SECJ2203: Software Engineering

System Documentation (SD)

2u2i Final Year Project with Industry (FYP-I)

Management System

Version 1.0

5 June 2022

School of Computing, Faculty of Engineering

Prepared by: Inferno

Name of Members:

1. Amir Iskandar Bin Norkhairulazaddin
2. Iman Ehsan Bin Hassan
3. Muhammad Aiman Bin Abdul Razak
4. Shahril Bin Saiful Bahri

Revision Page

a. Overview

In this system documentation version 1.0, it consists of the documentation components that are adapted from IEEE Recommended Practice for Software Requirements Specification, Software Design Description and Software Test Documentation that has been simplified and customized. In this version, it will cover all of the sections from the introduction until the specific requirements.

b. Target Audience

- Stakeholder
- Development Team
- Users

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
Shahril Bin Saiful Bahri (A20EC0144)	Team Leader	1.3 Definition, Acronyms and Abbreviations 1.4 References 2.1.1 User Interfaces 2.1.3 Software Interface 2.5 Software System Attribute <Use Case> UC007: Use Case <Notification Data> UC008: Use Case <View Notification> UC009: Use Case <Add Notification> UC010: Use Case <Update Notification>	Completed

		UC011: Use Case <Delete Notification> UC012: Use Case <Return Message>	
Amir Iskandar Bin Norkhairulazaddin (A20EC0011)	Team Member	1.1 Purpose 2.1.1 User Interfaces 2.1.2 Hardware Interface 2.2.13 UC013: Use Case <Upload Questions> 2.2.14 UC014: Use Case <Take Exam> 2.2.15 UC015 Use Case <View Grades> 2.4 Design Constraints	Completed
Iman Ehsan Bin Hassan (A20EC0048)	Team Member	1.2 Scope 2.1.1 User Interfaces 2.1.4 Communication interfaces 2.2.1 UC001: Use Case <Login> 2.2.2 UC002: Use Case <Verification>	Completed
Muhammad Aiman Bin Abdul Razak (A20EC0082)	Team Member	1.5 Overview 2.1.1 User Interfaces 2.2.3 UC003: Use Case <View Message> 2.2.4 UC004: Use Case <Add Message> 2.2.5 UC005: Use Case <Delete Message> 2.2.6 UC006: Use Case <Edit Message>	Completed

		2.3 Performance And Other Requirements	

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Shahril Bin Saiful Bahri (Team Leader)	Completed Chapter 1 and 2, Section 1.1 to 1.5 and 2.1 to 2.5	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. 10161998 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 - 5

1.1 Purpose 3

1.2 Scope 3, 4

1.3 Definitions, Acronyms and Abbreviations 4

1.4 References 4

1.5 Overview 4, 5

2 Specific Requirements 6 - 32

2.1 External Interface Requirements 6 - 8

2.1.1 User Interfaces 6, 7

2.1.2 Hardware Interfaces 7

2.1.3 Software Interfaces 7, 8

2.1.4 Communication Interfaces 8

2.2 System Features 8 - 31

2.2.1 UC001: Use Case <Login> 12, 13

2.2.2 UC002: Use Case <Verification> 14, 15

2.2.3 UC003: Use Case <View Message> 15, 16

2.2.4 UC004: Use Case <Add Message> 16, 17

2.2.5 UC005: Use Case <Delete Message> 17, 18

2.2.6 UC006: Use Case <Edit Message> 18, 19

2.2.7 UC007: Use Case <Notification Data> 19, 20

2.2.8 UC008: Use Case <View Notification> 20, 21

2.2.9 UC009: Use Case <Add Notification> 21, 22

2.2.10 UC010: Use Case <Update Notification> 22, 23

2.2.11 UC011: Use Case <Delete Notification> 23, 24

2.2.12 UC012: Use Case <Return Message> 24, 25

2.2.13 UC013: Use Case <Upload Question> 25, 26, 27

	2.2.14	UC014: Use Case <Take Exam>	27, 28, 29
	2.2.15	UC015: Use Case <View Grades>	29, 30, 31
2.3		Performance and Other Requirements	31
2.4		Design Constraints	31, 32
2.5		Software System Attributes	32

1. Introduction

1.1 **Purpose**

This Software Documentation describes the proposed 2u2i Final Year Project with Industry (FYP-I) Management System in which several sections will be clarified. The sections that will be described for this software documentation are the specific requirements, detailed description of components and the requirements matrix. This document contains the system design that explains the architecture of the system. This also includes the complete description of the components used in the system. The data design is also included in this documentation that contains the data description and data dictionary. This documentation was intended for the stakeholder, development team and the system's user. This is to provide them with necessary information about the system that will help them to understand more about the function available in the system. The stakeholder can make a decision, approval or changes to the system based on the documentation. Other than that, the development team can use the documentation to show and monitor their progress in the development process. The audience of the documentation also can provide necessary feedback during the development based on the documentation.

1.2 **Scope**

The software product is the internship system which mainly focuses on final year students that undergo internships. It is a system where all the students will interact to find their right place in the internship. The system add, delete and edit the messages to ease the process of the communication .This helps the students to ask their lecturer if they have any inquiries .Other than that, there are notification systems where the student and lecturer can view ,add,update and delete notifications .This helps students to stay up to date with any new information that is related to internship news.

The scope of this system includes

- User access by authentication
- View message
- Add message
- Delete message
- Edit message
- View notification
- Delete notification
- Upload questions
- View grades
- Take exam

This internship system is an online application that can be accessed by the website which provides both lecturer and student access and find the information more easily. The goal of the system is to create a platform for the student to find their company to do the internships. Meanwhile, the objective is to make lecturer and supervisor jobs easier and save their time of helping their students to find the place for an internship.

1.3 **Definitions, Acronyms and Abbreviation**

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

Terms, Acronyms/Abbreviation	Meaning
SD	System Documentation
Stakeholder	Individual or group that has an interest in any decision or activity of an organization
Inquiries	An act of asking information
Mnemonic	A system such as pattern of letters, ideas, or associaciations which assists in remembering something.
C++	An object-oriented computer language.
IDE	Integrated development environment is software for building applications that combines common developer tools into a single graphical user interface (GUI).

1.4 **References**

1. Sommerville, I. 2016. “Software Engineering”, 10th Edition, Pearson.

1.5 **Overview** .

The 2u2i Final Year Project with Industry (FYP-I) Management System consists of introduction, specific requirements, system architectural design, detailed description of components, data design, interface design, requirements matrix, test cases, as well as appendices. In the introductory phase, we describe the purpose and scope of the project. We have also provided definitions and abbreviations, as well as some references. In specific

requirements, we have covered several external interface requirements which includes user interfaces, hardware interfaces, software interfaces, and communication interfaces. We also explained about several system features which include user access, communication, notification data and assessment. The performance of the project as well as the constraints were also discussed. Other than that, software system attributes were also discussed. In system architectural design, we have explained about architectural style and rationale as well as component model. For detailed description of components, the complete package diagram as well as detailed description of its subsystems were discussed. Data design section includes the data description and data dictionary. For interface design, we have displayed an overview of the interface as a whole. We have also discussed the requirements matrix. For the last part, we discuss the test cases in the project.

2. Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

User access

- Users are required to enter their username and password.
- Student will enter the main page of student's profile while supervisor will enter the main page of supervisor profile
- Access will be given to the right username and password through verification
- The username and password will be save for future login
- There will be authentication for the user to make sure that the right user enter the system

Notification

- Users can access the notification by going to the interface and choosing notification
- Students are only able to view the notifications that are available in the notification database.
- Coordinator able to modify the notifications database by adding more notifications to the database.
- Coordinator able to modify the notifications database by updating the searched notifications in the database.
- Coordinator able to modify the notification database by deleting the searched notifications in the database.
- The system will display “no data found” if there is no data found in the database.

Communication

- Students, supervisors, coordinators, coaches, and examiners are able to have full accessibility to view, add, delete, and edit a message.
- Users are able to access messages in the system by going to the interface and choosing notification
- Users are able to view existing messages by them or by other people in the system.
- Users are able add a new message in the system.
- Users are able to delete an existing message in the system.
- Users are able to edit an existing message created by them in the system.

- The system will keep an organized list of messages added by users of the system.

Assessment

- Examiners can upload the questions by adding it into the database.
- Students can take the exam online by attempting it on the system.
- Students can view their grade after they have finished their exam.
- Examiners can view their student's grade after their student;s has finished the exam.

2.1.2 Hardware Interfaces

The 2u2i Final Year Project with Industry (FYP-I) Management System is a web based system. The system will be connected to the servers provided by Amazon Web Services and database from MySQL. The web server will have 8 GB of RAM to support the system load while using it. The database will have 1TB capacity to store all of the data. The system can be accessed from multiple devices such as smartphone, laptop or desktop. It also can be accessed by computers that use different operating systems such as Windows, MacOS or Linux. The clients are required to use modern web browsers such as Google Chrome, Microsoft Edge or Mozilla Firefox. The clients also need to have an Internet connection with minimum bandwidth of 10MB/s to access the system. The data from the server will be transferred and received by the clients using basic networking protocols. They also need to enable cookies to ensure they get the best experience using the system.

