CUST Internship Portal

Ch. Mohsin Javed	**(BCS183092)**
Ali Hamid	**(BCS183073)**
Shazib Hussain	**(BCS183056)**

Fall-2021

Supervised By

Mr. Omaid Ghayyur

Department of Computer Science

Capital University of Science & Technology, Islamabad

Submission Form for Final-Year

PROJECT REPORT

Version	v 1.0	**NUMBER OF MEMBERS**	3

TITLE	CUST Internship Portal v 1.0

SUPERVISOR NAME	Sir Omaid Ghayyur

MEMBER NAME	REG. NO.	EMAIL ADDRESS
Ch. Mohsin Javed	BCS183092	mohsinj50@gmail.com
Ali Hamid	BCS183073	raoalihamid810@gmail.com
Shazib Hussain	BCS183056	shazibali30@gmail.com

MEMBERS' SIGNATURES

____

____

____

Supervisor's Signature

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides the details of the project scope, the specifications of the project and the existing solutions of the project. The useful tools and technologies, project work break down and the timeline of the project are also discussed in details in this project.

1.1. Project Introduction

CUST Internship portal is a platform in which internship coordinators for multiple departments will be able to manage students' internship allocation, keep tracks of student performance during internship duration, issue internship letters and get reports from evaluators and much more. This web-based platform will help coordinators to perform main tasks automatically keep their record, system will help to manage all the process and workflow of students' internship progress like offer letters, internship completion certificates and their day-to-day progress reports. The internship coordinators will also have the benefit to assign the internship evaluation task to faculty members from the concerned department through online portal.

As managing all the records of students' internship progress manually is not time consuming task for coordinators like arranging and managing all the paperwork of student's internship records, CUST Internship Portal web application will help automate most of the process to minimize manual work and will keep records internships done by registered students in an organized way which will be easily trackable for coordinators to track the students' progress report and have benefit of assigning evaluation task to faculty members.

1.2. Main Scope

As this project covers numerous modules to be developed in designing CUST Internship portal, so our main module in designing this project is the Coordinator module in which the coordinator will view and manage the registered students who are applying in internship program term wise. The main role of the coordinator is to track the student progress and will apply the process of evaluation with the support of their respective faculty members. The coordinator will also verify the student’s current organization and keep the records of those organization for future purposes. The coordinator will also communicate and have the authority to apply changes in rules and regulations of the internship program.

As this project covers an entirely huge amount of requirements and their objectives to be implemented in this project, so our entirely main focus is on coordinator part which is basically the core part of designing CUST internship portal.

There will also be a few implementation on student’s module because it has a highly strong relation between student and coordinator through the system to run the internship process. Student part is to get the details of the organization as an interne and submit all the organization information to coordinator for verification of an organization. Student will also be tracked by coordinator and from multiple evaluators for the verification of internship progress.

1.3. Existing Examples / Solutions

CUST University has provided portal for students that provides an automated processes to register courses, view grades, view transcript, generate challan form, check attendance reports, fees summary report and courses, etc. Capital University in house development team has also developed portals for faculty including CMS, FYP, Advisory portal to facilitate faculty and automate the process to maximum. We are intend to semi automate internship process for coordinators. At present no such portal is under development or available to manage internship work with features to ease and automate maximum tasks of internship coordinators. Developing such internship web portal will provide a solution to save maximum time and help automate most of the process keep the record of organizations for internships, students’ reports, evaluation data etc.

1.4. Business Scope

Most universities may or may not offer internship opportunities for students per degree requirement. Even if they do, the workflow of their registration process, tracking students' records, managing their documents and all kinds of information about the students' internship records are done in an old-fashioned way. As this is an era of automated technology, CUST University has provided and will provide many ways to give benefits to students as well as faculty members to manage and manipulate all their corresponding process of information through online portal. We, as a student of this university, will provide the benefits to this university to be fully automated by creating this type of projects in which will increase the market value of this university.

1.5. Useful Tools and Technologies

Following are the useful tools and technologies in which we will intend to use in this project.

PHP (The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases.)

ReactJS (ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components.)

Laravel (Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern.)

VS Code (Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications.)

XAMPP (XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server.)

MySQL (MySQL is the most popular Open Source Relational SQL database management system. MySQL database management will be used to store data.)

1.6. Project Work Break Down

A project breakdown structure is a chart in which work elements are divided into different modules. We will complete the different modules of this project so that we can integrate the modules to make the whole project.

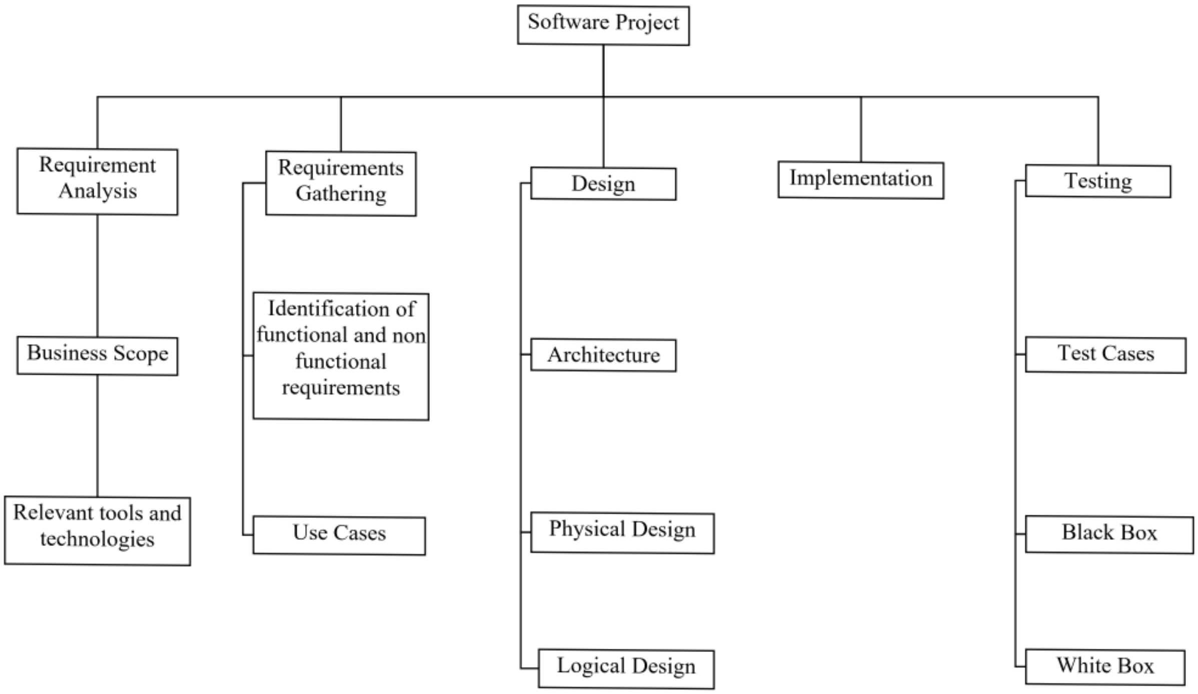


Figure 1. Project Work Breakdown

1.7. Project Time Line

A Gantt chart outlines what aspects of the project will be completed and by when. A chart between number of weeks and project breakdown are shown below.

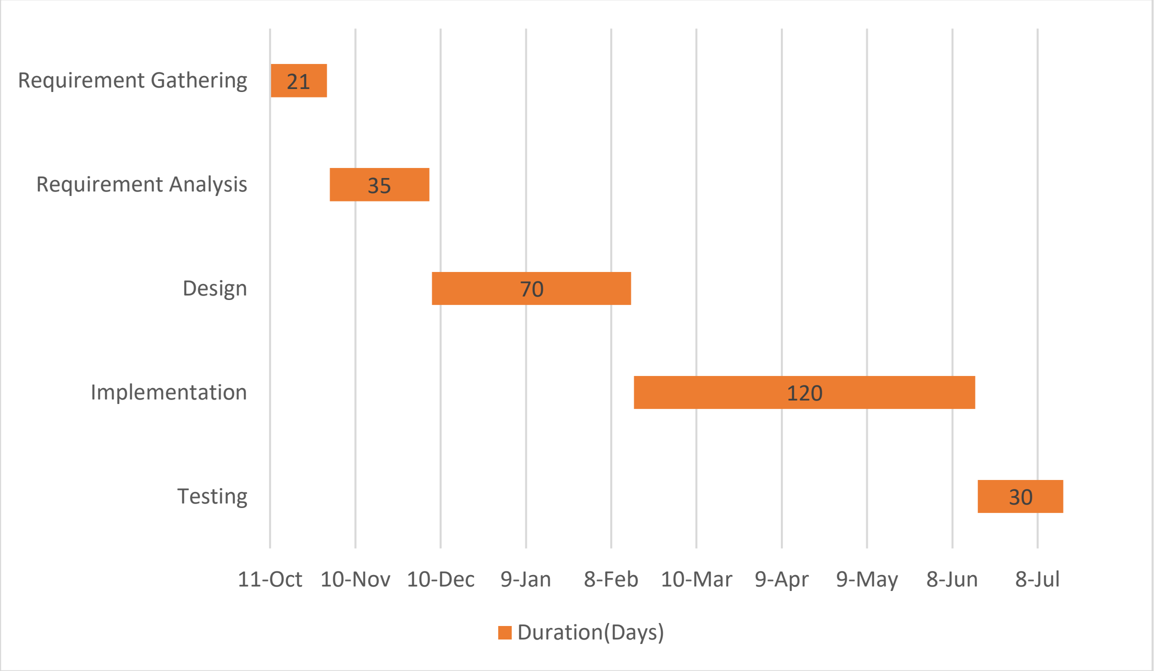


Figure 2. Project Timeline

Chapter 2

Requirement Specification and Analysis

Following are the requirement specifications and analysis that were gathered at the start of this project. It provides a description of system properties and its attributes and explains how system should behave or interact with user.

2.1. Functional Requirements

Mostly, all functional requirements about this project are core requirements. Following are the function requirements given below:

Table 1. Functional Requirements

S. No.	Functional Requirement	Status
	Coordinators	
1	Can view the registered student's internship program term wise	Pending
2	Can assign the recommendation letter	Pending
3	Can view student's result assigned by evaluators	Pending
4	Can make announcements through portal and SMS	Pending
5	Can extract the list of those students who had done all the necessary requirements of internship program term wise	Pending
6	Can provide the guidelines to students to comply the necessary details of internship program	Pending
7	Can view student's internship progress	Pending
8	Can view the info about organization of students	Pending
	Students	

9	Can login through the account	Pending
10	Can download recommendation letter	Pending
11	Can download the internship program rules and guidelines related document	Pending
12	Can submit the organization information i.e. NTN and SECP number.	Pending
13	Can submit the offer letter acquired from organization	Pending
14	Can submit day-to-day internship report	Pending
15	Can view the assigned evaluators for viva	Pending
16	Can view the announcements from internal evaluator or coordinator	Pending
17	Can submit the organization/external evaluator email, name and contact info	Pending
	Internal Evaluators	
18	Can view the assigned students for viva.	Pending
19	Can view the student’s organization info	Pending
20	Can assign grades to students based on their performance	Pending
21	Can send the notification about the student’s performance and grades to internship coordinator	Pending
22	Can have the option to reassign grades for under circumstances.	Pending
23	Can view the external evaluator email and contact info	Pending
24	Can view the student’s internship progress report	Pending
	External Evaluators	
25	Can view the students assigned as an internee in the organization	Pending
26	Can submit evaluation Performa based on the student’s performance with remarks	Pending

27	Can view the internship coordinator email and contact info	Pending

2.2. Non-Functional Requirements

Following are the non-functional requirements given below.

Table 2. Non Functional Requirements

S. No.	Non Functional Requirements	Categories
1	Extraction of the students term wise and doing analysis must be at low cost	Response Time
2	Satisfaction for the coordinator to provide documents and send announcements to specified user	Maintainability
3	Keeping students' record term wise	Backup
4	Verifying organization details to ensure that the student has provided all the right details of the organization	Availability

2.3. Selected Functional Requirements

Following are the list of selected functional requirements for current iteration.

Table 3. Selected Functional Requirements

S. No.	Functional Requirement	Type
	Students	
1	Can login with account	Implemented
2	Can download recommendation letter	Implemented
3	Can change login id and password	Implemented
4	Can download the internship program rules and guidelines related document	Implemented

Coordinator		
5	Can login with account	Implemented
6	Can send recommendation letter	Implemented
7	Can send login account information to all students.	Implemented
8	Can extract the list of students term wise	Implemented
9	Can change account password	Implemented

2.4. System Use Case Modeling

Following are the sequence of actions a system performs that yields an observable result of value to a particular actor. The functionality of a system is defined by different use cases, each of which represents a specific goal (to obtain the observable result of value) for a particular actor. The interactions among the elements of a system and the relationships between and among the actors and the figures are given as follows.

2.4.1. Students

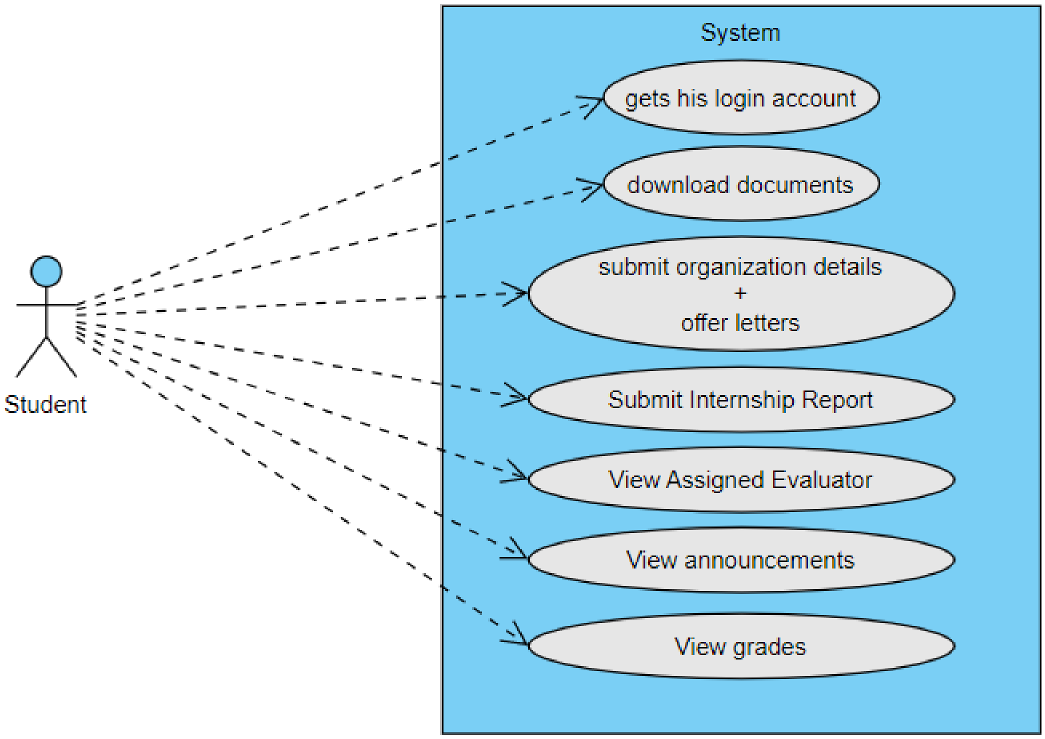


Figure 3. Use case diagram of Student

2.4.2. Coordinators

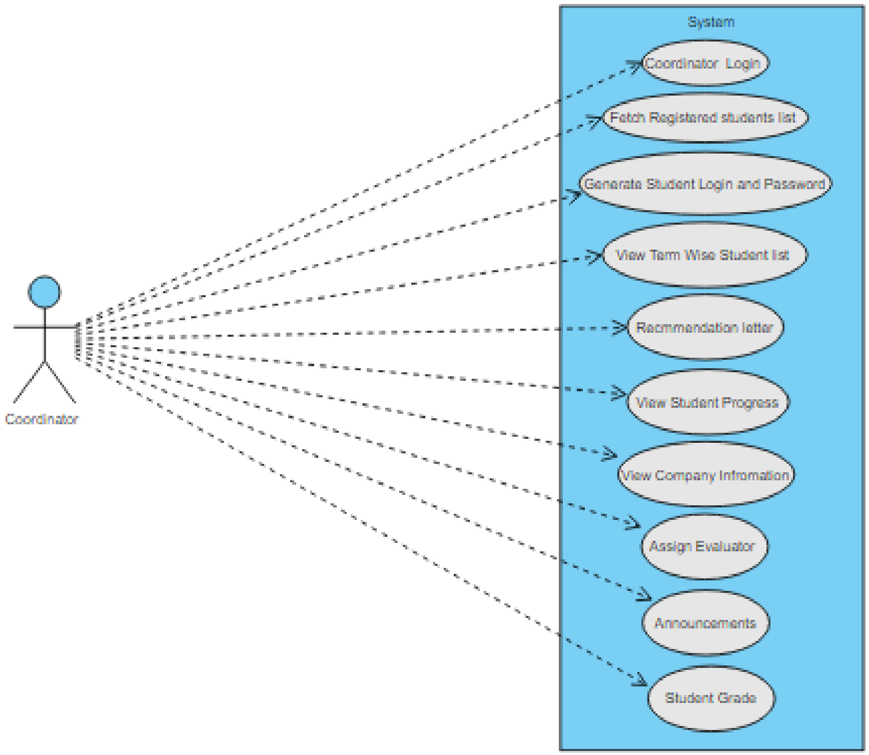


Figure 4. Use case diagram of Coordinator

2.5. Use case description

Following are the details of use cases of actors that have their specific roles of interaction with the system.

