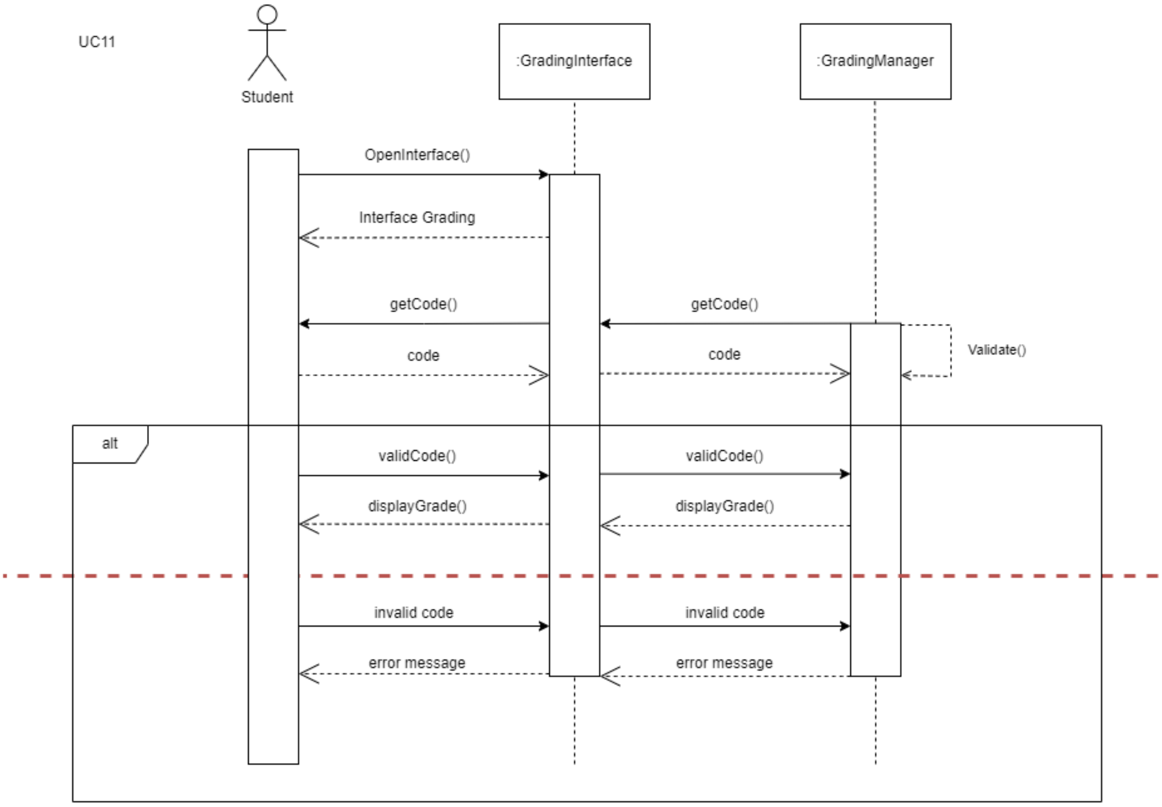


Figure 2.25: Sequence Diagram for <View Grading>

2.2.12.UC*012*: Use Case <Contact Coaches>

Use case: Contact Coaches
ID: UC12
Actors: Student, Industry Coach, University Coach
Preconditions: A valid Student's, Industry Coach's, University Coach's username and password to log into the system.
Flow of events: 1. The Student, Industry Coach, University Coach login to the system. 2. Student open message interface 3. Student click receiver's contact box 4. Student type at the chat box 5. Student press send to receiver
Postconditions: The student contacts the coach.

Table 2.12: Use Case Description for <Contact Coaches>

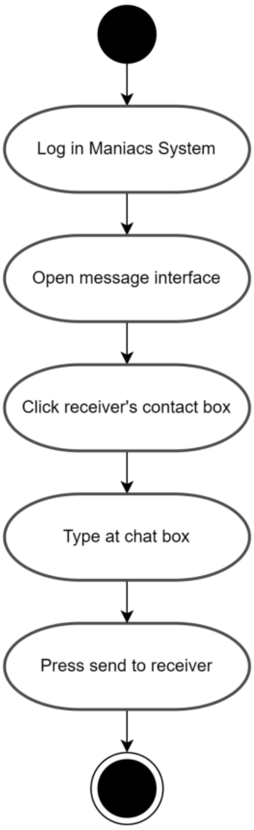


Figure 2.26: Activity Diagram for <Contact Coaches>

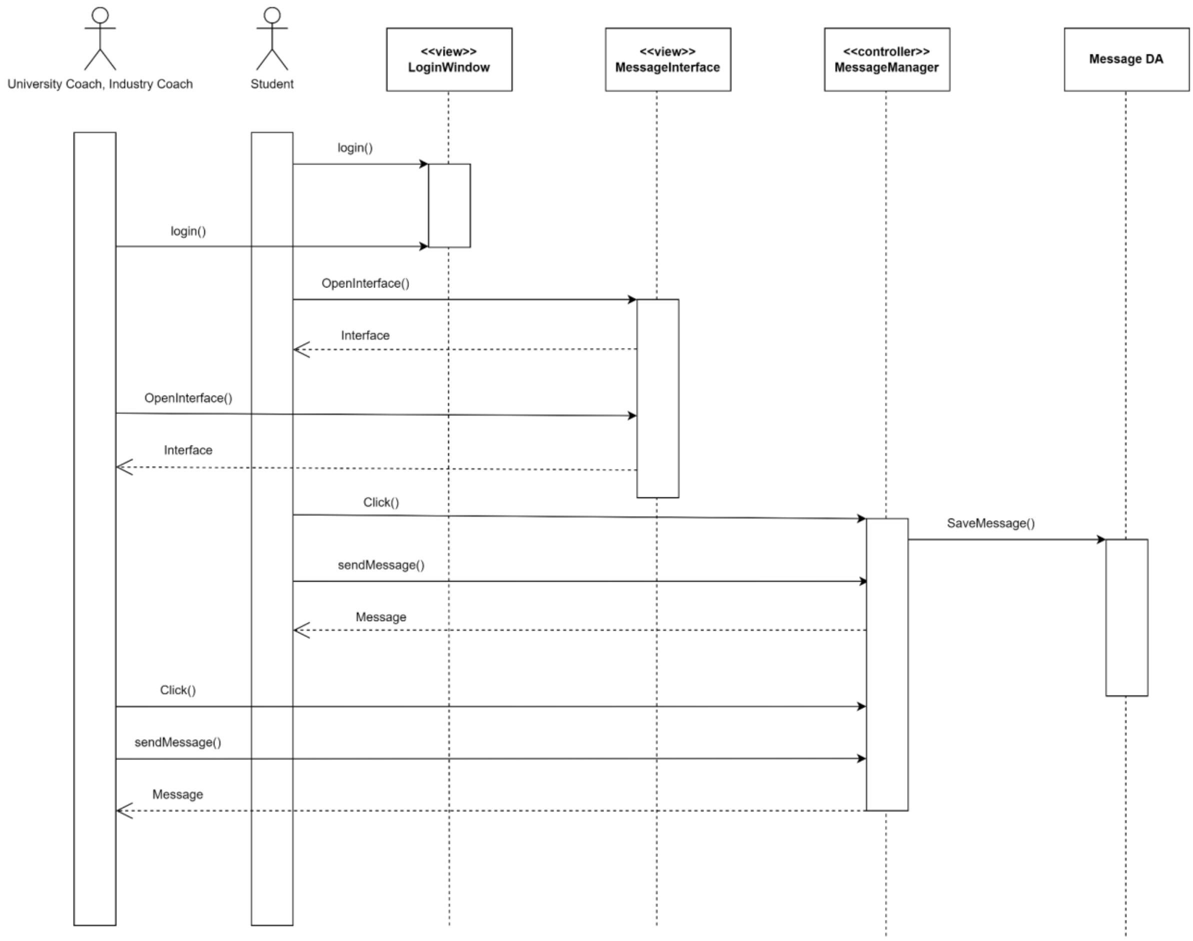


Figure 2.27: Sequence Diagram for <Contact Coaches>

2.3.Performance and Other Requirements

1. Response Time

Each system function will have its own reaction time, which will be used to gauge the system's overall performance. Although it is dependent on the device's hardware and software, the system will be created with the least amount of system performance in mind. After a successful login session, the system must fully load onto the system's main page in less than 5 seconds, according to a specified functional requirement.

2. Accuracy and Validity

The system will use a variety of data quality and policy tools, such as input masks and buttons, among others.

3. Timing and capacity

With the exception of scheduled and pre-notified system maintenance and upgrade downtimes, the system is designed to be available 24 hours a day, 365 days a year.

4. Failure Contingencies

The system is not in a critical state. Any user will not be inconvenienced by temporary inaccessibility for a maximum of three days. The system will alert the user by displaying error pages/images informing them of the system's failure. To avoid data loss, all information will be frozen. In addition, The data will be kept safe by UTM CICT, which will use daily backup processes in accordance with the UTM Policy.

2.4.**Design Constraints**

1. Safety Requirements

Several users should be able to access the system at different levels. A user login screen protects this, requiring a matric ID or staff ID and password. It will provide many perspectives and functionalities that may be accessed through the system. In the event of an emergency, the system can be restored.

2. Security Requirements

Users will be able to log in using their ID and password, which will protect them from gaining access to the various subsystems. The Maniac System will be available to students. Coaches, examiners, and coordinators have complete control over all subsystems.

3. Business Rules

The Maniac System will be used by a variety of users, including students, coordinators, examiners, industry coaches, and university coaches. The system is set up in such a manner that users' responsibilities and powers diminish as their rank rises.

2.5.Software System Attributes

1. Maintainability

The Maniac System is made to be simple to use and maintain. It is developed in such a way that it may be modified and extended to meet changing client demands. It is also appropriate for debugging, updating, and extending functionality as necessary.

2. Learnability

The Maniac System is intended to be simple to use. The user interfaces show data as realistically as possible and allow for efficient use of the software's flaws. It also simplifies and clarifies the user instructions.

3. Security

The Maniac System is well-designed to prevent errors from occurring. In the case of a system breakdown, the system's trustworthy software will not inflict physical harm. Malicious users will be unable to login to the system or cause damage to it.

4. Efficiency

The Maniac System has the potential to accomplish the goal in the most cost-effective and time-efficient manner possible.

5. Availability

During regular class sessions, the Maniac System will be available.

6. Correctness

The extent to which a software meets specifications and achieves the user's mission objectives.

7. Flexibility

The extent to which a software meets specifications and achieves the user's mission objectives.

8. Integrity

How the system secures the data in the system and prevents data loss. In database tables and interfaces, referential integrity is important.

9. Portability

The Maniac System will work on any device with a different operating system version.

10. Reliability

Specify the criteria that must be considered in order to determine the software system's needed reliability at the time of delivery. The average time between failures and the average time to recover.

11. Reusability

The capacity to incorporate the system's available components into other systems.

12. Testability

Effort required to test to ensure that it works as expected.

13. Usability

How effortlessly a person can take advantage of the system's benefits and user-friendliness.

14. Robustness

The system's ability to accurately handle system functions and maintain the database without experiencing unexpected failures.

3.System Architectural Design

3.1.Architecture Style and Rationale

We choose layered architecture as our architectural style. The reason for choosing layered architecture is that it divides the system into layers, each with its own set of functions. The lowest-level layers indicate essential services that are likely to be used across the system because each layer delivers services to the layer above it. Therefore, in the Maniac System, one of the three layers is the User Interface Layer which contains the actual Graphical User Interface for users to view or click. The second layer is the Domain Layer (Business Logic) which holds the business logic classes that represent the Maniac System business model. Finally, the third layer is the Data Access Layer which makes data stored in entity-relational databases more accessible. As Maniac System already has a manual system, layered architecture is a perfect choice because it is utilized when adding new facilities to current systems. Furthermore, the layered architecture enables for the replacement of entire layers as long as the interface is maintained, hence reducing the user interface replacement cost. Since the Maniac System may require external redundant facilities such as Student Information and Grades for each course subject, each layer might have a layered architecture to strengthen the system's reliability.

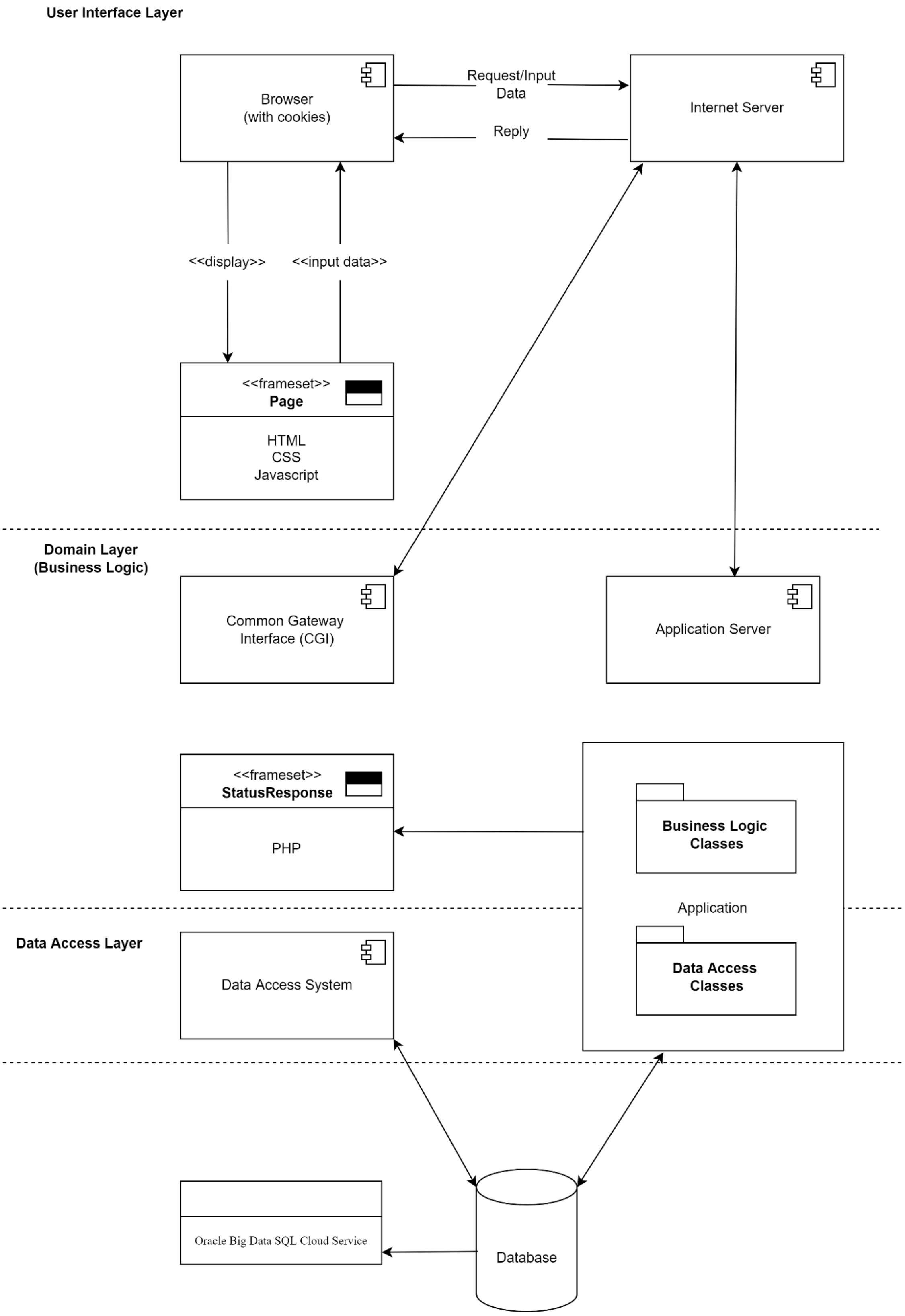


Figure 3.1: Complete Architectural Diagram of <Maniac System>

3.2. Component Model

In this component model, we have 4 subsystems which are **notification subsystem, submission subsystem, grading subsystem and communication subsystem.** So, for the notification subsystem, all the targets which are industry coach, student and university coach will receive the notification from the coordinator. Also, for the coordinator, it can edit and delete the notification. Then, for the submission subsystem, the coordinator will create the submission platform where it can edit the submission details. Students will submit the report at the submission platform and the examiner will check the report. Notification subsystem has a relationship with the submission subsystem. If there are any submission reports from students, the coordinator will send the notification because the coordinator created the submission platform or when the student reports finished.

Next, the grading subsystem is mainly about the grading marks where the coordinator will release the marks and students can view the grading marks. For examiner, they will recheck the grading marks to give it to the university coach while for industry coach, they will be given the student's grading marks from the examiner. So, there is a relationship between the grading subsystem and the submission subsystem where the coordinator releases the grading marks, they will key in the grades in the student report meanwhile for students, they can review their grading marks at the student report.

Lastly, the communication subsystem is a platform for the students to contact the coach such as university coach and industry coach. This subsystem is an independent relationship where it does not depend on other subsystems.

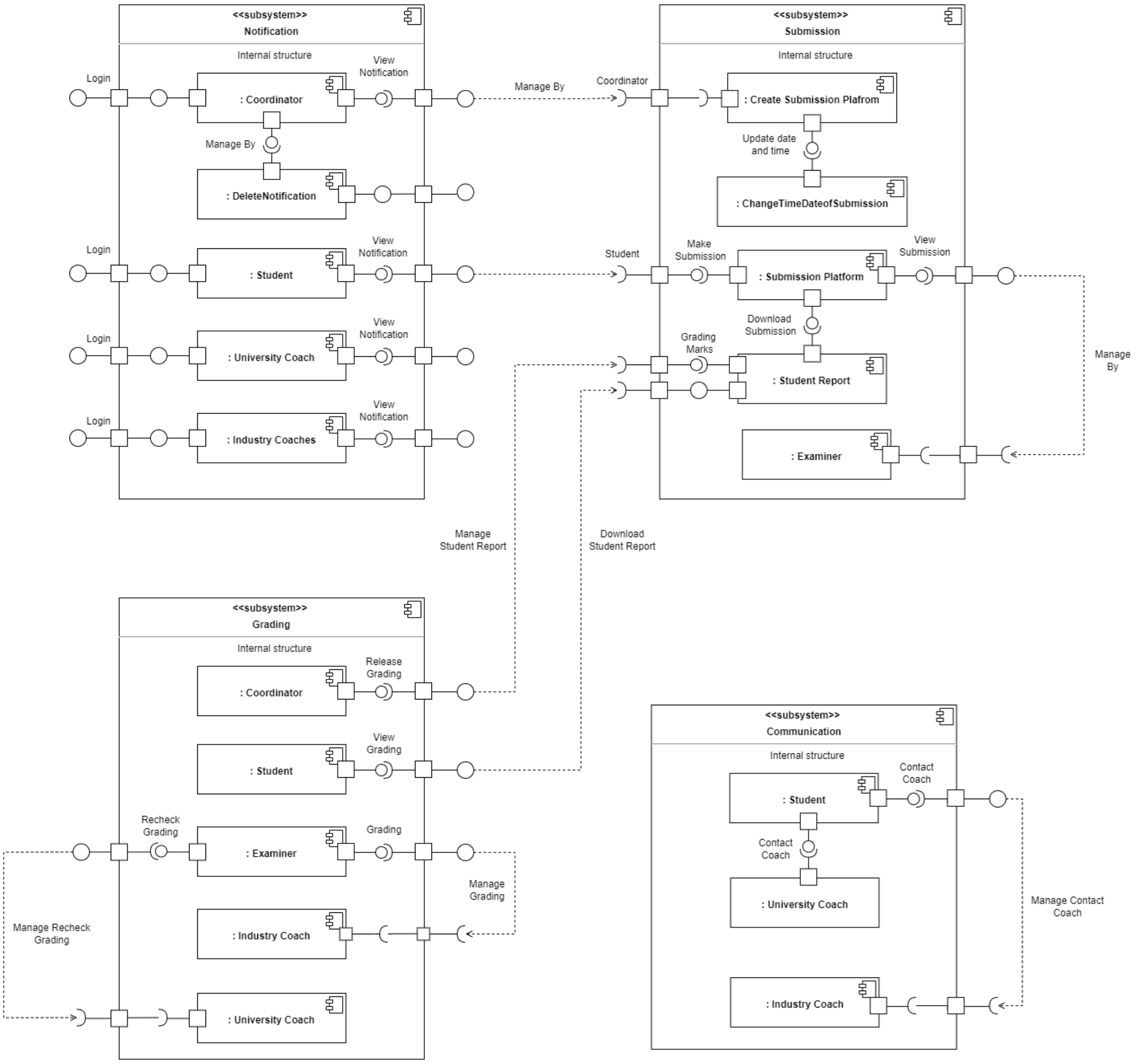
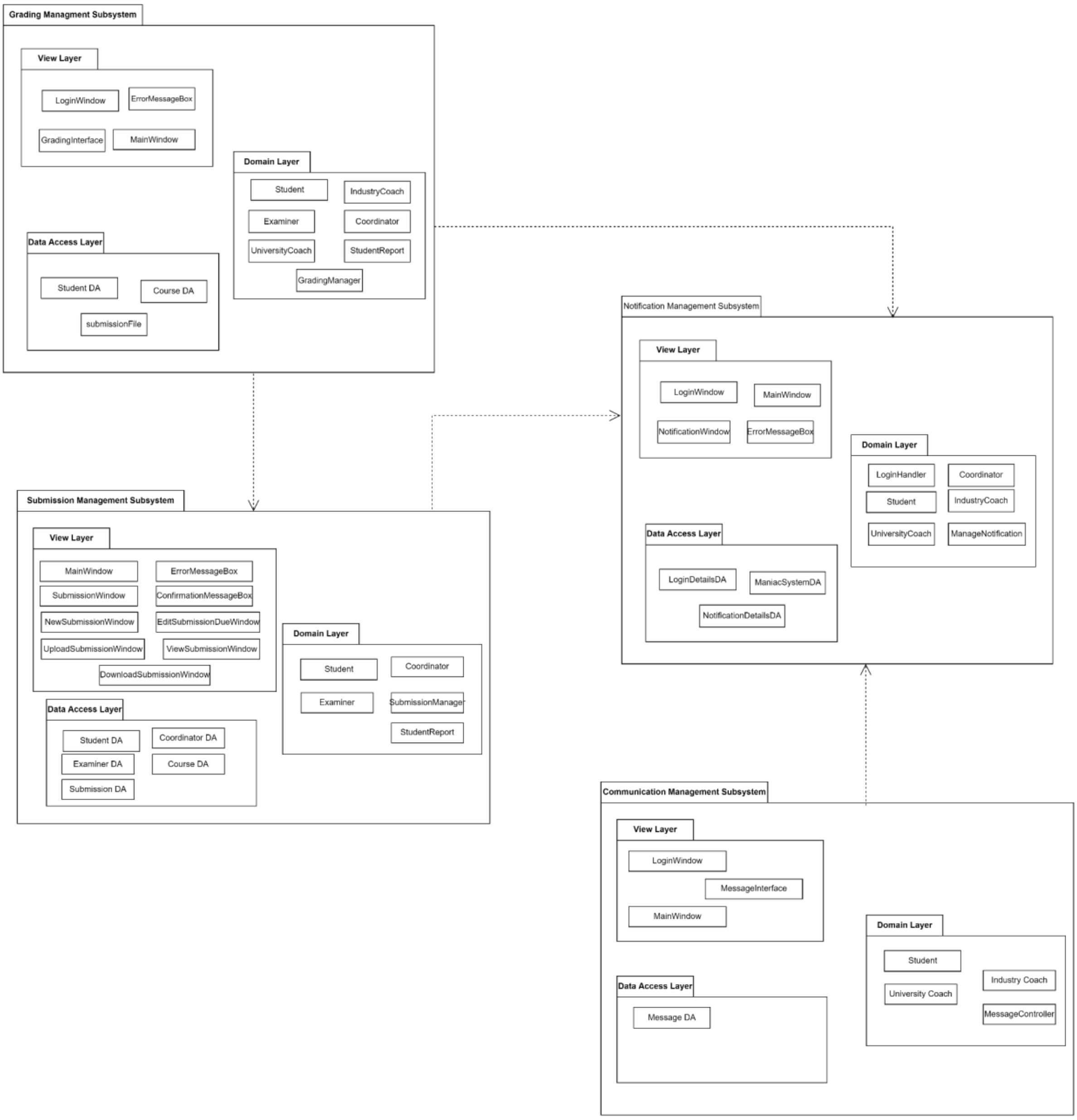