2.1.3 Software Interfaces

1. **Google Chrome (Web Browser)**
   Name: Google Chrome
   Mnemonic: Chrome, Google
   Specification number: 103.0.000.00
   Version number:
   103.0.5060.33 (Windows & macOS & Linux)
   103.0.5060.33(Android)
   103.0.5060.34(iOS)
   Sources: www.google.com/chrome/
   Discussion: We use this web browser to operate our systems because it's easier and fast for our users.

2. **MySQL (Database)**

   Name: MySQL

   Mnemonic: CREATETABLE, SELECT, INSERT INTO

   Specification number: 8.0.00

   Version number: 8.0.28

   Sources: https://www.mysql.com/

   Discussion: The database that we used to create, store, search, and also delete all the data in the system.

3. **VSC (Programming Software)**

   Name: Visual Studio Code

   Mnemonic: C++, Java, Python

   Specification number: 1.67.0

   Version number: 1.67.2

   Sources: https://code.visualstudio.com/

   Discussion: The best IDE for us to operate our programming software using many languages such as MySQL and also C++.

2.1.4 Communication Interfaces

This should list the different communication interfaces, such as local network protocols and so on. For receiving student and college management information and encrypting all important information from and into our system for data transfer security, our system will employ Hypertext Transfer Protocol Secure (https). HTTP is a protocol for sending and receiving data.

TCP/IP (Transmission Control Protocol/Internet Protocol) is also used by the system to link network devices via the internet. It may also be used to send data across the Internet. For data transfer, TCP is used. TCP is compatible with IP and enhances it. It's used to send and receive data through a network. It breaks down any communication into a sequence of packets that are transferred from source to destination, where they are reassembled.

2.2 System Features

The system features included in the use case diagram. The use case diagrams will be used to explain the features in the system. The domain model will be used to represent the relationship between the classes in the system. There are also use case descriptions

and sequence diagrams to explain all of the processes involved in the 2u2i Final Year Project with Industry (FYP-I) Management System. The use case description will explain all of the condition and flow of the events of a use case while the sequence diagram will explain the interaction of the classes while the use case is executed.

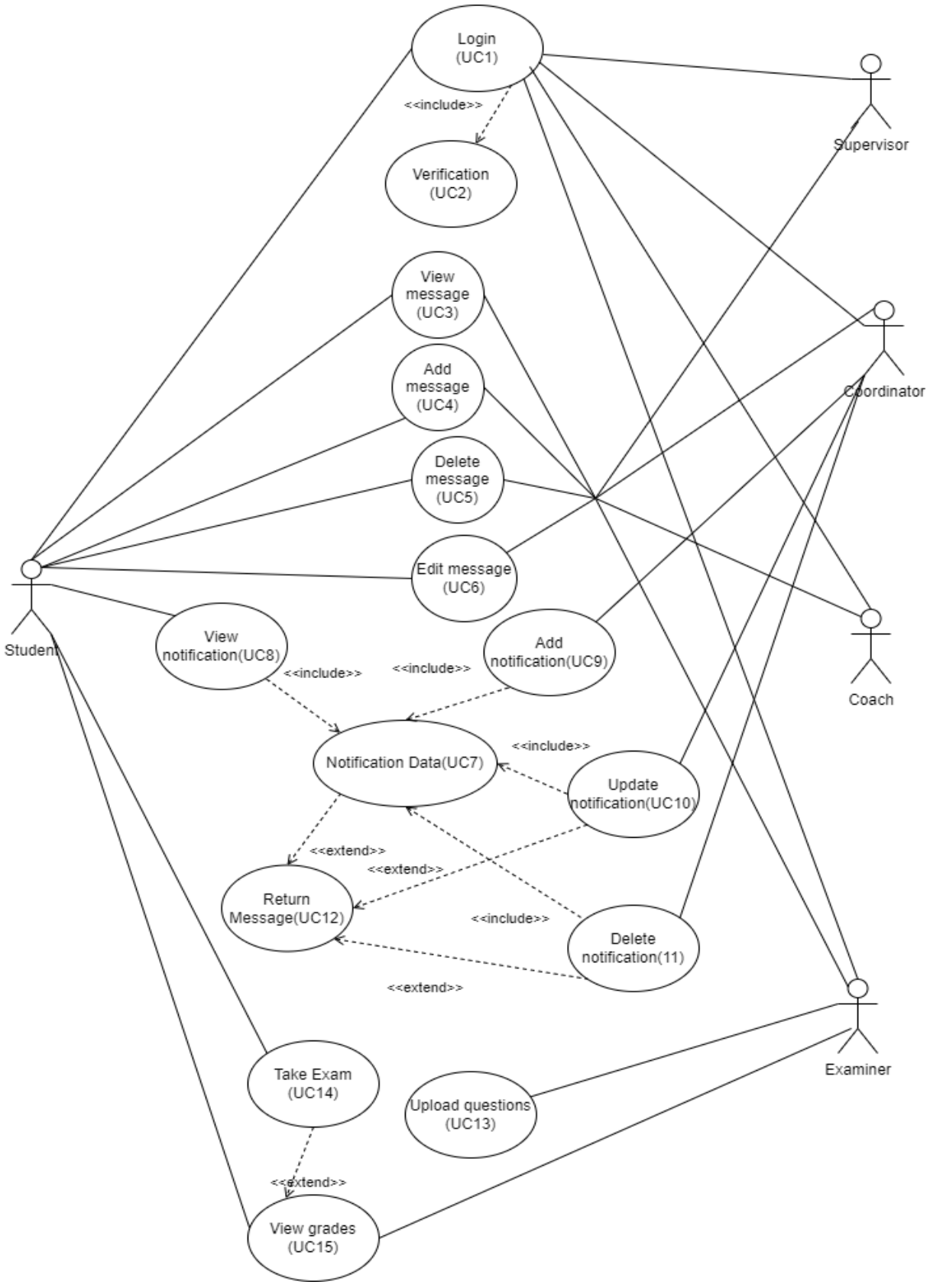

Figure 2.1: Use Case Diagram for <2u2i Final Year Project with Industry (FYP-I) Management System>

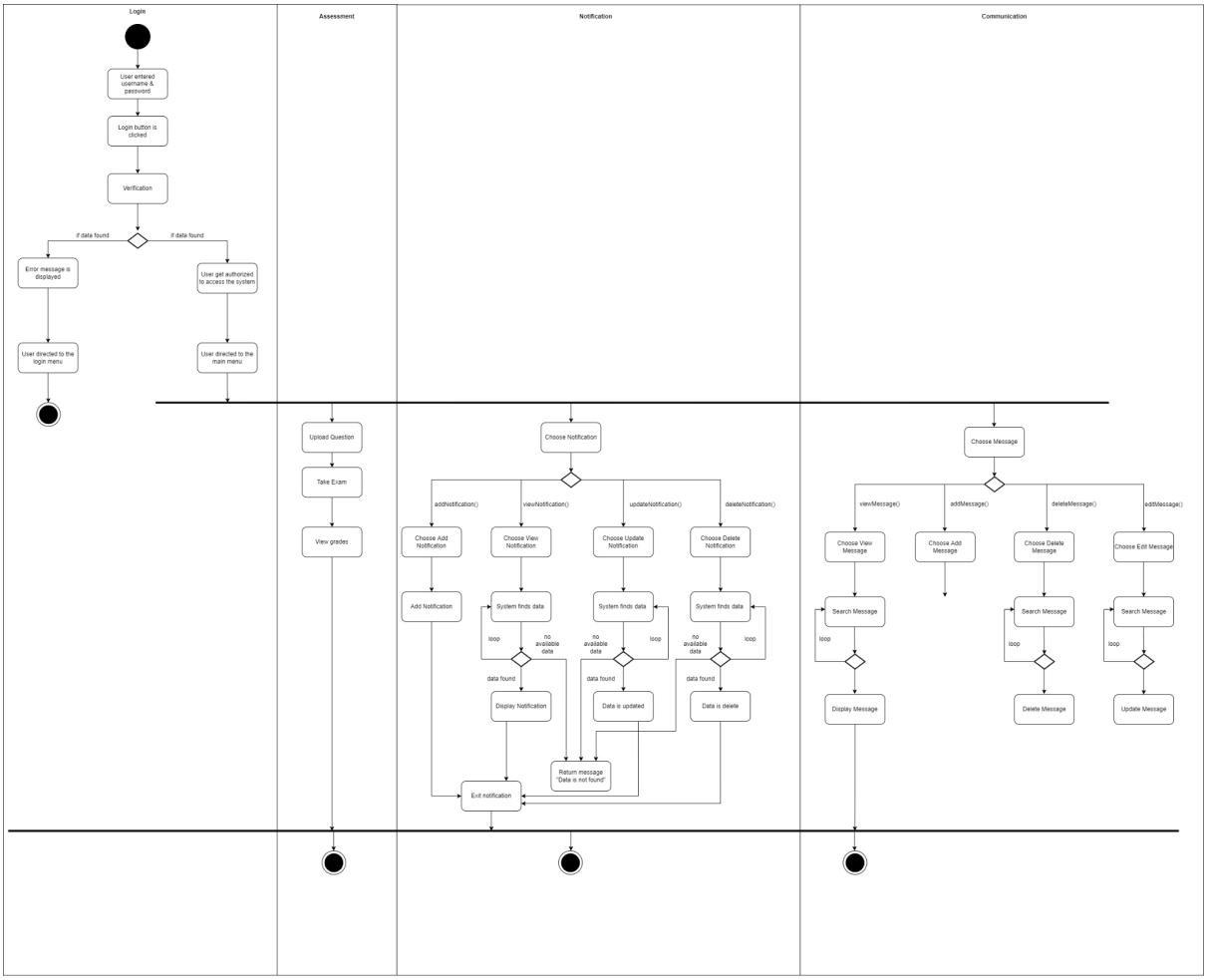

Figure 2.2: Activity Diagram for <2u2i Final Year Project with Industry (FYP-I) Management System>

