



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

Project Name
Intelligent Project Alliance

By

Azeem Sarwar	18-ARID-3011
Rehan Asghar	18-ARID-3036
Irtaza Zulfiqar	18-ARID-3018

Supervisor

Mr. Zeeshan Javed

Bachelor of Science in Software Engineering (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 the 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.

Student Name 1

Student Name 2

Student Name 3

Azeem Sarwar

Rehan Asghar

Irtaza Zulfiqar

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (SE) **“Intelligent Project Alliance”** was developed by **“Azeem Sarwar, 18-ARID-3011”**, **“Rehan Asghar, 18-ARID-3036”** and **“Irtaza Zulfiqar, 18-ARID-3018”** under the supervision of **“Mr. Zeeshan Javed”** and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelor of Science in Software Engineering.

Mr. Zeeshan Javed
Supervisor

External Examiner (If any)

Administrator UIIT

Executive Summary

To effectively manage different projects in an organization's managers need an application that makes it easy for them to do so. However, many different platforms are providing the management facilities to manage projects but all these runs by third parties. Thousands of managers visit multiple platforms daily to manage their projects on a daily, weekly, and monthly basis. So, they find it very difficult to manage their routines. According to our research about this problem, that is the most consuming thing that the Management face daily and this area needs to be explored till today. So, we have come up with a

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 **“Mr. Zeeshan Javed ”** for personal supervision, advice, valuable guidance, and completion of this project. We are deeply indebted to him/them 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.

Student Name 1

Student Name 2

Student Name 3

Azeem Sarwar

Rehan Asghar

Irtaza Zulfiqar

Table of Contents

Introduction 1

1.1	Brief.....	9
1.2	Relevance to Course Modules.....	9
1.3	Project Background.....	10
1.4	Literature Review.....	11
1.5	Methodology and Software Life Cycle.....	11

Problem Definition 12

2.1	Purpose.....	13
2.2	Product Functions.....	13
2.3	Proposed Architecture.....	14
2.4	Project Deliverables.....	14
2.5	Operating Environment.....	15
2.6	Assumptions and Dependencies.....	16

Requirement Analysis 17

3.1	Functional Requirements.....	18
3.2	Non – Functional Requirements.....	19
3.3	Use case Model.....	20
3.3.1	Use Case Diagram.....	20
3.3.2	Actors Description.....	20
3.3.3	Use Case Description.....	20

Design and Architecture

4.1.	System Architecture.....	31
4.2.	System Design.....	31
4.2.1	UML Structural Diagrams.....	31
4.2.1.2	Deployment Diagram.....	32
4.2.2	UML Behavioral Diagrams.....	33

4.2.2.1 Activity Diagram.....	33
4.2.2.2 ERD Diagram.....	40
4.2.3 UML Diagrams.....	40
4.2.4 Sequence Diagrams.....	41
4.3 Class Diagram.....	46
Implementation	47
5.1 Component Diagram.....	47
5.2 User Interface.....	48
Testing and Evaluation	52
6.1 Verification	52
6.1.1 Functional Testing	52
6.1.2 Static Testing	55
6.2 Validation.....	55
6.3 Usability Testing.....	56
6.4 Unit Testing	56
6.5 Integration Testing.....	56
6.6 System Testing	56
6.7 Acceptance testing.....	56
6.8 Stress Testing.....	57
6.9 Hardware Configuration Testing.....	57
6.10 Evaluation.....	57
6.11 Deployment.....	57
6.12 Maintenance.....	57
References	57
[1]: https://openProject.com/	
[2]: https://www.atlassian.com/software/jira	

List of Figures

Figure 1 .1: Agile LifeCycle	112
Figure 3.1: Use Case Diagram	18
Figure 4.1: Deployment Diagram.....	30
Figure 4.2: Activity Diagram of Intelligent Project Alliance.....	31
Figure 4.3: Activity Diagram for Schedule Management.....	32
Figure 4.4: Activity Diagram for Registration	33
Figure 4.5: Activity Diagram for Create New Project.....	34
Figure 4.6: Activity Diagram for Task Management	35
Figure 4.7: ER Diagram of Intelligent Project Alliance.....	36
Figure 4.8: Sequence Diagram for Registration.....	37
Figure 4.9: Sequence Diagram for Verify Email.....	38
Figure 4.10: Sequence Diagram for Login.....	38
Figure 4.11: Sequence Diagram for Verify Username & Password.....	39
Figure 4.12: Sequence Diagram for Create New Project.....	40
Figure 4.13: Sequence Diagram for Add Schedule.....	40
Figure 4.14: Sequence Diagram for Add Member.....	41
Figure 4.15: Sequence Diagram for Remove Member.....	41
Figure 4.16: Sequence Diagram for Add Cost	42
Figure 4.17: Sequence Diagram for Update Member.....	42
Figure 4.18: Sequence Diagram for Edit Report.....	43
Figure 4.19: Sequence Diagram for Generate Report.....	44
Figure 4.20: Sequence Diagram for Add Comments.....	44
Figure 4.21: Sequence Diagram for Edit Comments.....	45
Figure 4.22: Class Diagram of Intelligent Project Alliance.....	46
Figure 5.1: Component Diagram	48
Figure 5.2: Splash Screen	49
Figure 5.3: Sign Up	50
Figure 5.4: Dashboard.....	51
Figure 5.5: Add Mmber.....	51
Figure 5.6: Reporting Screen	52
Figure 5.7: Goal Management	52
Figure 5.8: Chat Module.....	53
Figure 5.9: Project Schedule.....	53

Chapter 1: Introduction

1.1. Brief

The "**Intelligent Project Alliance**" is a cross-platform desktop application. And it is a project management tool to manage projects. It is a platform that users and management communities can use to set up their routine tasks on a daily, weekly, or monthly basis and also can do a complete scope management and cost management in it. This application will help them to estimate their project cost according to their schedule and work packages. Intelligent Project Alliance will assist a team in organizing and managing their projects and tasks effectively. These tools will allow you to plan and delegate work in one place with tasks, subtasks, folders, templates, workflows, and calendars. User can build a better way of working with your team assign tasks, add comments and organize work.

The user can track and assess productivity and growth through resource management and reporting. It will take advanced features to track projects at a higher level like Flexible work views, Dashboards, Reportings and Gantt charts representations. Gantt charts will be used to show calendar time task assignments in days, weeks or months. The tool uses graphic representations to show start, elapsed, and completion times of each task within a project. Gantt charts are ideal for tracking progress. The number of days actually required to complete a task that reaches a milestone can be compared with the planned or estimated number. The actual workdays, from actual start to actual finish, are plotted below the scheduled days. This information helps target potential timeline slippage.

It will also generate the report that will provide the overview of the project in which include the complete schedule and cost of the project. It will also help users to track and analyze your team's work throughout a project. In cost management it allows the use to estimate for forming and monitoring a project's budget summary.

1.2. Relevance to Course Modules

The Software Project Management course taught us Organizing, scheduling, allocating resources, executing, monitoring, and delivering software and web projects. IT-related projects are usually handled in an Agile manner so that they can keep pace with the accelerating pace of business and iterate on existing products.

Our app will cover these following modules:

- ❖ Module that we will add in this app is task management. The developers can set their tasks on daily bases, weekly basis or on monthly basis. They will get alerts of their task before the end time.

- ❖ Our application will also help the user to estimate the correct cost of the project.

Similar to how managers will be able to **manage employee tasks**, they will also be able to monitor their employees' performance through this module on a regular basis.

1.3. Project Background

Basically, according to our research about project management, that is the most consuming thing that the developers and management communities face on a daily basis and this area needs to be explored till today. The purpose of this project is to provide a system for users to manage their work on a daily routine. The activities we will manage include Schedule Management, Scope Management, Requirements Management, and Design Management. Users can access all project management features through a desktop application. Although users may not be able to grasp the complex tool, they can use this application without difficulty. As a result, the user will be able to select the project and plan their tasks accordingly. A cost estimate can be made after the setup task. All schedule management features will be available in design view through which the user can better understand these features visually. Furthermore, the administration can check their employees' daily schedules and track their tasks. Using the Task Management feature, you will be able to organize and assign tasks to members of your team, in addition to owning and storing them. Everyone is kept informed about the progress and upcoming steps. Maintain work portfolios, automate reports, track comments, create user accounts, create administrator accounts, manage requirements, and manage designs. Cost Management will be handled by Time Estimation, Cost Estimation, Budgeting, and Cost Reporting. Keep track of past projects and keep a backlog for future projects. Additionally, the Project Settings can be used to set meeting schedules.

1.4. Literature Review

In large organizations or new start-ups, it is necessary for us to organize, assign and manage projects for we will be able to maximize our output with less resources and with less output. Project management also allows us to reduce failures and prevent waste. Many software tools help us to manage projects. These tools provide a multitude of features including Task Management, Team Collaboration and Release Planning. We have reviewed one system that offers almost the same functionality.

Open-project is a web-based project management system for location-independent team collaboration. It is used in Project Planning, HR Management and Project Collaboration. It is available for websites not for desktop and mobile apps. In Open-project members of a team can view all the information in one place like Milestones, phases, tasks. Whether they are in progress, completed, or new. In addition, they implement the work breakdown strategy by

managing their tasks on a daily basis. The Organization have to buy its license every year and all of our information and Data stored on their server.[\[1\]](#)

Core services:

- Basic Project Management features.
- Assign Projects to team members.
- Team Collaboration

- Team Communication.

MS Project, the project management software program by Microsoft, is a very handy tool for project managers that helps them develop a schedule, assign resources to tasks, track the progress, manage the budget, and analyze workloads for an ongoing project. MS Project need training to use because it is complex tool for new user.

1.5. Analysis from Literature Review

Project management also helps us to decrease failure chances and maximum success rate because we can easily track and manage our project. Many software programs help us to manage projects. They provide features like Project Planning and Scheduling, Task Management, Team Collaboration, Product Roadmap and release Planning, cost Reporting and Budgeting. We have reviewed one system that provides the functionalities, but it is having some defects that are mentioned below.

These tools provide the functionalities but mostly in a paid mood or in a trial version and all of our information saved on their database. These tools help the user to manage their routines. The user interface of these tools is very complex and not easily understandable for the users. The non-technical person did not even use it for its work. Their configuration is also a major issue in this tool.

1.6. Methodology and Software Lifecycle for This Project

The software methodology used in **Intelligent Project Alliance** is agile method. The development of this software originated from the previous platform and came up with the solutions for Management.

We are using this technique because Agile methods **can help us to manage work more efficiently and do the work more effectively** and it will help us to delivering highest quality product within the constraint of time limit.

□ **Modules of the project:**

SDLC (Software Development Lifecycle) Agile.

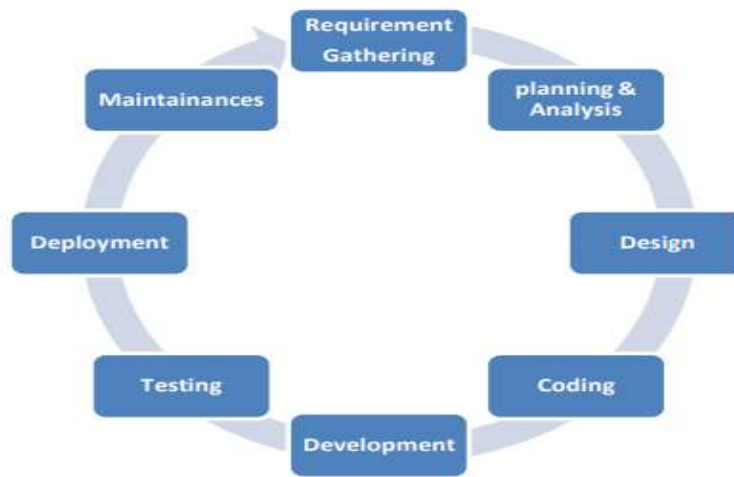


Figure 1.1: Agile Lifecycle

1.6.1. Rationale behind Selected Methodology

With Agile methodologies we aim to deliver the right product, incrementally and continuously, through small, self-organizing teams, so that we can get frequent feedback from customers and subsequently make changes as necessary. As stated in the Agile Manifesto, the four core values of Agile software development are interaction and collaboration over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and Sprint Reviews are held to inspect the result of the Sprint and to determine future adaptations.

Following are some benefits that Agile Process Model will offer:

- Generates working software quickly and early during the software life cycle.
- An extensive selection of agile development methodologies.
- Fixed scheduling and predictable task delivery.
- Improved final product quality and user experience.
- Greater transparency and visibility for key stakeholders.

Chapter 2: Problem Definition

2.1. Purpose:

Our aim is to develop a system that helps the management communities and large-scale organizations to manage their projects. In the context of the present age, we don't want to waste our money and time. Every organization needs to manage their projects to prevent failure rate and increase success rate. According to report in 2019 almost 1.9 trillion-dollar money lost just due software errors because those software are not develop by using with proper management. We want to save this money by providing software that helps in management and planning of software development. Our aim is to provide a system that helps run on the local server of organization and provide a desktop application and website. That can help the employees and administration to plan, manage projects, and assign resources to the team.

2.2. Product Functions:

Product Functionality is listed below:

- **Authentication**
- **Add Members**
- **Remove members**
- **create project**
- **Manage projects**
- **Project Schedule Development**
- **Manage Tasks**
- **Cost Management**
- **Reports generation**
- **Team Collaboration**

2.3. Proposed Architecture:

3 Tier Architecture has been used in our project.

We have three layers in 3 Tier Architecture:

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

- **Presentation Tier:**

We will develop a Desktop Application in which we will create interfaces by using ElectronJS and ReactJS Language.

- **Application Tier:**

In this layer we will apply logics in our Desktop Application by using JavaScript Language.

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

- **Data Tier:**

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

In this layer we will use Asp.net core & MySQL Server which will manage to store the data of users.

2.4. Project Deliverables:

Following are the deliverables of our project.

1. Project Alliance Application for Mac, Linux and windows.
2. C# backend build.
3. SRS.

2.5. Operating Environment:

Operating environment for the Intelligent Project Alliance is as listed below:

Operating System: Window 7, 8 ,10,11, Linux and MAC Operating system

Database: Asp.net & MySQL (.net 5.0)

Platform: Front end will be designed in **Electron JS & React JS** and backend will be designed in **Asp.net & MySQL**.

2.6. Assumptions and Dependencies:

Assumptions:

The desktop application developed from this technique is more efficient than others. This software gives perfect results in real time for finding and managing tasks.

Dependencies:

This software does not depend on any third-party tool but we need to deploy our backend on our server so that we can use it remotely and collaborate with our team.

Chapter 3: Requirement Analysis

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

3.1. Functional Requirements

A number of functional requirements are necessary to make our system work and facilitate the user's experience. Since all of these functional specifications are interface-less, they can also be assumed to be interface requirements.

- The system will authenticate the user through login functionality.
- The system will display and handle the projects of the organization.
- Admin can create new projects, add new members, assign projects to teams and generate reports of projects.
- The system will allow the admin to add new employees and assign projects to employees.
- Employees have a team lead role that Manage their Projects, Divide Projects into smaller and more manageable. Assign different tasks to his Teammates.
- Other Employees or teammates work on their assigned task or work.
- Admin or team lead can schedule meetings. Notify other concerns employed with notification.

3.2. Non-Functional Requirements

These requirements form the constraints of the system and can be divided into two parts.

- Hardware requirements
- Software requirements

3.2.1 Hardware Requirements:

These technologies need an immense hardware resources which includes.

- Minimum memory for application will be 500MB RAM.
- Server will be required for backend database storage
- A minimum processor speed of 1GHz.

A minimum free hard disk space of 500MB.

