SECJ2203: Software Engineering

System Documentation (SD)

Final Year Project (Maniacs System)

Version 1

Date: 5 JUN 2022

School of Computing, Faculty of Engineering

Prepared by: <Software Maniacs>

Revision Page

a. **Overview**

The current version of SD contains the Introduction of Maniac System that states about the purpose of the documentation, the scope of the system, definitions, acronyms and abbreviations, references needed to complete this documentation as well as the overview of the system. In addition, this version of SD also includes the details of specific requirements of the system such as External Interface Requirements (User Interface, Hardware Interface, Software Interface, Communication Interface), System Features (which contains the use case diagram, use case description, activity diagram and sequence diagram), Performance and other requirements, Design Constraints as well as Software System Attributes.

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	1. Use Case Diagram 2. Use Case <Contact Coaches> 3. Performance and Other Requirements 4. Design Constraints 5. Software System Attributes 6. Check and suggest improvements to all parts, especially the use cases and graphics.	Completed
LUE GUO MING	Group Member	1. Introduction 2. User Interfaces 3. Software Interfaces 4. Communication Interfaces 5. Use Case Diagram	Completed

		6. Use Case <Make Submission> 7. Use Case <View Submission> 8. Use Case <Download Submission>	
NUR HAZNIRAH BINTI HAZMAN	Group Member	1. Use Case Diagram 2. Use Case <View Notification> 3. Use Case <Delete Notification> 4. Use Case <Create Submission Platform> 5. Use Case <Change Time Date of Submission> 6. Domain Model 7. Overall Activity Diagram	Completed
ERICA DESIRAE MAURITIUS	Group Member	1. Introduction 2. User Interfaces 3. Hardware Interfaces 4. Communication Interfaces 5. Use Case Diagram 6. Use Case <Grading> 7. Use Case <Recheck Grading> 8. Use Case <Release Grading> 9. Use Case <View Grading>	Completed

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

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

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, and examiners. 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

