

**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*Sr. 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 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 **“Sr.Zeeshan Javed”** and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Software Engineering.

**Sr.Zeeshan Javed**

**Supervisor**

---------------------------------------

**External Examiner (If any)**

---------------------------------------

**Administrator UIIT**

**Executive Summary**

To effectively manage everyday tasks, managers need an application that makes it easy for them to do so. However, platforms are not providing the management facilities to developers. Thousands of developers visit multiple platforms on a daily basis for managing their tasks and daily routine. 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 developers face on a daily basis and this area needs to be explored till today. So, we have come up with a **solution** for developers and those people who want to set their daily routine according to time and find their solution on a single application. With this platform, all modules will be well organized for all type of people to reach. Activities like **Schedule Management**, **Scope Management**, **Requirements Management** and **Design Management** will be managed by us. It is a desktop application to provide the complete project management facilities to users. It is difficult for users to understand the complex tool, so they can use this application with ease. The user will select the project and manage their tasks according to their needs. After the setup task they can estimate the cost of their project. All the schedule management will be in a design **view** that will help its user to understand this in visual form. The administration can also go through all the tasks and check their employee daily routines. The **Task Management** feature will be well organized and prioritized, allowing you to own and **organize tasks**, assign them to team members, etc. Everyone is informed about the progress and upcoming steps. Organize work portfolios, **automate reports**, **keep track of comments**, **create user accounts**, create administrator accounts, manage requirements and manage designs. We have a category having Views & Dashboards, Work Packages, Automatic Reports, Comments and History, User Account, Administrator Account, Search Task, Requirement Management, Design Management.

**Cost Management** will be handled by **Time Estimation**, **Labor Cost**, **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.

In this way, through this **hub**, all tasks and issues will be placed in an organized way. This will be a great platform that will be very useful. Thus, we are developing an application that is **“Intelligent Project Alliance”** to help managers do so. Here we would like to create an application that provides them with the ability to manage these tasks on a daily, weekly or monthly basis.

We are developing an application that will be accessible offline as well. All functionalities will be built in comparison to other management tools but in offline mode in desktop App and it's all user interface will be organized in a simple way.

**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 **“Sr.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

**Abbreviations**

| **SRS** | Software Requirement Specification |
| --- | --- |
| **PC** | Personal Computer |
|  |  |
|  |  |
|  |  |

**Table of Contents**

**Introduction 1**

[1.1 Brief](#_heading=h.1ksv4uv) 9

[1.2 Relevance to Course Modules](#_heading=h.44sinio) 9

[1.3 Project Background](#_heading=h.2jxsxqh) 10

[1.4 Literature Review](#_heading=h.2jxsxqh) 11

[1.5 Methodology and Software Life Cycle](#_heading=h.z337ya) 11

**Problem Definition** 12

[2.1 Purpose](#_heading=h.3j2qqm3) 13

[2.2 Product Functions](#_heading=h.1y810tw) 13

[2.3 Proposed Architecture](#_heading=h.4i7ojhp) 14

[2.4 Project Deliverables](#_heading=h.2xcytpi) 14

[2.5 Operating Environment](#_heading=h.1ci93xb) 15

[2.6 Assumptions and Dependencies](#_heading=h.3whwml4) 16

**Requirement Analysis** 17

[3.1 Functional Requirements](#_heading=h.2bn6wsx) 18

[3.2 Non – Functional Requirements](#_heading=h.qsh70q) 19

[3.3 Use case Model](#_heading=h.2bn6wsx) 20

[3.3.1 Use Case Diag](#_heading=h.3as4poj)ram 20

[3.3.2 Actors D](#_heading=h.3as4poj)escription 20

[3.3.3 Use Case](#_heading=h.3as4poj) Description 20

**References** 21

**Table of Figures**

[Figure 1: Agile LifeCycle 13](#_heading=h.3rdcrjn)

# Chapter 1: Introduction

# 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 developers and management communities can use to set up their routine tasks on a daily, weekly, or monthly basis. This application will help them to estimate their project cost according to their schedule and work packages.

Intelligent Project Alliance will assist an individual or team in organizing and managing their projects and tasks effectively. Every individual can even use it for free.All the modules will be available without any cost.It tools 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, organize dashboards, and proof or approve changes.

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,Gantt charts.

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.

# 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.

# Project Background

Basically, According to our research about this problem, 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.

# Literature Review

In large organizations or new start-ups, it is necessary for us to organize, assign and manage tasks and 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, Issue Management and Project Collaboration. It is available for websites not for desktop and mobile apps. In **Open-project** users 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. Although all the functionalities are built are paid and not accessible after a trial version.[[1]](https://openproject.com/)

**Open Projects** provides collaborative communities with foundation-level support.

**Core services:**

* Community management support
* Collaboration tool set GitHub org, mailing group, calendar, roster, ballot, asynchronous chat, video meeting.
* Management of deliverables
* Set-up or transition of project assets such as repositories, code base, website, group communication channels, etc.
* Ongoing support for community-led marketing activities
* Outreach to secure more developers, sponsors, and adopters
* Free public access to your work in perpetuity

The association aims at the following objectives:

* Establishing and promoting an active and open community of developers, users, and companies for continuously developing the open-source project collaboration software **OpenProject**
* Defining and developing the project vision, the code of conduct, and principles of the application
* Creating development policies and ensuring their compliance
* Defining and evolving the development and quality assurance processes
* Providing the source code to the public
* Providing and operating the **OpenProject** platform.

# 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 versions.These tools help the user to manage their rotunies.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.

# 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 developers to entertain with this hub.

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.

Graphical user interface, application

Description automatically generated

**Figure 1: Agile LifeCycle**

* + 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**

# Purpose:

The aim of your gaze determines what you see. Now, at present there are only paid project management tools available, the main purpose would be to develop a system that can help managers to manage their project and workforce(employees). It is difficult for managers to understand the complex tool, so they can use this application with ease. The user will select the project and manage their tasks according to their needs. Managers can also estimate the cost of the project and also assign resources to different modules or tasks. The task, cost and work progress report also be maintained in this system. Manager or User can schedule his project and analyze them in gantt-chart and kanban board.The admin can also go through all the tasks and check their employee’s daily routines.

Open-project is a project management tool. It is available on the web but it is paid.open-project is available for linux but its configuration is not too easy. Every user needs a certification and full training to use it. And customization is also not permitted.

That’s why we are developing an application that will be accessible online as well offline. Intelligent project alliance will also provide the facility to generate reports automatically. Our application user interfacewill beuser-friendly, straightforward, providingquickaccesstocommonfeatures. Every user will be able to use it according to their requirements. We will provide every feature freeofcost and this project will be an open-source project.Open-source community can also make enhancements in it so that it can fulfill the needs of industry.

We will bring them this kind of application in which management communities can make their routines in an organized way. All the cost management, design management, requirement management modules will be displayed in different ways in which they can get their solution with ease.

# Product Functions:

Product Functionality is listed below:

* **Authentication**
* **Create new project**
* **Invite Members**
* **Remove members**
* **create project schedule**
* **Create project tasks and subtask**
* **Manage Tasks**
* **Cost Estimation**
* **Reports generation**

**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 and development requirements:

**2.4.1 Projects Modules:**

In this project the deliverables will consist of the input as equipment (i.e., 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 “Intelligent Project Alliance”). In this case product deliverables are the completed parts or modules of the project. Input will be software-based components. The project is divided into different modules and each module is a major milestone in the project.

1. **Authentication**
2. **Task Management:**

It will be a well organized and prioritized feature in which you can organize tasks and assign tasks to your team members.  All the tasks and communication in one place. Everyone is up to date about the progress and next steps.The following are the features that will be in task management:

* Overview & Dashboard
* Work Packages
* Tasks List
* Notifications
* Schedule
* Kanban board
* List of Task
* Comments and History

1. **Report Generate**
2. **Cost Management:**

* Time Estimation
* Labor Cost
* Cost Estimation
* Budgeting
* Cost Reporting

1. **Members Management:**

* Invite Member
* Assigning Tasks
* Track Routines

1. **Maintain Backlog Of Project:**

* Previous Record
* Track Changes

1. **Meetings Schedule:**
2. **Project Settings:**

**2.4.2 Software Requirements specification Document:**

The document contains all the information about the project and having functional and non- functional requirements.

**2.5. Operating Environment:**

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

**Operating System**: Window 10, Linux and MAC Operating system

**Database**: Asp.net & MySQl

**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 is dependent on some desktop device.

# 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.

# 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 task of users.
* Admin can create new projects, add new members, manage projects according to their requirements.
* Moderators can also work as an admin. Moderator will monitor the projects that will be assigned to teams. Moderator will lead individually.
* The system will confirm the user to Create your Organization.
* The system will allow the admin to define new users, add new users or handle different tasks between them.
* The system will manage the data into the database.

