SECJ2203: Software Engineering

SECJ2203 - 09

System Documentation (SD)

Data Engineering FYP System

Version 3.0

02 / 07 / 2023

Faculty of Computing

Prepared by: Techies

NAME	MATRIC NUMBER	ROLE
MAATHUREE VEERABALAN	A21EC0051	MEMBER
MALAVIKA BASKARAN	B22EC0069	MEMBER
NASRUL AMIN BIN AB HADI	A21EC0099	LEADER
THUVAARITHA SIVARAJAH	A21EC0137	MEMBER

Revision Page

a. Overview

Data Engineering FYP System version 3.0 describes the test case of our system.

Target Audience

Our target audience is 4th-year students of the Data Engineering course and staff who handle the Data Engineering final year project.

b. Project Team Members

Member Name	Role	Task	Status
MAATHUREE VEERABALAN	MEMBER	4.1 Complete Package Diagram 4.2 Upload Subsystem 5.1 Data Description 5.2 Data Dictionary 6.0 Requirement Traceability Matrix 7.2 Test Upload Subsystem	Completed
MALAVIKA BASKARAN	MEMBER	4.2 Submission Subsystem 5.1 Data Description 5.2 Data Dictionary 6.0 Requirement Traceability Matrix 7.4 Test Submission Subsystem	Completed
NASRUL AMIN BIN AB HADI	LEADER	3.2 Component Model 4.2 Assign Subsystem 5.1 Data Description 5.2 Data Dictionary 6.0 Requirement Traceability Matrix 7.3 Test Assign Subsystem	Completed
THUVAARITHA SIVARAJAH	MEMBER	3.1 Architecture Pattern and Rationale 4.2 Login Subsystem 5.1 Data Description 5.2 Data Dictionary 6.0 Requirement Traceability Matrix	Completed

		7.1 Test Login Subsystem	

c. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Malavika Baskaran	Combined all the parts	22/05/2023
	Thuvaaritha Sivarajah	Ensure correct formatting	22/05/2023
	Maathuree Veerabalan	Final editing	22/05/2023
	Nasrul Amin	Final Editing	22/05/2002
2.0	Malavika Baskaran	Combined all the parts	16/06/2023
	Thuvaaritha Sivarajah	Final editing	16/06/2023
	Maathuree Veerabalan	Ensure correct formatting	16/06/2023
	Nasrul Amin	Final editing	16/06/2023
3.0	Nasrul Amin	Ensure correct formatting	2/7/2023
	Malavika Baskaran	Combined all the parts	2/07/2023
	Maathuree Veerabalan	Final editing	2/07/2023
	Thuvaaritha Sivarajah	Final editing	2/07/2023

Table of Contents

1. Introduction

1.1 **Purpose**

The System Documentation describes the purpose of a Data Engineering FYP system which covers the specific requirement which includes user characteristic, the system features, software system attributes, design constraint, performance requirements, and other requirements.

1.2 **Scope**

The “Data Engineering FYP System” is to develop a collaborative online platform that facilitates the information of the Data Engineering Final Year Project. Students, coaches, examiners, and coordinators will get access to the system. The system is built to smoothen the communication between students, coaches, examiners, and coordinators, improving the efficiency in supervision and thus increasing the chances for students to enhance their performance.

The scope of the "Data Engineering FYP System" software product includes the following:

- The system will facilitate interactions such as sharing project updates, submitting deliverables, providing feedback, and managing project-related tasks.
- The system may incorporate features to track project progress, and monitor deadlines.
- The system will be developed using the waterfall methodology, ensuring it is easier to manage and control the project.
- The system will be developed following best practices in software engineering, including coding standards, testing procedures, and quality assurance processes.
- The system is more convenient and will ease the work for students, coaches, examiners, and coordinators.

1.3 **Definitions, Acronyms and Abbreviation**

Term	**Definition**
SD	System Documentation - documents that describe the system itself and its parts. It includes requirements documents, design decisions, architecture descriptions, program source code, and FAQs.
SRS	System Requirements Specification - a document that outline the requirements of the software product being developed
SDD	System Design Document - a document that describe the system architecture components and interfaces in details
STD	System Testing Document - document that outline the testing process and procedures for the developed software
FYP	Final Year Project

1.4 **References**

Athuralia, A. 2022. *Sequence Diagram Tutorial*. Creately. https://creately.com/guides/sequence-diagram-tutorial/

Athuralia, A. 2022. *Activity Diagram Tutorial*. Creately. https://creately.com/guides/activity-diagram-tutorial/

GeeksforGeeks. 2022. *Designing Use Cases for a Project*. https://www.geeksforgeeks.org/designing-use-cases-for-a-project/

1.5 **Overview**

The System Documentation (SD) provides a comprehensive description of the Data Engineering FYP system being developed. This document comprises three main sections: System Requirements Specification (SRS), System Design Document (SDD), and System Testing Document (STD).

The SD is organized as follows:

User Characteristics: This section outlines the target users of the Data Engineering FYP System, including their needs, expectations, and relevant characteristics.

System Features: This section provides a detailed description of the functional and non-functional requirements of the software product, including use cases, scenarios, and interfaces.

System Quality Attributes: This section outlines the quality attributes that are essential to the Data Engineering FYP System, , such as reliability, usability, maintainability, and portability.

Design Constraints: This section describes the performance requirements and design constraints that must be adhered to during the development of the software product.

This SD document is intended to serve as a reference for all stakeholders involved in the development, testing, and implementation of the Data Engineering FYP System. It provides a clear and concise overview of the software requirements, design, and testing procedures, and should be consulted throughout the software development lifecycle.

2. Specific Requirement

2.1 User Characteristic

The Data Engineering FYP System will be used by five main users: Students, Industry Coaches, University Coaches, Examiners, and Coordinators.

2.1.1 Students

- The UTM students who will use the FYP Management systems are expected to have basic computer skills, including familiarity with web-based applications.
- They have a background in data engineering and are completing their First Year Project (FYPi) as part of their bachelor's degree.
- They will be actively involved in producing project proposals, designing project solutions, and delivering project solutions.
- Students will be required to provide personal information such as name and email address in order to register for the system. The system will be designed with privacy and security measures in place to protect user information.

2.1.2 Industry Coaches

- Each student is assigned an industry coach who is a professional from the relevant field of study in the organization where the students are placed.
- Industry coaches guide the students through the project, provide domain-specific expertise, and ensure the project aligns with industry requirements.
- They will evaluate the students' work in all three objectives as stated in the case study using a predetermined rubric.
- Industry coaches will be required to provide personal information such as name and email address in order to register for the software. The software will be designed with privacy and security measures in place to protect user information.

2.1.3 University Coaches

- University coaches are faculty members who work with the industry coaches to ensure that the students meet the necessary academic requirements for the project proposal.
- They oversee the students' progress and provide guidance throughout the project.
- They will evaluate the students' work in all three objectives as stated in the case study using a predetermined rubric.
- University coaches will be required to provide personal information such as name and email address in order to register for the software. The software will be designed with privacy and security measures in place to protect user information.

2.1.4 Examiners

- They play a crucial role in evaluating students' work in all three assessments: project proposal, design and documentation.
- They review the project materials and provide feedback on the students' understanding of project requirements, technical skills, and overall project quality.
- They evaluate students' work based on predetermined rubrics and provide constructive feedback to help students improve their work.
- Examiners will be required to provide personal information such as name and email address in order to register for the software. The software will be designed with privacy and security measures in place to protect user information.

2.1.5 Coordinators

- Coordinators play a crucial role in managing the project proposal for the entire duration.
- They ensure that students meet project requirements, provide rubrics for assessments, prepare necessary forms, assign university coaches and examiners, and calculate students' marks.
- They work closely with university coaches, industry coaches, and examiners to ensure project progress and goals are met.
- Coordinators will be required to provide personal information such as name and email address in order to register for the software. The software will be designed with privacy and security measures in place to protect user information.

2.2 System Features

Data Engineering FYP System is built to smoothen the communication between students, coaches, examiners, and coordinators, improving the efficiency in supervision and increasing the enhancement of students' performance.

The system features are illustrated in Figure 2.1 below. The detailed description of each module and function is tabulated in Table 2.1.

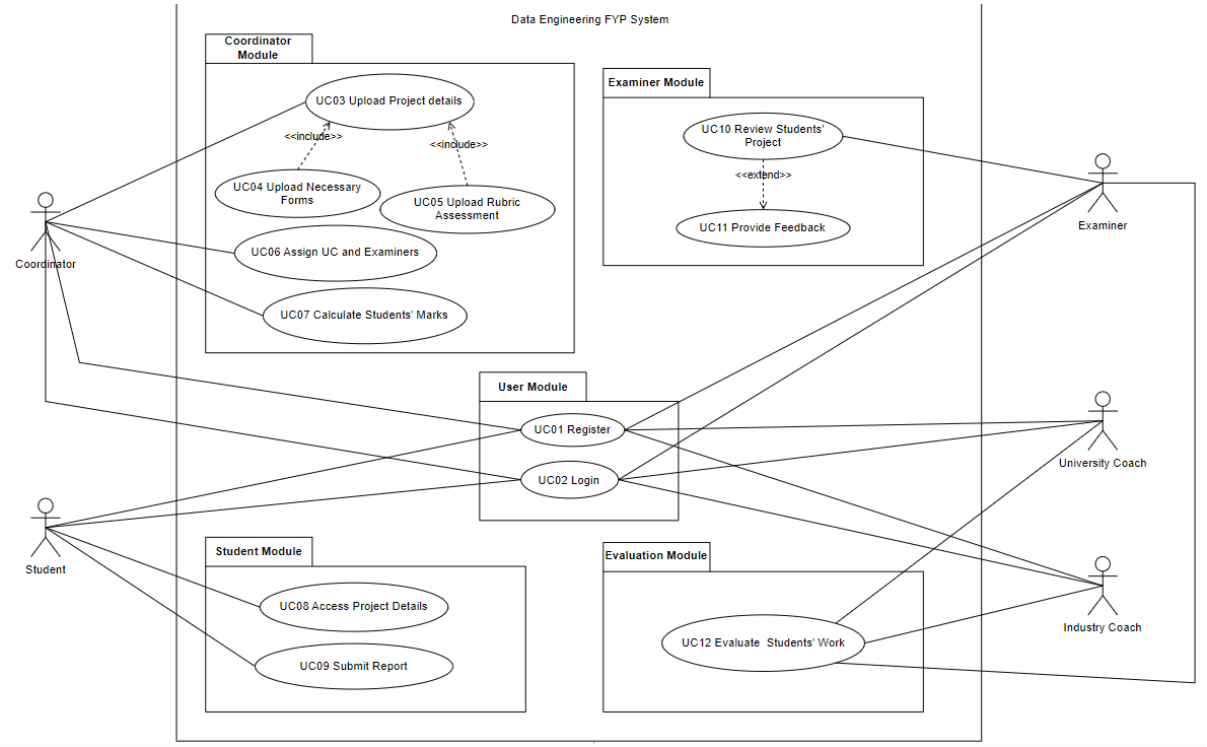

Figure 2.1: Use Case Diagram for Data Engineering FYP System

Table 2.1: Description of Module and Functions for Data Engineering FYP System

Module	***Function***	***Description***
User Module	UC01 – Register	This use case allows students to sign up as a user for the software
User Module	UC02 - Login	This use case allow user to login to the software
Coordinator Module	UC03 - Upload Project Details	This use case allows coordinators to upload project details
Coordinator Module	UC04 -Upload Necessary Forms	This use case allows to upload necessary forms for the project
Coordinator Module	UC05 - Upload Rubric Assessment	This use case allows to upload rubric assessment of the project
Coordinator Module	UC06 - Assign UC and Examiners	This use case allows to assign UC and Examiners to guide and examine students
Coordinator Module	UC07 - Calculate Students' marks	This use case allows to calculate students' total marks for the project
Student Module	UC08 - Access Project Details	This use case allows students to access to their project details
Student Module	UC09 - Submit Report	This use case allows students to submit their project report in the software
Examiner Module	UC10 - Review Students' project	This use case allows examiners to review students' materials
Examiner Module	UC11 - Provide Feedback	This use case allows examiners to provide feedback on students' project progress so improvements can be made
Evaluation Module	UC12 - Evaluate Students' Work	This use case allows examiners, university coach, and industry coach to evaluate students' work.

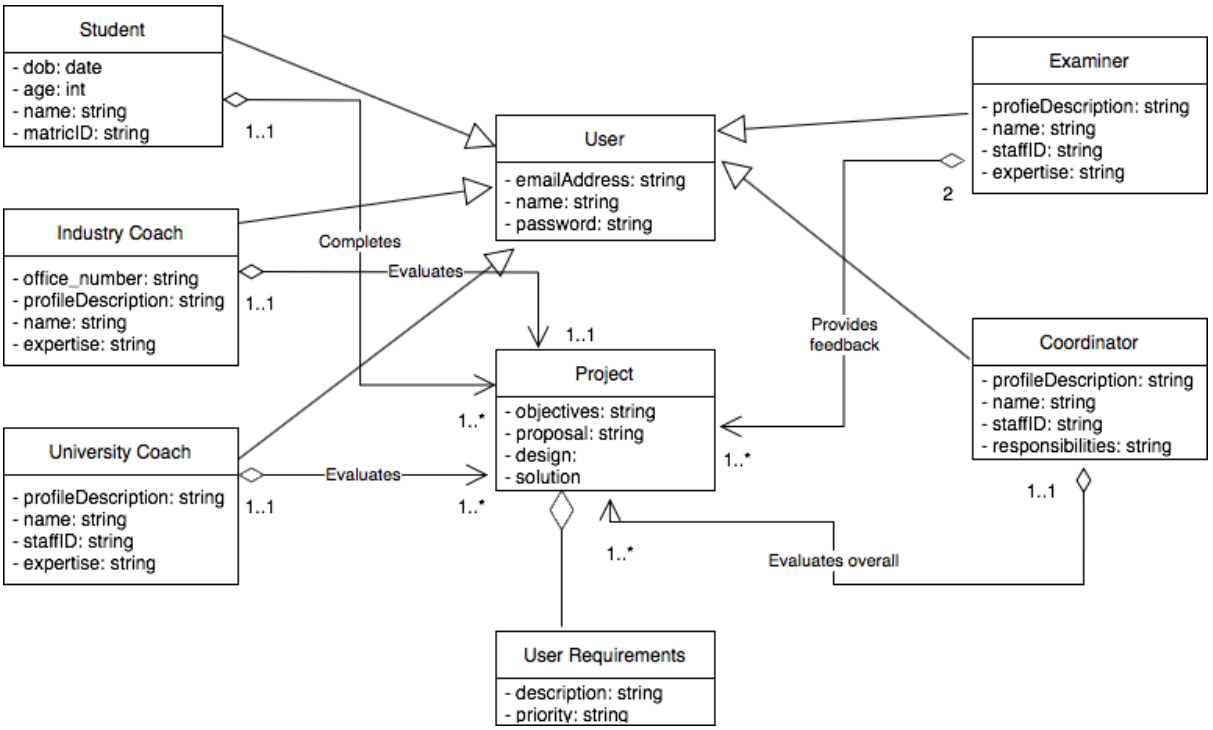

Figure 2.2: Domain Model for Data Engineering FYP System

2.2.1 UC01: Use Case <Register>

Table 2.2: Use Case Description for <Register>

Use case: <Register>
ID: UC01
Actors: Student, University Coach, Industry Coach, Examiner and Coordinator
Preconditions: Has active internet connections to the system.
Flow of events: 1. The use case starts when the user taps on “Create an account” at the sign up page. 2. System displays a registration page that contains a registration form. 3. The user fills up their details. 4. User tap “Sign Up”. 5. System validates the registration form. If the form is invalid, Exception 1 is followed 6. System adds the user account into the user database. 7. System redirect to home page. 8. The use case ends.
Post Conditions: 1. Register as a user in the system.
Exception flow (if any): 1. Invalid registration form 1. System displays an error message.

