**PROMAN**



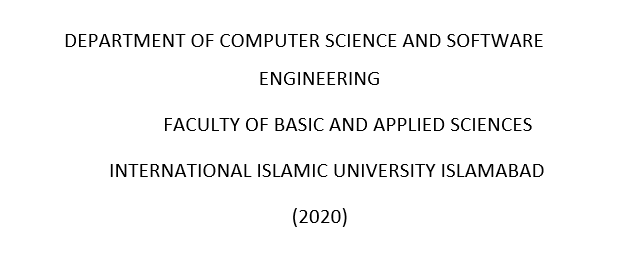
**Developed by**

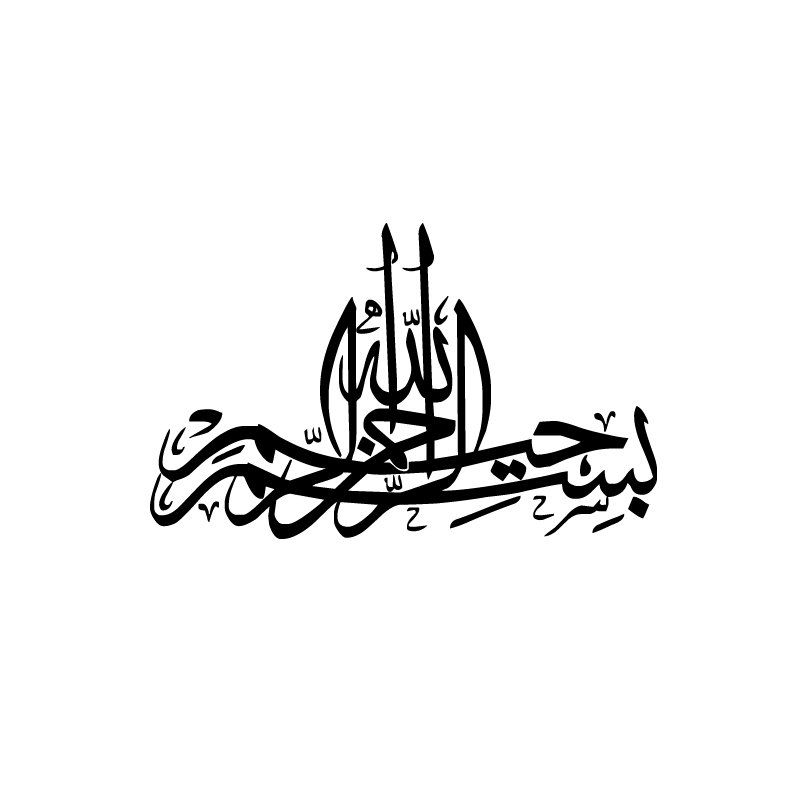
**Usama Majid 3206-FBAS/BSSE/F16**

**Kamran Nazir 3276-FBAS/BSSE/F16**

Supervisedb𝘺

**Dr. Qamar Abbas**





**Final Approval**

**Date:** \_\_\_\_\_\_\_\_\_\_\_\_

It is confirmed that we have perused the Project report title "PROMAN" submitted by Usama Majid 3206-FBAS/BSSE/F16 and Kamran Nazir 3276-FBAS/BSSE/F16 under the supervision of our regarded Teacher Dr. Qamar Abbas from DCS&SE, FBAS, IIUI. It is our notion that this task is of enough standards to complete the fulfillment of DCS and SE, International Islamic University Islamabad for the four year certification of Software Engineering.

**Dedication**

We devote this humble effort to our beloved parents and respectable teachers for their endless support, affection, trust, prayers and encouragement. With completion of our final year project after dedicating this accomplishment and effort to our lovely parents who have constantly supported and provided all the possible facilities that we could complete our studies and the project.

.

**Declaration**

We as a result of this document mention that the development of this project and its report is thoroughly depend on our own hard-work and learning accomplished under the absolute support and instruction of the assigned supervisor **Dr. Qamar Abbas**. Not a single part of this work appeared in this artifact has been proposed for any other degree or any other university or university or any other educational institute. We further announce that this project, all code, and related artifacts and reports are submitted as fractional prerequisites for the degree of Bachelors in Software Engineering.

**Kamran Nazir**

3276-FBAS/BSSE/F16

**Usama Majid**

3206-FBAS/BSSE/F16

**Acknowledgement**

All praise to **Allah Almighty** for his countless blessings and for helping us and giving us courage in completing this project and also blessings upon our last Prophet Muhammad (PBUH).

We want to pay our sincere and true appreciation to supervisor Dr. Qamar Abbas for his meaningful and continuous support and motivation in our project. We are thankful of his brilliant supervision throughout the completion of the project.

**Kamran Nazir**

3276-FBAS/BSSE/F16

**Usama Majid**

3206-FBAS/BSSE/F16

Project in Brief

|  |  |
| --- | --- |
| **Project Title:** | PROMAN |
| **Undertaken By**: | Usama Majid (3206-FBAS/BSSE/F16)  Kamran Nazir (3276-FBAS/BSSE/F16) |
| **Supervised By:** | Dr. Qamar Abbas  Assistant Professor  DCS and SE, IIUI. |
| **Starting Date:**  **Completion Date:** | December 2018  August 2019 |
| **Tool and Technologies used:** | React JS, Node JS, Express (framework for Node),  MongoDB, Ant design,  Visual Studio Code |
| **System Used:** | Core i5, 6 GB RAM and Core m3, 8 GB RAM |
| **Operating System:** | Microsoft Windows 10, Ubuntu 20.04 Chrome |

Table of Contents

[1. Introduction: 12](#_Toc47620134)

[**1.1 Overview:** 12](#_Toc47620135)

[**1.2 PROMAN:** 13](#_Toc47620136)

[**1.3 Problem statement:** 13](#_Toc47620137)

[**1.4 Solution of These Problems:** 13](#_Toc47620138)

[13](#_Toc47620139)

[**1.5 Control Flow Diagram (Leader):** 15](#_Toc47620140)

[**1.6 Control Flow Diagram (User):** 16](#_Toc47620141)

[**2.1 USE CASE DIAGRAM** 18](#_Toc47620142)

[19](#_Toc47620143)

