UTM

UNIVERSITI TEKNOLOGI MALAYSIA

SECJ2203: Software Engineering

System Documentation (SD)

Project Title : iWork
(SECP 4124 - Professional Practice)

Version 1.0

22th June 2022

School of Computing, Faculty of Engineering

Lecturer : Dr Aryati Bakri

Prepared by: Mimiz

NO	NAME	MATRIC NUMBER
1.	IRSYAD ROS BIN HISYAM ROS	A20EC0052
2.	NUR FATEHAH BINTI HAIROL NIZAM	A20EC0113
3.	MUQRISYA BINTI MAT NOOR	A20EC0218
4.	MUHAMMAD SHAUQIE DANISH BIN ABAS	A20EC0098

Revision Page

a. Overview

The software design in this report is the first version of the iWork system. This report consists of the introduction, system-specific requirement, system architectural design, detailed description of components, and data design.

b. Target Audience

The target audiences of the iWork system are the University of Technology Malaysia (UTM) students that are involved in the FYP and the UTM staff that are responsible for FYP.

c. Project Team Members

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

Member Name	Matric Number	Role	Task	Status
IRSYAD ROS BIN HISYAM ROS	A20EC0053	Team Leader	• 3.1 Architecture Style and Rational • 3.2 Component Model • 4.1 Complete Package Diagram	completed
NUR FATEHAH BINTI HAIROL NIZAM	A20EC0113	Member	• 2.2.1 UC001: Use Case Login • 2.2.2UC002:Use Case Register • 2.2.3 UC003: Use Case Edit Profile • 2.2.4 UC004 Use Case for View Task • 2.2.5 UC005 Use Case for Download	completed

			Task ● 2.2.6 UC006 Use Case for SubmitFinishTask ● 2.2.7 UC007 Use Case for DeleteFinishTask ● 2.2.8 UC008 Use Case for View Student Profile ● 2.2.9 UC009 Use Case for Upload Task ● 2.2.10 UC010 Use Case for View Student Finish Task ● 2.2.11 UC011 Use Case for Download Student Finish Task ● 2.2.12 UC012 Use Case for Rate Student Finish Task ● 2.2.13 UC013 Use Case for View Notification	
MUQRISYA BINTI MAT NOOR	A20EC0218	Member	● 2.1.1 User Interfaces ● 2.1.2 Hardware Interfaces ● 2.1.3 Software Interfaces ● 2.1.4 Communication Interfaces ● 2.3 Performance and Other Requirements ● 2.4 Design Constraints ● 2.5 Software System Attributes	completed
MUHAMMAD SHAUQIE DANISH BIN ABAS	A20EC0098	Member	● 1.1 Purpose ● 1.2 Scope ● 1.3 Definitions, ● 1.4 References ● 1.5 Overview ● 4.2 Detailed Description ● 5.1 Data Description ● 5.2 Data Dictionary	completed

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	IRSYAD ROS BIN HISYAM ROS	COMPLETED VERSION 1.0	22/06/2022

Note:
This System Documentation (SD) template is adapted from IEEE Recommended Practice for Software Requirements Specification (SRS) (IEEE Std. 830-1998), Software Design Descriptions (SDD) (IEEE Std. 10161998 1), and Software Test Documentation (IEEE Std. 829-2008) that are simplified and customized to meet the need of SECJ2203 course at School of Computing, UTM. Examples of models are from Arlow and Neustadt (2002) and other sources stated accordingly.

Table of Contents

1. Introduction

Software Requirement Specification (SRS), Software Design Document (SDD), and Software Testing Documentation (STD) are all part of System Documentation (SD). Software Requirement Specification (SRS) is the first. The goal of documentation is to specify the requirements. for the sake of the system It discusses the system's purpose and features in great detail. the system's interface design, as well as the structural and functional definitions, non-functional system specifications, system behaviour, and design. To achieve each criteria, you must work within the constraints or limitations of the environment. This data was gathered by interviewing one of the project's stakeholders. The most popular method for getting information is the system, and this technique is the most typical method for gathering information requirements.

The Software Design Document (SDD) is then used to specify the requirements in the system's design diagrams, which will help in the creation of the system. This SDD dives into the graphical documentation in depth, including use case models, sequence diagrams, object behaviour models, and other supporting required data.

Testing is an important part of the software development life cycle. As a result, Software Testing Documentation (STD) is the third document in this SD, and it contains the documentation of items that should be prepared prior to or during system testing. Software testing documentation helps in calculating the testing effort required, as well as test coverage, requirement tracking, and more. In general, it contributes to the reduction of system development costs and time. This testing documentation displays the results of the system's software testing, allowing us to identify and correct any flaws.

1.1 Purpose

This SD describes the system for software development progress with detailed explanations of what is to be created and how it is expected to be created. The purposes of this SD are as listed below:

1. To help students filling their log book not using handwritten.
2. The supervisors and JKPSMi can monitor student progress.
3. Organisation Supervisor and Faculty Supervisor able to rate students' work using the system.
4. Help students to get an easy access about their current achievements during their industrial training.

1.2 Scope

The software product is iWork system. It is a system that is made for the UTM students, especially the third and fourth year students and supervisors that are responsible for FYP. This system provides an online platform to replace the manual system. It can be accessed by both PC and mobile phone. The students can use this system to view the dashboard and tasks while always being notified with the latest tasks or feedback regarding the FYP. This system can also allow students to edit their profile. They can download, update and also delete their tasks in the system. Meanwhile, the organisational and faculty supervisor are responsible for giving student feedback and rate the students work.They can view the students profile. They also can download students' tasks and upload the rated tasks to the students. In general, this system creates an efficient platform for the users and eases the users to send information.

The advantages of the iWork system include to provide an efficient platform for the users to organise their tasks so that everyone can be alert of the current tasks and also the future tasks. Students can directly ask their responsible supervisors what should be done and what should be avoided during the completion of the FYP. The supervisors also profited from the system because this system significantly reduces the workload and time of the supervisor.

1.3 Definitions, Acronyms and Abbreviation

1.	UTM	Universiti Teknologi Malaysia
2.	SD	System Documentation
3.	SRS	Software Requirement Specification
4.	SDD	Software Design Document
5.	STD	Software Testing Documentation
6.	FYP	Final Year Project
7.	UML	Unified Modelling Language a standardised modelling language consisting of an integrated set of diagrams
8.	Domain Model	Domain model is a conceptual model to describe and model real world entities and the relationships between them. Domain class objects reflect business concern, policies, and constraints
9.	Use case diagram	Use case diagrams are used for interaction modelling. The purpose of a use case diagram is to capture the dynamic aspect of a system.

1.4 References

Specify a complete list of references using a standardised reference format.

1.5 Overview

The SD Document is divided into nine sections with some of them having various subsections. The sections of the Software Design Document are:

1.0 Introduction: This section consists of five subsections that covers the system purpose, system scope, system overview, references, document overview and the definitions of terms, acronyms and abbreviation used in this documentation. Overall, it explains about the contents of System Documentation (SD) and the system, its background, its functionalities, the audiences, and the references used to complete this system.

2.0 Specific Requirements: This section has six subsections that includes external interface requirements, system features, performance requirements, design constraints, software system attributes and other requirements. Generally, in the first subsection it discusses user interfaces, hardware interfaces, software interfaces and communication interfaces of the system Meanwhile, it describes the system features using use case description for each function that the system offers. In addition, this section briefly explains about the design constraints, performance requirements and software system attributes.

3.0 System Architectural Design: In this section, it describes the architectural design. The system should be divided into a set of communicating subsystems and components. It describes the decomposition, the components, the interfaces, and the rationales.

4.0 Detailed Description of Components: Here we describe the detailed description of components consisting of a complete package diagram to show 5 the files and subfiles contained within the system. Also, the details then elaborated according to the content of the package diagram where every file has their class diagram, sequence diagram, entity, method, and the algorithm table.

5.0 Data Design: Data design contains data description that contains every entity name and their description. Next is a data dictionary which lists out the attribute name. type and description for each entity.

2. Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

The aim for this project is to develop a beginner-friendly interface where all users from age of 19 until 60 can use. It is important to ensure user can understand the interface so we can increase efficiency.

a) System interface depends on who the user is as there will be different interface for each group of user such as students, faculty supervisor, organisation supervisor, and admin.
b) Admin only can read and register students to store or update students' data into the database.
c) Input will be taken for each click on the interface whether through touchscreen, cursor or keyboard.

These are how user will interact with the system, :

Group of user	**Interaction with interface**
Student	Student should log in into their registered account. Student also can customise their password into what they preferred after logged in for the first time. Besides, they can view, and edit their profile if they see any mistakes in their profile. When they edit their profile information, this will update the data of theirs in database server too. Student will receive some tasks from the supervisor and need to complete it outside the system. They can download the task assigned to them. When they finished the assessment, they need to upload it in the system. Student can delete

	the task they uploaded. Student also can see their marks for the report. They will be notified by the app too.
Faculty Supervisor	Faculty supervisor can view students' profile, download their finished tasks and mark students' report. They will be notified by the system of every students' progress.
Organisation Supervisor	Organisation supervisor can view the students' profile. Besides, they will upload tasks to the students through the system by entering the correct task code. Plus, they can view and download the students' finished task that has been uploaded by the student itself. They are responsible to mark the students' completed tasks. Moreover, they also will be notified by the system about the progress of the student.
JKPSMi	JKPSMi will register all the students' information into the database where the interface will give a prompt to enter the students' matric number, identification card number, full name, contact number, password for the student's account, email and their home address. JKPSMi only can view the student's profile after registering their account. JKPSMi can update their information in the database server too. JKPSMi will only be monitoring students' progress where they only can view the students' finished task.

2.1.2 Hardware Interfaces

- To maintain a smooth movement using the system, user should consider using a good internet connection whether using Wi-Fi, Ethernet cable or else. Not only that, the system can be used in any operating system like Mac, iOS, Windows or Android. It can be access on a computer, tablet, or mobile phone.

2.1.3 Software Interfaces

- Software requirements :
- In developing the system, we use Oracle as the database server, Google Drive as our cloud storage where all group of users can access, and Figma as our framework design. Not only that, the system only accept certain document data type which are in .pdf and .docx format (2007 and above).
- Operating system requirements :
- User must at least use iOS version 10.0 and above, and for Android version 9.0 and above.

2.1.4 Communication Interfaces

- It is important to ensure users can communicate with everything that is accessible on the system. It will require another interface from another system. Therefore, the system will have few features, such as:
- Users can access Google Drive as the cloud server where this will let them communicate with the data such as tasks.
- In the network system, to let users communicate with the system, there will be a LAN network to connect the host (device) with the internet.
- User will communicate with the system through the World Wide Web.
- Communication interface for the organisation supervisor will be more complex, because there are more processes going on than other users.

2.2 System Features

The system features include :