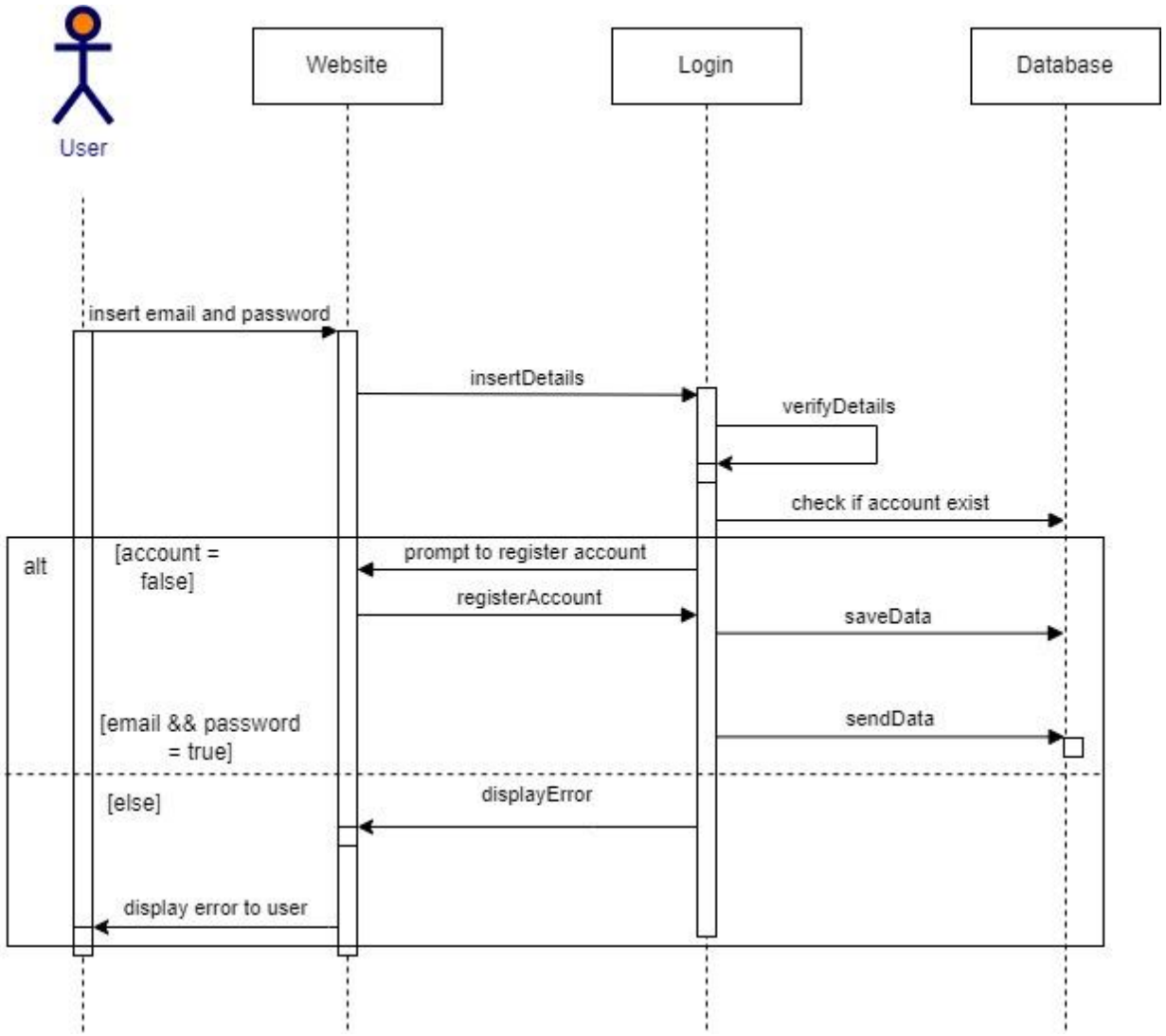

Figure 2.3: Sequence Diagram for <Register>

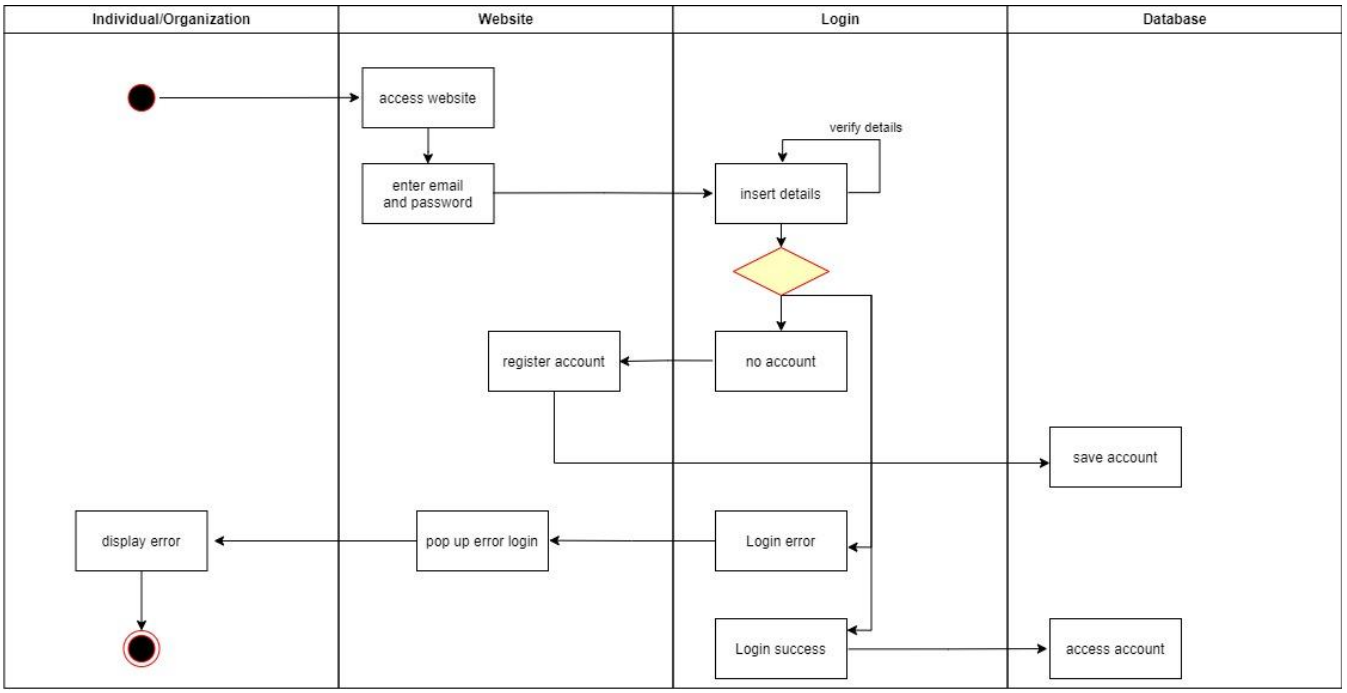

Figure 2.4: Activity Diagram for <Register>

2.2.2 UC02: Use Case <Login>

Table 2.3: Use Case Description for <Login Requirement>

Use case: <Login requirement>
ID: UC02
Actors: 1. Student 2. University Coach 3. Industry Coach 4. Coordinator 5. Examiner
Preconditions: 1. Has an active internet connection to the system for the user's device and already sign up the account.
Flow of events: 1. Starts when the user taps on "Login account" at the login page 2. Include (password verification) 3. If the user email and password is correct 3.1. Login system successfully 4. If the user do not have account 4.1. Register an account 4.2. Include (email validation)
Alternative flow : 1. If the user email and password is incorrect 1.1. Extend (display login error) 1.2. Flow events executed again
Postconditions: 1. User can use the system 2. The system able to record the login information into database

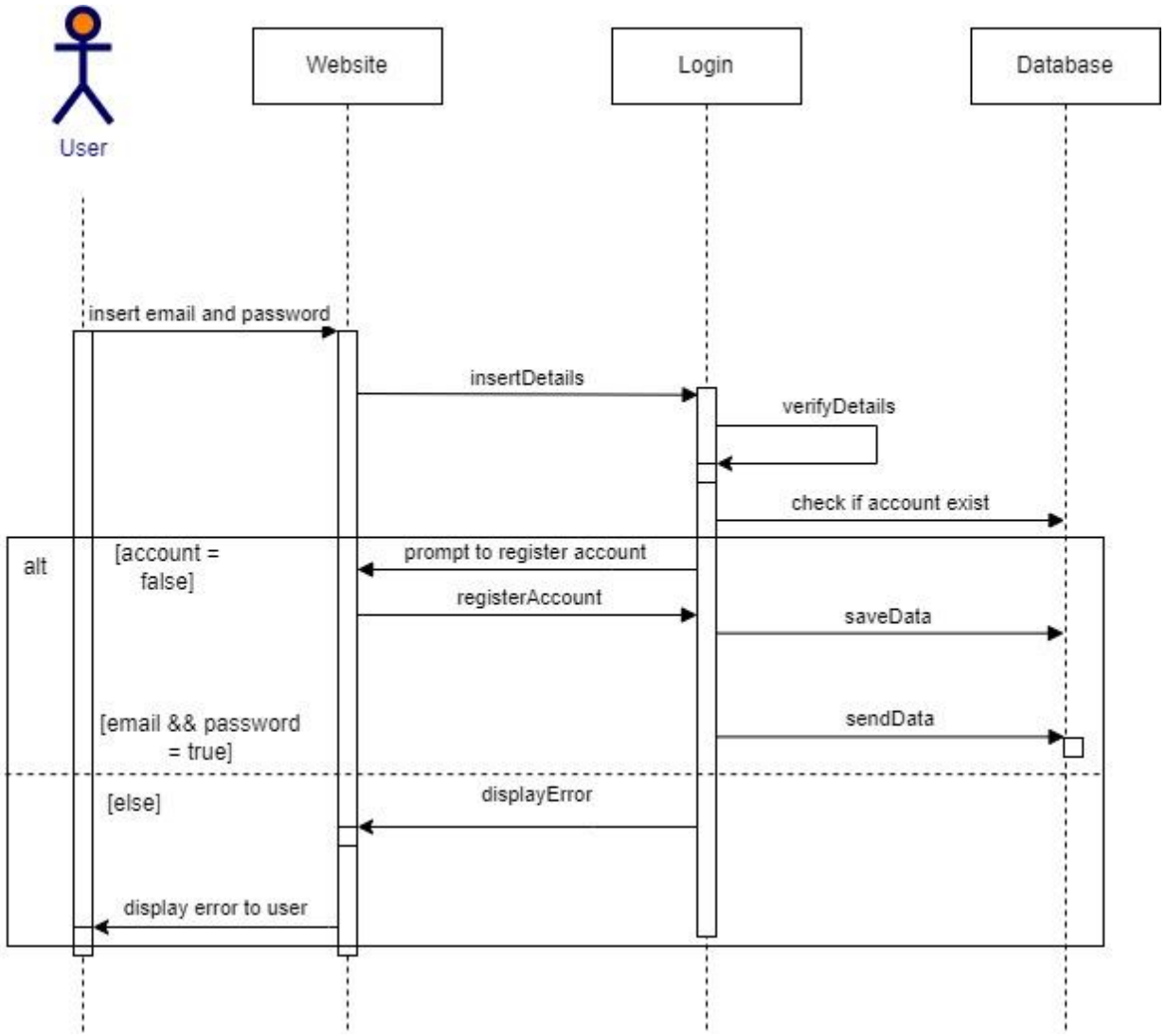

Figure 2.5: Sequence Diagram for <Login>

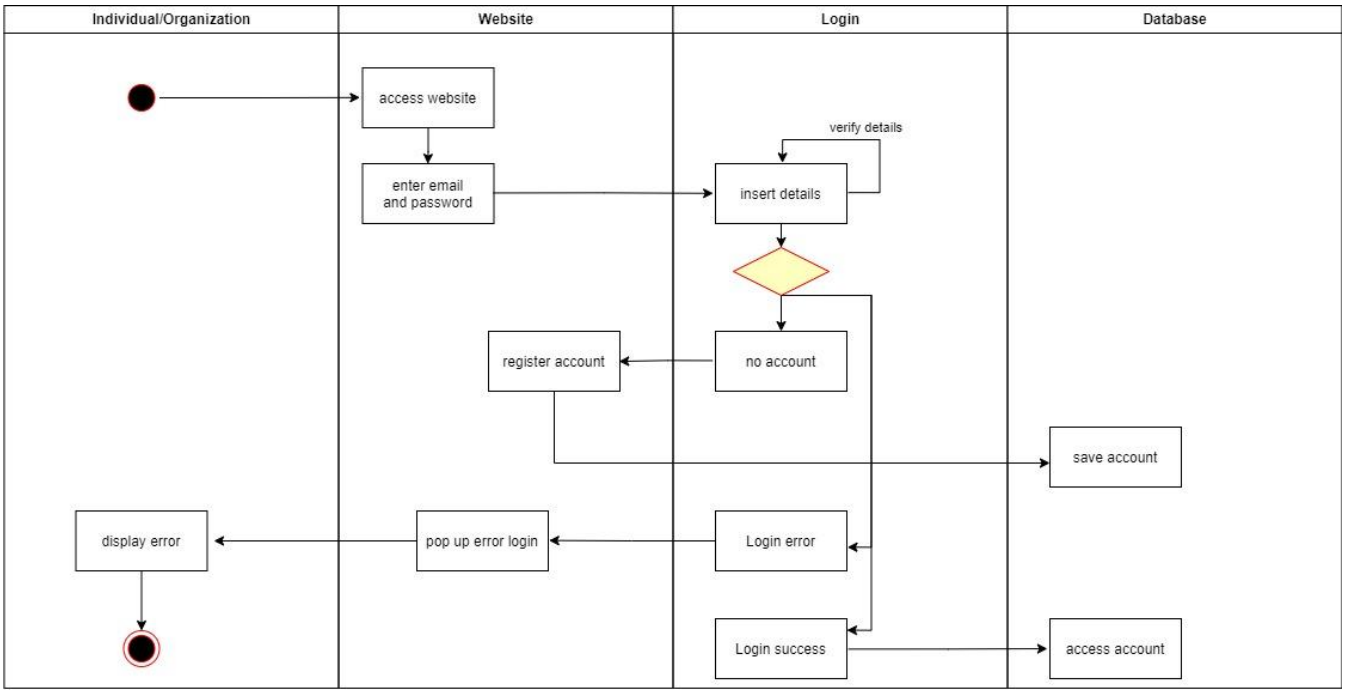

Figure 2.6: Activity Diagram for <Login>

2.2.2 UC03: Use Case <Upload Project Details>

Table 2.4: Use Case Description for <Upload Project Details>

Use case: <Upload Project Details>
ID: UC03
Actors: A1 Staffs
Preconditions: 1. Has active and stable internet connection to the system 2. Has access to the Online FYP Management System 3. Coordinator has already logged in into the system
Flow of events: 1. 1The coordinator logs into the Online FYP Management System 2. The system presents a list of students taking FYP this year. 3. The coordinator clicked upload button to upload documents. 4. Coordinator selects projects details folder from documents. 5. The coordinator uploads the project details on the system.
Alternative flow *n*: 1. If the project details didn't failed to update, reupload the document folder.
Postconditions: 1. Students can view the uploaded project details.

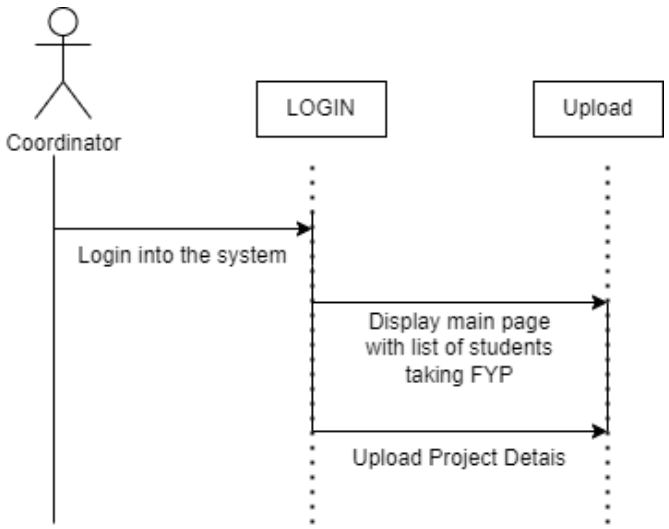

Figure 2.7: Sequence Diagram of Use Case <Upload Project Details>

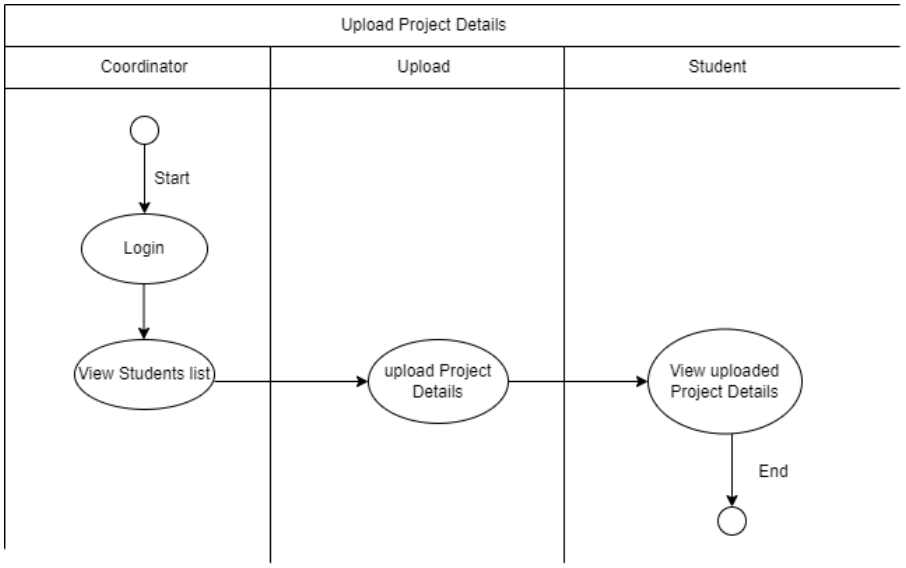

Figure 2.8: Activity Diagram of Use Case <Upload Project Details>

2.2.4 UC04: Use Case <Upload Necessary Forms>

Table 2.5: Use Case Description for <Upload Necessary Forms>

Use case: <Upload Necessary Forms>
ID: UC04
Actors: A1 Staffs
Preconditions: 1. Has active and stable internet connection to the system 2. Has access to the Online FYP Management System 3. Coordinator has already logged in into the system
Flow of events: 1. The coordinator logs into the Online FYP Management System 2. The system presents a list of students taking FYP this year. 3. The coordinator clicked upload button to upload necessary forms. 4. Coordinator selects all the forms from the folder 5. The coordinator uploads all the necessary forms on the system.j
Alternative flow *n*: 1. If the forms didn't failed to update, reupload the forms folder.
Postconditions: 1. Students will receive the uploaded necessary forms.

