

# **Final Year Project Management and Tracking System**

**Hina Mahmood Qureshi** (BCS163008)

**Muhammad Haroon Qureshi** (BCS163009)



**Spring-2020**

**Supervised By**

**Mr. Samir Obaid**

**Department of Computer Science**

**Capital University of Science & Technology, Islamabad**

Submission Form for Final-Year

# PROJECT REPORT



Version	V 1.4
---------	-------

NUMBER OF MEMBERS	2
----------------------	---

TITLE	Final Year Project Management and Tracking System
-------	---

SUPERVISOR NAME	Mr. Samir Obaid
--------------------	-----------------

MEMBER NAME	REG. NO.	EMAIL ADDRESS
Hina Mahmood Qureshi	BCS163008	hinaqureshi051@gmail.com
Muhammad Haroon Qureshi	BCS163009	haroonqureshi60@gmail.com

MEMBERS' SIGNATURES
---------------------

---

---

---

Supervisor's Signature
------------------------

2

# **APPROVAL CERTIFICATE**

This project, entitled as “Final Year Project Management and Tracking System” has been approved for the award of

## **Bachelors of Science in Computer Science**

**Committee Signatures:**

Supervisor: \_\_\_\_\_

(Mr. Samir Obaid)

Project Coordinator: \_\_\_\_\_

(Mr. Ibrar Arshad)

Head of Department: \_\_\_\_\_

(Dr. Nayyer Masood)

## **DECLARATION**

*I/We, hereby, declare, "No portion of the work referred to, in this project has been submitted in support of an application for another degree or qualification of this or any other university/institute or other institution of learning". It is further declared that this undergraduate project, neither as a whole nor as a part there of has been copied out from any sources, wherever references have been provided.*

### **MEMBERS' SIGNATURES**

---

---

---

## **ACKNOWLEDGEMENTS**

*We would like to express our great appreciation to Mr.Samir Obaid who has helped us throughout the whole Period by giving us valuable suggestions during our planning and development phase.*

# **Executive Summary**

The project “Final Year Project Management and Tracking System” is a web-based modular, which will help students, supervisors and coordinator to perform final year project activities. This will allow students, supervisors and coordinators to communicate through announcements. Coordinator will be performing Final year activities by creating a Term, Program Learning Objectives, Course Learning Objectives, Exams, and Questions etc. Only Coordinator will be able to register the students and the supervisors. Supervisors will be making the groups by selecting students and assigning them project title and sending the email to the coordinator about this project group. Students are registering themselves for the final Year Project, can view related questions to the Exams and emailing the documents to the project panel.

## Table of Contents

Chapter 1 .....	17
Introduction .....	17
1.1. Project Introduction .....	17
1.2. Existing Examples / Solutions .....	19
1.3. Business Scope .....	19
1.4. Useful Tools and Technologies .....	20
1.5. Project Work Break Down .....	20
1.6. Project Time Line .....	21
Chapter 2 .....	23
Requirement Specification and Analysis .....	23
2.1. Functional Requirements .....	23
2.2. Non-Functional Requirements .....	24
2.3. Selected Functional Requirements .....	25
2.4. System Use Case Modeling .....	26
2.5. System Sequence Diagrams .....	52
2.6. Domain Model .....	65
Chapter 3 .....	66
System Design .....	66
3.1. Software Architecture .....	66
3.2. Class Diagram.....	68
3.3. Sequence Diagram .....	69
3.4. Entity Relationship Diagram .....	87
3.5. Database Schema .....	88
3.6. User Interface Design.....	89
Chapter 4 .....	102

Software Development .....	102
4.1. Coding Standards .....	102
4.2. Development Environment.....	102
4.3. Software Description .....	102
Chapter 5 .....	127
Software Testing .....	127
5.1. Testing Methodology.....	127
5.2. Test Cases .....	127
Chapter 6 .....	151
Software Deployment .....	151
6.1. Installation / Deployment Process Description .....	151
Chapter 7 .....	152
Project Evaluation .....	152
7.1. Project Evaluation Report.....	152
References.....	153

## **List of Figures**

- Figure 1.1 Work Breakdown
- Figure 1.2 Project Timeline Part 1
- Figure 1.3 Project Timeline Part 2
- Figure 2.1 Use Case Diagram
- Figure 2.2 Coordinator Sign Up
- Figure 2.3 Coordinator Login
- Figure 2.4 Supervisor Login
- Figure 2.5 Student Login
- Figure 2.6 Register Supervisor and Students
- Figure 2.7 Manage Term
- Figure 2.8 Manage Program Learning Objectives
- Figure 2.9 Manage Course Learning Objectives
- Figure 2.10 Manage Exams
- Figure 2.11 Manage Questions
- Figure 2.12 Manage Supervisor's Panel
- Figure 2.13 Manage Supervisors
- Figure 2.14 Manage Announcements
- Figure 2.15 View Announcements by Supervisor
- Figure 2.16 View Announcements by Student
- Figure 2.17 Create Group
- Figure 2.18 Assign Project Title
- Figure 2.19 Send Email by Supervisor
- Figure 2.20 View Project and Panel Details

Figure 2.21 View Questions by Supervisor

Figure 2.22 View Questions by Student

Figure 2.23 Register Student

Figure 2.24 View Supervisor's List

Figure 2.25 Send Email by Student

Figure 2.26 Domain Model

Figure 3.1 Tier Architecture

Figure 3.2 Tier 1 Architecture

Figure 3.3 Layered Architecture

Figure 3.4 Class Diagram

Figure 3.5 Coordinator Sign Up

Figure 3.6 Coordinator Login

Figure 3.7 Supervisor Login

Figure 3.8 Student Login

Figure 3.9 Register Supervisors and Students

Figure 3.10 Create Term

Figure 3.11 Create Program Learning Objectives

Figure 3.12 Create Course Learning Objectives

Figure 3.13 Create Exam

Figure 3.14 Create Question

Figure 3.15 Create Panel

Figure 3.16 Create Supervisor

Figure 3.17 Create Announcement

Figure 3.18 View Announcements by Supervisor

Figure 3.19 View Announcements by Student

Figure 3.20 Create Group

Figure 3.21 Assign Project Title

Figure 3.22 Send Email by Supervisor

Figure 3.23 View Project and Panel Details

Figure 3.24 View Question by Supervisor

Figure 3.25 View Question by Student

Figure 3.26 Register Student

Figure 3.27 View Supervisor's List

Figure 3.28 Send Email by Student

Figure 3.29 Entity Relationship Diagram

Figure 3.30 Database Schema

Figure 3.31 Main Page

Figure 3.32 Coordinator Sign Up

Figure 3.33 Coordinator Login

Figure 3.34 Supervisor Login

Figure 3.35 Student Login

Figure 3.36 Register Supervisors and Student

Figure 3.37 Manage Term

Figure 3.38 Manage Program Learning Objectives

Figure 3.39 Manage Course Learning Objectives

Figure 3.40 Manage Exams

Figure 3.41 Manage Questions

Figure 3.42 Manage Supervisor's Panel

Figure 3.43 Manage Supervisors

Figure 3.44 Manage Announcements

Figure 3.45 View Announcements by Supervisor

Figure 3.46 View Announcements by Student

Figure 3.47 Create Group

Figure 3.48 Assign Project Title

Figure 3.49 Send Email by Supervisor

Figure 3.50 View Project and Panel Details

Figure 3.51 View Questions by Supervisor

Figure 3.52 View Questions by Student

Figure 3.53 Register Student

Figure 3.54 View Supervisor's List

Figure 3.55 Send Email by Student

## **List of Tables**

- Table 2.1 Functional Requirements
- Table 2.2 Non-Functional Requirements
- Table 2.3 Selected Functional Requirements
- Table 2.4 Coordinator Sign Up
- Table 2.5 Login
- Table 2.6 Register Supervisor and Students
- Table 2.7 Manage Terms
- Table 2.8 Manage Program Learning Objectives
- Table 2.9 Manage Course Learning Objectives
- Table 2.10 Manage Exams
- Table 2.11 Manage Questions
- Table 2.12 Manage Supervisor's Panel
- Table 2.13 Manage Supervisors
- Table 2.14 Manage Announcements
- Table 2.15 View Announcements
- Table 2.16 Create Group
- Table 2.17 Assign Project Title
- Table 2.18 Send Email by Supervisor
- Table 2.19 View Project and Panel Details
- Table 2.20 View Questions
- Table 2.21 Register Student
- Table 2.22 View Supervisors List
- Table 2.23 Send Email by Student
- Table 5.1 Sign Up Coordinator with matching password and Confirm password
- Table 5.2 Sign Up Coordinator with not matching password and Confirm password
- Table 5.3 Login Coordinator with email address and password
- Table 5.4 Login Coordinator with wrong email address and password
- Table 5.5 Login Supervisor with email address and password
- Table 5.6 Login Supervisor with wrong email address and password
- Table 5.7 Login Student with email address and password
- Table 5.8 Login Student with wrong email address and password
- Table 5.9 Register Supervisor with email address, matching password and confirm password
- Table 5.10 Register Supervisor with email address, not matching password and confirm password

- Table 5.11 Register Student with email address, matching password and confirm password
- Table 5.12 Register Student with email address, not matching password and confirm password
- Table 5.13 Create Term with valid Inputs
- Table 5.14 Create Term with invalid Inputs
- Table 5.15 Edit Term
- Table 5.16 Delete Term
- Table 5.17 Create Program Learning Objective with valid Inputs
- Table 5.18 Create Program Learning Objective with invalid Inputs
- Table 5.19 Edit Program Learning Objective
- Table 5.20 Delete Program Learning Objective
- Table 5.21 Create Course Learning Objective with valid Inputs
- Table 5.22 Create Course Learning Objective with invalid Inputs
- Table 5.23 Edit Course Learning Objective
- Table 5.24 Delete Course Learning Objective
- Table 5.25 Create Exam with valid Inputs
- Table 5.26 Create Exam with invalid Inputs
- Table 5.27 Edit Exam
- Table 5.28 Delete Exam
- Table 5.29 Create Questions with valid Inputs
- Table 5.30 Create Questions with invalid Inputs
- Table 5.31 Edit Questions
- Table 5.32 Delete Questions
- Table 5.33 Create Panel with valid Inputs
- Table 5.34 Create Panel with invalid Inputs
- Table 5.35 Edit Panel
- Table 5.36 Delete Panel
- Table 5.37 Create Supervisor with valid Inputs
- Table 5.38 Create Supervisor with invalid Inputs
- Table 5.39 Edit Supervisor
- Table 5.40 Delete Supervisor
- Table 5.41 Create Announcement with valid inputs
- Table 5.42 Edit Announcements
- Table 5.43 Delete Announcements
- Table 5.44 Create Group

Table 5.45 Create Group with invalid inputs

Table 5.46 Assign Project Title

Table 5.47 Register Student with valid inputs

Table 5.48 Register Student with invalid inputs



# Chapter 1

## Introduction

“Final Year Project Management and Tracking System” is a web based modular management system that manages all the aspects of final year project operations. This management system covers all the data and activities of project coordinator, supervisors and students. The Project coordinator will be the one doing all the activities related to a term and exam. The supervisors making the groups, projects and email it to the coordinator about this. Students register themselves, viewing question with respect to exams, and emailing the artifacts to supervisors of panel.

### 1.1. Project Introduction

A web based final year project management and tracking system facilitating project coordinator, supervisors and students in all activities. Final year project management and tracking system handles the issues in delay of selecting supervisors and groups. Final year project management and tracking system is a management system that will handle supervisor and student registration by coordinator, Project activities management by coordinator, group management, online registration, and proposal and report submissions, exam notifications through announcements. This project will give the interface for project coordinator, supervisors, and project panel and students to perform their final year project activities in a manageable way.

The beneficiary of the project will be Capital University of Science and Technology for managing the final year project activities. This will give the organization a proper definition of work by using this web-based application.

**Roles:** The three main roles for “Final year project management and tracking system” project are as follows:

**Project Coordinator:** Project Coordinator is an organizational member that handles all the activities regarding final year project. Project coordinator will register students and supervisors, create term, add Program learning objectives

in terms, add Course Learning Objectives in terms, create exam types, add questions in exams, create a panel and assign supervisor a panel, add announcements.

**Supervisors:** Supervisors are the members that guide the students through whole process of project. Supervisors will view the announcements created by the coordinator, create a group, assign project title to a group, send email to the coordinator for project creation, view project details by term, monitors students' progress.

**Students:** Student can view announcements created by supervisor, register themselves for the final year project, view supervisor's list and their slots, email their documents to supervisor and panel members, view questions related to each exam.

**Modules:** The five modules of "Final year project management and tracking system" project are as follows:

**Supervisor and Student Registration:** Coordinator would do Supervisor and student registration. Any outsider will not be able to access the web modular by signing up.

**Project Activities:** Project activities involve creating a new term, adding program learning objectives, course learning objectives, creating a exam, adding a question to exam, making a project panel, adding a supervisor to panel and adding the announcement in terms would be done by coordinator.

**Group Management:** Student shall be able to register them online providing all the data. Supervisor would make a group involving all those registered students and assign them a project title.

**Proposal, Report and Presentation Submissions:** System shall provide a interface in which students would send emails to the supervisors and project panels involving project documents.

**Announcements:** Coordinator shall be able to notify to supervisors and students about exam dates, and group allotment or any news important notice.

## 1.2. Existing Examples / Solutions

There are many project management and tracking process but none of them are aligned with the define process of Capital University of Science and Technology. All the existing examples/solutions works with their own organizational needs/process but none of them are aligned with our Capital University of Science and Technology final year project management system.

The existing example for current problems is as follows:

- **CSCUST** is web based project management tool
  - Helps student to upload documents.
  - Helps coordinators to assign project to supervisors.
  - Help Project panel to asses' projects.

### **Limitation:**

Tool does not facilitate to register students and asses project panel by students. It does not facilitate the supervisors to make a group and assign project title to it.

- **MOODLE** is open source campus management system with project management as a module.
  - Helps to maintain document.
  - Enables coordinator to notify students.

### **Limitation:**

However rather than managing separate groups the tool creates a signal group for entire batch.

## 1.3. Business Scope

The “Final year project management and tracking system” will be a liable and secure system that will help students and supervisors in computer science department of CUST to smoothly undertake and carry on undergraduate final year project. However, project features, as explained above, are generic and thus, it will possible to deploy the project in any department of CUST or in any other university.

## 1.4. Useful Tools and Technologies

We have planned to use following technologies for designing, development and testing of our project.

1. Programming Languages:
  - a. C# for backend development.
  - b. HTML and CSS for frontend design.
  - c. Ajax and JQuery.
2. Development Environment:
  - a. Visual Studio 2019
3. Database
  - a. SQL Server
4. Operating System: Window 10

## 1.5. Project Work Break Down

The project work break down structure is shown in Figure 1.1 Work

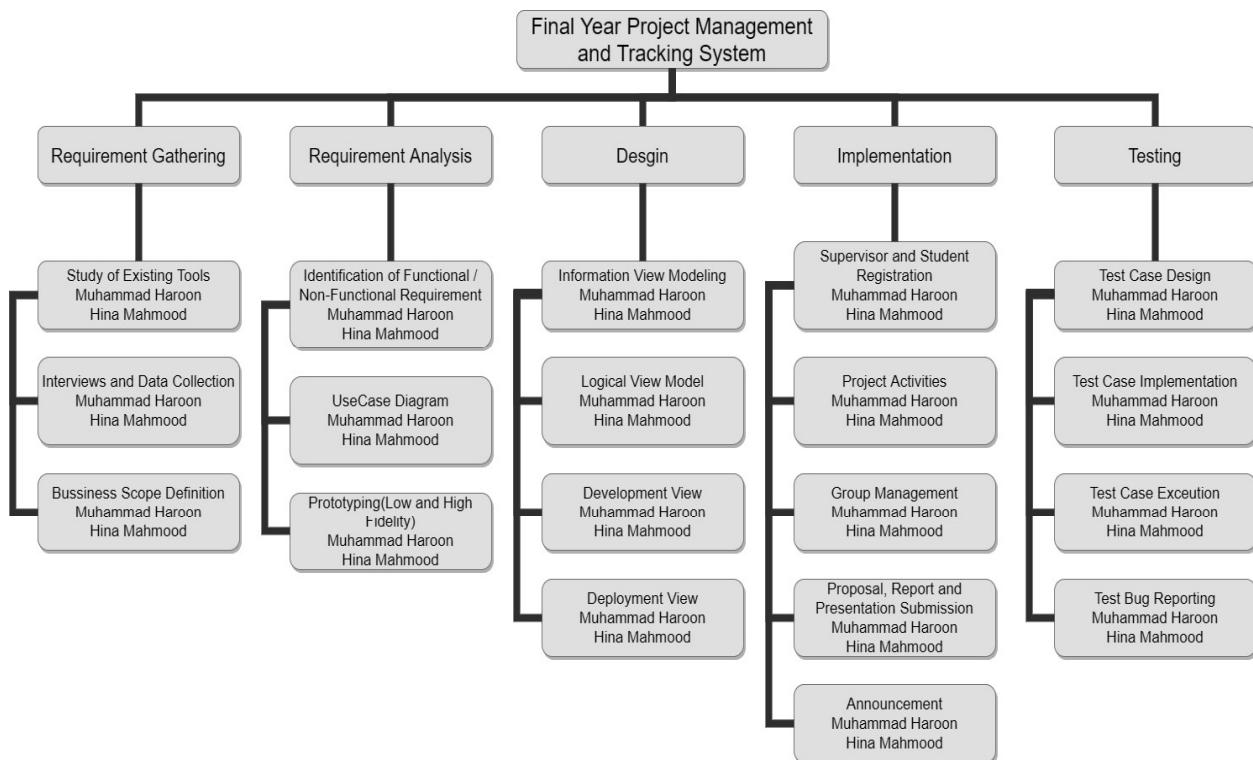


Figure 1.1 Work Breakdown

**Information View Modeling:** This includes identification of entities and their relationship, identification of classes, data ownership and information flow models.

**Logical View Model:** This includes execution sequence (sequence diagram) for different events identified from use case. This part will also focus on integration of system with different external entities such as database.

**Development View:** This part will focus on development structure such as Model, Views and Controller (MVC) and REST FULL web services (API's).

**Deployment View:** Deployment of the system on physical hardware and deploy of system with online system provider will be part of this.

## 1.6. Project Time Line

The project Gantt chart is shown in [Error! Reference source not found.](#) and in Figure 1.3.

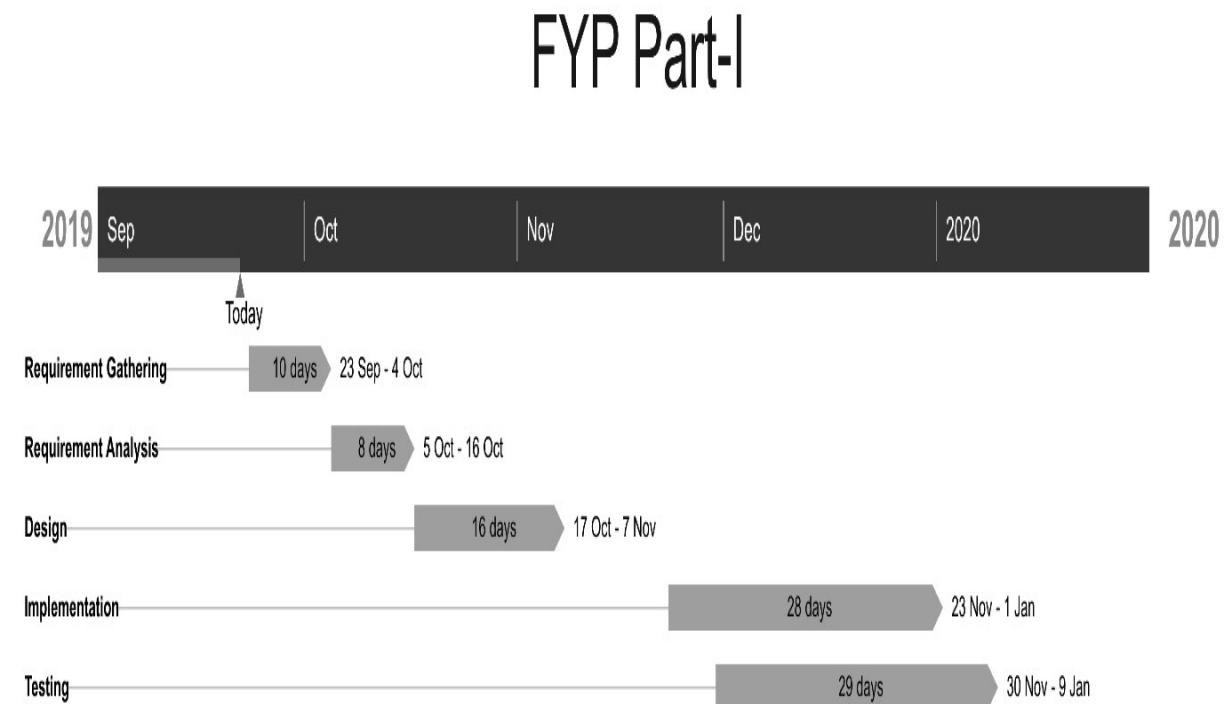


Figure 1-2 Project Timeline Part 1

## FYP Part-II

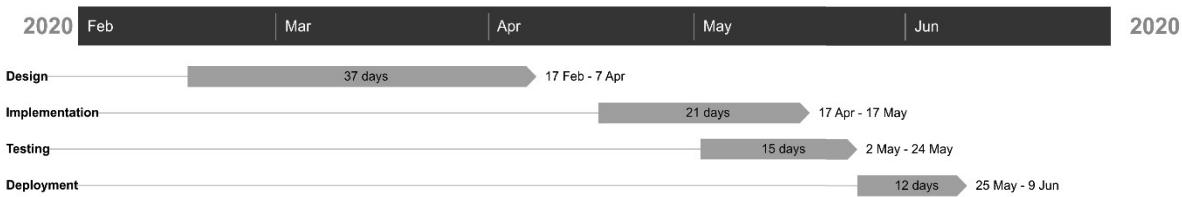


Figure 1-3 Project Timeline Part 2

# Chapter 2

## Requirement Specification and Analysis

### 2.1. Functional Requirements

Following Table 2.1 shows all the Functional Requirements of the project.

Table 2.1 Functional Requirements

S. No.	Functional Requirement	Type	Status
1	Coordinator can sign up to the system	Core	Implemented
2	User (Student, Supervisor and Coordinator) can login to the system	Core	Implemented
3	Coordinator can register supervisors and students	Core	Implemented
4	Coordinator can manage terms	Core	Implemented
5	Coordinator can manage Program Learning Objectives	Core	Implemented
6	Coordinator can manage Course Learning objectives	Core	Implemented
7	Coordinator can manage Exams	Core	Implemented
8	Coordinator can manage Questions	Core	Implemented
9	Coordinator can manage supervisor's panel	Core	Implemented
10	Coordinator can manage Supervisors	Core	Implemented
11	Coordinator can manage announcements	Core	Implemented
12	Supervisors can view announcements	Core	Implemented
13	Supervisors can create a group of students	Core	Implemented

14	Supervisors can assign project title to a group	Core	Implemented
15	Supervisors can send email to coordinator about project	Core	Implemented
16	Supervisors can view project details and panel details.	Core	Implemented
17	Supervisors can view questions to each exam	Core	Implemented
18	Student can view announcements	Core	Implemented
19	Students can register himself for the final year project	Core	Implemented
20	Students can view supervisor's List involving their name, slots and email id's	Core	Implemented
21	Students can send their documents and presentation via email to supervisors and project panel	Core	Implemented
22	Students can view questions to each exam	Core	Implemented

## 2.2. Non-Functional Requirements

Following Table 2.2 shows all the non-Functional Requirements.

Table 2.2 Non-Functional Requirements

S. No.	Non Functional Requirements	Category
1	It should be easy for students, supervisors, and coordinators to use the system with the help of helping forms provided by application without assistant of expert.	Operability in terms of usability.

2	System must ensure access control authorization.	Role Based Accessed Control and Authorization in terms of security.
---	--	---

## 2.3. Selected Functional Requirements

Following Table 2.3 shows all the selected functional requirements for current iteration.

Table 2.3-Selected Functional Requirement

S. No.	Selected Functional Requirement	Type
1	Coordinator can sign up to the system	Core
2	User (Student, Supervisor and Coordinator) can login to the system	Core
3	Coordinator can register supervisors and students	Core
4	Supervisors can send email to coordinator about project	Core
5	Student can view announcements	Core
6	Students can register himself for the final year project	Core
7	Students can view supervisor's List involving their name, slots and email id's	Core
8	Students can send their documents and presentation via email to supervisors and project panel	Core
9	Students can view questions to each exam	Core

## 2.4. System Use Case Modeling

Following Figure 2.1 shows the system use case model for Student, Supervisor and Coordinator.

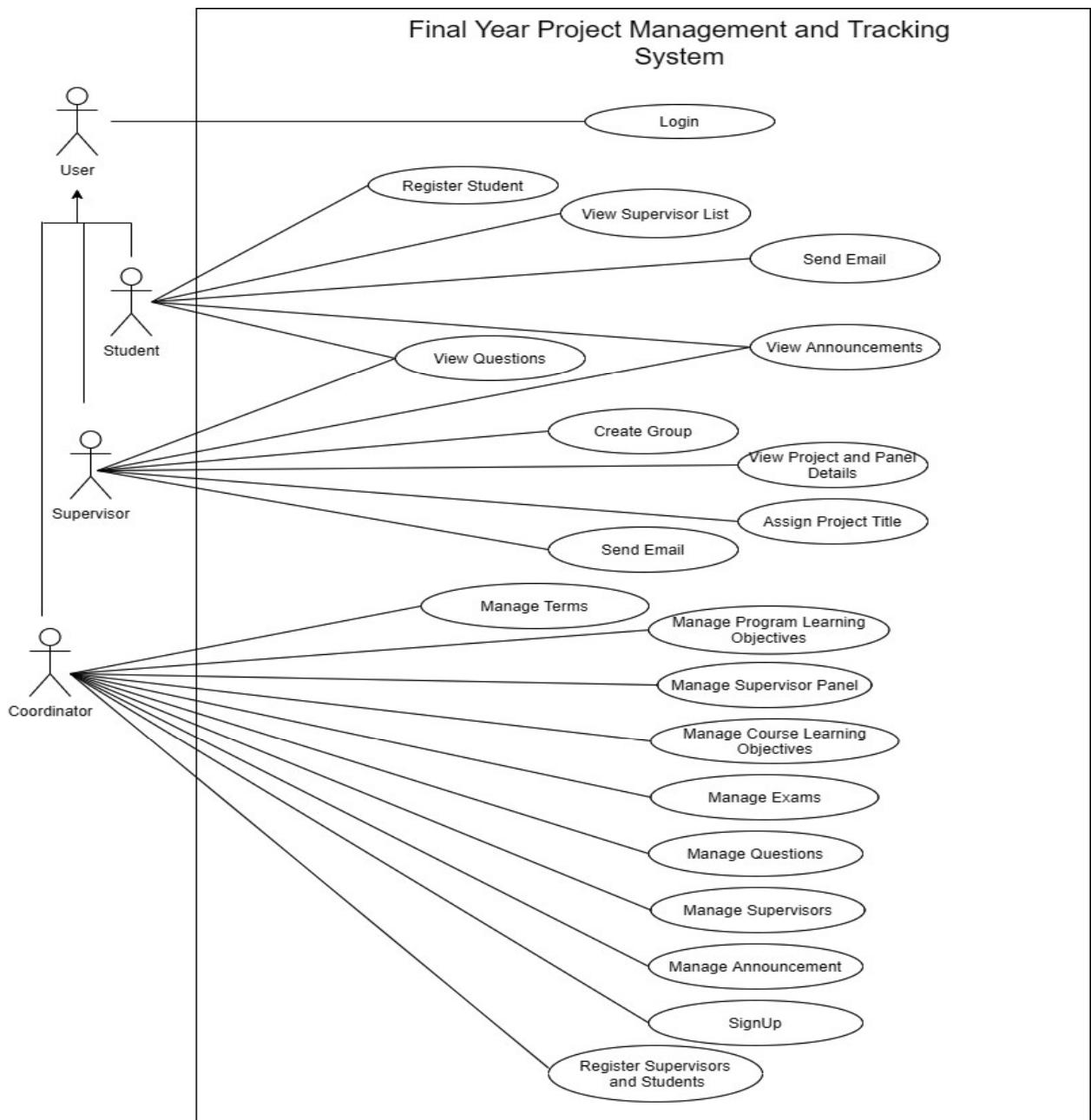


Figure 2.1 Use Case Diagram

## Use Case 1 Title: Sign Up

Coordinator can sign up into the application by providing email address; password and confirm password is shown in Table 2.4

Table 2.4 Coordinator Sign Up

<b>Use Case ID:</b>	1	
<b>Use Case Name:</b>	Sign Up	
<b>Created By:</b>	Haroon	<b>Last Updated By:</b> Hina
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b> November 15, 2019
<b>Actors:</b>	Coordinator	
<b>Description:</b>	Coordinator can sign up into the application by providing email address; password and confirm password.	
<b>Trigger:</b>	Signup Button.	
<b>Preconditions:</b>	Application should be in running state.	
<b>Post conditions:</b>	Coordinator is able to login to the application by providing registered email address and password	
<b>Normal Flow:</b>	Actor	System
	1) Coordinator clicks on sign up button. 3) Coordinator provides valid email address; password and confirm password 4) Coordinator clicks on sign up button.	2) System an interface in which the coordinator will be able to sign up to the system. 5) System navigates to login page.
<b>Alternative Flows:</b>	3) What happens if user provides incorrect password and	

	<p>confirm password?</p> <ol style="list-style-type: none"> <li>1. System notifies about invalid email/password.</li> <li>2. Coordinator provides correct password and confirm password.</li> <li>3. Coordinator clicks on sign up button</li> </ol>
--	--

## Use Case 2 Title: Login

User (Coordinator, Supervisor and Student) can login to the application by providing email address and password shown in Table 2.5

Table 2.5 Login

<b>Use Case ID:</b>	2		
<b>Use Case Name:</b>	Login		
<b>Created By:</b>	Haroon	<b>Last Updated By:</b>	Hina
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b>	November 15, 2019
<b>Actors:</b>	Coordinator, Supervisor and Student		
<b>Description:</b>	User (Coordinator, Supervisor and Student) can login to the application by providing email address and password.		
<b>Trigger:</b>	Login Button		
<b>Preconditions:</b>	User (Coordinator, Supervisor and Student) must be registered then provides email address and password for login to the application.		
<b>Post conditions:</b>	User (Coordinator, Supervisor and Student) is logged in to the application		
<b>Normal Flow:</b>	Actor	System	

	<p>1) User (Coordinator, Supervisor and Student) clicks on login button.</p> <p>3) User (Coordinator, Supervisor and Student) provides registered email address and password.</p> <p>4) User (Coordinator, Supervisor and Student) clicks on login button</p>	<p>2) System provides an interface in which user (Coordinator, Supervisor and Student) logged in to the application.</p> <p>5) System validates email address and password</p> <p>6) System navigates to their individual pages.</p>
<b>Alternative Flows:</b>	<p>3) What happens if user provides unregistered email address or password?</p> <ol style="list-style-type: none"> <li>1. System notifies about invalid email/password.</li> <li>2. User (Coordinator, Supervisor and Student) provides registered email address and password.</li> <li>3. User (Coordinator, Supervisor and Student) clicks on login button</li> </ol>	

### Use Case 3 Title: Register Supervisors and Students

Supervisors and Students are registered by the coordinator by providing email address, password and confirm password is shown in Table 2.6

Table 2.6 Register Supervisor and Students

<b>Use Case ID:</b>	3	
<b>Use Case Name:</b>	Register Supervisors and Students	
<b>Created By:</b>	Haroon	<b>Last Updated By:</b> Hina

<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b>	November 15, 2019								
<b>Actors:</b>	Coordinator										
<b>Description:</b>	Supervisors and Students are registered by the coordinator by providing email address, password and confirm password.										
<b>Trigger:</b>	Supervisor and Student Registration Button										
<b>Preconditions:</b>	Coordinator must have logged in to the application.										
<b>Post conditions:</b>	Supervisors and students are registered into the application.										
<b>Normal Flow:</b>	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1) Coordinator clicks on supervisor and student registration button.</td> <td>2) System provides an interface about registering a supervisor and student respectively.</td> </tr> <tr> <td>3) Coordinator registers the supervisor and student by providing email address, password and confirm password individually.</td> <td>5) System notifies that the supervisor or student have been registered individually.</td> </tr> <tr> <td>4) Coordinator clicks on sign up button individually.</td> <td></td> </tr> </tbody> </table>			Actor	System	1) Coordinator clicks on supervisor and student registration button.	2) System provides an interface about registering a supervisor and student respectively.	3) Coordinator registers the supervisor and student by providing email address, password and confirm password individually.	5) System notifies that the supervisor or student have been registered individually.	4) Coordinator clicks on sign up button individually.	
Actor	System										
1) Coordinator clicks on supervisor and student registration button.	2) System provides an interface about registering a supervisor and student respectively.										
3) Coordinator registers the supervisor and student by providing email address, password and confirm password individually.	5) System notifies that the supervisor or student have been registered individually.										
4) Coordinator clicks on sign up button individually.											
<b>Alternative Flows:</b>	<p>3) What happens if coordinator clicks on signup button without providing the email address, password and confirm password?</p> <ol style="list-style-type: none"> <li>1. System notifies provide the credentials.</li> <li>2. Coordinator provides email address, password and confirm password of individual supervisor and students.</li> </ol>										

	3. Coordinator clicks on sign up button.
--	--

#### Use Case 4 Title: Manage Terms

Coordinator can create, view, edit and delete the term is shown in Table 2.7

Table 2.7 Manage Terms

<b>Use Case ID:</b>	4	
<b>Use Case Name:</b>	Manage Terms	
<b>Created By:</b>	Haroon	<b>Last Updated By:</b> Hina
<b>Date Created:</b>	December 15,2019	<b>Last Revision Date:</b> December 15,2019
<b>Actors:</b>	Coordinator	
<b>Description:</b>	Coordinator can create, view, edit and delete the term	
<b>Trigger:</b>	Create New term Button	
<b>Preconditions:</b>	Coordinator must have click the create new term button.	
<b>Post conditions:</b>	Coordinator have created term and can view, edit, delete the term	
<b>Normal Flow:</b>	Actor	System
	1) Coordinator clicks on create new term button  3) Coordinator provides the term name, date from, date to and working term.  4) Coordinator clicks on	2) System provides a forum for adding credentials like term name, date from, date to and working term.  5) System displays created term with edit and deletes

	save button	option.
<b>Alternative Flows:</b>	3) What happens if Coordinator presses save button without filling the whole form? 1. System notifies that please fill out this field.	