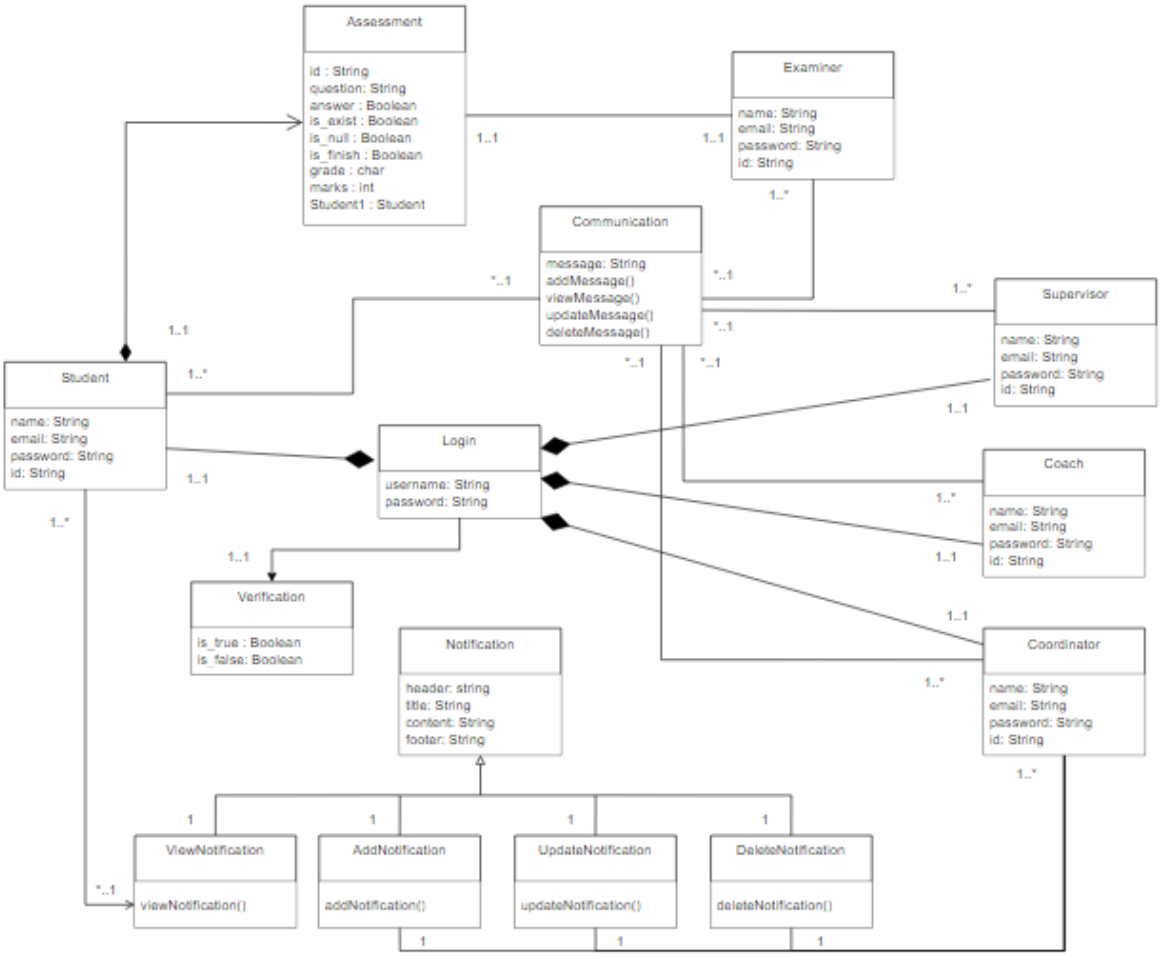

Figure 2.3: Domain Model for <2u2i Final Year Project with Industry (FYP-I) Management System>

2.2.1 UC001: Use Case <Login>

Use Cases: Login
ID: UC1
Actors: A1 Student A2 Supervisor A3 Coordinator A4 Coach A5 Examiner
Includes: UC2 Verification
Extensions Point:
Preconditions:
Flow of Events: 1. Users enter their username and password to the system. 2. Users click on the login button to proceed accessing the system. 3. Include Verification
Postconditions:

Figure 2.4: Use Case Description for <Login>

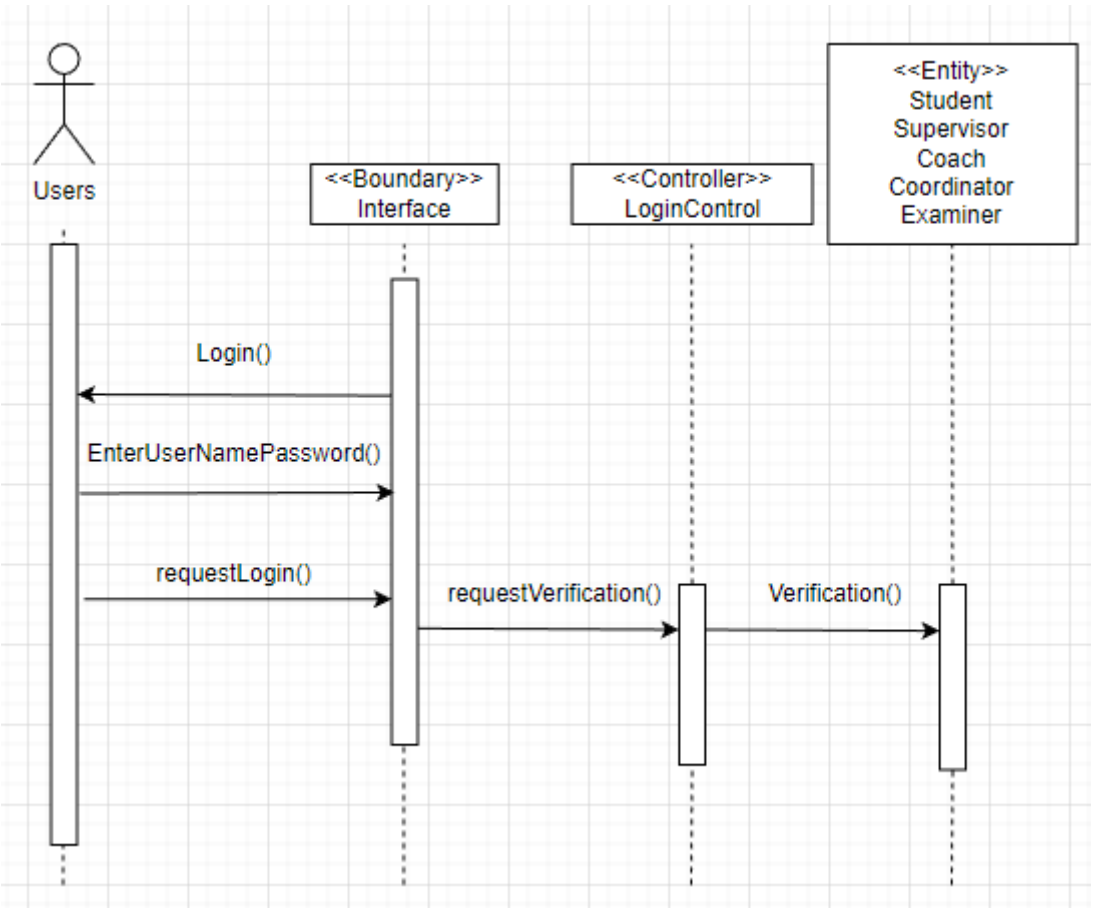

Figure 2.5: Sequence Diagram for <Login>

2.2.2 UC002: Use Case<Verification>

Use Cases: Verification
ID: UC2
Actors:
Includes:
Extensions Point:
Preconditions: 1. The username and password have been entered by the user.
Flow of Events: 1. The username and password will be verified with the data in the database. 2. If the username and password matched with the database 2.1 The user will be authorized and can access the system. 2.2 The user will be directed to the main menu. 3. Else 3.1 The system will display error message and redirected to login page. 3.2 The user needs to enter the correct username and password.
Postconditions: 1. Users are sent to the main page and can access the system.