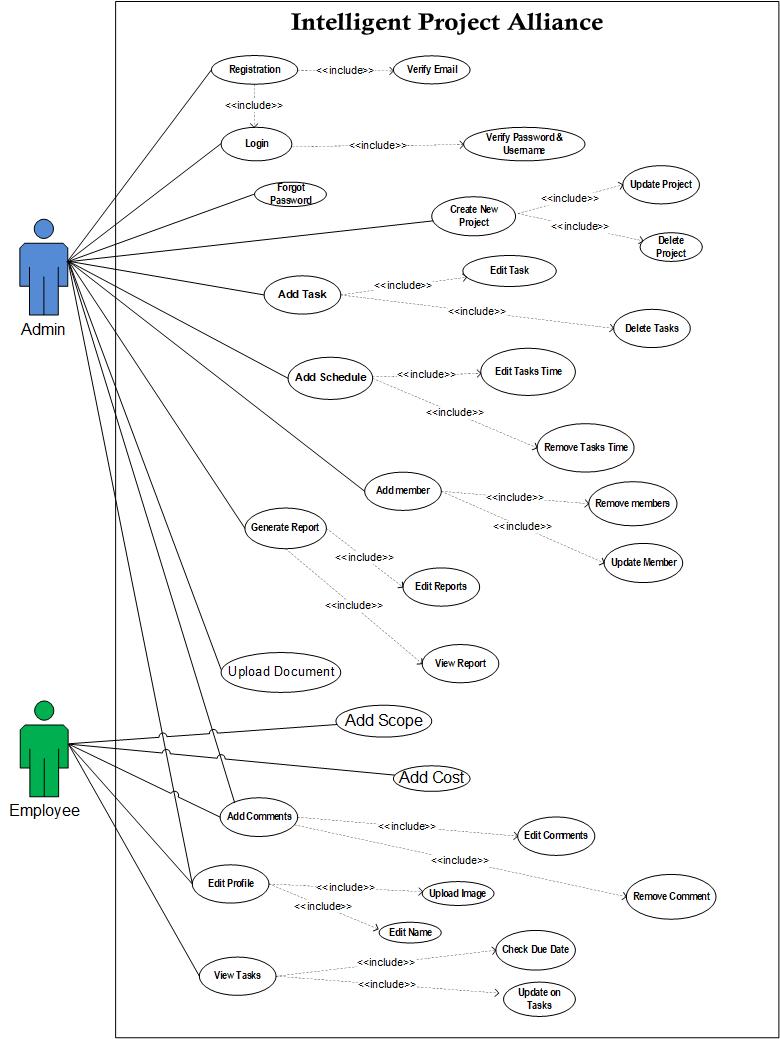
# Non-Functional Requirements

* 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.
* The system should be available for 24/7 of the time to handle the concurrent request of the users.
* The system should permit only authorized users to ensure its security.
* Interface and the system itself should be user friendly so that the user will feel it is easy to use.
* The system will authenticate the user by verifying the credentials to the database.

# 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:**

****

* + 1. **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 by itself.

* **Employee:**

Employees will be those people who will perform assigned tasks that are assigned by the admin. They will be able to view and submit project tasks and they will be able to add Profile Info.

* + 1. **Use Case Description:**

| **Use Case ID:** | ID-01 |
| --- | --- |
| **Use Case Name:** | Registration |
| **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:** | * Credentials will be entered by users * Credentials will be authenticated from database of the system * User will be successfully login |
| **Alternative Flows:** | * 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 3.1: Register**

| **Use Case ID:** | ID-02 |
| --- | --- |
| **Use Case Name:** | Login |
| **Actors:** | Admin |
| **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. |
| **Pre-conditions:** | Users should be authenticating.Users should have a desktop device. |
| **Post-conditions:** | The user can use the app according to their needs.. |
| **Normal Flow:** | * User will first gave their own registered userName and   Password.   * System will verify it if the details are correct then the user will login successfully. |
| **Alternative Flows:** | * 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:** | If a user wants to manage their tasks then the user must fulfill the credentials. |

**Table 3.2: Login**

| **Use Case ID:** | ID-03 |
| --- | --- |
| **Use Case Name:** | Forgot Password |
| **Actors:** | Admin |
| **Description:** | If the user forgot his password, He/She will be able to forget his/her password. |
| **Trigger:** | When individuals forgot a password in their system. |
| **Pre-conditions:** | Users should register their account.Then if they forgot their password they can get it.. |
| **Post-conditions:** | The user will get their password back.Then he will be able to use the system.. |
| **Normal Flow:** | * The user will use enter password login. * If he got the wrong password then he will go for the forgotten password. * Then he will give an email and the system will verify then will be able to rewrite his new password. * After getting a new password, he will login in the system with the new password. |
| **Alternative Flows:** | * Users can not forget their password without registering an account. * If the new password is incorrect again,,User can again forget. |
| **Notes and Issues:** | If a user wants to forget he/she must be registered first, then will be able to use it. |

**Table 3.3: Forgot Password**

| **Use Case ID:** | ID-04 |
| --- | --- |
| **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:** | * 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:** | * Users will be those people who will perform assigned tasks that are mentioned by the moderator. * They will follow the project tasks time limit and will give feedback back to the moderator. |
| **Alternative Flow:** | Users can add the existing project in the system. |
| **Assumptions:** | If the same name of the project exists then the user uses another name. |

**Table 3.4: Create New Project**

| **Use Case ID:** | ID-04 |
| --- | --- |
| **Use Case Name:** | Add Task |
| **Actors:** | Admin |
| **Description:** | Admin will assign tasks to every individual and specific users. |
| **Trigger:** | When a user clicks on an add task it will show the members and then he will assign tasks to them.. |
| **Preconditions:** | User must create a project and then he will assign tasks to them. |
| **Post-conditions:** | Admin can edit the tasks that are assigned to the people.And  He will also delete the tasks that will be assigned to the project. |
| **Normal Flow:** | * Admin will create the project * Assign tasks to every individual. * Admin can edit project * Admin can delete the tasks |
| **Alternative Flows:** | If the Admin wants to assign tasks then he should create a project first then he will assign tasks to individuals. |
| **Assumptions:** | If the user will not create a project then tasks will not be assigned to them. |

**Table 3.5: Add Tasks**

| **Use Case ID:** | ID-06 |
| --- | --- |
| **Use Case Name:** | Add Schedule |
| **Actors:** | Admin |
| **Description:** | Admin will add the schedule for the team member. And he will also give them specific time according to every task. |
| **Trigger:** | Admin will add time according to time.. |
| **Preconditions:** | Project should be created.Then the user will add the members in the project. |
| **Post-conditions:** | Admin will edit the project schedule and monitor them according to their task. |
| **Normal Flow:** | * Admin will add the member in the project * Admin will assign tasks to the members * Then he will assign the task limit. |
| **Alternative Flows:** | * If the time limit exceeds the given time, the admin can increase the time limit of the task. |
| **Notes and Issues:** | Users will be notified according to the time. |

**Table 3.6: Add Schedule**

| **Use Case ID:** | ID-07 |
| --- | --- |
| **Use Case Name:** | Add Comments |
| **Actors:** | Admin, Employee |
| **Description:** | Amin, employee and moderator 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:** | * Admin will be able to add new comments over there. * Employee and Moderator will also have an ability to add new comments |
| **Alternative Flows:** | * If they want to change their comments then they can also change comments over there. * It will be handled by the database through admin. |
| **Notes and Issues:** | Comments will remain through the projects. |

**Table 3.7: Add Comments**

| **Use Case ID:** | ID-09 |
| --- | --- |
| **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:** | * Admin will check the projects * Admin will check the required changes * Then admin will make the changes in the project |
| **Alternative Flows:** | * 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 3.9: Update Project**

| **Use Case ID:** | ID-10 |
| --- | --- |
| **Use Case Name:** | Add Schedule |
| **Actors:** | Admin |
| **Description:** | Admin will manage the tasks and edit project schedules and also manage their access in the project. |
| **Trigger:** | When an admin wants to add a project schedule. |
| **Preconditions:** | Project must be created before updating it. |
| **Post-conditions:** | Admin can successfully edit project schedules. |
| **Normal Flow:** | * Admin will click on the update button for editing |
| **Alternative Flows:** | * If there is a problem regarding editing, the admin will have a chance to change the time limit of the project. |
| **Notes and Issues:** | None. |