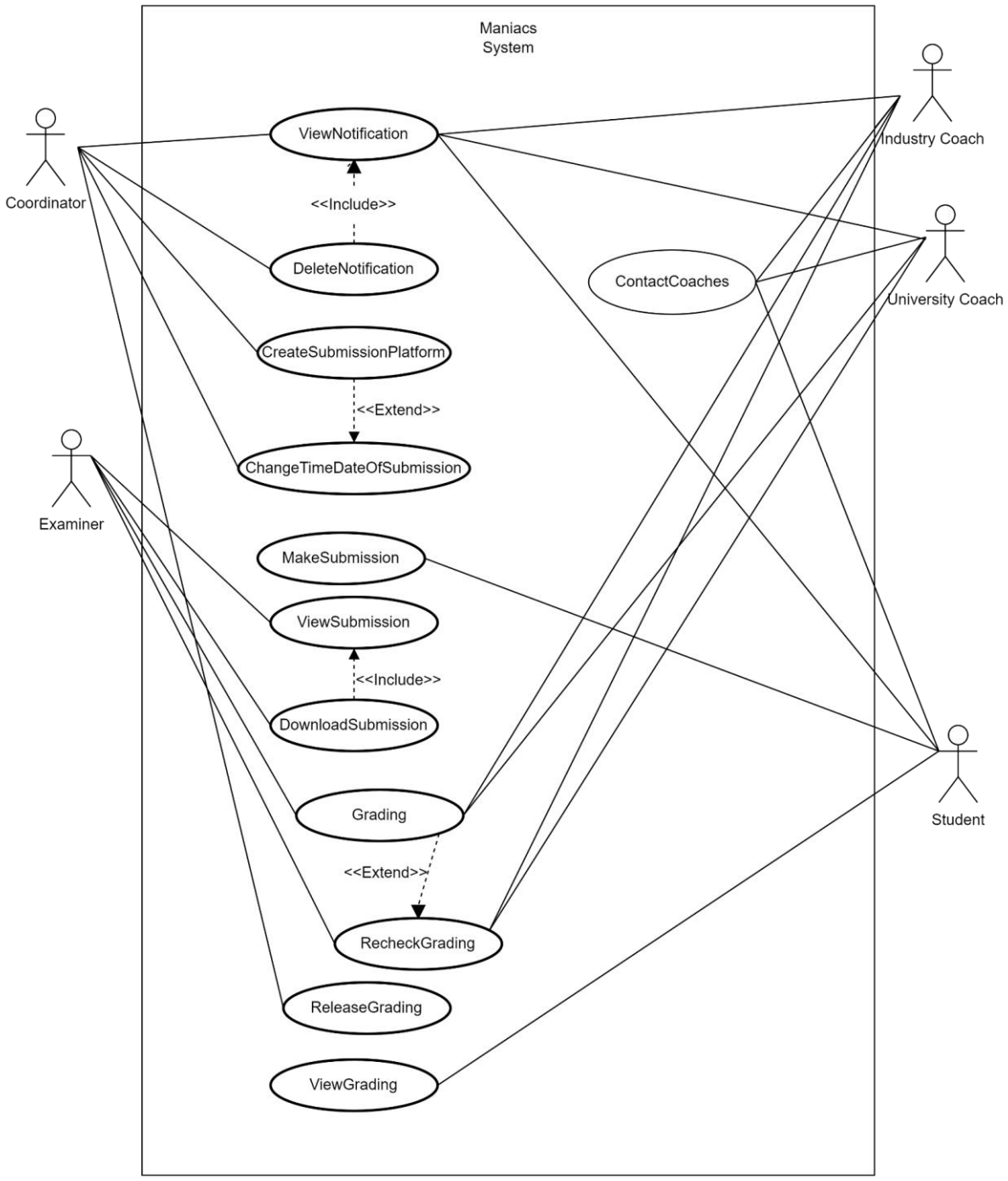

Figure 2.1: Use Case Diagram for <Maniac System>

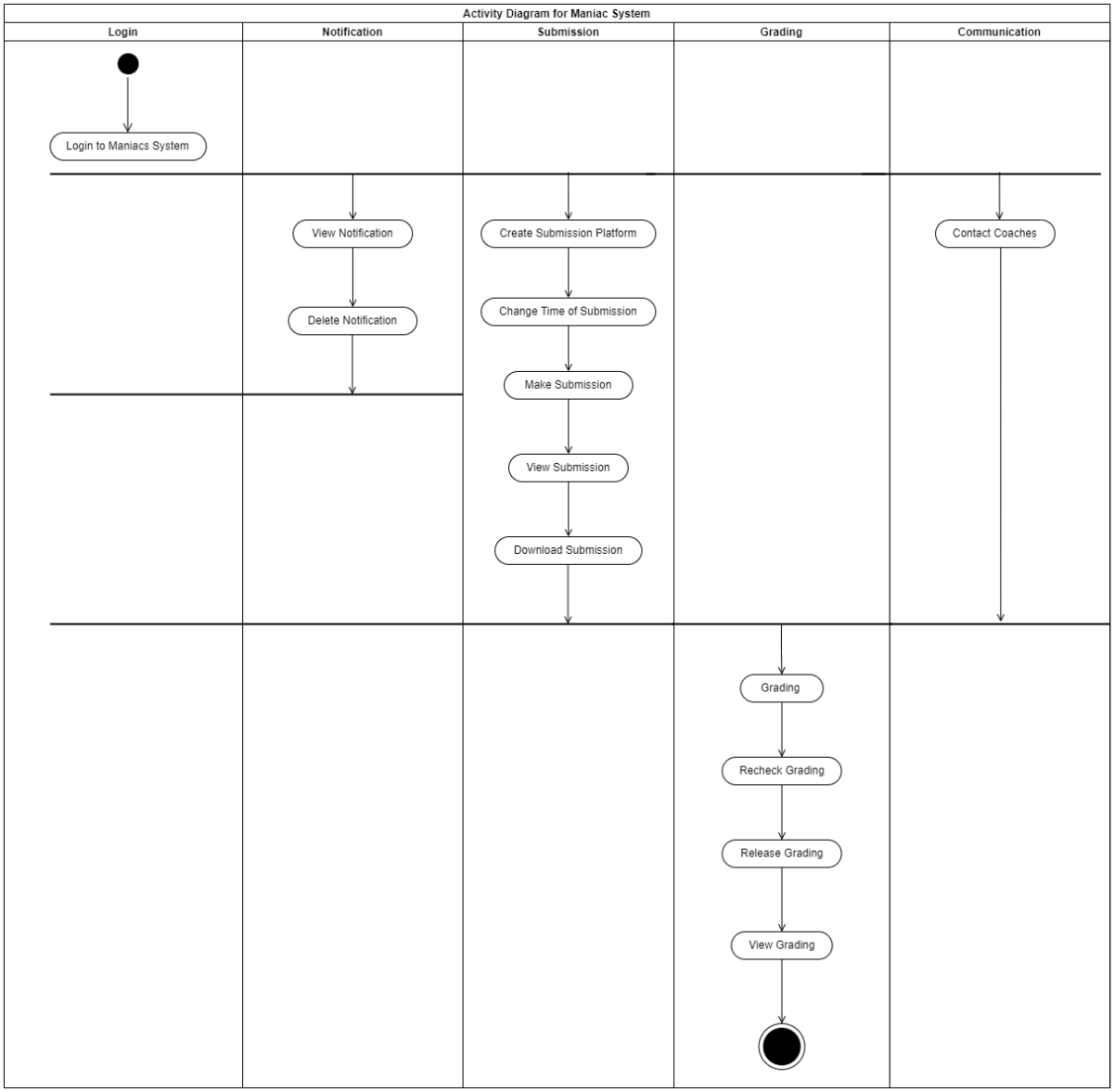

Figure 2.2: Activity Diagram for <Maniac System>

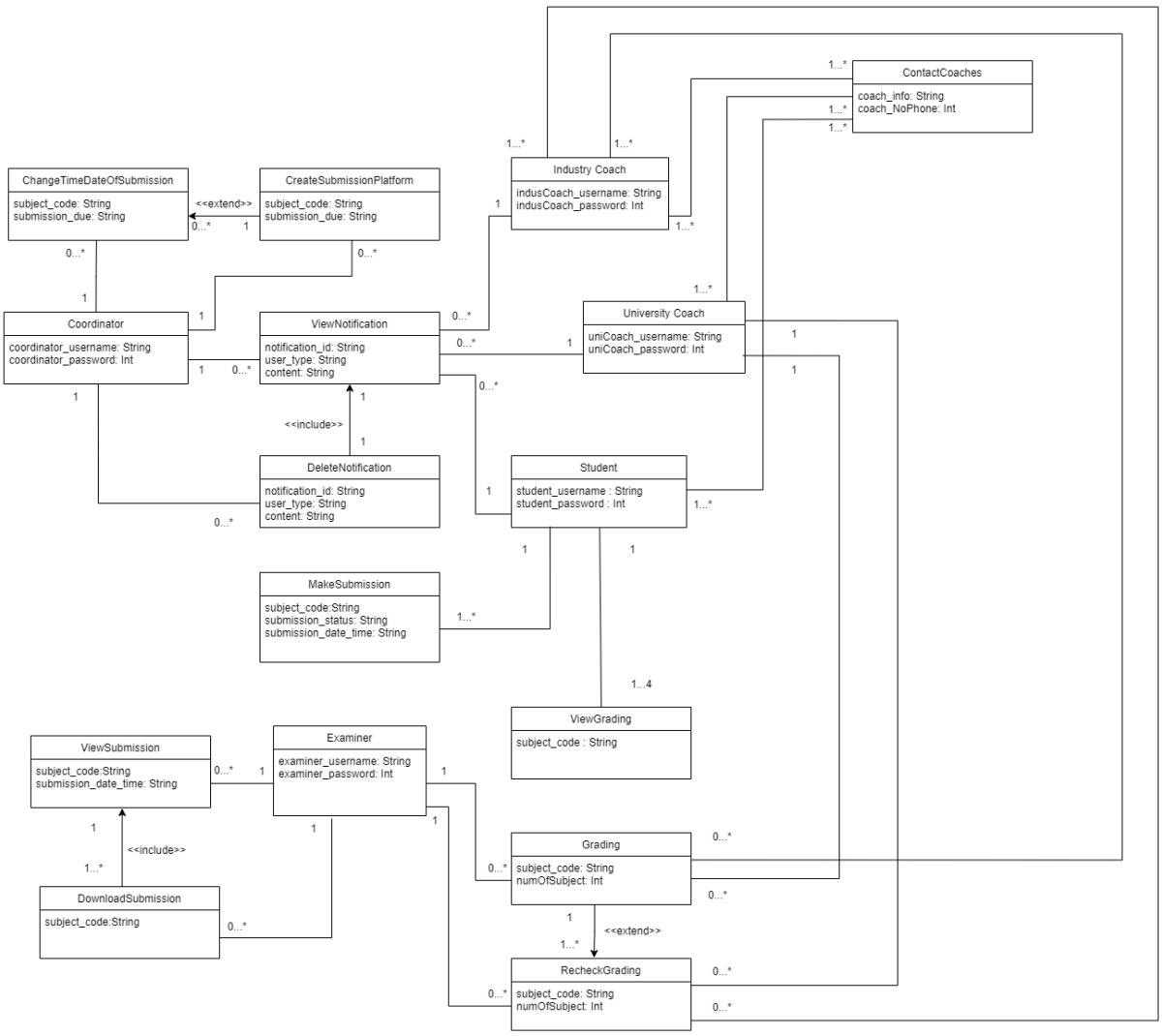

Figure 2.3: Domain Model for <Maniac System>

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>

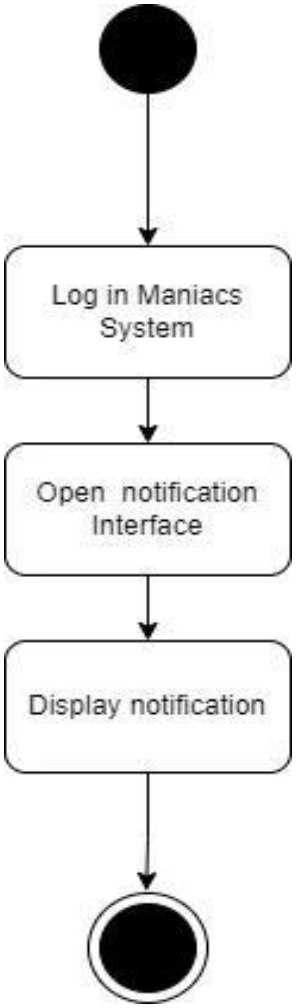

Figure 2.4: Activity Diagram for <View Notification>

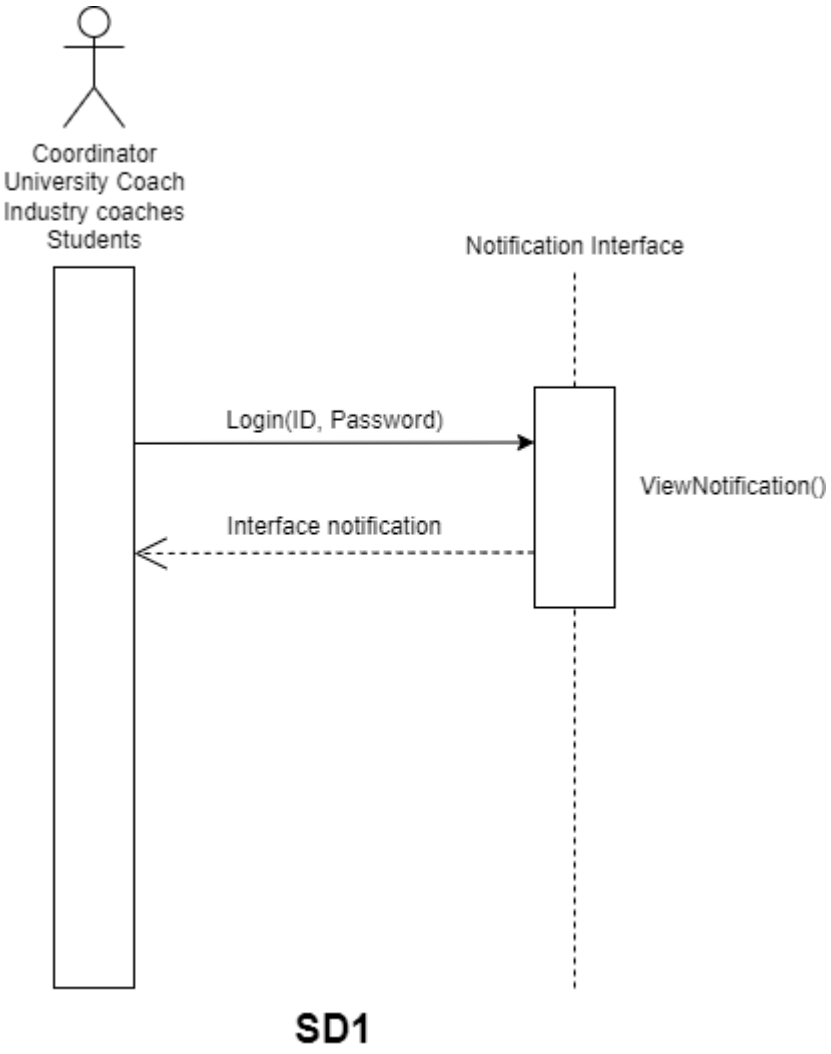

Figure 2.5: Sequence Diagram for <View Notification>

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>

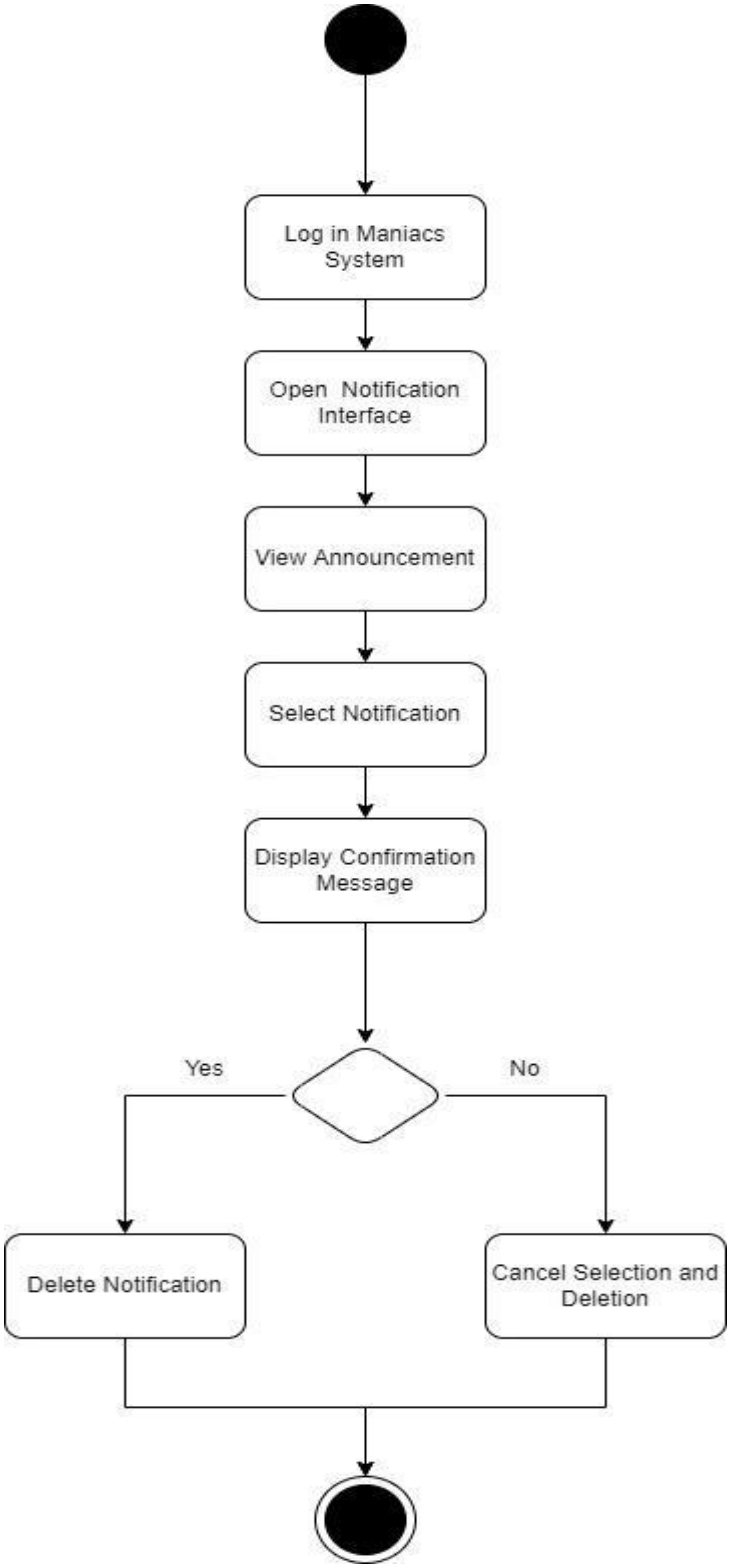

Figure 2.6: Activity Diagram for <Delete Notification>

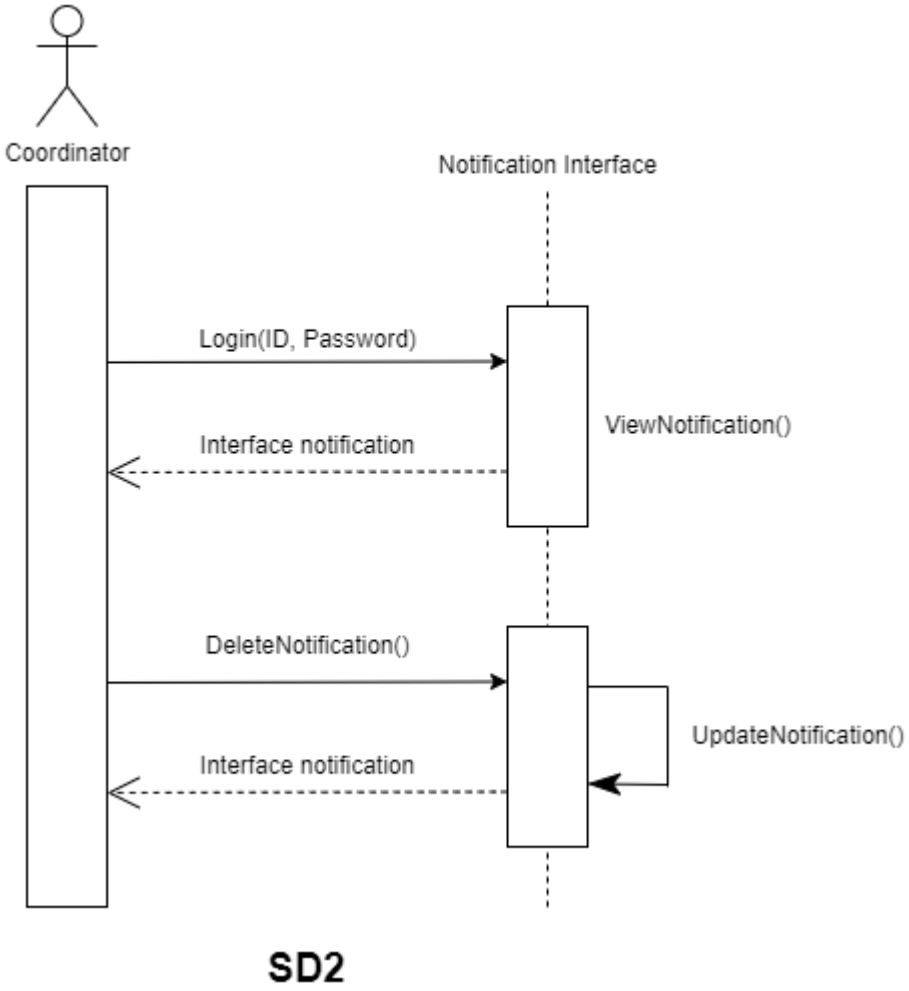

