# Project Documentation

## Research Coordinating System

### Supervised by

Nusrat Jahan

Senior Lecturer

### Submitted by

Md Sabbir Ahammed

171-35-1979

### Department of Software Engineering

### Daffodil International University

Fall-2020

## Acknowledgement

At first, I express my heartiest thanks and gratefulness to almighty Allah for divine blessing made it possible to complete this project successfully. I feel grateful to Nusrat Jahan, senior Lecturer, Department of Software Engineering, Daffodil International University. Deep knowledge keen interest of my supervisor in the field of technology influenced me to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this report. I would like to express my heartiest gratitude to Dr. Imran Mahmud, Head of the department, Department of SWE, Daffodil International University, for his kind help to finish my project and to other faculty member and all other staffs. Finally, I must acknowledge with due respect the constant support and patience of my family.

# Table of Content

## List of Figures

## List of Tables

# Chapter 1: Introduction

## 1.1 Project Overview

Research Coordinating System is a web-based application, which will help the faculty members to manage the working schedule for publishing process. Only the faculty members are allowed to use the system. For this, they need to be registered by an admin or a coordinator type user of this system. After successfully registered, a faculty member can submit their research paper details. System will check if the paper title is already submitted or not. If not, the system will accept the submission. The coordinator can check the paper details submitted by the faculty members and can make an overall report. Then he/she can send the report to the system admin. The admin can check the report, can assign tasks regarding the report to the coordinator. The coordinator will receive the tasks assign by the admin with due dates. Then he/she will pass those tasks to the faculty members according to their research papers. The admin can also promote a user as a coordinator and demote a coordinator to a faculty member.

## 1.2 Project Purpose

This "Research Coordinating System" will help the faculty members of a department to easily manage their research paper publishing process and keep track of their work schedule. So, the main purposes of this system could be:

- Managing research paper publishing process
- Keeping track of work schedule
- Working remotely
- Distributing tasks easily

### 1.2.1 Background

Many new things are discovered or being improved day to day. Universities are playing a vital role in this race. As a result, it's getting harder for the faculty members to cope with their research and publishing processes at a time. A lot of students want to complete their research and publish their paper during their graduation. So, every faculty member needs to go through more than one or two research papers of their students. Besides they have to keep track of publishing processes of their department.

Keeping all those difficulties in mind, the "Research Coordinating System" web-based application have been developed. It will automate many processes and will help the faculty members to keep track of their work. They will also have the benefit of working remotely.

### 1.2.2 Benefits & Beneficiaries

The project is all about to create a hub and repository of academic research papers and publications. There had the system before where all these tasks which are described below had to be maintained manually and therefore the reason it was very difficult to keep the track up to date and stockholder had to suffer very much.

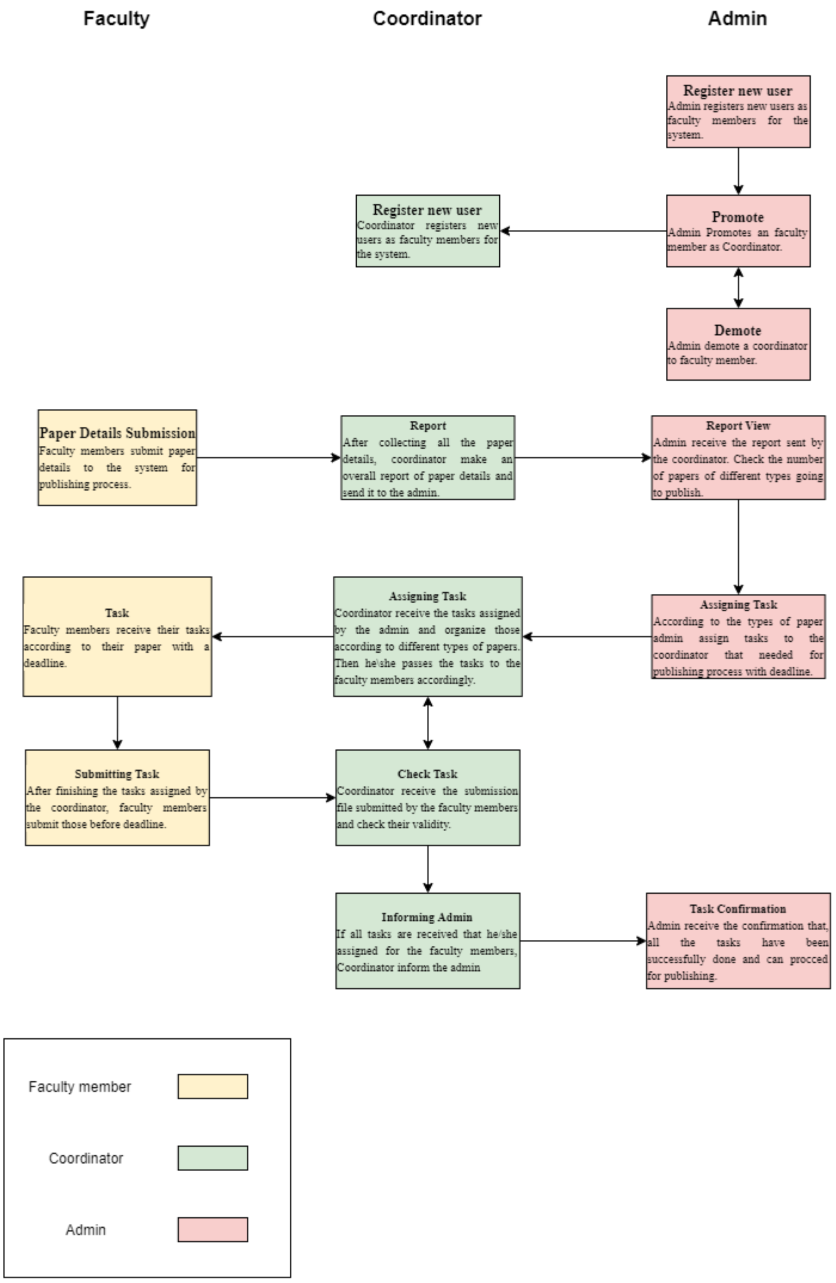
Admin and coordinator as well as all the stockholders will be the beneficiaries by using the system actually. Faculties don’t need to submit papers through email or post and it’s redundant to wait for the confirmation from the coordinators and these things were nothing but incertitude. Admin and coordinator will not have to wait and overcome the unnecessary delay to maintain the typical procedure. Every stockholder has their role and activity well defined and easy to use and I hope all these above circumstances help to make you understand the benefits.

## 1.3 Stakeholders

There are three stakeholders in the “Research Coordinating System”. They are-

- **Admin:** Admin is assigned automatically when the project start. Admin can assign user types and tasks to other users.
- **Coordinator:** Coordinator can register new users as faculty members and can assign tasks for them.
- **Faculty:** When an admin or coordinator register a new user to the system, he/she is considered as faculty member. A faculty member can submit paper details and tasks assigned by the coordinator. Tasks for him\her appear in his\her task menu.

## 1.4 Proposed System Model



**Register new user**
Admin registers new users as faculty members for the system.

**Register new user**
Coordinator registers new users as faculty members for the system.

**Promote**
Admin Promotes an faculty member as Coordinator.

**Demote**
Admin demote a coordinator to faculty member.

**Paper Details Submission**
Faculty members submit paper details to the system for publishing process.

**Report**
After collecting all the paper details, coordinator make an overall report of paper details and send it to the admin.

**Report View**
Admin receive the report sent by the coordinator. Check the number of papers of different types going to publish.

**Task**
Faculty members receive their tasks according to their paper with a deadline.

**Assigning Task**
Coordinator receive the tasks assigned by the admin and organize those according to different types of papers. Then he\she passes the tasks to the faculty members accordingly.

**Assigning Task**
According to the types of paper admin assign tasks to the coordinator that needed for publishing process with deadline.

**Submitting Task**
After finishing the tasks assigned by the coordinator, faculty members submit those before deadline.

**Check Task**
Coordinator receive the submission file submitted by the faculty members and check their validity.

**Informing Admin**
If all tasks are received that he/she assigned for the faculty members, Coordinator inform the admin

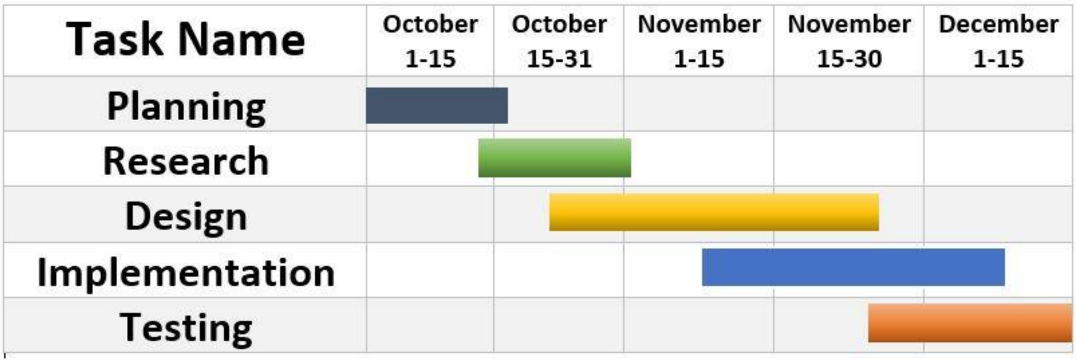
**Task Confirmation**
Admin receive the confirmation that, all the tasks have been successfully done and can procced for publishing.

Figure 1.1: Block Diagram

## 1.5 Project Schedule

### 1.5.1 Gantt Chart

Table 1.1: Gantt Chart



| Task Name | October 1-15 | October 15-31 | November 1-15 | November 15-30 | December 1-15 |
|---|---|---|---|---|---|
| Planning | ⬛ | ⬛ | | | |
| Research | 🟩 | 🟩 | 🟩 | | |
| Design | | 🟨 | 🟨 | 🟨 | |
| Implementation | | | 🟦 | 🟦 | 🟦 |
| Testing | | | | 🟧 | 🟧 |

### 1.5.2 Release Plane/Milestone

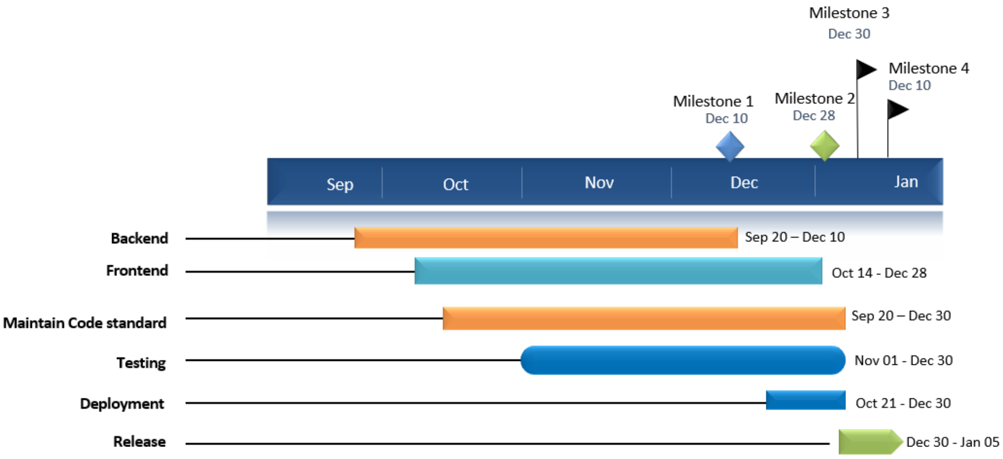


Figure 1.2: Milestone

# Chapter 2: Software Requirement Specification

## 2.1 Functional Requirements

Functionality requirements refers to the functions included in the system to understand the functionality requirements application. If an application is created, then of course functional requirements are required. Here I am going to discuss the functional requirements of the “Research Coordinating System”.

### 2.1.1 Log In

Table 2.1: Log In

| **FR-1** | **Log In** | | |
|---|---|---|---|
| **Description** | Admin can log in to this system with his/her login credentials. After registered and promoted by the admin coordinator and faculty can also log in to this system with credentials. | | |
| **Stakeholders** | Admin, Coordinator, Faculty | **Priority** | High |

### 2.1.2 Update Profile

Table 2.2: Update Profile

| **FR-2** | **Update Profile** | | |
|---|---|---|---|
| **Description** | Users of this system can update his/her profile after he/she is logged in. | | |
| **Stakeholders** | Admin, Coordinator, Faculty | **Priority** | Low |

### 2.1.3 Register New User

Table 2.3: Register New User

| **FR-3** | **Register New User** | | |
|---|---|---|---|
| **Description** | Admin and Coordinator can register a new user for this system with necessary information. | | |
| **Stakeholders** | Admin, Coordinator | **Priority** | High |

### 2.1.4 Assign Coordinator

Table 2.4: Assign Coordinator

| **FR-4** | **Assign Coordinator** | | |
|---|---|---|---|
| **Description** | Admin can promote a faculty member as a coordinator. He/she can also demote a coordinator to faculty member. | | |
| **Stakeholders** | Admin | **Priority** | High |

### 2.1.5 Assign Task

Table 2.5: Assign Task

| **FR-5** | **Assign Task** | | |
|---|---|---|---|
| **Description** | Admin can assign tasks for coordinator and coordinator can assign task for faculty members. | | |
| **Stakeholders** | Admin, Coordinator | **Priority** | Medium |

### 2.1.6 Submit Paper Details

Table 2.6: Submit Paper Details

| **FR-6** | **Submit Paper Details** | | |
|---|---|---|---|
| **Description** | Faculty members can submit paper details about their research in the system they want to publish. | | |
| **Stakeholders** | Faculty | **Priority** | Medium |

### 2.1.7 Report writing

Table 2.7: Report Writing

| **FR-7** | **Report Writing** | | |
|---|---|---|---|
| **Description** | Coordinator can make a report of overall paper details based on category after all the faculty members submitted their paper details. | | |
| **Stakeholders** | Coordinator | **Priority** | Medium |

### 2.1.8 Log Out

Table 2.8: Log Out

| **FR-8** | **Log out** | | |
|---|---|---|---|
| **Description** | Users of this system can log out from the system. All the session records will be destroyed from the browser immediately. Users need to log in again in order to do something in the system. | | |
| **Stakeholders** | Admin, Coordinator, Faculty | **Priority** | High |

## 2.2 Performance Requirements

Performance requirements determine how effective the system is in a given situation. Examples include software response speed, throughput, execution time and storage capacity. Service levels with performance requirements often support end-user tasks.

### 2.2.1 Speed and Latency Requirements

System's response time during working schedule is a major fact that specify an application's quality. Overall response time of this system is good. Speed of a web application also depends on its host. It can be said that, with a good hosting facility the "Research Coordinating System" will work perfectly smooth and quick.

### 2.2.2 Precision or Accuracy Requirements

Accuracy of data provide by a system is mandatory for a good quality of system. This system provides 100% accurate data with the right authorization. In this system, I used unit of work for the surety of providing 100% accurate data. In this case if something goes wrong during collecting data from a user, the system will role back the whole process and the collecting process will start again for accuracy.

### 2.2.3 Capacity Requirements

Advanced systems must be able to manage user data, provide accurate information, manage databases, manage http requests.