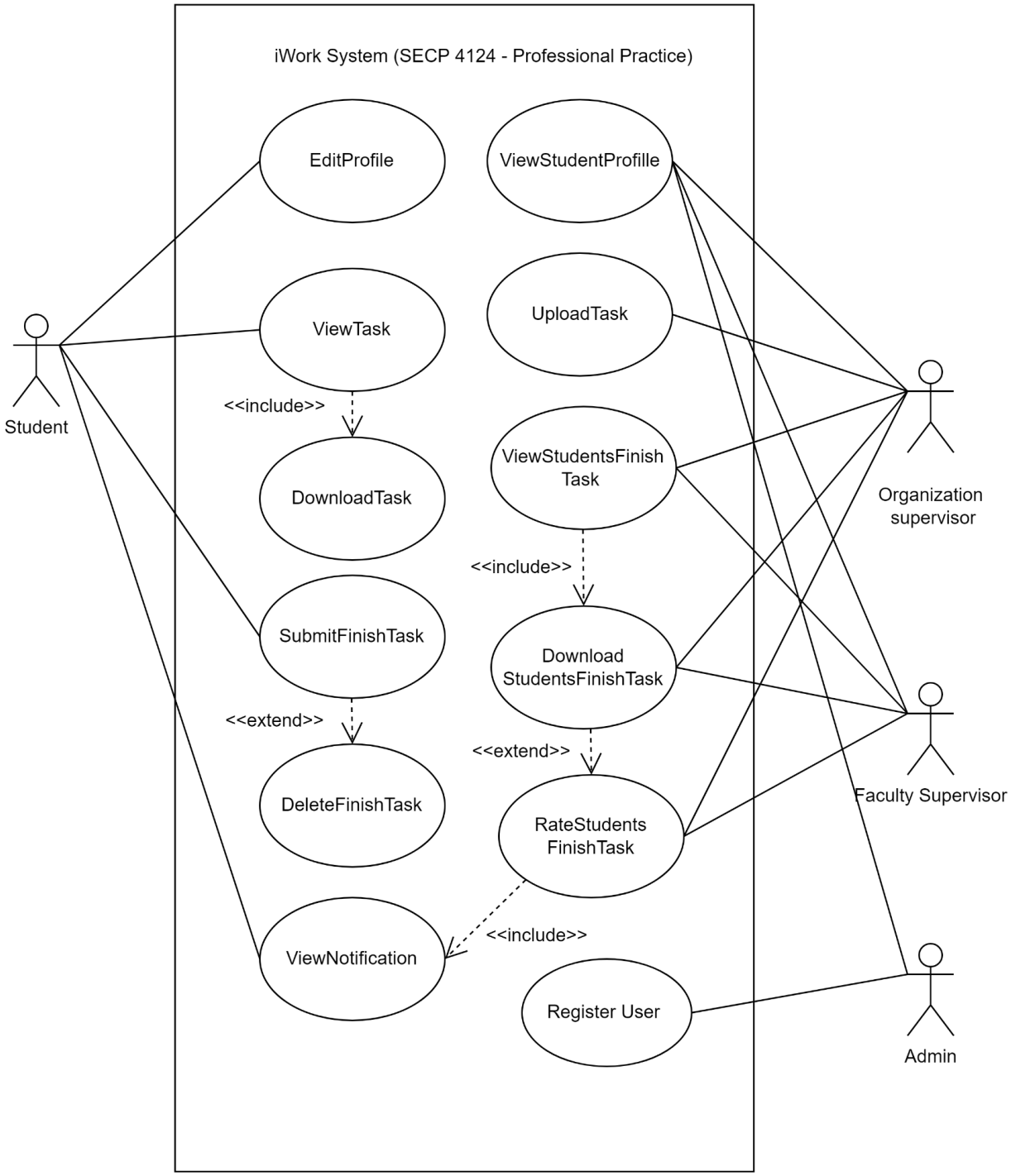


Figure 2.1: Use Case Diagram for iWork

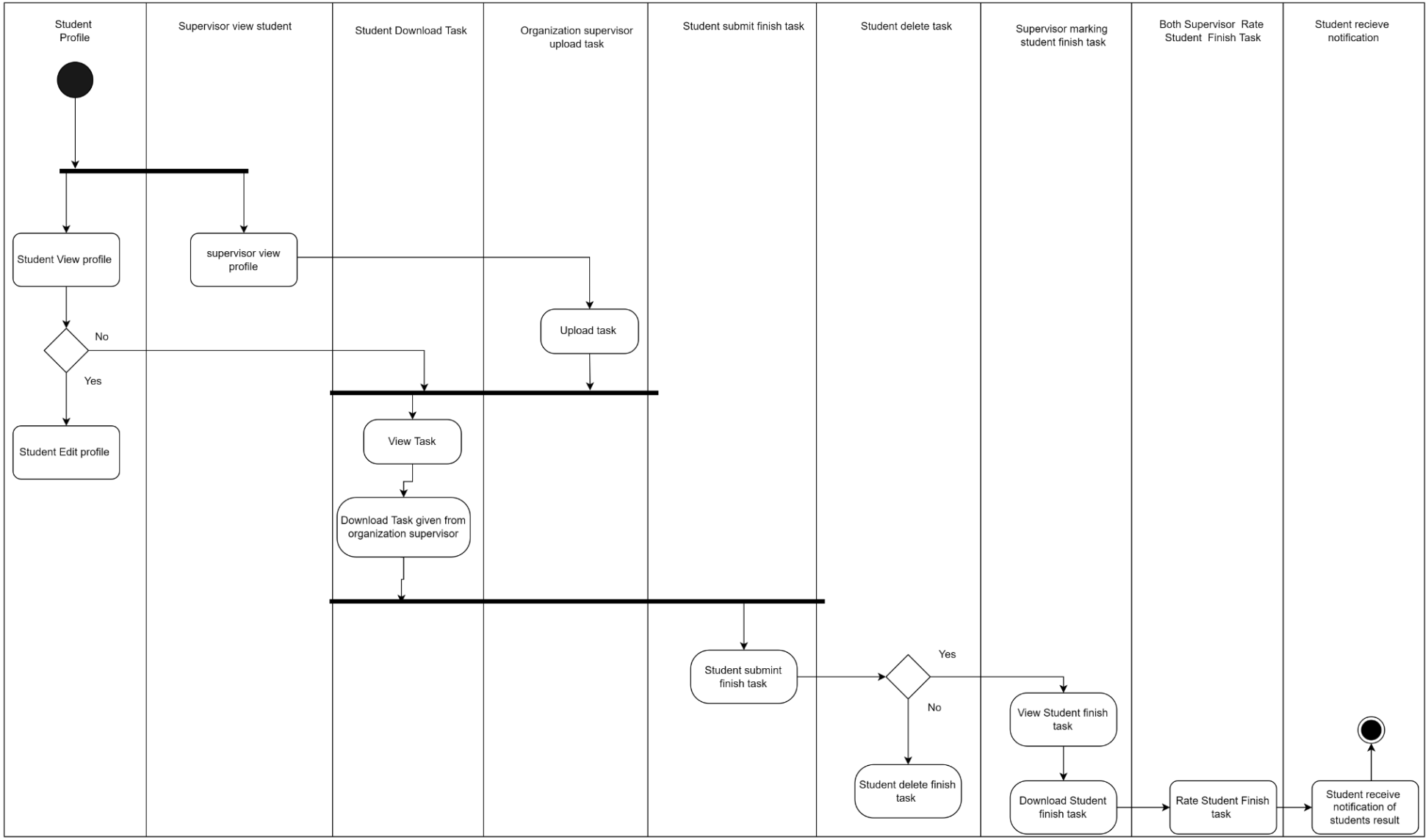


Figure 2.2: Activity Diagram for iWork

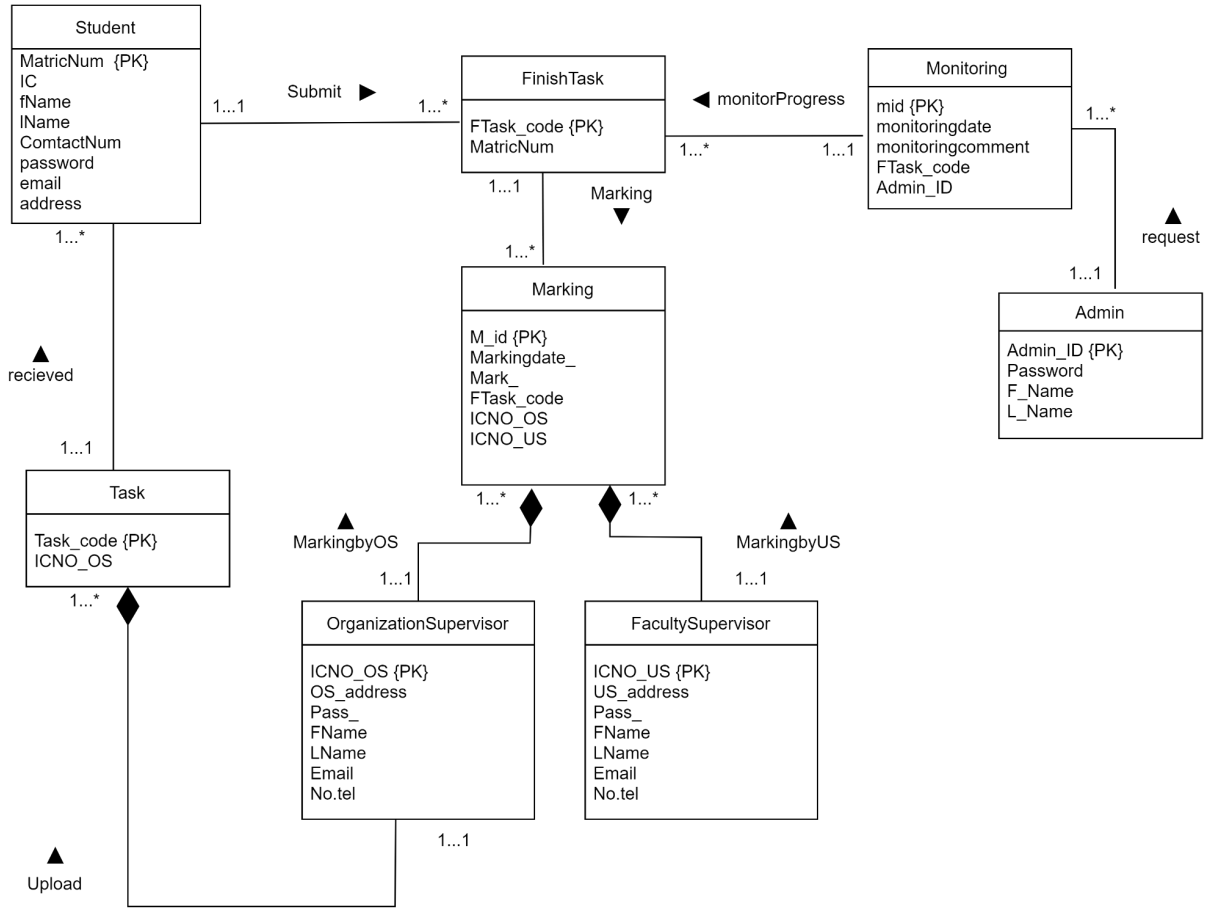


Figure 2.3: Domain Model for iWork

2.2.1 UC001: Use Case Login

Table 2.1: Use Case Description for Login`

Use case: Login
ID:IW001
Actors: Student , Faculty Supervisor & Organisation Supervisor
Preconditions: ● Access to system ● Internet connections ● Student identity ● University Supervisor Identity ● Organisation Supervisor Identity
Flow of events: 1.Users will fill the log in with their username that has already been provided. 2.System validate user identity with each of database 3.User able to enter system 4.Use Case end
Postconditions: -
Alternative flow *n*:-
Exception flow (if any): 1. Users cannot enter if the admin(JKPSMi) hasn't registered a student.. 1.1 will display the message “User does not exist. 1.2 return to login page back

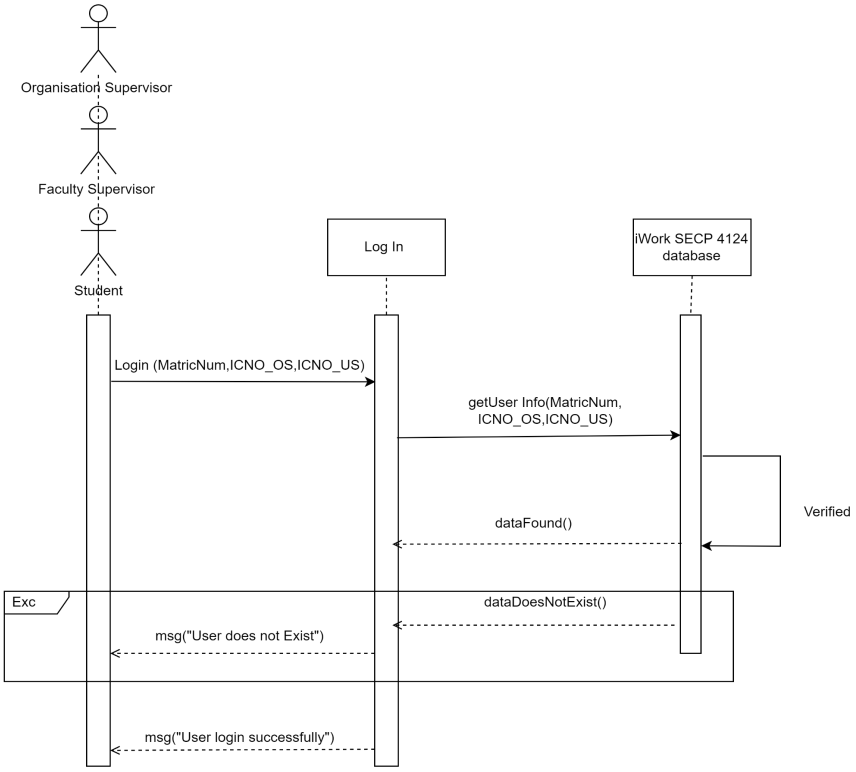


Figure 2.5: Sequence Diagram for Login

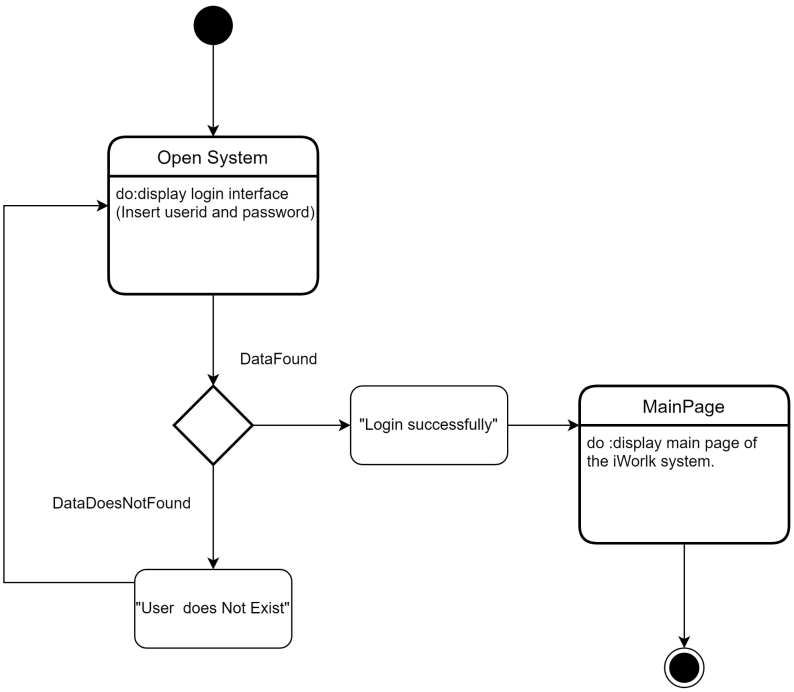


Figure 2.6: State Diagram for Login system

2.2.2 UC002: Use Case Register

Table 2.2: Use Case Description for Register

Use case: Register
ID:IW002
Actors: Admin
Preconditions: ● Access to system ● Internet connections ● Student identity
Flow of events: 1. Admin sign up student name and matric number inside the registration name and produce temporary password 2. System will search and compare student identity from the student database. 3. Student registered to iWork System 4. Admin has registered as a student. 5. Use case End
Alternative flow *n*:-
Postconditions:
Exception flow (if any): **1.** Students cannot be registered if they are not taking this subject. 1.1 a message will display “This student does not exist”

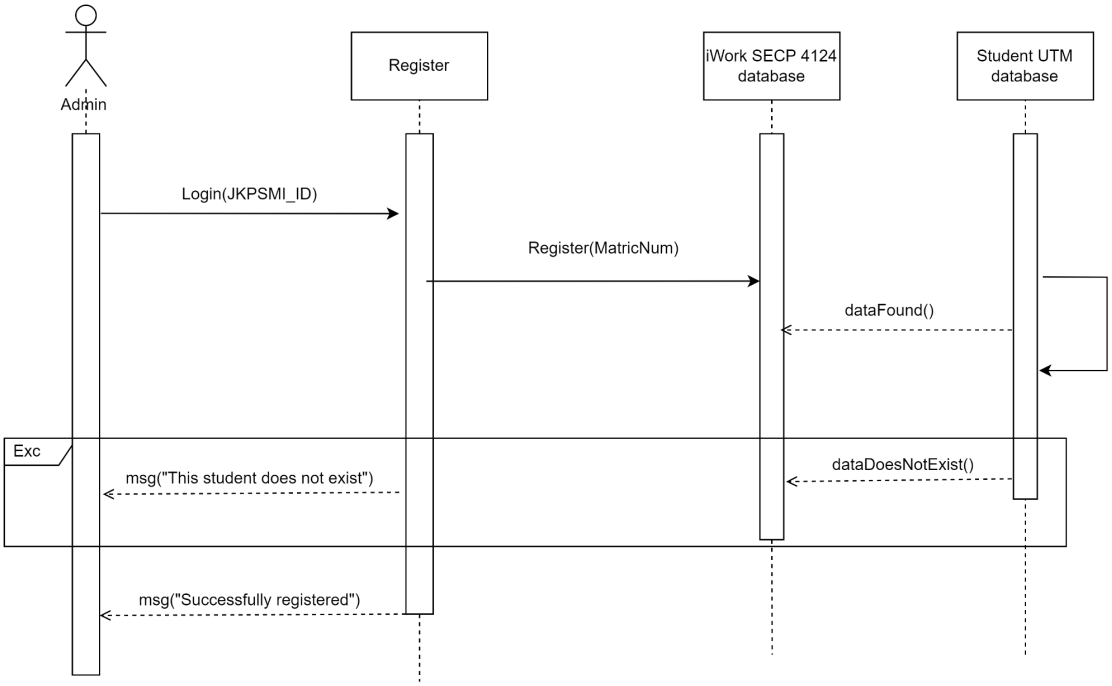


Figure 2.7: Sequence Diagram for Register

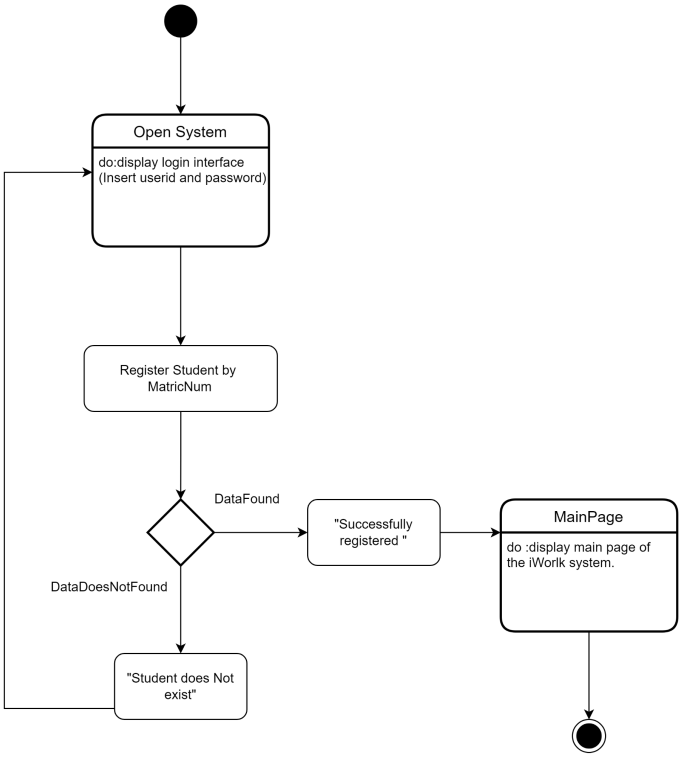


Figure 2.8: State Diagram for Register System

2.2.3 **UC003: Use Case Edit Profile**

Table 2.3: Use Case Description for Edit Profile

Use case: Edit Profile
ID:IW003
Actors: Student
Preconditions: ● Internet connections ● Students have to login inside the system.
Flow of events: 1. Student need to login inside the system 2. Go to the student profile page. 3. Students can edit their information (profile picture, number phone or email.
Alternative flow *n*:-
Postconditions:
Exception flow (if any): **1.** Students can undo it if they do not want to change any of their information. 2.Will go back to the student profile picture.