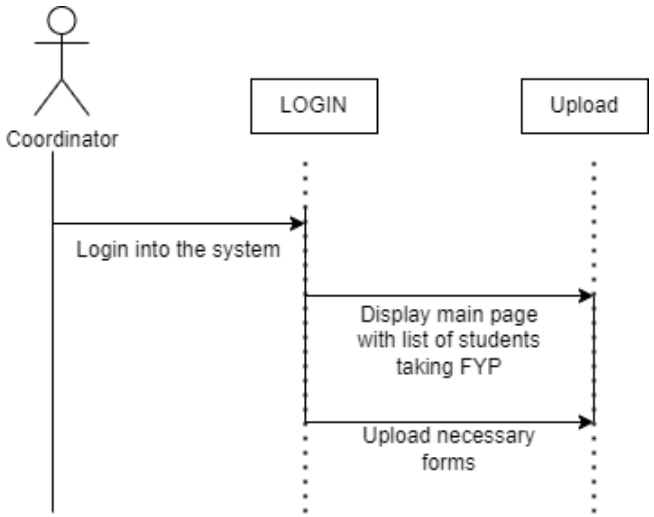

Figure 2.9 : Sequence Diagram of Use case: <Upload Necessary Forms>

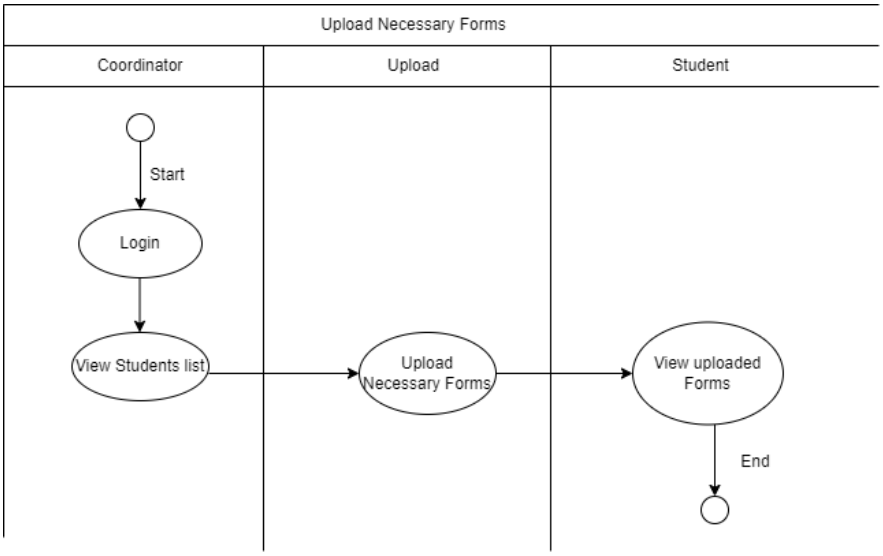

Figure 2.10 : Activity Diagram of Use case: <Upload Necessary Forms>

2.2.5 UC05: Use Case <Upload Rubric Assessment>

Table 2.6: Use Case Description for <Upload Rubric Assessment>

Use case: <Upload Rubric Assessment>
ID: UC05
Actors: A1 Staffs
Preconditions: 1. Has active and stable internet connection to the system 2. Has access to the Online FYP Management System 3. Coordinator has already logged in into the system
Flow of events: 1. The coordinator logs into the Online FYP Management System 2. The system presents a list of students taking FYP this year. 3. The coordinator clicked upload button to upload rubric assessment. 4. Coordinator selects all the forms from the folder. 5. The coordinator uploads all the necessary forms on the system
Alternative flow : 1. If the forms didn't failed to update, reupload the rubric assessment document.
Postconditions: 1. Students will receive the uploaded rubric assessment document.

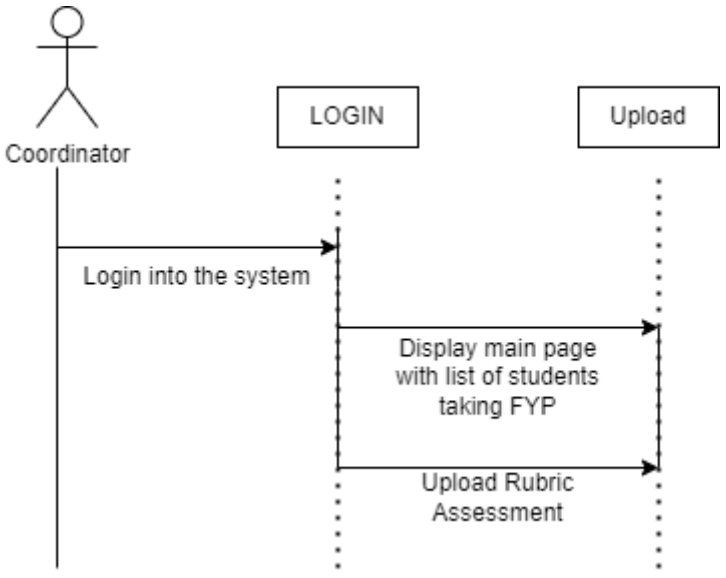

Figure 2.11: Sequence Diagram of Use case: <Upload Rubric Assessment>

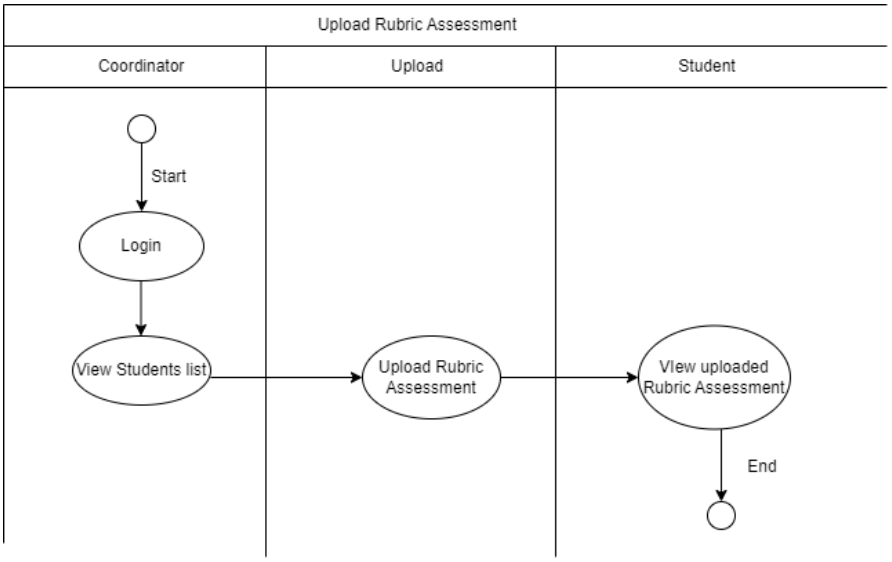

Figure 2.12 : Activity Diagram of Use case: <Upload Rubric Assessment>

2.2.6 UC06: Use Case <Assign UC and Examiners>

Table 2.7: Use Case Description for <Assign UC and Examiners>

Use case: <Assign UC and Examiners>
ID: UC06
Actors: A1 Staffs
Preconditions: 1. Has active and stable internet connection to the system 2. Has access to the Online FYP Management System 3. Coordinator has already logged in into the system
Flow of events: 1. The coordinator logs into the Online FYP Management System 2. The system presents a list of students taking FYP this year. 3. The system presents a list of UC and Examiners. 4. The coordinator assign UC to guide students in their FYP Project. 5. The coordinator assign examiners to examine students in their FYP Project.
Postconditions: 1. Students will receive notification on the UC and examiner that has assigned to them 2. UC and examiners receive notification on the students that they assigned to guide and examine.
Exception flow (if any): 1. Not assigning UC or examiners causes students couldn't learn how to do a proper final year project.

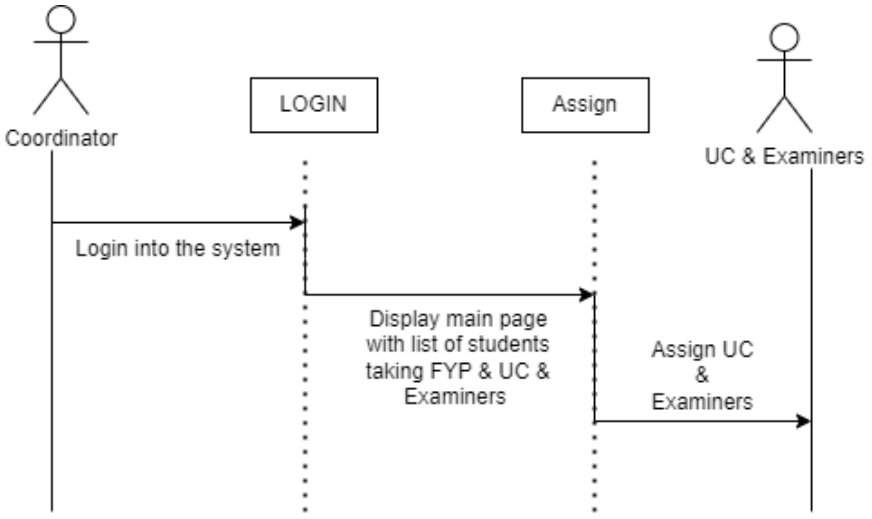

Figure 2.13 : Sequence Diagram of Use case: <Assign UC and Examiners>

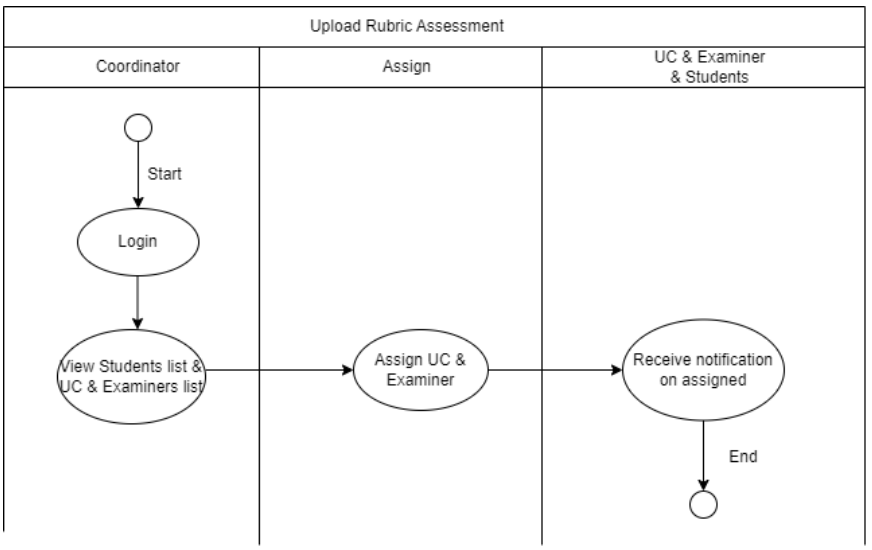

Figure 2.14 : Activity Diagram of Use case: <Assign UC and Examiners>

2.2.7 UC07: Use Case <Calculate Students Marks>

Table 2.8: Use Case Description for <Calculate Students Marks>

Use case: <Calculate Students Marks>
ID: UC07
Actors: A1 Staffs
Preconditions: 1. Has active and stable internet connection to the system 2. Has access to the Online FYP Management System 3. Coordinator has already logged in into the system
Flow of events: 1. The coordinator logs into the Online FYP Management System 2. The system presents a list of students taking FYP this year. 3. The marks given by examiners after evaluating students work will be present beside the students name on the screen. 4. The coordinator calculate the students total marks.
Postconditions: 1. The total calculated marks will be displayed on students profile.

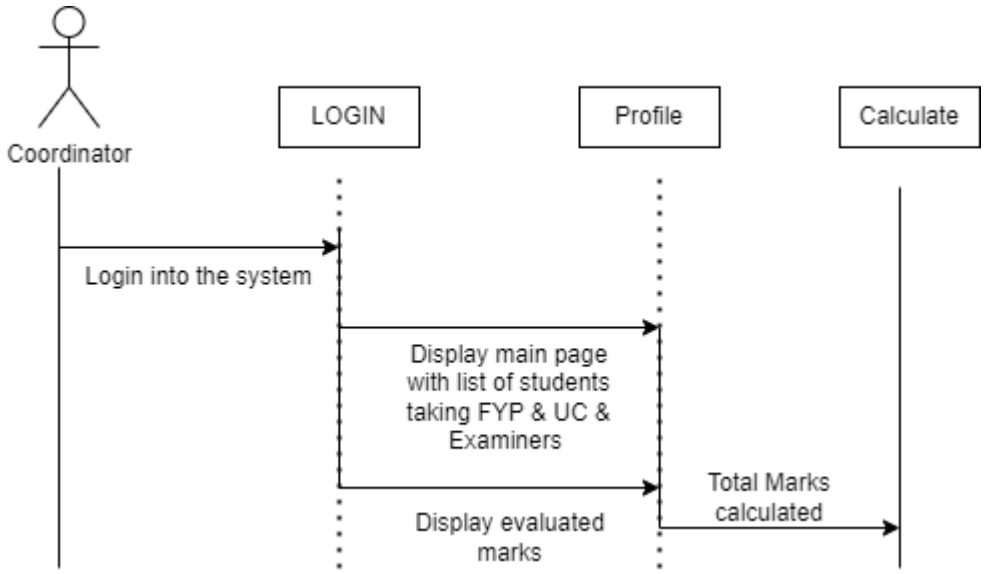

Figure 2.15 : Sequence Diagram of Use case: <Calculate Students Marks>

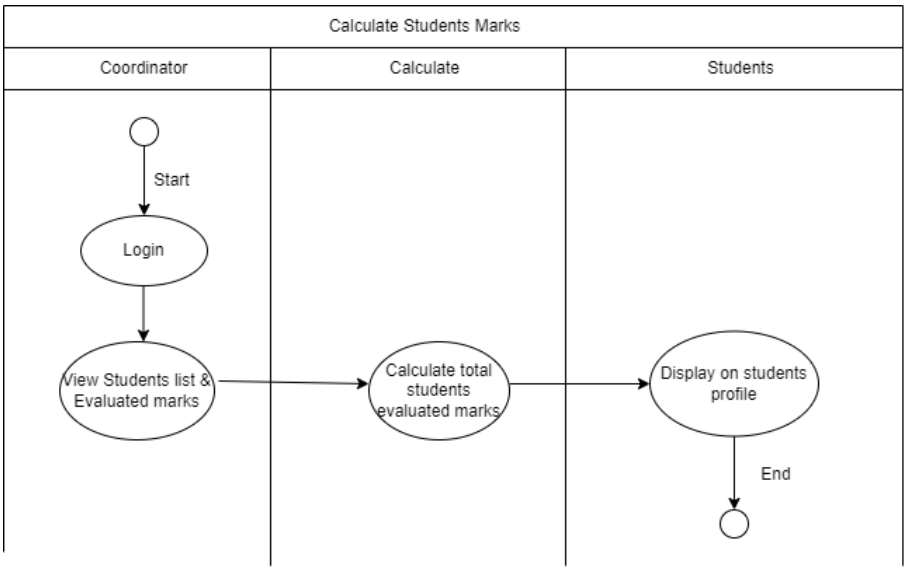

Figure 2.16 : Activity Diagram of Use case: <Calculate Students Marks>

2.2.8 UC08: Use Case <Access Project's Details>

Table 2.9: Use Case Description for <Access Project Details>

Use case: <Access Project's Details>
ID: UC08
Actors: 1. Student
Preconditions: 1. Stable network connection on the user's device. 2. Login to the Data Engineering FYP system.
Flow of events: 1. The use case starts when the students enter the "Access Details" button. 2. The students can view their project's details

