UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SECJ2203: Software Engineering

System Documentation (SD)

SCSP 4112 – Initial Industry Project Proposal System

Version 1.0

22th June 2022

School of Computing, Faculty of Engineering

Prepared by: Blue Clues Team group 4

Name	Matric
Nor Fadli Bin Ahmad	A20EC0109
Muhammad Izwan Sazmin	A20EC0092
Azlinah Binti Herman	A20EC0018
Muhammad Nur Azizi	B20EC3011

Revision Page

a. **Overview**

In this current version of the system, documentation contained four main topics which are Specific Requirements (2), System Architectural Design (3), Detailed Description of Components (4), and Data Design (5). All of the main topics will be explained clearly and more details included diagram.

b. **Target Audience**

1. 4th year students of Data engineering
2. Coordinators of FYPi
3. University coaches
4. Industry coaches
5. Examiners

c. **Project Team Members**

Member Name	**Role**	**Task**	**Status**
Azlinah Herman	Project Manager	2.1.1 User Interfaces 2.2 System Features 2.2.1 UC001: Use Case Choose title & presentation date 2.2.2 UC002: Use Case Give Approval 4.1 Complete Package Diagram 4.2.2 P002: Evaluation Subsystem	Completed
Nor Fadli Bin Ahmad	UX designer	1.3 Definitions, Acronyms and Abbreviations 1.4 References 2.1.4 Communication Interfaces 2.2.5 UC005: Use Case Upload & update Form A 2.2.6 UC006: Use Case Give Comment Form A 2.5 Software System Attributes 4.2.1 P001: Proposal Submission Subsystem	Completed
Muhammad Izwan Sazmin	Software engineer	1.1 Purpose 1.5 Overview 2.1.3 Software Interfaces	Completed

		2.2.7 UC007: Use Case Submit Proposal 2.2.8 UC008: Use Case View Status 2.2.9 UC009: Use Case View Document 2.4 Design Constraints 4.2.3 P003: Appointment Subsystem	
Muhammad Nur Azizi	Data Analyst	1.2 Scope 2.1.2 Hardware Interfaces 2.2.3 UC003: Use Case Receive Evaluation 2.2.4 UC004: Use Case Give Evaluation 2.3 Performance and Other Requirements 3 System Architectural Design 5 Data Design	Completed

d. **Version Control History**

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Azlinah Binti Herman	Section 1 – 5	22/6/2022

Table of Contents

1. Introduction

1.1 Purpose

Our main goal is to create a system that would assist data engineering students in completing their final year assignment, which included several PSMi guidelines, for this project. We decided to create a filing system that focuses on SCSP 4112, Initial Industrial Project Proposal, which focuses on writing the project progress, delivering the written initial project proposal, identifying the problem background and proper solution, and identifying the project objectives and scope. We can see from these scopes that a suitable online filing management system is required to assist data engineering students in correctly managing their work without resorting to archaic ways that are time-consuming and exhausting. Students can easily keep track of their work with the help of an online filing management system because the system can notify them of work that needs to be done, manage the file according to their needs, and submit the work online rather than handing it over to their lecturer's office. This method can also help students save money by reducing the number of papers they need to write their assignments plus we must conserve the environment by minimizing the amount of paper we consume.

First and foremost, students must register their credentials to use our team's online filing management system. The user must provide their full name, email address, matric numbers, and a secure password. After that, the user may log into the system and access the main menu. Students may pick from a variety of alternatives from the main menu, including reporting their project progress, submitting a project proposal, receiving comments from lecturers, and keeping track of their work. Each choice has its own set of features that might assist students in achieving their objectives. Our goal is also to assist instructors in tracking students' progress and simplify the grading procedure. This file system may alert lecturers to any project-related issues, such as students failing to submit work or sending it late, and it can also assist lecturers in organizing students' work according to their needs.

1.2 Scope

The system is an Final Year Project Management System.. The system will be capable of handling and managing lecturer-related tasks, such as student registration, submission of initial final year project materials (documents and photographs), and approval process processes. The system will not automate the full process of the final year project; rather, it will assist users in automating the majority of the operations in order to eliminate human error during manual jobs (incomplete submissions, etc.). The technology will be used to submit initial final year projects and for lecturers to approve them, replacing the current, manual procedure . There are various things that we must have in order to develop this system before we can deploy it and make it available to the public. The following is a list of the technologies that will be used to develop the system

1.1.2 Hardware Requirements

I. We chose AWS as our server because of its top one security matter. The security of the FYP platform is extremely crucial because there is a large amount of private and secret information that needs to be protected. . There are dozens of compliance processes built into AWS for its infrastructure, and they are provided with each service. Moreover, the services include extensive security and support networks aimed at identifying questionable activity on accounts and thereby revealing system weaknesses. AWS is also known for its stability and consistency. When compared to other cloud service providers, it has far less downtime. System availability is critical since both the lecturer and the student need to access the system all of the time in order to complete the project. How will students and lecturers complete their work if the system run under bad server and down all of the time? this thing need to be consider

II. Since FYP platform have thousand of database need to be store and database need to be maintain ,the best hardware for database already chosen.ORACLE was chosen as our database management system. Oracle database offerings include high-performance databases, maintainable database systems, no-SQL and MySQL databases, and self-contained databases. Oracle also offers cloud-based database services, allowing for implementation in our data centre when residency or latency are necessary, or in the cloud when scalability and capabilities are desired.

1.1.2 Software requirement

I. **PHP** - PHP will be use for backend(system) .Main reason PHP language is chosen for backend is PHP solution can work on various operating systems such as Windows, Linux, Mac, Unix . This programming language provides compatibility with AWS server.

II. **HTML** - HTML is used for frontend development. It helps with the building of websites. It has an effect on practically every website's front-end programming. HTML describes the manner in which content is presented in a browser.The important thing is HTML language can perfectly work along with PHP

III. **JavaScript** – This language is use for fontend (GUI) and adding interactive behavior to web pages for example show or hide more information with the click of a button, displaying animations, playing audio and video in a web page, Displaying a timer or count-down on a website and etc.

IV. **Web Browser –** Web browser such as Google chrome . Mozilla Firefox , Opera ,Microsoft Edge etc .This software is help the user which is lecturer and student to browse our platoform .

1.1.2 Modules and Activities

Module	Description
Login module	This module assists the user who enters the system in authenticating themselves before they may access the platform and execute any action such as posting or reviewing.Pupose of the module is to give security to the system and only authorize person can access the platform
Registration module	The student will be registered in this subject as a final year project student. It will form a part of a structure in which all of the student's information will be included, such as the title of the project, the subject of the project, and the supervisor.
Submission module	This section requires students to submit the initial draft of their final project. This module serves as a reminder to students to submit their work in the form of a PDF file. If a student uploads anything other than a PDF file, the system will immediately reject it and prompt the student to re-upload.
Approval module	Following the submission of a student's initial draft of their final year project, the lecturer will download and review the file. After reviewing the paper, if the lecturer believes the project is appropriate, he or she must offer approval. If the lecturer rejects, the system will seek comment from the lecturer and forward it to the student. The system will tell students that their submission has been rejected and they'll be able to view the reason for rejection. The system will prompt the student to re-upload. Following that, the same process will be repeated until the student is approved.

1.3 Definitions, Acronyms and Abbreviation

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

1. IC – Industry Coach
2. UC – University Coach
3. SD – System Documentation

1.4 References

Lucidchart. (n.d.). *UML Use Case Diagram Tutorial*. https://www.lucidchart.com/pages/uml-use-case-diagram

SADOS. (2022, March 7). *The 5 Benefits of AWS Cloud Servers (And the 3 Drawbacks).* https://sados.com/blog/aws-benefits-and-drawbacks/

Singh, H. (2022, May 26). *Oracle Database Advantages, Disadvantages and Features [Guide 2022]*. The NineHertz: https://theninehertz.com/blog/advantages-of-using-oracle-database

1.5 Overview

There are five sections in this paper. The aim of this paper, the scope of this project, and the overview are all briefly explained in the introductory section. The specifics for particular requirements, system architecture design, component detail descriptions, and data design are then listed. We include all of the key elements, including the logical properties of each interaction between the software product and its consumers, in the particular needs section. This system is aimed for fourth-year data engineering students, but it will also be used by university coaches, industry coaches, examiners, and coordinators. The design limitation section delves further into the system's design limits. The software system characteristics specify the functional specifications of the system that are required by particular users. Other criteria for customers/users or developers include quality qualities or non-functional needs. On the other hand, using class diagrams, pseudocodes, and sequence diagrams, this article will cover the usage of the proposal submission, evaluation, and appointment subsystems.

2. Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

a) The logical characteristics of each interface between the software product and its users.

required screen formats	can be used on multiple devices of different sizes such as desktops, laptops, tablets, or mobile phones.
page or window layouts	The layout for desktops and laptops will be landscapes while using tablets and mobile phones can be portrait or landscapes depending on the autorotate screen of the devices.
content of any reports or menus	The home page will have several menus depending on the user. Student: choose title and presentation date, Submit a proposal, Submit Form A, view status, and user profile. Industry and university coach, examiner: approval and evaluation Coordinator: view documents
availability of programmable function keys	Programmable function keys will be the same as another normal website. For example, the user can press “Enter” on the keyboard as a submission button, “Backspace” for the delete button, and usual keyboard shortcuts such as “Ctrl + C” for copy and “Ctrl + V” for paste function.

b) All the aspects of **optimizing the interface** with the person who must use the system.

Software System Attributes	**Do’s**	**Don’ts**
Usability	The interface will be simple, easy to use, user-friendly, and easy to learn by users.	A high learning curve and complicated navigations.
reliability	Every button and feature in the interface will give the corresponding function	System crash when high traffic

	independent of the environment and conditions	
maintainability	Easy to upgrade for a new feature in the future and low-cost	Impossible to update
efficiency	Can support high capacity of users at the same time and fast response time	Slow response time and cannot support all the users that are trying to access the website at the same time
security	Only registered users can access the system, high security for privacy and data loss	An unauthorized users can access the database

There are five different users in this system which are 4th year students of data engineering courses, university coaches, industry coaches, examiners, and coordinators. Below will describe the interface of each user.

4th year students of data engineering courses

Students must log in to the system and then the home page will appear. There will be five menu on the home page which are chosen title and presentation date, Submit a proposal, Submit Form A, status, and user profile. When the user clicks the menu, it will go to the corresponding function. When the user needs to upload documents such as the proposal or Form A, there will be confirmation text that the user successfully uploads the documents. The user also can view, undo and reupload the documents back to the system. There will be a notification when their university coach, industry coach, and examiners give approval and evaluation to them. Status menu is to view the status of the proposal and the user profile consists of student's data.

University coach, industry coach and examiner

These three users have a similar interface just a few different features. The user needs to log in to the system and the home page will appear. There will be two menu on the home page which are the approval and evaluation. When the user press the approval button, the list of title and presentation date of the student will appear. The approval button on that page is to approve the title and the presentation date of the student. University and Industry coaches can give approval to both title and date while examiners can only approve the presentation date. The evaluation page is to evaluate the proposal and presentation of the student. Only the examiners can give comments to Form A.

Coordinator

The user needs to log in to the system and the home page will appear. The home page will have a list of students' data with their related documents such as the proposal and Form A. The coordinator can view the contents of the documents and the status of the documents whether they have been approved, still in progress, or rejected.

2.1.2 Hardware Interfaces

These days, there are a variety of devices, including phones, tablets, laptops, and desktop computers, that are capable of opening websites. Users can use the Initial Industry Project Proposal System through any device of their choosing. However, they are required to be linked to the internet through WIFI, Ethernet Cross Cable, or mobile data. Examples of such devices are a phone, tablet, laptop, and computer b. Due to the adaptability of our framework, this system is compatible with a wide variety of operating systems, including Windows, Linux, OS X, iOS, and android. Users have the option of downloading the report in PDF format and printing it out on paper when it comes to the matter of the report.

2.1.3 Software Interfaces

No	Name	Mnemonic	Specification number / Version Number	Source	Discussion
1	Google Chrome		**Windows, macOS, Linux:** 102.0.5005.63 **iOS:** 102.0.5005.78 **Android:** 102.0.5005.87	www.google.com	Use as the platform to display the website of our system
2	Python		**Windows, macOS, Linux:** 2022.1.2	jetbrains.com/ pycharm/download /#section=windows	Use to code and develop the system
3	HTML		HTML5	https://code.visualstudio.com/ download	Use to code and develop the system on the website
4	JavaScript		Version 23	https://code.visualstudio.com/ download	Use to create dynamic and interactive web content

5	Microsoft SQL Server	MySQL	15.0.2000.5	https://dev.mysql.com/downloads/mysql/	To configure and update the database

2.1.4 Communication Interfaces

This is one of the most important components of the system. However, the key to this method is to make communication for the student, University coach, Industrial coach, Examiner and Coordinator simpler and more understandable. In order to have a seamless, simple and user-friendly, the following functionalities should be included in the interfaces: -

- The communication interfaces should be able for user to communicate each other through online.
- The common question (FAQ) should be put in the system in order to ease user to have quick answer for their questions.
- The communication interfaces for Coordinator, Industrial Coach, University coach and Examiner will have more option so that they can have almost overall control and future system updates.
- All users will be able to send their complaints via the online application for the convenience for themselves and for the future.