3.2.2 Software Requirements:

The system would require:

- 64-bit and 32-bit Windows, Windows 7, and windows 8, window 10 Operating system.
- Macbook Book Pro including M1 and Macbook Air.
- Linux

3.3. Non-Functional Requirements To The User:

3.3.1 Portability:

The effort required to move the software to a different target platform. The system can easily be moved and executed on new systems without any difficulties. This will facilitate software portability.

3.3.2 Availability:

The program should be flexible and should be available within a mean time and should work in any operating environment like (linux, Window and Macbook).

3.3.3 Useability:

Usability of our systems are straightforward to use. Any type of person can easily use it. Like the other complex softwares. It is not design only for tech people. Accessibility is another crucial element for our project.

3.3.4 Deployability:

Our software's deployability requirement type is very straightforward, low-risk and on a just a single-button event.

3.3. 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 Intelligent Project Alliance.

3.3.1 Use Case Diagram:

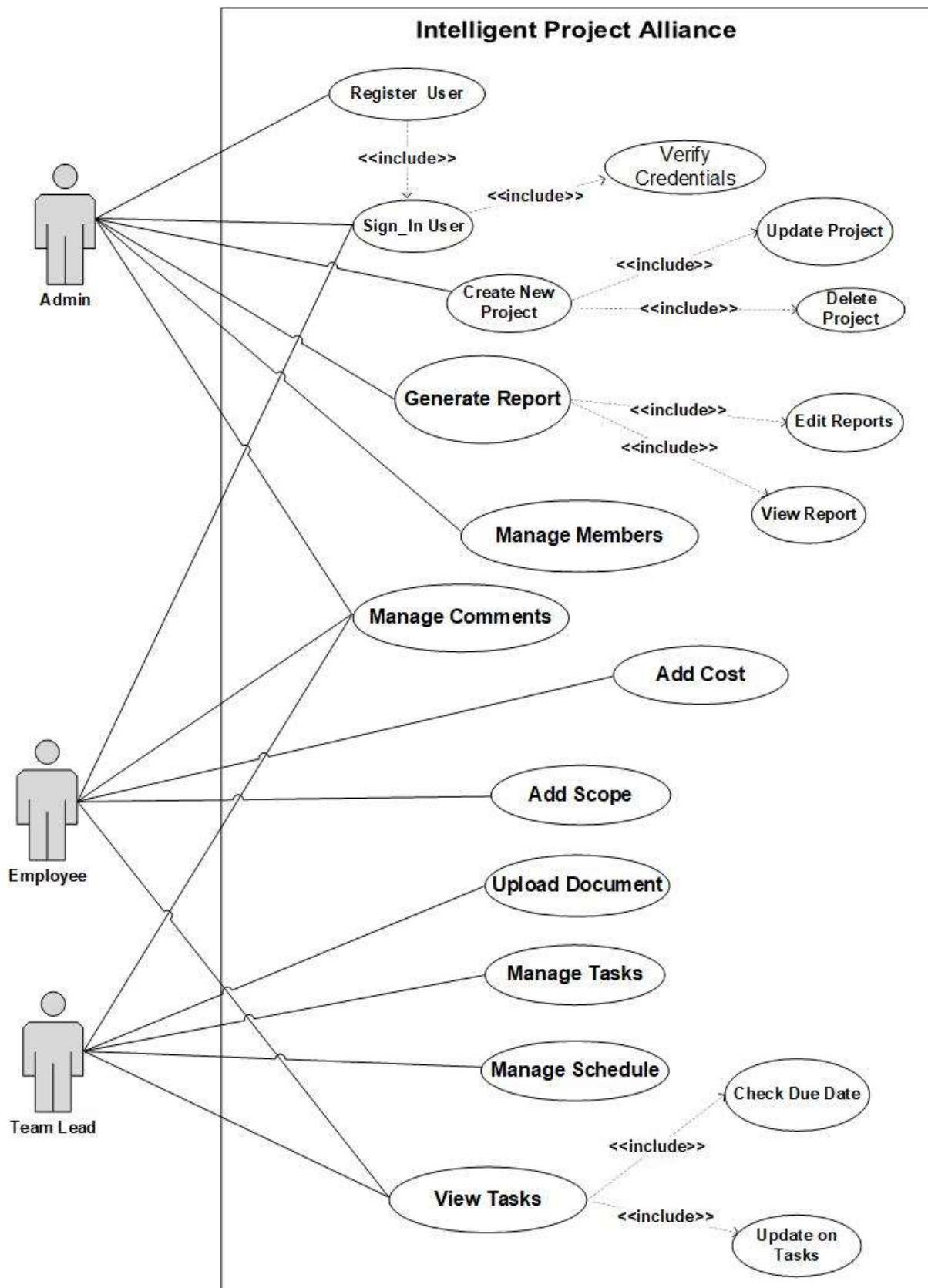


Figure 3.1: Use Case Diagram

3.3.2 Actors Description:

We have these actors in my use case.

- **Admin:**

Admin will be the primary actor which will directly interact with the application. Admin can register, Sign-in, create new Project, can manage project access, Can Add Project, Edit Projects, Edit Projects Schedule, Adding Requirements, Add Accounts, Edit Admin Info and Manage Cost Of Projects. Admin will give access to other persons that will be in the organization. He will manage all project activities.

- **Employee:**

Employees will be those people who will perform assigned projects that are assigned by the admin. They are able to develop the schedule of the project and divide the project into more manageable tasks and modules.

- **Team Lead:**

One of the employees will lead all the other employees. He will be working as team lead in the project. He will manage the project schedule and task distribution process in the project.

3.3.3 Use Case Description:

Table 3.1: Register User

Use Case ID:	ID-01
UseCase Name:	Register User
Actors:	Admin
Description:	Users will register and authenticate by the system through the database.
Trigger:	When the user clicks on the register then the database will verify its credentials.
Pre-conditions:	Username, Email and password must be provided by the users.

Post-conditions:	Users will register successfully.
Normal Flow:	<ul style="list-style-type: none"> ● Credentials will be entered by users ● Credentials will be authenticated from database of the system ● User will be successfully login
Alternative Flows:	<ul style="list-style-type: none"> ● If user is not already registered, then user will first sign up ● User will then provide credentials ● User will again be authenticated through database
Notes and Issues:	Users can retry again and again until the account successfully registered. If the user faced any issue he should wait until the system reactive.

Table 3.2: Sign In User

Use Case ID:	ID-02
Use Case Name:	Sign In User
Actors:	Admin, Employee
Description:	System will verify the password and then the user will be able to use it according to their needs.
Trigger:	When individuals login to their system or employees will login to the system.
Pre-conditions:	Users should be authenticating. users should register their account before login into the system.
Post-conditions:	The user can use the app according to their needs.
Normal Flow:	<ul style="list-style-type: none"> ● Users will first give their own registered user Name and Password. ● System will verify it if the details are correct then the

	user will login successfully.
Alternative Flows:	<ul style="list-style-type: none"> • Users can not login without a registration account. • If the password is incorrect, user can forget his password. • If the user is not registered, first he/she will register then will be able to use it.
Notes and Issues:	N/A

Table 3.3: Create New Project

Use Case ID:	ID-03
Use Case Name:	Create New Project
Actors:	Admin
Description:	Users will create new projects according to their needs.
Trigger:	Users will be pressed on the create project then they will be given details about the project.
Preconditions:	User Must be Login First to create the project. Without the login in the system, he will not be able to create a project.
Post-conditions:	<ul style="list-style-type: none"> • Users will use the project to manage their work routines. • After the project is created, user can update the project. • Users will be able to Delete the project.
Normal Flow:	<ul style="list-style-type: none"> • Admin will create a project • Then, he will give access to the employees • Employees will do the tasks according to their requirements
Alternative Flow:	Users can add the existing project in the system.
Assumptions:	N/A

Table 3.4: Manage Tasks

Use Case ID:	ID-04
Use Case Name:	Manage Tasks
Actors:	Team Lead
Description:	Team Lead will assign tasks to every member in the project.
Trigger:	When a user clicks on an add task it will show the members and then he will assign tasks to them.
Preconditions:	The admin must create a project and employees are added by the admin. Then they can create new tasks.
Post-conditions:	Team Lead can edit the tasks. Team Lead can remove tasks from every individuals.
Normal Flow:	<ul style="list-style-type: none"> ● Admin will create the project ● Assign Team Members to a project. ● Team Lead can edit projects. ● Team Lead can delete the tasks
Alternative Flows:	N/A
Assumptions:	If the user will not create a project, then tasks will not be assigned to them.

Table 3.5: Manage Comments

Use Case ID:	ID-05
Use Case Name:	Manage Comments
Actors:	Admin, Employee team Lead
Description:	Admin, Employees and Team Lead will be able to add comments regarding their managed tasks.
Trigger:	When the user clicks on the Add Comment button.
Preconditions:	It will provide the facility to change comments regarding

	their tasks.
Post-conditions:	They will successfully change user comments.
Normal Flow:	<ul style="list-style-type: none"> ● Admin will be able to add new comments over there. ● Employee and Team Lead will also have an ability to add new comments
Alternative Flows:	<ul style="list-style-type: none"> ● If they want to change their comments, then they can also change comments over there.
Notes and Issues:	N/A

Table 3.6: Update Project

Use Case ID:	ID-06
Use Case Name:	Update Project
Actors:	Admin
Description:	Admin have access to edit different projects.
Trigger:	When a user clicks on an edit project.
Preconditions:	Projects must be created for editing.
Post-conditions:	Admin will successfully edit and delete different projects.
Normal Flow:	<ul style="list-style-type: none"> ● Admin will check the projects ● Admin will check the required changes ● Then admin will make the changes in the project
Alternative Flows:	<ul style="list-style-type: none"> ● If a user wants to edit and delete their projects, then the admin will be able to do that.
Notes and Issues:	Admin will have only access to edit or delete different projects.

Table 3.7: Manage Schedule

Use Case ID:	ID-7
Use Case Name:	Manage Schedule
Actors:	Team Lead
Description:	Team Lead will manage the tasks and edit project schedules and also manage their schedule in the project.
Trigger:	When a team lead wants to add a project schedule.
Preconditions:	Project must be created before updating it.
Post-conditions:	Team Lead can successfully edit project schedules.
Normal Flow:	<ul style="list-style-type: none"> Team Lead will click on the update button for editing
Alternative Flows:	<ul style="list-style-type: none"> If there is a problem regarding editing, the Employees will have a chance to change the time limit of the project.
Notes and Issues:	None.

Table 3.8: Manage Members

Use Case ID:	ID-08
Use Case Name:	Manage Members
Actors:	Admin
Description:	Admin will be able to add new members in the project. Also, will be able to remove them from the project.
Trigger:	When the user clicks on login then the actor will give them an account for signing in the project.
Preconditions:	Admin should enter members accounts in the system.
Post-conditions:	When User will login successfully then a new account will be added to the database.
Normal Flow:	<ul style="list-style-type: none"> First admin will add the members Credentials

	<ul style="list-style-type: none"> ● Credentials will be authenticated from the database of the system. ● User will successfully login and then new account data will be added into the database.
Alternative Flows:	<ul style="list-style-type: none"> ● If the user is not already registered, then the user will first sign up. ● User will then provide credentials and then a new account will be generated.
Notes and Issues:	N/A

Table 3.9: Add Cost

Use Case ID:	ID-09
Use Case Name:	Add Cost
Actors:	Employee
Description:	Employees will also add the cost of the tasks that are assigned to the employee.
Trigger:	Employees will add it individually.
Preconditions:	Project must be created and assigned to the user.
Post-conditions:	Employee will accurately estimate the project cost.
Normal Flow:	<ul style="list-style-type: none"> ● Employees will add the tasks in the project. ● Then create the task list ● After admin will add the cost of tasks individually
Alternative Flows:	<ul style="list-style-type: none"> ● Employees can also edit or delete different documents.
Notes and Issues:	None

Table 3.10: Add Scope

Use Case ID:	ID-10
Use Case Name:	Add Scope
Actors:	Employee
Description:	Employees will also add the requirements of the tasks that are assigned to the employees.
Trigger:	Employees will add it individually.
Preconditions:	Projects must be created and assigned to the employees.
Post-conditions:	Employees will accurately estimate the project scope.
Normal Flow:	<ul style="list-style-type: none"> ● Employees will add the tasks in the project. ● Then create the task list ● Afterwards, employees will add the scope of tasks individually.
Alternative Flows:	<ul style="list-style-type: none"> ● Employees can also edit or delete different documents.
Notes and Issues:	None

Table 3.11: Update on Tasks

Use Case ID:	ID-11
Use Case Name:	Update on Tasks
Actors:	Employee
Description:	The Employees will be able to submit tasks that he knows he will be able to solve himself.
Trigger:	When the user clicks on Submit Tasks.
Preconditions:	Tasks must be resolved before submitting.
Post-conditions:	Employees will submit tasks successfully.
Normal Flow:	<ul style="list-style-type: none"> ● Employees will easily submit those tasks that can be resolved.

Alternative Flows:	<ul style="list-style-type: none"> Employees can submit different tasks before resolving them.
Notes and Issues:	Employees must complete their tasks before submitting.

Table 3.12: Upload Document

Use Case ID:	ID-12
Use Case Name:	Upload Document
Actors:	Team Lead
Description:	Team lead will also be able to add documents for the scope management. When user will add the document then all the employees will add the requirements one by one then admin will finalize the document
Trigger:	When a user clicks on Add Document.
Preconditions:	Team lead must be added to the project.
Post-conditions:	Team lead will Add requirements according to the need of the project.
Normal Flow:	<ul style="list-style-type: none"> Team lead can easily upload the document in the project then all the employees will add the requirements in the project.
Alternative Flows:	<ul style="list-style-type: none"> If the team lead did not have access to upload documents, first the admin will add them in the project. Document may have already been finalized.
Notes and Issues:	N/A

Table 3.13: Generate Report

Use Case ID:	ID-13
Use Case Name:	Generate Report
Actors:	Admin
Description:	Whenever the admin wants to generate the report, the system will generate the reports that will give a detailed view of the tasks and the entire cost of the project.
Trigger:	When an Admin clicks on generate report.
Preconditions:	Employees must be working on the project. All the tasks must be assigned to them to generate reports.
Post-conditions:	Admin will successfully view the generated report.
Normal Flow:	<ul style="list-style-type: none"> ● Admin will click on the button to generate a report. ● All the task and the cost will be in creating in XML and PDF form.
Alternative Flows:	<ul style="list-style-type: none"> ● If the report did not generate, then the admin must see they have any data to generate.
Notes and Issues:	None

Table 3.14: Edit Report

Use Case ID:	ID-14
Use Case Name:	Edit Reports
Actors:	Admin
Description:	Admin can also edit the report according to their needs.
Trigger:	When an Admin clicks on edit report.
Preconditions:	Report must be generated before editing.
Post-conditions:	Updated Report will be generated after the editing.

Normal Flow:	<ul style="list-style-type: none"> ● Admin will click check the report which changes required in the generated report. ● After this admin will change the required changes in the report ● And finally, the report will be generated.
Alternative Flows:	<ul style="list-style-type: none"> ● If system did not have any data in the system, then no report will be shown for editing
Notes and Issues:	None

Table 3.15: Edit Report

Use Case ID:	ID-15
Use Case Name:	View Reports
Actors:	Admin
Description:	Once the report is generated, the admin can view it and download it.
Trigger:	When an Admin clicks on a view report.
Preconditions:	Reports must be available in the system.
Post-conditions:	All the tasks and cost of the tasks and the employees will be in the generated report.
Normal Flow:	<ul style="list-style-type: none"> ● Admin will check the availability of the report. ● When the report is available then the admin can view it.
Alternative Flows:	<ul style="list-style-type: none"> ● Reports must be available in the system.
Notes and Issues:	None

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.

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 Intelligent Project Alliance.