Figure 2.6: Use Case Description for <Verification>

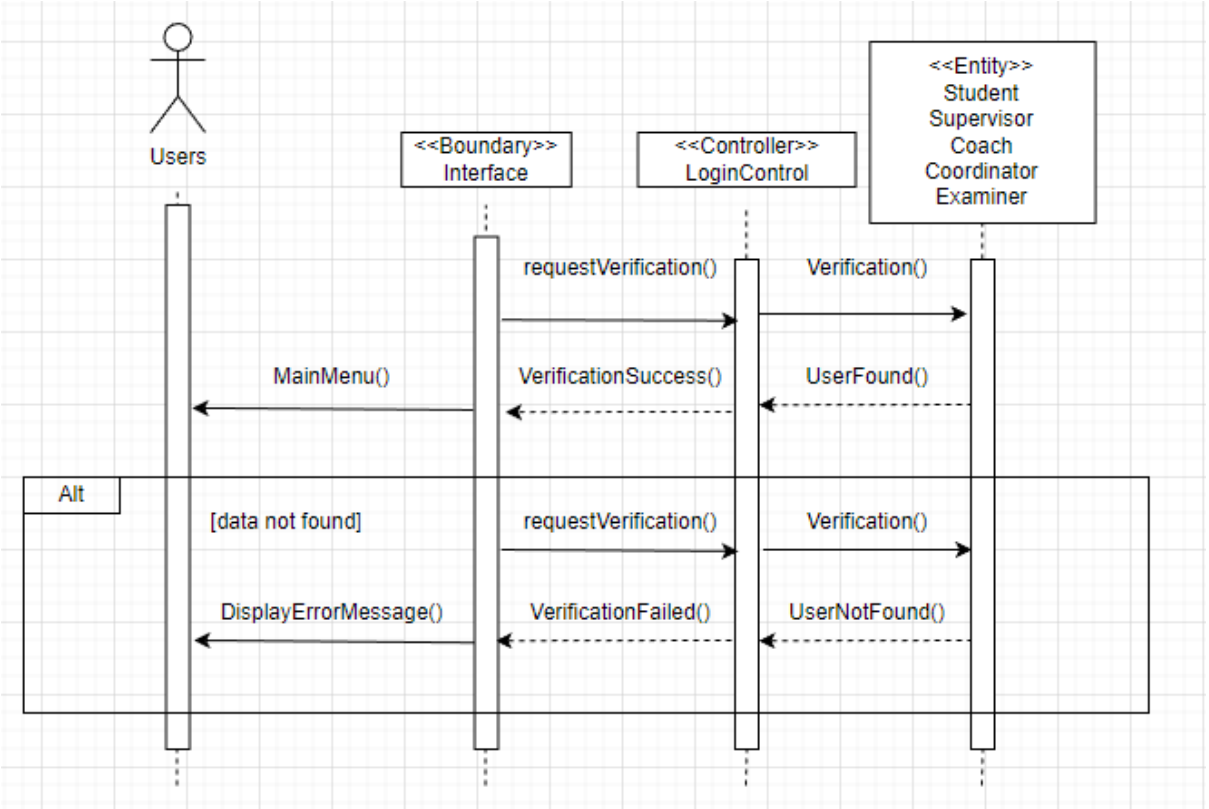

Figure 2.7: Sequence Diagram for <Verification>

2.2.3 UC003: Use Case<View Message>

Use Case: View Message
ID: UC3
Actor: A1 Student A2 Supervisor A3 Coordinator A4 Coach A5 Examiner
Includes:
Extension points:
Preconditions: 1. The actor has logged in to the system. 2. The actor has added message.
Flow of Events: 1. The actor logs into the system. 2. The system gives choices to the actor for messaging. 3. The actor chooses to view a message. 4. The system shows the actor the existing message. 5. The actor logs out of the system.
Postconditions: 1. The message has been viewed by the actor.

Figure 2.8: Use Case Description for <View Message>

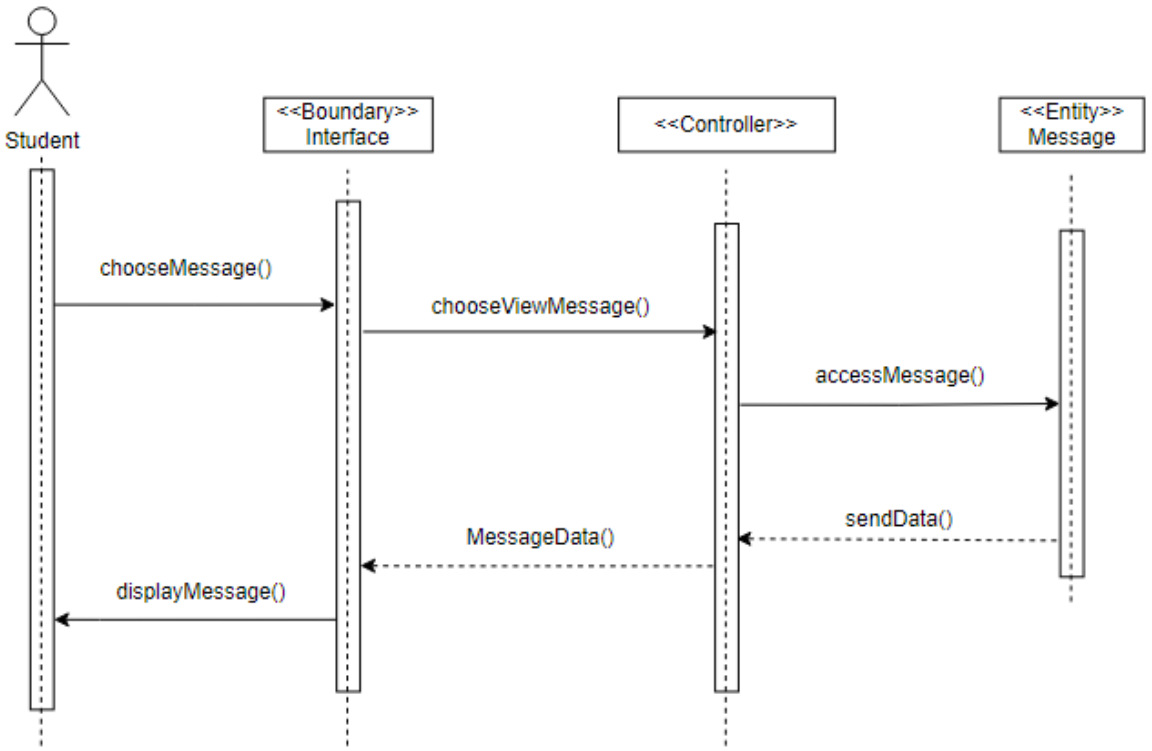

Figure 2.9: Sequence Diagram for <View Message>

2.2.4 UC004: Use Case<Add Message>

Use Case: Add Message
ID: UC4
Actor: A1 Student A2 Supervisor A3 Coordinator A4 Coach A5 Examiner
Includes:
Extension points:
Preconditions: 1. The actor has logged in to the system.
Flow of Events: 1. The actor logs into the system. 2. The system gives choices to the actor for messaging. 3. The actor chooses to add a message. 4. The system displays a text box to add a message. 5. The actor fills in the text box for the message. 6. The system updates. 7. The actor logs out of the system.
Postconditions: 1. The message has been added by the actor.

Figure 2.10: Use Case Description for <Add Message>

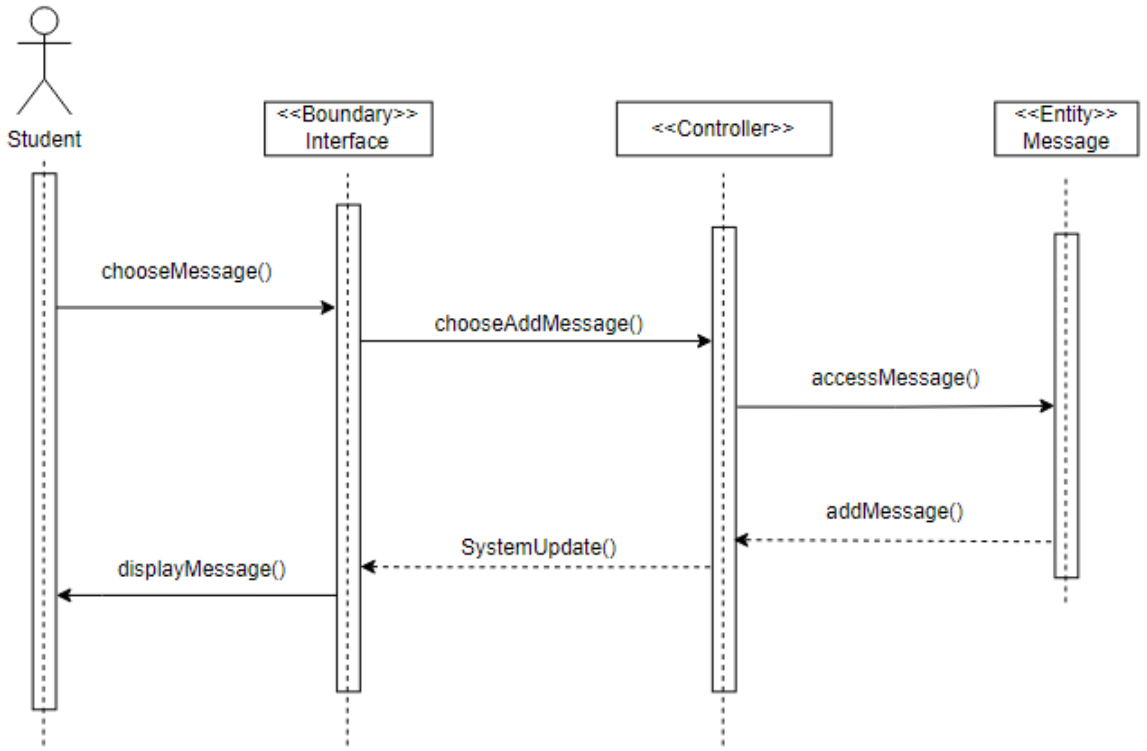

Figure 2.11: Sequence Diagram for <Add Message>

2.2.5 UC005: Use Case<Delete Message>

Use Case: Delete Message
ID: UC5
Actor: A1 Student A2 Supervisor A3 Coordinator A4 Coach A5 Examiner
Includes:
Extension points:
Preconditions: 1. The actor has logged in to the system. 2. The actor has added message.
Flow of Events: 1. The actor logs into the system. 2. The system gives choices to the actor for messaging. 3. The actor chooses to delete a message. 4. The system shows list of messages to delete. 5. The system asks for confirmation on deleting the message. 6. The system updates. 7. The actor logs out of the system.
Postconditions: 1. The message has been deleted by the actor.