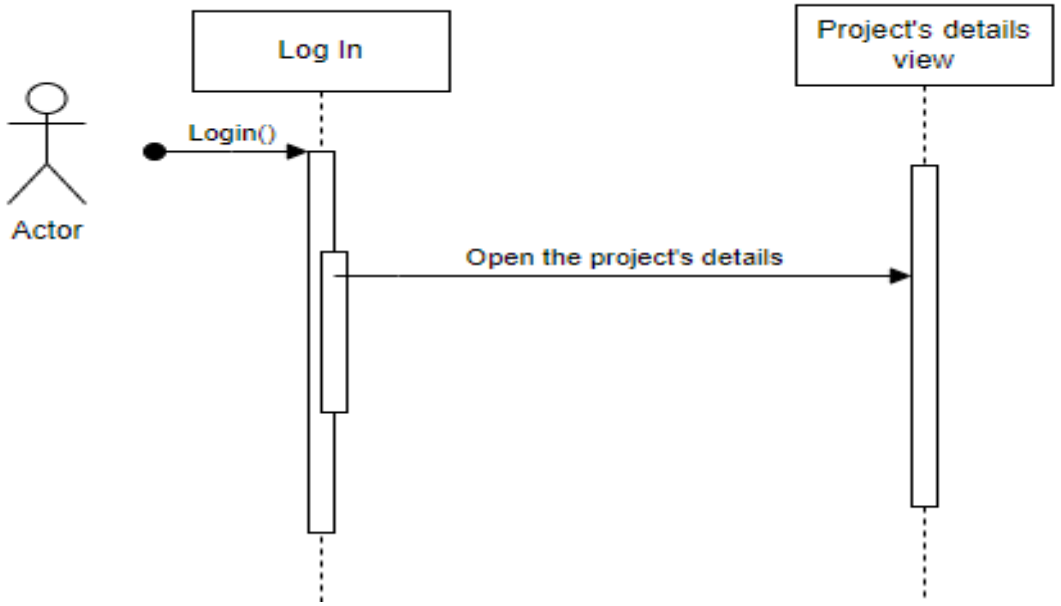

Figure 2.17: Sequence Diagram for <Access Project Details>

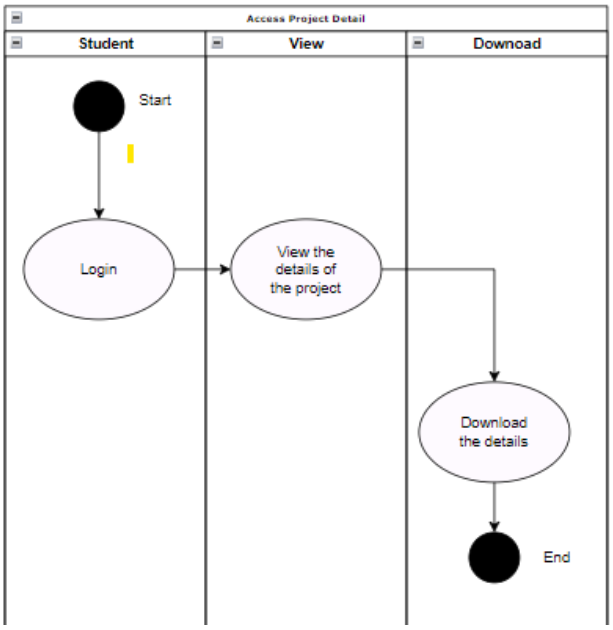

Figure 2.18: Activity Diagram for <Access Project Details>

2.2.9 UC09: Use Case <Submit Report>

Table 2.10: Use Case Description for <Submit Report>

Use case: <Submit Project>
ID: UC09
Actors: 1. Students
Preconditions: The students has validation to use the system
Flow of events: 1. The use case starts when students access the submission interface. 2. The student will have to click to the “Submit Report” button to submit their FYP report 3. To check whether the submission is successful a. Successful completion message pops out 4. Else a. Unsuccessful completion message pops out
Postconditions: 1. Examiner and coordinator can view the project that was uploaded.

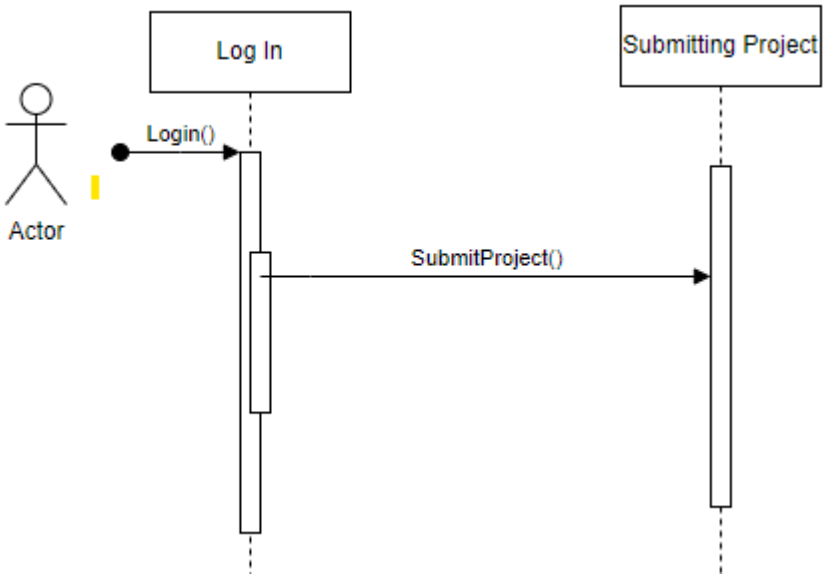

Figure 2.19 : Sequence Diagram for <Submit Report>

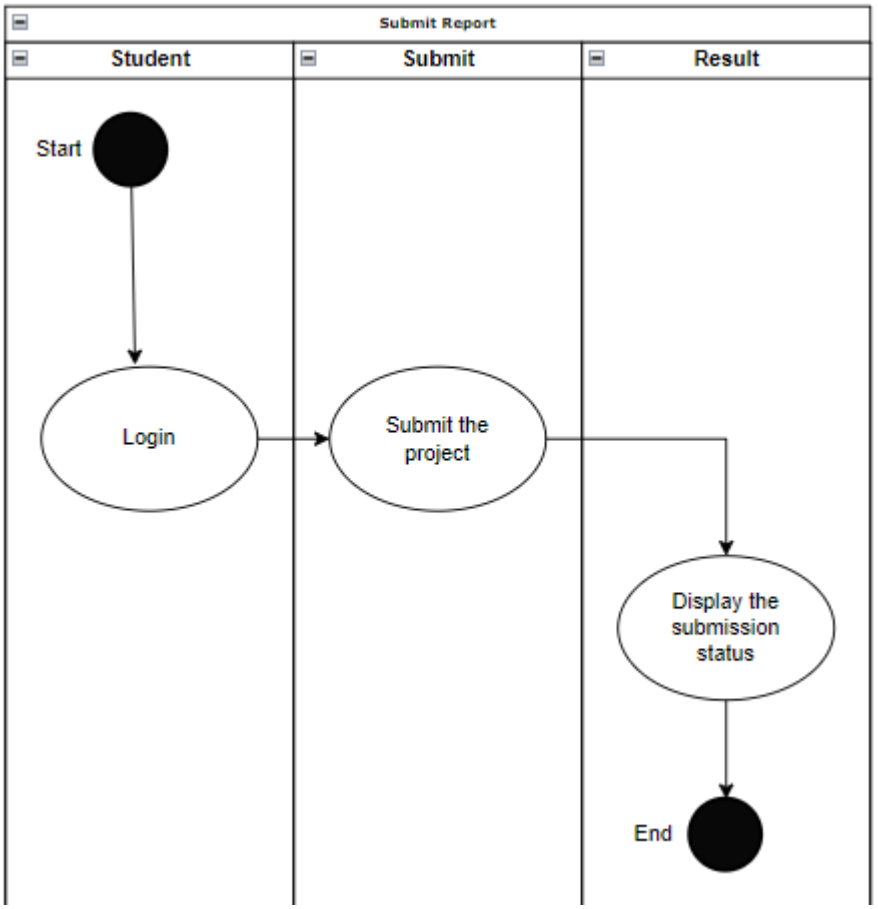

Figure 2.20: Activity Diagram for <Submit Report>

2.2.10 UC10: Use Case <Review Students' Project>

Table 2.11: Use Case Description for <Review Students' Project>

Use case: <Review Students' Project>
ID: UC10
Actors: A1 Examiners
Include: UC11 - Provide Feedback
Preconditions: 1. Has active and stable internet connection to the system 2. Has access to the Online FYP Management System. 3. The student has submitted their project proposal, design documentation, and presentation.
Flow of events: 1. The examiner logs into the Online FYP Management System. 2. The system presents a list of projects awaiting reviews. 3. The examiner selects a project from the list. 4. The system displays the project proposal, design documentation, and presentation materials. 5. The examiner carefully examines each component, considering project requirements, technical details, and overall project quality. 6. The examiner provides feedback, comments, and suggestions for improvement for each component of the project.
Alternative flow: 1. If the project materials submitted by the student are incomplete or do not meet the requirements, the examiner may request additional information or clarification from the student before proceeding with the review process.
Postconditions: 1. The student can use the feedback to improve their work.
Exception flow: 1. Examiners forget to add feedback resulting in students delaying their project improvement.

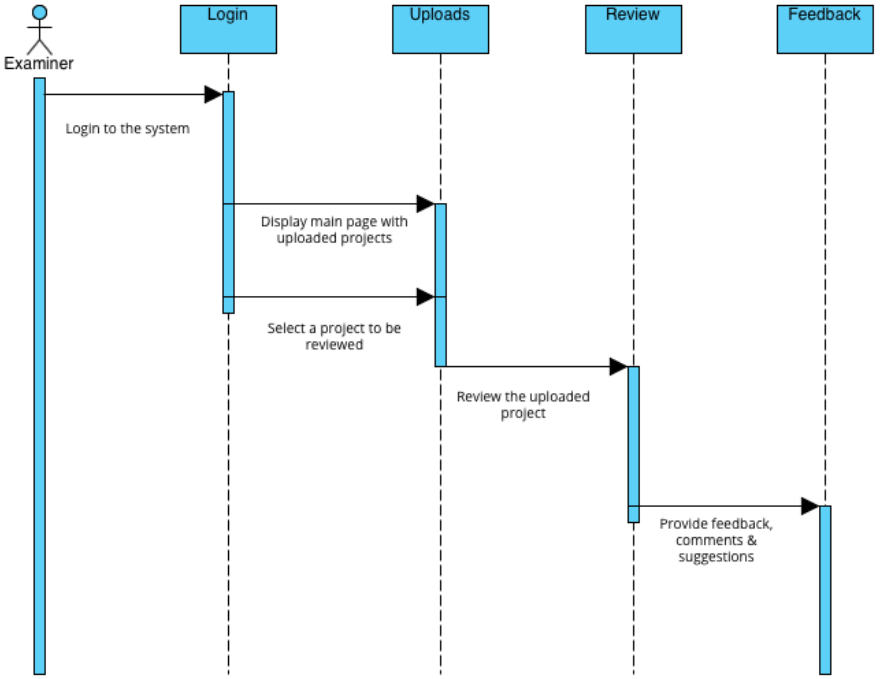

Figure 2.21 : Sequence Diagram for <Examiner Module>

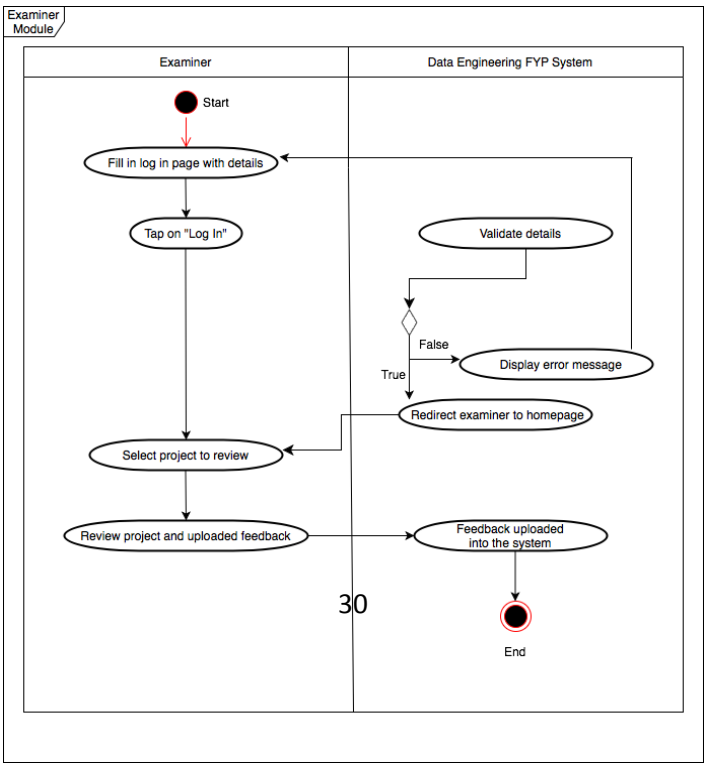

Figure 2.22 : Activity Diagram for <Examiner Module>

2.2.11 UC11: Use Case <Evaluate Students' Work>

Table 2.12: Use Case Description for <Evaluate Students' Work>

Use case: <Evaluate Students' Work>
ID: UC12
Actors: A1 Examiners A2 University Coach A3 Industry Coach
Preconditions: 1. Stable network connection on the user's device. 2. Login to the Data Engineering FYP system. 3. Access project rubric assessment. 4. Get students' work.
Flow of events: 1. Gather documents to evaluate. 2. Thoroughly evaluate students' work based on the rubric. 3. Based on the evaluation, generate an evaluation report. 4. Submit the evaluation report in the system.
Postconditions: 1. Generate an evaluation report from examiners, university coaches, and industry coaches.

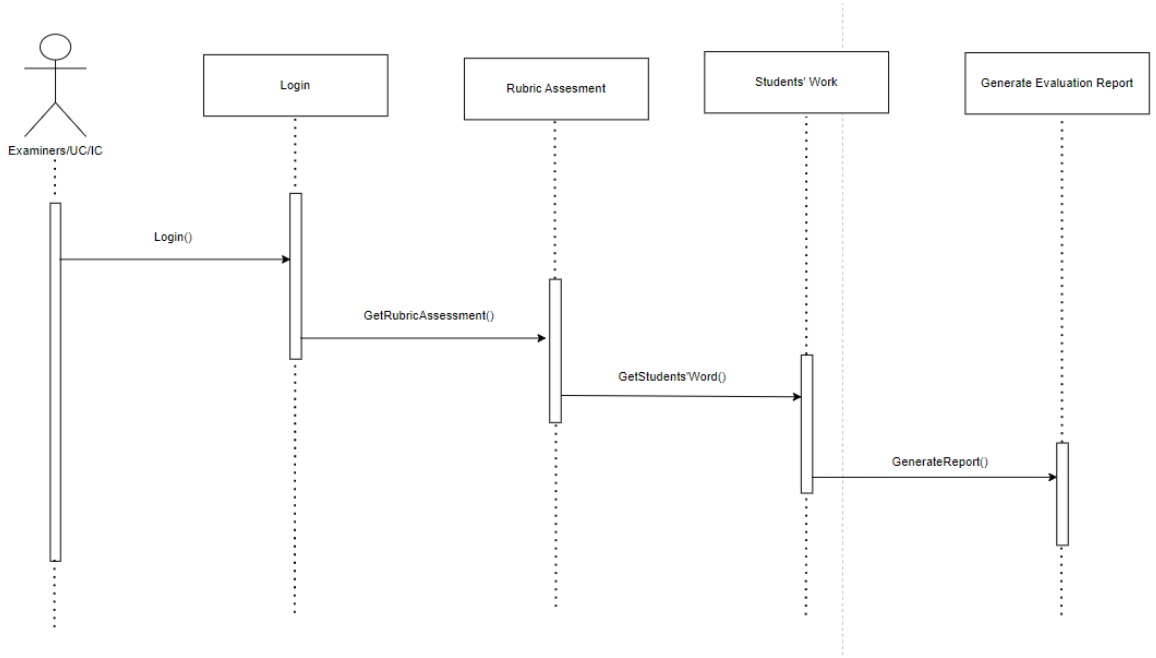

Figure 2.23: Sequence Diagram for <Evaluate Students' Work>

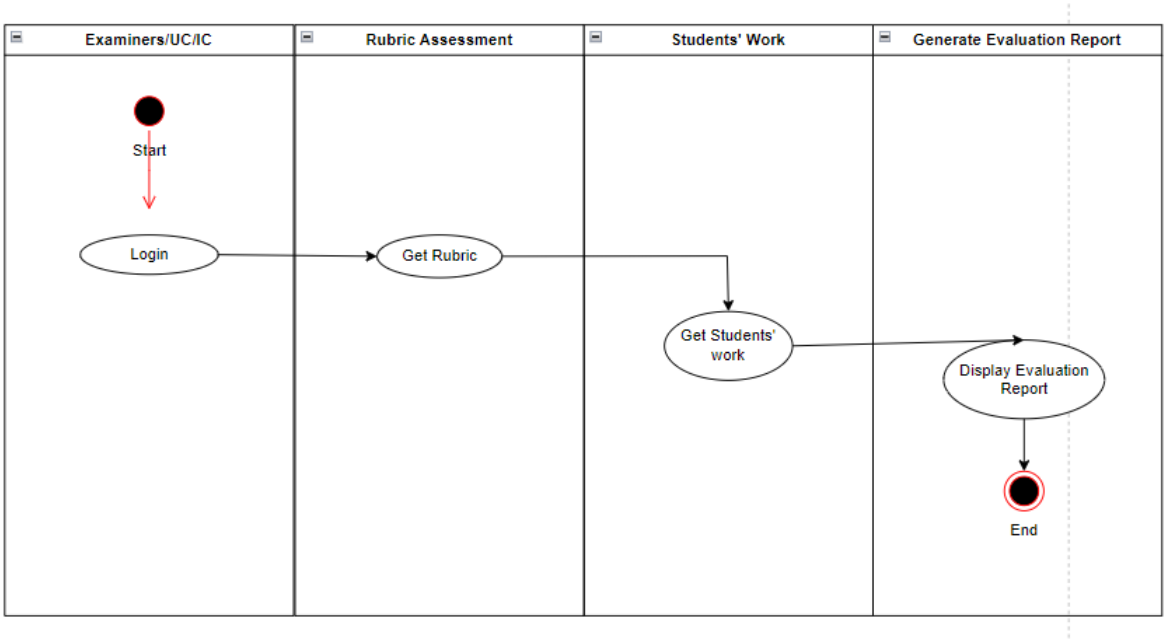

Figure 2.24: Activity Diagram for <Evaluate Students' Work>

2.3 Software System Attributes, Performance and Other Requirements

Non-functional requirements explain how the system should behave rather than what the system should do. These define system properties and constraints of the system such as response time, storage of the system, the system usability, system performance and system reliability. Non-functional requirements may be more critical than functional requirements because if they are not met, the system may be useless. The ISO/IEC/IEEE 29148 standard suggests that non-functional requirements should be specified under three main categories: Software System Attributes, Performance and Other Requirements.