2.5.1. Students

Table 4. Student Login UCD

Use Case ID:	1		
Use Case Name:	Student Login		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	10/12/2021	**Last Revision Date:**	N/A
Actors:	Student		
Description:	Student login into system by giving login ID and password.		
Trigger:	By Clicking login Button		
Preconditions:	Only registered student can login		
Post conditions:	Student successfully logged In into system		
Normal Flow:	Actors	System	
	1. Registered student in internship can login into the system. 2. Student can view details of internship Coordinator	3. System will display student dashboard.	
Alternative Flows:	Student forgot password Or Student is not registered in internship		
Exceptions:	N/A		

Table 5. Student Download Documents UCD

Use Case ID:	2		
Use Case Name:	Student Download Documents		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	10/12/2021	**Last Revision Date:**	N/A
Actors:	Student		
Description:	Student can download the documents which are uploaded by the Coordinator		
Trigger:	Click on Download Button		
Preconditions:	Student is logged in Document should be there to download		
Post conditions:	Downloaded Document		
Normal Flow:	Actors	System	
	1. Student can download Internship timeline document. 2. Student can download his recommendation letter. 3. Student can download documents uploaded by the coordinator.	4. System will Show document uploaded by the Coordinator	
Alternative Flows:	No document available to download		
Exceptions:	N/A		

Table 6. Offer Letter Submission UCD

Use Case ID:	3		
Use Case Name:	Offer letter Submission		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	10/12/2021	**Last Revision Date:**	N/A
Actors:	Student		
Description:	Student submits offer letter and organization information. Information includes NTN ,SECP and External evaluator information.		
Trigger:	By clicking submit Button		
Preconditions:	Student is logged in		
Post conditions:	Save all the information in to the database.		
Normal Flow:	Actors	System	
	1. Student will enter organizational details 2. Student will enter NTN and SECP number of the organization 3. Student will upload offer letter 4. Student will enter external evaluator information	5. System will save organizational data in organization database 6. System will save all the data in database 7. This information will be visible to coordinator, student and assigned internal evaluator	
Alternative Flows:	Student didn’t submit anything.		
Exceptions:	N/A		

Table 7. Submit Internship Report UCD

Use Case ID:	4		
Use Case Name:	Submit internship report		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	10/12/2021	**Last Revision Date:**	N/A
Actors:	Student		
Description:	Student will upload his internship report along with completion certificate. Also enter needed information if required.		
Trigger:	Upload internship report		
Preconditions:	Student is logged in. Student must have submitted offer letter and organizational details.		
Post conditions:	Student can view all the information he has submitted. He can also edit information		
Normal Flow:	Actors	System	
	1. Student will upload internship report 2. Student will upload certificate. 3. Student can review his report.	4. System will save organizational data in organization database 5. System will save all the data in database 6. This information will be visible to coordinator, student and assigned internal evaluator	
Alternative Flows:	Student will re-upload internship report if required..		
Exceptions:	N/A		

Table 8. Student View Announcements UCD

Use Case ID:	5		
Use Case Name:	Student View Announcements		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	10/12/2021	**Last Revision Date:**	N/A
Actors:	Student		
Description:	Student will view announcements by coordinator and assigned internal evaluator		
Trigger:	By clicking view announcement		
Preconditions:	Student is logged in		
Post conditions:	N/A		
Normal Flow:	Actors	System	
	1. Student can view announcement	2. System will show all the announcement made by coordinator and assigned evaluator.	
Alternative Flows:	No announcements have been made		
Exceptions:	N/A		

Table 9. View Assigned Evaluator UCD

Use Case ID:	6		
Use Case Name:	View Assigned Evaluator		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	10/12/2021	**Last Revision Date:**	N/A
Actors:	Student		
Description:	Student can view details about assigned internal evaluator		
Trigger:	By clicking View Evaluator		
Preconditions:	Student is logged in		
Post conditions:	N/A		
Normal Flow:	Actors	System	
	1. Student will view details about the internal evaluator	2. System will show details about assigned internal evaluator	
Alternative Flows:	No evaluator assigned		
Exceptions:	N/A		

Table 10. View Coordinator UCD

Use Case ID:	7		
Use Case Name:	View Coordinator		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	10/12/2021	**Last Revision Date:**	N/A
Actors:	Student		
Description:	Student can view details about coordinator		
Trigger:	By clicking View Coordinator		
Preconditions:	Student is logged in		
Post conditions:	N/A		
Normal Flow:	Actors	System	
	1. Student will view details about the coordinator	2. System will show details about the coordinator	
Alternative Flows:	N/A		
Exceptions:	N/A		

Table 11. View results UCD

Use Case ID:	8		
Use Case Name:	View Grades		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	10/12/2021	**Last Revision Date:**	N/A
Actors:	Student		
Description:	Student can view the grade of his/her internship		
Trigger:	By Clicking View Result		
Preconditions:	Student is logged in		
Post conditions:	N/A		
Normal Flow:	Actors	System	
	1. Student can view result of internship	2. System will show details about the coordinator	
Alternative Flows:	N/A		
Exceptions:	N/A		

Table 12. Download Recommendation letter UCD

Use Case ID:	9		
Use Case Name:	Download recommendation letter		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	10/12/2021	**Last Revision Date:**	N/A
Actors:	Student		
Description:	Student can download recommendation letter		
Trigger:	By clicking Download recommendation letter		
Preconditions:	Student is logged in		
Post conditions:	N/A		
Normal Flow:	Actors	System	
	1. Student can download his recommendation letter.	2. System will Show recommendation letter uploaded by the Coordinator	
Alternative Flows:	N/A		
Exceptions:	N/A		

2.5.2. Coordinator

Table 13. Coordinator Login UCD

Use Case ID:	10		
Use Case Name:	Coordinator Login		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	08/02/2022	**Last Revision Date:**	N/A
Actors:	Coordinator		
Description:	Coordinator login into system by giving login ID and password.		
Trigger:	By Clicking login Button		
Preconditions:	Only Predefined Coordinator can login by developer----temporary		
Post conditions:	Coordinator successfully logged In into system		
Normal Flow:	Actors	System	
	4. Coordinator can view registered students. 5. Coordinator can view and assign internal and external evaluator details.	6. System will display Coordinator dashboard.	
Alternative Flows:	Coordinator Forgot password		
Exceptions:	Database is not responding		

Table 14.View Registered Student UCD

Use Case ID:	11		
Use Case Name:	View Registered Student		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	08/02/2022	**Last Revision Date:**	N/A
Actors:	Coordinator		
Description:	Coordinator can view registered students for internship		
Trigger:	By Clicking View Registered Students		
Preconditions:	Coordinator must be logged in to system		
Post conditions:	Registered Students list can be seen		
Normal Flow:	Actors	System	
	1. There must be students registered for internship.	2. System will display registered students.	
Alternative Flows:	Students are not registered for internship		
Exceptions:	N/A		

Table 15.Generate Students Login UCD

Use Case ID:	12		
Use Case Name:	Generate Registered Student Username Password		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	08/02/2022	**Last Revision Date:**	N/A
Actors:	Coordinator		
Description:	Coordinator will generate students login and password for internship portal		
Trigger:	By clicking generate login		
Preconditions:	Only student registered in internship can have login and password		
Post conditions:	Student will receive user name and password by email.		
Normal Flow:	Actors	System	
	1. Coordinator will generate username for students. 2. Students will get their login id.	3. System will generate student username and login/ 4. System will send an email to students of their username and passwords.	
Alternative Flows:	No registered students		
Exceptions:	N/A		

Table 16.View Students Term wise UCD

Use Case ID:	13		
Use Case Name:	View term wise student list		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	08/02/2022	**Last Revision Date:**	N/A
Actors:	Coordinator		
Description:	Coordinator will view students by searching term wise internship.		
Trigger:	By Clicking Search term wise		
Preconditions:	Coordinator must be logged in		
Post conditions:	Coordinator will View term wise data of students		
Normal Flow:	Actors	System	
	1. Coordinator will get student list term wise 2. Coordinator can search Student in that list.	3. System will display lists of student.	
Alternative Flows:	No term wise data is available		
Exceptions:	N/A		

Table 17.Send Recommendation Letter UCD

<table>
<tr><td><b>Use Case ID:</b></td><td colspan="3">14</td></tr>
<tr><td><b>Use Case Name:</b></td><td colspan="3">Recommendation letters for student</td></tr>
<tr><td><b>Created By:</b></td><td>Ali Hamid</td><td><b>Last Updated By:</b></td><td>N/A</td></tr>
<tr><td><b>Date Created:</b></td><td>08/02/2022</td><td><b>Last Revision Date:</b></td><td>N/A</td></tr>
<tr><td><b>Actors:</b></td><td colspan="3">Coordinator</td></tr>
<tr><td><b>Description:</b></td><td colspan="3">Coordinator will send students recommendation letters.</td></tr>
<tr><td><b>Trigger:</b></td><td colspan="3">By Clicking send recommendation letters.</td></tr>
<tr><td><b>Preconditions:</b></td><td colspan="3">Recommendation letter format should be there.</td></tr>
<tr><td><b>Post conditions:</b></td><td colspan="3">All the student will get their recommendation letters.</td></tr>
<tr><td rowspan="2"><b>Normal Flow:</b></td><td colspan="2">Actors</td><td>System</td></tr>
<tr><td colspan="2">1. Coordinator will generate student’s recommendation letters.
2. Student will receive their recommendation letter and can download that.</td><td>3. System will send students their recommendation letters.</td></tr>
<tr><td><b>Alternative Flows:</b></td><td colspan="3">No registered students there to send recommendation letter</td></tr>
<tr><td><b>Exceptions:</b></td><td colspan="3">N/A</td></tr>
</table>

Table 18.View Student Progress UCD

Use Case ID:	15		
Use Case Name:	View student progress		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	08/02/2022	**Last Revision Date:**	N/A
Actors:	Coordinator		
Description:	Coordinator can view student progress		
Trigger:	By Clicking View student progress		
Preconditions:	Only registered students will progress		
Post conditions:	Student fulfil progress will able to see next task		
Normal Flow:	Actors	System	
	1. Coordinator view student progress. 2. Coordinator will see student submitted required information as progress	3. System will show students' progress	
Alternative Flows:	Student didn't do any task, No progress		
Exceptions:	N/A		

Table 19.View Company Information UCD

Use Case ID:	16		
Use Case Name:	View company information submitted by students		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	08/02/2022	**Last Revision Date:**	N/A
Actors:	Coordinator		
Description:	Coordinator will company view company information in which student is doing internship		
Trigger:	By Clicking View company information		
Preconditions:	Only registered student can upload company information		
Post conditions:	Company information will be displayed		
Normal Flow:	Actors	System	
	1. Coordinator will view company information uploaded by the students.	2. System will display company information.	
Alternative Flows:	No company records is there to display.		
Exceptions:	N/A		

Table 20. Assign Internal Evaluator UCD

Use Case ID:	17		
Use Case Name:	Assign Evaluator		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	08/02/2022	**Last Revision Date:**	N/A
Actors:	Coordinator		
Description:	Assign students to internal evaluator		
Trigger:	By Clicking Assign Evaluator		
Preconditions:	Registered Students must submit internship report, evaluation Performa and certificate		
Post conditions:	Assigned student will be notified		
Normal Flow:	Actors	System	
	1. Coordinator will assign students to internal evaluators. 2. Student and internal evaluator will be notified.	3. System will display assigned student to evaluator. 4. System will display student's internship records to evaluator.	
Alternative Flows:	No Students to assign.		
Exceptions:	N/A		

Table 21. Announcements UCD

Use Case ID:	18		
Use Case Name:	Announcements		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	08/02/2022	**Last Revision Date:**	N/A
Actors:	Coordinator		
Description:	Coordinator will make announcements for student and internal evaluator.		
Trigger:	By Clicking announcement		
Preconditions:	Coordinator must be logged in.		
Post conditions:	Notification will be send to student or evaluator		
Normal Flow:	Actors	System	
	1. Coordinator will make announcements.	2. System will display notification to user.	
Alternative Flows:	No announcement to made.		
Exceptions:	N/A		

Table 22. Student Grade View UCD

Use Case ID:	19		
Use Case Name:	Student grade		
Created By:	Ali Hamid	**Last Updated By:**	N/A
Date Created:	08/02/2022	**Last Revision Date:**	N/A
Actors:	Coordinator		
Description:	Coordinator will see assigned grades by internal evaluators and can change grades.		
Trigger:	By Clicking View Grades		
Preconditions:	Coordinator must be logged in.		
Post conditions:	Coordinator will view student grades.		
Normal Flow:	Actors	System	
	1. Coordinator will see grades of student.	2. System will display student grades.	
Alternative Flows:	No grades are there to display.		
Exceptions:	N/A		

2.6. System Sequence Diagrams

Sequence diagrams are created to show the sequence of events among user and the system to complete in action/use case. Following are the sequence diagram of actors.

2.6.1. Students

Here is the system sequence diagram in which shows interaction between the student and system use case (Reset Password SSD).

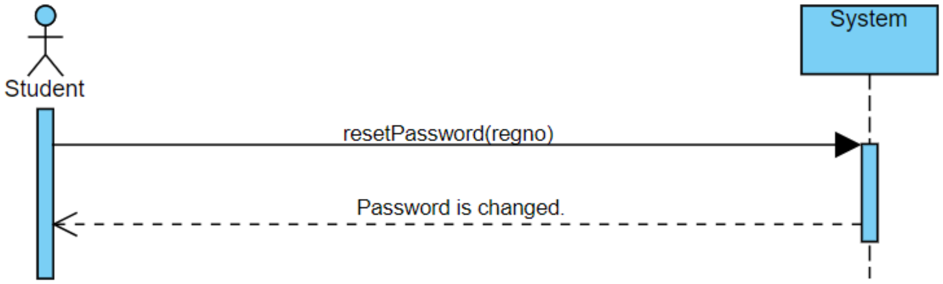


Figure 5. Reset Password SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (Forgot Password SSD).

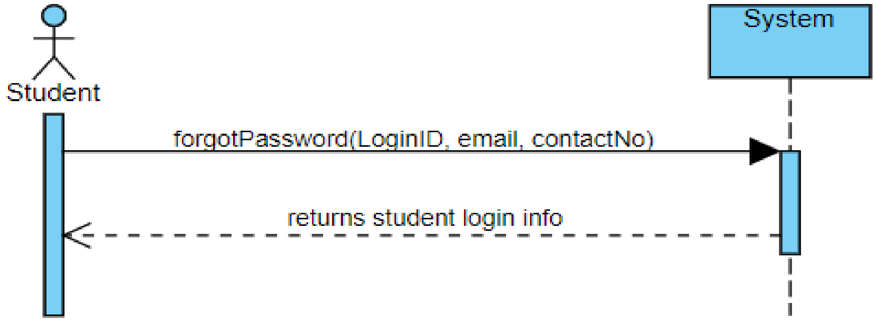

Student
System
forgotPassword(LoginID, email, contactNo)
returns student login info

Figure 6. Forgot Password SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (Get Coordinator Info SSD).

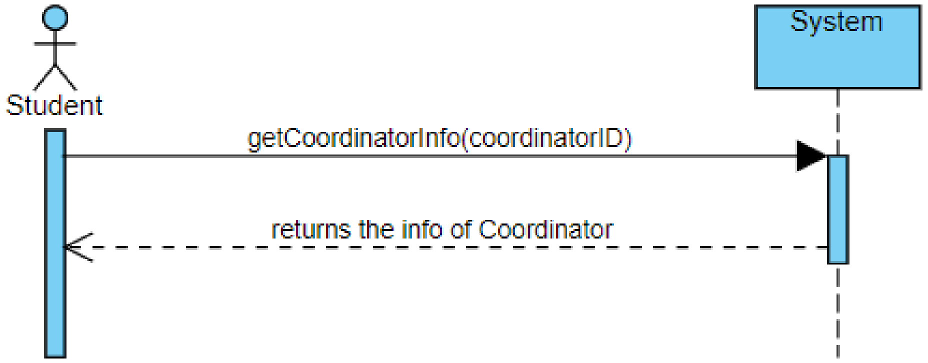

Student
System
getCoordinatorInfo(coordinatorID)
returns the info of Coordinator

Figure 7. Get Coordinator Info SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (Set Organization Details and offer letter SSD).

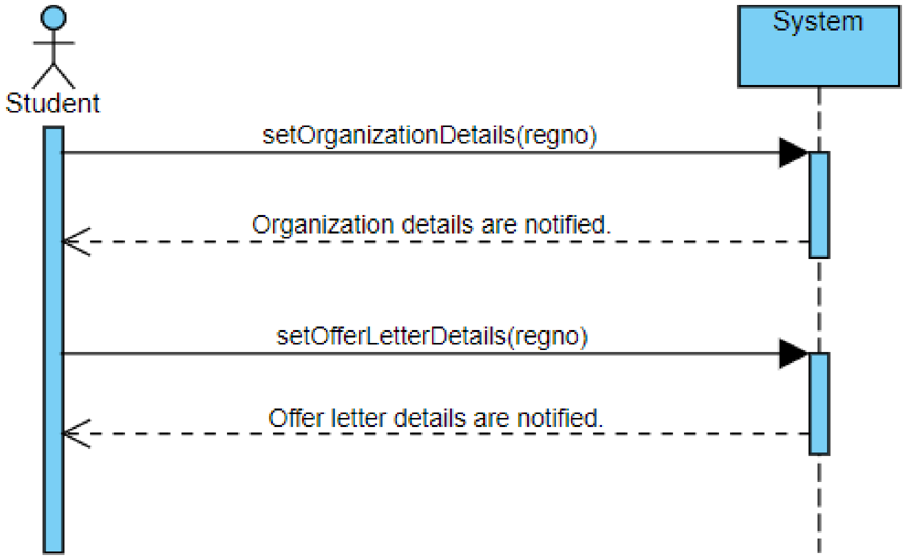

Student
System
setOrganizationDetails(regno)
Organization details are notified.
setOfferLetterDetails(regno)
Offer letter details are notified.

Figure 8. Set Organization details and offer letter SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (Set Report SSD).

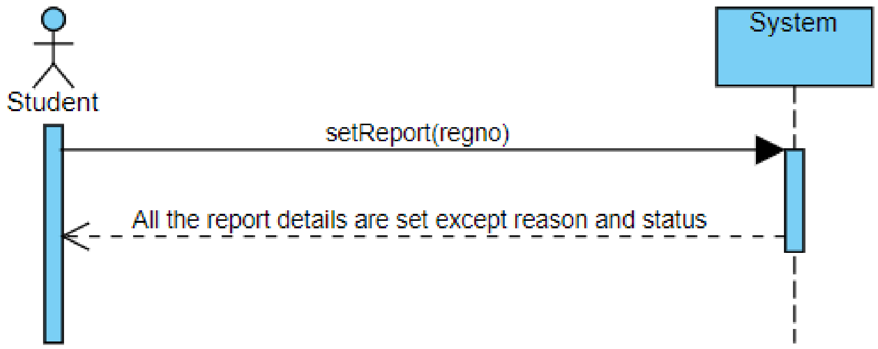


Figure 9. Set Report SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (Get Organization Info SSD).

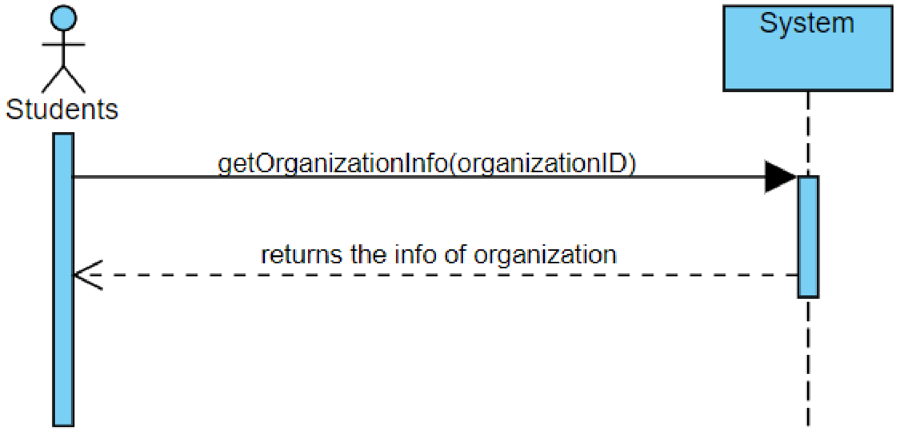

Students
System
getOrganizationInfo(organizationID)
returns the info of organization

Figure 10. Get Organization Info SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (View Grades SSD).

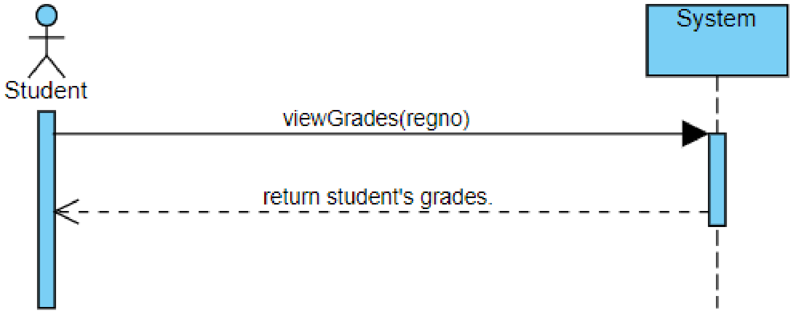


Figure 11. View Grades SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (Get External Evaluator info SSD).

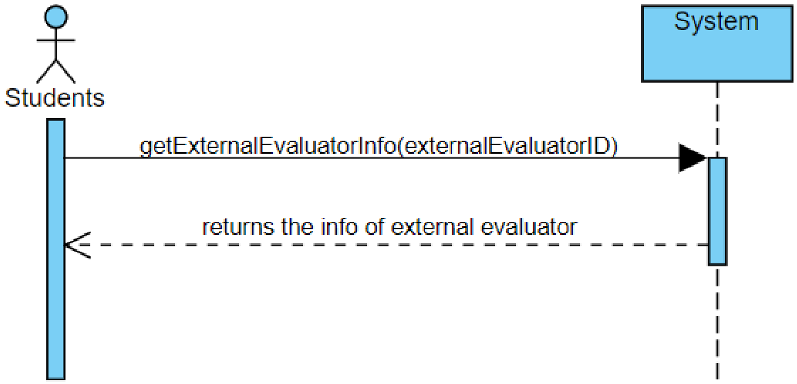

Students
System
getExternalEvaluatorInfo(externalEvaluatorID)
returns the info of external evaluator

Figure 12.Get External Evaluator Info SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (Show Documents SSD).