### Use Case 5 Title: Manage Program Learning Objectives

Coordinator is adding, viewing; updating and deleting Program Learning Objectives on basis of terms selected is shown in Table 2.8

Table 2.8 Manage Program Learning Objectives

<b>Use Case ID:</b>	5		
<b>Use Case Name:</b>	Manage Program Learning Objectives		
<b>Created By:</b>	Haroon	<b>Last Updated By:</b>	Hina
<b>Date Created:</b>	December 15,2019	<b>Last Revision Date:</b>	December 15,2019
<b>Actors:</b>	Coordinator		
<b>Description:</b>	Coordinator is adding, viewing; updating and deleting Program Learning Objectives on basis of terms selected		
<b>Trigger:</b>	PLO's Button		
<b>Preconditions:</b>	Coordinator must have created term first.		
<b>Post conditions:</b>	Coordinator has created Program Learning Objectives on basis of terms selected and will manage them.		
<b>Normal Flow:</b>	Actor	System	

	<p>1) Coordinator clicks on PLO button</p> <p>3) Coordinator selects term and add Program Learning Objectives</p> <p>4) Coordinator clicks on add button</p>	<p>2) System provides a forum for selecting specific term and adding Program Learning Objective</p> <p>5) System displays created Program Learning Objective with edit and removes option.</p>
<b>Alternative Flows:</b>	<p>3) What happens if Coordinator press add button without selecting the term?</p> <ol style="list-style-type: none"> <li>1. System shows by reloading the down panel that no program learning objectives is created.</li> </ol>	

### Use Case 6 Title: Manage Course Learning Objectives

Coordinator is adding, viewing; updating and deleting Course Learning Objectives on basis of Program Learning Objectives and terms selected is shown in Table 2.9

Table 2.9 Manage Course Learning Objectives

<b>Use Case ID:</b>	6		
<b>Use Case Name:</b>	Manage Course Learning Objectives		
<b>Created By:</b>	Haroon	<b>Last Updated By:</b>	Hina
<b>Date Created:</b>	December 15,2019	<b>Last Revision Date:</b>	December 15,2019
<b>Actors:</b>	Coordinator		
<b>Description:</b>	Coordinator is adding, viewing; updating and deleting Course Learning Objectives on basis of Program Learning		

	Objectives and terms selected	
<b>Trigger:</b>	CLO's Button	
<b>Preconditions:</b>	Coordinator must have created term and Program Learning Objective first	
<b>Post conditions:</b>	Coordinator has created Course Learning Objectives with program learning objective and term selected.	
<b>Normal Flow:</b>	Actor	System
	1) Coordinator clicks on CLO button 3) Coordinator selects term, Program Learning Objective and add Course Learning Objectives 4) Coordinator clicks on add button	2) System provides a forum for selecting specific term, Program Learning Objective and adding Course Learning Objective 5) System displays created Course Learning Objective with edit and delete option.
<b>Alternative Flows:</b>	3) What happens if Coordinator press add button without selecting the term and Program Learning Objective?  1. System shows by reloading the down panel that no course learning objectives is created.	

### Use Case 7 Title: Manage Exams

Coordinator is adding, viewing; updating and deleting exams on basis of terms selected is shown in Table 2.10

Table 2.10 Manage Exams

<b>Use Case ID:</b>	7
---------------------	---

<b>Use Case Name:</b>	Manage Exams	
<b>Created By:</b>	Haroon	<b>Last Updated By:</b> Hina
<b>Date Created:</b>	December 15,2019	<b>Last Revision Date:</b> December 15,2019
<b>Actors:</b>	Coordinator	
<b>Description:</b>	Coordinator is adding, viewing; updating and deleting exams on basis of terms selected	
<b>Trigger:</b>	Exams Button	
<b>Preconditions:</b>	Coordinator must have created term first.	
<b>Post conditions:</b>	Coordinator has created Exams on basis of term selected	
<b>Normal Flow:</b>	Actor	System
	1) Coordinator clicks on Exams button.  3) Coordinator selects term and adds exam name, weight age, announced date and deadline.  4) Coordinator clicks on add button.	2) System provides a forum for selecting specific term and adding exam name, weight age, announced date and deadline  5) System displays created Exam with edit and delete option.
<b>Alternative Flows:</b>	3) What happens if Coordinator press add button without selecting the term and exam information?  1. System shows by reloading the down panel that no exam is created.	

### Use Case 8 Title: Manage Questions

35

Coordinator is adding, viewing; updating and deleting questions on basis of terms, Course Learning Objectives, and exam selected is shown in Table 2.11

Table 2.11 Manage Questions

<b>Use Case ID:</b>	8									
<b>Use Case Name:</b>	Manage Questions									
<b>Created By:</b>	Haroon	<b>Last Updated By:</b> Hina								
<b>Date Created:</b>	December 15,2019	<b>Last Revision Date:</b> December 15,2019								
<b>Actors:</b>	Coordinator									
<b>Description:</b>	Coordinator is adding, viewing; updating and deleting questions on basis of terms, Course Learning Objectives, and exam selected									
<b>Trigger:</b>	Questions Button									
<b>Preconditions:</b>	Coordinator must have created term, Course Learning Objective and Exam type first.									
<b>Post conditions:</b>	Coordinator has created Questions with term, Course Learning objective and Exam type selected.									
<b>Normal Flow:</b>	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1) Coordinator clicks on Questions button.</td> <td>2) System provides a forum for selecting specific term, Course Learning Objective and exam type and adding question</td> </tr> <tr> <td>3) Coordinator selects term, Course Learning Objective and exam type and add question.</td> <td>5) System displays created Question with edit and</td> </tr> <tr> <td>4) Coordinator clicks on</td> <td></td> </tr> </tbody> </table>	Actor	System	1) Coordinator clicks on Questions button.	2) System provides a forum for selecting specific term, Course Learning Objective and exam type and adding question	3) Coordinator selects term, Course Learning Objective and exam type and add question.	5) System displays created Question with edit and	4) Coordinator clicks on		
Actor	System									
1) Coordinator clicks on Questions button.	2) System provides a forum for selecting specific term, Course Learning Objective and exam type and adding question									
3) Coordinator selects term, Course Learning Objective and exam type and add question.	5) System displays created Question with edit and									
4) Coordinator clicks on										

	add button.	deletes option.
<b>Alternative Flows:</b>	3) What happens if Coordinator press add button without selecting the term, Course Learning Objective, exam type and adding question?  1. System shows by reloading the down panel that no question is created.	

### Use Case 9 Title: Manage Supervisor's Panel

Coordinator is adding, viewing; updating and deleting supervisor's panel with term selected is shown in Table 2.12

Table 2.12 Manage Supervisor's Panel

<b>Use Case ID:</b>	9		
<b>Use Case Name:</b>	Manage Supervisor's Panel		
<b>Created By:</b>	Haroon	<b>Last Updated By:</b>	Hina
<b>Date Created:</b>	December 15,2019	<b>Last Revision Date:</b>	December 15,2019
<b>Actors:</b>	Coordinator		
<b>Description:</b>	Coordinator is adding, viewing; updating and deleting supervisor's panel with term selected		
<b>Trigger:</b>	Panels Button		
<b>Preconditions:</b>	Coordinator must have created term first.		
<b>Post conditions:</b>	Coordinator managing individual panels with term selected		

<b>Normal Flow:</b>	<b>Actor</b>	<b>System</b>
	1) Coordinator clicks on panel button 3) Coordinator selects the term and add the panel 4) Coordinator clicks on add button	2) System provides a form to select the term and add the panel 5) System displays the added panel with edit and delete button.
<b>Alternative Flows:</b>	3) What happens if Coordinator press add button without adding the panel name?  1. System shows by reloading the down panel that no panel is created.	

### **Use Case 10 Title: Manage Supervisors**

Coordinator is adding, viewing; updating and deleting supervisors in panels on basis of term selected is shown in Table 2.13

Table 2.13 Manage Supervisors

<b>Use Case ID:</b>	10		
<b>Use Case Name:</b>	Manage Supervisors		
<b>Created By:</b>	Haroon	<b>Last Updated By:</b>	Hina
<b>Date Created:</b>	December 15,2019	<b>Last Revision Date:</b>	December 15,2019
<b>Actors:</b>	Coordinator		
<b>Description:</b>	Coordinator is adding, viewing; updating and deleting supervisors in panels on basis of term selected		

<b>Trigger:</b>	Supervisor Button	
<b>Preconditions:</b>	Coordinator must have created term and panel	
<b>Post conditions:</b>	Coordinator have created Supervisors in panel with term selected	
<b>Normal Flow:</b>	Actor	System
	1) Coordinator clicks on Supervisor button. 3) Coordinator selects term, panel and adding supervisor's name, email, slot and designation. 4) Coordinator clicks on add button.	2) System provides a forum for selecting specific term, panel and adding supervisor's name, email, slot and designation. 5) System displays created supervisor with edit and deletes option.
<b>Alternative Flows:</b>	3) What happens if Coordinator press add button without selecting the term, panel and adding supervisor information? 1. System shows by reloading the down panel that no supervisor is created.	

### Use Case 11 Title: Manage Announcements

Coordinator is adding, viewing; updating and deleting announcements on basis of term selected is shown in Table 2.14

Table 2.14 Manage Announcements

<b>Use Case ID:</b>	11
<b>Use Case Name:</b>	Manage Announcements

<b>Created By:</b>	Hina	<b>Last Updated By:</b>	Haroon
<b>Date Created:</b>	December 15,2019	<b>Last Revision Date:</b>	December 15,2019
<b>Actors:</b>	Coordinator		
<b>Description:</b>	Coordinator is adding, viewing; updating and deleting announcements on basis of term selected		
<b>Trigger:</b>	Announcement Button		
<b>Preconditions:</b>	Coordinator must have created term first.		
<b>Post conditions:</b>	Coordinator has created announcements with term selected.		
<b>Normal Flow:</b>	Actor 1) Coordinator clicks on announcement button 3) Coordinator selects term and add announcement 4) Coordinator clicks on add button		System 2) System provides a forum for selecting specific term and adding announcements 5) System displays created announcements with edit and removes option.
<b>Alternative Flows:</b>	3) What happens if Coordinator press add button without selecting the term and adding announcements? 1. System shows by reloading the down panel that no announcement is created.		

### Use Case 12 Title: View Announcements

Supervisors and students can view announcements of the working term is shown in Table 2.15

Table 2.15 View Announcements

<b>Use Case ID:</b>	12	
<b>Use Case Name:</b>	View Announcements	
<b>Created By:</b>	Hina	<b>Last Updated By:</b> Haroon
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b> November 15, 2019
<b>Actors:</b>	Supervisors and Student	
<b>Description:</b>	Supervisors and students can view announcements of the working term	
<b>Trigger:</b>	Dashboard Button of Supervisors and Student	
<b>Preconditions:</b>	Student must have logged in to the application.	
<b>Post conditions:</b>	Student viewing the announcements by coordinator and gets up to date with the events.	
<b>Normal Flow:</b>	Actor	System
	1) Supervisor and Student select the dashboard page.  3) Supervisor and Student viewing all the announcements of the term	2) System gives all the announcements made by the coordinator of specific term in the page.
<b>Alternative Flows:</b>	1) What happens if supervisor and students clicks on another page instead of dashboard page?  1. System navigates the supervisor and the student to that page.	

### Use Case 13 Title: Create Group

41

Supervisor can create a group of students who are registered for final year project in a term is shown in Table 2.16

Table 2.16 Create Group

<b>Use Case ID:</b>	13	
<b>Use Case Name:</b>	Create Group	
<b>Created By:</b>	Hina	<b>Last Updated By:</b> Haroon
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b> November 15, 2019
<b>Actors:</b>	Supervisor	
<b>Description:</b>	Supervisor can create a group of students who are registered for final year project in a term.	
<b>Trigger:</b>	Make Group Button	
<b>Preconditions:</b>	Student must be registered for the final year project	
<b>Post conditions:</b>	Supervisor has created a group with students in a term and can view the group	
<b>Normal Flow:</b>	Actor	System
	1) Supervisor clicks on the Make Group Button  3) Supervisor can select the term, select the students and add a group number  4) Supervisor clicks on the submit button	2) System displays the form in which supervisor can select the term, students and add a group number and create a group.  5) System reloads the page showing the group has been created.
<b>Alternative Flows:</b>	3) What happens if supervisor select the term, select the students and does not write the group number and open up another page?	

	1. System navigates supervisor to that page.
--	--

### Use Case 14 Title: Assign Project Title

Supervisor assigns the project title to the group created by him/her in a term is shown in Table 2.17

Table 2.17 Assign Project Title

<b>Use Case ID:</b>	14	
<b>Use Case Name:</b>	Assign Project Title	
<b>Created By:</b>	Hina	<b>Last Updated By:</b> Haroon
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b> November 15, 2019
<b>Actors:</b>	Supervisor	
<b>Description:</b>	Supervisor assigns the project title to the group created by him/her in a term	
<b>Trigger:</b>	Make Group Button	
<b>Preconditions:</b>	Supervisor must have created a group in a term	
<b>Post conditions:</b>	Supervisor can view the assigned project title to a group	
<b>Normal Flow:</b>	Actor	System
	1) Supervisor selects the Make group button. 3) Supervisor selects the term, group and adds project title to it. 4) Supervisor clicks the submit button	2) System provides a interface in which supervisor selects the term, group and add project title to it. 5) System reloads the page showing the project title has

		been assigned.
<b>Alternative Flows:</b>	3) What happens if supervisor selects the term, group but does not write the title and opens up another page? 1. System navigates the supervisor to that page.	

### Use Case 15 Title: Send Email by Supervisor

Supervisor can send an email to the coordinator about creation of a project is shown in Table 2.18

Table 2.18 Send Email by Supervisor

<b>Use Case ID:</b>	15	
<b>Use Case Name:</b>	Send Email by Supervisor	
<b>Created By:</b>	Hina	<b>Last Updated By:</b> Haroon
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b> November 15, 2019
<b>Actors:</b>	Supervisor	
<b>Description:</b>	Supervisor can send an email to the coordinator about creation of a project.	
<b>Trigger:</b>	Send for Approval Button	
<b>Preconditions:</b>	Supervisor must have created a group and assign a project title to it.	
<b>Post conditions:</b>	Email sent to coordinator about the project creation by supervisor	
<b>Normal Flow:</b>	Actor	System

	1) Supervisor clicks on send for approval button.	2) System sends email to coordinator about the project creation.
<b>Alternative Flows:</b>	1) What happens if supervisor does not click on the send for approval button?  1. System does not send email to coordinator about the project creation.	

### Use Case 16 Title: View Project and Panel details

Supervisor can view the projects details (project title, panel Number, Term name, group name, supervisor) and panel details (panel member's name, designation and email id) of the project of selected term is shown in Table 2.19

Table 2.19 View Project and Panel details

<b>Use Case ID:</b>	16		
<b>Use Case Name:</b>	View Project and Panel details		
<b>Created By:</b>	Hina	<b>Last Updated By:</b>	Haroon
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b>	November 15, 2019
<b>Actors:</b>	Supervisor		
<b>Description:</b>	Supervisor can view the projects details (project title, panel Number, Term name, group name, supervisor) and panel details (panel member's name, designation and email id) of the project of selected term		
<b>Trigger:</b>	Project Details Button		
<b>Preconditions:</b>	Supervisor must have created a group of students and assigned a project title to it.		

<b>Post conditions:</b>	Supervisor can view project and panel details of a project of selected term.	
<b>Normal Flow:</b>	Actor	System
	1) Supervisor clicks on project details button. 3) Supervisor selects the term 5) Supervisor clicks on the specific project. 7) Supervisor views the project and panel details of the project.	2) System provides the interface in which supervisor selects the term. 4) System provides the projects. 6) System shows the project and panel details of the project.
<b>Alternative Flows:</b>	5) What happens if supervisor clicks on the project action button without selecting the term? 1. System notifies that please select term first.	

### Use Case 17 Title: View Questions

Supervisors and students can view the questions to each exam with course learning objectives with term working is shown in Table 2.20

Table 2.20 View Questions

<b>Use Case ID:</b>	17		
<b>Use Case Name:</b>	View Questions		
<b>Created By:</b>	Hina	<b>Last Updated By:</b>	Haroon
<b>Date Created:</b>	November 15, 2019	<b>Last Revision</b>	November 15, 2019

		<b>Date:</b>	
<b>Actors:</b>	Supervisor and student		
<b>Description:</b>	Supervisors and students can view the questions to each exam with course learning objectives with term working		
<b>Trigger:</b>	Project Details Button, Proposal Defense Button, Part 1 Mid Term Button, Part 1 Final Term Button, Part 2 Mid Term Button, Part 2 Final Term Button		
<b>Preconditions:</b>	Coordinator must have created the questions with Course learning objectives attach with it in the term		
<b>Post conditions:</b>	Supervisor and student view questions and course learning objectives of a term.		
<b>Normal Flow:</b>	Actor	System	
	1) Supervisor and student clicks on Project Details Button, Proposal Defense Button, Part 1 Mid Term Button, Part 1 Final Term Button, Part 2 Mid Term Button, Part 2 Final Term Button respectively.	2) System provides an interface where supervisor and student can view the questions and course learning objectives individually.	
	3) Supervisor and student can view the questions and the course learning objectives to it.		
<b>Alternative Flows:</b>	1) What happens if supervisor and student clicks on another page while viewing the questions and the attach course learning objectives?  1. System navigates the supervisor and student to that		

	page respectively.
--	--------------------

### Use Case 18 Title: Register Student

Student provides valid name, registration number, date of birth, phone number, email address, credit hours enrolled, credit hour completed, and cumulative GPA with term selected for registration of final year project is shown in Table 2.21

Table 2.21 Register Student

<b>Use Case ID:</b>	18	
<b>Use Case Name:</b>	Register Student	
<b>Created By:</b>	Hina	<b>Last Updated By:</b> Haroon
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b> November 15, 2019
<b>Actors:</b>	Student	
<b>Description:</b>	Student provides valid name, registration number, date of birth, phone number, email address, credit hours enrolled, credit hour completed, and cumulative GPA with term selected for registration of final year project.	
<b>Trigger:</b>	Registration Button	
<b>Preconditions:</b>	Student must be logged in to the system and semester term must be created by the coordinator	
<b>Post conditions:</b>	Student has registered him/her for final year project.	
<b>Normal Flow:</b>	Actor	System
	1) Student clicks on registration button.	2) System provides an interface in which student can select term and add

	<p>3) Student selects the term and add valid name, registration number, date of birth, phone number, email address, credit hours enrolled, credit hour completed, and cumulative GPA</p> <p>4) Student clicks on add button</p>	<p>valid name, registration number, date of birth, phone number, email address, credit hours enrolled, and credit hour completed, and cumulative GPA.</p> <p>5) System notifies that you are registered successfully and reload the page.</p>
<b>Alternative Flows:</b>	<p>3) What happens if student selects the term and does not provide the valid name, registration number, date of birth, phone number, email address, credit hours enrolled, credit hour completed, and cumulative GPA and presses add button?</p> <p>1. System notifies that fill out the form first.</p>	

### Use Case 19 Title: View Supervisor's List

Students can view the supervisor's list involving their name, slots available and email id's of working term is shown in Table 2.22

Table 2.22 View Supervisor's List

<b>Use Case ID:</b>	19		
<b>Use Case Name:</b>	View Supervisor's List		
<b>Created By:</b>	Hina	<b>Last Updated By:</b>	Haroon
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b>	November 15, 2019

<b>Actors:</b>	Student	
<b>Description:</b>	Students can view the supervisor's list involving their name, slots available and email id's of working term	
<b>Trigger:</b>	Supervisor List Button	
<b>Preconditions:</b>	Coordinator must have added the supervisor's name, slots and email id's in the term.	
<b>Post conditions:</b>	Student view the supervisor's list involving their name, slots available and email id's.	
<b>Normal Flow:</b>	Actor	System
	1) Student clicks on supervisor list button.  3) Student views the supervisor's name, slots available and the email ids.	2) System provides an interface in which the student can view the supervisor name, slots available and the email ids.
<b>Alternative Flows:</b>	3) What happens if student clicks on another button while viewing the supervisor's list?  1. System navigates him/her to that page.	

### Use Case 20 Title: Send Email by Student

Student can send presentation and document to supervisor and project panels by sending email is shown in Table 2.23

Table 2.23 Send Email by Student

<b>Use Case ID:</b>	20
<b>Use Case</b>	Send Email by Student

<b>Name:</b>			
<b>Created By:</b>	Hina	<b>Last Updated By:</b>	Haroon
<b>Date Created:</b>	November 15, 2019	<b>Last Revision Date:</b>	November 15, 2019
<b>Actors:</b>	Student		
<b>Description:</b>	Student can send presentation and document to supervisor and project panels by sending email		
<b>Trigger:</b>	Proposal Defense Button, Part 1 Mid Term Button, Part 1 Final Term Button, Part 2 Mid Term Button, Part 2 Final Term Button		
<b>Preconditions:</b>	Student's group must have assigned a project title by the supervisor.		
<b>Post conditions:</b>	Student sending presentation and document to the supervisor and project panel through email		
<b>Normal Flow:</b>	Actor	System	
	1) Student clicks on Proposal Defense Button, Part 1 Mid Term Button, Part 1 Final Term Button, Part 2 Mid Term Button, Part 2 Final Term Button individually.  3) Student enters the supervisor email, his/her own email, his/her email password and choose the presentation and document from the computer	2) System provides an interface in which student enters the supervisor email, student's own email, student's email password and choose the presentation and document from the computer separately.  5) System sends the email to the selected supervisor's email and reloads the page.	

	<p>separately</p> <p>4) Student clicks on send email button</p>	
<b>Alternative Flows:</b>	<p>3) What happens if student enters the supervisor email, his/her own email, his/her email password, choose the presentation, and document from the computer separately but does not click on send email button, clicks on another button?</p> <p>1. System navigates the student to that page.</p>	

## 2.5. System Sequence Diagrams

Coordinator can sign up into the application by providing email address; password and confirm password is shown in Figure 2.2

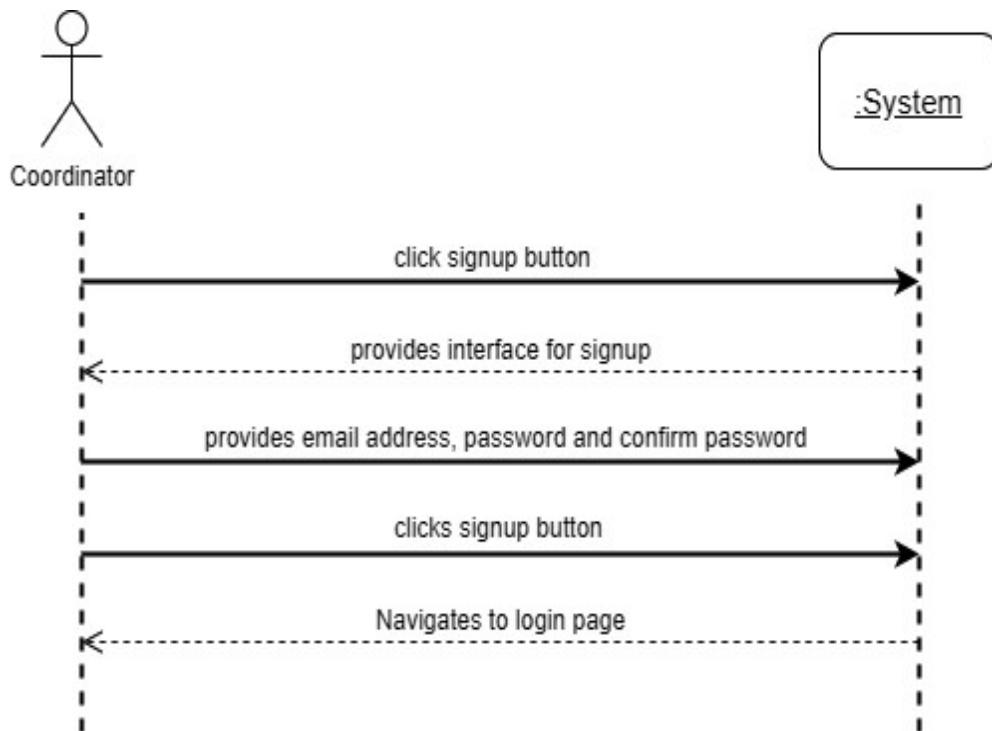


Figure 2.2 Coordinator Sign Up

Coordinator can login to the application by providing email address and password as shown in Figure 2.3

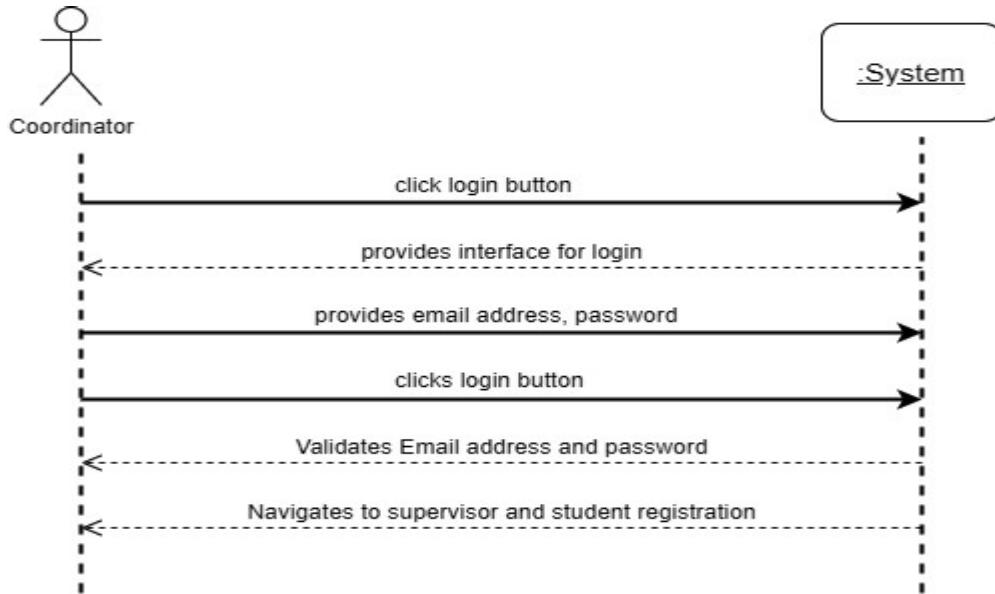


Figure 2.3 Coordinator Login

Supervisor can login to the application by providing email address and password as shown in Figure 2.4

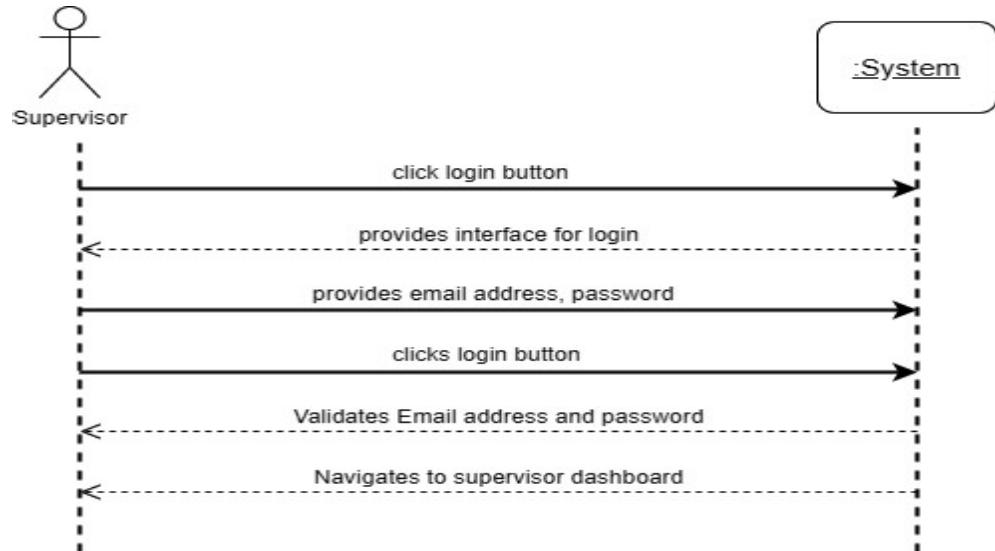


Figure 2.4 Supervisor Login

Student can login to the application by providing email address and password as shown in Figure 2.5

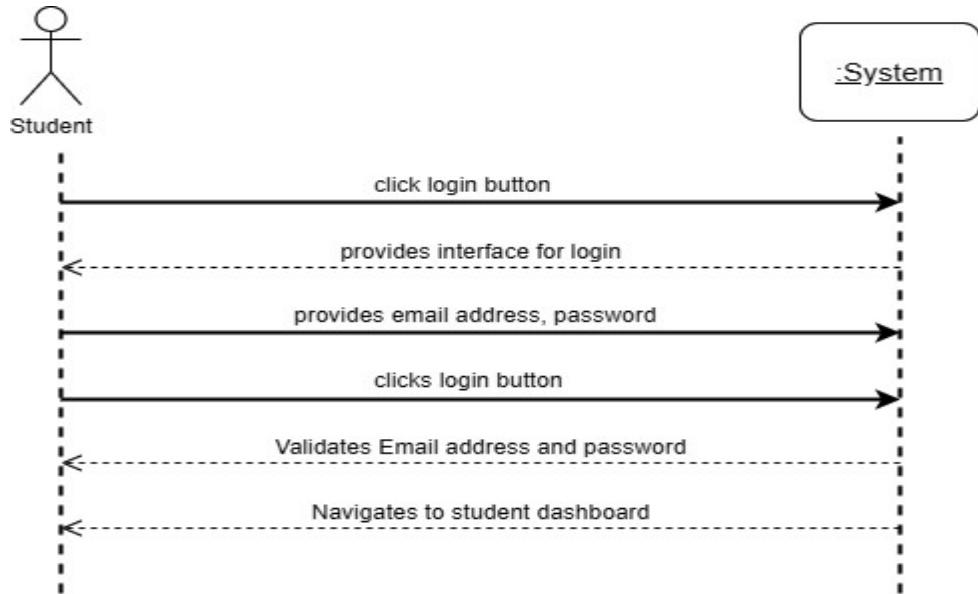


Figure 2.5 Student Login

Supervisors and Students are registered by the coordinator by providing email address, password and confirm password as shown in Figure 2.6

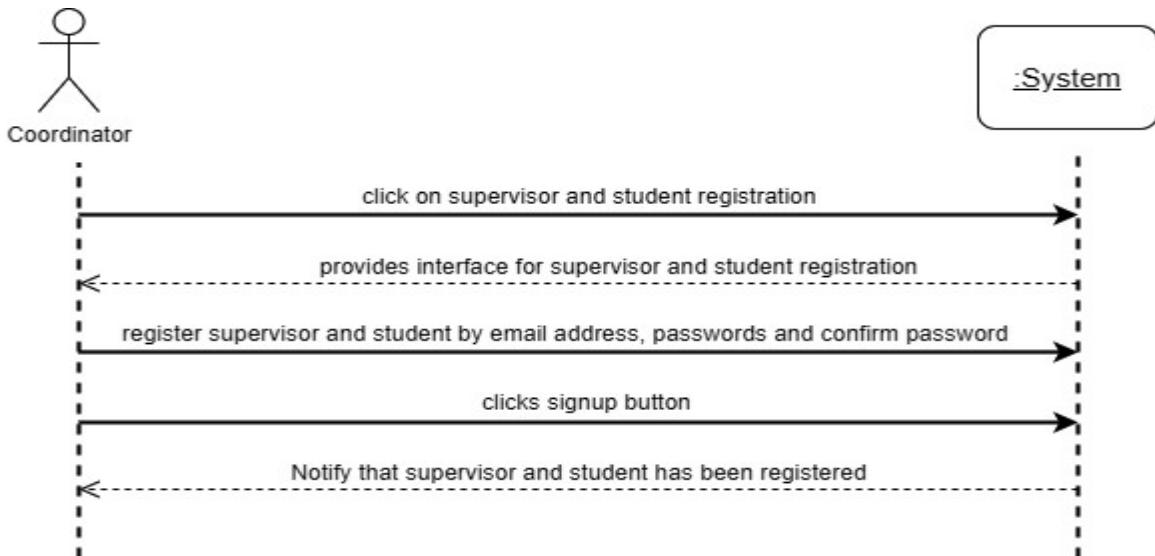


Figure 2.6 Register Supervisors and Students

Coordinator can create, view, edit, and delete the term as shown in Figure 2.7

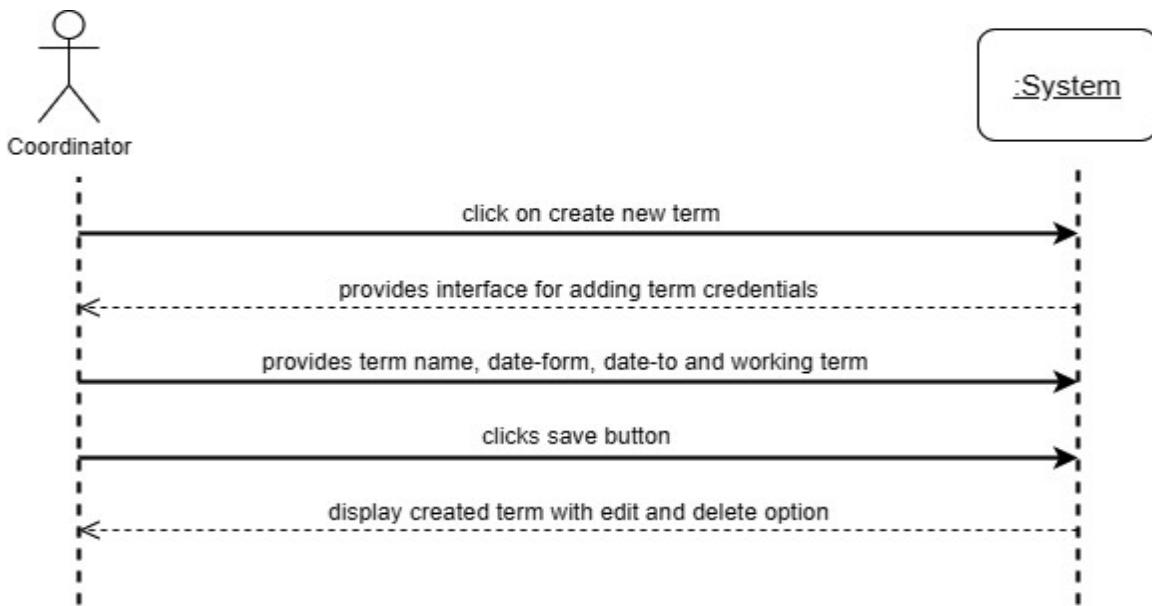


Figure 2.7 Manage Term

Coordinator is adding, viewing, updating and deleting Program Learning Objectives on basis of terms selected as shown in Figure 2.8