Figure 3.2: Component Diagram of <Maniac System>

4.Detailed Description of Components

4.1.Complete Package Diagram



https://drive.google.com/file/d/1yyH0TUExzyQ6H6BCrNVnoTvL6ty4YF3x/view?usp=sharing

Figure 4.1: Package Diagram for <Maniac System>

4.2. Detailed Description

4.2.1. P001: <Notification Management> Subsystem

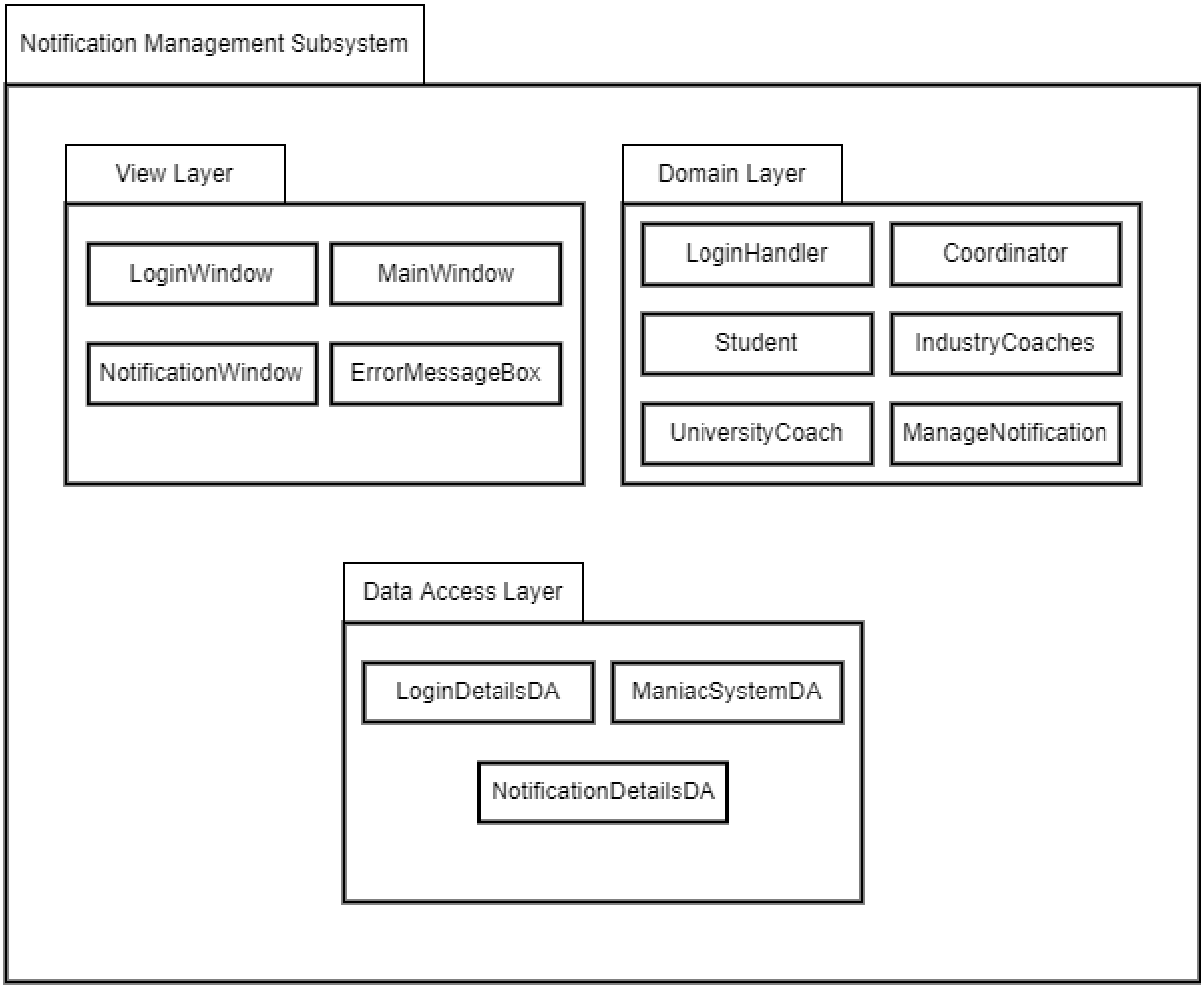


Figure 4.2: Package Diagram for <Notification Management> Subsystem

4.2.1.1.**Class Diagram**

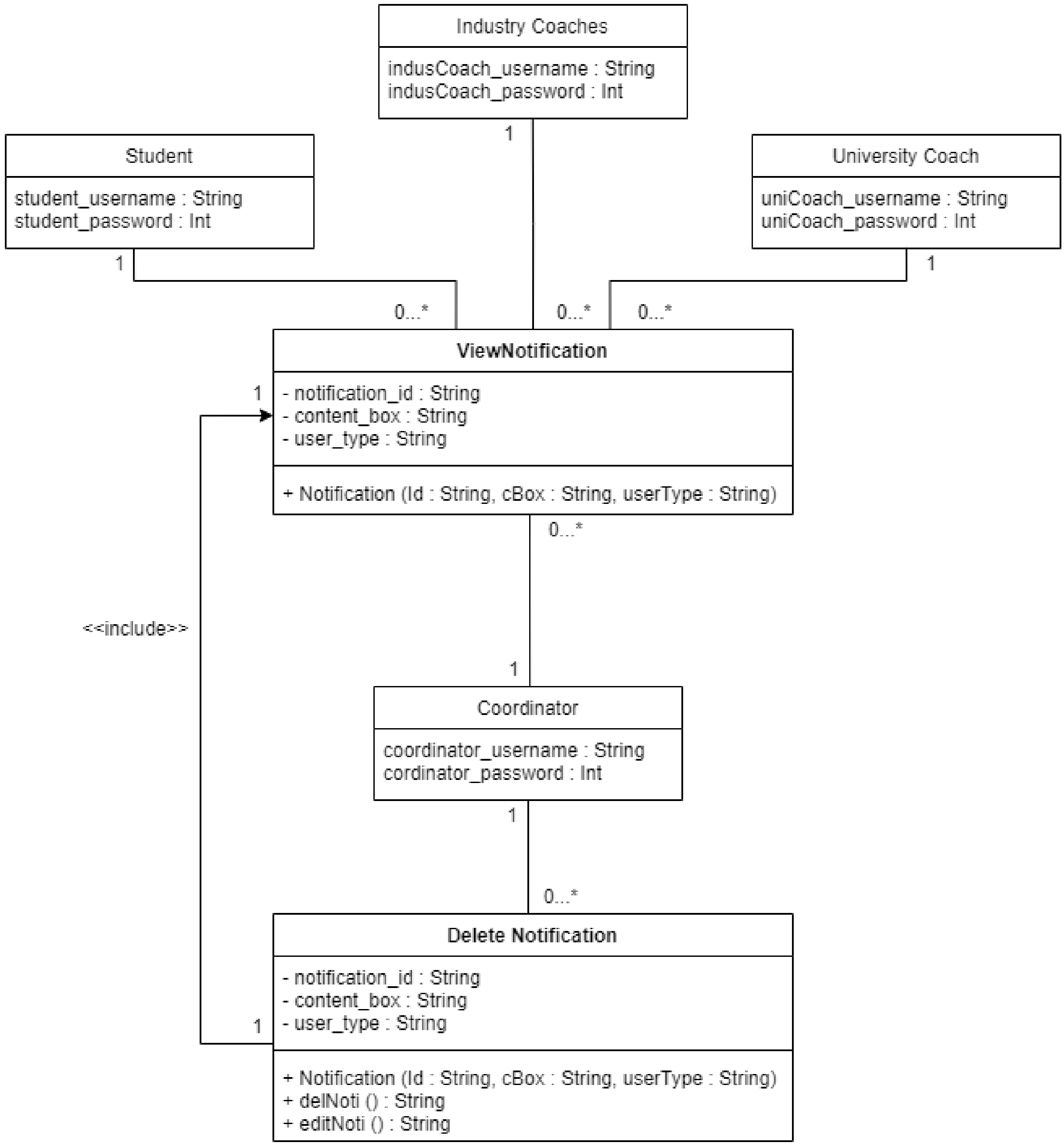

Industry Coaches
indusCoach_username : String
indusCoach_password : Int
Student
student_username : String
student_password : Int
University Coach
uniCoach_username : String
uniCoach_password : Int
1
1
1
0...*
0...*
0...*
ViewNotification
- notification_id : String
- content_box : String
- user_type : String
+ Notification (Id : String, cBox : String, userType : String)
1
0...*
<<include>>
1
Coordinator
coordinator_username : String
cordinator_password : Int
1
0...*
Delete Notification
- notification_id : String
- content_box : String
- user_type : String
1
+ Notification (Id : String, cBox : String, userType : String)
+ delNoti () : String
+ editNoti () : String

Figure 4.3: Class Diagram for <Notification Management> Subsystem

UC001: View Notification

Entity Name	Coordinator, University coach, Industry coach, Student
Method Name	Notification()
Input	notification_id, content_box, user_type
Output	View the notification
Algorithm	1. Start 2. Login System as a coordinator 3. Click "Notification Management" in main page 4. If there are any changes, click "Edit" 5. Can edit any info which is the text inside the box 6. After finish editing, click "View" to see the updated info 7. Next, login System as a university coach or industry coach or student 8. Click "Notification Management" in main page 9. Click "View" to read more about the notification 10. End

UC002: Delete Notification

Entity Name	Coordinator
Method Name	Notification()
Input	notification_id, content_box, user_type
Output	Delete the notification
Algorithm	1. Start 2. Login System as a coordinator 3. Click "Notification Management" in main page 4. Click "Delete" button if the coordinator wants to delete the notification 5. End

4.2.1.2.**Sequence Diagram**

a) SD001: Sequence diagram for View Notification

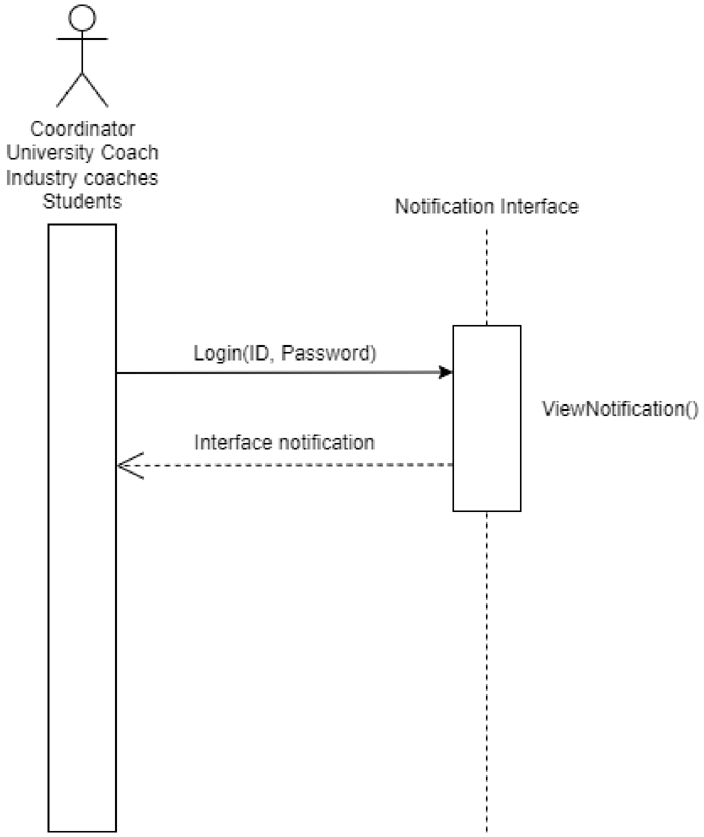


Figure 4.4: Sequence Diagram for <View Notification>

b) SD002: Sequence diagram for Delete Notification

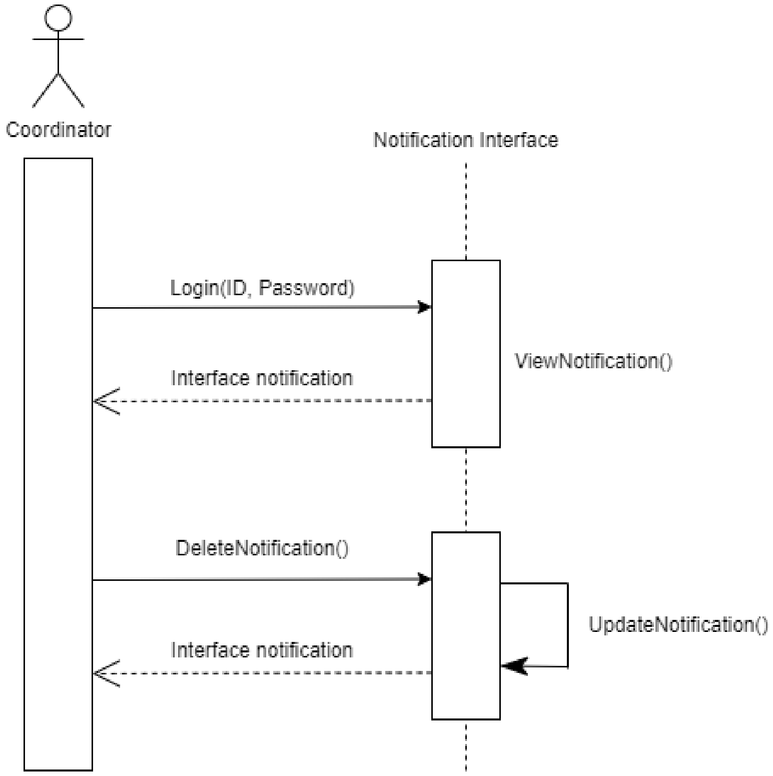
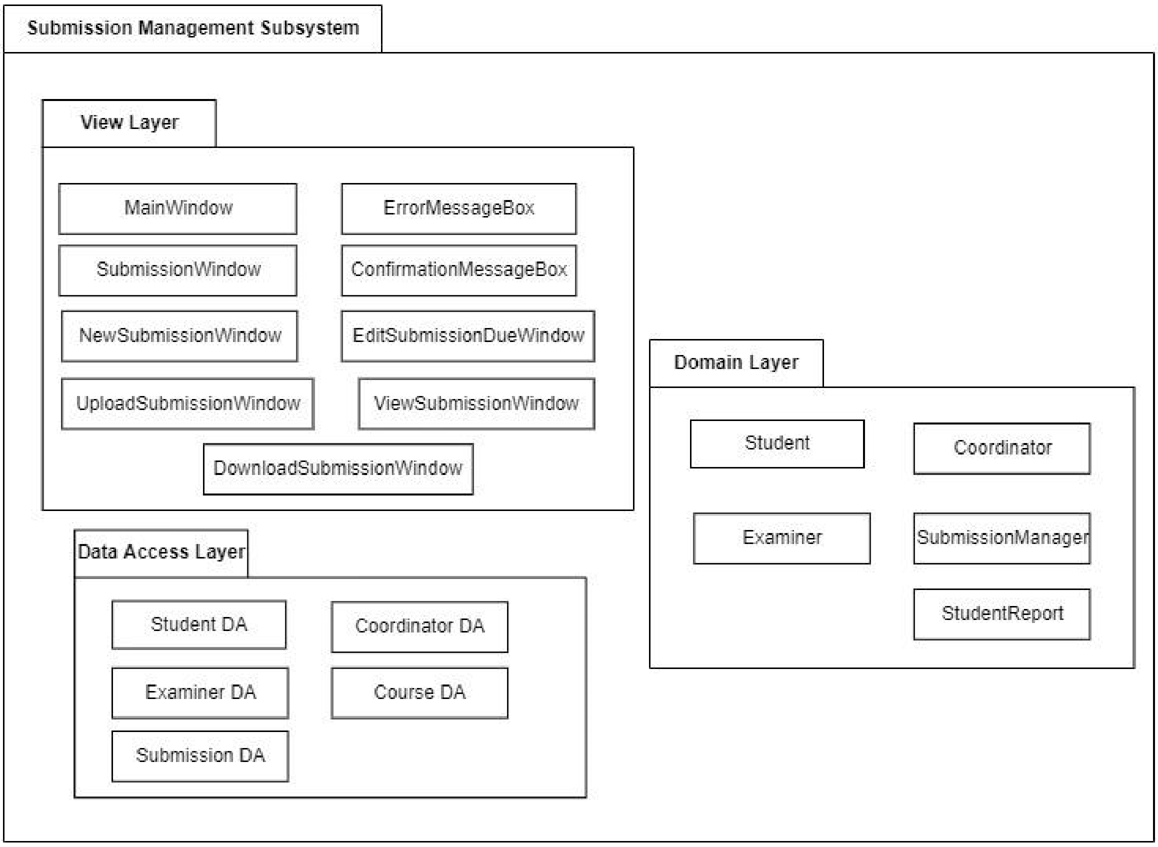
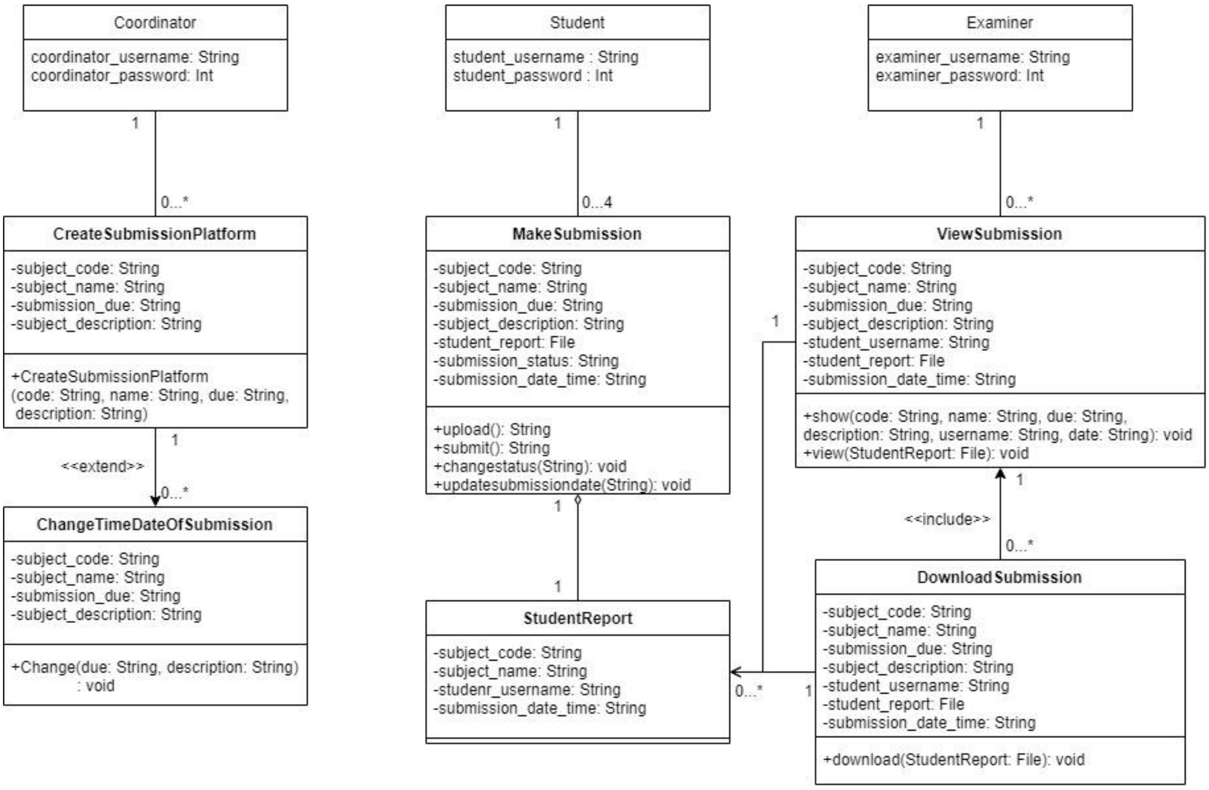


Figure 4.5: Sequence Diagram for <Delete Notification>

4.2.2.**P002: <Submission Management> Subsystem**



4.2.2.1.**Class Diagram**


Coordinator
coordinator_username: String
coordinator_password: Int
Student
student_username : String
student_password : Int
Examiner
examiner_username: String
examiner_password: Int
1
0...*
1
0...4
1
0...*
CreateSubmissionPlatform
-subject_code: String
-subject_name: String
-submission_due: String
-subject_description: String
+CreateSubmissionPlatform
(code: String, name: String, due: String,
description: String)
1
<<extend>>
0...*
ChangeTimeDateOfSubmission
-subject_code: String
-subject_name: String
-submission_due: String
-subject_description: String
+Change(due: String, description: String)
: void
MakeSubmission
-subject_code: String
-subject_name: String
-submission_due: String
-subject_description: String
-student_report: File
-submission_status: String
-submission_date_time: String
+upload(): String
+submit(): String
+changestatus(String): void
+updatesubmissiondate(String): void
1
1
StudentReport
-subject_code: String
-subject_name: String
-studenr_username: String
-submission_date_time: String
ViewSubmission
-subject_code: String
-subject_name: String
-submission_due: String
-subject_description: String
-student_username: String
-student_report: File
-submission_date_time: String
+show(code: String, name: String, due: String,
description: String, username: String, date: String): void
+view(StudentReport: File): void
1
1
<<include>>
0...*
DownloadSubmission
-subject_code: String
-subject_name: String
-submission_due: String
-subject_description: String
-student_username: String
-student_report: File
-submission_date_time: String
+download(StudentReport: File): void
0...*
1

UC003: Create Submission Platform

Entity Name	Coordinator
Method Name	CreateSubmissionPlatform()
Input	subject_code, subject_name, submission_due, subject_description
Output	submission platform to submit the report
Algorithm	1. Start 2. Login System as a coordinator 3. Click “Submission Management” in main page 4. Click symbol ‘+’ at the bottom page if want to create new submission by enter the subject code, subject name and submission due 5. Can add notes at the description 6. Click “Create” button to create the new submission platform 7. If there are any changes, click “Edit” 8. Can edit any info which are subject code, subject name, submission due and notes at the description box 9. After finish editing, click “Save” button to save the updated submission details 10. End