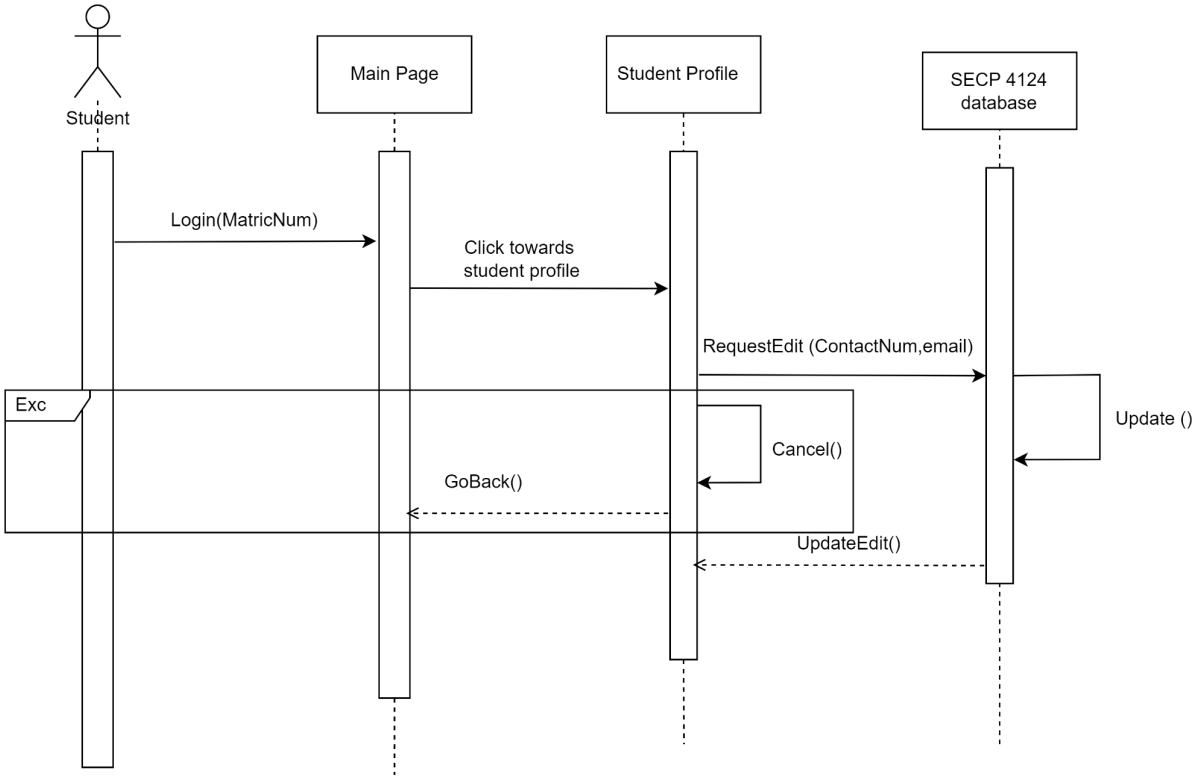


Figure 2.8: Sequence Diagram for Edit Profile

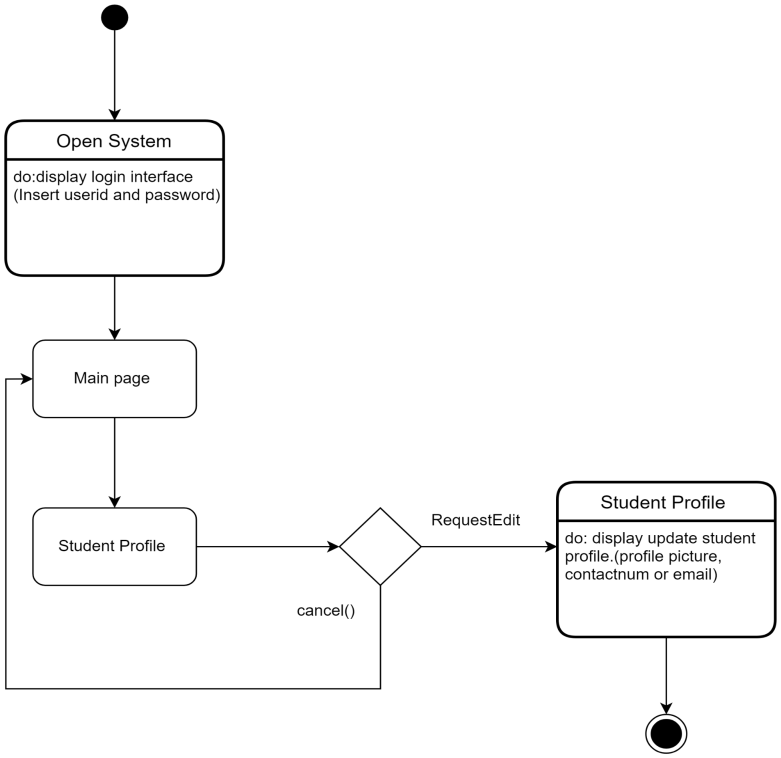


Figure 2.9: State Diagram for Edit Profile

2.2.4 UC004 Use Case for View Task

Table 2.4: Use Case Description for View Task

Use case: View Task
ID:IW004
Actors: Student
Preconditions: ● Internet connections ● Students have to login inside the system. ● Students need to be assigned an Organisation Supervisor. ● Student need to login inside the system
Flow of events: 1. Students go to the task page. 2. Students go to the task to view the task.
Alternative flow *n*:-
Postconditions:
Exception flow (if any):

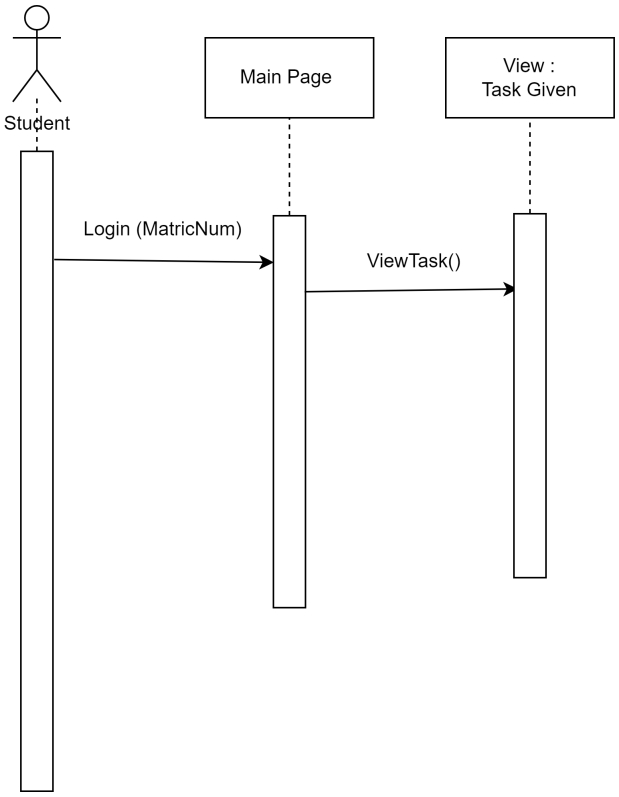


Figure 2.9: Sequence Diagram for View Task

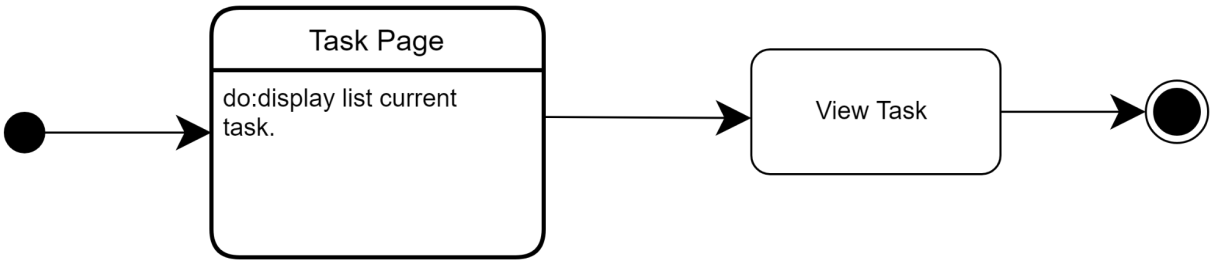


Figure 2.9: State Diagram for View Task

2.2.5 UC005 Use Case for Download Task

Table 2.5: Use Case Description for Download Task

Use case: Download Task
ID:IW005
Actors: Student
Preconditions: ● Internet connections ● Students have to login inside the system. ● Students need to be assigned an Organisation Supervisor. ● Student need to login inside the system
Flow of events: 1. Students go to the task page. 2. Students go to the task to view the task. 3. Students download the task given by lecturer
Alternative flow *n*:-
Postconditions: -
Exception flow (if any):

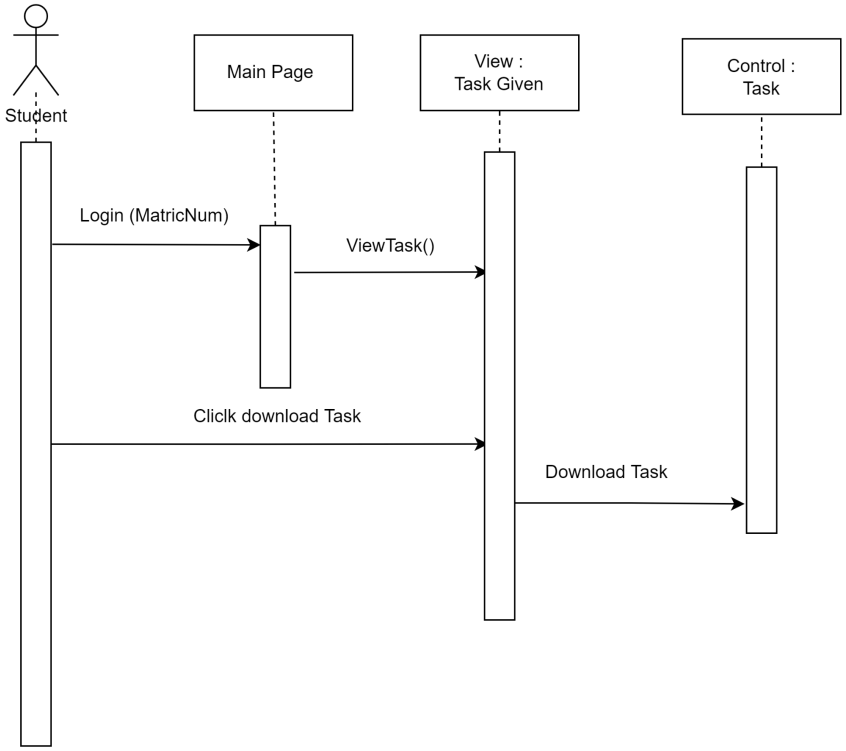


Figure 2.10: Sequence Diagram for Download Task

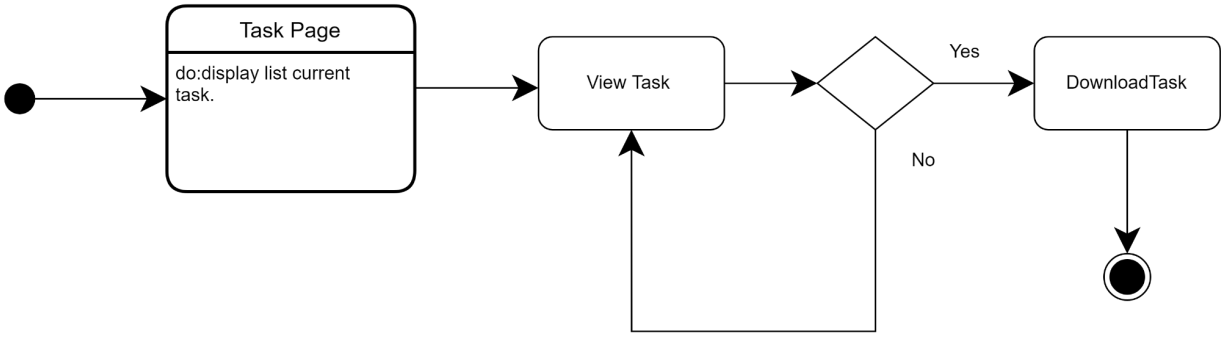


Figure 2.9: State Diagram for Download Task

2.2.6 UC006 Use Case for SubmitFinishTask

Table 2.6: Use Case SubmitFinishTask

Use case: SubmitFinishTask
ID:IW006
Actors: Student
Preconditions: ● Internet connections ● Students have to login inside the system. ● Students need to be assigned an Organisation Supervisor. ● Student need to login inside the system
Flow of events: 1. Students will go to the task page. 2. Students will send the finished task on the submission page.
Alternative flow *n*:
Postconditions: -
Exception flow (if any):

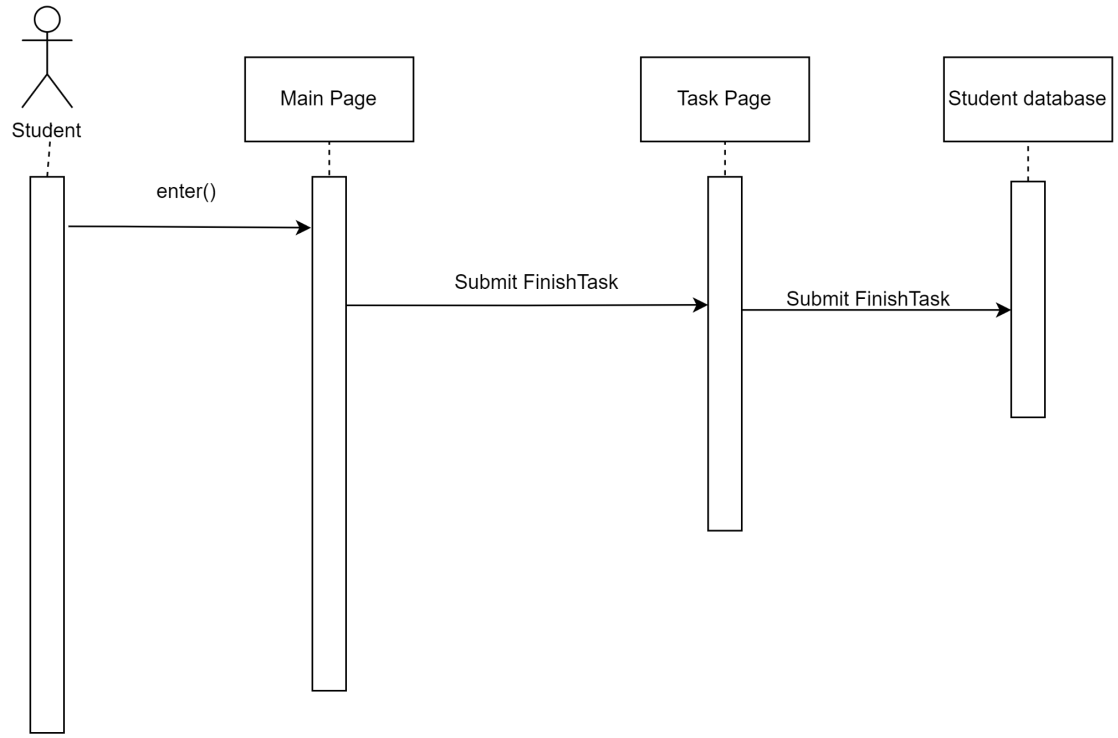


Figure 2.10: Sequence Diagram for Submit Finish Task

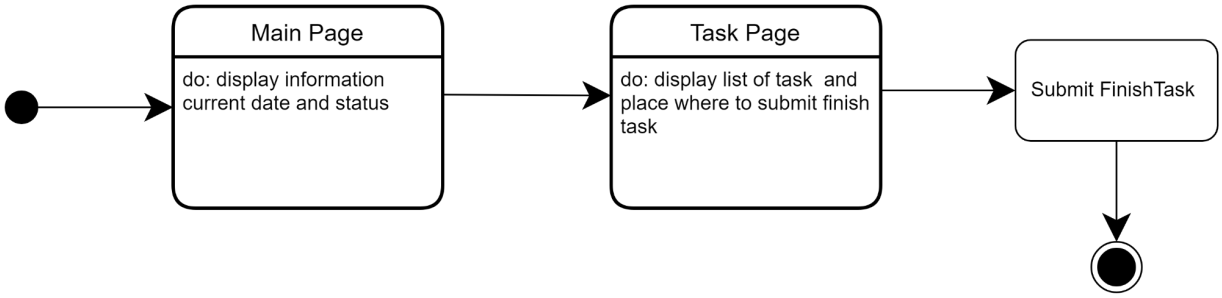


Figure 2.10: State Diagram for Download Task

2.2.7 UC007 Use Case for DeleteFinishTask

Table 2.7: Use Case DeleteFinishTask

Use case: DeleteFinishTask
ID:IW007
Actors: Student
Preconditions: ● Internet connections ● Students have to login inside the system. ● Students need to be assigned an Organisation Supervisor. ● Student need to login inside the system
Flow of events: 1. Students will go to the task page. 2. Students will send the finished task on the submission page. 3. Students decide to delete the finishing task.
Alternative flow *n*:-
Postconditions: -
Exception flow (if any):