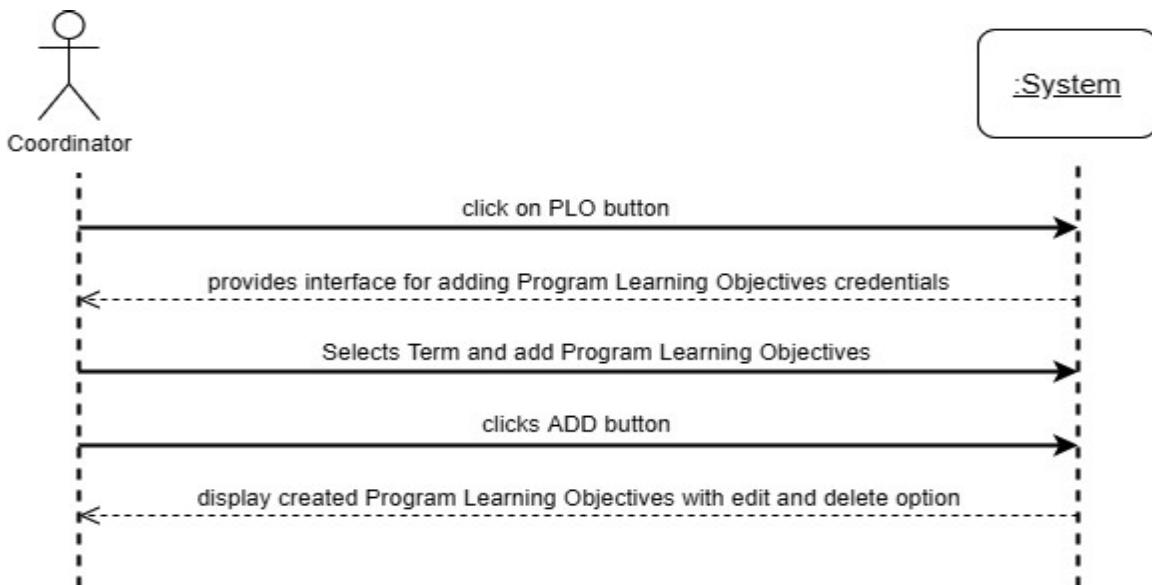


Figure 2.8 Manage Program Learning Objectives

Coordinator is adding, viewing; updating and deleting Course Learning Objectives on basis of Program Learning Objectives and terms selected as shown in Figure 2.9

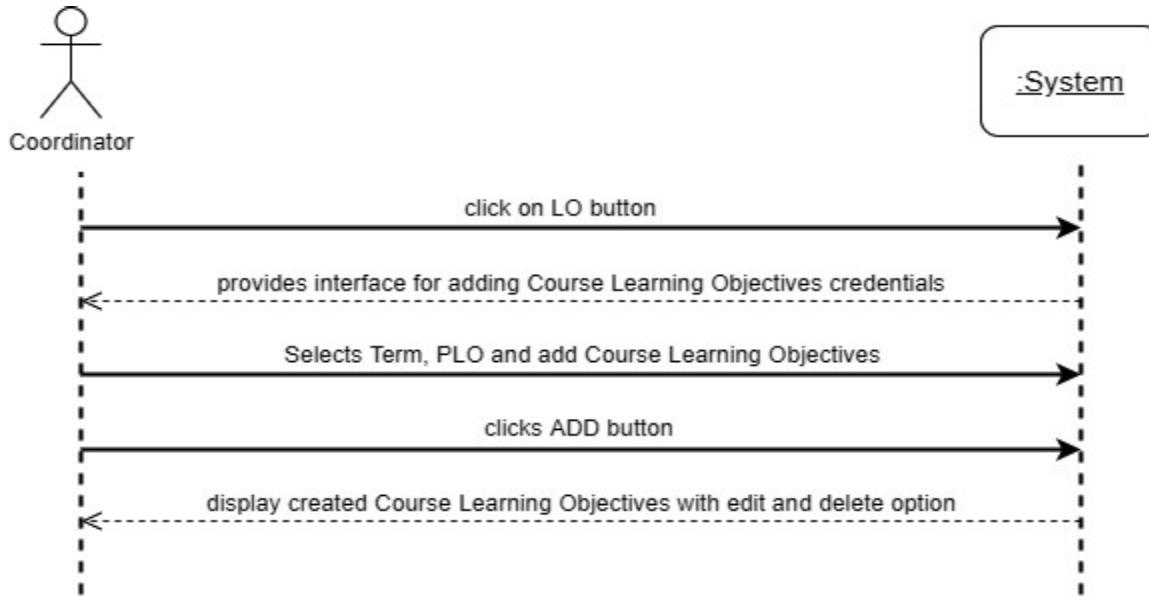


Figure 2.9 Manage Course Learning Objectives

Coordinator is adding, viewing; updating and deleting exams on basis of terms selected as shown in Figure 2.10

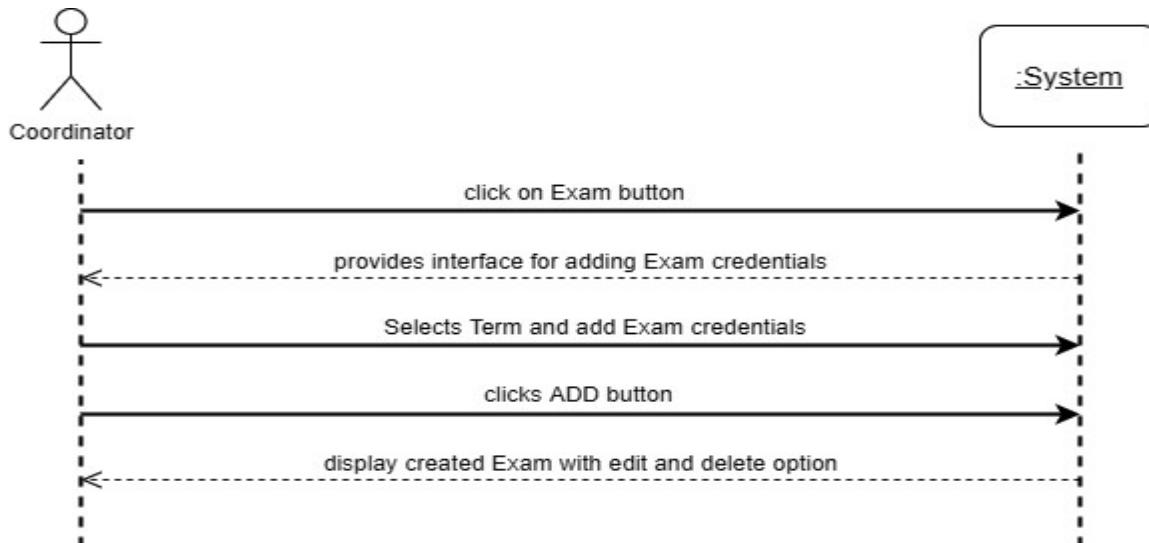


Figure 2.10 Manage Exams

Coordinator is adding, viewing; updating and deleting questions on basis of terms, Course Learning Objectives, and exam selected as shown in Figure 2.11

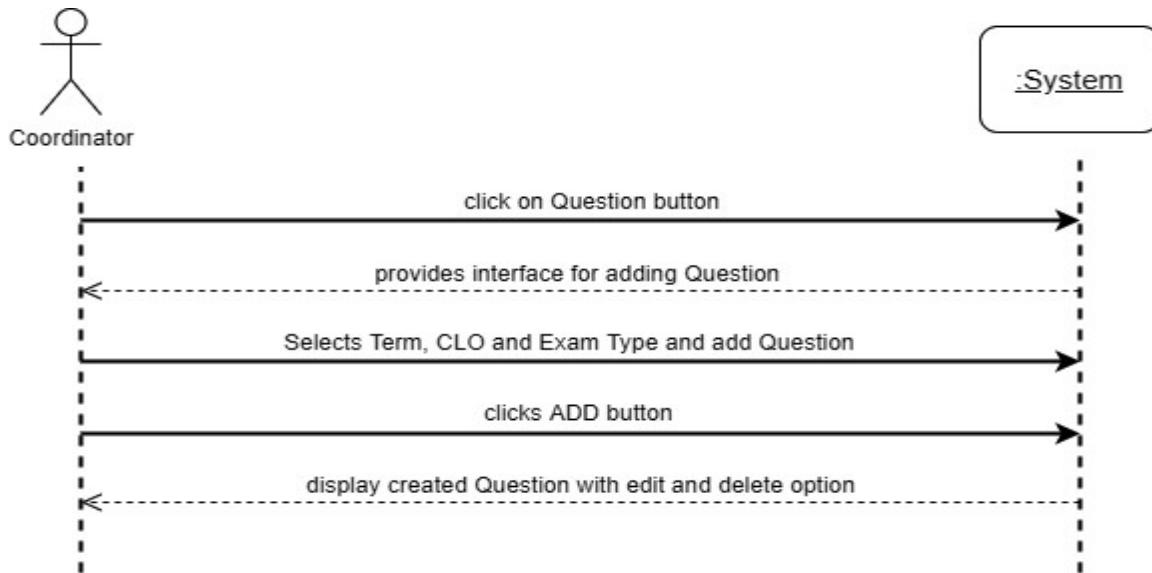


Figure 2.11 Manage Questions

Coordinator is adding, viewing; updating and deleting supervisor's panel with term selected as shown in Figure 2.12

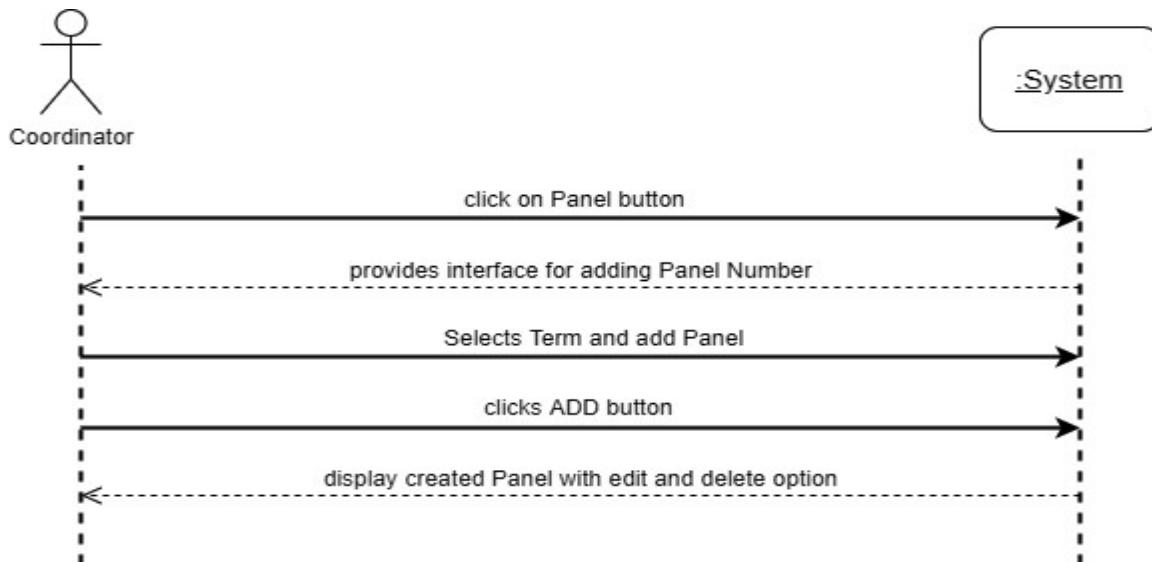


Figure 2.12 Manage Supervisor's Panel

Coordinator is adding, viewing; updating and deleting supervisors in panels on basis of term selected as shown in Figure 2.13

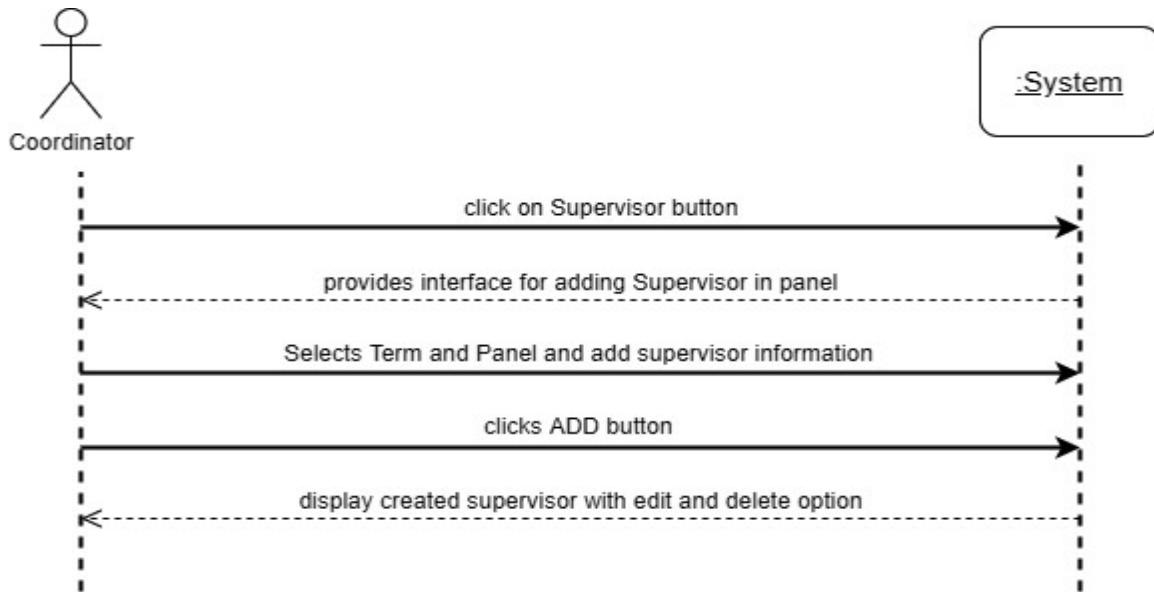


Figure 2.13 Manage Supervisors

Coordinator is adding, viewing; updating and deleting announcements on basis of term selected as shown in Figure 2.14

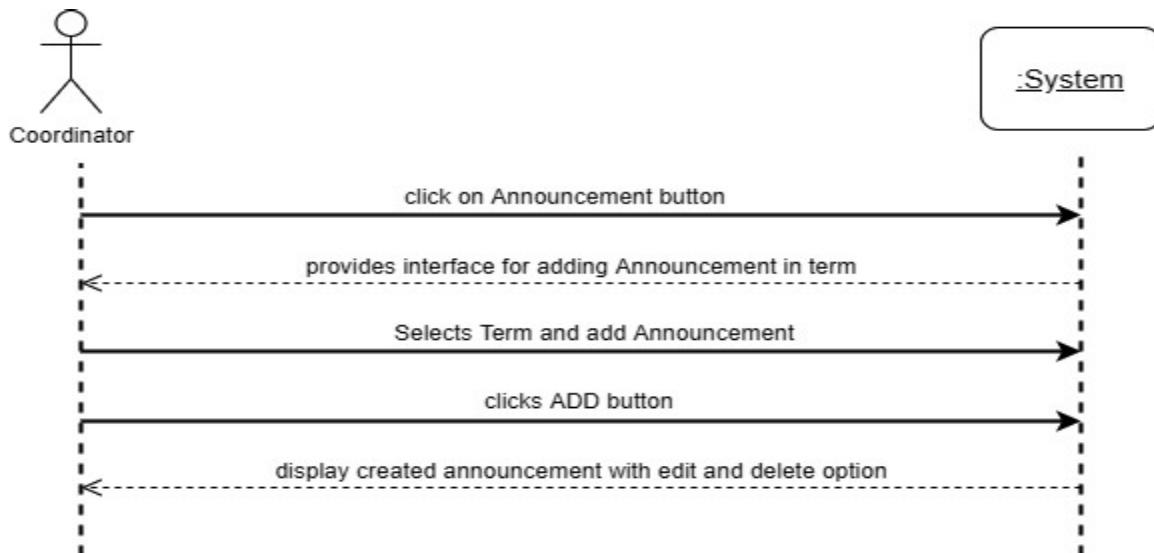


Figure 2.14 Manage Announcements

Supervisors can view announcements of the working term as shown in Figure 2.15

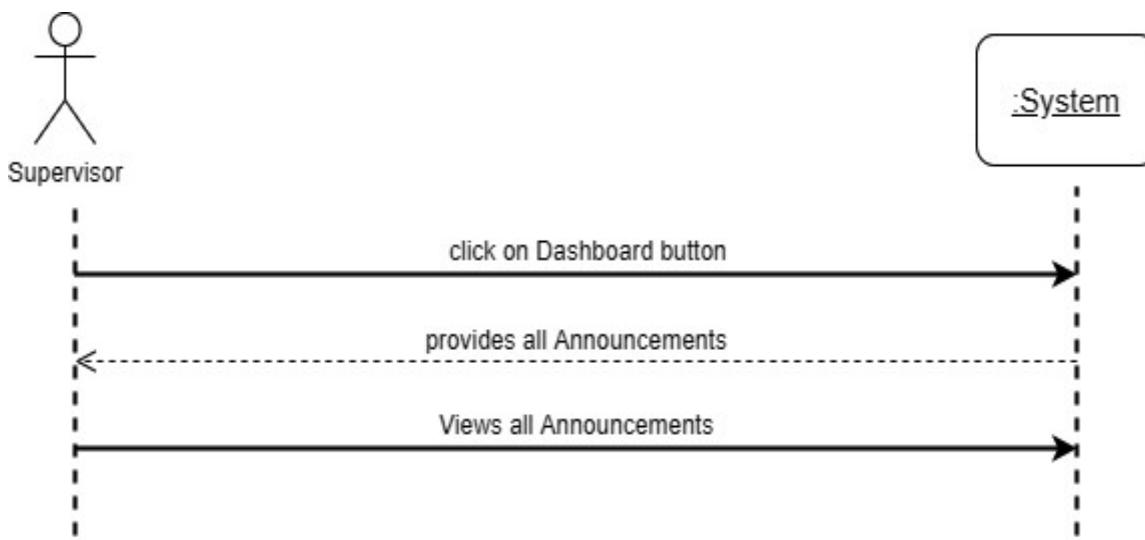


Figure 2.15 View Announcements by Supervisor

Students can view announcements of the working term as shown in Figure 2.16

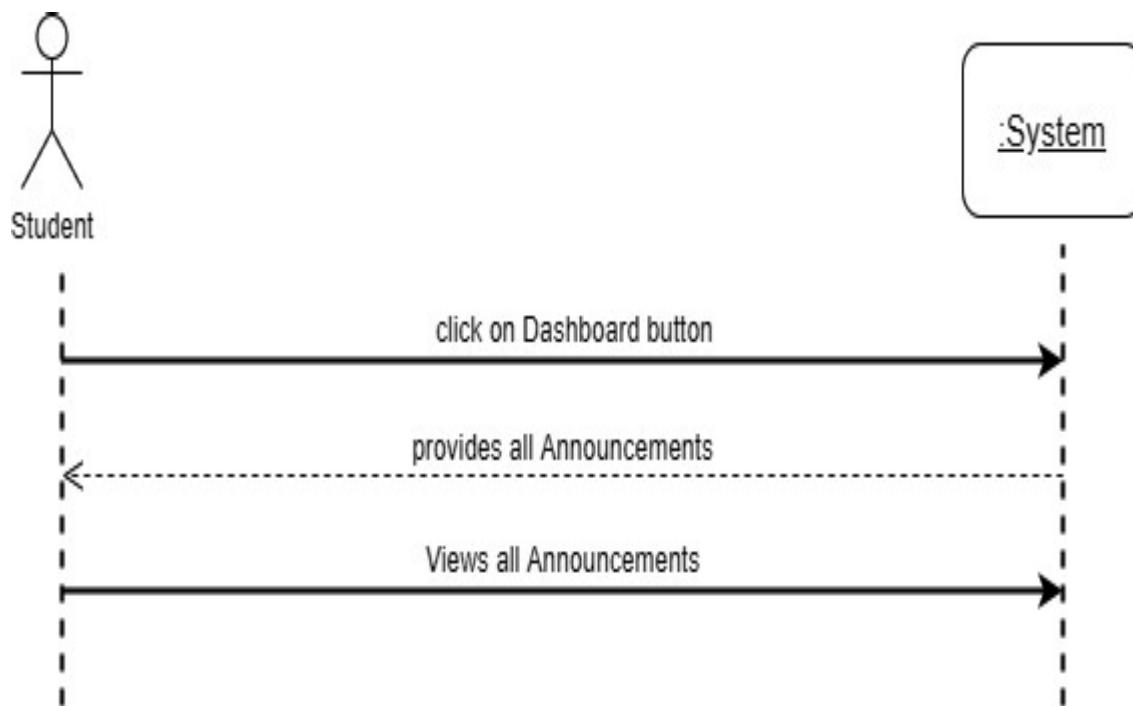


Figure 2.16 View Announcements by Student

Supervisor can create a group of students who are registered for final year project in a term as shown in Figure 2.17

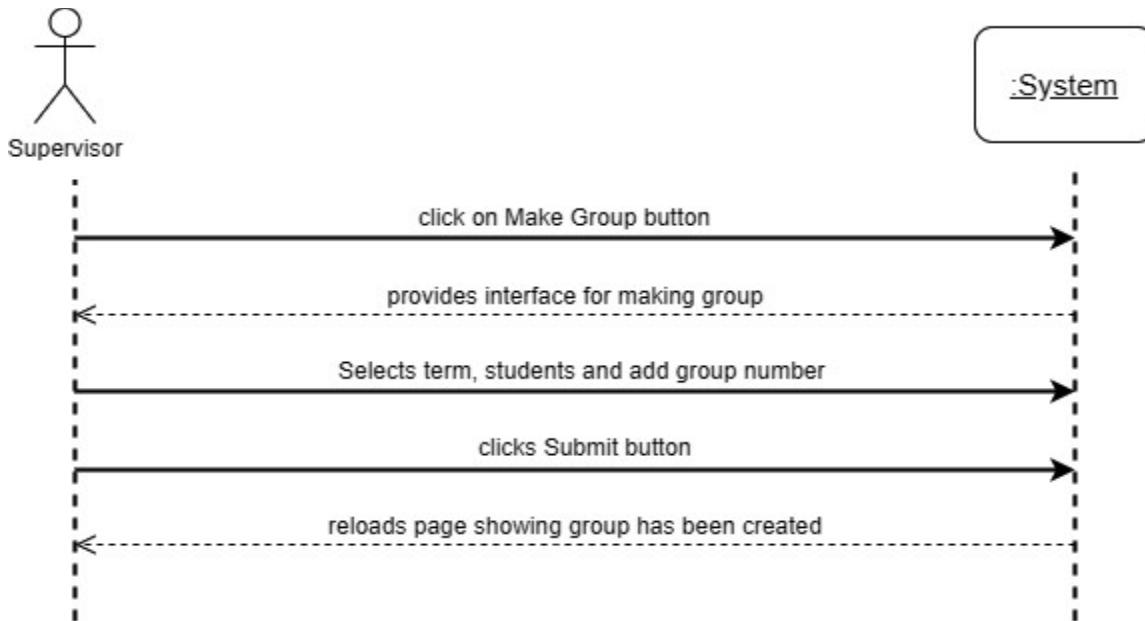


Figure 2.17 Create Group

Supervisor assigns the project title to the group created by him/her in a term as shown in Figure 2.18

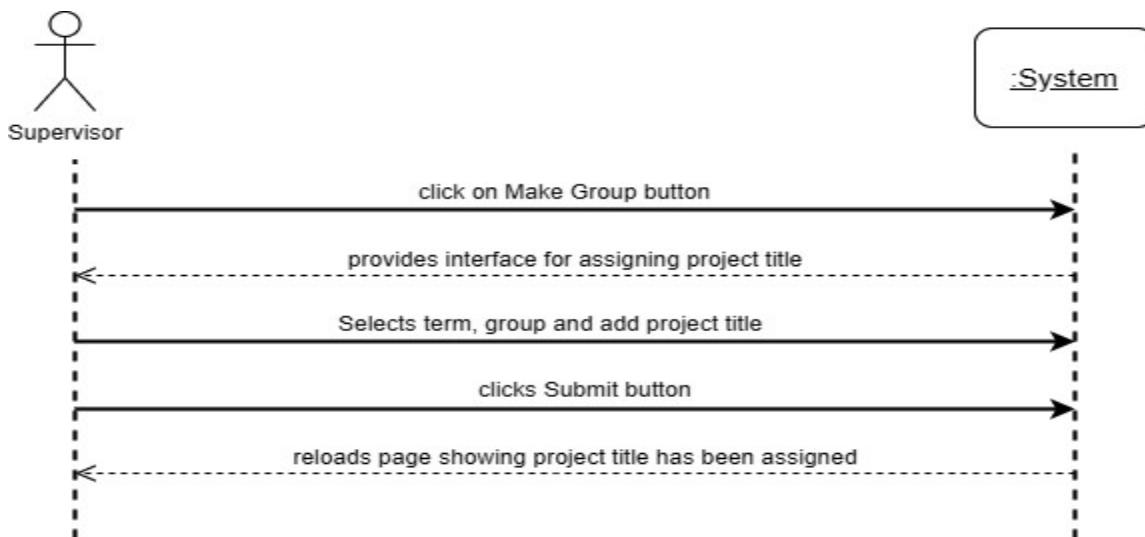


Figure 2.18 Assign Project Title

Supervisor can send an email to the coordinator about creation of a project as shown in Figure 2.19

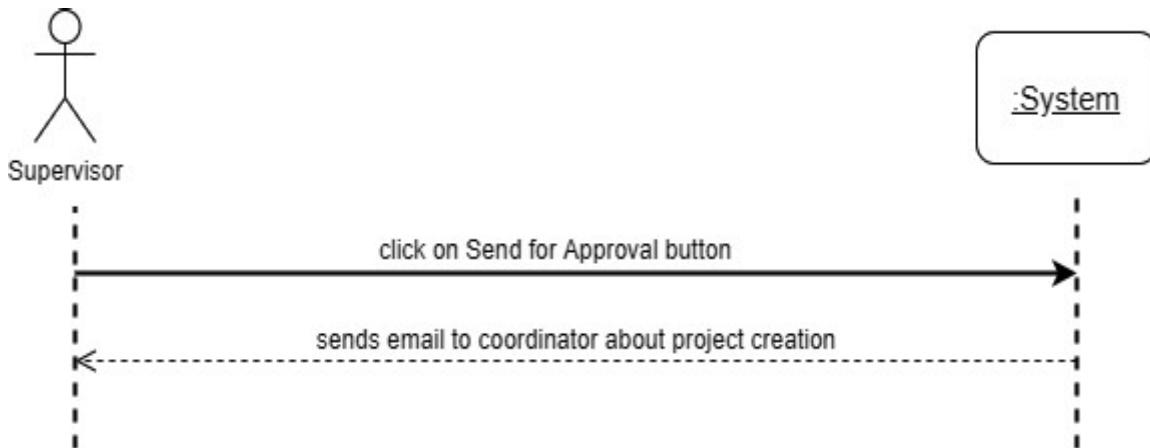


Figure 2.19 Send Email by Supervisor

Supervisor can view the projects details (project title, panel Number, Term name, group name, and supervisor) and panel details (panel member's name, designation and email id) of the project of selected term as shown in Figure 2.20

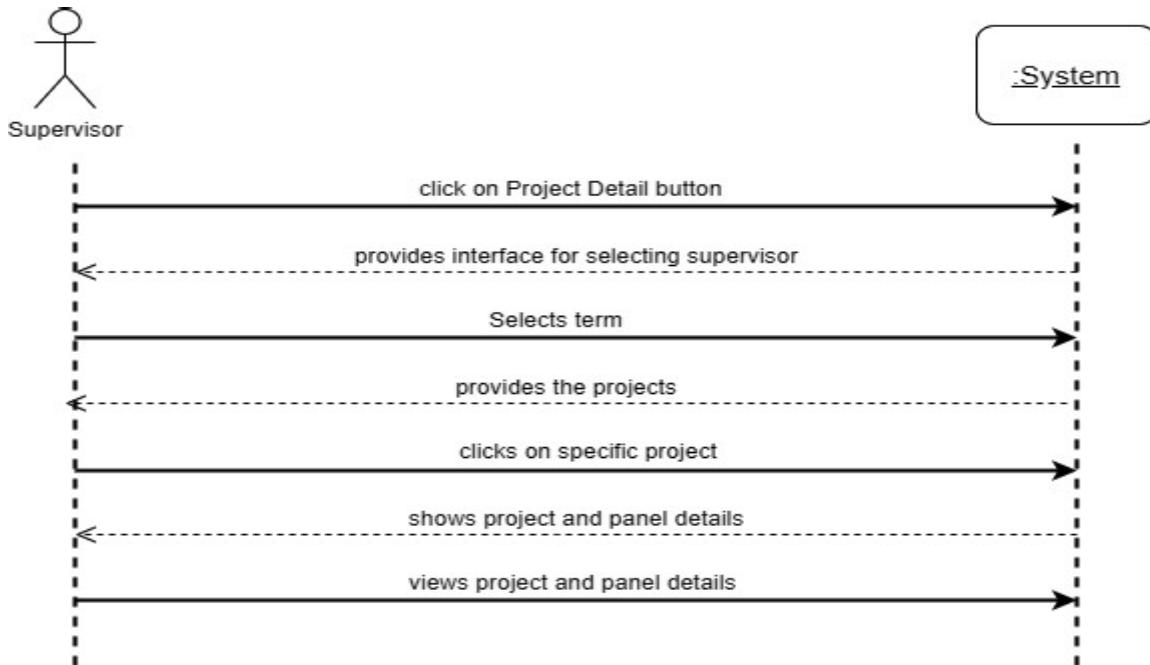


Figure 2.20 View Project and Panel Details

Supervisors can view the questions to each exam with course learning objectives with term working as shown in Figure 2.21

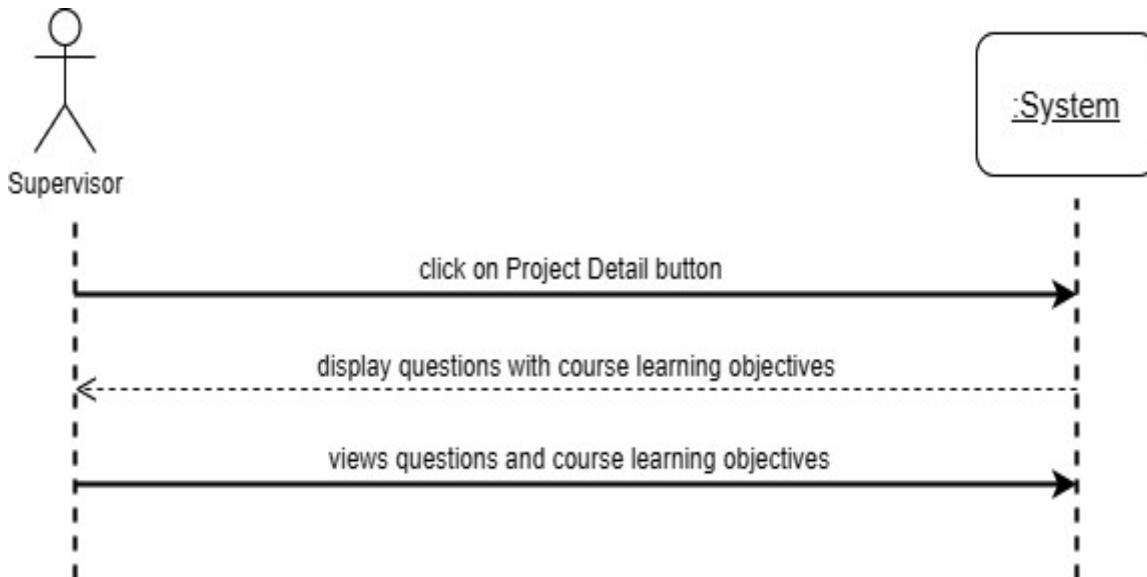


Figure 2.21 View Questions by Supervisor

Students can view the questions to each exam with course learning objectives with term working as shown in Figure 2.22

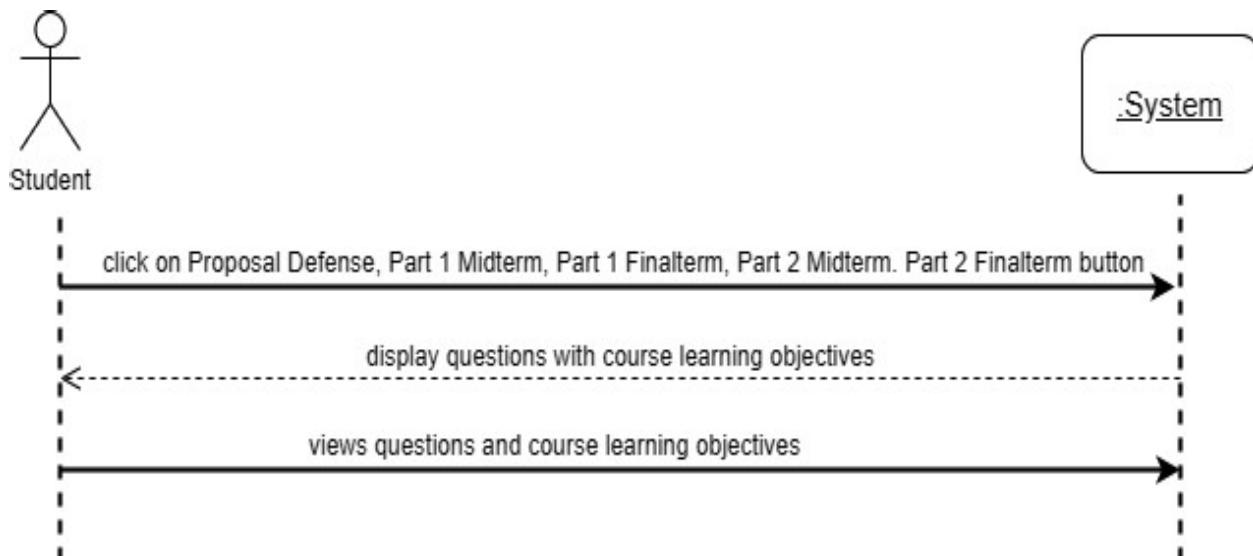


Figure 2.22 View Questions by Student

Student provides valid name, registration number, date of birth, phone number, email address, credit hours enrolled, credit hour completed, and cumulative GPA with term selected for registration of final year project as shown in Figure 2.23

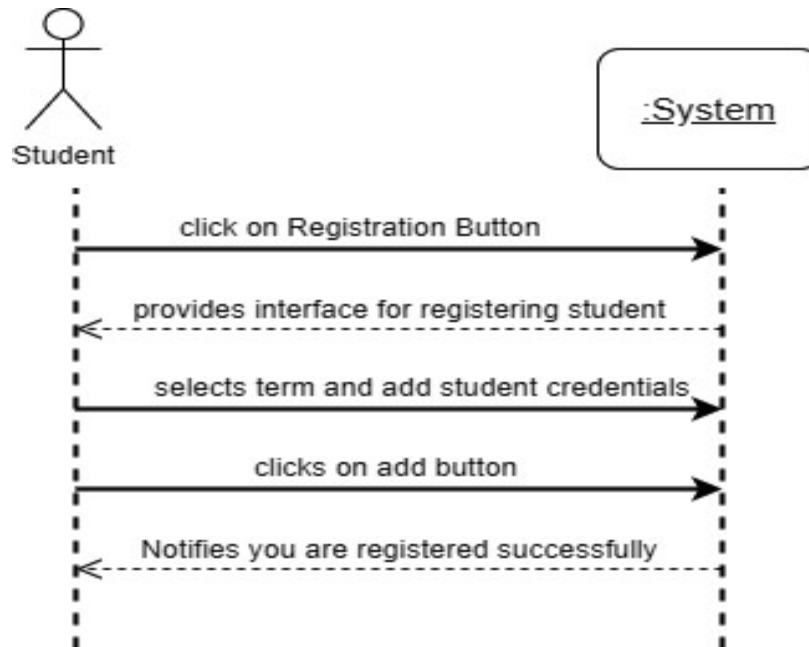


Figure 2.23 Register Student

Students can view the supervisor's list involving their name, slots available and email id's of working term as shown in Figure 2.24

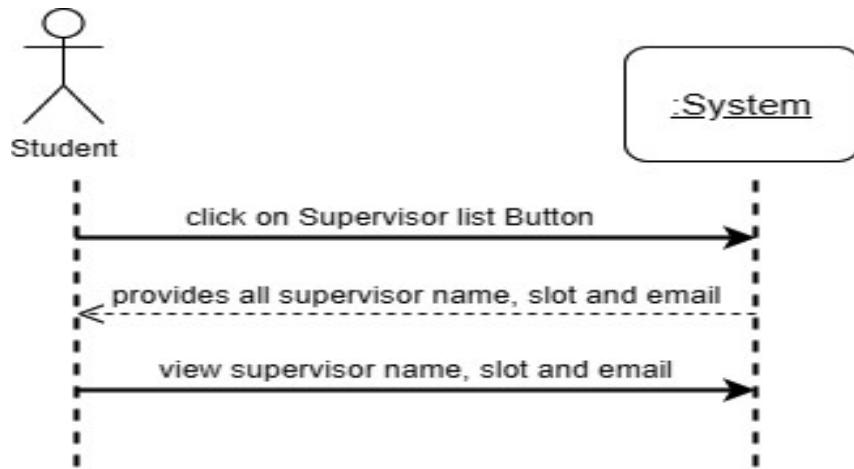


Figure 2.24 View Supervisor's List

Student can send presentation and document to supervisor and project panels by sending email as shown in Figure 2.25

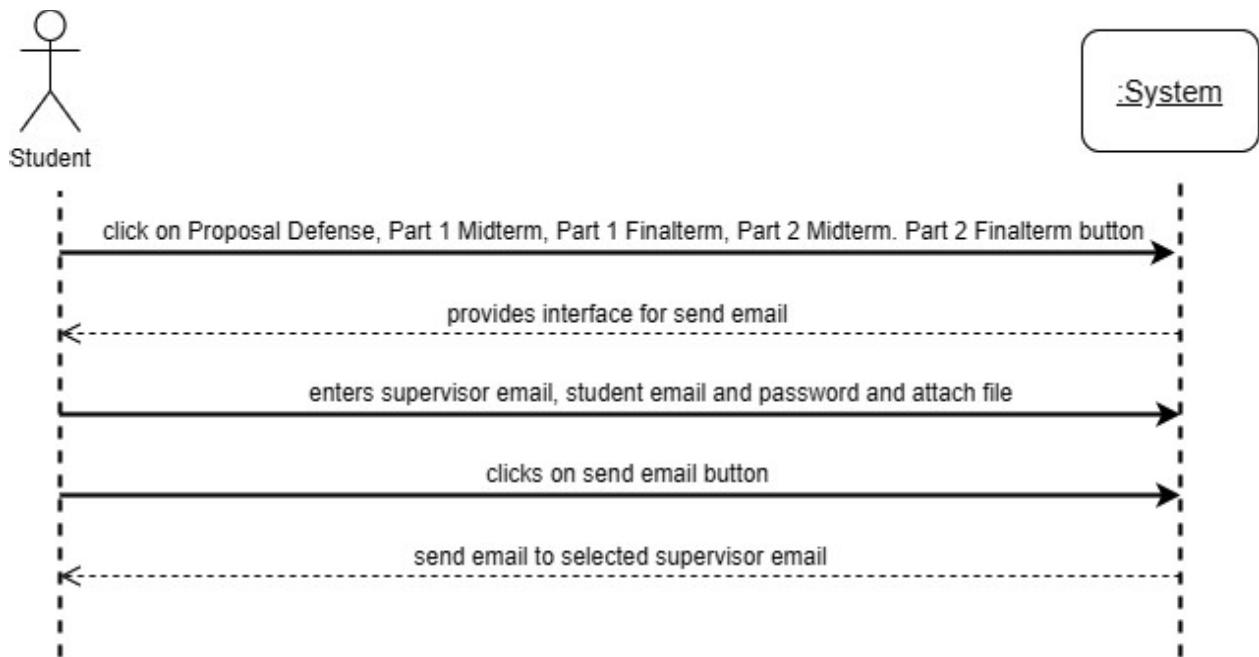


Figure 2.25 Send Email by Student

## 2.6. Domain Model

The following Figure 2.26 shows the domain model of the Final Year project Management and Tracking System.