Software system attributes define the overall qualities or characteristics of the software. These attributes are the foundation on which the software is built, and they include the following:

a) **Functionality:** Functionality refers to how well the software meets the specified requirements and performs the intended tasks or functions. This includes its features, capabilities, and the accuracy of its behavior.

b) **Reliability:** Reliability pertains to the software system's ability to consistently and accurately perform its intended functions without errors or failures. It involves aspects such as fault tolerance, error handling, recovery from failures, and the system's ability to handle exceptional conditions.

c) **Usability:** Usability focuses on how easy and efficient it is for users to interact with the software system. This includes factors like the design of the user interface, how intuitive it is, how quickly users can learn to use it, how efficiently they can accomplish tasks, and their overall satisfaction with the system.

d) **Performance:** Performance concerns the speed, responsiveness, and efficiency of the software system. It encompasses attributes such as response time, throughput, resource utilization, scalability, and the optimization of system resources to ensure efficient operation.

e) **Maintainability:** Maintainability refers to how easily the software system can be modified, extended, or repaired over time. This includes attributes like code readability, modularity, documentation, adherence to coding standards, and the availability of tools that facilitate maintenance activities.

f) **Security:** Security involves measures taken to protect the software system from unauthorized access, data breaches, and malicious attacks. This includes attributes such as authentication, authorization, data encryption, secure communication protocols, and the management of vulnerabilities to ensure the system's integrity and confidentiality.

g) **Testability:** Testability refers to how easily the software system can be tested and evaluated to ensure its correctness, reliability, and compliance with requirements. This includes attributes like the ease of generating test cases, the observability and controllability of the system during testing, and the maintainability of test suites.

Performance requirements define the system's capability to respond to user requests and handle data in a timely manner. These requirements include the following:

a) **Response Time:** Response time refers to the duration it takes for a system to react to a user's request or complete a specific action. Enhancing response time involves reducing delays and ensuring the system promptly responds to user interactions. Strategies like employing efficient algorithms, caching, and parallel processing can be utilized to optimize response time.

b) **Throughput:** Throughput measures the system's capacity to handle a certain number of operations or transactions within a given timeframe. It represents how well the system can process multiple requests simultaneously. Improving throughput often entails optimizing resource allocation, minimizing contention, and implementing concurrency techniques to efficiently handle concurrent requests.

c) **Scalability:** Scalability refers to a software system's ability to accommodate increasing workloads or growing user demands without significant performance deterioration. Scalability is crucial to ensure the system can handle higher user traffic or larger data volumes. Horizontal scaling (adding more servers) and vertical scaling (increasing server resources) are common approaches to achieve scalability.

d) **Memory Usage:** Memory usage optimization focuses on reducing the amount of memory consumed by an application. This can be achieved by minimizing unnecessary object creation, efficient memory allocation and deallocation, and utilizing data structures that require less memory overhead.

e) **Network Efficiency:** In distributed systems or client-server architectures, network efficiency plays a critical role in overall performance. Techniques such as minimizing network round-trips, reducing data transfer size, and employing compression or caching mechanisms can enhance network efficiency and decrease latency.

2.4 Design Constraints

i. **Compatibility of Software**

- The system version 1.0 should be compatible with all the other systems in order to respond on time and also be able to handle a certain amount of workload in the given timeframe. It has to work in alignment with the other systems so that it can be responsive to the user's request.

ii. **Operation**

- The unsuccessful logins will display an login error and also will request to re login the account. There will be a forget password option in order for the students or the staff to reset their password and login back to their account. Once logged in there will be a display of logged in signs with a tick on the screen.

iii. **Performance**

- The system must be able to maximize workload in a given time frame, handle a certain number of operations together and also be able to respond to students' requests on time without delay.

iv. **Data Securence**

- The uploaded fyp documents will only be seen by the staff and also students. It won't be leaked or shared to the other students so that there won't be any plagiarism.

3. System Architectural Design

3.1 Architecture Pattern and Rationale

The chosen architecture pattern is the Modern-View-Controller (MVC) Pattern as it is a common design architecture utilized in web-based application systems which is exactly what we need.

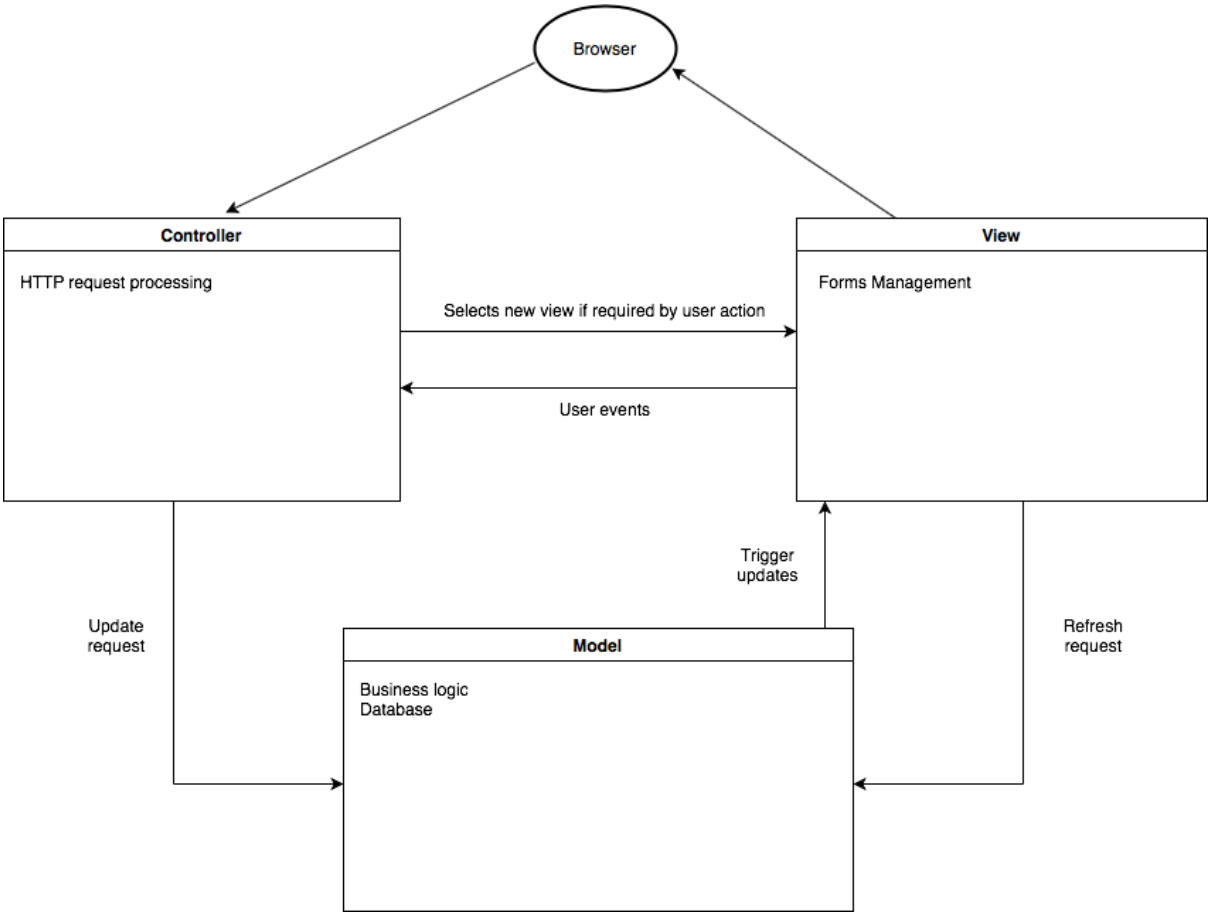

Figure 3.1 Architecture Diagram for Data Engineering FYP System

3.2 Component Model

The four main components of this architectural design are the Login component, Upload component, Submission component, and Assign component. The system's data and associated processes are under the control of the model components. The user must register using their ID or institution email and password before using the login subsystem, which is the initial stage of the procedure. If the user already has an account, they may log in right away and, if wanted, change their password. If the user agrees to the website's cookies, the system will check the user's information to see if it is valid or invalid after the user enters it. If it is valid, the login information will be stored in the database and with the persistence feature so that the user need not enter it again when logging in. Next is Submission where students can submit, re-submit and delete their submission. Then the upload is where the coordinator uploads the rubric. Lastly, Assign is where the coordinator will assign student to their respective Industrial Coach.