UC004: Change Time Date of Submission

Entity Name	Coordinator
Method Name	Change()
Input	submission_due
Output	time and date of the submission change
Algorithm	1. Start 2. Login System as a coordinator 3. Click “Submission Management” in main page 4. Click “Edit” button to edit the submission due 5. Change the date 6. After finish editing, click “Save” button to save the updated submission details 7. End

UC005: Make Submission

Entity Name	Student, StudentReport, SubmissionManager

Method Name	upload()
	submit()
	changestatus()
	updatesubmissiondate()

Input	Submission_File, subject_code, student_username, submission_status, submission_date_time
Output	File upload to submission database, submission status and submission date, time change and update in student database.
Algorithm	1. Start 2. Login System as a student 3. Open “Make Submission” in main page 4. Select a submission 5. Check the submission details include subject code, subject name, submission due, subject description 6. Click “Upload” button to upload a file from valid drive 7. Click “Submit” button to submit the file uploaded 8. Confirm the submission 8.1. If submission is confirmed 8.1.1. File upload to submission database 8.1.2. Submission date and time updated. 8.1.3. Submission Status change to “submitted” 8.2. ELSE 8.2.1 Return back to “Make Submission” page .END

UC006: View Submission

Entity Name	Examiner, StudentReport, SubmissionManager

Method Name	show()
	view()

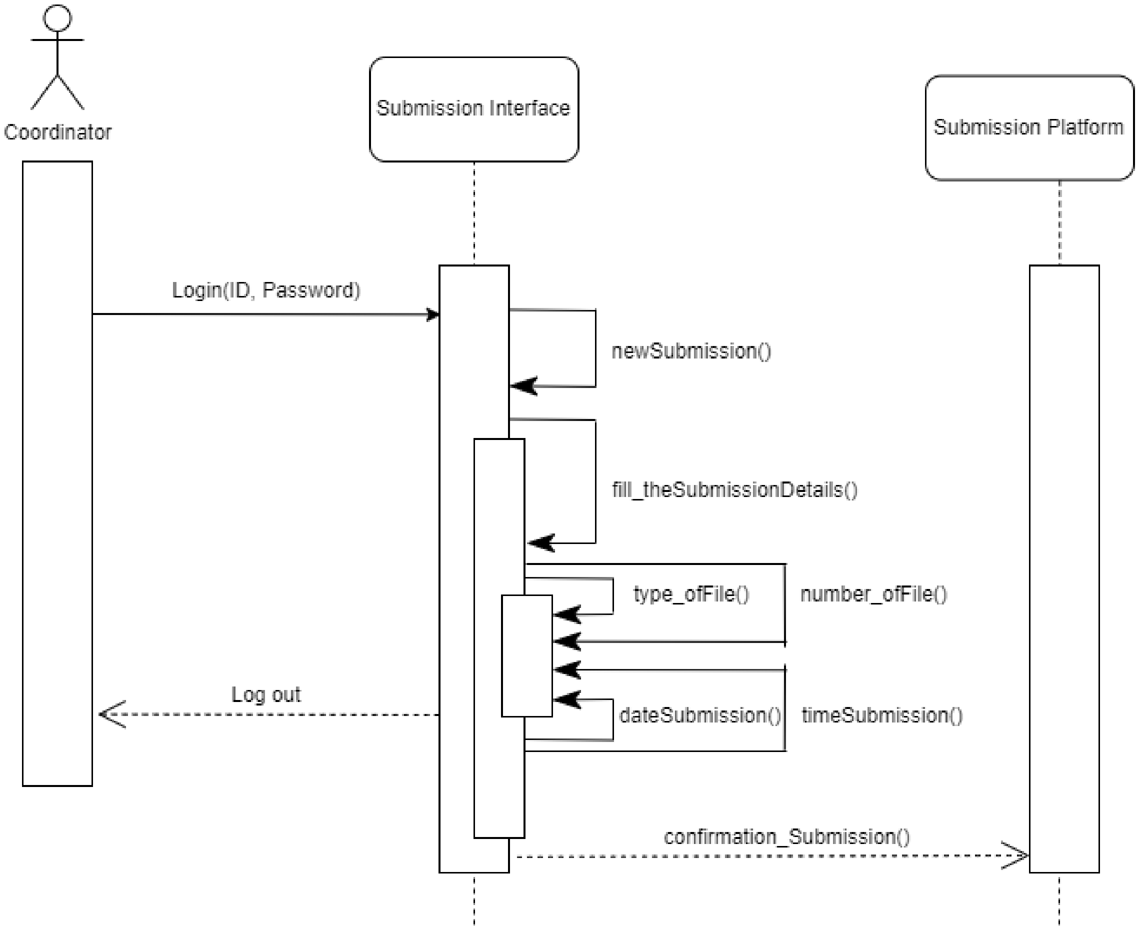
Input	subject code, student_username
Output	System displays a submission file from the submission database to the "View Submission" page.
Algorithm	1. Start 2. Login System as an examiner 3. Open "Download Submission" on the main page 4. Check the submission details include subject code, subject name, submission due, subject description 5. Select to view a list of submissions 6. System shows all students who made a submission and their submission date, time 7. Enter a student_username/ID to find the student submission 8. Click the "View" button to call the file from the submission database 9. End

UC007: Download Submission

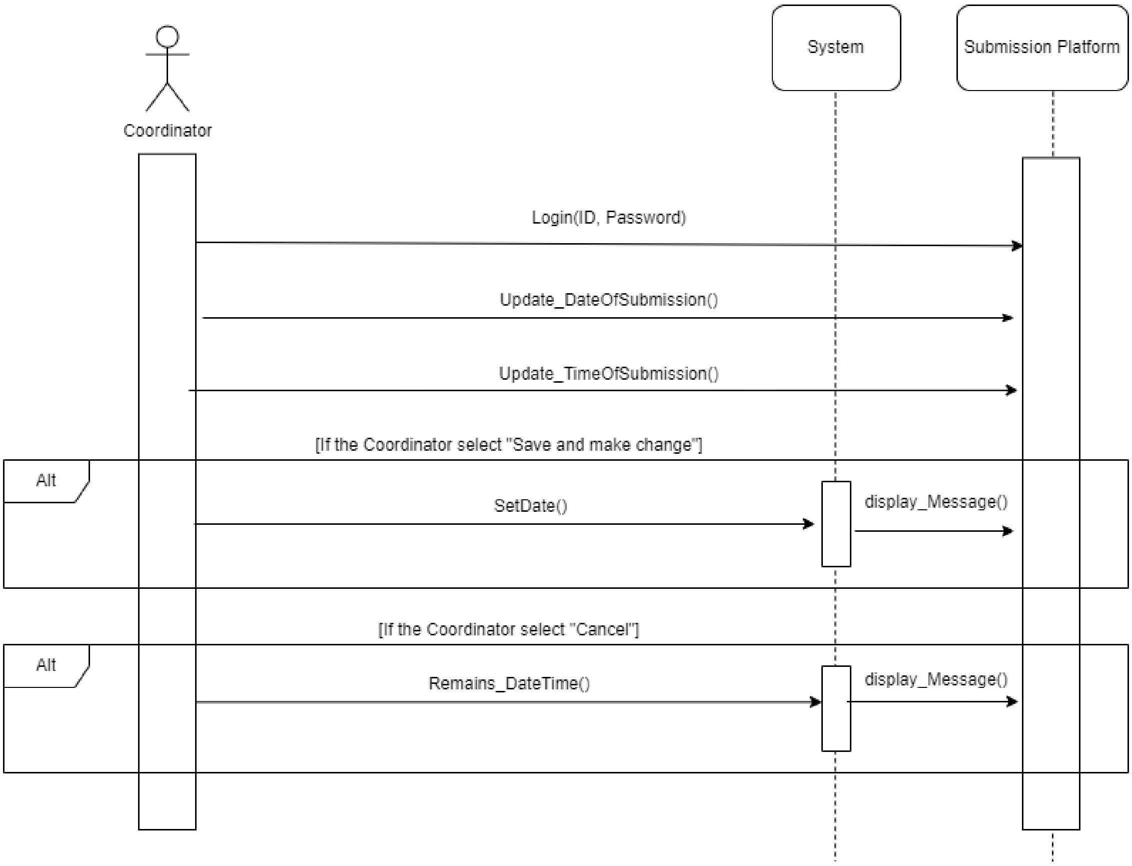
Entity Name	Examiner, StudentReport, SubmissionManager
Method Name	download()
Input	subject code, student_username
Output	Submission file download from database to examiner local drive
Algorithm	1. Start 2. Login System as an examiner 3. Open "Download Submission" on the main page 4. Check the submission details include subject code, subject name, submission due, subject description 5. Select to view a list of submissions 6. System shows all students who made a submission and their submission date, time 7. Enter a student_username/ID to find the student submission 8. Click the "Download" button to download the file to the local drive 9. End

4.2.2.2.**Sequence Diagram**

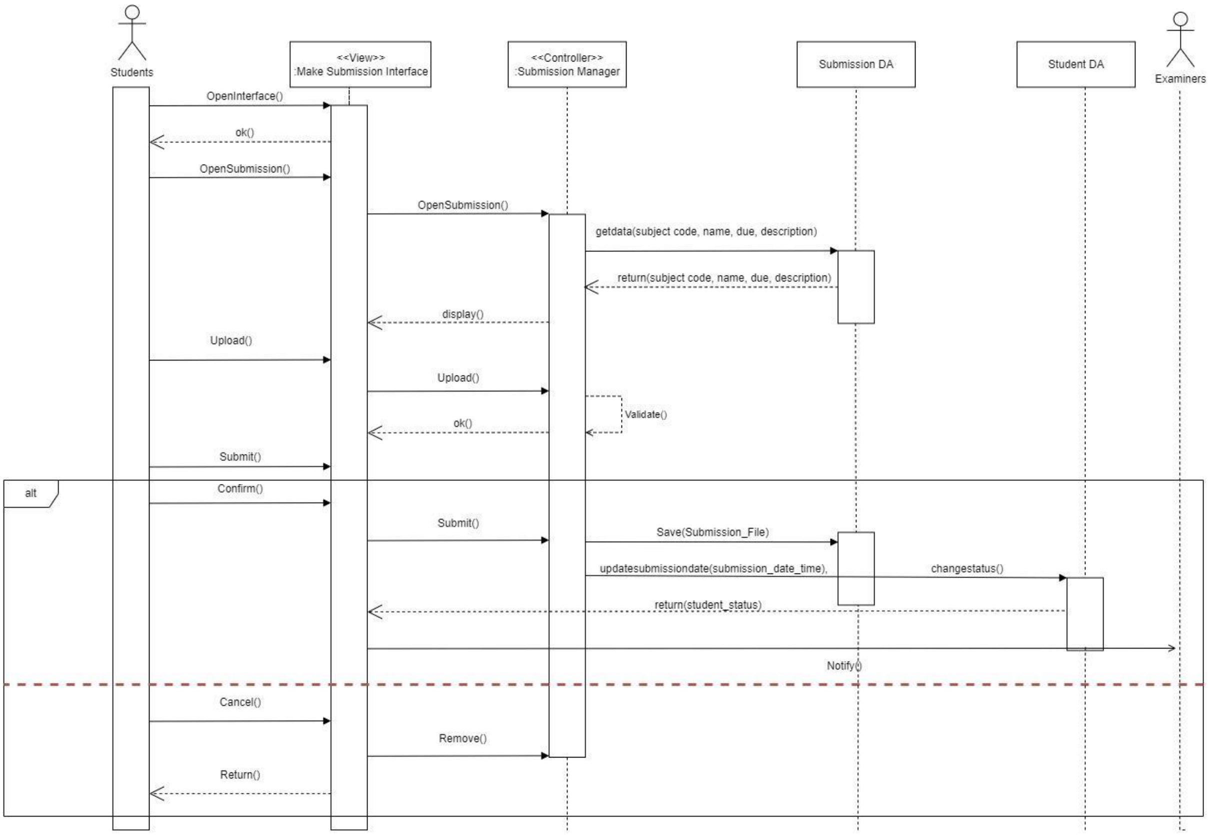
a.) SD003: Sequence Diagram for Create Submission Platform



b.) SD004: Sequence Diagram for Change Time Date Of Submission

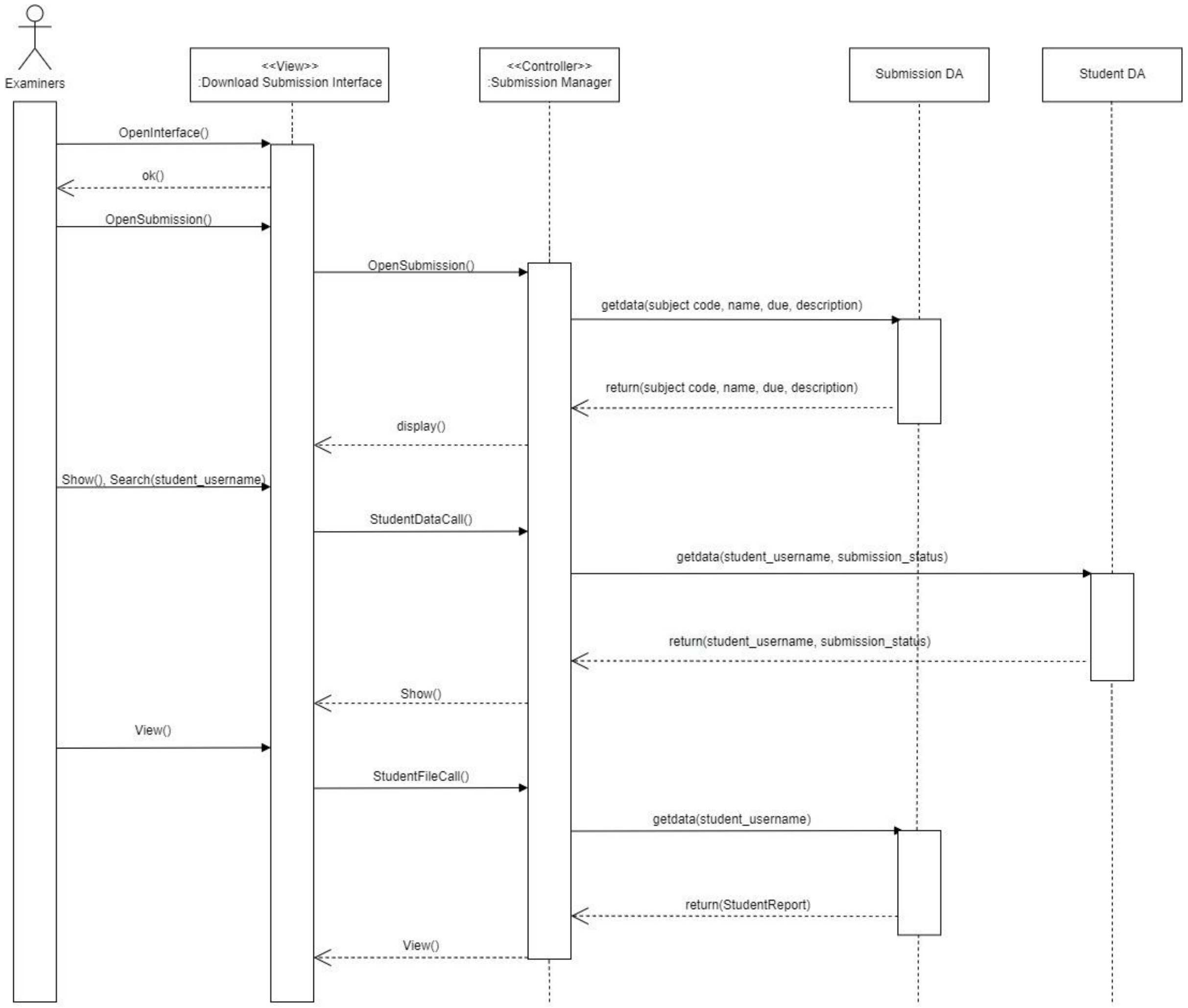

System
Submission Platform
Coordinator
Login(ID, Password)
Update_DateOfSubmission()
Update_TimeOfSubmission()
[If the Coordinator select "Save and make change"]
Alt
SetDate()
display_Message()
[If the Coordinator select "Cancel"]
Alt
Remains_DateTime()
display_Message()

c.) SD005: Sequence Diagram for Make Submission



SD5

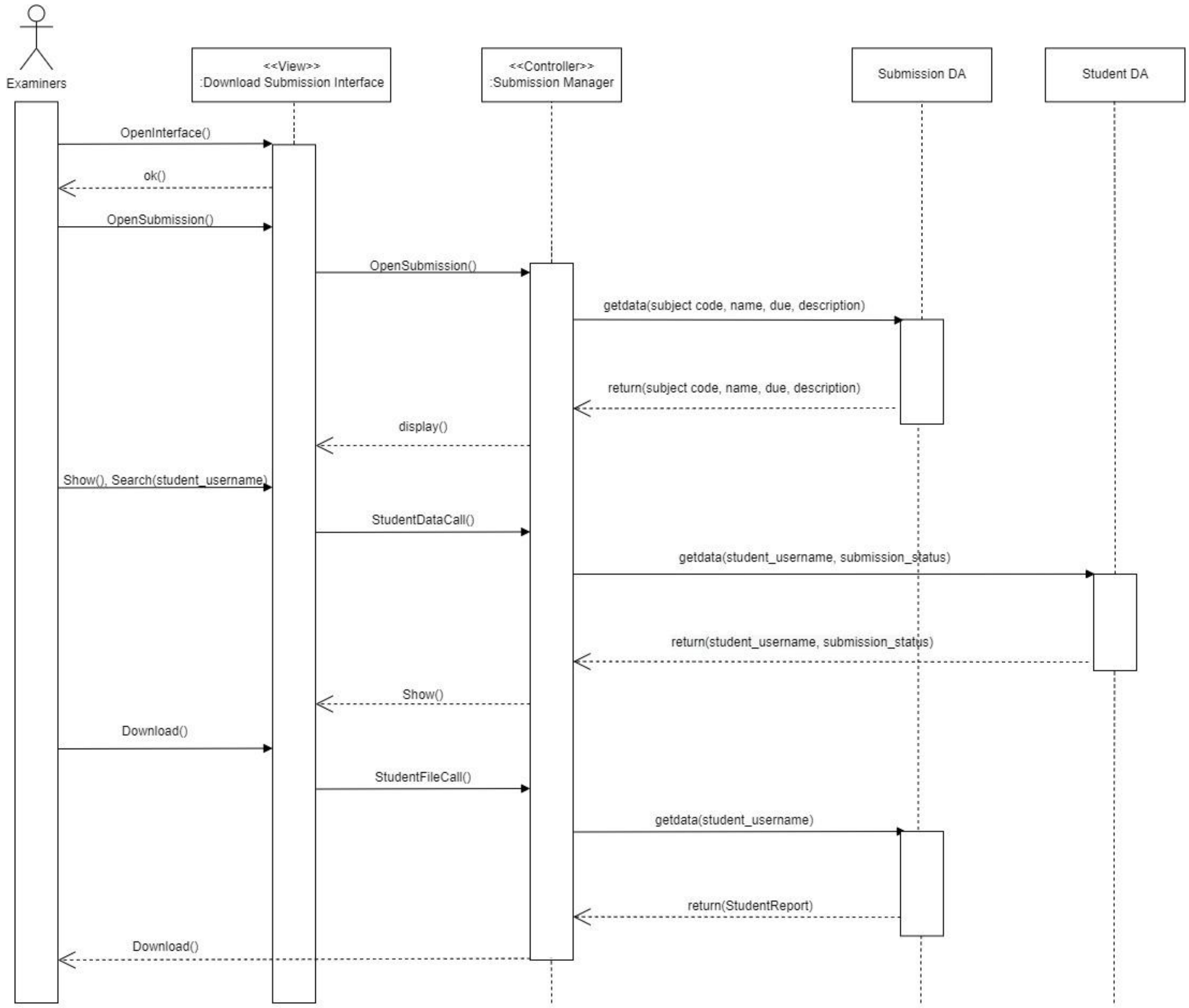
d.) SD006: Sequence Diagram for View Submission



Examiners
<<View>>
:Download Submission Interface
<<Controller>>
:Submission Manager
Submission DA
Student DA
OpenInterface()
ok()
OpenSubmission()
OpenSubmission()
getdata(subject code, name, due, description)
return(subject code, name, due, description)
display()
Show(), Search(student_username)
StudentDataCall()
getdata(student_username, submission_status)
return(student_username, submission_status)
Show()
View()
StudentFileCall()
getdata(student_username)
return(StudentReport)
View()

