



**University Institute of Information Technology,  
PMAS-Arid Agriculture University,  
Rawalpindi Pakistan**

## **Product Lifecycle Management**

*By*

**Abdullah Talat    18-ARID-2595**

**Sharjeel Wakeel    18-ARID-2709**

*Supervisor*

**Mr. Zeeshan Javed**

## ***Bachelor of Science in Computer Science (2018-2022)***

**The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.**

# **DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software documentation and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Abdullah Talat

Sharjeel Wakeel

# CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS(CS) “**Product Lifecycle Management**” was developed by “Abdullah Talat, **18-arid-2595**” and “Sharjeel Wakeel, **18-arid-2709**” under the supervision of “Mr. Zeeshan Javed” and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Science.

-----  
**Supervisor**

-----  
**External Examiner**

-----  
**Administrator UIIT**

# Executive Summary

Software is delivered to the end-user when it performs their task successfully but the management of software is too tough because software delivers after the completion of several phases. In our project, we manage these phases. In our project, there are two main users one is an admin who is able to create products and add members. Admin also has the authority to edit or delete the member and a product. Admin assign project to a member. And the other user is members who analyze what cost occur on a project which risk factor are faced during deployment how to increase the quality of the product and what resources need to accomplish the project. Member also check the feasibility of a product. In the existing system, we faced configuration problems even member isn't able to create an account or manage his project. He needs some training then he is able to use those systems. He tries to overcome this issue through our project.

# Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Zeeshan Javed” for personal supervision, advice, valuable guidance and completion of this project. We are deeply indebted to him for encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Abdullah Talat

Sharjeel Wakeel

# Abbreviations

<b>SRS</b>	Software Requirement Specification
<b>PC</b>	Personal Computer

# Table of Contents

<b>Introduction.....</b>	<b>1</b>
1.1 Brief .....	2
1.2 Relevance to Course Modules .....	2
1.3 Project Background .....	3
1.4 Literature Review .....	3
1.5 Methodology and Software Life Cycle.....	5
<b>Problem Definition.....</b>	<b>6</b>
2.1 Purpose .....	7
2.2 Product Functions .....	7
2.3 Proposed Architecture .....	7
2.4 Project Deliverables .....	8
2.5 Operating Environment .....	8
2.6 Assumptions and Dependencies .....	8
<b>Requirement Analysis .....</b>	<b>9</b>
3.1 Functional Requirments .....	10
3.2 Non – Functional Requirments .....	11
3.2.1 Usability.....	11
3.2.2 Reliability .....	11
3.2.3 Performance.....	11
3.2.4 Supportability .....	11
3.2.5 Design Constraints.....	11
3.2.6 Licensing Requirements .....	11
3.3 Use case Model.....	12
3.3.1 Use Case Diagarm .....	12
3.3.2 Actors Discription.....	16
3.3.3 Use Case Discription .....	17
<b>The Design .....</b>	<b>21</b>
4.1 UML Structural Diagrams .....	22

4.1.1	Component Diagram.....	22
4.1.2	System Component Diagram.....	24
4.1.3	Package Diagram.....	25
4.1.4	Deployment Diagram.....	26
4.2	UML Behavioral Diagrams.....	27
4.2.1	Activity Diagrams.....	27
4.2.2	State Machine Diagrams.....	29
4.3	UML Interaction Diagrams.....	30
4.3.1	Sequence Diagrams.....	30
4.4	Node Structure.....	31
4.5	Communication Design Protocol.....	32
<b>Implementation.....</b>		<b>33</b>
5.1	Communication Protocol Implementation.....	34
5.2	PC Application Implementation.....	36
5.3	Embedded Application Implementation.....	38
5.4	Wireless Sensor Application Implementation.....	51
<b>Testing and Evaluation.....</b>		<b>52</b>
6.1	Verification.....	53
6.1.1	Functional Testing.....	53
6.1.2	Static Testing.....	57
6.2	Validation.....	57
6.3	Usability Testing.....	57
6.4	Unit Testing.....	57
6.5	Integration Testing.....	57
6.6	System Testing.....	57
<b>GUI.....</b>		<b>58</b>
<b>Future Work.....</b>		<b>70</b>
<b>References.....</b>		<b>72</b>



# List of Figures

Fig 3.1: Use Case diagram.....	21
Fig 3.2: ERD.....	23
Fig 4.1: System Architecture .....	23
Fig 4.2: Class diagram .....	24
Fig 4.3: View Member Activity Diagram.....	24
Fig 4.4: Assign Project Activity Diagram .....	25
Fig 4.5: Add Stakeholder Activity Diagram.....	26
Fig 4.7: Add Risk Activity Diagram.....	26
Fig 4.8: Add Procurement Activity Diagram .....	27
Fig 4.9: Feasibility Study Activity Diagram.....	28
Fig 4.10: Add Cost Activity Diagram.....	29
Fig 5.1: Sequence Diagram Add Member .....	29
Fig 5.3: Sequence Diagram sign in.....	30
Fig 5.4: Sequence Diagram View Project.....	30
Fig 5.5: Sequence Diagram Add Procurement .....	31
Fig 5.6: Sequence Diagram Add Schedule .....	32
Fig 5.7: Sequence Diagram Add Risk .....	32
Fig 5.8: Sequence Diagram Add Cost .....	33

# List of Tables

Table 3.1 Sign in.....	32
Table 3.2 Create project.....	32
Table 3.3 Add User.....	33
Table 3.4 View Project .....	34
Table 3.5 Assign Project.....	37
Table 3.6 Notificaion .....	36
Table 3.7 Chat.....	36
Table 3.8 Feasibilty Study .....	37
Table 3.9 Add Scope.....	38
Table 3.10 Add Risk .....	38
Table 3.11 Add Schedule.....	39

# Chapter 1: Introduction

In this chapter we will discuss over view of our whole project, its brief introduction, and how it is relevant to the courses which we have studied during our degree. We will also discuss project literature review and its analysis and methodology that we will use in project.

## 1.1. Brief

A process that is suitable to carry out a software project usually consists of several phases going far beyond the implementation of the software. These phases include, among others, project management, requirements analysis and management, quality assurance, and of course the technical phases such as implementation, testing, and maintenance. A holistic development process should address all these relevant phases of the software lifecycle, which we also refer to as project disciplines. Jira and FlinkISO are the software that provides the platform where you manage the product lifecycle but these software have some limitations. In our project, we overcome discussing software issues. In our system, there are three actors one of the project admin who is able to create projects, add members, view projects, view member's detail, and assign projects to members. Project Admin is also able to edit and delete the project and member detail and chat with members. On the other hand, members are able to see assigned projects and the history of recent assign project detail. Member able to risk analysis, schedule management, procurement management, resources management, cost management, review proposal, integration management, stakeholder management, and communicate with the admin. The last actor who is a web manager who manages a website adds new features and if a new language introduces in the market, then he adds in the select box field if a stakeholder of a system faces some issue in the website then he overcomes them. He is able to view the overall traffic of the system.

## 1.2. Relevance to Course Modules

Almost everything which we are using in our project “Product Lifecycle Management” is totally relevant to our course materials.

- The website which we are developing in our project is relevant to the subject “Web Development”.
- The data of the users which will be stored in Database and status will be shown on screen, is relevant to the subject Database Management System.

### 1.3. Project Background

Generally, when we want to develop something, first of all, we make a strategy where we start first and where we end, where we focus most you answer these questions. After that, you start working on your project to achieve the desired goal but what if a strategy doesn't support you to achieve your desired goal? what if the platform interface is difficult and you want to make a strategy or take a guess? What if you aren't able to create an account on these platforms?. The existing system has a limitation which we raise in the form of a question in the above discussion. So our aim is to provide users simple and user-friendly interface where he easily does their work and easily configure his account and communicate with each other.

### 1.4. Literature Review

Most of the prevailing product lifecycle management doesn't have a significant system. Some of them don't have a communication system between the team and some of them don't have a user-friendly interface. Some of them need to give training for use because of the difficult interface. and some of them are difficult to configure so these issues occur with the existing system. Every system is good but incomplete to provide complete facilities.

**Flinkiso** is an existing system in which we manage, audit the project. We also assure project quality but this platform has a problem that is following.

- It doesn't have defect tracking
- It doesn't have a feature of managing the equipment.
- It doesn't manage the risk.
- It doesn't not manage maintenance.

**Jira** is a software application used for issue tracking and project management. The tool, developed by the Australian software company Atlassian, has become widely used by agile

development teams to track bugs, stories, epics, and other tasks. But it has following limitations.

- The tool is hard to set up and get used to.
- It's complicated user interface can make managing tasks difficult.
- No built-in timeline to track your project progress.
- No collaboration features to communicate with your team.
- It's mainly built for engineering and software development teams.
- No idea management features to keep track of your ideas and plans.
- The tool can be expensive.
- It's known for being a slow tool with long query load times.

### **1.5. Analysis from Literature Review (in the context of your project)**

The goal is to provide user-friendly interface users easily interact with our system and do their work. They don't need the training to use this system. They easily create an account on this system because this system provides an easy method to create an account. This system has communication where teammates communicate with each other.

### **1.6. Methodology and Software Lifecycle for This Project**

We use Agile methodology. The proposed system is going to follow the agile development method because this method assists in responding to the unpredictability of constructing software. This method offers a light framework and focus on rapid delivery of the software. Moreover, agile method facilitates us to create and respond to change in an uncertain and turbulent situation. Our focus is to collaborate with stakeholders during project to ensure the product quality which is impossible without using agile methodology.

### **1.6.1. Rationale behind Selected Methodology**

- Creativity and innovation
- Lower costs
- Improved quality
- Customer satisfaction
- Focus on users
- Early and predictable delivery
- Predictable costs and schedule
- Allows for change

## Chapter 2: Problem Definition

### 2.1. Purpose

Existing systems are good but they have some weaknesses like their interface is not user-friendly and user account configuration is complex. Teammates don't communicate with each other. Sometimes users need the training to use these software because of the complex interface. These software have some limitations so they perform their task in some limitations so users faced some difficulty by using that software. Our aim is to provide a user-friendly interface that helps users to use the software easily. they easily configure their account on this software. They communicate with teammates and project admin.

### 2.2. Product Functions

The product functions of Product Lifecycle Management are, **Register:** In registration the project admin can add member personal information and the information can be verified and store in to the database and the member account created **Login:** The user provide their account credential after that he see his profile **.Manage Database:** The web admin can manage database i.e., update user, add user, view user, search users, delete users etc. **Create Project:** Project admin can add a new project he adds the basic idea of a project with a proposal and submit after submission form validate if it validates successfully then it project detail store in database and shows success message. **View Projects:** Project admin can view all project detail also able to edit and delete operations on project details. **View Member:** Project admin can view all project detail also able to edit and delete operations on project details. **Assign project:** Project admin able to assign a project to a member he put a project and work detail and submit form after submission form validate then store into database and system show success message. **Add Cost:** Member can add module cost of assign project after addition he submit form after validation store into database then system show success message. **Add Schedule:** Member can add module schedule of assign project after addition he submit form after validation store into database then system show success message. **Add Risk:** Member can add module risk of assign project after addition he submit form after validation store into database then system show success message. **Add Resources:** Member can add resources of assign

project after addition he submit form after validation store into database then system show success message. **Add Procurement:** Member can add module procurements of assign project after addition he submit form after validation store into database then system show success message. **Chat:** Teammates communicate to each other. **Notification:** Admin and member receive notification if action perform on project.

- login
- Manage Database
- Create Project
- View Project
- View Member
- Assign Member
- Add Cost
- Add Schedule
- Add Risk
- Add Resources
- Add Procurement
- Chat
- Notification

### **2.3. Proposed Architecture:**

3Tier Architecture has been used in our project.

We have three layers in 3Tier Architecture:

1. Presentation Tier
2. Application Tier
3. Data Tier

- **Presentation Tier:**

We will develop Web Application in which we will create interfaces by using HTML,CSS and BOOTSTRAP . These interfaces will display all the information of User.

- **Application Tier:**

In this layer we will apply logics in our Web Application by using JavaScript and PHP Language.

This tier- also called Middle tier, Logic tier and Business tier, pulled from presentation tier.



- **Data Tier:**

Data in this tier is kept independent of web servers and Business logic.

In this layer we will use Mysql which will manage to store the data of users.

## **2.4. Project Deliverables:**

Following are the deliverables and development requirements:

### **2.4.1 Projects Deliverables:**

In this project the deliverables will consist of the input as equipment (i.e., hardware components and software components) then there will be a process (i.e., development phases in which development will be completed) applied on the input and then as a result there will be an output (Project being completed “Product Lifecycle management”). In this case product deliverables are the completed parts or modules of the project. Input will be software and software-based components. The project is divided into different modules and each module is a major milestone in the project.

### **2.4.2 Development Requirements:**

Development requirements are the requirements needed for the development purposes without which the development is not possible. It can be hardware, software or any kind of requirements. These include the software and hardware equipment's, time and date constraints, budget, planning, following the SDLC etc. development requirements are met accordingly to make sure that the end result does not differ from what is expected and that it can perform its functionality accurately and perfectly without any glitches.

## **2.5. Operating Environment:**

Operating environment for the Product Lifecycle Management is as listed below:

**Operating System:** Window 10

**Database:** Mysql

**Platform:** Visual Studio Code in which web application will be developed using PHP language. Front end will be designed in HTML,CSS,BOOTSTRAP and backend will be designed in PHP.

## **2.6. Assumptions and Dependencies:**

### **Assumptions:**

The Web Application developed from this technique is more efficient than others. This software gives perfect results in real-time for long queries. user don't face difficulties that they faced in an existing system.

### **Dependencies:**

This software is dependent on internet connectivity.

## Chapter 3: Requirement Analysis

In this chapter we will define all the requirements of proposed system that include functional and non-functional requirements. We will also discuss about use cases of the system and see how our system will respond to various use cases.

### 3.1. Functional Requirements

For our system to work and facilitate the user number of functional requirements have been are needed. These functional requirements also be presumed as interface requirements as they are all but interface.

- The system will authenticate the user through login functionality.
- The system will display the previous projects.
- The system will display the member list to project admin.
- The system will allow project admin to create new projects .
- The system will allow project admin to perform edit delete operations on created projects.
- The system will allow to perform edit delete operation on member.
- System allow admin to assign project to different members.
- The system will allow member to test the feasibility study.
- The system will allow member to add cost.
- The system will allow member to add risk.
- The system will allow member to add Integration.
- The system will allow member to add stakeholder.
- The system will allow member to add module.
- The system will allow member to add procurement.
- The system will allow member to add Resources.
- The system will allow member to view proposal.
- The system will allow member to chat with its teammate and project admin.
- Member receive notification if teammate or project admin send message.

### **3.1. Non-Functional Requirements**

- The system should be able to handle the concurrent requests from different users.
- The system should provide confidentiality for user data.
- The system should be stable and reliable enough to handle the exceptions.
- Interface and the system itself should be user friendly so that the user will feel it easy to use.
- The system will authenticate the user by verifying the credentials to database.
- If user fill form completely then he will be able to submit otherwise system give error message on a specific fields.

### **3.1. Use Case Model**

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. Following are the use cases of the Product Lifecycle Management

### 3.3.1 Use Case Diagram:

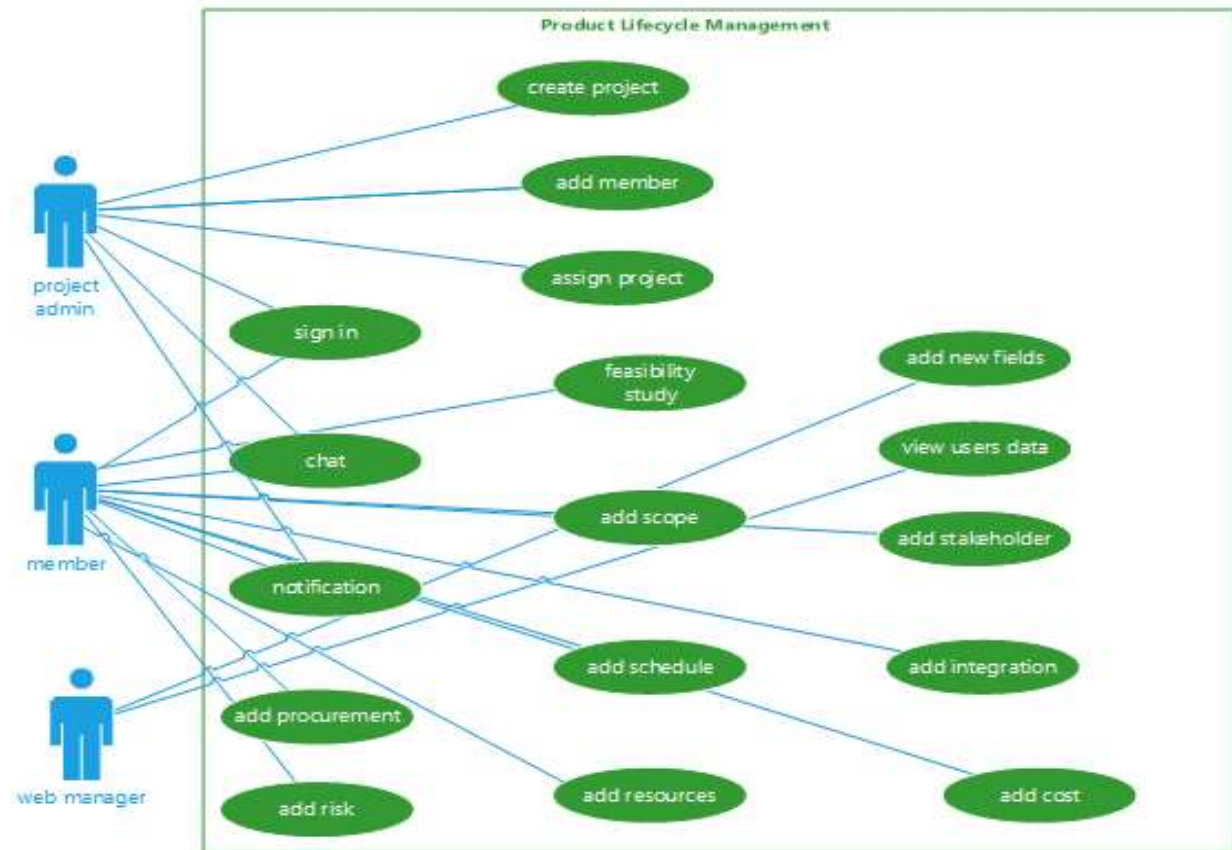


Fig 3.1: Use Case diagram

#### Actors Description:

**We have four types of actors.**

##### **Project manager:**

Project manager is a primary actor which directly interacts with a website. He can add projects ,add users, assign projects and is also able to edit and delete project, users.

##### **User:**

User is also the primary actor .He can view assigned projects and is able to accept or reject projects and perform some operation on the project.

##### **Web Manager:**

Web Manager can add new fields and manage the website.

##### **Database:**

Database stores all the information and data fetched through it as well.

### Use Case Description

<b>User Case Name:</b>	ID-01
<b>Use Case Name:</b>	Sign in
<b>Actors:</b>	Project admin, user
<b>Description:</b>	User will provide its credentials and be authenticated by the system through the database.
<b>Trigger:</b>	When the user clicks on the sign in button
<b>Preconditions:</b>	User should be sign in page
<b>Postconditions:</b>	User will login successfully and redirect to home page
<b>Normal Flow:</b>	1.user enter email and password. 2.system validates the email and password. 3.system match email and password through database 4.system redirect to home page
<b>Alternative Flows:</b>	2a. Error show please fill required fields 3a.Error show please enter valid email and password 4a.don't redirect to the home page

**Table 3.1 sign in**

<b>User Case ID:</b>	ID-02
<b>Use Case Name:</b>	Create Project
<b>Actors:</b>	Project admin
<b>Description:</b>	Project admin clicks on create project option and fills the field after the validation form is submitted and gets a successful message.
<b>Trigger:</b>	When the admin clicks on create project.
<b>Preconditions:</b>	Admin should be on the home page.
<b>Postconditions:</b>	Fill the form and submit an admin get success message.
<b>Normal Flow:</b>	1.click on the add project option. 2.fill all fields. 3.validate all fields. 4.store in the database. 5.get a success message.
<b>Alternative Flows:</b>	3a.Error show please fill the required field.

**Table 3.2: Create Project**

<b>Use Case ID:</b>	ID-03
<b>Use Case Name:</b>	Add User
<b>Actors:</b>	Project admin
<b>Descriptions:</b>	Project admin clicks on add user option and fills the field after the validation form is submitted and gets a successful message.
<b>Trigger:</b>	When the admin clicks on add user.
<b>Preconditions:</b>	Admin should be on the home page.
<b>Postconditions:</b>	Fill the form and submit an admin get success message.

<b>Normal Flow:</b>	1.click on the add project option. 2.fill all fields. 3.validate all fields. 4.store in the database. 5.get a success message.
<b>Alternative Flows:</b>	3a.Error show please fill the required field.