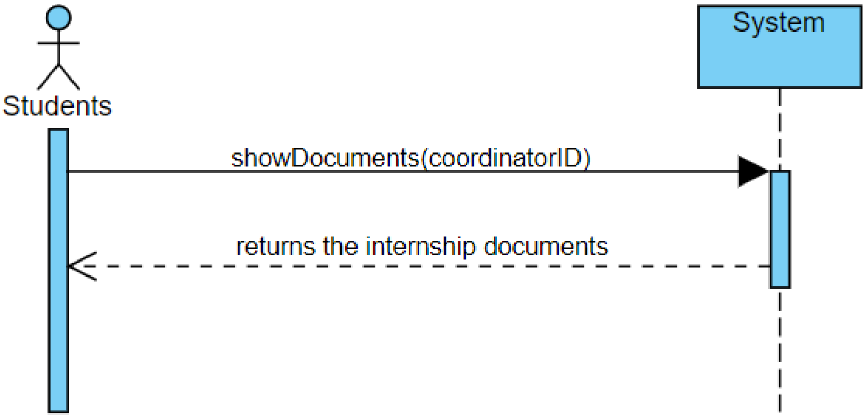

Students
System
showDocuments(coordinatorID)
returns the internship documents

Figure 13. Show Documents SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (Show Performa SSD).

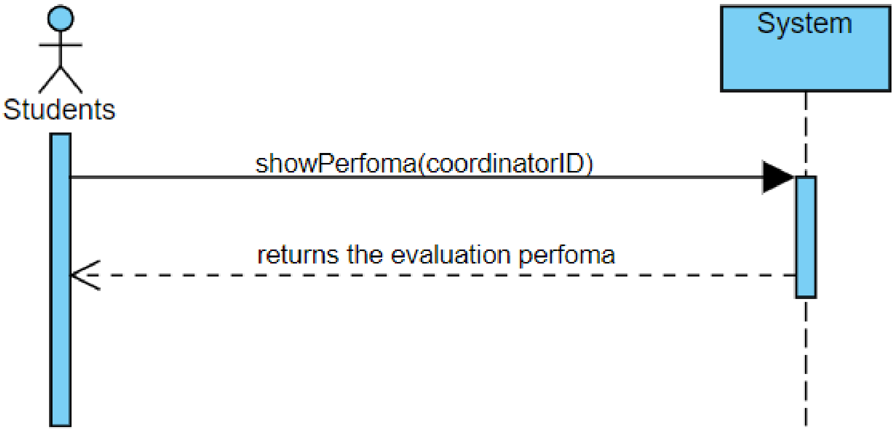


Figure 14. Show Performa SSD

Here is the system sequence diagram in which shows interaction between the student and system use case (View Announcements SSD).

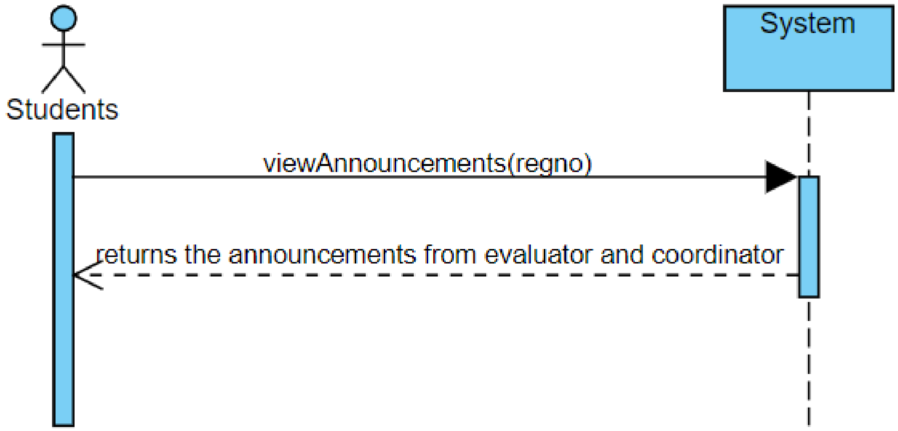


Figure 15. View Announcements SSD

2.6.2. Coordinator

Here is the system sequence diagram in which shows interaction between the coordinator and system use case (Reset Password SSD);

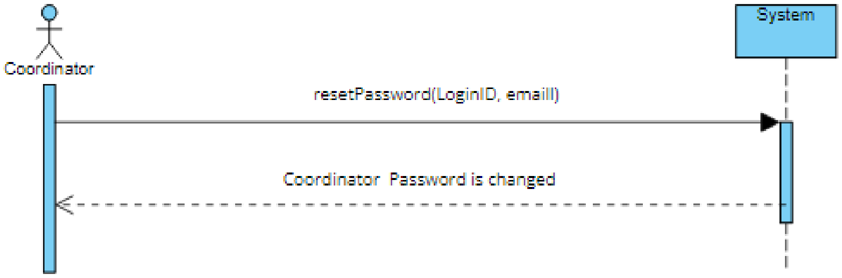

Coordinator
System
resetPassword(LoginID, emaill)
Coordinator Password is changed

Figure 16.Reset Password SSD

Here is the system sequence diagram in which shows interaction between the coordinator and system use case (Announcements SSD).

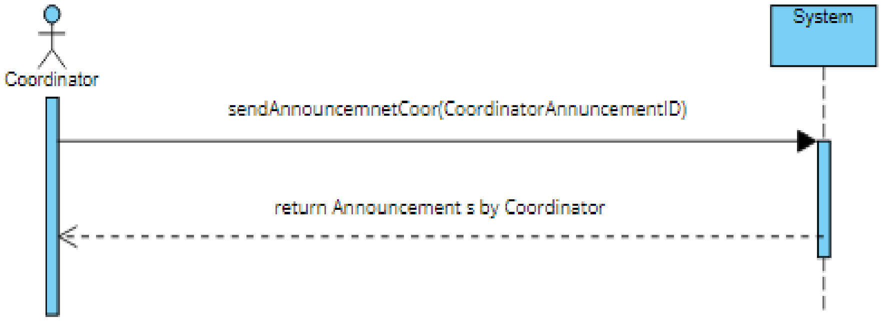

Coordinator
System
sendAnnouncemnetCoor(CoordinatorAnnuncementID)
return Announcement s by Coordinator

Figure 17. Announcements SSD

Here is the system sequence diagram in which shows interaction between the coordinator and system use case (Generate login for students SSD).

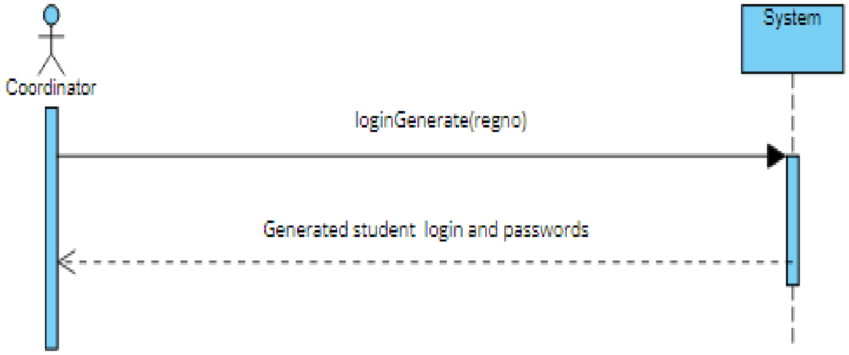


Figure 18. Generate Login SSD

Here is the system sequence diagram in which shows interaction between the coordinator and system use case (Assign Internal Evaluator SSD).

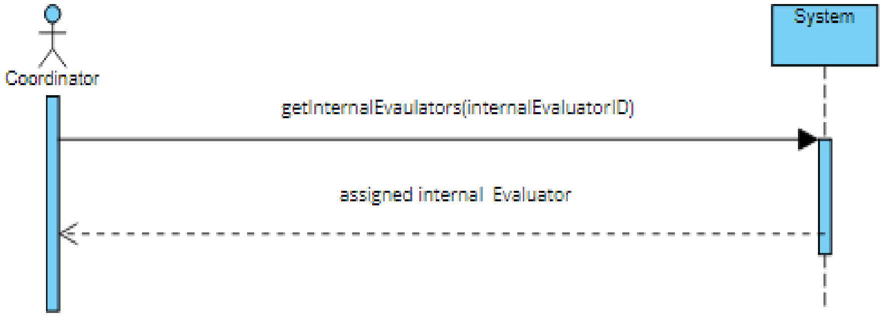

Coordinator
System
getInternalEvaulators(internalEvaluatorID)
assigned internal Evaluator

Figure 19. Assign Internal Evaluator SSD

Here is the system sequence diagram in which shows interaction between the coordinator and system use case (Get Organization Details SSD).

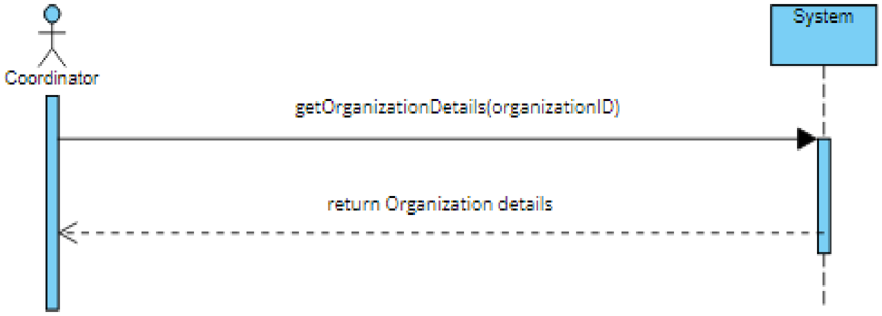

Coordinator
System
getOrganizationDetails(organizationID)
return Organization details

Figure 20. Get Organizational Information SSD

Here is the system sequence diagram in which shows interaction between the coordinator and system use case (Send Recommendation letters to Students SSD).

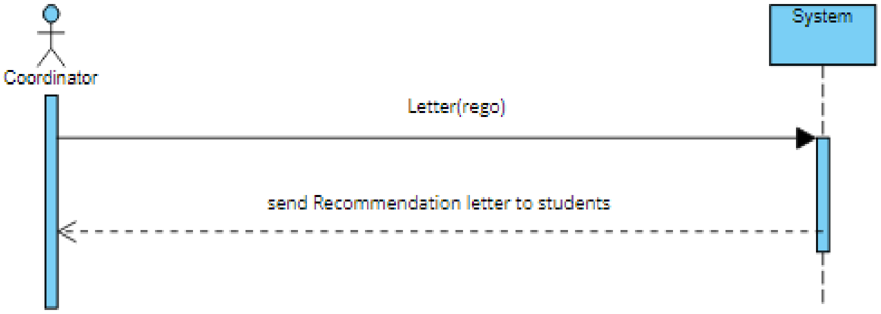


Figure 21.Send Recommendation letters SSD

Here is the system sequence diagram in which shows interaction between the coordinator and system use case (View Students Term wise SSD).

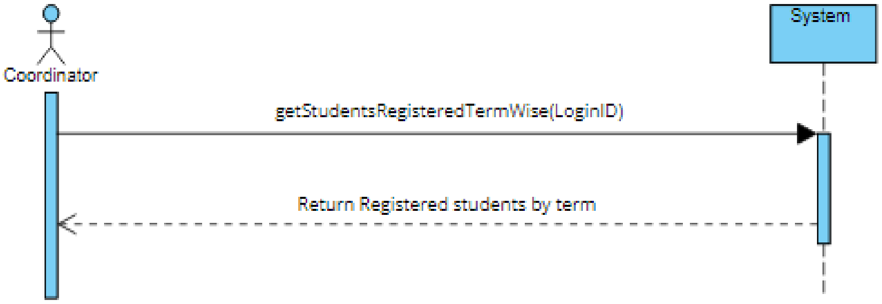

Coordinator
System
getStudentsRegisteredTermWise(LoginID)
Return Registered students by term

Figure 22. View Registered Students term wise SSD

Here is the system sequence diagram in which shows interaction between the coordinator and system use case (View Student Grades SSD).

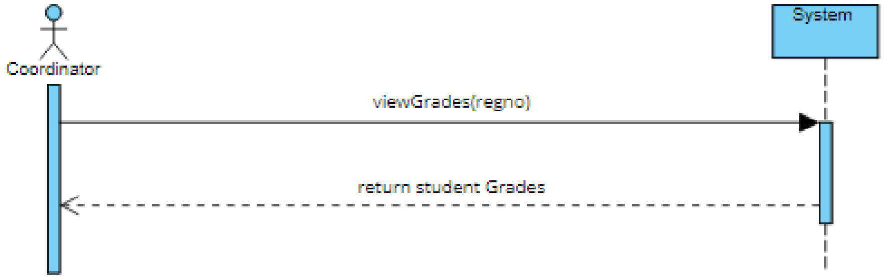


Figure 23. View Student Grades SSD

Here is the system sequence diagram in which shows interaction between the coordinator and system use case (View Student Progress SSD).

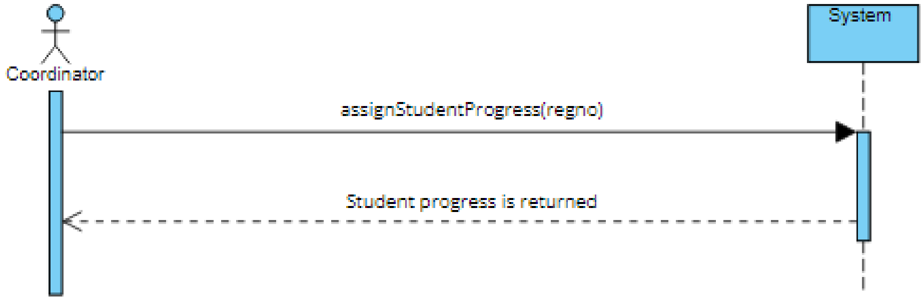


Figure 24. View Student Progress SSD

2.7. Domain Model

Here is the domain Model system of abstractions that describes selected aspects of a sphere of knowledge, influence or activity.

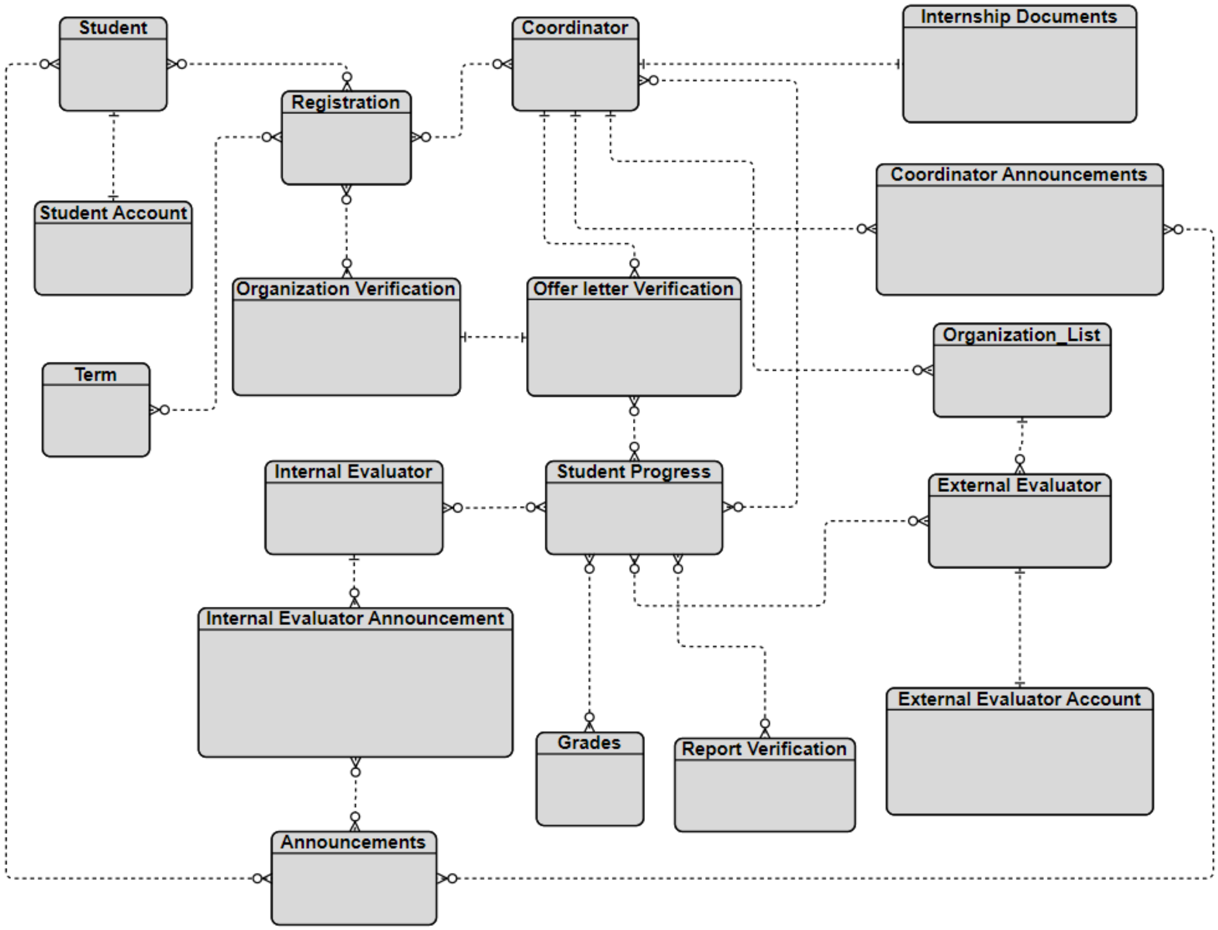


Figure 25. Domain Model

Chapter 3

System Design

Following are the system design of this project which was done by the analysis from requirements specification. It provides the details of how the system will communicate to the user efficiently and effectively. We will follow the structural view of designing the software architecture

3.1. Software Architecture

As the software architecture of this project is based on MVC (Model-View-Controller) pattern commonly used for developing user interfaces that divides the work into four interconnected elements. This is done to separate presentation layers from business layers and logical layers.

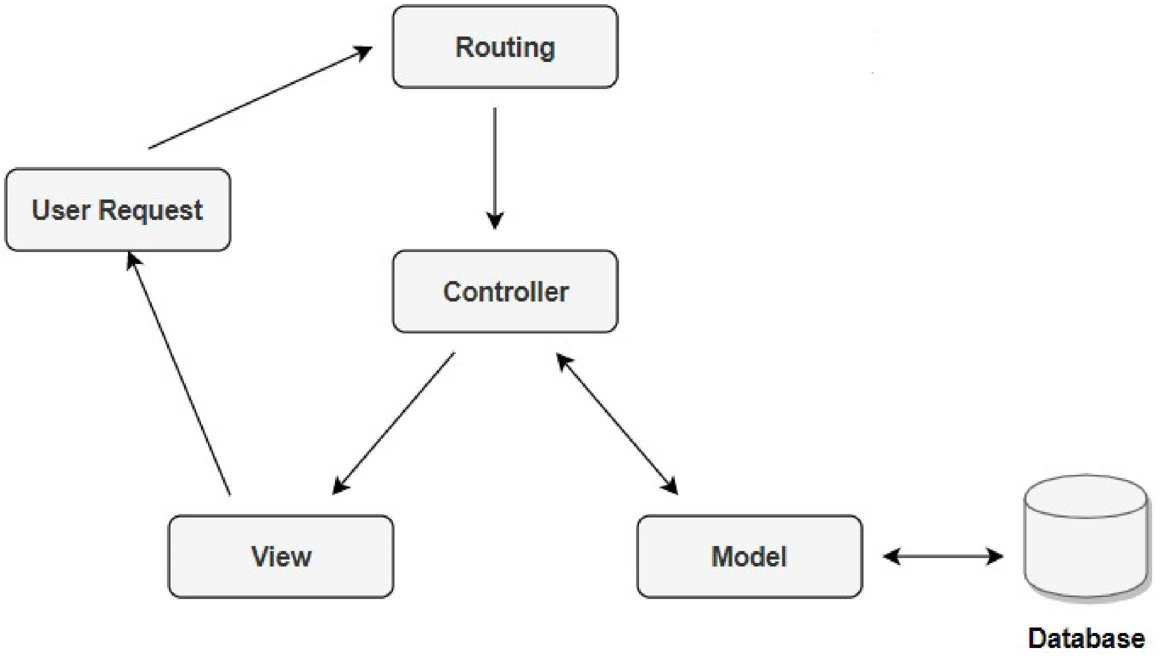


Figure 26. Software Architecture Diagram

3.2. Class Diagram

The class diagram of this project provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages, modeling the domain-specific data structure to detailed design of the target system. Following is the class diagram given below.