Figure 2.12: Use Case Description for <Delete Message>

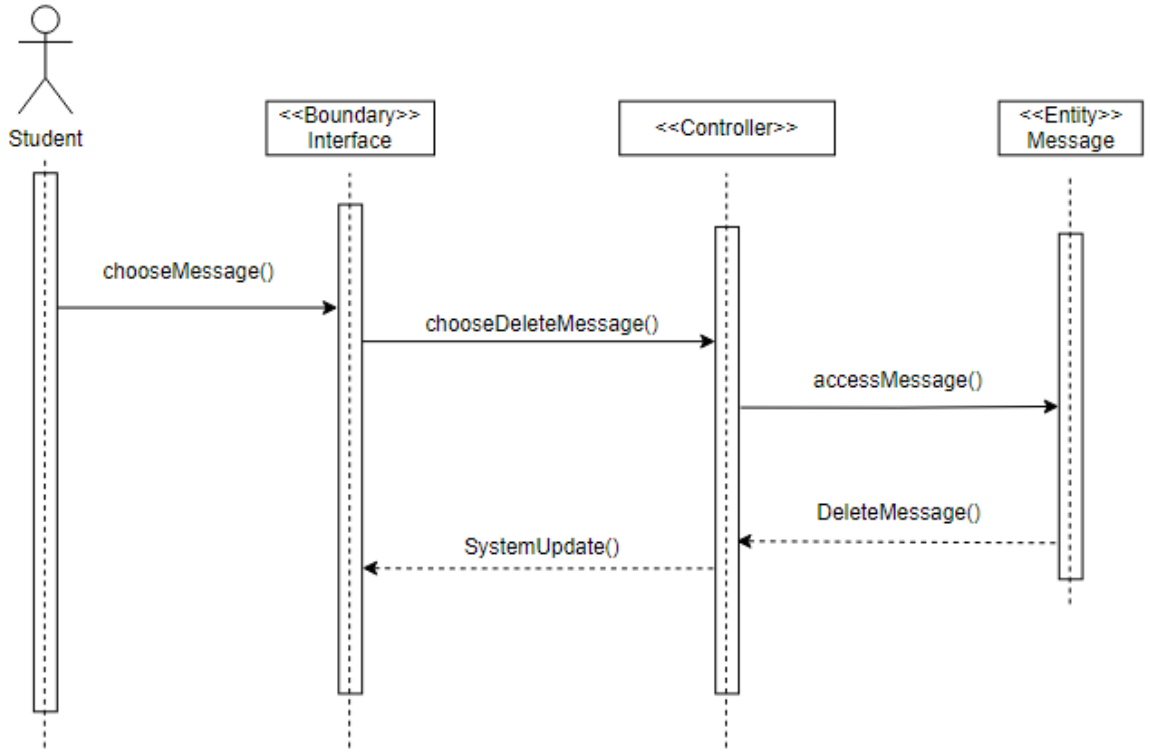

Figure 2.13: Sequence Diagram for <Delete Message>

2.2.6 UC006: Use Case<Edit Message>

Use Case: Edit Message
ID: UC6
Actor: A1 Student A2 Supervisor A3 Coordinator A4 Coach A5 Examiner
Includes:
Extension points:
Preconditions: 1. The actor has logged in to the system. 2. The actor has added message.
Flow of Events: 1. The actor logs into the system. 2. The system gives choices to the actor for messaging. 3. The actor chooses to edit a message. 4. The system displays a text box to edit the message. 5. The actor fills in the text box for the message. 6. The system updates. 7. The actor logs out of the system.
Postconditions: 1. The message has been edited by the actor.

Figure 2.14: Use Case Description for <Edit Message>

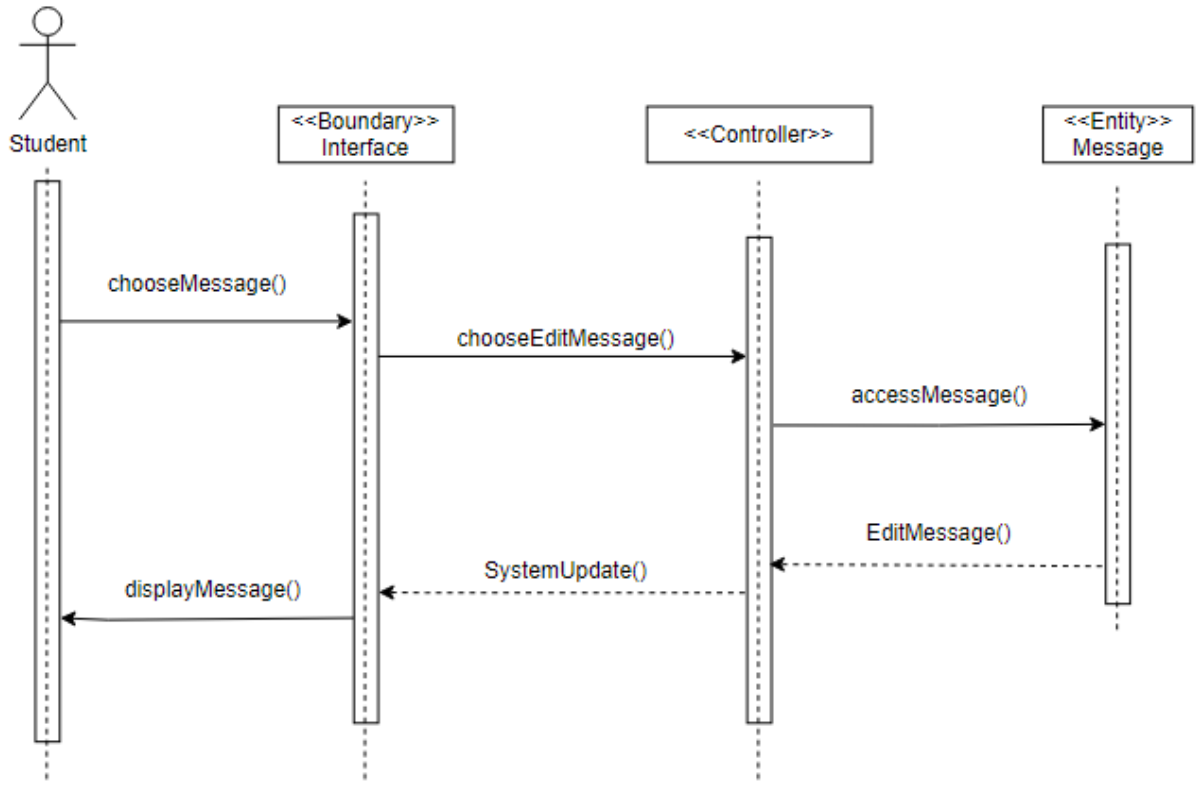

Figure 2.15: Sequence Diagram for <Edit Message>

2.2.7 UC007: Use Case <Notification Data>

Use cases: Notification Data
ID: UC7
Extends: <returnMessage>
Preconditions:
Flow of Events: 1. If there are data in notification 1.1. Return data to users. 1.2. System returns back to previous use case. 2. Else 2.1. There are no data available in the system <returnMessage>
Postconditions: 1. User able to get the data from Notification Data 2. User will get an error message if there is no data.