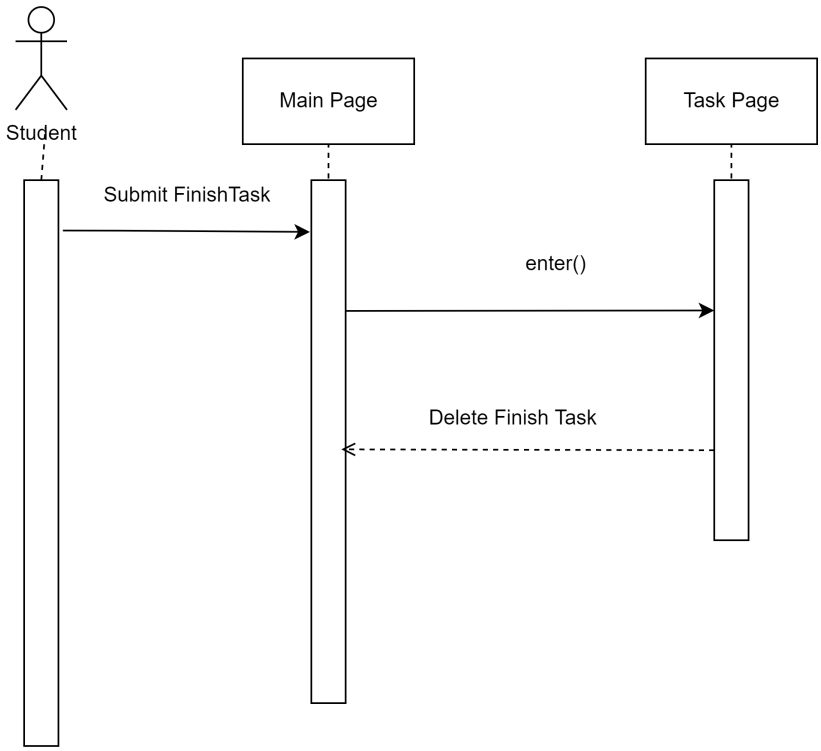


Figure 2.11: Sequence Diagram for Delete Finish Task

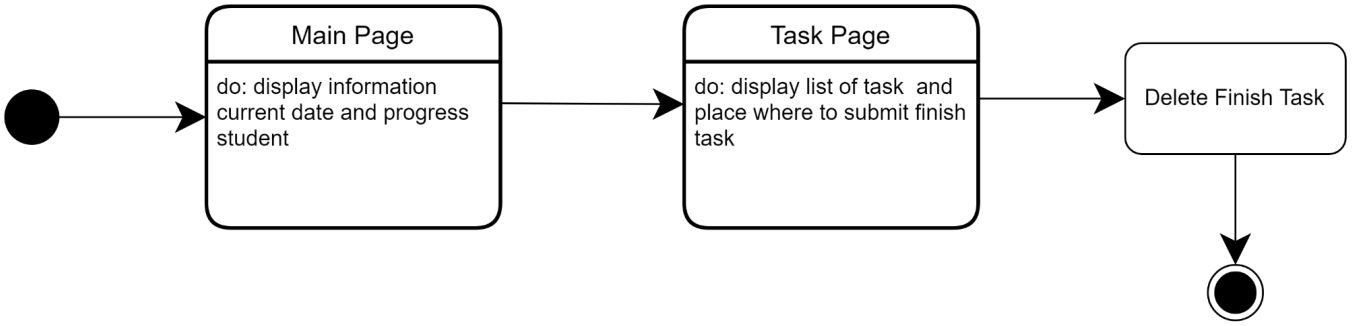


Figure 2.11: State Diagram for Delete Finish Task

2.2.8 UC008 Use Case forView Student Profile

Table 2.8: Use Case View Student Profile

Use case: View Student Profile
ID:IW008
Actors: Organization supervisor, Faculty Supervisor,JKPSMi
Preconditions: ● Internet connections ● Organization supervisor, University Supervisor,JKPSMi have to login inside the system. ● Students need to be assigned an Organisation Supervisor and University Supervisor
Flow of events: 1. Organization supervisor, University Supervisor,JKPSMi will go to the list of students. 2. Click on the student's name. 3. It will go to a student's profile . 4. Organization supervisor, University Supervisor,JKPSMi can see their finished task and progress.
Alternative flow *n*:-
Postconditions: -
Exception flow (if any): -

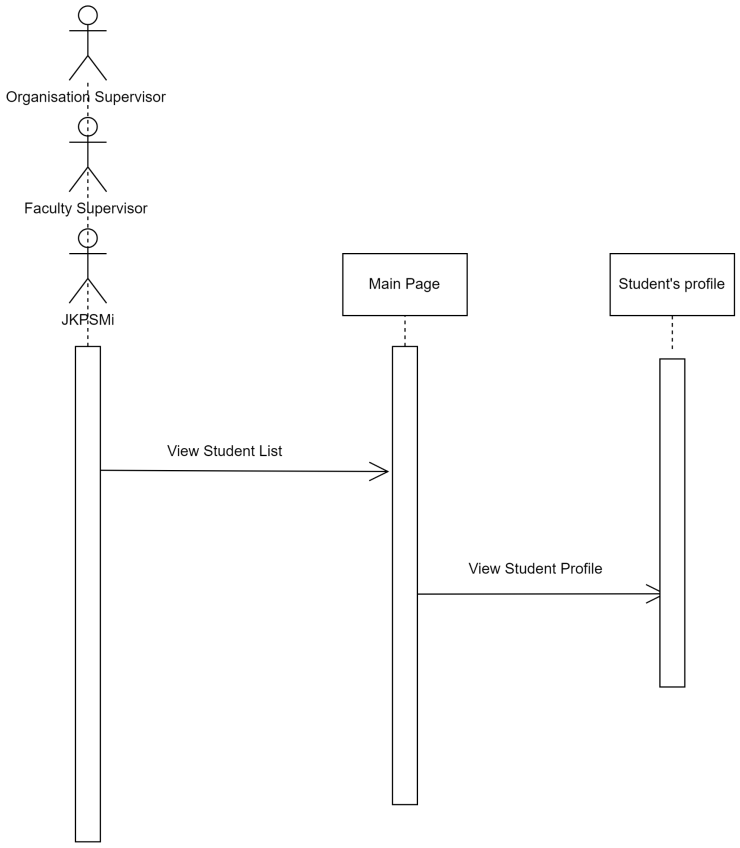


Figure 2.12: Sequence Diagram for View Student Profile

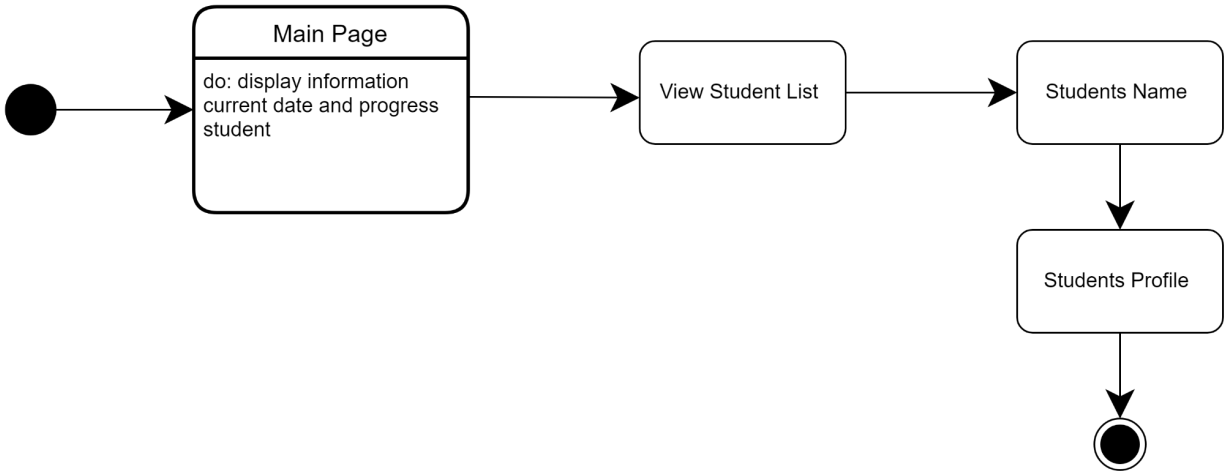


Figure 2.12: State Diagram for View Student Profile

2.2.9 UC009 Use Case for Upload Task

Table 2.9: Use Case Upload Task

Use case: Upload task
ID:IW009
Actors: Organization Supervisor
Preconditions: ● Internet connectionsr ● Students need to be assigned an Organisation Supervisor.
Flow of events: 1. Organization supervisor will go to his task page. 2. Organization supervisor uploads the task given towards the students. 3. Organization supervisor set a date and time for student to submit.
Alternative flow *n*:-
Postconditions: -
Exception flow (if any): - 1. Organization Supervisor can delete the task.

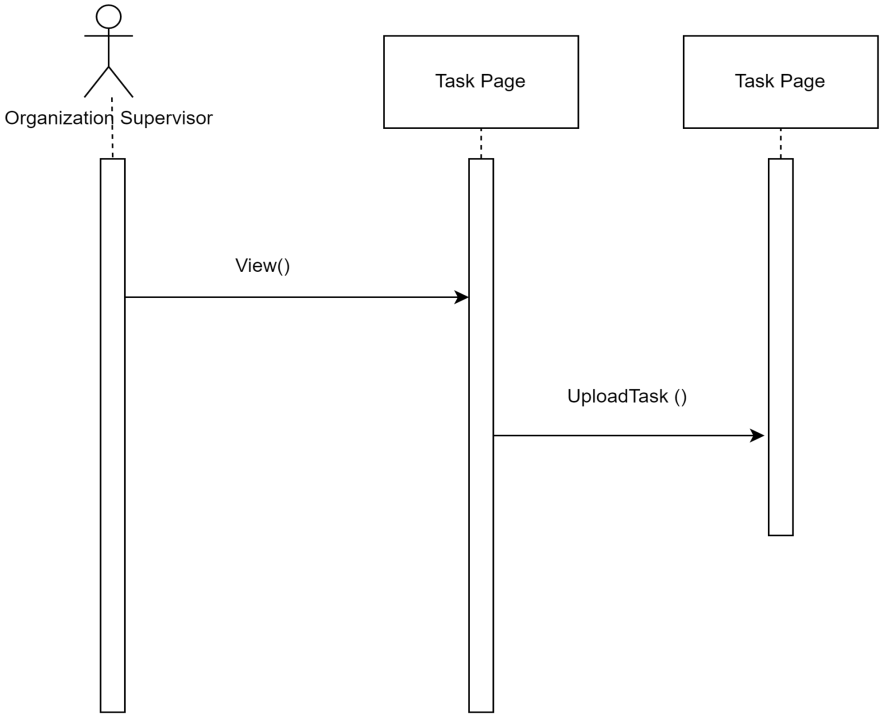


Figure 2.13: Sequence Diagram for Upload Task

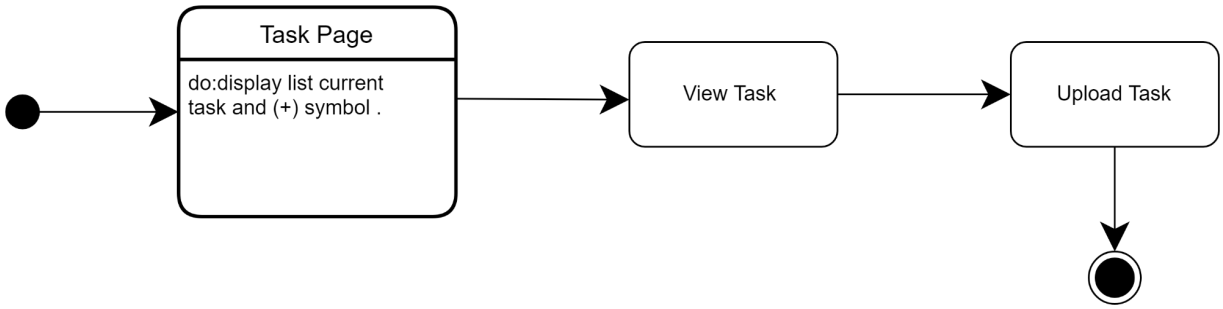


Figure 2.13: State Diagram for Upload Task

2.2.10 UC010 Use Case for View Student Finish Task

Table 2.10: Use Case View Student Finish Task

Use case: View Student Finish Task
ID:IW010
Actors: Organization Supervisor
Preconditions: ● Internet connections ● Students need to be assigned an Organisation Supervisor and University Supervisor
Flow of events: 1. Organization supervisor and University supervisor will go to the task page. 2. Both will view students finish tasks.
Alternative flow *n*:-
Postconditions: -
Exception flow (if any): -

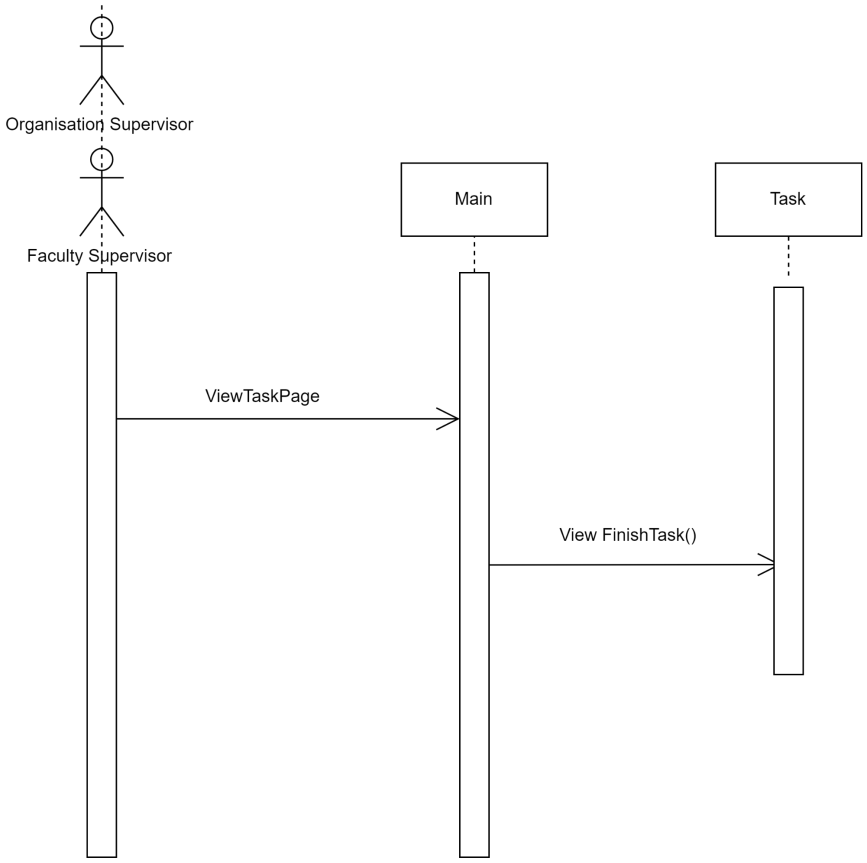


Figure 2.14: Sequence Diagram for View Finish Task

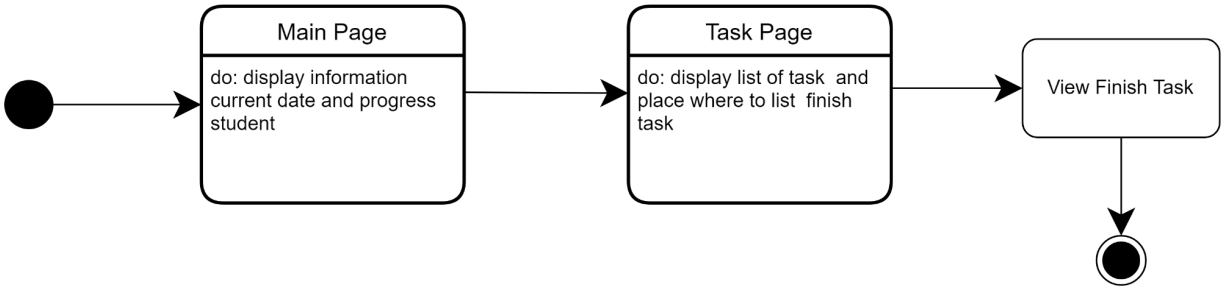


Figure 2.14: State Diagram for View Finish Task

2.2.11 UC011 Use Case for Download Student Finish Task

Table 2.11: Use Case Download Student Finish Task

Use case: Upload task
ID:IW011
Actors: Organization supervisor,Faculty supervisor
Preconditions: ● Internet connections ● Students need to be assigned an Organisation Supervisor and University Supervisor
Flow of events: 1. Organization supervisor and University supervisor will go to the task page. 2. Both will view students finish tasks. 3. Both of the supervisors will download finishing task and mark them.
Alternative flow *n*:-
Postconditions: -
Exception flow (if any): -