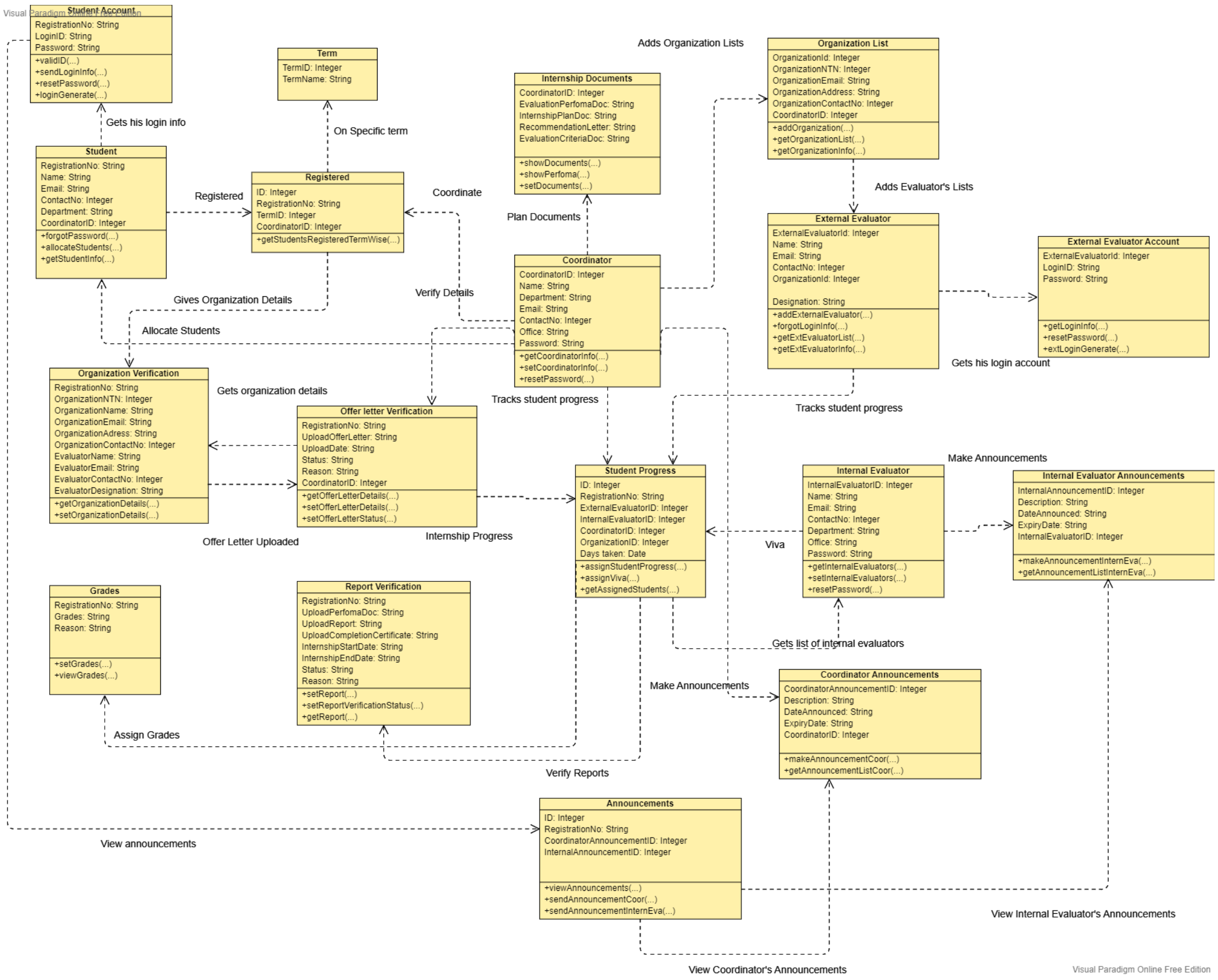


Figure 27. Class Diagram

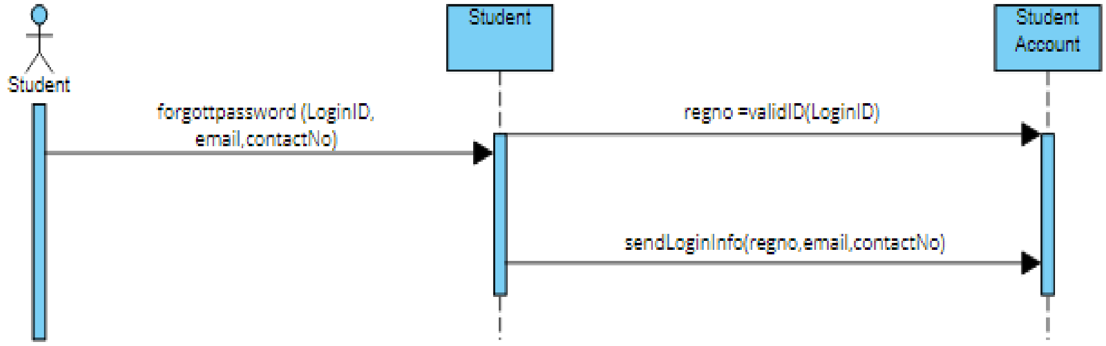
3.3. Sequence Diagram

Sequence diagrams are basically used in conjunctions with class diagrams that provides an extremely effective communication mechanism. So following are the sequence diagram in which how system behaves when multiple actors interact with the system.

3.3.1. Students

The details of students interacting with the system are given in system sequence diagram in Chapter 2. Following are the sequence diagrams in which system performs as per student's request.

Here is the sequence diagram in which shows interaction of system functionality per user request (forgot Password SD).



Student
Student
Student Account
forgottpassword (LoginID, email,contactNo)
regno =validID(LoginID)
sendLoginInfo(regno,email,contactNo)

Figure 28. Forgot Password SD

Here is the sequence diagram in which shows interaction of system functionality per user request (reset Password SD).

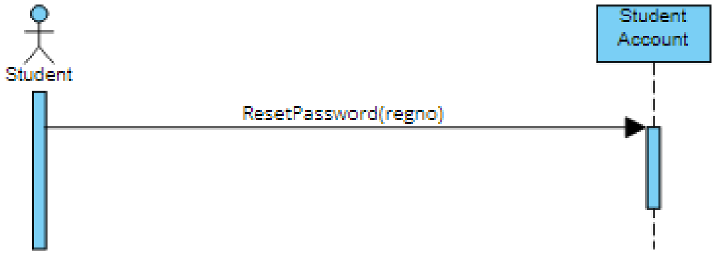


Figure 29. Reset Password SD

Here is the sequence diagram in which shows interaction of system functionality per user request (get Coordinator Info SD).

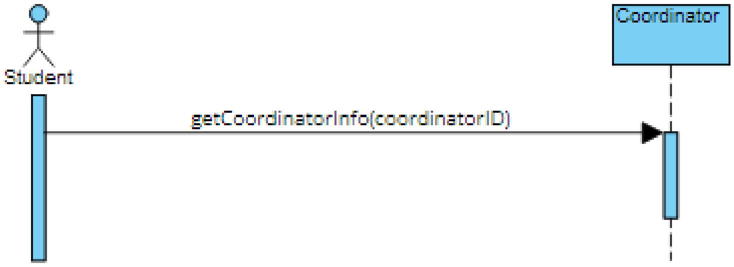


Figure 30. Get Coordinator Info SD

Here is the sequence diagram in which shows interaction of system functionality per user request (set Organization Details SD).

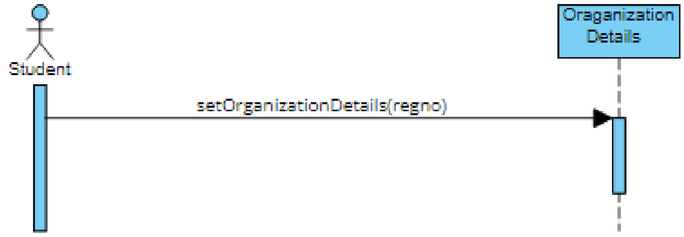


Figure 31. Set Organization Details SD

Here is the sequence diagram in which shows interaction of system functionality per user request (set Report SD).

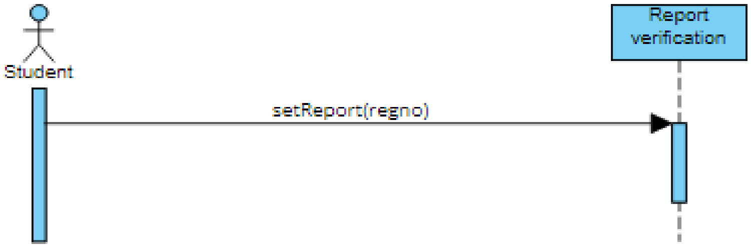


Figure 32. Set Report SD

Here is the sequence diagram in which shows interaction of system functionality per user request (view Grades SD).

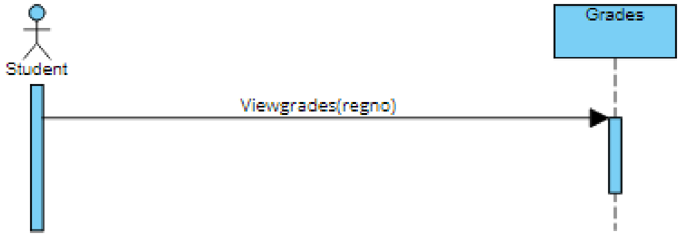


Figure 33. View Grades SD

Here is the sequence diagram in which shows interaction of system functionality per user request (get organization info SD).

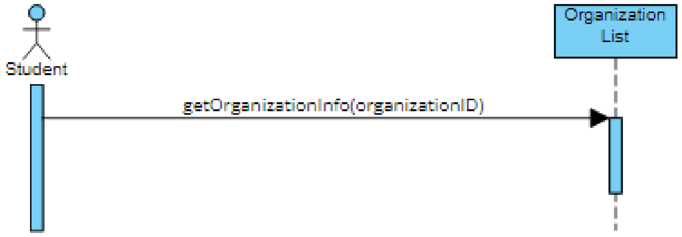


Figure 34. Get Organization Info SD

Here is the sequence diagram in which shows interaction of system functionality per user request (get external evaluator info SD).

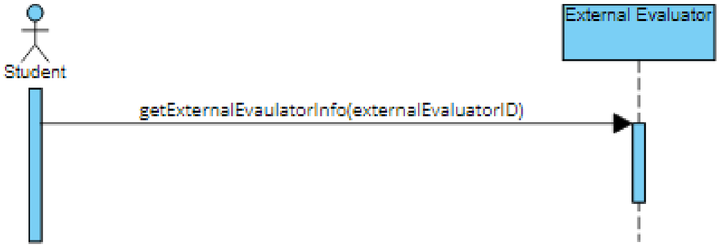


Figure 35. Get External Evaluator Info SD

Here is the sequence diagram in which shows interaction of system functionality per user request (Show Documents SD).

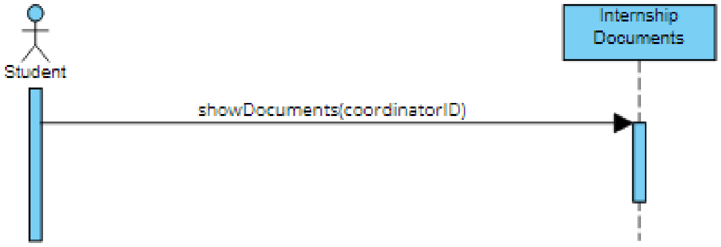


Figure 36. Show Documents SD

Here is the sequence diagram in which shows interaction of system functionality per user request (Show Documents SD).

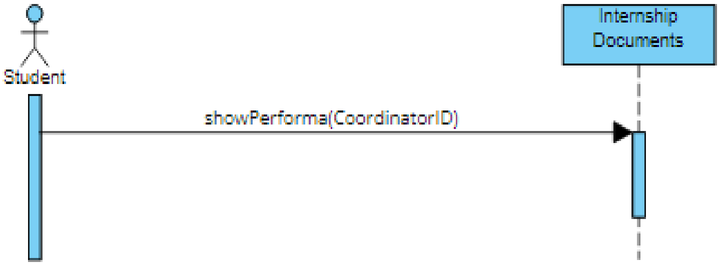


Figure 37. Show Performa SD

3.3.2. Coordinator

Here is the sequence diagram in which shows interaction of system functionality per user request (Announcements SD).

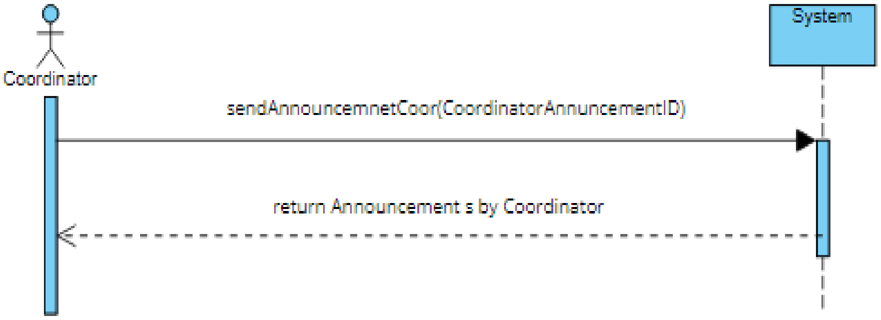

Coordinator
System
sendAnnouncemnetCoor(CoordinatorAnnuncementID)
return Announcement s by Coordinator

Figure 38. Announcements SD

Here is the sequence diagram in which shows interaction of system functionality per user request (Assign Internal Evaluator SD).

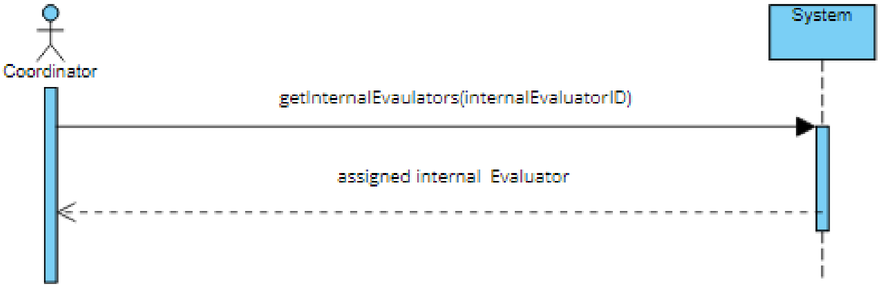

System
Coordinator
getInternalEvaulators(internalEvaluatorID)
assigned internal Evaluator

Figure 39. Assign Internal Evaluator SD

Here is the sequence diagram in which shows interaction of system functionality per user request (View Organization Information SD).

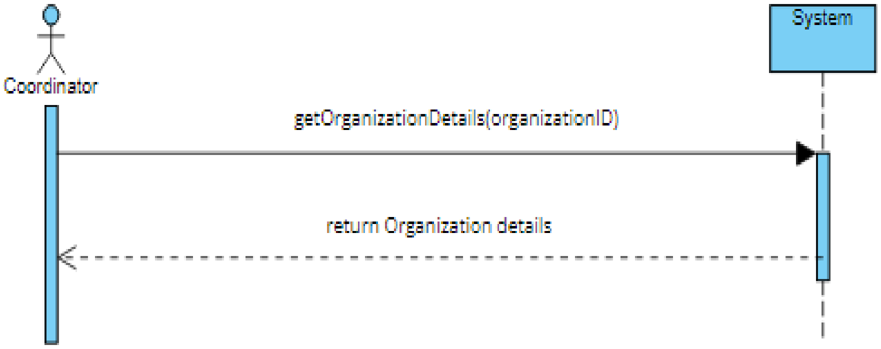

System
Coordinator
getOrganizationDetails(organizationID)
return Organization details

Figure 40.View Organization Information SD

Here is the sequence diagram in which shows interaction of system functionality per user request (Generate Login for Students SD).

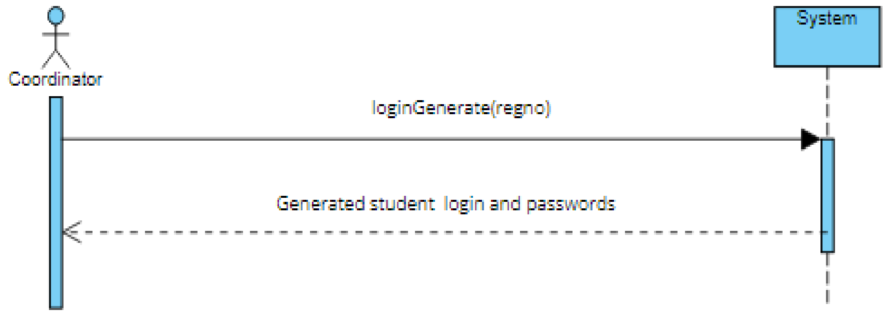


Figure 41. Generate Login for Students SD

Here is the sequence diagram in which shows interaction of system functionality per user request (Reset Password SD).

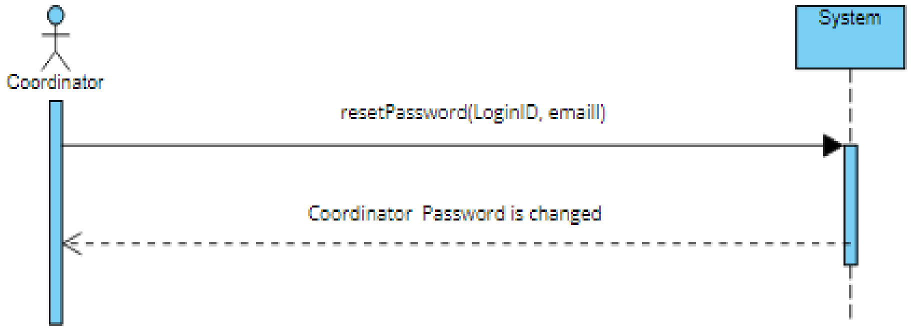

Coordinator
System
resetPassword(LoginID, emaill)
Coordinator Password is changed

Figure 42. Reset Password SD

Here is the sequence diagram in which shows interaction of system functionality per user request (View Students Term wise SD).

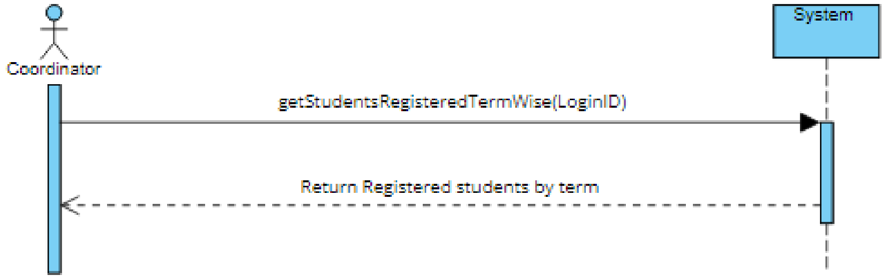

Coordinator
System
getStudentsRegisteredTermWise(LoginID)
Return Registered students by term

Figure 43. View Student Term Wise SD

Here is the sequence diagram in which shows interaction of system functionality per user request (View Student Grades SD).

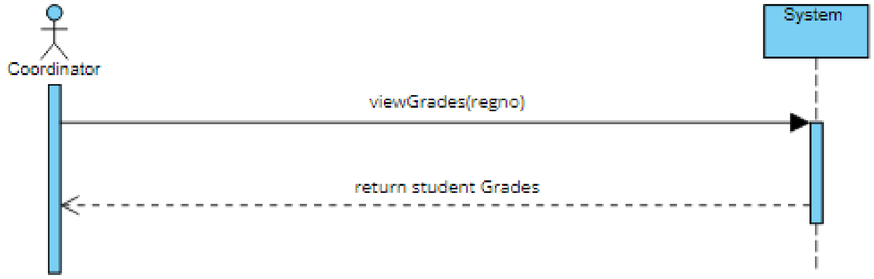


Figure 44. View Student Grades SD

Here is the sequence diagram in which shows interaction of system functionality per user request (View Student Progress SD).

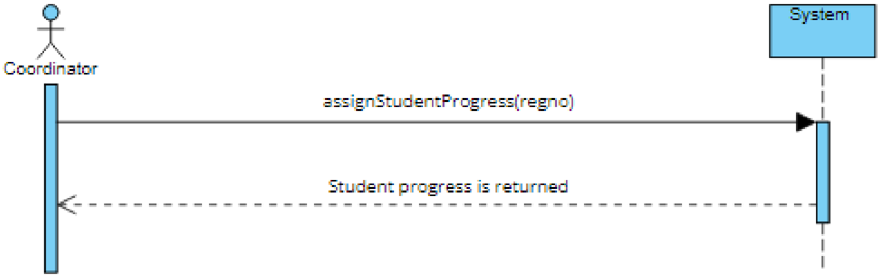


Figure 45. View Student Progress SD

3.4. Entity Relationship Diagram

Following is the conceptual representation of the data in a software system. This model is mapped in to the physical database model.