Figure 2.16: Use Case Description for <Notification Data>

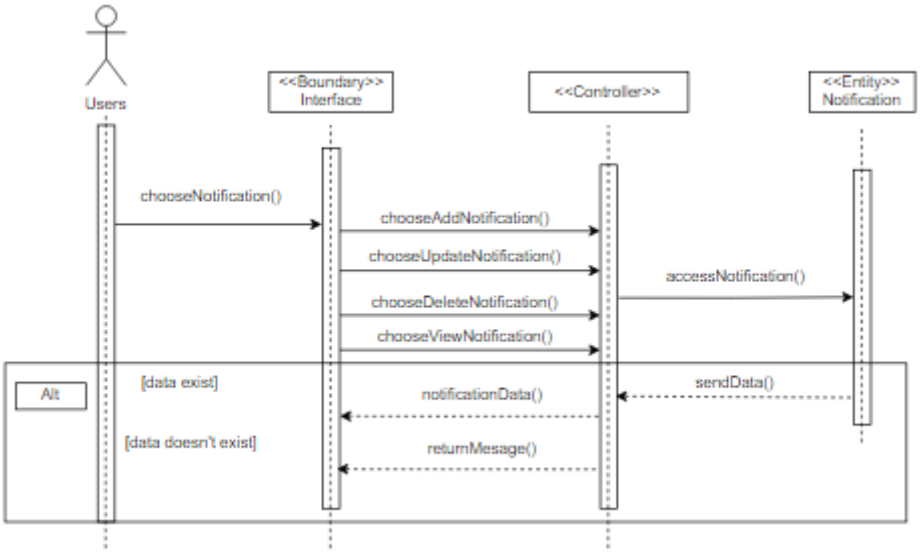

Figure 2.17: Sequence Diagram for <Notification Data>

2.2.8 UC008: Use Case <View Notification>

Use cases: View Notification
ID: UC8
Actors: A1 Student
Includes: UC7 Notification Data
Preconditions: 1. The student has logged in into the system. 2. Student has entered the interface of the system.
Flow of Events: 1. Student selects “View Notification” 2. include (Notification Data) 3. System will send a request to access the notification database 4 For each data found 4.1. System will print out notification data 5. There is no more data 5.1 System will exit the use case
Postconditions: 1. System able to display the notification to Students

Figure 2.18: Use Case Description for <View Notification>

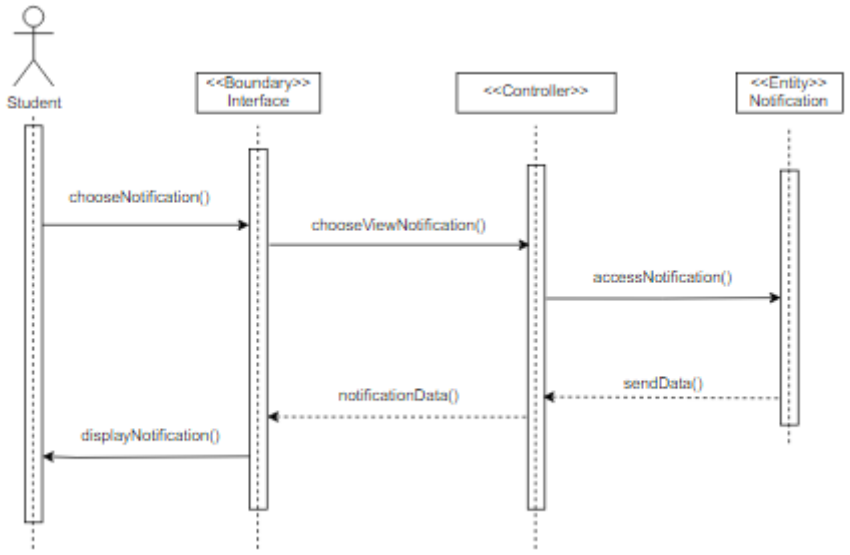

Figure 2.19: Sequence Diagram for <View Notification>

2.2.9 UC009: Use Case <Add Notification>

Use cases: Add Notification
ID: UC9
Actors: A3 Coordinator
Includes: UC7 Notification Data
Preconditions: 1. Coordinator is logged in into the system.
Flow of Events: 1. Coordinator selects “Add Notification” 2. include (Notification Data) 3. System will send a request to access the notification database 4. Coordinator add notification to the database.
Postconditions: 1. Coordinator able to add notification to the system for Student to view.

Figure 2.20: Use Case Description for <Add Notification>

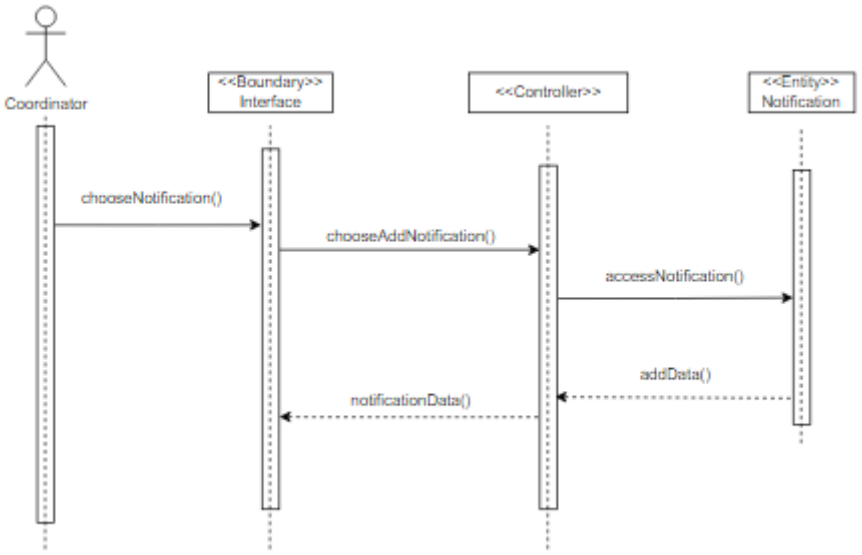

Figure 2.21: Sequence Diagram for <Add Notification>

2.2.10 UC010: Use Case <Update Notification>

Use cases: Update Notification
ID: UC10
Actors: A3 Coordinator
Includes: UC7 Notification Data
Extends: <returnMessage>
Preconditions: 1. Coordinator is logged in into the system.
Flow of Events: 1. Coordinator selects “Update Notification” 2. include (Notification Data) 3. System will send a request to access the notification database 4. While searching for notification 4.1. If notification data is found 4.1.1. Coordinator is able to update the notification 4.2 Else <returnMessage>
Postconditions: 1. Coordinator able to update the notification data

Figure 2.22: Use Case Description for <Update Notification>

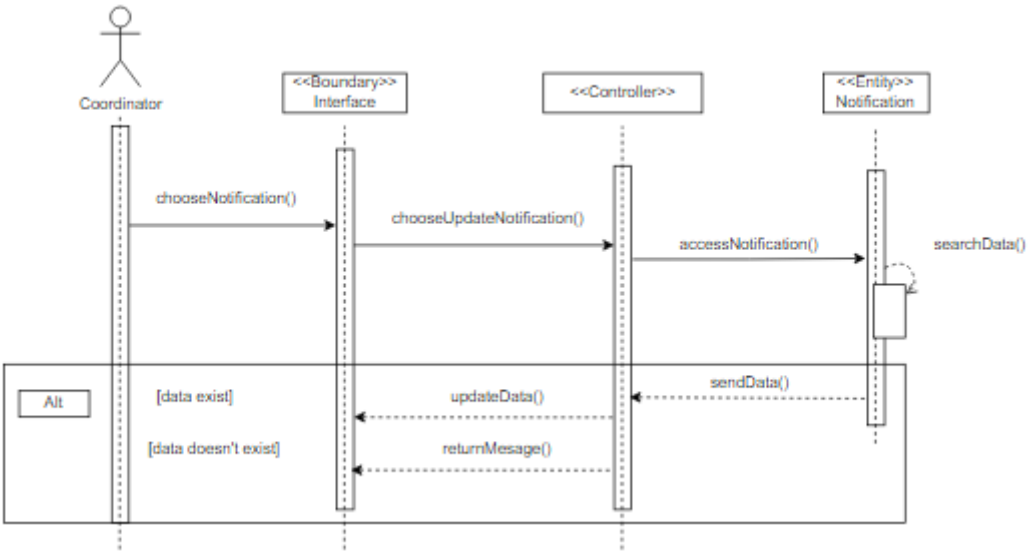

Figure 2.23: Sequence Diagram for <Update Notification>

2.2.11 UC011: Use Case <Delete Notification>

Use cases: Delete Notification
ID: UC11
Actors: A3 Coordinator
Includes: UC7 Notification Data
Extends: <returnMessage>
Preconditions: 1. Coordinator is logged in into the system.
Flow of Events: 1. Coordinator selects “Delete Notification” 2. include (Notification Data) 3. System will send a request to access the notification database 4. While searching for notification 4.1. If notification data is found 4.1.1. Coordinator is able to delete the notification 4.2 Else <returnMessage>
Postconditions: 1. Coordinator able to delete the notification data

Figure 2.24: Use Case Description for <Delete Notification>

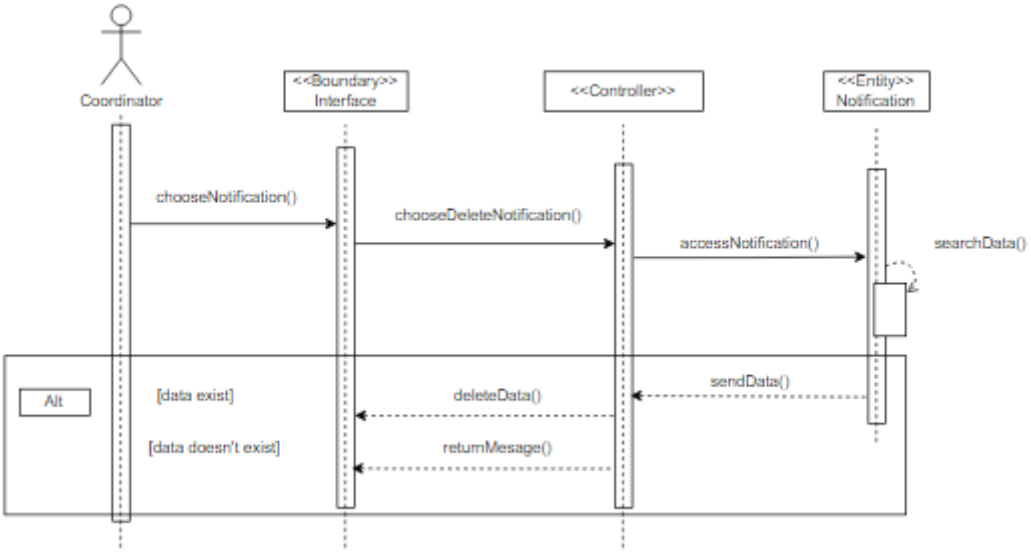

Figure 2.25: Sequence Diagram for <Delete Notification>

2.2.12 UC012: Use Case <Return Message>

Extension use cases: Return Message
ID: UC12
Extends: UC7 Notification Data at <returnMessage> UC10 Update Notification at <returnMessage> UC11 Delete Notification at <returnMessage>
Flow of Events: 1. System will print out a message “No notification data available”
Postconditions: 1. Users can add types of news that they want in their notification 2. Users can view the news that they had selected for their notification