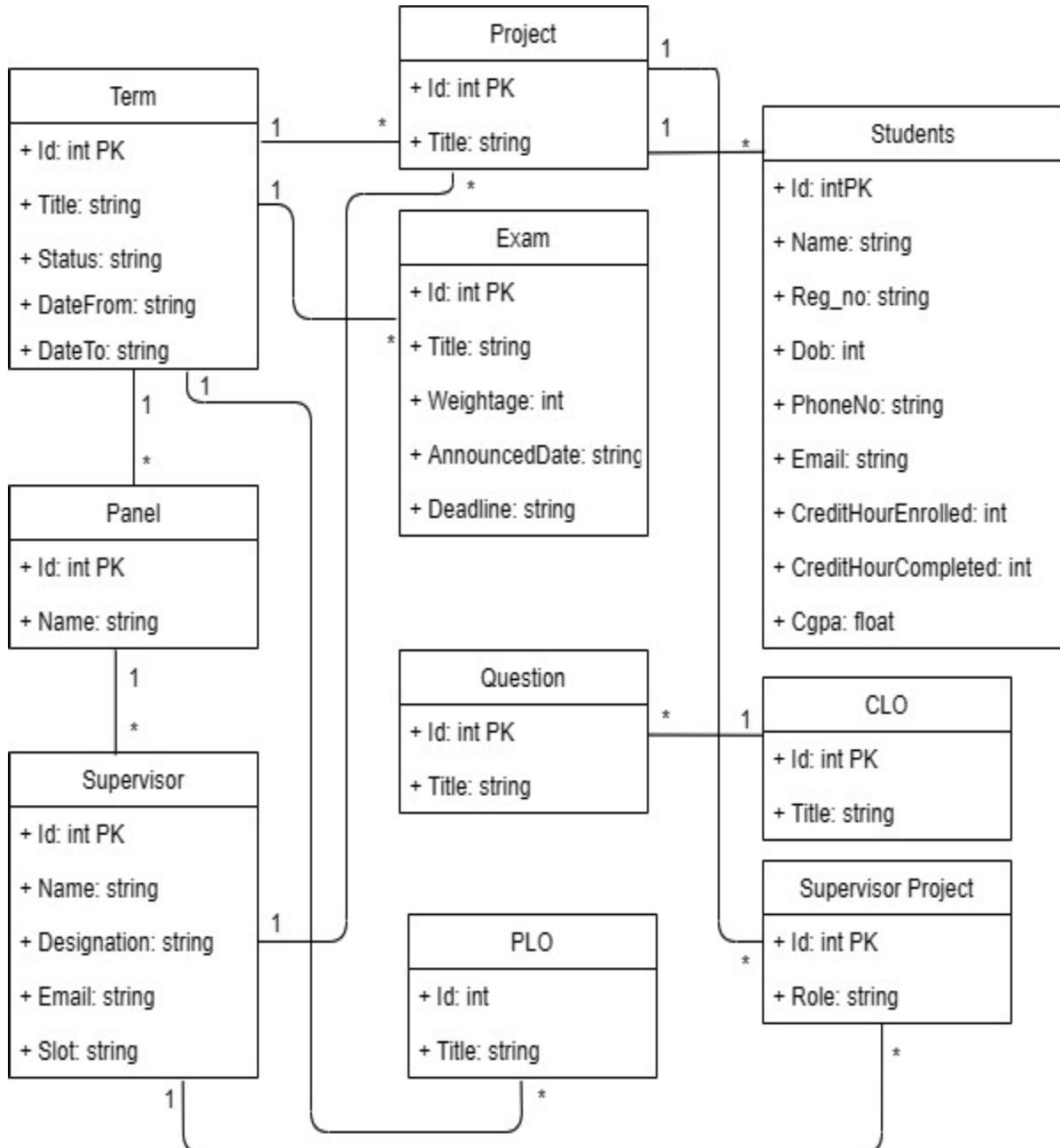


Figure 2.26 Domain Model

# Chapter 3

## System Design

### 3.1. Software Architecture

#### Tier Architecture

The following Figure 3.1 shows the tier architecture of the final year project management and tracking system.

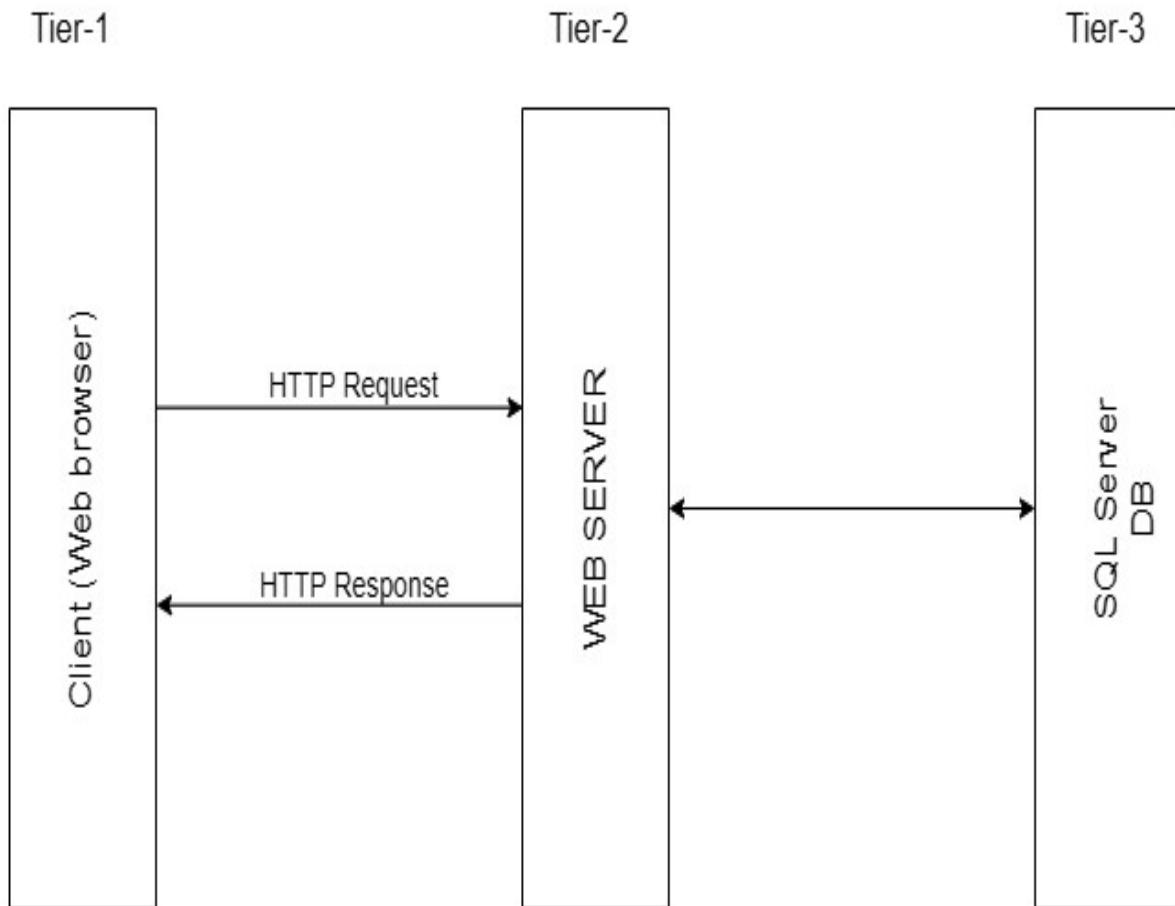


Figure 3.1 Tier Architecture

## Tier-1 Architecture

The following Figure 3.2 shows the architecture of tier-1

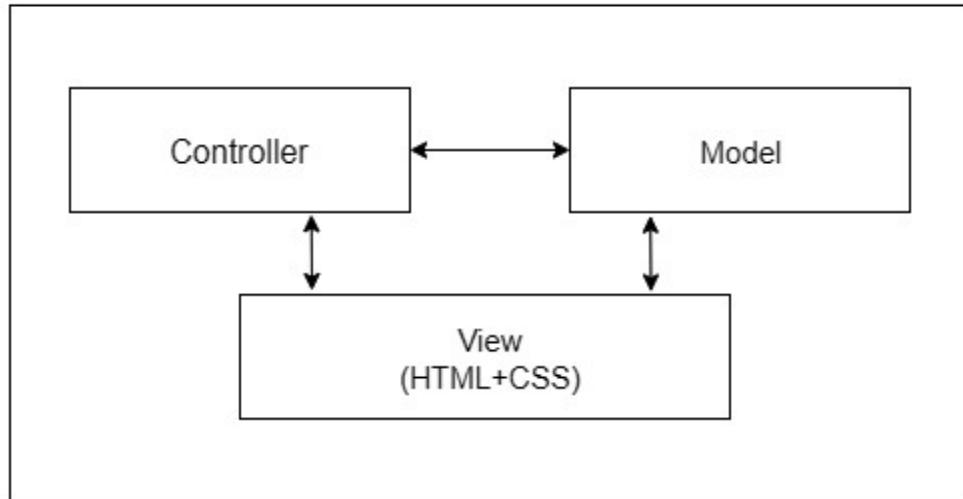


Figure 3.2 Tier-1 Architecture

## Layered Architecture

The following Figure 3.3 shows the layered architecture of tier-2 and tier-3

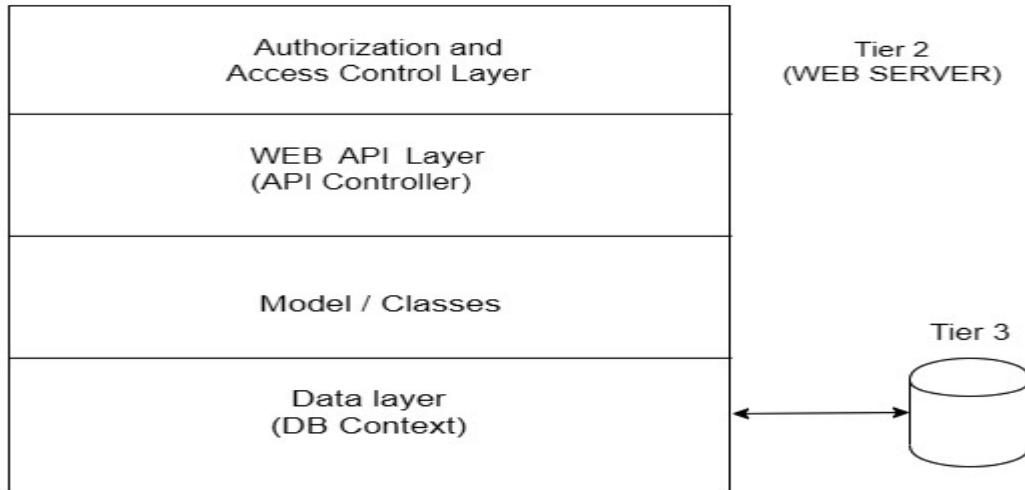


Figure 3.3 Layered Architecture

## 3.2. Class Diagram

The following Figure 3.4 shows the class diagram

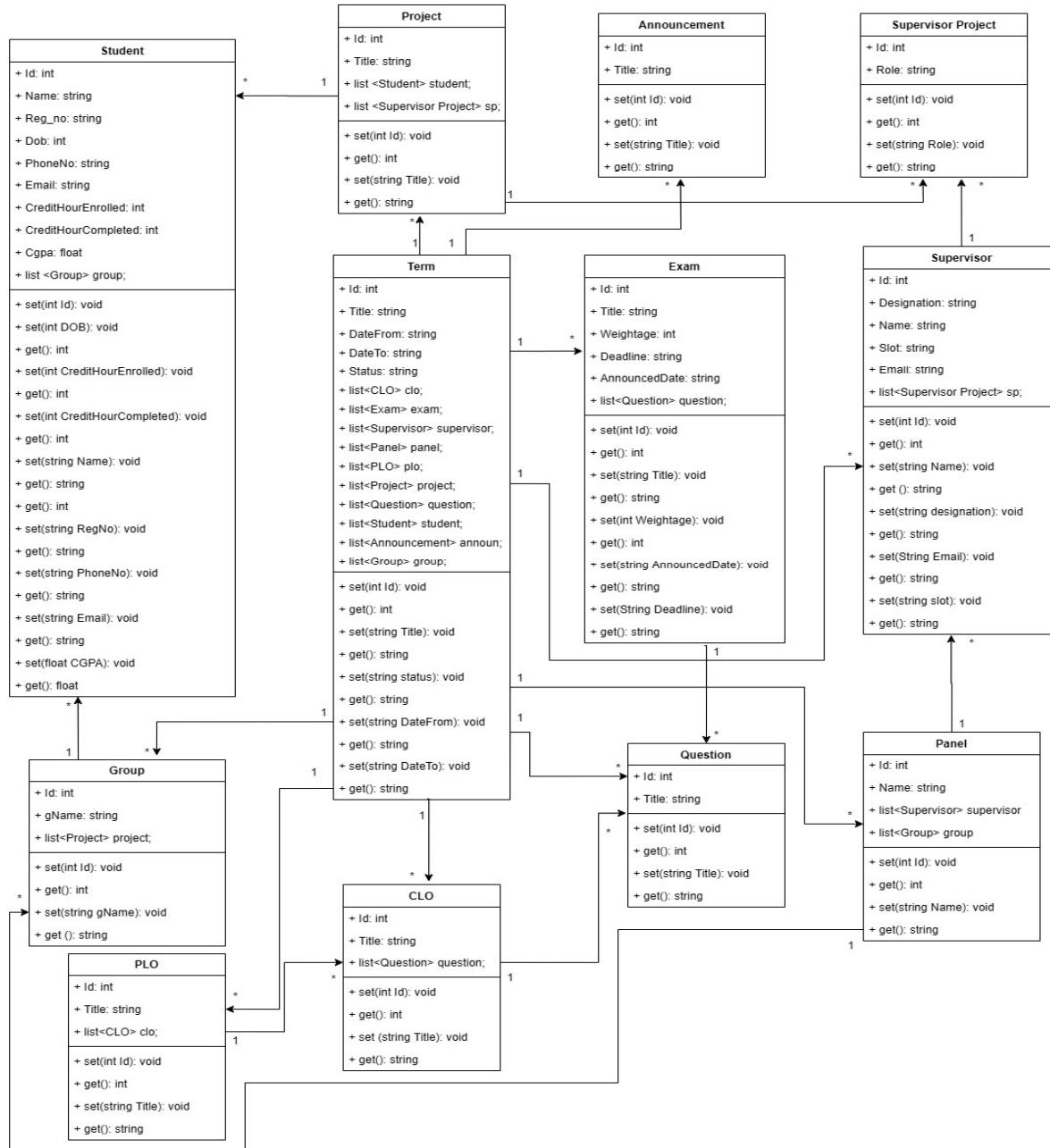


Figure 3.4 Class Diagram

### 3.3. Sequence Diagram

The following Figure 3.5 shows the signup sequence of coordinator

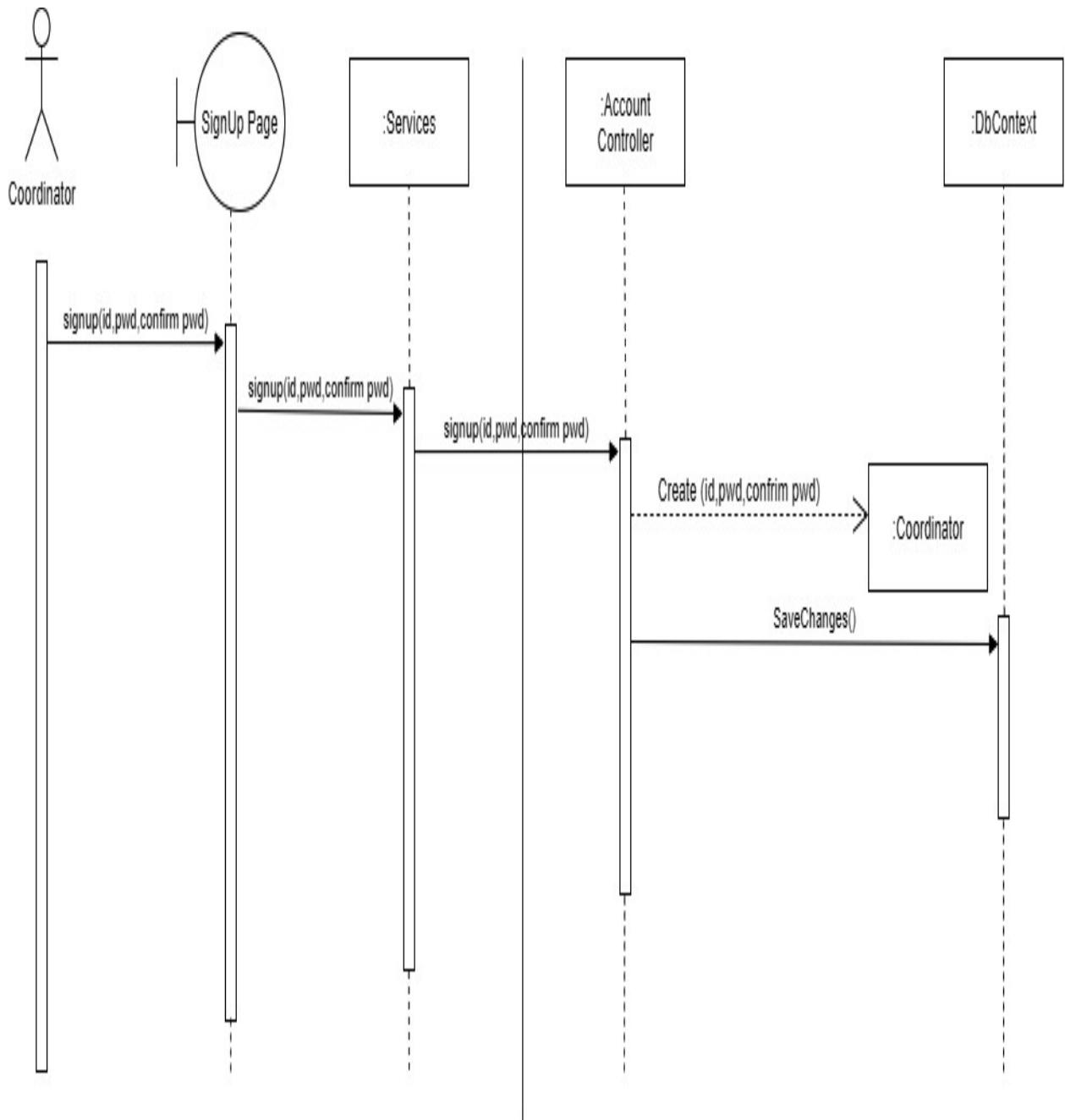


Figure 3.5 Coordinator Sign Up

The following Figure 3.6 shows the login sequence of coordinator

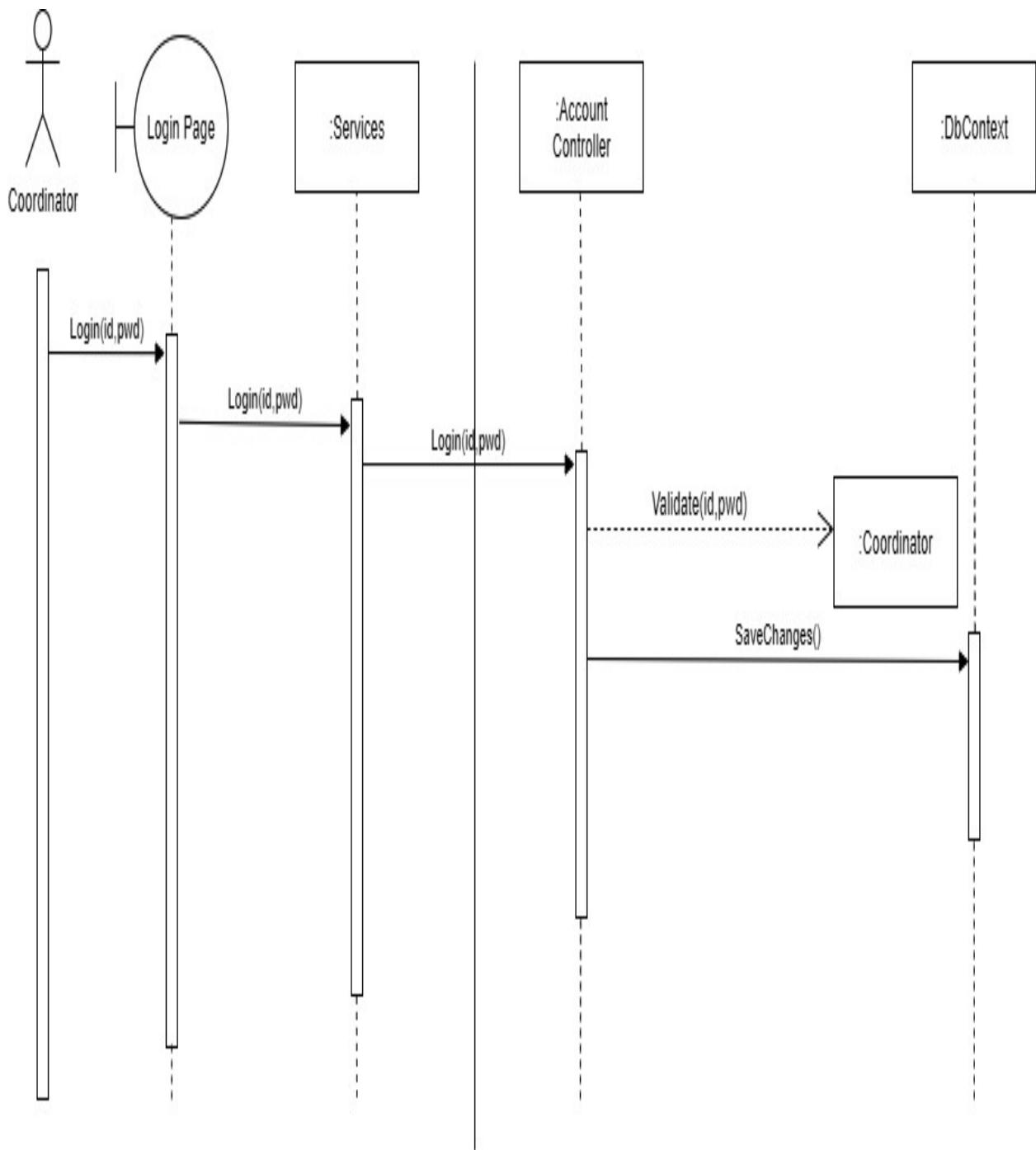


Figure 3.6 Coordinator Login

The following Figure 3.7 shows the login sequence of supervisor

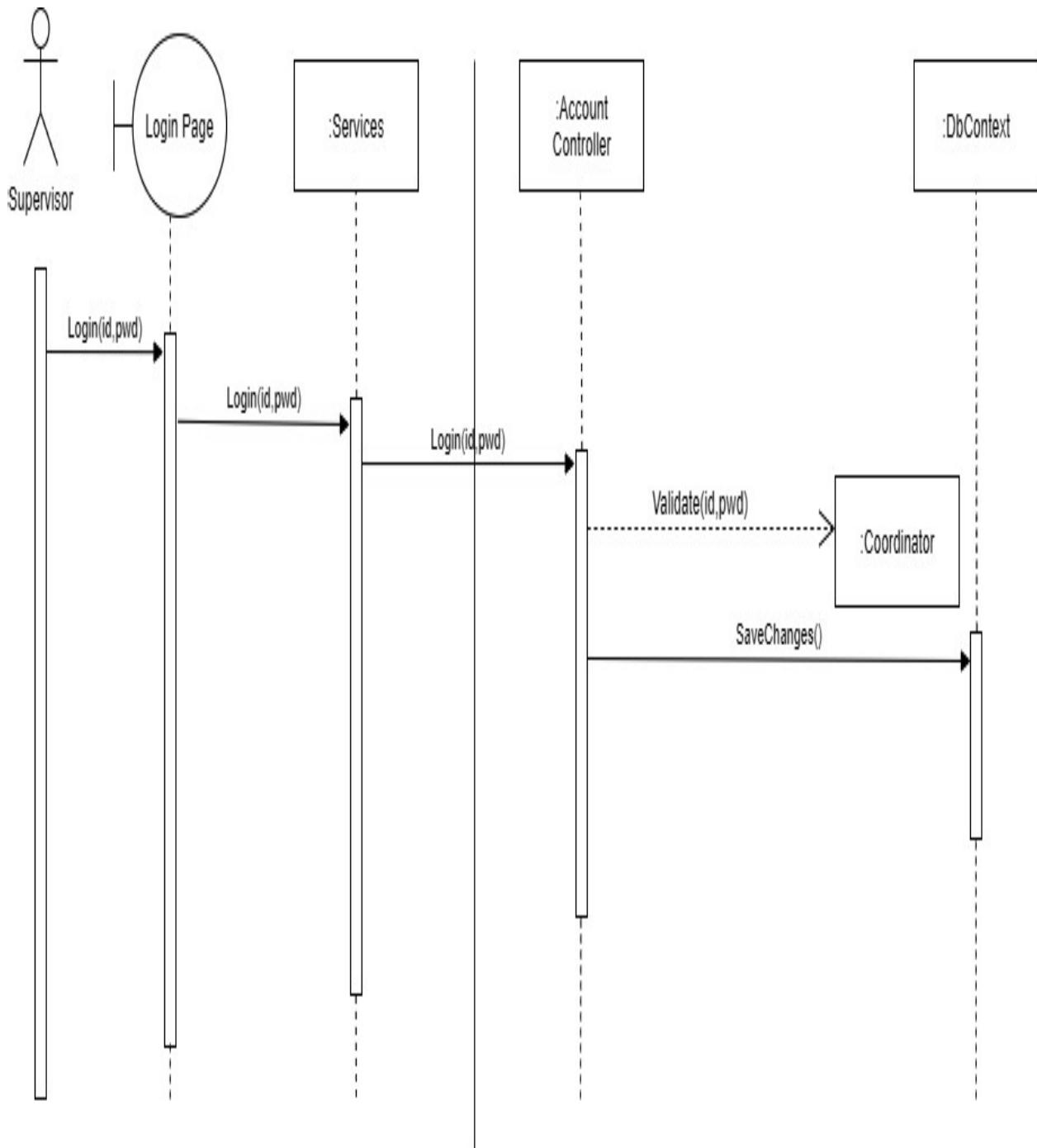


Figure 3.7 Supervisor Login

The following Figure 3.8 shows the login sequence of student

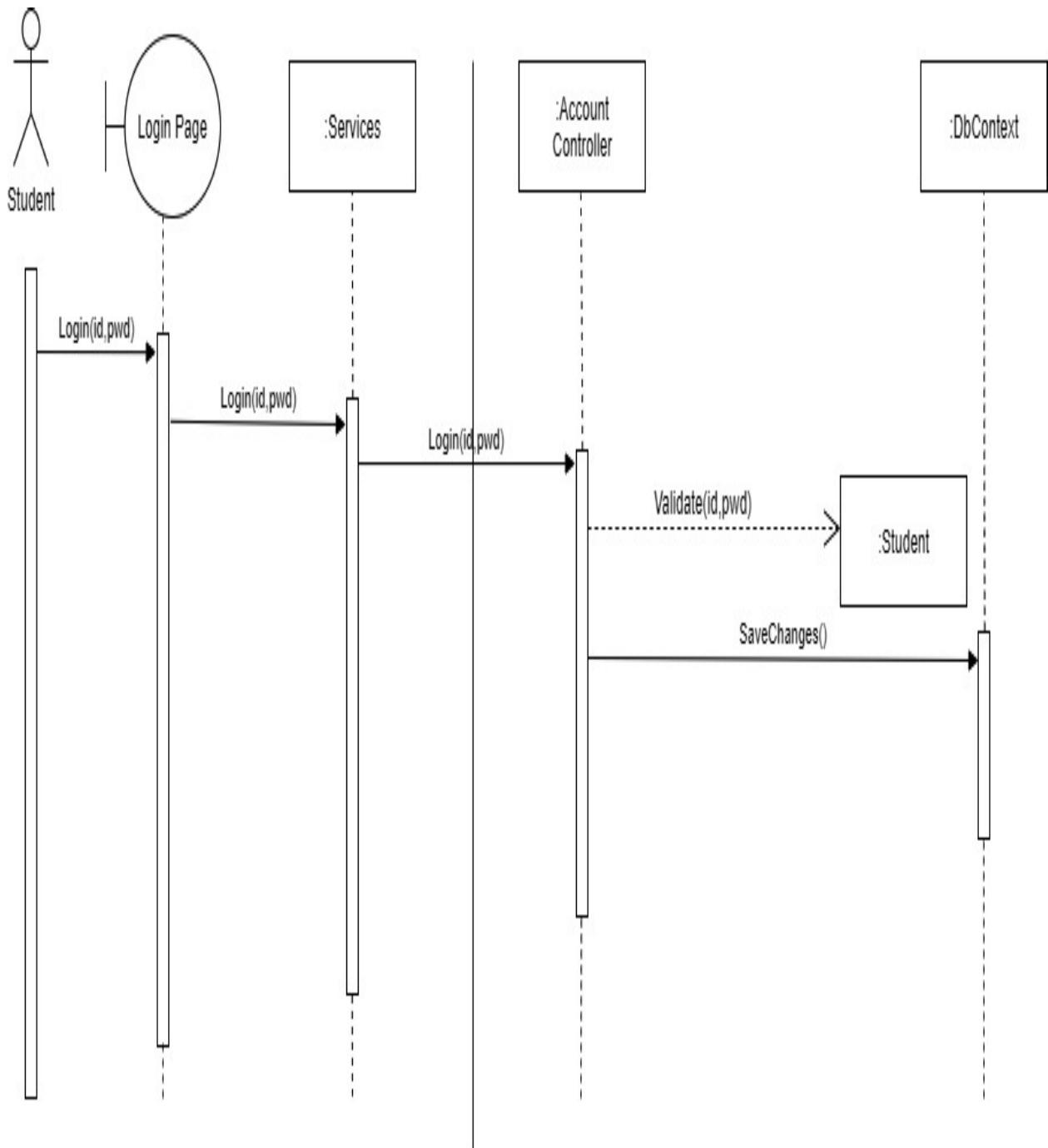


Figure 3.8 Student Login

The following Figure 3.9 shows the register sequence of supervisor and student of coordinator