[**2.2 Use Case Description in Detail Expended Format:** 20](#_Toc47620144)

[**2.2.1 Use Case Register:** 20](#_Toc47620145)

[**2.2.2 Use Case Login** 21](#_Toc47620146)

[**2.2.3 Use Case Add Project:** 22](#_Toc47620147)

[**2.2.4 Use Case Add Team:** 23](#_Toc47620148)

[**2.2.5 Use Case Check Team Details:** 24](#_Toc47620149)

[**2.2.6 Use Case Add Member to Team:** 25](#_Toc47620150)

[**2.2.7 Use Case Open Project** 26](#_Toc47620151)

[**2.2.8 Use Case Add Member To Project:** 27](#_Toc47620152)

[**2.2.9 Use Case Open Leader Dashboard:** 28](#_Toc47620153)

[**2.2.10 Use Case Add Task List:** 29](#_Toc47620154)

[**2.2.11 Use Case Add Task:** 30](#_Toc47620155)

[**2.2.12 Use Case Add Sub Task:** 31](#_Toc47620156)

[**2.2.13 Use Case Add Comment:** 32](#_Toc47620157)

[**2.2.14 Use Case Add Attachment:** 33](#_Toc47620158)

[**2.2.15 Use Case Review Task:** 34](#_Toc47620159)

[**2.2.16 Use Case Update Task Status:** 35](#_Toc47620160)

[**2.2.17 Use Case Subscribe:** 36](#_Toc47620161)

[**2.2.18 Use Case Unsubscribe:** 37](#_Toc47620162)

[**2.2.19 Use Case Profile:** 38](#_Toc47620163)

[**2.2.20 Use Case Check Project Details:** 39](#_Toc47620164)

[**2.2.21 Use Case Check Overview of Project:** 40](#_Toc47620165)

[**2.2.22 Use Case Check all Member details of project:** 41](#_Toc47620166)

[**2.2.23 Use Case Check Task by task details:** 42](#_Toc47620167)

[**2.2.24 Use Case Task by member details:** 43](#_Toc47620168)

[**2.3 SEQUENCE DIAGRAMS:** 44](#_Toc47620169)

[**2.3.1 Register:** 44](#_Toc47620170)

[**2.3.2 Login:** 44](#_Toc47620171)

[**2.3.3 Add Project:** 45](#_Toc47620172)

[**2.3.4 Add Team:** 45](#_Toc47620173)

[**2.3.5 Check Team Details:** 46](#_Toc47620174)

[**2.3.6 Add Member to Team:** 46](#_Toc47620175)

[**2.3.7 Open Project:** 47](#_Toc47620176)

[**2.3.8 Add Member to Project:** 47](#_Toc47620177)

[**2.3.9 Open Leader Dashboard:** 48](#_Toc47620178)

[**2.3.10 Add Task List** 48](#_Toc47620179)

[**2.3.11 Add Task:** 49](#_Toc47620180)

[**2.3.12 Add Sub Task:** 49](#_Toc47620181)

[**2.3.13 Add Comment:** 50](#_Toc47620182)

[**2.3.14 Add Attachment:** 50](#_Toc47620183)

[**2.3.15 Review Task:** 51](#_Toc47620184)

[**2.3.16 Update Task Status:** 51](#_Toc47620185)

[**2.3.17 Subscribe:** 52](#_Toc47620186)

[**2.3.18 Unsubscribe:** 52](#_Toc47620187)

[**2.3.19 Profile:** 53](#_Toc47620188)

[**2.3.20 Check Project Details:** 53](#_Toc47620189)

[**2.3.21 Check Overview of Project:** 54](#_Toc47620190)

[**2.3.22 Check All Member Details of Project:** 54](#_Toc47620191)

[**2.3.23 Check Task By Task Details:** 55](#_Toc47620192)

[**2.3.24 Check Task by Member Details:** 55](#_Toc47620193)

[**2.4 Domain Model** 56](#_Toc47620194)

[**3.1 System Sequence Diagram: (Leader)** 57](#_Toc47620195)

[58](#_Toc47620196)

[**3.2 System Sequence Diagram: (User)** 59](#_Toc47620197)

[59](#_Toc47620198)

[**3.4 Alternate to ERD:** 60](#_Toc47620199)

[60](#_Toc47620200)

[4. Implementation: 61](#_Toc47620201)

[**4.1 Functional Features of the Model:** 61](#_Toc47620202)

[**5.1 Black Box Testing:** 62](#_Toc47620203)

[**5.2 White Box Testing:** 62](#_Toc47620204)

[**5.3 Test Cases:** 62](#_Toc47620205)

[**5.3.1 Register:** 62](#_Toc47620206)

[**5.3.2 Login:** 63](#_Toc47620207)

[**5.3.3 Add Project:** 63](#_Toc47620208)

[**5.3.4 Add Team:** 64](#_Toc47620209)

[**5.3.5 Add Members to Team:** 64](#_Toc47620210)

[**5.3.6 Add Task List:** 65](#_Toc47620211)

[**5.3.7 Add Task:** 65](#_Toc47620212)

[**5.3.8 Add Sub Task:** 66](#_Toc47620213)

[**5.3.9 Add Members to Project:** 66](#_Toc47620214)

[**5.3.10 Add Comment:** 67](#_Toc47620215)

[**5.3.11 Add Attachment:** 67](#_Toc47620216)

[**5.3.12 Subscribe Task:** 68](#_Toc47620217)

[**5.3.13 Unsubscribe task:** 68](#_Toc47620218)

[**5.3.14 See Profile:** 69](#_Toc47620219)

[**5.3.15 Review Task:** 69](#_Toc47620220)

[Chapter No. 6 71](#_Toc47620221)

[Conclusion 71](#_Toc47620222)

[**6. Conclusion:** 71](#_Toc47620223)

[Appendix-A 72](#_Toc47620224)

[User Manual 72](#_Toc47620225)

[**A. User manual:** 72](#_Toc47620226)

[**A.1 Public View:** 72](#_Toc47620227)

[**A.2 Register:** 73](#_Toc47620228)

[**A.3 Login:** 74](#_Toc47620229)

[**A.4 Add new Project:** 75](#_Toc47620230)

[**A.5 Add new Project:** 75](#_Toc47620231)

[**A.6 Project Overview:** 76](#_Toc47620232)

[**A.7 Task Overview:** 77](#_Toc47620233)

[**A.8 Project overview in table format:** 77](#_Toc47620234)

[**A.9 Timeline:** 78](#_Toc47620235)

[**A.10 Leader Dashboard:** 79](#_Toc47620236)

# 1. Introduction:

“PROMAN" is basically a project manager which is designed for people to manage their projects online with many and many of options. People can use our platform to manage their project easily and more efficiently than any other such existing system. We are giving people multiple options like they can use our online system for individual as a normal project manager and as well as for a team to manage bigger software projects and can calculate software cost as well.