2.2 System Features

The system features include:

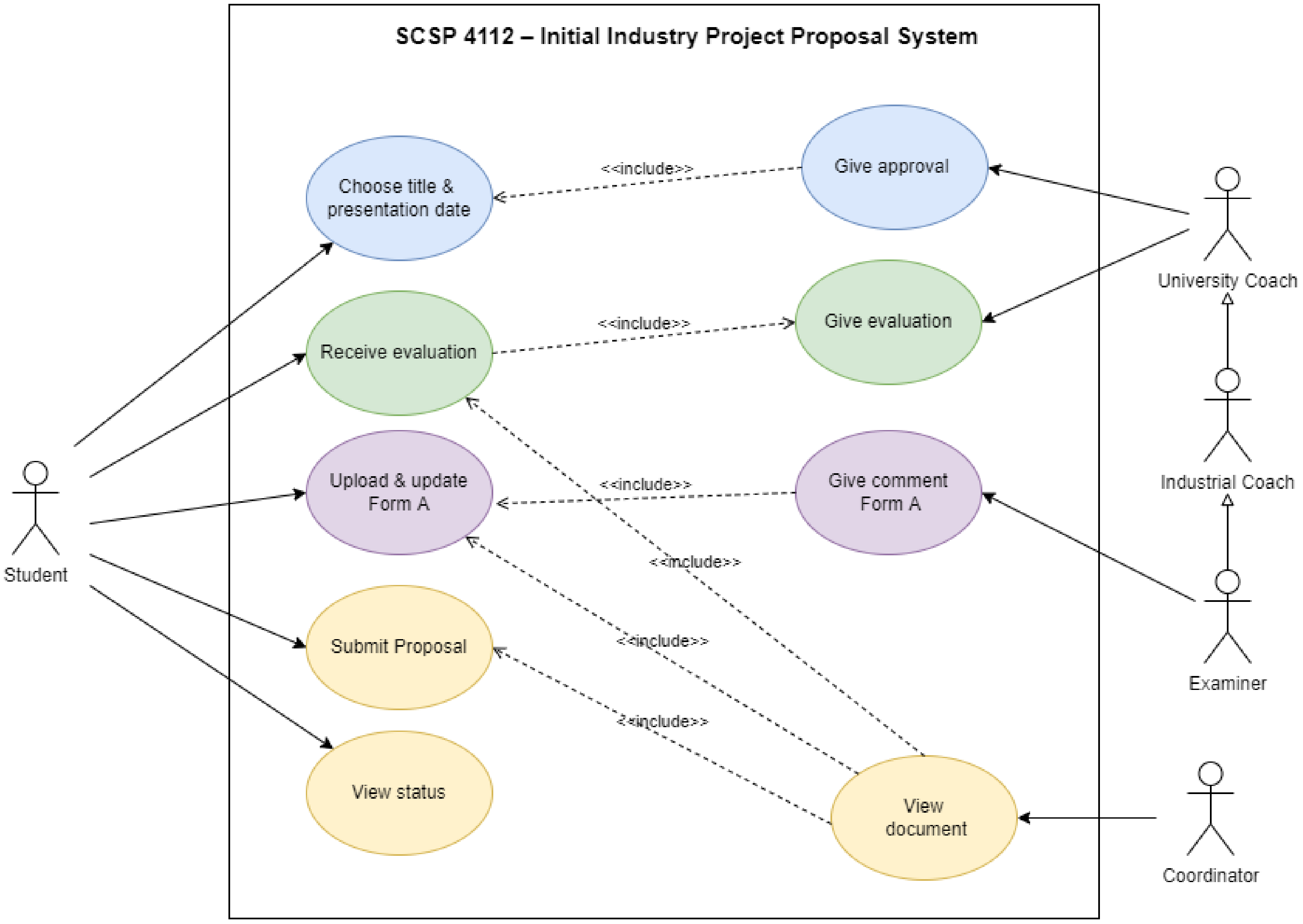


Figure 2.1: Use Case Diagram for SCSP 4112 – Initial Industry Project Proposal System

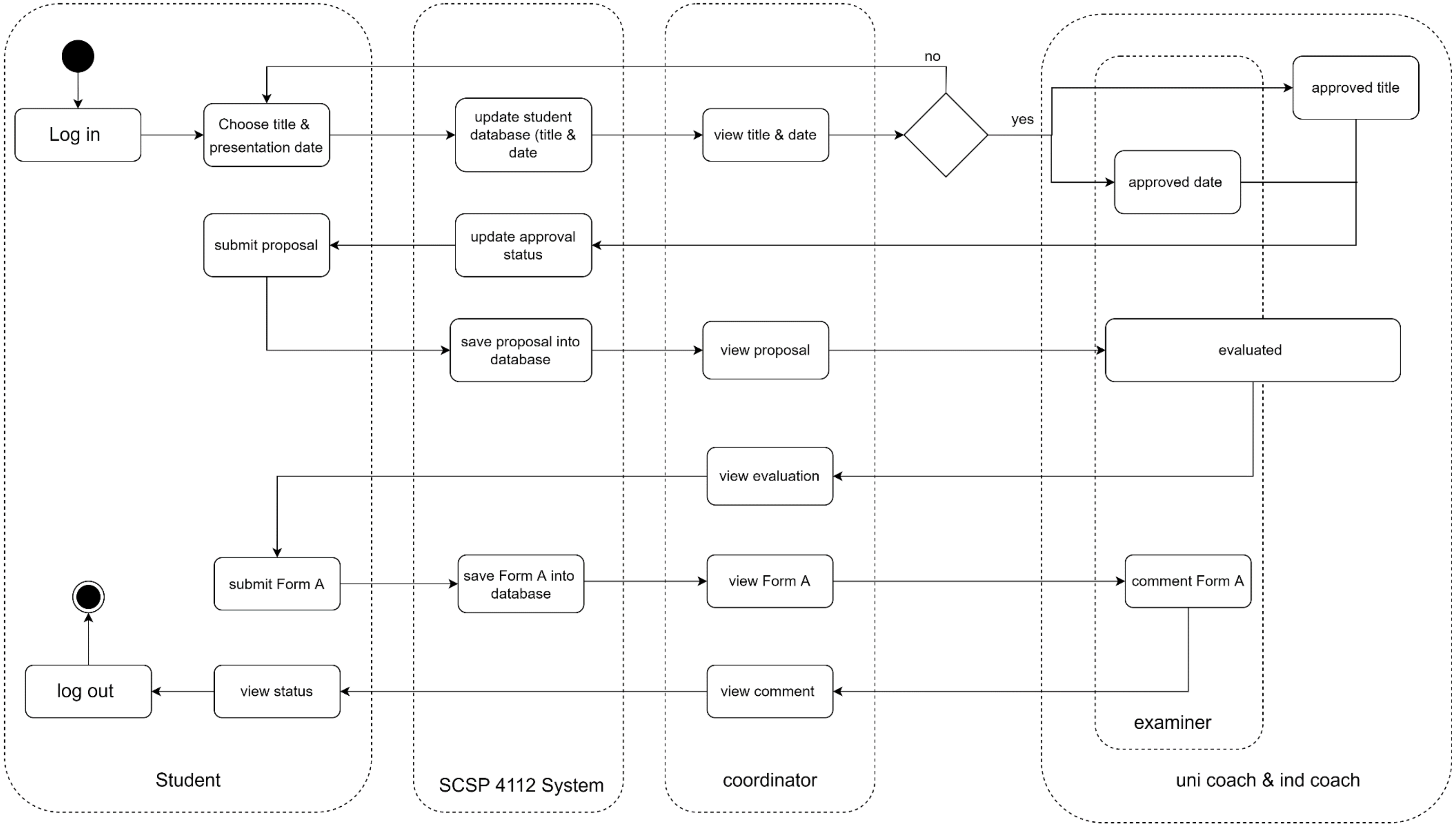


Figure 2.2: Activity Diagram for SCSP 4112 – Initial Industry Project Proposal System

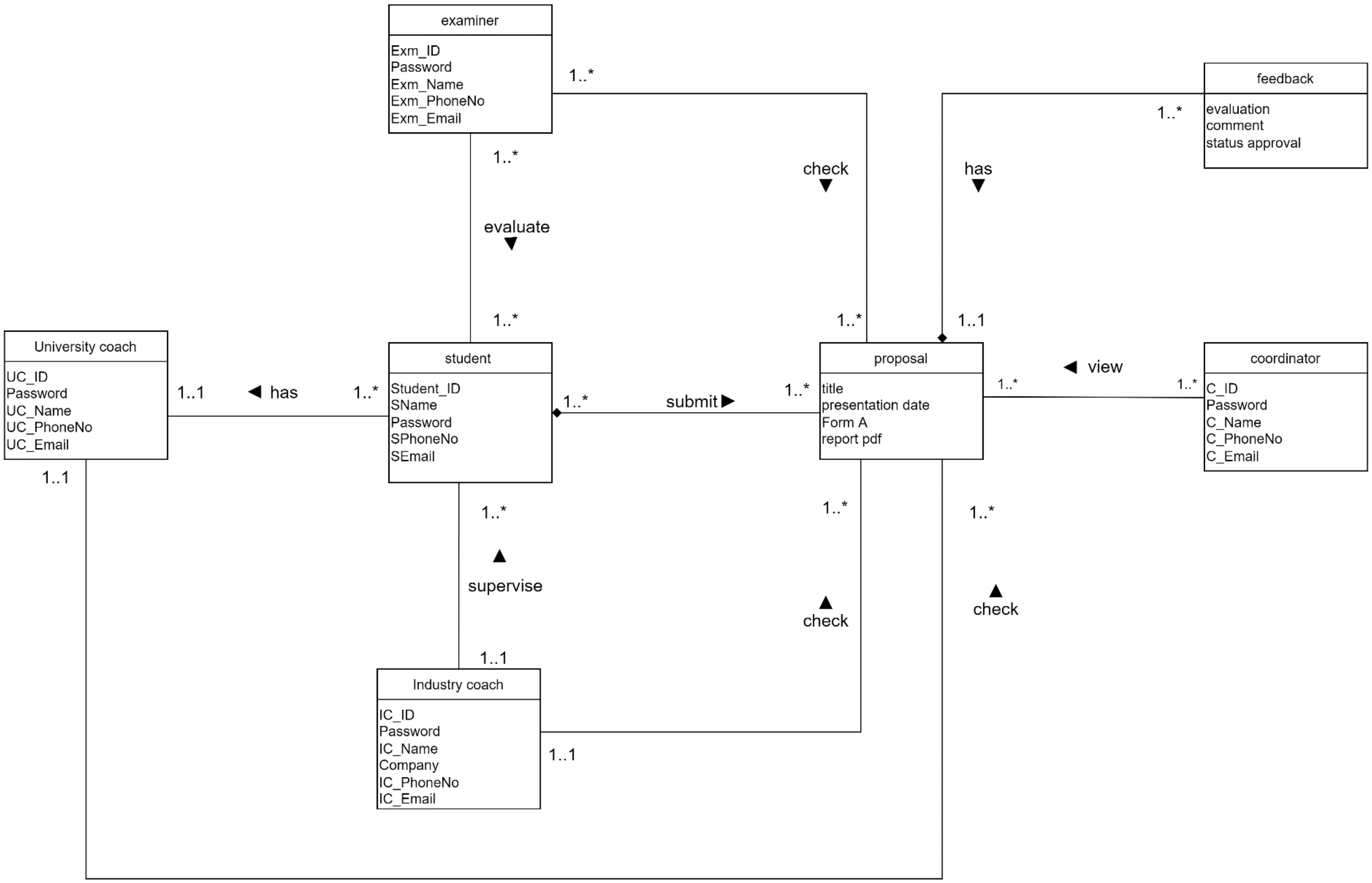


Figure 2.3: Domain Model for SCSP 4112 – Initial Industry Project Proposal System

2.2.1 UC001: Use Case Choose title & presentation date

Table 2.1: Use Case Description for Choose title & presentation date

Use case: Choose title & presentation date
ID: UC001
Actors: Student
Preconditions: - Log in to the system
Flow of events: 1. Chose the menu for “Choose title & presentation date” in the home page 2. Fill in the chosen title for the proposal of the Final Year Project intern (Fypi) 3. Press submit button 4. Choose the available date in the calendar for the presentation date 5. Press submit button 6. Confirmation of the title and presentation date that has been chosen 7. Use case end
Postconditions: Title and presentation date for student exist in database
Alternative flow *n*: No alternative flow
Postconditions: No postconditions for alternative flow
Exception flow (if any): 1. chosen title and presentation date cannot be submitted 1.1 System will prompt the message “data cannot be submitted. Draft saved”

Figure 2.4: Sequence Diagram for Choose title & presentation date

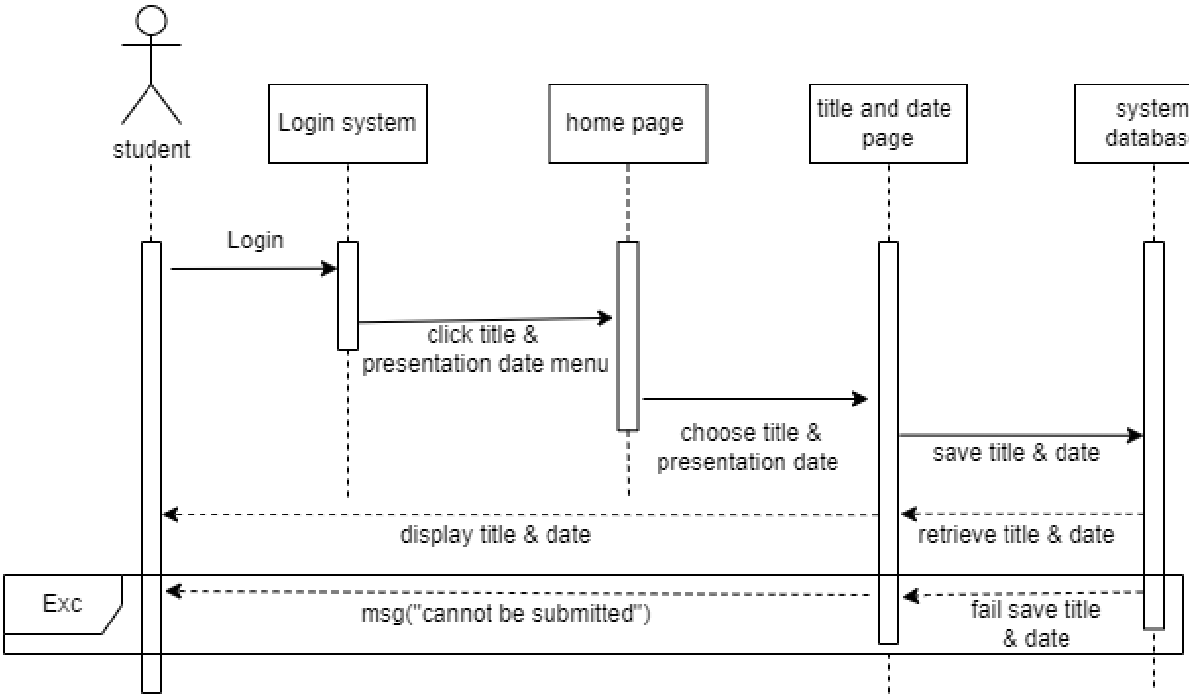
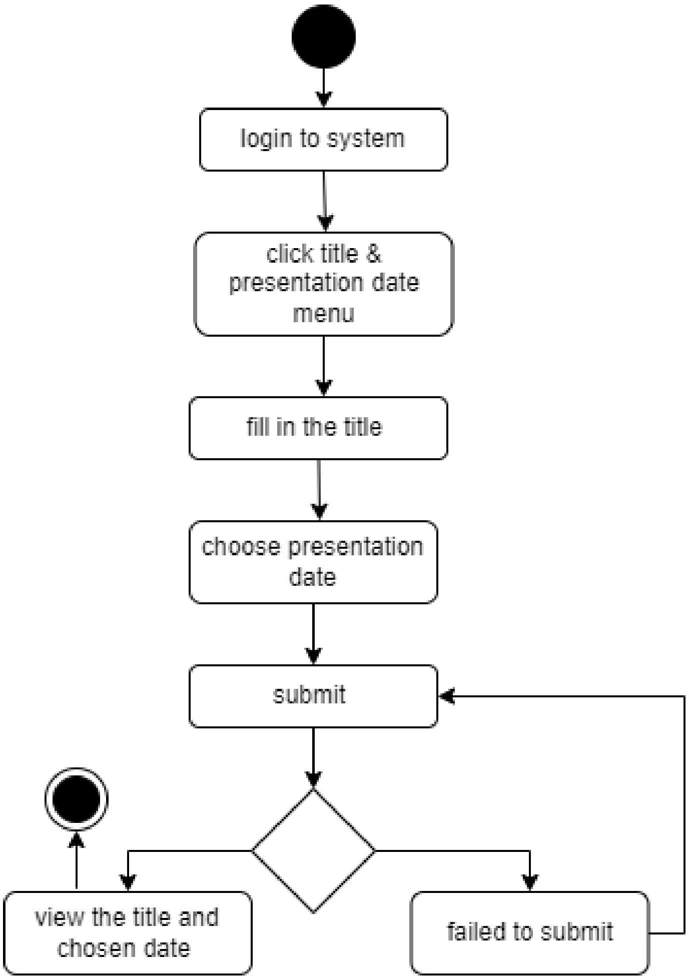