**Table 3.10: Add Schedule**

| **Use Case ID:** | ID-11 |
| --- | --- |
| **Use Case Name:** | Add Members |
| **Actors:** | Admin |
| **Description:** | As admin will be able to manage different tasks and handle new added projects so admin will also add new accounts. |
| **Trigger:** | When the user clicks on login then the actor will gave them  account for signing in the project. |
| **Preconditions:** | Admin should enter members' accounts inthe system.. |
| **Post-conditions:** | When User will login successfully then a new account will be added to the database. |
| **Normal Flow:** | * First admin will add the project the project Credentials * Credentials will be authenticated from database of the system * User will successfully login and then new account data will be added into the database. |
| **Alternative Flows:** | * If user is not already registered, then user will first sign up * User will then provide credentials and then a new account will be generated. |
| **Exceptions:** | Admin act as Primary actor that will access to store the data. |
| **Special Requirements:** | None |
| **Assumptions:** | None |
| **Notes and Issues:** | Users will have only 3 attempts for login. After 3 attempts the user will have to wait for 30 seconds to retry. |

**Table 3.11: Add Members**

| **Use Case ID:** | ID-12 |
| --- | --- |
| **Use Case Name:** | Adding Tasks Cost |
| **Actors:** | Admin |
| **Description:** | Admin will also the cost of the task that is assigned to the employee. |
| **Trigger:** | Admin will add it individually. |
| **Preconditions:** | Project must be created and assigned to the user.. |
| **Post-conditions:** | Admin will accurately estimate the project cost.. |
| **Normal Flow:** | * Admin will add the tasks in the project. * Then create the task list * After admin will add the cost of tasks individually |
| **Alternative Flows:** | * Admin can also edit or delete different documents. |
| **Notes and Issues:** | None |

**Table 3.12: Adding Tasks Cost**

| **Use Case ID:** | ID-13 |
| --- | --- |
| **Use Case Name:** | Edit Name |
| **Actors:** | Admin,Employee |
| **Description:** | All the individuals in the project will have an ability to edit their profile info. |
| **Trigger:** | When admin,moderator and employee when they want to change profile info can click on profile info. |
| **Preconditions:** | They must have an account over there to edit info. |
| **Post-conditions:** | Admin will easily edit their profile. |
| **Normal Flow:** | * . |
| **Alternative Flows:** | * If an actor wants to edit profile info then they easily edit from there. * Data will be managed by database. |
| **Exceptions:** | Databases act as Secondary actors that will access to store or edit the data. |
| **Special Requirements:** | None |
| **Assumptions:** | None |
| **Notes and Issues:** | None |

**Table 3.13: Edit Name**

| **Use Case ID:** | ID-14 |
| --- | --- |
| **Use Case Name:** | Adding Cost per Employee |
| **Actors:** | Admin |
| **Description:** | Admin will also the cost of the employee that will be in project |
| **Trigger:** | Admin will add it individually. |
| **Preconditions:** | Project must be created and assigned to the user.. |
| **Post-conditions:** | Admin will accurately estimate the employee cost.. |
| **Normal Flow:** | * Admin will add the tasks in the project. * Then create the task list * After admin will add the cost of members individually |
| **Alternative Flows:** | * Admin can also edit or delete different documents. |
| **Notes and Issues:** | None |

**Table 3.13: Adding Cost Per Employee**

| **Use Case ID:** | ID-15 |
| --- | --- |
| **Use Case Name:** | Edit Tasks |
| **Actors:** | Admin |
| **Description:** | Admin will be able to view tasks after checking projects due dates. |
| **Trigger:** | When the actor clicks on view tasks. |
| **Preconditions:** | Due dates must be mentioned before viewing their tasks. |
| **Post-conditions:** | Tasks will be viewed easily by employees. |
| **Normal Flow:** | * Tasks will be viewed after checking due dates. |
| **Alternative Flows:** | * Every employee will be easily viewed and their tasks. * And also be able to assign dates according to their schedule. |
| **Notes and Issues:** | Actors viewing their tasks. |

**Table 3.15: Edit Tasks**

| **Use Case ID:** | ID-16 |
| --- | --- |
| **Use Case Name:** | Update on Tasks |
| **Actors:** | Employee |
| **Description:** | The actor 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:** | Actor will submit tasks successfully. |
| **Normal Flow:** | * Actors will easily submit those tasks that can be resolved. |
| **Alternative Flows:** | * Actors can submit different tasks before resolving them. |
| **Notes and Issues:** | Actors must complete their tasks before submitting. |

**Table 3.16: Update on Tasks**

| **Use Case ID:** | ID-17 |
| --- | --- |
| **Use Case Name:** | Edit Profile |
| **Actors:** | Employee,Admin |
| **Description:** | Actors will also be able to add profile info by editing their name or bio and also by uploading new images. |
| **Trigger:** | When a user clicks on Add Profile Info. |
| **Preconditions:** | Actors must have an account for editing over there. |
| **Post-conditions:** | Actor will Add Profile Info successfully. |
| **Normal Flow:** | * Actors can easily edit their profile info after creating their account over there. |
| **Alternative Flows:** | * If the user is not already registered, then the user will first sign up. * After that actors will be able to edit their name, bio and also be able to upload new images in their profile. |
| **Notes and Issues:** | Users will have an account over there for adding profile info. |

**Table 3.17: Edit Profile**

| **Use Case ID:** | ID-20 |
| --- | --- |
| **Use Case Name:** | Generate Report |
| **Actors:** | Amdin |
| **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:** | * Admin will click on the button to generate a report. * All the taks and the cost will be in creating in xml and pdf form. |
| **Alternative Flows:** | * If the report did not generate, then the admin must see they have any data to generate. |
| **Notes and Issues:** | None |

**Table 3.20: Generate Report**

| **Use Case ID:** | ID-21 |
| --- | --- |
| **Use Case Name:** | Edit Reports |
| **Actors:** | Amdin |
| **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:** | * 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:** | * If system did not have any data in the system then no report will be shown for editing |
| **Notes and Issues:** | None |

**Table 3.21: EditReport**

| **Use Case ID:** | ID-22 |
| --- | --- |
| **Use Case Name:** | ViewReports |
| **Actors:** | Amdin |
| **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:** | * Admin will check the availability of the report. * When the report is available then the admin can view it. |
| **Alternative Flows:** | * Reports must be available in the system. |
| **Notes and Issues:** | None |

**Table 3.22: EditReport**

# 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.

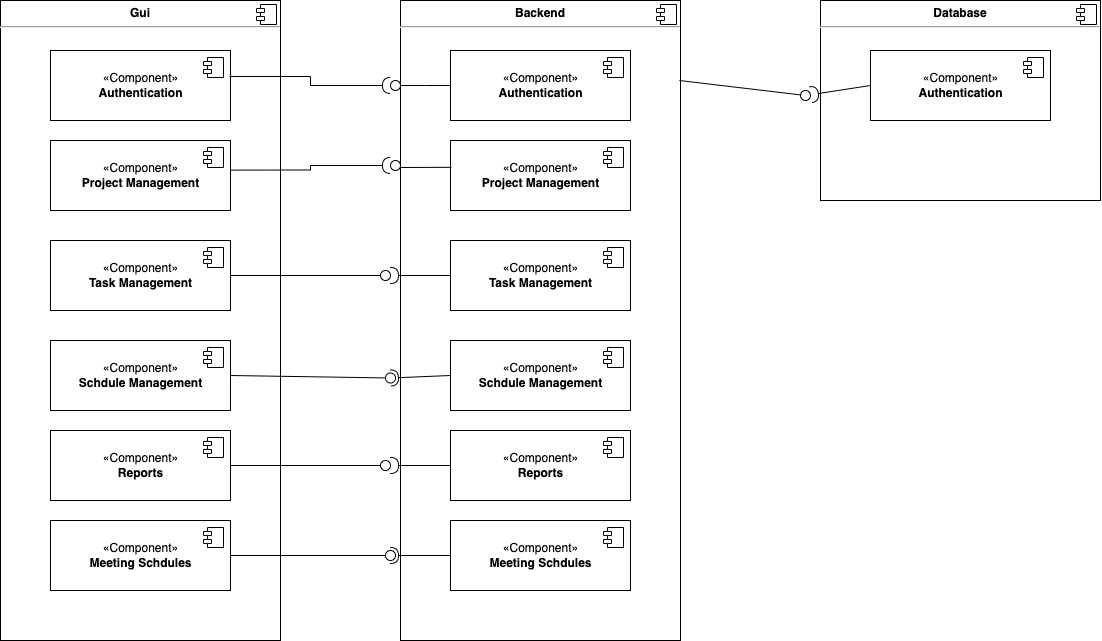
**4.2. System Design**

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.