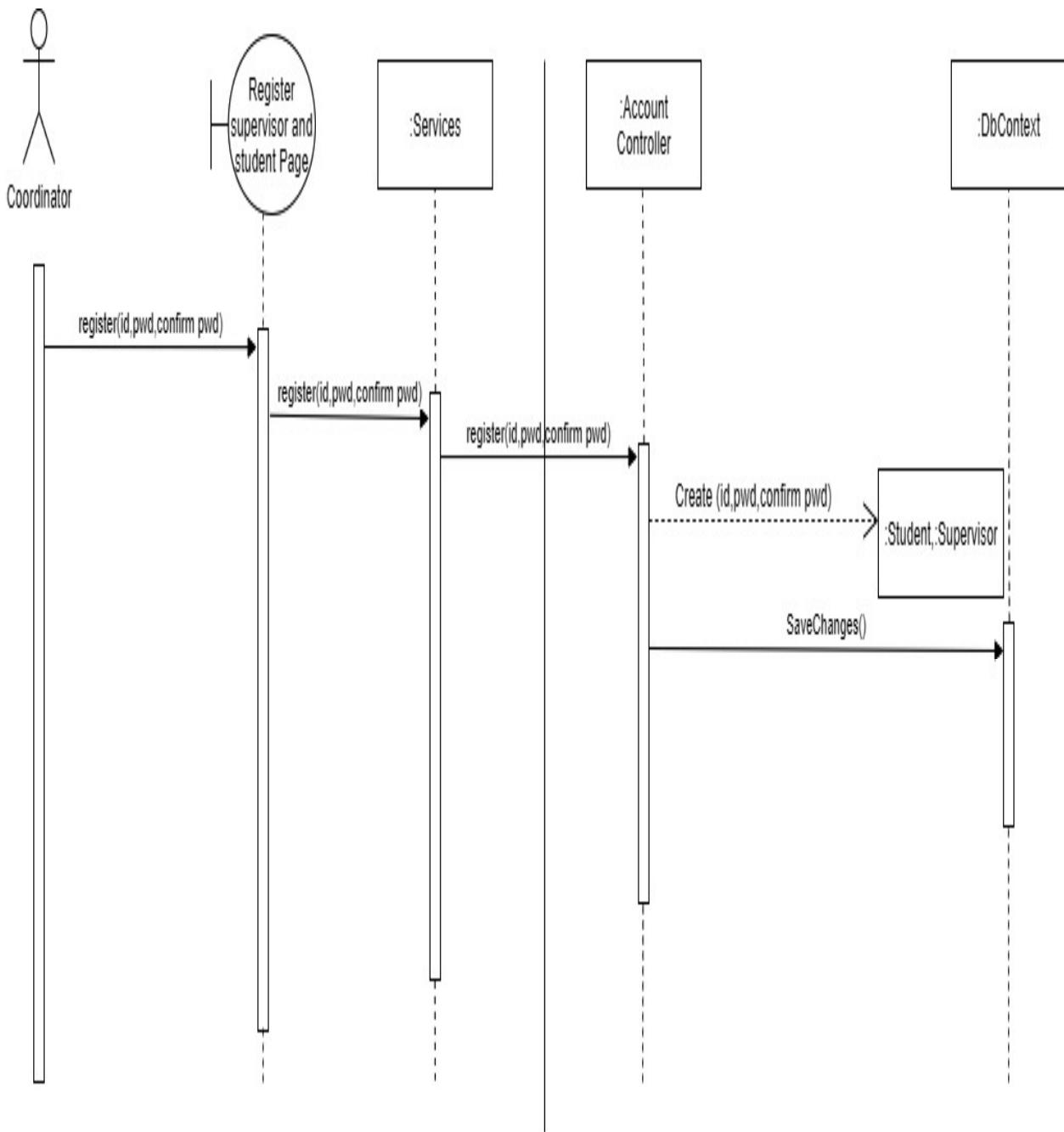


Figure 3.9 Register Supervisors and Students

The following Figure 3.10 shows the create term sequence of coordinator

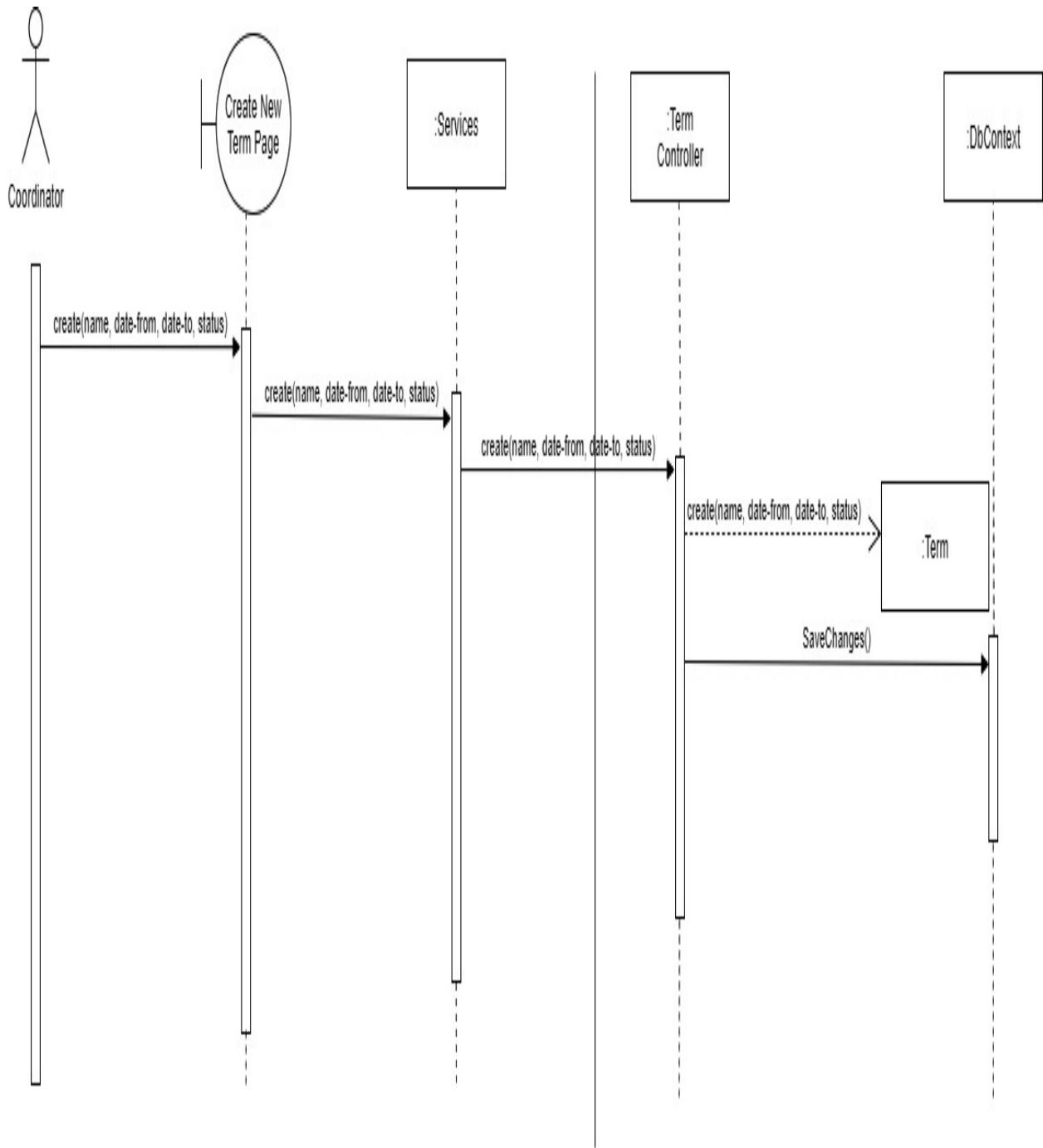


Figure 3.10 Create Term

The following Figure 3.11 shows the create Program Learning Objectives sequence of coordinator

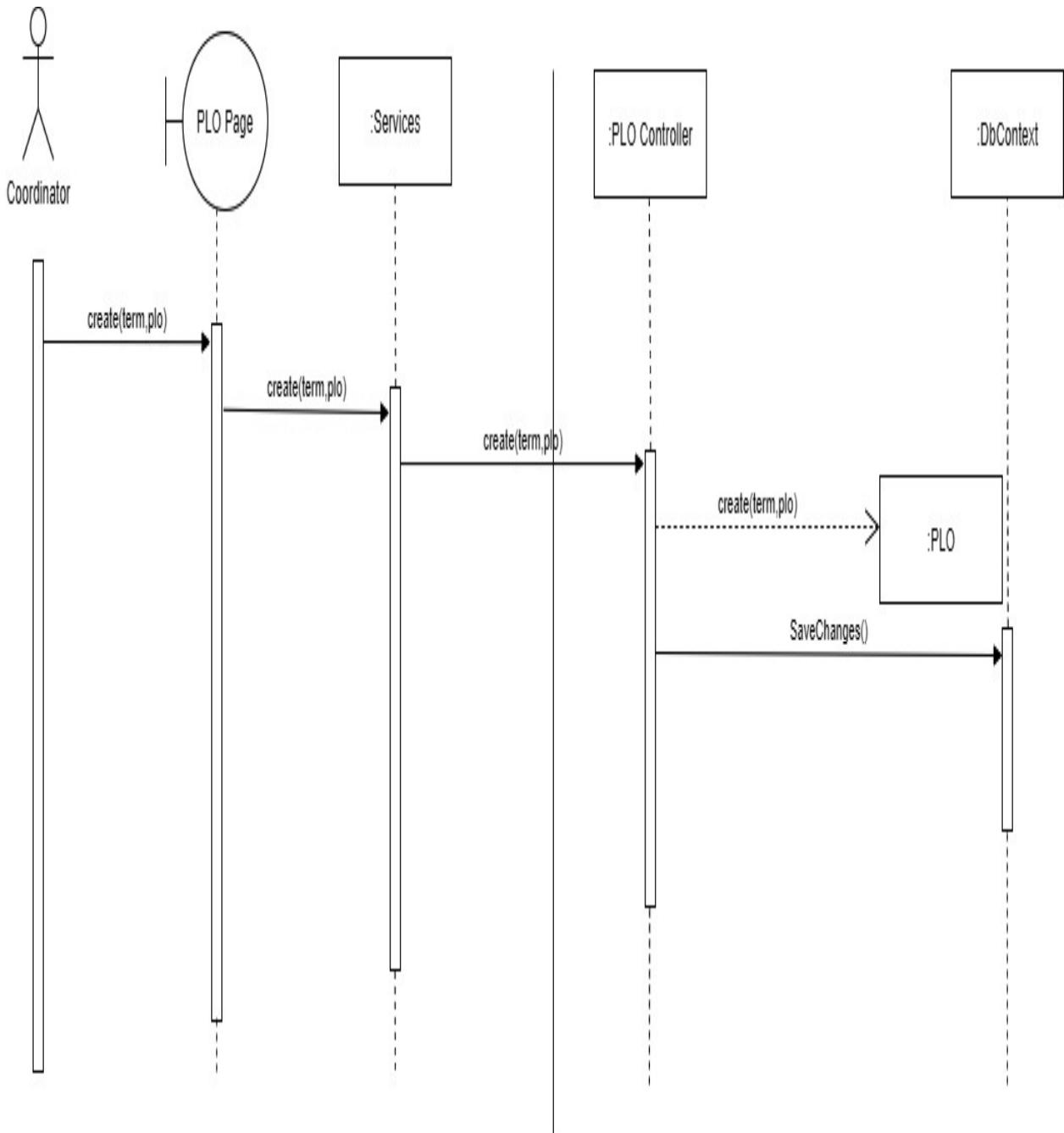


Figure 3.11 Create Program Learning Objectives

The following Figure 3.12 shows the create Course Learning Objectives sequence of coordinator