4.2.1 UML Structural Diagrams

Following are the UML structural diagrams of our system:

4.2.1.1 Deployment Diagram

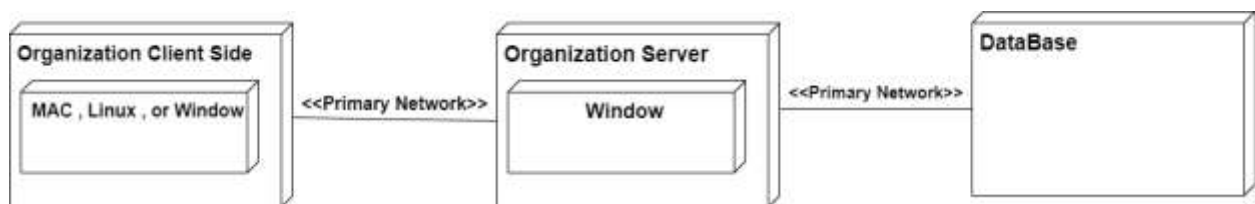


Figure 4.1: Deployment Diagram

4.2.2 UML Behavioral Diagrams

Following are the behavioral diagrams of our system:

4.2.2.1 Activity Diagram

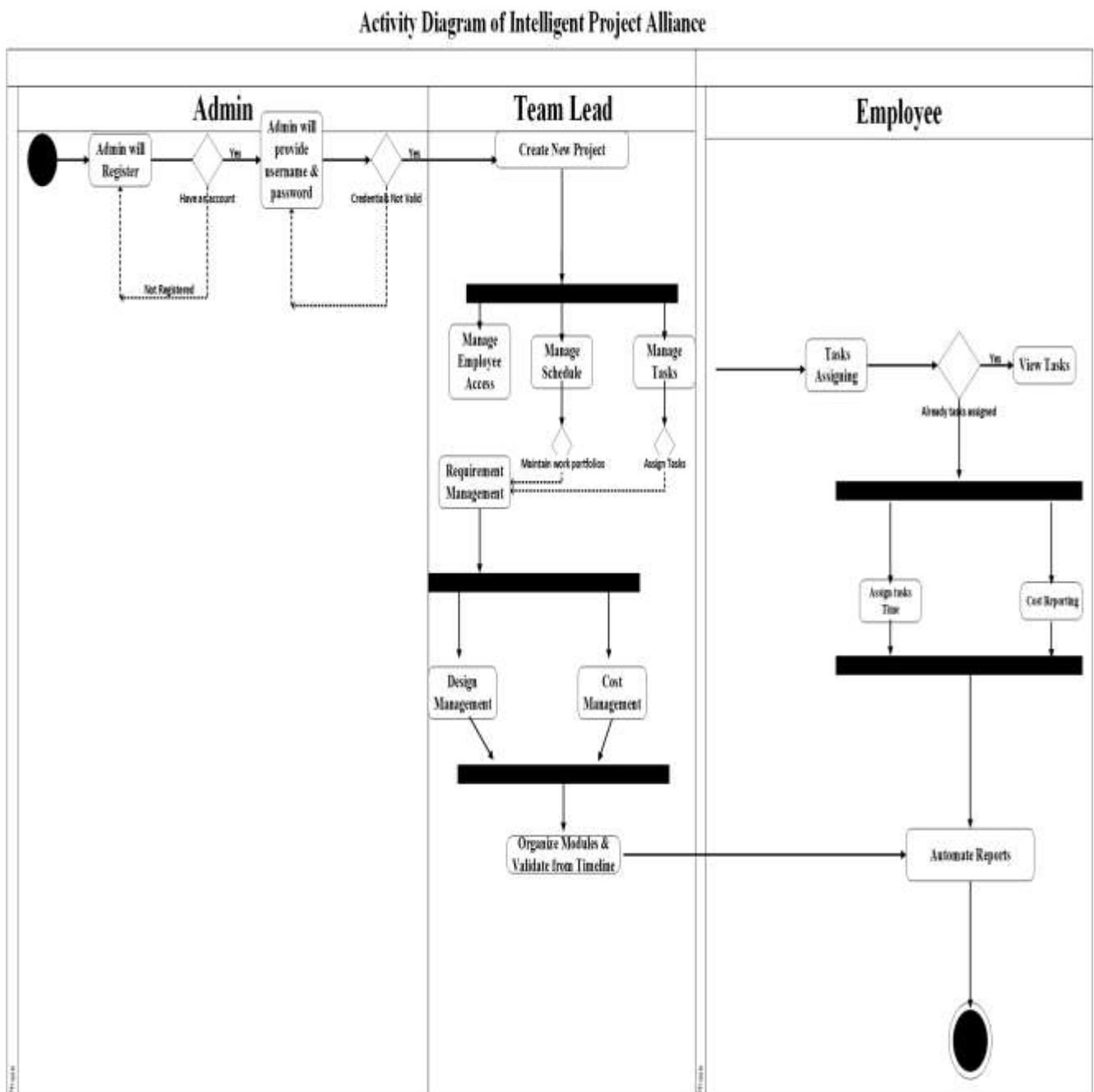


Figure 4.2: Activity Diagram of Intelligent Project Alliance

Activity Diagram For Schedule Management

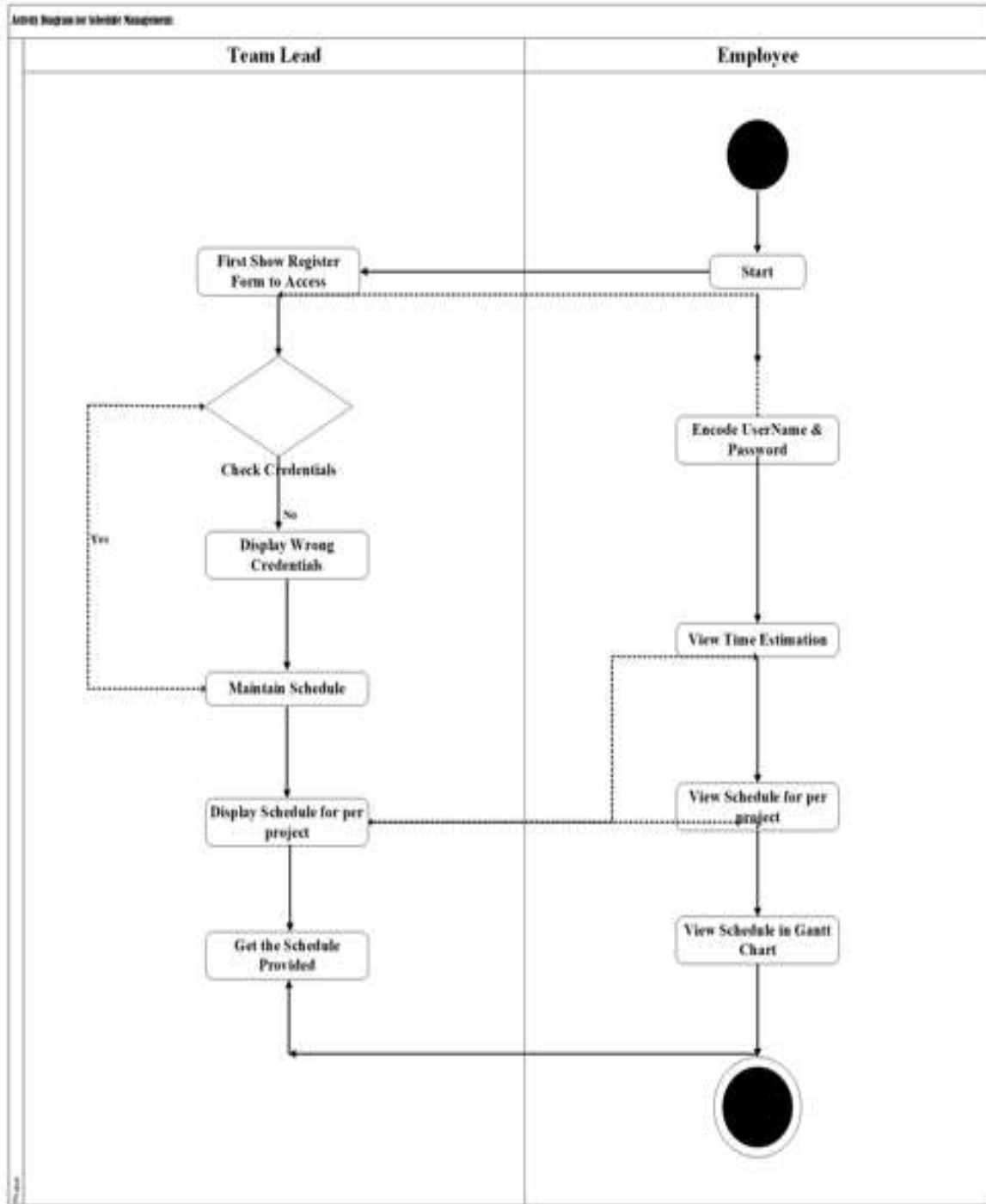


Figure 4.3: Activity Diagram for Schedule Management

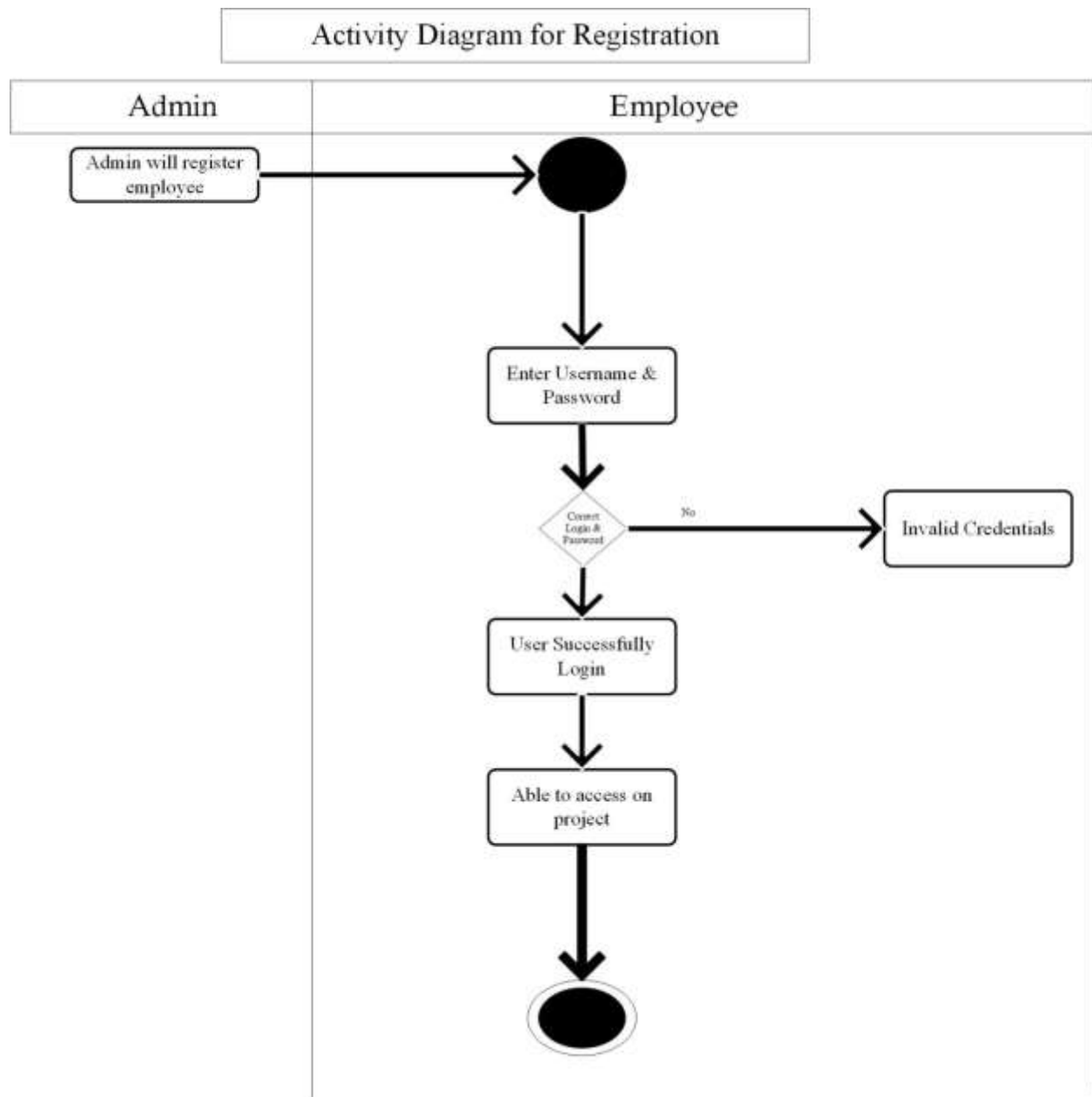