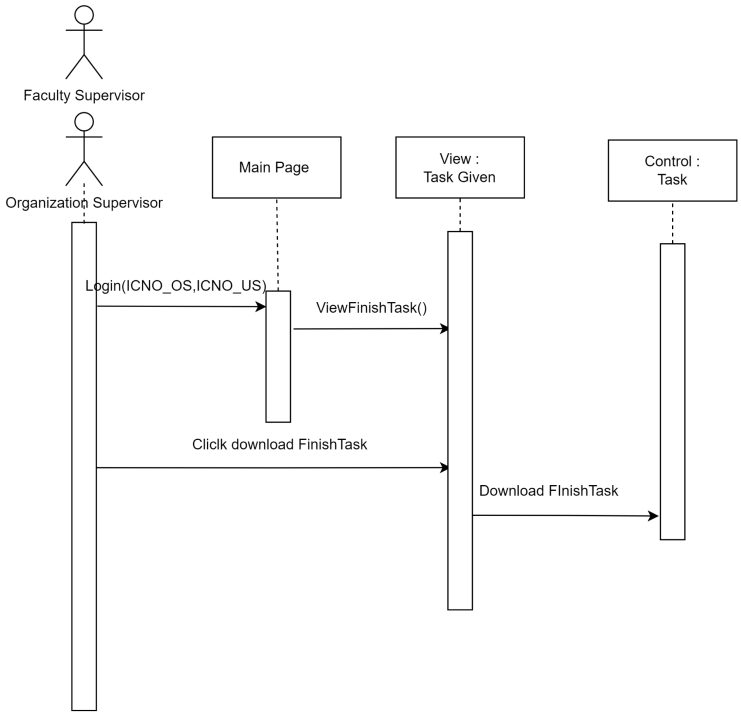


Figure 2.15: Sequence Diagram for Download Student Finish Task

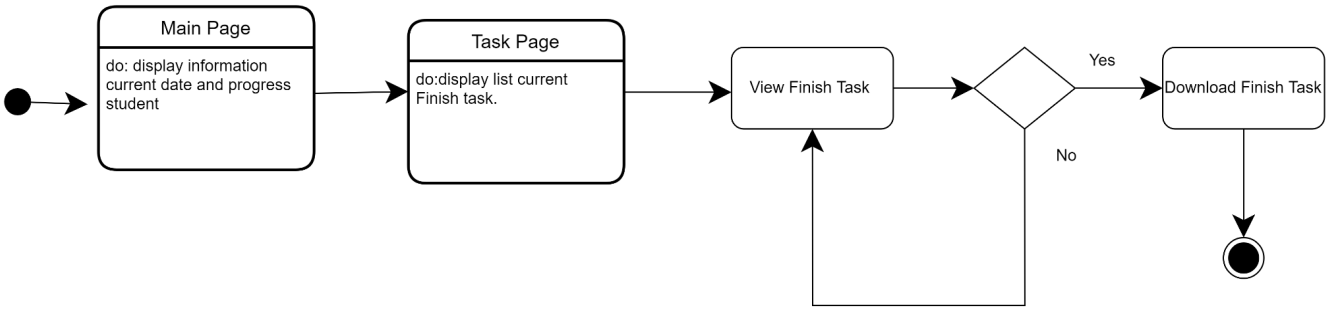


Figure 2.15: State Diagram for Download Student Finish Task

2.2.12 UC012 Use Case for Rate Student Finish Task

Table 2.12: Use Case Rate Student Finish Task

Use case: Rate Student Finish Task
ID:IW012
Actors: Organization Supervisor,Faculty Supervisor
Preconditions: ● Internet connections ● Students need to be assigned an Organisation Supervisor and University Supervisor
Flow of events: 1. Both supervisors marked students' tasks. 2. Result for the task was uploaded back and marks will be stated.
Alternative flow *n*:-
Postconditions: -
Exception flow (if any): -

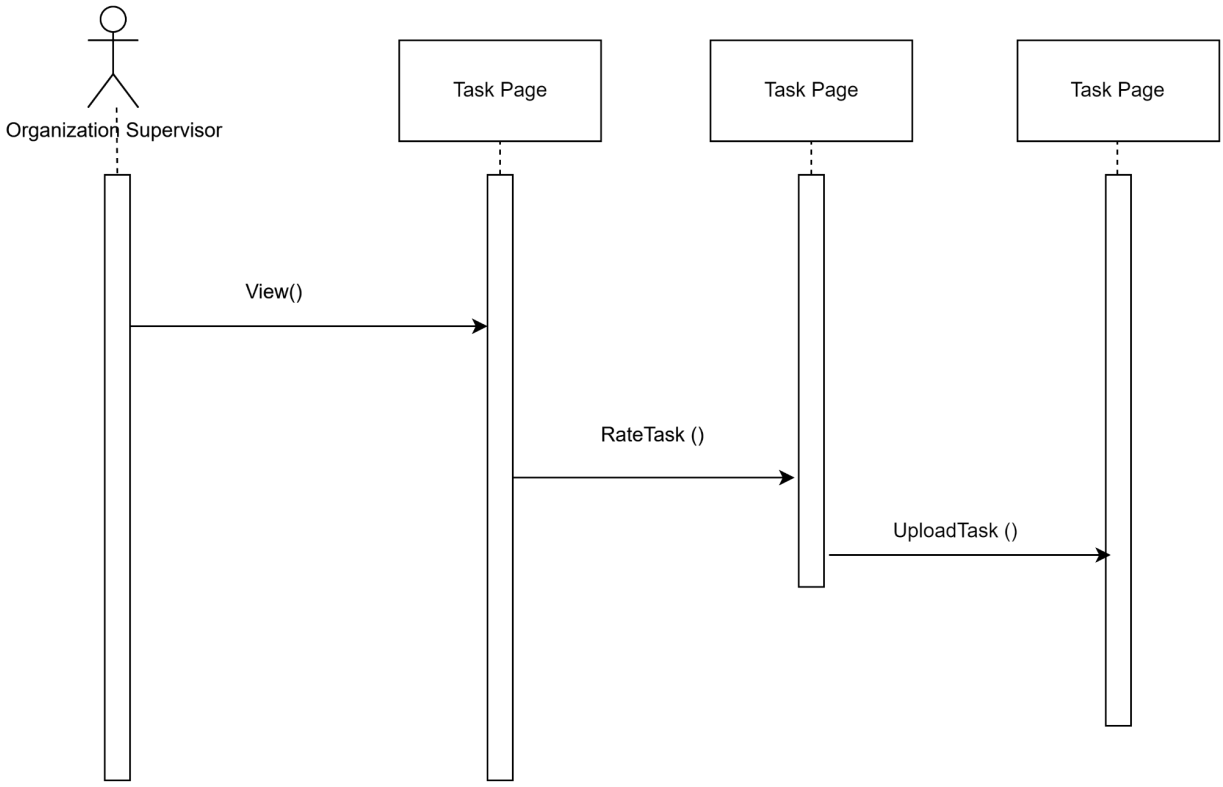


Figure 2.16: Sequence Diagram for Rate Student Finish Task

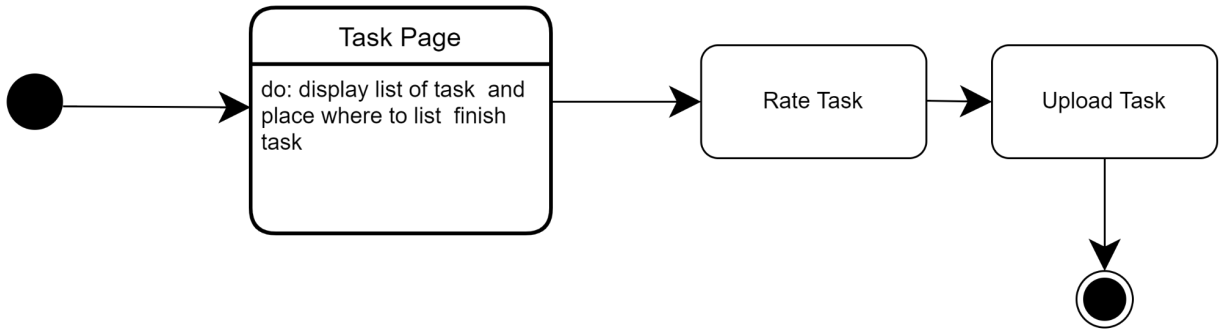


Figure 2.16: State Diagram for Rate Student Finish Task

2.2.13 UC013 Use Case for View Notification

Table 2.13: Use Case View Notification

Use case: View Notification
ID:IW013
Actors: Student
Preconditions: ● Internet connections ● Students need to be log in inside the system ● Students need to be assigned an Organisation Supervisor and University Supervisor
Flow of events: 1. Student will receive notification from system once the mark already been up 2. Students will look at the mark and feedback from both supervisors.
Alternative flow *n*:-
Postconditions: -
Exception flow (if any): -

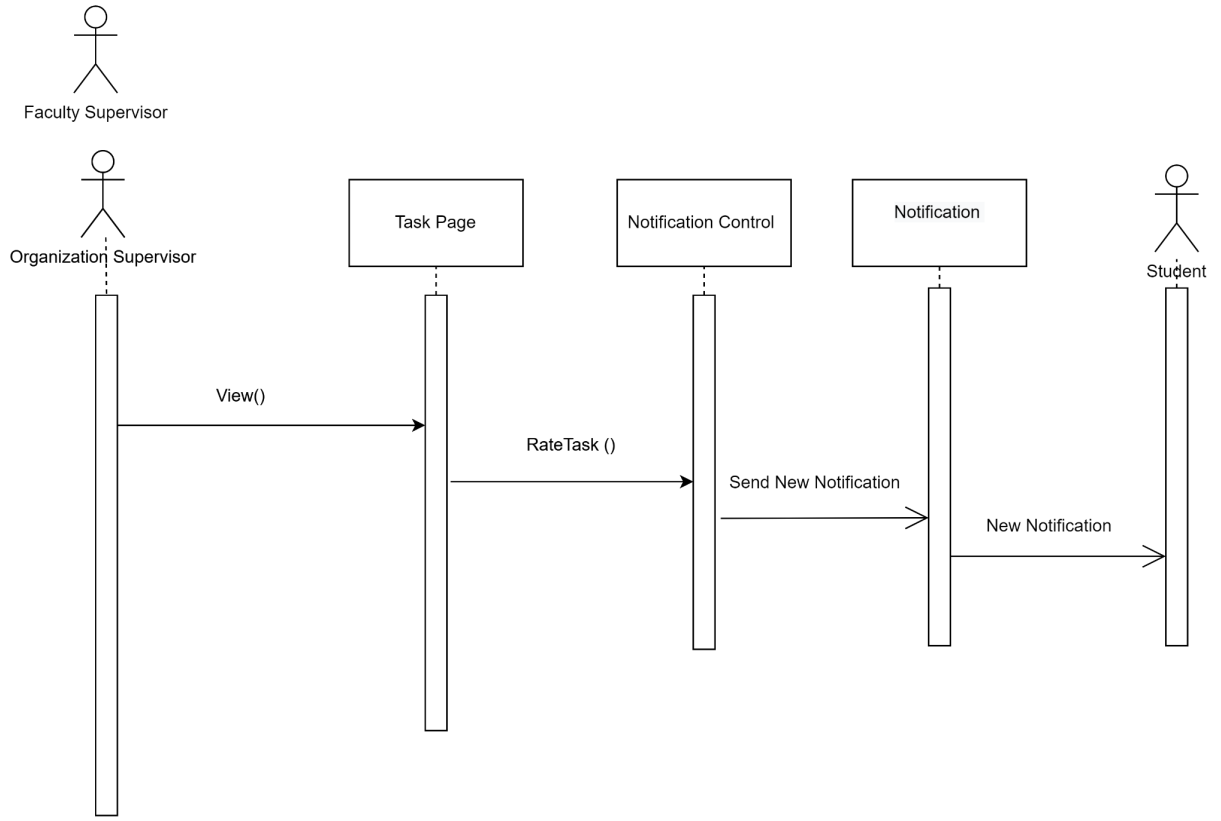


Figure 2.17: Sequence Diagram for View Notification

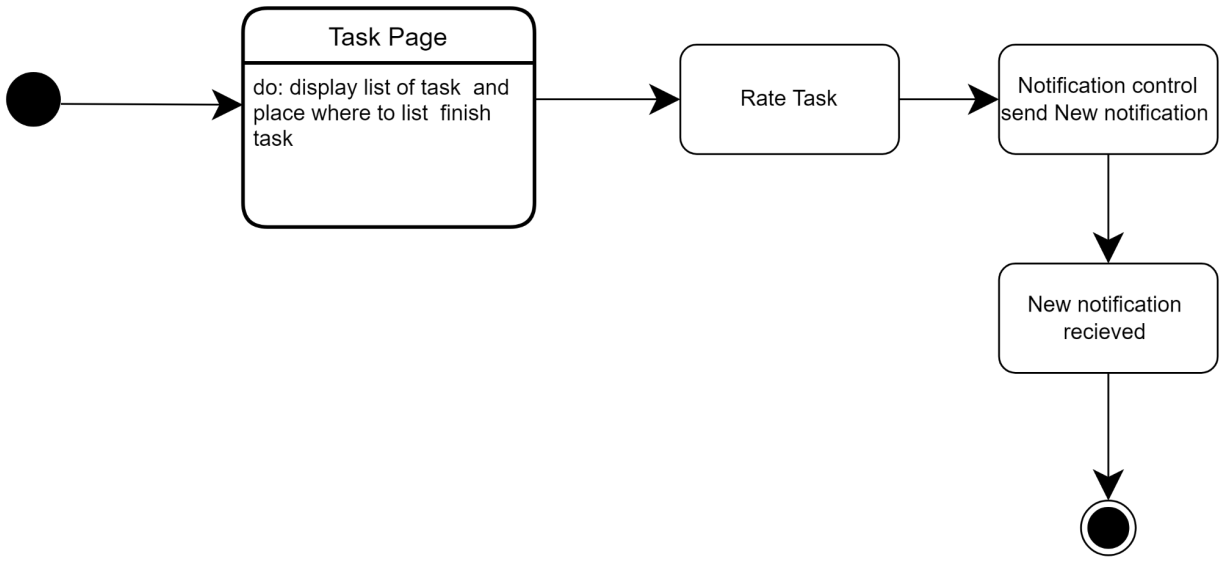


Figure 2.17: Sequence Diagram for View Notification

2.3 Performance and Other Requirements

Adaptability	The system is able to manage to adapt with some changes in the future. Example, if there is only one supervisor need to supervise students, then it is easily to remove the other supervisor in the system and can proceed using the system as usual.
Flexibility	System is flexible for different types of users to have different process for each group of users. For instance, there is different process for supervisors and students. Both have different processes and also interfaces.
Interoperability	This system will be cooperating with another software which are database software (Oracle)
Maintainability	The system is easy to change if there are any changes in request. Average time taken to recover the system if it fails is about 1-2 months.
Portability	System can be used in different operating systems such as Mac, iOS, and Chrome.
Reliability	The system adequately follows the defined performance specifications during a specific time in an environment.
Reusability	The system can be reused for years as long there is no changes made in requirements. However, if there are changes, system must be fixed first.
Usability	The system is easy to use as the interface theme will be minimalist, and there is not a lot of process going on in

	the system. Every process has their own path and user only has to follow one by one.

2.4 Design Constraints

- Student should enter their username and password correctly to log in
- Faculty and organisation supervisors should enter their identification card number and password correctly to log in
- Organisation supervisor should ensure the task code entered is correct to upload the tasks to the students
- All users should have an internet connection to access the system
- New user might having hard time to understand the processes in the system
- Student cannot change their matric card number in the system
- Student cannot change the marks for their reports
- System will only use Google Cloud as the server
- Oracle will be used as the database server
- Quizy, Figma and Adobe XD will be used in creating the GUI interface.

2.5 Software System Attributes

Interface	Interface should be easily understandable because there are many users will be using the system, from the age of 19 until 60 including students, supervisors and administrators.
	Language that will be used in the system is level 3 of proficiency in English.
Security	Customised password should be a secured password where it must contain capital letter, small letter, numbers and character.
	Informations of students, and supervisors need to be always updated and kept in the database server.
Software	Users except 8admin can only upload documents or tasks in certain data types which are in .pdf and .docx format.
	System should notify students about the progress, marks, and due of the tasks given.

3. System Architectural Design

3.1 Architecture Style and Rationale

Layered Architecture was chosen as the architecture style for our system. The concept organises modules or components with similar functionality into horizontal layers, which in turn causes each layer to serve a distinct purpose inside the system application. Our system is made of 3 layers which are the data layer, the web server layer and the front end layer. Each layer will be able to interact with other layers within the architecture with notable consistency. Changes are easy to implement due to the fact that the Layered Architecture has great agility and finding specific objects within a layer is easier. It also helps us to determine each task that is required for each layer.