## **1.1 Overview:**

“PROMAN” is a web based application or system which gives people a great approach to manage their business and software projects very easily and efficiently with was a headache for them before.

## **1.2 PROMAN:**

“PROMAN” is an online platform for people to manage their projects easily and efficiently. People can manage their individual projects as well as team oriented projects. People can manage multiple project at the same time and can be leader in one project and can be ordinary member in other project at the same time. People can calculate cost of software project as well and can chat and discuss issues related to project with other project members. Leader can assign task to members and members can complete and can push it to the leader for his approval before due date. Every member have an efficiency through which one can easily check his performance and leader have many more options to check projects and its details.

## **1.3 Problem statement:**

* Managing Your Project is a difficult task.
* It is difficult to keep track what the team members are doing.
* Assigning tasks to different members is sometimes headache especially if members are not in one place (working remotely).
* It is difficult to see the overall progress of the project.
* It is difficult for members to discuss issues with each other while working remotely.

## **1.4 Solution of These Problems:**

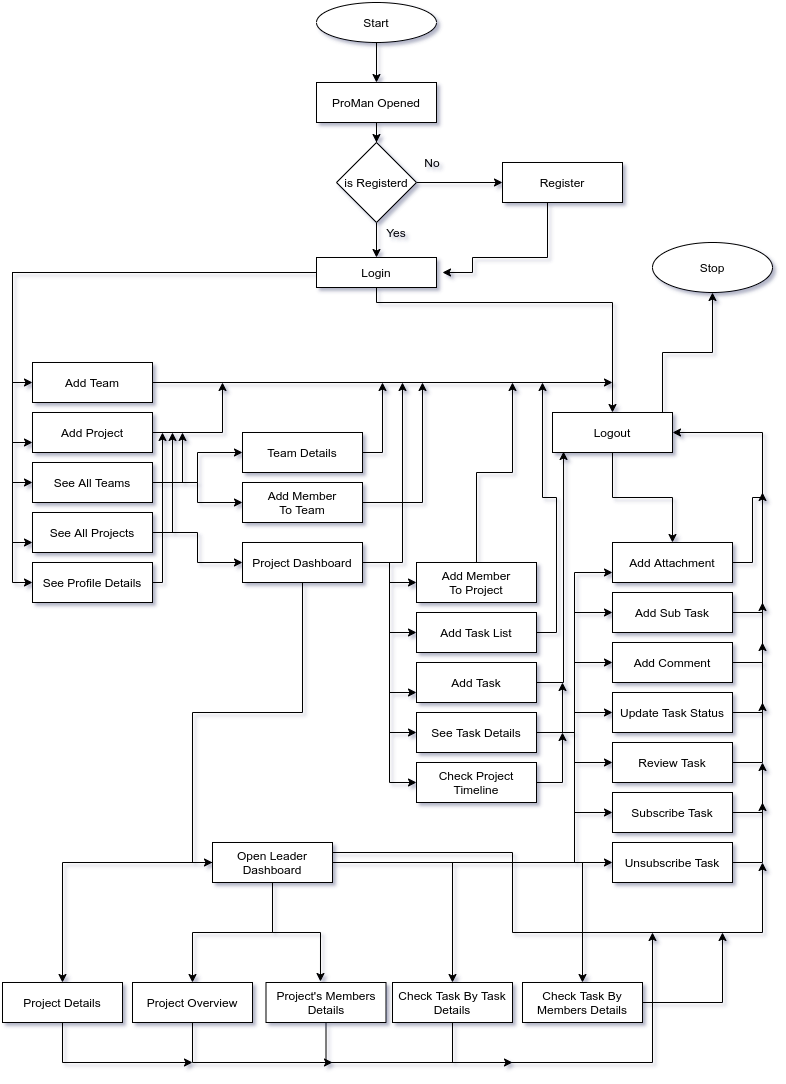
## We are proposing an environment where people can manage their projects remotely. Projects can be managed individually as well as in team oriented environment. People can manage software projects and can estimate cost as well. The beauty of our proposed environment is that even an ordinary person can use it as a task manager for him and can set due date and time so he can work on time and can examine the comprehensive perspective of the project and the remaining time as well.

In PROMAN, user can create project and tasks and can assign it to different members while they are working remotely. User can add comments and attachments to their tasks and can talk to their project fellow members through chat and can easily discuss issues without leaving the environment. One can track and see every minor detail about the project that what is going on and how much work is completed. Leader have other multiple options to approve the work of member, can add members to project and have many options to see every minor detail about the project, task and members.