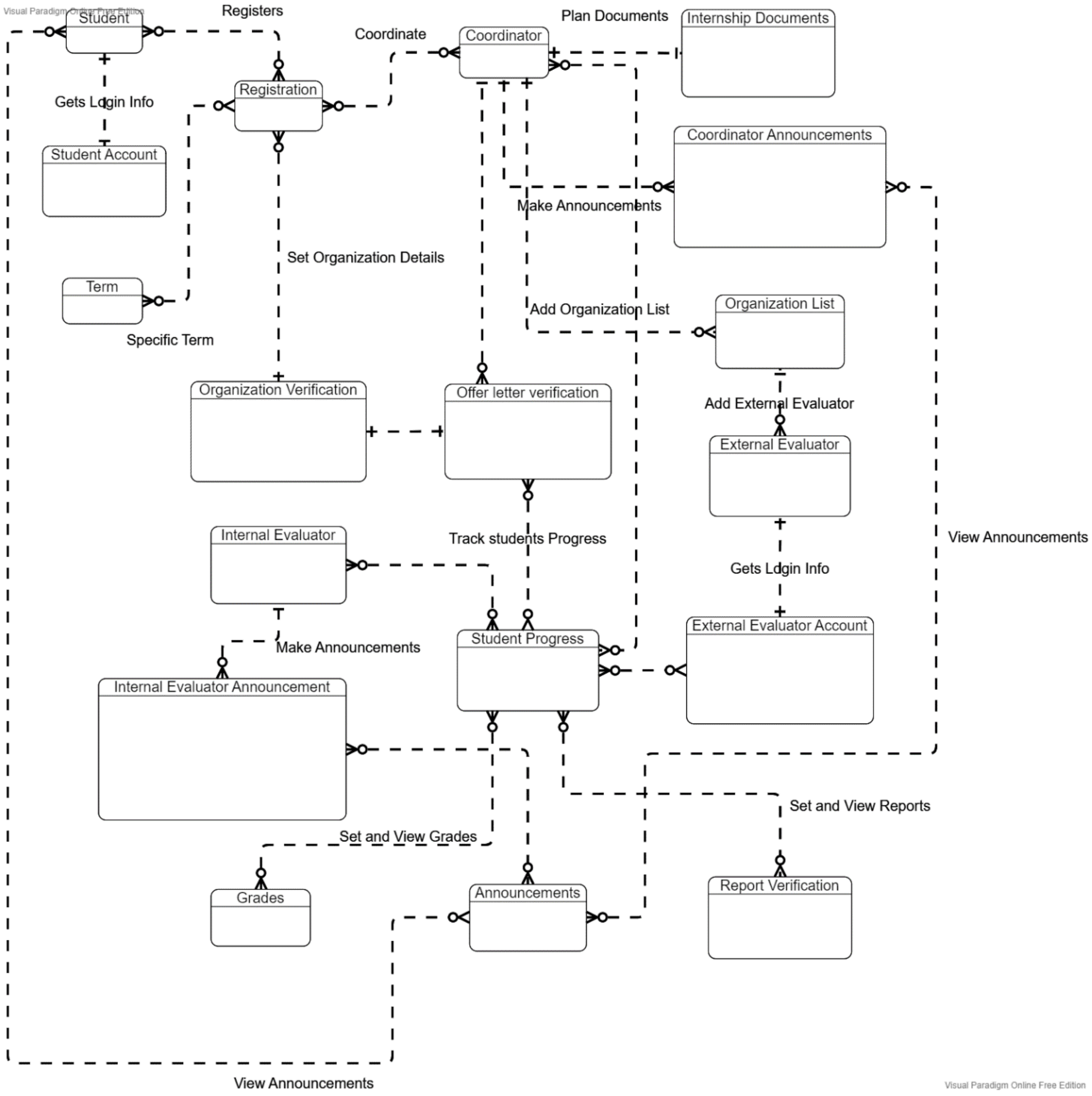
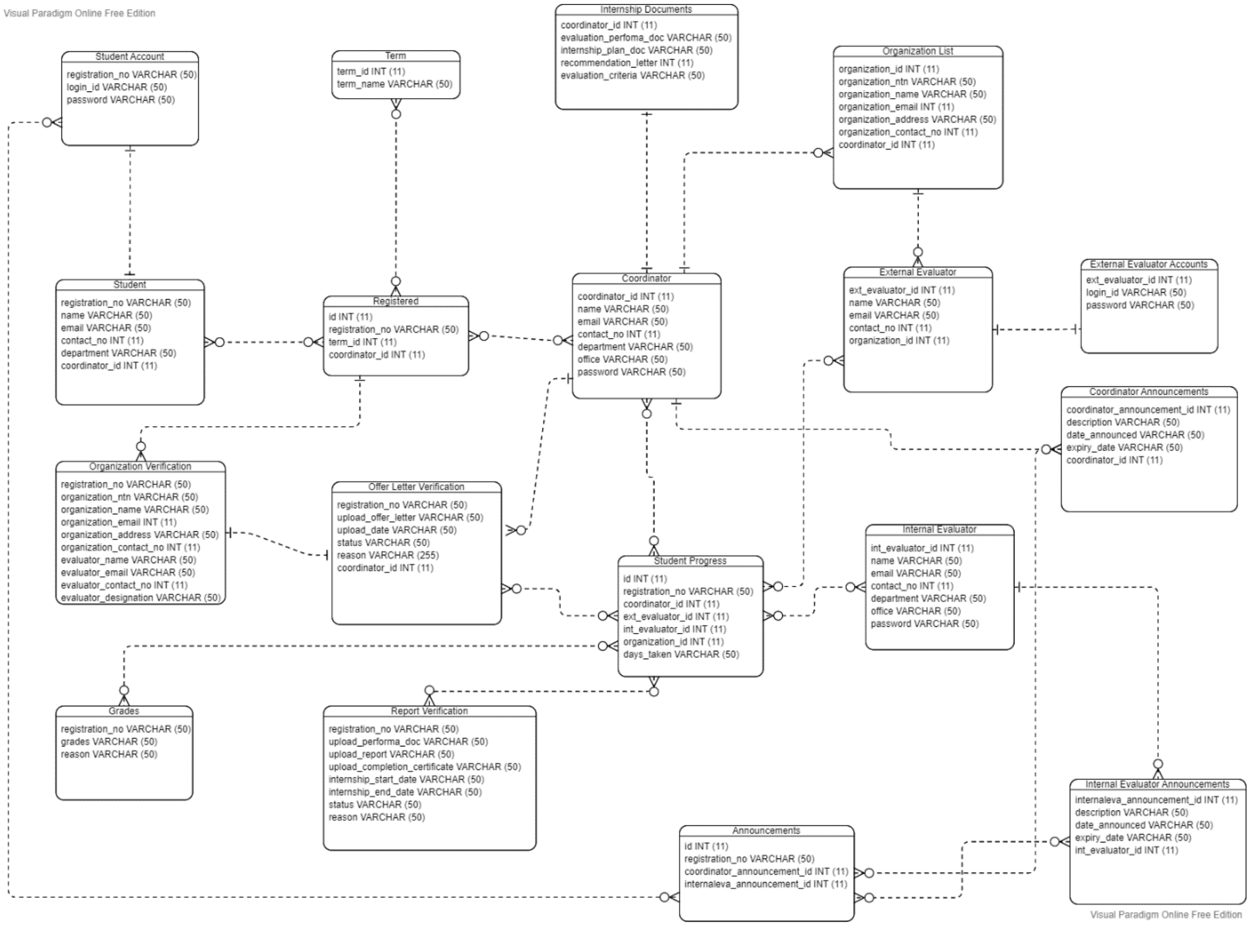


Figure 46. ER Diagram

3.5. Database Schema

Following are the logical configuration of all or part of a relational database that is related to our project. The rules of this configuration are expressed in SQL and the database management system we will use is MySQL. A database schema includes information related to primary and secondary keys, normalizing and indexing are given below.



Visual Paradigm Online Free Edition
Student Account
registration_no VARCHAR (50)
login_id VARCHAR (50)
password VARCHAR (50)
Term
term_id INT (11)
term_name VARCHAR (50)
Internship Documents
coordinator_id INT (11)
evaluation_perfoma_doc VARCHAR (50)
internship_plan_doc VARCHAR (50)
recommendation_letter INT (11)
evaluation_criteria VARCHAR (50)
Organization List
organization_id INT (11)
organization_ntn VARCHAR (50)
organization_name VARCHAR (50)
organization_email INT (11)
organization_address VARCHAR (50)
organization_contact_no INT (11)
coordinator_id INT (11)
Student
registration_no VARCHAR (50)
name VARCHAR (50)
email VARCHAR (50)
contact_no INT (11)
department VARCHAR (50)
coordinator_id INT (11)
Registered
id INT (11)
registration_no VARCHAR (50)
term_id INT (11)
coordinator_id INT (11)
Coordinator
coordinator_id INT (11)
name VARCHAR (50)
email VARCHAR (50)
contact_no INT (11)
department VARCHAR (50)
office VARCHAR (50)
password VARCHAR (50)
External Evaluator
ext_evaluator_id INT (11)
name VARCHAR (50)
email VARCHAR (50)
contact_no INT (11)
organization_id INT (11)
External Evaluator Accounts
ext_evaluator_id INT (11)
login_id VARCHAR (50)
password VARCHAR (50)
Coordinator Announcements
coordinator_announcement_id INT (11)
description VARCHAR (50)
date_announced VARCHAR (50)
expiry_date VARCHAR (50)
coordinator_id INT (11)
Organization Verification
registration_no VARCHAR (50)
organization_ntn VARCHAR (50)
organization_name VARCHAR (50)
organization_email INT (11)
organization_address VARCHAR (50)
organization_contact_no INT (11)
evaluator_name VARCHAR (50)
evaluator_email VARCHAR (50)
evaluator_contact_no INT (11)
evaluator_designation VARCHAR (50)
Offer Letter Verification
registration_no VARCHAR (50)
upload_offer_letter VARCHAR (50)
upload_date VARCHAR (50)
status VARCHAR (50)
reason VARCHAR (255)
coordinator_id INT (11)
Student Progress
id INT (11)
registration_no VARCHAR (50)
coordinator_id INT (11)
ext_evaluator_id INT (11)
int_evaluator_id INT (11)
organization_id INT (11)
days_taken VARCHAR (50)
Internal Evaluator
int_evaluator_id INT (11)
name VARCHAR (50)
email VARCHAR (50)
contact_no INT (11)
department VARCHAR (50)
office VARCHAR (50)
password VARCHAR (50)
Grades
registration_no VARCHAR (50)
grades VARCHAR (50)
reason VARCHAR (50)
Report Verification
registration_no VARCHAR (50)
upload_performa_doc VARCHAR (50)
upload_report VARCHAR (50)
upload_completion_certificate VARCHAR (50)
internship_start_date VARCHAR (50)
internship_end_date VARCHAR (50)
status VARCHAR (50)
reason VARCHAR (50)
Internal Evaluator Announcements
internaleva_announcement_id INT (11)
description VARCHAR (50)
date_announced VARCHAR (50)
expiry_date VARCHAR (50)
int_evaluator_id INT (11)
Announcements
id INT (11)
registration_no VARCHAR (50)
coordinator_announcement_id INT (11)
internaleva_announcement_id INT (11)
Visual Paradigm Online Free Edition

Figure 47. Database Schema

3.6. User Interface Design

Following are the user interface design which mainly focus on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand and use to facilitate those actions.

Here is the design in which multiple actors will interact with the system (Login UI Design).

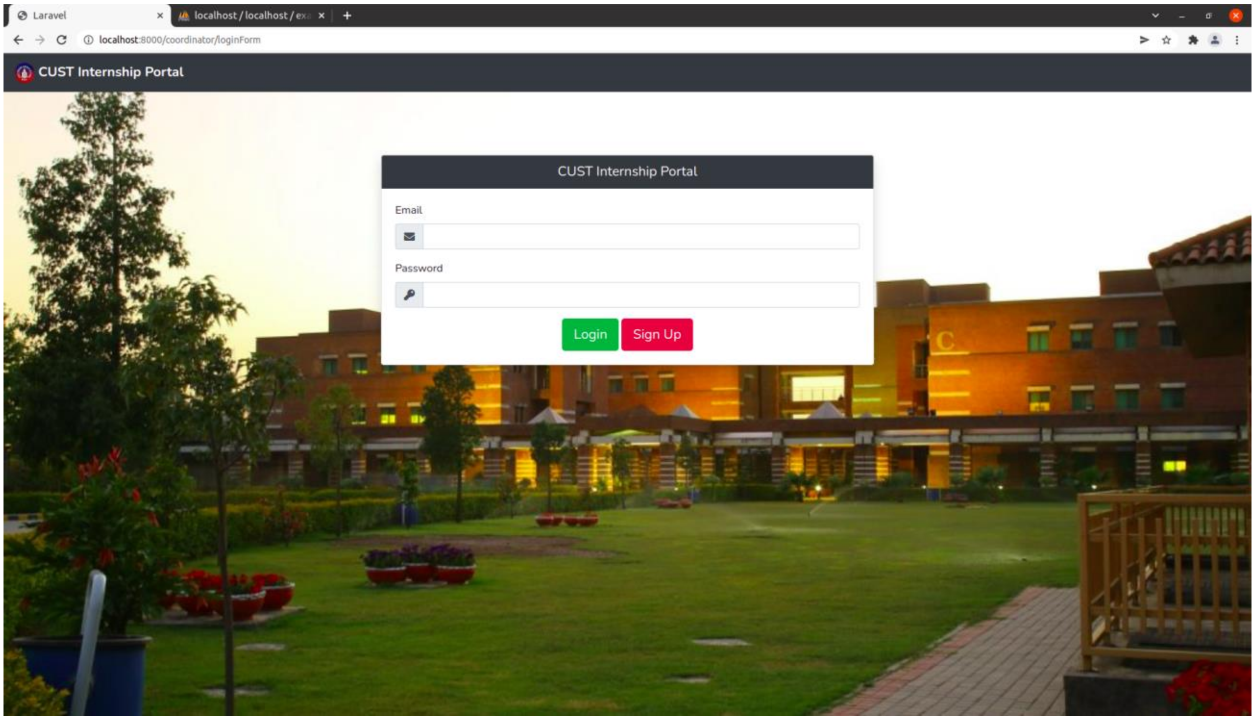
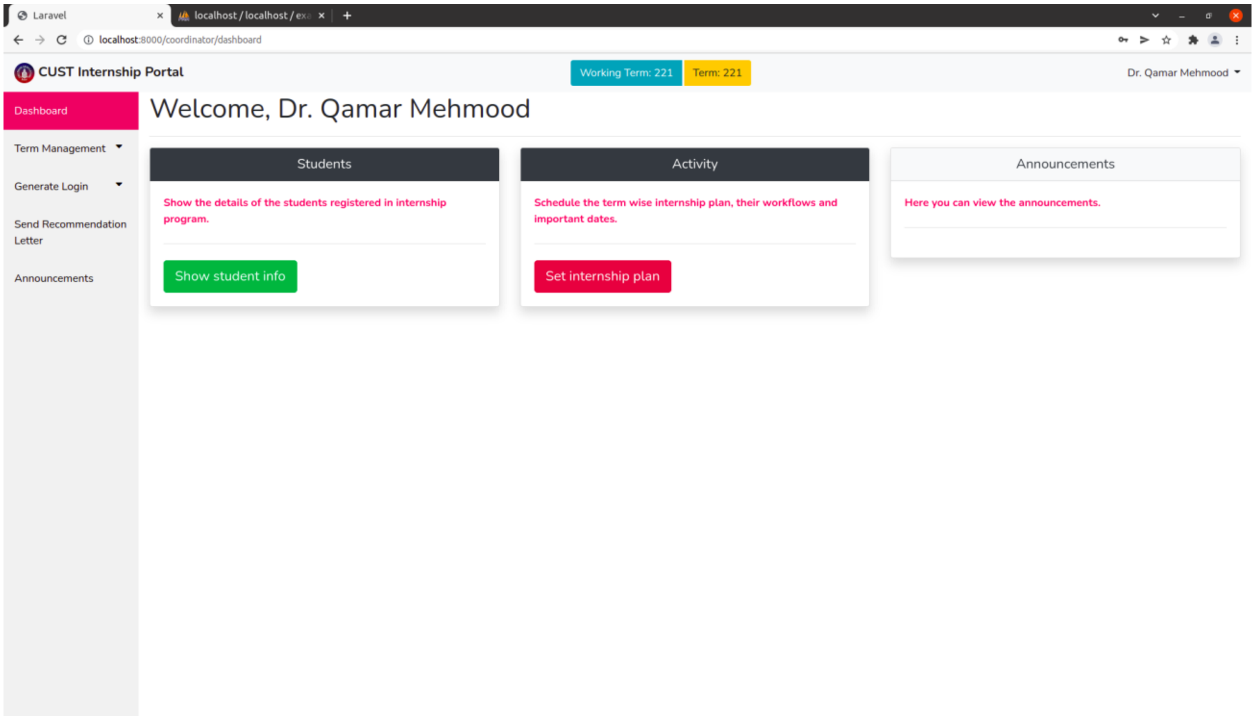


Figure 48. Login UI Design

Here is the design in which multiple actors will interact with the system (Dashboard UI Design).



Welcome, Dr. Qamar Mehmood

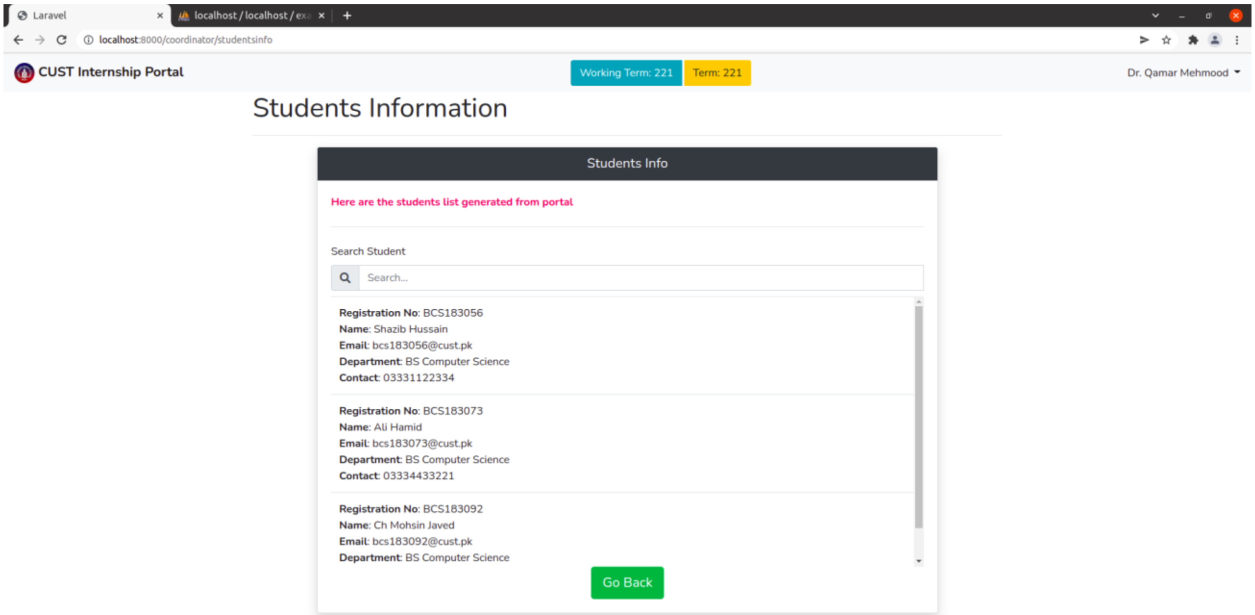
Show the details of the students registered in internship program.

Schedule the term wise internship plan, their workflows and important dates.

Here you can view the announcements.

Figure 49. Dashboard UI Design

Here is the design in which multiple actors will interact with the system (list UI Design).



Students Information

Students Info

Here are the students list generated from portal

Search Student

Search...