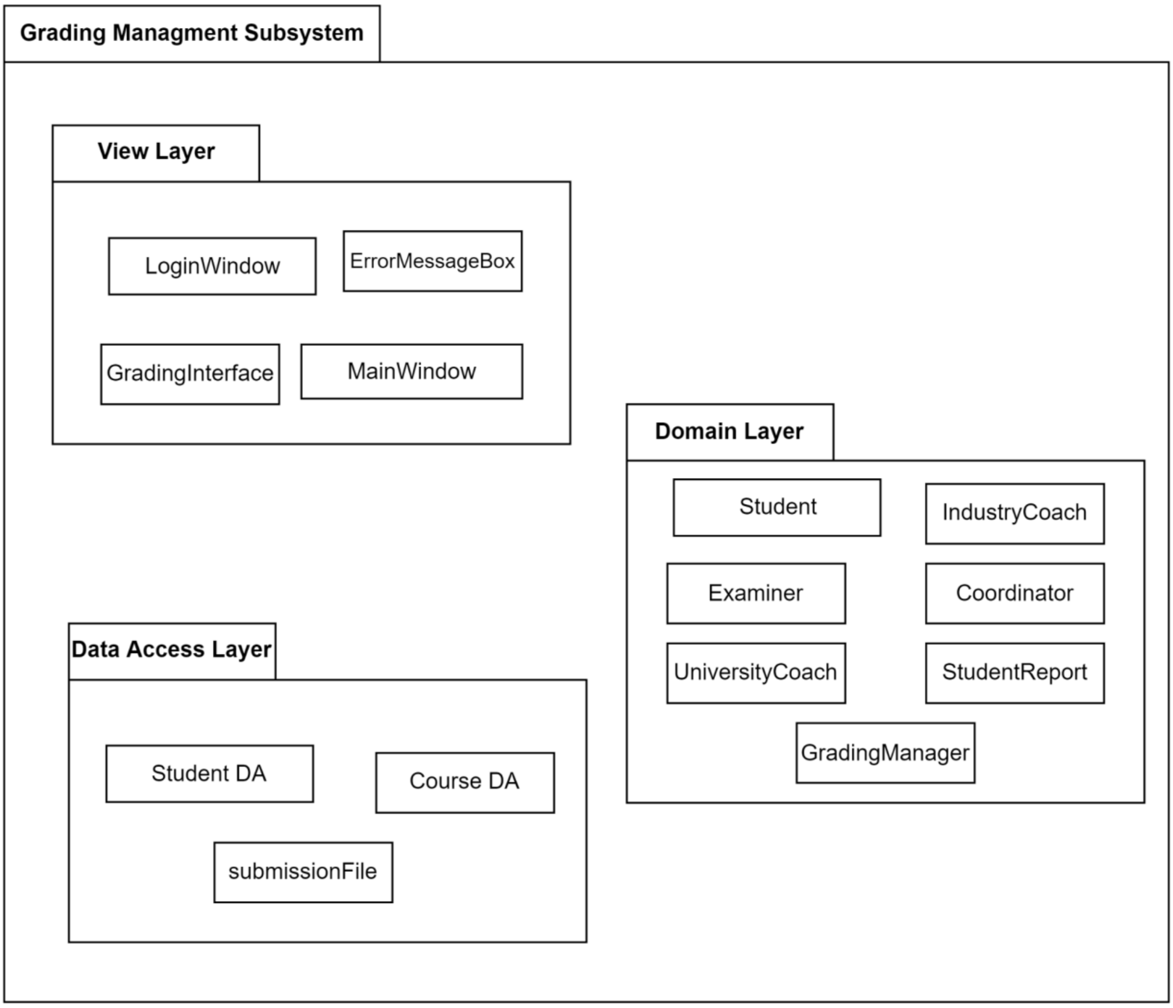
SD6

e.) SD007: Sequence Diagram for Download Submission

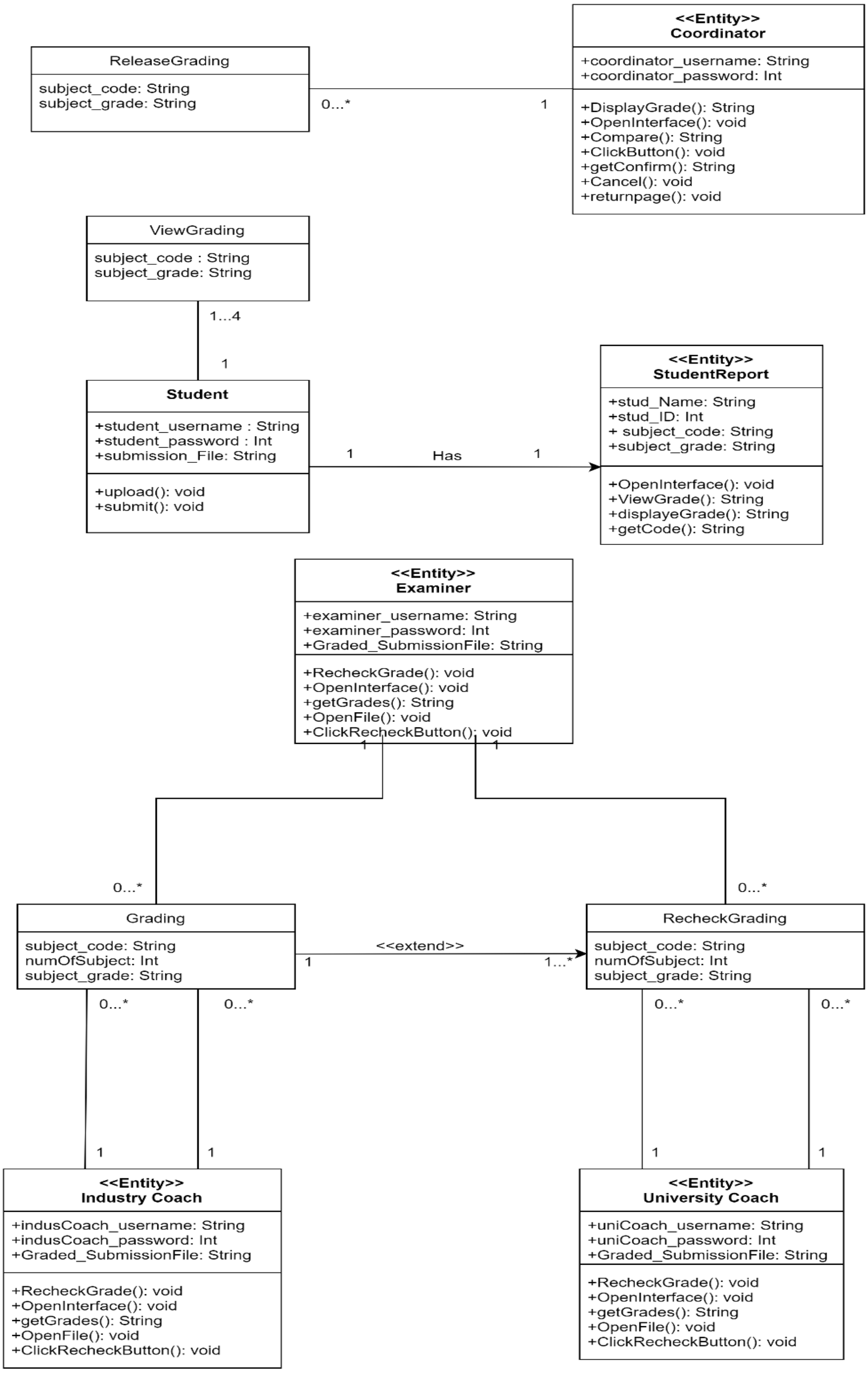

Examiners
<<View>>
:Download Submission Interface
<<Controller>>
:Submission Manager
Submission DA
Student DA
OpenInterface()
ok()
OpenSubmission()
OpenSubmission()
getdata(subject code, name, due, description)
return(subject code, name, due, description)
display()
Show(), Search(student_username)
StudentDataCall()
getdata(student_username, submission_status)
return(student_username, submission_status)
Show()
Download()
StudentFileCall()
getdata(student_username)
return(StudentReport)
Download()

SD7

4.2.3.**P003: <Grading Management> Subsystem**



4.2.3.1.**Class Diagram**



UC008: Grading

Entity Name	Examiner, Industry Coach, University Coach

Method Name	getGrades()
	OpenFile()
	OpenInterface()

Input	Submission_File, subject_code
Output	Graded_SubmissionFile
Algorithm	2.Start 3.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 4.User submit Graded_SubmissionFile to Grading Interface 5.Save and record the Grades 6.End

UC009: Recheck Grading

Entity Name	Examiner, Industry Coach, University Coach

Method Name	RecheckGrade()
	OpenInterface()
	getGrades()
	OpenFile()
	ClickRecheckButton()

Input	Graded_SubmissionFile, subject_code
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

Entity Name	StudentReport

Method Name	OpenInterface()
	ViewGrade(): String
	displayeGrade(): String
	getCode()

Input	subject_code
Output	Grades of a subject has been displayed
Algorithm	1. Start 2. Select Grading Interface 3. Select Enter Subject Code 4. Student enter Subject Code 4.1 IF code is valid 4.1.1 Display Grade 4.2 ELSE 4.2.1 Return Grading Interface 5. End

UC011: View Grading

Entity Name	Coordinator

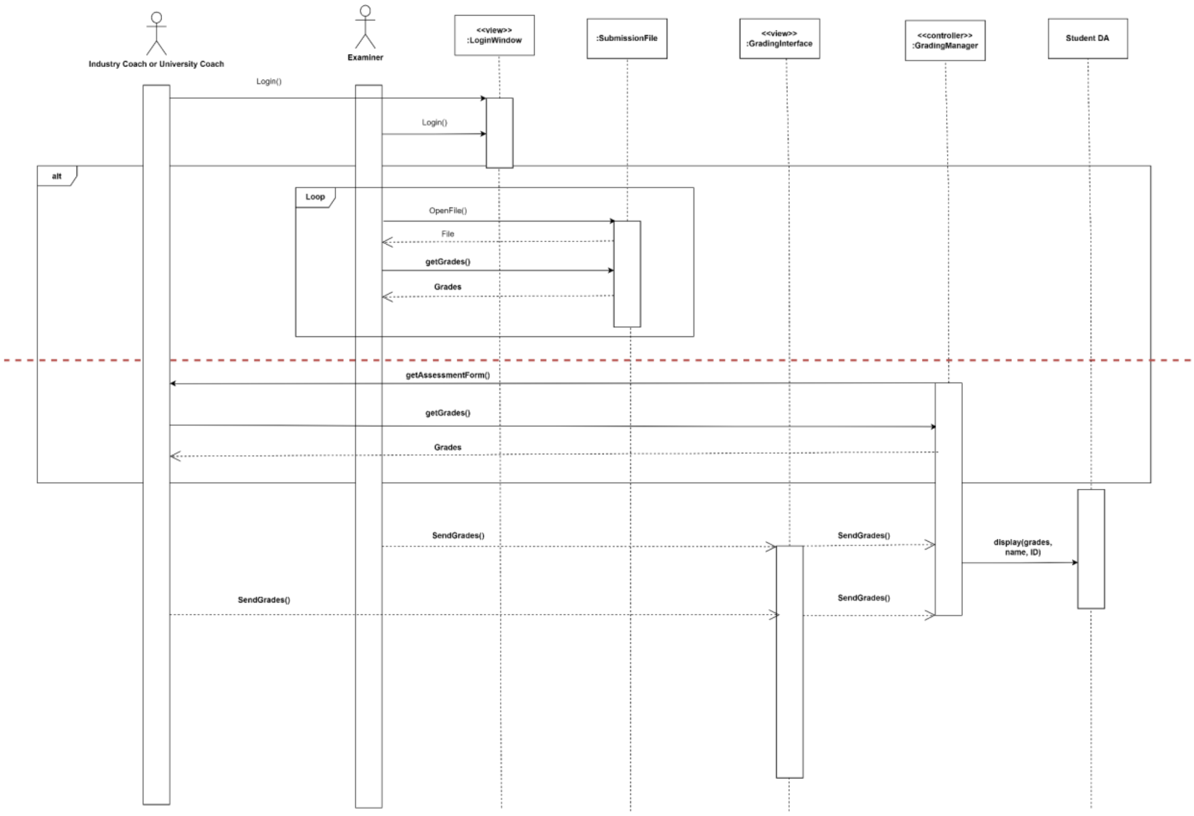
Method Name	DisplayGrade(): String
	OpenInterface()
	Compare()
	ClickButton()
	getConfirm()
	Cancel()
	returnpage()