**Table 3.3: Add User**

<b>Use Case ID:</b>	ID-04
<b>Use Case Name:</b>	View Projects
<b>Actors:</b>	Project Admin
<b>Descriptions:</b>	Project admin click on view project and view projects
<b>Trigger:</b>	When a user clicks view project.



<b>Preconditions:</b>	Admin should be on the home page.
<b>Postconditions:</b>	View all projects with project id and project title with little description.
<b>Normal Flow:</b>	1.click on the view project. 2.View all projects project id and project title with little description.

**Table 3.4: View Projects**

<b>Use Case ID:</b>	ID-05
<b>Use Case Name:</b>	View User
<b>Actors:</b>	Project Admin
<b>Descriptions:</b>	Project admin click on view users and view users.
<b>Trigger:</b>	When a user clicks view user.
<b>Preconditions:</b>	Admin should be on the home page.
<b>Postconditions:</b>	View all user with user id and user name with designation.
<b>Normal Flow:</b>	1.click on view user. 2.View all user with user id and user name with designation.

**Table 3.5: View Users**

<b>Use Case ID:</b>	ID-06
<b>Use Case Name:</b>	Assign Project
<b>Actors:</b>	Project Admin
<b>Descriptions:</b>	Admin click on assign button then fill fields and submit get success message
<b>Trigger:</b>	When a user clicks the assign button.
<b>Preconditions:</b>	Admin should be on the view profile page.
<b>Postconditions:</b>	Admin fill fields press the assign button get success message
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1.click on the assign button.</li> <li>2.fill the form.</li> <li>3.Validate form field</li> <li>4.store in database</li> <li>5.system gives a success message</li> </ol>
<b>Alternative Flows:</b>	3a.Error show please fill required fields

**Table 3.6: Assign Project**

<b>Use Case ID:</b>	ID-07
<b>Use Case Name:</b>	View Notification
<b>Actors:</b>	Project Admin, User
<b>Descriptions:</b>	Users get notification through the system if someone performs some activity on a given project.
<b>Trigger:</b>	When user clicks on notification
<b>Preconditions:</b>	User should be on the view profile page.
<b>Postconditions:</b>	User check notifications.
<b>Normal Flow:</b>	1.click on notification. 2.check notifications.
<b>Alternative Flows:</b>	None

**Table 3.7:View Notification**

<b>Use Case ID:</b>	ID-08
<b>Use Case Name:</b>	View Chat
<b>Actors:</b>	Project Admin, User
<b>Descriptions:</b>	Users get notification through the system if someone texts.
<b>Trigger:</b>	When a user clicks the message button.
<b>Preconditions:</b>	User should be on the home page.
<b>Postconditions:</b>	See who text him and answer him

<b>Normal Flow:</b>	1.get alert 2.click on the message button. 3.see who sent a message to him. 4.answer him. 5.store answer in the database.
<b>Alternative Flows:</b>	None

<b>Use Case ID:</b>	ID-09
<b>Use Case Name:</b>	Study Feasibility
<b>Actors:</b>	User
<b>Descriptions:</b>	User click on feasibility option fill field.
<b>Trigger:</b>	When a user clicks the feasibility option.
<b>Preconditions:</b>	User should be on the home page.
<b>Postconditions:</b>	User fills the field and clicks the submit button to get success message.
<b>Normal Flow:</b>	1.click on the feasibility button. 2.fill the form. 3.validate form. 4.store in the database. 5.get a success message.

<b>Alternative Flows:</b>	3a.Errro show please fill required field.
---------------------------	---

**Table 3.9: Study Feasibility**

<b>Use Case ID:</b>	ID-10
<b>Use Case Name:</b>	Add Scope
<b>Actors:</b>	User
<b>Descriptions:</b>	User clicks on the add scope option to fill field.
<b>Trigger:</b>	When a user clicks the add scope option.
<b>Preconditions:</b>	User should be on the home page.
<b>Postconditions:</b>	User fills the field and clicks the submit button to get success message.

<b>Normal Flow:</b>	1.click on the scope button. 2.fill the form. 3.validate form. 4.store in the database. 5.get a success message.
<b>Alternative Flows:</b>	3a.Errro show please fill required field.

**Table 3.10: Add Scope**

<b>Use Case ID:</b>	ID-11
<b>Use Case Name:</b>	Add Risk
<b>Actors:</b>	User
<b>Descriptions:</b>	User click on add risk option to fill field.
<b>Trigger:</b>	When a user clicks the feasibility option.

<b>Preconditions:</b>	User should be on the home page.
<b>Postconditions:</b>	User fills the field and clicks the submit button to get success message.
<b>Normal Flow:</b>	1.click on the risk button. 2.fill the form. 3.validate form. 4.store in the database. 5.get a success message.
<b>Alternative Flows:</b>	3a.Errro show please fill required field.

**Table 3.11: Add Risk**

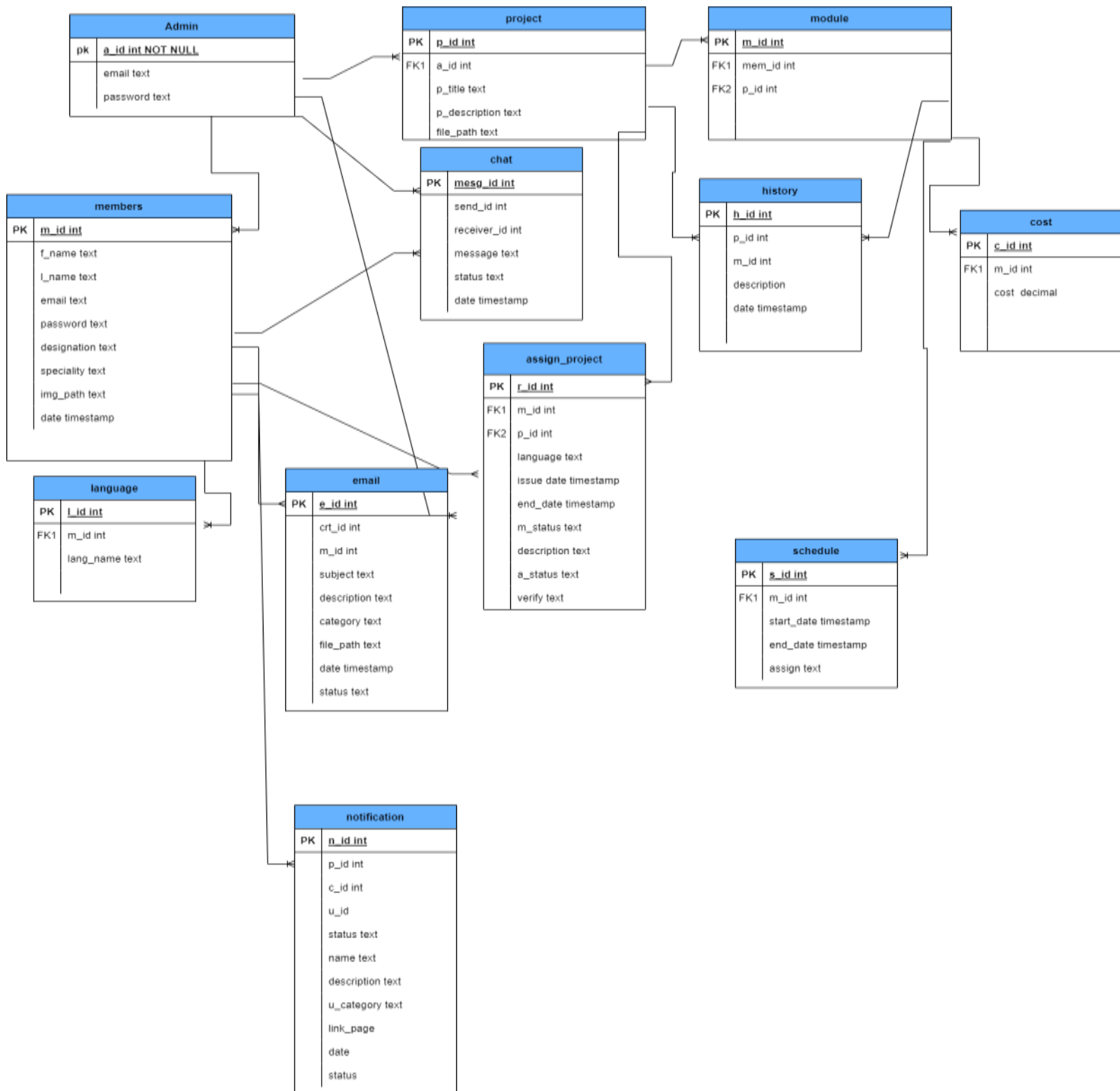
<b>Use Case ID:</b>	ID-12
<b>Use Case Name:</b>	Add schedule
<b>Actors:</b>	User
<b>Descriptions:</b>	User click on schedule option to fill the fields.
<b>Trigger:</b>	When a user clicks the add schedule option.

<b>Preconditions:</b>	User should be on the home page.
<b>Postconditions:</b>	User fills the field and clicks the submit button to get success message.
<b>Normal Flow:</b>	1.click on the schedule button. 2.fill the form. 3.validate form. 4.store in the database. 5.get a success message.
<b>Alternative Flows:</b>	3a.Error show please fill required field.

**Table 3.12: Add Schedule**



### 3.4 Entity Relational Diagram (ERD):



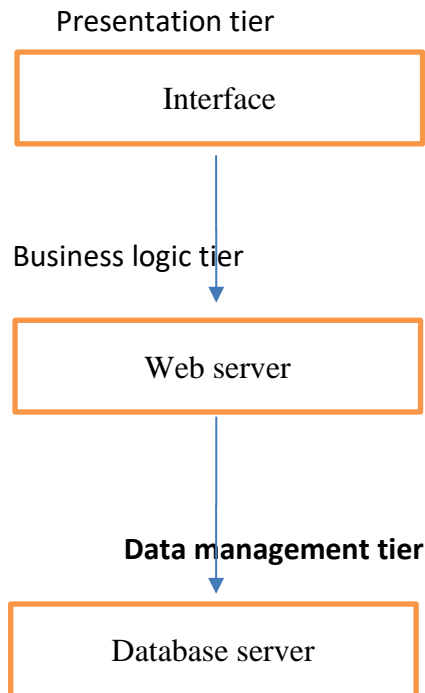
**Fig 3.2: ERD**

## **Chapter 4: Design and Architecture**

In this chapter we will discuss the design and architecture of our system.

### **4.1. System Architecture**

As system design varies from system to system, therefore user need to have the architecture view of the whole system.



**Fig 4.1: System Architecture**

### **4.2. System Design**

Systems design is the process of defining elements of a system like components, modules, architecture and their interfaces and data for a system based on the specified

requirements. The purpose of the System Design process is to provide sufficient detailed data and information about the system. Following is the system design of the Product Lifecycle Management.

#### 4.2.1 Class Diagram:

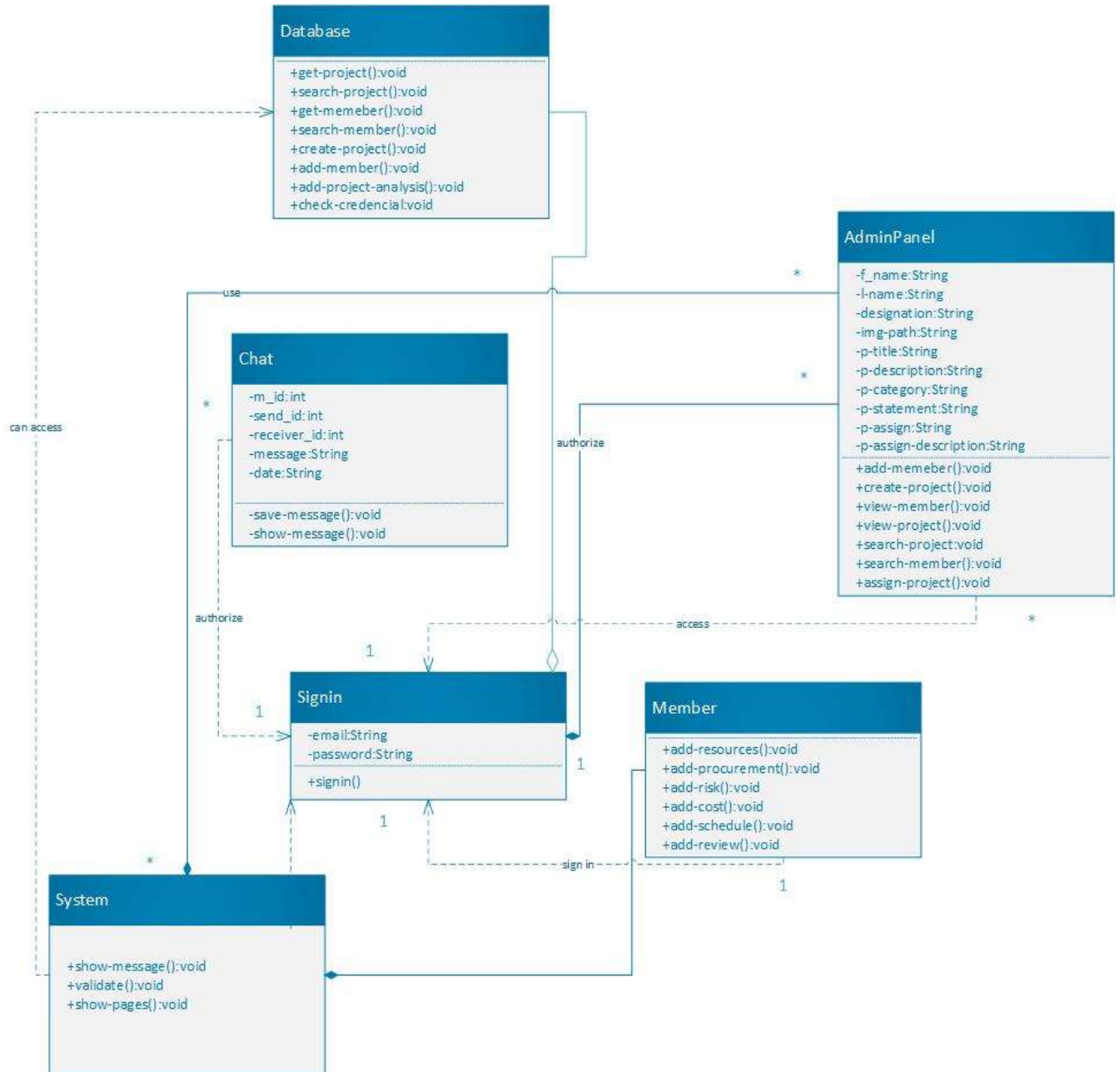
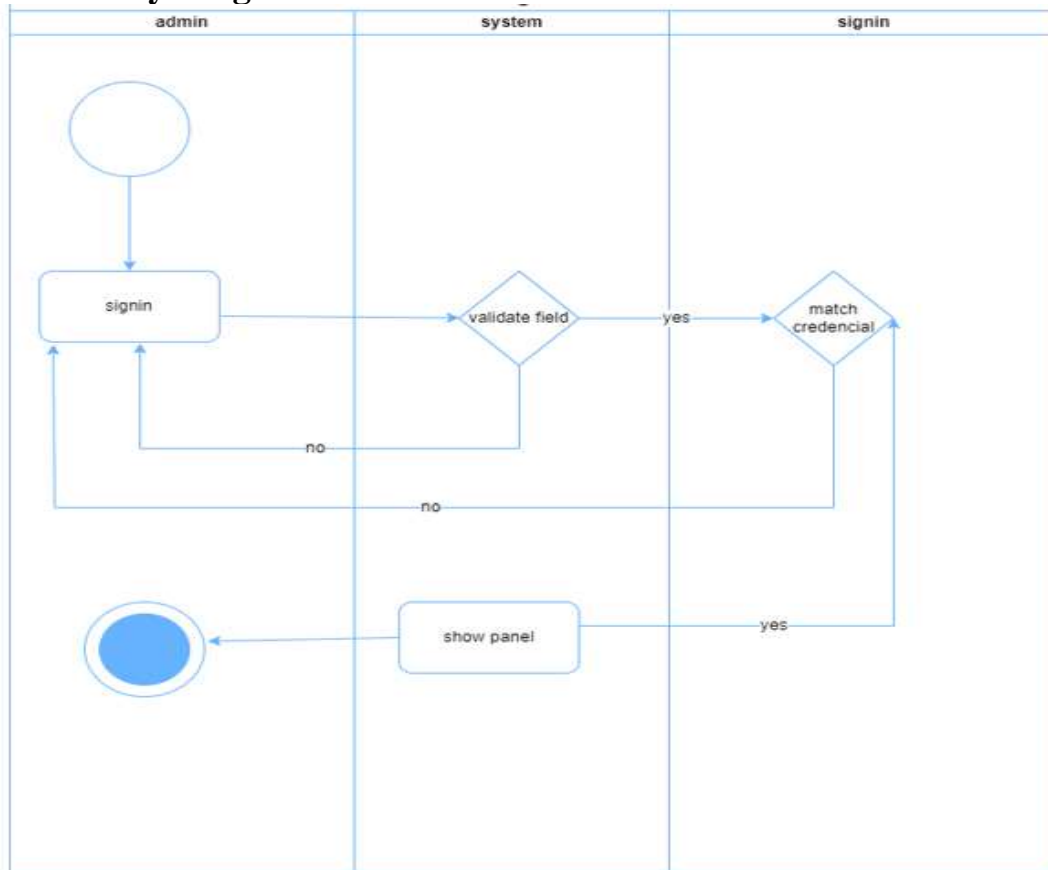
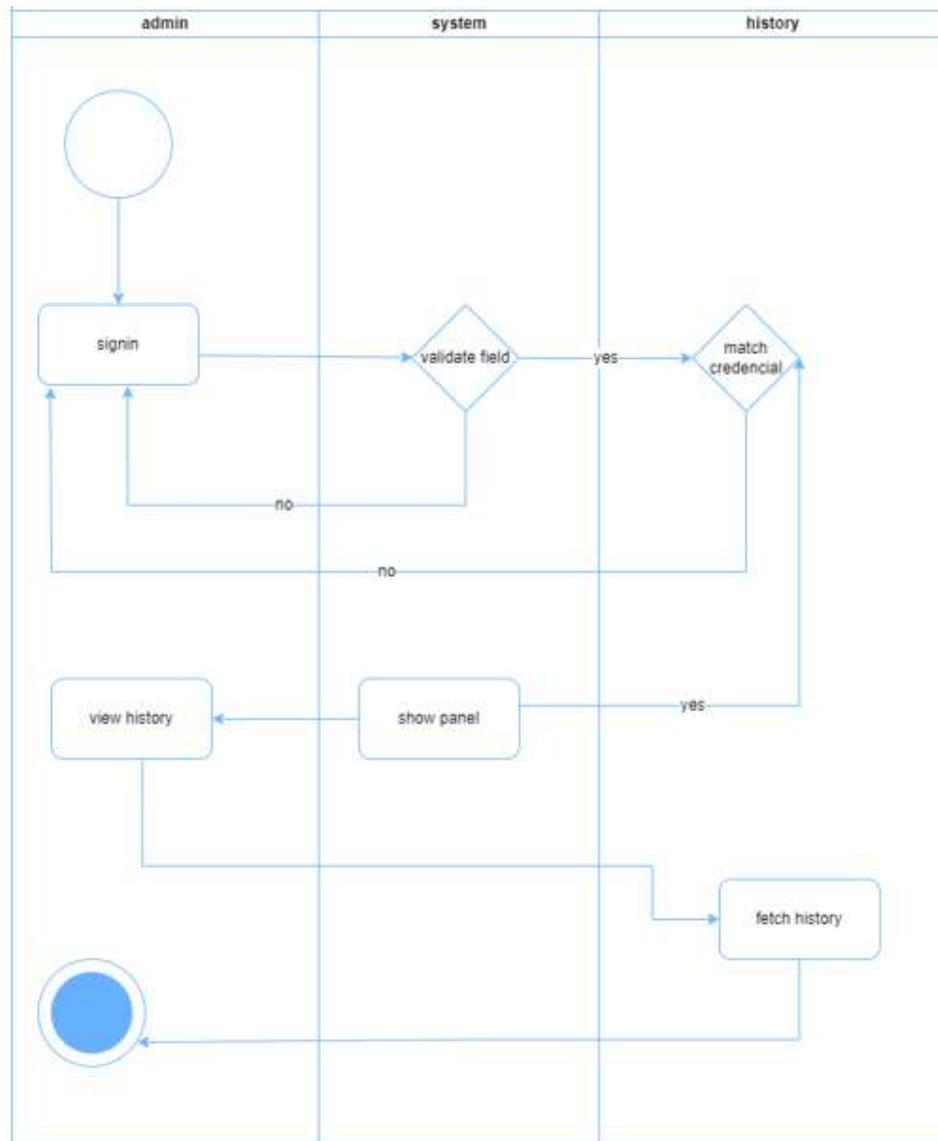