Figure 2.26: Use Case Description for <Return Message>

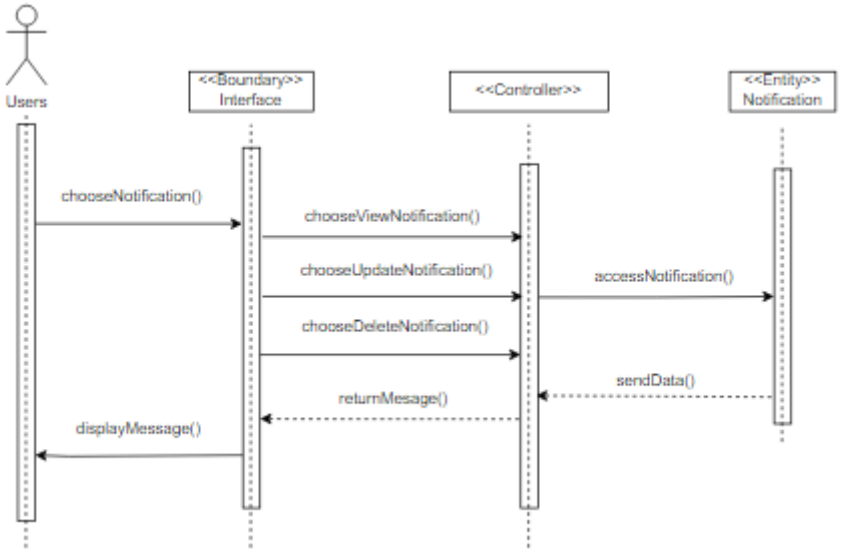

Figure 2.27: Sequence Diagram for <Return Message>

2.2.13 UC013: Use Case<Upload Question>

Use case: Upload Question
ID: UC13
Actors: A5 Examiner
Includes:
Extension Points:
Preconditions: 1. The examiner has logged in to the system.
Flow of Events: 1. The examiner clicks on the button that they want to upload the question. 2. If the question not uploaded yet 2.1 The file that contains the questions can be uploaded by clicking the upload button. 2.2 The answer for each of the question need to be set by clicking the right answer for the question. 2.3 The system stores the questions and answers 2.4 The exam is ready to be taken by the students 3. Else 3.1 The message about the question has been uploaded will be displayed. 3.2 The examiner will be redirected to main page.
Postconditions: 1. The question has been uploaded to the system.

Figure 2.28: Use Case Description for <Upload Question>

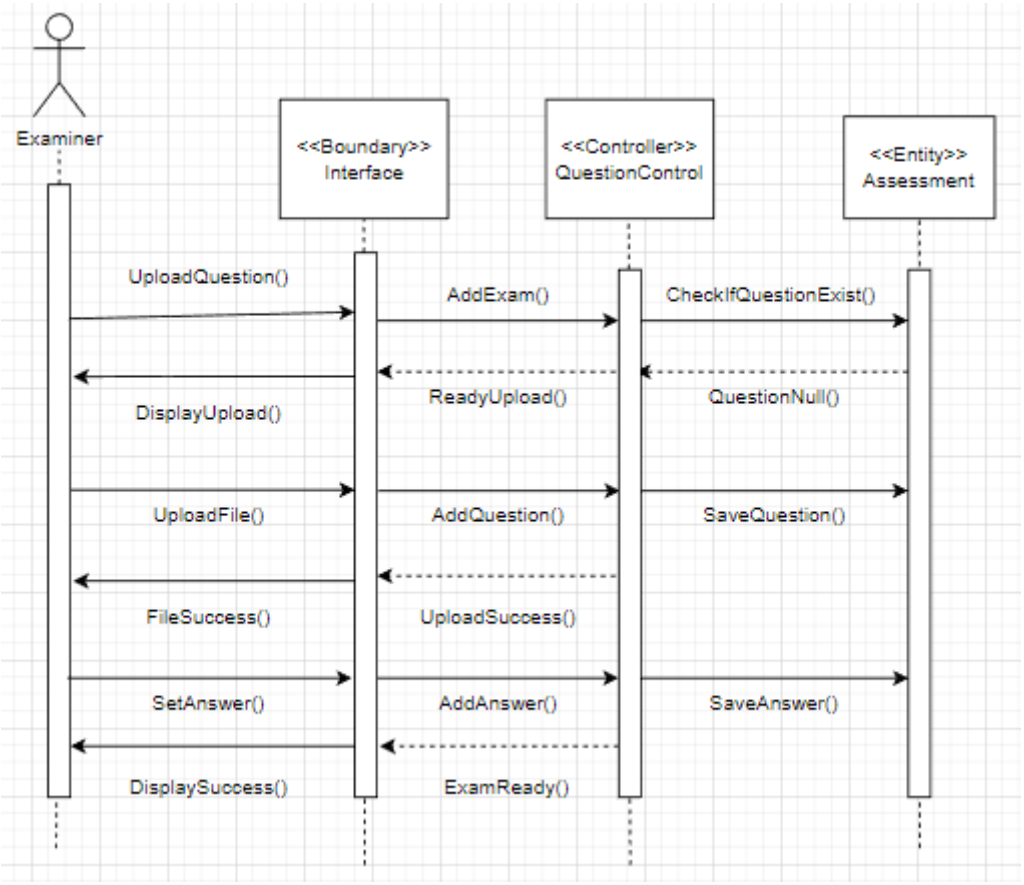

Figure 2.29: Sequence Diagram for <Upload Question> if Question Not Available

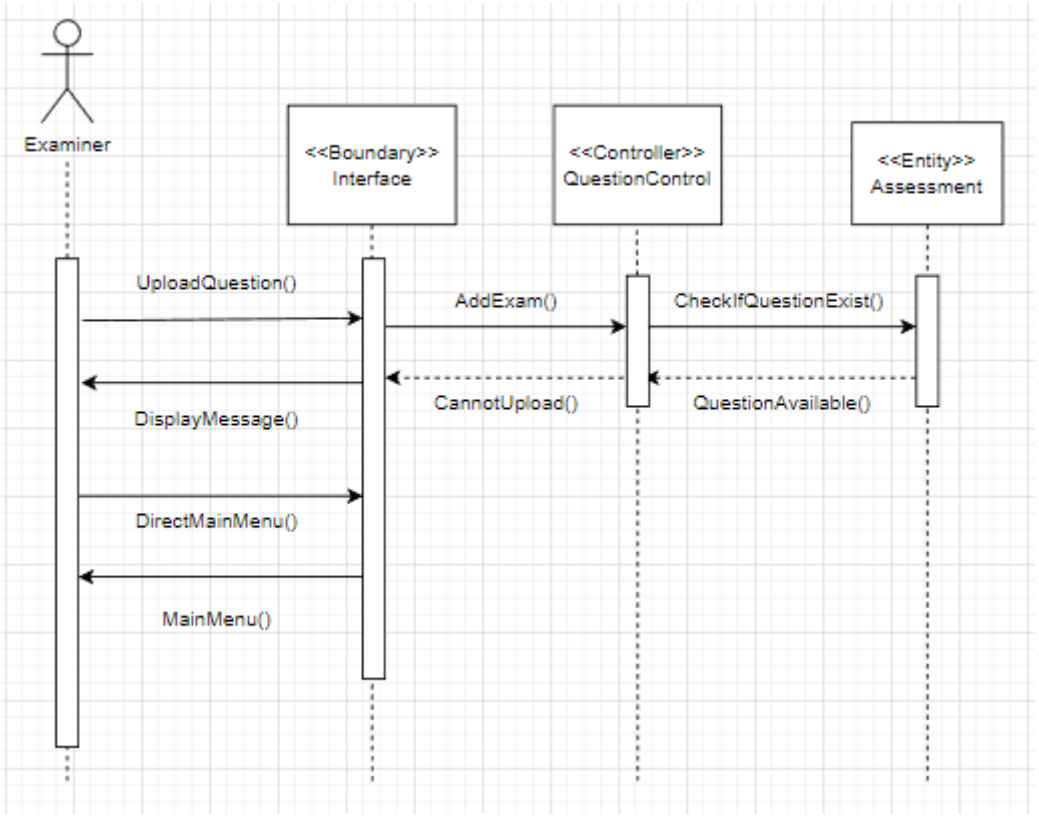

Figure 2.30: Sequence Diagram for <Upload Question> if Question Available

2.2.14 UC014: Use Case<Take Exam>

Use case: Take Exam
ID: UC14
Actors: A1 Student
Includes:
Extension Points: UC014 View Grades
Preconditions: 1. The student has logged in to the system. 2. The question has been uploaded on to the system.
Flow of Events: 1. The students click on the exam that they need to attend. 2. The student attempts the exam by clicking the start button. 3. All the questions are displayed and need to be answered by the student. 4. After all the question has been answered, the student can finish the student by clicking the finish button. 5. If the student has finished the exam 5.1 Extend View Grades
Postconditions: The student has finished the exam.