Table 2.9: Capacity Requirements

| **CR-1** | **The system will handle more than thousands of data** | | |
|---|---|---|---|
| **Description** | The system needs to handle thousands of data every moment | | |
| **Stakeholders** | Admin, Coordinator, Faculty | **Priority** | High |

## 2.3 Dependability Requirements

Dependability is measured on the basis of four dimensions. Like:

### 2.3.1 Reliability Requirements

Table 2.10: Reliability Requirement

| **RR-1** | **The system will available 24*7** | | |
|---|---|---|---|
| **Description** | This system will give service to its user all day long, will be malware free and will be updated when needed. | | |
| **Stakeholders** | Admin, Coordinator, Faculty | **Priority** | Medium |

### 2.3.2 Availability Requirements

It is important to ensure a Zero percent crash to ensure error tolerance benefits for end users. It’s also mandatory to shows accurate results.

Table 2.11: Availability Requirement

| **AR-1** | **The system handles every user access without errors** | | |
|---|---|---|---|
| **Description** | It’s possible that all the user tries to log in or doing something in the system at a time. In this situation system must handle their request without system errors. | | |
| **Stakeholders** | N/A | **Priority** | Medium |

### 2.3.3 Robustness or Fault-Tolerance Requirements

Providing after service and support to the end user is very important.

### 2.3.4 Safety-Critical Requirements

Scalability requirements define specific scalability requirements for stakeholders. This system is designed for maintenance, avoiding single points of failure and supplying as much as necessary data.

## 2.4 Security Requirements

Software security requirements should be its functional requirement. Software protection implements the protection of an application. Software security related functionality can be either directly tested or monitored. Below are some safety requirements:

- A proper way of sign in.
- Sign in credentials shouldn’t be disclosed to anyone in any situation.
- Gaining access according to the user type.
- Proper control swapping in the time of promote and demote
- Denying unauthorized registration
- Clearing session properly as a user log out

When users access the system, each and every module must be supplied from the central authentication process.

### 2.4.1 Access Requirements

Table 2.12: Access Requirement

| **ACR-1** | **Application Provides Secure Log In System** | | |
|---|---|---|---|
| **Description** | Each and every step of the system designed in such a way that it only allows the authorized users. | | |
| **Stakeholders** | N/A | **Priority** | High |

### 2.4.2 Integrity Requirements

Integrity requirements refer to a security system that ensures data quality expectations. It also ensures that all data on the system is never exposed to malicious changes or unexpected destruction.

### 2.4.3 Privacy Requirements

Ensuring the privacy of system users is very important. To ensure privacy, the central database is protected by anonymity. Users are allowed access to the data they are authorized to use.

## 2.5 Usability and Human-Interaction Requirements

Systems may fail for usability. That’s shy I build this application very user friendly, easy to understand and easy to manage.

### 2.5.1 Personalization and Internationalization Requirements

There is no personalization and internationalization requirements.

### 2.5.2 Understandability and Politeness Requirements

This system is built for organizational use. The interface is designed in a way that is very easy to understand. There are diagrams to fully understand the systems main working mechanism.

### 2.5.3 Accessibility Requirements

This system is built for organizational use only. So, I prefer that only the registered users have the accessibility for the system. And no one can register himself. Only the admin and coordinator can register a user. Then the user will be a valid user for the system.

### 2.5.4 User Documentation Requirements

There is no user documentation required in the system.

### 2.5.5 Training Requirements

No training requirements needed for this system.

## 2.6 Look and Feel Requirements

If a system does not look structured, users feel annoyed and does not want to go further. There are requirements to see and feel what the system will look like and how the system's user interface or graphical user interface will be displayed to users.

### 2.6.1 Appearance Requirements

The system is built in an understandable way that the users can easily use. For an example if admin added a task for coordinator, then the coordinator will be notified about his/her task. Accordingly, faculty members will be notified if coordinator add any task for them. Also, they can check if the task is completed by the users they appointed for. Users will understand the system very easily after they started working in it.

### 2.6.2 Style Requirements

Table 2.13: Style Requirement

| **SR-1** | **All content must be appearing within a format** | | |
|---|---|---|---|
| **Description** | Input field and other view result show a specific format | | |
| **Stakeholders** | Admin, Coordinator, Faculty | **Priority** | Medium |

## 2.7 Operational and Environmental Requirements

There are no operational and environmental requirements in this system.

## 2.8 Legal Requirements

There are no legal requirements in this system.

# Chapter 3: System Analysis

## 3.1 Use Case Diagram

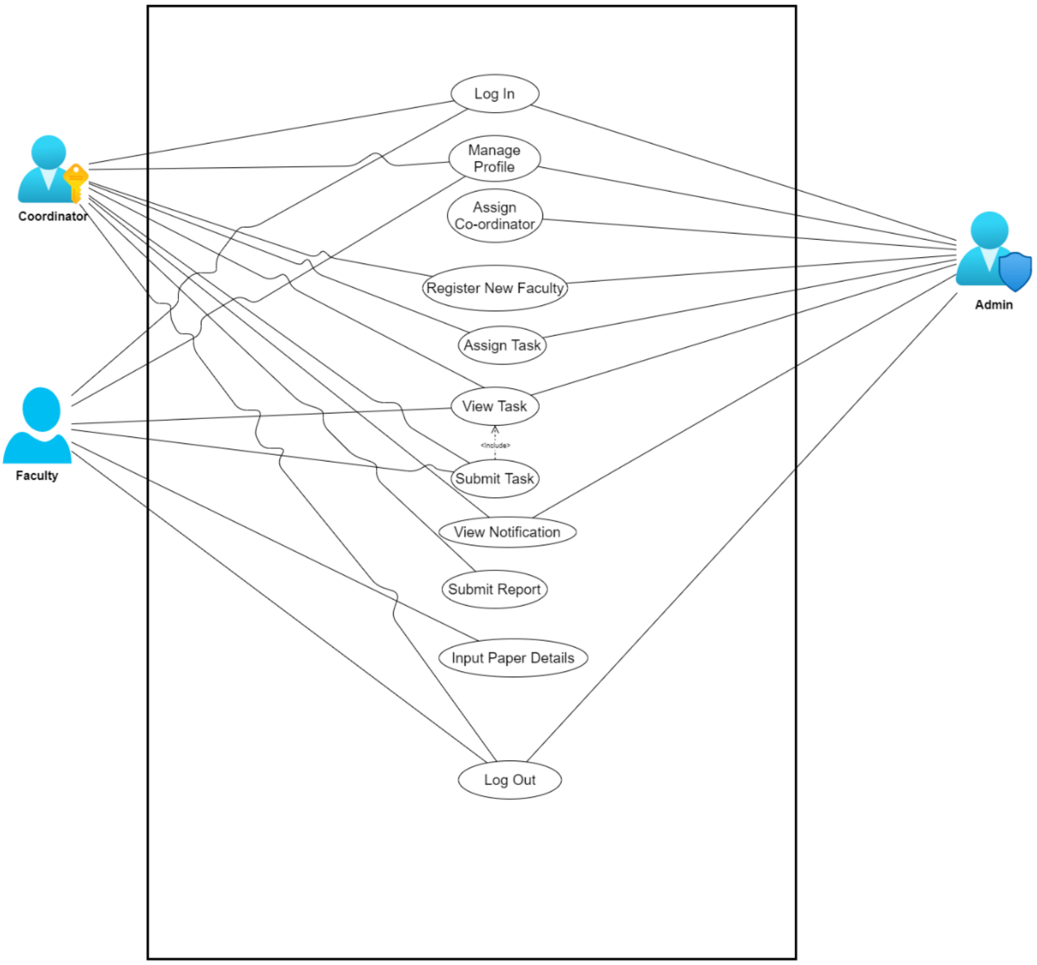


Figure 3.1: Use Case Diagram for "Research Coordinating System"

## 3.2 Use Case Description

### 3.2.1 Log In

Table 3.1: Log In

| **Use Case Title** | **Log In** | |
|---|---|---|
| **Goal** | Entering the dashboard according to user type | |
| **Preconditions** | Must be a registered user | |
| **Success End Condition** | Successfully logged in to dashboard | |
| **Failure End Condition** | Incorrect Email or Password | |
| **Primary Actor:**<br>**Secondary Actor:** | User<br>System | |
| **Trigger** | Log in button | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | User Input log in credentials |
| | 2 | Log in successful to dashboard |
| | 3 | User can use his/her dedicated work flows |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | Input incorrect or unregistered data |
| | 2 | Log in failed due to incorrect credentials |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.2 Manage Profile

Table 3.2: Manage Profile

| **Use Case Title** | **Manage Profile** | |
|---|---|---|
| **Goal** | Updating Name or adding phone number | |
| **Preconditions** | Must be a logged in | |
| **Success End Condition** | Well formatted data for required field | |
| **Failure End Condition** | Using bad formatted data for required field | |
| **Primary Actor:**<br>**Secondary Actor:** | User<br>System | |
| **Trigger** | Edit Profile Menu | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | User click edit profile |
| | 2 | User serve needed information |
| | 3 | Profile updated |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | Input incorrect information |
| | 2 | Update failed |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.3 Assign Coordinator

Table 3.3: Coordinator Assign

| | | |
|---|---|---|
| **Use Case Title** | **Assign Coordinator** | |
| **Goal** | Promoting a Faculty member as a Coordinator | |
| **Preconditions** | Must be logged in as an admin | |
| **Success End Condition** | The targeted user must be a registered faculty member | |
| **Failure End Condition** | The targeted user is already a Coordinator or an Admin or is not registered yet | |
| **Primary Actor:**<br>**Secondary Actor:** | Admin<br>Faculty | |
| **Trigger** | Assign Coordinator Menu | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | User must be an Admin |
| | 2 | Target the user needed to assign as Coordinator |
| | 3 | Click on Edit role and select Coordinator |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | Selecting Faculty again instead of Coordinator |
| | 2 | Targeting a Coordinator for assigning as coordinator |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.4 Register New Faculty

Table 3.4: Register New Faculty

| **Use Case Title** | **Register New Faculty** | |
|---|---|---|
| **Goal** | Adding a user as faculty member in the system | |
| **Preconditions** | Must be logged in as Coordinator or Admin | |
| **Success End Condition** | Serving valid information to the system | |
| **Failure End Condition** | Serving invalid information to the system | |
| **Primary Actor:**<br>**Secondary Actor:** | Admin, Coordinator<br>Faculty | |
| **Trigger** | Register New Faculty from Menu | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | Logged in as Coordinator or Admin |
| | 2 | Supplying valid information |
| | 3 | Registered user successfully |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | Supplying invalid information |
| | 2 | Incorrect information causes registration failure |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.5 Assign Task

Table 3.5: Assig Task

| **Use Case Title** | **Assign Task** | |
|---|---|---|
| **Goal** | Assigning task for the coordinator and faculty to progressing the publishing process | |
| **Preconditions** | Logged in as Admin or Coordinator | |
| **Success End Condition** | Providing valid information about task | |
| **Failure End Condition** | Providing invalid information about task | |
| **Primary Actor:**<br>**Secondary Actor:** | Admin, Coordinator<br>Faculty | |
| **Trigger** | Task from the side menu | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | Providing task including needed information |
| | 2 | Selecting candidate for this task |
| | 3 | Select deadline for the task |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | Incomplete information about task |
| | 2 | Task couldn’t be assigned |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.6 View Task

Table 3.6: View Task

| | | |
|---|---|---|
| **Use Case Title** | **View Task** | |
| **Goal** | View tasks that assigned by the user and for the user | |
| **Preconditions** | Must be a registered user | |
| **Success End Condition** | Tasks must be assigned | |
| **Failure End Condition** | No tasks assigned | |
| **Primary Actor:**<br>**Secondary Actor:** | Coordinator, Faculty<br>Admin, Coordinator | |
| **Trigger** | Task from side menu | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | Click task menu from side menu bar |
| | 2 | View tasks assigned for me and assigned by me |
| | 3 | Click on the specific one to view details |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | Request for view tasks |
| | 2 | No task assigned |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.7 Submit Task

Table 3.7: Submit Task

| **Use Case Title** | **Submit Task** | |
|---|---|---|
| **Goal** | Submit the assigned task before deadline | |
| **Preconditions** | Logged in and tasks must be assigned | |
| **Success End Condition** | Tasks submitted successfully | |
| **Failure End Condition** | Tasks is not submitted | |
| **Primary Actor:**<br>**Secondary Actor:** | Coordinator, Faculty<br>Admin, Coordinator | |
| **Trigger** | Task from side menu bar | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | Select the task that need to be submitted |
| | 2 | Upload the file that contains submission content |
| | 3 | Assigner receive the file successfully |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | Submitting without content file |
| | 2 | Submitting file without any content |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.8 View Notification

Table 3.8: View Notification

| **Use Case Title** | **View Notification** | |
|---|---|---|
| **Goal** | Alert about deadlines and tasks assigned | |
| **Preconditions** | Must be logged in | |
| **Success End Condition** | Nearby deadline or new task assigned for the user | |
| **Failure End Condition** | No tasks available for the user | |
| **Primary Actor:**<br>**Secondary Actor:** | Users<br>System | |
| **Trigger** | Notification from side menu bar | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | User check the notification menu |
| | 2 | Notification menu shows the nearby deadlines and new tasks assigned for the user |
| | 3 | Select specific notification to complete it |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | No tasks available for the user |
| | 2 | Notification window contain nothing |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.9 Submit Report

Table 3.9: Submit Report

| | | |
|---|---|---|
| **Use Case Title** | **Submit Report** | |
| **Goal** | Informing admin about research papers going to publish according to category | |
| **Preconditions** | Must be logged in as Coordinator | |
| **Success End Condition** | All the paper details must be submitted before making report | |
| **Failure End Condition** | Making report before submitting paper details | |
| **Primary Actor:**<br>**Secondary Actor:** | Coordinator<br>Admin | |
| **Trigger** | Report writing from side menu bar | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | Coordinator request for report |
| | 2 | System categorify all paper details |
| | 3 | Coordinator passes the report to the admin |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | Coordinator request for report before all paper details have been submitted |
| | 2 | Invalid informational report created |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.10 Input Paper Details

Table 3.10: Input Paper Details

| **Use Case Title** | **Input Paper Details** | |
|---|---|---|
| **Goal** | Providing information about the research papers going to publish | |
| **Preconditions** | Logged in as faculty members | |
| **Success End Condition** | Providing valid information in the required fields | |
| **Failure End Condition** | Providing invalid information or Existing information | |
| **Primary Actor:**<br>**Secondary Actor:** | Faculty<br>Coordinator | |
| **Trigger** | Paper Details from the side menu bar | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | Faculty input a paper detail |
| | 2 | Submit the details for review |
| | 3 | Coordinator receive the details |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | Faculty input an existing detail |
| | 2 | System reject the submission |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

### 3.2.11 Log Out

Table 3.11: Log Out

| **Use Case Title** | **Log Out** | |
|---|---|---|
| **Goal** | Exit the system | |
| **Preconditions** | Must be logged in | |
| **Success End Condition** | User is logged in | |
| **Failure End Condition** | User is already logged out | |
| **Primary Actor:**<br>**Secondary Actor:** | User<br>System | |
| **Trigger** | Log out button | |
| **Description**<br>**Main Success Scenario** | **Step** | **Action** |
| | 1 | User completed his/her work on system |
| | 2 | User clicked log out button |
| | 3 | System logged out the user and clear his/her session records |
| **Alternative flows** | **Step** | **Branching Action** |
| | 1 | User close the browser instead of log out |
| | 2 | System will catch his/her session record for a defined time for that browser. Then it will be cleaned also. |
| **Quality Requirements** | **Step** | **Requirements** |
| | | N/A |