Figure 2.5: Activity Diagram for Choose title & presentation date



2.2.2 UC002: Use Case Give Approval

Table 2.2: Use Case Description for Give Approval

Use case: Give Approval
ID: UC002
Actors: Uni coach, ind coach, examiner
Preconditions: - Log in to the system - Chosen title and presentation date of selected student exist in database
Flow of events: 1. Choose the “Approval” button on the home page 2. System will redirect the user to the page that contains a list of titles and dates of presentation of specific students 3. User approved the title and presentation date 4. Use case end
Postconditions: Students will get notifications that their chosen title and presentation are approved
Alternative flow *n*: 1. Logged in as examiners 1.1 Choose the “Approval” button on the home page 1.2 System will redirect the user to the page that contains a list of presentation dates. 1.3 User approved the presentation date 1.4 Resume to normal flow 4
Postconditions: Students will get notifications that their chosen presentations are approved
Exception flow (if any): 1. The title and presentation dates are rejected

Figure 2.6: Sequence Diagram for Give Approval

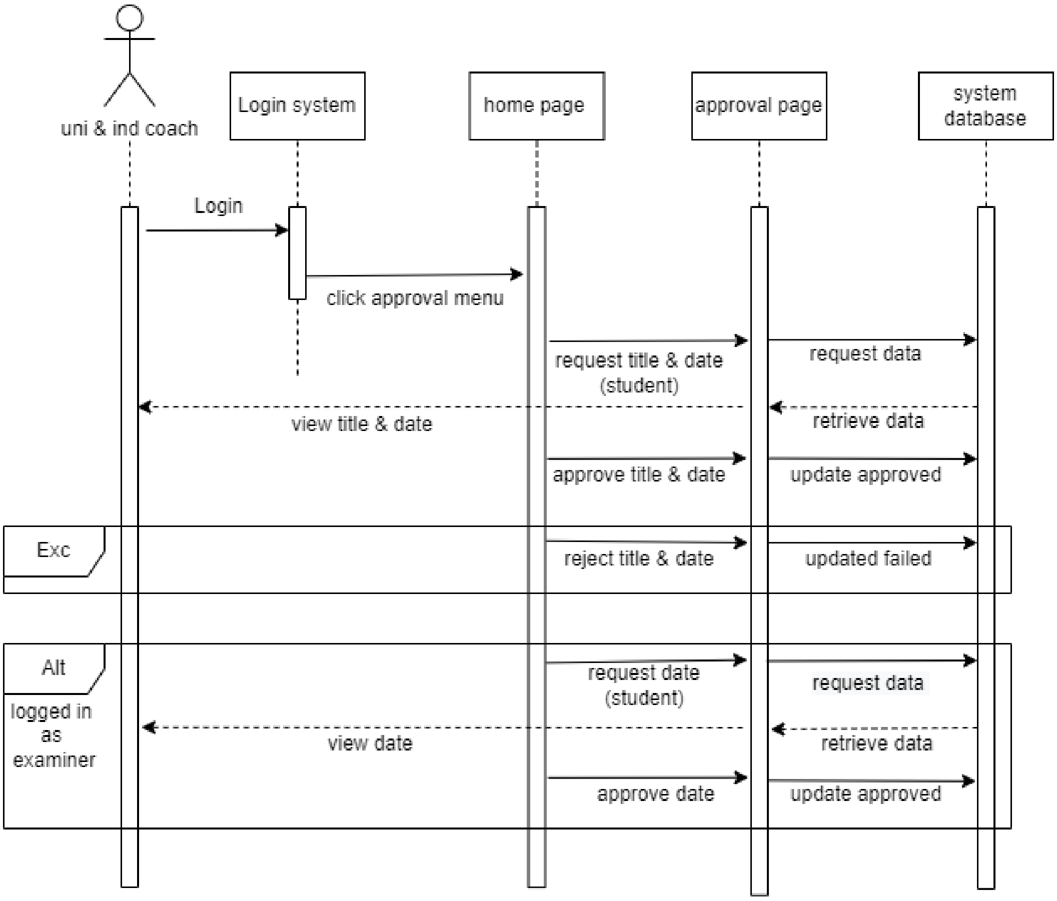
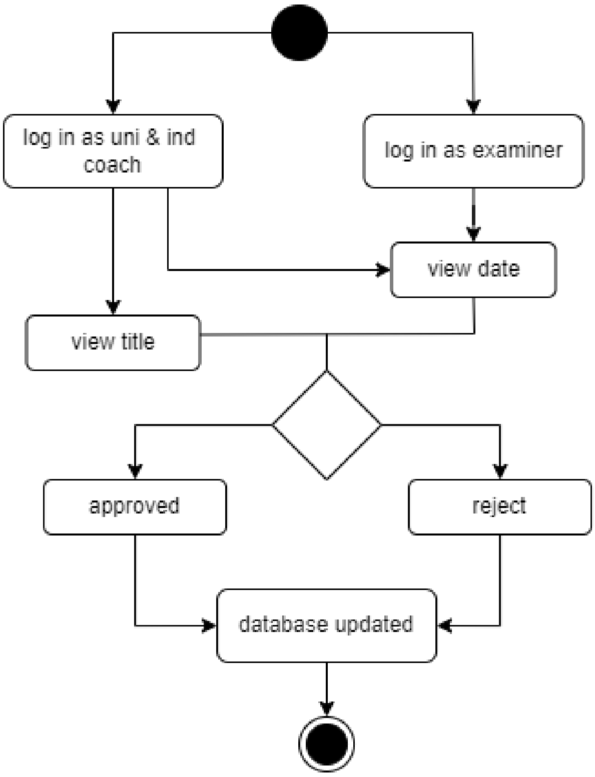


Figure 2.7: Activity Diagram for Give Approval



2.2.3 UC003: Use Case Receive Evaluation

Table 2.3: Use Case Description for Receive Evaluation

Use case: Receive Evaluation
ID: UC003
Actors: Student
Preconditions: 1. University coach must give evaluation on student's submission first before student can receive the evaluation. 2. The student must be a valid user logged on to the system
Flow of events: 1. Student open evaluation section on the system. 2. System notified that student proposal has been evaluated. 3. Student open evaluation form on the system. 4. System will check the evaluation data and fetch from Evaluation Form Database. 5. System found evaluation form data and display to student 6. Use case end.
Postconditions: 1.Evaluation form viewed
Alternative flow *1*: 1. The system should offer the student the option to download the evaluation form as a PDF for future reference.
Postconditions: 1.Evaluation form viewed and downloaded.
Exception flow (if any): 1.System will give message to student if there is no data found on the database.

Figure 2.8: Sequence Diagram for Receive Evaluation

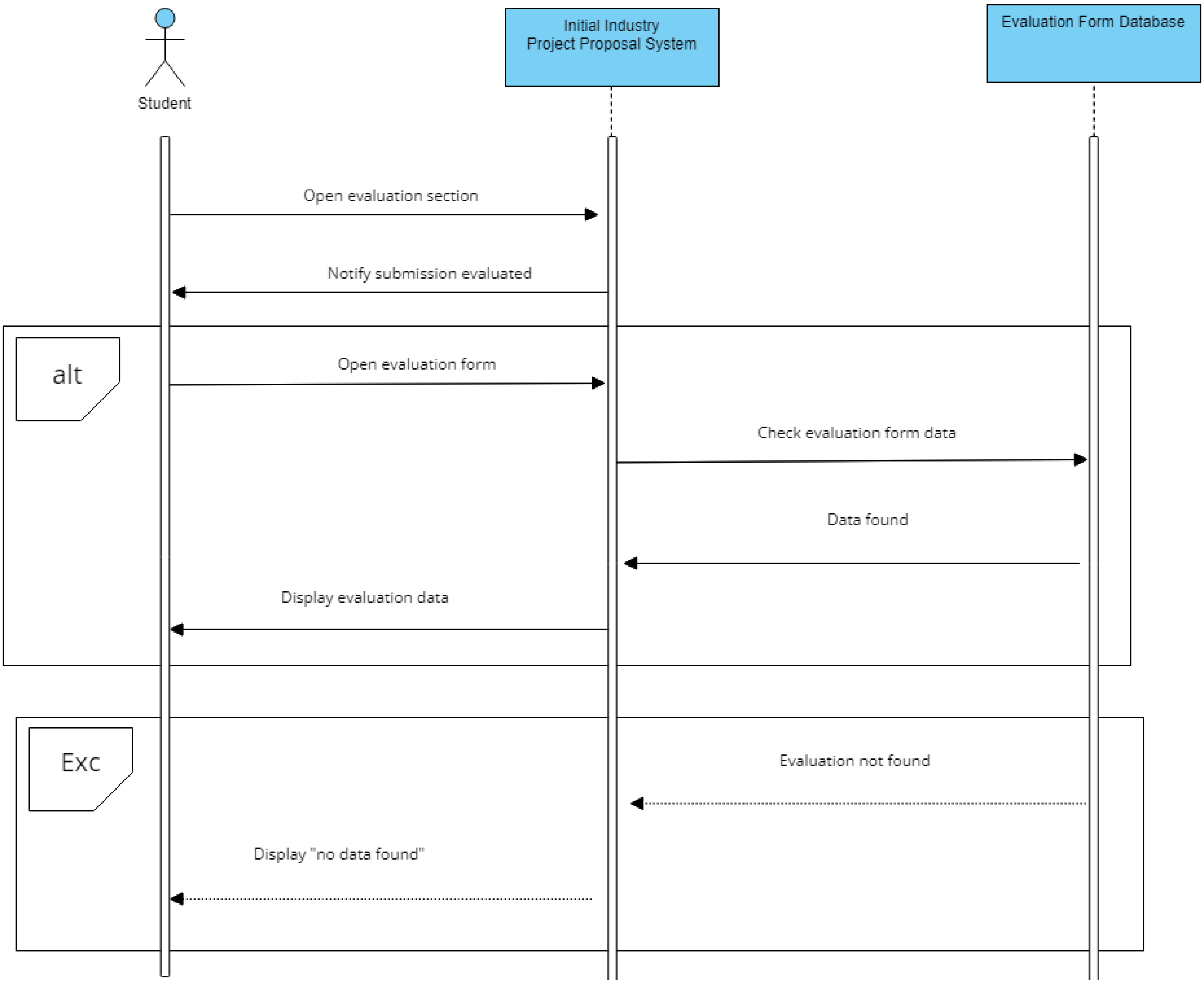
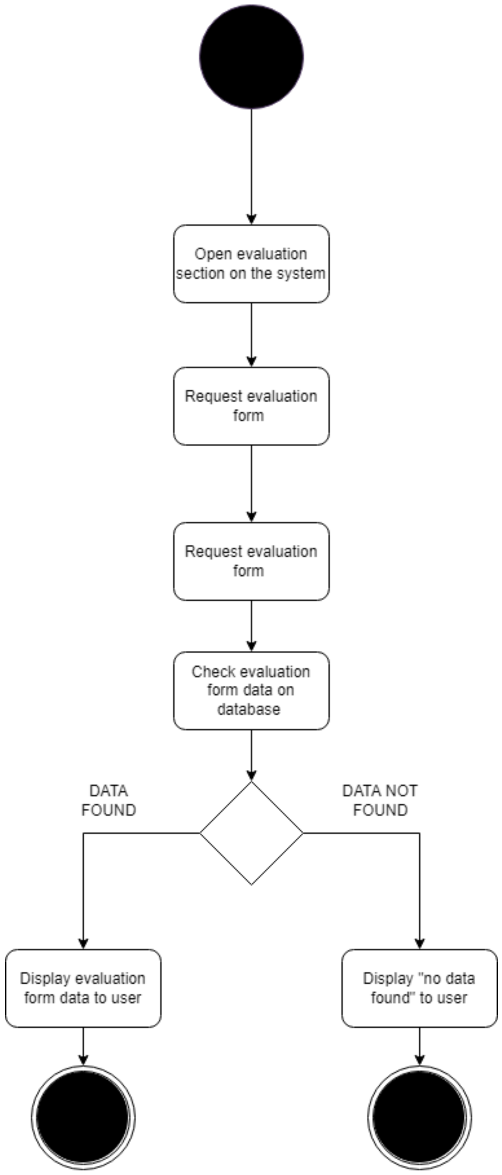


Figure 2.9: Activity Diagram for Receive Evaluation


Open evaluation section on the system
Request evaluation form
Request evaluation form
Check evaluation form data on database
DATA FOUND
DATA NOT FOUND
Display evaluation form data to user
Display "no data found" to user

2.2.4 UC004: Use Case Give Evaluation

Table 2.4: Use Case Description for Give Evaluation

Use case: Give Evaluation
ID: UC004
Actors: University coach and Industrial coach
Preconditions: 1. Before a coach may evaluate a student, the student is required to submit their submission. 2. The coach must be a valid user logged on to the system
Flow of events: 1. University coach and Industrial coach open evaluation section on the system. 2. Coach choose which proposal they want to evaluate. 3. Coach fill in mark in evaluation form and submit. 4. System will notify coach which proposal has evaluated. 5. Use case end.
Postconditions: 1.Proposal evaluated.
Exception flow (if any): 1.System will give message to coach if there is no proposal data found on the database.