Registration No: BCS183056
Name: Shazib Hussain
Email: bcs183056@cust.pk
Department: BS Computer Science
Contact: 03331122334

Registration No: BCS183073
Name: Ali Hamid
Email: bcs183073@cust.pk
Department: BS Computer Science
Contact: 03334433221

Registration No: BCS183092
Name: Ch Mohsin Javed
Email: bcs183092@cust.pk
Department: BS Computer Science

Figure 50. List UI Design

Chapter 4

Software Development

4.1. Coding Standards

In this chapter will provide the details about the coding standard, we have adopted during implementation phase.

4.1.1. Indentation

Four spaces are used as the unit of indentation and space is used for tab. The indentation pattern should be consistently followed throughout.

4.1.2. Declaration

One declaration per line is used to enhance the clarity of code. The order and position of declaration is as follows:

- Variables are Private.
- Functions are public.
- Classes are public

4.1.3. Statement Standards

Each line contains at most one statement. While compound statements are statements that contain lists of statements enclosed in braces. The enclosed statements are indented one more level than the compound statement. The opening brace at the end of the line that begins the compound statement. The closing brace to begin a line and be indented to the beginning of the compound statement. Braces are used around all statements, even single statements, when they are part of a control structure, such as an if-else or for statement. A Boolean expression / function is compared to a Boolean constants.

4.1.4. Naming Conventions

Naming conventions make programs more understandable by making them easier to read. Following conventions are followed while naming a class or a member:

We used full English descriptors that accurately describe the variable, method or class. For example, use of names like Registered Students, Coordinator Information instead of names like x1, y1.

4.2. Development Environment

4.2.1. Visual Studio

We used visual studio IDE for our project, because it is standard platform to make this software. It is a mature, powerful and versatile IDE that is used across the globe to create websites, desktop software and games in a number of popular programming languages, like C#, C++ and php.

4.2.2. Laravel Framework

Laravel is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern and based on Symfony. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance.

4.3. Database Management System

4.3.1. MySQL

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and

WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

4.4. Software Description

4.4.1. Students

The modules that are implemented under this category are as follows:

- **Account Login**

Where student can login the account to access the internship portal.

```
**GET method**

public function loginForm(){

        return view('Student.login');

    }


**POST method**

public function login(Request $request){

        $validated = $request->validate([

            'loginid' => 'required|max:100',

            'password' => 'required|max:100',

        ],

        [

            'loginid.required' => 'Please enter login',

            'password.required' => 'Please enter password',

        ]);


        $studentaccount = StudentAccount::where('login_id', $request-
>loginid)->first();


        if ($studentaccount){
            if ($request->password == $studentaccount->password){

                session(['registration_no' => $studentaccount-
>registration_no]);

                return Redirect('/student/dashboard');

            }

        }


        return Redirect()->back();

    }
```

- **Download Recommendation Letter**

Where student can download the recommendation letter to continue internship progress.

```
**POST method**

public function downloadrecletter(){

        $registration_no = session('registration_no');

        if ($registration_no){

            $studentdocs = StudentDocs::where('registration_no',
$registration_no)->first();


            if ($studentdocs){

                return response()->download($studentdocs-
>recommendation_letter);

            } else {

                return Redirect()->back();

            }


        } else {
            return Redirect("/student/loginForm");

        }

    }
```

Change of login ID and password

Where student can change login id and password for security premises.

```
**GET method**

public function accountsettings(){

        $registration_no = session('registration_no');

        if ($registration_no){

            $student  =  Student::where('registration_no',  $registration_no)-
>first();

            $term = Term::all()->last();

            return view('Student.accountinfo', compact('student', 'term'));

        } else {

            return Redirect("/student/loginForm");

        }

    }


**POST method for login id**

public function setloginid(Request $request){

        $registration_no = session('registration_no');

        if ($registration_no){

            $validated = $request->validate([

                'curloginid' => 'required|max:100',

                'newloginid' => 'required|max:100',

                'confirmloginid' => 'required|max:100',
            ],
            [
                'curloginid.required' => 'Please enter current login id',
                'newloginid.required' => 'Please enter new login id',
                'confirmloginid.required' => 'Please enter confirm login id',
            ]);

            $studentaccount = StudentAccount::where('registration_no',
$registration_no)->first();
            if($studentaccount->login_id == $request->curloginid){
                if($request->newloginid == $request->confirmloginid){
                    $studentaccount->where('registration_no',
$registration_no)->update([
                        'login_id' => $request->confirmloginid,
                    ]);
                } else {
                    return Redirect()->back();
                }
            } else {
                return Redirect()->back();
            }

            return Redirect()->back();
        } else {
            return Redirect("/student/loginForm");
        }
    }
```

```
**POST method for password**

public function setpassword(Request $request){

        $registration_no = session('registration_no');

        if ($registration_no){

            $validated = $request->validate([

                'curpassword' => 'required|max:100',

                'newpassword' => 'required|max:100',

                'confirmpassword' => 'required|max:100',

            ],

            [

                'curpassword.required' => 'Please enter current password',

                'newpassword.required' => 'Please enter new password',

                'confirmpassword.required' => 'Please enter confirm 
password',

            ]);


            $studentaccount = StudentAccount::where('registration_no', 
$registration_no)->first();

            if($studentaccount->password == $request->curpassword){

                if($request->newpassword == $request->confirmpassword){

                    $studentaccount->where('registration_no', 
$registration_no)->update([

                        'password' => $request->confirmpassword,

                    ]);

                } else {

                    return Redirect()->back();

                }

            } else {

                return Redirect()->back();

            }
            return Redirect()->back();
        } else {
            return Redirect("/student/loginForm");
        }
    }
```

- **Download internship plan**

Where student can download the internship plan for guidelines.

```
**POST method**
public function downloadinternshipplan(){
        $registration_no = session('registration_no');
        if ($registration_no){
            $studentdocs = StudentDocs::where('registration_no', 
$registration_no)->first();

            if($studentdocs){
                return response()->download($studentdocs->internship_plan);
            } else {
                return Redirect()->back();
            }

        } else {
            return Redirect("/student/loginForm");
        }
    }
```

4.4.2. Coordinators

The modules that are implemented under this category are as follows:

- **Login Account**

Where coordinator can login the account to access the internship portal.

```
**GET method**

public function loginForm(){

        return view('Coordinator.login');

    }


**POST method**

public function login(Request $request){

        $validated = $request->validate([

            'email' => 'required|max:100',

            'password' => 'required|max:100',

        ],

        [

            'email.required' => 'Please enter email',

            'password.required' => 'Please enter password',

        ]);


        $coordinator = Coordinator::where('email', $request->email)->first();


        if ($coordinator){

            if (Hash::check($request->password, $coordinator->password)){

                session([

                    'id' => $coordinator->id,
                    'term' => Term::all()->last()->term_name,

                    'output' => null

                ]);

                return Redirect('/coordinator/dashboard');

            }

        }


        return Redirect()->back();

    }
```

- **Send recommendation letter**

Where coordinator can send recommendation letter through portal to students via email.

```
**GET method**

public function sendLetter(){

        $id = session('id');

        if ($id){

            $student = TermRegistered::where('term_name', session('term'))-
>distinct()->get(['registration_no']);

            $root = TermRegistered::where([['term_name', session('term')],
['coordinator_id', session('id')]])->first();

            $term = Term::all()->last();

            return view('Coordinator.sendletter', compact('student', 'root',
'term'));

        } else {

            return Redirect('/coordinator/loginForm');

        }

    }
```

```
**POST method**
public function letter(Request $request){
        $id = session('id');
        if ($id){
            $validated = $request->validate([
                'regno' => 'required|max:100',
            ],
            [
                'regno.required' => 'Please select registration no',
            ]);

            foreach ($request->regno as $r){
                $rendererName = Settings::PDF_RENDERER_TCPDF;
                $renderedLibraryPath = "../vendor/tecnickcom/tcpdf";
                Settings::setPdfRenderer($rendererName,
$renderedLibraryPath);

                $student = Student::where('registration_no', $r)->first();

                $templateProcessor = new
TemplateProcessor('templates/recLetter.docx');
                $templateProcessor->setValue('description', $request-
>description);
                $templateProcessor->setValue('registration_no', $student-
>registration_no);
                $templateProcessor->setValue('name', $student->name);
                $templateProcessor->setValue('department', $student-
>department);
                $templateProcessor->setValue('contactno', $student-
>contactno);
                $templateProcessor->setValue('date', Carbon::now()-
>toFormattedDateString());
```

```
                $templateProcessor->saveAs($student-
>registration_no.'.docx');


                $objReader = PhpWord\IOFactory::createReader();
                $pdfWord = $objReader->load($student-
>registration_no.'.docx');
                $objWriter = PhpWord\IOFactory::createWriter($pdfWord,
'PDF');
                $file = $student->registration_no.".pdf";
                $objWriter->save($file);


                $pdffile = "files/".$file;
                rename($file, $pdffile);
                unlink($student->registration_no.'.docx');


                Mail::to($student->email)->send(new LetterMail($student,
$pdffile));


                $studentdocs = StudentDocs::where('registration_no', $r)-
>first();


                if($studentdocs){
                    $studentdocs->where('registration_no', $r)->update([
                        'recommendation_letter' => $pdffile,
                        'internship_plan' =>
"files/InternshipSummer2021Plan.pdf"
                    ]);
                } else {
                    $studentdocs = new StudentDocs;
                    $studentdocs->registration_no = $student-
>registration_no;
                    $studentdocs->recommendation_letter = $pdffile;
                    $studentdocs->internship_plan = 
"files/InternshipSummer2021Plan.pdf";

                    $studentdocs->save();

                }

            }

            return Redirect()->back();

        } else {

            return Redirect('/coordinator/loginForm');

        }

    }
```

- **Send login account**

Where coordinator can send login account information through portal to students via email.

```
**POST method**

public function studentlogininfo(){

        $id = session('id');

        if ($id){

            $studentaccount = StudentAccount::all();

            if(count($studentaccount) > 0){


            } else {

                $student = Student::all();

                foreach ($student as $s){

                    $studentaccount = new StudentAccount;

                    $studentaccount->registration_no = $s->registration_no;

                    $studentaccount->login_id = $this-
>generate_random_letters();

                    $studentaccount->password = $this-
>generate_letter_pass();
```

```
                    $studentaccount->save();
                }
                foreach ($student as $s){
                    Mail::to($s->email)->send(new LoginInfoMail($s));
                }
            }

            return Redirect()->back();
        } else {
            return Redirect('/coordinator/loginForm');
        }
    }
```

Extract the list of students

Where coordinator can extract the list of students term wise.

```
**GET method**
public function studentsinfo(){
        $id = session('id');
        if ($id){
            $root = TermRegistered::where([['term_name', session('term')],
['coordinator_id', session('id')]])->first();
            $student = TermRegistered::where('term_name', session('term'))-
>distinct()->get(['registration_no']);
            $term = Term::all()->last();
            return view('Coordinator.studentsinfo', compact('root',
'student', 'term'));
        } else {
            return Redirect('/coordinator/loginForm');
        }
```

```
    }
```

- **Change account password**

Where coordinator can extract the list of students term wise.

```
**GET method**

public function changePassword(){

        $id = session('id');

        if ($id){

            $root = TermRegistered::where([['term_name', session('term')],
['coordinator_id', session('id')]])->first();

            $term = Term::all()->last();

            return view('Coordinator.changepassword', compact('root',
'term'));

        } else {

            return Redirect('/coordinator/loginForm');

        }

    }


**POST method**

public function password(Request $request){

        $id = session('id');

        if ($id){

            $validated = $request->validate([

                'curpassword' => 'required|max:100',

                'newpassword' => 'required|max:100',

                'confirmpassword' => 'required|max:100',

            ],

            [
                'curpassword.required' => 'Please enter current password',
                'newpassword.required' => 'Please enter new password',
                'confirmpassword.required' => 'Please enter confirm
password',
            ]);

            $coordinator = Coordinator::where('id', $id)->first();

            if ($coordinator){
                if (Hash::check($request->curpassword, $coordinator-
>password)){
                    if ($request->newpassword == $request->confirmpassword){
                        $coordinator->password = Hash::make($request-
>newpassword);
                        $coordinator->save();
                        return Redirect('/coordinator/dashboard');
                    } else {
                        return Redirect()->back();
                    }
                } else {
                    return Redirect()->back();
                }
            }

        } else {
            return Redirect('/coordinator/loginForm');
        }
    }
```

Chapter 5

Software Testing

Software Testing is the most crucial part of Software Development Process. It is the investigation or evaluation of a software component, improving them, and finding bugs and defects. Testing is usually done by executing a system in such a way that it identifies any gaps, errors, or missing requirements in contrary to the actual requirements.

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 black box testing in our testing phase. We used black box testing because it is very efficient and it contains following benefits. Black box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. Black box unit testing is used in our project. Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use:

- Black box tests are reproducible.
- Find software bugs early.
- Facilitates change.
- The environment the program is running is also tested.
- The invested effort can be used multiple times.
- More effective on larger units of code than glass box testing.

Tester needs no knowledge of implementation, including specific programming languages.

5.2. Testing Environment

We tested manually here. Create test cases and check the system according to test cases.

5.3. Test Cases

5.3.1. Coordinator

5.3.1.1. Login

Table 23.Test Case 1

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Coordinator Login	Test ID: 1
Version: 1	Test Type: Unit Testing
Input: Username: aleeri@cust. Password: 123cdd45678 Click Button: Login	
Expected Result: Login Failed, Invalid input	
Actual Result: Passed	

Table 24. Test Case 2

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Coordinator Login	Test ID: 2
Version: 1	Test Type: Unit Testing
Input: Username: arli@cust.edu Password: 12345678 Click Button: Login	
Expected Result: Login Failed, Incorrect Username	
Actual Result: Passed	

Table 25. Test Case 3

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Coordinator Login	Test ID: 3
Version: 1	Test Type: Unit Testing
Input: Username: ali@gmail.com Password: 1888345678 Click Button: Login	
Expected Result: Login Failed, Incorrect Password	
Actual Result: Passed	

Table 26. Test Case 4

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Coordinator Login	Test ID: 4
Version: 1	Test Type: Unit Testing
Input: Username: ali@gmail.com Password: 12345678 Click Button: Login	
Expected Result: Login Successful, Redirecting to Homepage	
Actual Result: Passed	

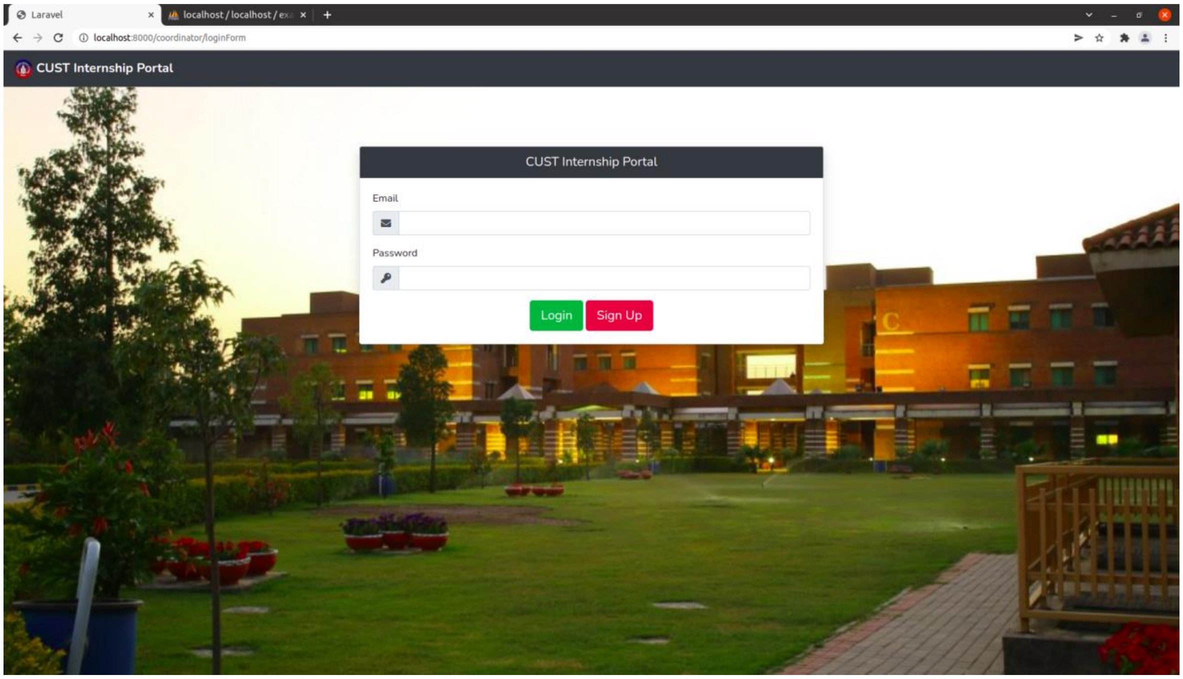
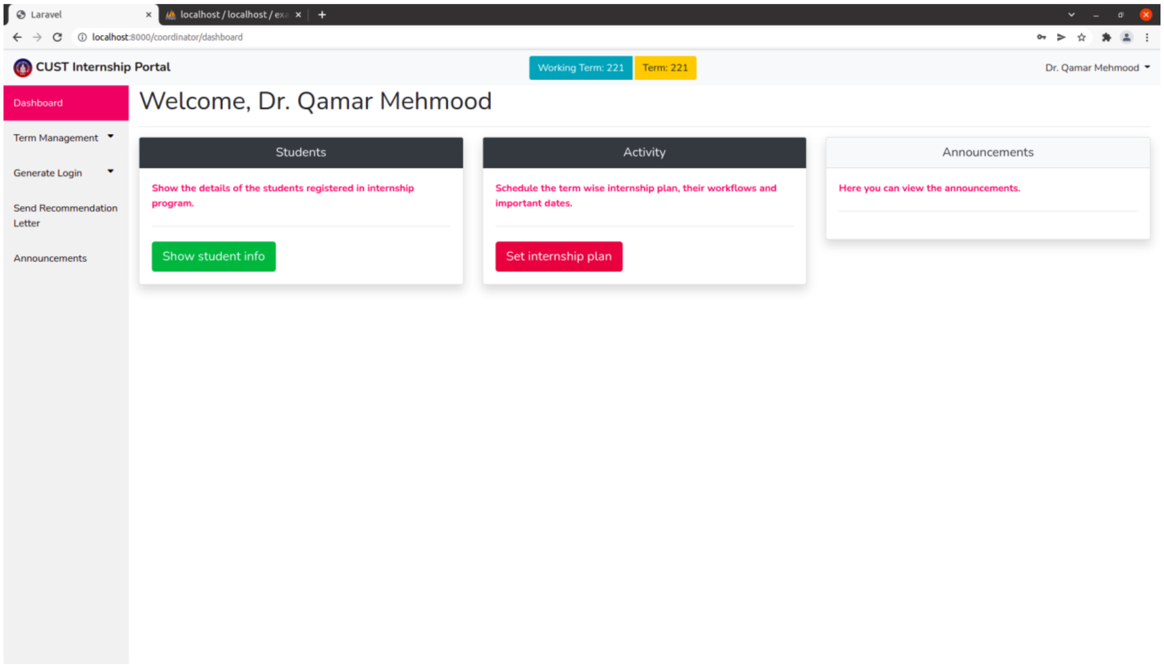


Figure 51. Login Screen



Welcome, Dr. Qamar Mehmood

Show the details of the students registered in internship program.

Schedule the term wise internship plan, their workflows and important dates.

Here you can view the announcements.