Figure 4.4: Activity Diagram for Registration

Activity Diagram for Create New Project

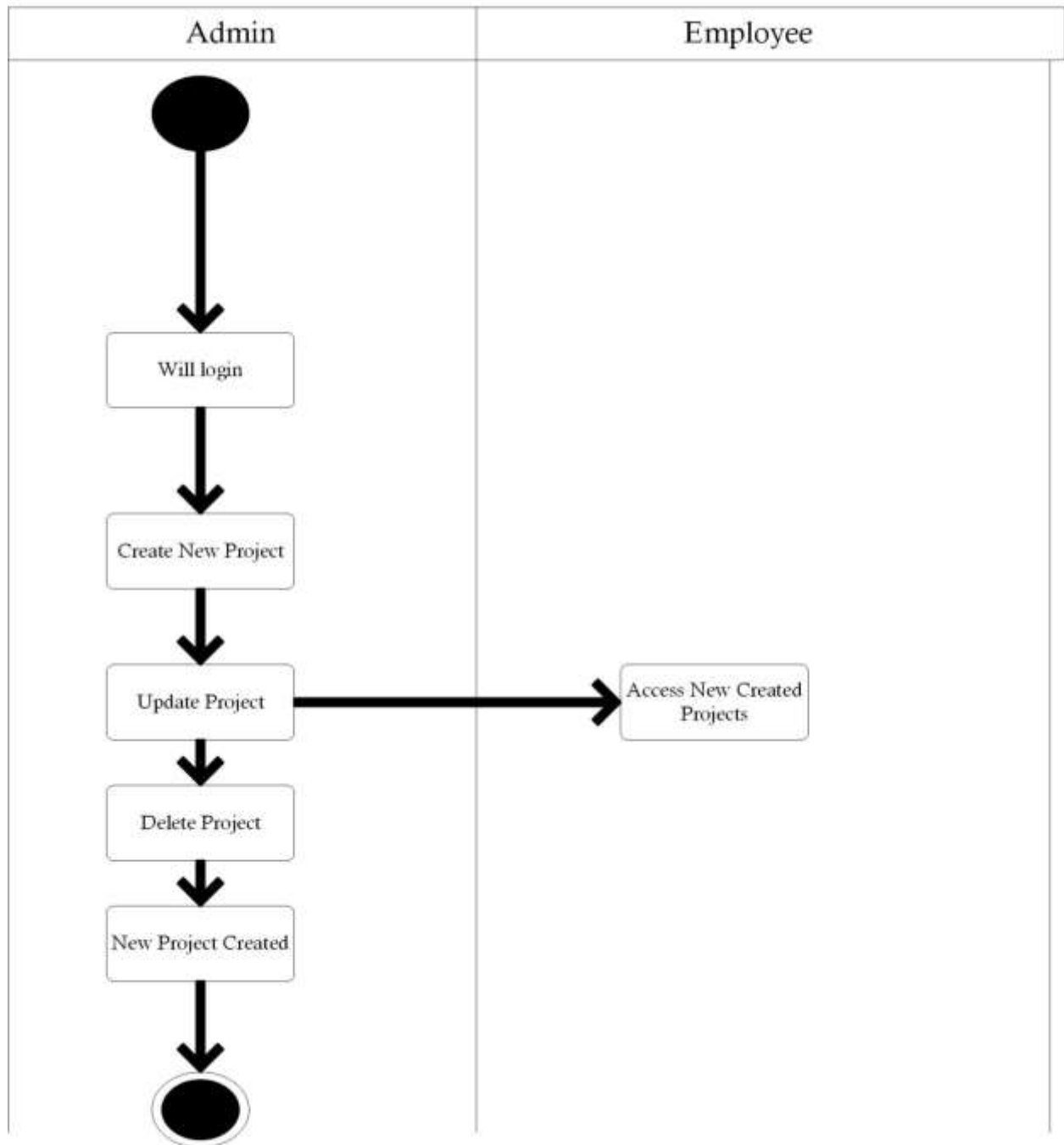


Figure 4.5: Activity Diagram for Create New Project

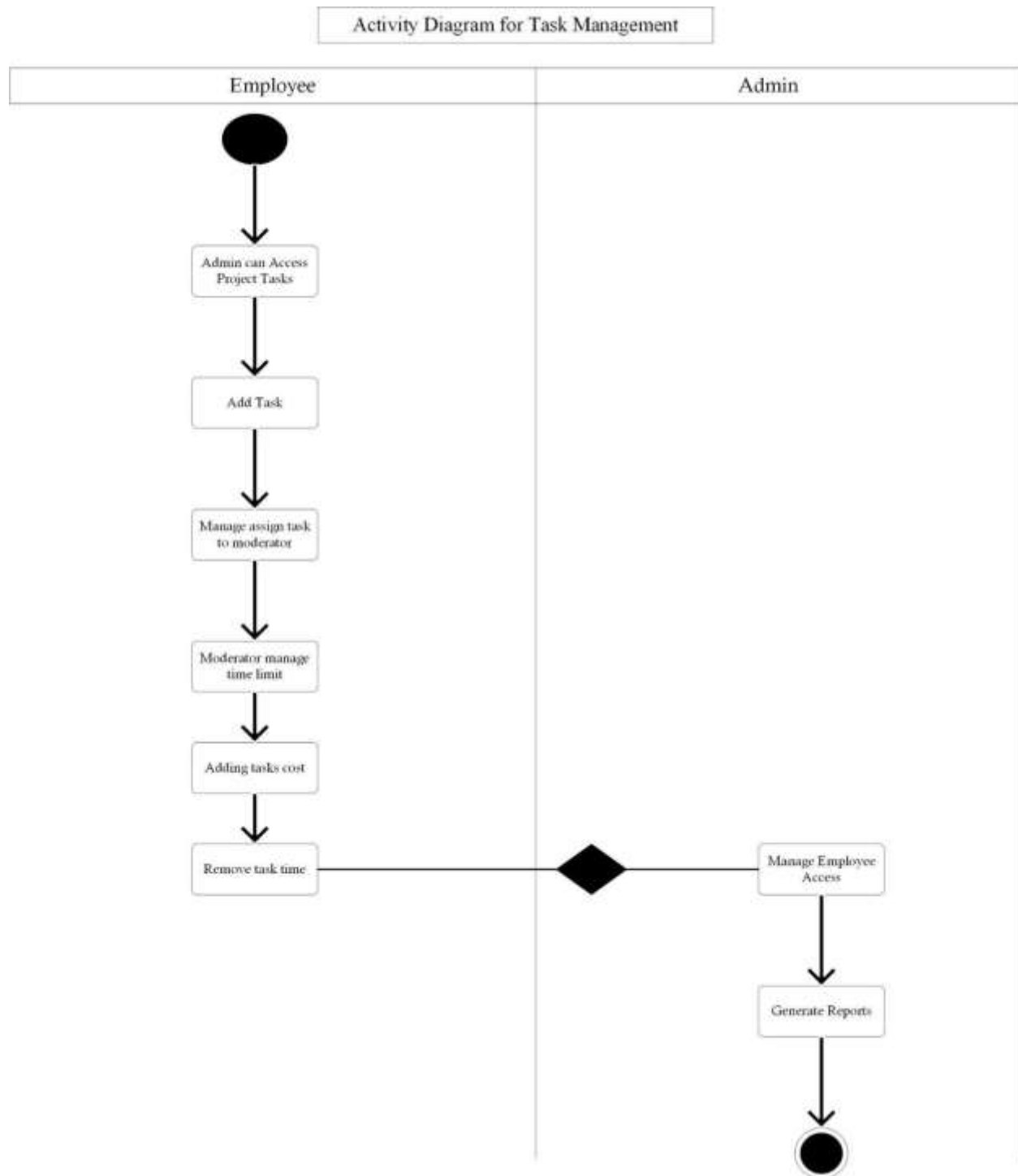


Figure 4.6: Activity Diagram for Task Management

4.2.2.2 ER Diagram

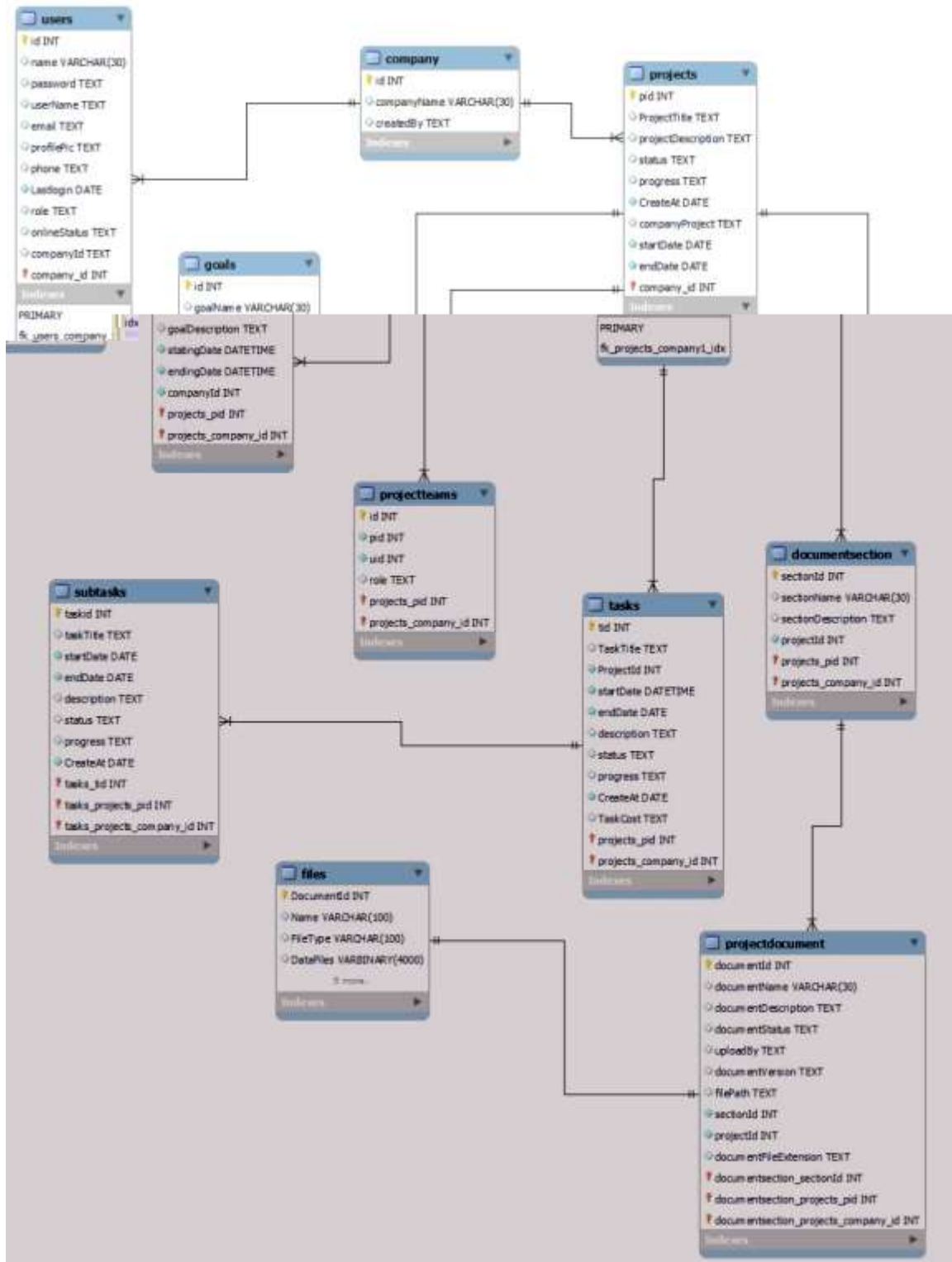


Figure 4.7: ER Diagram of Intelligent Project Alliance

4.2.3 UML Interaction Diagrams

Following are the UML interaction diagrams of our system:

4.2.2 Sequence Diagrams

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together.