3.2 Component Model

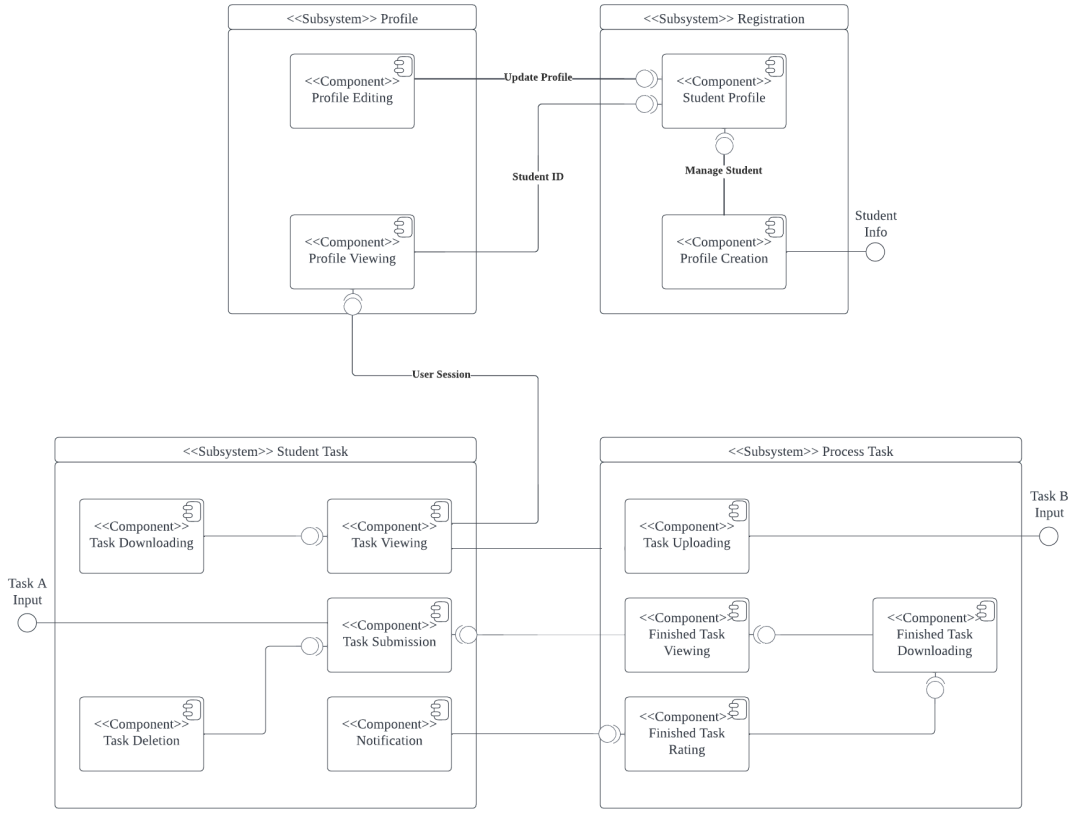


Figure 3.1: Component Diagram of iWork System

4. Detailed Description of Components

4.1 Complete Package Diagram.

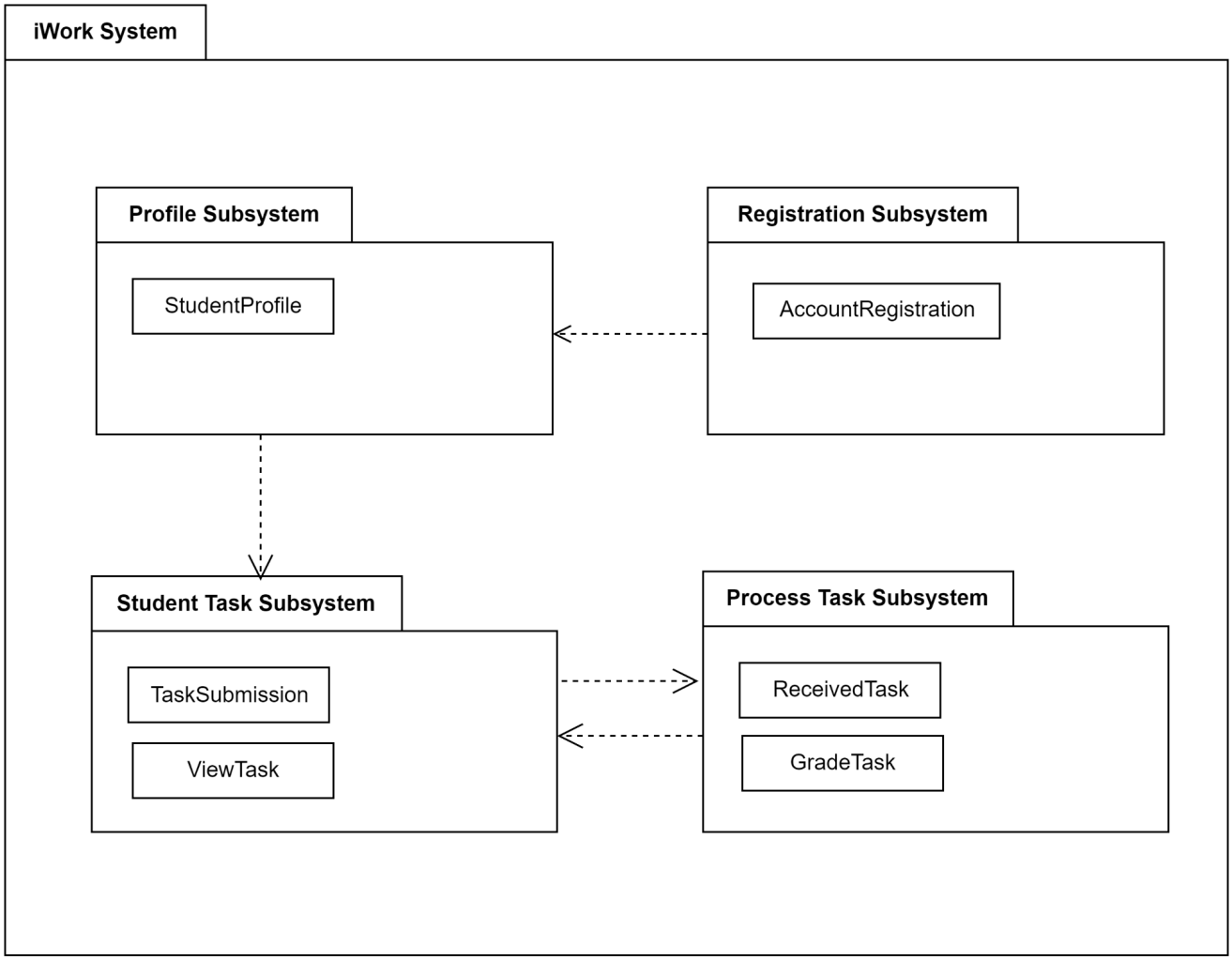


Figure 4.1: Package Diagram for iWork System

4.2 Detailed Description

iWork system consists of 4 subsystems: profile, register, student task, process task subsystems. Each subsystem is divided into view, domain, data access. The profile subsystem consists of StudentProfile which consists of view profile and edit profile. The view profile allows users to view students' profiles and edit profile allows students to edit their profiles thus saved into the subsystem. Next, the Registration subsystem consists of AccountRegistration which has a register. It allows users to register thus updated into the subsystem. The Student Task subsystem consists of TaskSubmission and ViewTask. TaskSubmission allows students to submit their finished tasks to the subsystem. ViewTask allows students to view tasks to do. Lastly, the Process Task subsystem consists of ReceivedTask and GradeTask. ReceivedTask allows students to receive already finished tasks from the supervisors. GradeTask allows students to view the grade of the tasks.

4.2.1 P001: Profile Subsystem

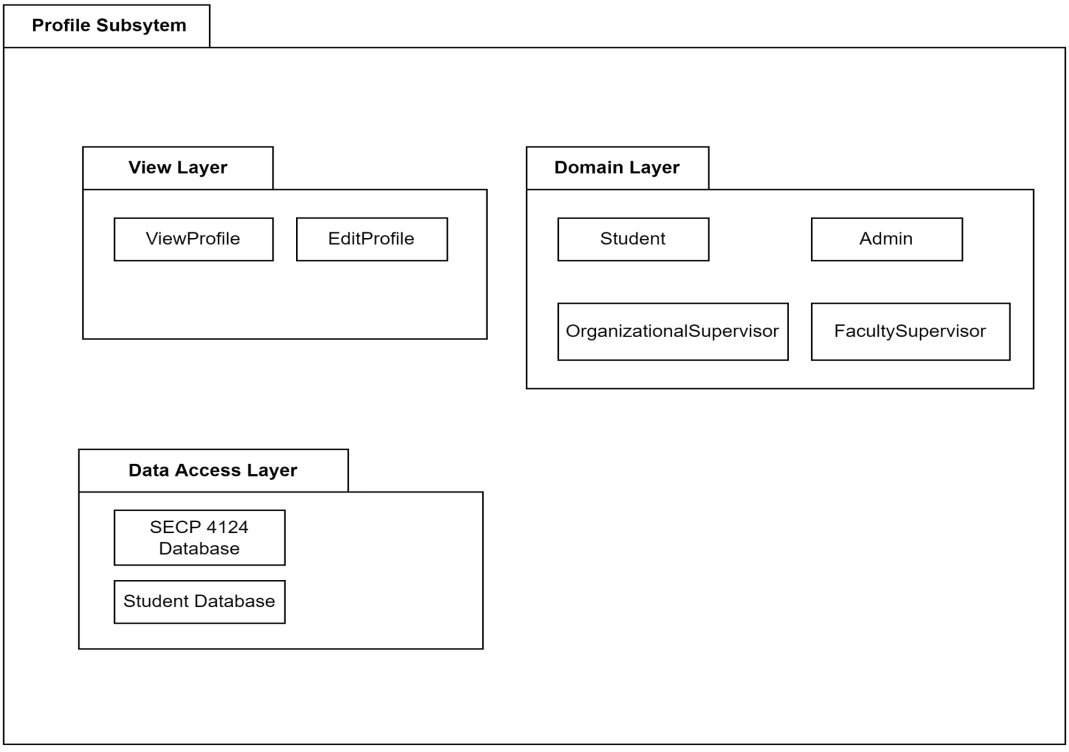


Figure 4.2: Package Diagram for Profile Subsystem

4.2.1.1 Class Diagram

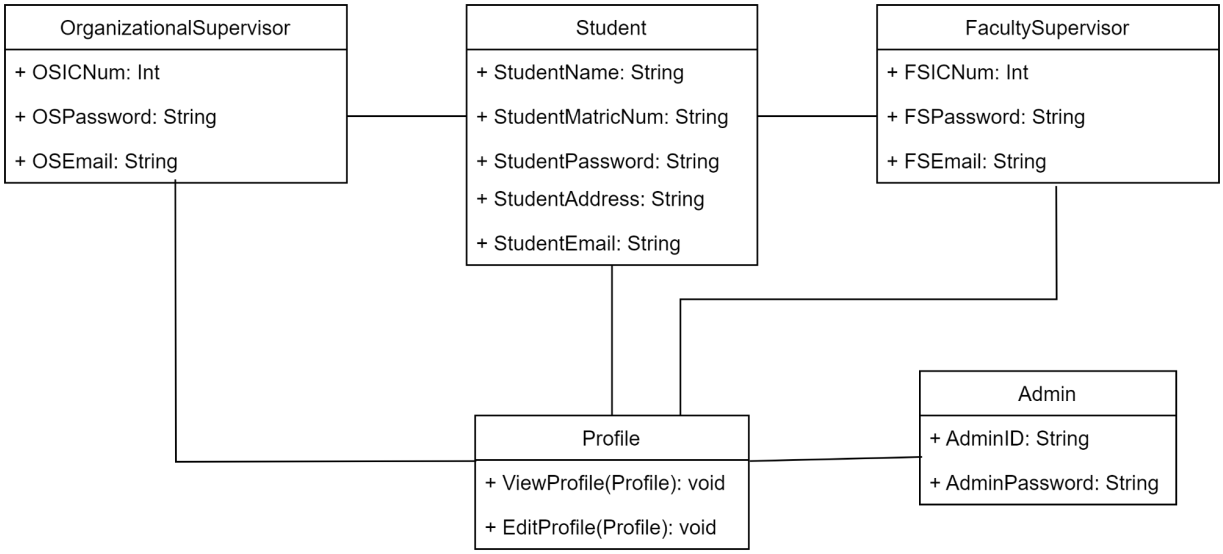


Figure 4.3: Class Diagram for Profile Subsystem

Entity Name	Profile
Method Name	ViewProfile()
Input	-
Output	Display student profile
Algorithm	1. Start 2. Login 3. Click on Student Profile 4. View student profile 5. End

Entity Name	Profile
Method Name	EditProfile()
Input	New student info
Output	New student profile info
Algorithm	1. Start 2. Login 3. Click on Student Profile 4. Edit profile name, matric number, address, email 5. Save profile 6. End