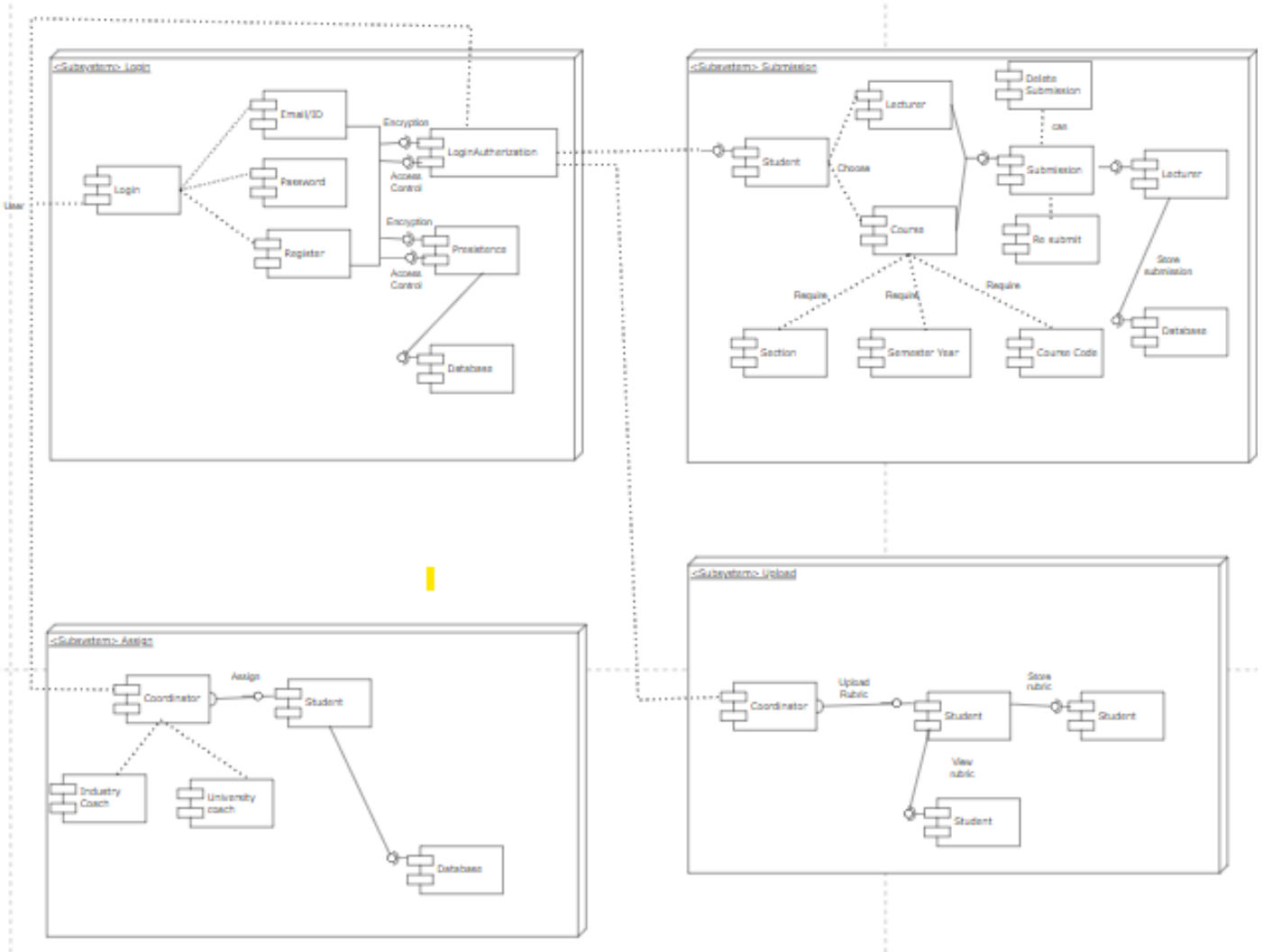

Figure 3.2: Component Diagram of <Data Engineering FYP System>

4. Detailed Description of Components

4.1 Complete Package Diagram

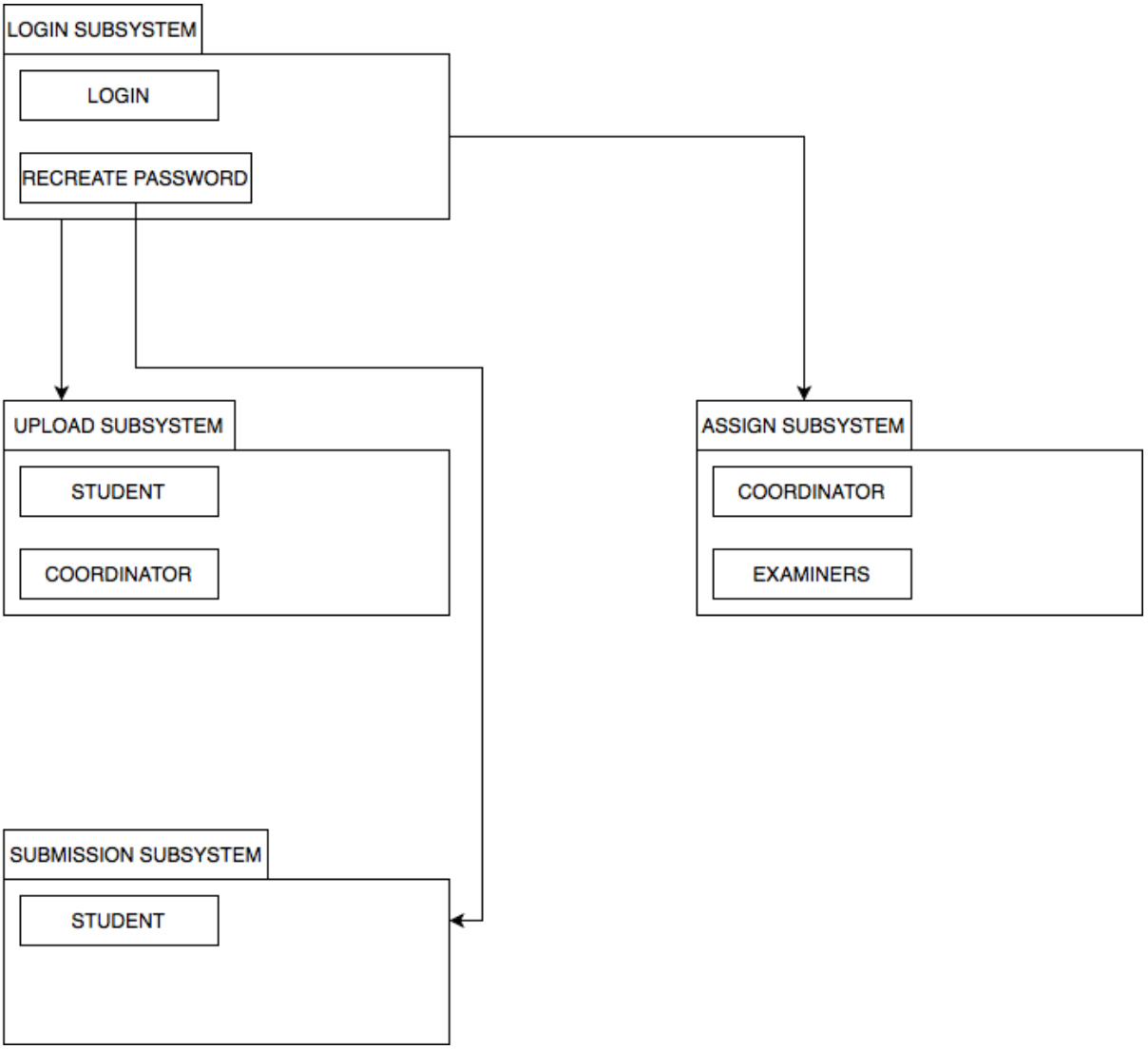

Figure 4.1: Package Diagram for <Data Engineering FYP System>

4.2 Detailed Description

4.2.1 P001: <Login> Subsystem

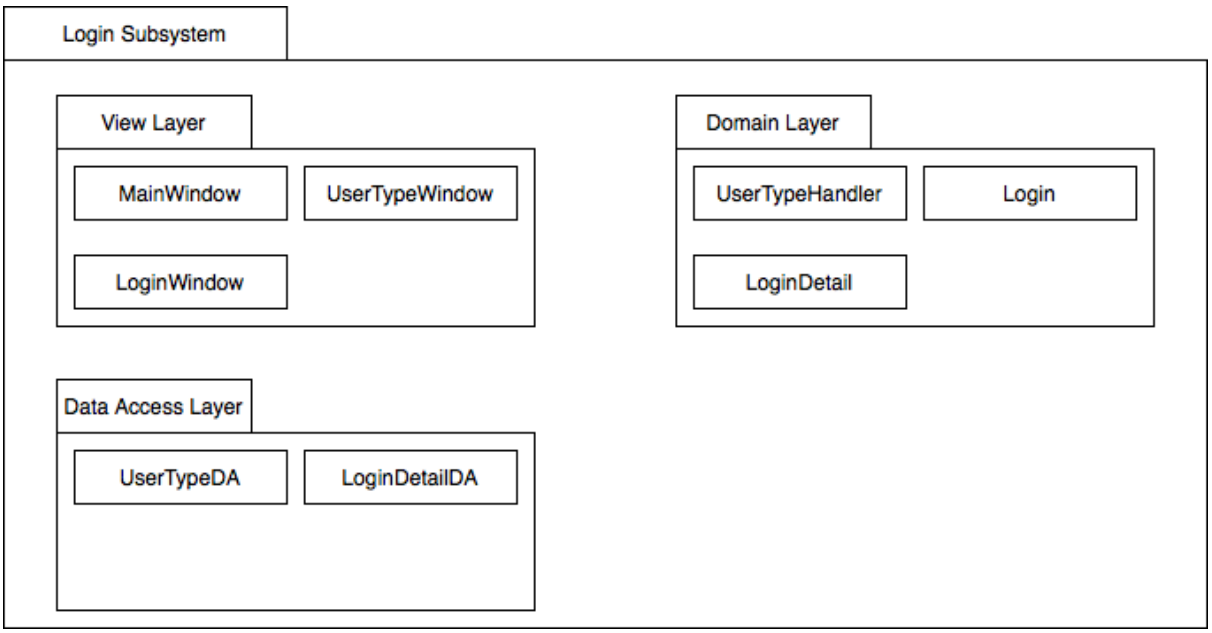

Figure 4.2: Package Diagram for <Login> Subsystem

4.2.1.1 Class Diagram

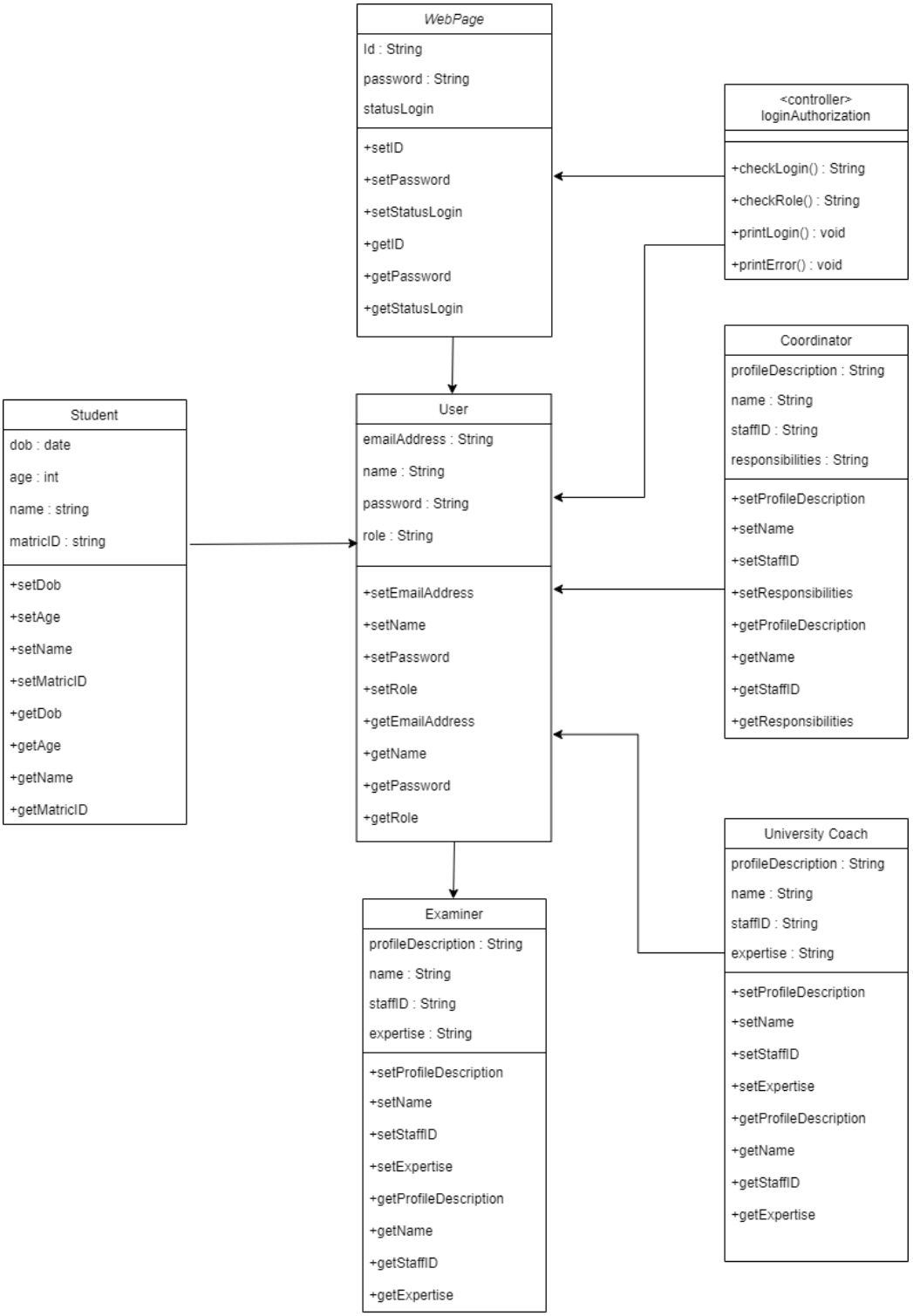

Figure 4.3: Class Diagram for <Login> Subsystem

Entity Name	WebPage
Method Name	CheckLogin()
Input	Called by method from loginAuthorizatiom
Output	Print login verification
Algorithm	1. Start 2. Read value of id from WebPage class 3. IF id exists 3.1 Read password 3.2 IF password exists 3.2.1 setStatusLogin = 1 3.2.2 printLogin() 3.3 ELSE print “Please enter the correct password” 4. ELSE 4.1 PrintError() 4.2 setStatusLogin = 0 5. END

Entity Name	User
Method Name	CheckRole()
Input	Called by method from loginAuthorizatiom
Output	N/A
Algorithm	1. Start 2. Read value of role from User class 3. IF role == Student 3.1 Read password 3.2 setMatricID() 3.3 setName() 4. ELSE IF role == Coordinator 4.1 setStaffID() 5. ELSE IF role == University Coach 5.1 setStaffID() 6. ELSE IF role == Examiner 6.1 setStaffID() 7. ELSE 7.1 Print “Please enter the correct role” 8. END

Entity Name	Login Authorization
Method Name	printLogin(void)
Input	Data from user information
Output	Display user information
Algorithm	1. Start 2. Get id 3. IF id == user id 3.1 Print id, full name, email address, role 4. END

Entity Name	User
Method Name	PrintError(void)
Input	Data from user information
Output	Print login verification
Algorithm	1. Start 2. Read value of id and password from WebPage class 3. IF (! ID && ! password) 3.1 Print “Account does not exist” 3.2 Print “Please create a new account” 4. ELSE 4.1 checkLogin() 5. END

4.2.1.2 Sequence Diagram

SD001: Sequence diagram for <Login> Scenario

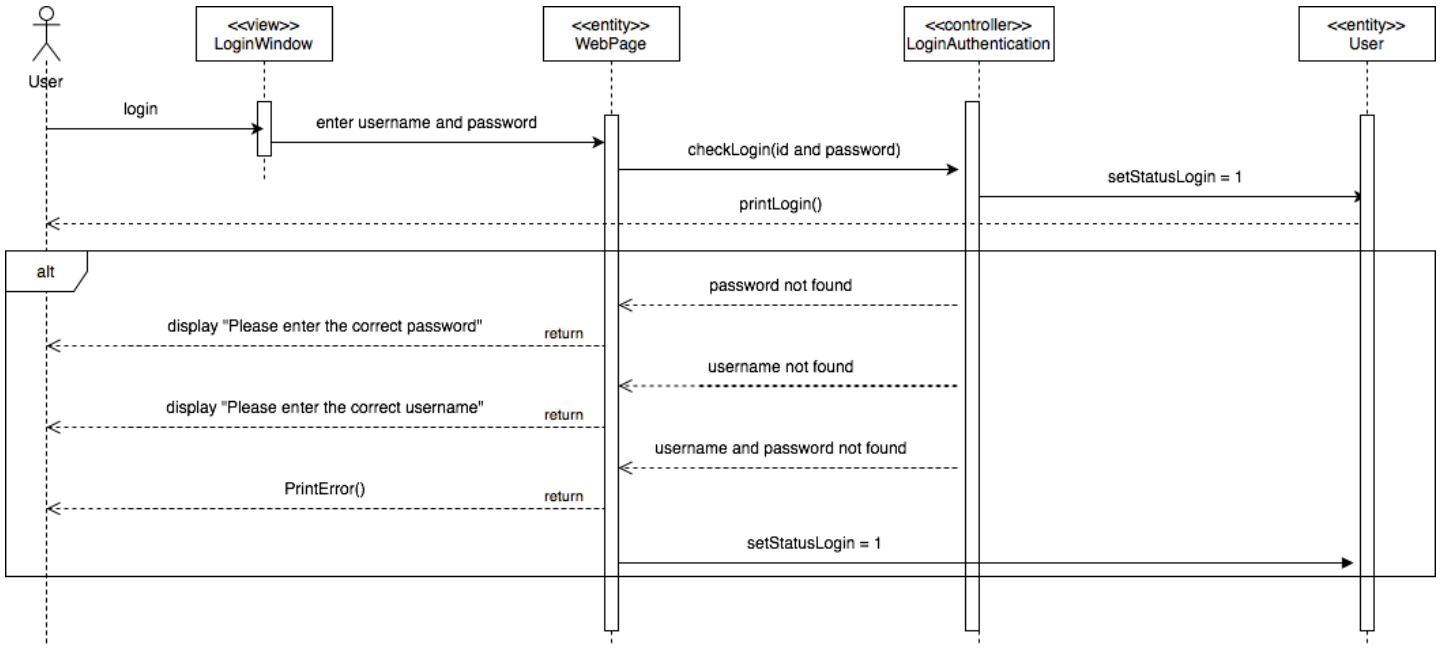

Figure 4.4: Sequence Diagram for <Login> Scenario

4.2.2 P002: Upload Subsystem

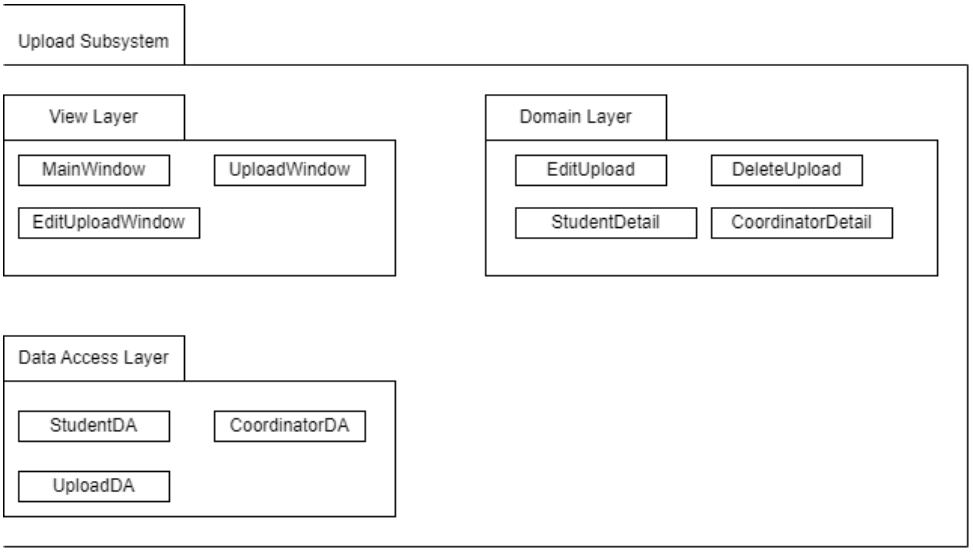

Figure 4.5: Package Diagram for Upload Subsystem

4.2.2.1 Class Diagram

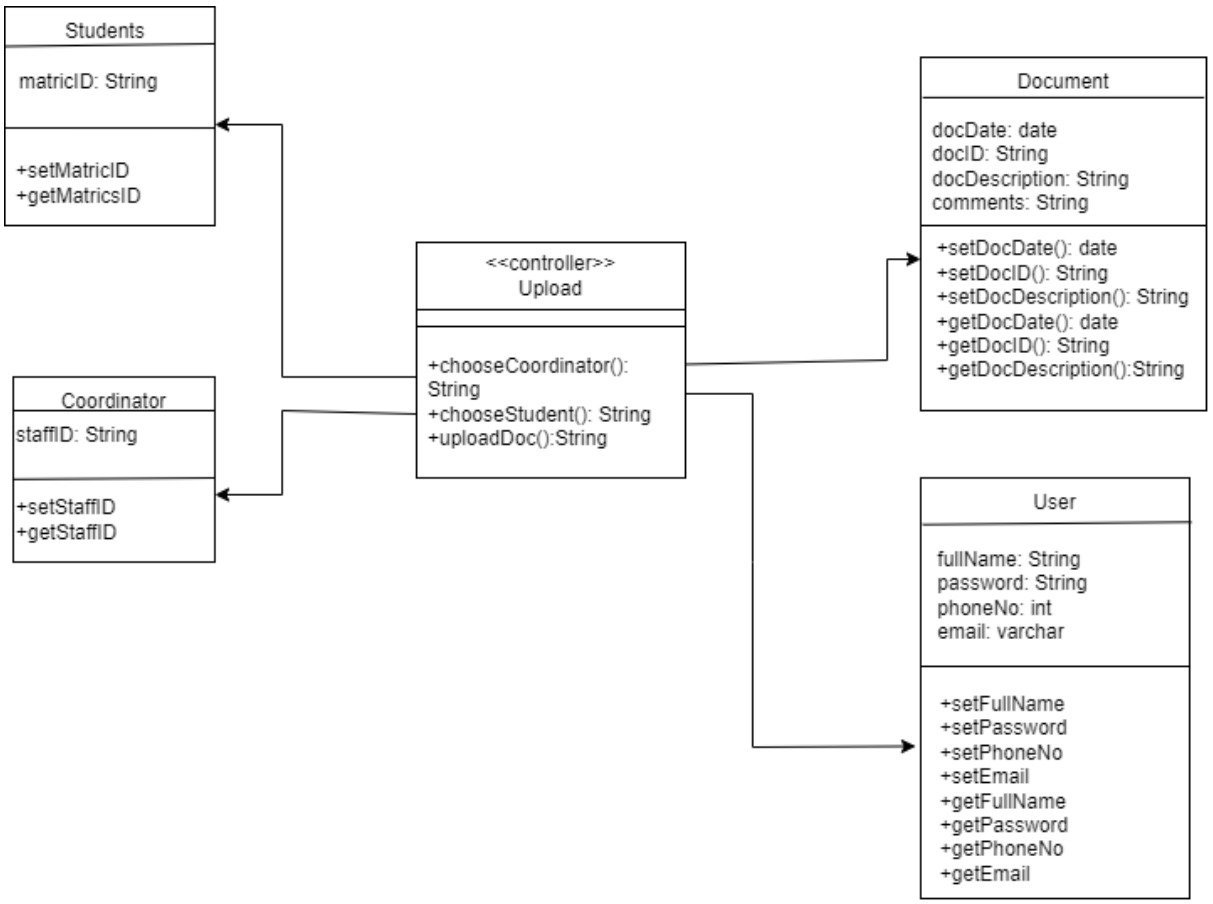

Figure 4.6: Class Diagram for Upload Subsystem

Entity Name	Coordinator
Method Name	chooseCoordinator()
Input	Called by method from upload
Output	Choose the selected Coordinator
Algorithm	1. Start 2. Read ID value entered by the user for staffID 3. IF id == staffID 3.1 Return coordinator's name, email address, phone number 4. ELSE 5. Print "Enter a valid StaffID" 6. END