## 3.3 Activity Diagram

### 3.3.1 Activity Diagram (Log In)

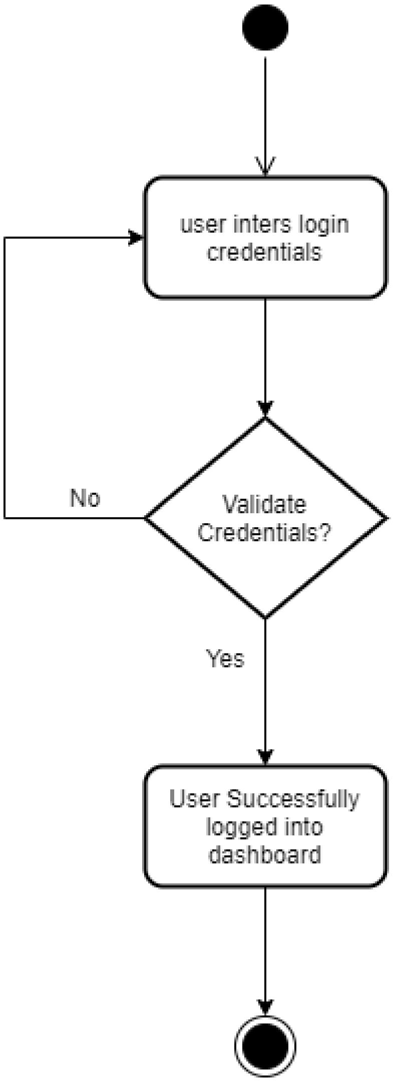

user inters login credentials
No
Validate Credentials?
Yes
User Successfully logged into dashboard

Figure 3.2: Activity Diagram for Log in

### 3.3.2 Activity Diagram (Manage Profile)

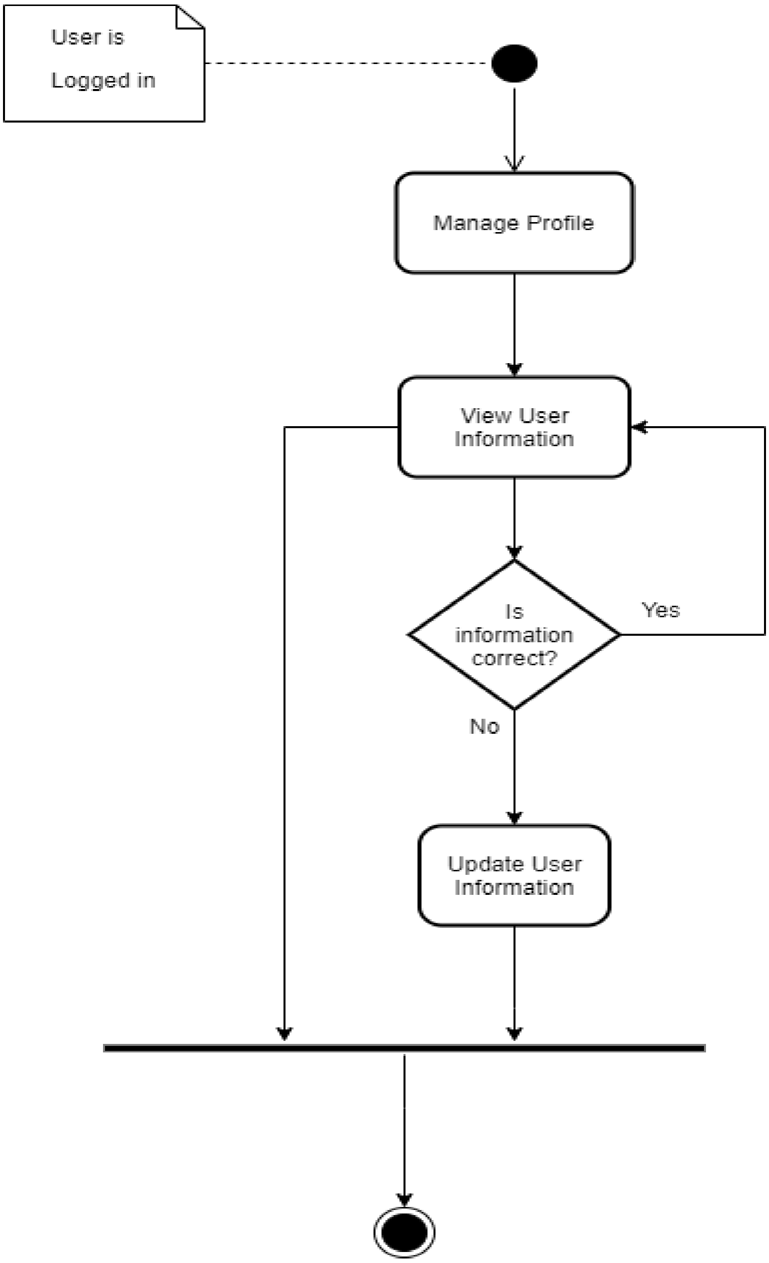


Figure 3.3: Activity diagram for Manage Profile

### 3.3.3 Activity Diagram (Assign Coordinator)

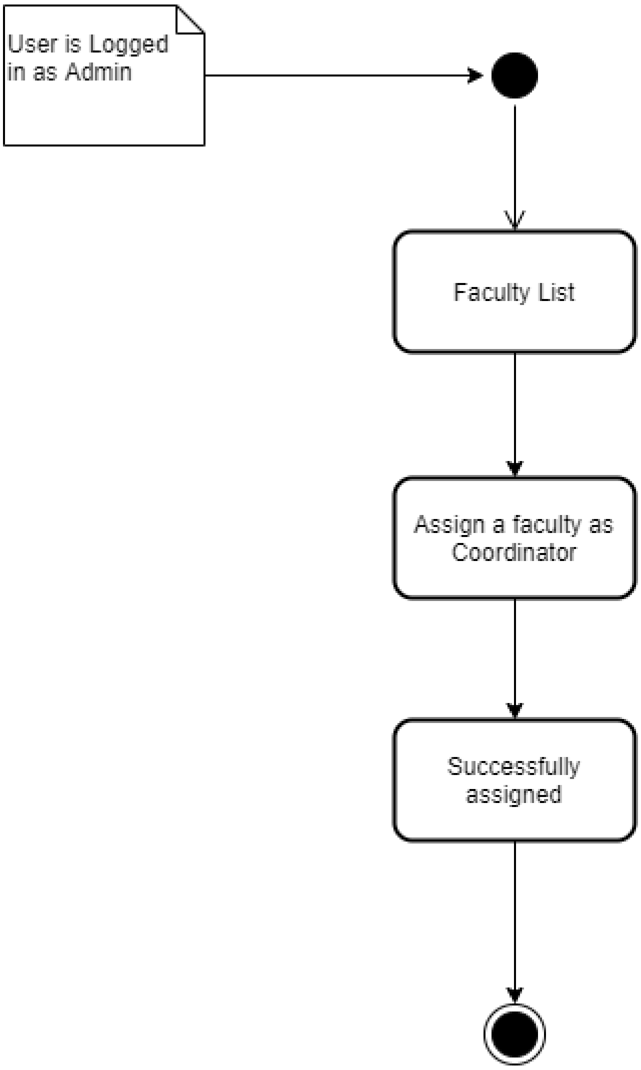


Figure 3.4: Activity diagram for Assign Coordinator

### 3.3.4 Activity Diagram (Register New Faculty)

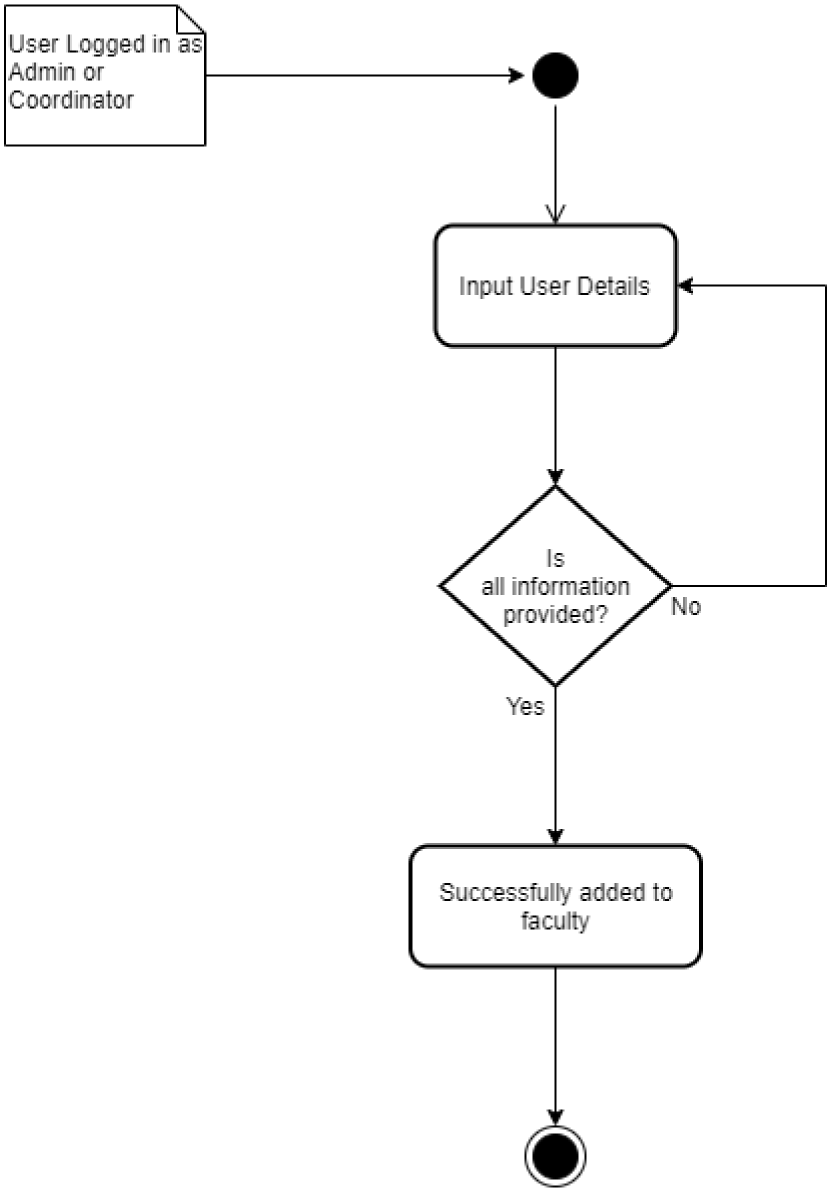

User Logged in as Admin or Coordinator
Input User Details
Is all information provided?
No
Yes
Successfully added to faculty