Figure 2.7: Sequence Diagram for <Delete Notification>

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>

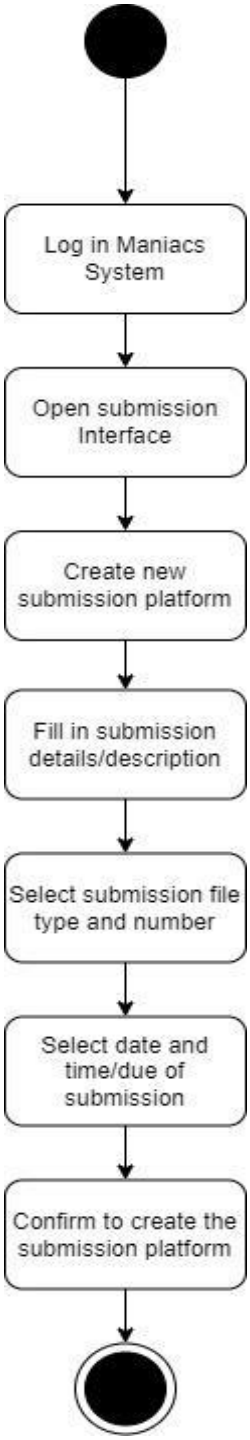

Figure 2.8: Activity Diagram for <Create Submission Platform>

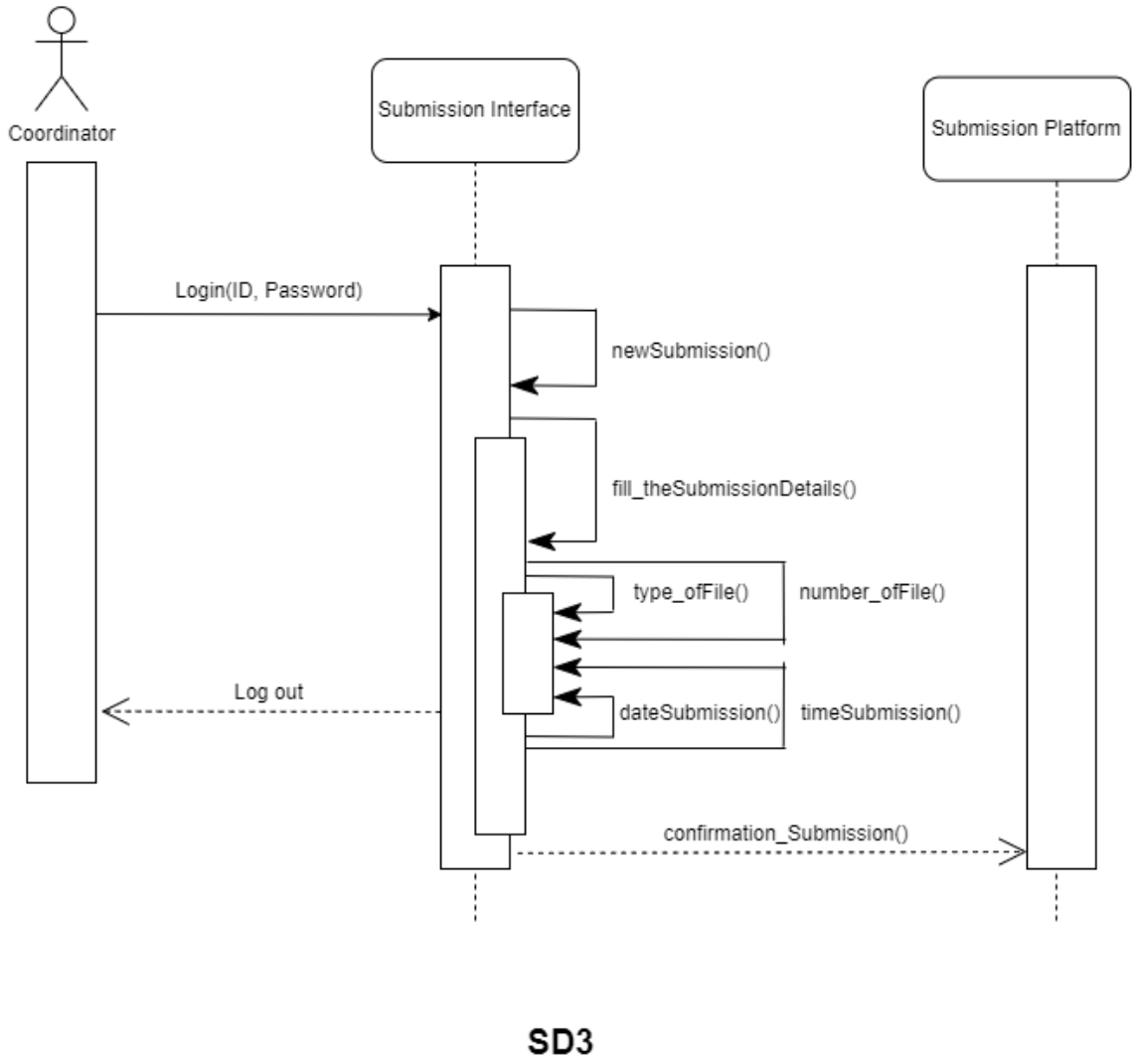

Figure 2.9: Sequence Diagram for <Create Submission Platform>

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: .The Coordinator selects to edit the submission platform. .The Coordinator set a new date and time for submission. .The system display the confirmation message .If the Coordinator select “Save and make change” .System sets a new date and time for the submission platform. .If the Coordinator select “Cancel” .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>