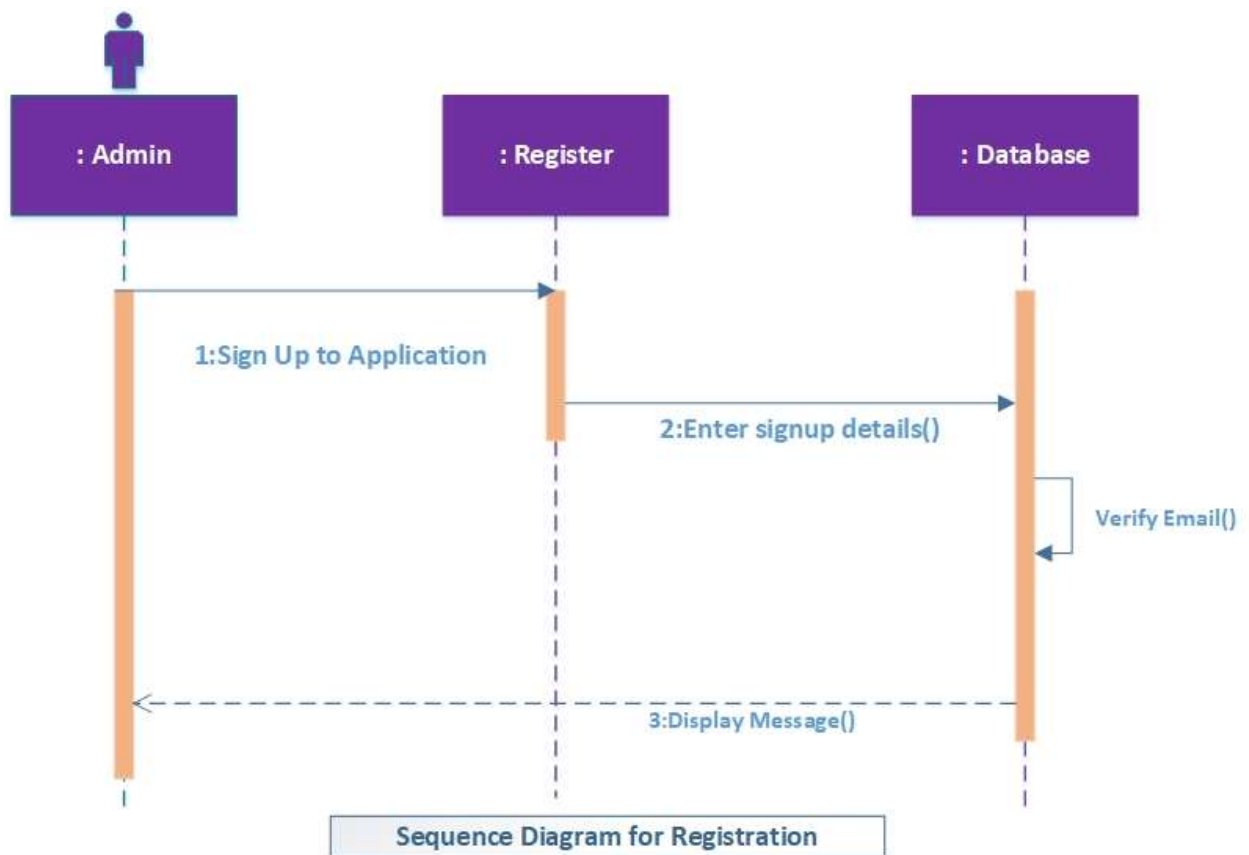


Figure 4.8: Sequence Diagram for Registration

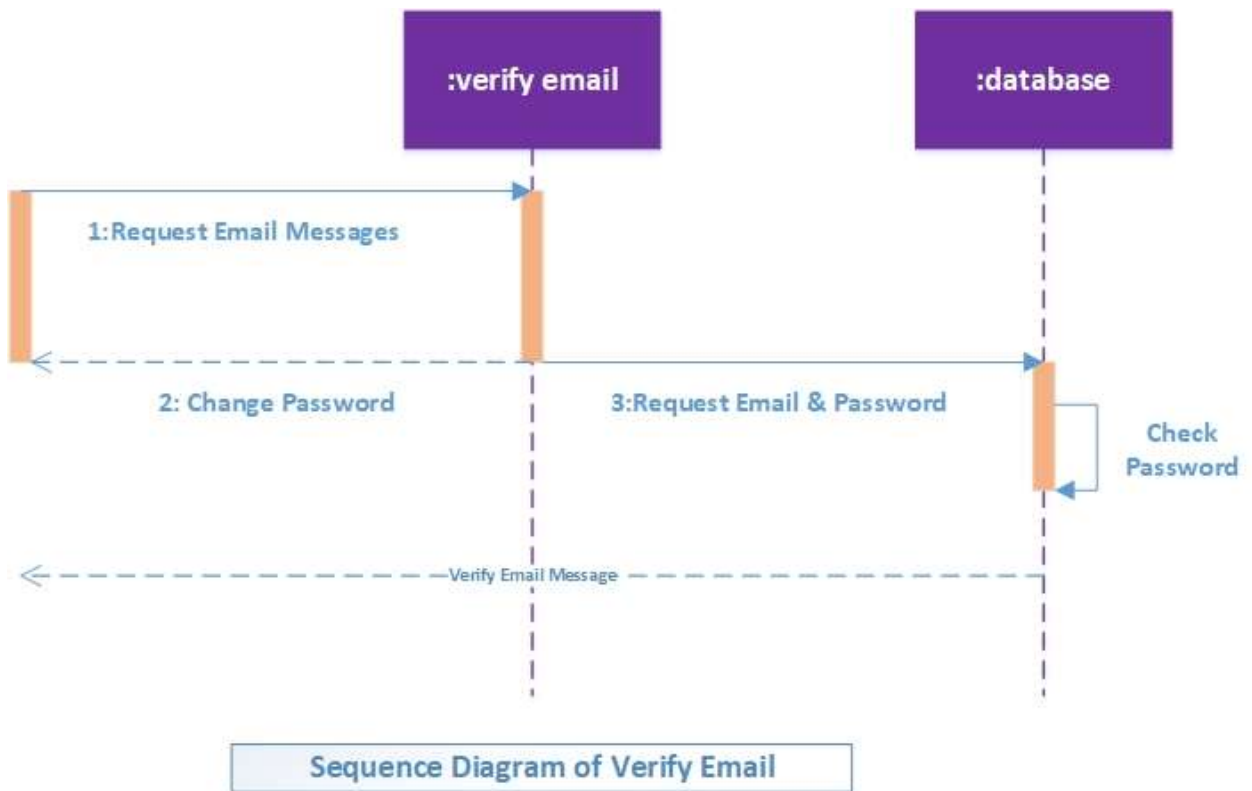


Figure 4.9: Sequence Diagram for Verify Email

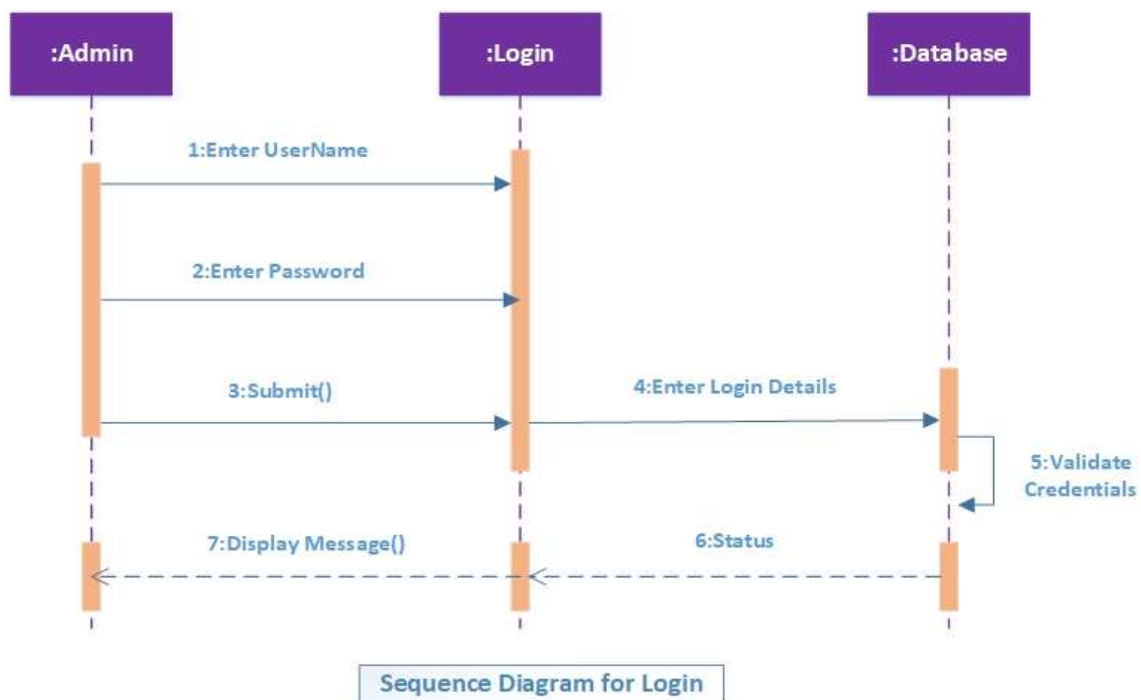


Figure 4.10: Sequence Diagram for Login

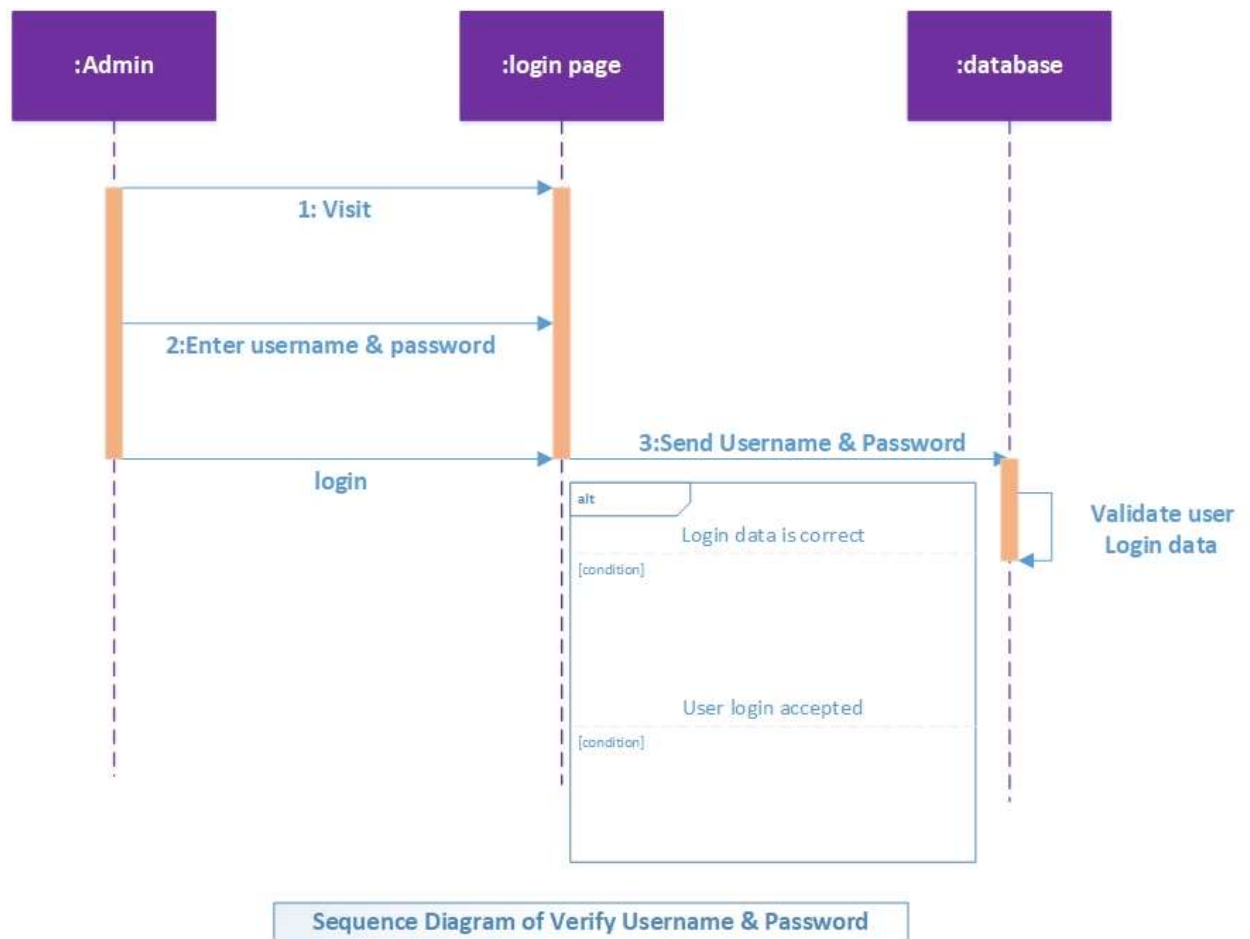


Figure 4.11: Sequence Diagram for Verify Username & Password

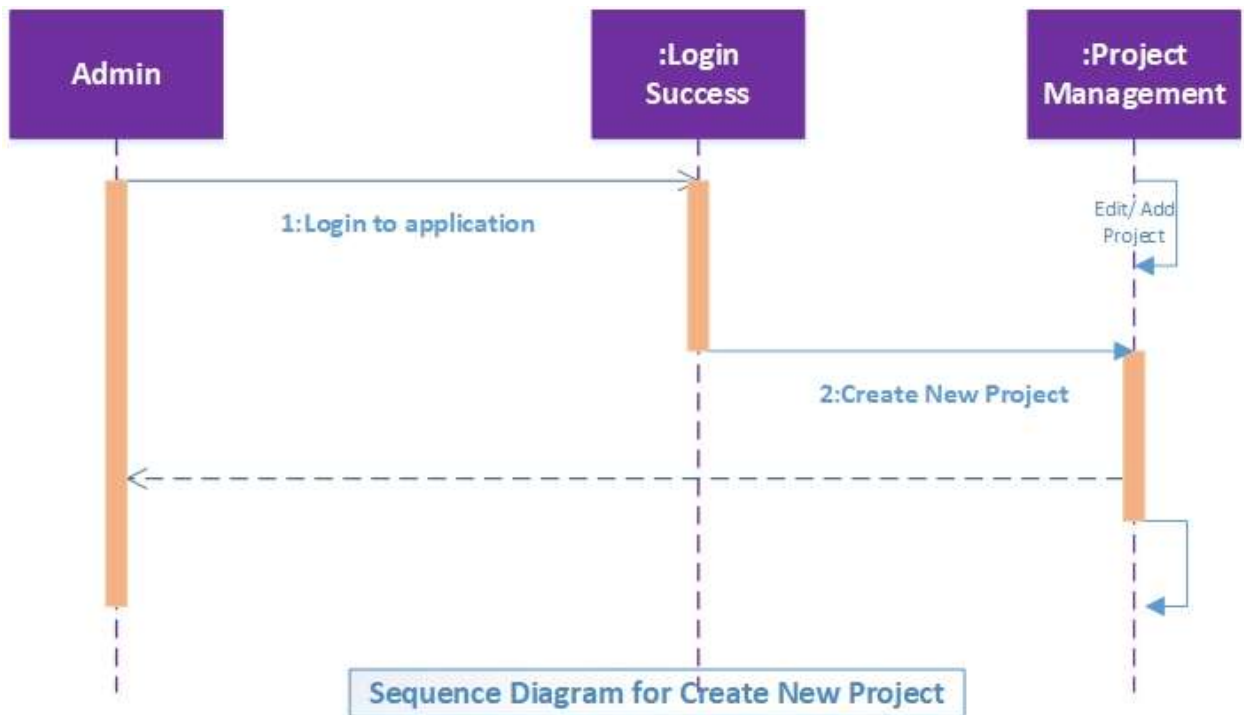


Figure 4.12: Sequence Diagram for Create New Project

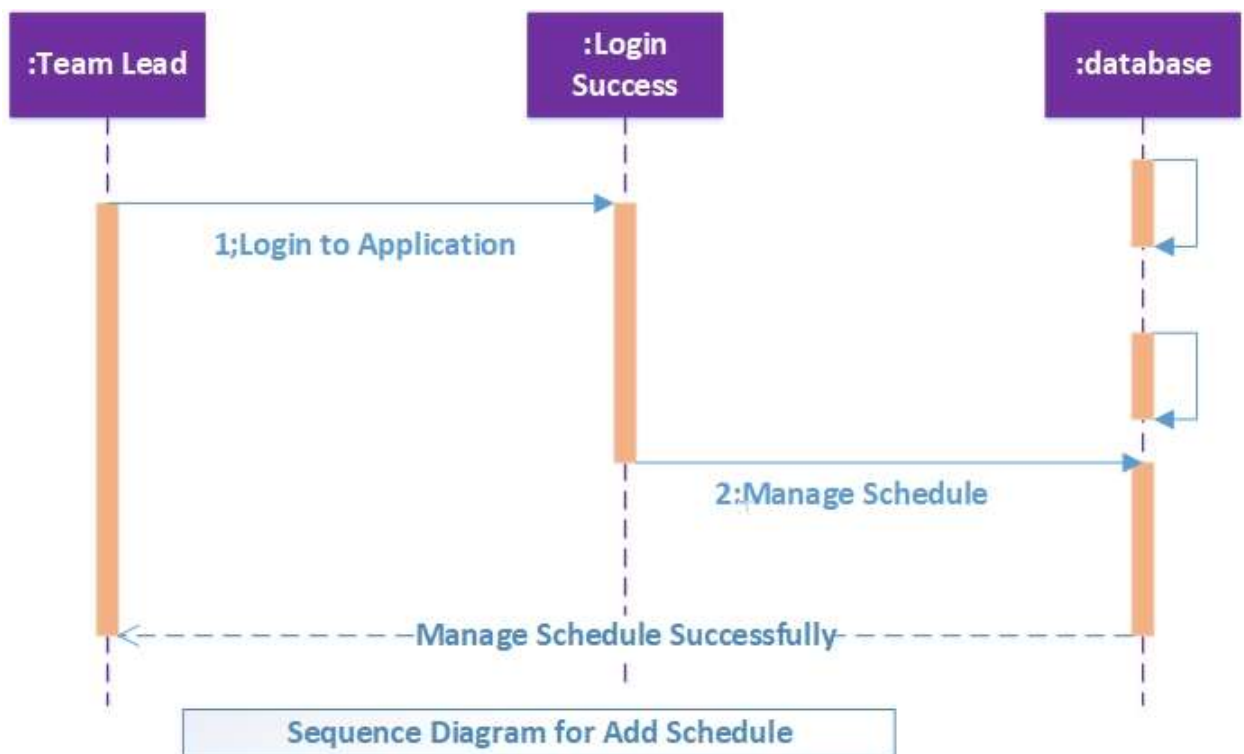


Figure 4.13: Sequence Diagram for Add Schedule

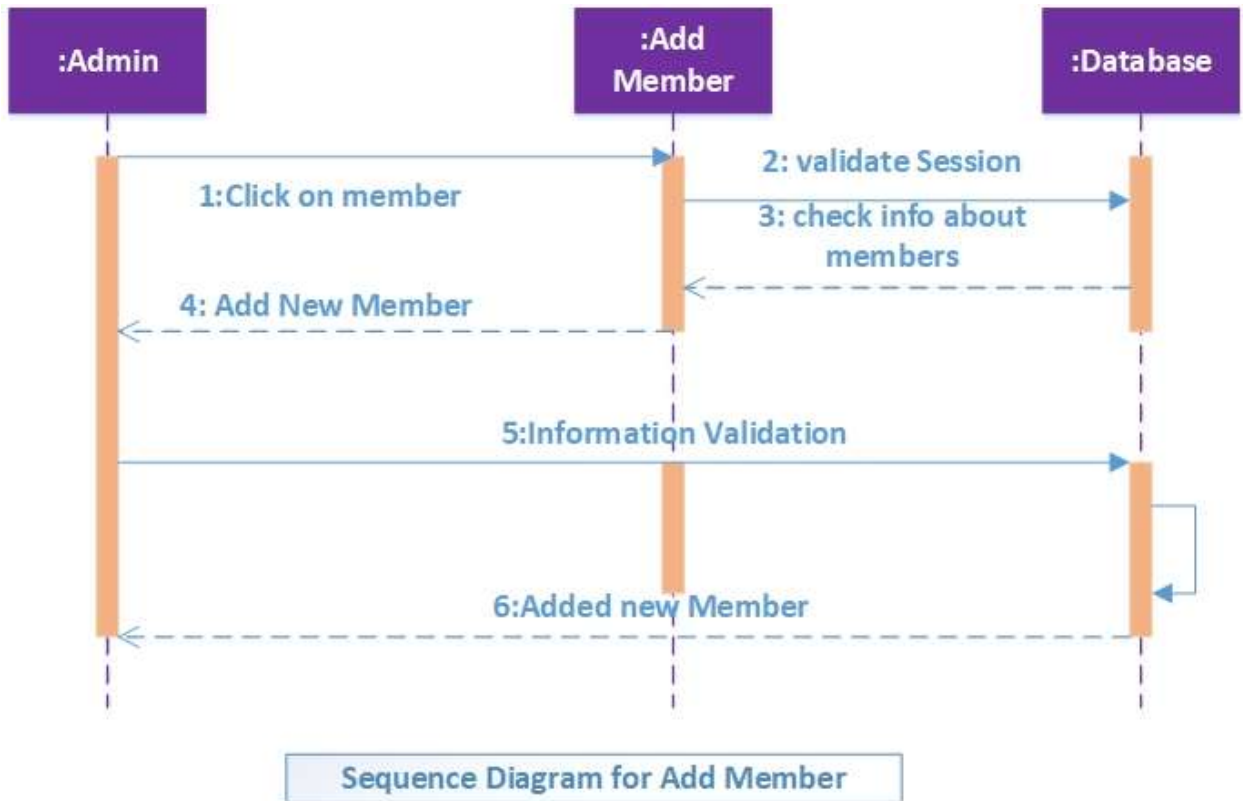


Figure 4.14: Sequence Diagram for Add Member

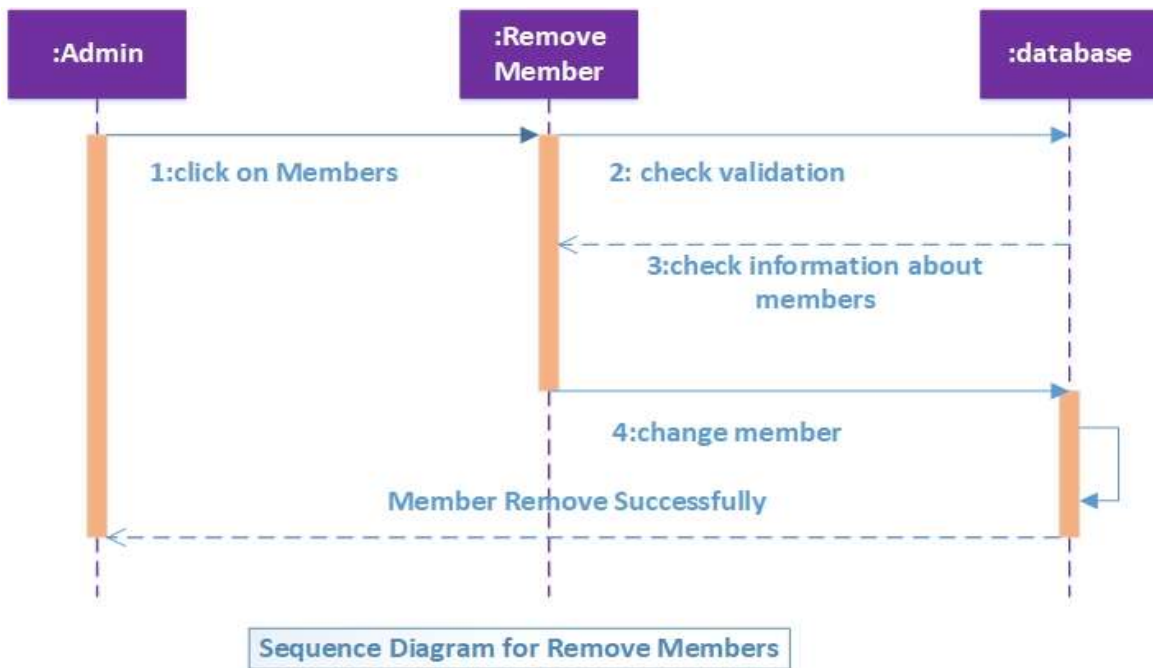


Figure 4.15: Sequence Diagram for Remove Member

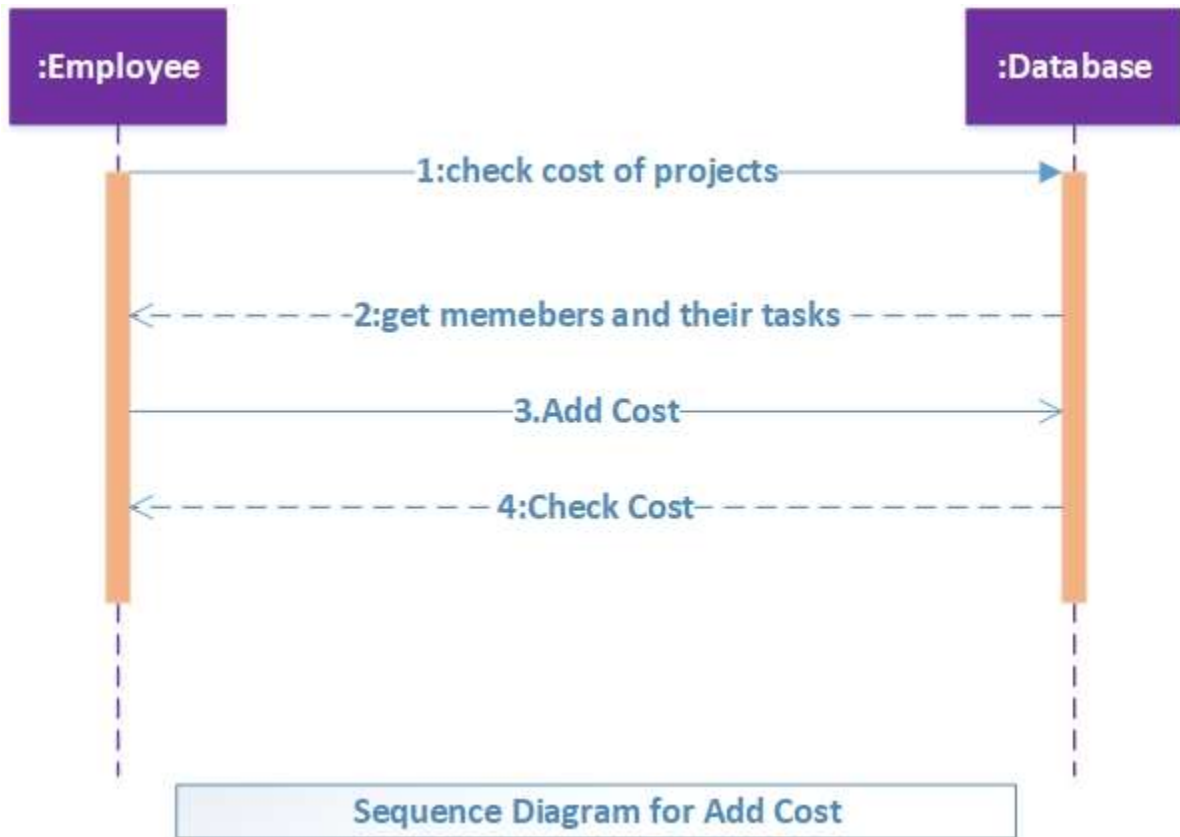


Figure 4.16: Sequence Diagram for Add Cost

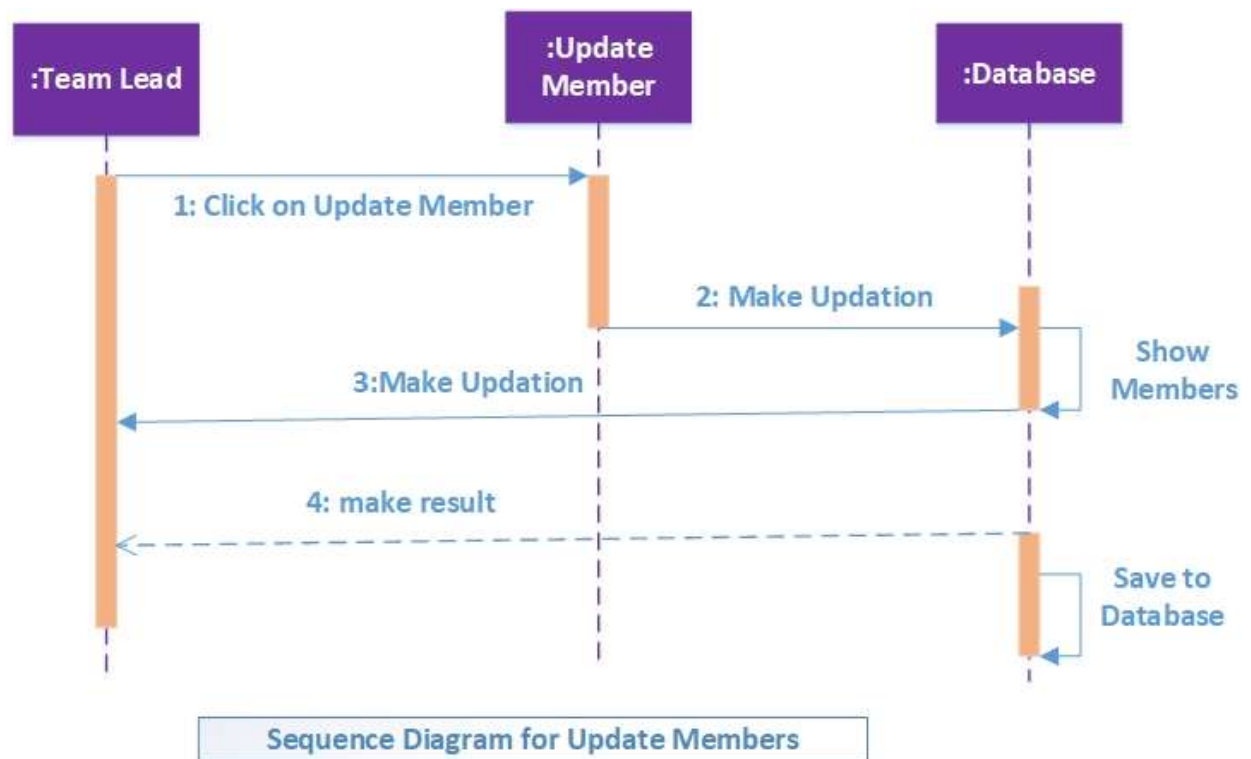


Figure 4.17: Sequence Diagram for Update Member

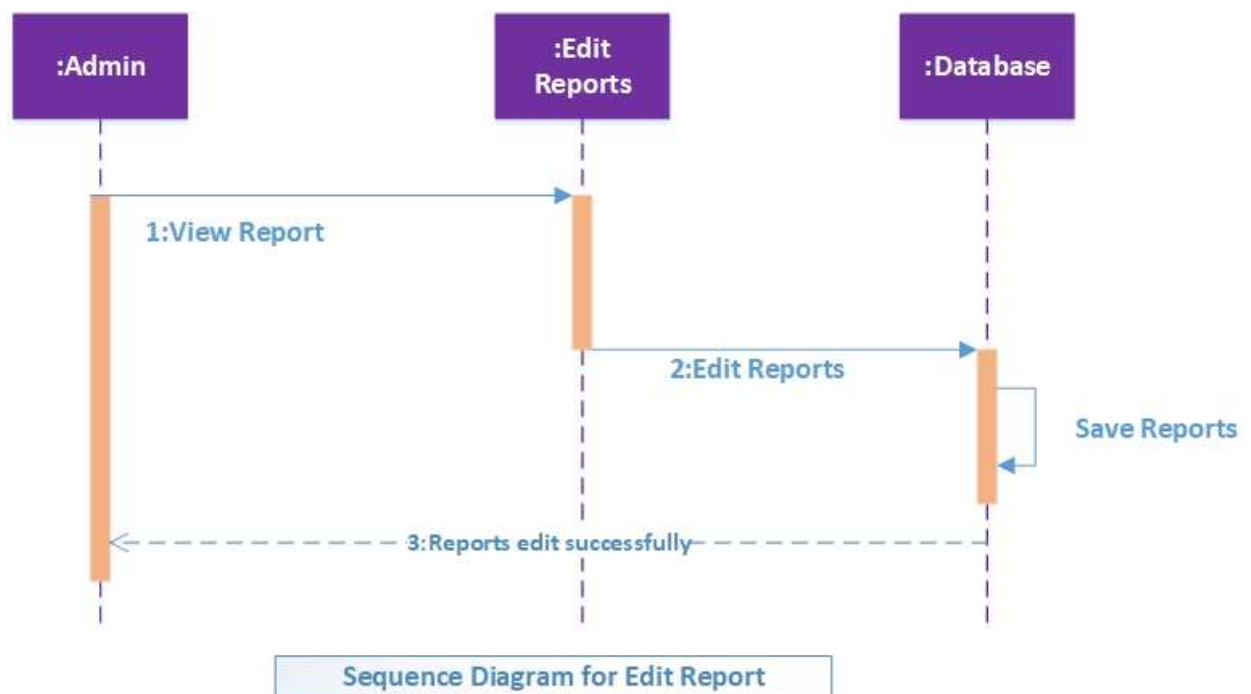


Figure 4.18: Sequence Diagram for Edit Report

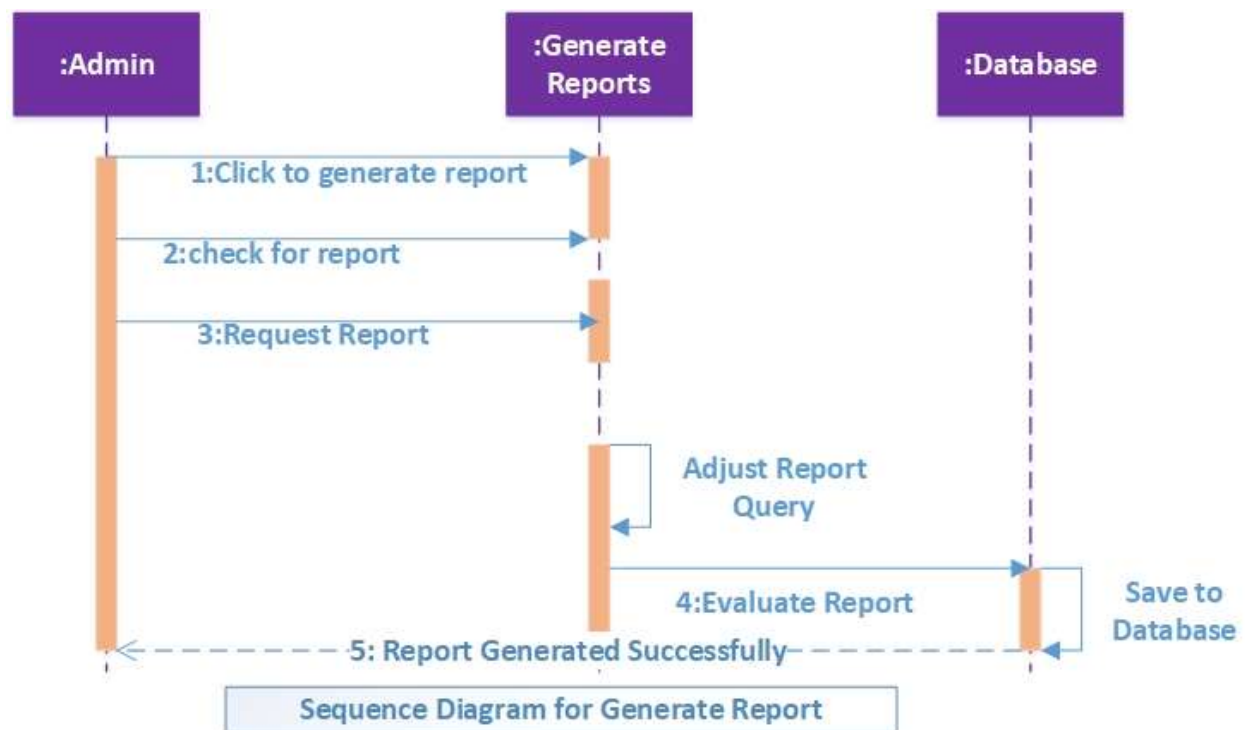


Figure 4.19: Sequence Diagram for Generate Report

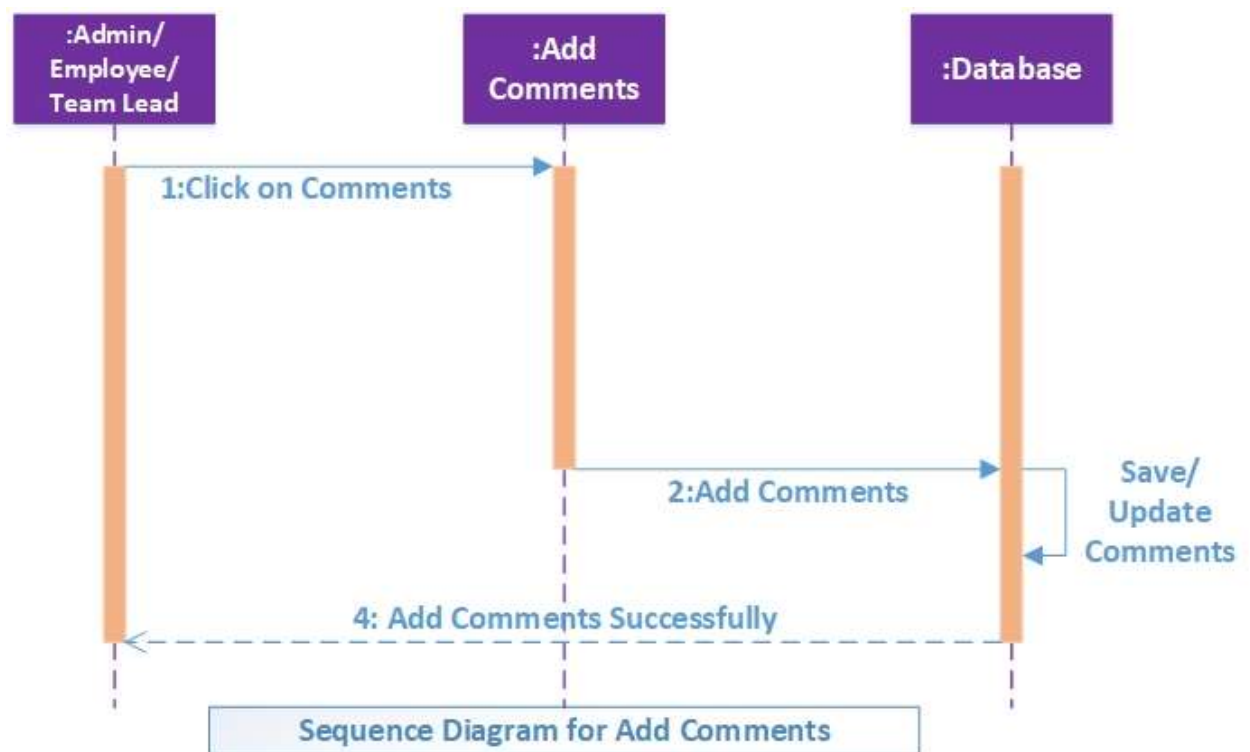


Figure 4.20: Sequence Diagram for Add Comments