Figure 3.5: Activity diagram for Register New Faculty

### 3.3.5 Activity Diagram (Assign Task)

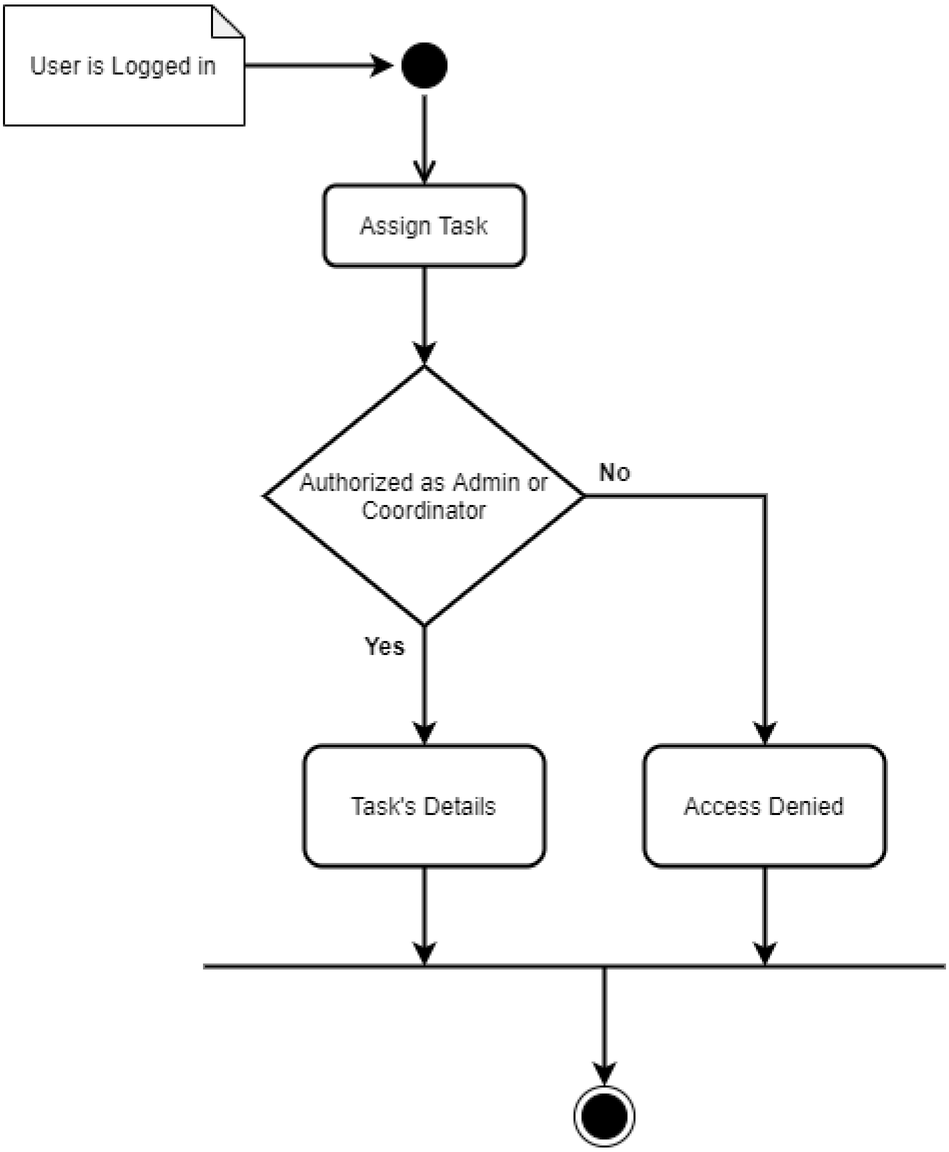


Figure 3.6: Activity diagram for Assign Task

### 3.3.6 Activity Diagram (view Task)

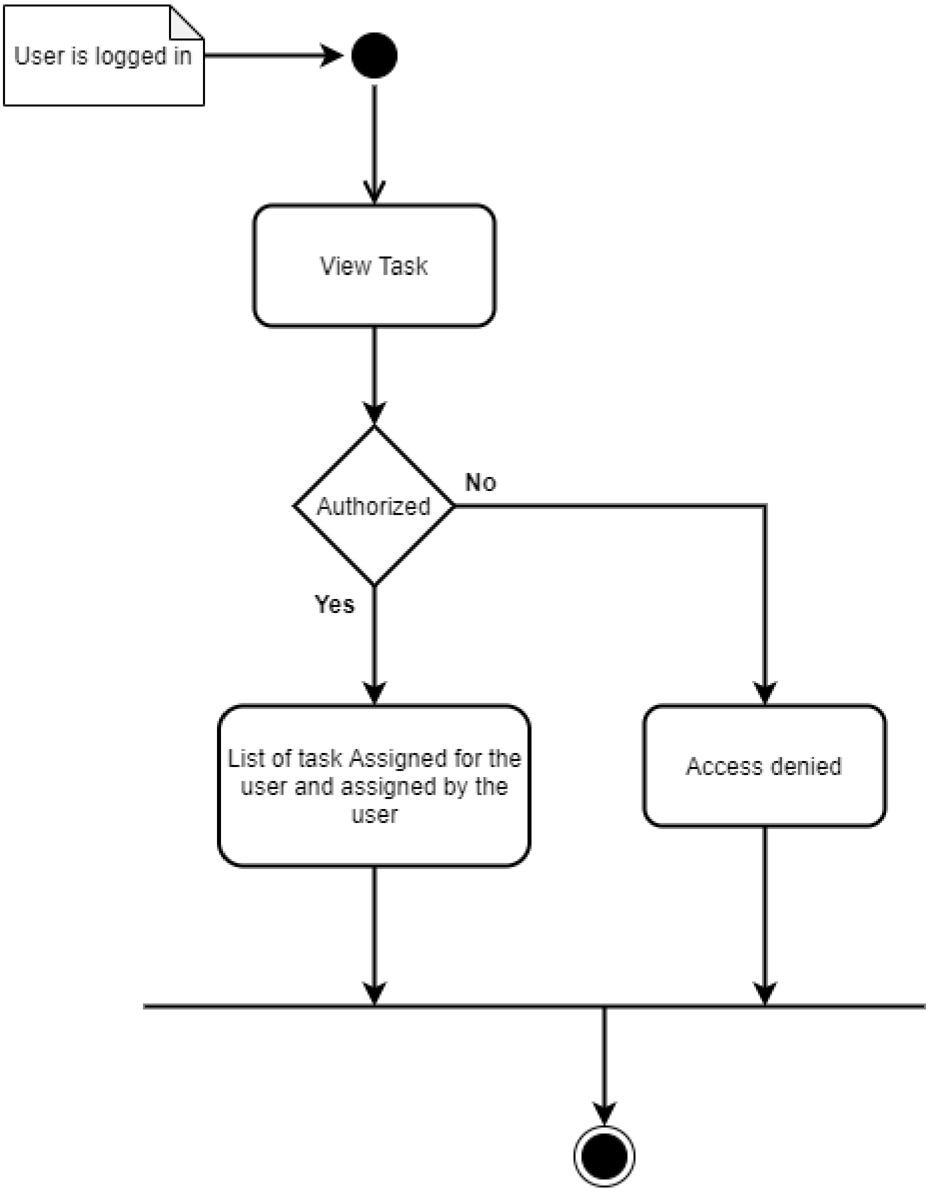


Figure 3.7: Activity diagram for View Task

### 3.3.7 Activity Diagram (Submit Task)

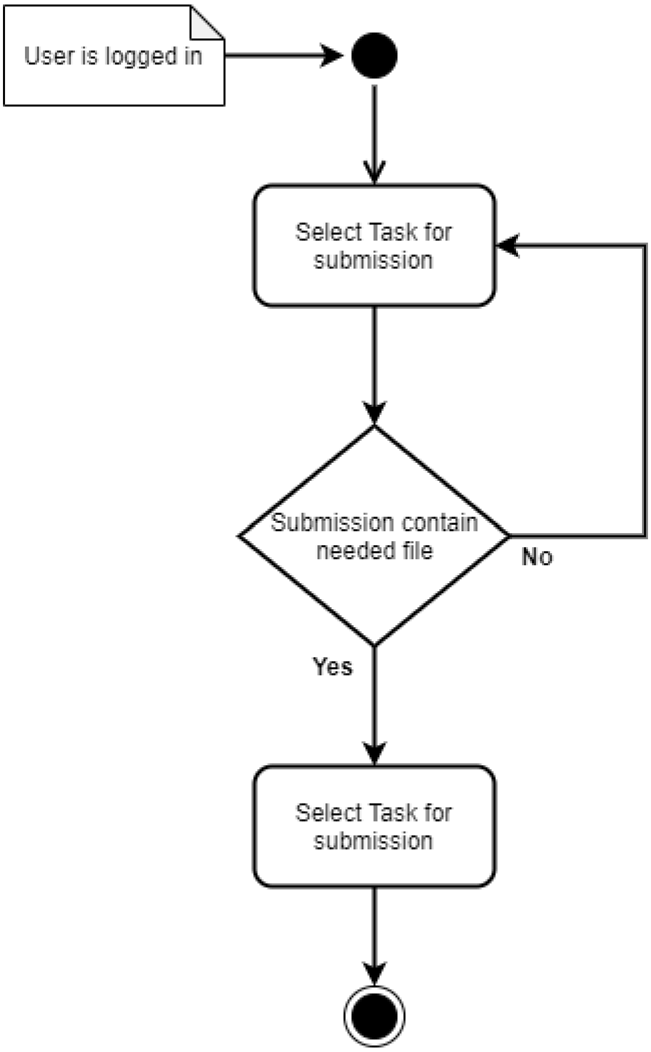

User is logged in
Select Task for submission
Submission contain needed file
No
Yes
Select Task for submission

Figure 3.8: Activity diagram for Submit Task

### 3.3.8 Activity Diagram (View Notification)

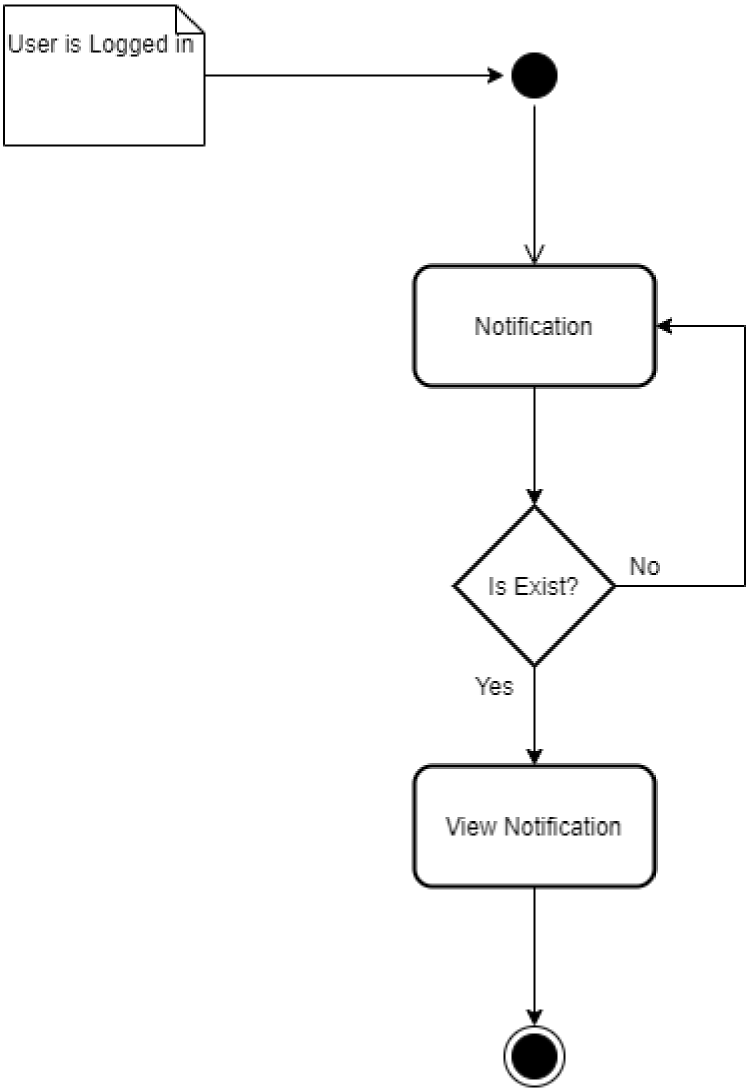


Figure 3.9: Activity diagram for Notification

### 3.3.9 Activity Diagram (Report Writing)

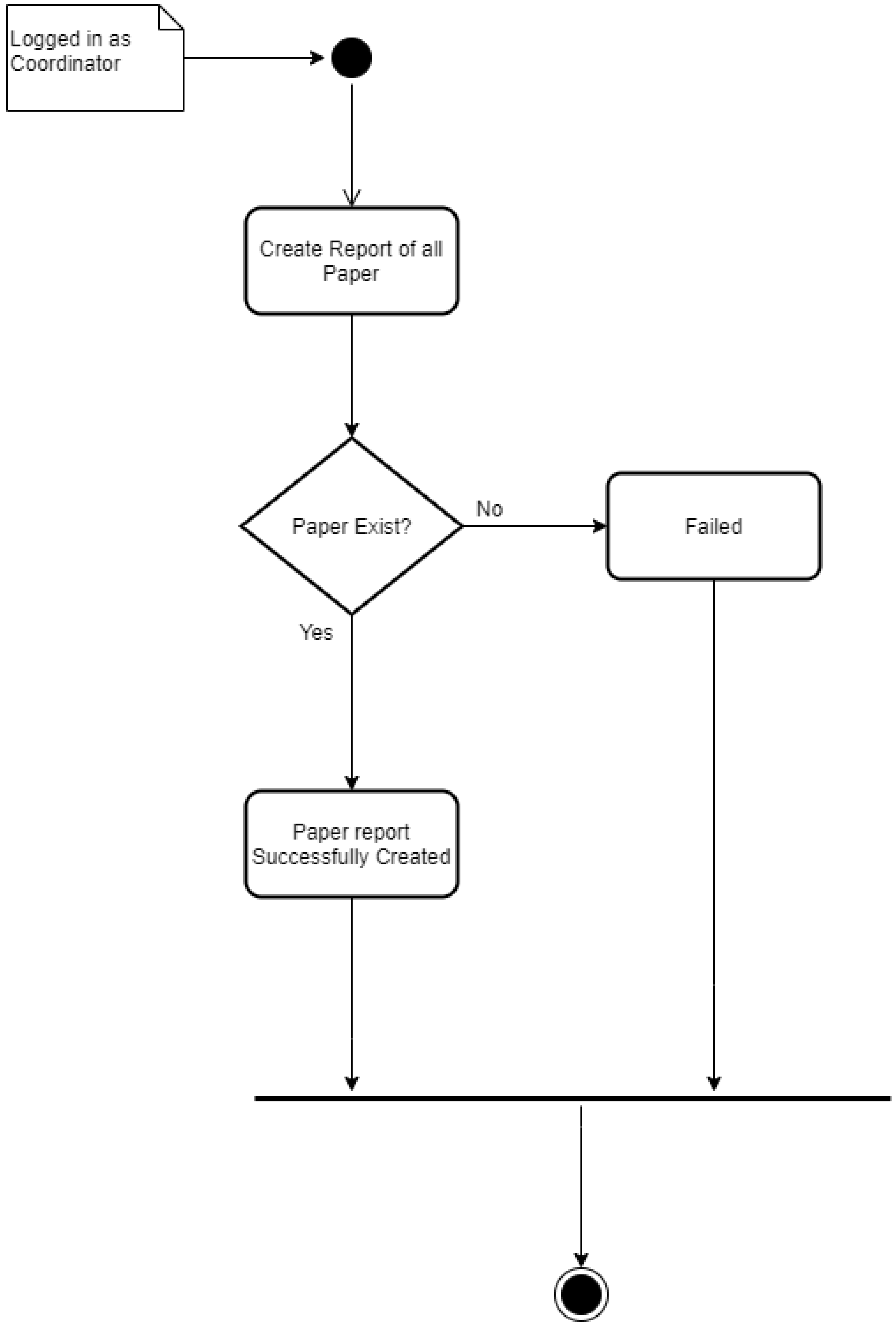


Figure 3.10: Activity diagram for Report Writing

### 3.3.10 Activity Diagram (Input Paper Details)

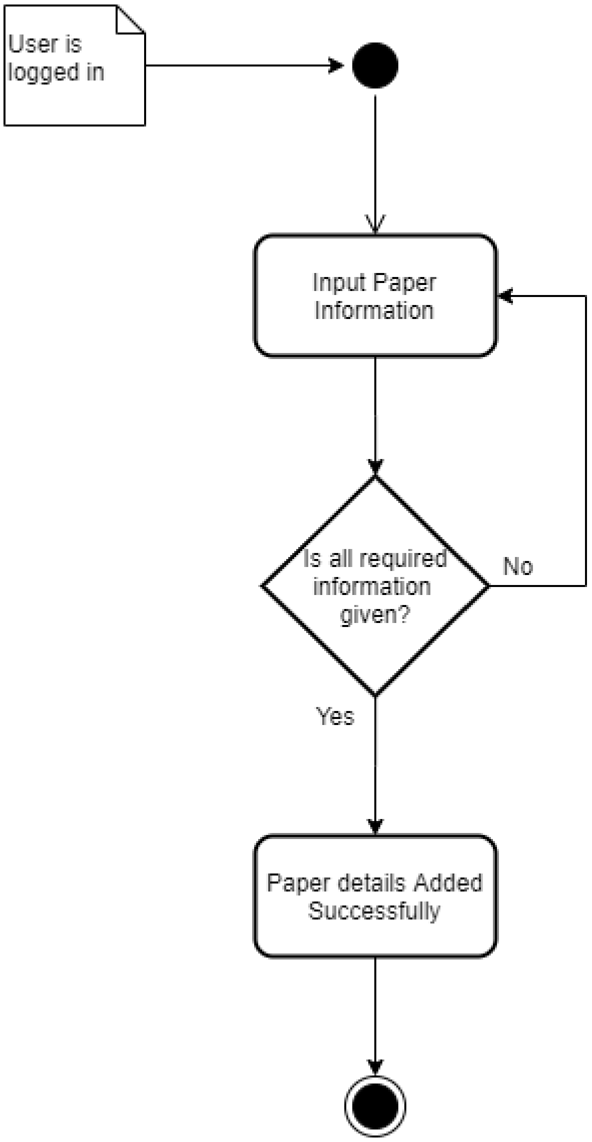

User is logged in
Input Paper Information
Is all required information given?
No
Yes
Paper details Added Successfully