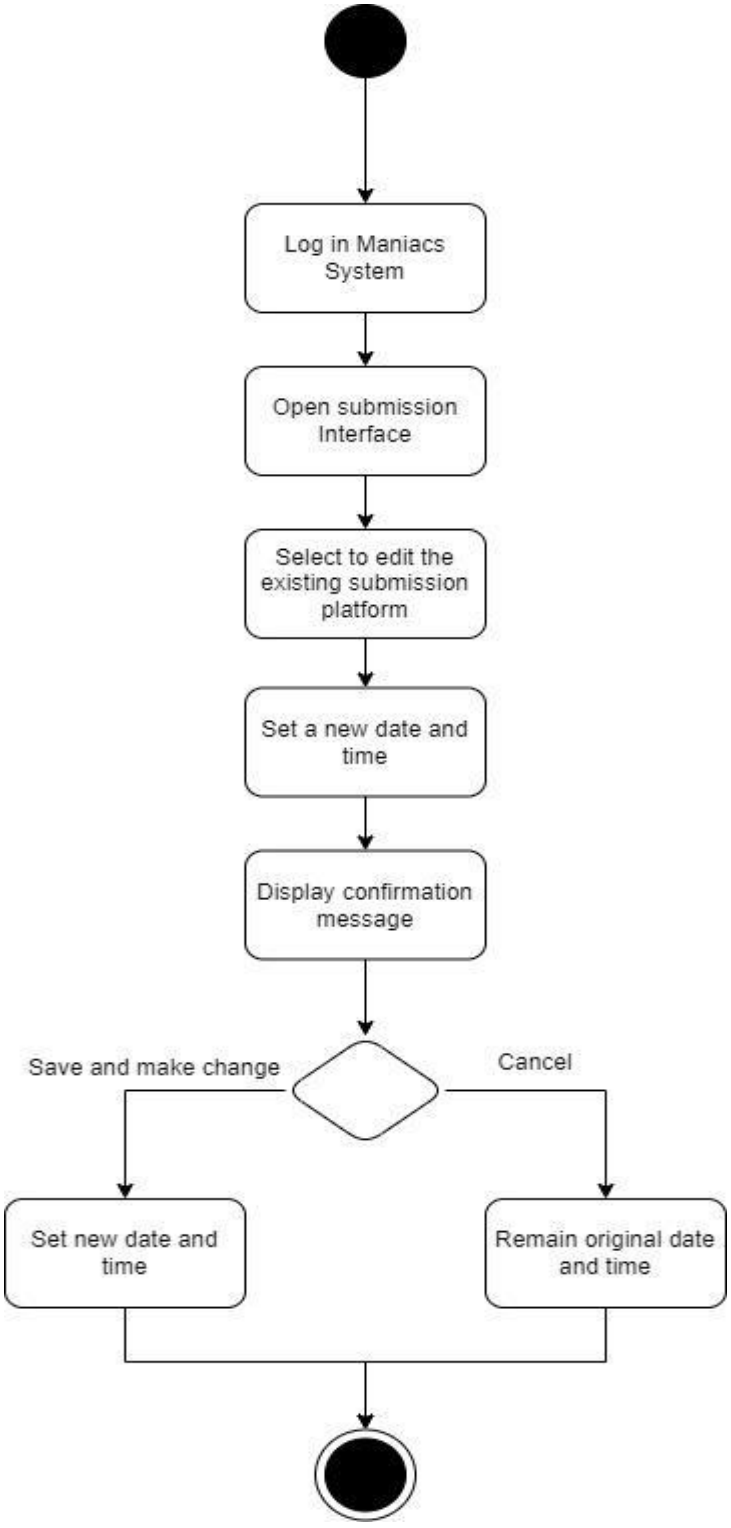

Figure 2.10: Activity Diagram for <Change Time Date of Submission>

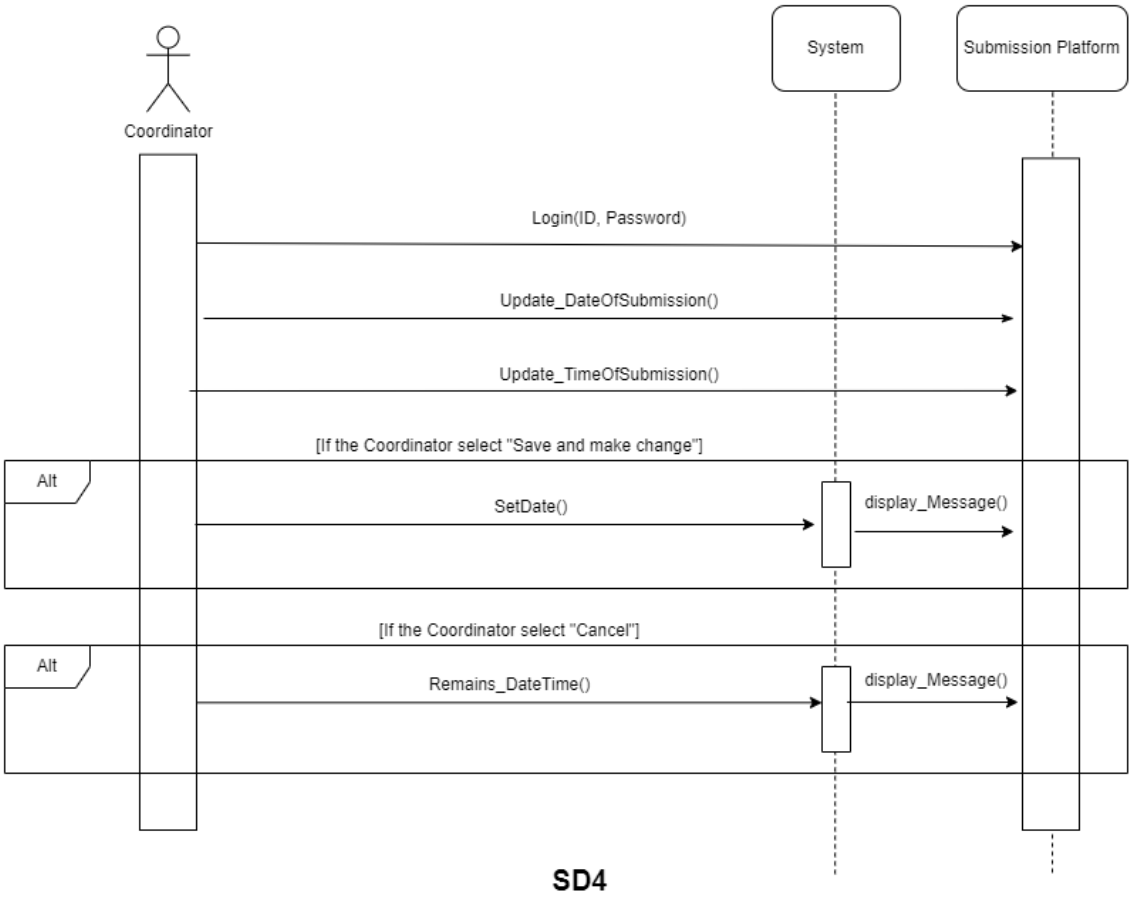

Figure 2.11: Sequence Diagram for <Change Time Date of Submission>

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: .The student logins to the system. .The student opens the submission interface. .The student clicks on the submission link. .The student selects and uploads his work/file from a valid drive. .The system displays confirmation messages. .If the student select “Save and make change” .System save and submit the uploaded file. .System change submission status to “Submitted”. .If the student select “Cancel” .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>