Figure 2.10: Sequence Diagram for Give Evaluation

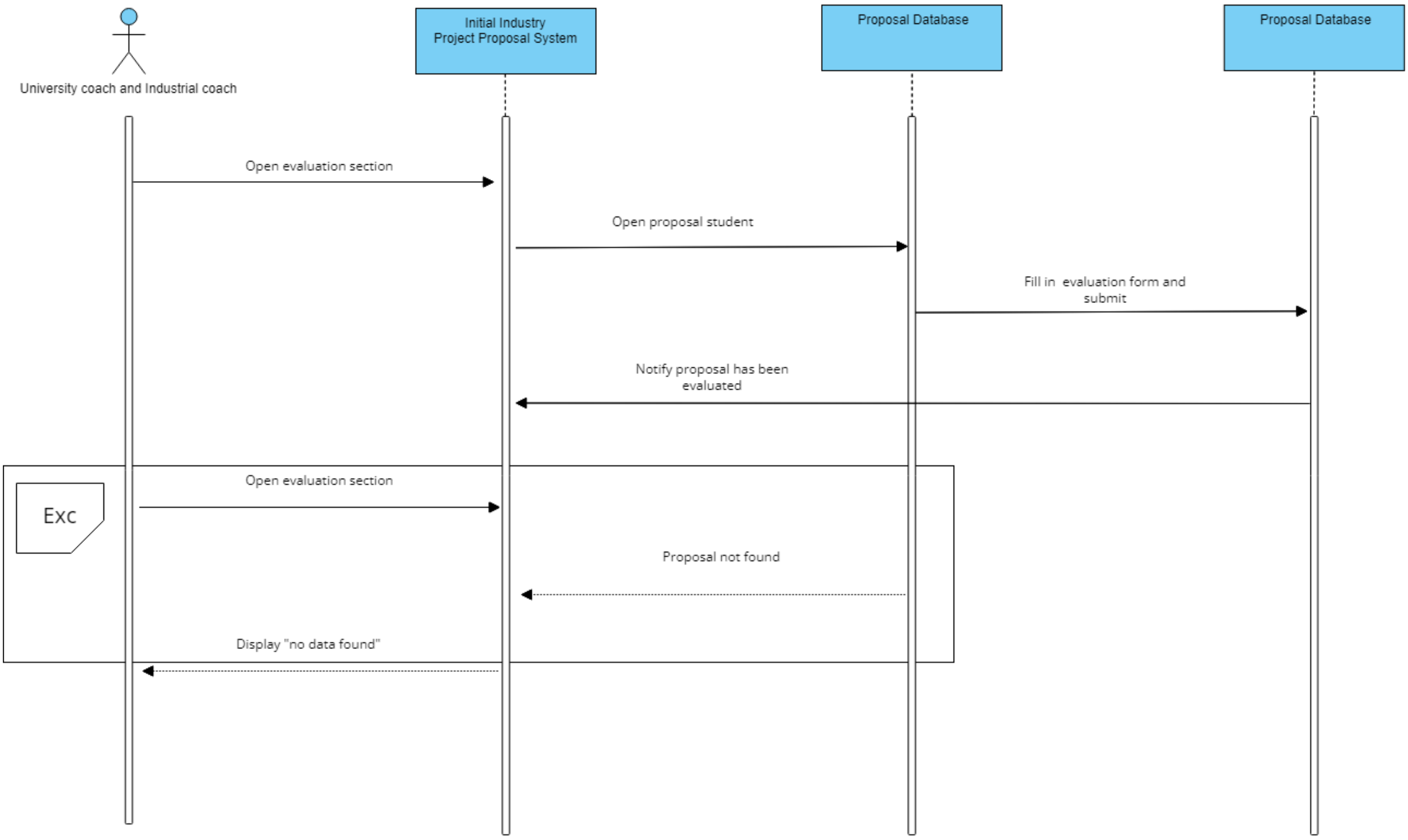
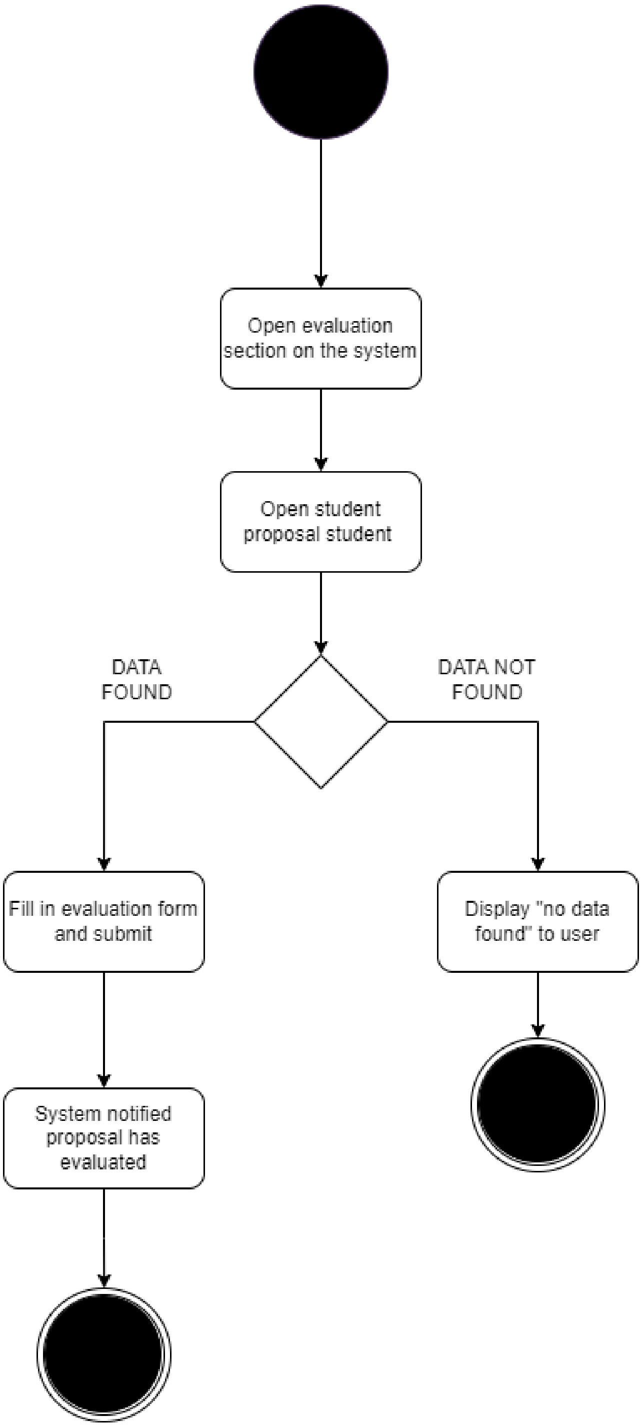

University coach and Industrial coach
Initial Industry Project Proposal System
Proposal Database
Proposal Database
Open evaluation section
Open proposal student
Fill in evaluation form and submit
Notify proposal has been evaluated
Exc
Open evaluation section
Proposal not found
Display "no data found"

Figure 2.11: Activity Diagram for Give Evaluation


Open evaluation section on the system
Open student proposal student
DATA FOUND
DATA NOT FOUND
Fill in evaluation form and submit
Display "no data found" to user
System notified proposal has evaluated

2.2.5 UC005: Use Case Upload & update Form A

Table 2.5: Use Case Description for upload and update form A

Use case: Upload and Update Form A
ID: UC005
Actor(s): Student
Preconditions: The student log in to the system using their valid ID.
Flow of events: 1. The student clicks the submission section for form A 2. The student Upload the form 3. The student clicks submit button 4. Student send request update to update their form
Postconditions: The form has been submitted.
Exception flow (if any): Student only can update their form if they have permission by examiner.

Figure 2.12: Sequence Diagram for Upload and Update Form A

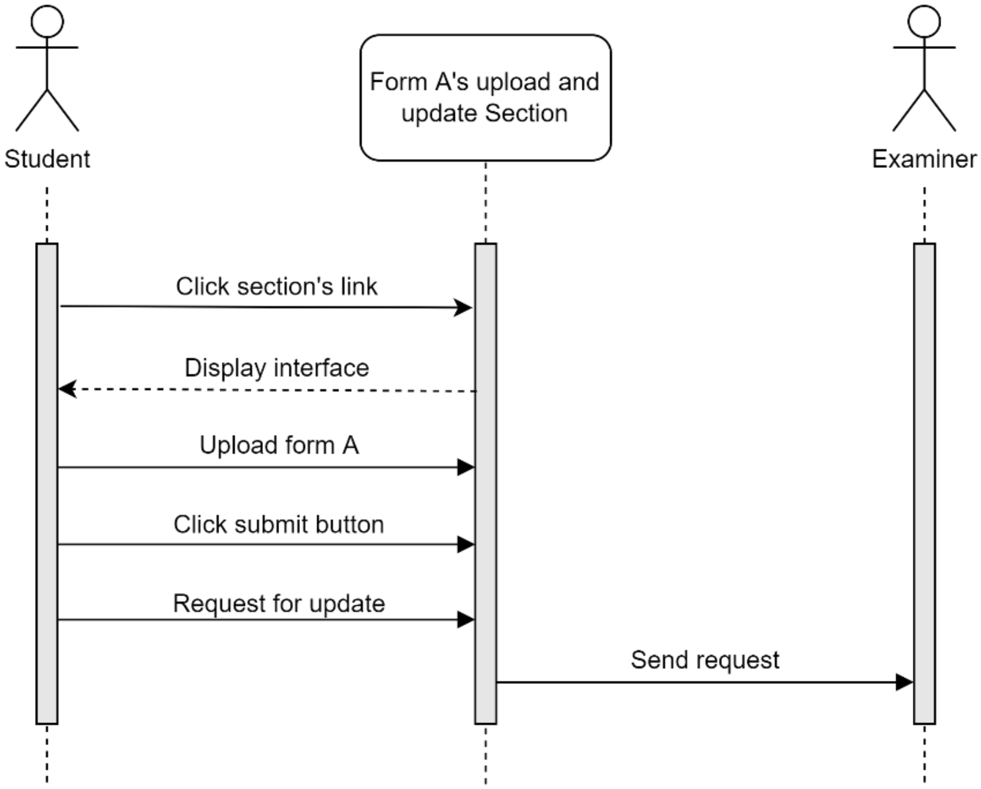
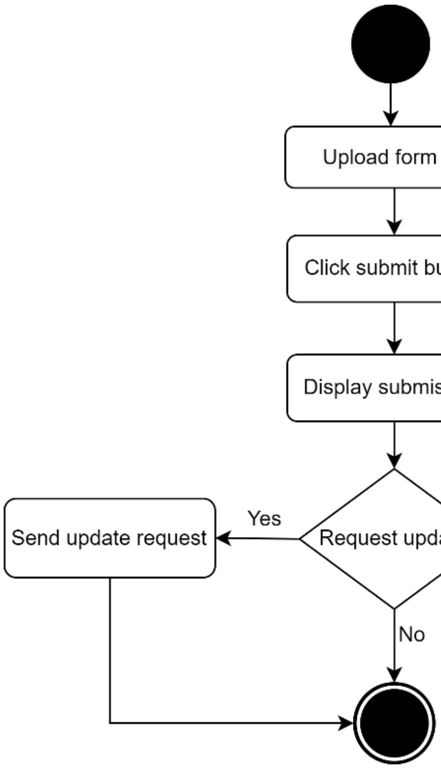


Figure 2.13: Activity diagram for upload and update form A



2.2.6 UC006: Use Case Give Comment Form A

Table 2.6: Use Case Description for Give Comment Form A

Use case: Give Comment Form A
ID: UC006
Actor(s): Examiner
Preconditions: Examiner login to the system using their valid ID.
Flow of events: 1. The Examiner clicks the form A uploaded by student 2. Examiner click the comment section on the From A’s submission interface 3. Examiner give their comment. 4. Examiner receives request update 5. Examiner approve / deny the request
Postconditions: The comment has been submitted.
Exception flow (if any): Only examiner will be able to grant permission to update the form.

Figure 2.14: Sequence Diagram for Give Comment Form A

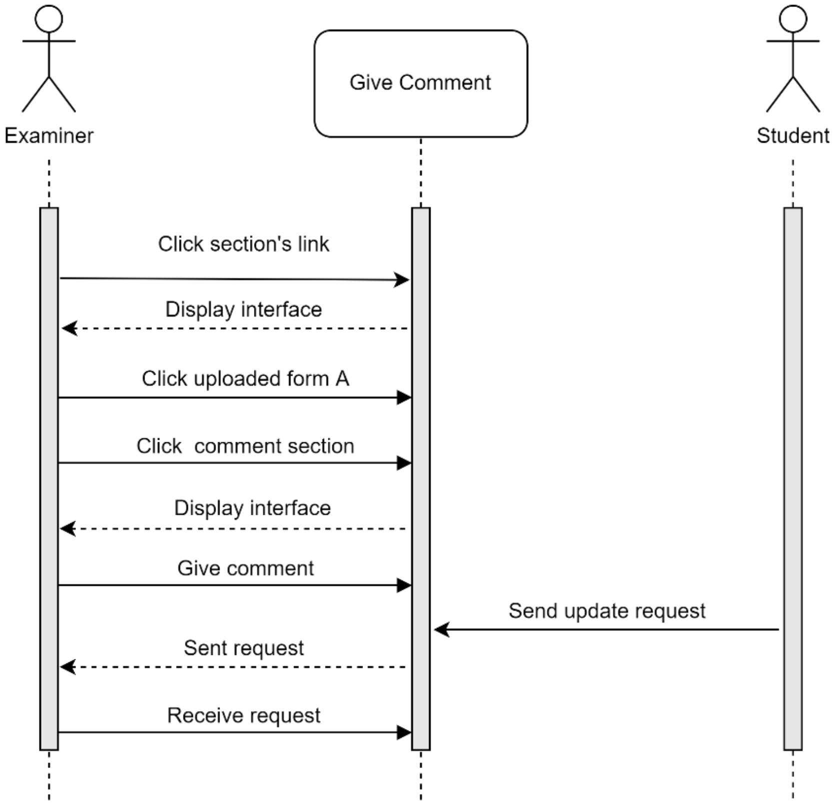
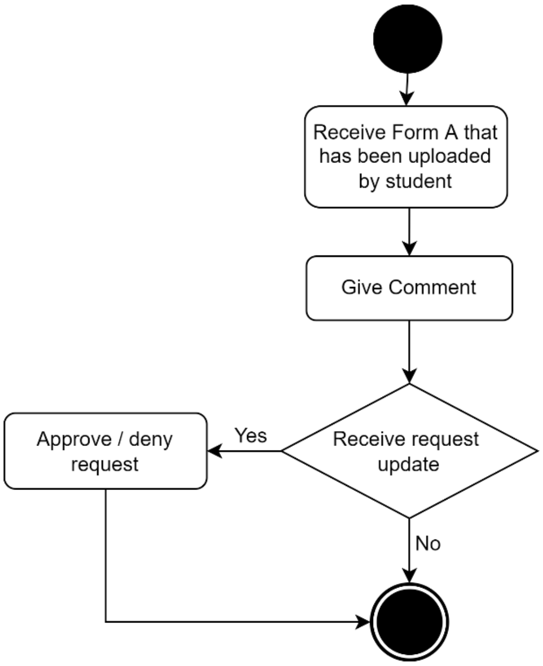


Figure 2.15 Activity diagram for give comment form A



2.2.7 UC007: Use Case Submit Proposal

Table 2.7: Use Case Description for Submit Proposal

Use case: Submit Proposal
ID: UC007
Actors: Student, Coordinator
Preconditions: The actors need to login into the system
Flow of events: 1. Student need to choose the “Submit proposal” button on the main menu. 2. Click the “Add submission” button. 3. Choose their proposal file from the folder and click “Upload this file” button. 4. Student can choose the “Save” button and finish their work or they can choose the “Cancel” button to select another file. 5. System will notify the coordinator the status of student’s submission. 6. Coordinator can click “student’s proposal” button to view their work.
Postconditions: Student’s proposal has been uploaded into the system database.