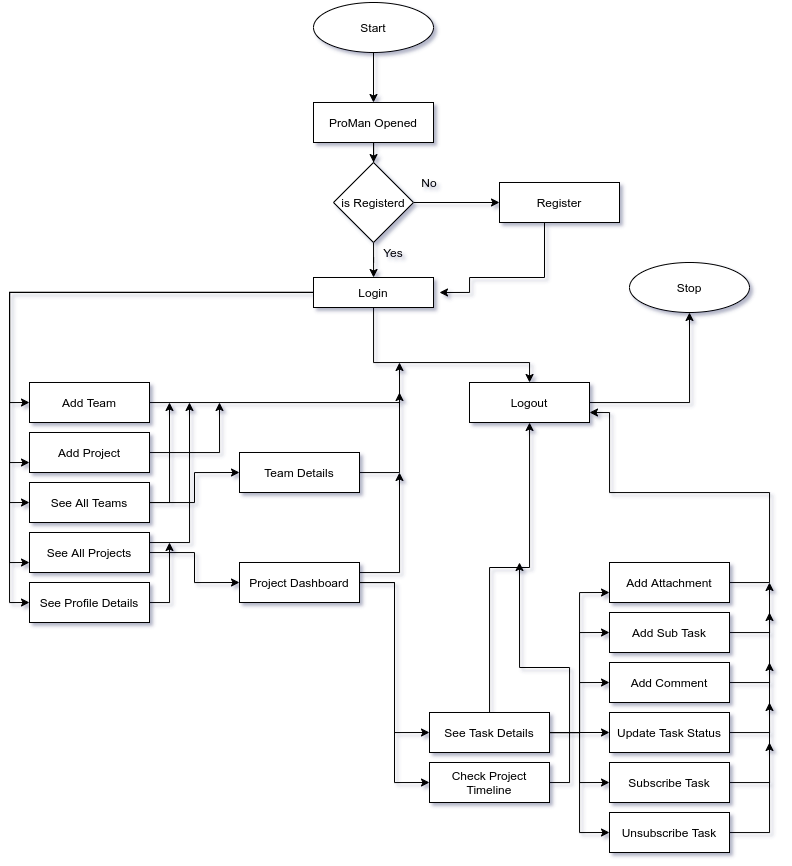
Following are the benefits of our proposed system:

* People can easily manage their project.
* Members can work on different projects at the same time without any conflicts.
* Members can chat with other project members to discuss issues and avoid conflicts.
* Provide large functionality for leader to see the overall project, task and member details.
* Will provide the overall statistics and performance of members which will help leader to identify who is doing well and who is not.

## **1.5 Control Flow Diagram (Leader):**



## **1.6 Control Flow Diagram (User):**



# **2. System Analysis:**

System analysis is basically a methodology in which we identify the problems we are going to face in our system in upcoming phases, checking the facts and decide what is possible and what is not. We take an overview of the whole project in which we consider every single detail about the project. We check the existing system having such type an take some ideas and inputs from that system and then also explore the new domains and functionalities which are required in our required system. Breaking down the system into smaller components or modules is also done in the system analysis phase. It was designed to check the system in such a detailed manner so we can identify its purpose as well as its objective. It is a problem solving technique which ensures us that the system and its components will work fine and will fulfill the functionality for which they are going to be designed and developed.

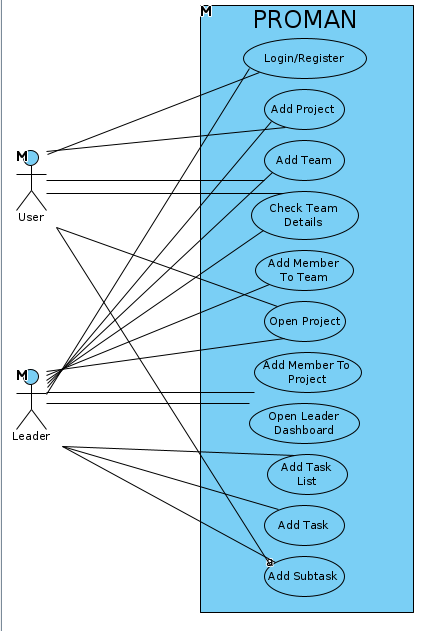
Analysis model includes the following:

1. Use case diagram

2. Detailed Use Cases

3. System sequence diagram (SSD)

## **2.1 USE CASE DIAGRAM**



# 

## **2.2 Use Case Description in Detail Expended Format:**

### **2.2.1 Use Case Register:**

1. Use Case Register that specifies how a user can register himself into our application and use it.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-1** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Register |
| 1. **Primary Actors** | 1. User, Leader |
| 1. **Goals** | 1. To enable users to run the system. |
| 1. **Pre-Conditions** | 1. System must be running. |
| 1. **Post Conditions** | 1. Account will be Created, User will be registered successfully. |
| 1. **Success Scenarios** | 1. User asks for registration enter his personal details and credentials. Then he clicks on register button and will be registered. |
| 1. **Alternate** | 1. The user email already exists. (User already registered). |

### **2.2.2 Use Case Login**

1. Use Case Login that specifies how a user can log into their respective account.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-2** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Login |
| 1. **Primary Actors** | 1. User, Leader |
| 1. **Goals** | 1. Enabling users to logon to their respective account. |
| 1. **Pre-Conditions** | 1. System is running and user, leader already registered their account. |
| 1. **Post Conditions** | 1. User enter the credentials and login to their account successfully. |
| 1. **Success Scenarios** | 1. User entered its email and password and hit login and user successfully login to their account. |
| 1. **Alternate** | 1. The user email or password is wrong so login failed. |

### **2.2.3 Use Case Add Project:**

1. Use Case Add Project that specifies how a user can create a new project to start work on it.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-3** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Add Project |
| 1. **Primary Actors** | 1. User, Leader |
| 1. **Goals** | 1. Enabling user to make a new project |
| 1. **Pre-Conditions** | 1. User, Leader is already logged in. |
| 1. **Post Conditions** | 1. Project creation successful. |
| 1. **Success Scenarios** | 1. User entered project name, description, start date, end date etc. 2. And click on create and project created successfully. |
| 1. **Alternate** | 1. User does not fill form properly. |

### **2.2.4 Use Case Add Team:**

1. Use Case Add Team that specifies how a user can create a new team for its future needs.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-4** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Add Team |
| 1. **Primary Actors** | 1. User, Leader |
| 1. **Goals** | 1. Enabling Use to make a new team |
| 1. **Pre-Conditions** | 1. User is already logged in. |
| 1. **Post Conditions** | 1. Team creation successful. |
| 1. **Success Scenarios** | 1. User will provide Team name, description. 2. And click on create and team created successfully. |
| 1. **Alternate** | 1. User does not fill form properly. |

### **2.2.5 Use Case Check Team Details:**

1. Use Case Team Details that specifies how a user can check the team details.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-5** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Check Team Details |
| 1. **Primary Actors** | 1. User, Leader |
| 1. **Goals** | 1. To enable user to check its team details e. g. its members. |
| 1. **Pre-Conditions** | 1. User is already logged in. And team is already created. |
| 1. **Post Conditions** | 1. Check team details successfully. |
| 1. **Success Scenarios** | 1. User select Team name, and saw its details like its name, description its members. |
| 1. **Alternate** | 1. No team is present to check details. |