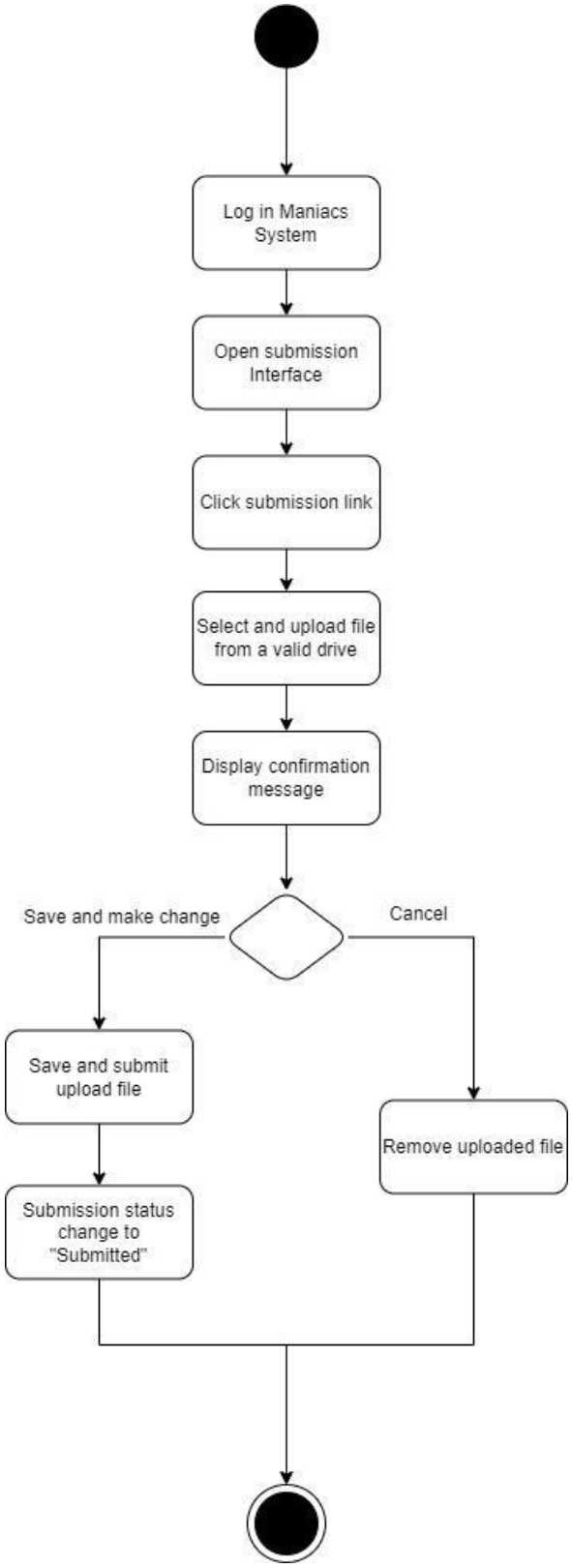

Figure 2.12: Activity Diagram for <Make Submission>

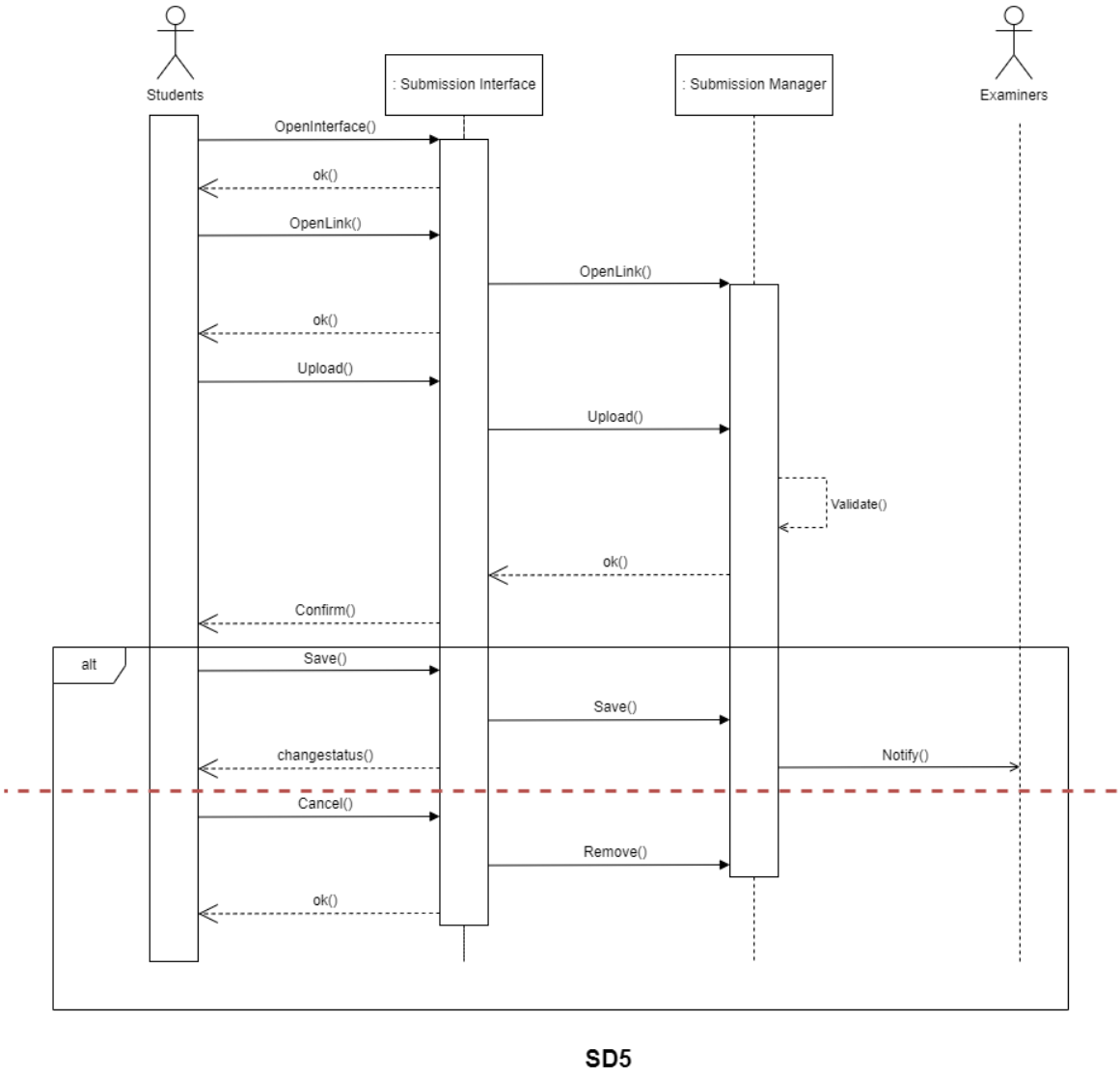

Figure 2.13: Sequence Diagram for <Make Submission>

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: The examiner logins to the system. The examiner opens the submission interface. A list of submission files displayed with its author(student), submission date and time. The examiner clicks and selects one file to view its content.