4.2.1.2 Sequence Diagram

a) SD001: Sequence diagram for View Profile

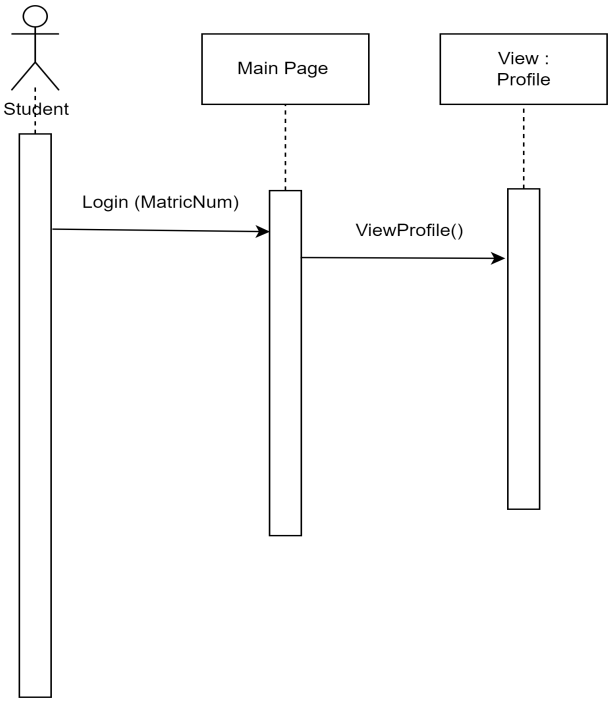


Figure 4.4: Sequence Diagram for <View Profile>

b) SD002: Sequence diagram for Edit Profile

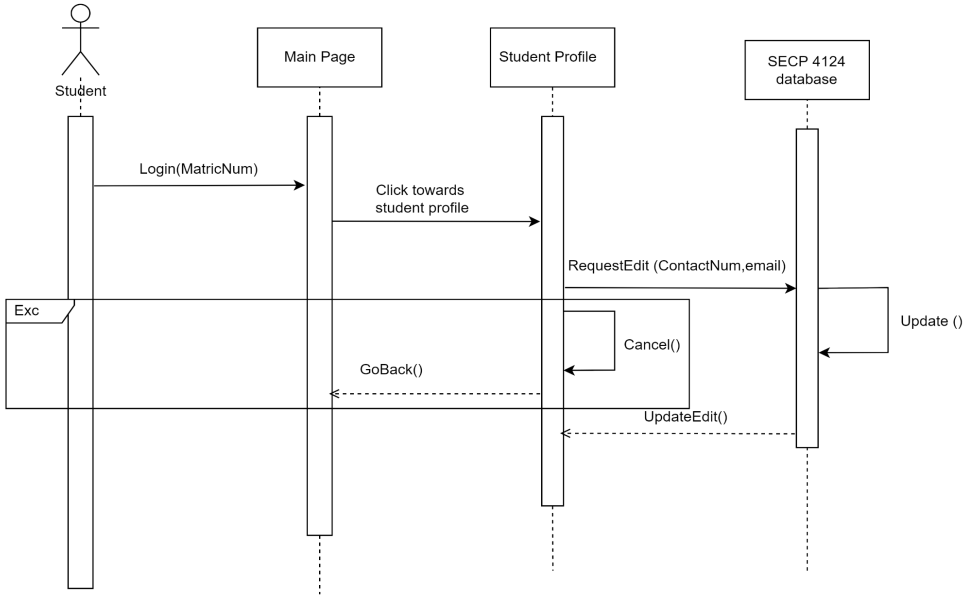


Figure 4.5: Sequence Diagram for <Edit Profile>

4.2.2 **P002: Registration Subsystem**

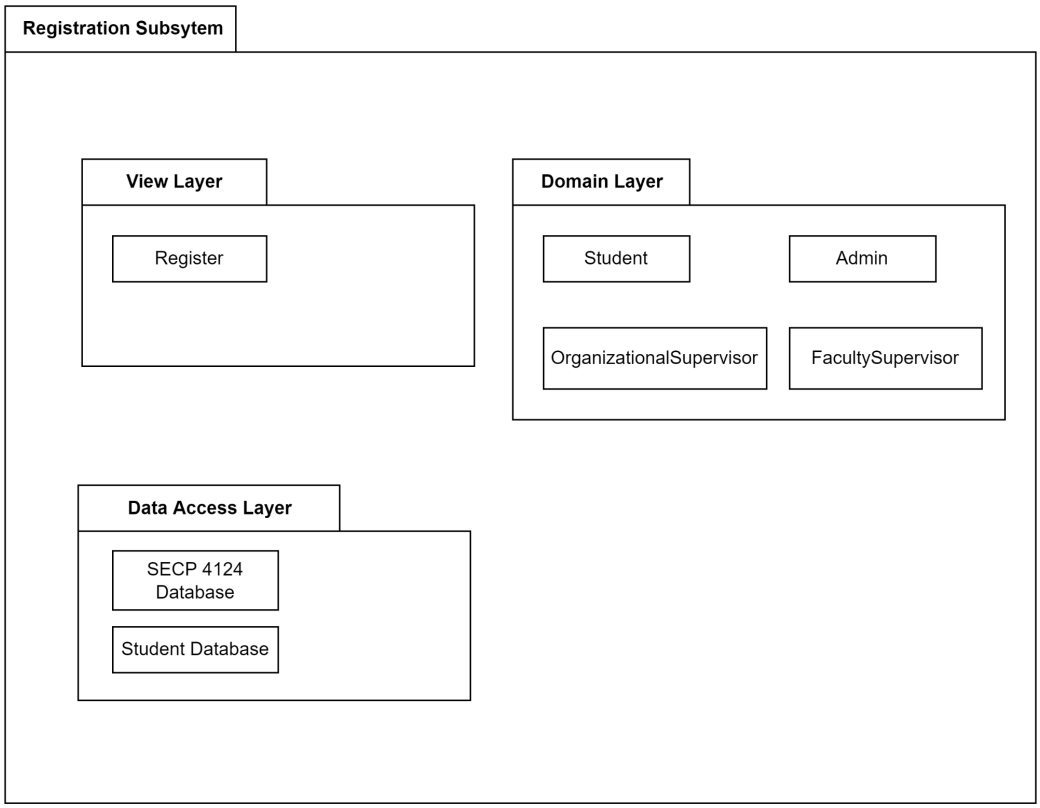


Figure 4.6: Package Diagram for Registration Subsystem

4.2.2.1 **Class Diagram**

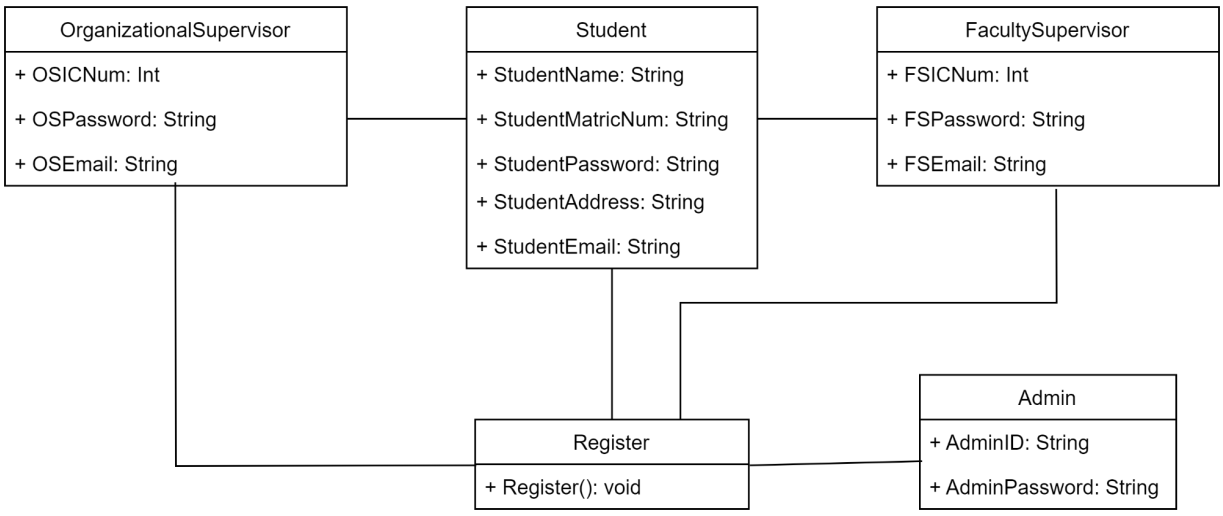


Figure 4.7: Class Diagram for Registration Subsystem

4.2.2.2 Sequence Diagram

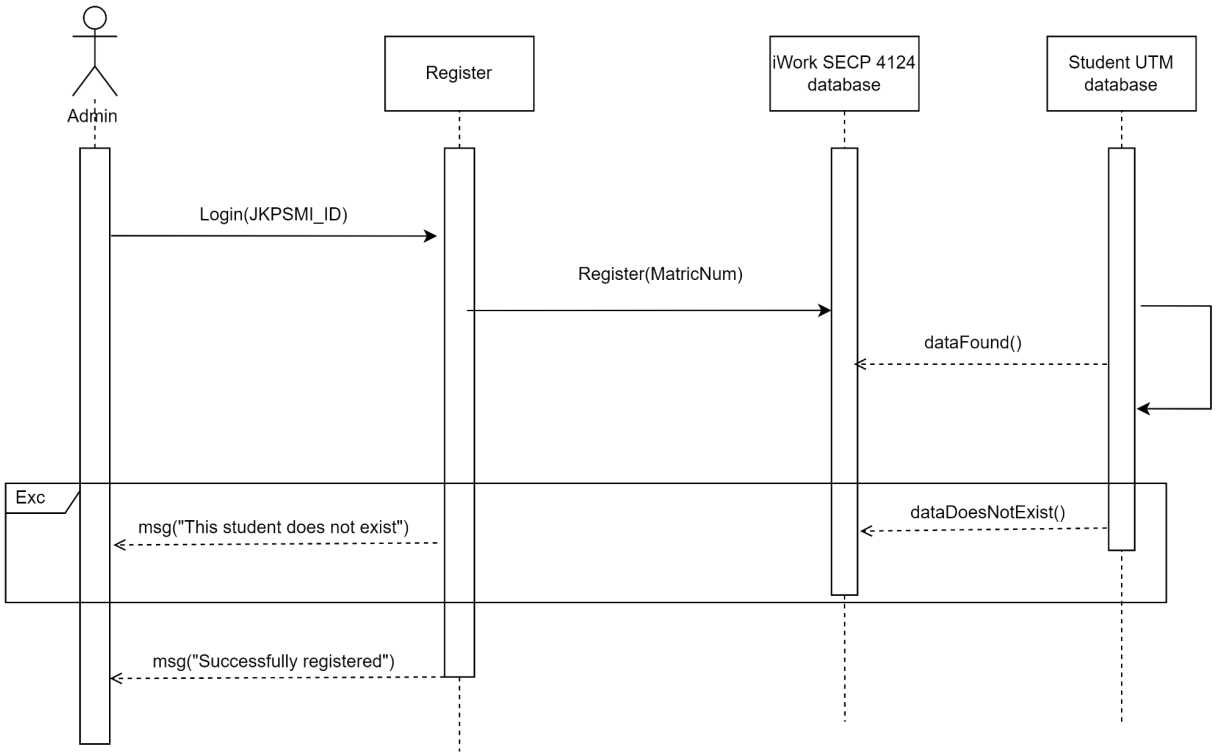


Figure 4.8: Sequence Diagram for <Register>

Entity Name	Register
Method Name	Register()
Input	Register new user
Output	New user registered
Algorithm	1. Start 2. Login 3. If cannot login, click register for new user 4. Enter info to register 5. Successfully registered 6. End

4.2.3 **P003: Student Task Subsystem**

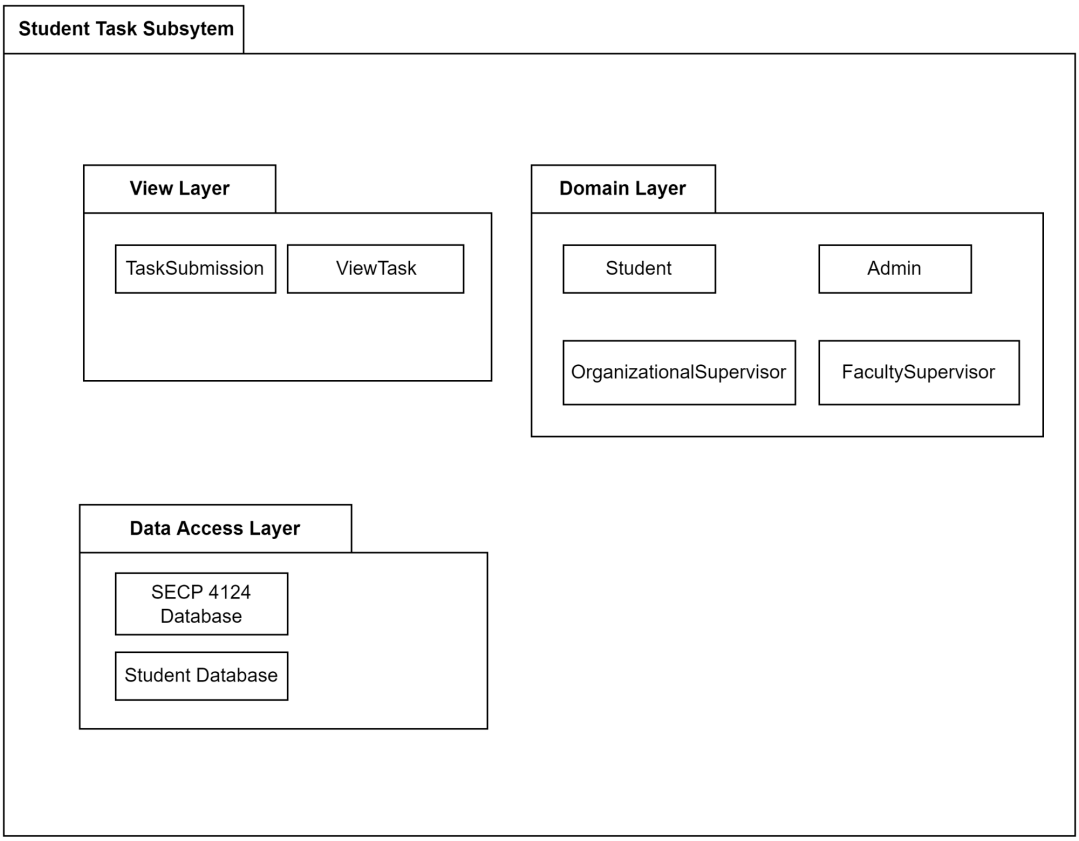


Figure 4.9: Package Diagram for Student Task Subsystem

4.2.3.1 Class Diagram

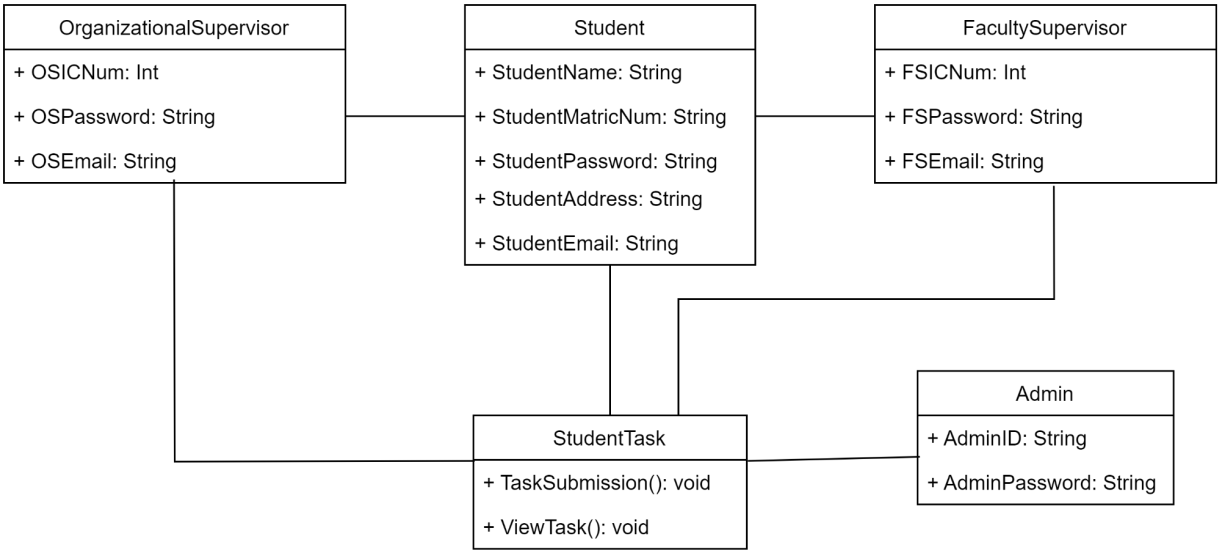


Figure 4.10: Class Diagram for Student Task Subsystem

4.2.3.2 Sequence Diagram

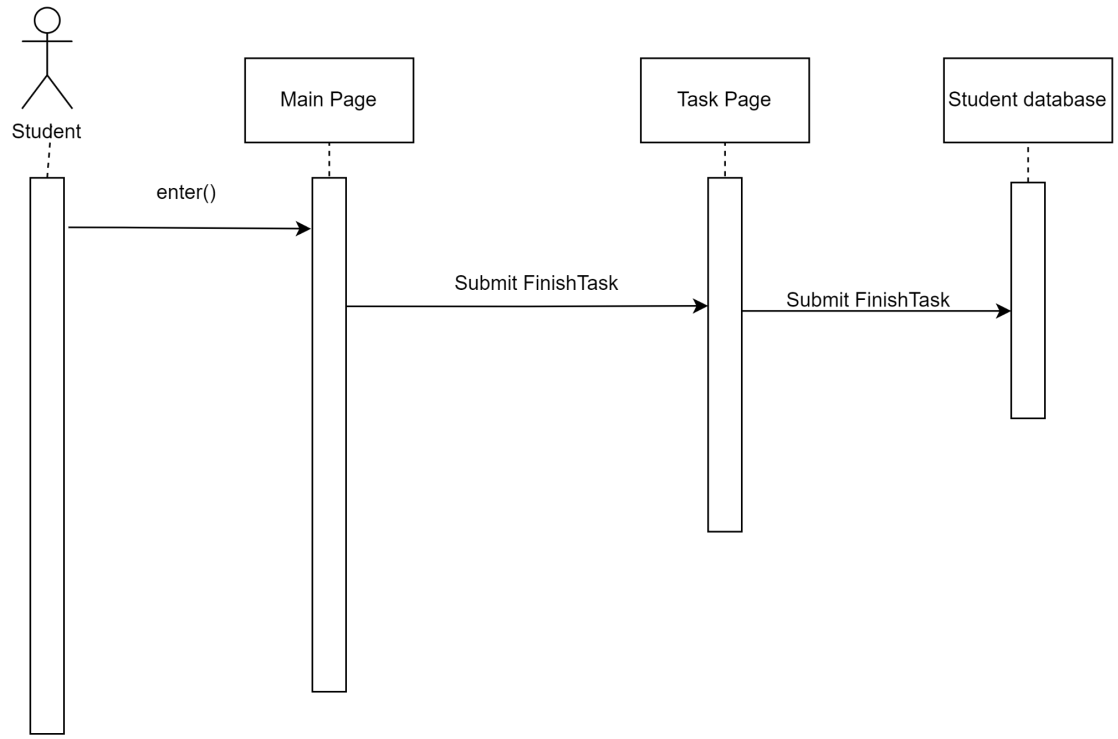


Figure 4.11: Sequence Diagram for TaskSubmission

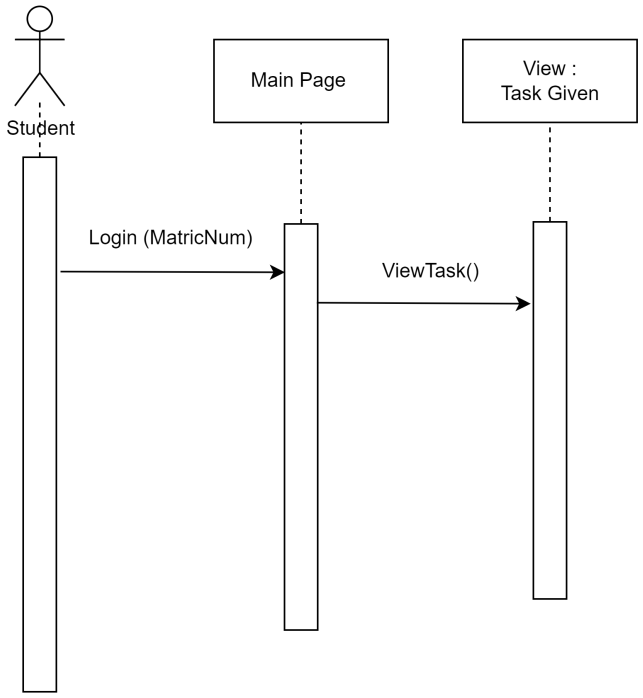


Figure 4.12: Sequence Diagram for ViewTask

Entity Name	Task
Method Name	TaskSubmission()
Input	Submit finish task
Output	Task submitted
Algorithm	1. Start 2. Login 3. Click on Submit tasks 4. Confirm submit 5. Task submitted successfully 6. End