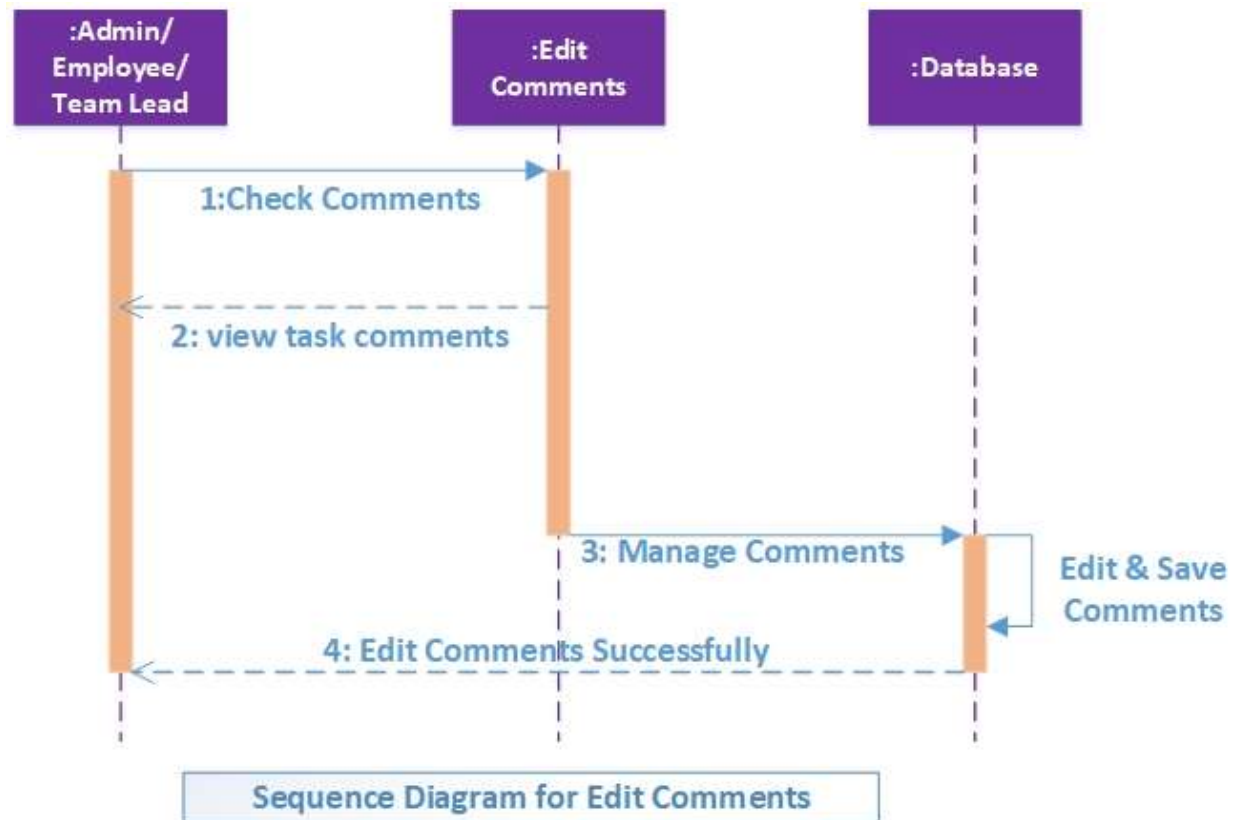


Figure 4.21: Sequence Diagram for Edit Comments

4.3 Class Diagram

Here is the class Diagram of our FYP project:

Chapter 5: Implementation

This chapter will discuss implementation details supported by UML diagrams of Intelligent Project Alliance.

5.1. Component Diagram

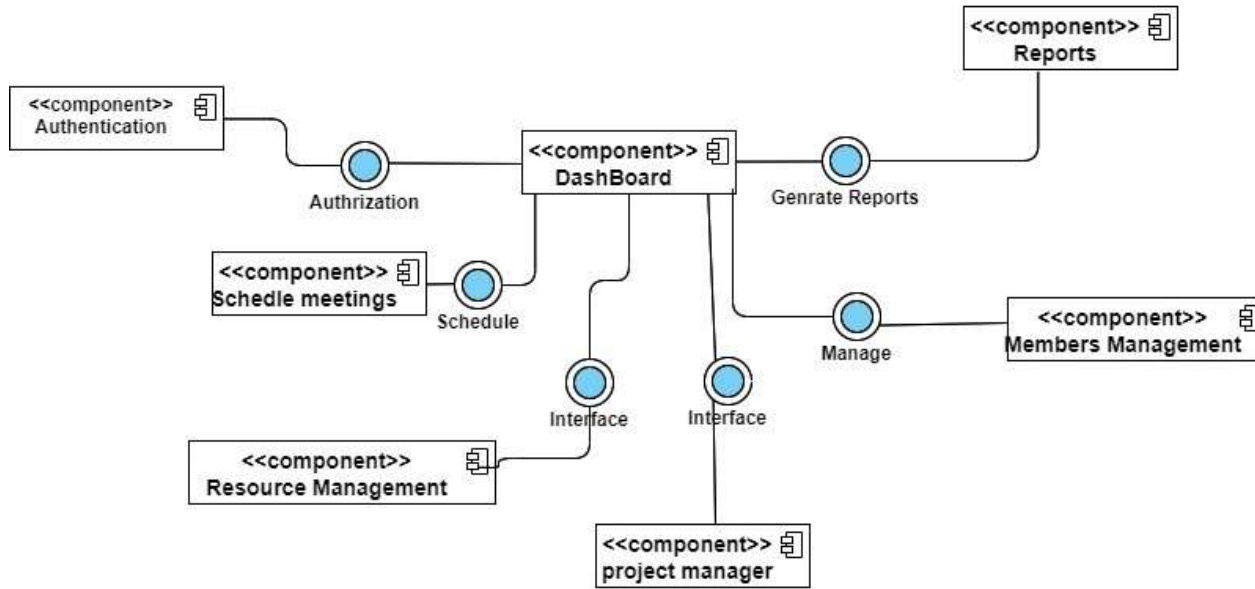


Figure 5.1: Component Diagram

Explanation:

In component diagram it has been shown that the admin register and manage the dashboard. They can check Schedule Meetings, Manage Members, and then generate reports in an efficient way.

5.2. User Interface

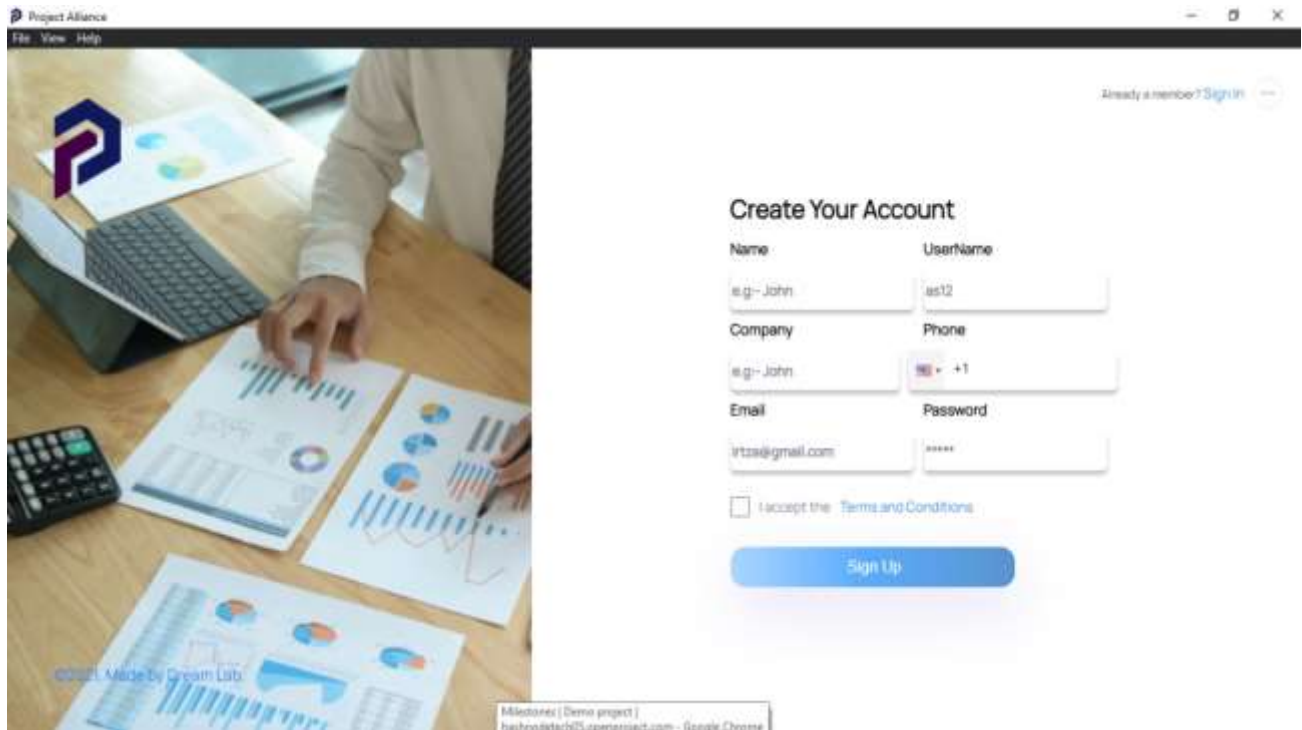
Splash Screen



Figure 5.2: Splash Screen

This is a splash screen, and it will be shown at the start of the application for 3 seconds.

Signup



Project Alliance

File View Help

Already a member? [Sign In](#)

Create Your Account

Name	Username
<input type="text" value="e.g.- John"/>	<input type="text" value="as12"/>
Company	Phone
<input type="text" value="e.g.- John"/>	<input type="text" value="+1"/>
Email	Password
<input type="text" value="jtsa@gmail.com"/>	<input type="password" value=""/>

☐ I accept the [Terms and Conditions](#)

[Sign Up](#)