Table 2.6: Use Case Description for <View Submission>

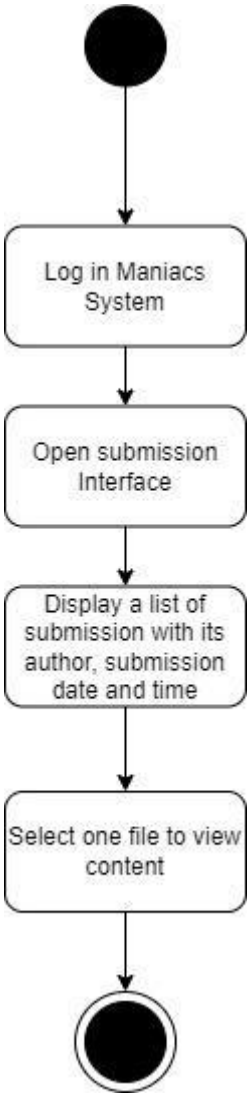

Figure 2.14: Activity Diagram for <View Submission>

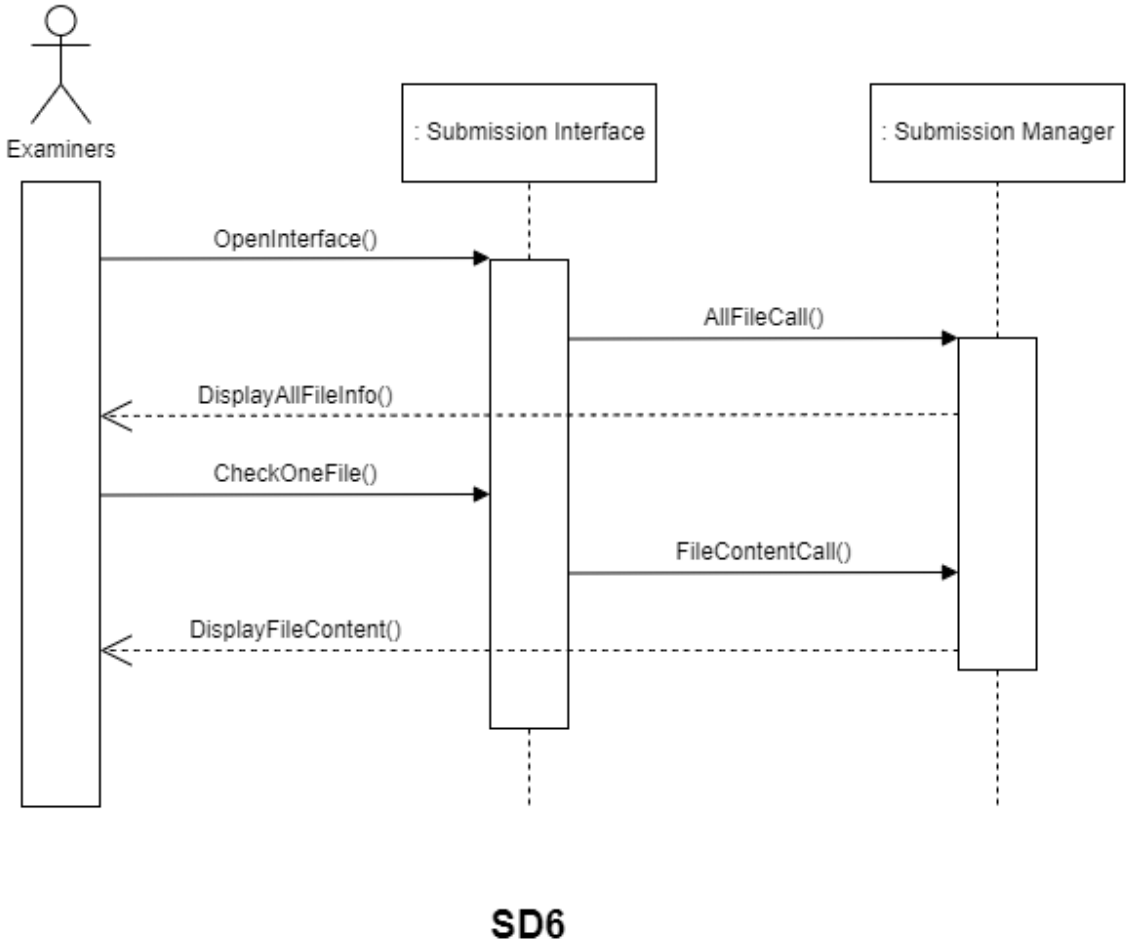

Figure 2.15: Sequence Diagram for <View Submission>

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>

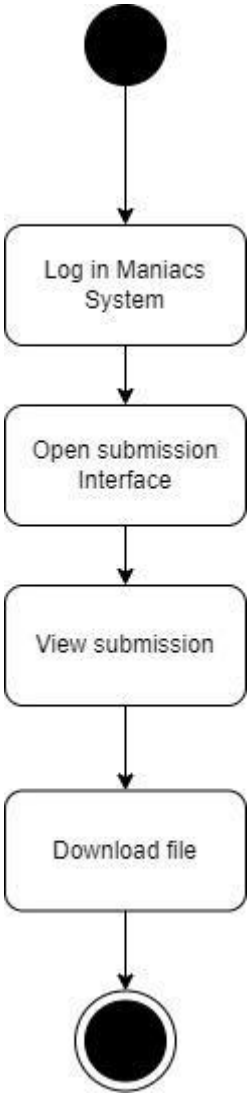

Figure 2.16: Activity Diagram for <Download Submission>

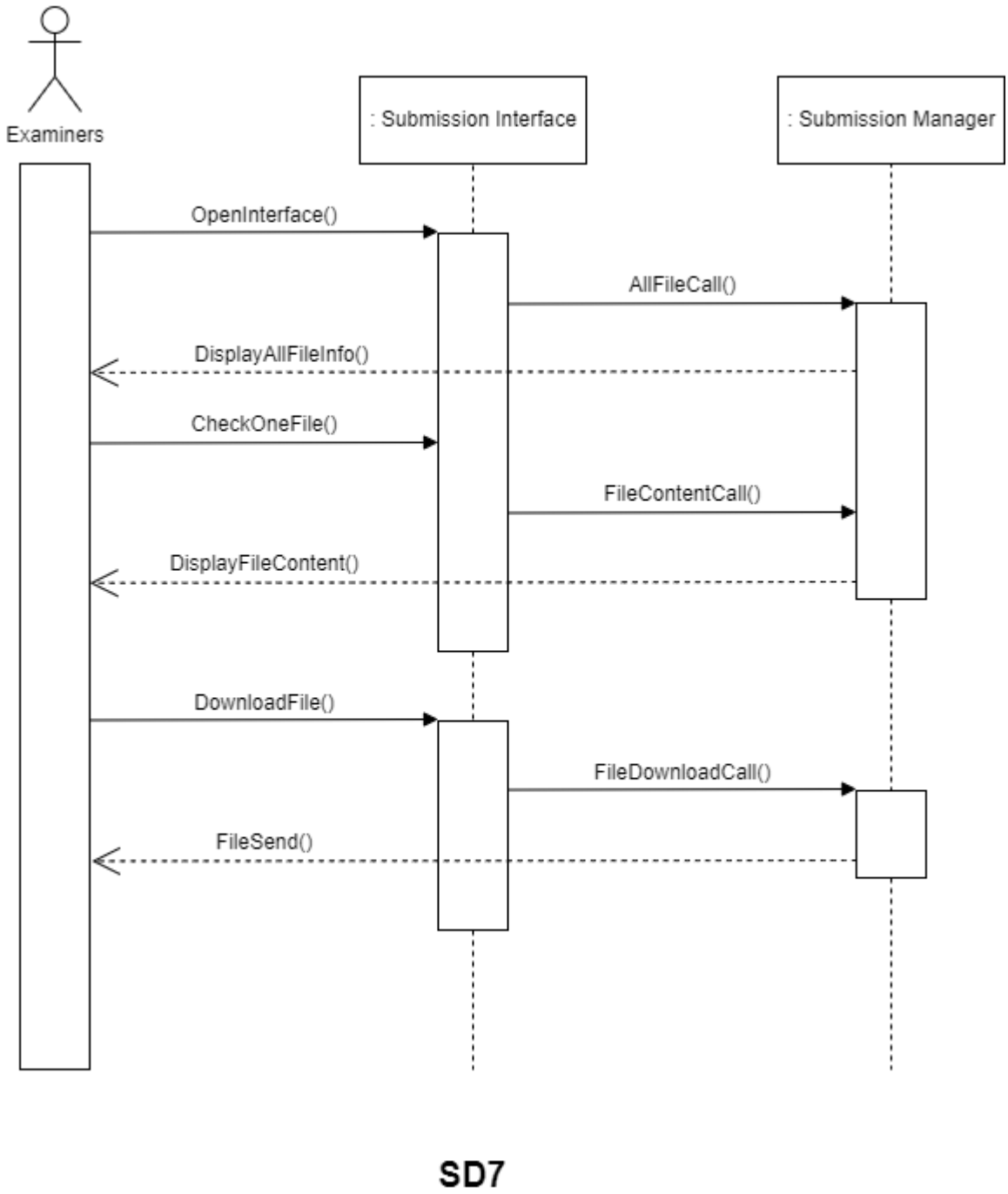

Figure 2.17: Sequence Diagram for <Download Submission>

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>

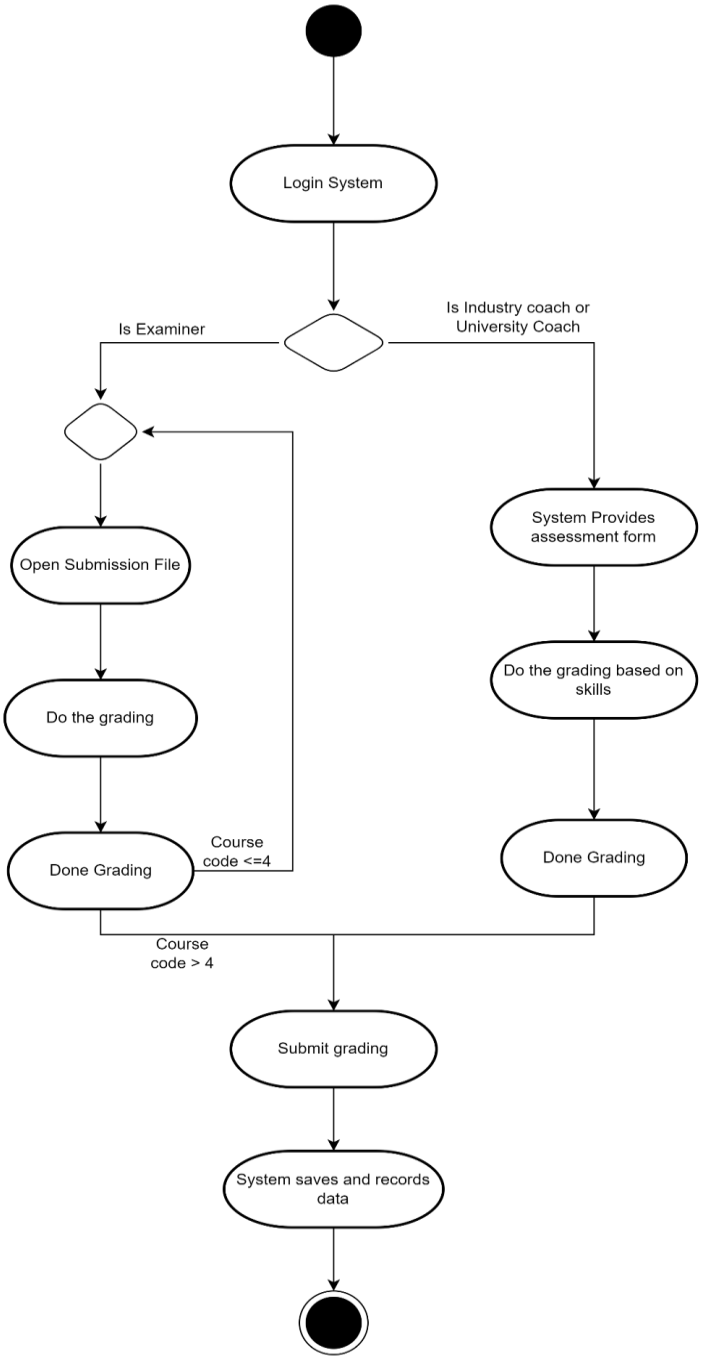

Figure 2.18: Activity Diagram for <Grading>

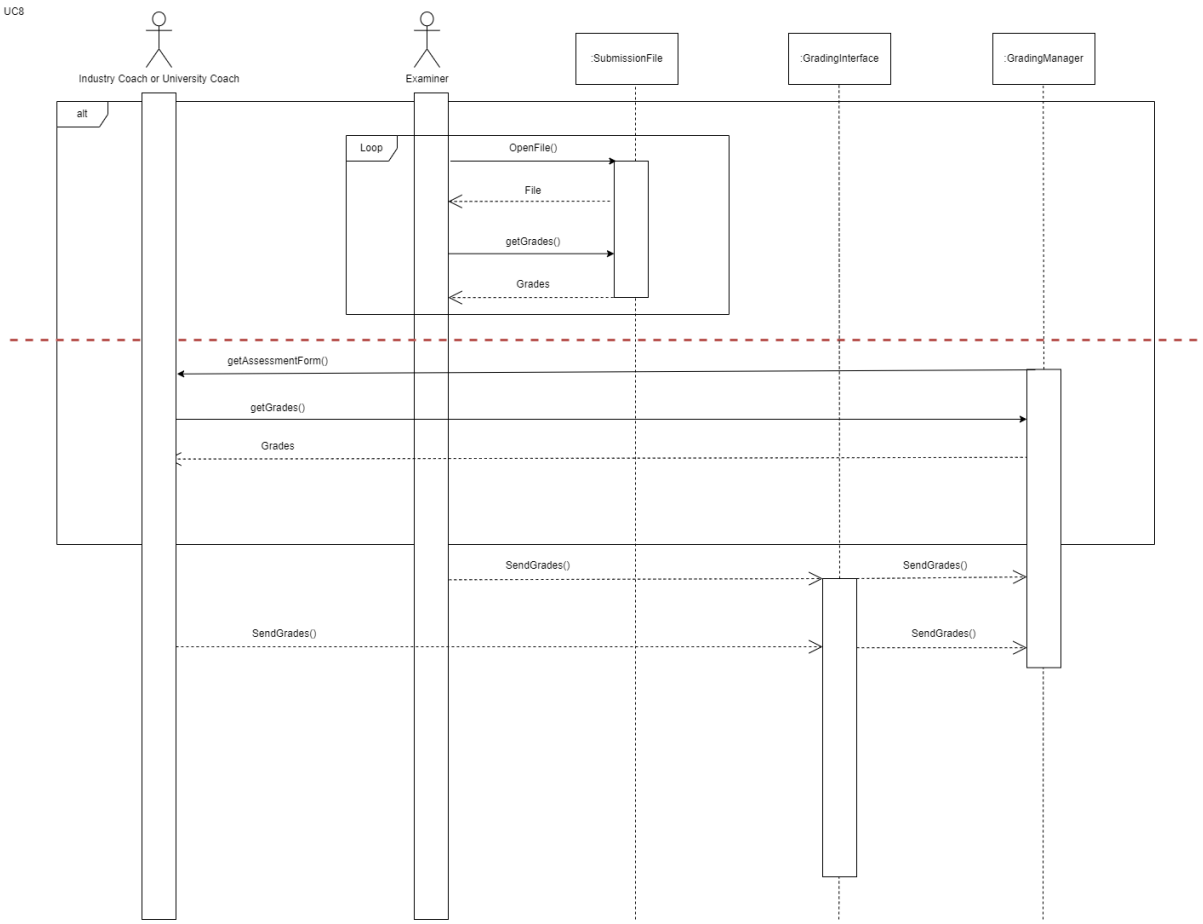