Figure 2.31: Use Case Description for <Take Exam>

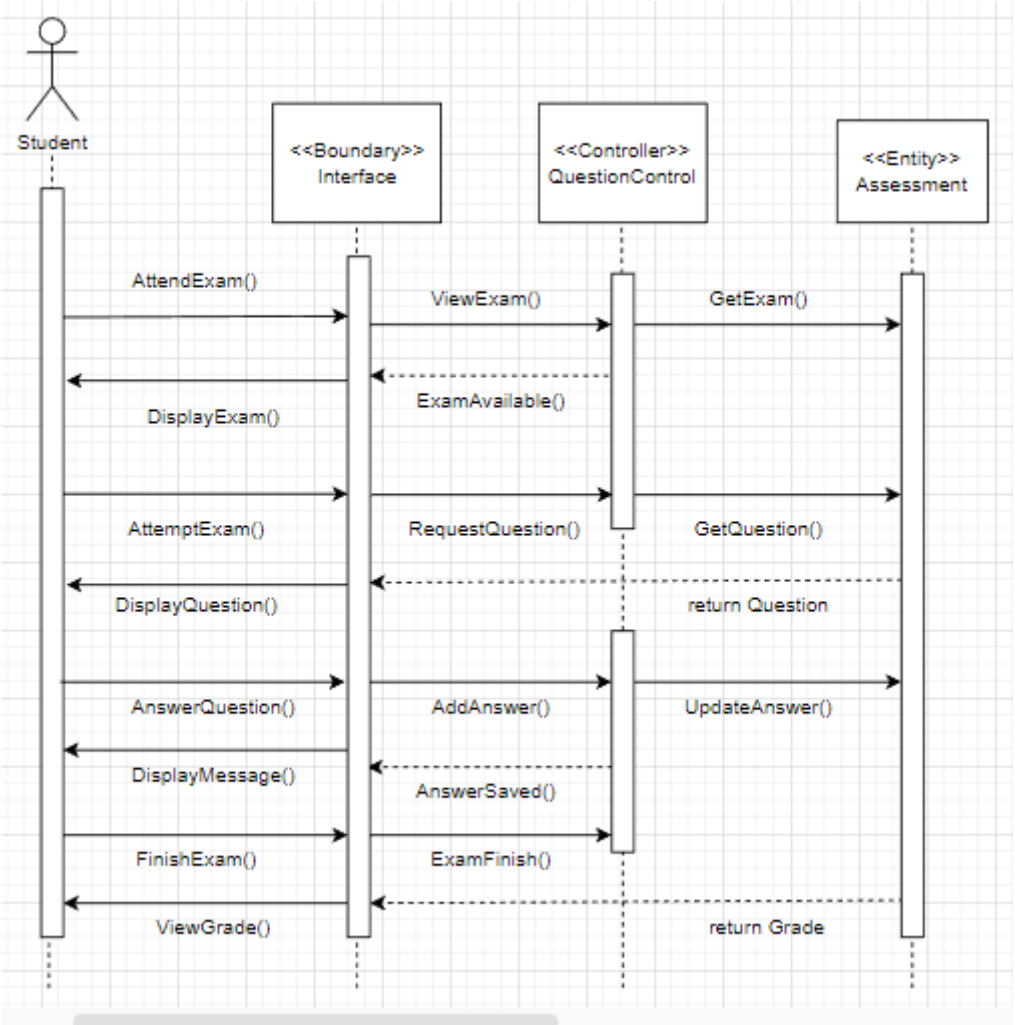

Figure 2.32: Sequence Diagram for <Take Exam>

2.2.15 UC015: Use Case<View Grades>

Use case: View Grades
ID: UC15
Actors: A1 Student A5 Examiner
Includes:
Extension Points:
Preconditions: 1. The student has logged in to the system. 2. The examiner has logged in to the system. 3. The student has finished their exams
Flow of Events: 1. The system will check the student's answer based on the answer that has been set by examiners. 2. The grades obtained after the checking has finished. 3. If the student wants to view grades 1.1 The grades displayed immediately after the exams. 4. If the examiner wants to view grades 2.1 A list of students that has finished the exam will be displayed. 2.2 The examiner can click on the student's name and their grades will be displayed.
Postconditions:

Figure 2.33: Use Case Description for <View Grades>

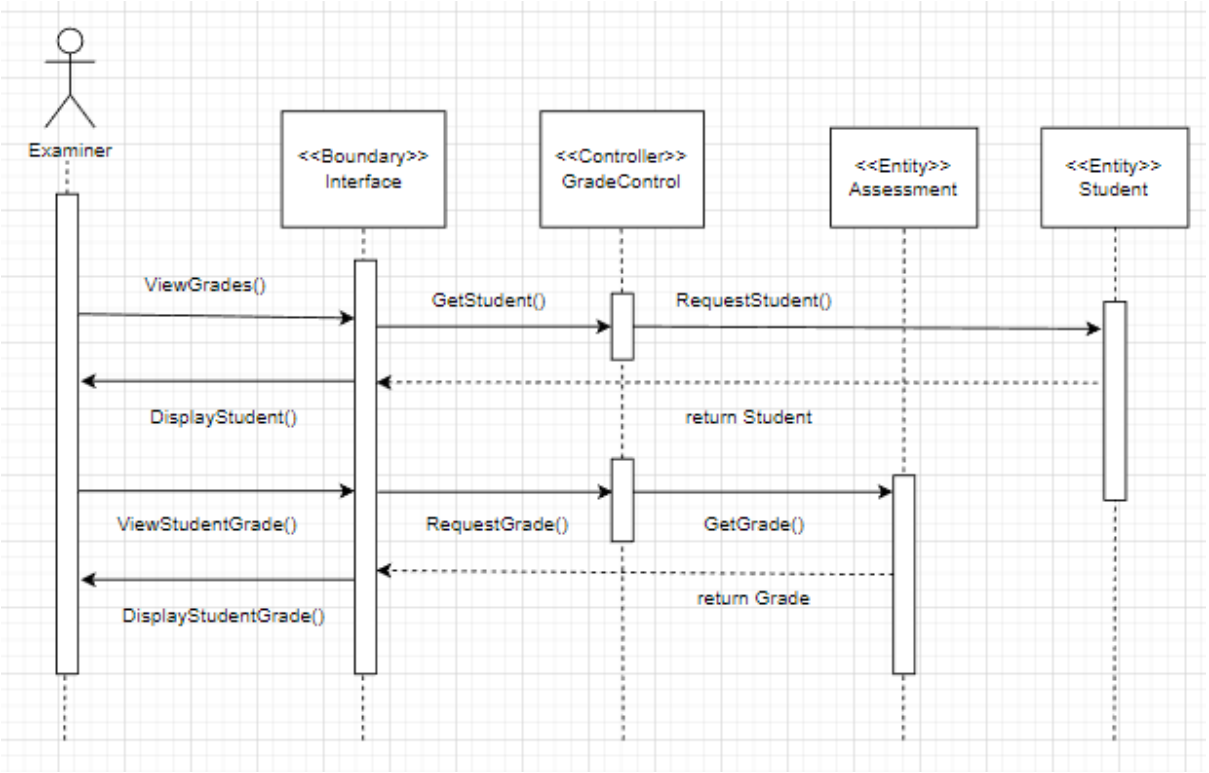

Figure 2.34: Sequence Diagram for <View Grades>

2.3 **Performance and Other Requirements**

- The **RESPONSE TIME** of the system should be only 1.0 second after the user clicks on the system.
- The **WORKLOAD** of the system should be able to handle more than 1000 students (including staff) and at least 200 users at one time during peak hours.
- The **THROUGHPUT** of the system should be around 10 operations per second. This means that the system can handle updating, adding, deleting and viewing content all at the same time.
- The **CAPACITY** of the system should be around 10 contents in a page with scrolling involved.

2.4 **Design Constraints**

- The system will be represented following the standards that are used on websites nowadays so the users can easily understand the system.
- The contents of the system shall follow the regulations by the Malaysia Government and Universiti Teknologi Malaysia (UTM).

- The language that will be used in the system is only limited to English and Bahasa Melayu.
- The system shall store the file related to the Final Year Project of 2u2i students that are currently doing their industrial training including their assessment.
- The file stored can only be accessed by those who have authorities.
- The website is constrained by the capacity of access that it can have at one time. Heavy load can cause the transfer of data to be slower and may cause server down.

2.5 **Software System Attributes**

- The system is available on any web browser
- Unique username and password registered for every user. They can use the username and password to log in to the system.
- Users can view, add, delete, edit messages to any users that are registered to the system.
- Students are able to view notifications that have been sent by the coordinator.
- The students are able to view their grades after they finished the exam.