Figure 2.16: Sequence Diagram for Submit Proposal

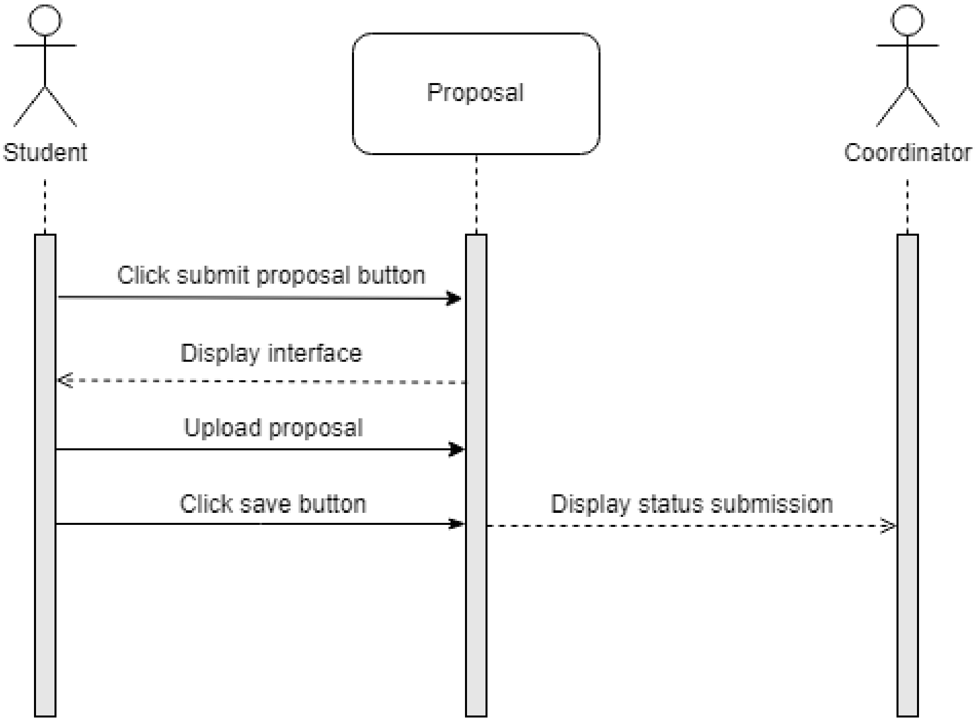
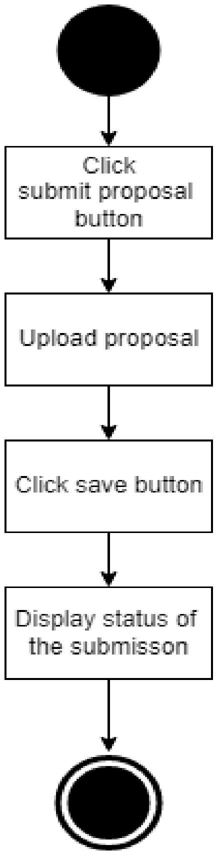


Figure 2.17: Activity Diagram for Submit Proposal



2.2.8 UC008: Use Case View Status

Table 2.8: Use Case Description for View status

Use case: View status
ID: UC008
Actors: Student
Preconditions: Student need to login into the system
Flow of events: 1. Student need to choose the "View status" button on the main menu. 2. Select "UC", "IC" or "Examiners" button. 3. System will display the approval status for their work.
Postconditions: -

Figure 2.18: Sequence Diagram for View status

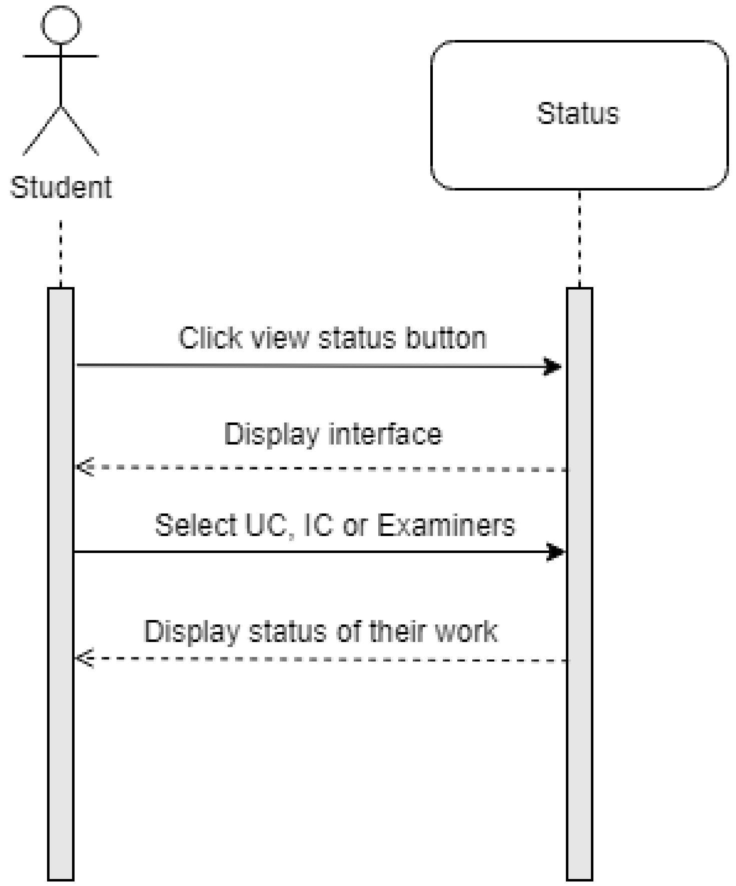
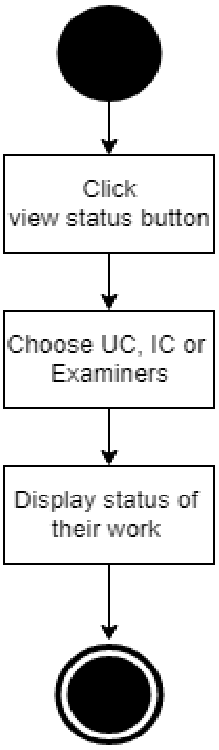


Figure 2.19: Activity Diagram for View Status



2.2.9 UC009: Use Case View Document

Table 2.9: Use Case Description for View document

Use case: View document
ID: UC009
Actors: Coordinator
Preconditions: Coordinator need to login into the system
Flow of events: 1. Coordinator need to choose the "View documents" button on the main menu. 2. Select which document to view 3. Choose the student 4. System will display the student's document
Postconditions: -

Figure 2.20: Sequence Diagram for View documents

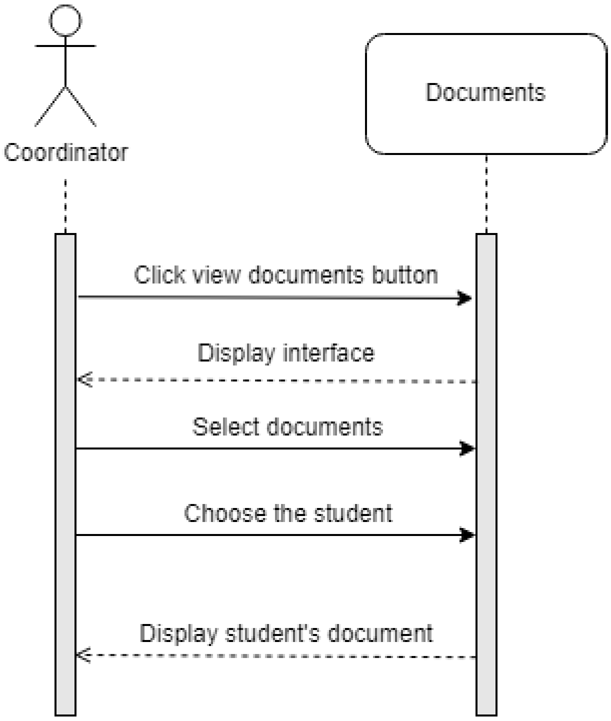
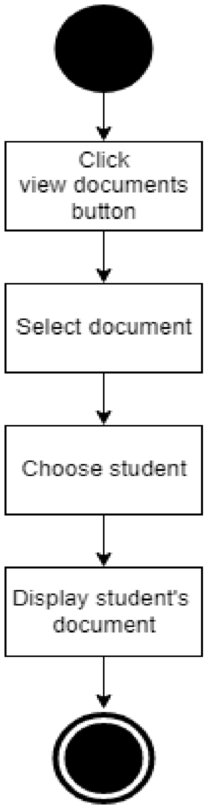


Figure 2.21: Activity Diagram for View documents



2.3 Performance and Other Requirements

1. After beginning the process of looking through files, the system should be able to produce search results within five seconds.
2. The system will be equipped with a centralized database that will be used to store all of the relevant information.
3. The system will be compatible with mobile devices in addition to desktop PCs in terms of its ability to function.
4. The system must be able to support at least one thousand users at the same time.

2.4 Design Constraints

1. The system is only available in English.
2. Users can use any platform to access the system, including laptops, mobile phones, tablets, and other devices.
3. The system should respond to the user quickly and without delay.
4. All of the information on the system must be informative, responsive, and well-organized.
5. At any given moment, the system should be able to accommodate up to 250 users.
6. The database must be able to update all of the information in a timely and secure manner.

2.5 Software System Attributes

With the online platform, students, coordinator, examiner, Industrial coach and University coach will have a place to communicate to each other and to submit any requirement for their Final Year Project (FYP). They do not have to submit it physically. This will make the submission and communication between two parties become faster and convenient. There will be four level for this system. One for students, one for coordinator, one for examiner and the last level is for university coach and industrial coach. The primary target audience is student. Most of the system functionalities will be geared toward them. The other users will have higher access to the system.

3. **System Architectural Design**

3.1 Architecture Style and Rationale

Because it is the design that works best with the system, we have decided to employ the Layered Architectural style when it comes to architecture patterns. This style is also known as an n-tier architecture. Because it divides the system into multiple different horizontal levels, each of which performs a specialised function inside the programme, this architectural framework is quite popular in the field of software engineering. This architectural choice was made with the goal of making it simpler to deliver implementations on several platforms. (Web Application, Mobile Application) If one layer is altered, the others will continue to maintain their previous state. As a result, if there are adjustments that need to be made in one component of the system, it will not have an impact on the other components of the system. The entire system has been properly arranged, and the members of the development team can navigate around it with ease. To make it simpler for the development team to make changes to the system's components, the various areas of the system have been carefully segmented. Because the code organisation is so clearly defined, it is simple for any development team to keep up with its maintenance. Members of the development team are able to work on separate aspects of the programme in parallel with only limited reliance on the work of other team members thanks to the layered architecture.

3.2 Component Model

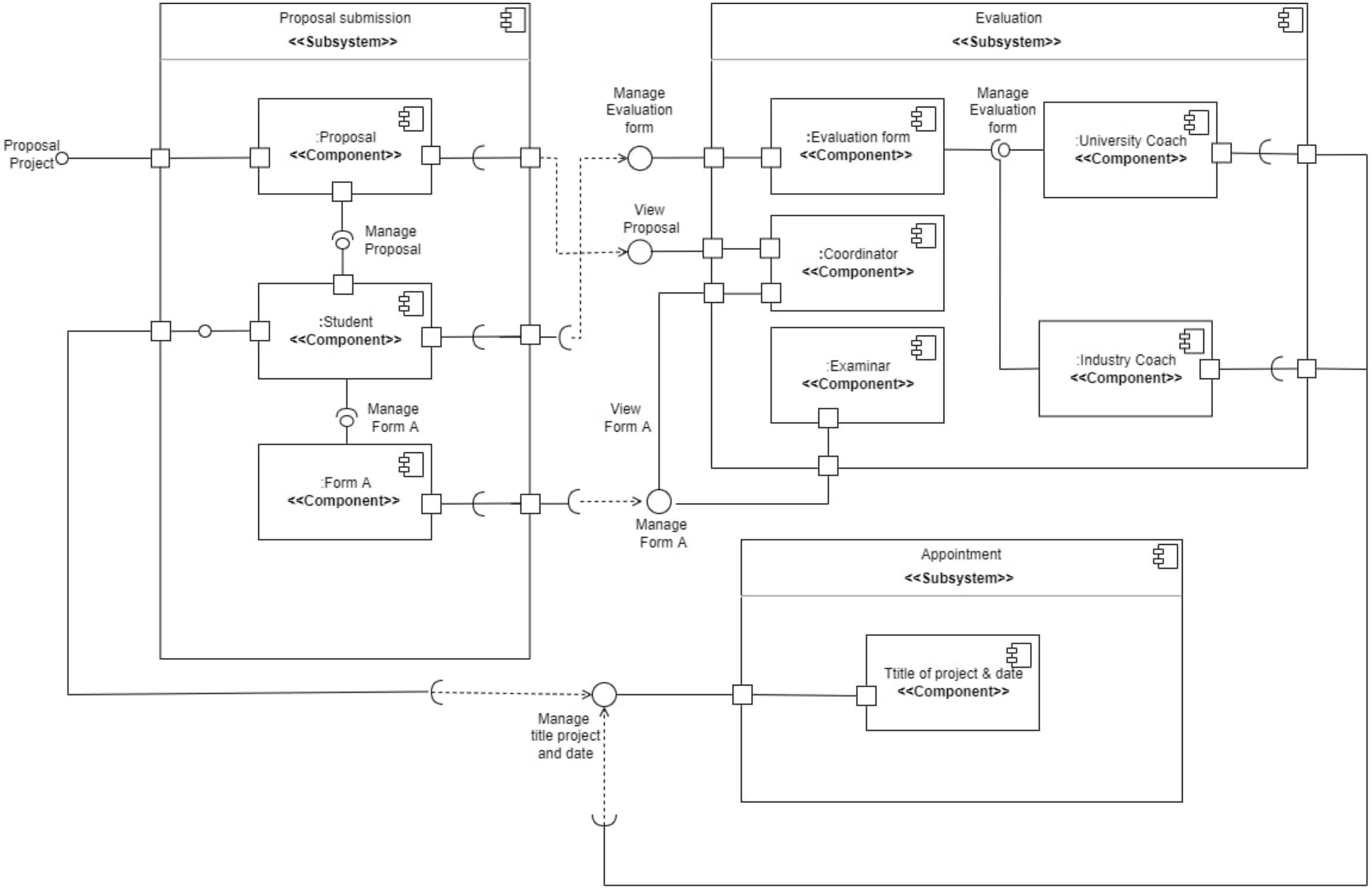


Figure 3.1: Component Diagram of SCSP 4112 – Initial Industry Project Proposal System