### **2.2.6 Use Case Add Member to Team:**

1. Use Case Add Member that specifies how a user can add a new member to the team.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-6** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Add Member To Team |
| 1. **Primary Actors** | 1. Leader |
| 1. **Goals** | 1. To enable leader to add a new member to existing team. |
| 1. **Pre-Conditions** | 1. Leader is already logged in. And team is already created. |
| 1. **Post Conditions** | 1. Add new member to team successfully. |
| 1. **Success Scenarios** | 1. Leader clicked on add member and entered the user email and then add user to team. |
| 1. **Alternate** | 1. No team is present to add new member. 2. User not found due to wrong email. |

### **2.2.7 Use Case Open Project**

1. Use Case Open Project that specifies how a user can open the project in which user is enrolled or user created it yourself.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-7** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Open Project |
| 1. **Primary Actors** | 1. User, Leader |
| 1. **Goals** | 1. To enable user to open the project and see its details. |
| 1. **Pre-Conditions** | 1. User is already logged in. And project is already created in which user in enrolled. |
| 1. **Post Conditions** | 1. Open the project successfully. |
| 1. **Success Scenarios** | 1. User clicked on proceed button on desired project. And project is open successfully. |
| 1. **Alternate** | 1. No project was present in which user is enrolled. |

### **2.2.8 Use Case Add Member To Project:**

1. Use Case Add Member that specifies how a leader can add an another member to the existing project.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-8** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Add Member To Project |
| 1. **Primary Actors** | 1. Leader |
| 1. **Goals** | 1. To enable leader to add a new member to existing project. |
| 1. **Pre-Conditions** | 1. Leader already opened the project. |
| 1. **Post Conditions** | 1. Add new member to project successfully. |
| 1. **Success Scenarios** | 1. Leader clicked on add member and entered the user email and then add user to project. |
| 1. **Alternate** | 1. User not found due to wrong email. |

### **2.2.9 Use Case Open Leader Dashboard:**

1. Use Case Leader Dashboard that specifies how a leader can open the project’s leader dashboard in which all details related to the project is showing (only for leader).

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-9** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Open Leader Dashboard |
| 1. **Primary Actors** | 1. Leader |
| 1. **Goals** | 1. To enable leader to open the leader dashboard. |
| 1. **Pre-Conditions** | 1. User is already logged in. And project is already created in which user is a leader. |
| 1. **Post Conditions** | 1. Open the leader dashboard successfully. |
| 1. **Success Scenarios** | 1. User clicked on open leader dashboard button. And then leader dashboard opened successfully. |
| 1. **Alternate** | 1. User was not leader of the project. |

### **2.2.10 Use Case Add Task List:**

1. Use Case Add Task List that specifies how a leader can add a new task list.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-10** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Add Task List |
| 1. **Primary Actors** | 1. Leader |
| 1. **Goals** | 1. To enable leader to create new task list. |
| 1. **Pre-Conditions** | 1. User is already logged in. And project is already created in which user is a leader. |
| 1. **Post Conditions** | 1. Add a new Task List successfully. |
| 1. **Success Scenarios** | 1. Leader clicked on add task list, and leader enter name and description for task list and task list created successfully. |
| 1. **Alternate** | 1. User was not leader of the project. |

### **2.2.11 Use Case Add Task:**

1. Use Case Add Task that specifies how a leader can add a new task.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-11** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Add Task |
| 1. **Primary Actors** | 1. Leader |
| 1. **Goals** | 1. To enable leader to create new task. |
| 1. **Pre-Conditions** | 1. User is already logged in. And project is already created in which user is a leader. |
| 1. **Post Conditions** | 1. Add a new Task successfully. |
| 1. **Success Scenarios** | 1. Leader clicked on add task, and leader enter name and description, assign members etc for task and task created successfully. |
| 1. **Alternate** | 1. User was not leader of the project. |

### **2.2.12 Use Case Add Sub Task:**

1. Use Case Add Sub Task that specifies how a user can add a new sub task.

|  |  |
| --- | --- |
| 1. **Use Case ID no**. | 1. **UC-12** |
| 1. **Scopes** | 1. Web Application for managing Projects |
| 1. **Name** | 1. Use Case Add Sub Task |
| 1. **Primary Actors** | 1. User, Leader |
| 1. **Goals** | 1. Enabling user to make a new sub task. |
| 1. **Pre-Conditions** | 1. User is already logged in. And project is already created in which user is already enrolled. A task is already assigned. |
| 1. **Post Conditions** | 1. Add a new Sub Task successfully. |
| 1. **Success Scenarios** | 1. User clicked on create new sub task button and enter its details and sub task created successfully. |
| 1. **Alternate** | 1. User was not among the assigned member of the task. |

### **2.2.13 Use Case Add Comment:**

Use Case Add Comment that specifies how the leader or user can add comments.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-13** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Add Comment |
| **Primary Actors** | Leader, User |
| **Goals** | To Enable Leader and user to add comments. |
| **Pre-Conditions** | Leader or user should be registered and logged in.  Project should be existed.  Task must be existed. |
| **Post Conditions** | Comment will be added. |
| **Success Scenarios** | User will write comment and it will be posted when user will click add comment button. |
| **Alternate** | Task does not exists. |

### **2.2.14 Use Case Add Attachment:**

Use Case add attachment which specifies the leader or user how to add attachment.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-14** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Add Attachment |
| **Primary Actors** | Leader, User |
| **Goals** | To Enable Leader or User to add attachments. |
| **Pre-Conditions** | Leader or user should be registered and logged in.  Project should be existed.  Task must be existed. |
| **Post Conditions** | Attachment will be added. |
| **Success Scenarios** | User will add attachment by clicking on add attachment or by drag and drop on add attachment button. |
| **Alternate** | Task does not exists. |

### **2.2.15 Use Case Review Task:**

Use Case Review task which specifies the Leader how to review task.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-15** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Review Task |
| **Primary Actors** | Leader |
| **Goals** | To enable Leader to review task. |
| **Pre-Conditions** | Leader must be registered and logged in.  Project must be existed.  Task must be existed. |
| **Post Conditions** | Task will be reviewed. |
| **Success Scenarios** | Leader will be able to alter the status of task from pending to done and pending to in progress. |
| **Alternate** | Task is not available for review. |