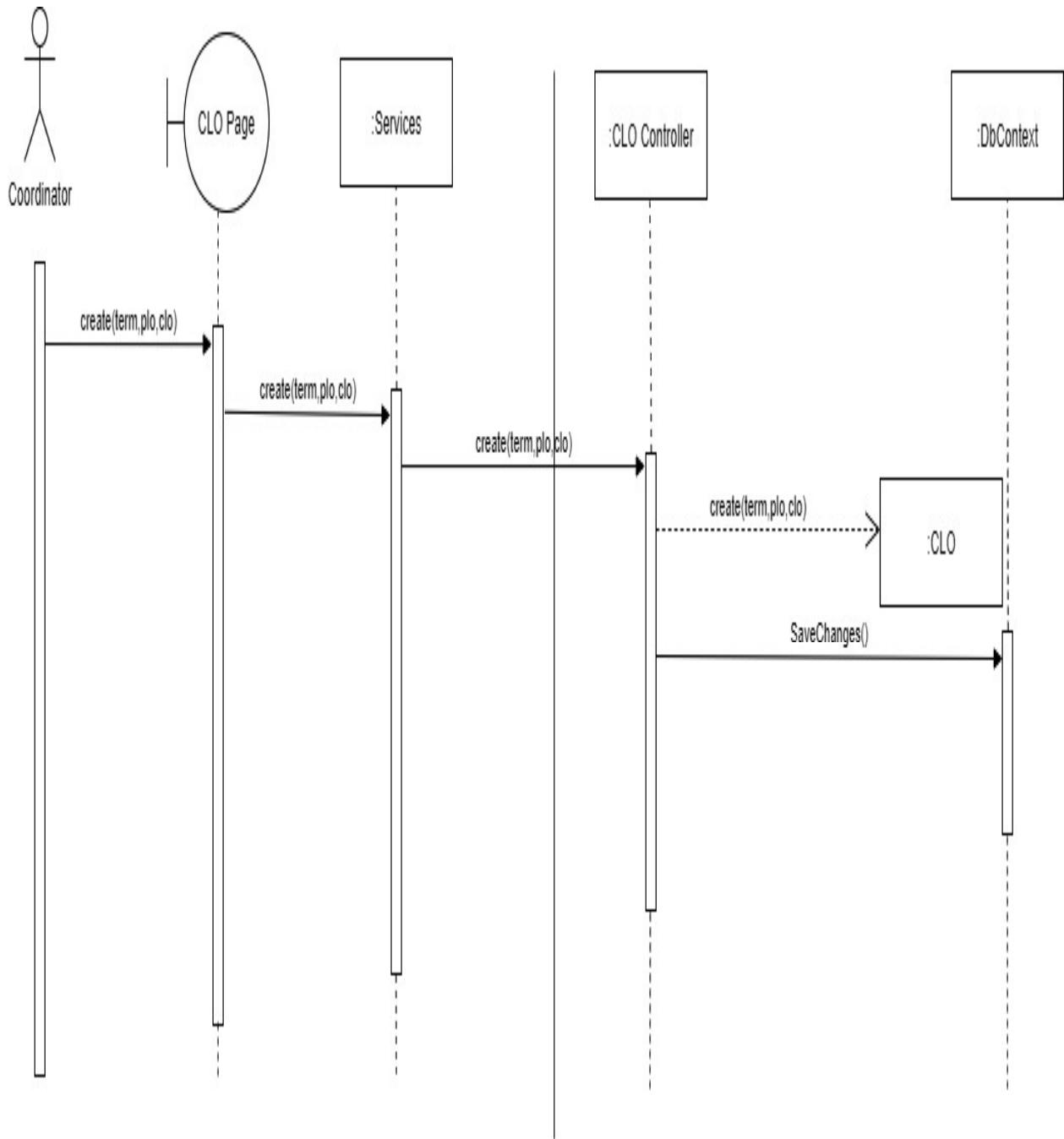


Figure 3.12 Create Course Learning Objectives

The following Figure 3.13 shows the create exam sequence of coordinator

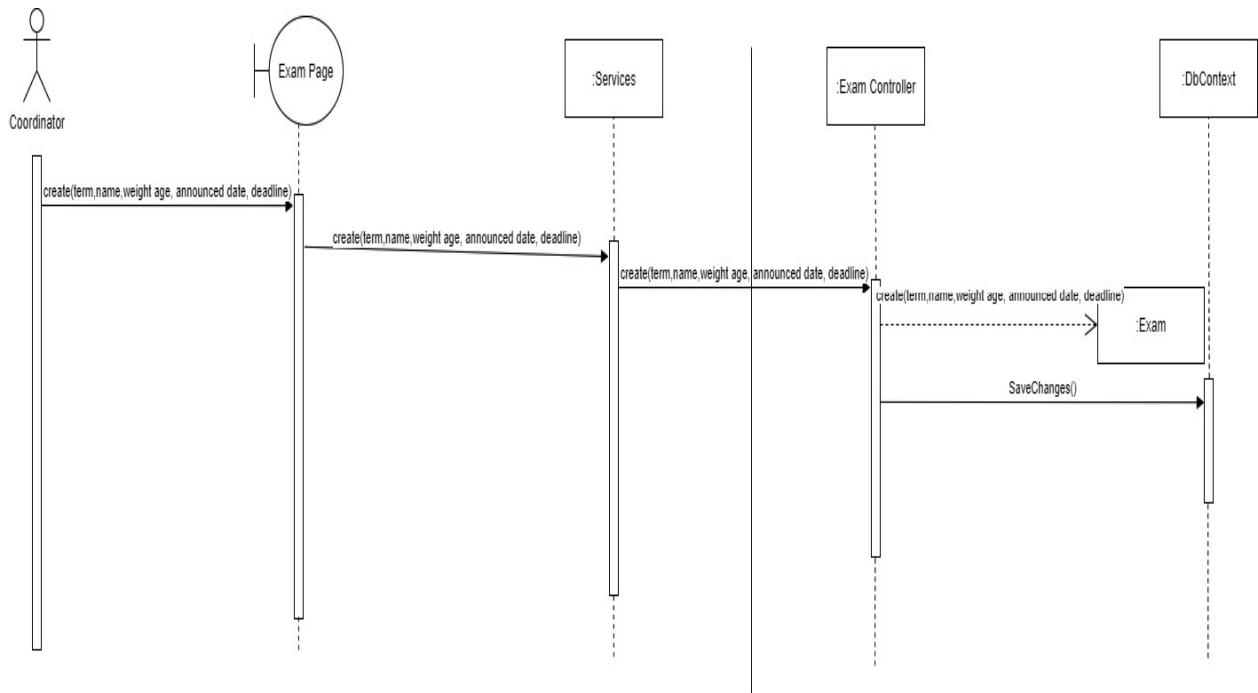


Figure 3.13 Create Exam

The following Figure 3.14 shows the create question sequence of coordinator

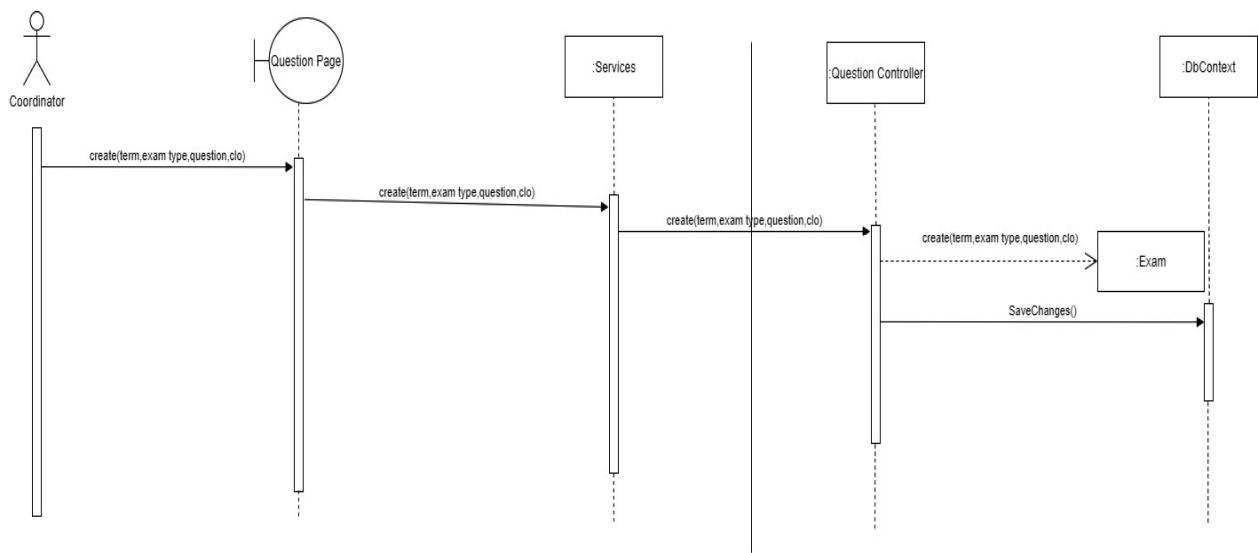


Figure 3.14 Create Question

The following Figure 3.15 shows the create panel sequence of coordinator

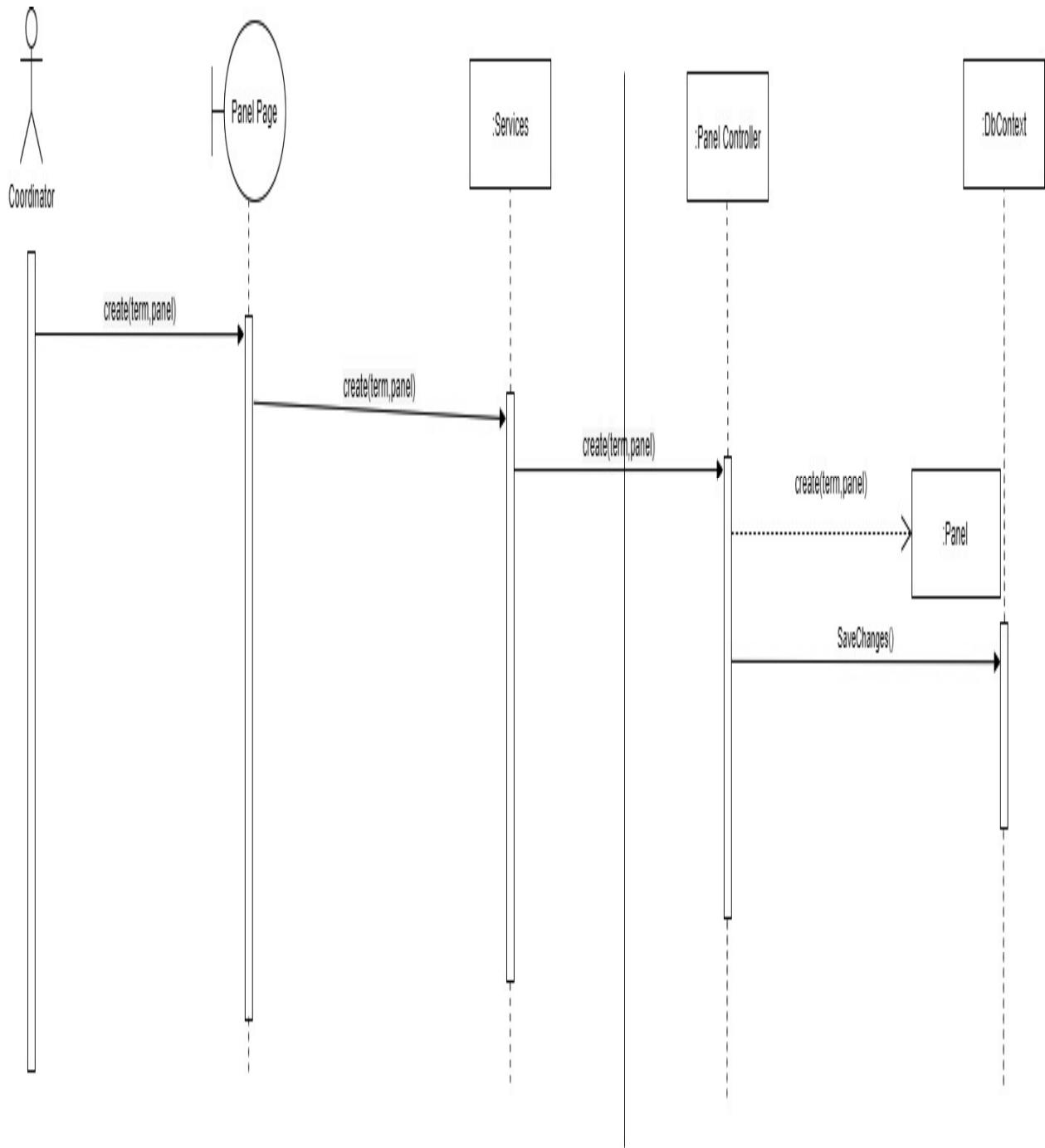


Figure 3.15 Create Panel

The following Figure 3.16 shows the create supervisor sequence of coordinator

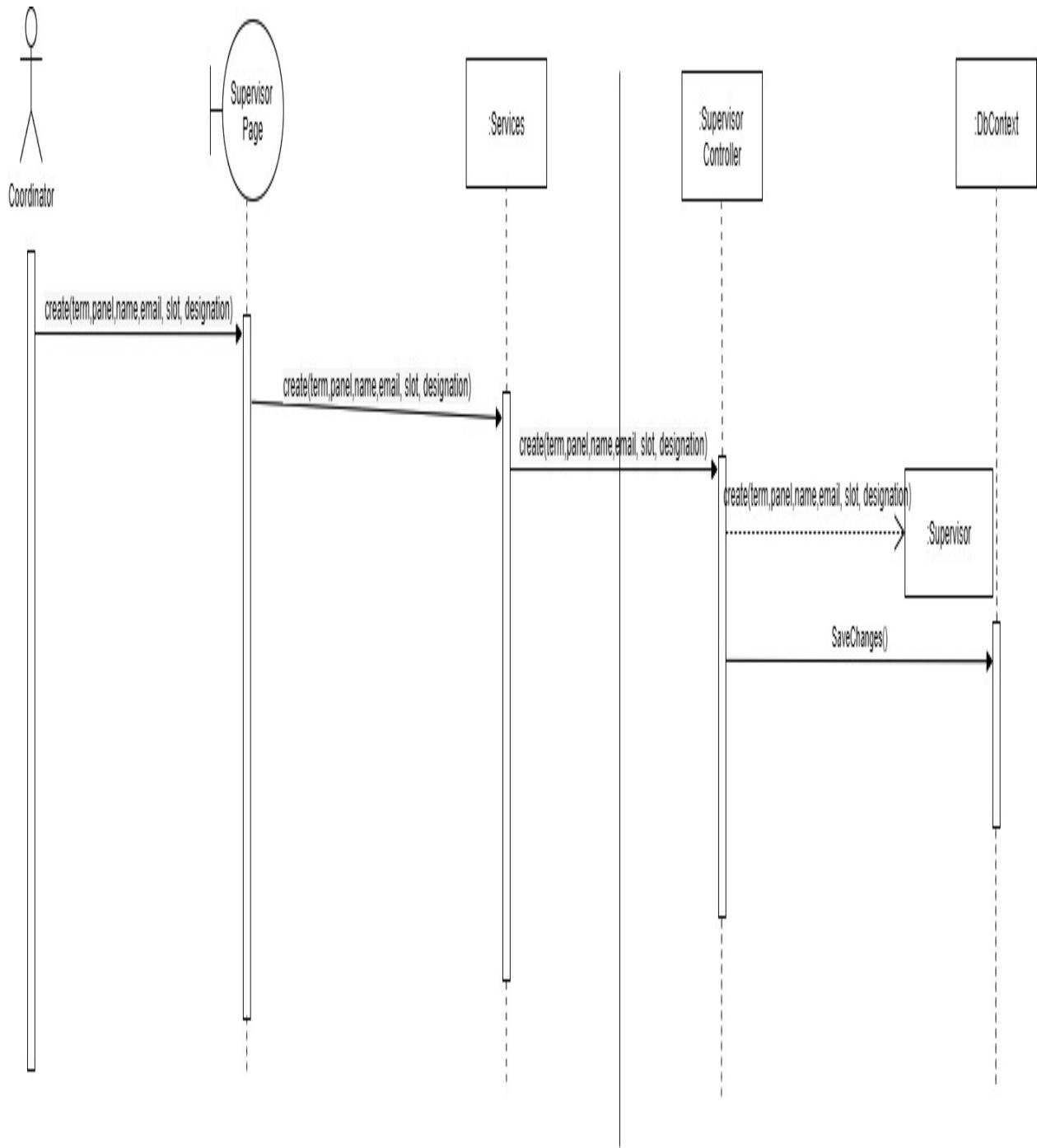


Figure 3.16 Create Supervisor

The following Figure 3.17 shows the create announcement sequence of coordinator

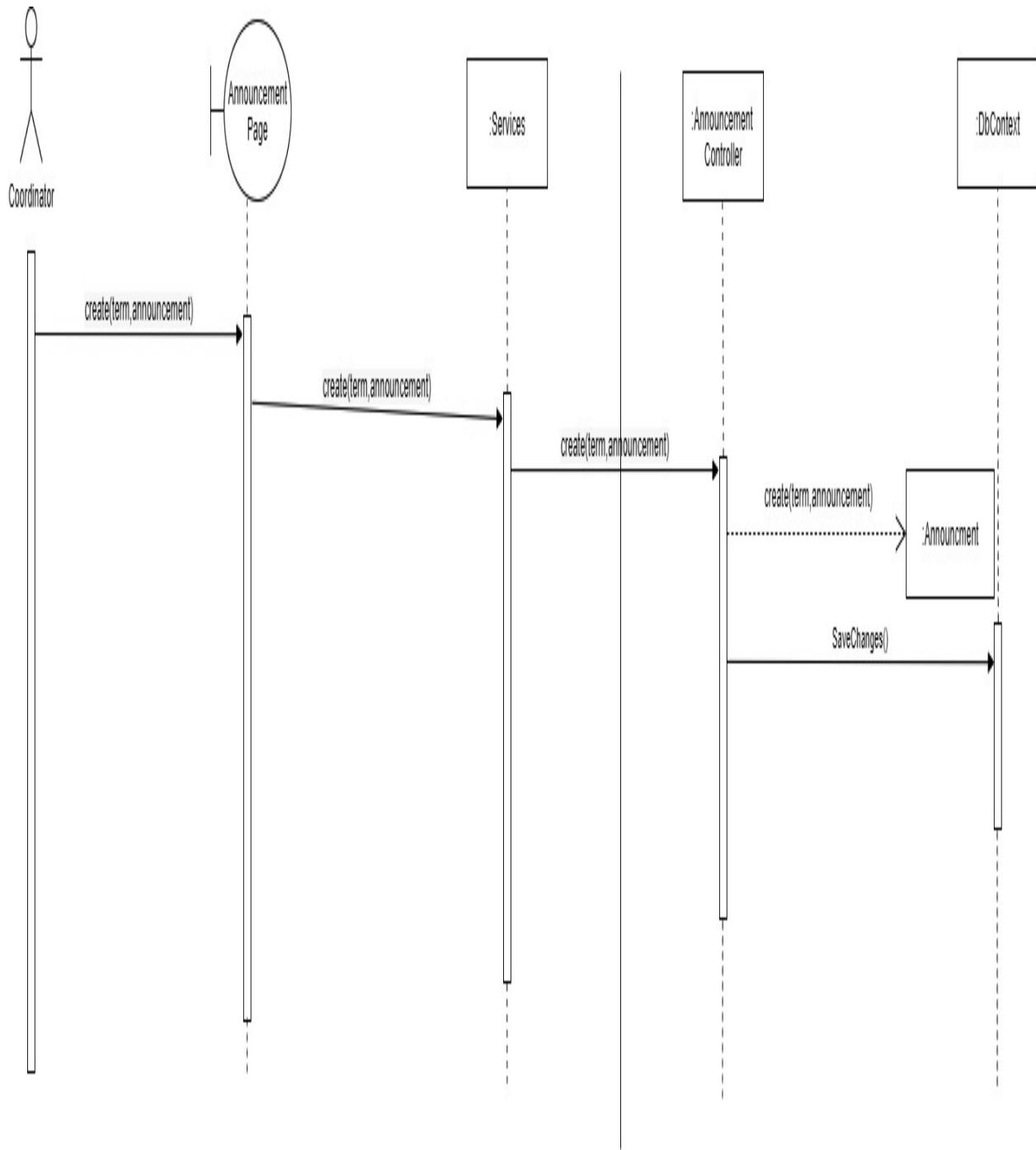


Figure 3.17 Create Announcement

The following Figure 3.18 shows the view announcement sequence of supervisor

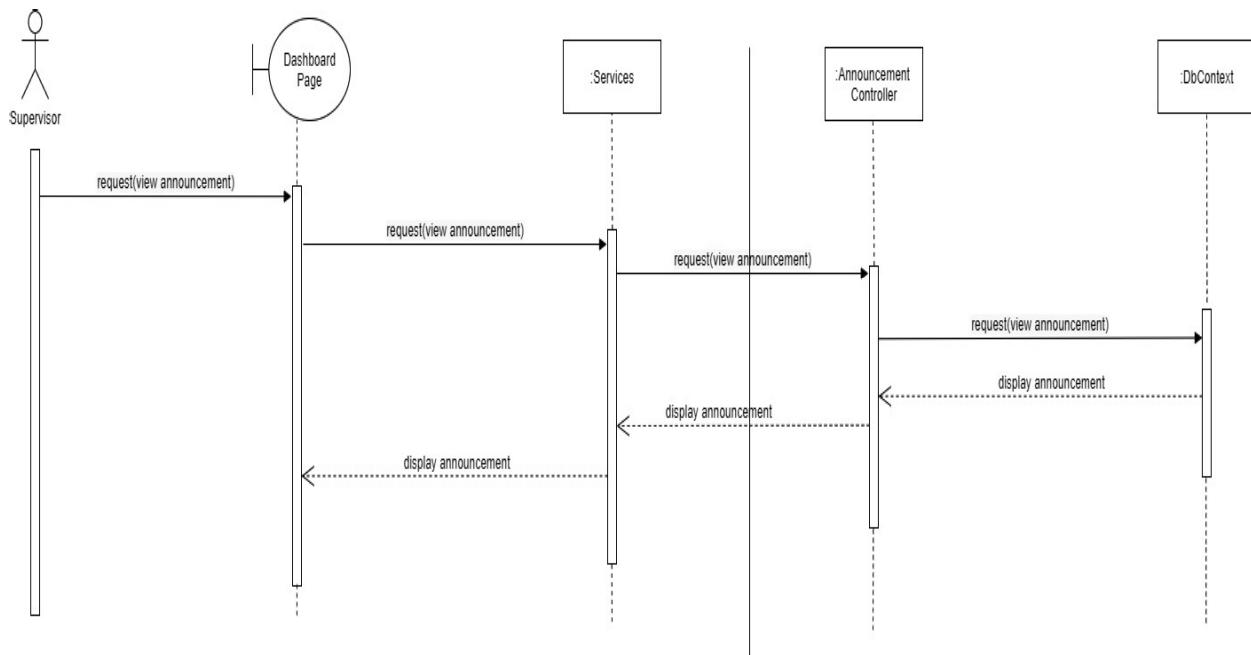


Figure 3.18 View Announcements by Supervisor

The following Figure 3.19 shows the view announcement sequence of student

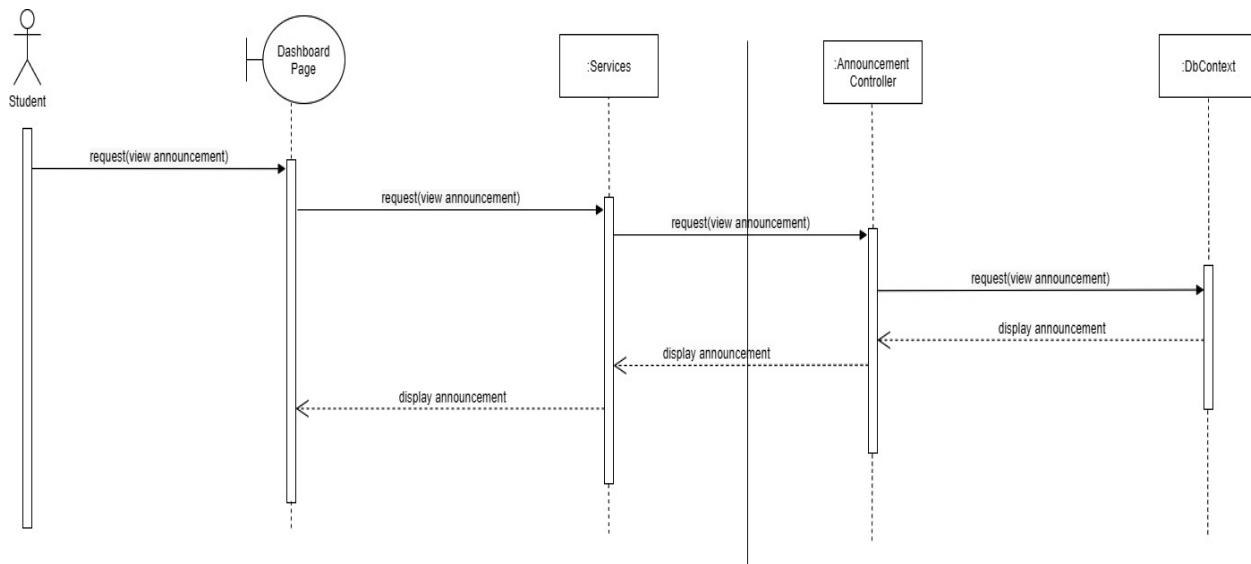


Figure 3.19 View Announcements by Student

The following Figure 3.20 shows the create group sequence of supervisor

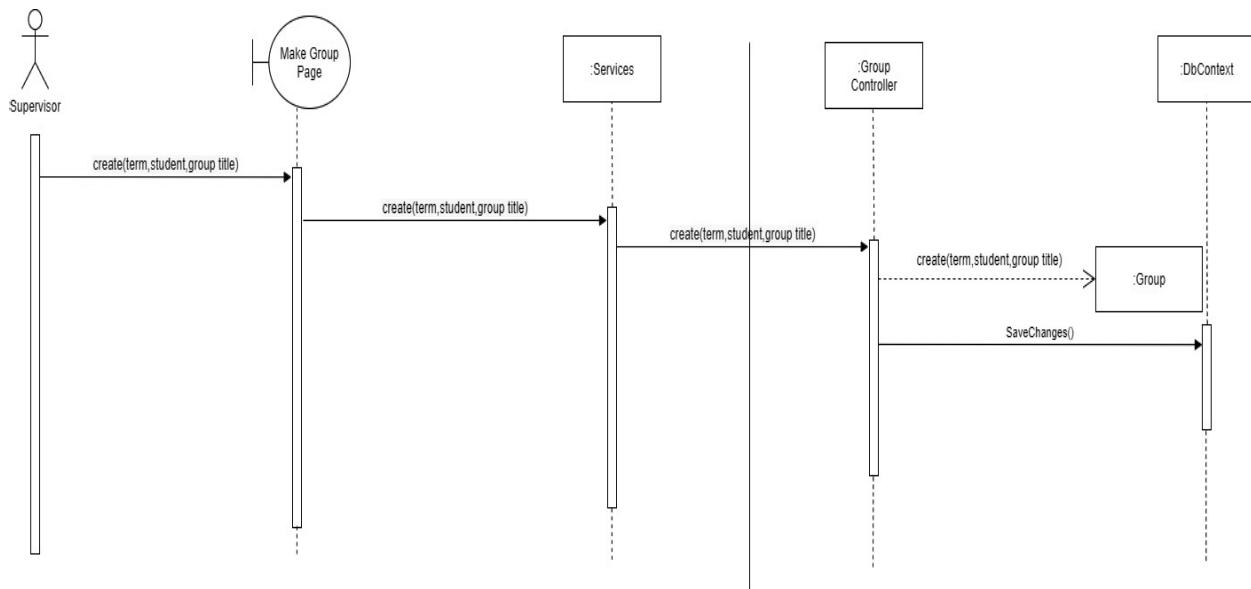


Figure 3.20 Create Group

The following Figure 3.21 shows the Assign Project Title sequence of supervisor

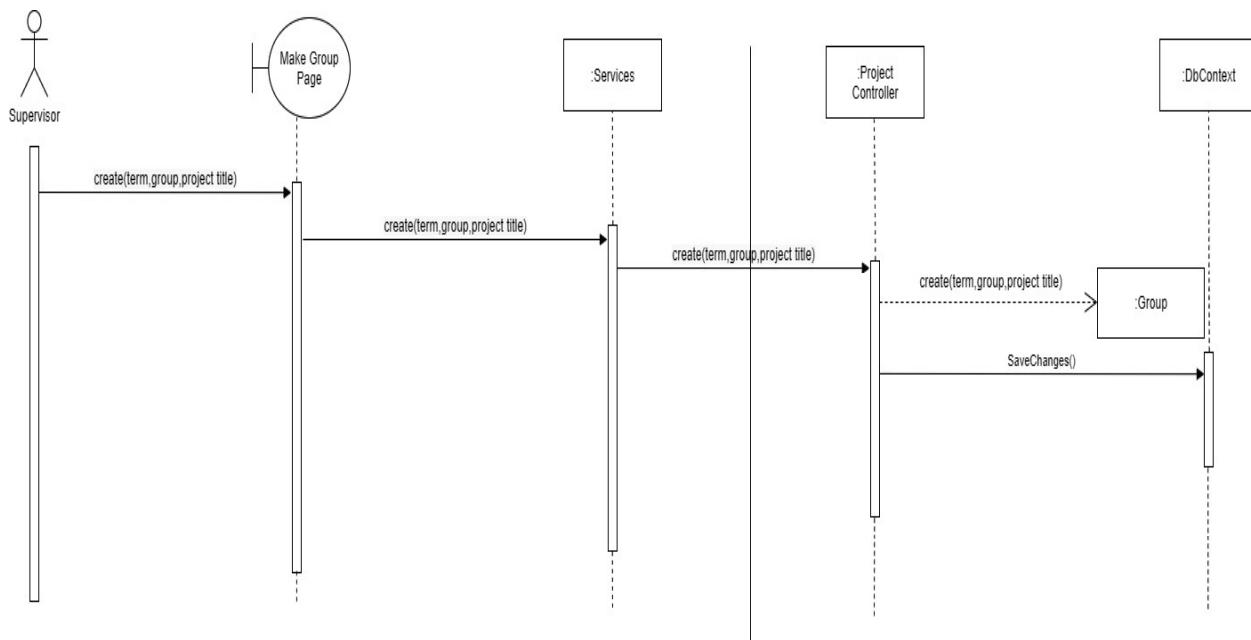


Figure 3.21 Assign Project Title

The following Figure 3.22 shows the send email sequence of supervisor

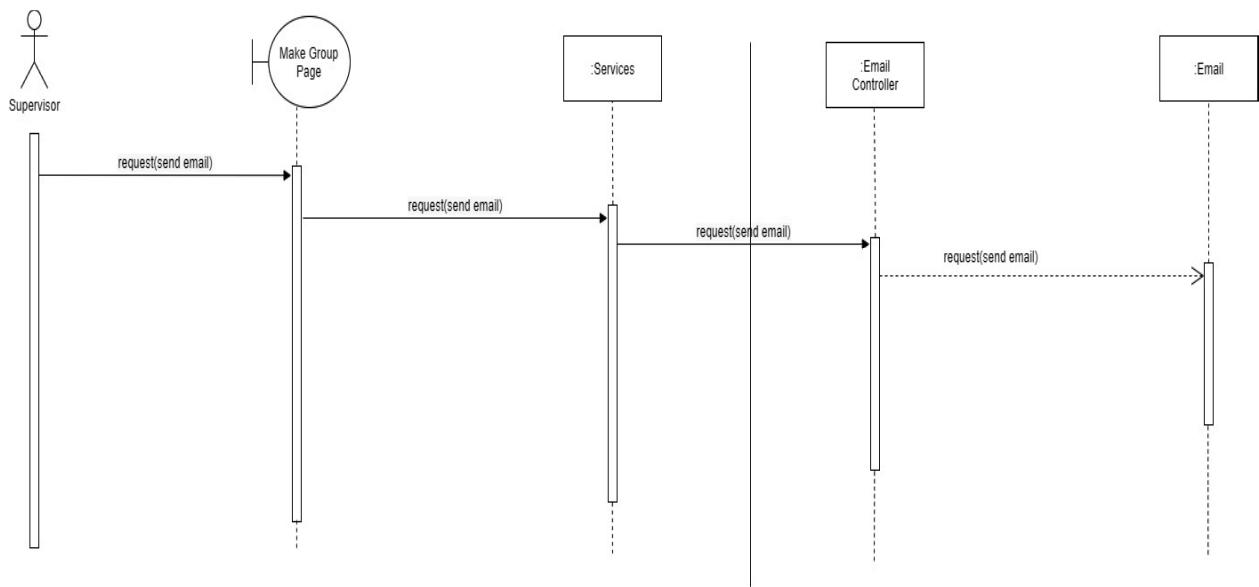


Figure 3.22 Send Email by Supervisor

The following Figure 3.23 shows the view project and panel details sequence of supervisor

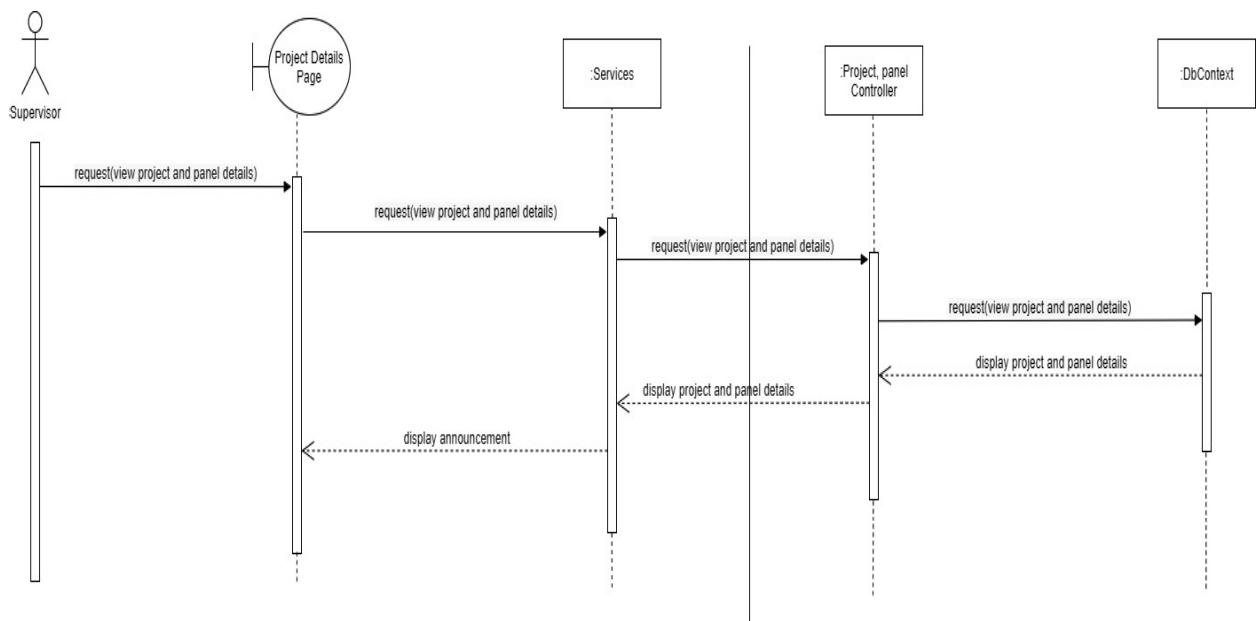


Figure 3.23 View Project and Panel details

The following Figure 3.24 shows the view question sequence of supervisor

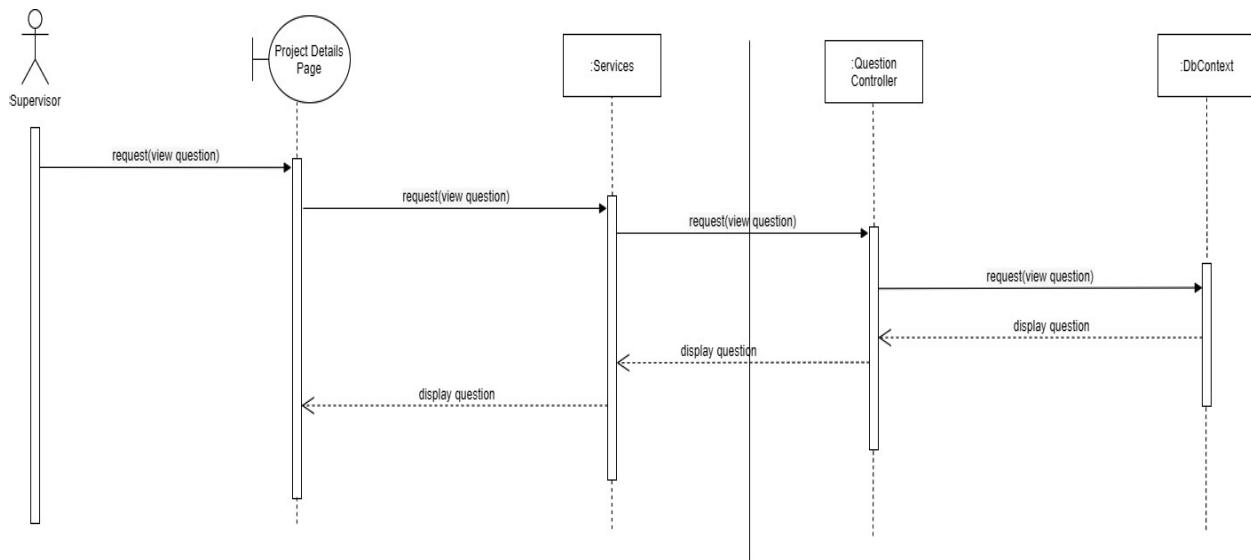


Figure 3.24 View Question by Supervisor

The following Figure 3.25 shows the view question sequence of student

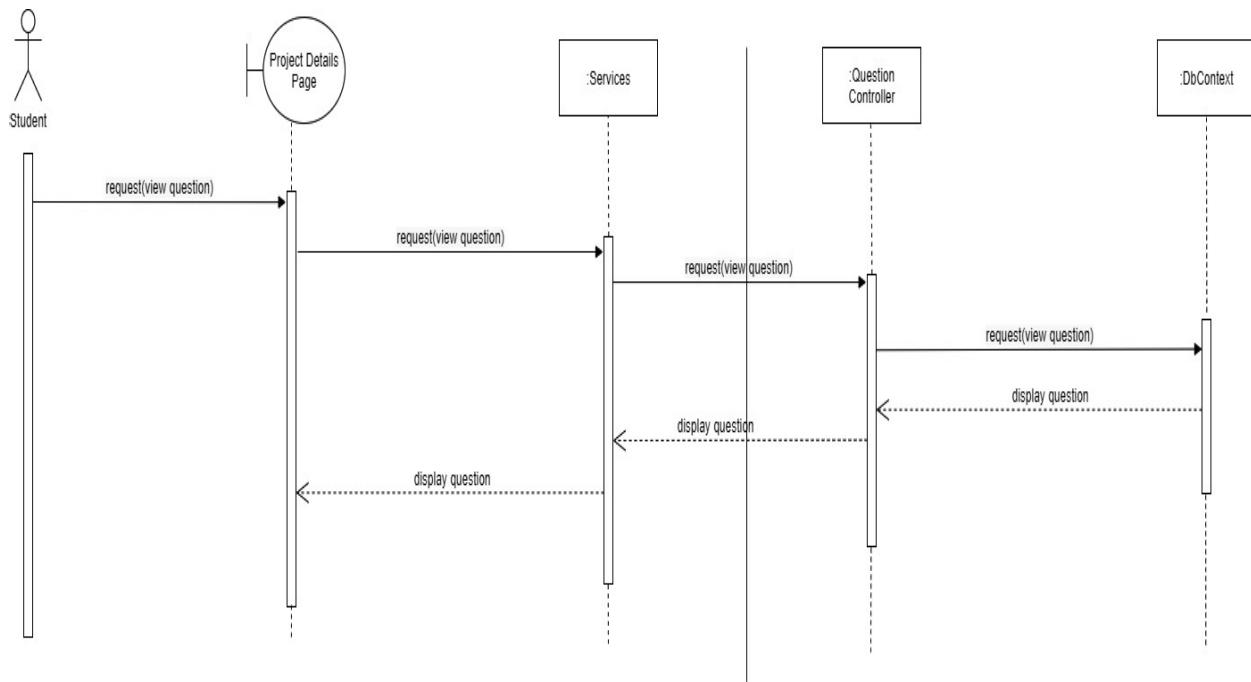


Figure 3.25 View Question by Student

The following Figure 3.26 shows the register student sequence of student

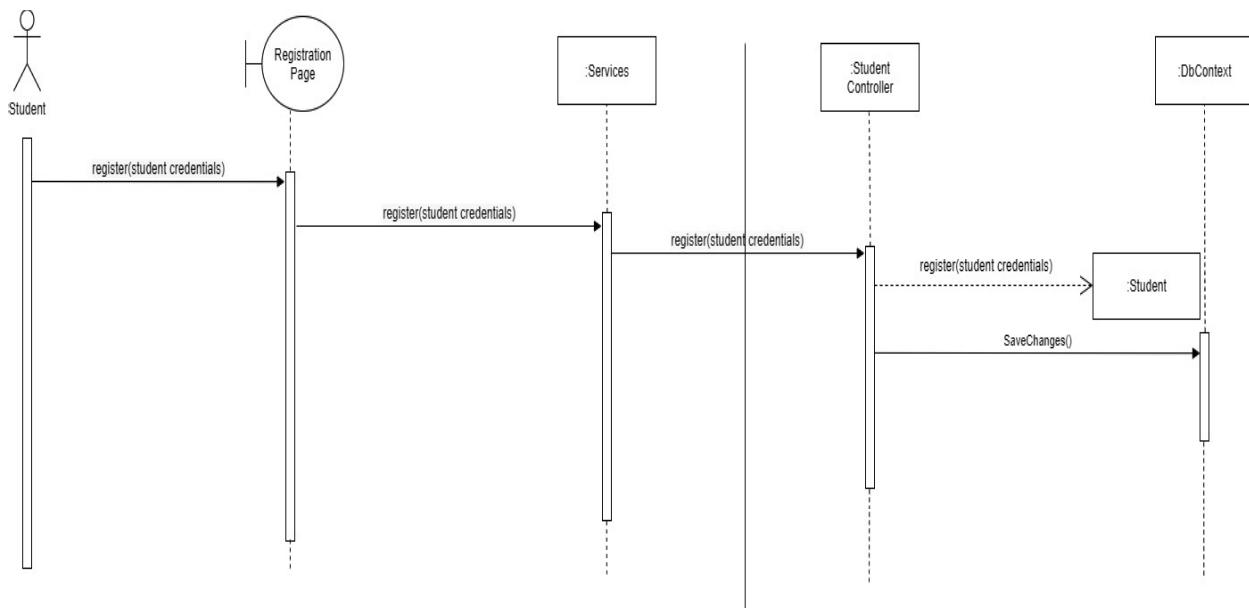


Figure 3.26 Register Student

The following Figure 3.27 shows the view supervisor's list sequence of student

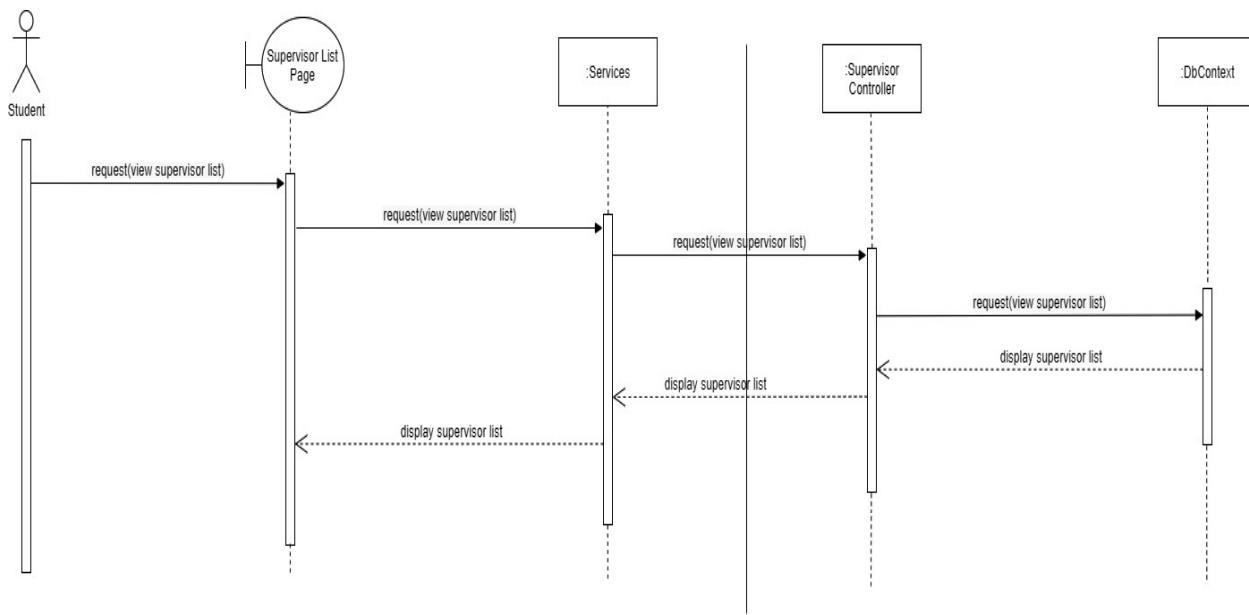


Figure 3.27 View Supervisor's List

The following Figure 3.28 shows the send email sequence of student

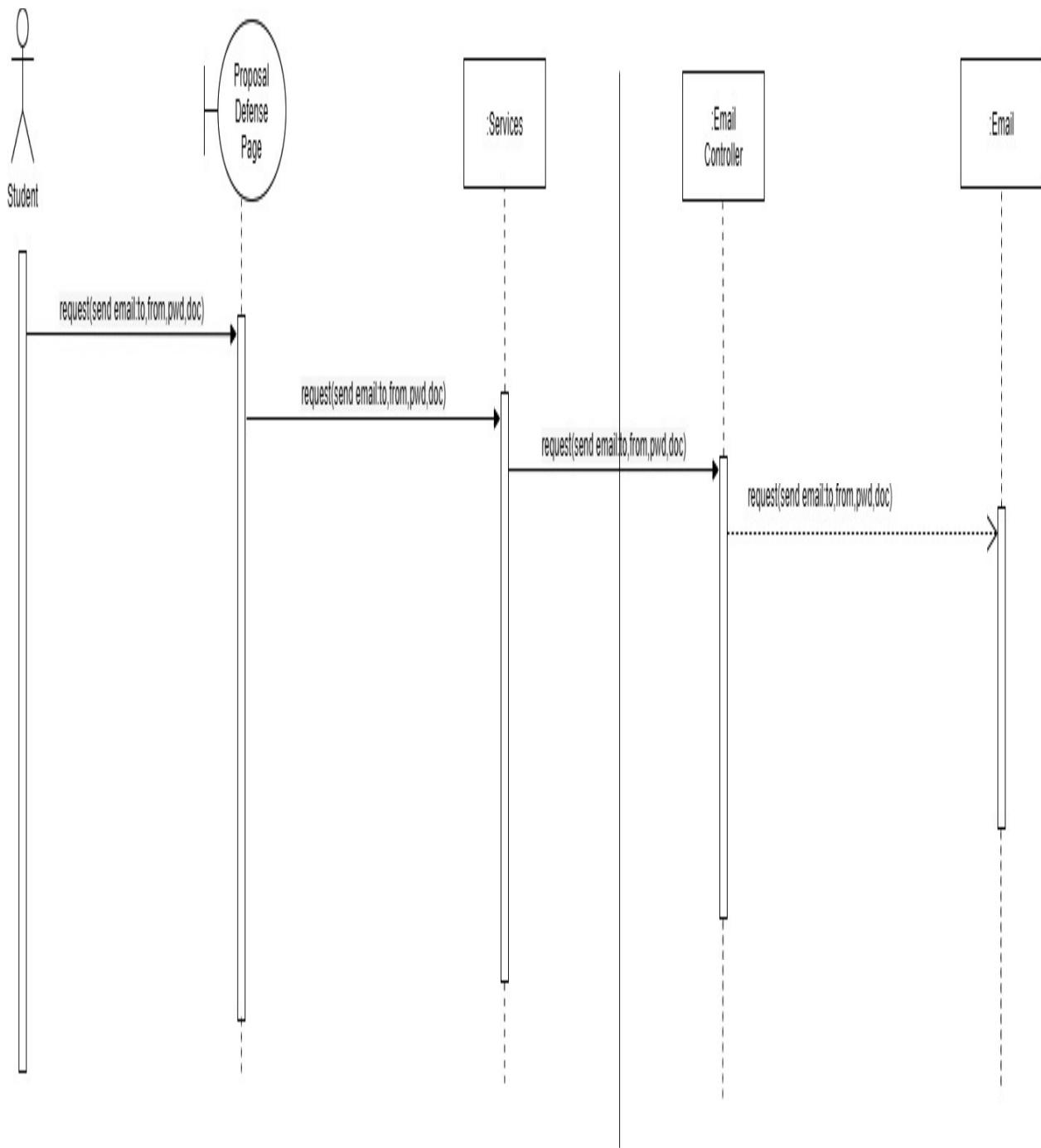


Figure 3.28 Send Email by Student

### 3.4. Entity Relationship Diagram

The following Figure 3.29 shows the Entity Relationship diagram of the project

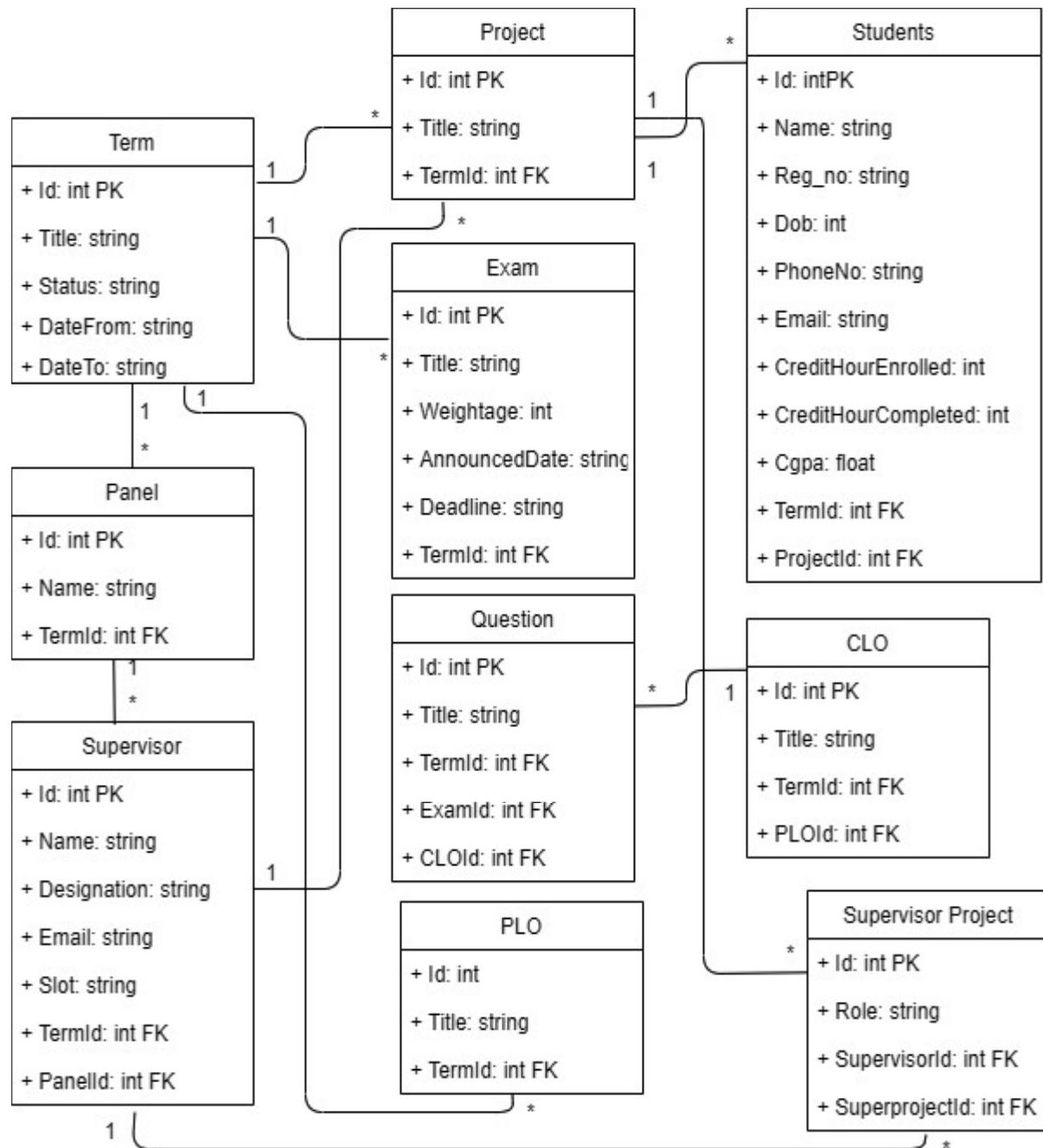


Figure 3.29 Entity Relationship Diagram

### 3.5. Database Schema

The following Figure 3.30 shows the Database schema of the project



Figure 3.30 Database Schema

### 3.6. User Interface Design

Coordinator, Supervisor and Student can interact with the system with main page as shown in Figure 3.31



Figure 3.31 Main Page

Coordinator can sign up into the application by providing email address; password and confirm password as shown in Figure 3.32

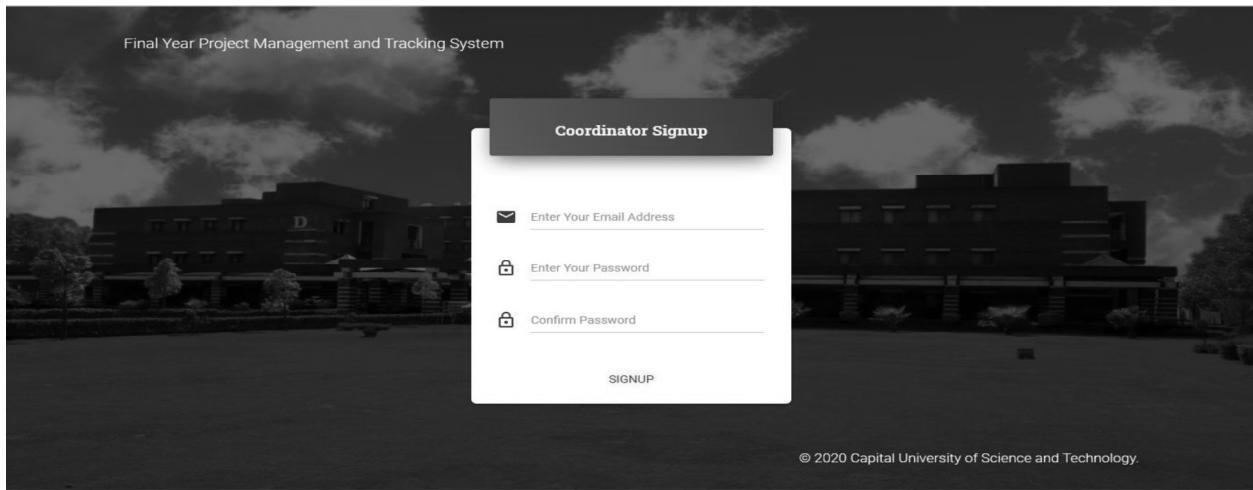


Figure 3.32 Coordinator Sign Up

Coordinator can login to the application by providing email address and password as shown in Figure 3.33

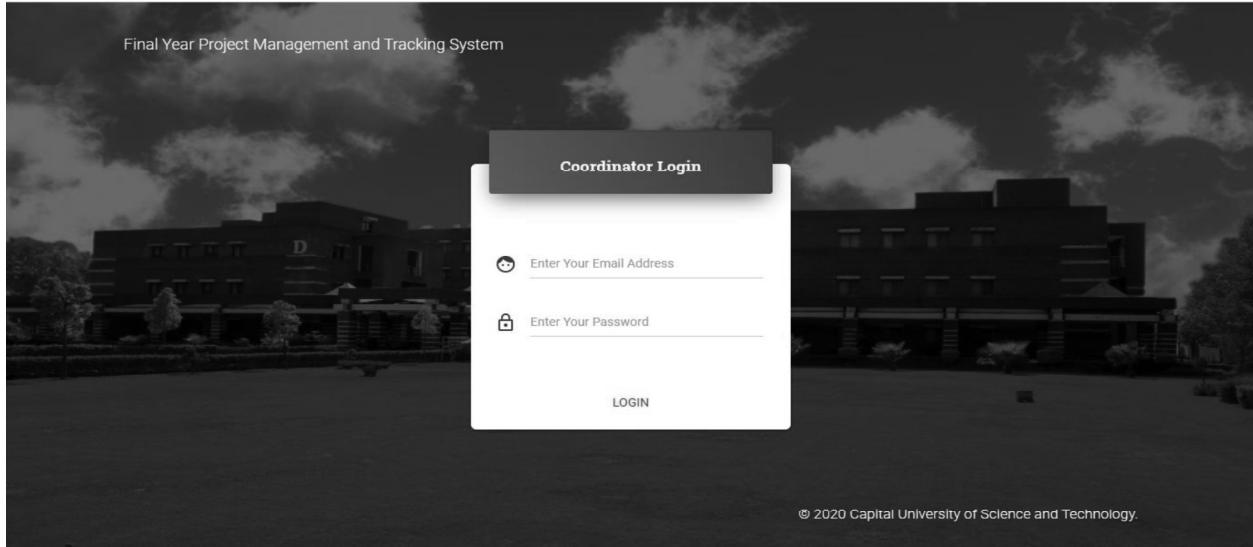


Figure 3.33 Coordinator Login

Supervisor can login to the application by providing email address and password as shown in Figure 3.34

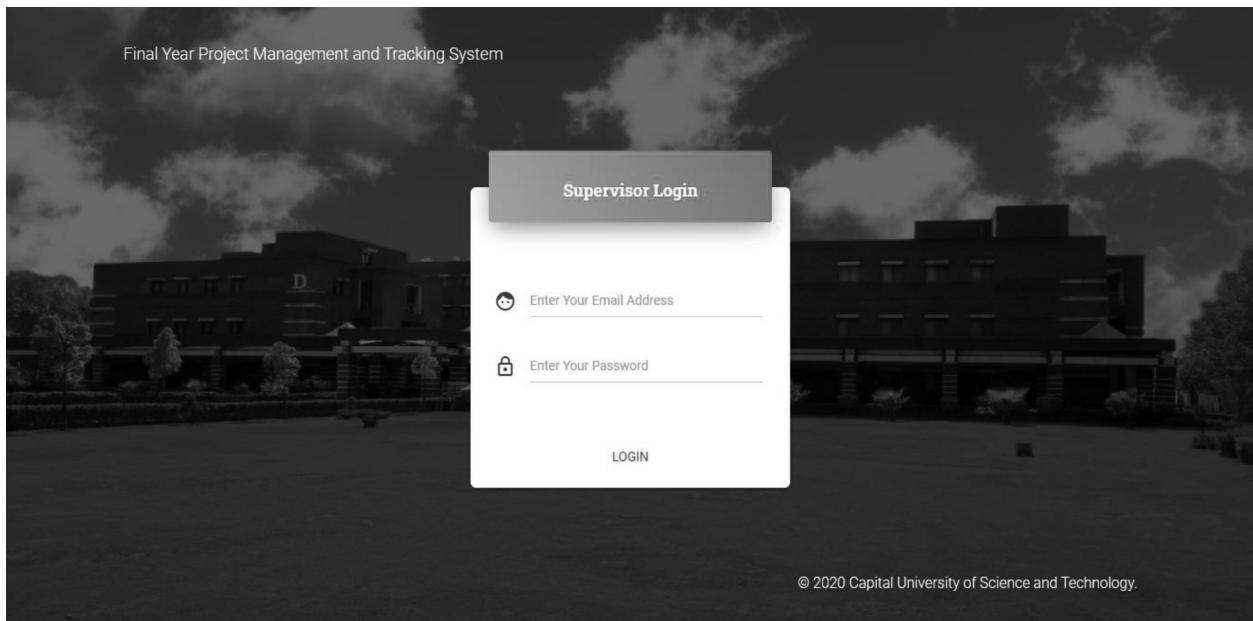


Figure 3.34 Supervisor Login

Student can login to the application by providing email address and password as shown in Figure 3.35