Milestones (Demo project)
hashrodetach@supertproject.com - Google Chrome

Figure 5.3: Sign Up

The user will enter the data for account creation. The validations applied on Signup page are given below:

1. Email should be valid.
2. Username should contain number and alphabets.
3. Name should only contain alphabets.
4. CNIC should be valid.
5. Password must be more than 5 digits.

After checking the given validations and when user press “Signup” button, the user account will be created successfully.

At login page the user will enter his valid email and password. If the email or password is invalid/empty the user will not be able to Log In to his/her account, and if the user does not have any account, then user should first create his/her account on clicking “Sign up” button.

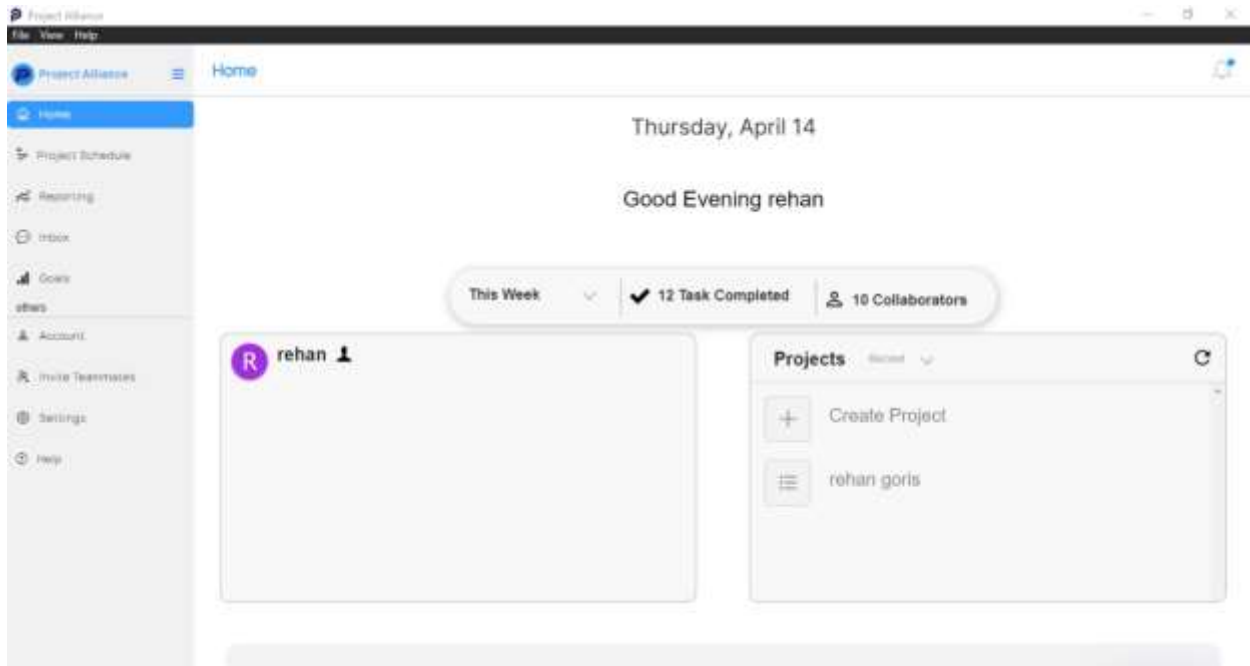


Figure 5.4: Dashboard

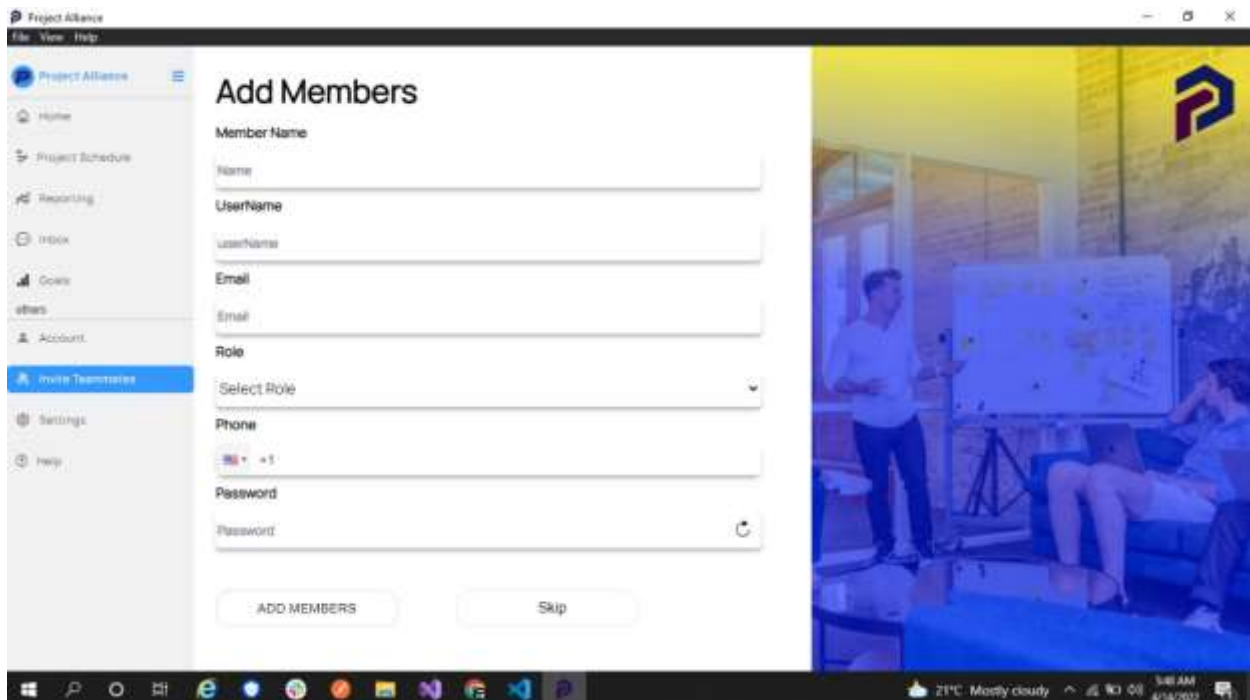


Figure 5.5: Add Member

In Add Member page the user will enter his member's name, username and valid email and then enter the role of that member. If the credentials are invalid/empty the user will not be able to add members over there , if they don't want then they will be able to skip that step.