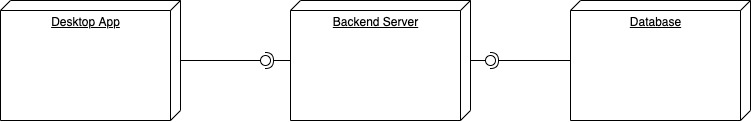
**4.2.1 UML Structural Diagrams**

Following are the UML structural diagrams of our system:

**4.2.1.1 Component Diagram**

****

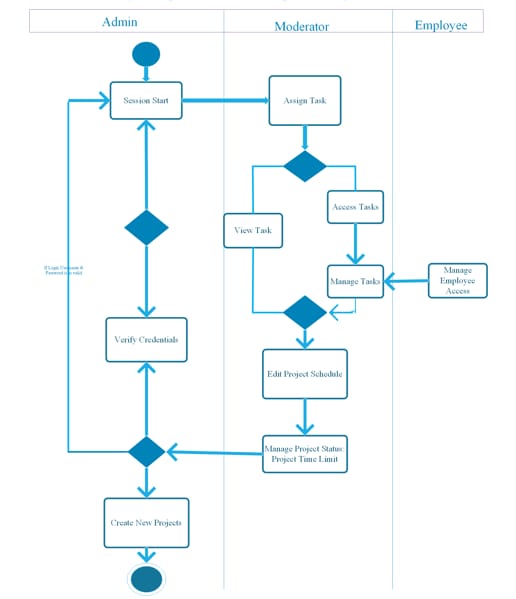
**4.2.1.2 Deployment Diagram:**

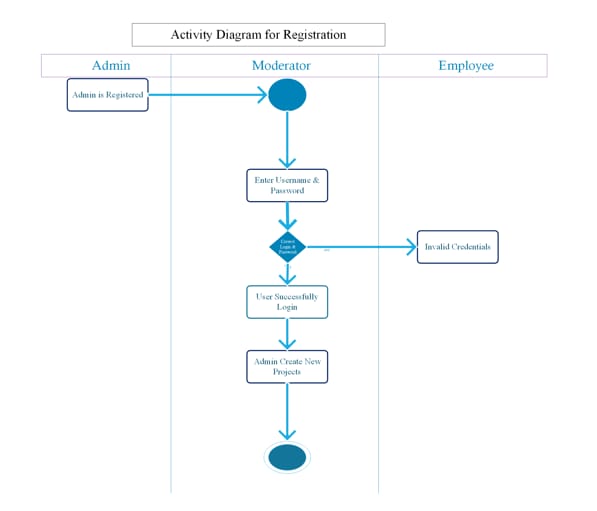
****

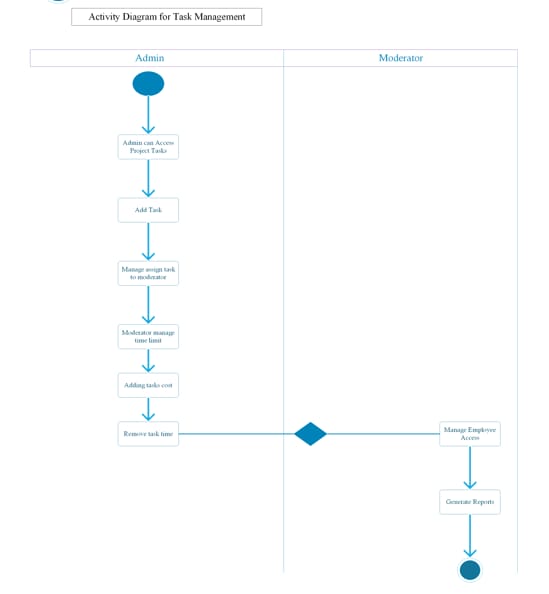
**4.2.2 UML Behavioral Diagrams**

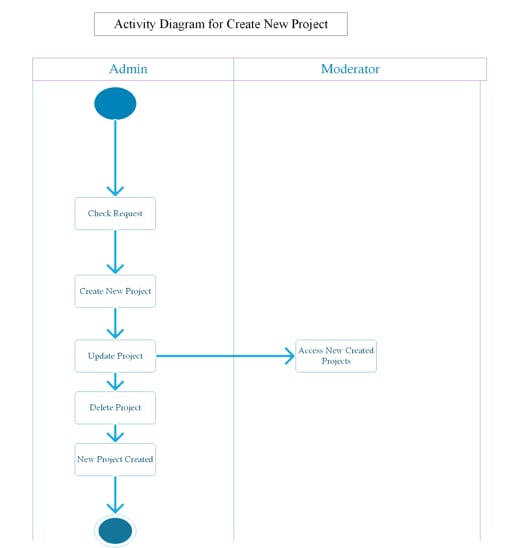
Following are the behavioral diagrams of our system:

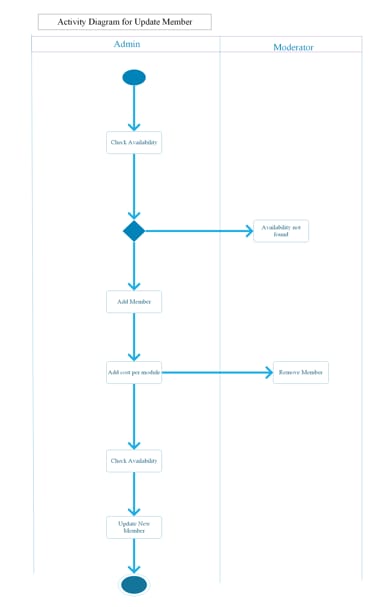
**4.2.2.1 Activity Diagram**

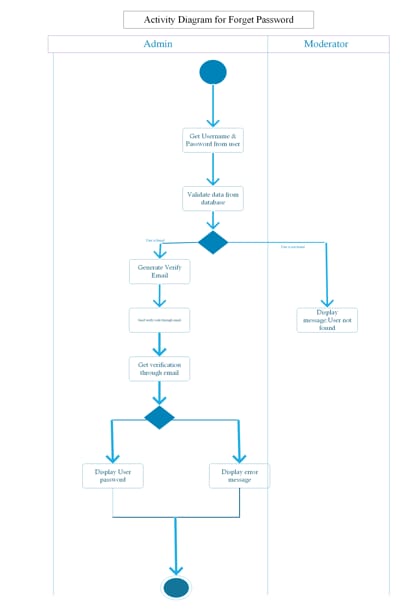
****

****

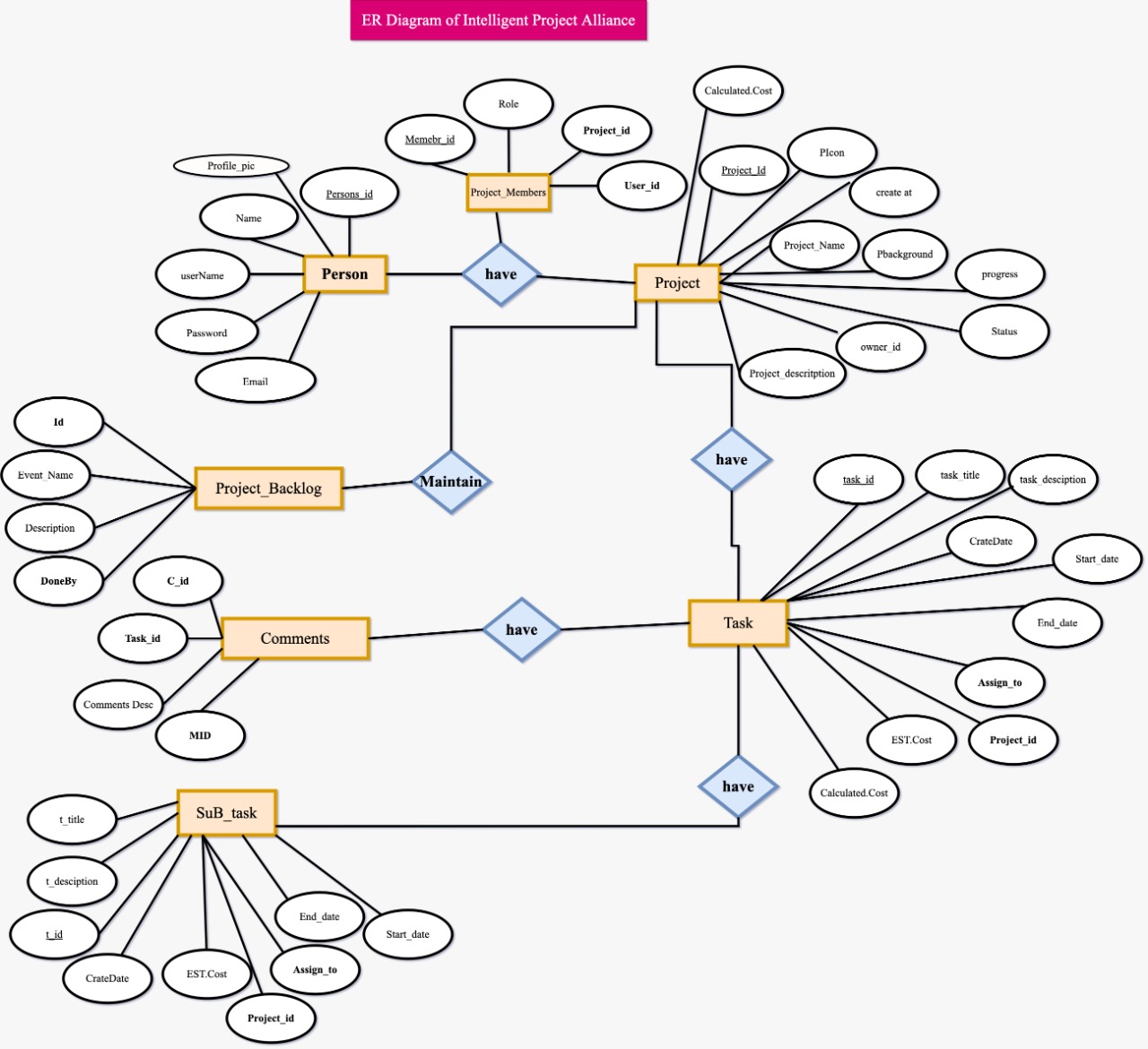
****

****

****

****

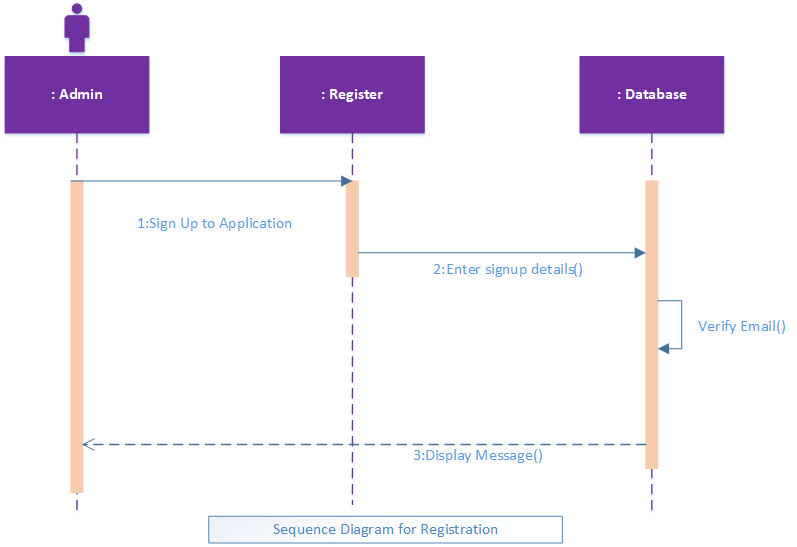
**4.2.2.2 ER Diagram**

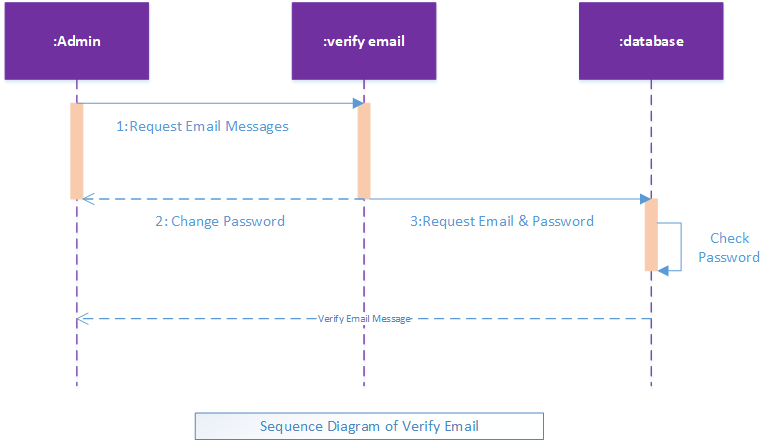
****

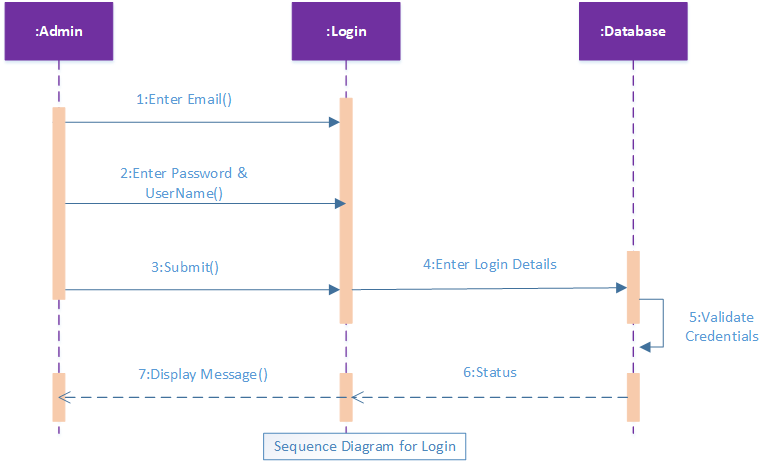
**4.2.3 UML Interaction Diagrams**

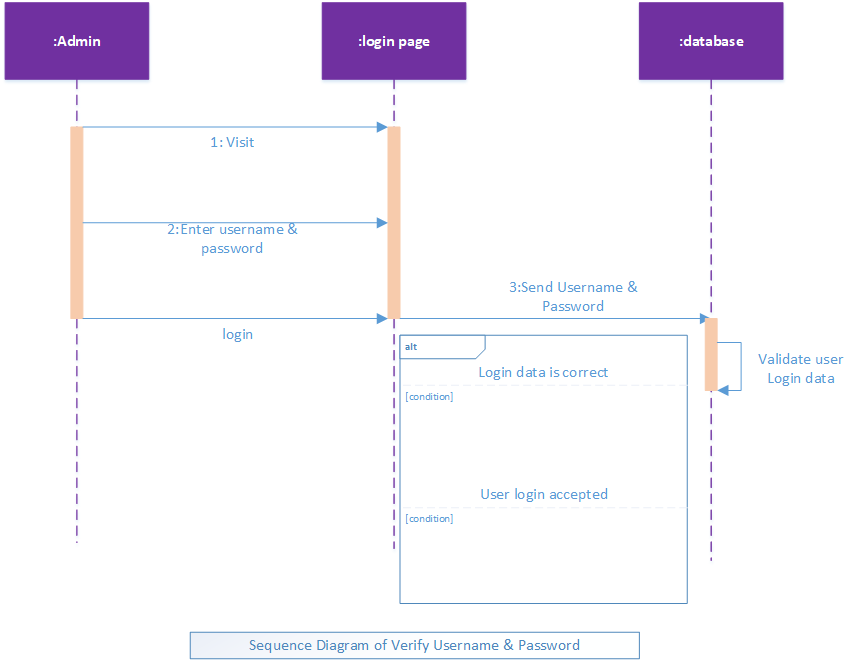
Following are the UML interaction diagrams of our system:

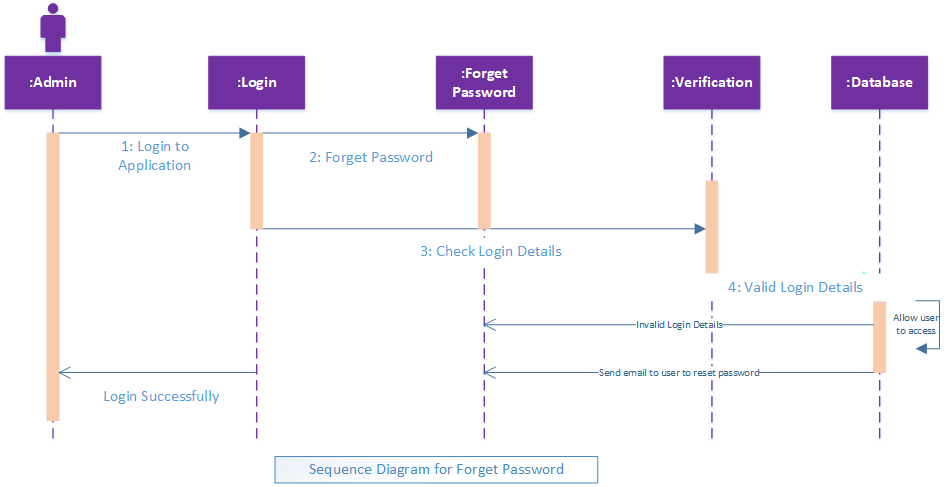
**4.2.2 Sequence Diagrams**

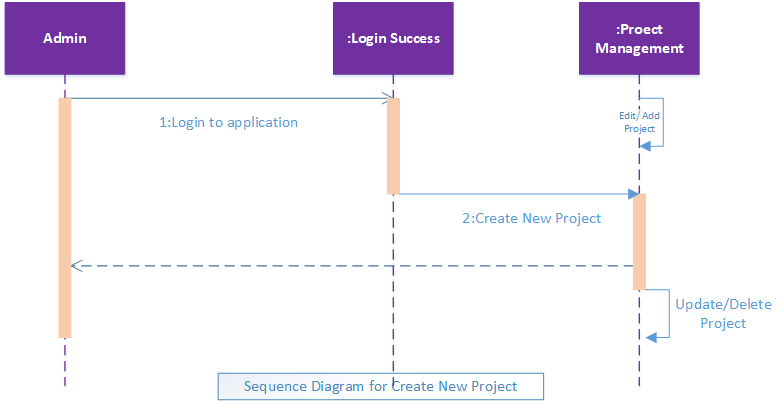
****

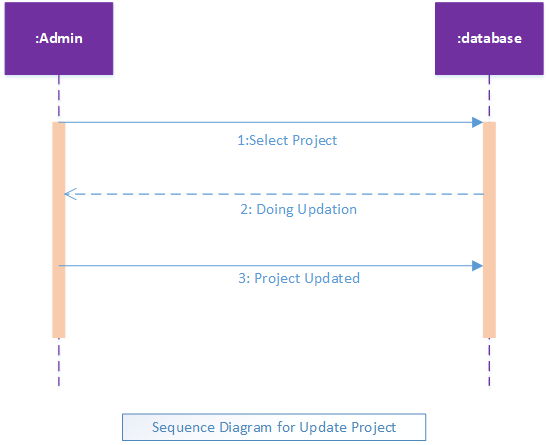
****

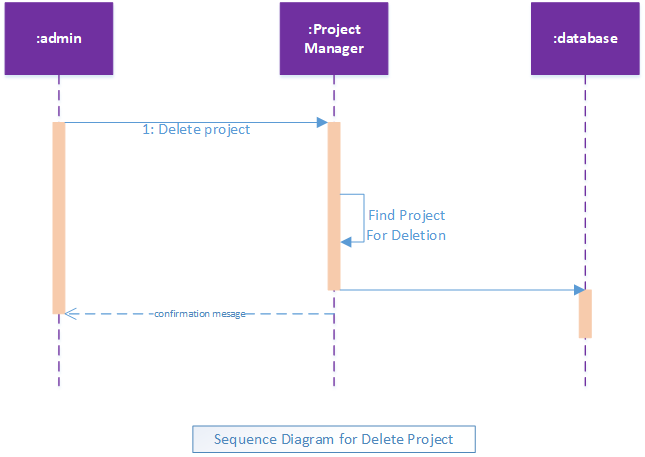
****

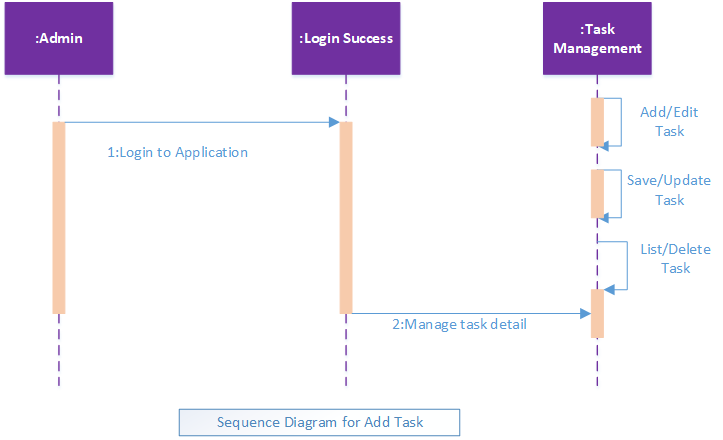
****

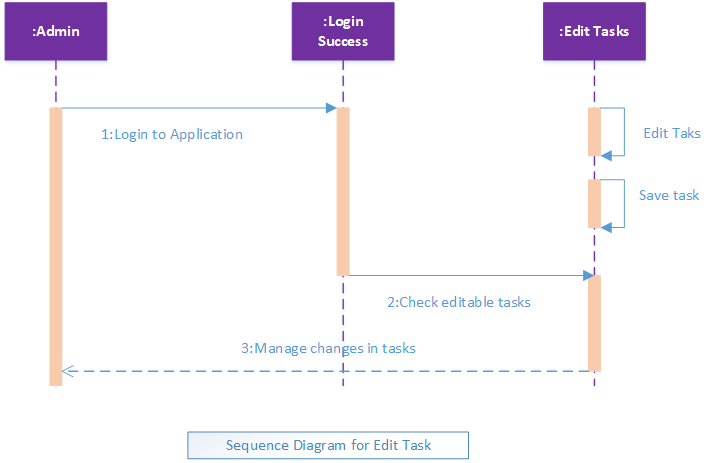
****

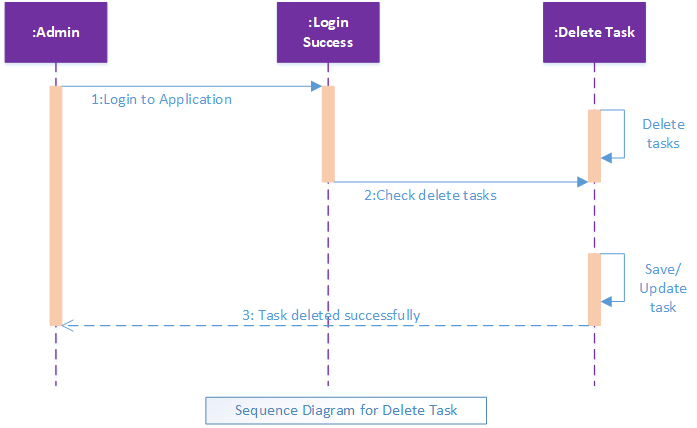
****

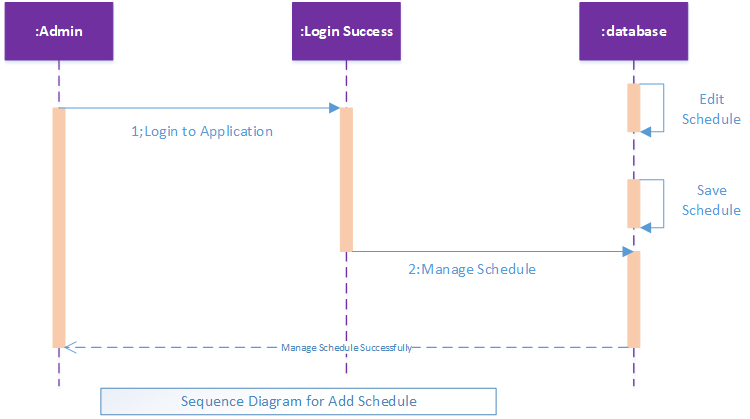
****

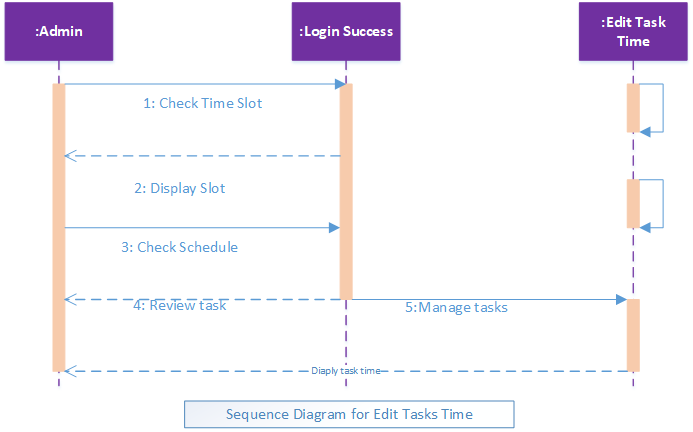
****

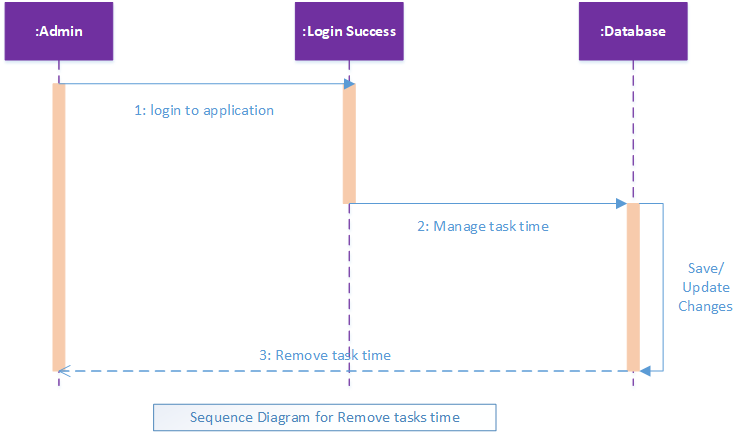
****

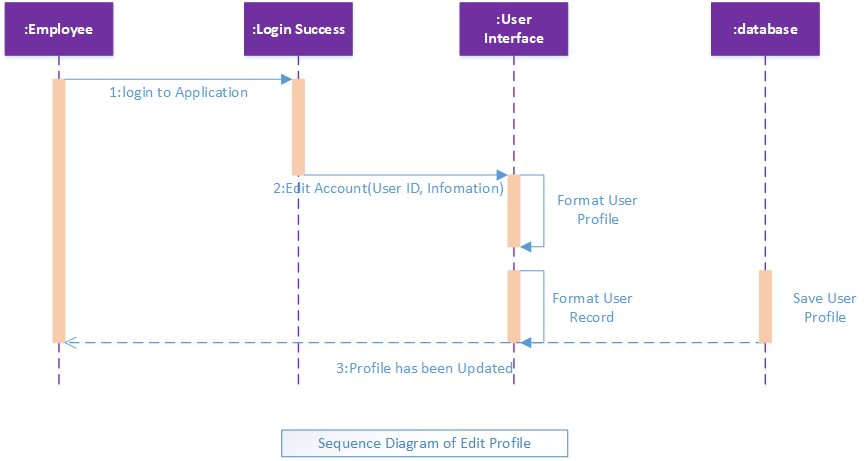