Entity Name	Student
Method Name	chooseStudent()
Input	Called by method from upload
Output	Choose the selected Student
Algorithm	7. Start 8. Read No value entered by the user for MatricNo 9. IF No == MatricNo 3.1 Return student's name, email address, phone number 10. ELSE 11. Print "Enter a valid MatricNo" 12. END

Entity Name	Document
Method Name	uploadDocument()
Input	Called by method from upload
Output	Update upload database with document information
Algorithm	1. Start 2. Read value for docID, docDescription 3. IF docID == exists a. Upload to upload database 4. ELSE a. Print" Upload Unsuccessfully" 5. End

4.2.2.2 Sequence Diagram

SD002: Sequence diagram for <Upload> Scenario

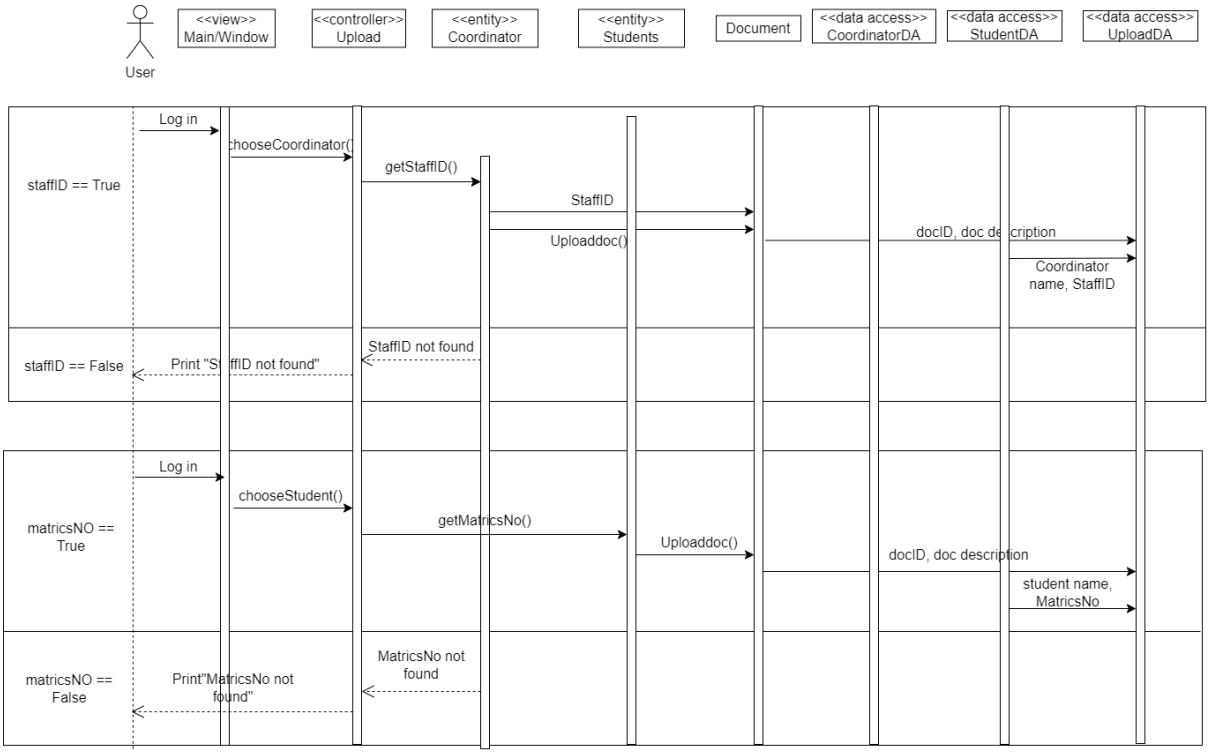

Figure 4.7: Sequence Diagram for <Upload> Scenario

4.2.3 P003: Assign Subsystem

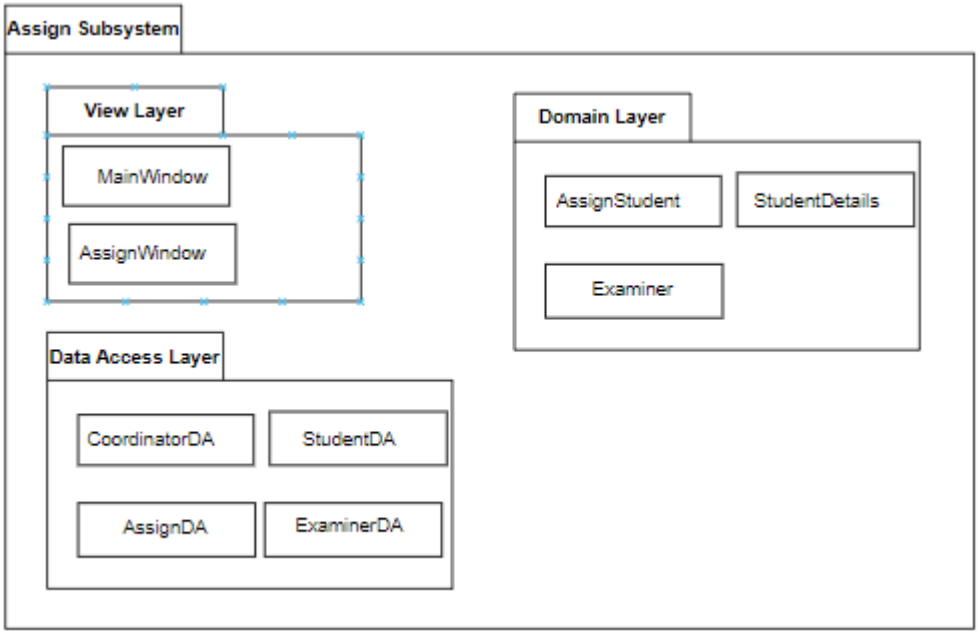

Figure 4.8: Package Diagram for Assign Subsystem

4.2.3.1 Class Diagram

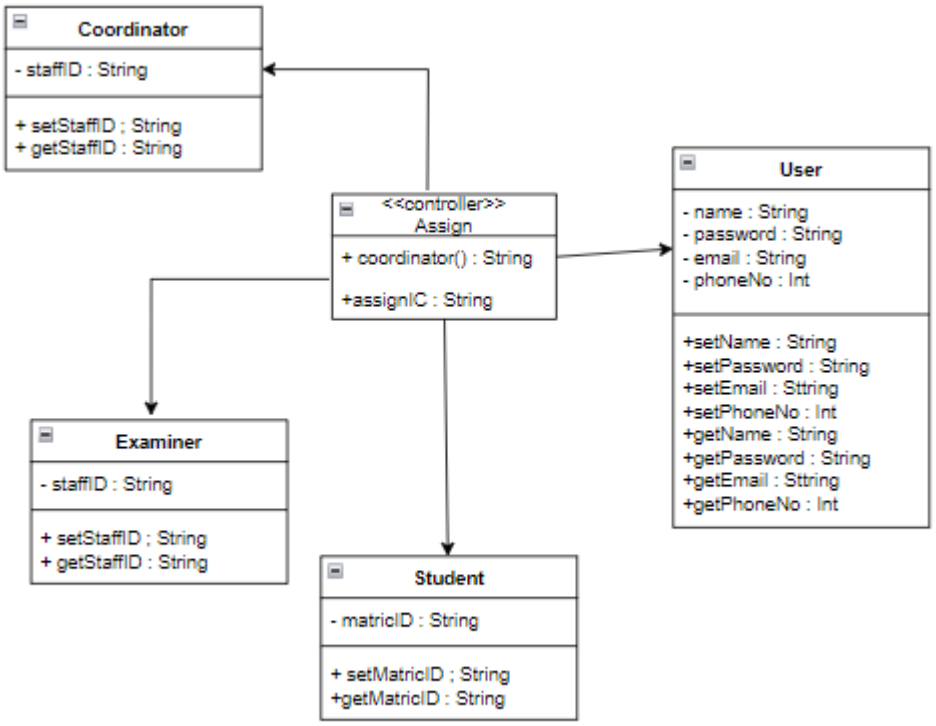

Figure 4.9: Class Diagram for Assign Subsystem

Entity Name	User
Method Name	CheckRole()
Input	Called by method from loginAuthorizatiom
Output	N/A
Algorithm	1. Start 2. Read value of role from User class 3. IF role == Student 3.1 Read password 3.2 setMatricID() 3.3 setName() 4. ELSE IF role == Coordinator 4.1 setStaffID() 5. ELSE IF role == University Coach 5.1 setStaffID() 6. ELSE IF role == Examiner 6.1 setStaffID() 7. ELSE 7.1 Print “Please enter the correct role” 8. END

Entity Name	Coordinator
Method Name	AssignIC()
Input	Called by method from Assign
Output	Student has assigned to their respective Industrial Coach
Algorithm	9. Start 10. Read ID value entered by the user for staffID 11. IF id == staffID 11.1 Return coordinator's name, email address, phone number 11.2. DO assign UC 11.3. DO assign Examiners 12.ELSE 13.Print "Enter a valid StaffID" 14.END

Entity Name	Student
Method Name	chooseStudent()
Input	Called by method from Assign
Output	Choose the selected Student
Algorithm	13. Start 14. Read No value entered by the user for MatricNo 15. IF No == MatricNo 3.1 Return student's name, email address, phone number 16. ELSE 17. Print "Enter a valid MatricNo" 18. END

Entity Name	Examiner
Method Name	chooseExaminer()
Input	Called by method from Assign
Output	Choose the selected Examiner
Algorithm	19. Start 20. Read No value entered by the user for StaffNo 21. IF No == StaffNo 3.1 Return Examiner's name, email address, phone number 22. ELSE 23. Print "Enter a valid StaffNo" 24. END

4.2.3.2 Sequence Diagram

SD003: Sequence diagram for Assign University Coach and Examiner

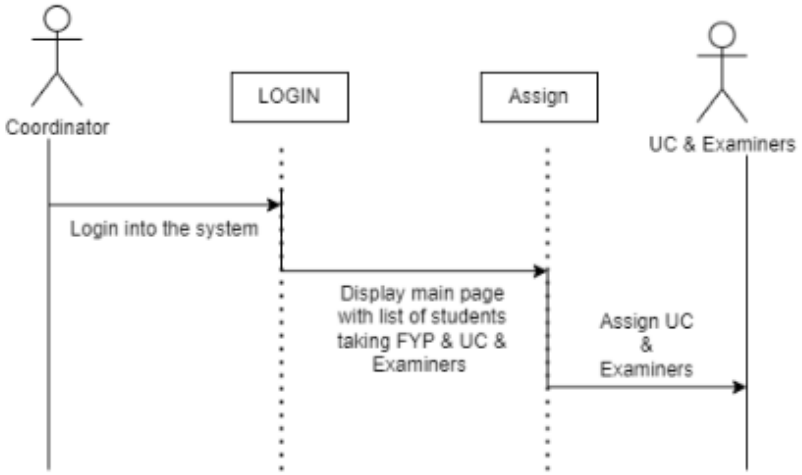

Figure 4.10: Sequence Diagram for <Assign> Scenario

4.2.4 P004: Submission Subsystem

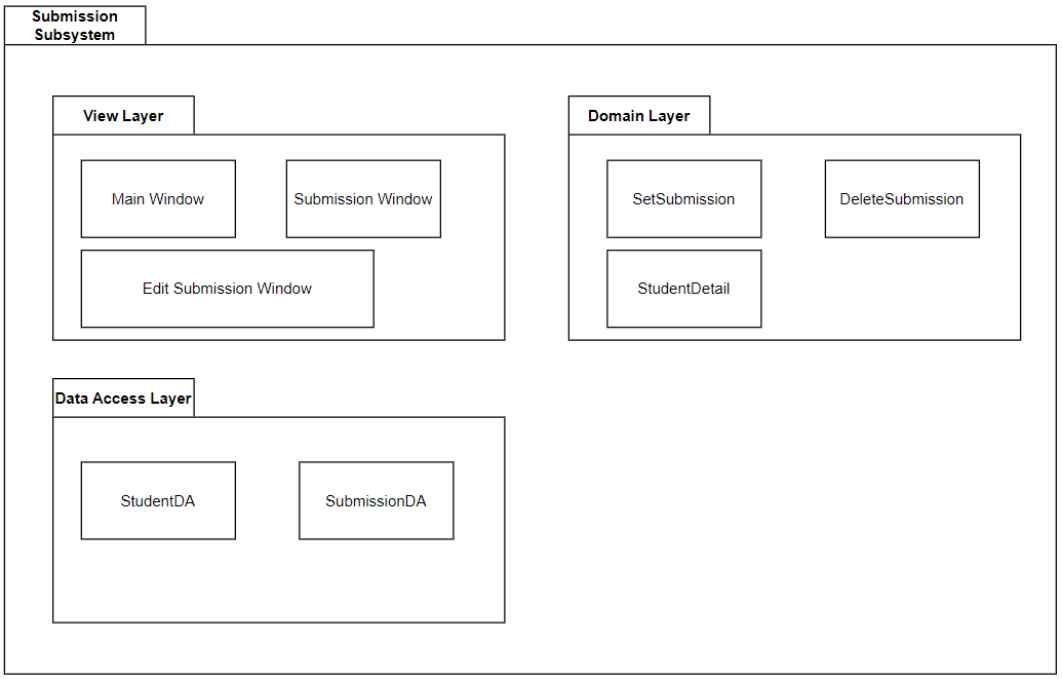

Figure 4.11: Package Diagram for <Submission Subsystem>

4.2.4.1 Class Diagram

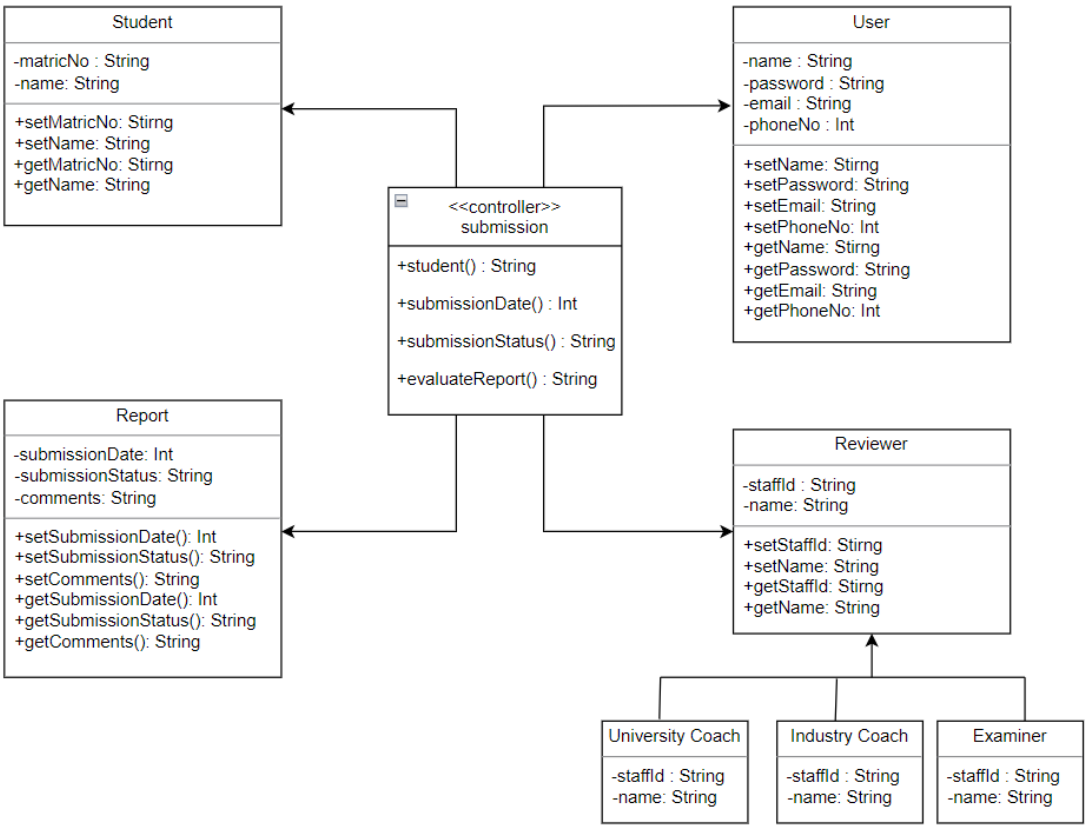

Figure 4.12: Class Diagram for <Submission Subsystem>

Entity Name	Student
Method Name	student()
Input	Call by method submission
Output	Student Profile
Algorithm	1. Start 2. Read id value enter by student for matricNo 2.1 If id == matricNo 2.2 Return student name 3. Else 3.1 Incorrect matriNo 4. End

Entity Name	Report
Method Name	submissionDate()
Input	Call by method submission
Output	Print submission date of the students' report
Algorithm	1. Start 2. View submission page 2.1 If student submit report 2.2 Return submission date 3. Else 3.1 No submission date 4. End

Entity Name	Report
Method Name	submissionStatus()
Input	Call by method submission
Output	Print submission status of the students' report
Algorithm	1. Start 2. View submission page 2.1 If student submit report 2.2 Return submission status : Accepted 3. Else 3.1 Return no submission 4. End