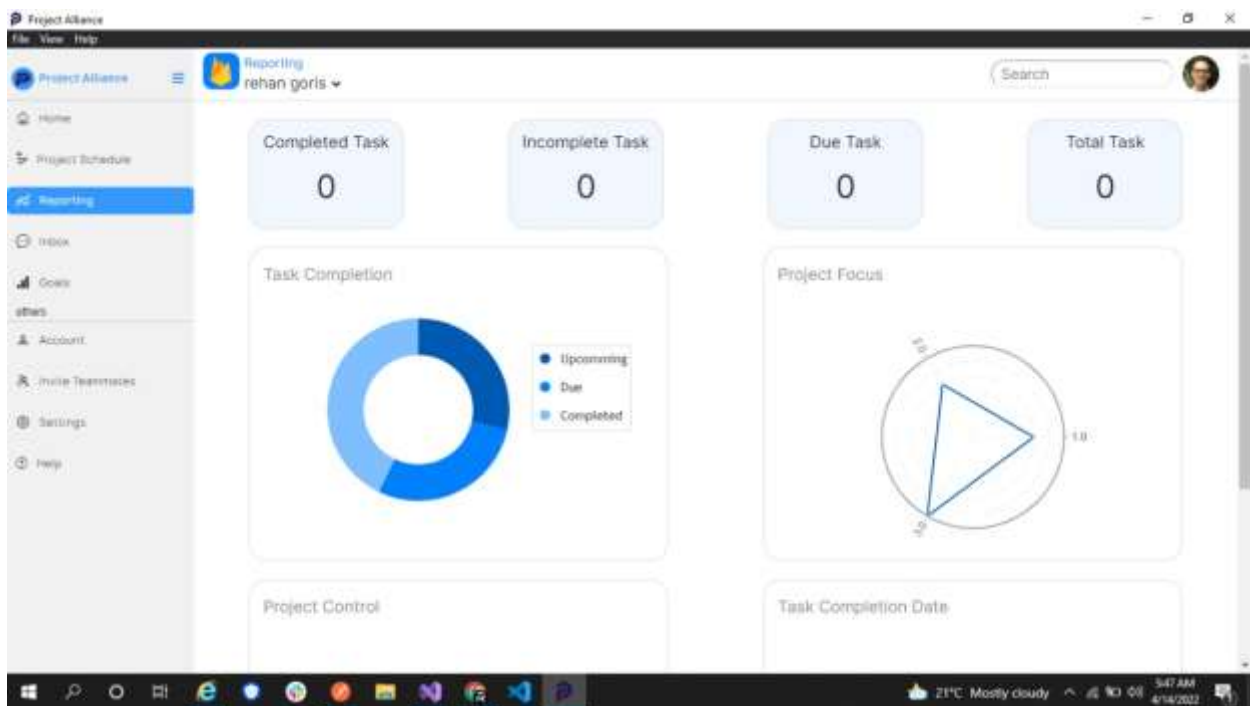


Figure 5.6:Reporting Screen

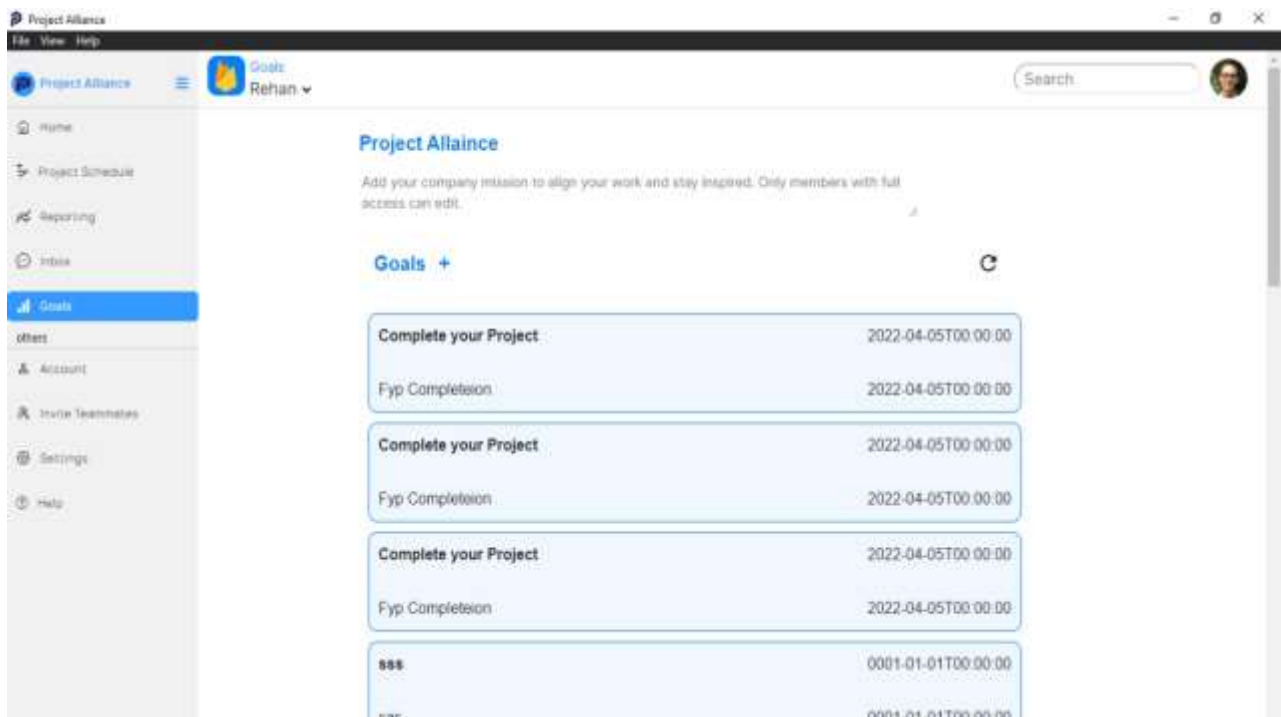


Figure 5.7: Goal Management

At this page the user will check the project goals according to his/her need. The records related to slot are given at this interface which helps the user to get know how many projects will be completed.

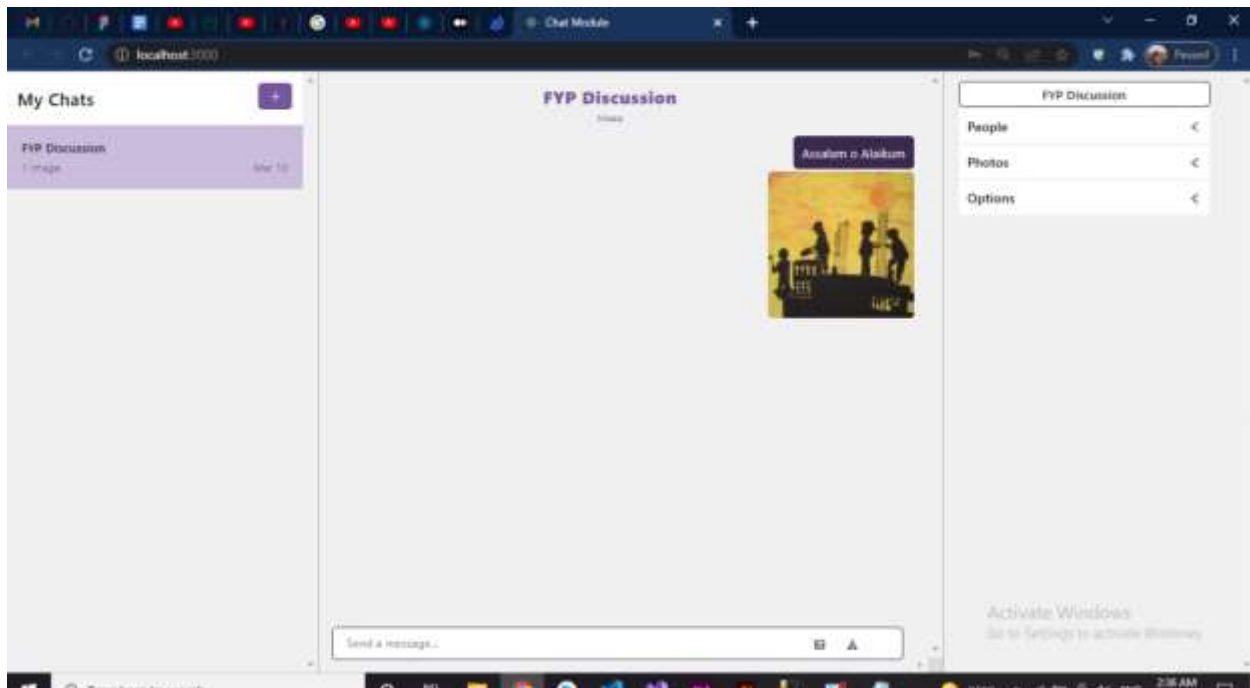


Figure 5.8: Chat Module

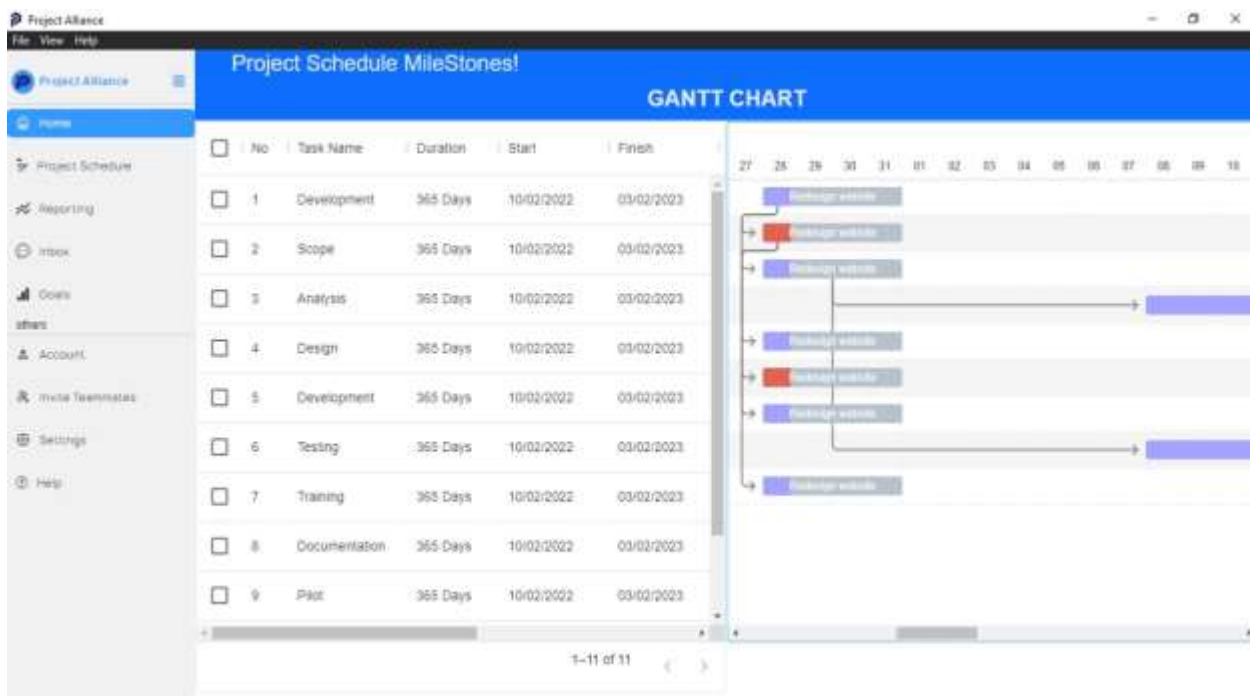


Figure 5.9: Project Schedule

Gantt-Chart are given at this interface which helps the user to get know the project schedule. Gantt-Chart is basically visualizing schedule, that monitoring specific tasks and resources in the project.

Project Schedule help teams to plan work around deadlines and properly allocate resources. They depict, among other things, the relationship between the **start and end dates** of the tasks, milestones, and dependent tasks.

The main benefit of Gant-Chart is easy to understand, clear and visual representation of time frames. An easy to add dependencies and predecessors, creatively showing project plans.

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 desktop application.
- Our system gives error message when user enter incorrect fields.
- Our system successfully shows the record in database of registered users.

1. Sign up

Tested By	Azeem Sarwar
Test type	Unit testing
Test case number	01
Test case name	Sign up
Test case description	This test case will check sign up form details
Procedural Steps	Procedural Steps
1	User will fill sign up form
2	All information must be entered correct
3	User account will be created

Sign up form

2. Login

Tested By

Procedural Steps	Procedural Steps
1	User will enter email and press send button
2	A dialog box appears confirming that the user wants to reset password or not
3	On selecting yes password will be reset

Forgot Password

4. Project Status

Tested By	Rehan Asghar
Test type	Unit testing
Test case number	04
Test case name	Project Status
Test case description	This test case will check Project status details
Procedural Steps	Procedural Steps
1	User after login will move towards Project Status page where he can see all the details.
2	On clicking status button application will show To-do, In-Progress, Completed project detail.
3	If user click to check project description, then popup message displayed.

Project Status

5. Admin

Tested By	Azeem Sarwar
Test type	Unit testing
Test case number	05
Test case name	Admin
Test case description	This test case will check admin details
Procedural Steps	Procedural Steps
1	Admin can check and search user details
2	Details can be searched by name and email

Admin page

6.1.2. 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., Scope and Cost module.

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 Schedule module that you should add some other ways to check through Gantt Chart.

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 Scope and Goal 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 Chat 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 user. 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.

6.9. Evaluation:

Evaluation was done as we completed our project by correctly evaluated things and completed them on time.

6.10. Deployment:

Our project deploys in real time environment, and it works accurate. There is proper exchange of information between system and application, hence deployment was also done.

6.11. Maintenance:

Maintenance was done as we fulfilled all the objectives of the project and maintained all the detection and correction of errors in the system.

1.7. References:

Link and Reference are as under:

[1]:<https://openProject.com/>

[2]:<https://www.atlassian.com/software/jira>