4. Detailed Description of Components

4.1 Complete Package Diagram

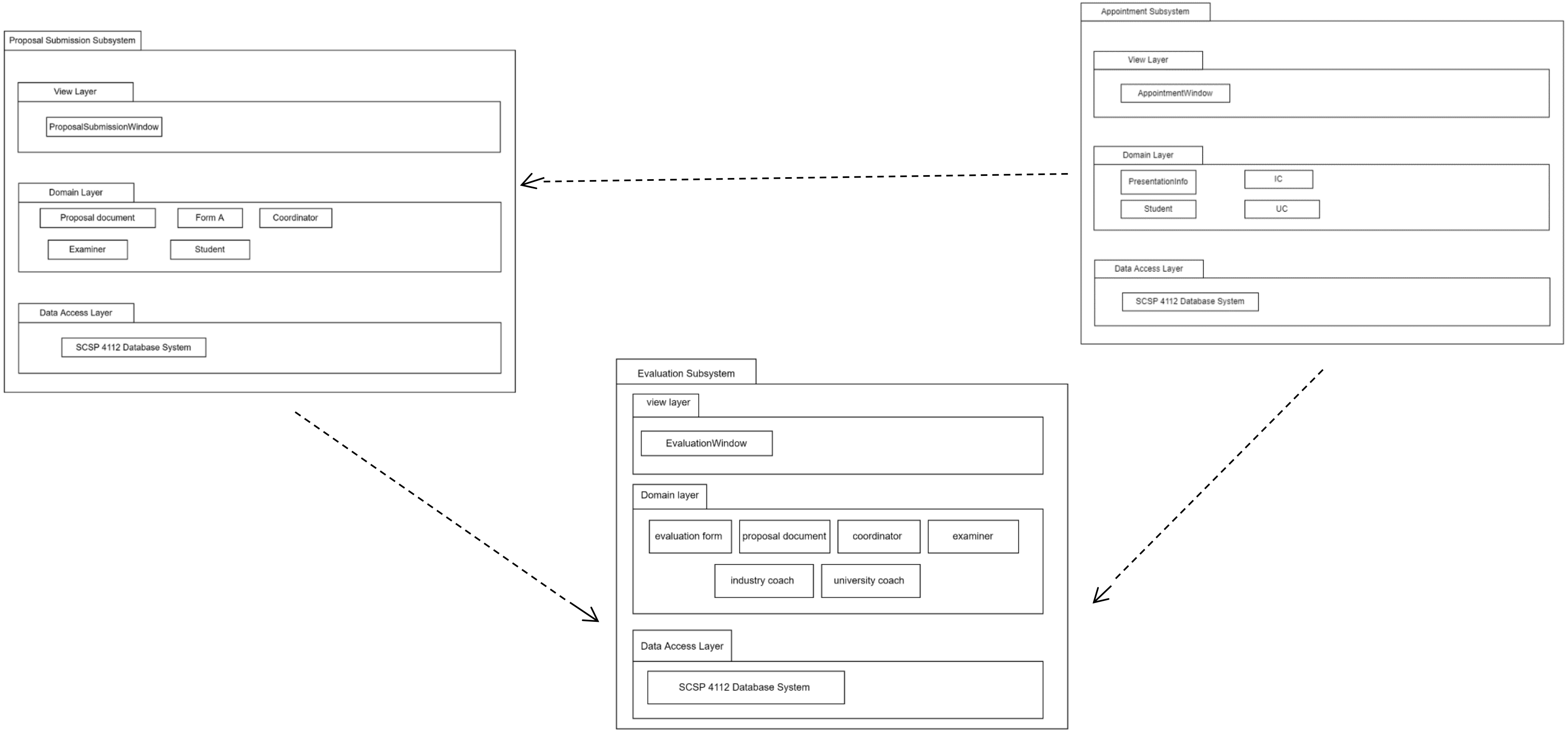


Figure 4.1: Package Diagram for SCSP 4112 – Initial Industry Project Proposal System

4.2 Detailed Description

4.2.1 P001: Proposal Submission Subsystem

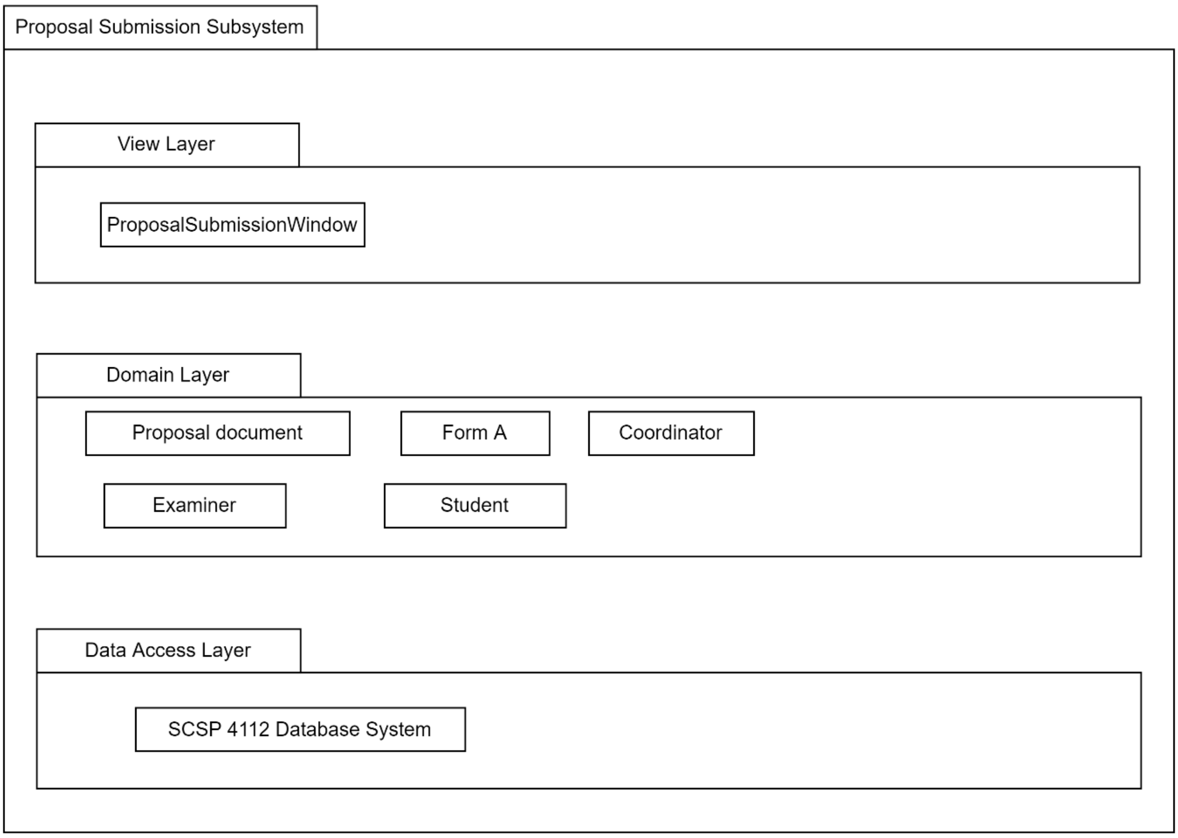


Figure 4.2: Package Diagram for Proposal Submission Subsystem

Entity Name	Proposal Submission Application
Method Name	displayDocument()
Input	Upload file, Request document
Output	Display confirmation status, Display document
Algorithm	1. Start 2. If Student 2.1 Upload Form A 2.2 Upload proposal 3. Else if Examiner 3.1 request Form A 4. if Coordinator 4.1 request Form A 4.2 request Proposal 5. Display Document 6. End

4.2.1.1 Class Diagram

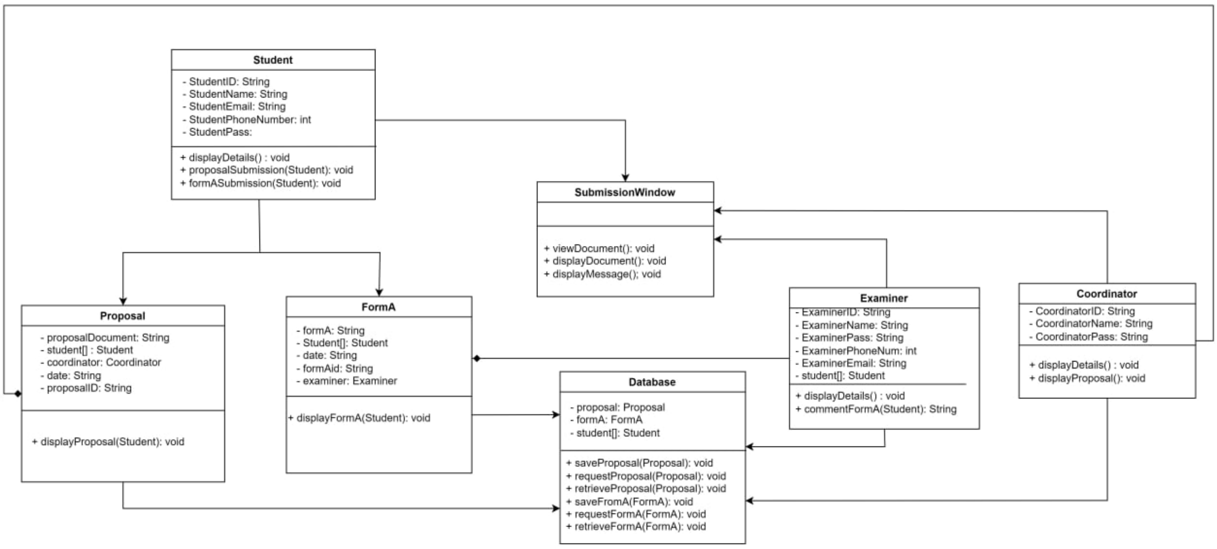

Student
- StudentID: String
- StudentName: String
- StudentEmail: String
- StudentPhoneNumber: int
- StudentPass:
+ displayDetails() : void
+ proposalSubmission(Student): void
+ formASubmission(Student): void
SubmissionWindow
+ viewDocument(): void
+ displayDocument(): void
+ displayMessage(); void
Proposal
- proposalDocument: String
- student[] : Student
- coordinator: Coordinator
- date: String
- proposalID: String
+ displayProposal(Student): void
FormA
- formA: String
- Student[]: Student
- date: String
- formAid: String
- examiner: Examiner
+ displayFormA(Student): void
Database
- proposal: Proposal
- formA: FormA
- student[]: Student
+ saveProposal(Proposal): void
+ requestProposal(Proposal): void
+ retrieveProposal(Proposal): void
+ saveFromA(FormA): void
+ requestFormA(FormA): void
+ retrieveFormA(FormA): void
Examiner
- ExaminerID: String
- ExaminerName: String
- ExaminerPass: String
- ExaminerPhoneNum: int
- ExaminerEmail: String
- student[]: Student
+ displayDetails() : void
+ commentFormA(Student): String
Coordinator
- CoordinatorID: String
- CoordinatorName: String
- CoordinatorPass: String
+ displayDetails() : void
+ displayProposal(): void