Fig 4.2: Class diagram

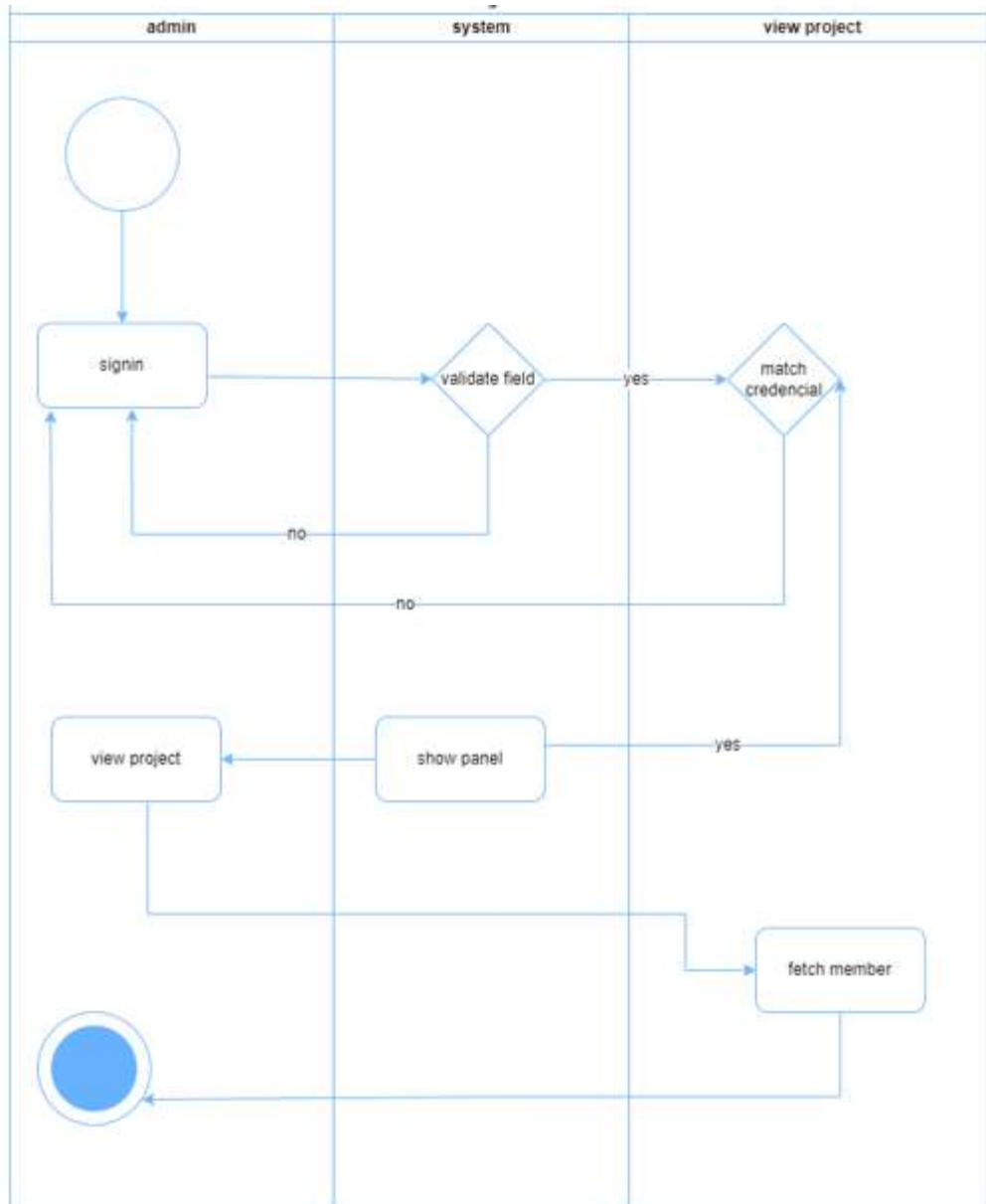
### 4.2.2 Activity Diagram:



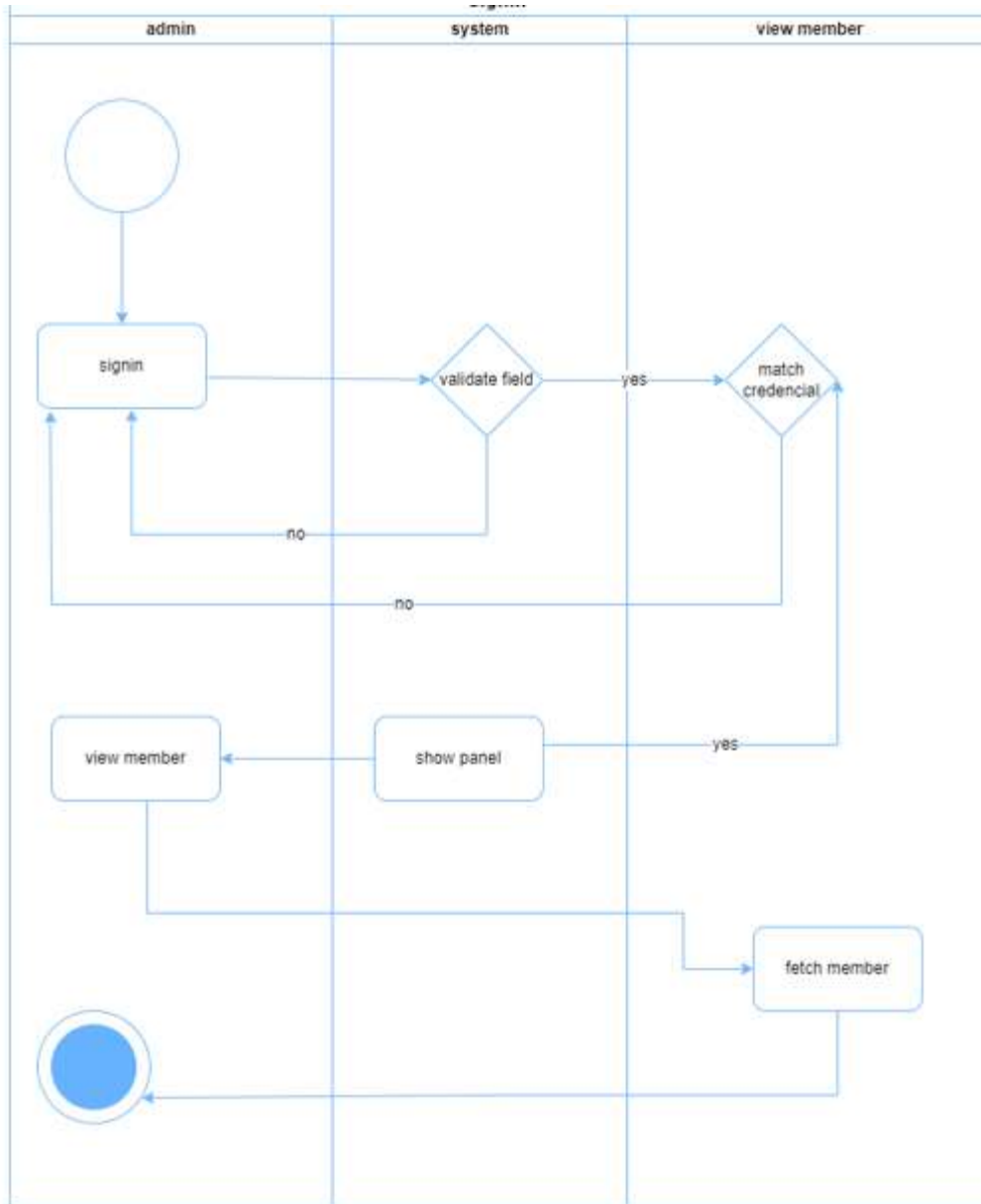
**Fig 4.3: Sign in Activity Diagram**



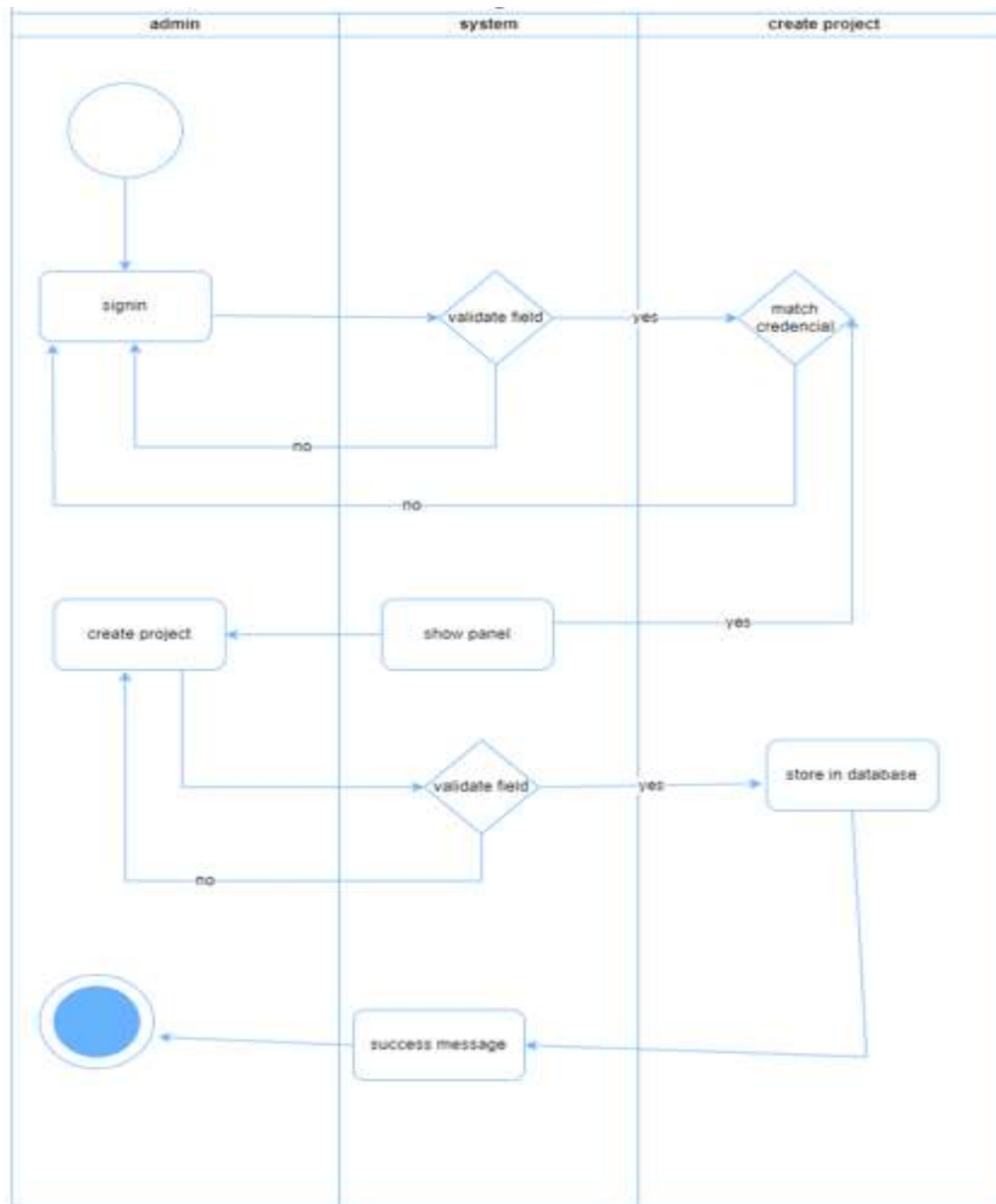
**Fig 4.3: History Activity Diagram**



**Fig 4.4: View Project Activity Diagram**

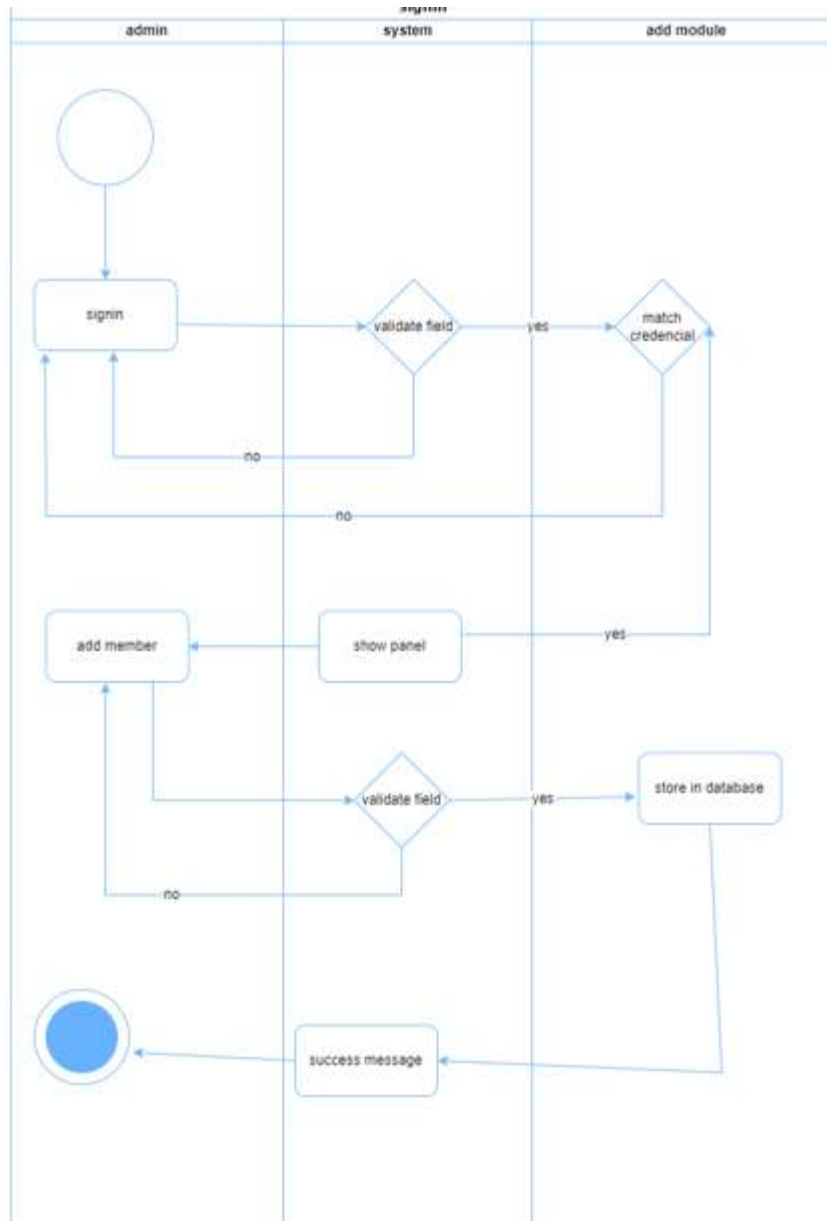


**Fig 4.5: View Member Activity Diagram**

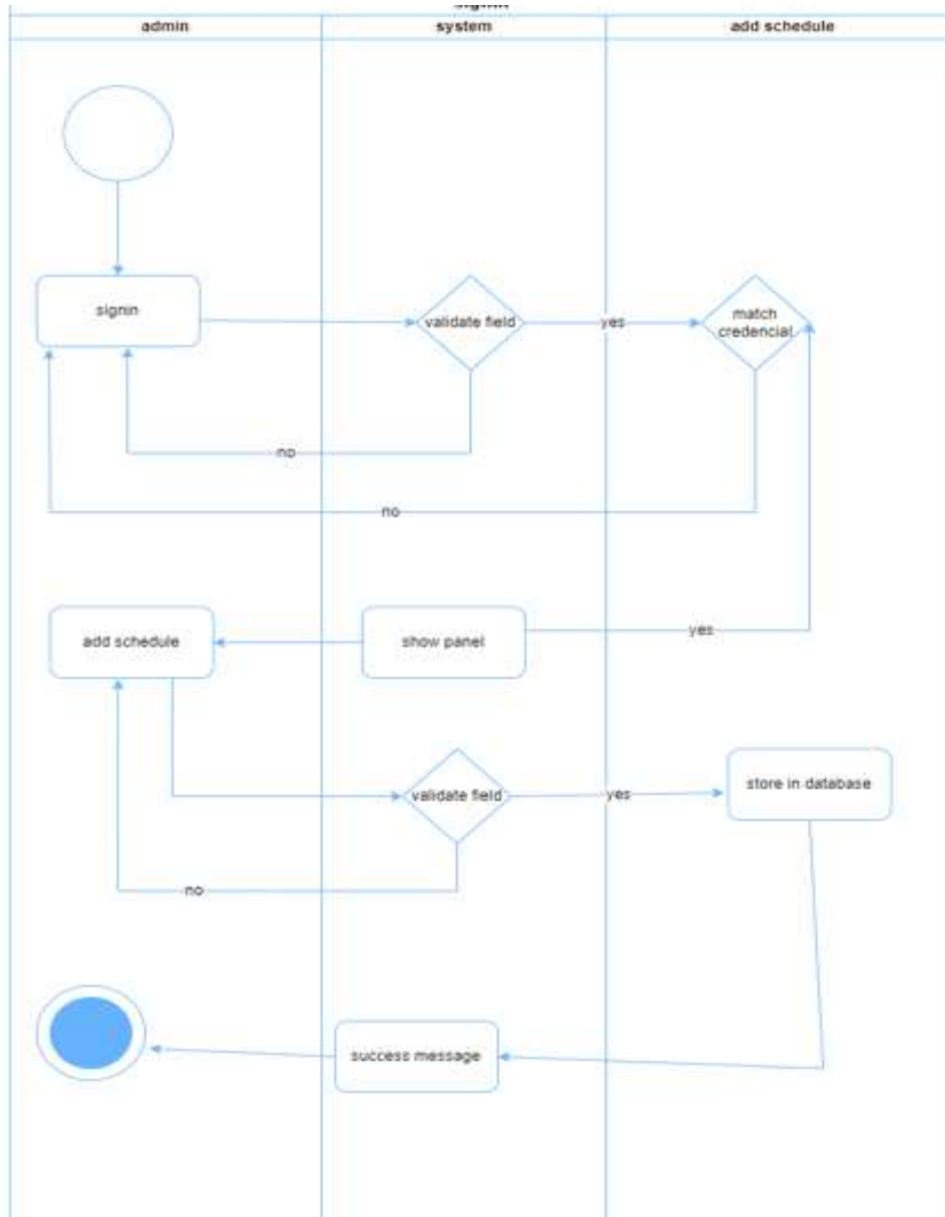


**Fig 4.6: Create Project Activity Diagram**

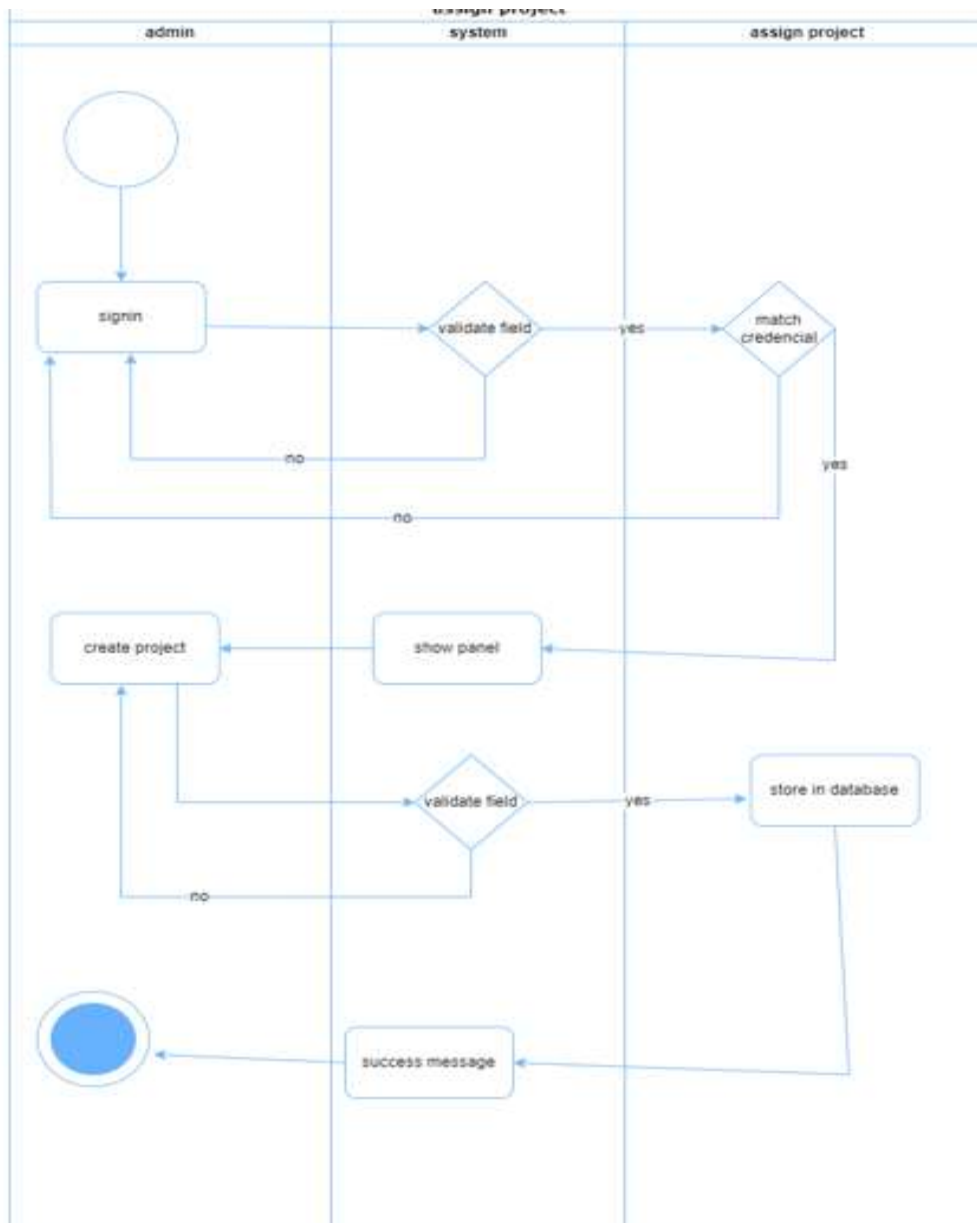




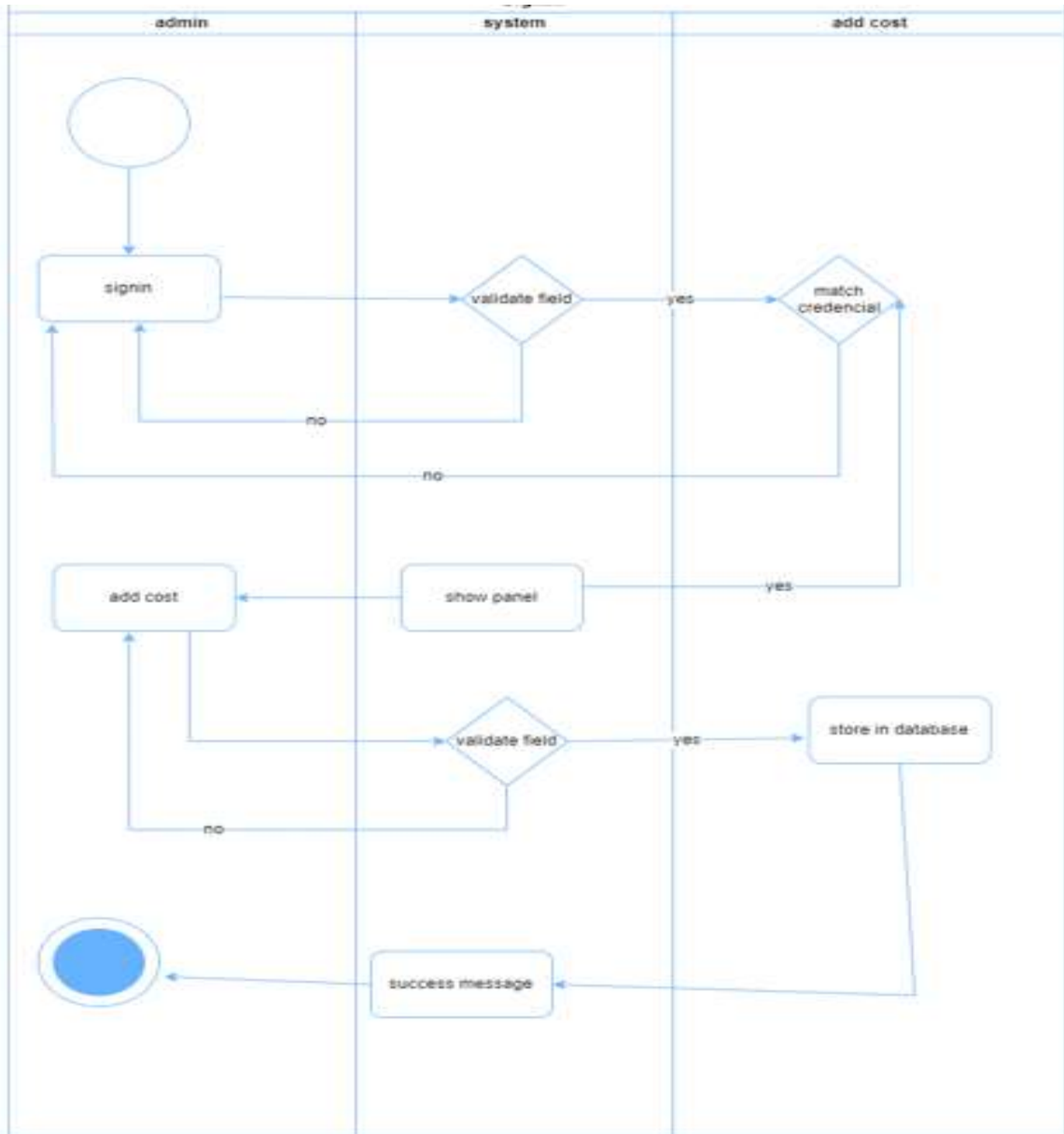
**Fig 4.7: Add Module Activity Diagram**



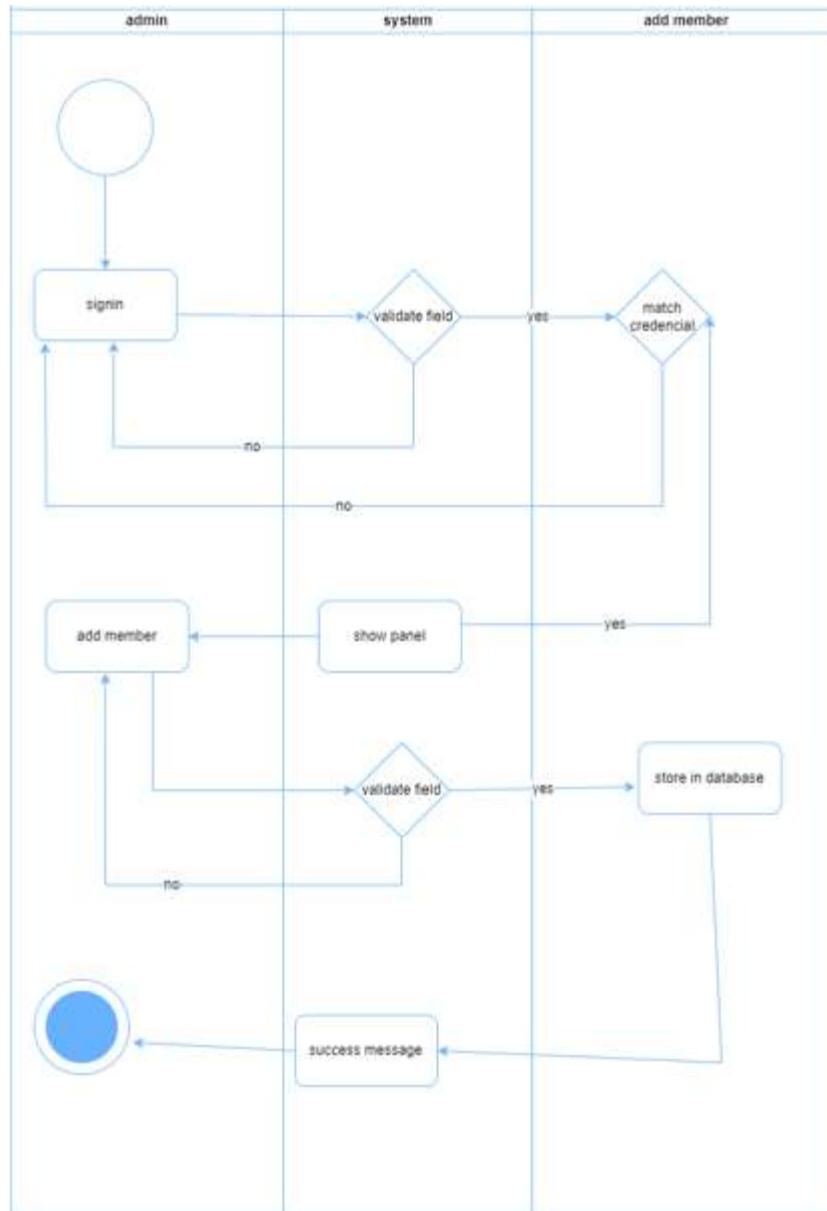
**Fig 4.8: Add Schedule Activity Diagram**