Figure 3.11: Activity diagram for Paper Details

### 3.3.11 Activity Diagram (Log Out)

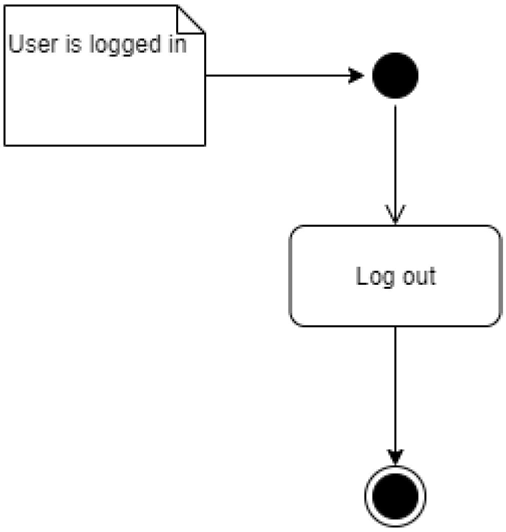


Figure 3.12: Activity diagram for Log Out

## 3.4 System Sequence Diagram

### 3.4.1 Sequence Diagram (Log In)

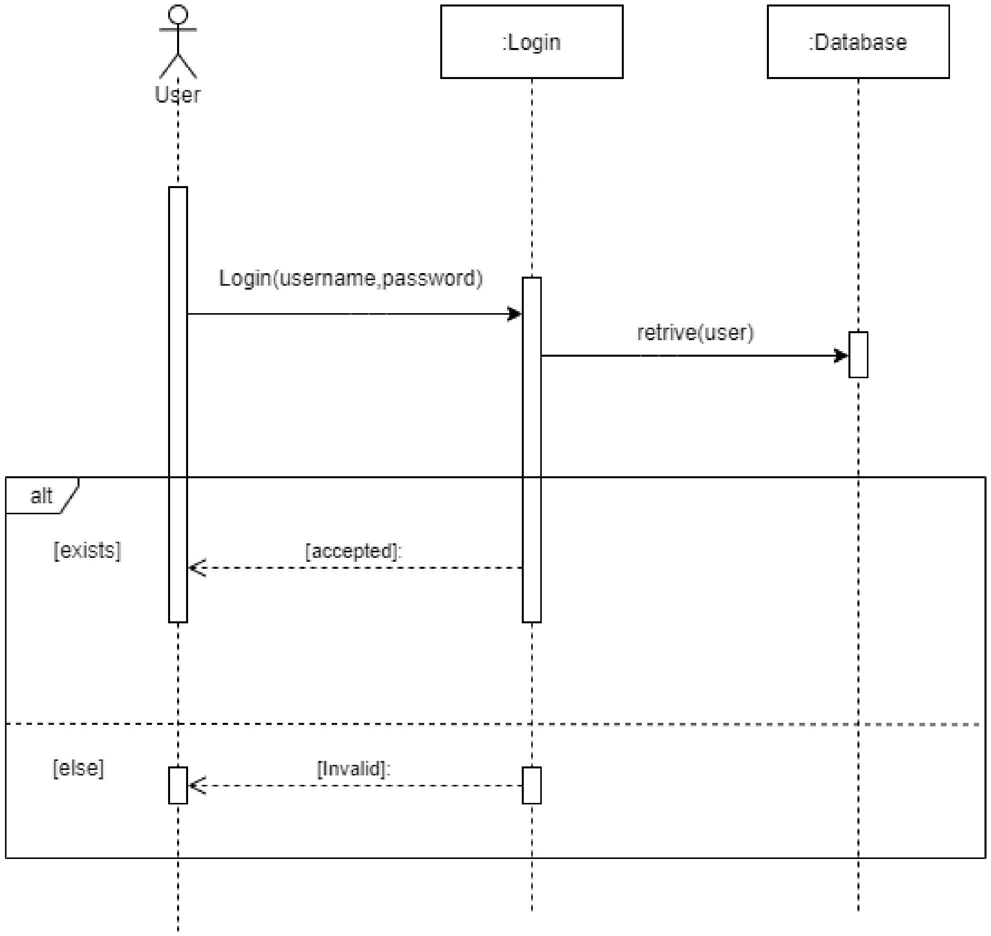


Figure 3.13: Sequence diagram for Log in

### 3.4.2 Sequence Diagram (Manage Profile)

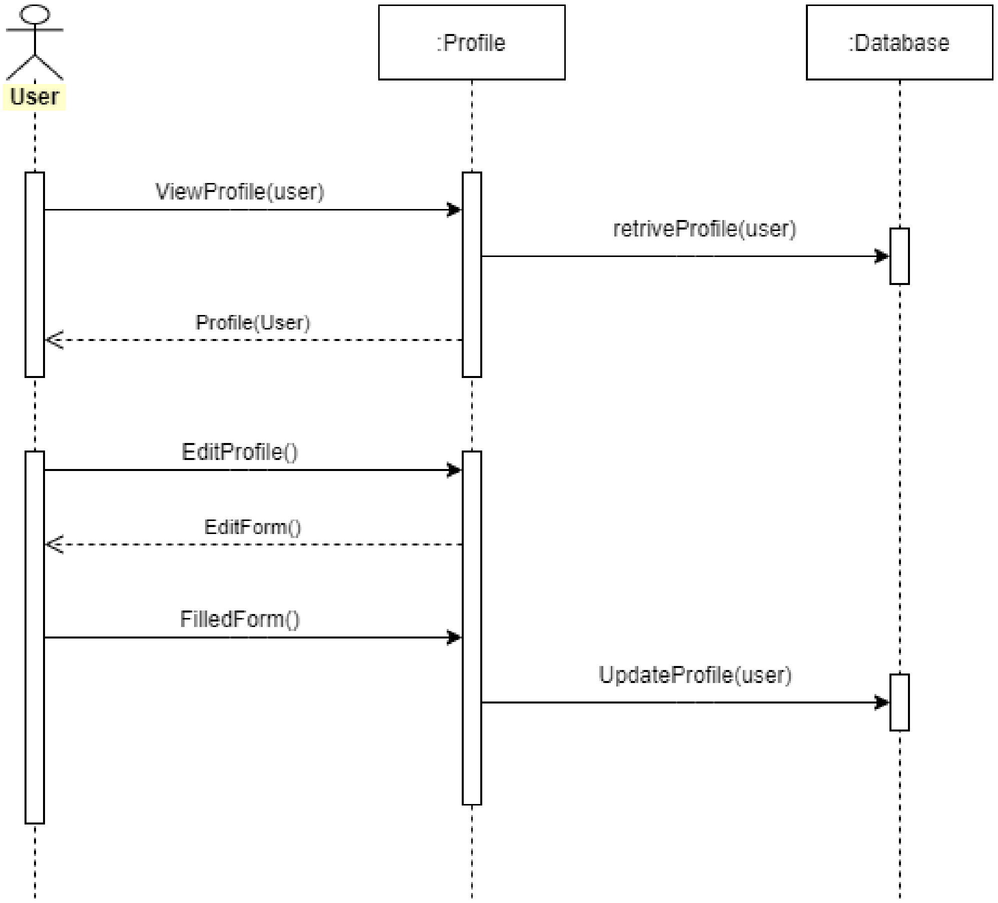


Figure 3.14: Sequence diagram for Manage Profile

### 3.4.3 Sequence Diagram (Assign Coordinator)

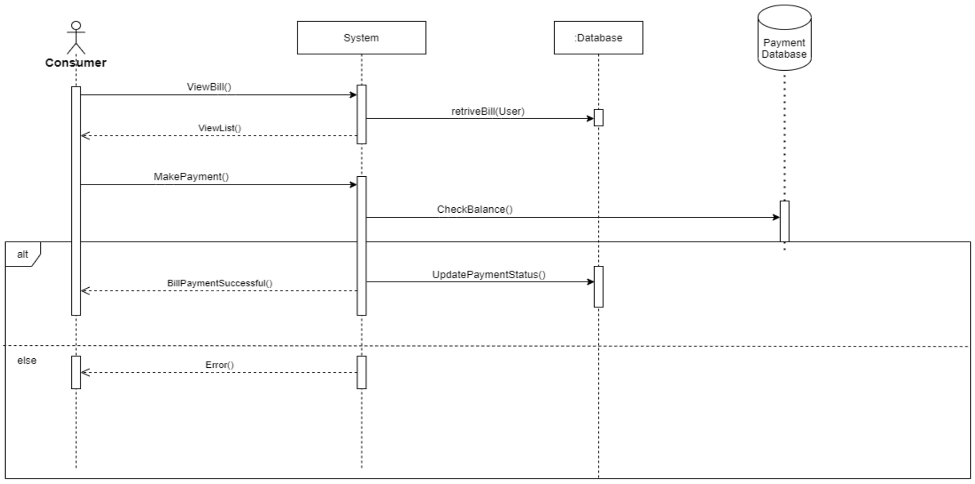


Figure 3.15: Sequence diagram for Assign Coordinator

### 3.4.4 Sequence Diagram (Register New Faculty)

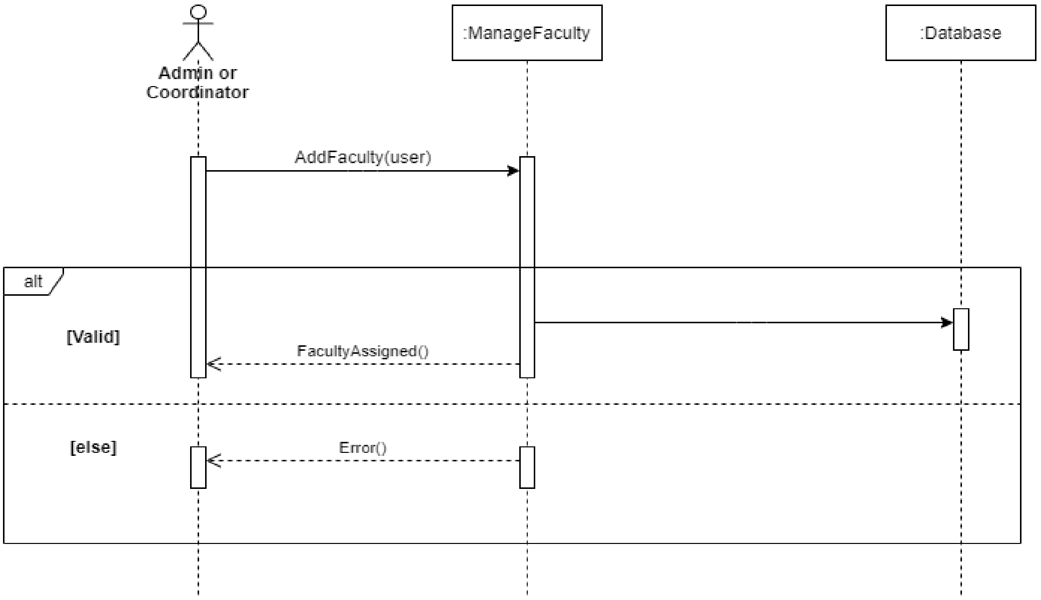


Figure 3.16: Sequence diagram for Register New User

### 3.4.5 Sequence Diagram (Assign Task)

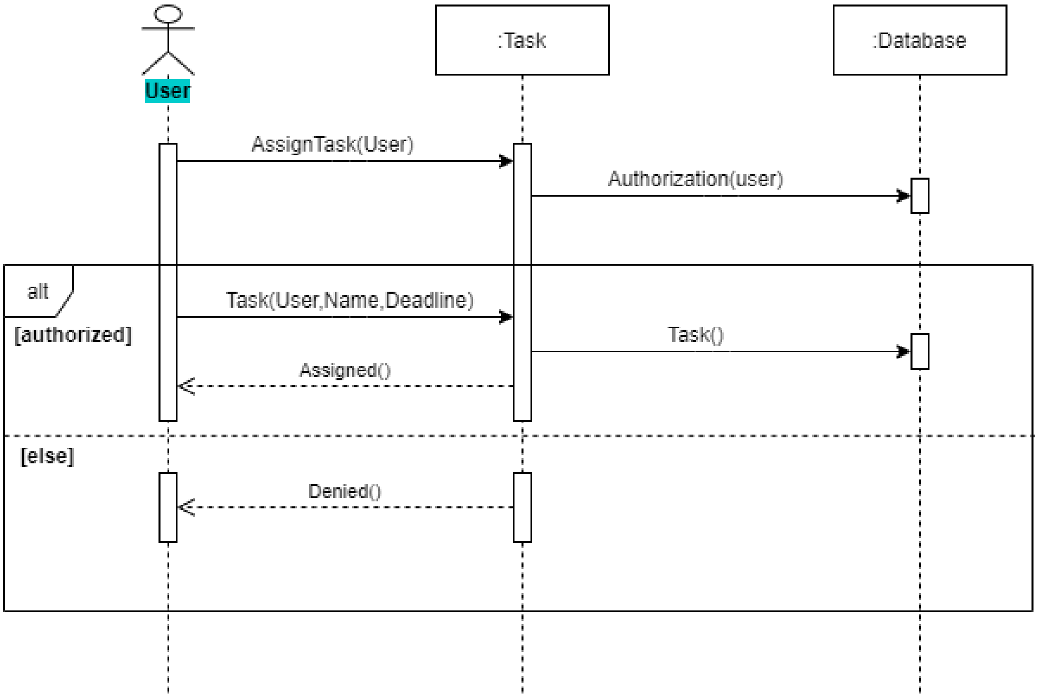


Figure 3.17: Sequence diagram for Assign Task

### 3.4.6 Sequence Diagram (View Task)

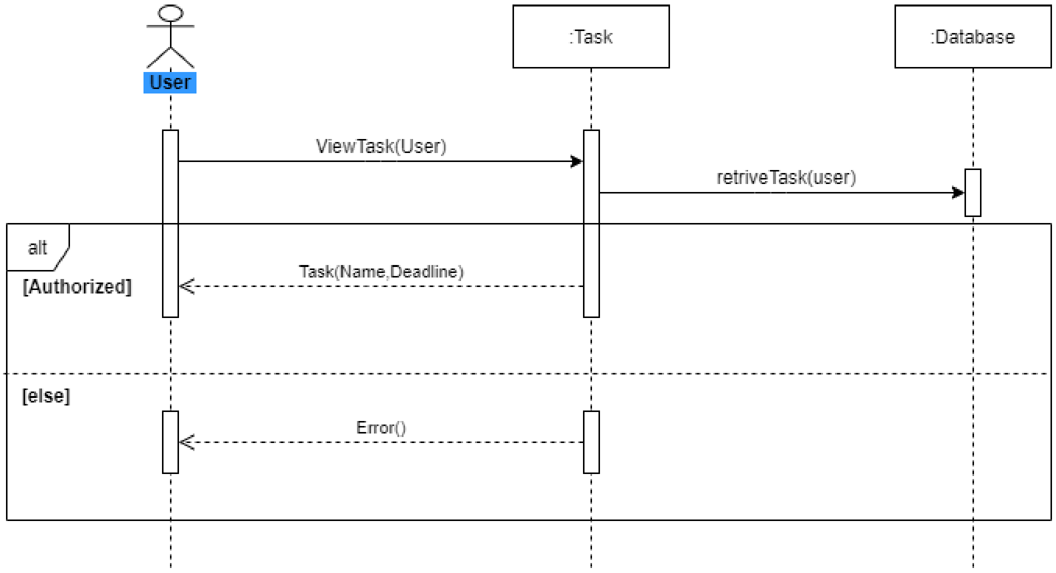


Figure 3.18: Sequence diagram for View Task

### 3.4.7 Sequence Diagram (Submit Task)

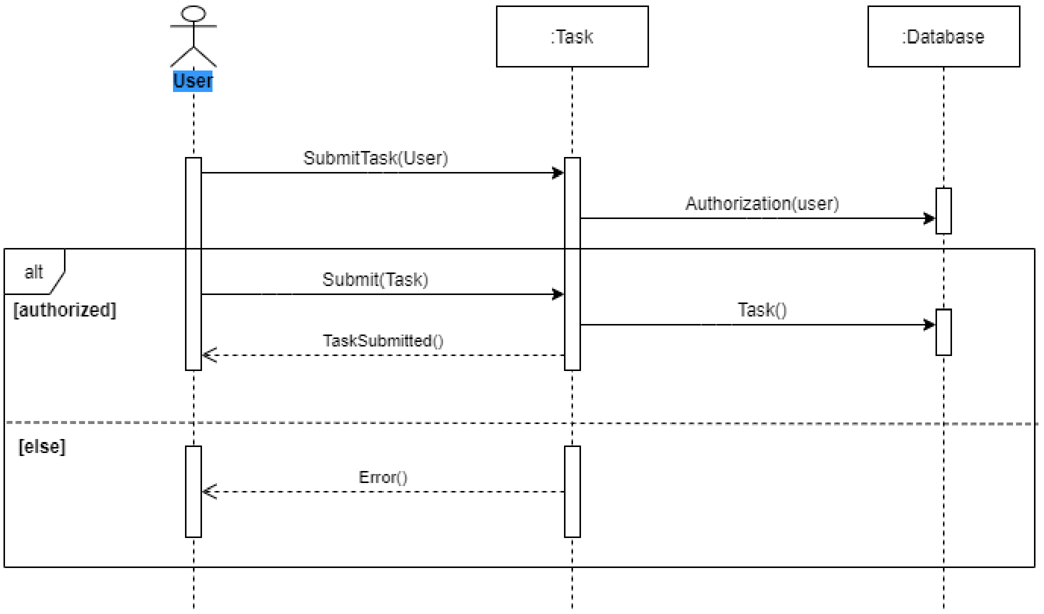


Figure 3.19: Sequence diagram for Submit Task