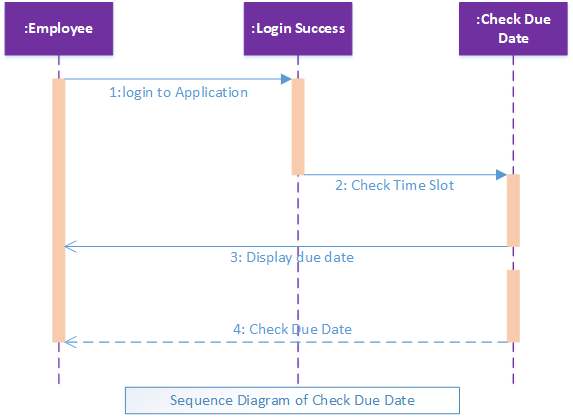
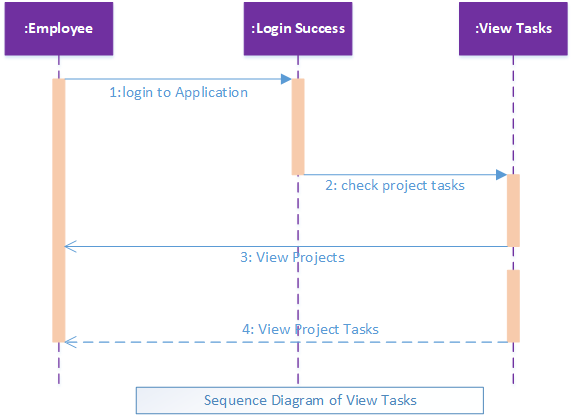
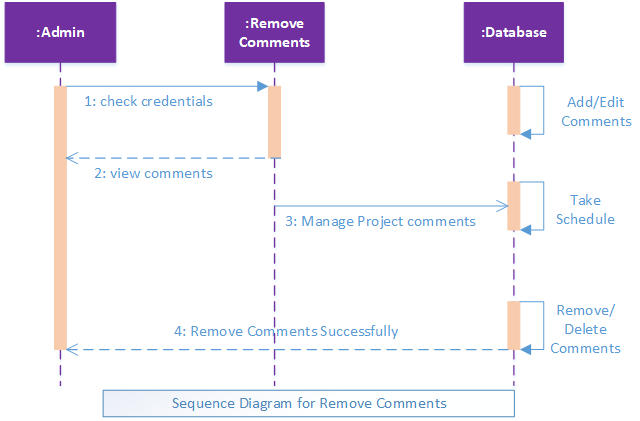
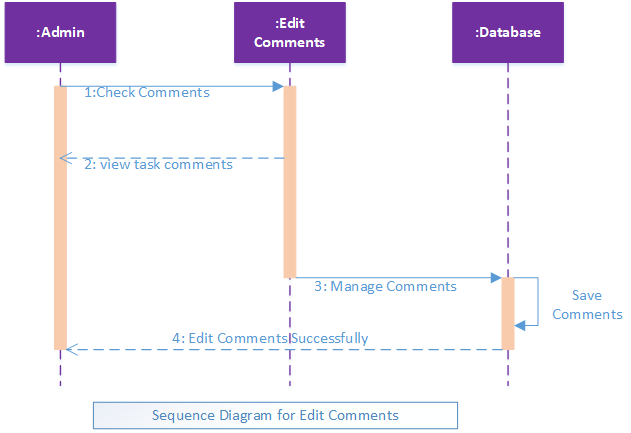
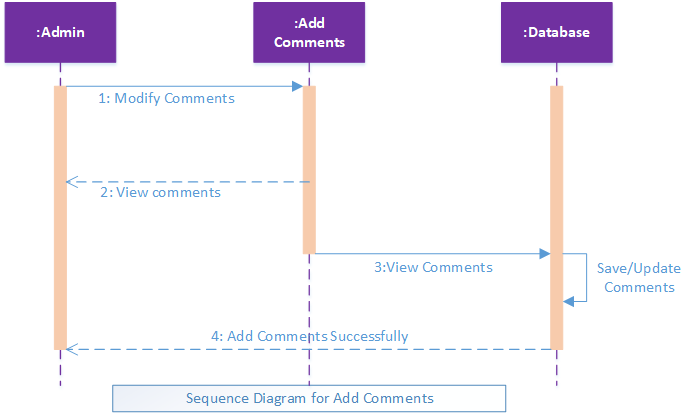
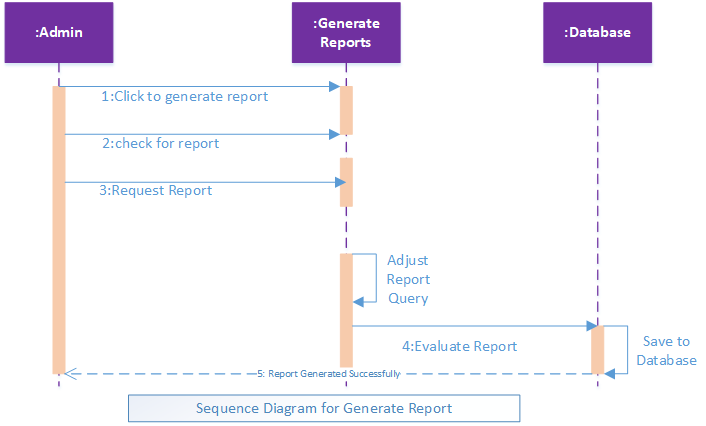
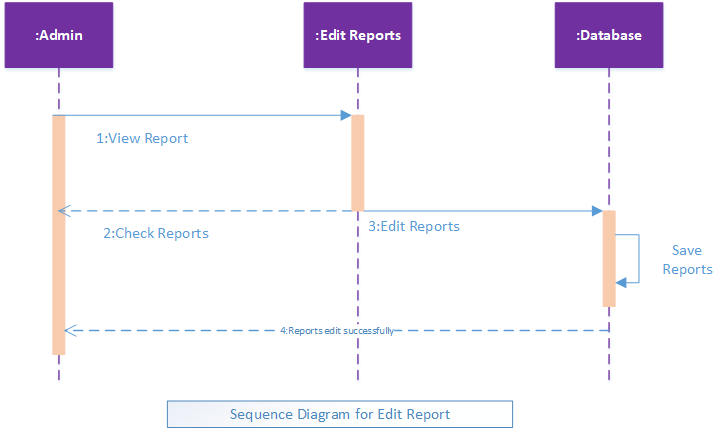
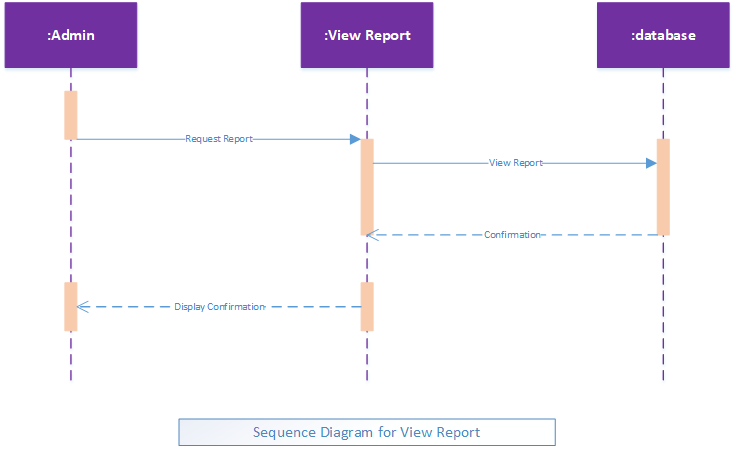
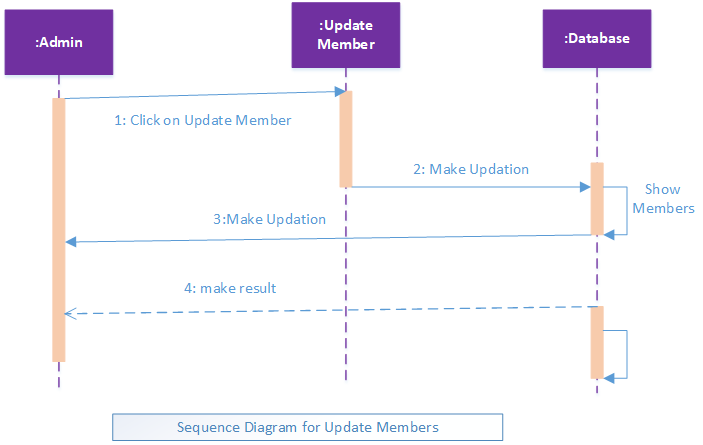
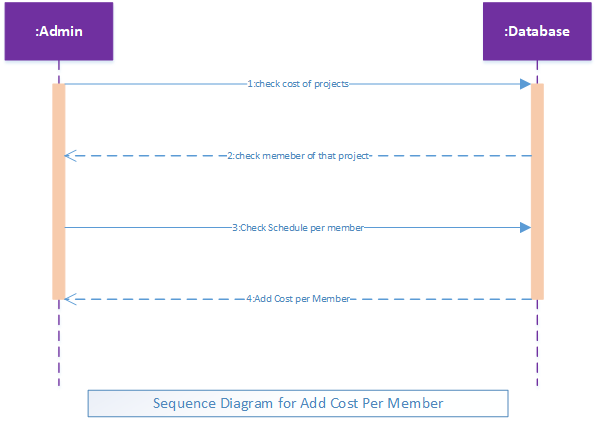
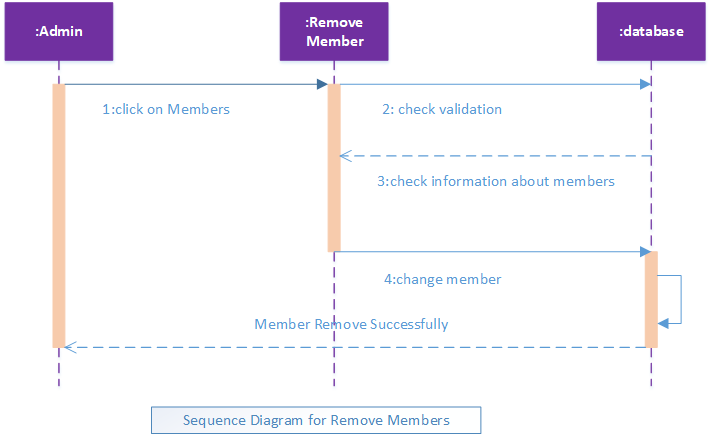
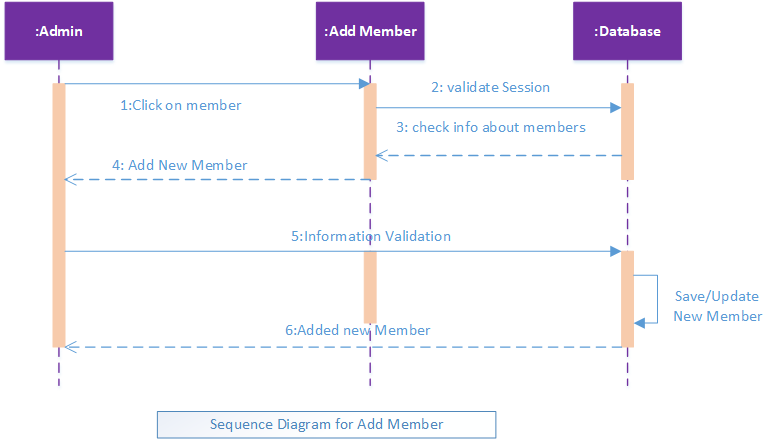
****