**Fig 4.9: Assign project Activity Diagram**

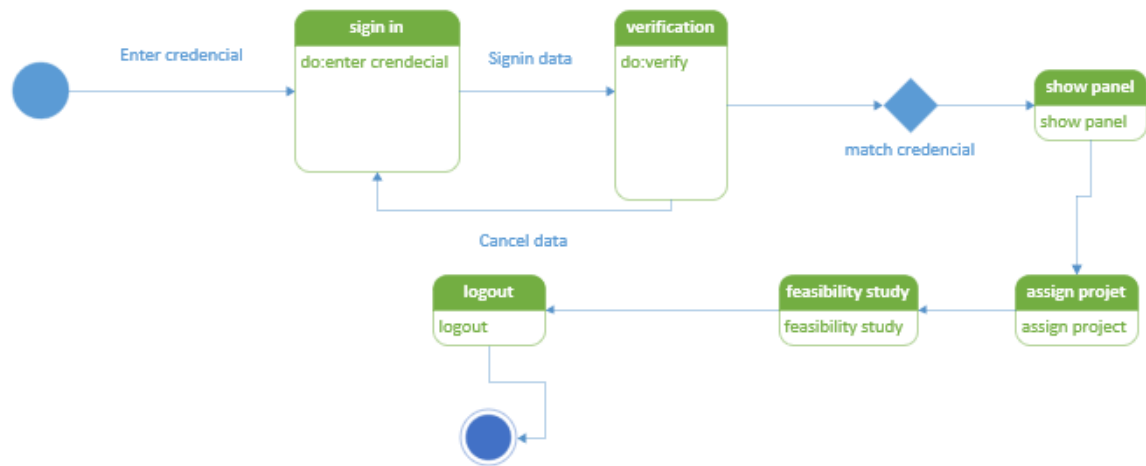


**Fig 4.10: Add Cost Activity Diagram**

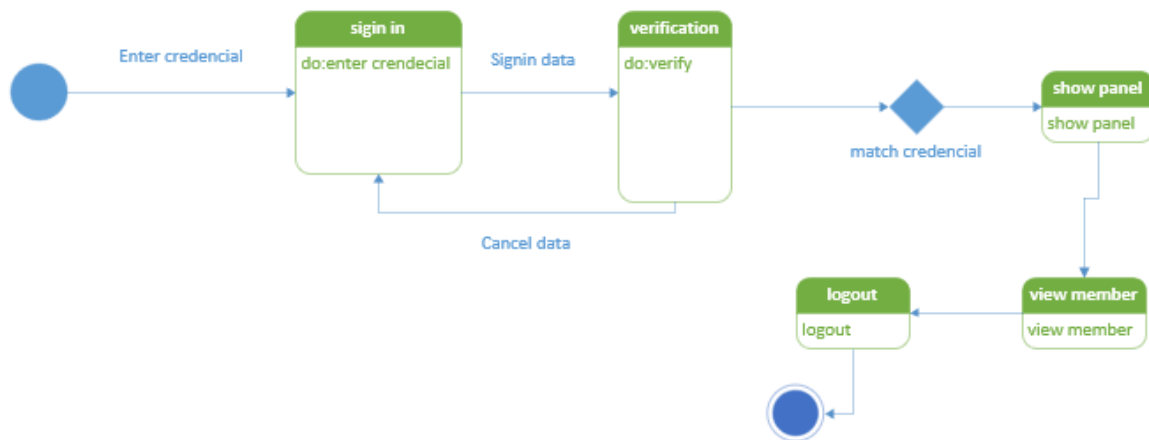


**Fig 4.10: Add Member Activity Diagram**

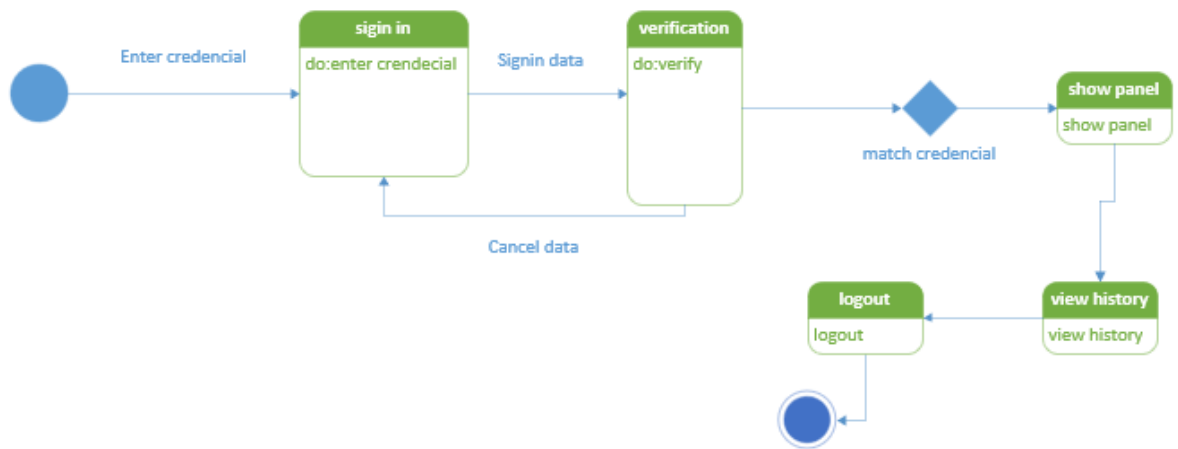
## 4.2.2 State Machine Diagram



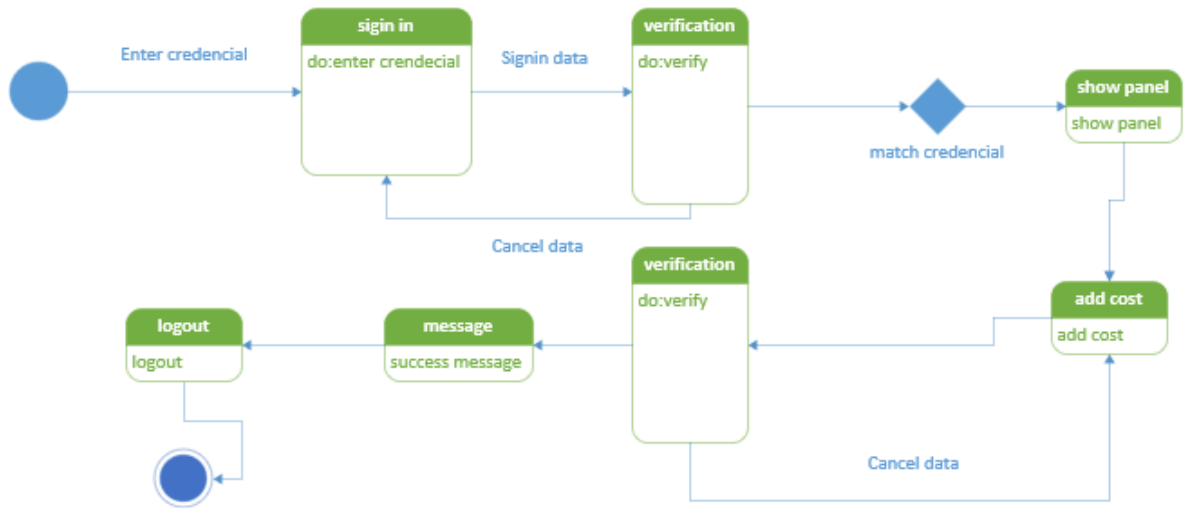
**Sign In**



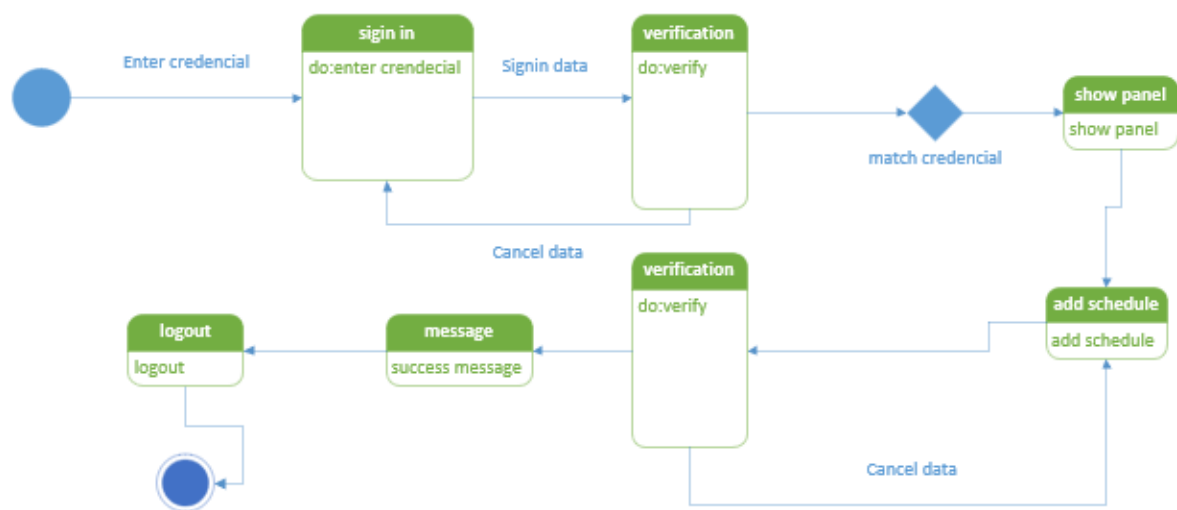
**View Member**



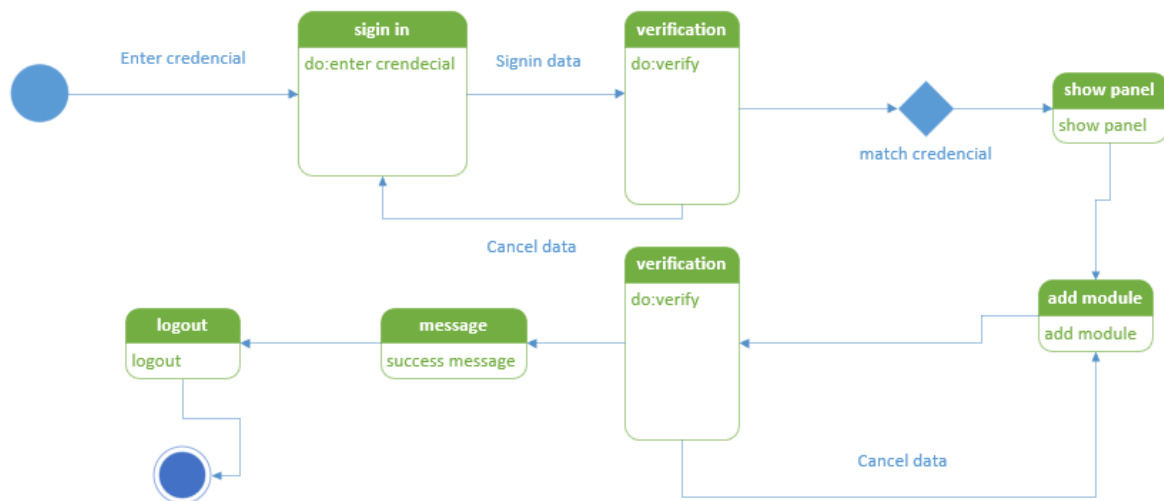
### View history



### Add cost

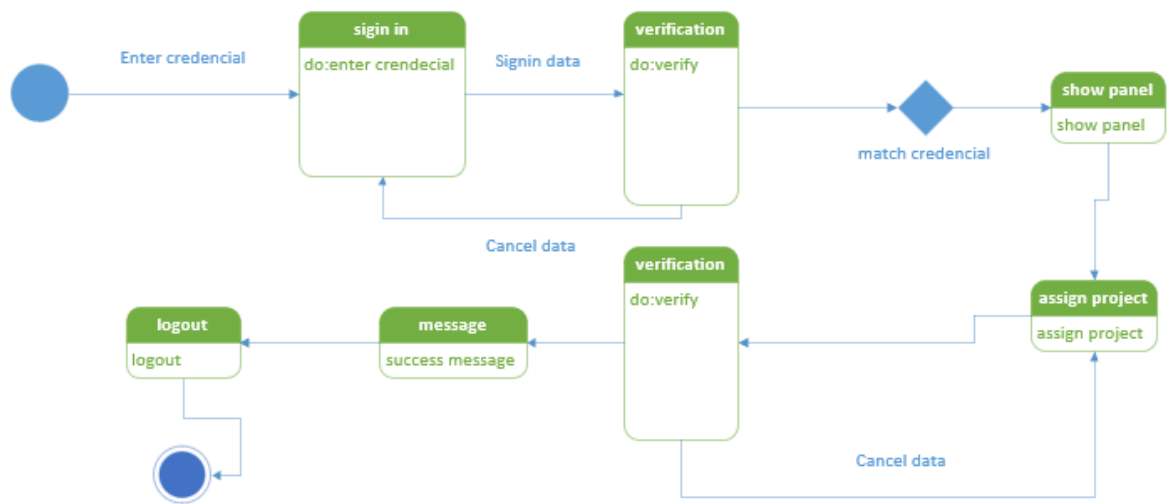


### Add schedule

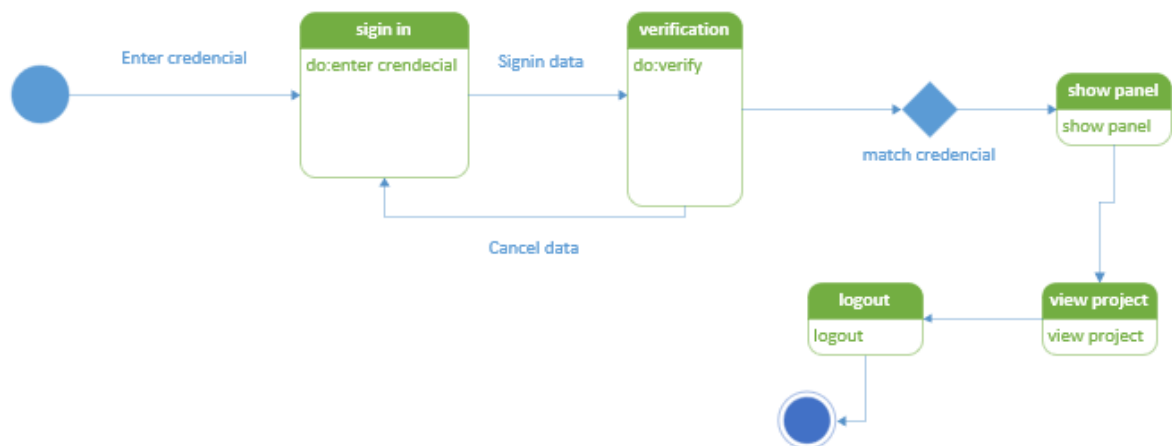


### Add module

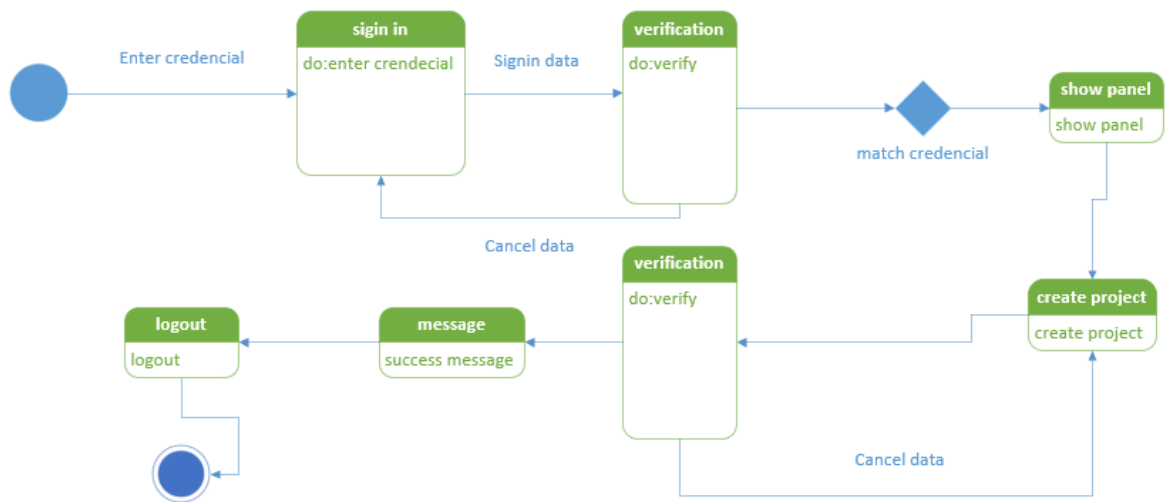




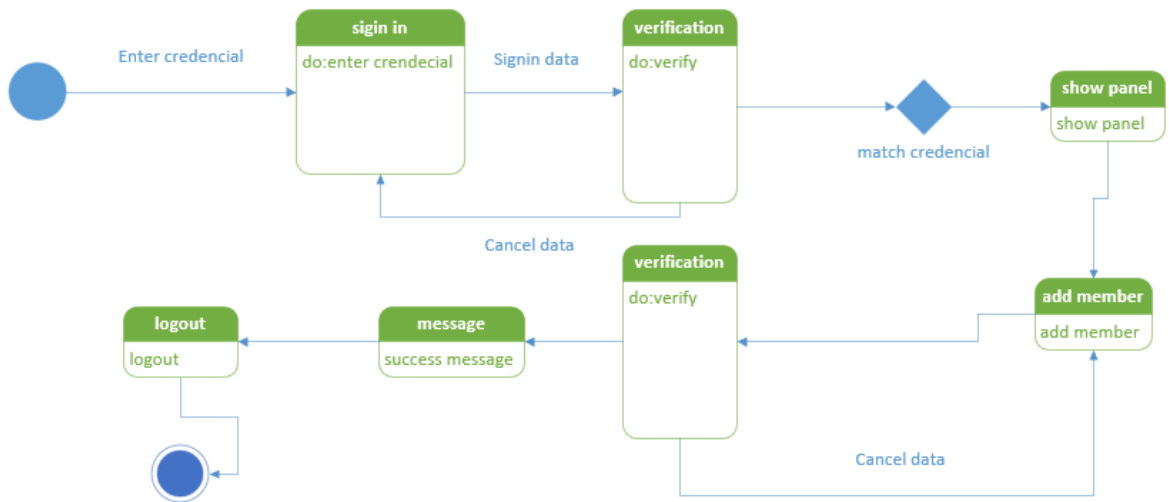
### Assign project



### View project

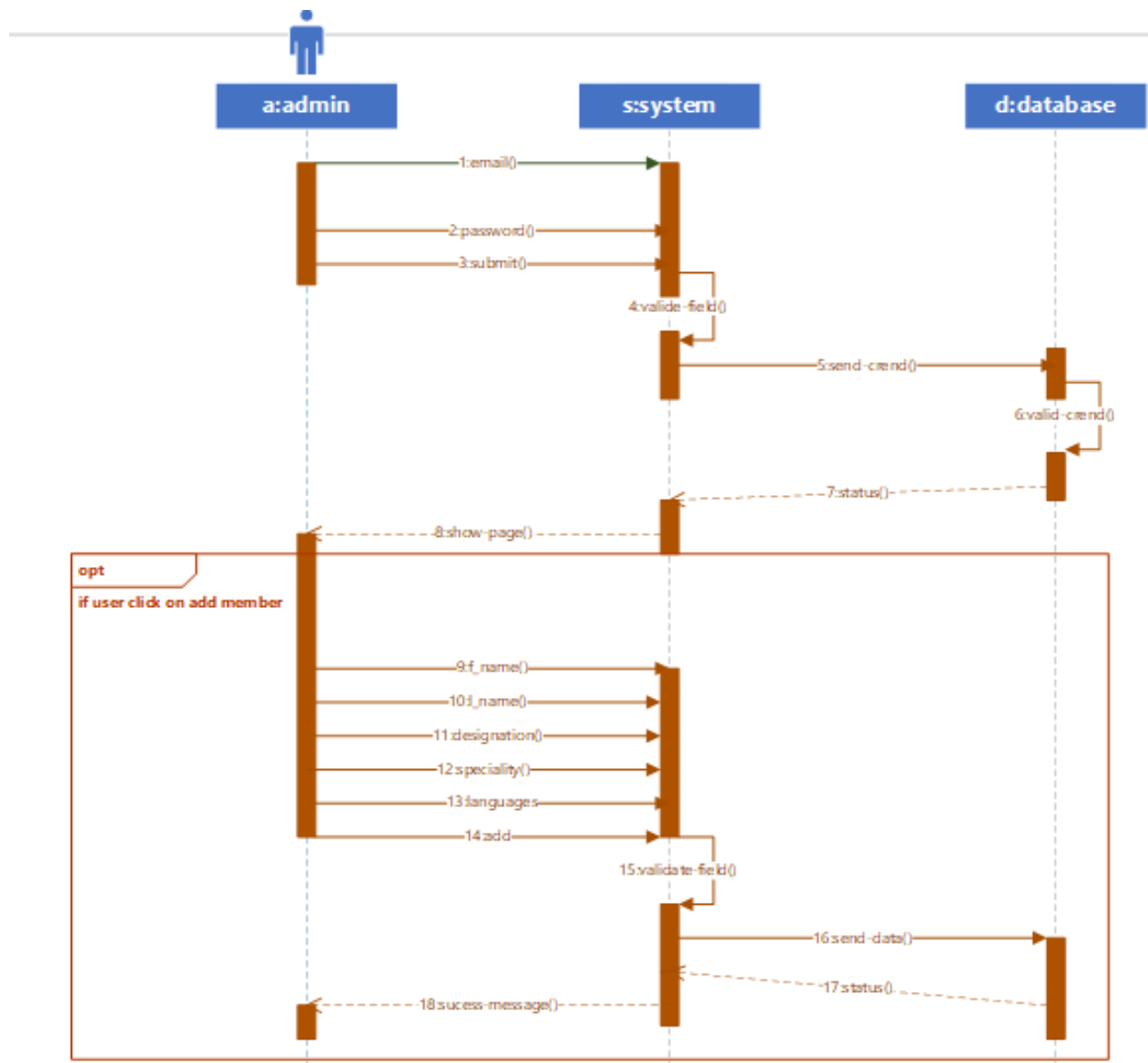


### Create Project

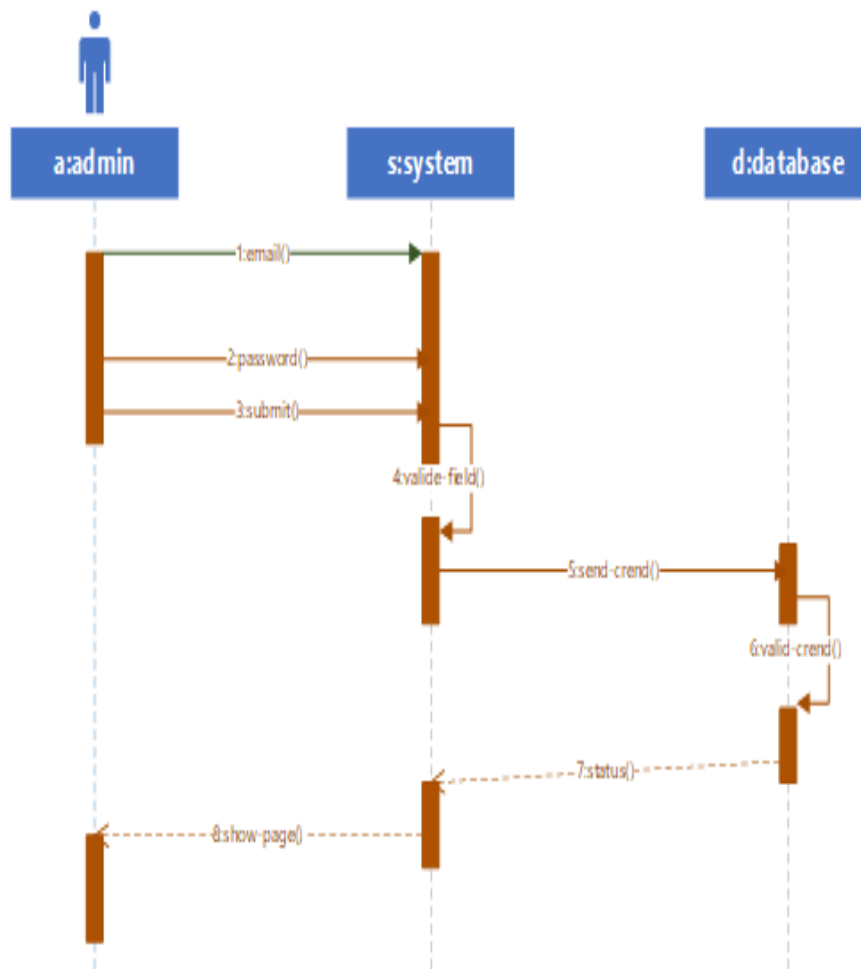


### Add member

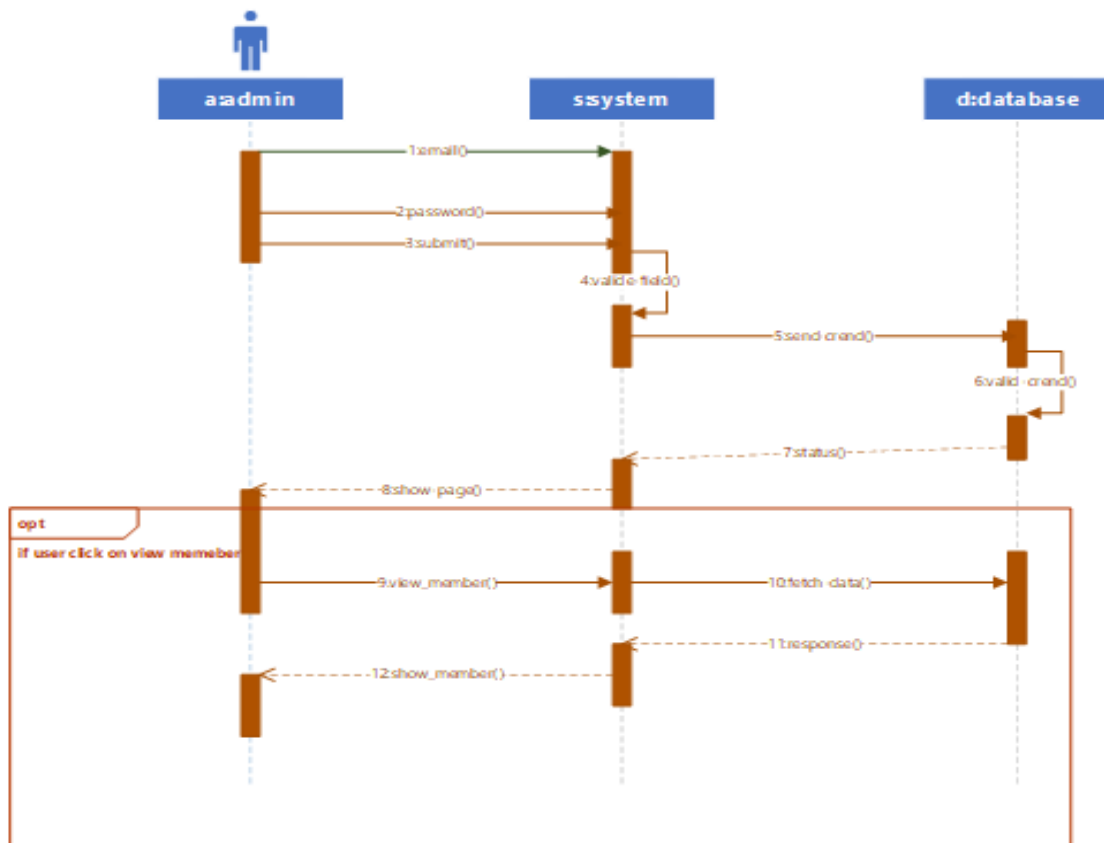
## 4.2.3 Sequence Diagram:



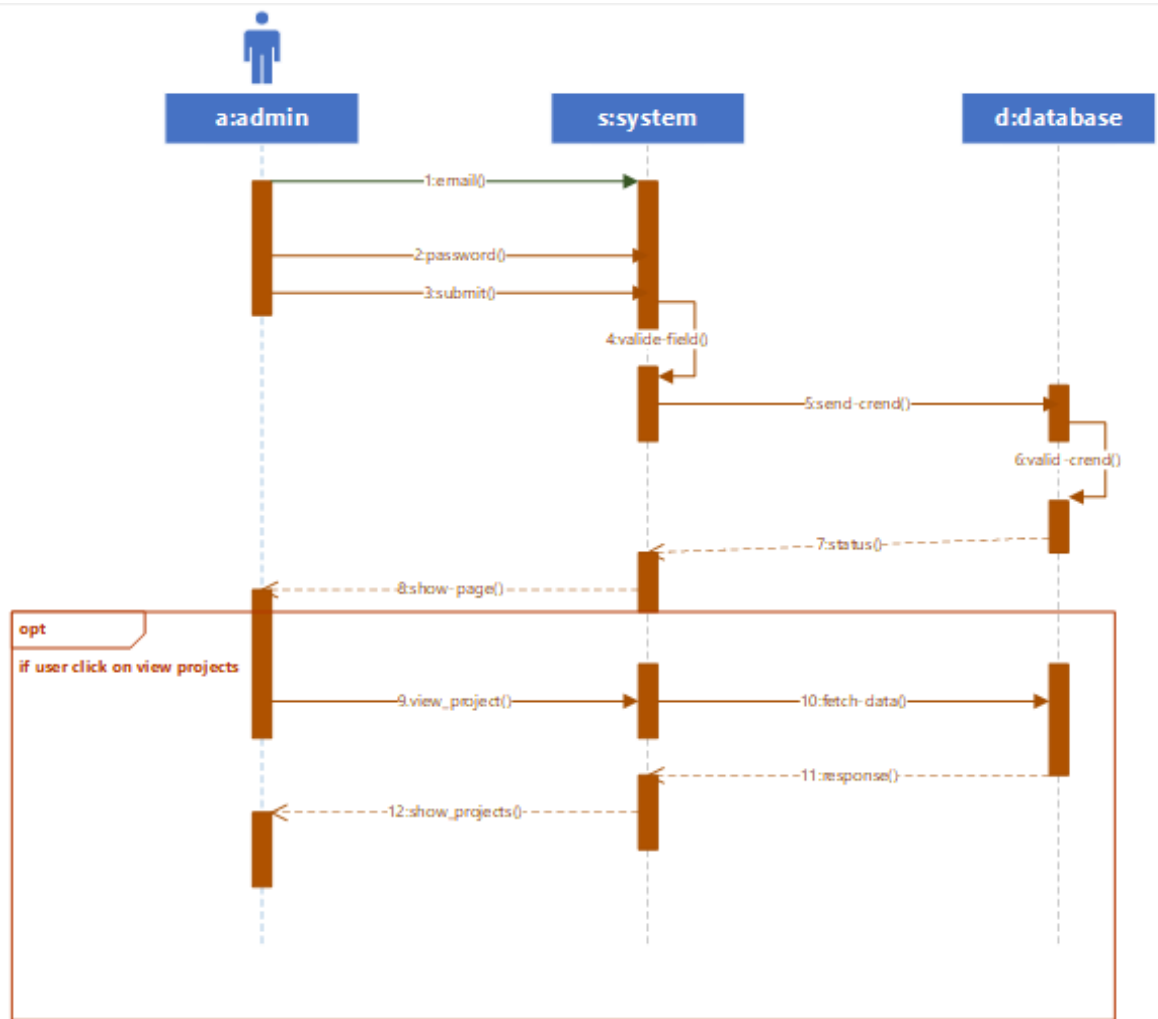
**Fig 5.1: Sequence Diagram Add Member**