### 3.4.8 Sequence Diagram (View Notification)

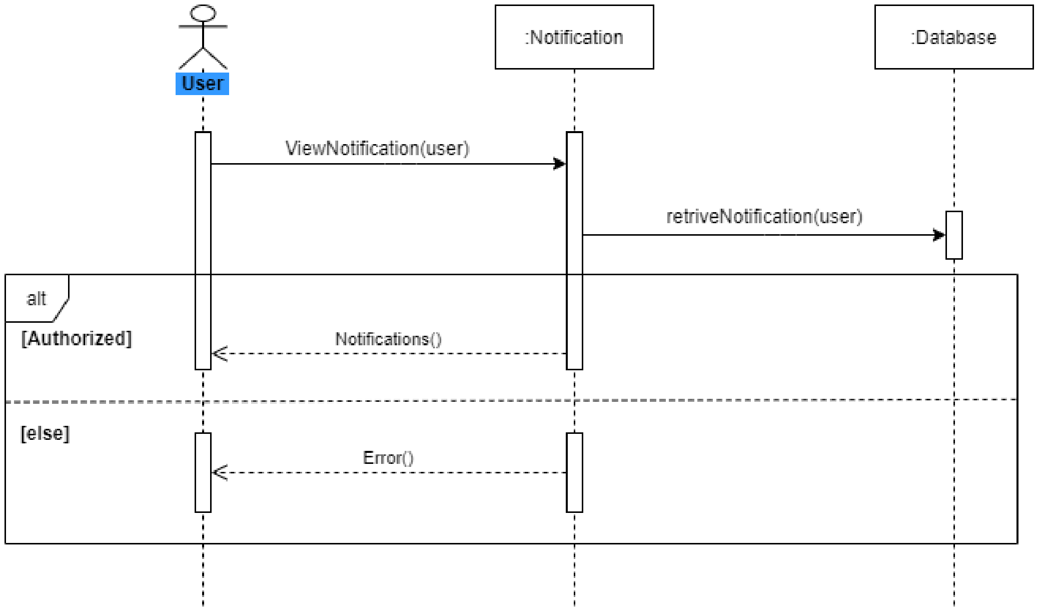


Figure 3.20: Sequence diagram for View Notification

### 3.4.9 Sequence Diagram (Report Writing)

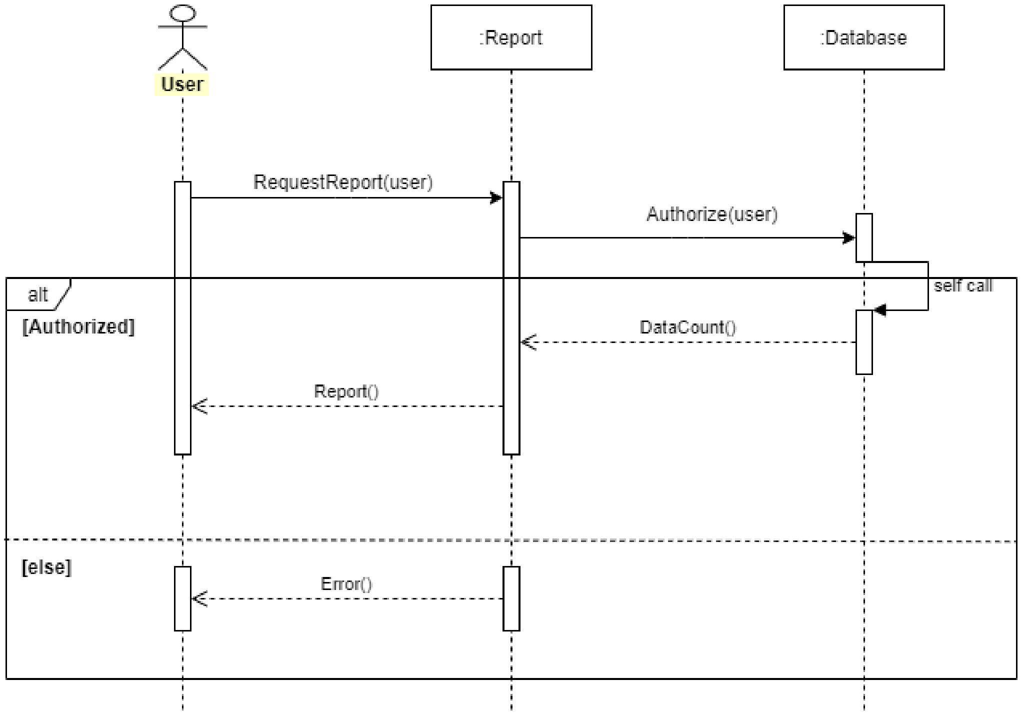


Figure 3.21: Sequence diagram for Report Writing

### 3.4.10 Sequence Diagram (Input Paper Details)

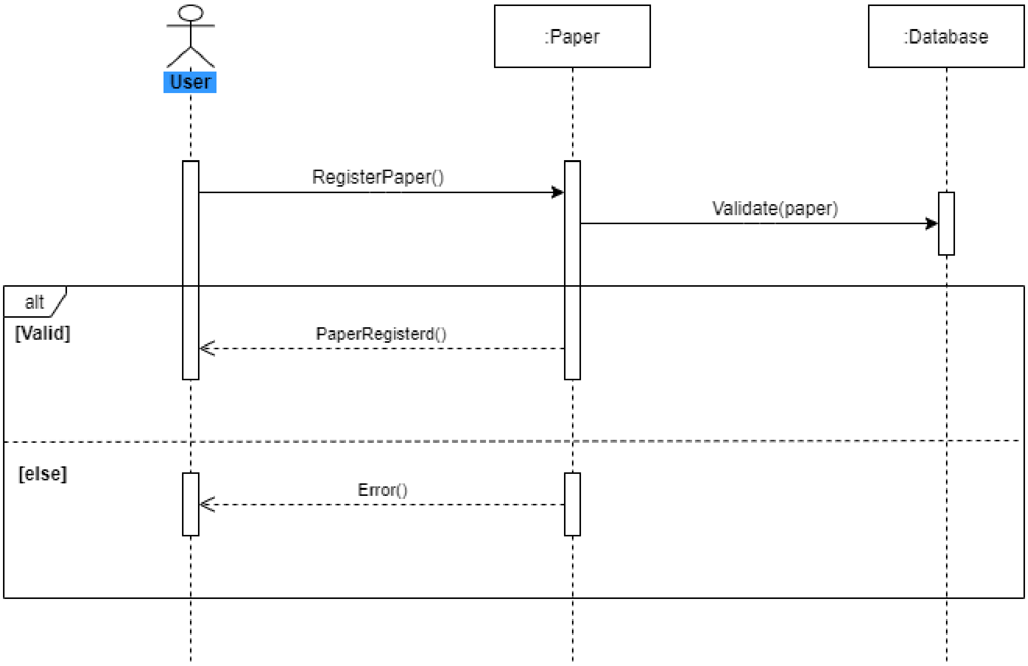


Figure 3.22: Sequence diagram for Input Paper Details

### 3.4.11 Sequence Diagram (Log Out)

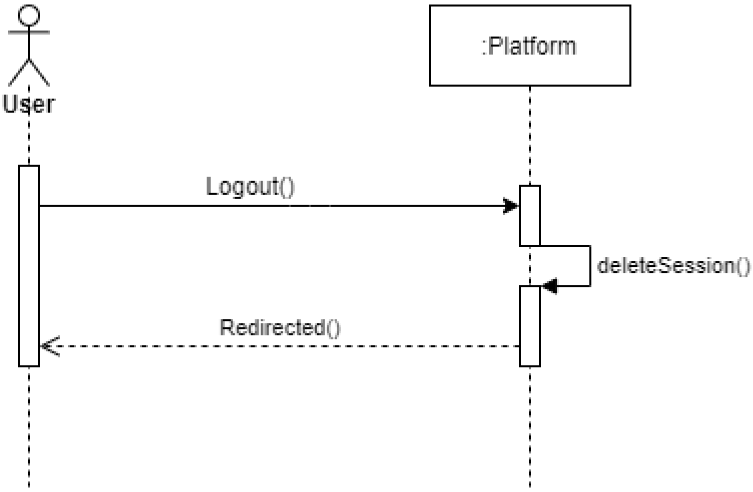
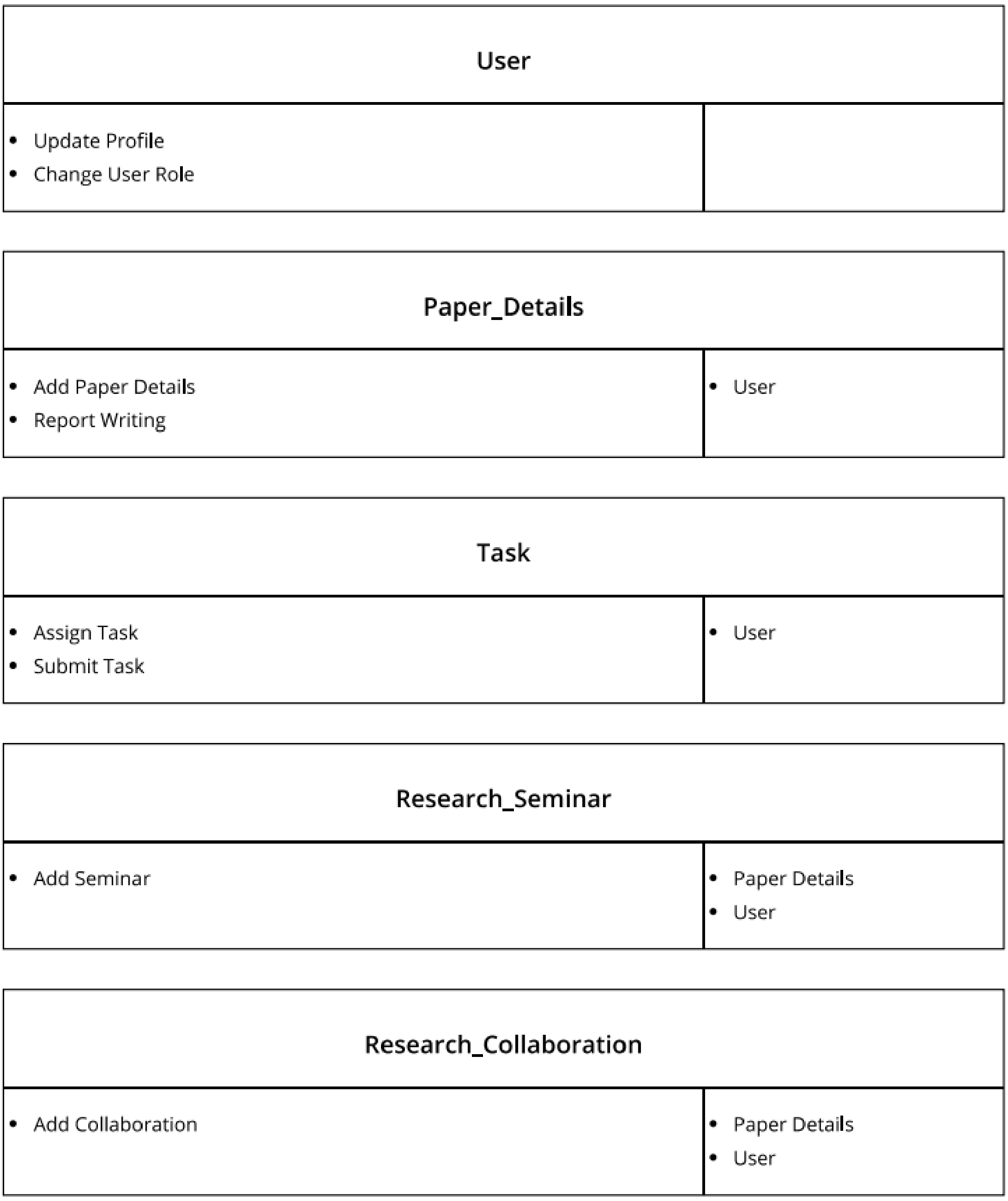


Figure 3.23: Sequence diagram for Log Out

# Chapter 4: System Design Specification

## 4.1 Class Responsibilities Collaboration (CRC) Cards



| User | |
|---|---|
| • Update Profile<br>• Change User Role | |

| Paper_Details | |
|---|---|
| • Add Paper Details<br>• Report Writing | • User |

| Task | |
|---|---|
| • Assign Task<br>• Submit Task | • User |

| Research_Seminar | |
|---|---|
| • Add Seminar | • Paper Details<br>• User |

| Research_Collaboration | |
|---|---|
| • Add Collaboration | • Paper Details<br>• User |

Figure 4.1: CRC cards for “Research Coordinating System”