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

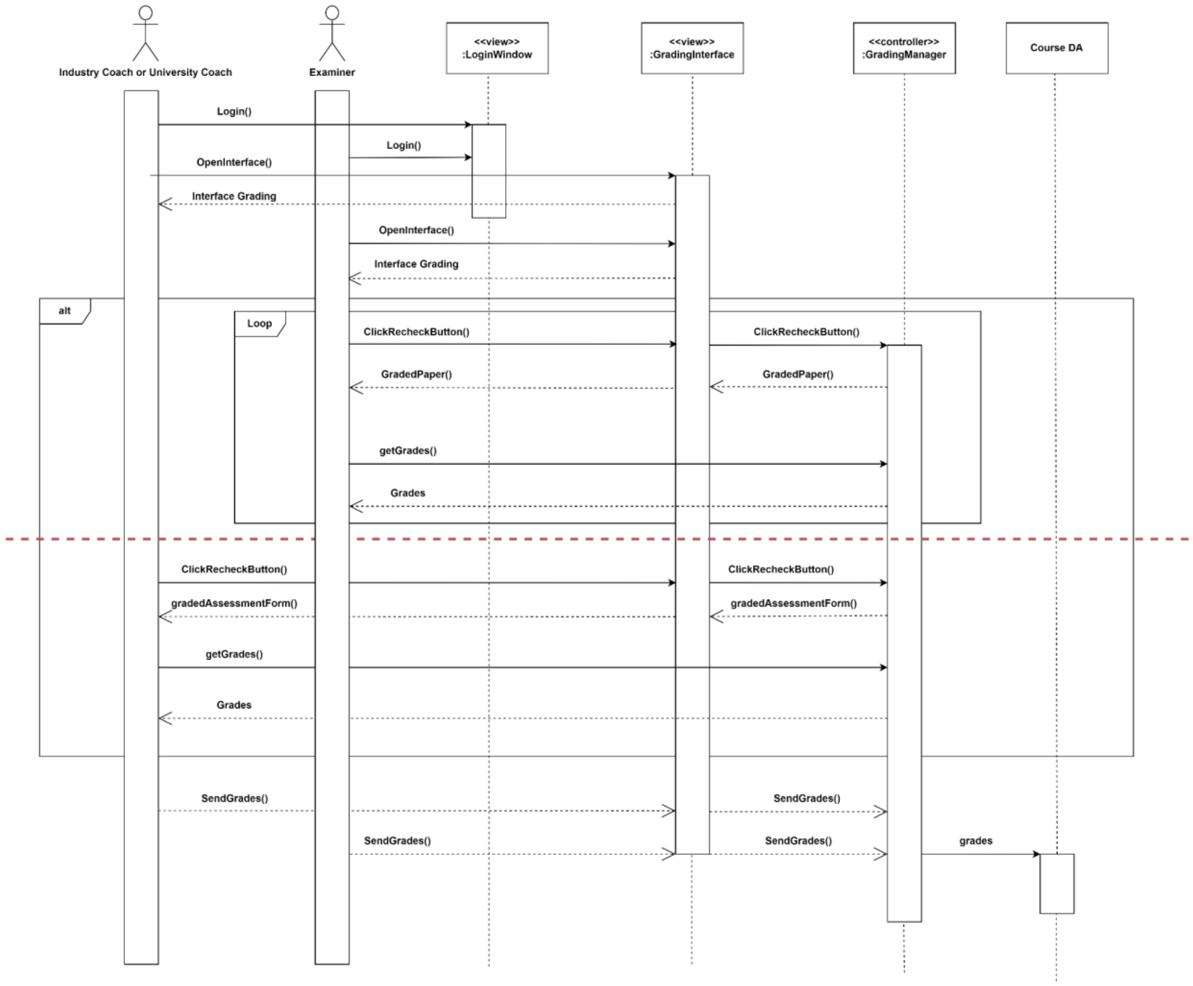
	6.2.1 Return Grading Interface 7. END

4.2.3.2. Sequence Diagram

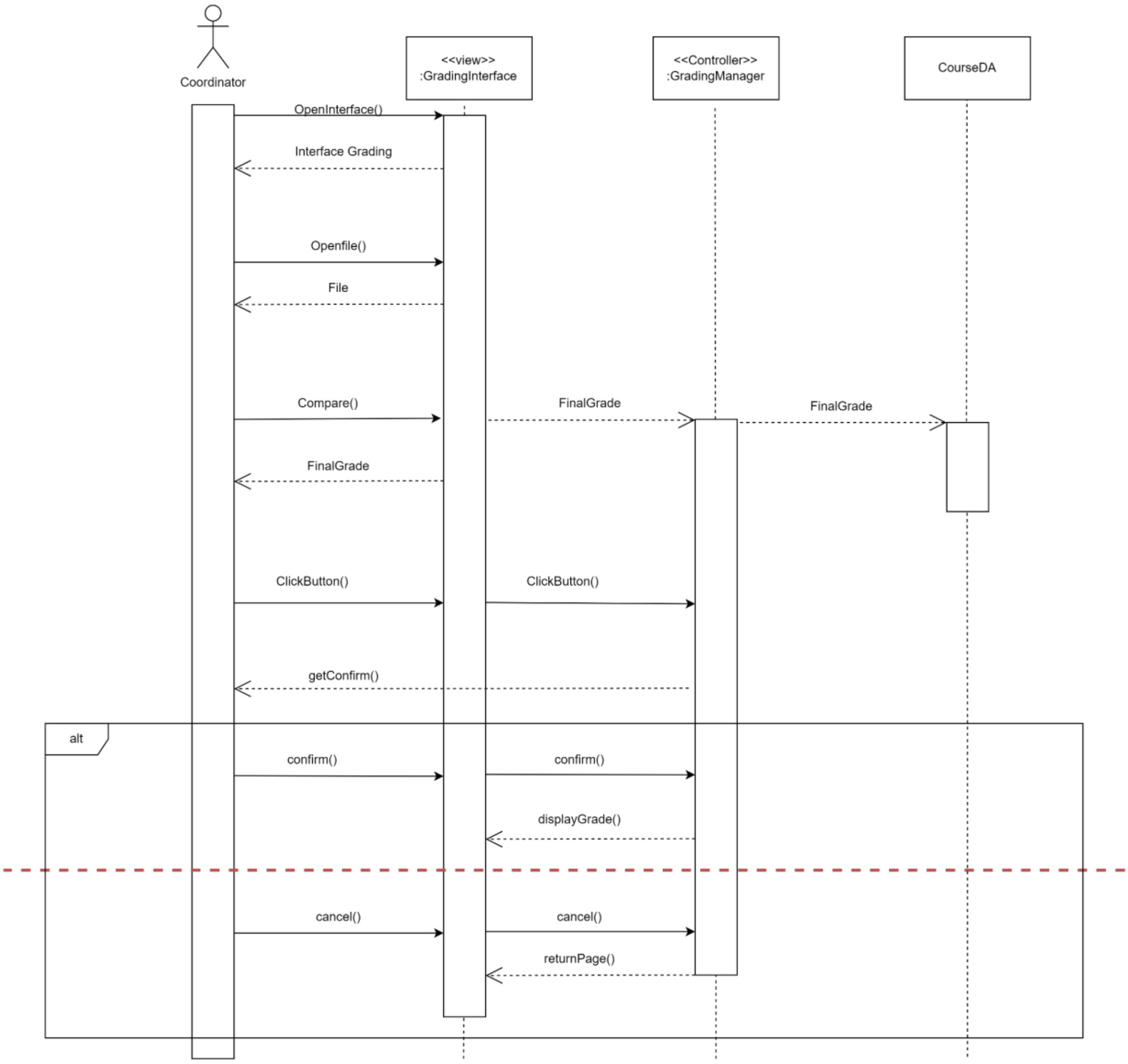
a) SD008: Sequence Diagram for Grading



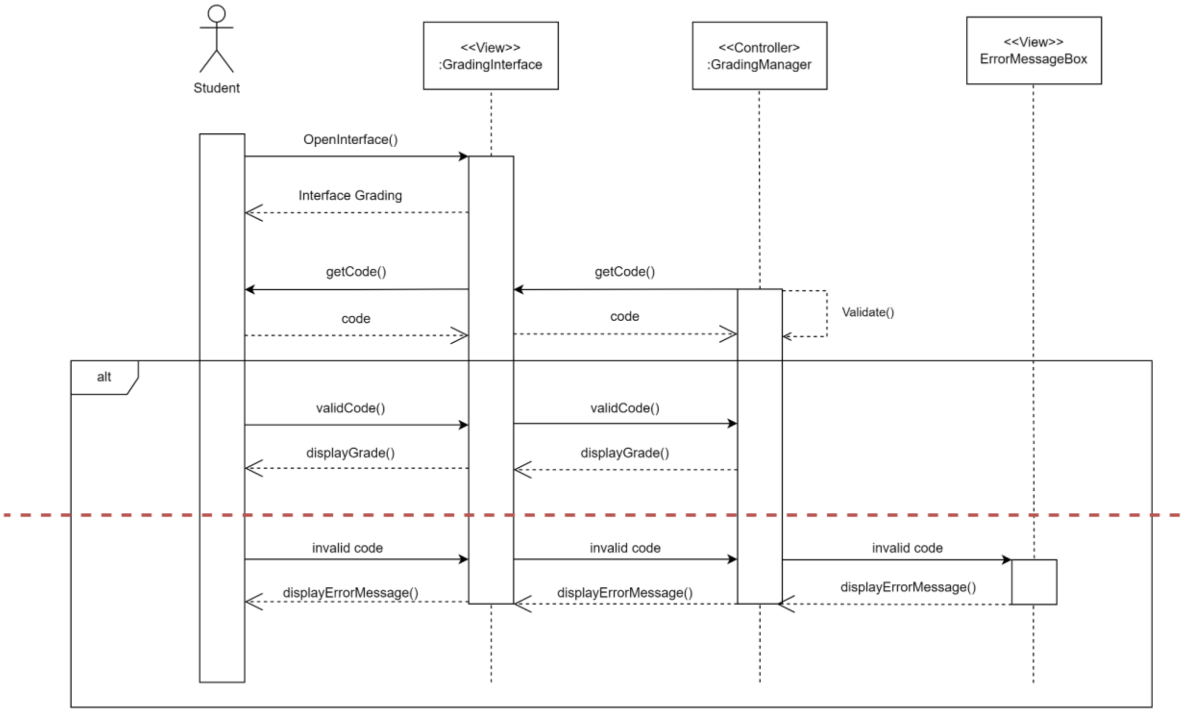
b) Sequence Diagram for Recheck Grading



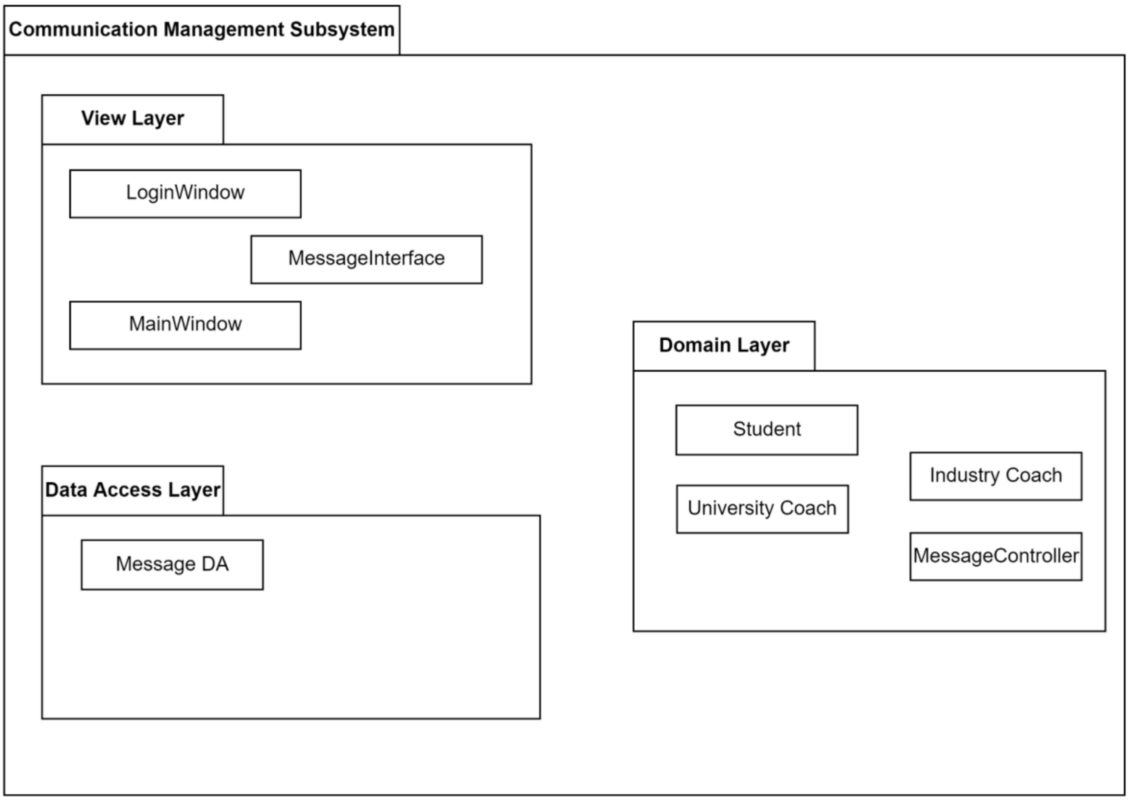
c) Sequence Diagram for Release Grading



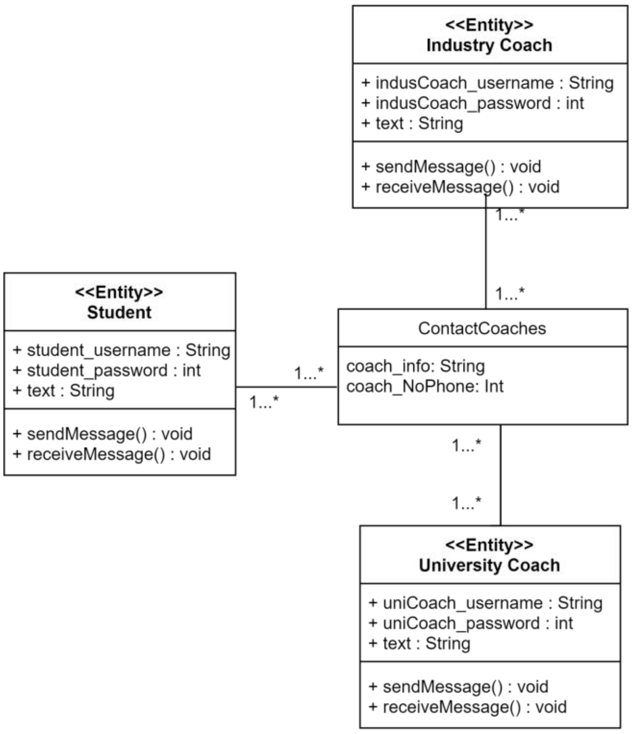
d) Sequence Diagram for View Grading



4.2.4 P004: **<Communication Management> Subsystem**



4.2.4.1 Class Diagram



UC012: Contact Coaches

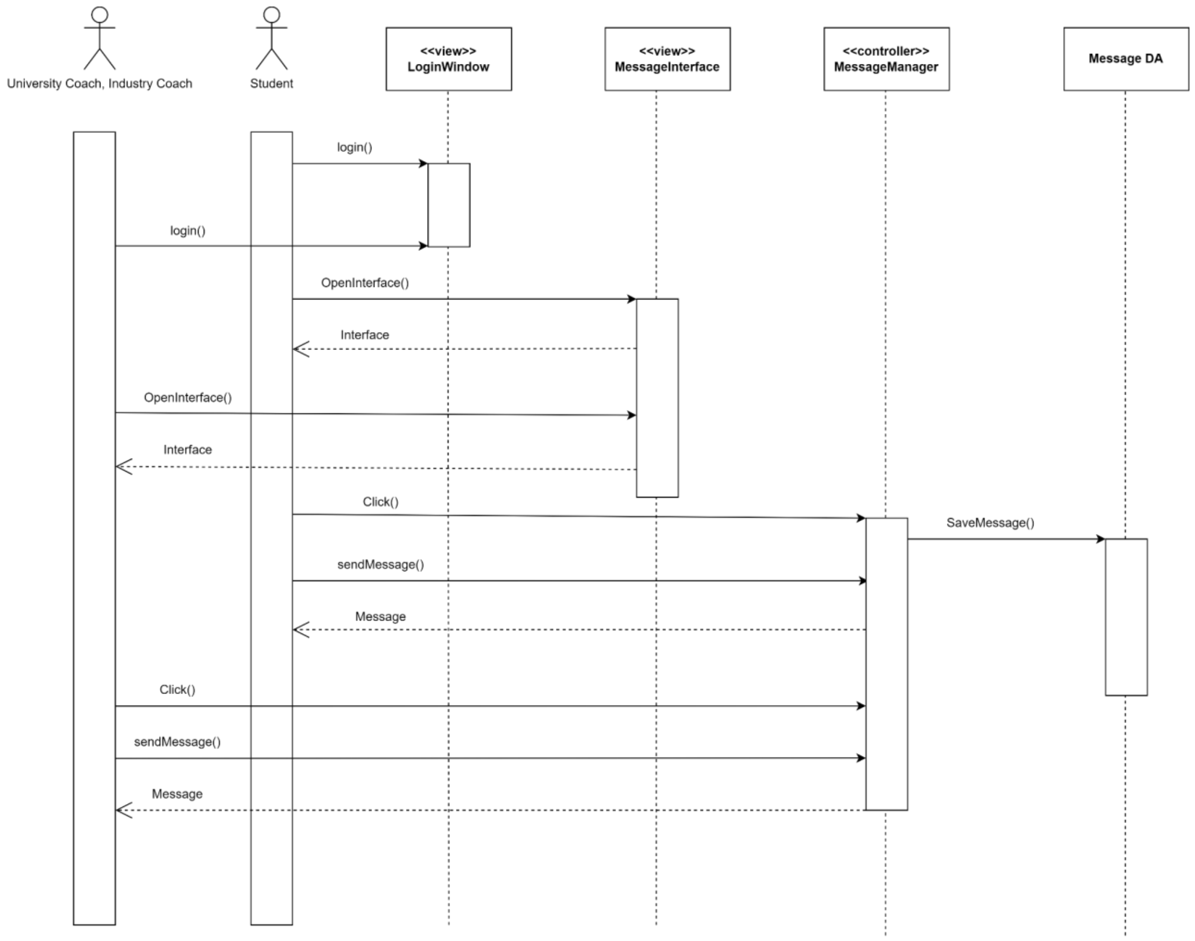
Entity Name	Student, Industry Coach, University Coach

Method Name	sendMessage()
	receiveMessage()

Input	text
Output	Message sent successfully
Algorithm	1. Start 2. Login to system 3. Select Communication Interface 4. Select Coach Contact 5. Input text 6. Send message 7. Display message 8. End

4.2.4.2 Sequence Diagram

Sequence Diagram for Contact Coaches



5.Data Design

5.1.Data Description

The major data or systems entities are stored into a relational database named as Maniac System, processed and organized into *17* entities as listed in Table 5.1.

Table 5.1: Description of Entities in the Database

No.	Entity Name	Description
1.	Student	It will store Student Information (stud_username, stud_password) in order to be able to login to the system. It will also contain attributes (Submission_File) as Students need a submission file in order to receive a grade.
2.	Examiner	It will store Examiner Information (examiner_username, examiner_password) in order to be able to login to the system and do grading.
3.	Industry Coach	It will store Industry Coach Information (indusCoach_username, indusCoach_password) to be able to login to the system and do grading.
4.	University Coach	It will store Industry Coach Information (uniCoach_username, uniCoach_password) to be able to login to the system and do grading.
5.	Coordinator	It will store Industry Coach Information (coordinator_username, coordinator_password) to be able to login to the system.
6.	StudentReport	It will be showing the final year project of each where it will display (stud_Name, stud_ID, subject_code, subject_grade). The student must enter a valid subject_code in order to view the grading.
7.	ViewNotification	Here it will notify the user on important notice (notification_id), for Students it will notify about the Submission Time and Date, meanwhile for Examiner will be notified about the Grading, whereas for Industry Coach and University Coach

		will be notified about Grading and also Messages. Coordinator can view all the notifications.
8.	DeleteNotification	Here it will delete unnecessary notifications that can only be done by the Coordinator, it will contain (notification_id).
9.	CreateSubmissionPlatform	Here it will have a platform for students to submit their Final Year Project (Submission_File, subject_code, student_username, submission_status, submission_date_time), this is done by the Coordinator.
10.	ChangeTimeDateOfSubmisison	Here it will be changing the time and date of Submission, (submission_date_time) this is doned by the Coordinator only.
11.	MakeSubmission	Here it will show that Students can make a submission for their Final Year Project (Submission_File, subject_code, student_username, submission_status, submission_date_time).
12.	ViewSubmission	Here it will be showing the Submission File by the Students where it can be viewed by the Examiner, University Coach and Industry Coach. (Submission_File, subject_code, student_username, submission_status, submission_date_time).
12.	DownloadSubmission	Here it will be allowing the downloading of submission files in order to do grading. (Submission_File, subject_code, student_username, submission_status, submission_date_time).
13.	Grading	Here it will be allowing the User to do a grading on the downloaded submission file. (Submission_File, subject_code, student_username, subject_grade)
14.	RecheckGrading	Here it will be making sure the grades of the graded Submission_File. are graded fair and square, (Graded_SubmissionFile, subject_code, student_username, subject_grade)
15.	Release Grading	Here it will be releasing the final grades of the students. (subject_code, subject_grades).

16.	View Grading	Here it will display the report of Final Year for the students, (stud_Name, stud_ID, subject_code, Final_grade).
17.	ContactCoaches	Here it will show the conversation of between University Coach or Industry Coach with Students, (text)

5.2.**Data Dictionary**

5.2.1.**Entity: <Student>**

Attribute Name	**Type**	**Description**
stud_username	String	Username of the Student
stud_password	Int	Password of the Student
Submission_File	String	Submission File for each subject of the Student

5.2.2.**Entity: <Examiner>**

Attribute Name	**Type**	**Description**
examiner_username	String	Username of the Examiner
examiner_password	Int	Password of the Examiner

5.2.3.**Entity: <University Coach>**

Attribute Name	**Type**	**Description**
uniCoach_username	String	Username of the University Coach
uniCoach_password	Int	Password of the University Coach

5.2.4.**Entity: <Industry Coach>**

Attribute Name	**Type**	**Description**
indusCoach_username	String	Username of the Industry Coach
indusCoach_password	Int	Password of the Industry Coach

5.2.5.**Entity: <Coordinator>**

Attribute Name	**Type**	**Description**
coordinator_username	String	Username of the Coordinator
coordinator_password	Int	Password of the Coordinator

5.2.6.**Entity: <StudentReport>**

Attribute Name	**Type**	**Description**
stud_Name	String	Name of Student
stud_ID	Int	Uniquely identifies the Student
subject_code	String	Uniquely identifies the code of subjects
Final_grade	String	The final grade of the subject for each student

5.2.7.Entity: <ViewNotification>

Attribute Name	**Type**	**Description**
notification_id	Int	Uniquely Identifies the ID of notification

5.2.8.Entity: <DeleteNotification>

Attribute Name	**Type**	**Description**
notification_id	Int	Uniquely Identifies the ID of notification

5.2.9.Entity: <CreateSubmissionPlatform>

Attribute Name	**Type**	**Description**
Submission_File	String	Submission File for each subject of the Student
subject_code	String	Uniquely identifies the code of subjects
student_username	String	Username of the Student
submission_status	String	Identifies the status of the submission
submission_date_time	String	Identifies the date and time of submission

5.2.10.Entity: <ChangeTimeDateOfSubmission>

Attribute Name	Type	Description
submission_date_time	String	Identifies the date and time of submission

5.2.11.Entity: <MakeSubmission>

Attribute Name	Type	Description
Submission_File	String	Submission File for each subject of the Student
subject_code	String	Uniquely identifies the code of subjects
student_username	String	Username of the Student
submission_status	String	Identifies the status of the submission
submission_date_time	String	Identifies the date and time of submission

5.2.12.Entity: <ViewSubmission>

Attribute Name	Type	Description
Submission_File	String	Submission File for each subject of the Student
subject_code	String	Uniquely identifies the code of subjects
student_username	String	Username of the Student
submission_status	String	Identifies the status of the submission
submission_date_time	String	Identifies the date and time of submission

5.2.13.Entity: <DownloadSubmission>

Attribute Name	Type	Description
Submission_File	String	Submission File for each subject of the Student
subject_code	String	Uniquely identifies the code of subjects
student_username	String	Username of the Student
submission_status	String	Identifies the status of the submission
submission_date_time	String	Identifies the date and time of submission

5.2.14.Entity: <Grading>

Attribute Name	Type	Description
Submission_File	String	Submission File for each subject of the Student
subject_code	String	Uniquely identifies the code of subjects
student_username	String	Username of the Student
subject_grade	String	The grades for each subject of the Student

5.2.15.Entity: <RecheckGrading>

Attribute Name	Type	Description
Graded_SubmissionFile	String	Graded Submission File for each subject of the Student
subject_code	String	Uniquely identifies the code of subjects
student_username	String	Username of the Student
subject_grade	String	The grades for each subject of the Student

5.2.16.Entity: <ReleaseGrading>

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

5.2.17.Entity: <ViewGrading>

Attribute Name	Type	Description
stud_Name	String	Name of Student
stud_ID	Int	Uniquely identifies the Student
subject_code	String	Uniquely identifies the code of subjects
Final_grade	String	The final grade of the subject for each student

5.2.18.Entity: <ContactCoaches>

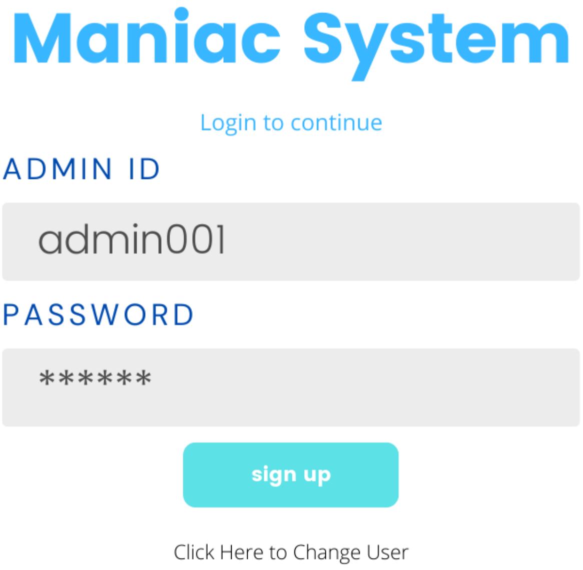
Attribute Name	Type	Description
text	String	The text messages between University Coach or Industry Coach and Students

6. Interface Design

6.1 Overview of Interface

The interface design for Maniac System focuses on the user/software interface, which are notification interface, submission interface, grading interface and communication interface. Those interfaces also indicate different subsystems. There are 4 types of different users (coordinator, student, examiner, industry and university coaches) having different roles, authorities and accessibilities towards the system functions.

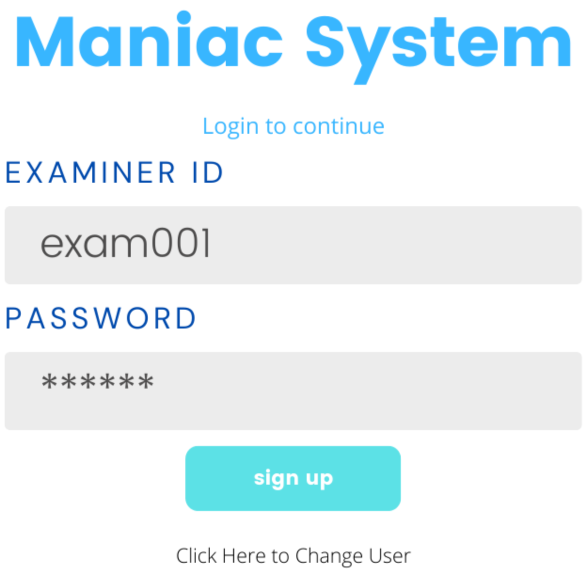
Login Interface



ADMIN ID: admin001

PASSWORD: ******

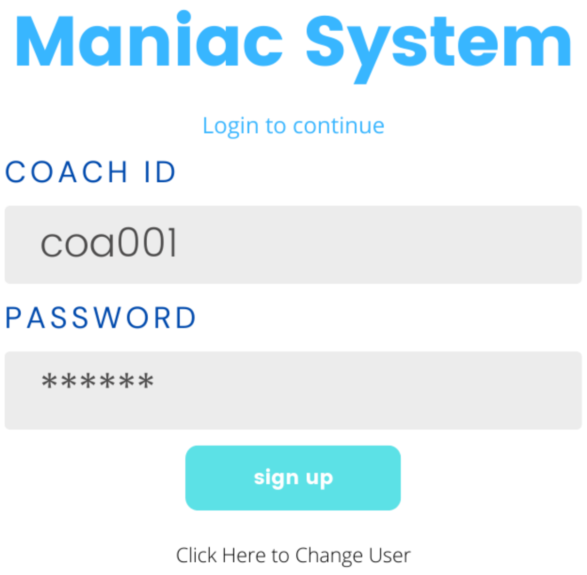
Figure 6.1: Overview of Login Page for Admin



EXAMINER ID: exam001

PASSWORD: ******

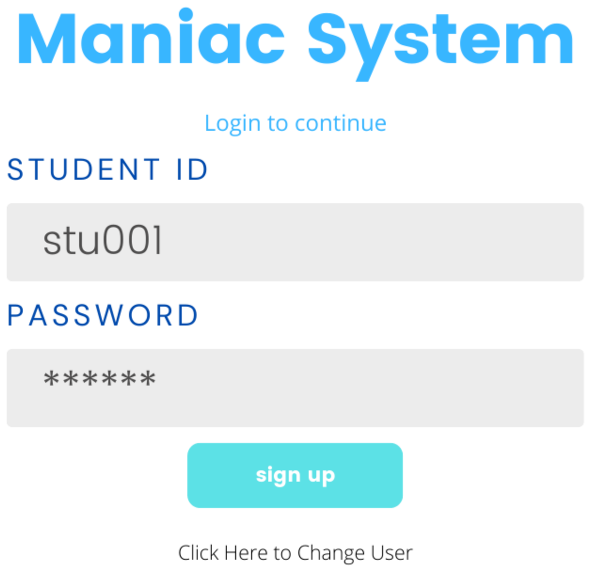
Figure 6.2: Overview of Login Page for Examiner



COACH ID: coa001

PASSWORD: ******

Figure 6.3: Overview of Login Page for Coach



Maniac System

Login to continue

STUDENT ID

stu001

PASSWORD

******

Figure 6.4: Overview of Login Page for Student

Login interface involves all users, who are admin (coordinator), examiner, coach and student. For admin, they have to enter admin ID and password in order to login. Same concept applies to examiner, coach and student, but instead of admin ID, they have to input examiner ID, coach ID or student ID.