### **2.2.16 Use Case Update Task Status:**

Use Case Update Task status which specifies the Leader and user how to Update Task Status.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-16** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Update Task Status |
| **Primary Actors** | Leader, User |
| **Goals** | Enabling User to forward task from in progress to pending. |
| **Pre-Conditions** | Leader must be registered and logged in.  Project must be existed.  Task must be existed. |
| **Post Conditions** | Status of task will be updated. |
| **Success Scenarios** | Leader or User will click on Push button to change the status from in progress to pending. |
| **Alternate** | No alternate. |

### **2.2.17 Use Case Subscribe:**

Use Case Subscribe which specifies the leader and user how to subscribe a task.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-17** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Subscribe |
| **Primary Actors** | Leader, User |
| **Goals** | Enabling Leader or User to get notifications about the task. |
| **Pre-Conditions** | Leader or User should be logged in.  Project should be existed.  Task should be existed. |
| **Post Conditions** | Task will be subscribed. |
| **Success Scenarios** | Leader or User will click on Subscribe button and should get the notifications about that task. |
| **Alternate** | User is among the assigned member of the task so will not be able to subscribe. |

### **2.2.18 Use Case Unsubscribe:**

Use Case Unsubscribe which specifies the leader and user how to unsubscribe a task.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-18** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Unsubscribe |
| **Primary Actors** | Leader, User |
| **Goals** | To enable Leader or User to not be able to get notifications about that particular task. |
| **Pre-Conditions** | Leader or User should be logged in.  Project should be existed.  Task must be existed.  Task must be subscribed. |
| **Post Conditions** | Task will be subscribed. |
| **Success Scenarios** | Leader or User will click on Unsubscribe button and will not get notifications about that task in future. |
| **Alternate** | Task is not Subscribed. |

### **2.2.19 Use Case Profile:**

Use Case Profile which specifies the leader and user how to check Profile.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-19** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Check Profile |
| **Primary Actors** | Leader, User |
| **Goals** | Enabling Leader or User to see profile details. |
| **Pre-Conditions** | Leader or User must be registered and logged in. |
| **Post Conditions** | Leader or User will be have option to see profile. |
| **Success Scenarios** | Leader or User will click on avatar (Navbar) and then will click on profile option to see the profile. |
| **Alternate** | Leader or User is not registered or logged in. |

### **2.2.20 Use Case Check Project Details:**

Use Case Check Project Details which specifies the Leader how to Check Project Details.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-20** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Check Project Details |
| **Primary Actors** | Leader |
| **Goals** | To enable Leader to see project details. |
| **Pre-Conditions** | Leader should be logged in.  Project should be existed and User should be the Leader. |
| **Post Conditions** | Leader will be able to see project details. |
| **Success Scenarios** | Leader will click on Leader Dashboard and then Project details button in side bar to see project details. |
| **Alternate** | User is not a leader. |

### **2.2.21 Use Case Check Overview of Project:**

Use Case Check Overview of Project which specifies the Leader how to Check Overview of Project.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-21** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Check Overview of Project |
| **Primary Actors** | Leader |
| **Goals** | Enabling Leader to check overview of project. |
| **Pre-Conditions** | Leader should be logged in.  Project should be existed and User should be the Leader. |
| **Post Conditions** | Leader will be able to see overview of project. |
| **Success Scenarios** | Leader will click on leader dashboard from project and then click on project overview button from sidebar to see project overview. |
| **Alternate** | User is not a leader. |

### **2.2.22 Use Case Check all Member details of project:**

Use Case check all Member details of project which specifies the Leader how to see all Member details of project.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-22** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Check all Member details of project |
| **Primary Actors** | Leader |
| **Goals** | To enable Leader to see member details of project. |
| **Pre-Conditions** | Leader must be registered and logged in.  Project must be existed.  Members must be existed.  User must be the Leader. |
| **Post Conditions** | Leader will be able to see member details of project. |
| **Success Scenarios** | Leader will click on leader dashboard from project and then click on member details button from sidebar to see project member’s details. |
| **Alternate** | User is not a leader. |

### **2.2.23 Use Case Check Task by task details:**

Use Case Check Task by task details which specifies the Leader how to see Task by task details.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-23** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Task by task details |
| **Primary Actors** | Leader |
| **Goals** | Enabling Leader to see the overall task details. |
| **Pre-Conditions** | Leader must be registered and logged in.  Project must be existed.  Task must be existed.  User must be the Leader. |
| **Post Conditions** | Leader will be able to see the overall task details. |
| **Success Scenarios** | Leader will click on leader dashboard from project and then click on task by task button from sidebar to see overall task details. |
| **Alternate** | User is not a leader. |

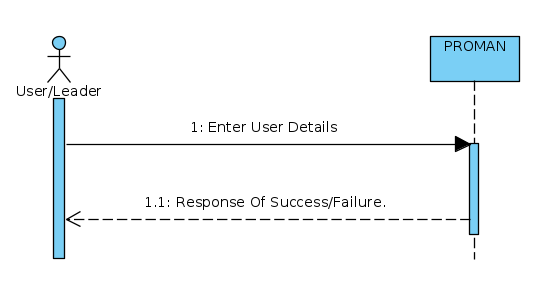
### **2.2.24 Use Case Task by member details:**

Use Case Check Task by member details which specifies the Leader how to see Task by member details.

|  |  |
| --- | --- |
| **Use Case ID no**. | **UC-24** |
| **Scopes** | Web Application for managing Projects |
| **Name** | Task by member details |
| **Primary Actors** | Leader |
| **Goals** | Enabling Leader to see task by member details. |
| **Pre-Conditions** | Leader must be registered and logged in.  Project must be existed.  Task must be existed.  Members must be existed.  User must be the Leader. |
| **Post Conditions** | Leader will be able to see task by member details. |
| **Success Scenarios** | Leader will click on leader dashboard from project and then click on member by task details button from sidebar to see task by member details. |
| **Alternate** | User is not a leader. |