Entity Name	Task
Method Name	ViewTask()
Input	-
Output	View remaining tasks
Algorithm	1. Start 2. Login 3. Click on tasks 4. View tasks

	5. End

4.2.4 **P004: Process Task Subsystem**

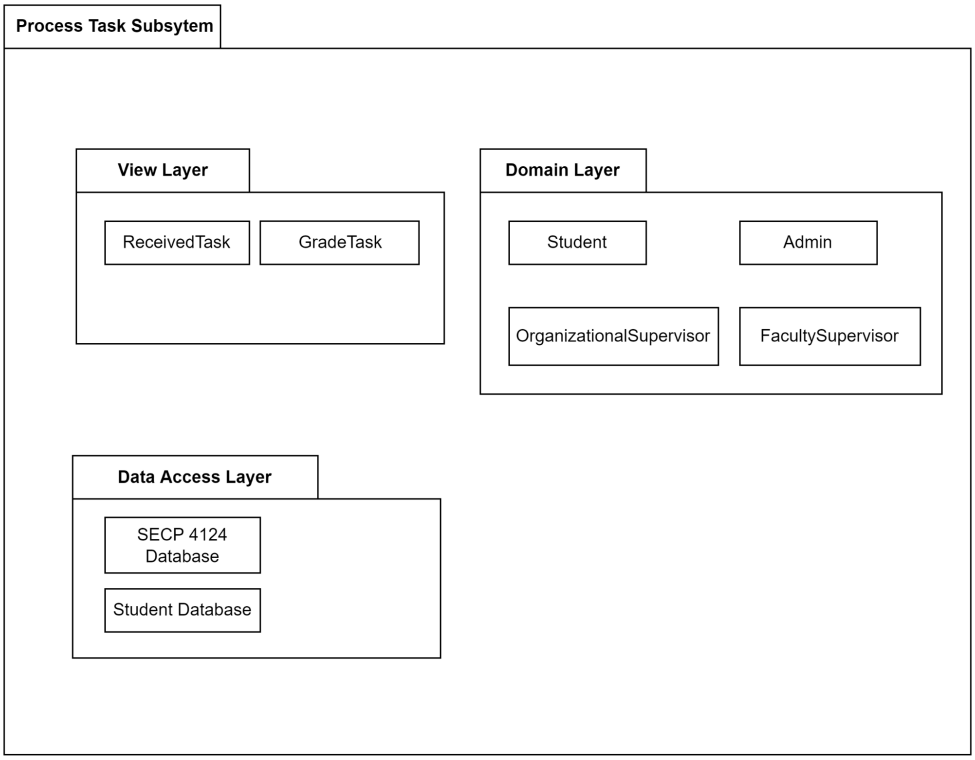


Figure 4.13: Package Diagram for Process Task Subsystem

4.2.4.1 **Class Diagram**

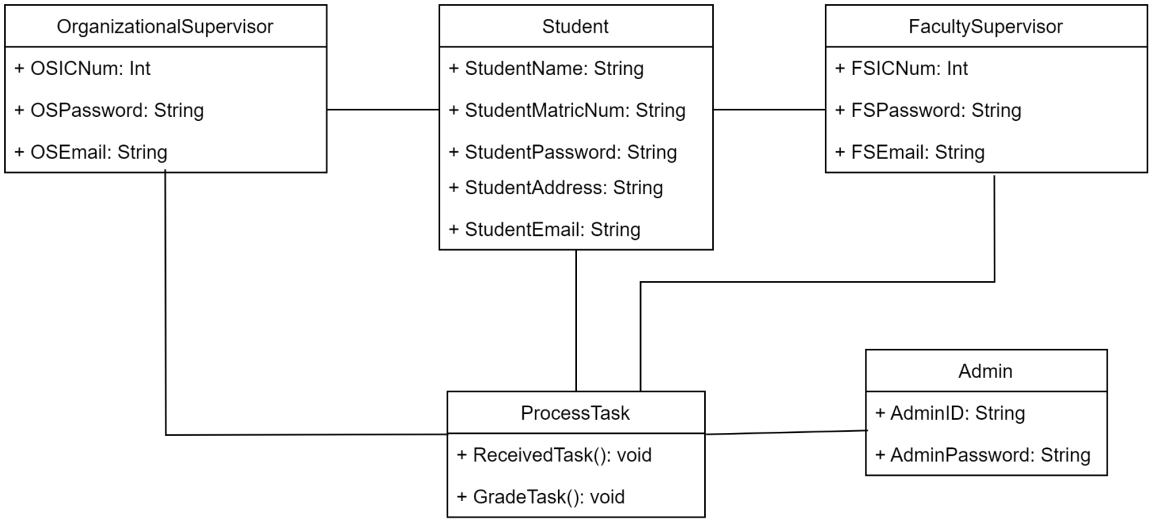


Figure 4.14: Class Diagram for Process Task Subsystem

4.2.4.2 Sequence Diagram

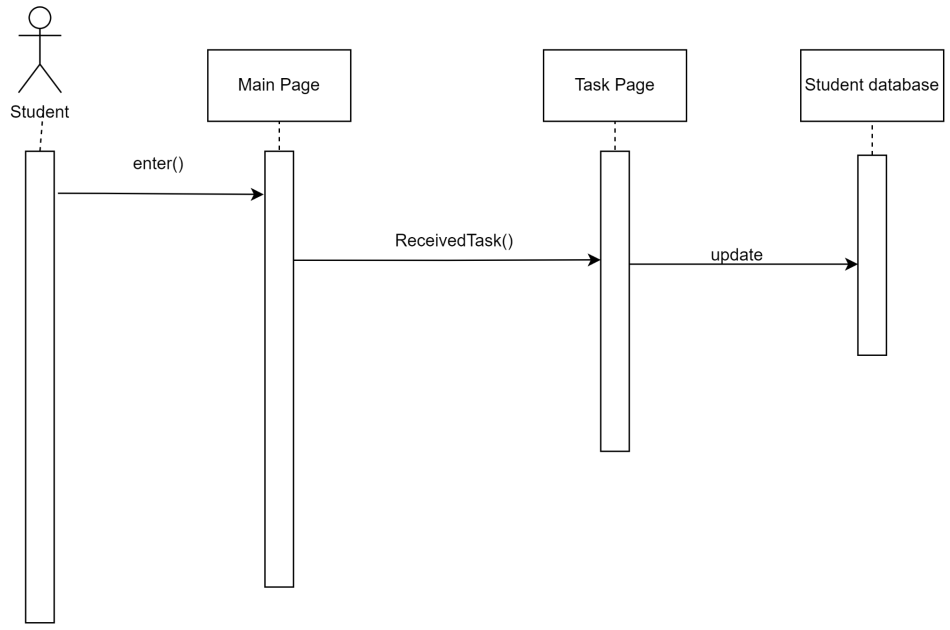


Figure 4.15: Sequence Diagram for ReceivedTask

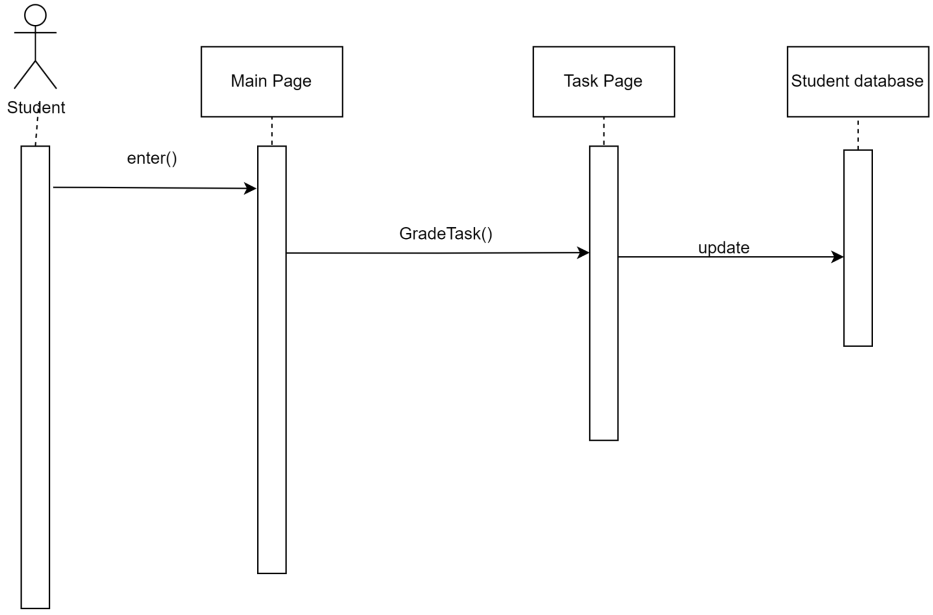


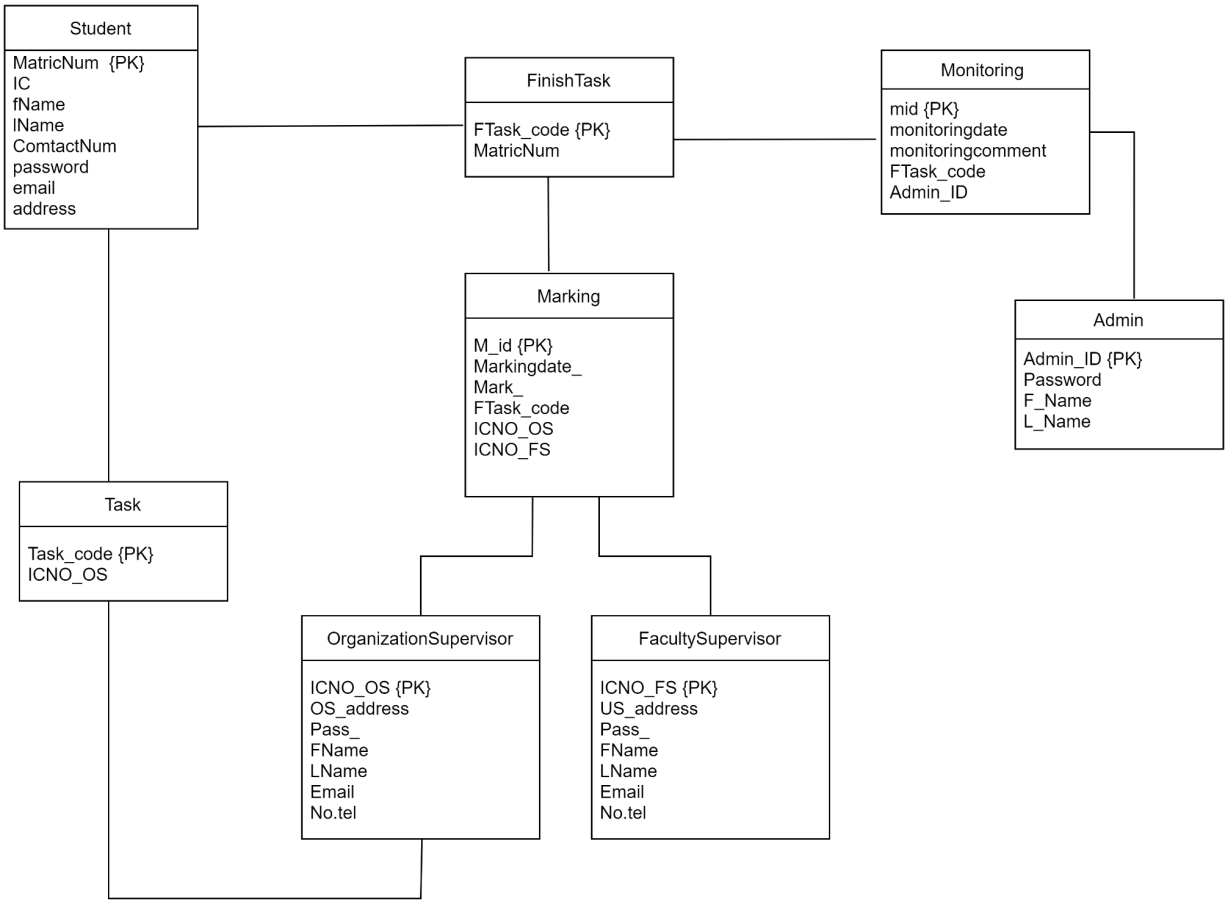
Figure 4.16: Sequence Diagram for GradeTask

Entity Name	ProcessTask
Method Name	ReceivedTask()
Input	-
Output	Receive finished task from supervisors
Algorithm	1. Start 2. Login 3. Click on tasks 4. Click on received task 5. View the received tasks 6. End

Entity Name	Task
Method Name	GradeTask()
Input	-
Output	View grade for the tasks
Algorithm	1. Start 2. Login 3. Click on tasks 4. Click on grade tasks 5. View the grade of the tasks 6. End

5. Data Design

5.1 Data Description



The major data or systems entities are stored into a relational database named as system database, processed and organized into *8* entities as listed in Table 5.1.

Table 5.1: Description of Entities in the Database

No.	Entity Name	Description
1.	Student	Entity consists of data related to student
2.	OrganizationSupervisor	Entity consists of data related to organizational supervisor
3.	FacultySupervisor	Entity consists of data related to faculty supervisor
4.	Admin	Entity consists of data related to admin
5.	Task	Entity consists of data related to task
6.	FinishTask	Entity consists of data related to finish task
7.	Marking	Entity consists of data related to marking

8.	Monitoring	Entity consists of data related to monitoring

5.2 Data Dictionary

5.2.1 Entity: Student

Attribute Name	Type	Description
MatricNum	varchar	Unique identifier for students
IC	int	Identifier for students
fName	String	First name of students
lName	String	Last name of students
ContactNum	int	Contact number for students
password	varchar	Password for students
email	varchar	Email for students
address	varchar	Address for students

5.2.2 Entity: OrganizationalSupervisor

Attribute Name	Type	Description
ICNO_OS	int	Unique identifier for organizational supervisor
OS_address	varchar	Address for organizational supervisor
Pass_	varchar	Password for organizational supervisor
FName	String	First name for organizational supervisor
LName	String	Last name for organizational supervisor
Email	varchar	Email for organizational supervisor
Notel	int	Phone number for organizational supervisor

5.2.3 Entity: FacultySupervisor

Attribute Name	Type	Description
ICNO_OS	int	Unique identifier for faculty supervisor
OS_address	varchar	Address for faculty supervisor
Pass_	varchar	Password for faculty supervisor
FName	String	First name for faculty supervisor
LName	String	Last name for faculty supervisor
Email	varchar	Email for faculty supervisor
Notel	int	Phone number for faculty supervisor

5.2.4 Entity: Admin

Attribute Name	Type	Description
Admin_ID	varchar	Unique identifier for admin

Password	varchar	Password for admin
F_Name	String	First name for admin
L_Name	String	Last name for admin

5.2.5 **Entity: Task**

Attribute Name	Type	Description
Task_code	varchar	Unique identifier for task
ICNO_OS	int	Unique identifier for organizational supervisor

5.2.6 **Entity: FinishTask**

Attribute Name	Type	Description
FTask_code	varchar	Unique identifier for finish task
MatricNum	varchar	Unique identifier for students

5.2.7 **Entity: Marking**

Attribute Name	Type	Description
M_id	varchar	Unique identifier for marking
Markingdate_	varchar	Address for marking
Mark_	int	Final mark for the task
FTask_code	varchar	Unique identifier for finish task
ICNO_OS	int	Unique identifier for organizational supervisor
ICNO_FS	int	Unique identifier for faculty supervisor

5.2.8 **Entity: Monitoring**

Attribute Name	Type	Description
mid	varchar	Unique identifier for monitoring
monitoringdate	varchar	Address for monitoring
monitoringcomment	varchar	Comment for monitoring
FTask_code	varchar	Unique identifier for finish task
Admin_ID	varchar	Unique identifier for admin