Figure 4.3: Class Diagram for Proposal Submission Subsystem

4.2.1.2 Sequence Diagram

a) SD001: Sequence diagram for Form A and proposal submission

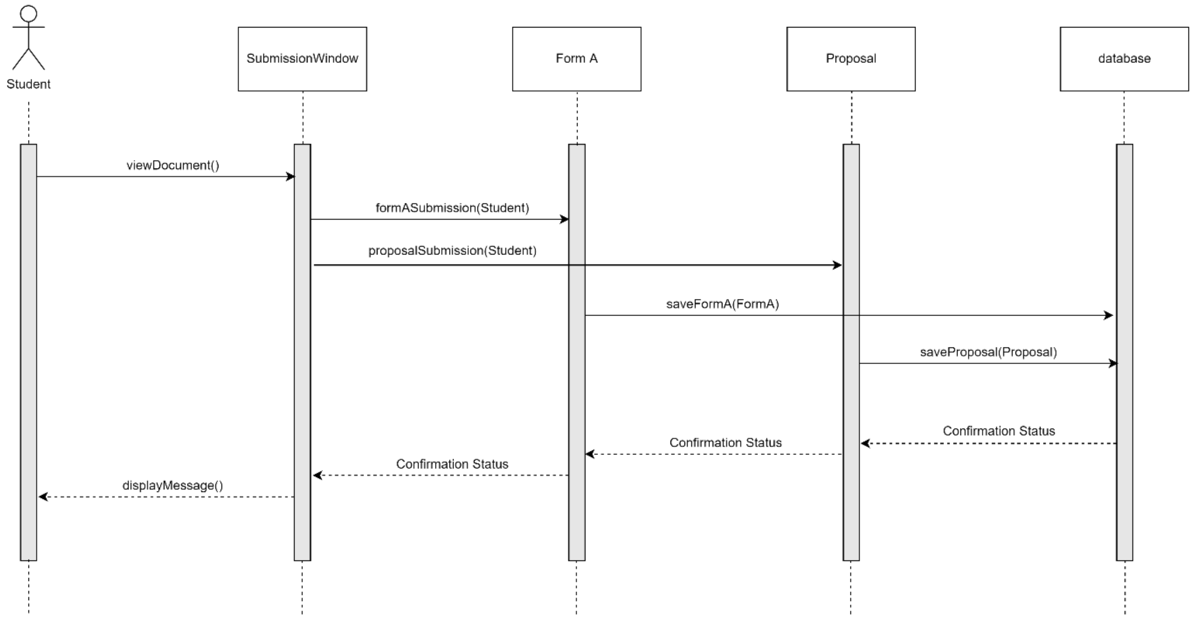


Figure 4.4: Sequence Diagram for Form A and proposal submission

b) SD002: Sequence diagram for View Form A

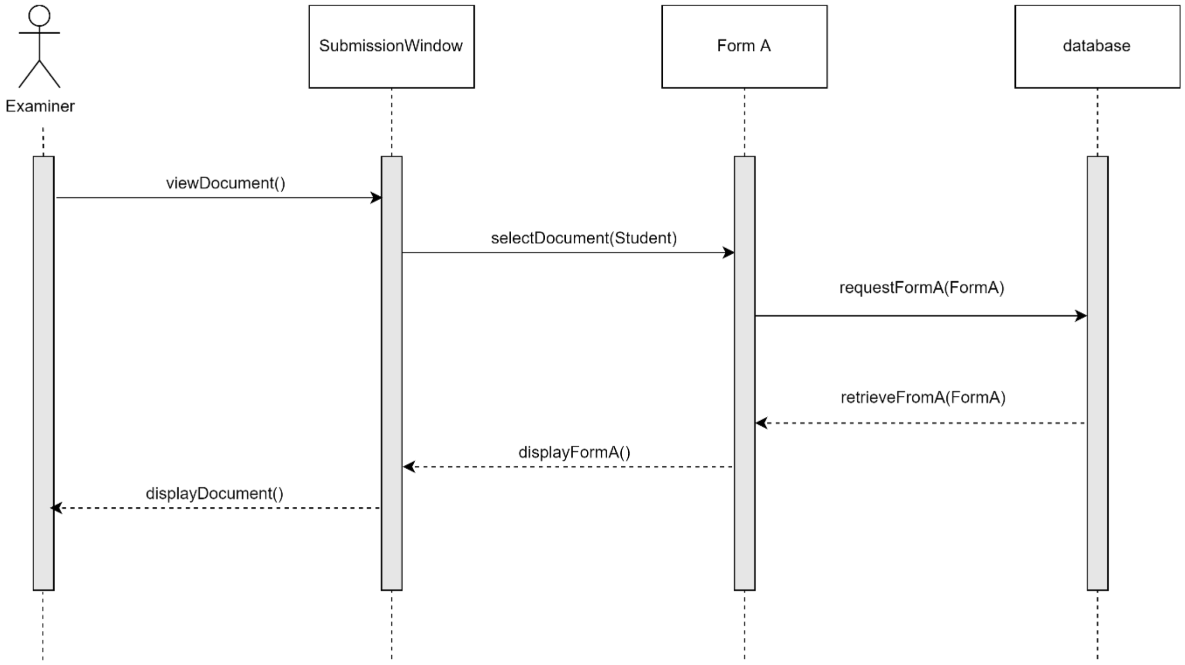


Figure 4.5: Sequence Diagram for view form A

c) SD003: Sequence diagram for view Form A and Proposal

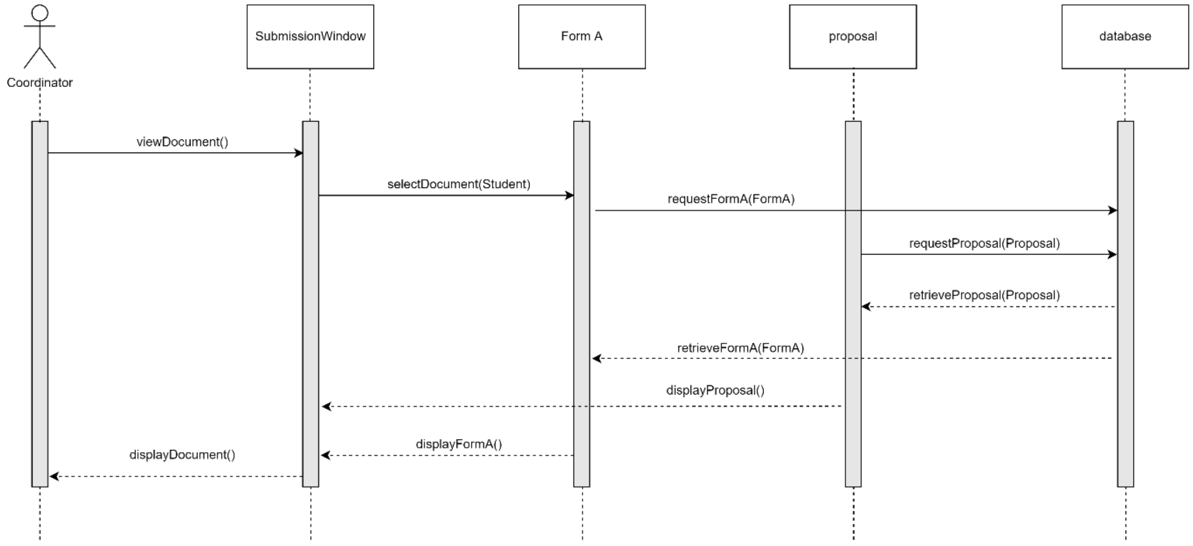


Figure 4.6: Sequence Diagram for view Form A and Proposal

4.2.2 P002: Evaluation Subsystem

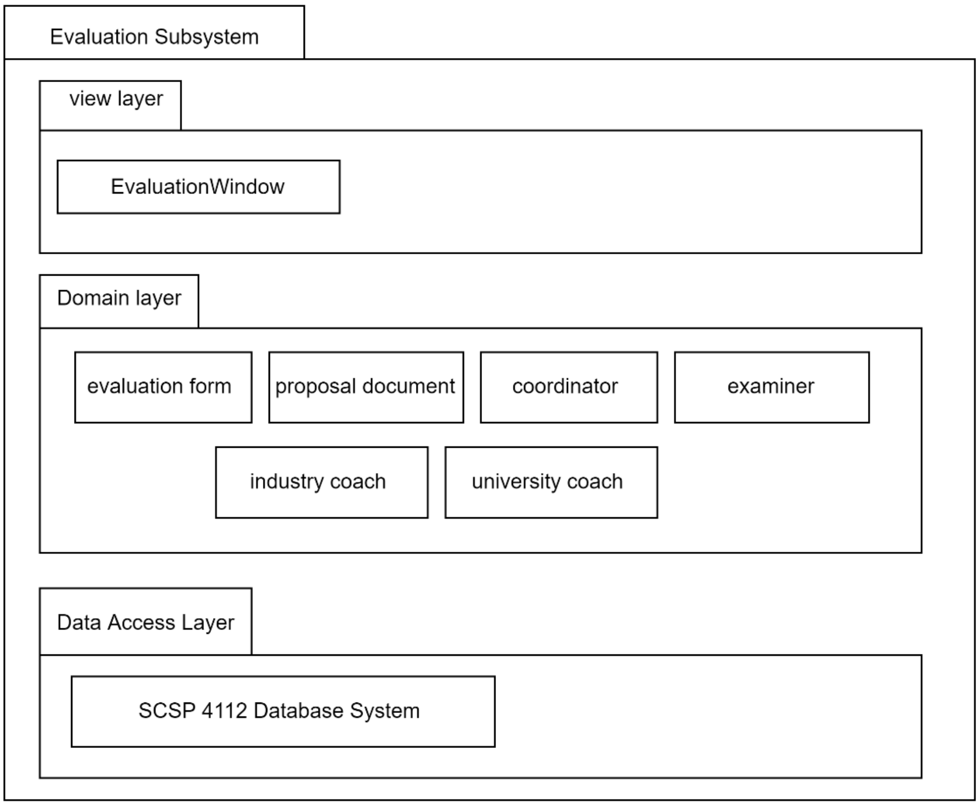


Figure 4.7: Package Diagram for Proposal Submission Subsystem

4.2.2.1 Class Diagram

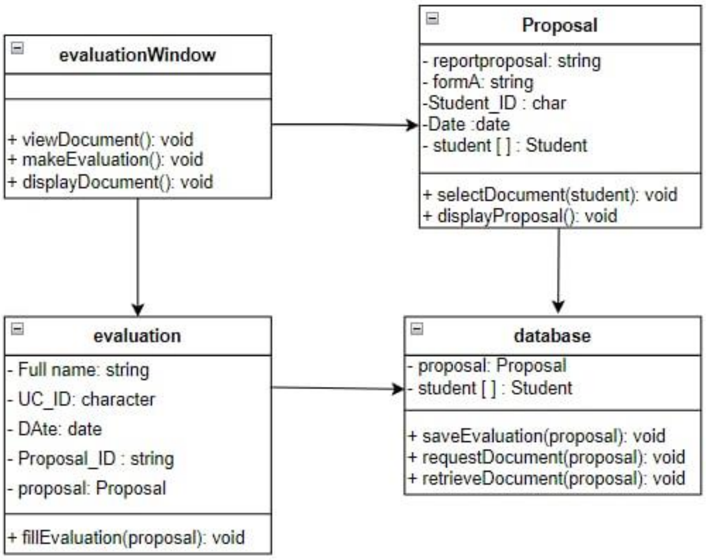


Figure 4.8: Class Diagram for Proposal Submission Subsystem

Entity Name	evaluationWindow
Method Name	+ viewDocument()
Input	User select “view Document” button from page
Output	Invocation of Proposal’s class method
Algorithm	1. Start 2. Click on “view Document” button 3. List of documents send to user from Proposal method 4. End
Method Name	+ makeEvaluation()
Input	User select “make Evaluation” button from page
Output	Invocation of Proposal’s class method
Algorithm	1. Start 2. Click on “make evaluation” button 3. List of documents send to user from Proposal method 4. End
Method Name	+ displayDocument()
Input	Data from proposal method
Output	Display selected documents
Algorithm	1. Start 2. Get data from displayProposal() and database 3. Display selected documents 4. End

Entity Name	Proposal
Method Name	+ selectDocument(student)
Input	Invocation from evaluationWindow
Output	Invocation of database's class method
Algorithm	1. Start 2. User's choice from evaluationWindow 3. requestDocumet method from database class are called 4. End
Method Name	+ displayProposal()
Input	Invocation from database's class method
Output	Invocation of evaluationWindow
Algorithm	1. Start 2. Invocation of retrieveDocument from database's class 3. Receive documents from the database 4. Called displayDocuments method from evaluationWindow 5. End

Entity Name	evaluation
Method Name	+ fillEvaluation(proposal)
Input	Invocation from evaluationWindow
Output	Invocation of database's class method
Algorithm	1. Start 2. User's choice from evaluationWindow 3. Fill the evaluation form 4. saveEvaluation(proposal) method from database class are called 5. End

4.2.2.2 Sequence Diagram

a) SD004: Sequence diagram to View Documents

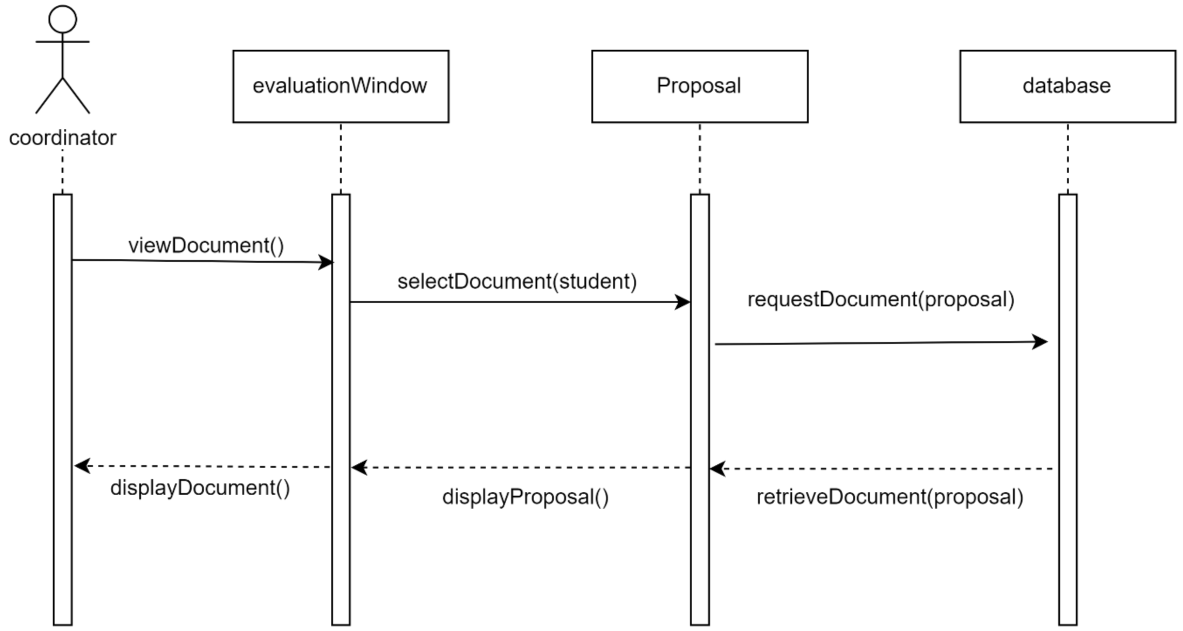


Figure 4.9: Sequence Diagram for View Documents

b) SD005: Sequence diagram to Give Evaluation

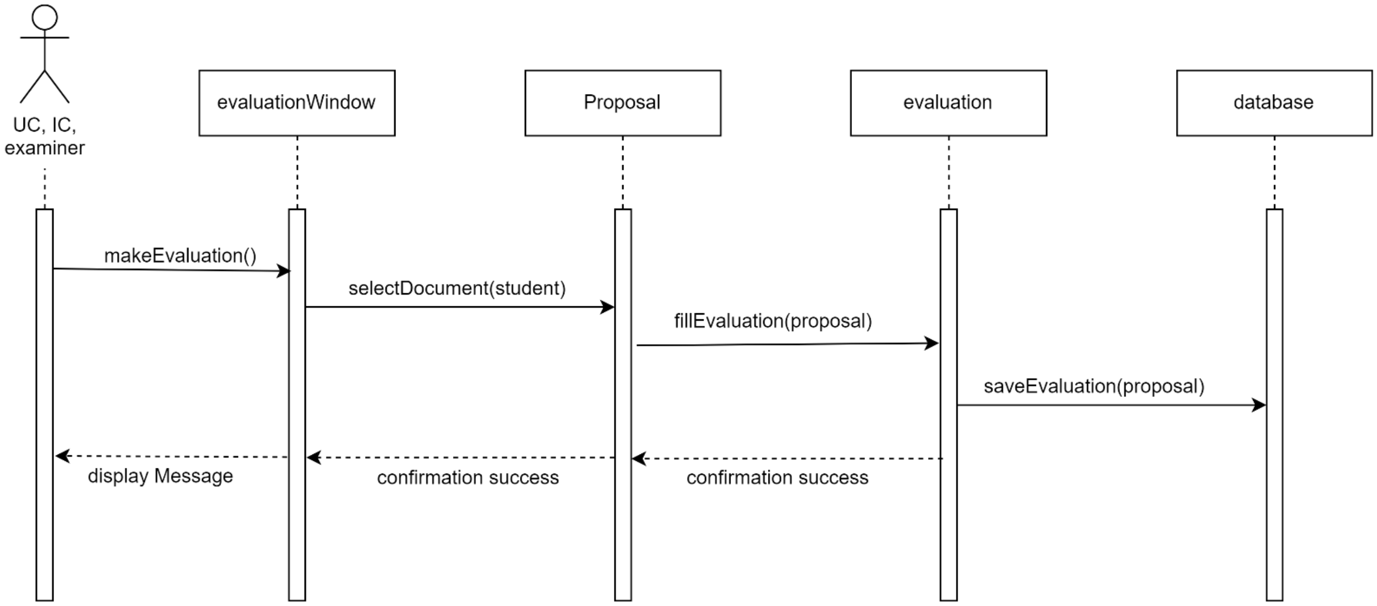


Figure 4.10: Sequence Diagram for Give Evaluation

4.2.3 P003: Appointment Subsystem

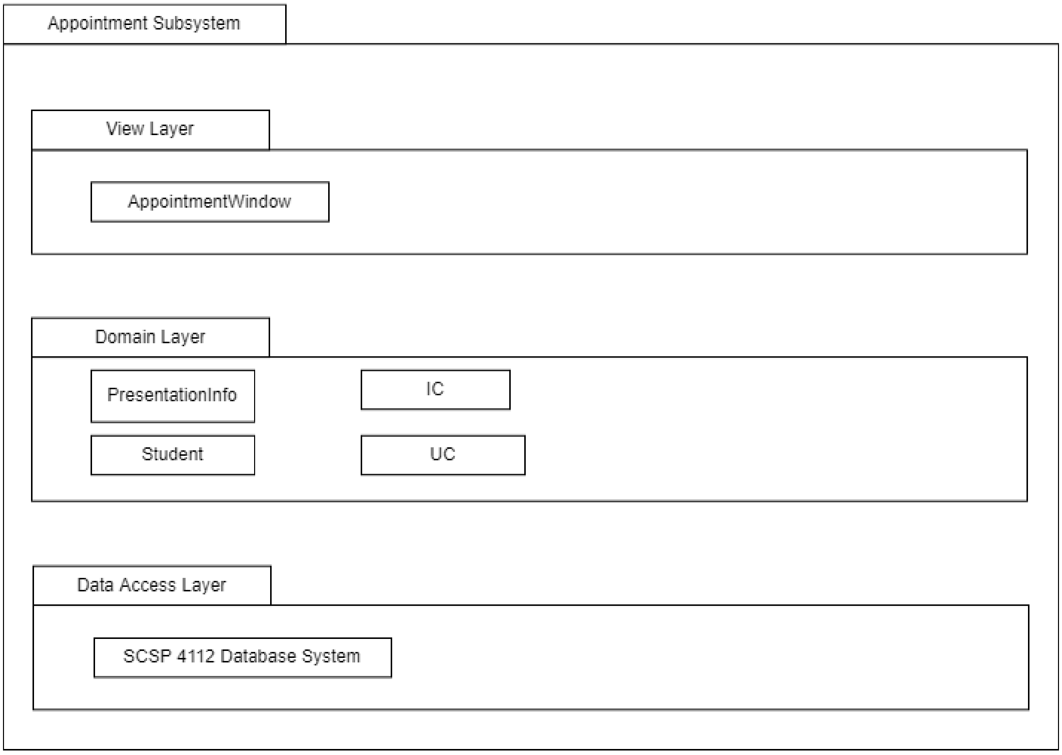


Figure 4.11: Package Diagram for Appointment Subsystem

Entity Name	PresentationInfo
Method Name	setTitle(String)
Input	newtitle
Output	-
Algorithm	1. Start 2. Set title == newtitle 3. End