Main Page Interface

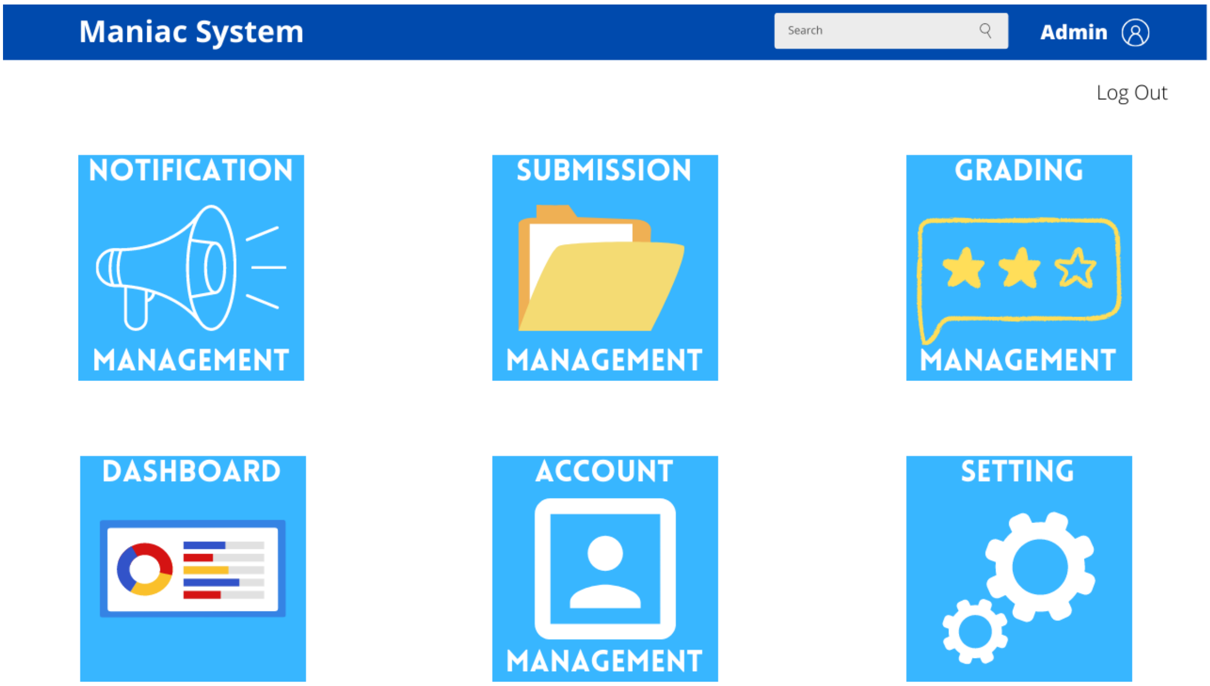


Figure 6.5: Overview of Main Page for Admin

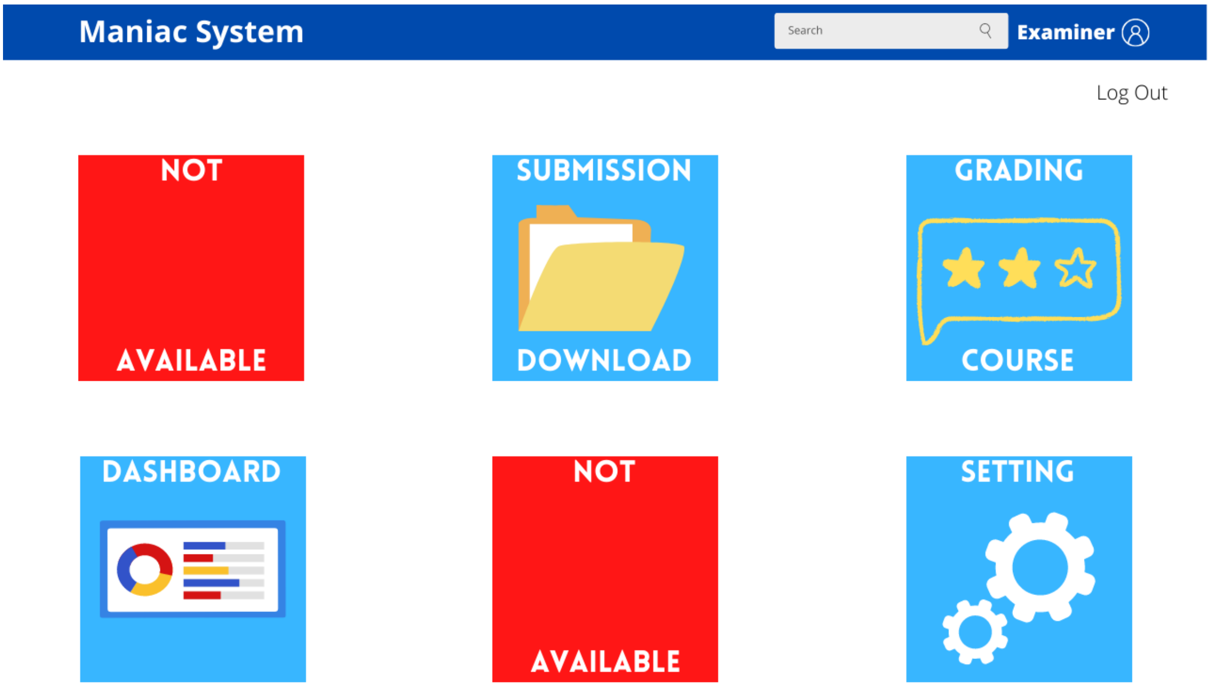


Figure 6.6: Overview of Main Page for Examiner

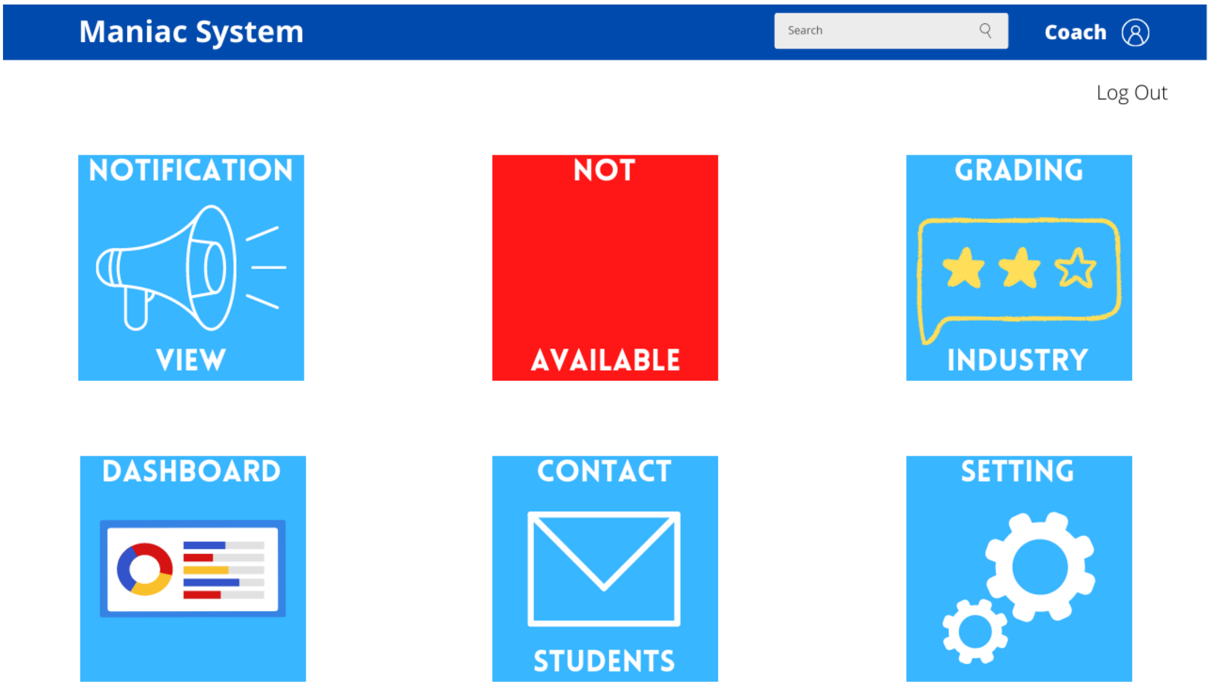


Figure 6.7: Overview of Main Page for Coach

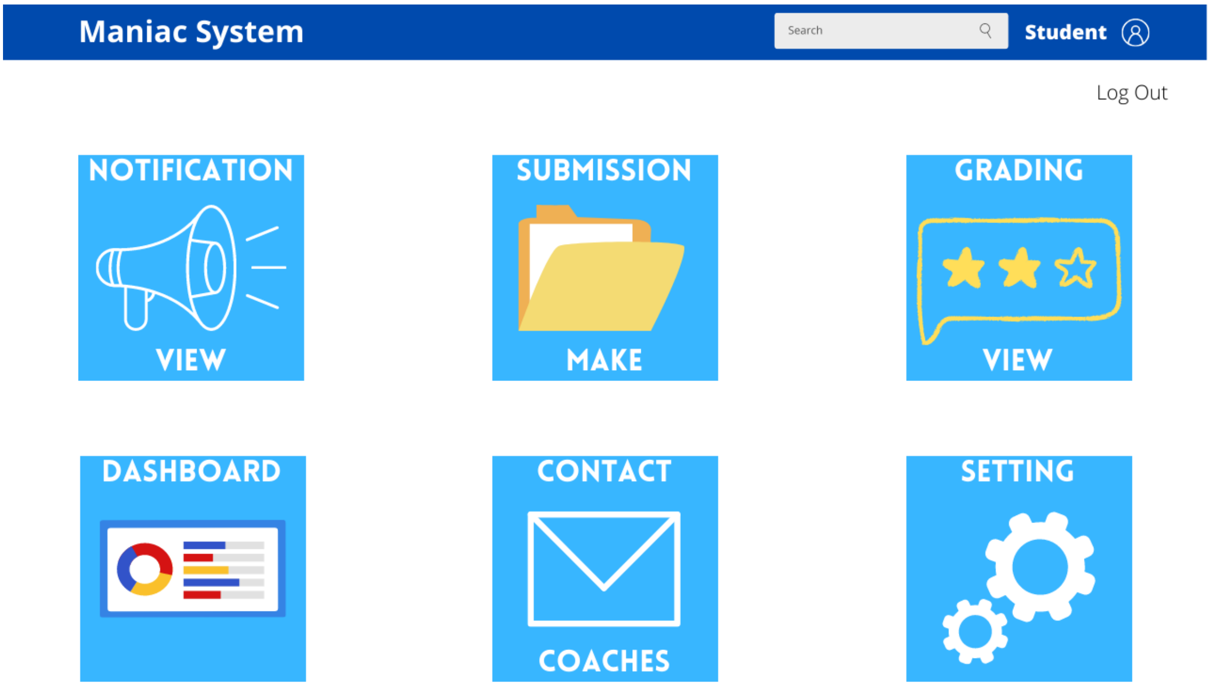
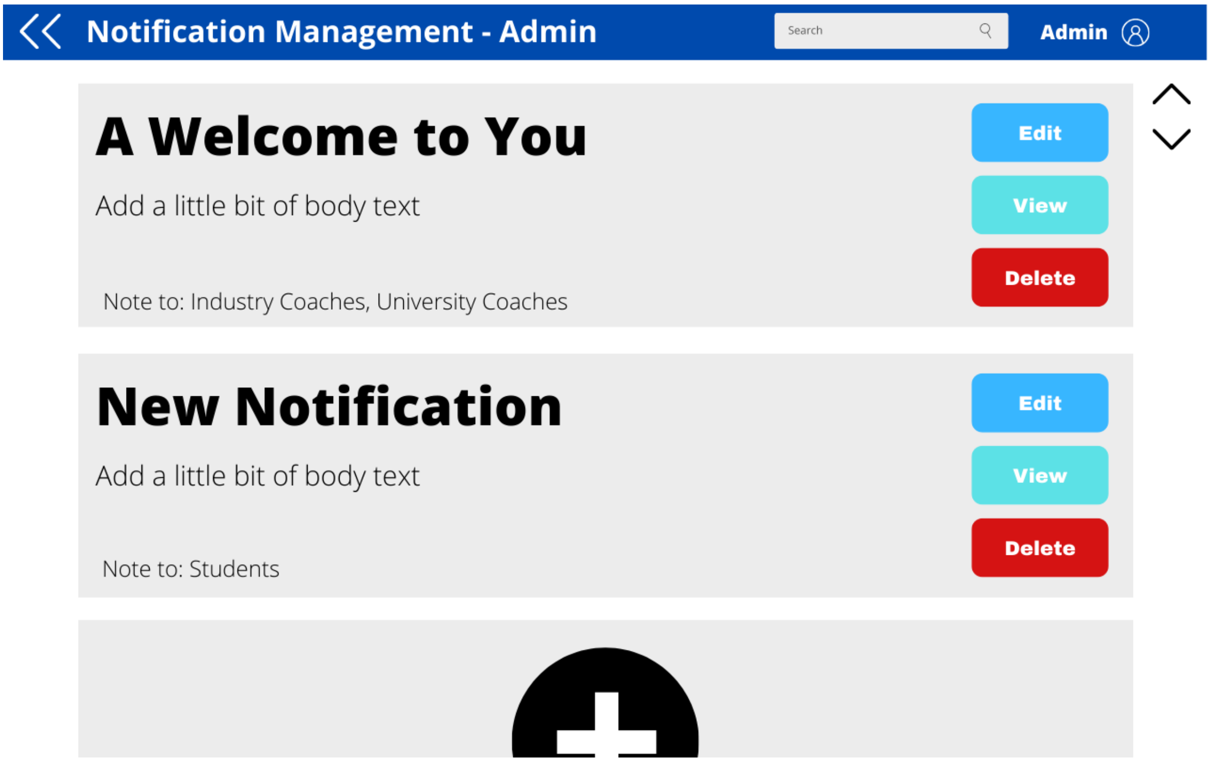


Figure 6.8: Overview of Main Page for Student

Admin, Examiner, Coach and Student are involved in the Main Page Interface. After successfully logged in to the system, the user will be able to access the main page of the system.

All of the users' main pages have a search bar and profile button on the top of the page including the log out button below the header. For the Admin, the main page contains notification management, submission management, grading management, dashboard, account management and setting. The Examiner can access submission download, grading industry, dashboard and setting. Next, the coach has access to notification view, grading industry, dashboard, contact student and setting. Lastly, the student can open the notification view, submission make, grading view, dashboard, contact coach and setting.

Notification Interface



A Welcome to You

Add a little bit of body text

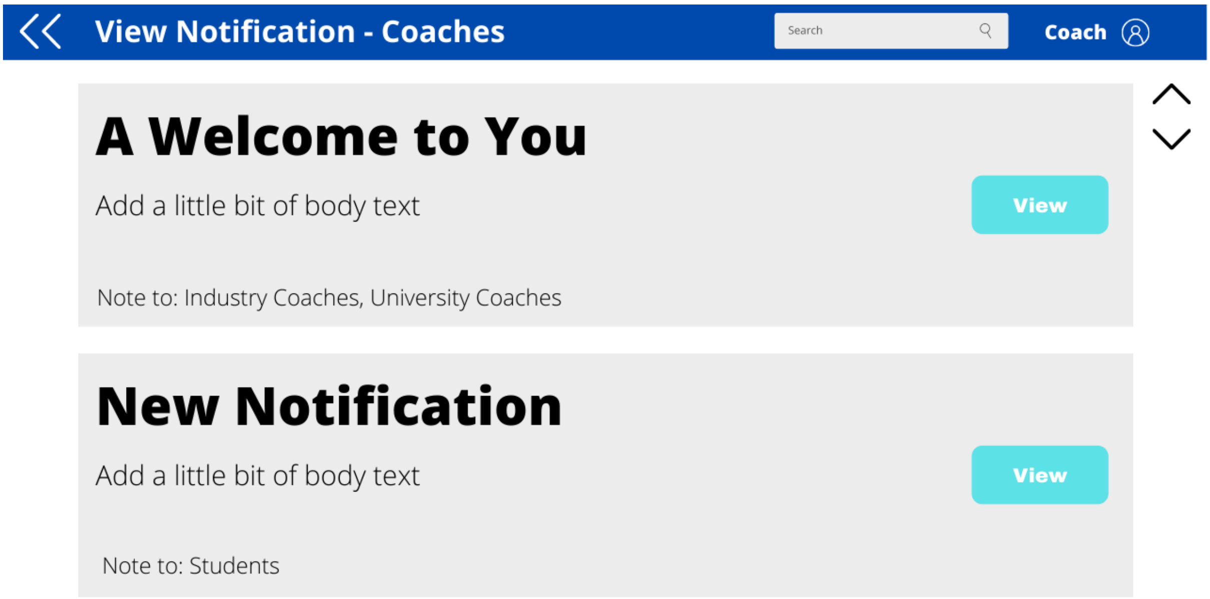
Note to: Industry Coaches, University Coaches

New Notification

Add a little bit of body text

Note to: Students

Figure 6.9: Overview of Notification Management Page for Admin



A Welcome to You

Add a little bit of body text

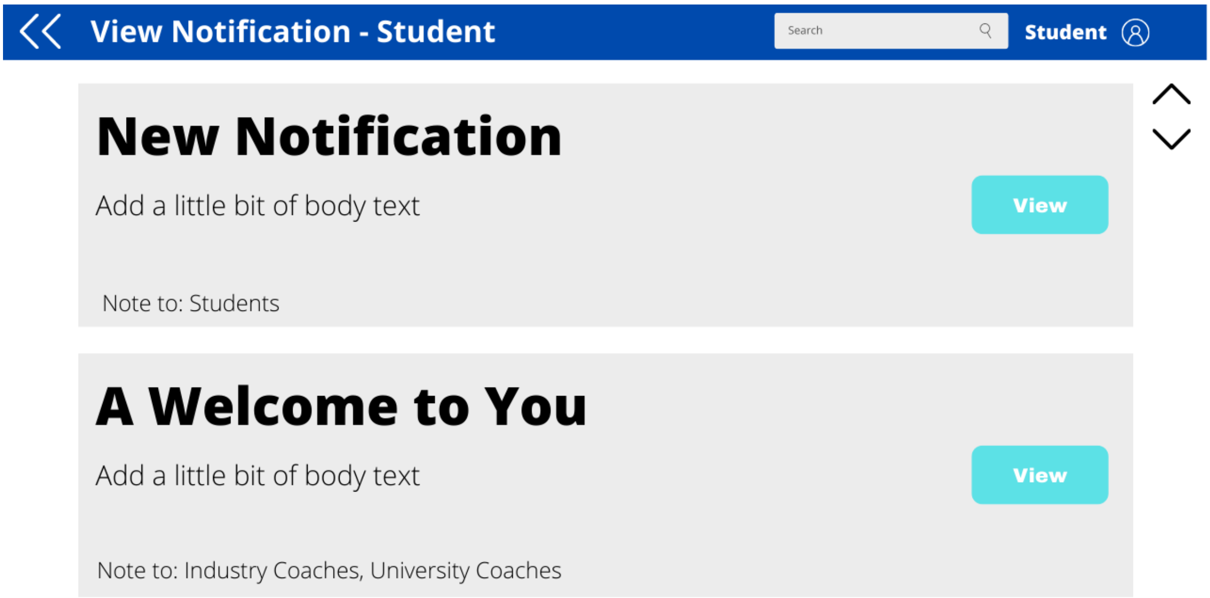
Note to: Industry Coaches, University Coaches

New Notification

Add a little bit of body text

Note to: Students

Figure 6.10: Overview of View Notification Page for Coach



New Notification

Add a little bit of body text

Note to: Students

A Welcome to You

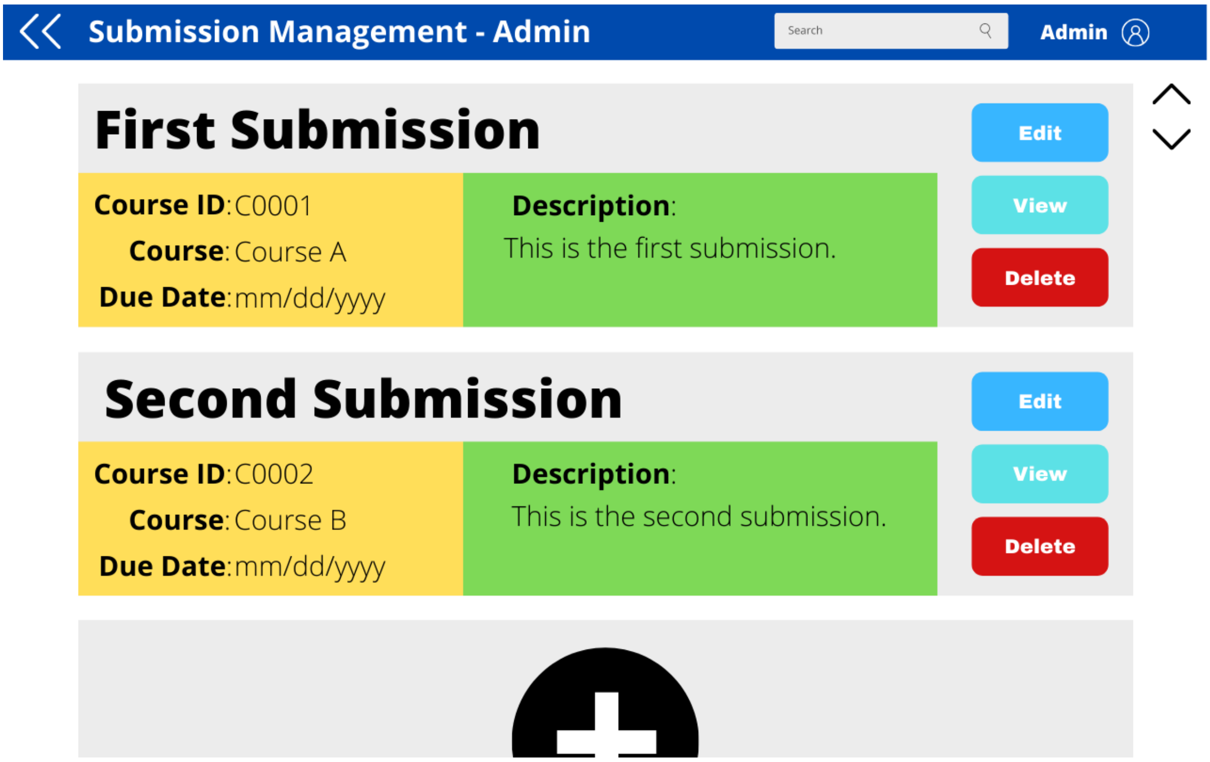
Add a little bit of body text

Note to: Industry Coaches, University Coaches

Figure 6.11: Overview of View Notification Page for Student

Notification Interface (subsystem) only can be accessed by Admin, Coach and Student. The admin can manage the notification by adding a new notification by clicking the add button. The notification also can be edited, viewed and deleted. For Coach and Student, they only can view the notification once they received the notification from the system.

Submission Interface


Submission Management - Admin

First Submission

Course ID:C0001
Course: Course A
Due Date:mm/dd/yyyy

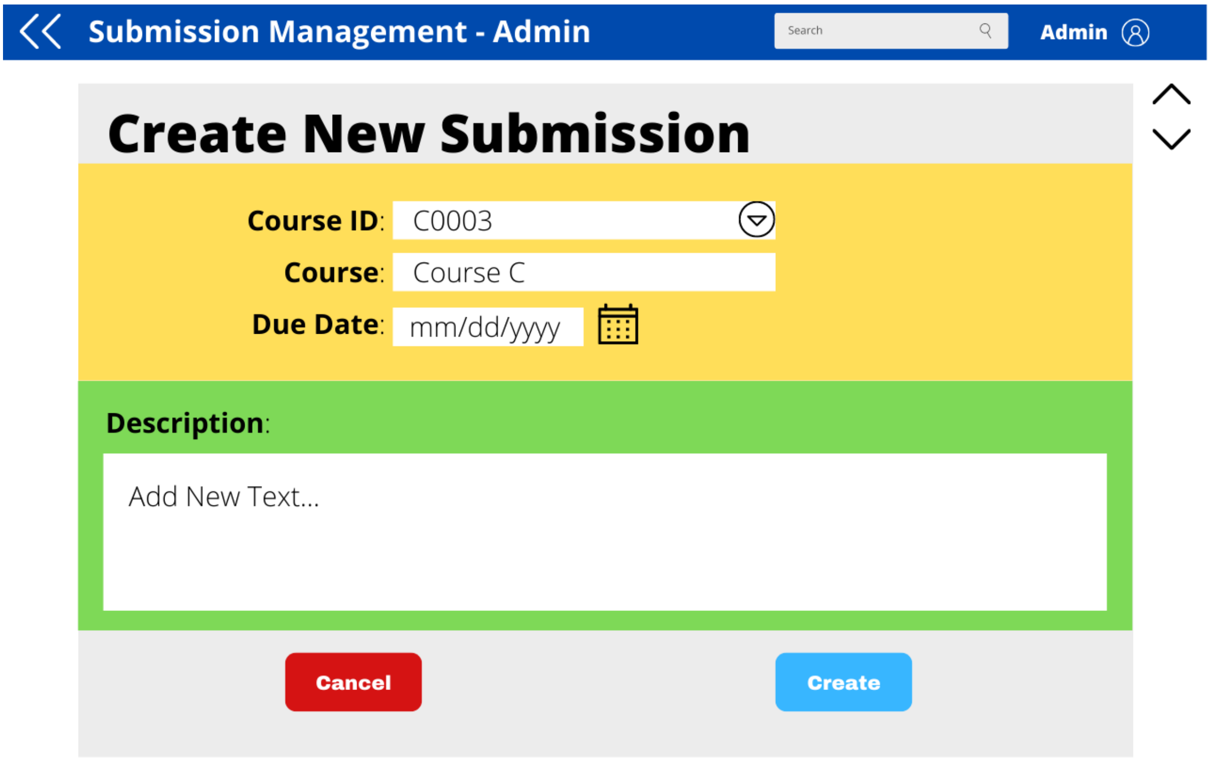
Description:
This is the first submission.

Second Submission

Course ID:C0002
Course: Course B
Due Date:mm/dd/yyyy

Description:
This is the second submission.

Figure 6.12: Overview of the list of submission page for Admin

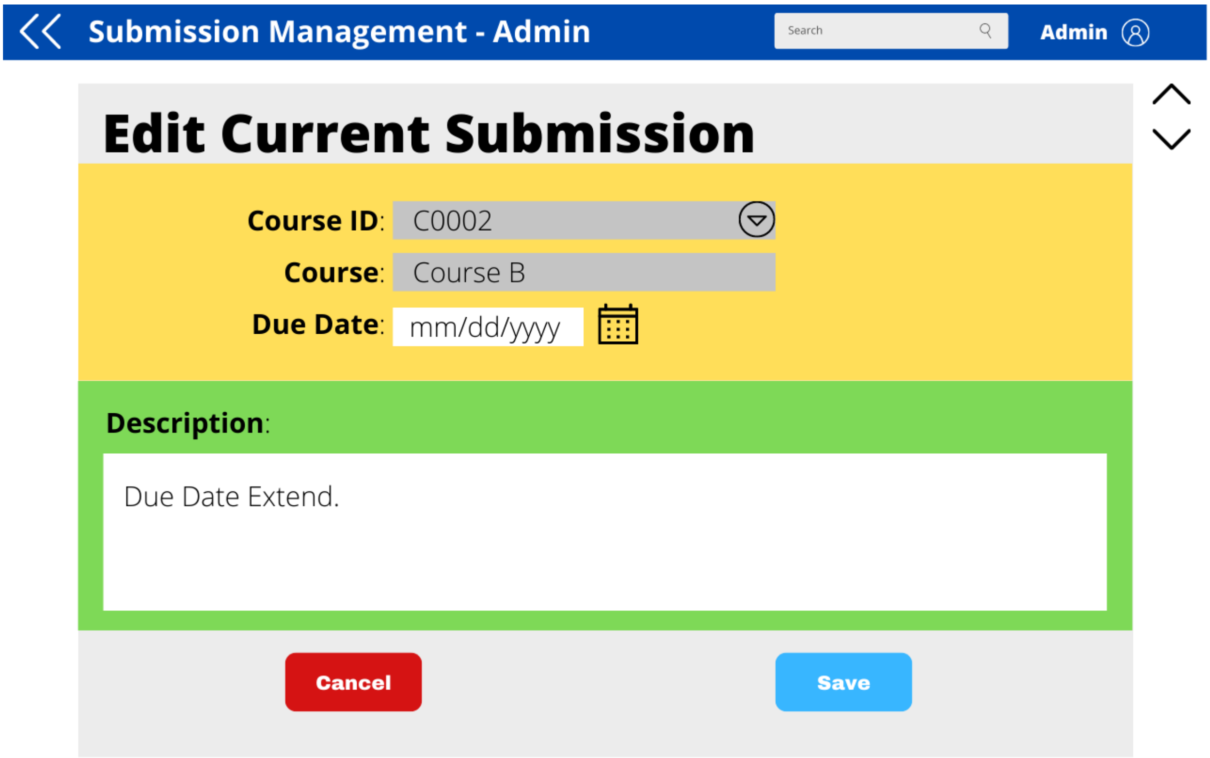

Submission Management - Admin

Create New Submission

Course ID: C0003
Course: Course C
Due Date: mm/dd/yyyy

Description:
Add New Text...