****

****

****

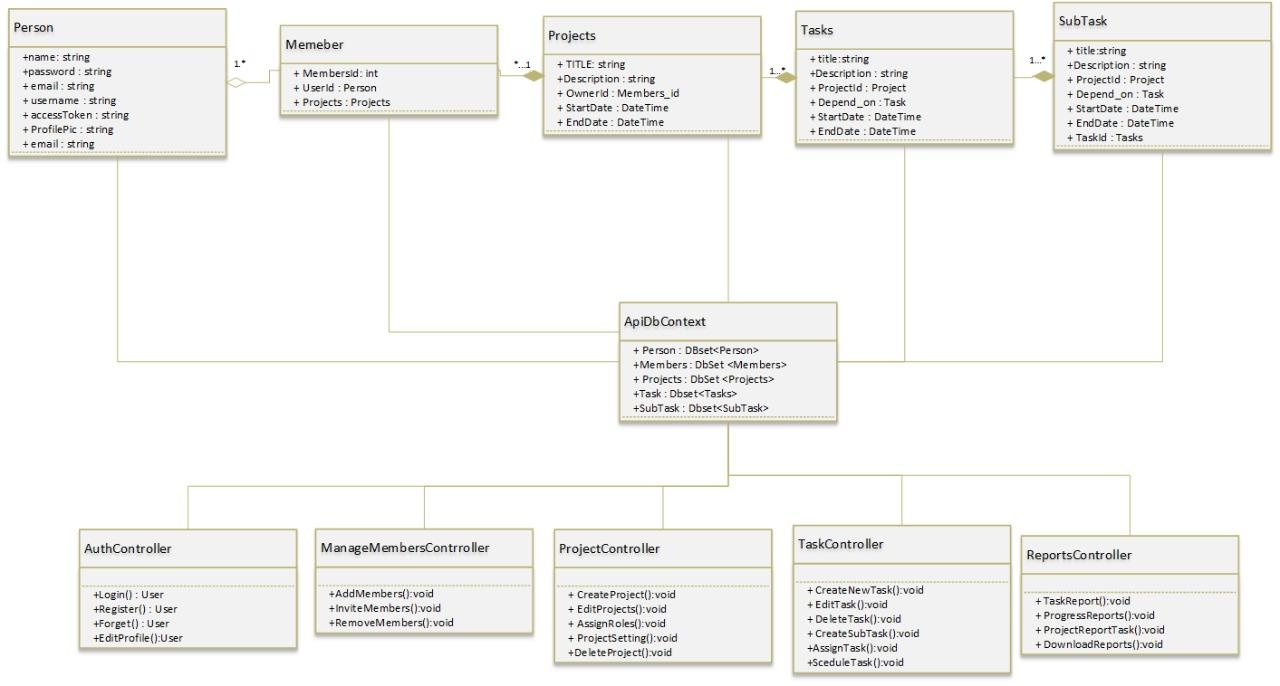
****

****

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

**4.3 Class Diagram**

Here is the class Diagram of our FYP project:

****

* 1. **References:**

Link and Reference are as under:

[1]:[https://openProject.com/](https://openproject.com/)

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