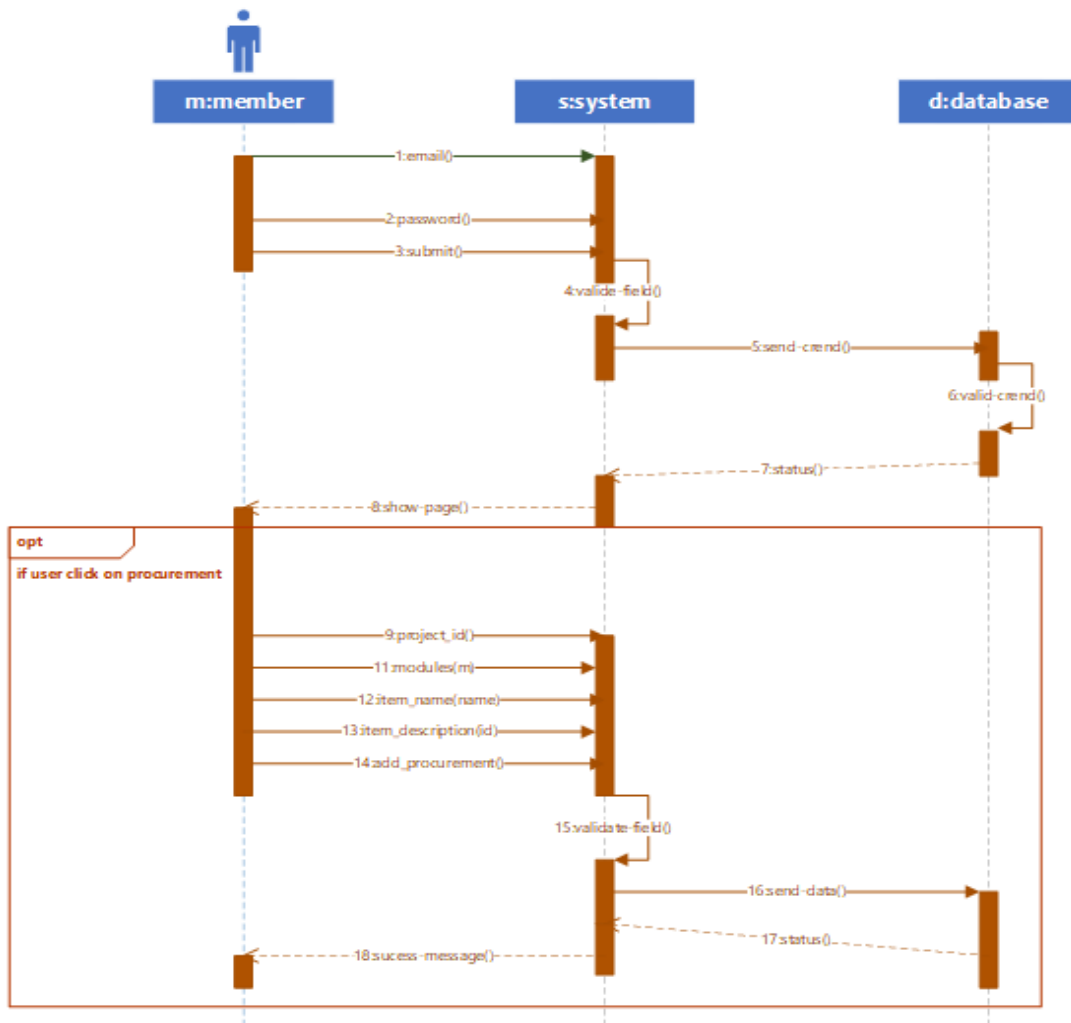
**Fig 5.2: Sequence Diagram sign in**



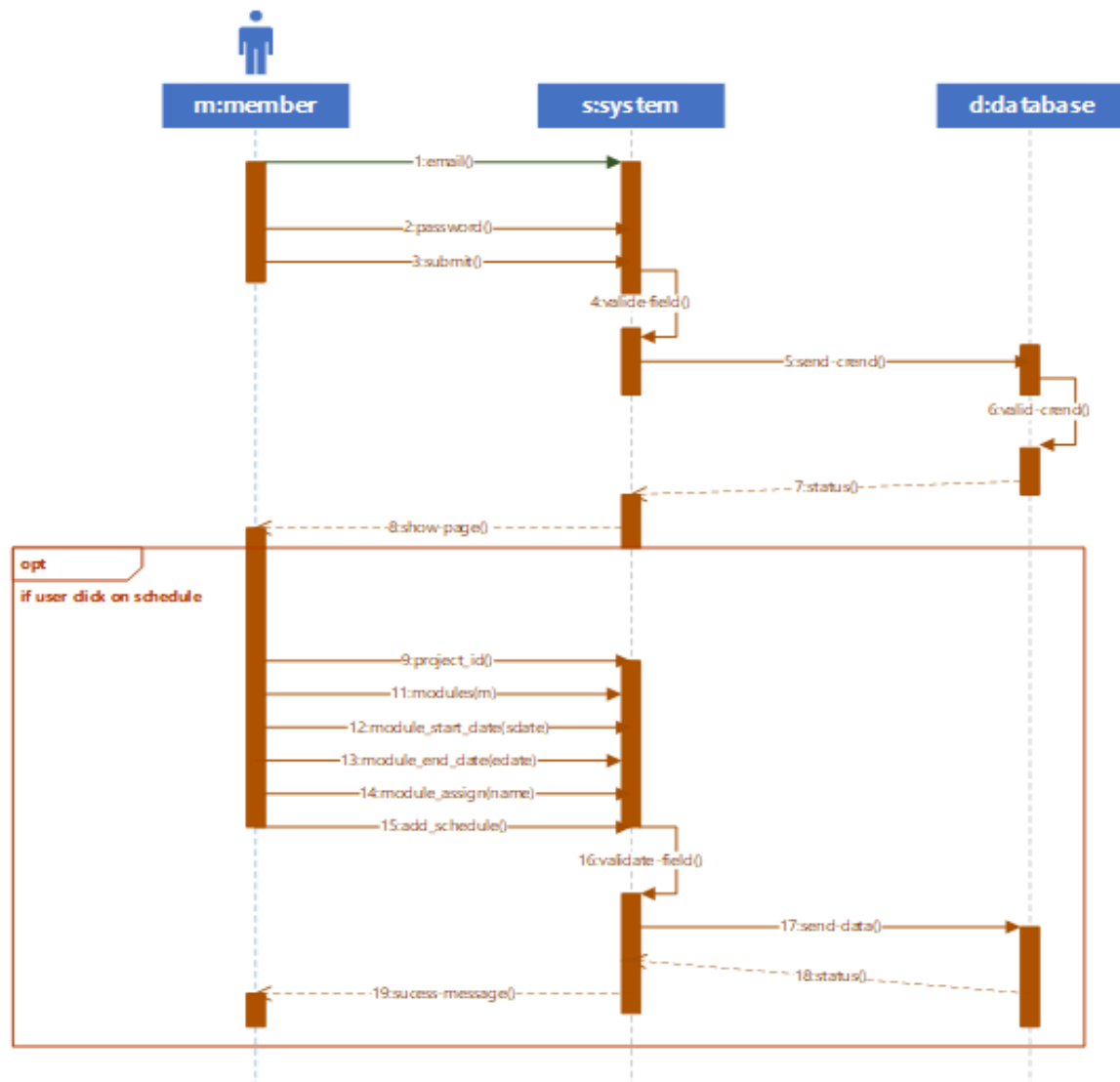
**Fig 5.3: Sequence Diagram View Member**



**Fig 5.4: Sequence Diagram View Project**

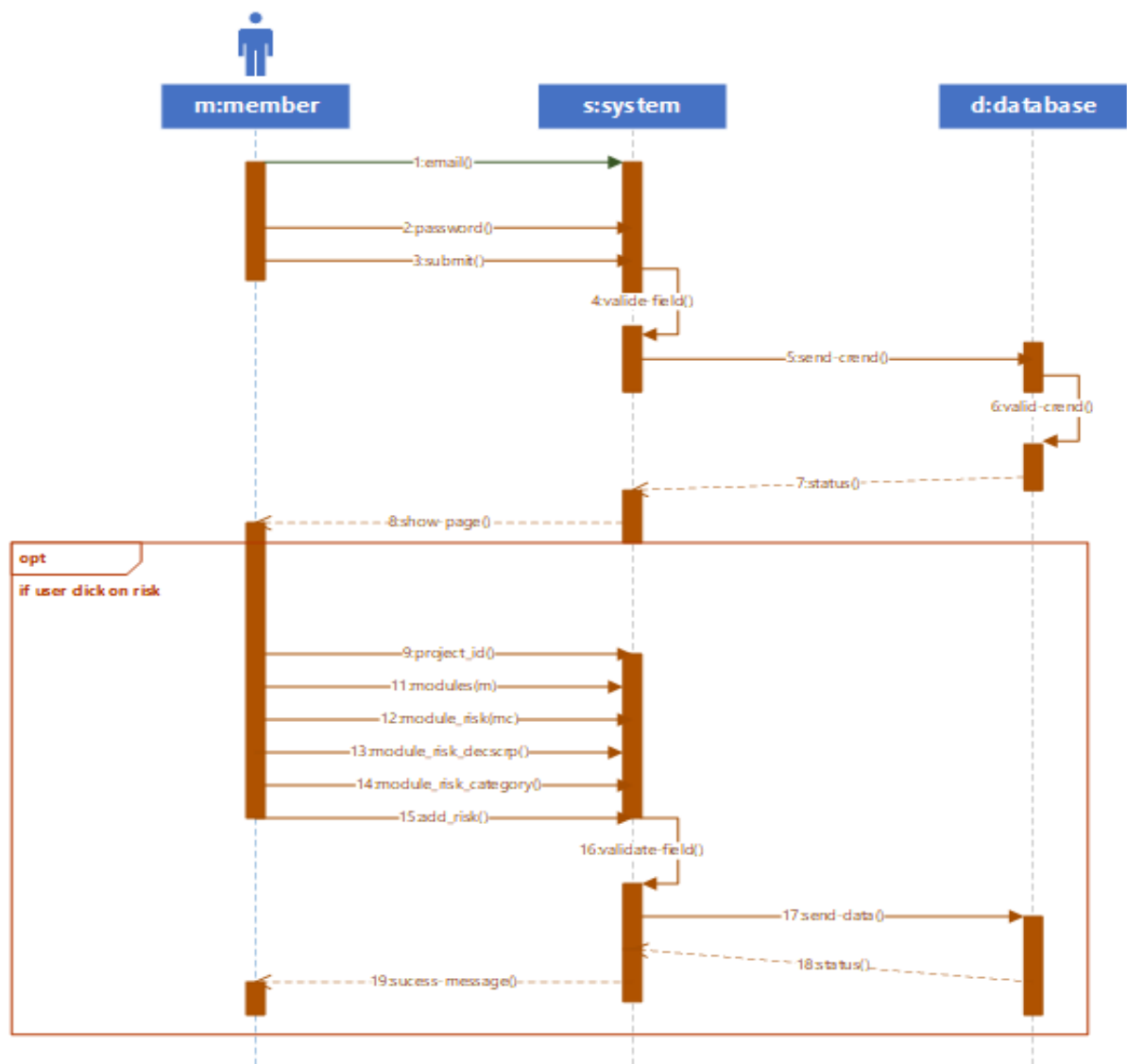


**Fig 5.5: Sequence Diagram Add Procurement**

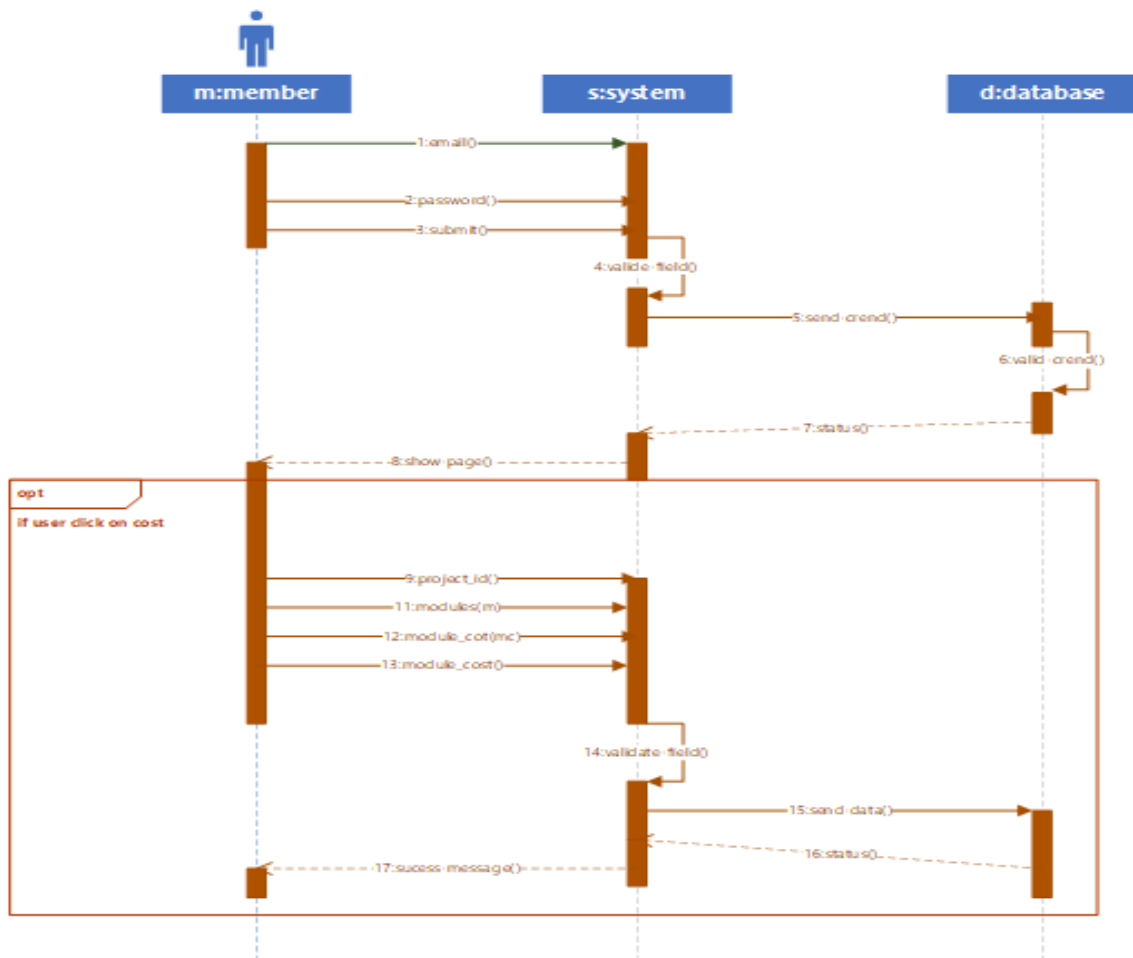


**Fig 5.6: Sequence Diagram Add Schedule**





**Fig 5.7: Sequence Diagram Add Risk**

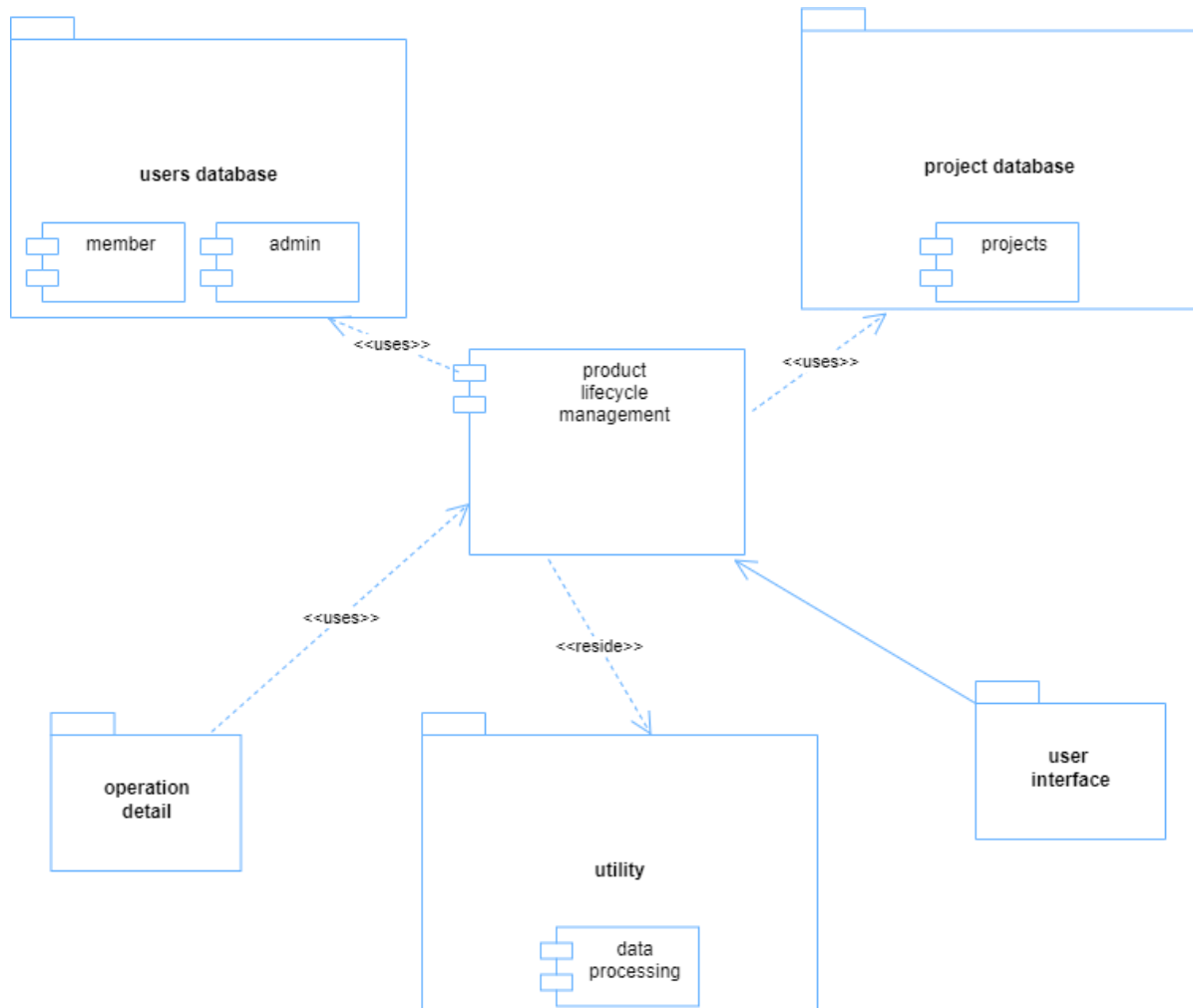


**Fig 5.8: Sequence Diagram Add Cost**

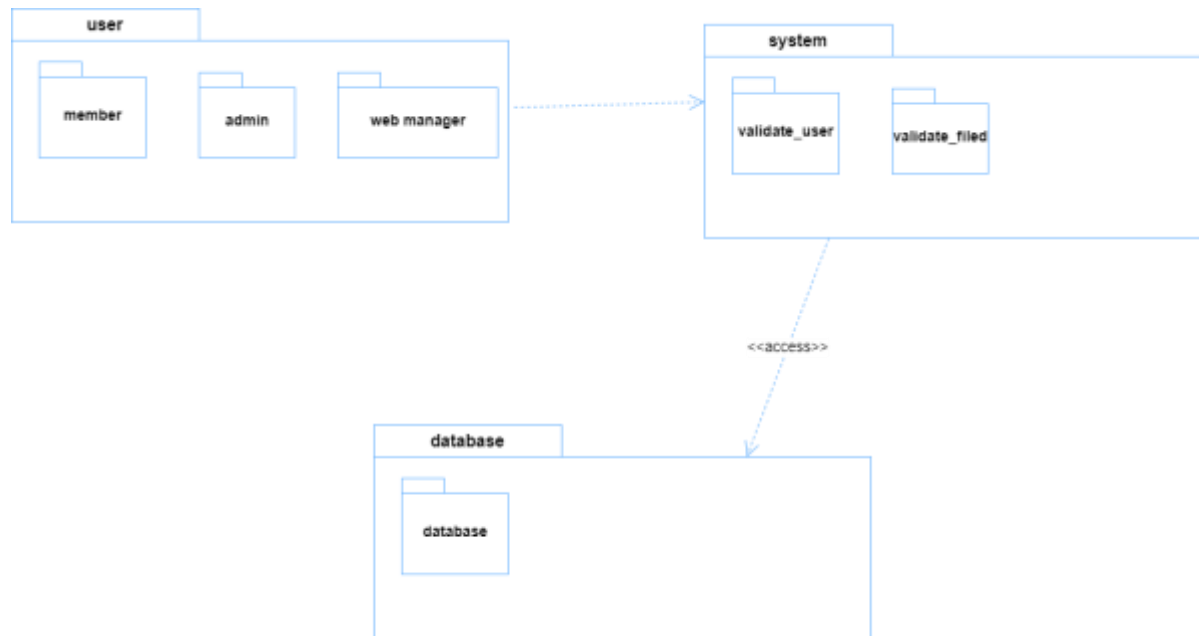
## 4.2.1 UML Structural Diagrams

Following are the UML structural diagrams of our system:

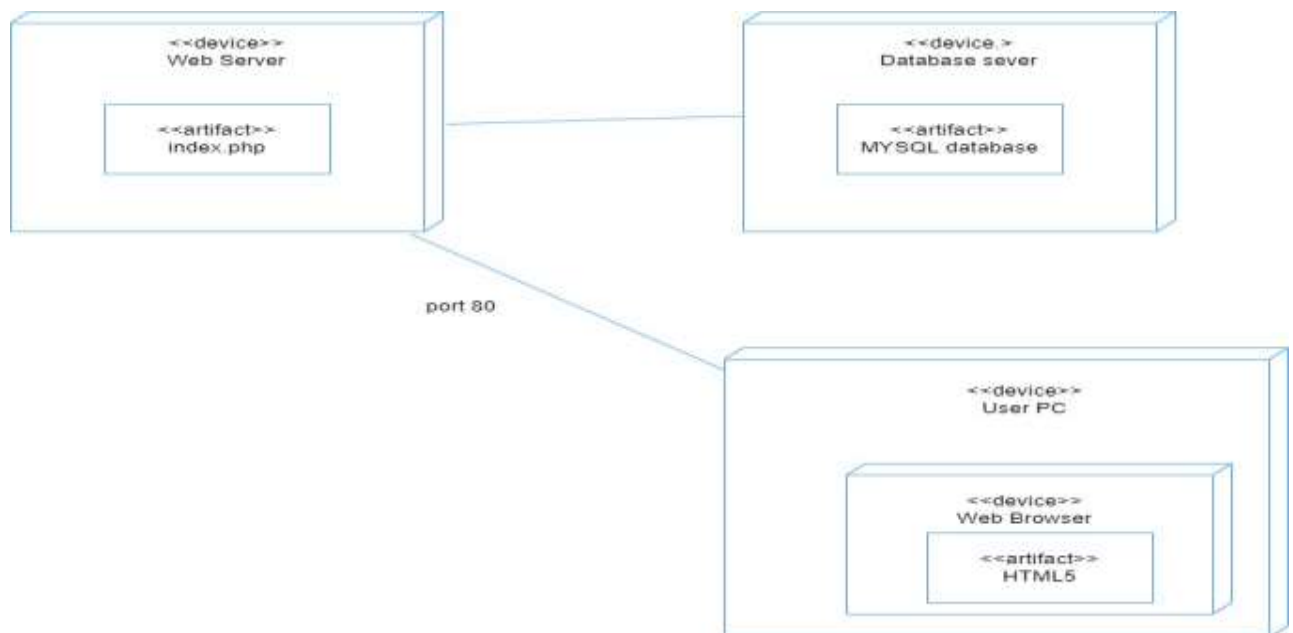
### 4.2.1.1 Component Diagram



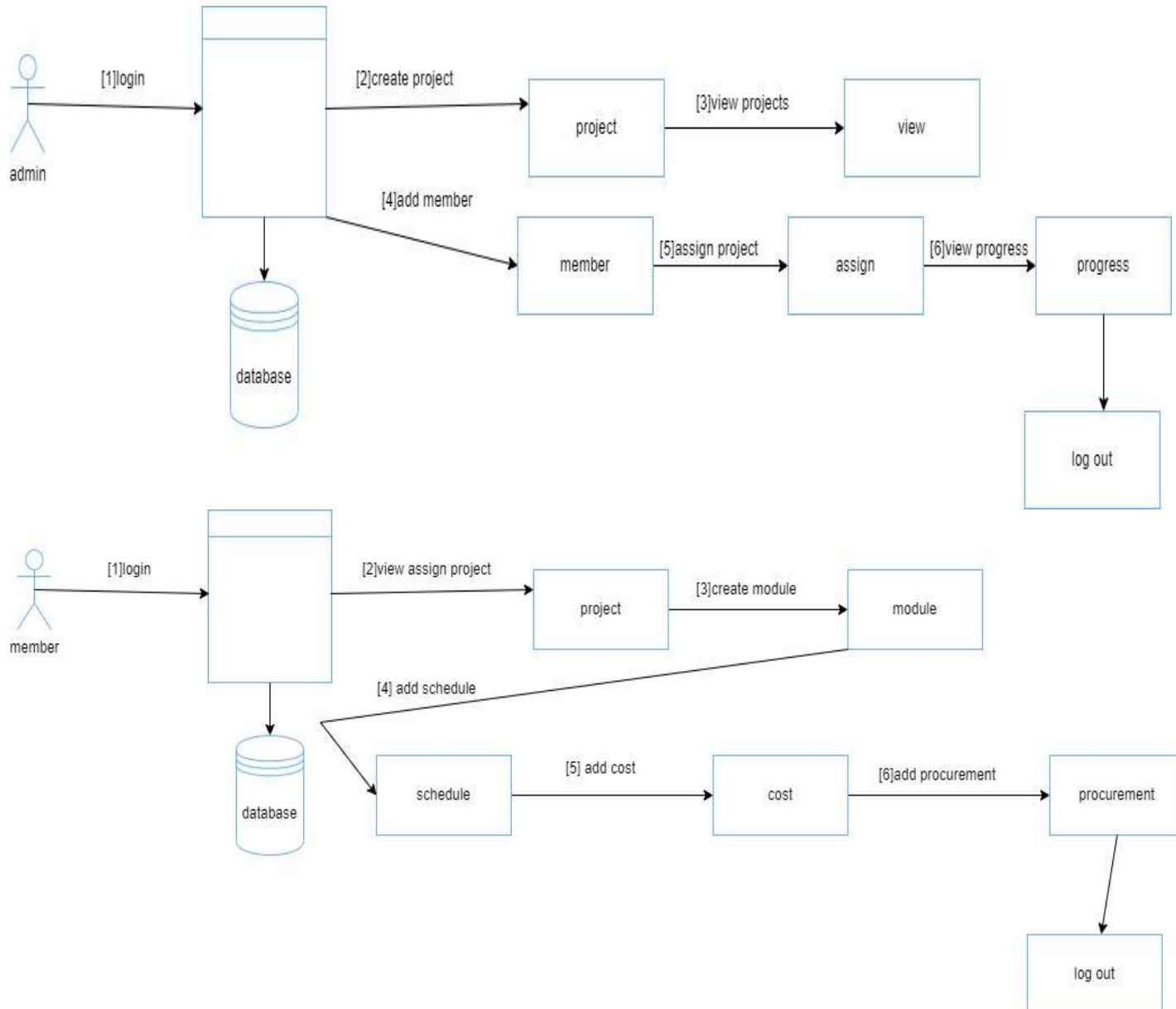
#### 4.2.1.2. Package Diagram



#### 4.2.1.3 Deployment Diagram

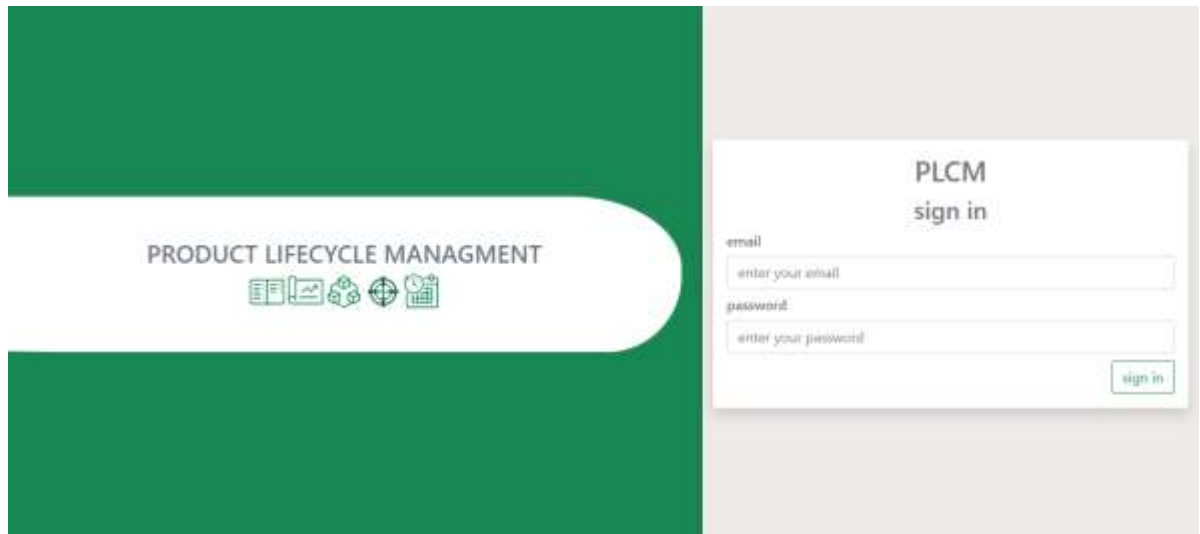


## 4.5 Communication Design Protocol



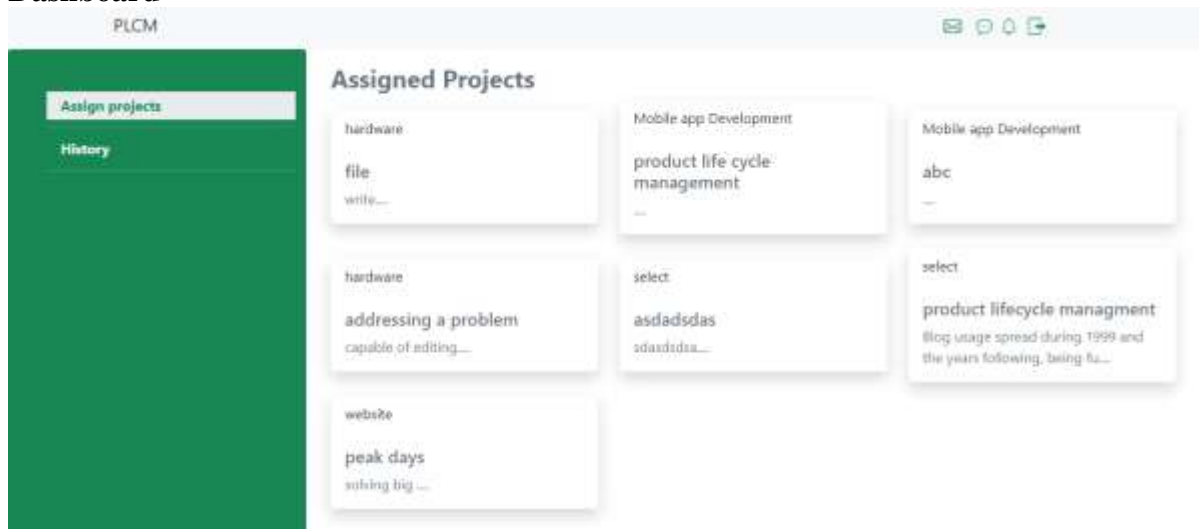
## 4.6 Interfaces of Product Lifecycle Management:

### ➤ Sign in



The sign-in interface features a green sidebar on the left with the text "PRODUCT LIFECYCLE MANAGEMENT" and a row of five icons: a document, a bar chart, a gear, a globe, and a factory. The main area is light gray and contains a white sign-in box. The box is titled "PLCM sign in" and includes two input fields: "email" with the placeholder "enter your email" and "password" with the placeholder "enter your password". A "sign in" button is located at the bottom right of the box.

### ➤ Dashboard



The dashboard interface has a green sidebar on the left with two buttons: "Assign projects" and "History". The main area is light gray and features a header bar with the text "PLCM" and three icons: an envelope, a speech bubble, and a bell. Below the header bar is a section titled "Assigned Projects" which displays a grid of project cards. Each card contains a title, a subtitle, and a description. The cards are arranged in three rows and three columns.

Project Title	Subtitle	Description
hardware	file	write...
Mobile app Development	product life cycle management	...
Mobile app Development	abc	...
hardware	addressing a problem	capable of editing...
select	asdadsdas	sdadsdas...
select	product lifecycle managment	Blog usage spread during 1999 and the years following, being fa...
website	peak days	solving big ...

➤ Project details:

PLCM

Review Proposal

Road Map

Work Package

Assigned to me

Created by me

Add Schedule

Add Cost

### Assigned project detail

file

**Problem Statement**  
write

**Problem Description**  

#### Origins

Before blogging became popular, digital communities took many forms including Usenet, commercial online services such as GEnie, Byte Information Exchange (BIX) and the early CompuServe, e-mail lists<sup>[14]</sup> and Bulletin Board Systems (BBS). In the 1990s, Internet forum software created running conversations with "threads". Threads are topical connections between messages on a virtual "corkboard". From June 14, 1993, Mosaic Communications Corporation maintained their "What's New"<sup>[15]</sup> list of new websites, updated daily and archived monthly. The page was accessible by a special "What's New" button in the Mosaic web browser.

The earliest instance of a commercial blog was on the first business-to-consumer Web site created in 1990 by Ty, Inc., which featured a blog in a section called "Online Diary". The entries were maintained by featured Beanie Babies that were voted for monthly by Web site visitors.<sup>[16]</sup>

The modern blog evolved from the online diary where people would keep a running account of the events in their personal lives. Most such writers called themselves diarists, journalists, or journalets. Justin Hall, who began personal blogging in 1994 while a student at Swarthmore College, is generally recognized as one of the earlier bloggers.<sup>[17]</sup> as is Jerry Pournelle.<sup>[18]</sup> Dave Winer's Scripting News is also credited with being one of the older and longer running weblogs.<sup>[19][20]</sup> The Australian Netguide magazine maintained the Daily Net News<sup>[21]</sup> on their web site from 1996. Daily Net News ran links and daily reviews of new websites, mostly in Australia.

Another early blog was Wearable Wireless Webcam, an online shared diary of a person's personal life combining text, digital video, and digital pictures transmitted live from a wearable computer and EyeTap device to a web site in 1994. This practice of semi-automated blogging with live video together with text was referred to as *streamblogging*, but such streamlogs were not used as frequently as text entries. Such early bloggers such as The Blogfather

➤ Work Packages:

PLCM

Review Proposal

Road Map

Work Package

Assigned to me

Created by me

Add Schedule

### Work Packages

Project Name:file

subject

type

status

Assigned

Priority

subject

Select

Select

Select

Select

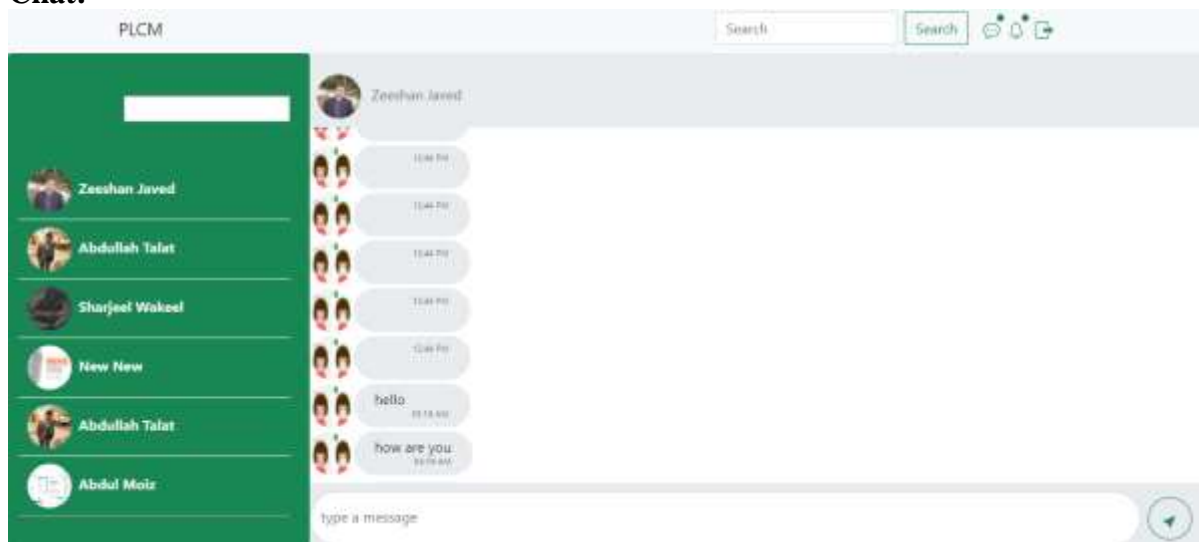
upload file

create

Search:

ID	Subject	Type	Status	Assigned	Priority
51	include	Task	Schedule	zeeshan	Low
50	no way	Task	Schedule	zeeshan	Normal
49	testing again	Milestone	To be schedule	zeeshan	Low
48	from me	Task	Schedule	zeeshan	Normal

➤ Chat:



➤ Emails:





## ➤ Add Schedule

PLCM

Search

Search

Task name Start time Duration

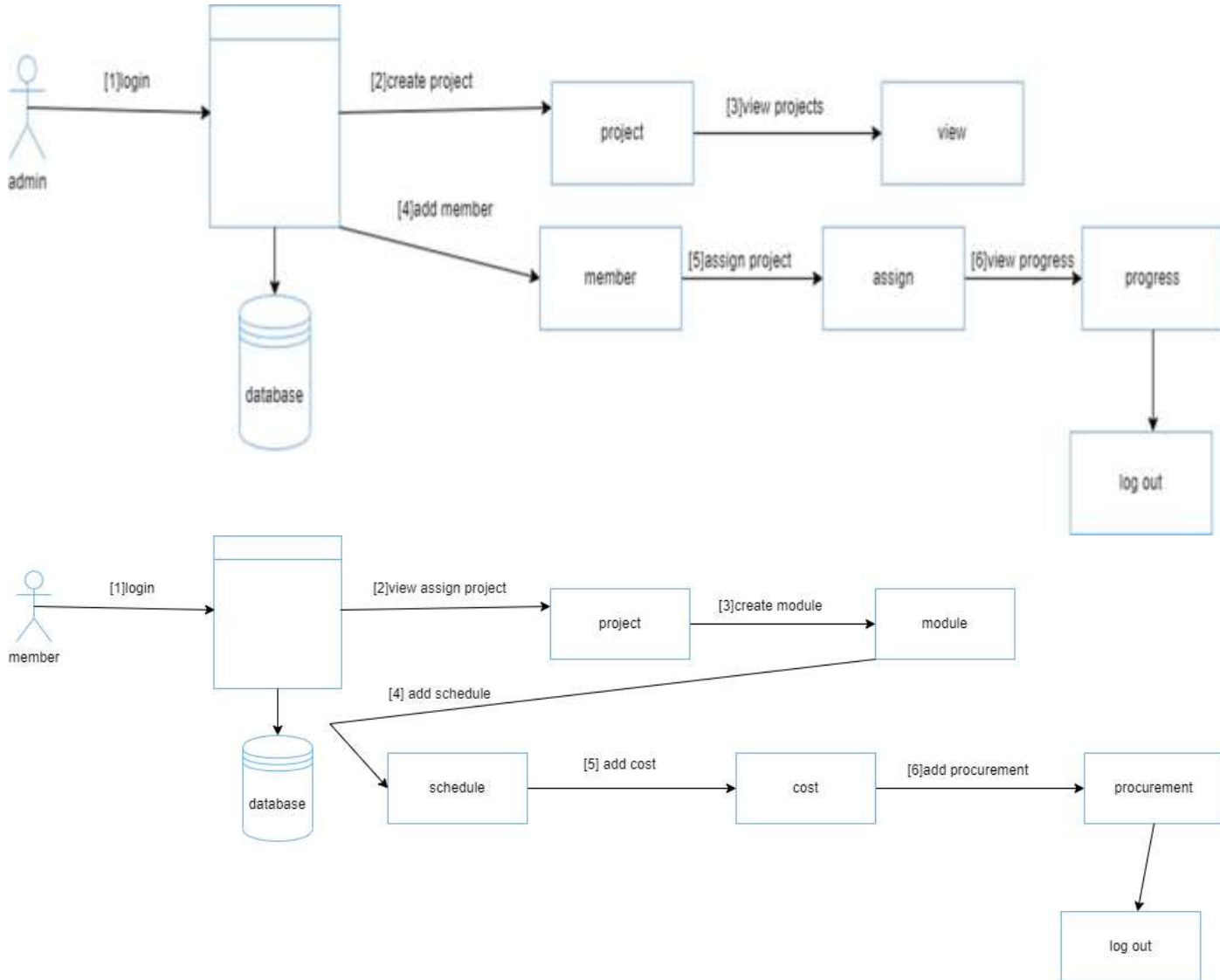
Task name	Start time	Duration
rutly	2022-03-06	303
teeling	2022-03-21	402
rene	2022-04-25	010
06 March 2022 - 07 March 2022	New task	
Description		
New task		
Time period		
0 March 2022 - 07 March 2022		
Save	Cancel	Delete
ken	2022-03-06	0
placeta	2022-04-04	676
teeling again	2022-04-24	07
include	2022-04-26	127
New task	2022-03-06	0

New task

## Chapter 5: Implementation

This chapter will discuss implementation details supported by UML diagrams (if applicable). You will not put your source code here. Any of the following sections may be included based on your project.