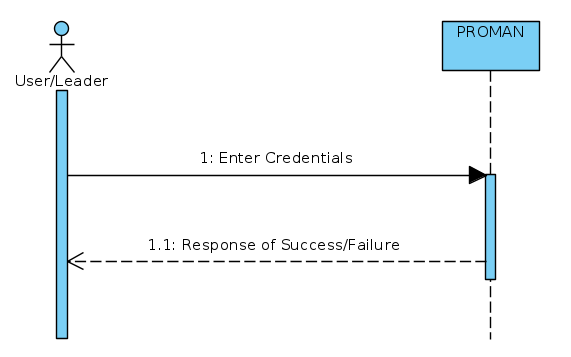
## **2.3 SEQUENCE DIAGRAMS:**

### **2.3.1 Register:**

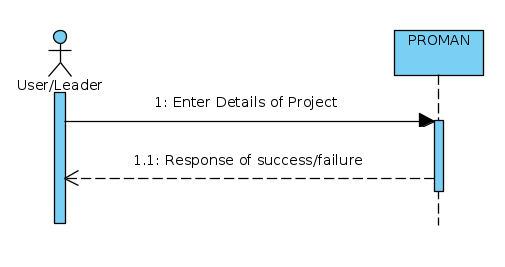


### 

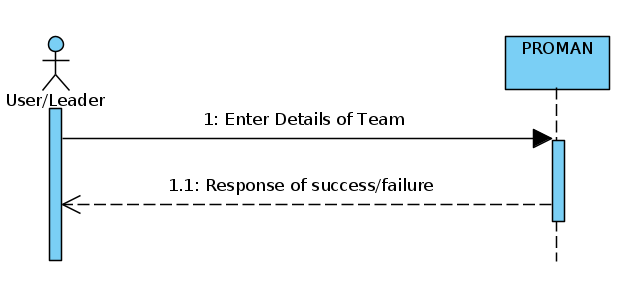
### **2.3.2 Login:**



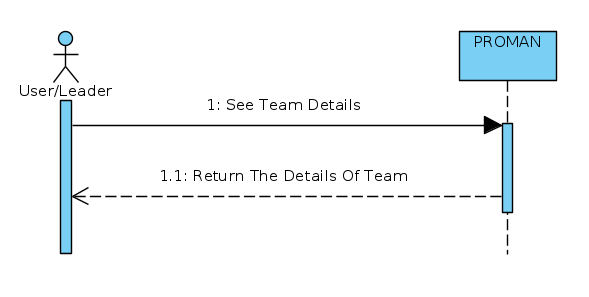
### **2.3.3 Add Project:**



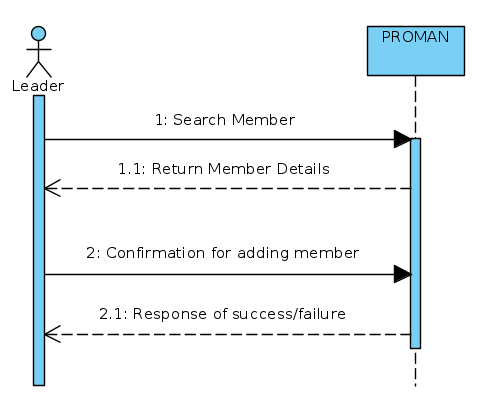
### **2.3.4 Add Team:**



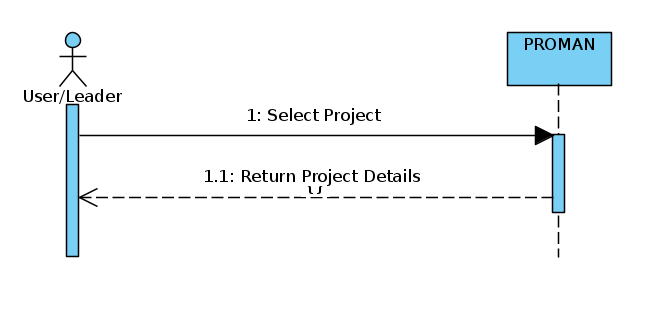
### **2.3.5 Check Team Details:**



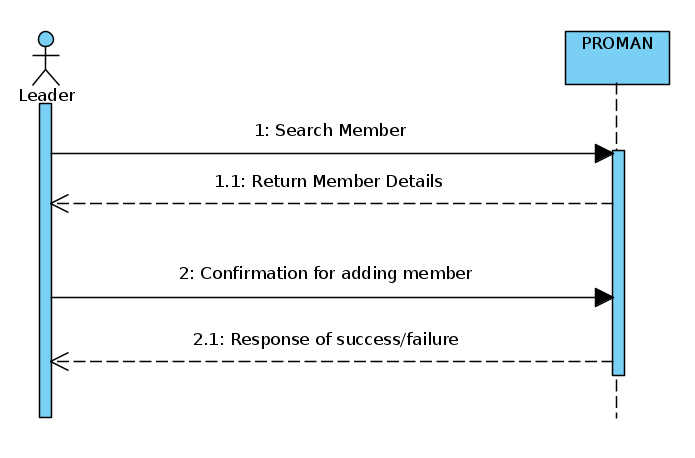
### **2.3.6 Add Member to Team:**



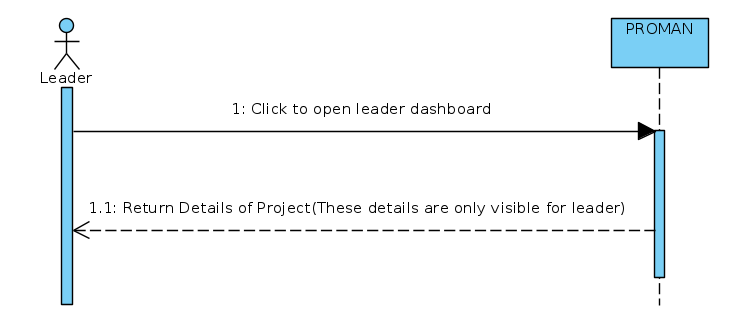
### **2.3.7 Open Project:**



### **2.3.8 Add Member to Project:**

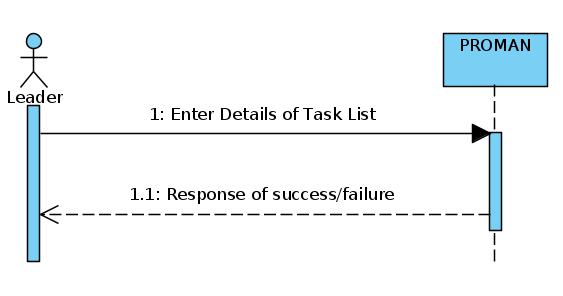