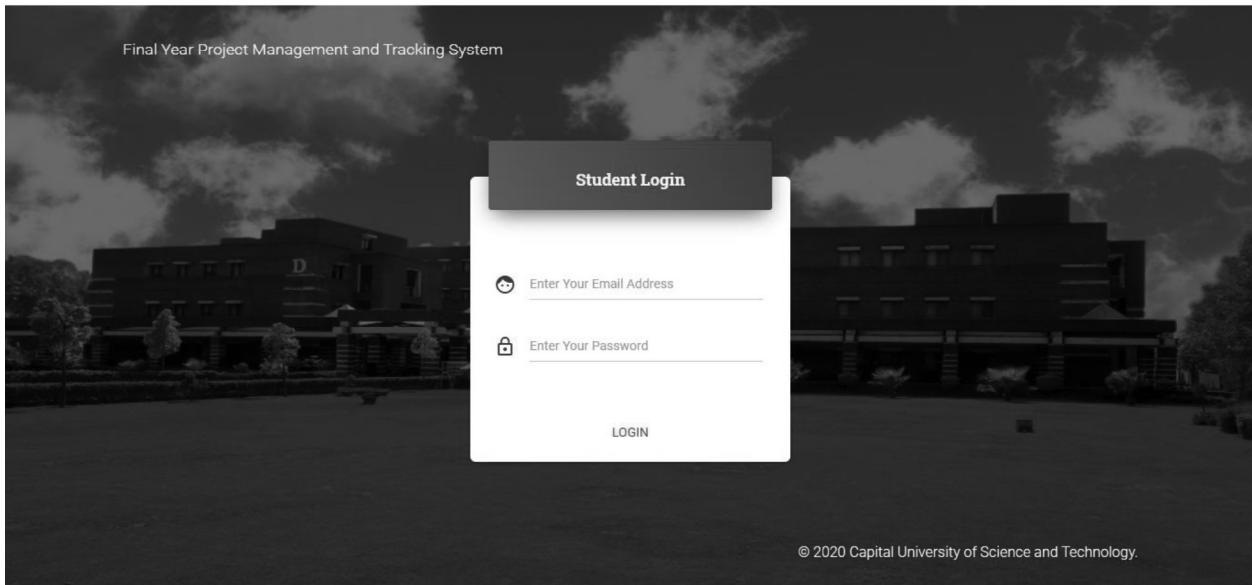


Figure 3.35 Student Login

Supervisors and Students are registered by the coordinator by providing email address, password and confirm password as shown in Figure 3.36

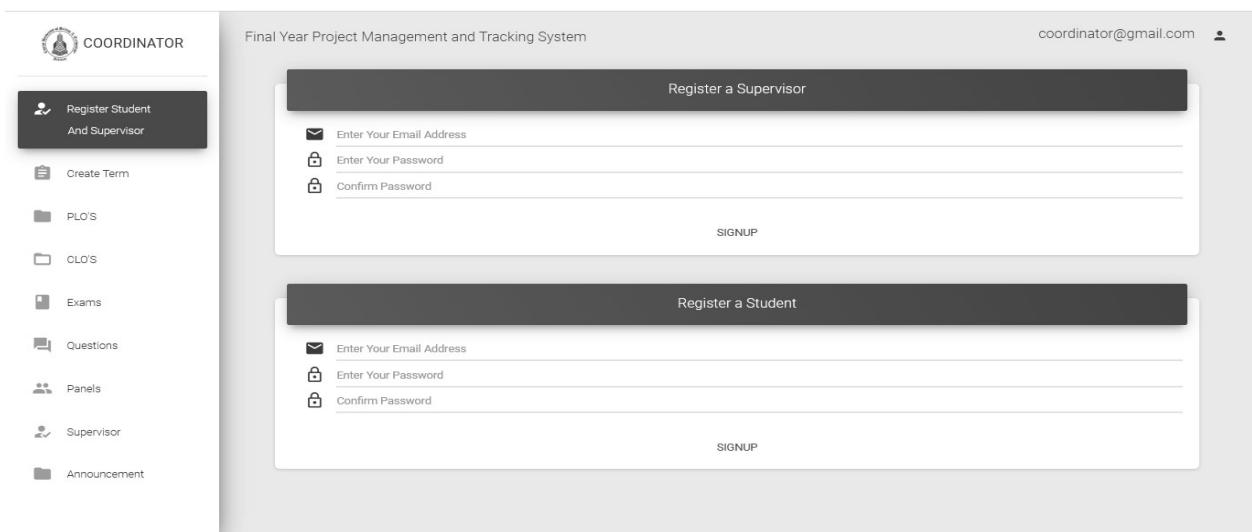


Figure 3.36 Register Supervisors and Students

Coordinator can create, view, edit and delete the term as shown in Figure 3.37

The screenshot shows the 'COORDINATOR' dashboard of the system. On the left sidebar, there are several menu items: Register Student And Supervisor, Create Term (which is highlighted in a dark grey box), PLO'S, CLO'S, Exams, Questions, Panels, Supervisor, and Announcement.

**Create New Term:** A modal window titled 'Create New Term' is open. It has fields for 'Term Name' (empty), 'Date From' (dd----2020), 'Date To' (dd----2020), and 'Working Term' (True/False). There is also a 'SAVE' button at the bottom right.

**Terms:** Below the modal, a table titled 'Terms' lists two entries:

Term Name	Date From	Date To	Working Term	Edit	Remove
193	2020-02-02	2020-08-02	False	<b>EDIT</b>	<b>REMOVE</b>
183	2020-02-02	2020-02-07	False	<b>EDIT</b>	<b>REMOVE</b>

Figure 3.37 Manage Term

Coordinator is adding, viewing, updating and deleting Program Learning Objectives on basis of terms selected as shown in Figure 3.38

The sidebar menu is identical to Figure 3.37, with 'PLO'S' being the active item.

**Select Term:** A dropdown menu shows the option '193'.

**ADD Program Learning Objectives:** A modal window titled 'ADD Program Learning Objectives' contains a text input field for 'Program Learning Objectives:' and a 'ADD' button.

**Program Learning Objectives:** Below the modal, a table lists three objectives:

Program Learning Objectives	Edit	Remove
Problem Analysis	<b>EDIT</b>	<b>REMOVE</b>
System Design	<b>EDIT</b>	<b>REMOVE</b>
Computing tool usage	<b>EDIT</b>	<b>REMOVE</b>

Figure 3.38 Manage Program Learning Objectives

Coordinator is adding, viewing; updating and deleting Course Learning Objectives on basis of Program Learning Objectives and terms selected as shown in Figure 3.39

Course Learning Objectives	Edit	Remove
	<b>EDIT</b>	<b>REMOVE</b>

Figure 3.39 Manage Course Learning Objectives

Coordinator is adding, viewing; updating and deleting exams on basis of terms selected as shown in Figure 3.40

Name	Weightage	Announced Date	Deadline	Edit	Remove
Proposal Defense	10	2020-02-02	2020-03-03	<b>EDIT</b>	<b>REMOVE</b>
Mid Exam 1	20	2020-03-04	2020-04-08	<b>EDIT</b>	<b>REMOVE</b>

Figure 3.40 Manage Exams

Coordinator is adding, viewing; updating and deleting questions on basis of terms, Course Learning Objectives, and exam selected as shown in Figure 3.41

Figure 3.41 Manage Questions

Coordinator is adding, viewing; updating and deleting supervisor's panel with term selected as shown in Figure 3.42

Figure 3.42 Manage Supervisor's Panel

Coordinator is adding, viewing; updating and deleting supervisors in panels on basis of term selected as shown in Figure 3.43

The screenshot shows the 'COORDINATOR' dashboard of the system. On the left sidebar, under the 'Supervisor' section, the 'Announcement' option is highlighted. The main content area displays the 'Add Supervisor' form with fields for Term (193), Panel (Panel No. 7), Name, Email, Slot, and Designation. Below this is a table titled 'List of Supervisors' showing two entries:

Name	Email	Slot	Designation	Edit	Remove
Dr. Shahid Iqbal Malik	shahidiqbalmalik@gmail.com	3	Professor	<button>EDIT</button>	<button>REMOVE</button>
Mr. Samir Obaid	samirobaid@gmail.com	3	Professor	<button>EDIT</button>	<button>REMOVE</button>

Figure 3.43 Manage Supervisors

Coordinator is adding, viewing; updating and deleting announcements on basis of term selected as shown in Figure 3.44

The screenshot shows the 'COORDINATOR' dashboard with the 'Announcement' option selected from the sidebar. The main content area shows a table titled 'Announcements' with two entries:

Announcement	Edit	Remove
Announcement01	<button>EDIT</button>	<button>REMOVE</button>
Announcement02	<button>EDIT</button>	<button>REMOVE</button>

Figure 3.44 Manage Announcements

Supervisors can view announcements of the working term as shown in Figure 3.45

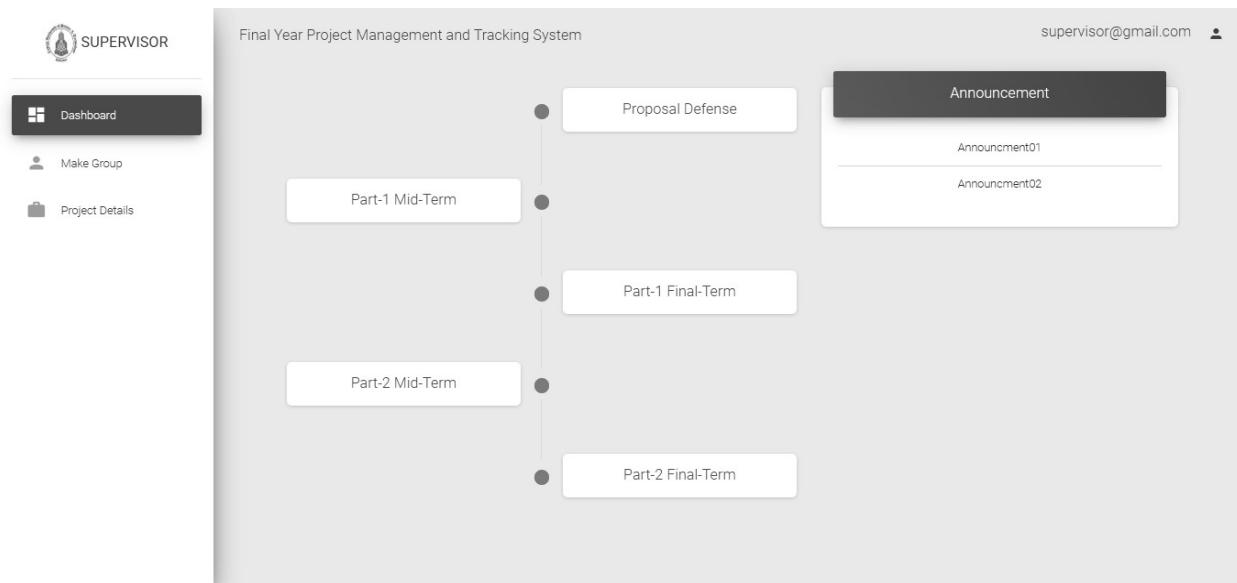


Figure 3.45 View Announcements by Supervisor

Students can view announcements of the working term as shown in Figure 3.46

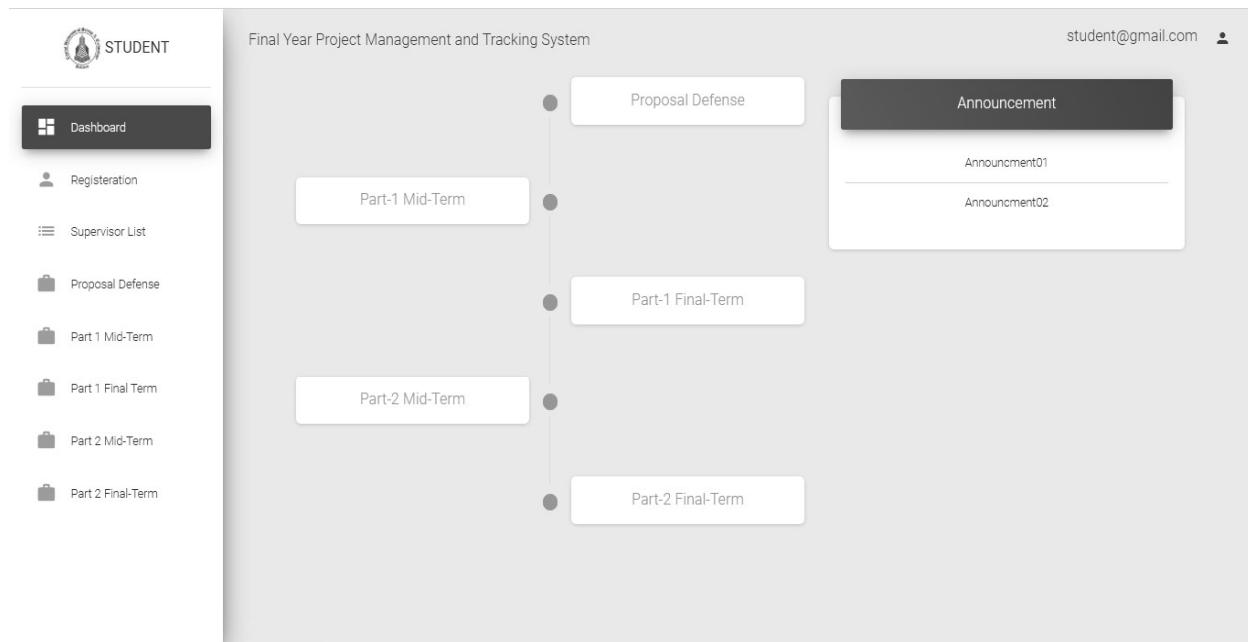


Figure 3.46 View Announcements by Student

Supervisor can create a group of students who are registered for final year project in a term as shown in Figure 3.47

The screenshot shows the 'Final Year Project Management and Tracking System' interface. On the left sidebar, under the 'SUPERVISOR' section, the 'Make Group' option is selected. The main content area has two tabs: 'Generate Group' and 'Assign Group Project Title'. The 'Generate Group' tab is active, showing a form with 'Group Title' (Enter Group Title) and three student names: 'bcs365 - Tahir', 'bcs163086 - rafy', and 'bcs163098 - osamak'. A 'SUBMIT' button is at the bottom. The 'Assign Group Project Title' tab shows a form with 'Please Select a Group' and 'Enter Project Title', with a 'SUBMIT' button. Below these tabs is a 'Projects' section table:

Project Name	Member1	Member2	Member3	Group	Action
Final Year Project	Muhammad Haroon Qureshi	Hina Mahmood Qureshi	Syed Qamar Abbas	Group 1	<button>SEND FOR APPROVAL</button>
Alumi	Saira Tanveer	Tahira	HinaZ	Group 2	<button>SEND FOR APPROVAL</button>

Figure 3.47 Create Group

Supervisor assigns the project title to the group created by him/her in a term as shown in Figure 3.48

The screenshot shows the 'Final Year Project Management and Tracking System' interface. The 'Assign Group Project Title' tab is active, showing a form with 'Group 3' and 'Enter Project Title', with a 'SUBMIT' button. Below this is a 'Projects' section table:

Project Name	Member1	Member2	Member3	Group	Action
Final Year Project	Muhammad Haroon Qureshi	Hina Mahmood Qureshi	Syed Qamar Abbas	Group 1	<button>SEND FOR APPROVAL</button>
Alumi	Saira Tanveer	Tahira	HinaZ	Group 2	<button>SEND FOR APPROVAL</button>

Figure 3.48 Assign Project Title

Supervisor can send an email to the coordinator about creation of a project as shown in Figure 3.49

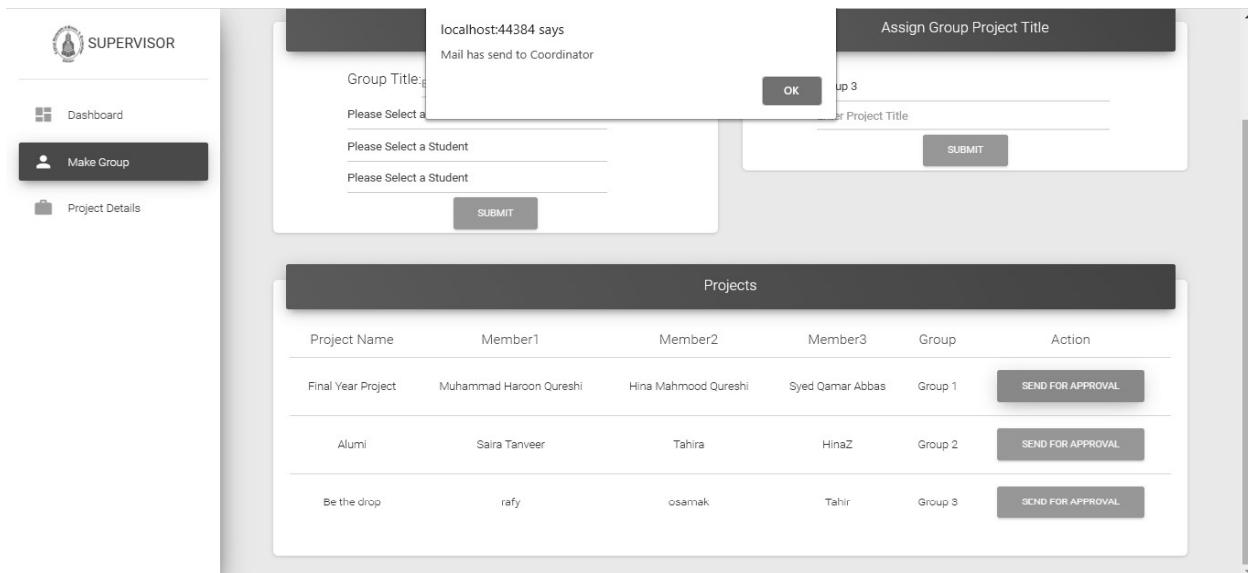


Figure 3.49 Send Email by Supervisor

Supervisor can view the projects details (project title, panel Number, Term name, group name, and supervisor) and panel details (panel member's name, designation and email id) of the project of selected term as shown in Figure 3.50

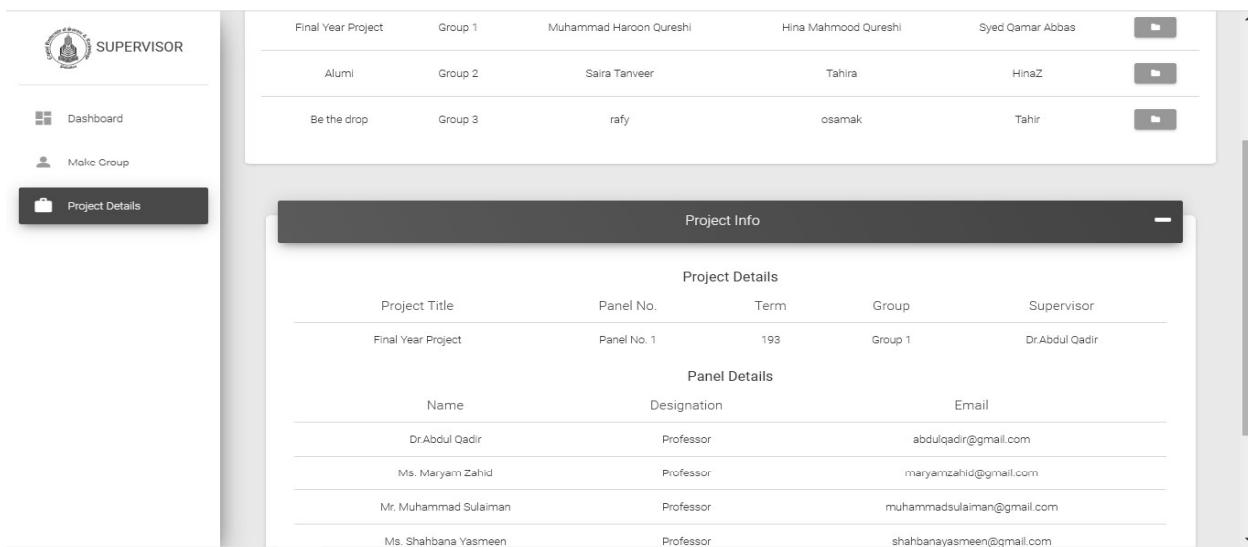


Figure 3.50 View Project and Panel Details

Supervisors can view the questions to each exam with course learning objectives with term working as shown in Figure 3.51

The screenshot shows a supervisor's dashboard with a sidebar on the left containing icons for Dashboard, Make Group, and Project Details (which is currently selected). The main area displays a list of questions under 'Project Info' for 'Final Year Project Part 1'. The questions are:

- Question1: Stakeholder analysis (background study) and Domain Knowledge CLO:To effectively communicate with stakeholders to identify the scope of the project and present as a final year project
- Question2: Project Scope w.r.t team members CLO:To effectively communicate with stakeholders to identify the scope of the project and present as a final year project
- Question3: Selection of tools, frameworks and other technologies CLO:To effectively communicate with stakeholders to identify the scope of the project and present as a final year project
- Question4: Project Plan and Work Breakdown Structure CLO:To effectively communicate with stakeholders to identify the scope of the project and present as a final year project
- Question5: Communication Skills CLO:To effectively communicate with stakeholders to identify the scope of the project and present as a final year project

At the bottom right, there is a button labeled 'Mid 1 +'

Figure 3.51 View Questions by Supervisor

Students can view the questions to each exam with course learning objectives with term working as shown in Figure 3.52

The screenshot shows a student's dashboard with a sidebar on the left containing icons for Dashboard, Registration, Supervisor List, and Proposal Defense (which is currently selected). The main area has two sections: 'Upload Presentation' and 'Upload Document'. Below these sections is a heading 'Question For Evaluation' followed by the same five questions listed in Figure 3.51.

Figure 3.52 View Questions by Student

Student provides valid name, registration number, date of birth, phone number, email address, credit hours enrolled, credit hour completed, and cumulative GPA with term selected for registration of final year project as shown in Figure 3.53

Figure 3.53 Register Student

Students can view the supervisor's list involving their name, slots available and email id's of working term as shown in Figure 3.54

S No.	Supervisor Name	Email	Slot Available
1	Dr Abdul Qadir	abdulqadir@gmail.com	3
2	Ms. Maryam Zahid	maryamzahid@gmail.com	3
3	Mr. Muhammad Sulaiman	muhammadsulaiman@gmail.com	3
4	Ms. Shahbana Yasmeen	shahbanayasmeen@gmail.com	3
5	Dr. Nayyer Masood	nayyermasood@gmail.com	3
6	Mr. Salman Ahmed	salmanahmed@gmail.com	3
7	Mr. Shehzad Qaiser	shehzadqaiser@gmail.com	3
8	Mr. Hasnat Ali	hasnatali@gmail.com	3

Figure 3.54 View Supervisor's List

Student can send presentation and document to supervisor and project panels by sending email as shown in Figure 3.55

The screenshot shows a web-based application interface for a student. On the left, there is a sidebar with a logo and the text "STUDENT". Below the logo are several menu items: "Dashboard", "Registration", "Supervisor List", "Proposal Defense" (which is highlighted with a dark background), "Part 1 Mid-Term", "Part 1 Final Term", "Part 2 Mid-Term", and "Part 2 Final-Term". The main content area has a header "Final Year Project Management and Tracking System" and a sub-header "student@gmail.com". The main content is titled "Proposal Defense" and contains two sections: "Upload Presentation" and "Upload Document". The "Upload Presentation" section includes fields for "From:", "To:", and "Password:", and a file upload button labeled "Choose File" with the message "No file chosen" and an "UPLOAD" button. The "Upload Document" section includes a file upload button labeled "Choose File" with the message "No file chosen" and an "UPLOAD" button. Below these sections, there is a heading "Question For Evaluation" followed by three questions:

- Question1: Stakeholder analysis (background study) and Domain Knowledge CLO:To effectively communicate with stakeholders to identify the scope of the project and present as a final year project
- Question2: Project Scope w.r.t team members CLO:To effectively communicate with stakeholders to identify the scope of the project and present as a final year project
- Question3: Selection of tools, frameworks and other technologies CLO:To effectively communicate with stakeholders to

Figure 3.55 Send Email by Student

## **Chapter 4**

# **Software Development**

### **4.1. Coding Standards**

This chapter will provide the details about the coding standard, we adopted during implementation phase.

The declaration in our project is meaningful that each function represents a task i.e. GetTermById(), DeleteTermById(), UpdateTerm() etc. The naming conventions we have used for the project is Camel case. Each line contains at most one statement.

### **4.2. Development Environment**

The development environment we have used is Visual Studio 2019. We have used this software because it is focused and productive with improved performance, instant code cleanup, and better search results.

We have used MYSQL database because it provides complete workflow control, high performance and flexibility of open source. MYSQL database is storing coordinator, supervisors, students and projects information.

### **4.3. Software Description**

Working of our project is:

- Coordinator signs up into the system
- Coordinator Login to the system
- Supervisor Login to the system
- Student Login to the system
- Coordinator registers supervisors and students
- Coordinator managing the terms
- Coordinator managing the Program Learning Objectives
- Coordinator managing the Course Learning Objectives
- Coordinator managing the exams
- Coordinator managing the questions
- Coordinator managing the supervisor's panel
- Coordinator managing the supervisors

- Coordinator managing the announcements
- Supervisors viewing the announcements
- Students viewing the announcements
- Supervisors creating the group
- Supervisors assigning the project title
- Supervisor sending email to the coordinator
- Supervisor viewing project and panel details
- Supervisor viewing the questions
- Students viewing the questions
- Students registering himself/herself
- Students viewing the supervisor's list
- Student sending email to the supervisors and panel members

### **Coordinator signs up into the system:**

Coordinator can sign up into the application by providing email address; password and confirm password

```
$("#signup").click(function () {
    var user = {
        grant_type: 'password',
        Email: $("#Email").val(),
        password: $("#password").val(),
        confirmPassword: $("#confirmPassword").val()
    };
    $.ajax({
        type: 'POST',
        url: "/api/Account/RegisterCoordinator",
        data: user,
        contentType: "application/x-www-form-urlencoded",
        dataType: "text",
        success: function (data) {
            alert("You are Registered. Please Login!");
            window.location.href = "coordinator.html";
        },
        error: function (response) {
```

```

        alert("Invalid Email/Password");
    }
});
});

```

### **Coordinator Login to the system:**

Coordinator can login to the application by providing email address and password

```

$("#login").click(function () {
    var user = {
        grant_type: 'password',
        username: $("#email").val(),
        password: $("#password").val()
    };
    $.ajax({
        type: "POST",
        url: "/token",
        data: user,
        contentType: "application/x-www-form-urlencoded",
        dataType: "json",
        success: function (data) {
            bearer = data.access_token;
            localStorage.setItem('token', bearer);
            window.location.href = "studentandsupervisor.html";
        },
        error: function (response) {
            alert("Invalid Email/Password");
        }
    });
}
);

```

### **Supervisor Login to the system:**

Supervisor can login to the application by providing email address and password

```

$("#login").click(function () {
    var user = {
        grant_type: 'password',

```

```

        username: $("#email").val(),
        password: $("#password").val()
    };
    $.ajax({
        type: "POST",
        url: "/token",
        data: user,
        contentType: "application/x-www-form-urlencoded",
        dataType: "json",
        success: function (data) {
            bearer = data.access_token;
            localStorage.setItem('token', bearer);
            window.location.href = "supervisor_dashboard.html";
        },
        error: function (response) {
            alert("Invalid Email/Password");
        }
    });
});

```

### **Student Login to the system:**

Student can login to the application by providing email address and password

```

$("#login").click(function () {
    var user = {
        grant_type: 'password',
        username: $("#email").val(),
        password: $("#password").val()
    };
    $.ajax({
        type: "POST",
        url: "/token",
        data: user,
        contentType: "application/x-www-form-urlencoded",
        dataType: "json",
        success: function (data) {

```

```

        bearer = data.access_token;
        localStorage.setItem('token', bearer);
        window.location.href = "student_dashboard.html";
    },
    error: function (response) {
        alert("Invalid Email/Password");
    }
});
});

```

### **Coordinator registers supervisors and students:**

Supervisors and Students are registered by the coordinator by providing email address, password and confirm password

```

//student register
$("#signups").click(function () {
    var user = {
        grant_type: 'password',
        Email: $("#Emails").val(),
        password: $("#passwords").val(),
        confirmPassword: $("#confirmPasswords").val()
    };
    $.ajax({
        type: 'POST',
        url: "/api/Account/RegisterStudent",
        data: user,
        contentType: "application/x-www-form-urlencoded",
        dataType: "text",
        success: function (data) {
            alert("Student has been Registered");
            location.reload(true);
        },
        error: function (response) {
            alert("Provide the Credentials");
        }
    });
});

```

```

});
//supervisor register
$("#signup").click(function () {
    var user = {
        grant_type: 'password',
        Email: $("#Email").val(),
        password: $("#password").val(),
        confirmPassword: $("#confirmPassword").val()
    };
    $.ajax({
        type: 'POST',
        url: "/api/Account/RegisterSupervisor",
        data: user,
        contentType: "application/x-www-form-urlencoded",
        dataType: "text",
        success: function (data) {
            alert("Supervisor has been Registered");
            location.reload(true);
        },
        error: function (response) {
            alert("Provide the Credentials");
        }
    });
});

```

### **Coordinator managing the terms:**

Coordinator can create, view, edit and delete the term

```

$("#buttonSave").click(function () {
    var Term = {
        Title: $("#textName").val(),
        DateFrom: $("#textDateFrom").val(),
        DateTo: $("#textDateTo").val(),
        Status: $("#isWorkingTerm").val(),
        IsActive: ("True")
    };

```

```

$.ajax({
    url: "/api/Terms",
    data: JSON.stringify(Term),
    type: "POST",
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    headers: {
        'Content-Type': 'application/json; charset-utf-8',
        'Accept': 'application/json',
        'Authorization': 'bearer ' + localStorage.getItem('token')
    },
    success: function (result) {
        LoadTerms();
    },
    error: function (errormessage) {
        alert(errormessage.responseText);
    }
});
});

```

### **Coordinator managing the Program Learning Objectives:**

Coordinator is adding, viewing, updating and deleting Program Learning Objectives on basis of terms selected

```

$("#PLOADD").click(function () {
    var PLO = {
        Termld: $("#DropDownItem option:selected").val(),
        Title: $("#PLO").val(),
        IsActive: ("True")
    };
    $.ajax({
        url: "/api/PLOes",
        data: JSON.stringify(PLO),
        type: "POST",
        contentType: "application/json; charset=utf-8",
        dataType: "json",

```

```

headers: {
    'Content-Type': 'application/json; charset-utf-8',
    'Accept': 'application/json',
    'Authorization': 'bearer ' + localStorage.getItem('token')
},
success: function (result) {
    loadPLOs($("#DropDownItem option:selected").val());
},
error: function (errormessage) {
    alert(errormessage.responseText);
}
});
});

```

### **Coordinator managing the Course Learning Objectives:**

Coordinator is adding, viewing; updating and deleting Course Learning Objectives on basis of Program Learning Objectives and terms selected

```

//Adds CLOs on button click
$("#CLOADD").click(function () {
var CLO = {
    Termld: $("#DropDownItem option:selected").val(),
    PLOId : $("#DropDownPLO option:selected").val(),
    Title: $("#CLO").val(),
    IsActive:("True")
};
$.ajax({
    url: "/api/CLOes",
    data: JSON.stringify(CLO),
    type: "POST",
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    headers: {
        'Content-Type': 'application/json; charset-utf-8',
        'Accept': 'application/json',
        'Authorization': 'bearer ' + localStorage.getItem('token')
    }
});
});

```

```

        },
        success: function (result) {
            DisplayCLOs( $("#DropDownPLO option:selected").val());
        },
        error: function (errormessage) {
            alert(errormessage.responseText);
        }
    });
});

```

### **Coordinator managing the exams:**

Coordinator is adding, viewing; updating and deleting exams on basis of terms selected

```

//Adds Exams on button click
$("#EXAMADD").click(function () {
    var exam = {
        TermId: $("#DropDownItem option:selected").val(),
        Title: $("#textTitle").val(),
        Weightage: $("#textWeightage").val(),
        Deadline: $("#textDeadLine").val(),
        AnnouncedDate: $("#textAnnouncedDate").val(),
        IsActive:("True")
    };
    $.ajax({
        url: "/api/Exams",
        data: JSON.stringify(exam),
        type: "POST",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset-utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {

```

```

        loadExams($("#DropDownItem option:selected").val());
    },
    error: function (errormessage) {
        alert(errormessage.responseText);
    }
});
});

```

### **Coordinator managing the questions:**

Coordinator is adding, viewing; updating and deleting questions on basis of terms, Course Learning Objectives, and exam selected

```

$("#QUESTIONADD").click(function () {
    var question = {
        TermId: $("#DropDownItem option:selected").val(),
        CLOId: $("#displayclo option:selected").val(),
        ExamId: $("#DropDownexam option:selected").val(),
        Title: $("#Question").val(),
        IsActive:("True")
    };
    $.ajax({
        url: "/api/Questions",
        data: JSON.stringify(question),
        type: "POST",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset=utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {
            DisplayQuestions( $("#DropDownexam option:selected").val());
        },
        error: function (errormessage) {
            alert(errormessage.responseText);
        }
    });
});

```

```
        }
    });
});
```

### **Coordinator managing the supervisor's panel:**

Coordinator is adding, viewing; updating and deleting supervisor's panel with term selected

```
//Adds panel on button click
$("#panelADD").click(function () {
    var panel = {
        TermId: $("#DropDownItem option:selected").val(),
        Name: $("#panel").val(),
        IsActive: ("True")
    };
    $.ajax({
        url: "/api/Panels",
        data: JSON.stringify(panel),
        type: "POST",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset-utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {
            loadpanel($("#DropDownItem option:selected").val());
        },
        error: function (errormessage) {
            alert(errormessage.responseText);
        }
    });
});
```