Entity Name	Reviewer
Method Name	evaluateReport()
Input	Call by method submission
Output	Print comments by the reviewer
Algorithm	1. Start 2. Read id value enter by reviewer for staffId 2.1 If id == staffId 2.2 Return staff profile 2.3 View student's report and add comments 3. Else 3.1 Incorrect staffId 4. End

4.2.4.2 Sequence Diagram

SD004: Sequence diagram for <Submission> Scenario

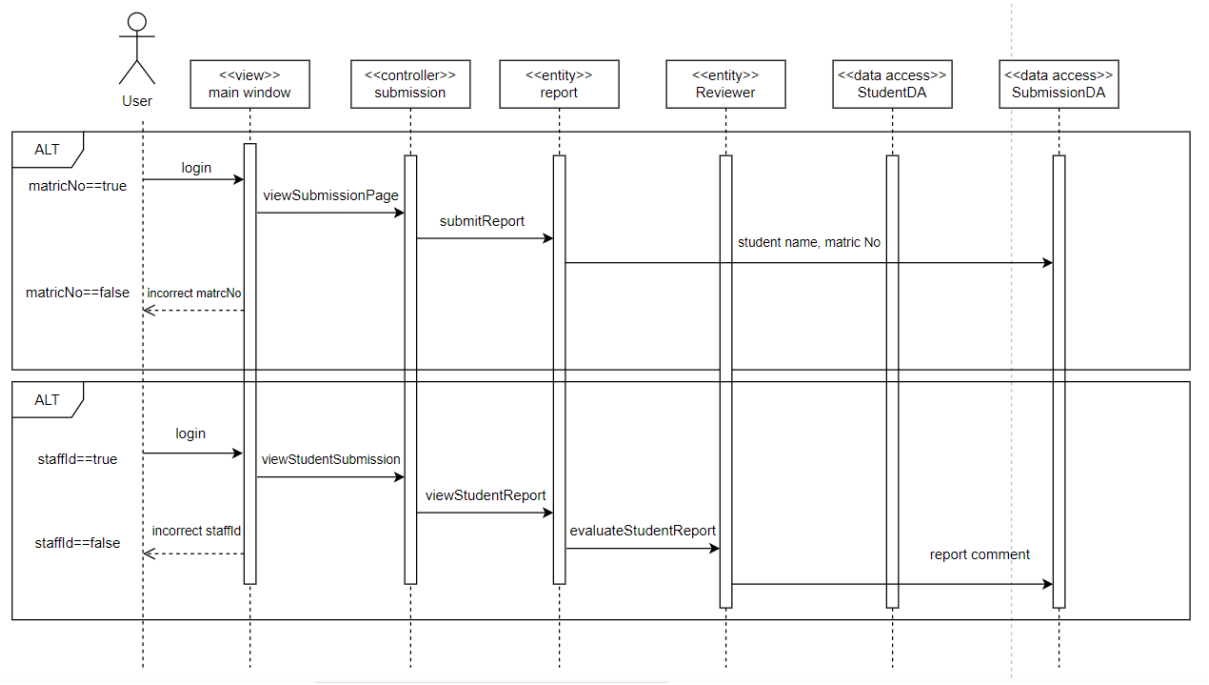

Figure 4.13: Sequence Diagram for <Submission Subsystem>

5. Data Design

5.1 Data Description

The major data or systems entities are stored in a system database divided into coordinator database, student database, examiner database, university coach database, and industry coach database.

Table 5.1: Description of Entities in the Database

No.	Entity Name	Description
1	Coordinator	This entity consists of the data related to the coordinator.
2	Student	This entity consists of the data related to the student.
3	Examiner	This entity consists of the data related to the examiner.
4	University Coach	This entity consists of the data related to the university coach.
5	Industry Coach	This entity consists of the data related to the industry coach.
6	Report	This entity consists of the data related to the students' project report.
7	Reviewer	This entity consists of the data related to the reviewer

5.2 Data Dictionary

5.2.1 Entity: Coordinator

Attribute Name	Type	Description
email	varchar(20)	Email of the coordinator
staffID	varchar(20)	Unique staffID for coordinator
name	varchar(50)	Name of the coordinator
password	varchar(12)	Password to login into the system
phoneNo	varchar(10)	Phone number of coordinator
Username	varchar(20)	Unique identifier for coordinator
role	varchar(20)	Role of the coordinator

5.2.2 Entity: Student

Attribute Name	Type	Description
Email	varchar(20)	Email of the student
MatricNo	varchar(20)	Matric Number of the student
Name	varchar(50)	Name of the student
Password	varchar(12)	Password to login into the system
Username	varchar(20)	Unique identifier for the student
phoneNo	varchar(10)	Phone number of student
role	varchar(20)	Role of the student

5.2.3 Entity: Examiner

Attribute Name	Type	Description
email	varchar(20)	email of the examiner
staffID	varchar(20)	Unique staffID for examiner
name	varchar(50)	Name of the examiner
password	varchar(12)	Password to login into the system
phoneNo	varchar(10)	Phone Number of examiner
username	varchar(20)	Unique identifier for examiner
departmentID	varchar(20)	Unique department ID for examiner
role	varchar(20)	Role of the examiner

5.2.4 Entity: University Coach

Attribute Name	Type	Description
email	varchar(50)	Email of the university coach
staffID	varchar(20)	Unique staffID for university coach
name	varchar(50)	Name of the university coach
password	varchar(12)	Password to login into the system
phoneNo	varchar(10)	Phone number of university coach
username	varchar(20)	Unique identifier for university coach
role	varchar(20)	Role of the university coach

5.2.5 Entity: Industry Coach

Attribute Name	Type	Description

email	varchar(50)	Email of the university coach
staffID	varchar(20)	Unique staff ID for industry coach
name	varchar(50)	Name of the industry coach
password	varchar(12)	Password to login into the system
phoneNo	varchar(10)	Phone number of industry coach
username	varchar(20)	Unique identifier for industry coach
companyID	varchar(20)	unique company id for the industry coach

5.2.6 Entity: Report

Attribute Name	Type	Description
submissionDate	Date	The date of report submitted
submissionStatus	varchar(100)	The status of report submitted
comments	varchar(100)	Comments by reviewer

5.2.7 Entity: Reviewer

Attribute Name	Type	Description
staffId	varchar(20)	Unique staff ID for reviewer
name	varchar(50)	Name of the reviewer

5.2.8 Entity: Upload

Attribute Name	Type	Description

file	varchar(200)	The proposal file
keyword	varchar(50)	The proposal keyword
semester	varchar(10)	The user needed to upload in required semester
status	varchar(50)	The upload status
subject	varchar(50)	Subject Code

5.2.9 Entity: Document

Attribute Name	**Type**	**Description**
date	Date	The date of the document uploaded
description	varchar(100)	The description of the document
documentID	varchar(10)	The document ID

6. Requirements Traceability Matrix

Package Item	Use Case ID	Use Case Description	Sequence Diagram ID	Sequence Diagram Description	Test Case ID
Package 1	UC01	Allows users to sign up to gain access to the software	SD001	Allows users to sign up to gain access to the software	TC001_01
	UC02	Allows users to login to the software	SD001	Allows users to login to the software	TC001_01
Package 2	UC03	Upload Project Details	SD002	Upload Project Details	TC002_01
	UC04	Upload Necessary Forms	SD003	Upload Necessary Forms	TC002_02
	UC05	Upload Rubric Assessment	SD004	Upload Rubric Assessment	TC002_03
Package 3	UC06	Assign University Coach an Examiner	SD005	Assign University Coach an Examiner	TC003_01
Package 4	UC09	Submit Report	SD009	Submit Report	TC004_01
	UC10	Review Project	SD010	Review Project	TC004_02
	UC12	Evaluate Project	SD012	Evaluate Project	TC004_03
	UC07	Calculate Marks	SD007	Calculate Marks	TC004_04

Table 6.1: RTM for <Data Engineering FYP System>

7. Test Cases

7.1 TC001: Test <Login> Subsystem

This test contains the following test cases:

(a) TC001_01: Test <Login> Scenario (SD001)

7.1.1 TC001_01: Test <Login> Scenario (SD001)

This test contains the following scenarios:

(a) TC001_01_01: Test normal flow of <Login> (SD001)

(b) TC001_01_02: Test exception flow of <Login> (SD001)

Table 7.1.1.1: TC001_01_01 - Test normal flow of <Login> (SD001)

Test Case ID	TC001_01	**Test Case Description**	Test the Login Functionality in the Data Engineering FYP System		
Created By	Thuvaaritha	**Reviewed By**		**Version**	

QA Tester's Log					
Tester's Name		**Date Tested**		**Test Case (Pass/Fail/Not Executed)**	

S #	Prerequisites:
1	User not logged in
2	User data exists in database
3	
4	

S #	Test Data
1	Username = mg12345
2	Password = df12@434c
3	
4	

Test Scenario	Verify on entering valid username and password, the user can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to Data Engineering FYP Website	Site should open		
2	Enter Username & Password	Credentials can be entered		
3	Click Login	User is logged in		
4				

Table 7.1.1.2: TC001_01_01 - Test exception flow of <Login> (SD001)

Test Case ID	TC001_01	**Test Case Description**	Test the Login Functionality in the Data Engineering FYP System		
Created By	Thuvaaritha	**Reviewed By**		**Version**	

QA Tester's Log					
Tester's Name		**Date Tested**		**Test Case (Pass/Fail/Not Executed)**	

7.2. TC002: Test Uploading Subsystem

This test contains the following test cases:

(a) TC002_01: Test Upload Project Details (SD002)

(b) TC002_02: Test Upload Necessary Forms (SD003)

(c) TC002_03: Test Upload Rubric Assessment (SD004)

7.2.1 TC002_01: Test Upload Project Details (SD002)

This test contains the following scenarios:

(a) TC002_01: Test normal flow of Upload Project Details (SD002)

Test Case ID	TC002_01	Test Case Description	Test the Upload Project Details Functionality		
Created By	Maathuree	Reviewed By		Version	

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	User is logged in as Lecturer
2	Final list of FYIP Students
3	
4	

S #	Test Data
1	Sample Documents (eg. Project paper, Templates)
2	
3	
4	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Select upload	Display upload page		
2	Select saved project details	Page refresh with selected list of project details		
3	Select Confirm to upload	Project Details will be uploaded and displayed on screen		

Table 7.2.1.1: TC002_01 - Test normal flow of Upload Project Details (SD002)

(b) TC002_02: Test normal flow of Upload Necessary Forms (SD003)

Test Case ID	TC002_01	Test Case Description	Test the Upload Necessary Forms Functionality		
Created By	Maathuree	Reviewed By		Version	

QA Tester's Log

Tester's Name		Date Tested	20 June	Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	User is logged in as Lecturer
2	List of FYIP students
3	
4	

S #	Test Data
1	Sample forms (eg. Personal Details, VIVA application form)
2	
3	
4	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Select upload	Display upload page		
2	Select saved forms	Page refresh with selected list of forms		
3	Select Confirm to upload	Forms will be uploaded and displayed on screen		

Table 7.2.1.3: TC002_03 - Test normal flow of Upload Necessary Forms (SD003)

(c) TC002_03: Test Upload Rubric Assessment (SD004)

Test Case ID	TC002_01	Test Case Description	Test the Upload Rubric Assessment Functionality		
Created By	Maathuree	Reviewed By		Version	

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	User is logged in as Lecturer
2	Project Details been uploaded
3	
4	

S #	Test Data
1	Sample documents (eg. Marking Scheme sheet)
2	
3	
4	

Test Scenario

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Select upload	Display upload page		
2	Select saved rubric assessment documents	Page refresh with selected rubric assessment documents		
3	Select Confirm to upload	Documents will be uploaded and displayed on screen		
4				

Table 7.2.1.3: TC002_03 - Test normal flow of Upload Rubric Assessment (SD004)

7.3 TC003: Test Assign Subsystem

This test contains the following test cases:

(a) TC003_1: Test Assign University Coach and Examiner (SD003)

7.3.1 TC003_01: Test Assign University Coach and Examiner (SD005)

This test contains the following scenarios:

(a) TC003_01_01: Test normal flow of Assign University Coach and Examiner (SD005)

Test Case ID	TC003_01	Test Case Description	Test the Assign University Coach and Examiner		
Created By	Nasrul Amin	Reviewed By		Version	

QA Tester's Log					
Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	User is logged as coordinator
2	Has access to the Online FYP Management System

S #	Test Data
1	StaffID = nas123
2	Pass = 2f3er42

Test Scenario	Verify on assigning university coach and examiner

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	The coordinator can have to access the assigning page	Assigning page could be access		
2	The system will display the students' lists	Students list displayed		
3	The system will display the examiners lists	Students list displayed		
4	Coordinator assign examiner to the student	Examiners assigned to the student		

Table 7.3.1.1: TC003_01_01 - Test normal flow of Assign University Coach and Examiner (SD005)

(b) TC003_01_02: Test exception flow of Assign University Coach and Examiner (SD005)

Test Case ID	TC003_01	Test Case Description	Test the Assign University Coach and Examiner		
Created By	Nasrul Amin	Reviewed By		Version	

QA Tester's Log					
Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	User is logged as coordinator
2	Has access to the Online FYP Management System

S #	Test Data
1	StaffID = nas123
2	Pass = 2f3er42

Test Scenario	Verify on assigning university coach and examiner

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	The coordinator can have to access the assigning page	Cannot access		
2	The system will display the students' lists	Students list does not displayed		
3	The system will display the examiners lists	Students list does not displayed		
4	Coordinator assign examiner to the student	Examiners are not assigned to the student		

Table 7.3.1.2: TC003_01_02 - Test exception flow of Assign University Coach and Examiner (SD005)

7.4 TC004: Test <Submission> Subsystem

This test contains the following test cases:

(a) TC004_01: Test <Submit Report> Scenario (SD009)

(b) TC004_02: Test <Review Project> Scenario (SD010)

(c) TC004_03: Test <Evaluate Project> Scenario (SD012)

(d) TC004_04: Test <Calculate Marks> Scenario (SD007)

7.4.1 TC004_01: Test <Submit Report> Scenario (SD009)

This test contains the following scenarios:

(a) TC004_01_01: Test normal flow of <Submit Report> (SD009)

Test Case ID	TC004_01	Test Case Description	Test the Submit Report Functionality		
Created By	Malavika	Reviewed By		Version	

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	Login to the system as Student
2	Access to the submission link
3	
4	

S #	Test Data
1	Submit report file (document)
2	
3	
4	

Test Scenario Verify on entering valid userid and password, student can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Login to the system	Display main page		
2	Access to the submission link	Display submission page		
3	Click Submit	Student submit report		

(b) TC004_02_01: Test normal flow of <Review Project> (SD010)

Test Case ID	TC004_02_01	Test Case Description	Test the Review Project Functionality		
Created By	Malavika	Reviewed By		Version	

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	Login to the system as Examiner
2	Access to the students' report
3	
4	

S #	Test Data
1	Student report (document)
2	
3	
4	

Test Scenario: Verify on entering valid userid and password, Examiner can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Login to the system	Display main page		
2	Access to the student reports	Student reports		
3	Click download	Download students' report		

(c) TC004_02_02: Test exception flow of <Review Project> (SD010)

Test Case ID	TC004_02_02	Test Case Description	Test the Review Project Functionality		
Created By	Malavika	Reviewed By		Version	

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	Login to the system as Examiner
2	Access to the students' report
3	
4	

S #	Test Data
1	Feedback (e.g: Revise use case diagram)
2	
3	
4	

Test Scenario: Verify on entering valid userid and password, Examiner can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Login to the system	Display main page		
2	Access to the student reports	Student reports		
3	Click download	Download students' report		
4	Provide feedback	Give comments		

(d) TC004_03_01: Test normal flow of <Evaluate Project> (SD012)

Test Case ID	TC004_03	Test Case Description	Test the Evaluate Project Functionality		
Created By	Malavika	Reviewed By		Version	

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	Login to the system as UC, IC and Examiner
2	Access to the rubric assessment
3	Access to the student reports
4	

S #	Test Data
1	Rubric file (marks)
2	
3	
4	

Test Scenario: Verify on entering valid userid and password, UC, IC, and Examiner can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Login to the system	Display main page		
2	Access to the rubric assessment	Download rubric assessment		
3	Access to the student reports	Student reports		
4	Click download	Download students' report		
5	Submit rubric assessment	Submission link page		

(e) TC004_04_01: Test normal flow of <Calculate Marks> (SD007)

Test Case ID	TC004_04	Test Case Description	Test the Calculate Marks Functionality		
Created By	Malavika	Reviewed By		Version	

QA Tester's Log

Tester's Name		Date Tested		Test Case (Pass/Fail/Not Executed)	

S #	Prerequisites:
1	Login to the system as Coordinator
2	Access to the rubric assessment
3	
4	

S #	Test Data
1	Calculate marks (e.g: marks : 50,60,80,100)
2	
3	
4	

Test Scenario: Verify on entering valid userid and password, Coordinator can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Login to the system	Display main page		
2	Access to the rubric assessment by UC,IC and examiner	Download the rubric assessment		
3	Calculate marks	Upload marks in the system		