***Figure 52.* Coordinator Dashboard**

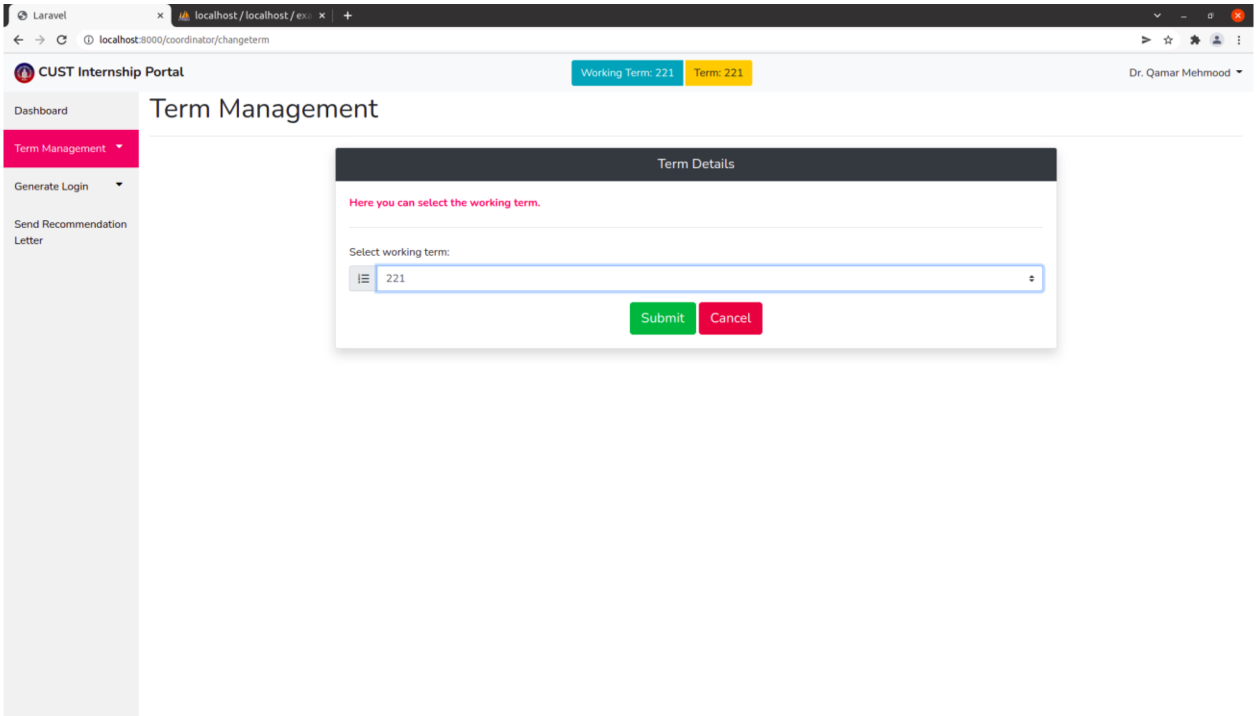
5.3.1.2. Changing Working Term

Table 27.Test Case 5

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: View Students term Wise	Test ID: 5
Version: 1	Test Type: Unit Testing
Input: Click Button: Change term	
Expected Result: If Coordinator is logged in, Redirecting to Term details page	
Actual Result: Passed	

Table 28. Test Case 6

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: View Students term Wise	Test ID: 6
Version: 1	Test Type: Unit Testing
Input: Click Drop Down: Select working term	
Expected Result: If Coordinator is logged in, Changed working term	
Actual Result: Passed	



localhost:8000/coordinator/changeterm

Here you can select the working term.

Select working term: 221

Figure 53. Working term View

5.3.1.3. Generate Student Login

Table 29. Test Case 7

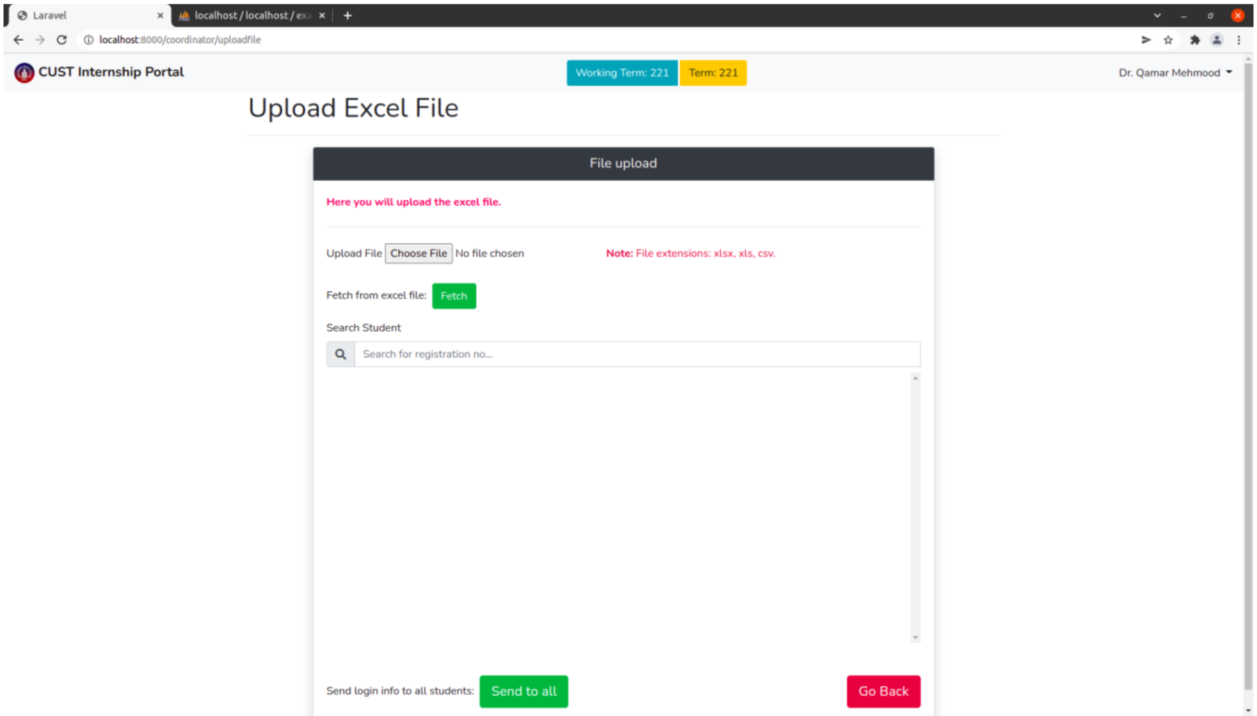
Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Generate Student Login	Test ID: 7
Version: 1	Test Type: Unit Testing
Input: Click Button : Fetch from portal	
Expected Result: If Coordinator is logged in, get the list of registered students from portal	
Actual Result: Passed	

Table 30.Test Case 8

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Generate Student Login	Test ID: 8
Version: 1	Test Type: Unit Testing
Input: Click Button : Upload File File Type: .docx	
Expected Result: File type is not Supported, Upload failed	
Actual Result: Passed	

Table 31. Test Case 9

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Generate Student Login	Test ID: 9
Version: 1	Test Type: Unit Testing
Input: Click Button : Upload File File Type: .csv, xls, xlsx	
Expected Result: File type is Supported, Upload failed	
Actual Result: Passed	



Upload Excel File

File upload

Here you will upload the excel file.

Upload File: No file chosen

Note: File extensions: xlsx, xls, csv.

Fetch from excel file:

Search Student

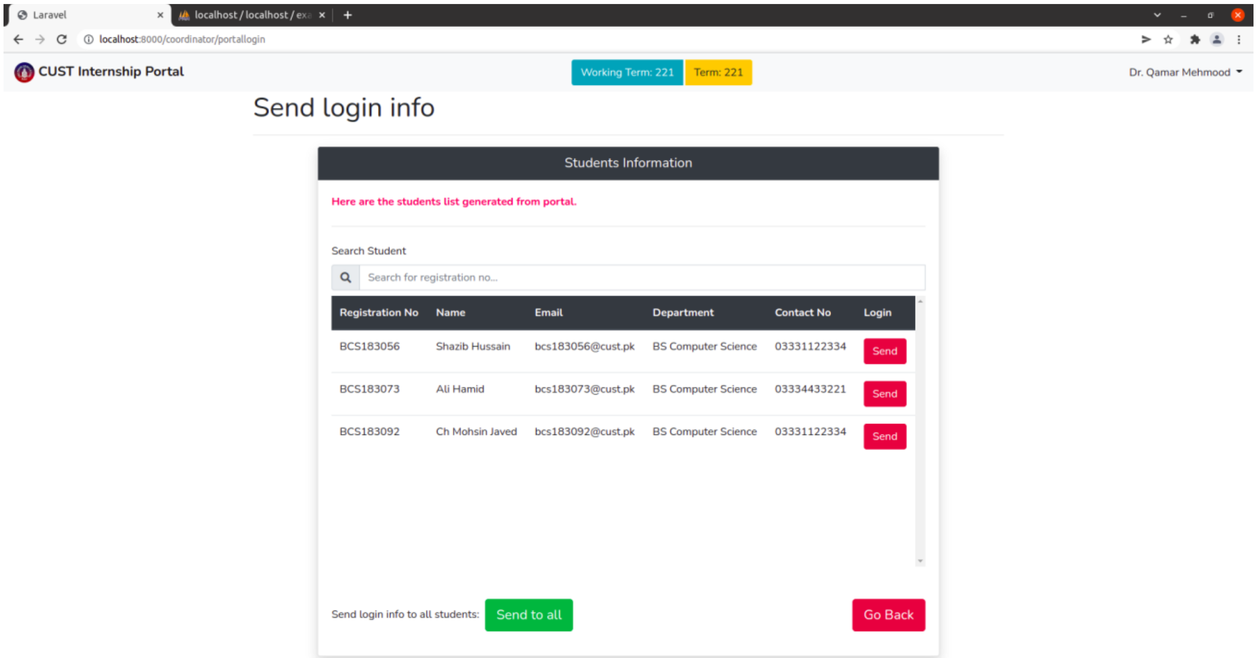
Search for registration no...

Send login info to all students:

Figure 54. Fetch from portal

Table 32. Test Case 10

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Generate Student Login	Test ID: 10
Version: 1	Test Type: Unit Testing
Input: Click Button : Select Students or Select all	
Expected Result: If Coordinator is logged in, Send students email of username and password	
Actual Result: Passed	



Send login info

Students Information

Here are the students list generated from portal.

Search Student

Search for registration no...

Registration No	Name	Email	Department	Contact No	Login
BCS183056	Shazib Hussain	bcs183056@cust.pk	BS Computer Science	03331122334	Send
BCS183073	Ali Hamid	bcs183073@cust.pk	BS Computer Science	03334433221	Send
BCS183092	Ch Mohsin Javed	bcs183092@cust.pk	BS Computer Science	03331122334	Send

Send login info to all students: Send to all

Go Back

Figure 55. Generate Student login

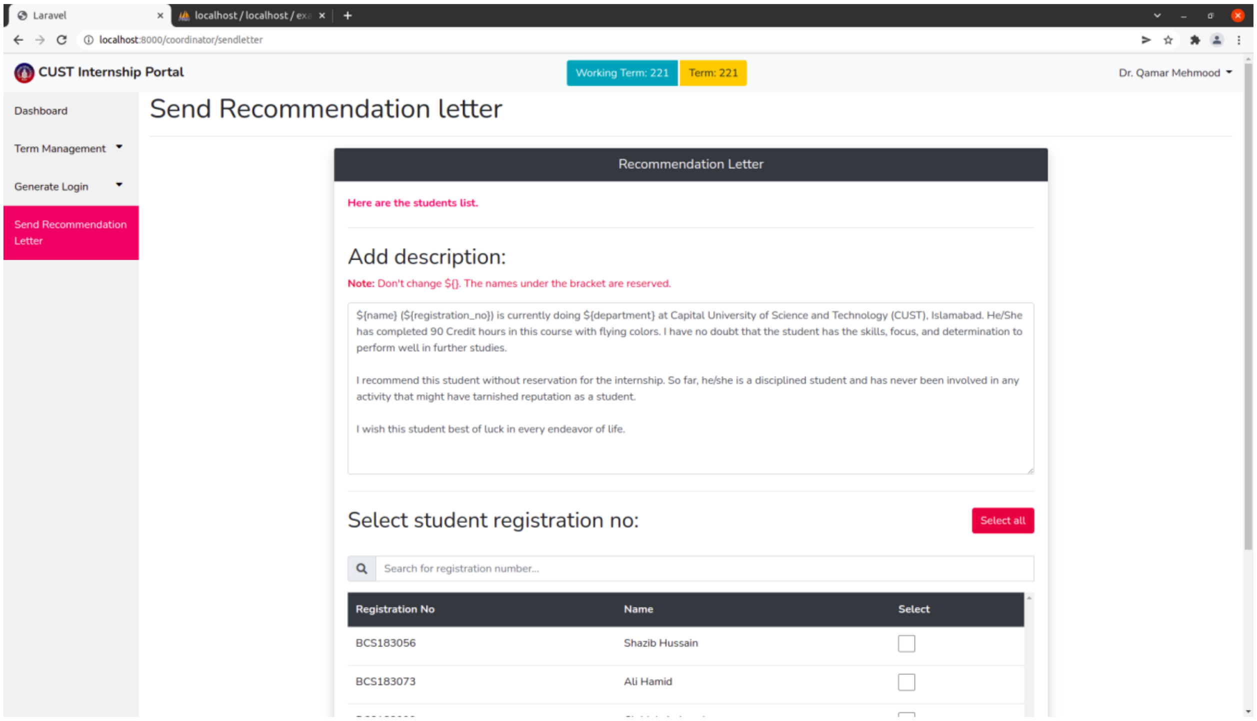
5.3.1.4. Send Recommendation Letter

Table 33. Test Case 11

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Send Recommendation letter	Test ID: 11
Version: 1	Test Type: Unit Testing
Input: Click Button : Send Recommendation letter	
Expected Result: If Coordinator is logged in, Redirect to Recommendation letter page	
Actual Result: Passed	

Table 34.Test Case 12

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Send Recommendation letter	Test ID: 12
Version: 1	Test Type: Unit Testing
Input: Click Button : Select Students or Select all	
Expected Result: If Coordinator is logged in, Send Recommendation letters pdf to student portal and email	
Actual Result: Passed	


Laravel
localhost / localhost / exa
localhost:8000/coordinator/sendletter
CUST Internship Portal
Working Term: 221
Term: 221
Dr. Qamar Mehmood
Dashboard
Term Management
Generate Login
Send Recommendation Letter
Select all
Search for registration number...

Send Recommendation letter

Recommendation Letter

Here are the students list.

Add description:

Note: Don't change ${}. The names under the bracket are reserved.

${name} (${registration_no}) is currently doing ${department} at Capital University of Science and Technology (CUST), Islamabad. He/She has completed 90 Credit hours in this course with flying colors. I have no doubt that the student has the skills, focus, and determination to perform well in further studies.

I recommend this student without reservation for the internship. So far, he/she is a disciplined student and has never been involved in any activity that might have tarnished reputation as a student.

I wish this student best of luck in every endeavor of life.

Select student registration no:

Registration No	Name	Select
BCS183056	Shazib Hussain	□
BCS183073	Ali Hamid	□

Figure 56. Send Recommendation letter

5.3.1.5. Password Change

Table 35. Test Case 13

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Password	Test ID: 13
Version: 1	Test Type: Unit Testing
Input: Click Button : Change Password Setting	
Expected Result: If Coordinator is logged in, Redirect to Password changing settings page	
Actual Result: Passed	

Table 36.Test Case 14

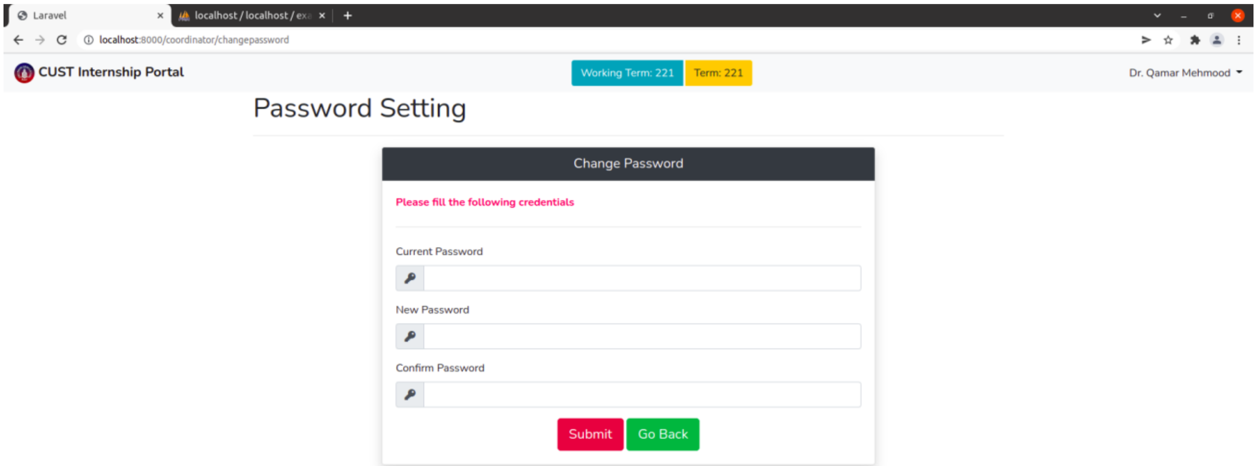
Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Password	Test ID: 14
Version: 2	Test Type: Unit Testing
Input: Current Password; *********** New Password; *********** Confirm Password; *********** Click Button : Submit	
Expected Result: Password changing failed, Incorrect Current Password	
Actual Result: Passed	

Table 37.Test Case 15

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Password	Test ID: 15
Version: 1	Test Type: Unit Testing
Input: Current Password; *********** New Password; *********** Confirm Password; *********** Click Button : Submit	
Expected Result: Password changing failed, New password does not match with confirm password	
Actual Result: Passed	

Table 38.Test Case 16

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Password	Test ID: 16
Version: 1	Test Type: Unit Testing
Input: Current Password; *********** New Password; *********** Confirm Password; *********** Click Button : Submit	
Expected Result: Password Changing Successful , Password changed	
Actual Result: Passed	



Password Setting

Change Password

Please fill the following credentials

Current Password

New Password

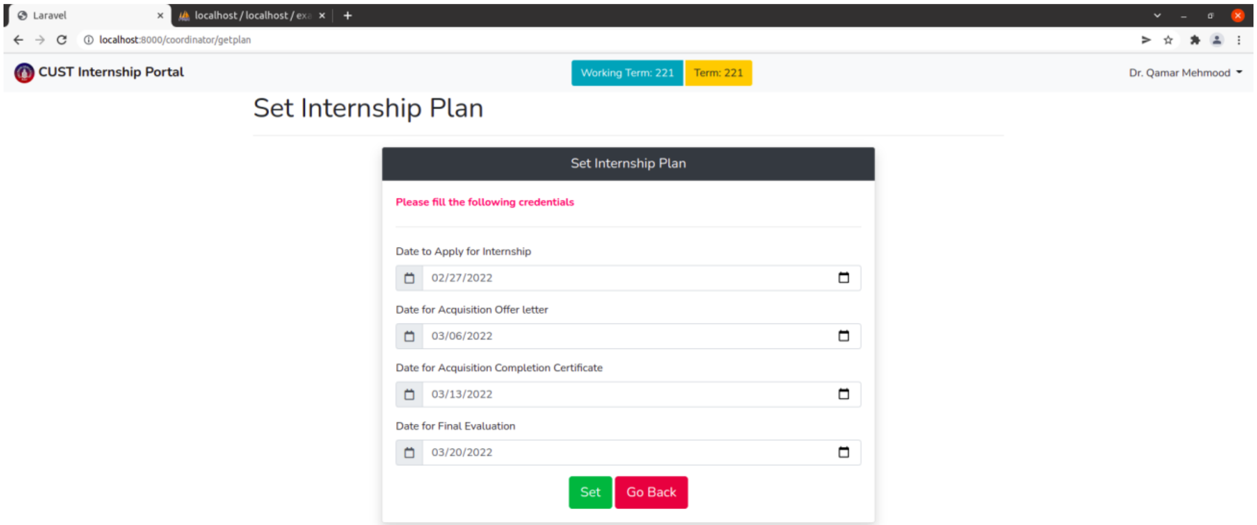
Confirm Password

Figure 57. Change Password

5.3.1.6. Set Internship Plan

Table 39. Test Case 17

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Password	Test ID: 17
Version: 1	Test Type: Unit Testing
Input: Date: Enter important Dates Click Button : Set	
Expected Result: If Coordinator is logged in, Set Internship Plan Successful	
Actual Result: Passed	



Set Internship Plan

Set Internship Plan

Please fill the following credentials

Date to Apply for Internship: 02/27/2022

Date for Acquisition Offer letter: 03/06/2022

Date for Acquisition Completion Certificate: 03/13/2022

Date for Final Evaluation: 03/20/2022

Figure 58. Set Internship Plan

5.3.2. Student

5.3.2.1. Login

Table 40.Test Case 18

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Student Login	Test ID: 18
Version: 1	Test Type: Unit Testing
Input: Username: aleeri@cust. Password: 12345678 Click Button: Login	
Expected Result: Login Failed, Invalid input	
Actual Result: Passed	