Entity Name	PresentationInfo
Method Name	setDate(int)
Input	newdate
Output	-
Algorithm	1. Start 2. Set newdate == date 3. End

Entity Name	PresentationInfo
Method Name	displayPresentationInfo()
Input	-
Output	Display the presentation info
Algorithm	1. Start 2. Print student info 3. Print university coach info 4. Print industry coach info 5. Print tittle of the project 6. Print date of the presentation 3. End

4.2.3.1 Class Diagram

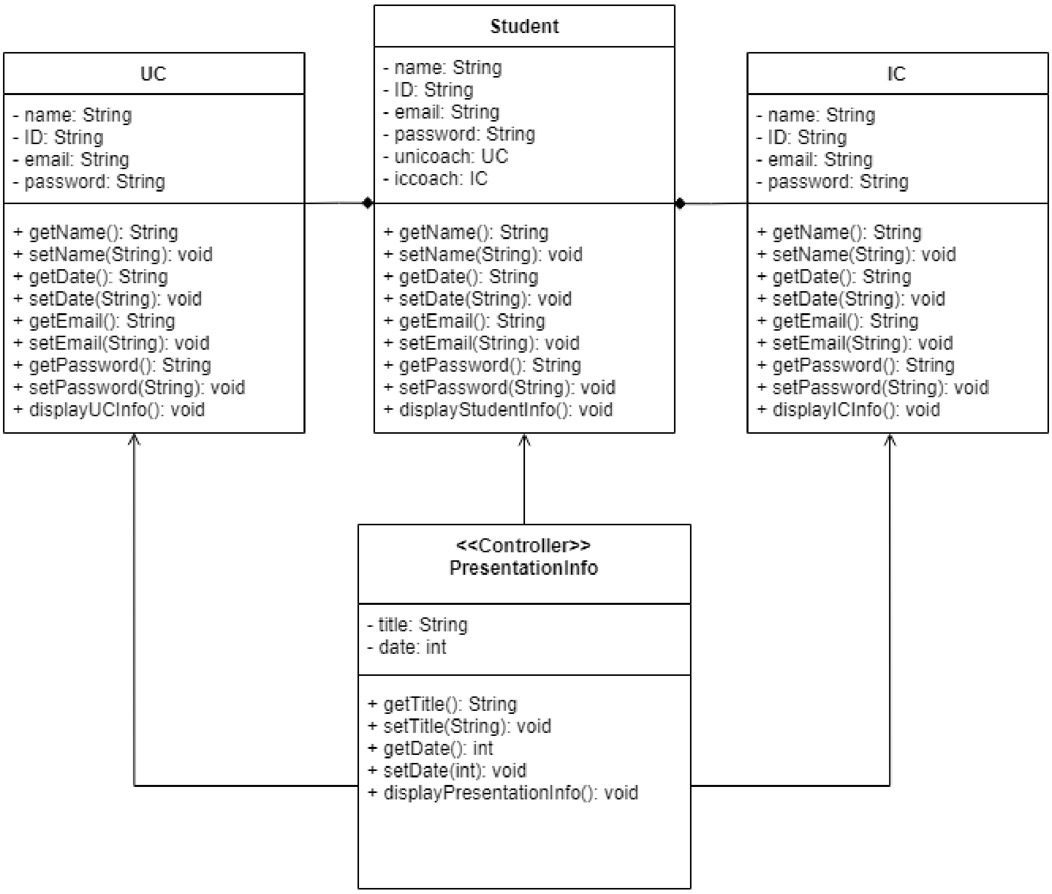


Figure 4.12: Class Diagram for Appointment Subsystem

4.2.3.2 Sequence Diagram

a) SD006: Sequence diagram for Set New Title for Project

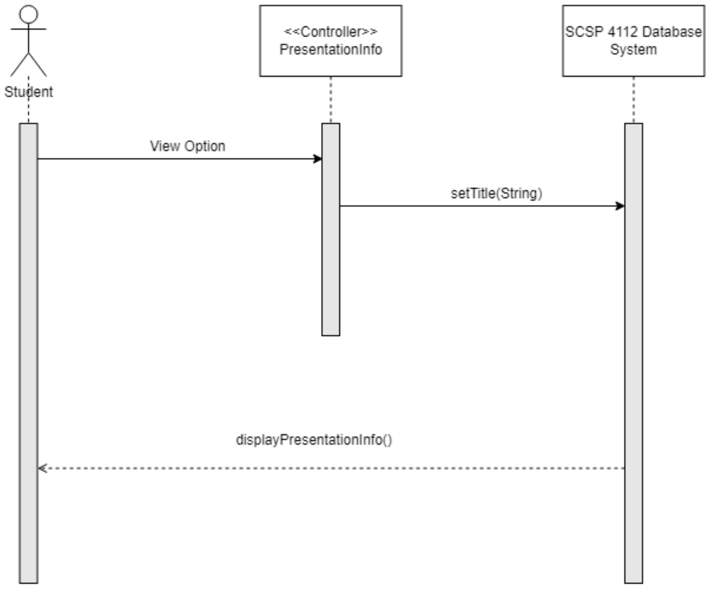

Student
<<Controller>> PresentationInfo
SCSP 4112 Database System
View Option
setTitle(String)
displayPresentationInfo()

Figure 4.13: Sequence Diagram for Set New Title for Project

b) SD007: Sequence diagram for Set New Date for Presentation

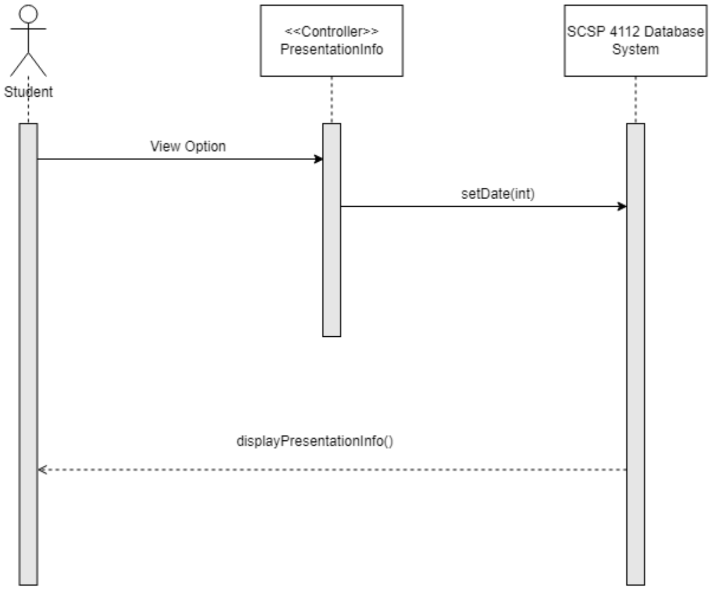

Student
<<Controller>> PresentationInfo
SCSP 4112 Database System
View Option
setDate(int)
displayPresentationInfo()

Figure 4.14: Sequence Diagram for Set New Date for Presentation

5. Data Design

5.1 Data Description

The major data or systems entities are stored into a relational database named as SCSP 4112 system, processed and organized into *7* entities as listed in Table 5.1.

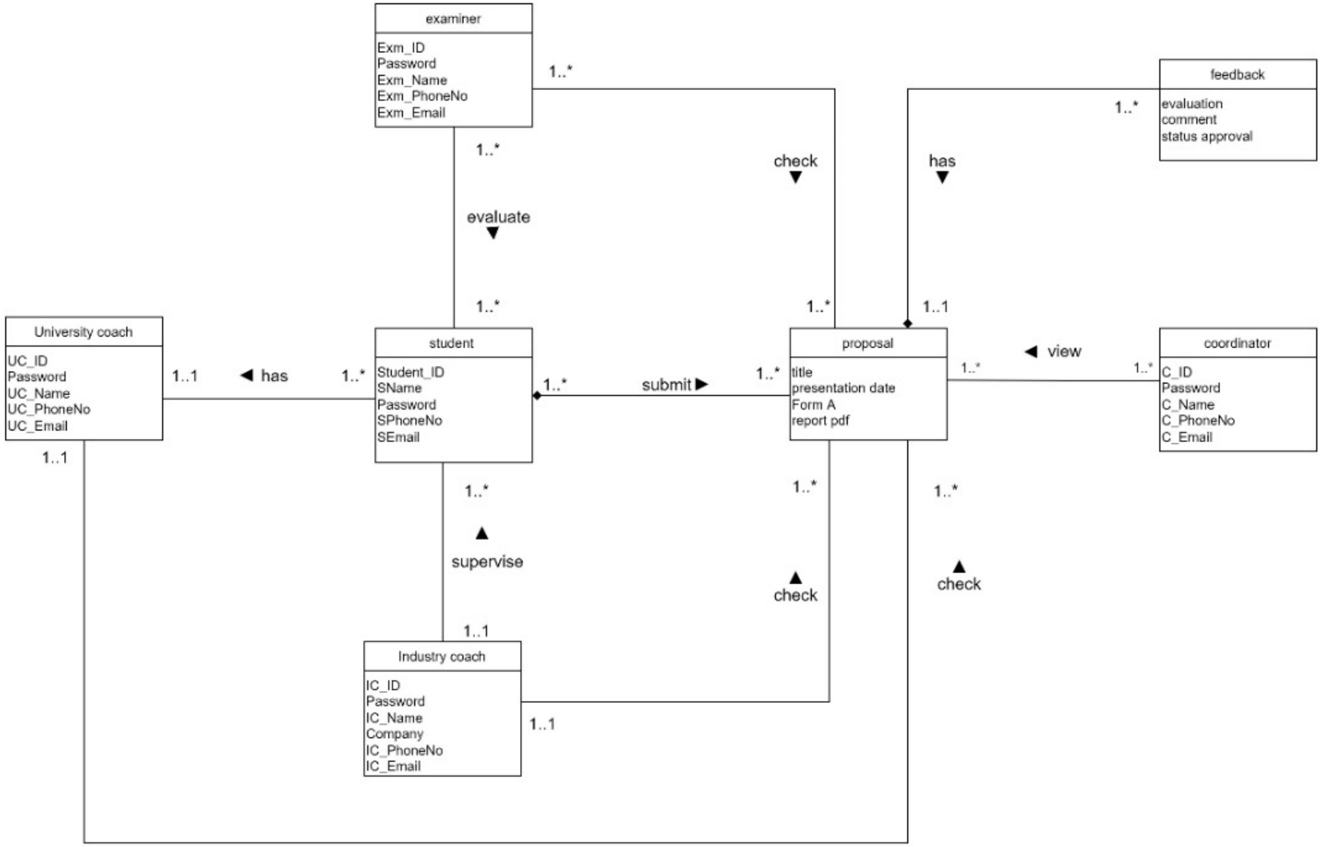


Table 5.1: Description of Entities in the Database

No	Entity Name	Description
1	University coach	Store basic information about University coach such as ID , name, email ,etc.
2	Student	Store basic information about student such as ID , name, email ,etc.
3	Examiner	Store basic information about examiner such as ID , name, email ,etc.
4	Industry coach	Store basic information about industry coach such as ID , name, email ,etc.
5	Proposal	Store all material submitted by student such as tittle, presentation date, Form A and report .
6	Feedback	Store all document that related to evaluation process .
7	Coordinator	Store basic information about coordinator such as ID , name, email ,etc.

5.2 Data Dictionary

5.2.1 Entity: University coach

Attribute Name	Type	Description
UC_ID	Varchar2 (20)	Unique Staff Code of a University coach, Primary Key.
Password	nvarchar(50)	Password of the University coach.
UC_Name	Varchar2 (50)	Name of the University coach.
UC_PhoneNo	int	Phone number of the University coach.
UC_Email	Varchar2 (40)	Email of the University coach.

5.2.2 Entity: Student

Attribute Name	Type	Description
Student_ID	Varchar2 (20)	Unique Code of student, Primary Key.
Password	nvarchar(50)	Password of the student.
SName	Varchar2 (50)	Name of the student.
SPhoneNo	int	Phone number of the student.
SEmail	Varchar2 (40)	Email of the student.

5.2.3 Entity: Examiner

Attribute Name	Type	Description
Exm_ID	Varchar2 (20)	Unique Staff Code of a examiner, Primary Key.
Password	nvarchar(50)	Password of the examiner.
Exm _Name	Varchar2 (50)	Name of the examiner.
Exm _PhoneNo	int	Phone number of the examiner.
Exm _Email	Varchar2 (40)	Email of the examiner.

5.2.4 Entity: Industry Coach

Attribute Name	Type	Description
IC_ID	Varchar2 (20)	Unique Staff Code of a Industry coach, Primary Key.
Password	nvarchar(50)	Password of the Industry coach.
IC _Name	Varchar2 (50)	Name of the Industry coach.
IC _PhoneNo	int	Phone number of the Industry coach.
IC _Email	Varchar2 (40)	Email of the Industry coach.
Company	Varchar2 (40)	Company of the Industry coach.

5.2.5 Entity: Proposal

Attribute Name	Type	Description
title	Varchar2 (50)	Title of project that chosen by student.
presentation date	Varchar2 (50)	Presentation date of project that chosen by student.
Form A	Varchar2 (50)	Submission of Form A by student.
report pdf	Varchar2 (50)	Report of Form A by student.

5.2.6 Entity: Feedback

Attribute Name	Type	Description
evaluation	Varchar2 (50)	Evaluation form submitted by University coach and Industry coach.
comment	Varchar2 (50)	Comment form submitted by University coach and Industry coach.
status approval	Varchar2 (50)	Approval by coach and examiner.

5.2.7 Entity: Coordinator

Attribute Name	Type	Description
C_ID	Varchar2 (20)	Unique staff code of a coordinator, Primary Key.
Password	nvarchar(50)	Password of the coordinator.
C _Name	Varchar2 (50)	Name of the coordinator.
C_PhoneNo	int	Phone number of the coordinator.
C_Email	Varchar2 (40)	Email of the coordinator.