Figure 6.13: Overview of the create new submission page for Admin



Edit Current Submission

Course ID: C0002

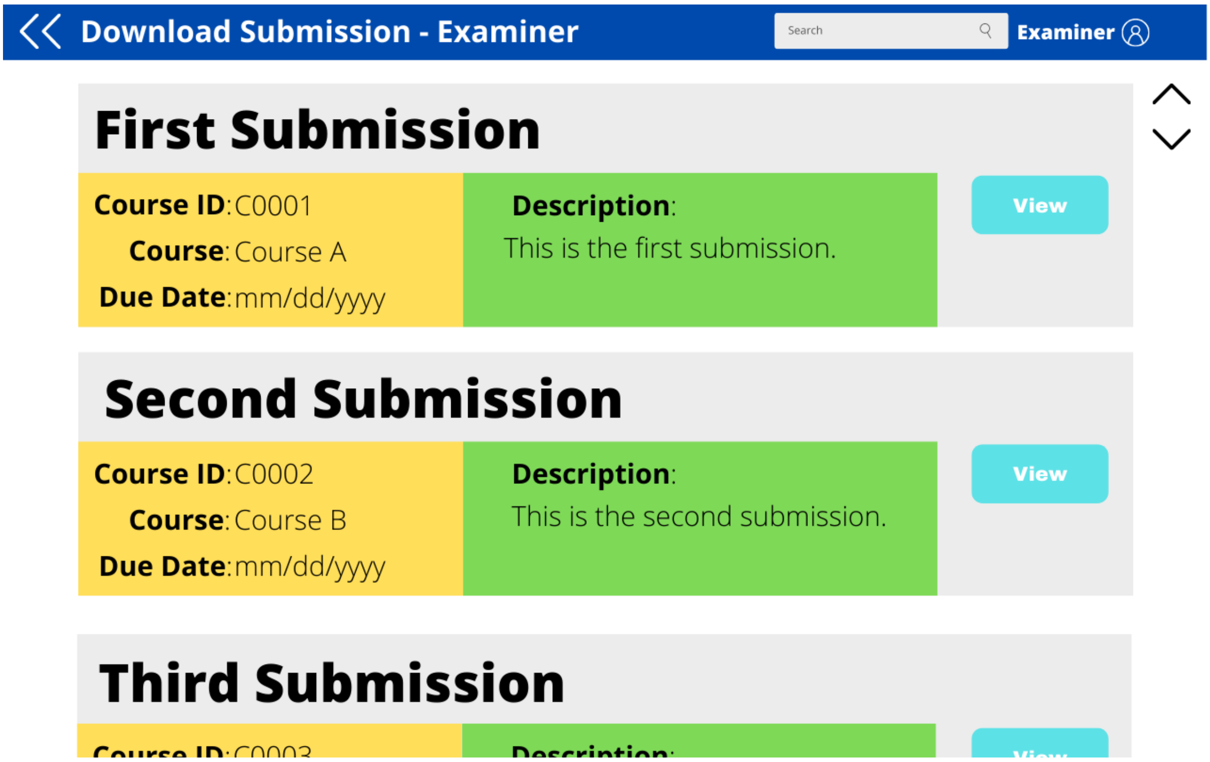
Course: Course B

Due Date: mm/dd/yyyy

Description:

Due Date Extend.

Figure 6.14: Overview of the edit current submission page for Admin



First Submission

Course ID:C0001
Course:Course A
Due Date:mm/dd/yyyy

Description:
This is the first submission.

Second Submission

Course ID:C0002
Course:Course B
Due Date:mm/dd/yyyy

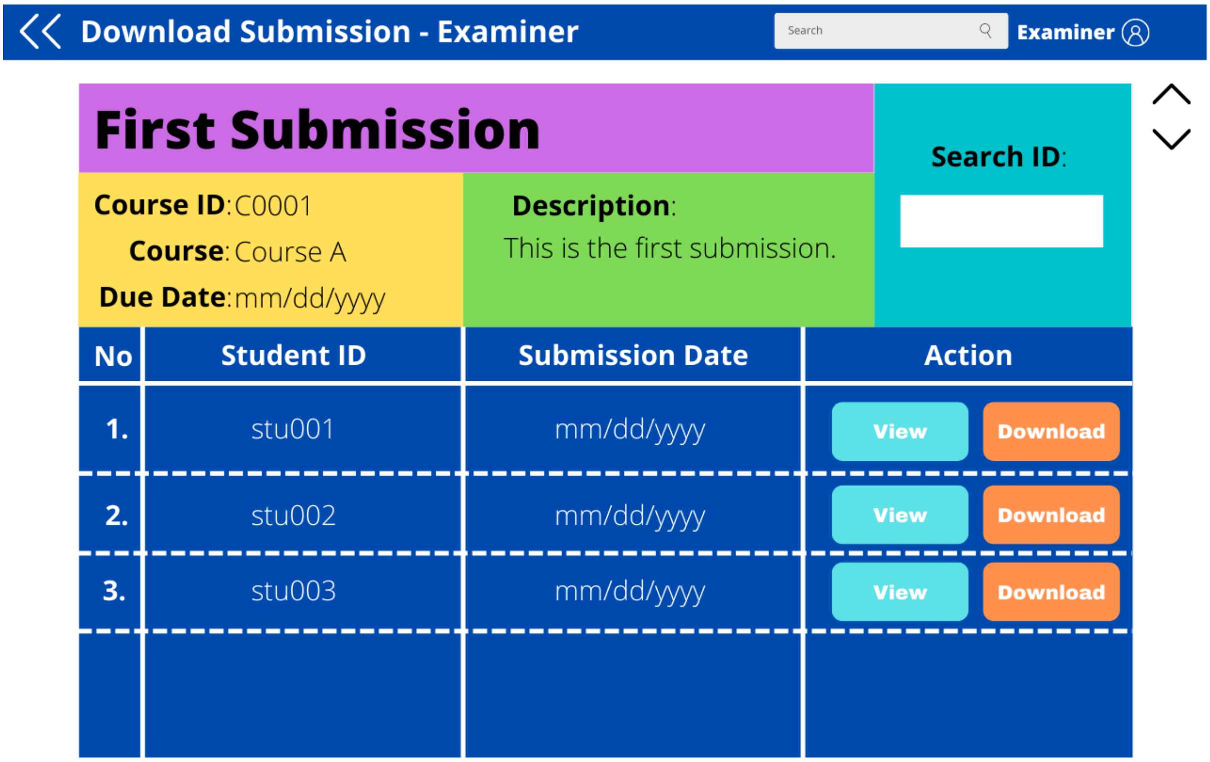
Description:
This is the second submission.

Third Submission

Course ID:C0003

Description:

Figure 6.15: Overview of the list of submission page for Examiner



Download Submission - Examiner

First Submission

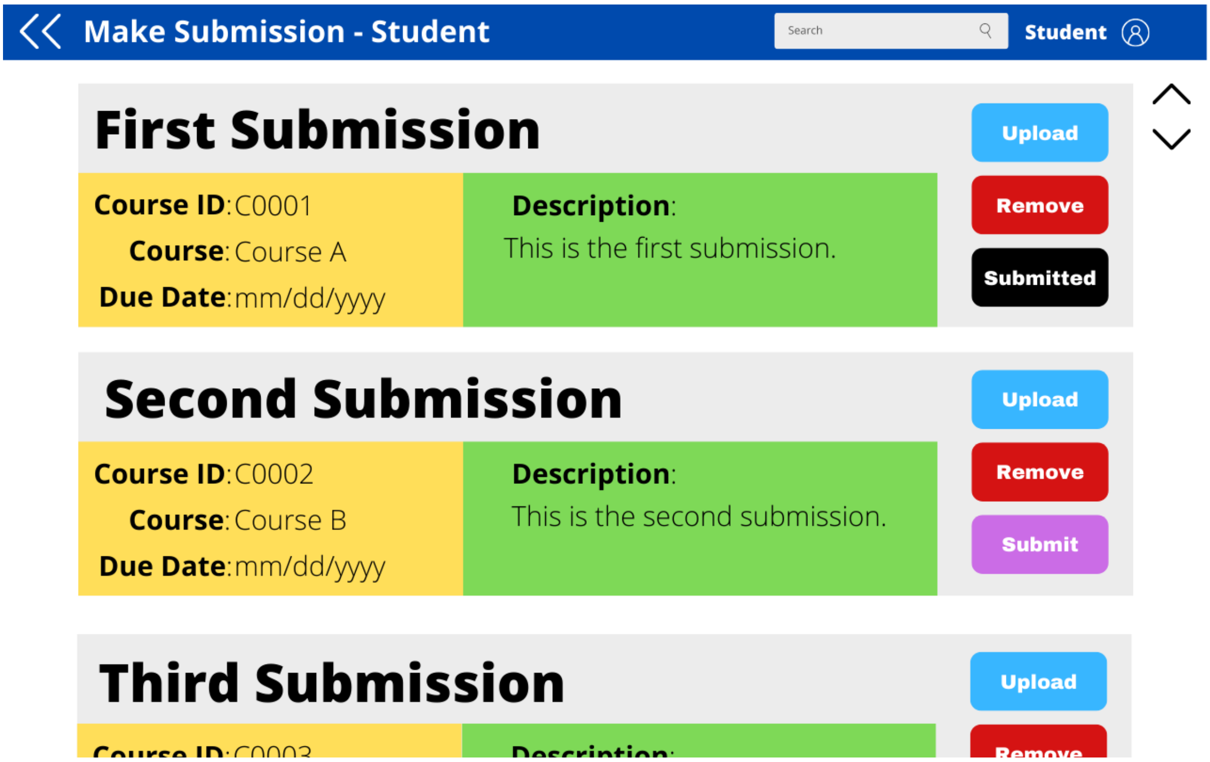
Course ID:C0001
Course: Course A
Due Date:mm/dd/yyyy

Description:
This is the first submission.

Search ID:

No	Student ID	Submission Date	Action
1.	stu001	mm/dd/yyyy	View Download
2.	stu002	mm/dd/yyyy	View Download
3.	stu003	mm/dd/yyyy	View Download

Figure 6.16: Overview of the download list of submission page for Examiner



Make Submission - Student

First Submission

Course ID:C0001
Course: Course A
Due Date:mm/dd/yyyy

Description:
This is the first submission.

Second Submission

Course ID:C0002
Course: Course B
Due Date:mm/dd/yyyy

Description:
This is the second submission.

Third Submission

Course ID:C0003

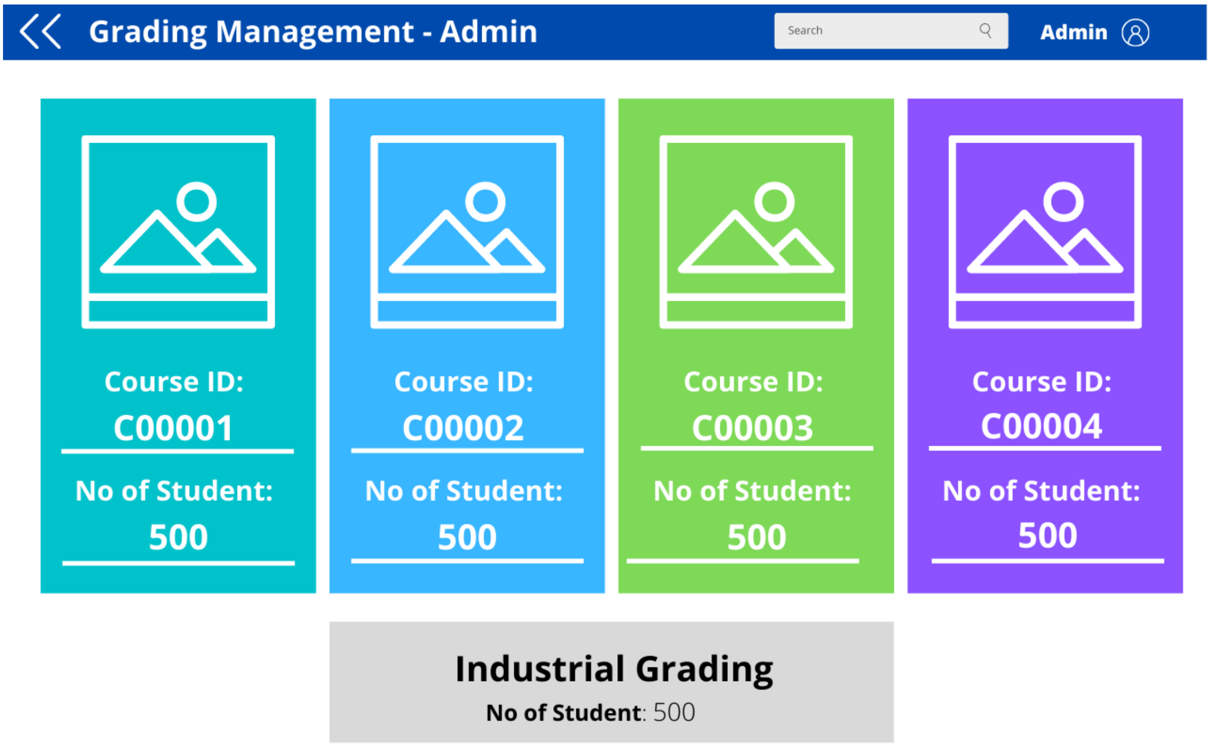
Description:

Figure 6.17: Overview of the list of make submission page for Student

Submission Interface (subsystem) includes Admin, Examiner and Student as the user. For the Admin, once they click the submission management option, it will bring the user to the submission management page. On the page, there is a list of the submissions with its course ID, course, due date and description. On the right side of the column, there are three buttons which are edit, view and delete. At the bottom of the page, Admin can add a new submission by clicking the add button. After clicking the button, a create new submission page will be opened. A new submission can be created by putting the course ID, course, due date and description. After all of the fields are filled, the Admin has to click the create button. If the Admin wants to edit the current submission, they have to click the edit button. An edit current submission page will be reloaded and the admin can edit any details of the submission then save it by pressing the save button.

If the user is Examiner, after they clicked the submission download option, they will be brought to the list of submissions page. In order for them to download the submission made by students, the user has to click the view button then a page of the list of students' submissions will be displayed. The list contains the student ID and its submission date. The examiner can either perform two actions which are view or download. In addition, on the right side of the page, the examiner can search for the student's submission by inputting the student's ID. Lastly, if the user is Student, they will be able to access the list of submissions that need to be made after clicking the make submission button. The student can upload, remove and submit a submission. Once they click the submit button, the button will change to submitted to show that a submission has been made.

Grading Interface



Course ID: C00001

No of Student: 500

Course ID: C00002

No of Student: 500

Course ID: C00003

No of Student: 500

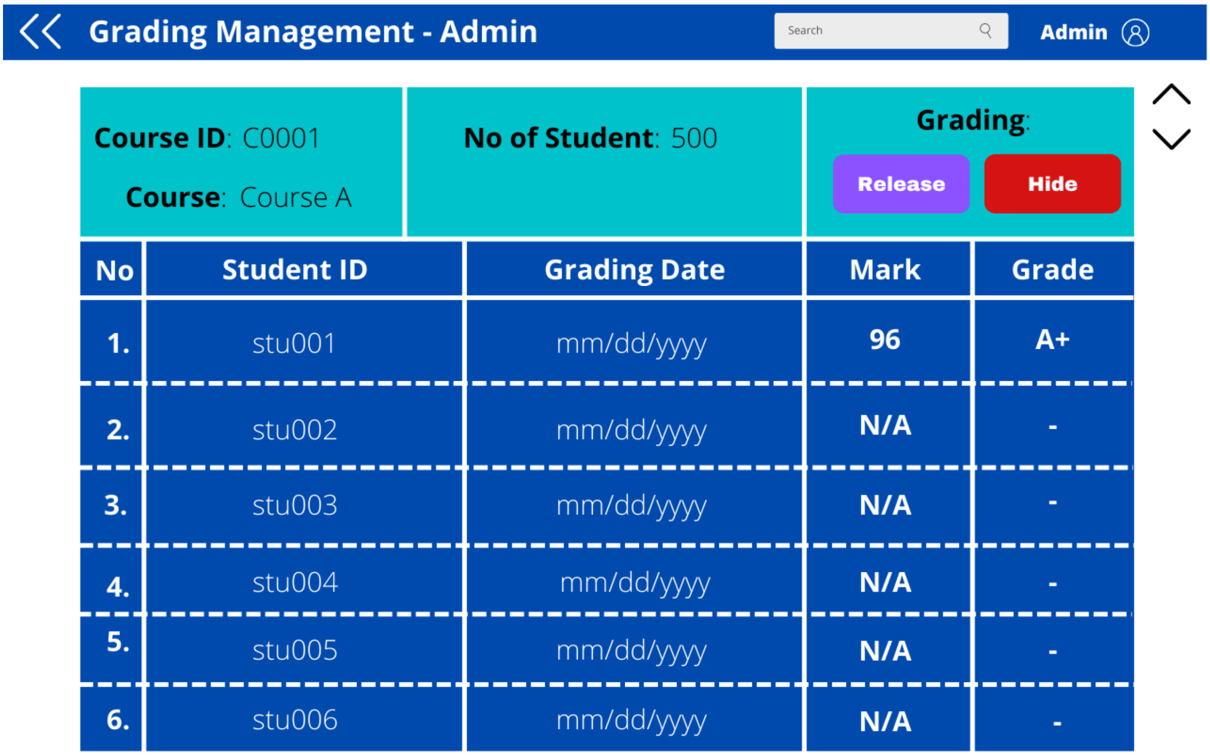
Course ID: C00004

No of Student: 500

Industrial Grading

No of Student: 500

Figure 6.18: Overview of Grading Management for Admin (Coordinator)



Course ID: C0001

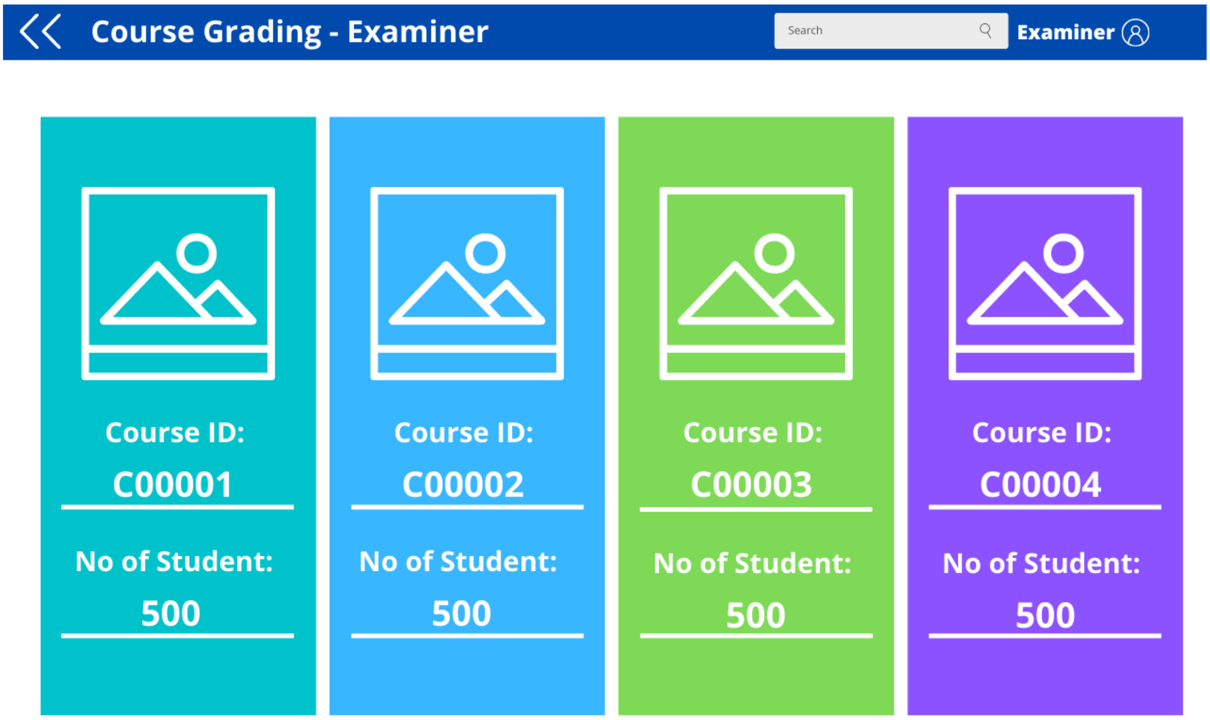
Course: Course A

No of Student: 500

Grading:

No	Student ID	Grading Date	Mark	Grade
1.	stu001	mm/dd/yyyy	96	A+
2.	stu002	mm/dd/yyyy	N/A	-
3.	stu003	mm/dd/yyyy	N/A	-
4.	stu004	mm/dd/yyyy	N/A	-
5.	stu005	mm/dd/yyyy	N/A	-
6.	stu006	mm/dd/yyyy	N/A	-

Figure 6.19: Overview of List of Grading per Course/Industrial for Admin (Coordinator)