Table 41.Test Case 19

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Coordinator Login	Test ID: 19
Version: 1	Test Type: Unit Testing
Input: Username: XyStYR Password: 123cdd45678 Click Button: Login	
Expected Result: Login Failed, Username Incorrect	
Actual Result: Passed	

Table 42.Test Case 20

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Coordinator Login	Test ID: 20
Version: 1	Test Type: Unit Testing
Input: Username: qrtopR Password: 123c45678 Click Button: Login	
Expected Result: Login Failed, Incorrect Password	
Actual Result: Passed	

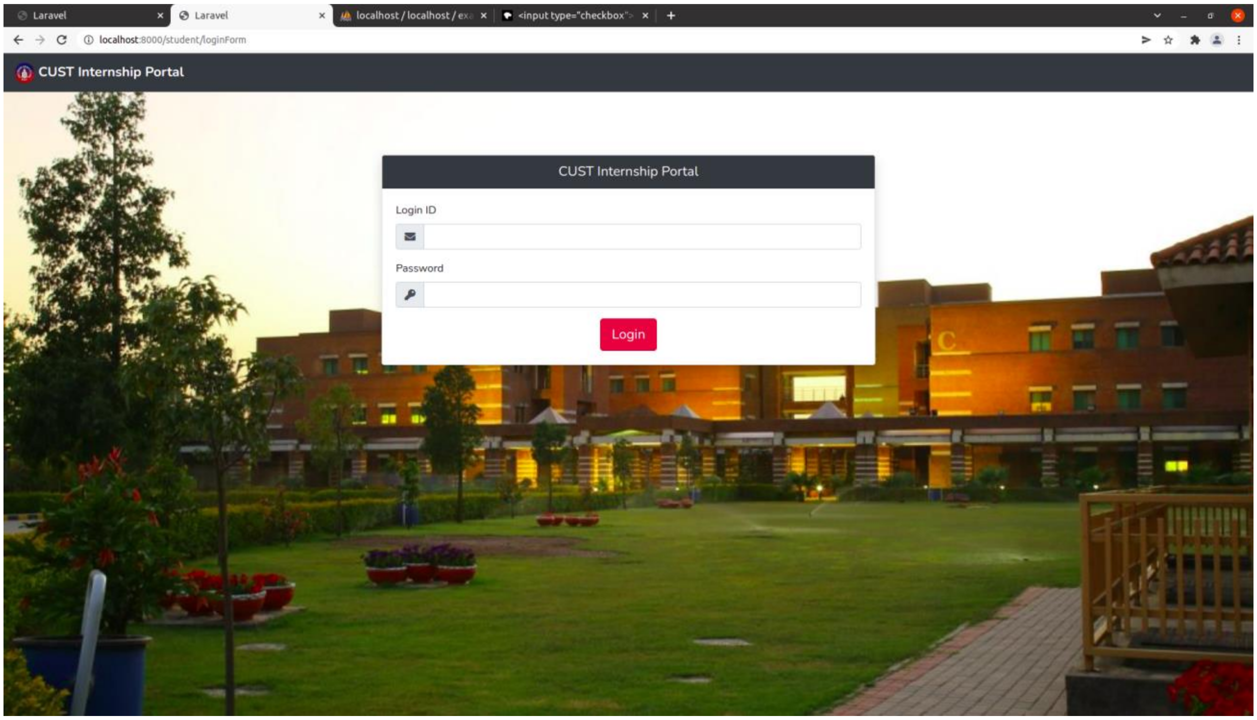
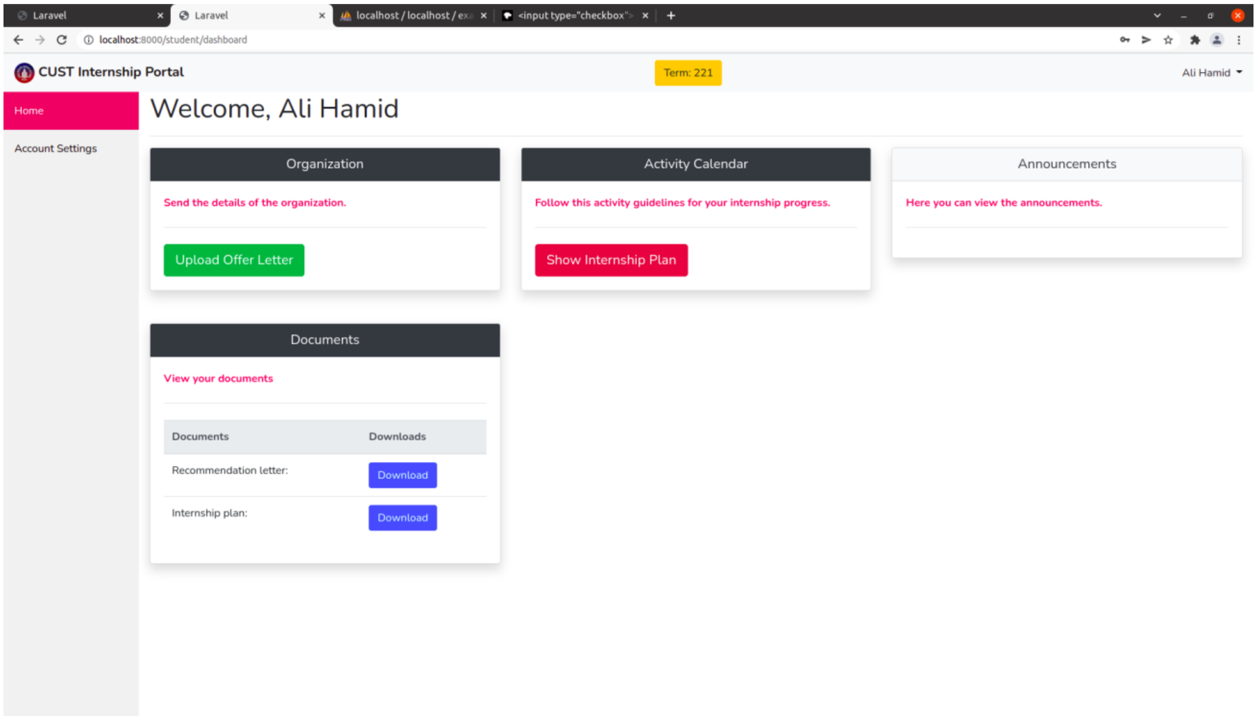


Figure 59.Login Student

Table 43.Test Case 21

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Coordinator Login	Test ID: 21
Version: 1	Test Type: Unit Testing
Input: Username: qrtopR Password: efrti98r Click Button: Login	
Expected Result: Login Successful , Redirect Student Home Page	
Actual Result: Passed	



Welcome, Ali Hamid

Send the details of the organization.

Follow this activity guidelines for your internship progress.

Here you can view the announcements.

View your documents

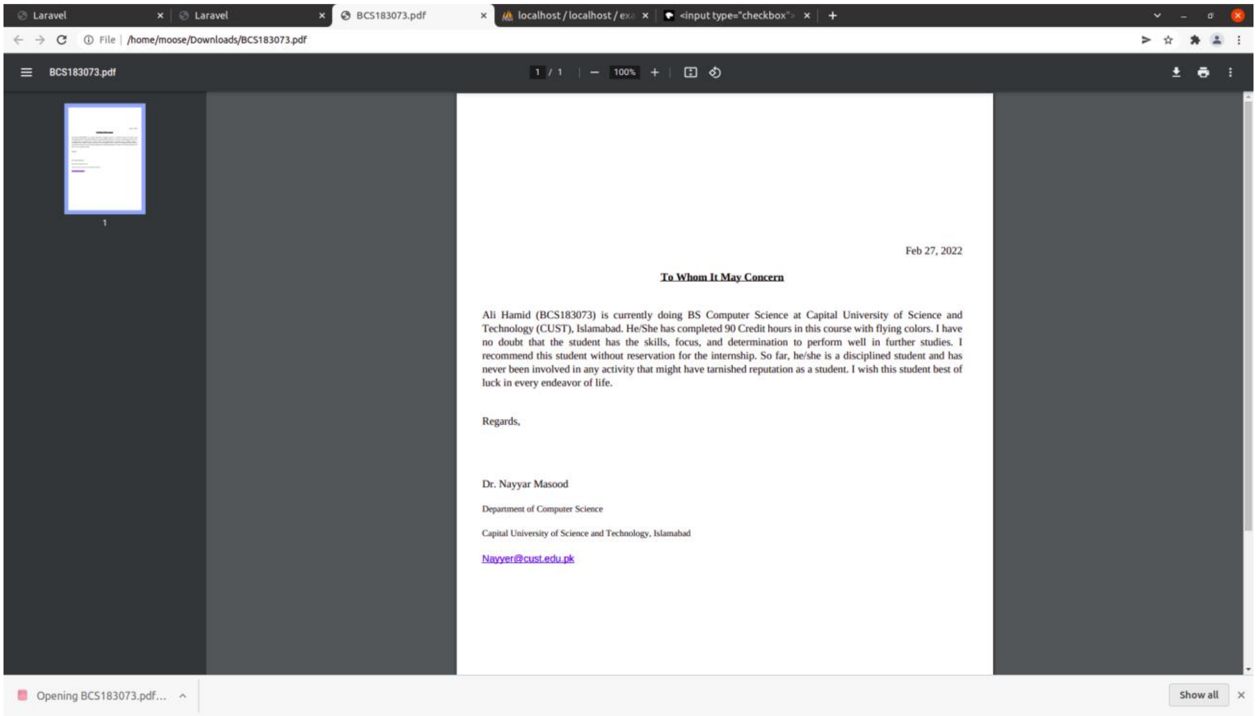
Documents	Downloads
Recommendation letter:	Download
Internship plan:	Download

Figure 60.Student Dashboard

5.3.2.2. Recommendation Letter Download

Table 44.Test Case 22

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Download Recommendation Letter	Test ID: 22
Version: 1	Test Type: Unit Testing
Input: Click Button : Download	
Expected Result: If Coordinator is logged in, Download Recommendation letter	
Actual Result: Passed	



Feb 27, 2022

To Whom It May Concern

Ali Hamid (BCS183073) is currently doing BS Computer Science at Capital University of Science and Technology (CUST), Islamabad. He/She has completed 90 Credit hours in this course with flying colors. I have no doubt that the student has the skills, focus, and determination to perform well in further studies. I recommend this student without reservation for the internship. So far, he/she is a disciplined student and has never been involved in any activity that might have tarnished reputation as a student. I wish this student best of luck in every endeavor of life.

Regards,

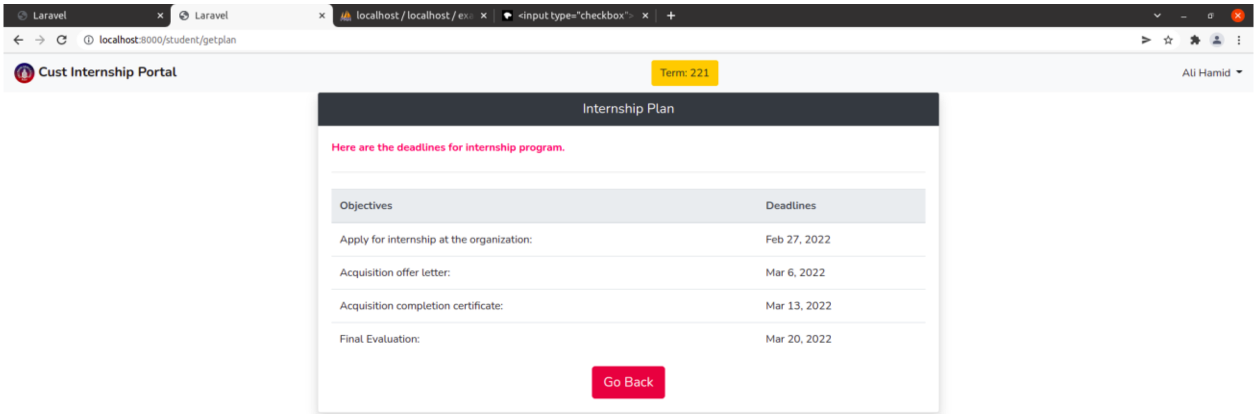
Dr. Nayyar Masood
Department of Computer Science
Capital University of Science and Technology, Islamabad
Nayyer@cust.edu.pk

Figure 61.Recommendation Letter

5.3.2.3. Internship Plan

Table 45.Test Case 23

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: View Internship Plan	Test ID: 23
Version: 1	Test Type: Unit Testing
Input: Click Button : View Internship Plan	
Expected Result: If Coordinator is logged in, Display Internship plan	
Actual Result: Passed	



Internship Plan

Here are the deadlines for internship program.

Objectives	Deadlines
Apply for internship at the organization:	Feb 27, 2022
Acquisition offer letter:	Mar 6, 2022
Acquisition completion certificate:	Mar 13, 2022
Final Evaluation:	Mar 20, 2022

Figure 62.Internship Plan

5.3.2.4. Password Change

Table 46.Test Case 24

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Password	Test ID: 24
Version: 1	Test Type: Unit Testing
Input: Click Button : Account Setting	
Expected Result: If Coordinator is logged in, Redirect to Account Setting page	
Actual Result: Passed	

Table 47.Test Case 25

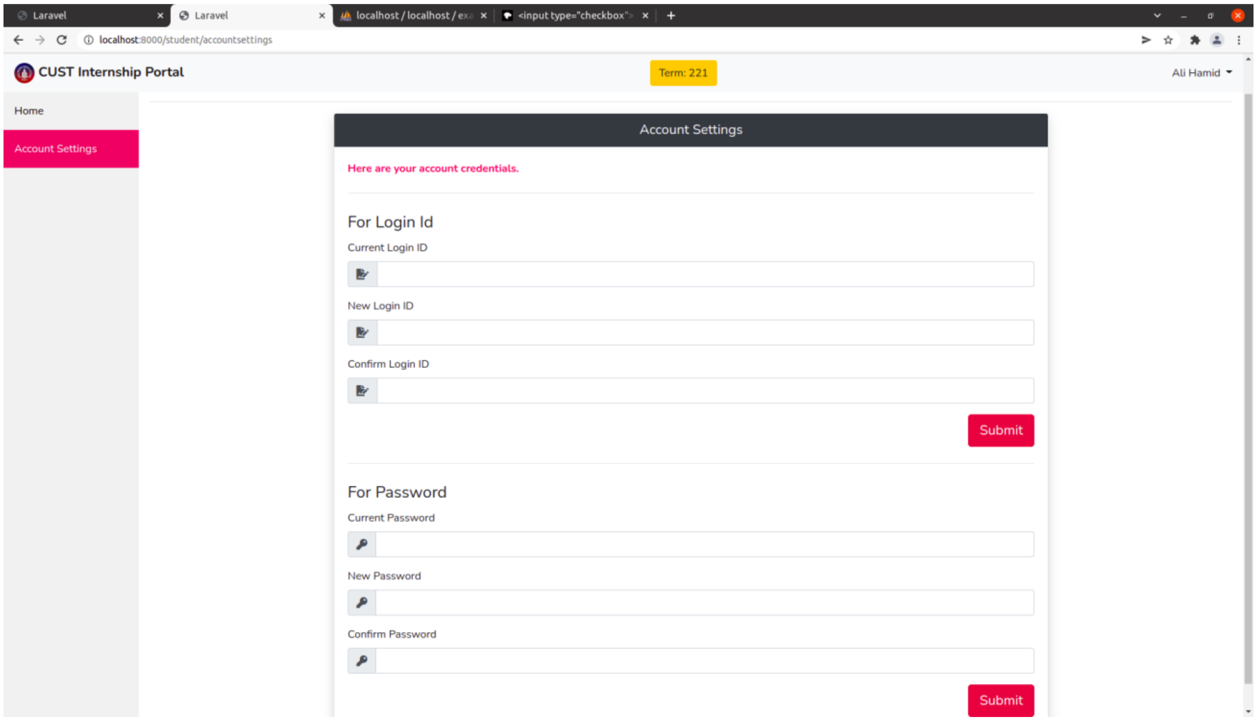
Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Password	Test ID: 25
Version: 2	Test Type: Unit Testing
Input: Current Password; *********** New Password; *********** Confirm Password; *********** Click Button : Submit	
Expected Result: Password changing failed, Incorrect Current Password	
Actual Result: Passed	

Table 48.Test Case 26

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Password	Test ID: 26
Version: 1	Test Type: Unit Testing
Input: Current Password; *********** New Password; *********** Confirm Password; *********** Click Button : Submit	
Expected Result: Password changing failed, New password does not match with confirm password	
Actual Result: Passed	

Table 49..Test Case 27

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Password	Test ID: 27
Version: 1	Test Type: Unit Testing
Input: Current Password; *********** New Password; *********** Confirm Password; *********** Click Button : Submit	
Expected Result: Password Changing Successful , Password changed	
Actual Result: Passed	



Account Settings

Here are your account credentials.

For Login Id

Current Login ID

New Login ID

Confirm Login ID

For Password

Current Password

New Password

Confirm Password

Figure 63.Change Password

5.3.2.5. Login Id Change

Table 50.Test Case 28

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Login ID	Test ID: 28
Version: 1	Test Type: Unit Testing
Input: Click Button : Account Setting	
Expected Result: If Coordinator is logged in, Redirect to Account Setting settings page	
Actual Result: Passed	

Table 51.Test Case 28

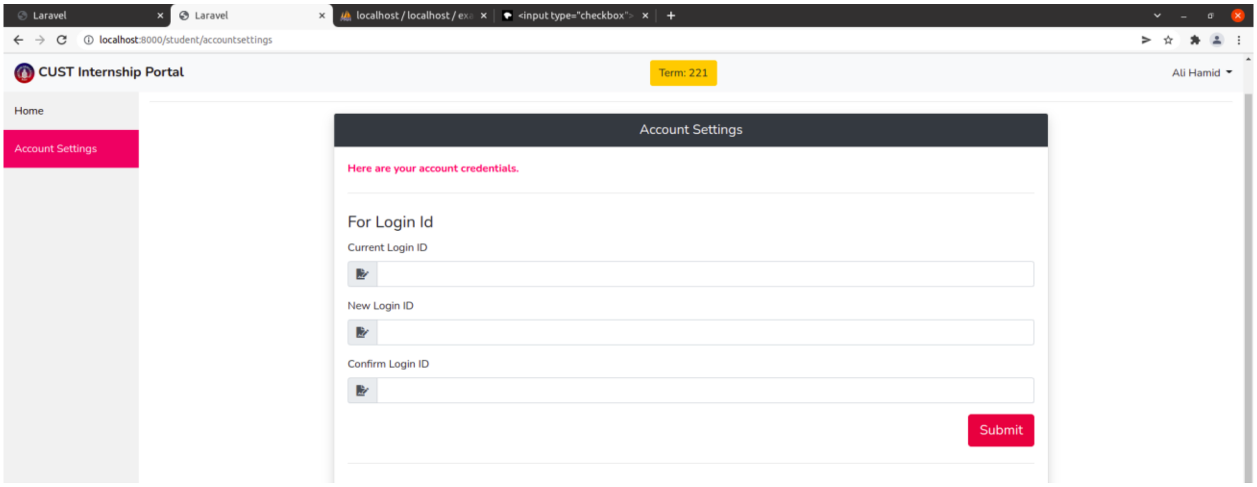
Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Login ID	Test ID: 28
Version: 2	Test Type: Unit Testing
Input: Current Login Id ; TrRewade New Login Id ; ali Confirm login Id ; ali Click Button : Submit	
Expected Result: Login Id changing failed, Incorrect Current Id	
Actual Result: Passed	

Table 52.Test Case 29

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Login ID	Test ID: 29
Version: 1	Test Type: Unit Testing
Input: Current Login Id ; TrRewade New Login Id ; ali Confirm login Id ; erf Click Button : Submit	
Expected Result: Login Id changing failed, New Login Id and Confirm login Id not matched	
Actual Result: Passed	

Table 53..Test Case 30

Date: February 25, 2022	
System: CUST Internship Portal	
Objective: Change Login ID	Test ID: 30
Version: 1	Test Type: Unit Testing
Input: Current Login Id ; rtYtree New Login Id ; ali Confirm login Id ;ali Click Button : Submit	
Expected Result: Login Id is changed , New Login Id is changed Successful	
Actual Result: Passed	



Here are your account credentials.

Figure 64.Change Login Id