## 4.2 Class Diagram

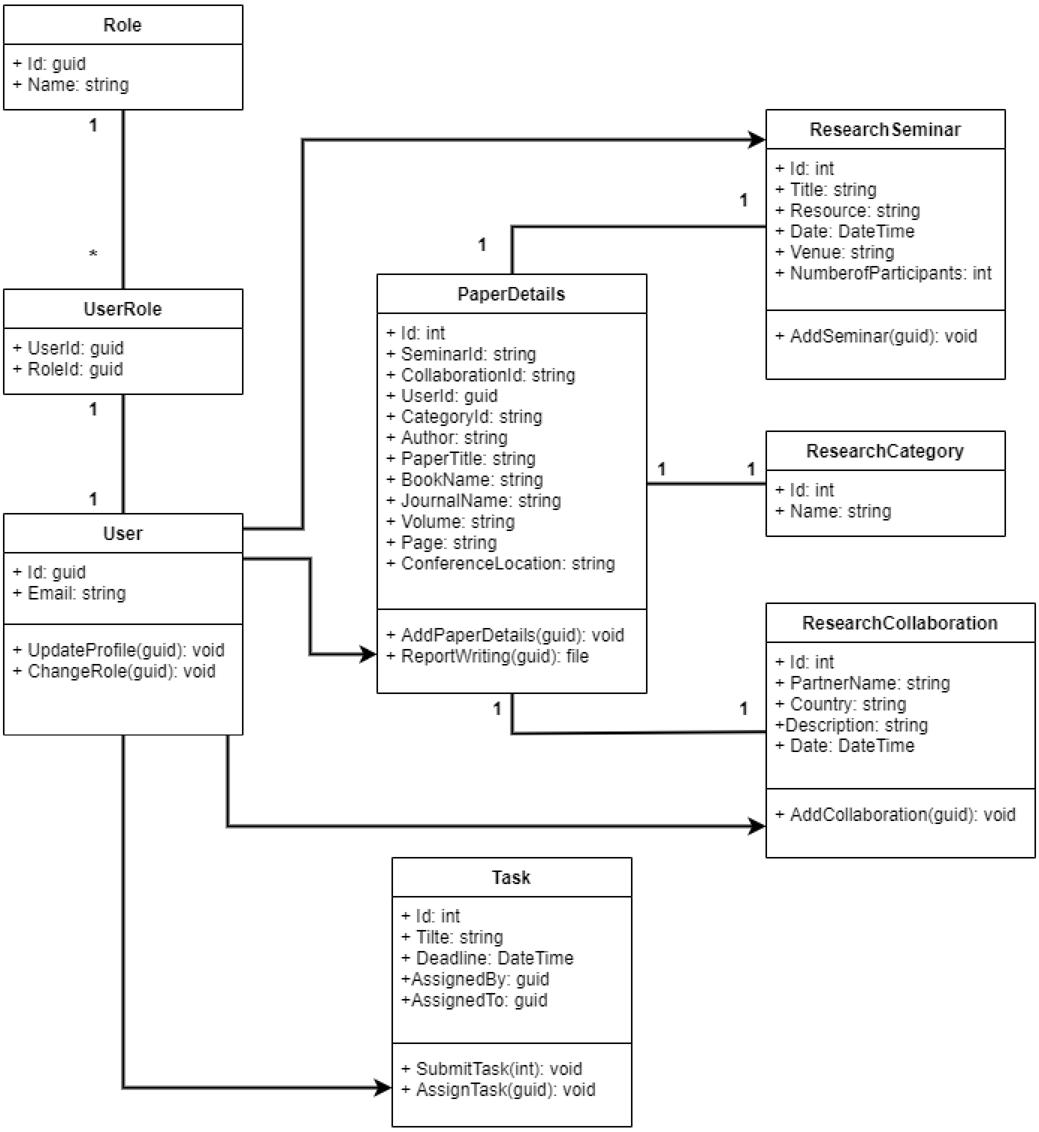


Figure 4.2: Class Diagram for “Research Coordinating System”

## 4.3 Database Design Diagram

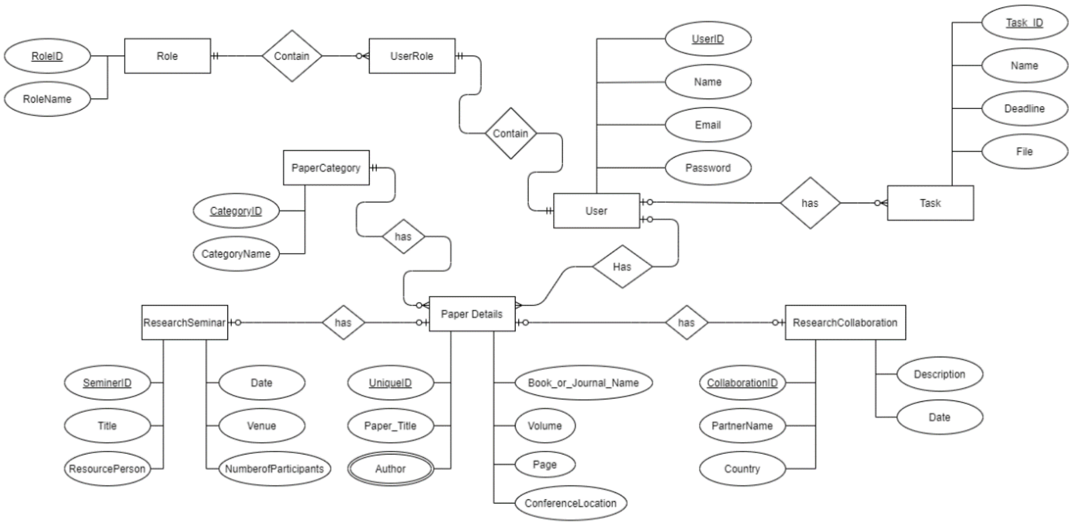


Figure 4.3: Database Design Diagram

## 4.4 Development Tools & Technology

For developing a quality software, development tools are used. Various development tools like programming tools, debugging tools, testing tools and so on are used to develop different types of applications. For the “Research Coordinating System” web application I also use those types of tools and technologies. They are:

### 4.4.1 User Interface Technology

The user interface means the visual part of a software. This interface has been created following the needs of the users so that they can easily interact with the system. The ultimate goal of the user interface is to deploy the user to the system. A good user interface makes an application effective, reliable and efficient.

### 4.4.1.1 ASP.NET MVC Framework

For the “Research Coordinating System” web-based application, the ASP.NET MVC framework is used. Security is a valuable part of web-based applications. ASP.NET offers more secure than many other frameworks. MVC (Model View Controller) is also a good architecture for web-based systems.

### 4.4.1.2 jQuery

In this system, ajax jQuery is used to retrieve data from dB context to datatables in user interface. It makes the data more effective and usable. It also decreases the retrieve time.

#### 4.4.1.3 CSS Framework and Bootstrap

CSS means “Cascading Style Sheets”. It helps the html elements to appear in a good-looking way. Bootstrap makes our web-application responsive. It’s a free and open-source framework. Bootstrap is also used in the “Research Coordinating System” application so that the layout matches the variety screen size easily.

### 4.4.2 Implementation Tools & Platforms

Selecting the tools and platforms applied is also an important factor in getting the application done properly. Anyone who wants to apply must analyze which equipment and platform is appropriate with the system. So, another challenge for the developer is to find the best tools to optimize his/her application.

#### 4.4.2.1 Microsoft Visual Studio 2019

A code editor or IDE is required to develop an application system. An IDE is used to edit the source code of applications. My used IDE for this project is Microsoft Visual Studio 2019. The community version is free and it has many great features to standardized source code for any application.

#### 4.4.2.2 MSSQL Server 2018

MSSQL server is used in this application as database server. Database server refers to the back-end system of a database application. MSSQL server is free, easy to use and it also maintain security angles.

#### 4.4.2.3 .NET Runtime

CLR (Common Language Runtime) is an application virtual machine that provides services like protection, memory management and exception handling. .NET runtime is free CLR by Microsoft. The latest version is cross platform.

## Chapter 5: System Testing

### 5.1 Testing Features

### 5.1.1 Features to be tested

- Log in
- Register new faculty
- Assigning Coordinator
- Adding paper details
- Assigning task

### 5.1.2 Features not to be tested

### 5.2 Testing Strategies

### 5.2.1 Unit Testing

Unit is the smallest testable part of an application like function, classes, procedures, interfaces. Unit testing is created and executed by software developer during the development process.

- **Log in**

Table 5.1: Log in

| | |
|---|---|
| Test case: TC001 | Test designed by: Sabbir |
| Test priority: High | Test design date: 01-12-2020 |
| Model name: Log in | Test executed by: Sabbir |
| Description: User can log in to the system with valid credentials. | Test execute date: 01-12-2020 |

- **Register New Faculty**

Table 5.2: Register New Faculty

| | |
|---|---|
| Test case: TC002 | Test designed by: Sabbir |
| Test priority: High | Test design date: 01-12-2020 |
| Model name: Register new faculty | Test executed by: Sabbir |
| Description: Registration process done by the admin or coordinator. | Test execute date: 01-12-2020 |

- **Assigning Coordinator**

Table 5.3: Assigning Coordinator

| | |
|---|---|
| Test case: TC003 | Test designed by: Sabbir |
| Test priority: High | Test design date: 01-12-2020 |
| Model name: Assign coordinator | Test executed by: Sabbir |
| Description: Admin assign a faculty member as coordinator. | Test execute date: 01-12-2020 |

- **Adding Paper details**

Table 5.4: Adding Paper Details

| | |
|---|---|
| Test case: TC004 | Test designed by: Sabbir |
| Test priority: Medium | Test design date: 01-12-2020 |
| Model name: Add paper details | Test executed by: Sabbir |
| Description: Faculty members add paper details they want to publish. | Test execute date: 01-12-2020 |

- **Assigning Task**

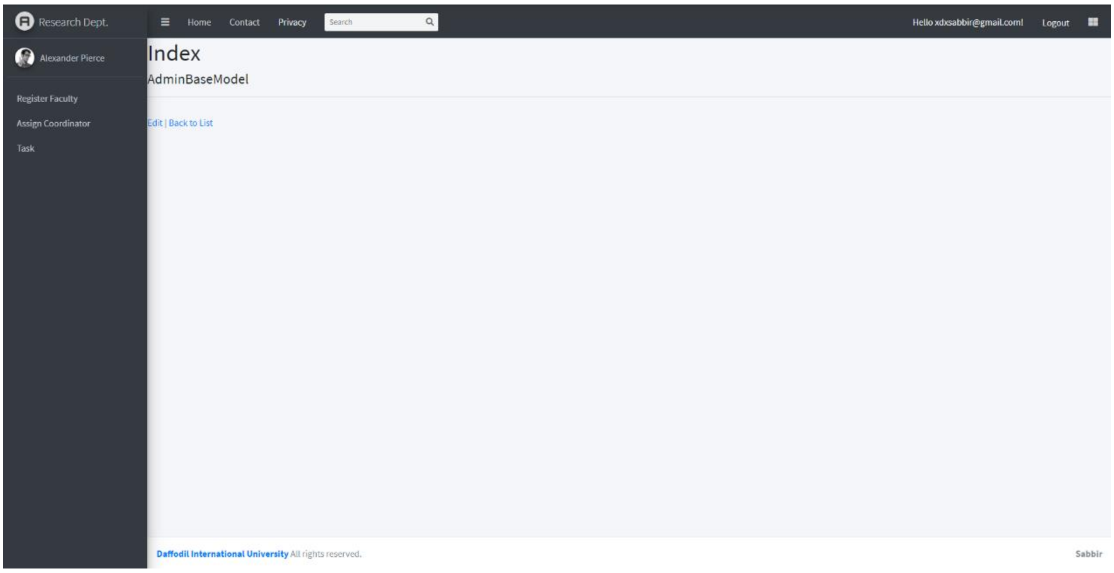
Table 5.5: Assign Task

| | |
|---|---|
| Test case: TC005 | Test designed by: Sabbir |
| Test priority: Medium | Test design date: 01-12-2020 |
| Model name: Task | Test executed by: Sabbir |
| Description: Admin and Coordinator assign task for the progress. | Test execute date: 01-12-2020 |

# Chapter 6: User Manual

## 6.1 User Manual (Admin)

### 6.1.1 Admin Dashboard



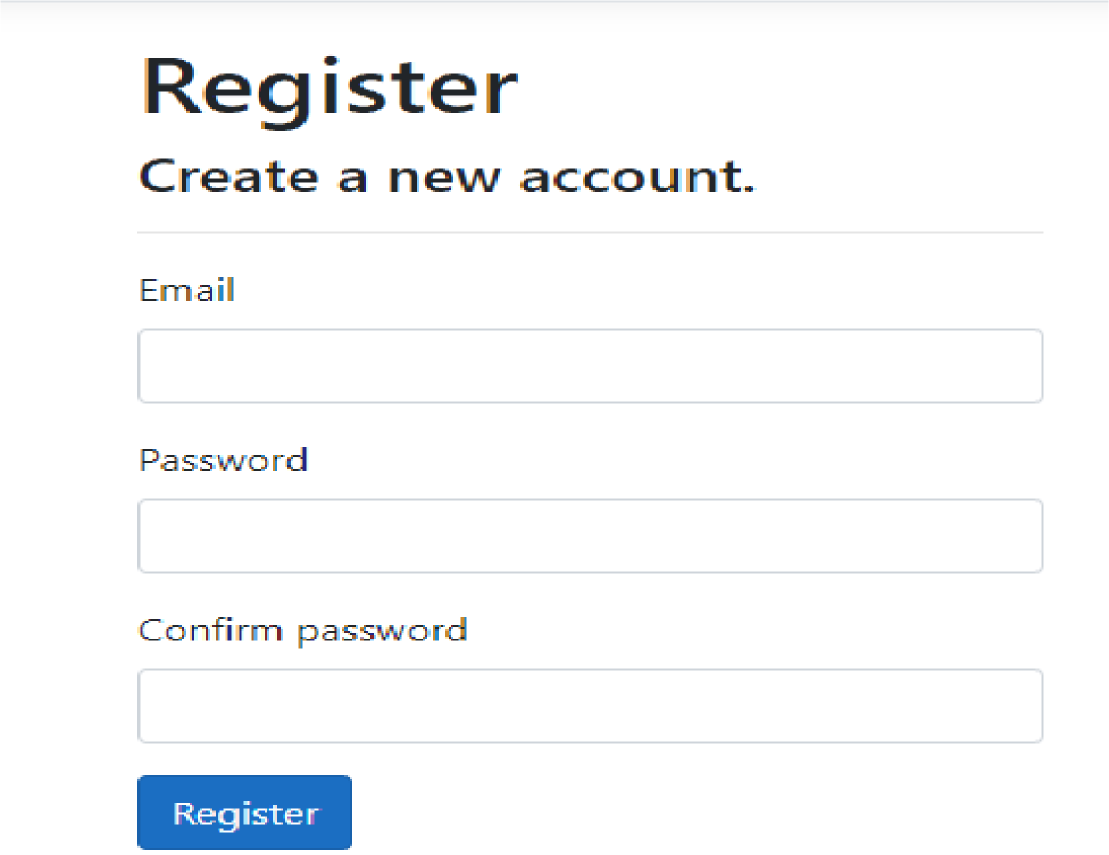
Index

AdminBaseModel

Daffodil International University All rights reserved.