Figure 2.19: Sequence Diagram for <Grading>

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>

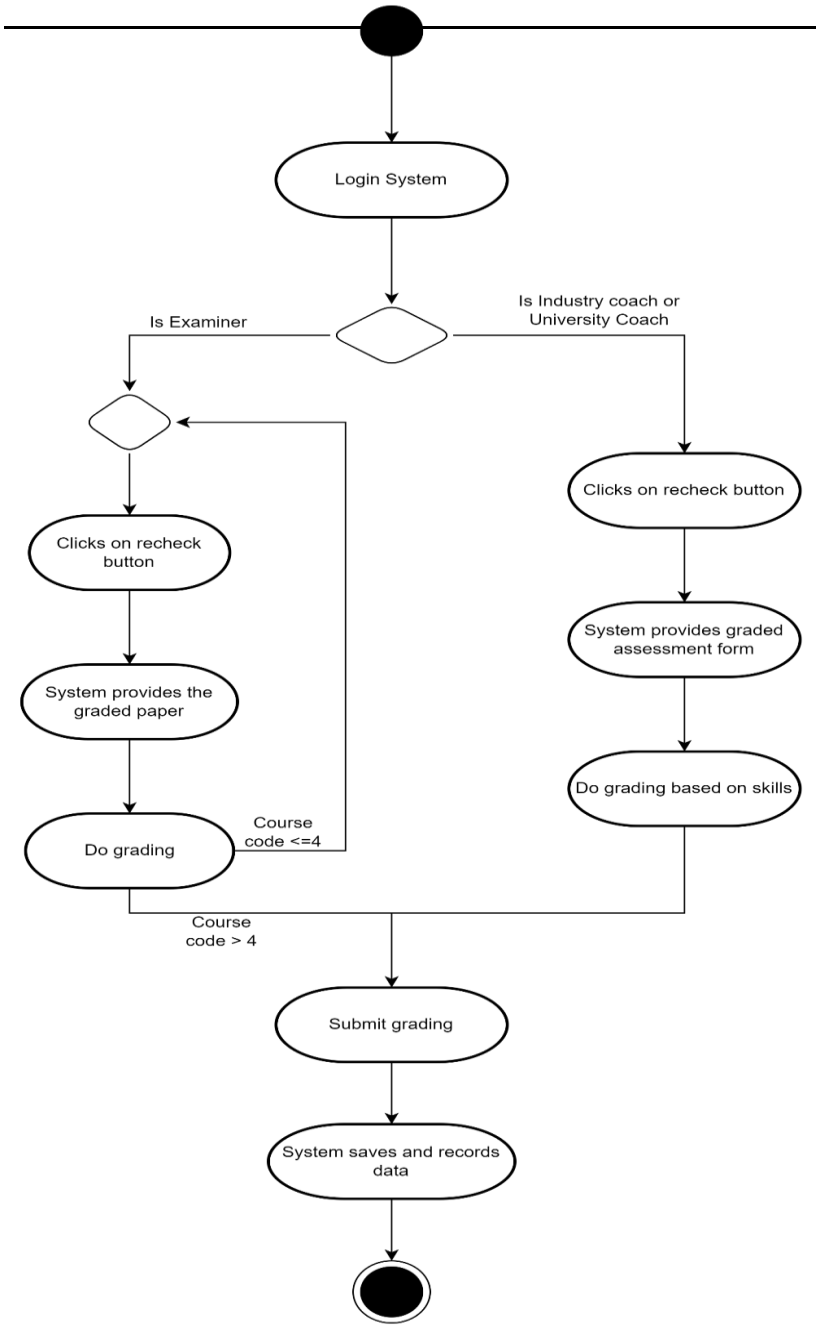

Figure 2.20: Activity Diagram for <Recheck Grading>

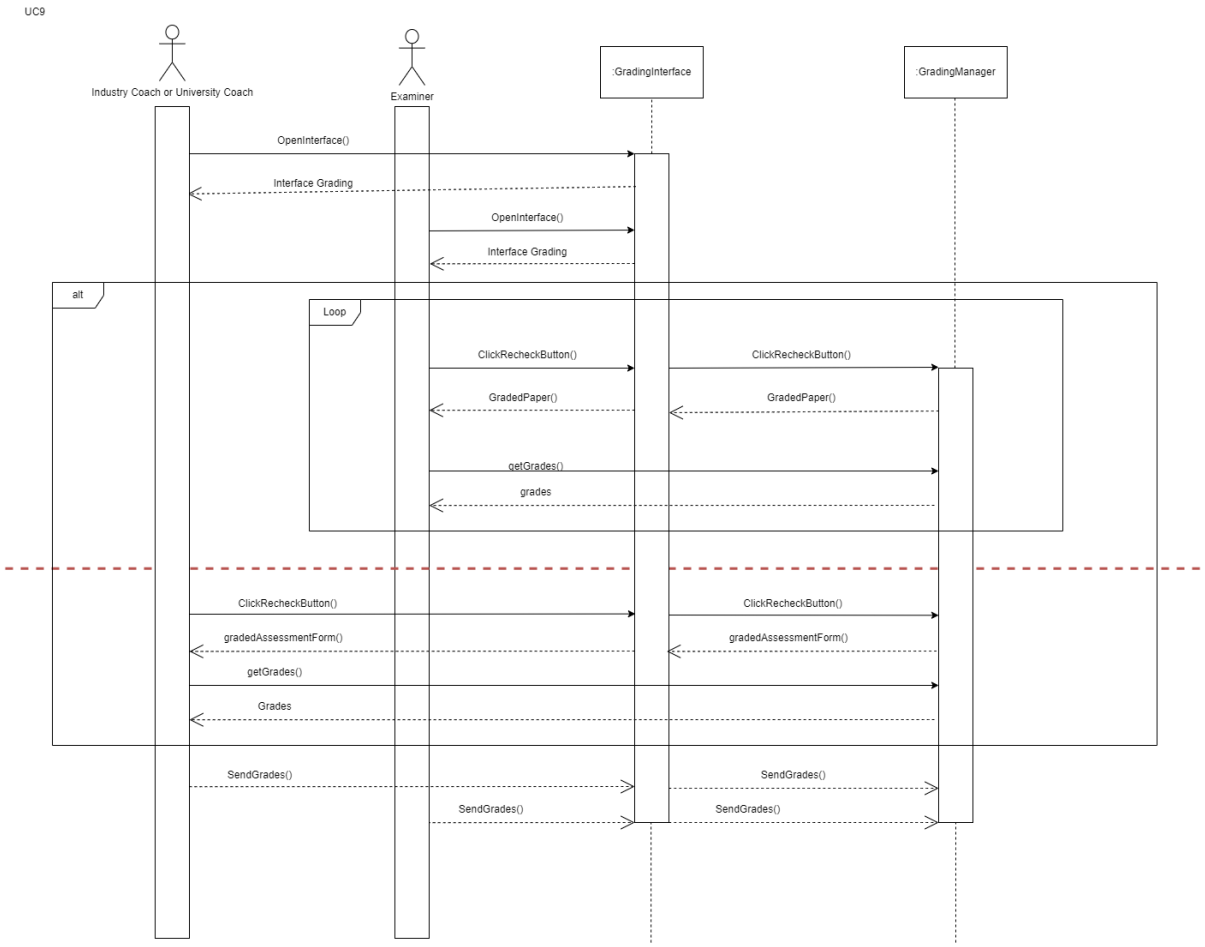

Figure 2.21: Sequence Diagram for <Recheck Grading>

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>

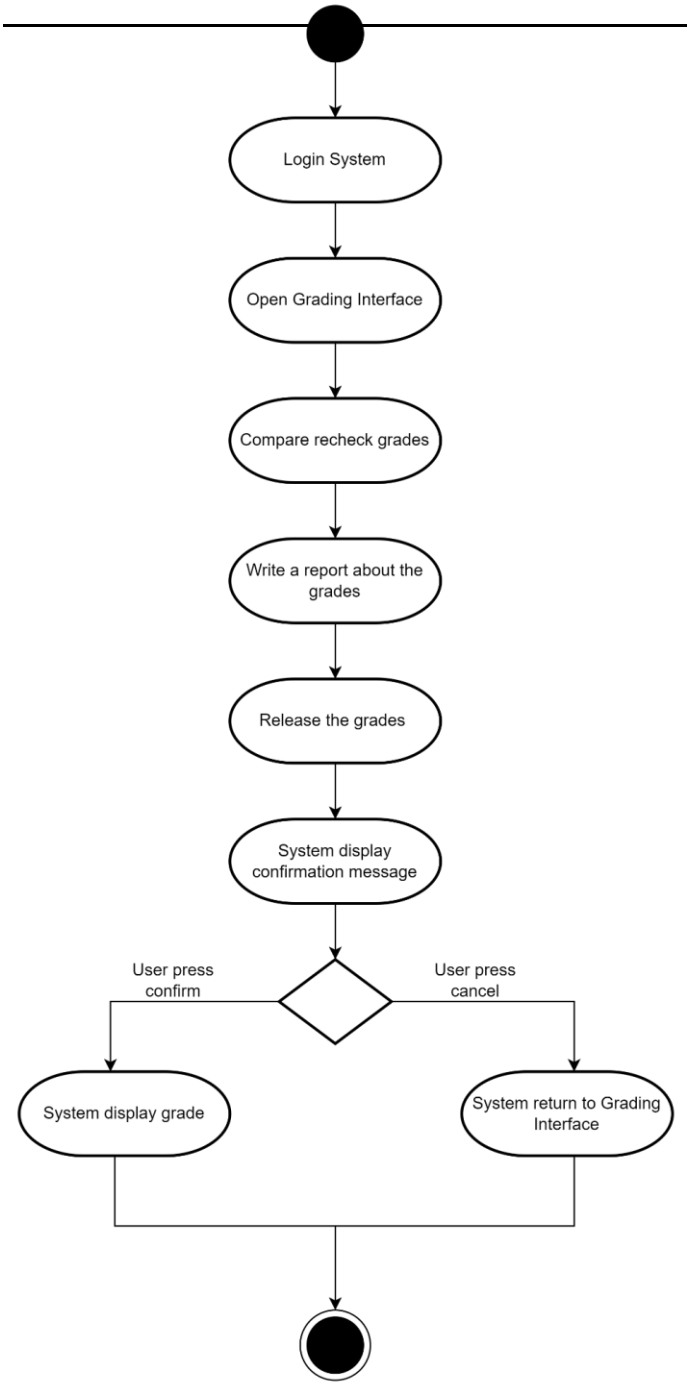

Figure 2.22: Activity Diagram for <Release Grading>

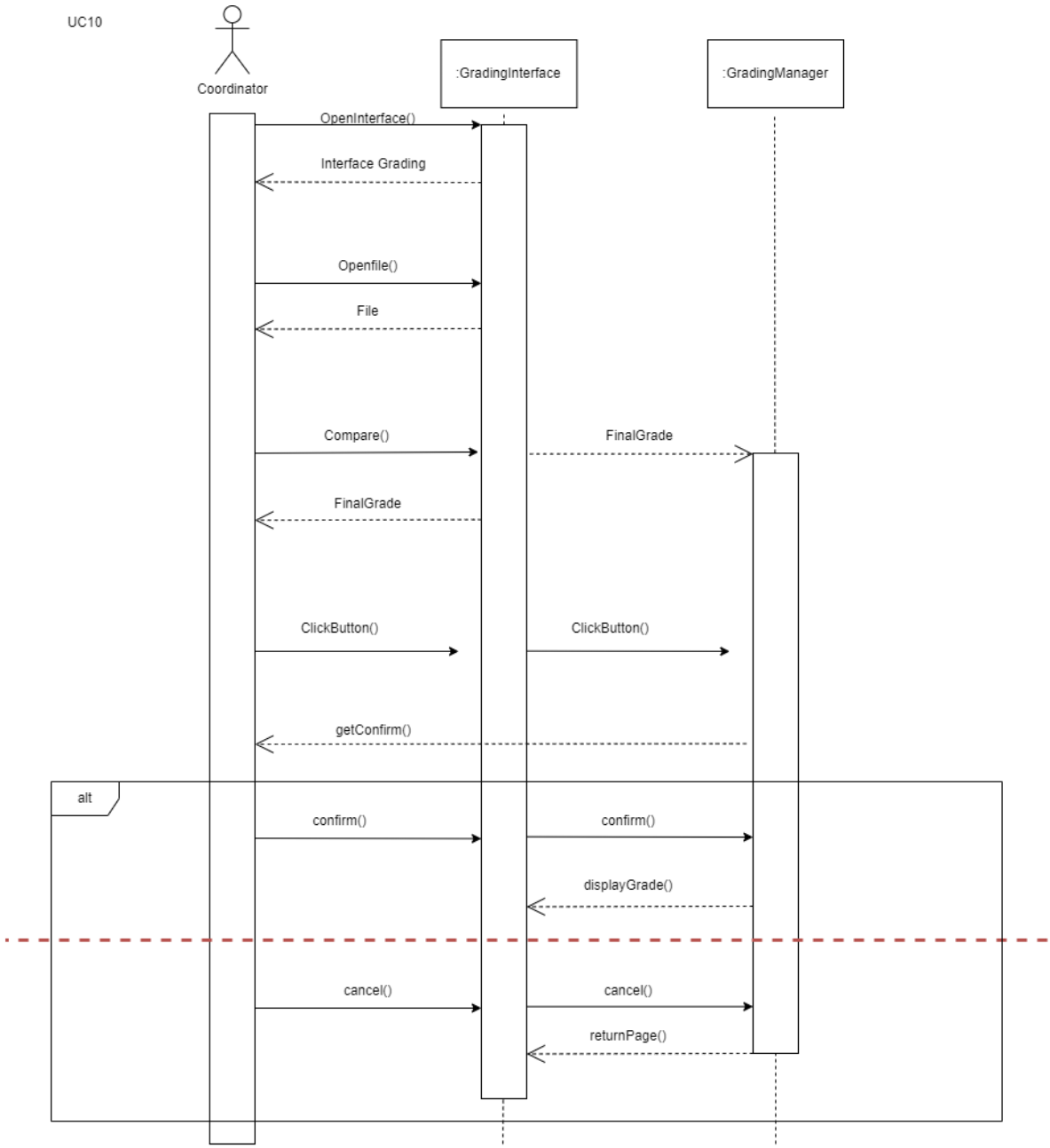

Figure 2.23: Sequence Diagram for <Release Grading>

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>

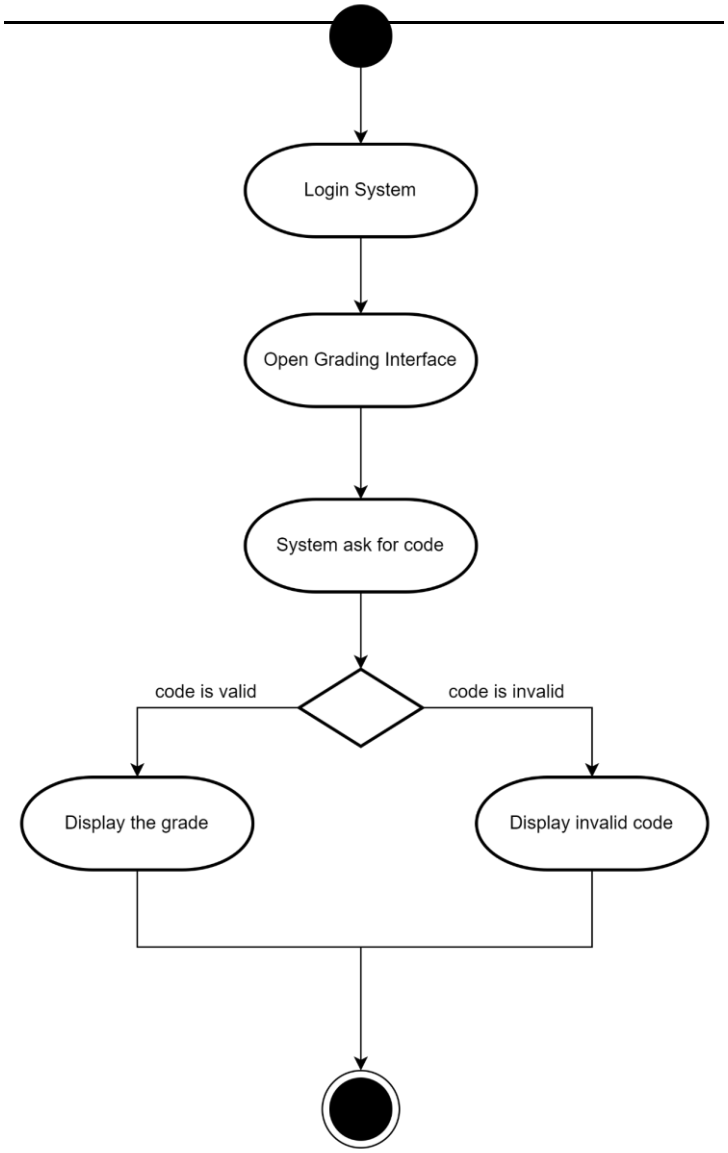