### 5.1. Communication Protocol Implementation

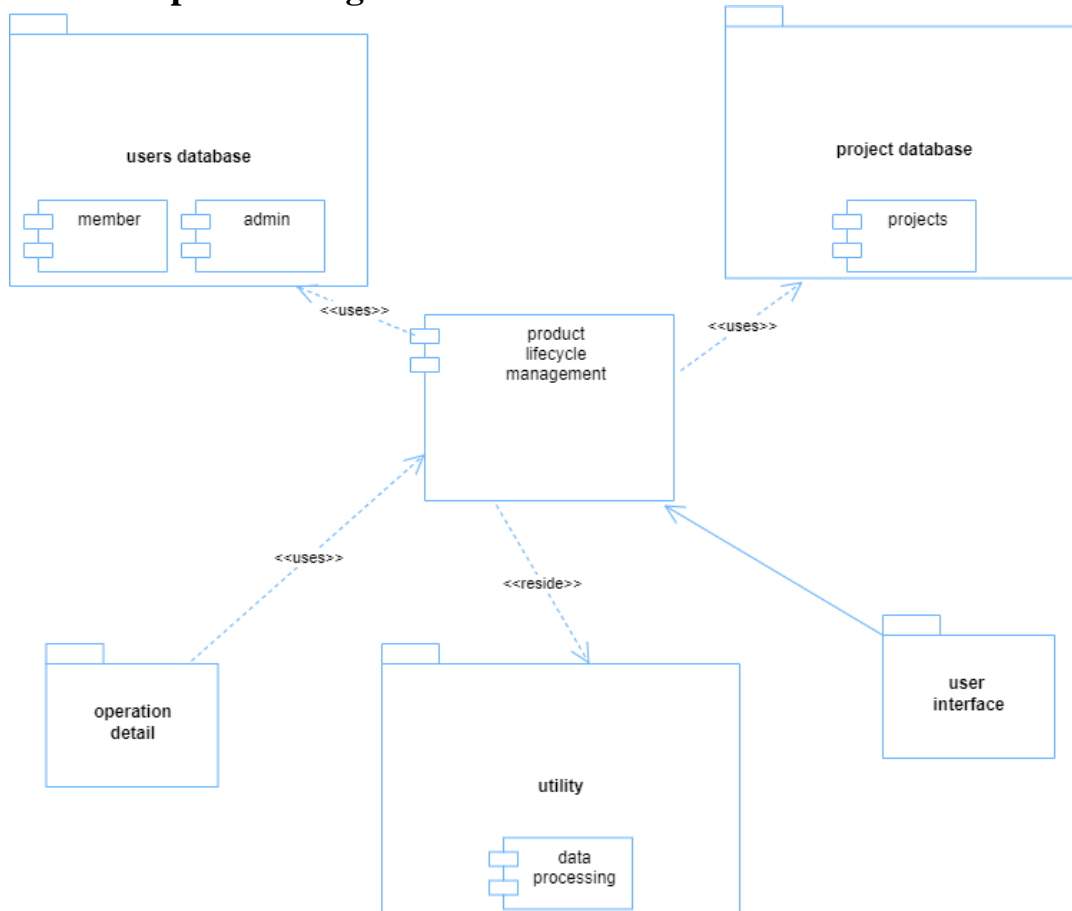


#### Explanation:

This project product lifecycle management is managed whole project life cycle in which the admin creates a project and assigns the project to the member. Admin can communicate with members and receive alerts when members do something on assigned projects. Members are able to add modules, schedules,

costs, and different functions to the project and report to the admin through communication and mail.

## 5.2. Component Diagram

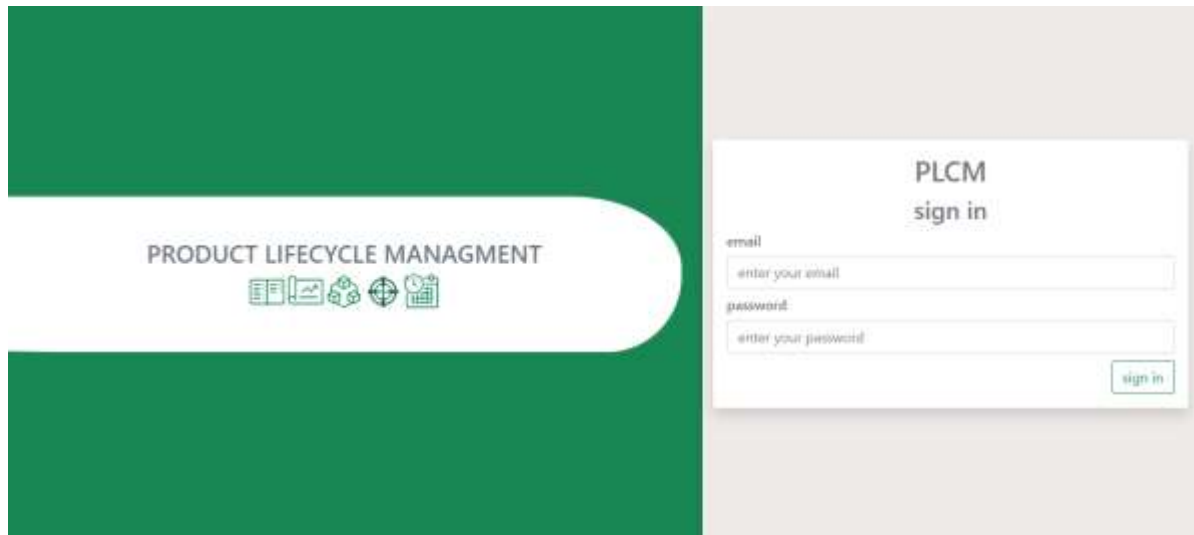


### Explanation:

In the Component diagram, it has been shown that the admin and member interact with the system and perform functions on the project. A system able to hold projects and provide interface according to the user category and authorized functionality according to the user.

## 5.1. User Interface

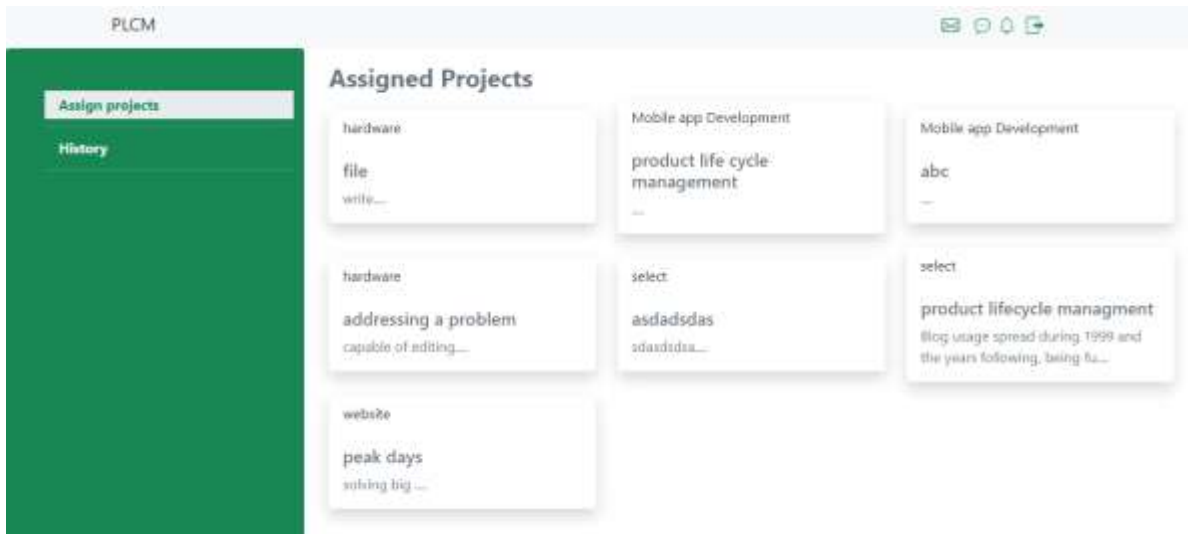
### Sign up



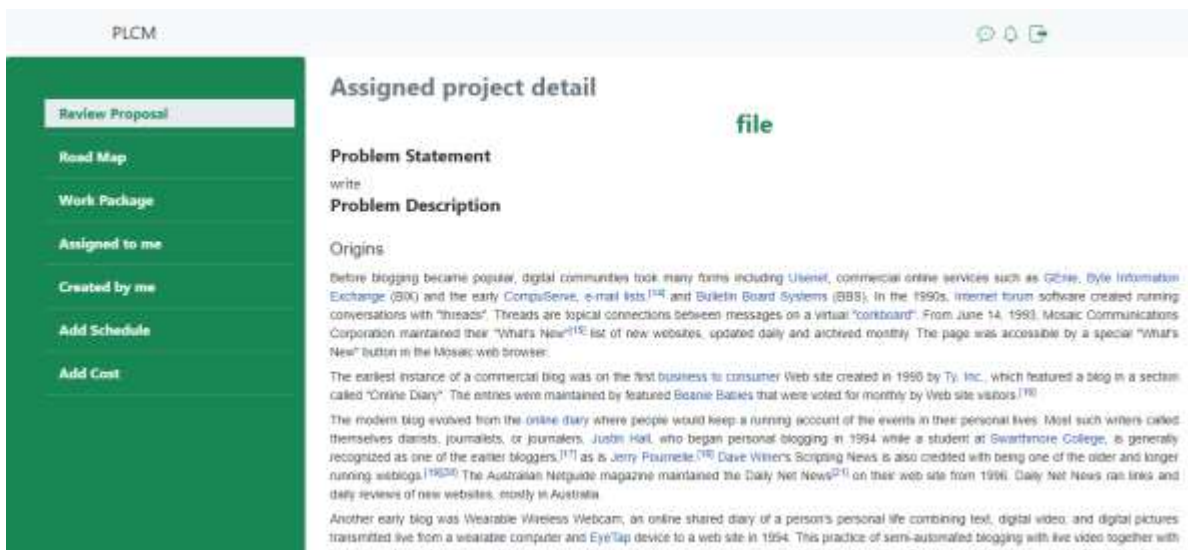
The image shows a user interface mockup for a system titled "PRODUCT LIFECYCLE MANAGEMENT". The interface is split into two main sections. On the left, a green background features a white rounded rectangle containing the text "PRODUCT LIFECYCLE MANAGEMENT" and a row of five icons: a document, a bar chart, a gear, a globe, and a calendar. On the right, a light gray background contains a white sign-in form. The form has the title "PLCM" and the subtitle "sign in". It includes two input fields: "email" with the placeholder text "enter your email" and "password" with the placeholder text "enter your password". A "sign in" button is located at the bottom right of the form.

The user will provide their credentials and click on sign-in button then the system validates the field and after validation credentials validate through the database.

- 1.user enter email and password.
- 2.system validates the email and password.
- 3.system match email and password through database.
- 4.system redirect to home page.



After signing user redirect to the home page where he sees his assigned project and history if he clicks on an assigned project he gets all detail of the project and works on the project.



After clicking on an assigned project. The user redirected to the work packages where he performed the function on a project.

PLCM

Review Proposal

Road Map

Work Package

Assigned to me

Created by me

Add Schedule

Work Packages

Project Name:file

subject

type

status

Assigned

Priority

subject

Select

Select

Select

Select

upload file

create

Search:

ID	Subject	Type	Status	Assigned	Priority
51	include	Task	Schedule	zeeshan	Low
50	no way	Task	Schedule	zeeshan	Normal
49	testing again	Milestone	To be schedule	zeeshan	Low
48	from me	Task	Schedule	zeeshan	Normal

In this section, you can add a module on which you want to work and it also show you a previous module and its current situation.

PLCM

Zeeshan Javed

11:46 PM

11:46 PM

11:46 PM

11:46 PM

11:46 PM

hello

how are you?

type a message

Zeeshan Javed

Abdullah Talat

Sharjeel Wakeel

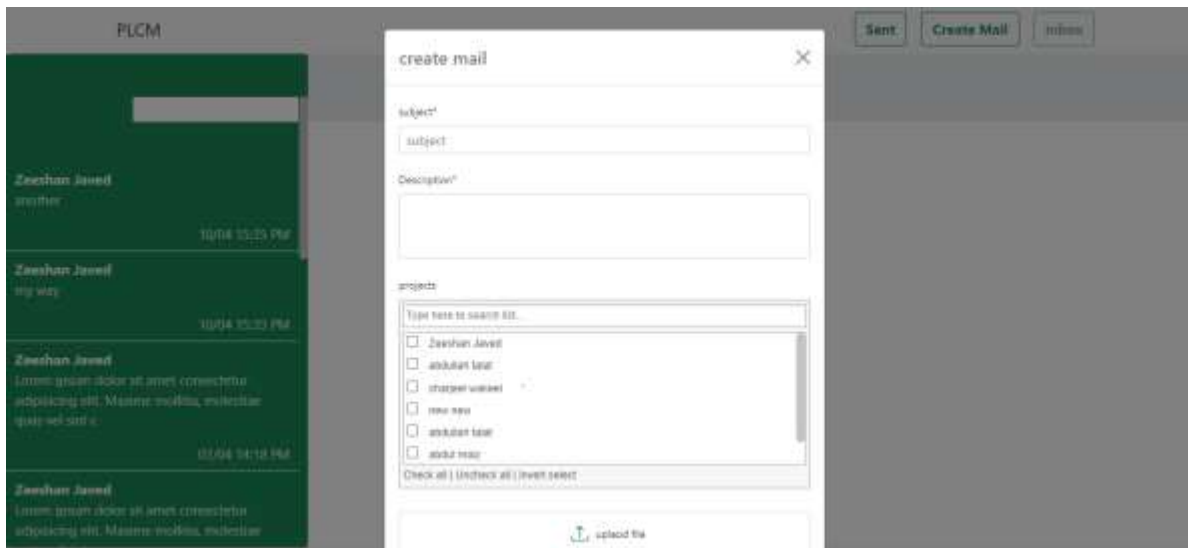
New New

Abdullah Talat

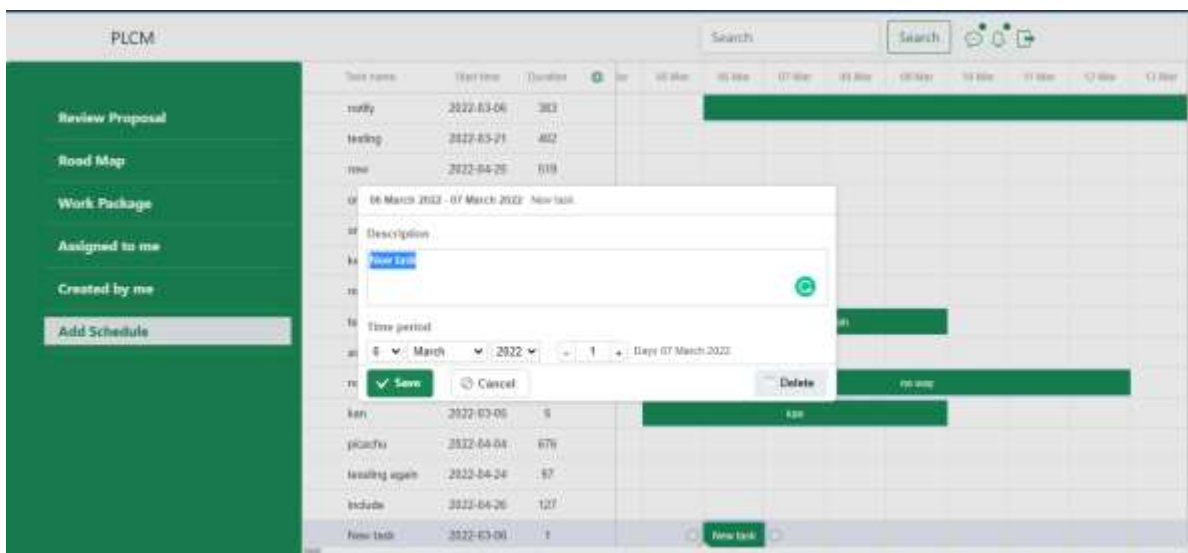
Abdul Moiz

Teams members and admin can communicate with each other and discuss the project problem and next scheduling.

60



Team members and users can email each other and exchange data. they have inbox where they receive incoming mail.



if you inserted a module in work packages you will be able to schedule it if the module deleted then schedule also deleted.

## Chapter 6: Testing and Evaluation

### 6.1. Verification:

Verification is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have.

### 6.1.1. Functional Testing:

Functional Testing is a type of black box testing whereby each part of the system is tested against functional specification/requirements.

- After giving right credentials we are able to login in our system.
- Our system gives error message when user enter incorrect card number.
- Our system successfully shows the record in database of registered users.

#### 1. Login

Tested By	Abdullah Talat
Test type	Unit testing
Test case number	01
Test case name	Sign in
Test case description	This test case will check login info
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User will enter email and password
2	On clicking sign in button user will be signed in
3	If not able to sign in, then first have to create account

**Sign in page**

#### 2.Create project

Tested By	Abdullah Talat
Test type	Unit testing



Test case number	02
Test case name	Create project
Test case description	This test case is for creating project
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User fill the fields of form.
2	On clicking create button system validate fields.
3	After validation form store in database and show the success message.

#### **Create project page**

### **3.Add user**

Tested By	Abdullah Talat
Test type	Unit testing
Test case number	03
Test case name	Add user
Test case description	This test case is for adding member
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User fill the fields of form.
2	On clicking add button system validate fields.

3	After validation form store in database and show the success message.
---	---

#### **Add user page**

#### **4.View Project**

Tested By	Abdullah Talat
Test type	Unit testing
Test case number	04
Test case name	View project
Test case description	This test case is for viewing project
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User click on view project
2	System fetch record of project.
3	User view list of project.

#### **View project page**

#### **5.View user**

Tested By	Abdullah Talat
Test type	Unit testing
Test case number	05
Test case name	View user

Test case description	This test case is for viewing project
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User click on view member
2	System fetch record of project.
3	User view list of member.

**View User page**

**6.Assign project**

Tested By	Abdullah Talat
Test type	Unit testing
Test case number	06
Test case name	Assign project
Test case description	This test case is for assigning project.
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User fill the fields of form.
2	On clicking add button system validate fields.
3	After validation form store in database and show the success message.

**Assign Project page**

## 7.View Notification

Tested By	Sharjeel Wakeel
Test type	Unit testing
Test case number	07
Test case name	View Notification
Test case description	This test case is for viewing notification
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User receive alert when notification receive.
2	User open notification list click new notification which is bold from seen notifications.
3	User click on new notification and redirect to concern page.

### View Notification page

## 8.View Chat

Tested By	Sharjeel Wakeel
Test type	Unit testing
Test case number	08
Test case name	View Chat
Test case description	This test case is for viewing chat

<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User receive alert when someone send message.
2	User click on chat button sender seen on top of list of members
3	User click on member profile view chat between each other.

**View Chat page**

### 9.View Feasibility study

Tested By	Sharjeel Wakeel
Test type	Unit testing
Test case number	09
Test case name	View Feasibility study
Test case description	This test case is for viewing feasibility study
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User click on assign project.
2	User see assigned projects
3	User click on one project and check feasibility.

**View feasibility study page**

### 10.Add scope

Tested By	Sharjeel Wakeel
Test type	Unit testing
Test case number	10
Test case name	Add scope
Test case description	This test case is for adding scope
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User click on assign project.
2	User see assigned projects
3	User click on one project and check scope.

#### **Add scope page**

### **11.Add Risk**

Tested By	Sharjeel Wakeel
Test type	Unit testing
Test case number	11
Test case name	Add Risk
Test case description	This test case is for adding risk
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User click on assign project.

2	User see assigned projects
3	User click on one project and check risk.

#### **Add Risk page**

### **12.Add Schedule**

Tested By	Sharjeel Wakeel
Test type	Unit testing
Test case number	12
Test case name	Add Schedule
Test case description	This test case is for adding schedule
<b>Procedural Steps</b>	<b>Procedural Steps</b>
1	User click on assign project.
2	User see assigned projects
3	User click on one project and add schedule.

#### **Add Schedule page**

#### **6.1.1. Static testing:**

Static test techniques provide a great way to improve the quality and productivity of software development. It includes the reviews and provides the overview of how they are conducted.

#### **6.2. Validation:**

Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e., it checks what we are developing is the right product. it is validation of actual

and expected product. Validation involves white box testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. E.g.; scheduling method, adding members.

### **6.3. Usability Testing:**

**Usability Testing** also known as User Experience (UX) Testing, is a testing method for measuring how easy and user-friendly a software application is. A small set of target end-users, use software application to expose usability defects. Usability testing mainly focuses on user's ease of using application, flexibility of application to handle controls and ability of application to meet its objectives. In our project some users test our system and they all are successfully registered. One user finds the defect in our scheduling module then we replace with other scheduling method.

### **6.4. Module/Unit Testing:**

Unit testing is a confusing part of the software development process. Unit testing involves individually testing unit of code separately to make sure that it works on its own, independent of the other units. Unit testing is essentially a set of paths, test performed to examine the several different paths through the modules. Unit testing is remarkably done by programmers with the help of Unit framework. Unit testing is usually an automated process and performed within the programmers IDE. Unit testing is an action used to validate that separate units of source code remains working properly. In our project we test the object detection, scheduling and creating project modules individually for better results.

### **6.5. Integration Testing:**

**Integration Testing** is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. In our project, we integrate scheduling and creating project modules and they show results successfully.

### **6.6. System Testing:**

System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS). In our project we give different inputs as user required from our system initially and get ninety percent accuracy in it.

### **6.7. Acceptance testing:**

Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not. Acceptance



Testing is the last phase of software testing performed after System Testing and before making the system available for actual use. In our system user check all details, initially face some defects but after negotiating and resolving some defects they accept our system.

### **6.8. Stress Testing:**

**Stress Testing** is a type of software testing that verifies stability & reliability of software application. The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn't crash under crunch situations. It even tests beyond normal operating points and evaluates how software works under extreme conditions.

## **Chapter 7: Conclusion and Future Work**

This chapter concludes the project and highlights future work.

### **7.1. Conclusion**

Project have two user admin and members.

Admin have authority to create projects and assign to the member. Admin also add members who work on project.

Members work on a project they have work space where they perform function on a project.

Both admin and member communicate to each other.

They receive notification when someone perform function on an assigned project.

### **7.1. Future Work**

We are planning to do following advancements in our project in future.

- Add live meeting session between admin and member on projects discussion purpose.
- Add voice note message functionality in chat system.
- Appreciation system embedded when task done by member
- Improvements in the designing and performance of software.