### **Coordinator managing the supervisors:**

Coordinator is adding, viewing; updating and deleting supervisors in panels on basis of term selected

```
//Adds supervsiors on button click
$("#Supervisoradd").click(function () {
    var SUPERVISOR = {
        TermId: $("#DropDownItem option:selected").val(),
        Name: $("#name").val(),
        PanelId: $("#DropDownpanel option:selected").val(),
        Designation: $("#designation").val(),
        Email: $("#email").val(),
        Slot: $("#slot").val(),
        IsActive:("True")
    };
    $.ajax({
        url: "/api/Instructors",
        data: JSON.stringify(SUPERVISOR),
        type: "POST",
        contentType: "application/json;charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset-utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {
            loadInstructors($("#DropDownpanel option:selected").val());
        },
        error: function (errormessage) {
            alert(errormessage.responseText);
        }
    });
});
```

### **Coordinator managing the announcements:**

Coordinator is adding, viewing; updating and deleting announcements on basis of term selected

```
//Adds announcement on button click
$("#announcementADD").click(function () {
    var announcement = {
        TermId: $("#DropDownItem option:selected").val(),
        Title: $("#announcement").val(),
        IsActive_: ("True")
    };
    $.ajax({
        url: "/api/Announcements",
        data: JSON.stringify(announcement),
        type: "POST",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset-utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {
            loadannouncement($("#DropDownItem option:selected").val());
        },
        error: function (errormessage) {
            alert(errormessage.responseText);
        }
    });
});
```

### **Supervisors viewing the announcements:**

Supervisors can view announcements of the working term

```
//Loads announcement on page load
$.ajax({
    url: "/api/Announcements",
```

```

        type: "GET",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset-utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {
            LoadAnnouncements();
        },
        error: function (errormessage) {
            alert(errormessage.responseText);
        }
    });
});

//Load Announcements
function LoadAnnouncements() {
    $.ajax({
        url: "/api/Announcements",
        type: "GET",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset-utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {
            var s = "";
            for (var i = 0; i < result.length; i++) {
                s += '<tr>';
                s += '<td id="Title">' + result[i].Title + '</td >';
                s += '</tr>';
            }
            $("#Announcementdisplay").html(s);
        }
    });
}

```

```

},
error: function (errormessage) {
    alert(errormessage.responseText);
}
});

```

### **Students viewing the announcements:**

Students can view announcements of the working term

```

//Loads announcement on page load
$.ajax({
    url: "/api/Announcements",
    type: "GET",
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    headers: {
        'Content-Type': 'application/json; charset-utf-8',
        'Accept': 'application/json',
        'Authorization': 'bearer ' + localStorage.getItem('token')
    },
    success: function (result) {
        LoadAnnouncements();
    },
    error: function (errormessage) {
        alert(errormessage.responseText);
    }
});
//Load Announcements
function LoadAnnouncements() {
    $.ajax({
        url: "/api/Announcements",
        type: "GET",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {

```

```

'Content-Type': 'application/json; charset-utf-8',
'Accept': 'application/json',
'Authorization': 'bearer ' + localStorage.getItem('token')
},
success: function (result) {
    var s = "";
    for (var i = 0; i < result.length; i++) {
        s += '<tr>';
        s += '<td id="Title">' + result[i].Title + '</td >';
        s += '</tr>';
    }
    $("#Announcementdisplay").html(s);
},
error: function (errormessage) {
    alert(errormessage.responseText);
}
});
}

```

### **Supervisors creating the group:**

Supervisor can create a group of students who are registered for final year project in a term

```

var group = {
    TermId: $("#DropDownItem option:selected").val(),
    StudentId1: $("#DropDownstudent").val(),
    StudentId2: $("#DropDownstudent1").val(),
    StudentId3: $("#DropDownstudent2").val(),
    gName: $("#groupTitle").val(),
    PanelId: s,
    IsActive: ("True")
};
$.ajax({
    url: "/api/Groups",
    data: JSON.stringify(group),
    type: "POST",

```

```

        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset-utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {
            location.reload(true);
        },
        error: function (errormessage) {
            alert("Make Group First");
        }
    });
}

```

### **Supervisors assigning the project title:**

Supervisor assigns the project title to the group created by him/her in a term

```

//Adds project of to database on button click
$("#makeproject").click(function () {
    var project = {
        TermId: $("#DropDownItem option:selected").val(),
        Title: $("#projecttitle").val(),
        GroupId: $("#DropDowngroup option:selected").val(),
        IsActive: ("True")
    };
    $.ajax({
        url: "/api/Projects",
        data: JSON.stringify(project),
        type: "POST",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset-utf-8',
            'Accept': 'application/json',

```

```

        'Authorization': 'bearer ' + localStorage.getItem('token')
    },
    success: function (result) {
        loadProjects($("#DropDownItem option:selected").val());
        setinstproj(result.id);
    },
    error: function (errormessage) {
        alert("Assign Project Title");
    }
});
});

```

### **Supervisor sending email to the coordinator:**

Supervisor can send an email to the coordinator about creation of a project

```

function sendEmail() {
    $("#send").click(function () {
        var user = {
            To: $("#email").val()
        };
        $.ajax({
            type: "POST",
            url: "/Email",
            data: user,
            contentType: "application/x-www-form-urlencoded",
            dataType: "json",
            success: function (data) {
                alert("Mail has send to Coordinator");
            }
        });
        alert("Mail has send to Coordinator");
    });
}

```

### **Supervisor viewing project and panel details:**

Supervisor can view the projects details (project title, panel Number, Term name, group name, and supervisor) and panel details (panel member's name, designation and email id) of the project of selected term

```
//load project detail action click
function loadprojectdetails(ProjectId) {
    $.ajax({
        url: "/api/Projects/" + ProjectId + "/ProjectId",
        type: "GET",
        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset=utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {
            var projectss = "";
            for (var i = 0; i < result.length; i++) {
                projectss += '<tr>';
                projectss += '<td id="title">' + result[i].Title + '</td >';
                projectss += '<td id="panelno">' + result[i].panelname + '</td >';
                projectss += '<td id="term">' + result[i].termtitle + '</td >';
                projectss += '<td id="group">' + result[i].gName + '</td >';
                projectss += '<td id="supervisor">' + result[i].Name + '</td >';
                projectss += '</tr>';
            }
            $('#projectdetaildisplay').html(projectss);
        },
        error: function (errormessage) {
            alert(errormessage.responseText);
        }
    });
}

//load panel details on action click
function loadpaneldetails(ProjectId) {
```

```

$.ajax({
    url: "/api/Instructors/" + ProjectId + "/ProjectId",
    type: "GET",
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    headers: {
        'Content-Type': 'application/json; charset-utf-8',
        'Accept': 'application/json',
        'Authorization': 'bearer ' + localStorage.getItem('token')
    },
    success: function (result) {
        var projectss = "";
        for (var i = 0; i < result.length; i++) {
            projectss += '<tr>';
            projectss += '<td id="panelname">' + result[i].Name + '</td >';
            projectss += '<td id="design">' + result[i].Designation + '</td >';
            projectss += '<td id="email">' + result[i].Email + '</td >';
            projectss += '</tr>';
        }
        $('#paneldetaildisplay').html(projectss);
    },
    error: function (errormessage) {
        alert(errormessage.responseText);
    }
});
}

```

### **Supervisor viewing the questions:**

Supervisors can view the questions to each exam with course learning objectives with term working

```

//load questions details on action click
function loadproposaldetails1(ProjectId) {
    $.ajax({
        url: "/api/Questions/" + ProjectId + "/ProjectId",
        type: "GET",

```

```

        contentType: "application/json; charset=utf-8",
        dataType: "json",
        headers: {
            'Content-Type': 'application/json; charset-utf-8',
            'Accept': 'application/json',
            'Authorization': 'bearer ' + localStorage.getItem('token')
        },
        success: function (result) {
            var projectss = "";
            var j = 1;
            for (var i = 0; i < result.length; i++) {
                projectss += '<div class="row" style="padding-left:13px;padding-right:13px;text-align:justify">';
                projectss += '<h4 id="qid" hidden>' + result[i].Id + '</h4>';
                projectss += '<h5 id="qtitle"><strong>Question' + j + ':</strong> ' +
                result[i].Title + '<strong> CLO:</strong>' + result[i].clotitle + '</h5 >';
                projectss += '</div>';
                j++;
            }
            $('#proposalquestiondetail').html(projectss);
        },
        error: function (errormessage) {
            alert(errormessage.responseText);
        }
    });
}

```

### **Students viewing the questions:**

Students can view the questions to each exam with course learning objectives with term working

```

$.ajax({
    url: "/api/Questions/" + 1 + "/ProjectId",
    type: "GET",
    contentType: "application/json; charset=utf-8",
    dataType: "json",

```

```

headers: {
    'Content-Type': 'application/json; charset-utf-8',
    'Accept': 'application/json',
    'Authorization': 'bearer ' + localStorage.getItem('token')
},
success: function (result) {
    var projectss = "";
    var j = 1;
    for (var i = 0; i < result.length; i++) {
        projectss += '<div class="row" style="padding-left:13px;padding-right:13px;text-align:justify">';
        projectss += '<h4 id="qid" hidden>' + result[i].Id + '</h4>';
        projectss += '<h5 id="qtitle"><strong>Question' + j + ':</strong> ' +
        result[i].Title + '<strong> CLO:</strong>' + result[i].clotitle + '</h5 >';
        projectss += '</div>';
        j++;
    }
    $('#proposalquestiondetail').html(projectss);
},
error: function (errormessage) {
    alert(errormessage.responseText);
}
});

```

### **Students registering himself/herself:**

Student provides valid name, registration number, date of birth, phone number, email address, credit hours enrolled, credit hour completed, and cumulative GPA with term selected for registration of final year project.

```

// student data saved in database on button click
$("#submit").click(function () {
    var Student = {
        TermId: $("#DropDownItem option:selected").val(),
        Name: $("#name").val(),
        RegNo: $("#regno").val(),
        DOB: $("#dob").val(),

```

```

    PhoneNo: $("#phone").val(),
    Email: $("#email").val(),
    CreditHourEnrolled: $("#che").val(),
    CreditHourCompleted: $("#chc").val(),
    Cgpa: $("#cgpa").val()
};

$.ajax({
    url: "/api/Students",
    data: JSON.stringify(Student),
    type: "POST",
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    headers: {
        'Content-Type': 'application/json; charset-utf-8',
        'Accept': 'application/json',
        'Authorization': 'bearer ' + localStorage.getItem('token')
    },
    success: function (result) {
        alert("You are Registered sucessfully");
        location.reload(true);
    },
    error: function (errormessage) {
        alert("Fill out the form first");
    }
});
});

```

### **Students viewing the supervisor's list:**

Students can view the supervisor's list involving their name, slots available and email id's of working term

```

$.ajax({
    url: "/api/Instructors",
    type: "GET",
    contentType: "application/json; charset=utf-8",
    dataType: "json",

```

```

headers: {
    'Content-Type': 'application/json; charset=utf-8',
    'Accept': 'application/json',
    'Authorization': 'bearer ' + localStorage.getItem('token')
},
success: function (result) {
    var s = "";
    var j = 1;
    for (var i = 0; i < result.length; i++) {
        s += '<tr>';
        s += '<td>' + j + '</td>';
        s += '<td id="name">' + result[i].Name + '</td >';
        s += '<td id="email">' + result[i].Email + '</td>';
        s += '<td id="slot">' + result[i].Slot + '</td >';
        s += '</tr>';
        j++;
    }
    $("#listdisplay").html(s);
},
error: function (errormessage) {
    alert(errormessage.responseText);
}
});
});

```

### **Student sending email to the supervisors and panel members:**

Student can send presentation and document to supervisor and project panels by sending email

```

$("#uploadp").click(function () {
    var formData = new FormData();
    var file = document.getElementById("presentation").files[0];
    formData.append("fileUploader", file);
    var data = {
        From: $("#from").val(),
        To: $("#to").val(),

```

```

        Password: $("#password").val(),
        Subject: $("#subject").val(),
        Body:( "Below is the Attachement File")
    };
    $.ajax({
        type: "POST",
        url: "/SendMailer",
        data: "objModelMail=" + data + "fileUploader=" + formData,
        data: "{objModelMail:" + data + ",fileUploader:" + formData + "}",
        data: "objModelMail=" + JSON.stringify(data) + "&fileUploader =" +
    formData,
        async: false,
        //data: JSON.stringify(parameter),
        contentType: false,
        processData: false,
        dataType: "json",
        success: function (data) {
            alert("Mail has sent");
        }
    });
});

```

# Chapter 5

## Software Testing

This chapter provides a description about the adopted testing procedure. This includes the selected testing methodology, test suite and the test results of the developed software.

### 5.1. Testing Methodology

We have used manual testing methodology in our testing phase. We give different inputs to our system to test the software and examine the functionality of the application.

### 5.2. Test Cases

#### Test Case 1: Sign Up Coordinator

**Test:** Sign up into the application by providing email address; password and confirm password

Table 5.1 Sign up Coordinator with matching password and Confirm Password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Sign Up with matching password and confirm Password	Test ID:1
Version:1	Test Type: System
<i>Input:</i> Email Id: <u>hinaqureshi051@gmail.com</u> Password: <u>Hina123@#</u> Confirm Password: <u>Hina123@#</u>	
Expected Result: Coordinator will be signed up into the application	
Actual Result: passed	

#### Test Case 2: Sign Up Coordinator

**Test:** Sign up into the application by providing email address; and wrong password and confirm password

Table 5.2 Sign up Coordinator with not matching password and Confirm Password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Sign Up with not matching password and confirm Password	Test ID:2
Version:1.1	Test Type: System
<i>Input:</i>	
Email Id: <u>hinaqureshi051@gmail.com</u>	
Password: Hina123@#	
Confirm Password: Hina123@	
<i>Expected Result:</i> Coordinator will not be signed up into the application	
<i>Actual Result:</i> passed	

### Test Case 3: Login Coordinator

**Test:** Login into the application by providing email address and password

Table 5.3 Login Coordinator with email address and password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Login with email address and password	Test ID:3
Version:1	Test Type: System
<i>Input:</i>	
Email Id: <u>hinaqureshi051@gmail.com</u>	
Password: Hina123@#	
<i>Expected Result:</i> Coordinator will be logged in to the application	
<i>Actual Result:</i> passed	

### Test Case 4: Login Coordinator

**Test:** Login into the application by providing wrong email address and password

Table 5.4 Login Coordinator with wrong email address and password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	

<i>Objective:</i> Login with wrong email address and password	<i>Test ID:</i> 4
<i>Version:</i> 1.1	<i>Test Type:</i> System
<i>Input:</i>	
Email Id: <u>hinaqureshi051@gmail.com</u>	
Password: Hina123@	
<i>Expected Result:</i> Coordinator will not be logged in to the application	
<i>Actual Result:</i> passed	

### Test Case 5: Login Supervisor

**Test:** Login into the application by providing email address and password

Table 5.5 Login Supervisor with email address and password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
<i>Objective:</i> Login with email address and password	<i>Test ID:</i> 5
<i>Version:</i> 1	<i>Test Type:</i> System
<i>Input:</i>	
Email Id: <u>samiroabaid@gmail.com</u>	
Password: Samir123@#	
<i>Expected Result:</i> Supervisor will be logged in to the application	
<i>Actual Result:</i> passed	

### Test Case 6: Login Supervisor

**Test:** Login into the application by providing wrong email address and password

Table 5.6 Login Supervisor with wrong email address and password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
<i>Objective:</i> Login with wrong email address and password	<i>Test ID:</i> 6
<i>Version:</i> 1.1	<i>Test Type:</i> System

<i>Input:</i> Email Id: <u>samirobaid@gmail.com</u> Password: Samir123@	
<i>Expected Result:</i> Supervisor will not be logged in to the application	
<i>Actual Result:</i> passed	

### Test Case 7: Login Student

**Test:** Login into the application by providing email address and password

Table 5.7 Login Student with email address and password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Login with email address and password	<i>Test ID:</i> 7
Version:1	<i>Test Type:</i> System
<i>Input:</i> Email Id: <u>haroonqureshi60@gmail.com</u> Password: Haroon123@#	
<i>Expected Result:</i> Student will be logged in to the application	
<i>Actual Result:</i> passed	

### Test Case 8: Login Student

**Test:** Login into the application by providing wrong email address and password

Table 5.8 Login Student with wrong email address and password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Login with wrong email address and password	<i>Test ID:</i> 8
Version:1.1	<i>Test Type:</i> System
<i>Input:</i> Email Id: <u>haroonqureshi60@gmail.com</u> Password: Haroon123@#	
<i>Expected Result:</i> Student will not be logged in to the application	

<i>Actual Result:</i> passed
------------------------------

### **Test Case 9: Register Supervisor**

**Test:** Register into the application by providing email address, and matching password and confirm password

Table 5.9 Register Supervisor with email address, matching password and confirm password

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Register with email address, matching password and confirm password	<i>Test ID:</i> 9
<i>Version:</i> 1	<i>Test Type:</i> System
<i>Input:</i> Email Id: <u>samirobaid@gmail.com</u> Password: Samir123@#	
<i>Expected Result:</i> Supervisor will be register in to the application	
<i>Actual Result:</i> passed	

### **Test Case 10: Register Supervisor**

**Test:** Register into the application by providing email address, and not matching password and confirm password

Table 5.10 Register Supervisor with email address, not matching password and confirm password

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Register with email address, not matching password and confirm password	<i>Test ID:</i> 10
<i>Version:</i> 1.1	<i>Test Type:</i> System
<i>Input:</i> Email Id: <u>samirobaid@gmail.com</u> Password: Samir123@	
<i>Expected Result:</i> Supervisor will not be register in to the application	
<i>Actual Result:</i> passed	

### **Test Case 11: Register Student**

**Test:** Register into the application by providing email address, and matching password and confirm password

Table 5.11 Register Student with email address, matching password and confirm password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Register with email address, matching password and confirm password	Test ID:11
Version:1	Test Type: System
<i>Input:</i>	
Email Id: <u>haroonqureshi60@gmail.com</u>	
Password: Haroon123@#	
<i>Expected Result:</i> Student will be register in to the application	
<i>Actual Result:</i> passed	

### **Test Case 12: Register Student**

**Test:** Register into the application by providing email address, and not matching password and confirm password

Table 5.12 Register Student with email address, not matching password and confirm password

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Register with email address, not matching password and confirm password	Test ID:12
Version:1.1	Test Type: System
<i>Input:</i>	
Email Id: <u>haroonqureshi60@gmail.com</u>	
Password: Haroon123@	
<i>Expected Result:</i> Student will not be register in to the application	
<i>Actual Result:</i> passed	

### **Test Case 13: Create Term**

**Test:** Create term with all valid inputs

Table 5.13 Create Term with valid inputs

Date: 23 January 2020	
<b>System:</b> Final Year Project Management & Tracking System	
<b>Objective:</b> Create term with valid inputs	<i>Test ID:13</i>
<b>Version:</b> 1	<i>Test Type: System</i>
<i>Input:</i>	
Term Name= 201	
Date From= 17 Feb, 2020	
Date To= 20 June, 2020	
Working Term= True	
<i>Expected Result:</i> Data will be added to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

#### **Test Case 14: Create Term**

**Test:** Create term with all invalid inputs

Table 5.14 Create Term with invalid inputs

Date: 23 January 2020	
<b>System:</b> Final Year Project Management & Tracking System	
<b>Objective:</b> Create term with invalid inputs	<i>Test ID:14</i>
<b>Version:</b> 1.1	<i>Test Type: System</i>
<i>Input:</i>	
Term Name= 201	
Date From= 20 June, 2020	
Date To= 17 Feb, 2020	
Working Term= Working	
<i>Expected Result:</i> Data will not be added to the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

#### **Test Case 15: Edit Term**

**Test:** Edit term by changing Date To field

Table 5.15 Edit Term

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Edit term by changing Date To field	<i>Test ID:</i> 15
<i>Version:</i> 1.2	<i>Test Type:</i> System
<i>Input:</i> Date To=18 Feb, 2020	
<i>Expected Result:</i> Data will be edited to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 16: Delete Term

**Test:** Delete term

Table 5.16 Delete Term

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Delete term	<i>Test ID:</i> 16
<i>Version:</i> 1.3	<i>Test Type:</i> System
<i>Input:</i> <i>Remove Button Clicked</i>	
<i>Expected Result:</i> Data will be deleted from the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 17: Create Program Learning Objective

**Test:** Create Program Learning Objective with valid inputs

Table 5.17 Create Program Learning Objective with valid inputs

Date: 23 January 2020	
-----------------------	--

<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Create Program Learning objective with valid inputs	<i>Test ID:</i> 17
<i>Version:</i> 1	<i>Test Type:</i> System
<i>Input:</i>	
Term Name= 201 Program Learning Objective: Ethical Values	
<i>Expected Result:</i> Data will be added to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 18: Create Program Learning Objective

**Test:** Create Program Learning Objective with previously made Program Learning Objective

Table 5.18 Create Program Learning Objective with invalid inputs

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Create Program Learning Objective with previously made Program Learning Objective	<i>Test ID:</i> 18
<i>Version:</i> 1.1	<i>Test Type:</i> System
<i>Input:</i>	
Term Name= 201 Program Learning Objective: Problem Analysis	
<i>Expected Result:</i> Data will not be added to the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 19: Edit Program Learning Objective with new Program Learning Objective statement

**Test:** Edit Program Learning Objective

Table 5.19 Edit Program Learning Objective

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Edit Program Learning Objective	Test ID:19
Version:1.2	Test Type: System
<p><i>Input:</i> Program Learning Objective: Programming Analysis</p>	
<p><i>Expected Result:</i> Data will be edited to the database and will be shown by system in a table</p>	
<p><i>Actual Result:</i> passed</p>	

### Test Case 20: Delete Program Learning Objective

**Test:** Delete Program Learning Objective

Table 5.20 Delete Program Learning Objective

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Delete Program Learning Objective	Test ID:20
Version:1.3	Test Type: System
<p><i>Input:</i> Remove Button Clicked</p>	
<p><i>Expected Result:</i> Data will be deleted from the database and will not be shown by system in a table</p>	
<p><i>Actual Result:</i> passed</p>	

### Test Case 21: Create Course Learning Objective

**Test:** Create Course Learning Objective with term name, Program Learning Objective name

Table 5.21 Create Course Learning Objective with valid inputs

Date: 23 January 2020	
-----------------------	--

<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Create Course Learning Objective with term name, Program Learning Objective name	<i>Test ID:</i> 21
<i>Version:</i> 1	<i>Test Type:</i> System
<i>Input:</i>	
Term Name= 201 Program Learning Objective: Ethical Values Course Learning Objective: To identify the requirements	
<i>Expected Result:</i> Data will be added to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 22: Create Course Learning Objective

**Test:** Create Course Learning Objective with previously made Course Learning Objective

Table 5.22 Create Course Learning Objective with invalid inputs

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Create Course Learning Objective with previously made Course Learning Objective	<i>Test ID:</i> 22
<i>Version:</i> 1.1	<i>Test Type:</i> System
<i>Input:</i>	
Term Name= 201 Program Learning Objective: Ethical Values Course Learning Objective: To identify the requirements	
<i>Expected Result:</i> Data will not be added to the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 23: Edit Course Learning Objective with new Course Learning Objective statement

**Test:** Edit Course Learning Objective

Table 5.23 Edit Course Learning Objective

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Edit Course Learning Objective	Test ID:23
Version:1.2	Test Type: System
<i>Input:</i> Course Learning Objective: To identify the document	
<i>Expected Result:</i> Data will be edited to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 24: Delete Course Learning Objective

**Test:** Delete Course Learning Objective

Table 5.24 Delete Course Learning Objective

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Delete Course Learning Objective	Test ID:24
Version:1.3	Test Type: System
<i>Input:</i> Remove Button Clicked	
<i>Expected Result:</i> Data will be deleted from the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 25: Create Exam

**Test:** Create Exam with Term name, Exam name, weightage, announced date, deadline

Table 5.25 Create Exam with valid inputs

Date: 23 January 2020	
-----------------------	--

<b>System:</b> Final Year Project Management & Tracking System	
<b>Objective:</b> Create Exam with Term name, Exam name, weightage, announced date, deadline	<i>Test ID:</i> 25
<b>Version:</b> 1	<i>Test Type:</i> System
<i>Input:</i>	
Term Name= 201 Name: Proposal Defense Weightage: 20 Announced Date: 17 Feb, 2020 Deadline: 1 <sup>st</sup> March, 2020	
<i>Expected Result:</i> Data will be added to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 26: Create Exam

**Test:** Create Exam with Term name and invalid Exam name, weightage, announced date, deadline

Table 5.26 Create Exam with invalid inputs

Date: 23 January 2020	
<b>System:</b> Final Year Project Management & Tracking System	
<b>Objective:</b> Create Exam with Term name and invalid Exam name, weightage, announced date, deadline	<i>Test ID:</i> 26
<b>Version:</b> 1.1	<i>Test Type:</i> System
<i>Input:</i>	
Term Name= 201 Name: Proposal Defense Weightage: 20 Announced Date: 1 <sup>st</sup> March, 2020 Deadline: 17 Feb, 2020	
<i>Expected Result:</i> Data will not be added to the database and will not be shown by system in a table	

<i>Actual Result:</i> passed
------------------------------

### **Test Case 27: Edit Exam with new Exam statement**

**Test:** Edit Exam

Table 5.27 Edit Exam

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Edit Exam	Test ID:27
Version:1.2	Test Type: System
<i>Input:</i> Name: Mid Term	
<i>Expected Result:</i> Data will be edited to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### **Test Case 28: Delete Exam**

**Test:** Delete Exam

Table 5.28 Delete Exam

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Delete Exam	Test ID:28
Version:1.3	Test Type: System
<i>Input:</i> Remove Button Clicked	
<i>Expected Result:</i> Data will be deleted from the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### **Test Case 29: Create Questions**

**Test:** Create Questions with Term name, Exam type, Course Learning Objective

Table 5.29 Create Questions with valid inputs

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Create Questions with Term name, Exam type, Course Learning Objective	Test ID:29
Version:1	Test Type: System
<p><i>Input:</i>            Term Name= 201            Course Learning Objective: To identify the requirements</p>	
<i>Expected Result:</i> Data will be added to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### **Test Case 30: Create Questions**

**Test:** Create question without attaching Course Learning Objective

Table 5.30 Create Questions with invalid inputs

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Create Questions without attaching Course Learning Objective	Test ID:30
Version:1.1	Test Type: System
<p><i>Input:</i>            Course Learning Objective: " "</p>	
<i>Expected Result:</i> Data will not be added to the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### **Test Case 31: Edit Question with new question statement**

**Test: Edit Question**

Table 5.31 Edit Questions

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Edit Question	Test ID:31
Version:1.2	Test Type: System
<i>Input:</i> Question 1: " "	
<i>Expected Result:</i> Data will not be edited to the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

**Test Case 32: Delete Questions****Test: Delete Questions**

Table 5.32 Delete Questions

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Delete Questions	Test ID:32
Version:1.3	Test Type: System
<i>Input:</i> Remove Button Clicked	
<i>Expected Result:</i> Data will be deleted from the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

**Test Case 33: Create Panel****Test: Create panel with valid input**

Table 5.33 Create Panel with valid inputs

Date: 23 January 2020	
-----------------------	--

<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Create panel with valid input	<i>Test ID:</i> 33
<i>Version:</i> 1	<i>Test Type:</i> System
<i>Input:</i> Panel: Panel No. 1	
<i>Expected Result:</i> Data will be added to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 34: Create Panel

**Test:** Create panel with previously made panel

Table 5.34 Create Panel with invalid inputs

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Create panel with previously made panel	<i>Test ID:</i> 34
<i>Version:</i> 1.1	<i>Test Type:</i> System
<i>Input:</i> Panel: Panel No. 1	
<i>Expected Result:</i> Data will not be added to the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 35: Edit panel with new panel

**Test:** Edit panel

Table 5.35 Edit Panel

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Edit panel	<i>Test ID:</i> 35

<i>Version:1.2</i>	<i>Test Type: System</i>
<i>Input:</i> Panel: “ ”	
<i>Expected Result:</i> Data will not be edited to the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### **Test Case 36: Delete Panel**

**Test:** Delete panel

Table 5.36 Delete Panel

<i>Date: 23 January 2020</i>	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Delete panel	<i>Test ID:36</i>
<i>Version:1.3</i>	<i>Test Type: System</i>
<i>Input:</i> <i>Remove Button Clicked</i>	
<i>Expected Result:</i> Data will be deleted from the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### **Test Case 37: Create Supervisor**

**Test:** Create supervisor with term name, panel number, supervisor name, email, designation

Table 5.37 Create Supervisor with valid inputs

<i>Date: 23 January 2020</i>	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Create supervisor with term name, panel number, supervisor name, email, designation	<i>Test ID:37</i>
<i>Version:1</i>	<i>Test Type: System</i>

<i>Input:</i> Term Name: 201 Panel: Panel No 1 Name: Mr. Samir Obaid Email: <a href="mailto:samirobaid@gmail.com">samirobaid@gmail.com</a> Designation: Professor	<i>Expected Result:</i> Data will be added to the database and will be shown by system in a table
<i>Actual Result:</i> passed	

### Test Case 38: Create Supervisor

**Test:** Create supervisor with previously made supervisor and invalid email address

Table 5.38 Create Supervisor with invalid inputs

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Create supervisor with previously made supervisor and invalid email address	Test ID:38
Version:1.1	Test Type: System
<i>Input:</i> Term Name: 201 Panel: Panel No 1 Name: Mr. Samir Obaid Email: samir@ Designation: Professor	
<i>Expected Result:</i> Data will not be added to the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 39: Edit supervisor with new supervisor name

**Test:** Edit supervisor

Table 5.39 Edit Supervisor

Date: 23 January 2020	
-----------------------	--

<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Edit supervisor	<i>Test ID:</i> 39
<i>Version:</i> 1.2	<i>Test Type:</i> System
<i>Input:</i> Name: Ms. Ghazala Mehmood	
<i>Expected Result:</i> Data will be edited to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 40: Delete Supervisor

**Test:** Delete Supervisor

Table 5.40 Delete Supervisor

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Delete Supervisor	<i>Test ID:</i> 40
<i>Version:</i> 1.3	<i>Test Type:</i> System
<i>Input:</i> <i>Remove Button Clicked</i>	
<i>Expected Result:</i> Data will be deleted from the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 41: Create Announcements

**Test:** Create announcement with term name

Table 5.41 Create Announcement with valid inputs

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Create announcement with term name	<i>Test ID:</i> 41
<i>Version:</i> 1	<i>Test Type:</i> System

<i>Input:</i> Term Name: 193 Announcement: "Exam will be held next Tuesday"
<i>Expected Result:</i> Data will be added to the database and will be shown by system in a table
<i>Actual Result:</i> passed

### Test Case 42: Edit announcement

**Test:** Edit Announcement

Table 5.42 Edit Announcement

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Edit announcement	<i>Test ID:</i> 42
Version:1.1	<i>Test Type:</i> System
<i>Input:</i> Announcement: "Exam will be held next thursday"	
<i>Expected Result:</i> Data will be edited to the database and will be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 43: Delete Announcement

**Test:** Delete Announcement

Table 5.43 Delete Announcement

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Delete Announcement	<i>Test ID:</i> 43
Version:1.2	<i>Test Type:</i> System
<i>Input:</i> <i>Remove Button Clicked</i>	
<i>Expected Result:</i> Data will be deleted from the database and will not be shown	

by system in a table
Actual Result: passed

### Test Case 44: Create Group

**Test:** Create group with term name, student 1, student 2, student 3.

Table 5.44 Create Group

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Create group with term name, student 1, student 2, student 3	Test ID:44
Version:1	Test Type: System
<i>Input:</i>	
Term Name: 193	
Student 1: Hina Mahmood Qureshi	
Student 2: Muhammad Haroon Qureshi	
Student 3: Syed Qamar Abbass	
Group No: Group No. 1	
<i>Expected Result:</i> Data will be added to the database and will be shown in dropdown	
Actual Result: passed	

### Test Case 45: Create Group

**Test:** Create group with term name and adding three student's name but with same group no.

Table 5.45 Create Group with invalid inputs

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Create group with term name and adding three student's name but with same group no.	Test ID:45
Version:1.1	Test Type: System

<i>Input:</i> Term Name: 193 Student 1: Rafay Student 2: Hajra Student 3: Qasim Group No: Group No. 1	
<i>Expected Result:</i> Data will not be added to the database and will not be shown by system in a table	
<i>Actual Result:</i> passed	

### Test Case 46: Assign Project Title

**Test:** Assigning project title to a group already created

Table 5.46 Assign Project Title

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Assign project title to a group already created	<i>Test ID:</i> 46
<i>Version:</i> 1	<i>Test Type:</i> System
<i>Input:</i> Term Name: 193 Group No: Group No. 1 Project Title: Car System	
<i>Expected Result:</i> Data will be added to the database and will be shown in table	
<i>Actual Result:</i> passed	

### Test Case 47: Register Student for final year project

**Test:** Register student for final year project with all the required inputs.

Table 5.47 Register student with valid inputs

Date: 23 January 2020	
<i>System:</i> Final Year Project Management & Tracking System	
<i>Objective:</i> Register student for valid inputs	<i>Test ID:</i> 47
<i>Version:</i> 1	<i>Test Type:</i> System

<i>Input:</i>
Term Name: 193
Name: Hina Mahmood Qureshi
Registration number: BCS163008
Date of Birth: 21 December 1996
Phone number: 03241234567
Email address: <a href="mailto:hinaqureshi051@gmail.com">hinaqureshi051@gmail.com</a>
Credit hours enrolled: 17
Credit hour completed: 117
Cumulative GPA: 3.65
<i>Expected Result:</i> Data will be added to the database
<i>Actual Result:</i> passed

### Test Case 48: Register Student for Final year project

**Test:** Register student for final year project without adding all the inputs

Table 5.48 Register Student with invalid inputs

Date: 23 January 2020	
System: Final Year Project Management & Tracking System	
Objective: Register student for final year project without adding all the inputs	Test ID: 48
Version: 1.1	Test Type: System
<i>Input:</i>	
Term Name: 193	
Name:	
Registration number:	
Date of Birth:	
Phone number:	
Email address:	
Credit hours enrolled:	
Credit hour completed:	
Cumulative GPA:	
<i>Expected Result:</i> Data will not be added to the database	
<i>Actual Result:</i> passed	

# **Chapter 6**

## **Software Deployment**

### **6.1. Installation / Deployment Process Description**

Web app deployed on Somee.com and the website is custfyp.somee.com

Somee.com provides window hosting, asp .net hosting, MS SQL hosting.

Somee follows a proper format for user verification and authentication so that a verified/authentic user uses the application. The format includes following procedure as narrated

Somee gives a wide range of products including the package, we use that firstly the somee.com verifies us by sending a confirmation email. As by then we become a verified user. Somee.com gives database-uploading option to live our database such that the stored data will be retrieving from online data storage. On the web-hosting site, we deployed all the files, which run our project smoothly gradually. After uploading our web application, it becomes live and can be used in any part of the world.

# **Chapter 7**

## **Project Evaluation**

This includes the examiners evaluation report, including the points to be revised/included along with the selected requirements in the next iteration.

### **7.1. Project Evaluation Report**

<b>Examiner Name:</b>	
<b>S. No.</b>	<b>Suggestion</b>
1	
2	
3	
4	
5	
6	
7	
8	

**Other Comments (If any):**

---

---

---

---

**Signature**

## References

Moodle.org