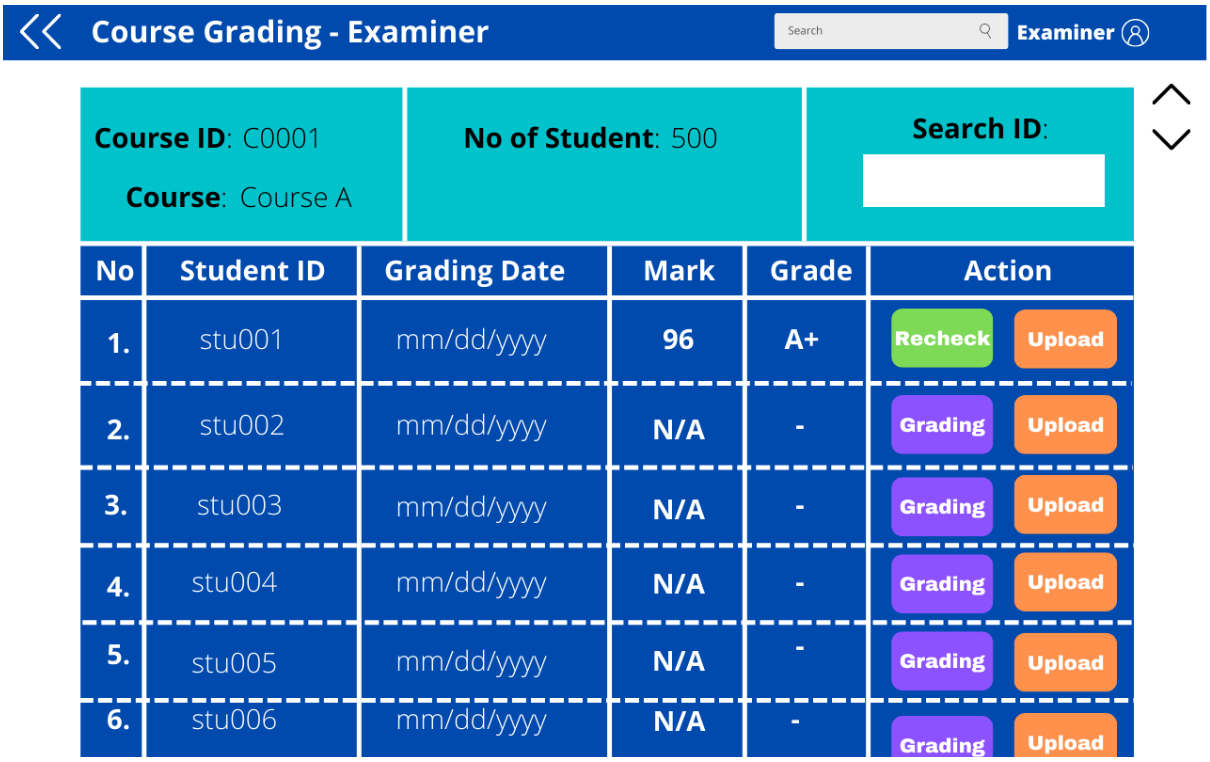
Course ID: C00001
No of Student: 500

Course ID: C00002
No of Student: 500

Course ID: C00003
No of Student: 500

Course ID: C00004
No of Student: 500

Figure 6.20: Overview of Course Grading for Examiner



Course ID: C0001

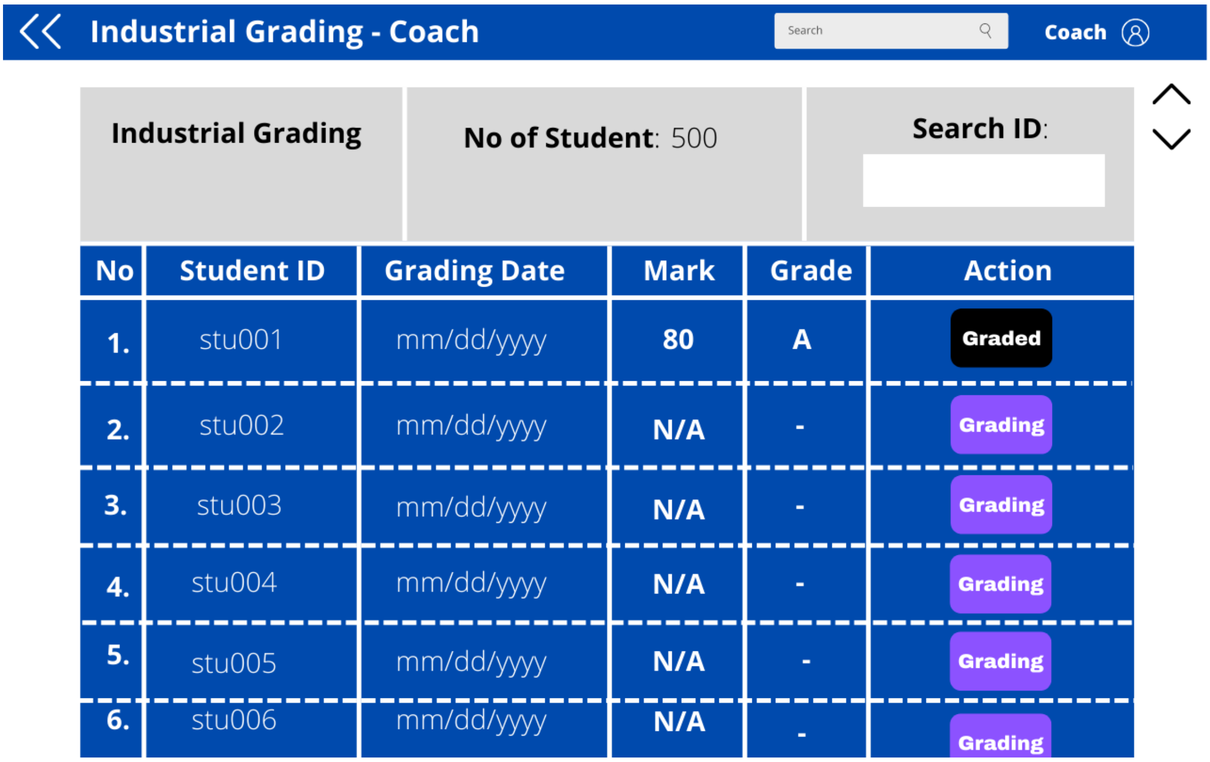
Course: Course A

No of Student: 500

Search ID:

No	Student ID	Grading Date	Mark	Grade	Action
1.	stu001	mm/dd/yyyy	96	A+	Recheck Upload
2.	stu002	mm/dd/yyyy	N/A	-	Grading Upload
3.	stu003	mm/dd/yyyy	N/A	-	Grading Upload
4.	stu004	mm/dd/yyyy	N/A	-	Grading Upload
5.	stu005	mm/dd/yyyy	N/A	-	Grading Upload
6.	stu006	mm/dd/yyyy	N/A	-	Grading Upload

Figure 6.21: Overview of List of Grading per Course for Examiner



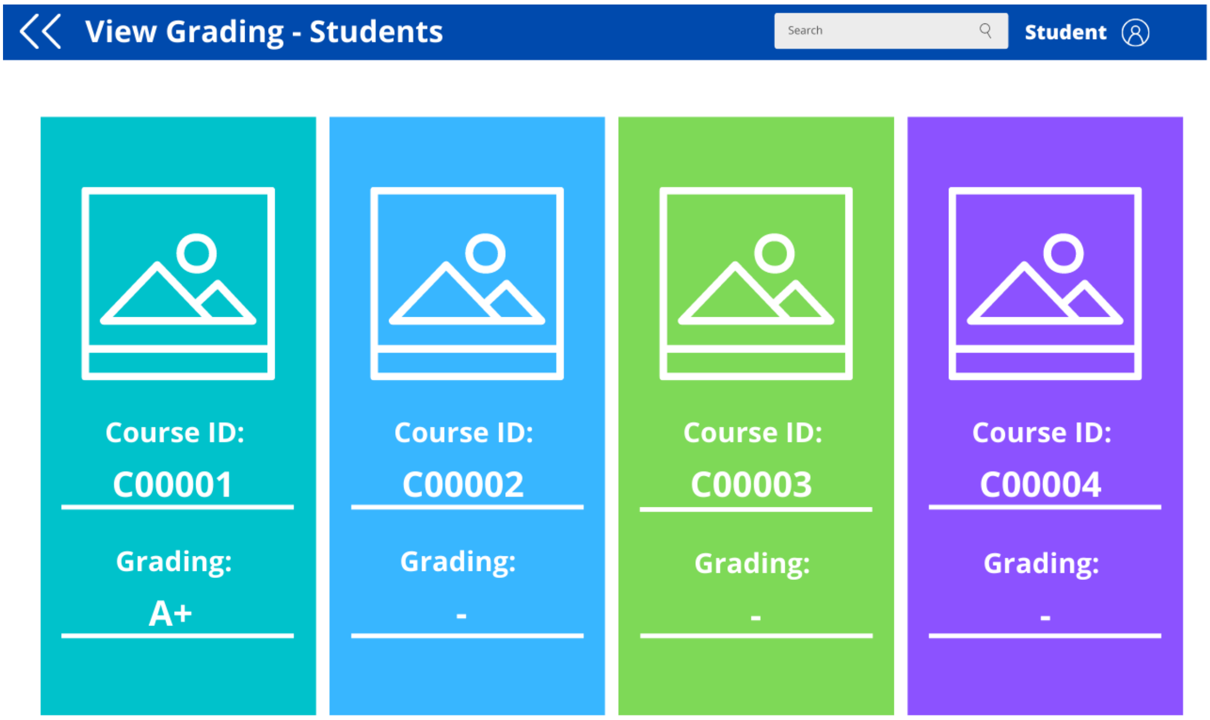
Industrial Grading

No of Student: 500

Search ID:

No	Student ID	Grading Date	Mark	Grade	Action
1.	stu001	mm/dd/yyyy	80	A	Graded
2.	stu002	mm/dd/yyyy	N/A	-	Grading
3.	stu003	mm/dd/yyyy	N/A	-	Grading
4.	stu004	mm/dd/yyyy	N/A	-	Grading
5.	stu005	mm/dd/yyyy	N/A	-	Grading
6.	stu006	mm/dd/yyyy	N/A	-	Grading

Figure 6.22: Overview of List of Industrial Grading for Coach



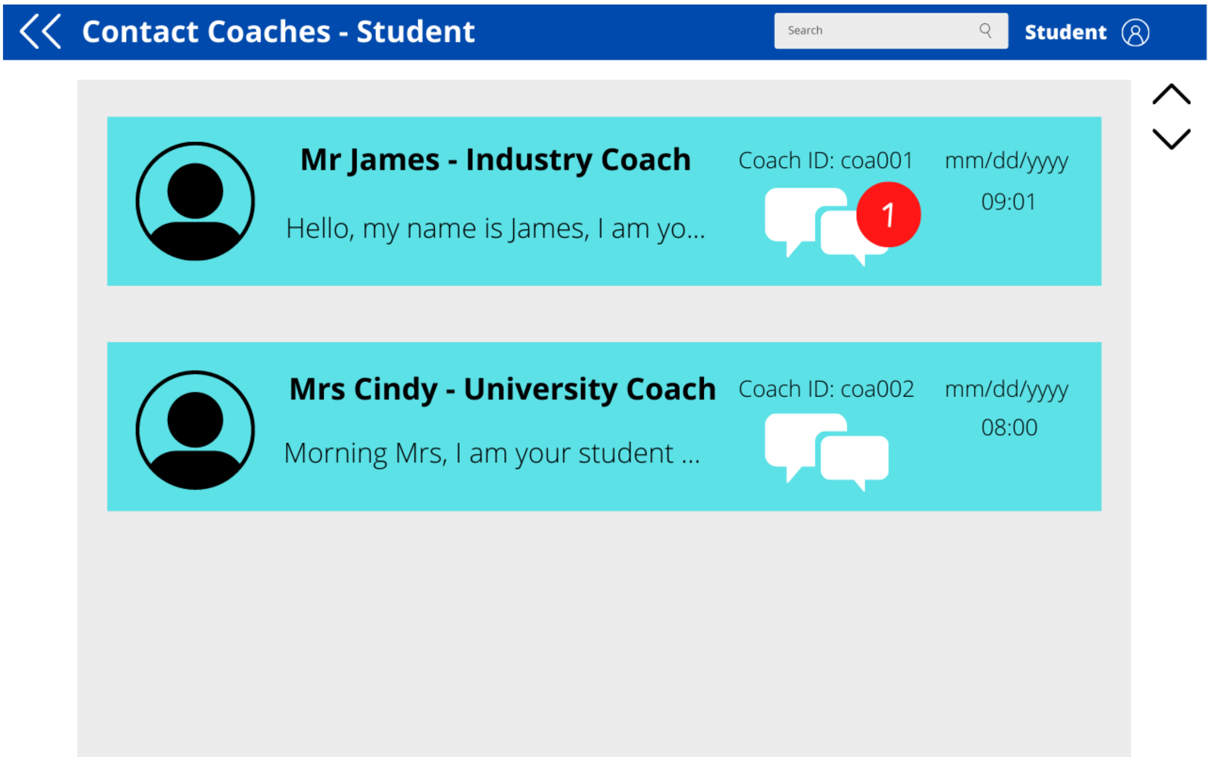
Course ID:	Grading:
C00001	A+
C00002	-
C00003	-
C00004	-

Figure 6.23: Overview of View Grading for Students

Grading interface (subsystem) involves all users, who are admin (coordinator), examiner, coach and student. When the admin clicks on the “Grading Management” button, the system will direct the admin to page Grading Management. There is a list of four courses and one industrial grading displayed with course ID and number of students on that page. By clicking on the course, a list of grading will be shown with the student ID (student username) and the grading date. Admin can select to hide or release the grading. Once grading is released, students can view their course grading by entering the View Grading page.

Examiners can do grading for students by entering the Course Grading page. Same as admin, there is a list of four courses displayed with course ID and number of students but there is no industrial grading. By clicking on one course, a list of student IDs (student username) are displayed with the grading date, mark and grade. Examiners can take action to make a grade by keying a mark and uploading the graded file. After a grade is done, a button “Recheck” will replace “Grading” for the rechecking grade process. Coaches do not be displayed with a list of courses. Once they click on industrial grading, they will be directed to a list of students with grading date, mark and grade. Coaches just need to do a grading based on the industrial skill of a student and they do not need to recheck the grading. Both examiners and coaches can find a specific student to do grading by entering the student ID as a search key.

Communication Interface



Mr James - Industry Coach

Hello, my name is James, I am yo...

Coach ID: coa001

mm/dd/yyyy

09:01

Mrs Cindy - University Coach

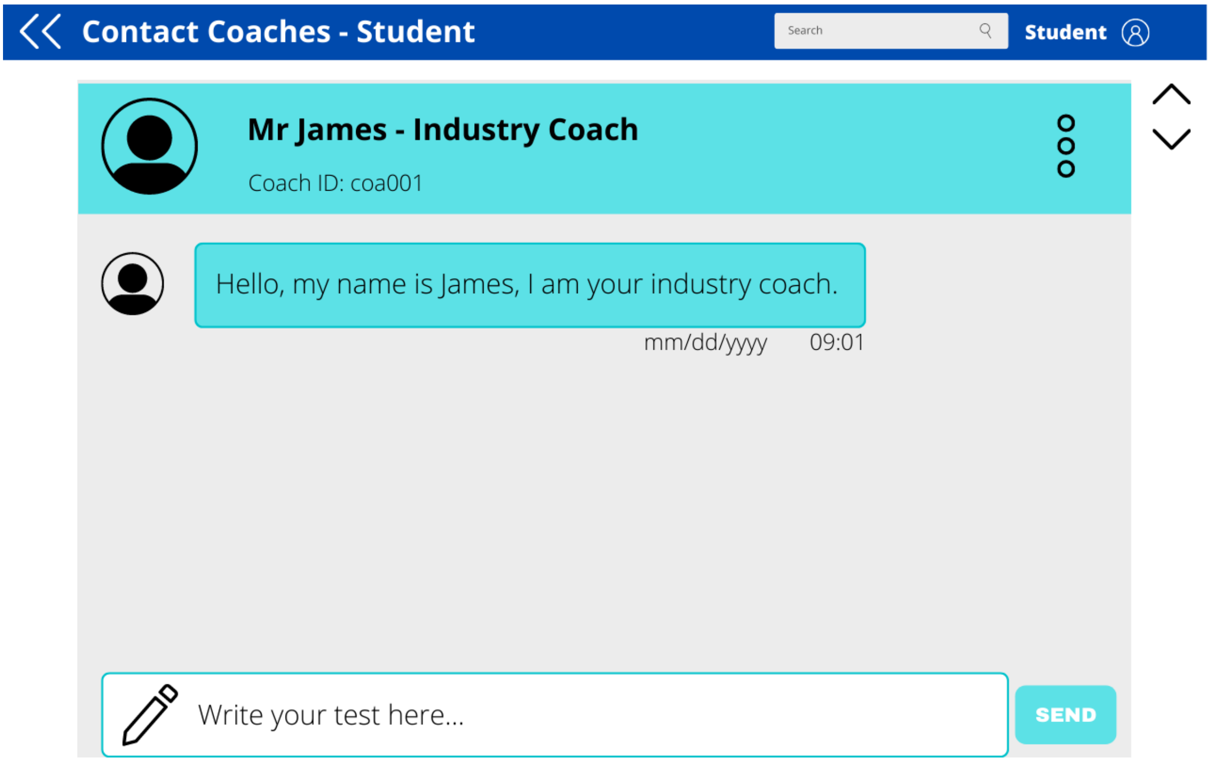
Morning Mrs, I am your student ...

Coach ID: coa002

mm/dd/yyyy

08:00

Figure 6.24: Overview of Contact Coaches for Students



Mr James - Industry Coach

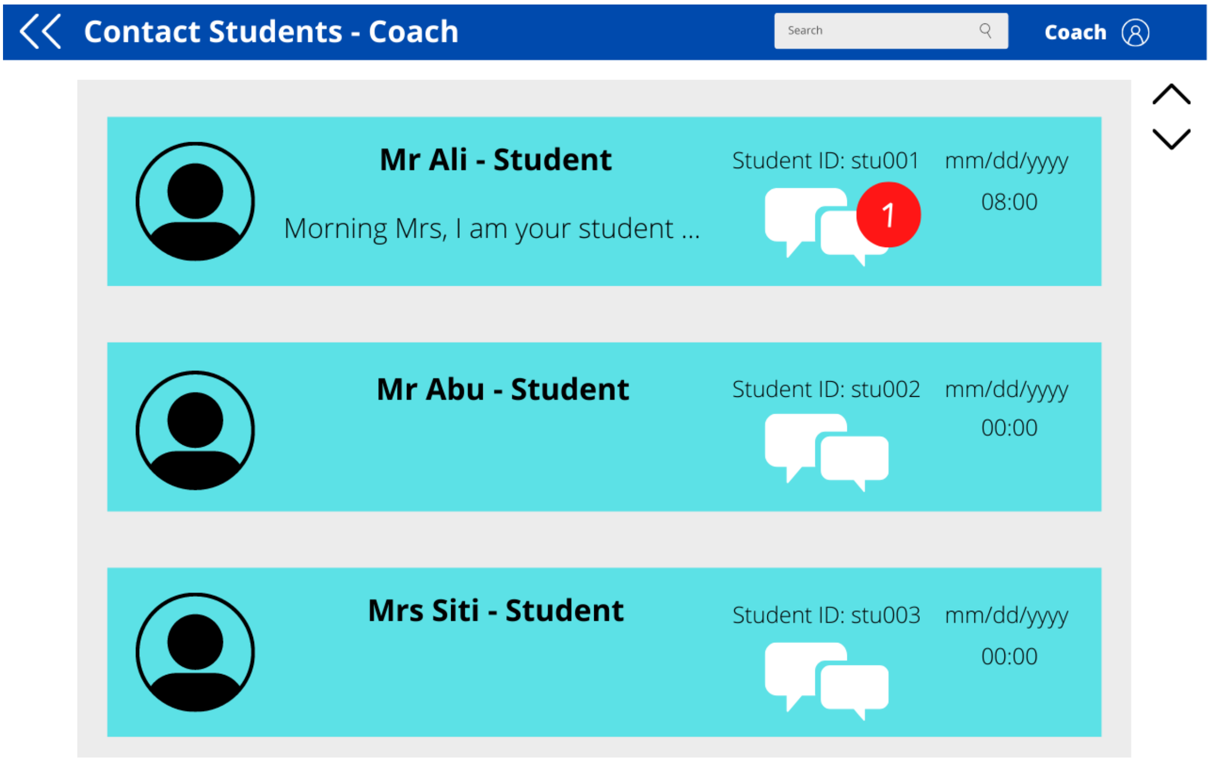
Coach ID: coa001

Hello, my name is James, I am your industry coach.

mm/dd/yyyy 09:01

Write your test here...

Figure 6.25: Overview of Contact Box with Coach for Students



Mr Ali - Student

Morning Mrs, I am your student ...

Student ID: stu001 mm/dd/yyyy 08:00

1

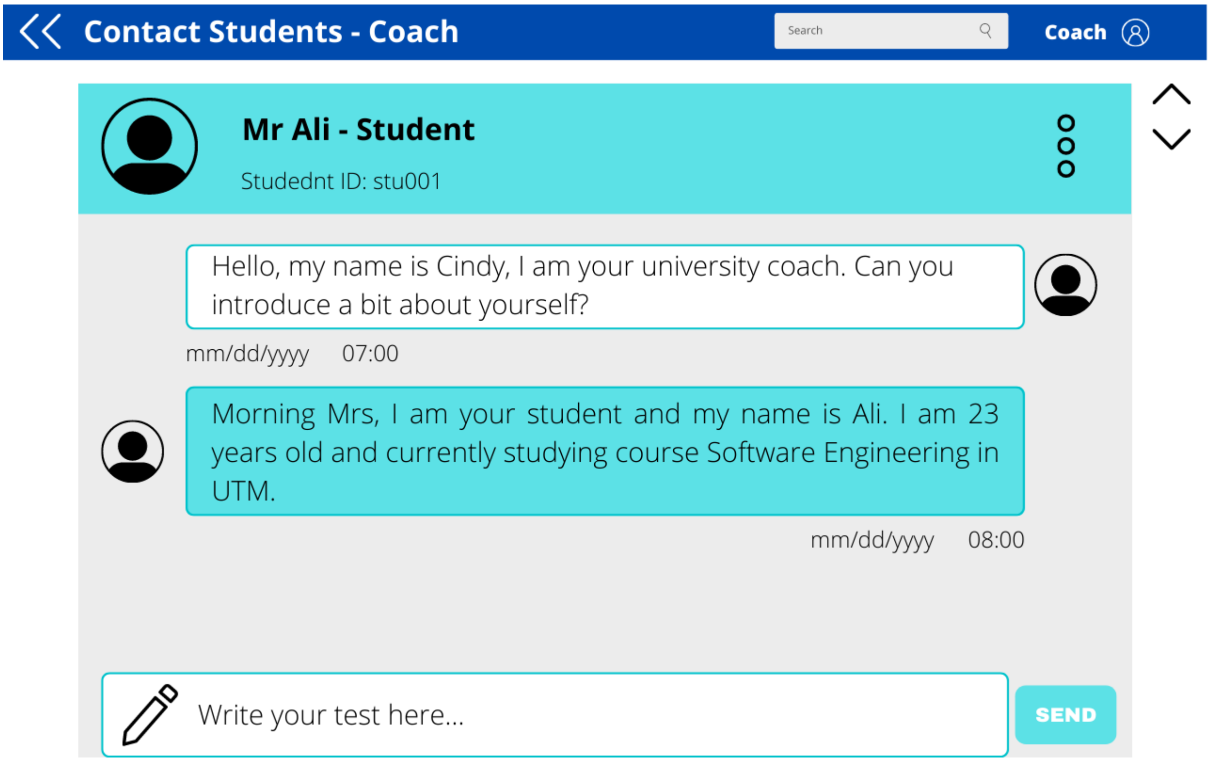
Mr Abu - Student

Student ID: stu002 mm/dd/yyyy 00:00

Mrs Siti - Student

Student ID: stu003 mm/dd/yyyy 00:00

Figure 6.26: Overview of Contact Students for Coaches



Mr Ali - Student

Studednt ID: stu001

Hello, my name is Cindy, I am your university coach. Can you introduce a bit about yourself?

mm/dd/yyyy 07:00

Morning Mrs, I am your student and my name is Ali. I am 23 years old and currently studying course Software Engineering in UTM.

mm/dd/yyyy 08:00

Write your test here...

Figure 6.27: Overview of Contact Students for Coaches

Communication interface (subsystem) involves only two types of users, coaches and students. Both users access the similar functions to communicate between each other, the differences are just the data category and flow. Students can only contact coaches while coaches can only contact students through the communication channel. The system will display the profile picture and ID (username) of user contact, and message date, time. The system will show a red dot with a number indicating the unread message(s). Users can type their message in a text box and send a message by clicking on the “SEND” button. When a message is displayed, the message in white box indicates the message sent by the user while the message in the mint-colored box indicates the message received from the contact person(sender).