Figure 6.1: Admin Dashboard

### 6.1.2 Register New User



# Register

## Create a new account.

Email

Password

Confirm password

Figure 6.2: Register new faculty

### 6.1.3 Assign Coordinator

## 6.2 User Manual (Coordinator)

### 6.2.1 Dashboard

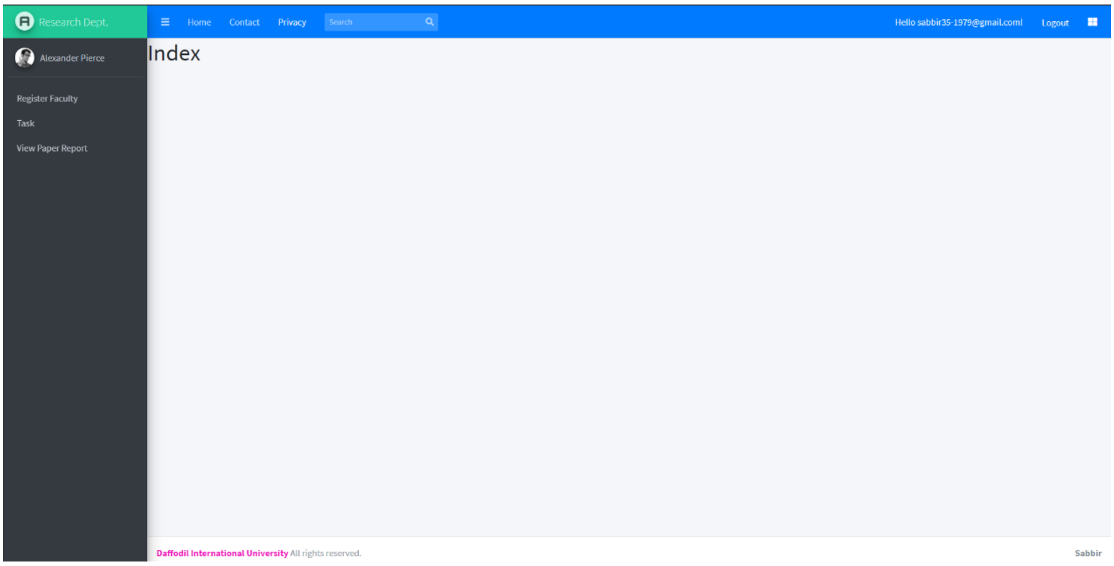
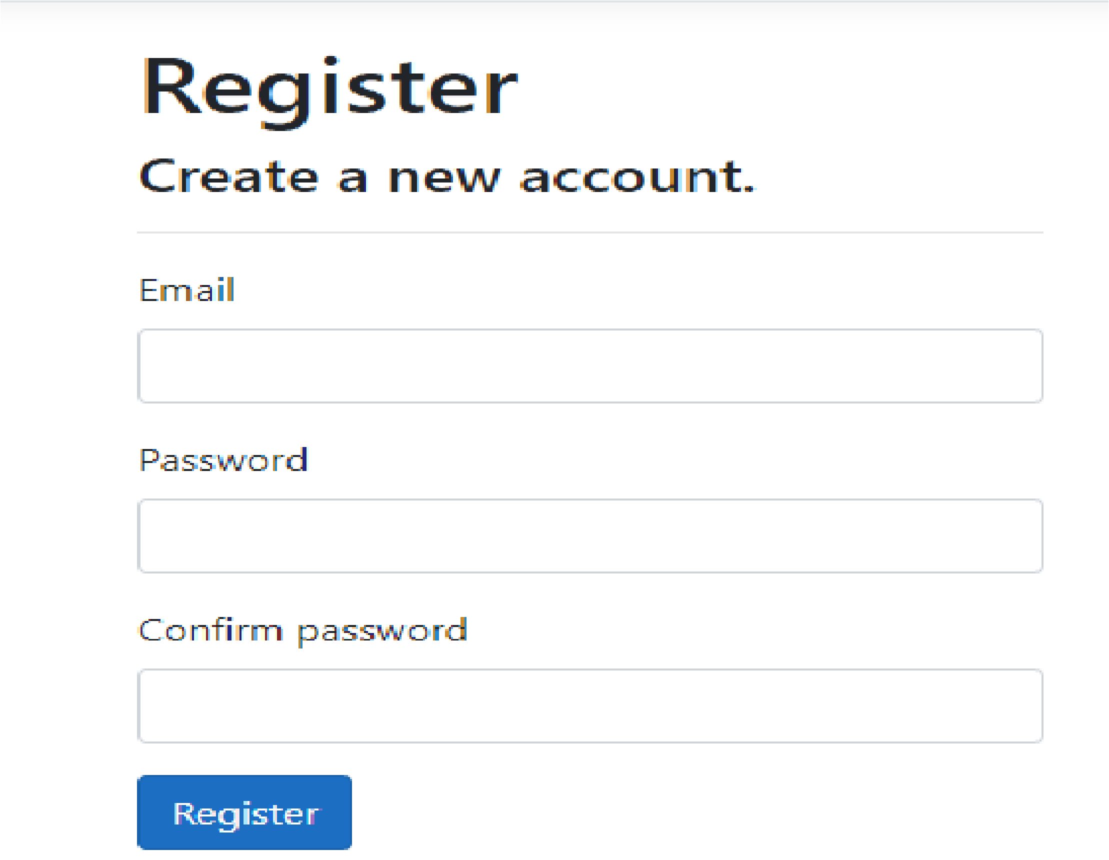


Figure 6.4: Coordinator Dashboard

### 6.2.2 Register New Faculty



# Register

## Create a new account.

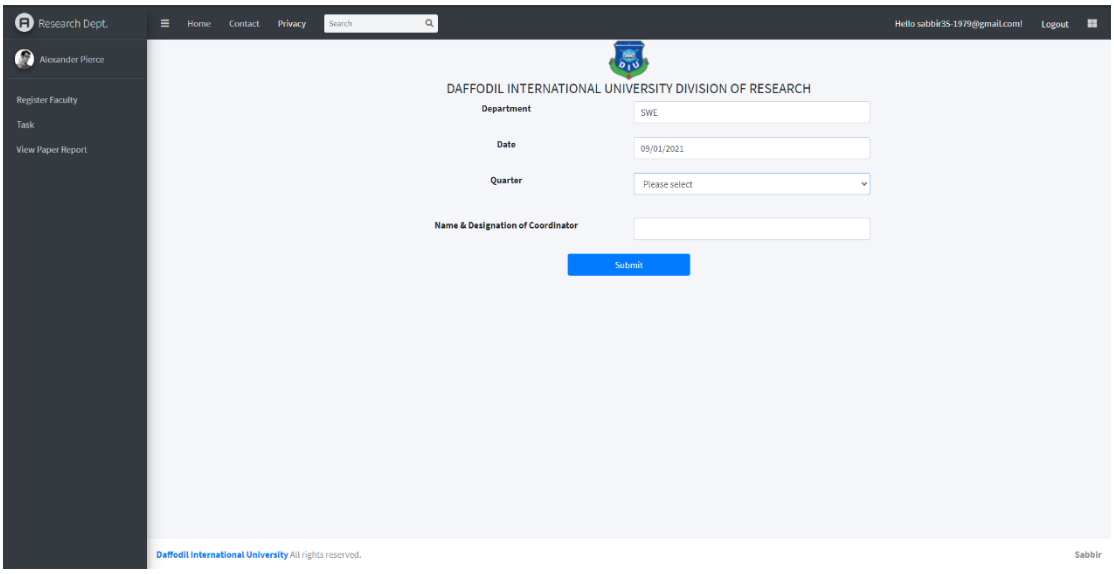
Email

Password

Confirm password

Figure 6.5: Register New Faculty

### 6.2.3 Report Writing



DAFFODIL INTERNATIONAL UNIVERSITY DIVISION OF RESEARCH

Department: SWE

Date: 09/01/2021

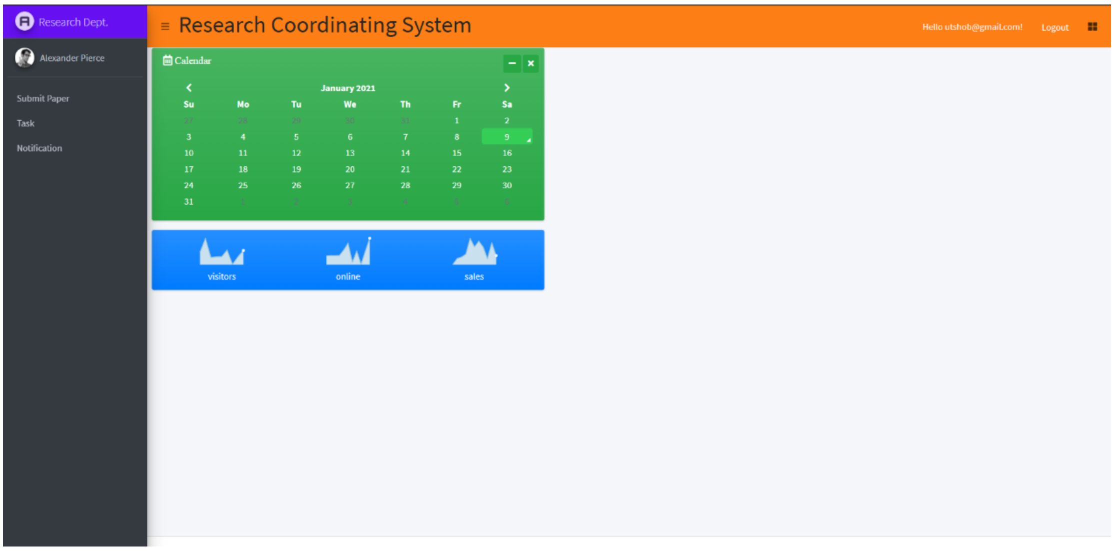
Quarter: Please select

Name & Designation of Coordinator:

Figure 6.6: Report writing

## 6.3 User Manual (Faculty)

### 6.3.1 Dashboard

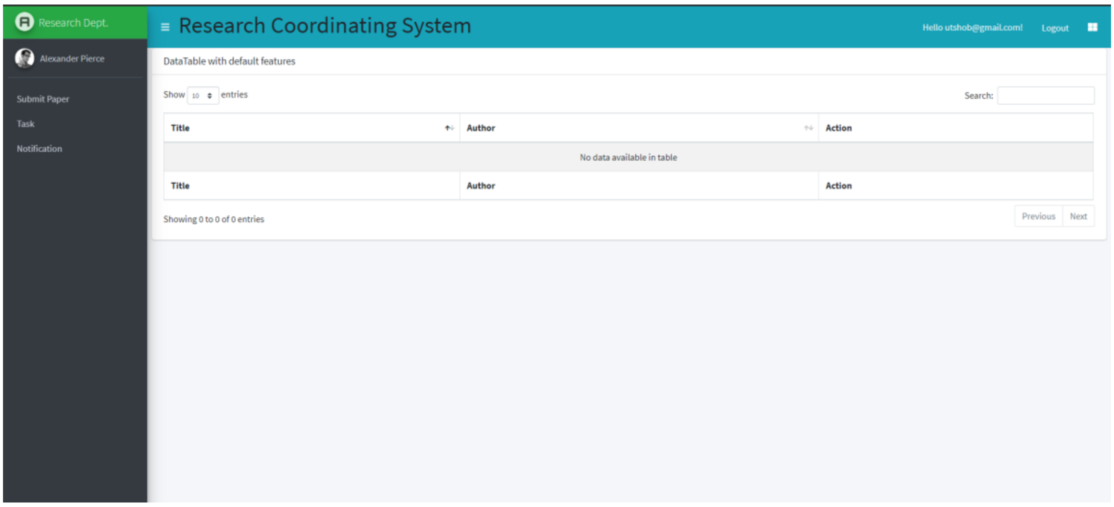


January 2021

| Su | Mo | Tu | We | Th | Fr | Sa |
|---|---|---|---|---|---|---|
| 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 |

Figure 6.7: Faculty Dashboard

### 6.3.2 Paper Details Submission



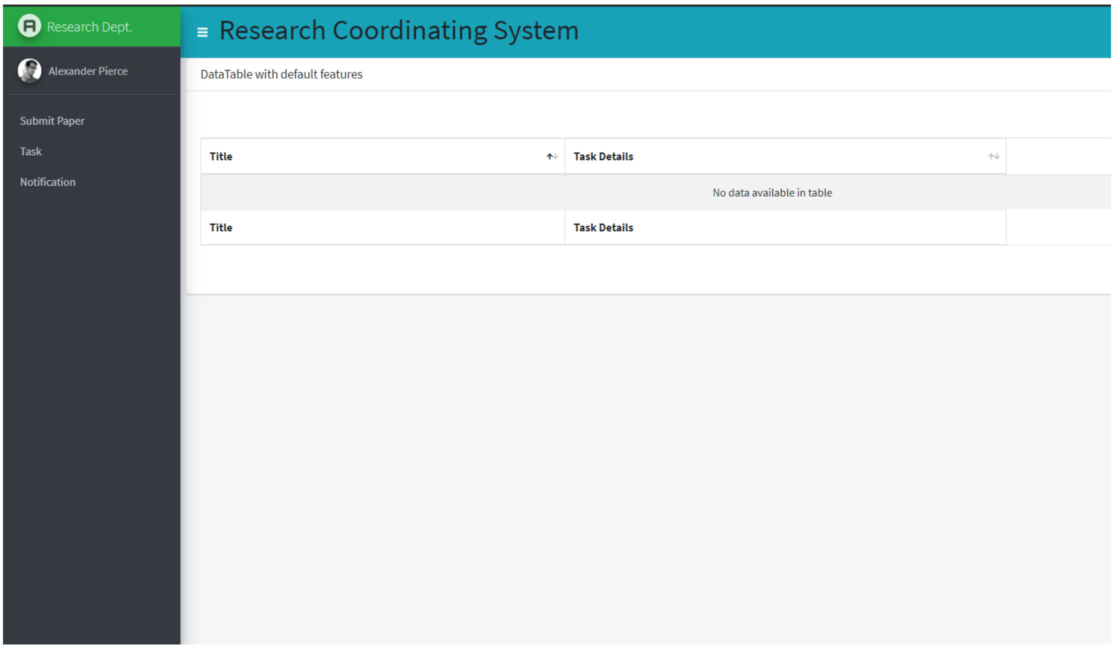
DataTable with default features

| Title | Author | Action |
|---|---|---|
| No data available in table | | |
| Title | Author | Action |

Showing 0 to 0 of 0 entries

Figure 6.8: Paper Details Submission and submitted

### 6.3.3 Tasks assigned



DataTable with default features

| Title | Task Details |
|---|---|
| No data available in table | |
| Title | Task Details |

Figure 6.9: Tasks

# Chapter 7: Project Summary

## 7.1 GitHub Link

https://github.com/I-am-Sabbir/Research_Coordinating_System

## 7.2 Limitations

Every application has some limitations as does this application.

- Notification alert can’t see at real time.
- Editing profile is limited.
- Mobile verification hasn’t set yet.

## 7.3 Future Scope

- More user-friendly frontend design.
- Overcome the limitations
- Connect worldwide departments with dynamic design

# Chapter 8: References

## 8.1 Documentation Idea:

Thomas T. Barker, (1998). *Writing Software Documentation*. Retrieved from https://cutt.ly/djMAyO3

## 8.2 System Requirement Specification:

Andreas Rüping, ISBN: 978-0-470-85617-8, (September 11, 2003). *Agile Documentation: A Pattern Guide to Producing Lightweight Documents for Software Projects* (1st Chapter, Topic-19, P-(26-28)). Retrieved from https://cutt.ly/ajMSGj2

## 8.3 Structure and Architecture:

Luke Hohmann, (2003). *Beyond Software Architecture*. Retrieved from https://cutt.ly/2jMDe8U

## 8.4 Unified Modeling Language:

Luke Hohmann, (2003). *Beyond Software Architecture*. Retrieved from https://cutt.ly/2jMDe8U

## 8.5 Testing:

Paul C. Jorgensen, ISBN: 9781138628076, (November 28, 2013- by Auerbach Publications). *Software Testing: A Craftsman's Approach, Fourth Edition 4th Edition* (Part-3, Beyond Unit Testing, P-(207-219)). Retrieved from https://cutt.ly/JjMDAXS

## 8.6 Tools and Technologies:

**Diagrams and Design:** https://app.diagrams.net/

**Articles: https://cutt.ly/CjMD2c7**