Figure 2.24: Activity Diagram for <View Grading>

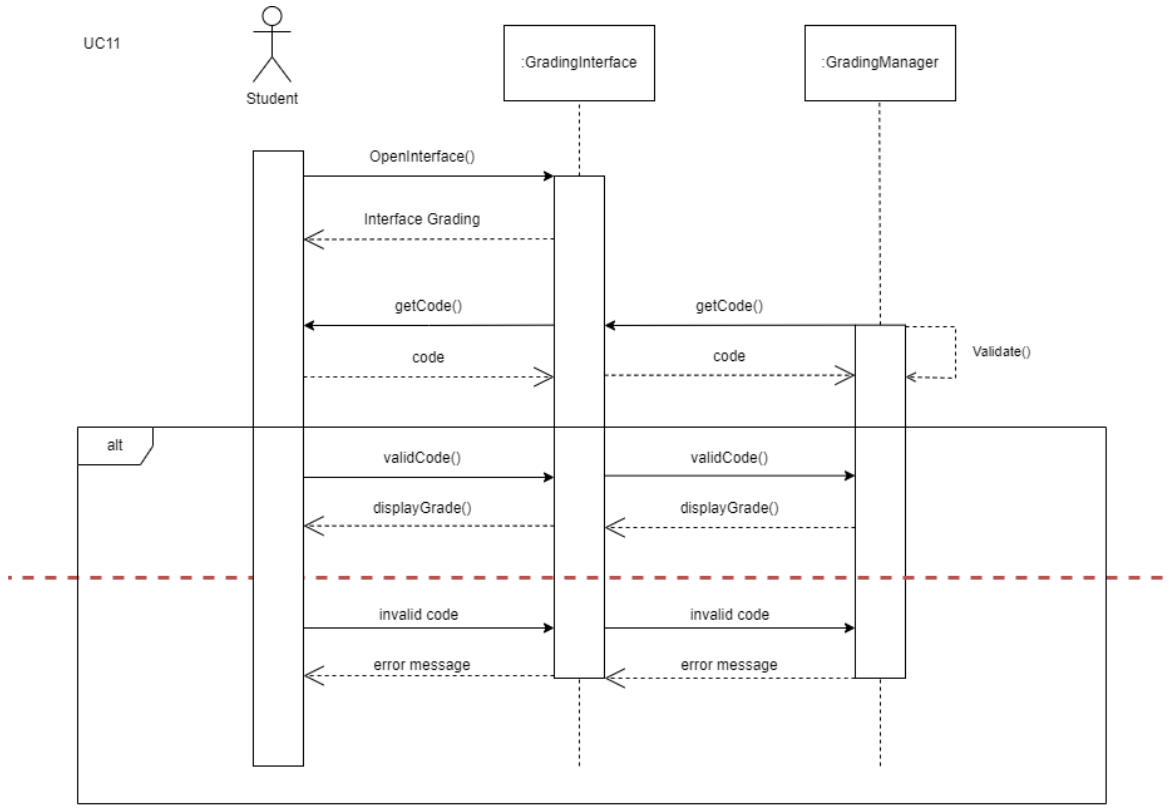

Figure 2.25: Sequence Diagram for <View Grading>

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

Use case: Contact Coaches
ID: UC12
Actors: Student, Industry Coach, University Coach
Extension Points: <<send message>> <<display message>>
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 <<send message>> <<display message>>
Postconditions: The student contacts the coach.

Table 2.12: Use Case Description for <Contact Coaches>

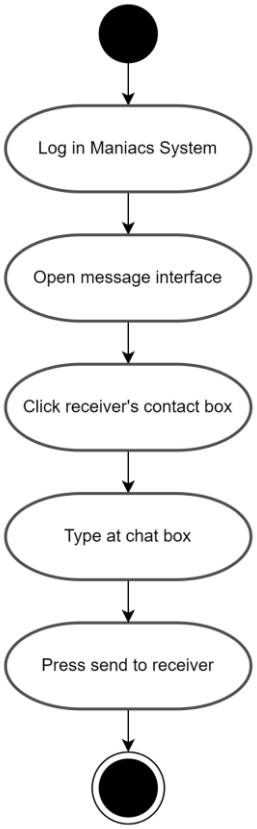

Figure 2.26: Activity Diagram for <Contact Coaches>

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.