### **2.3.9 Open Leader Dashboard:**

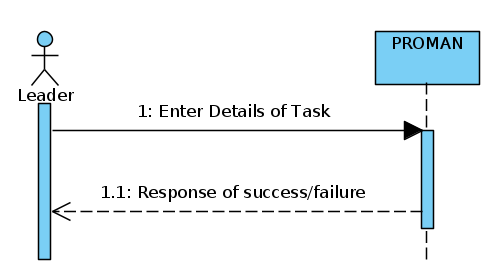


### 

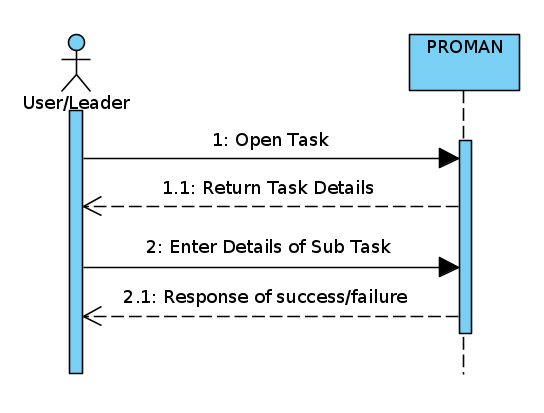
### **2.3.10 Add Task List**



### **2.3.11 Add Task:**

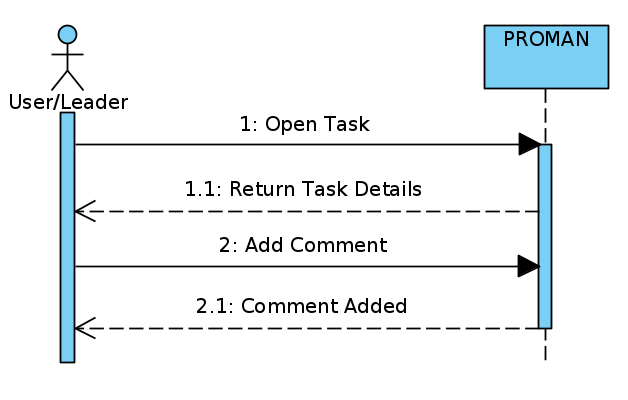


### **2.3.12 Add Sub Task:**

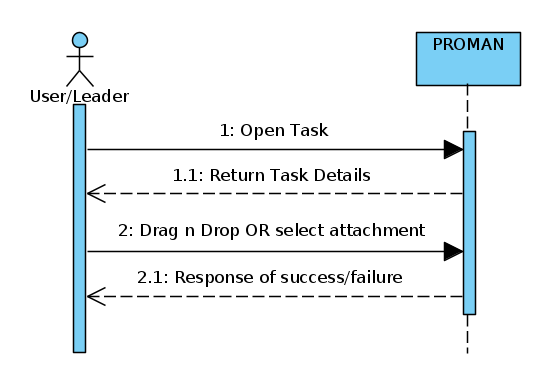


### 

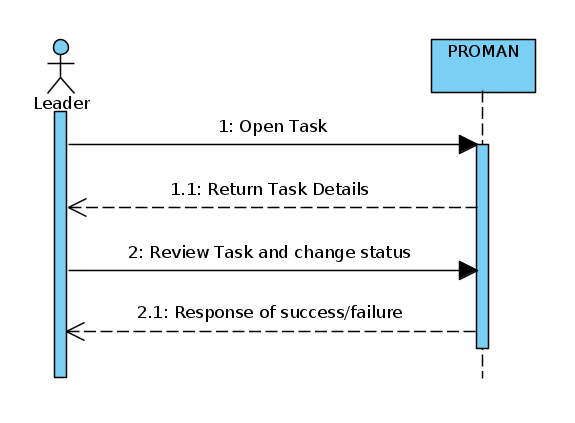
### **2.3.13 Add Comment:**



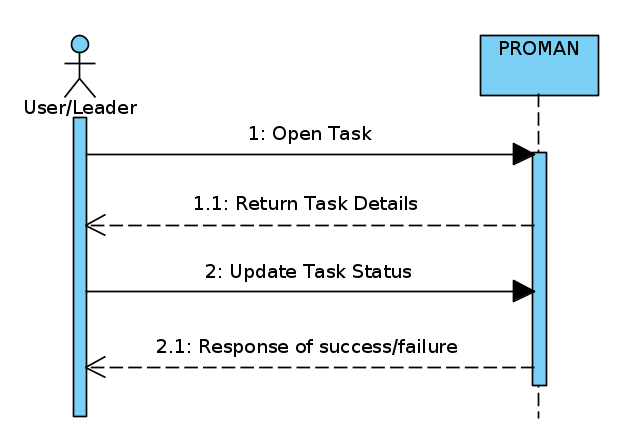
### **2.3.14 Add Attachment:**



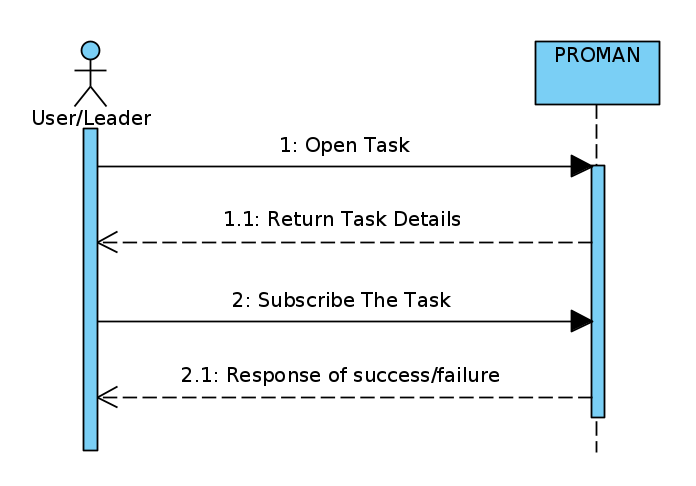
### **2.3.15 Review Task:**



### **2.3.16 Update Task Status:**

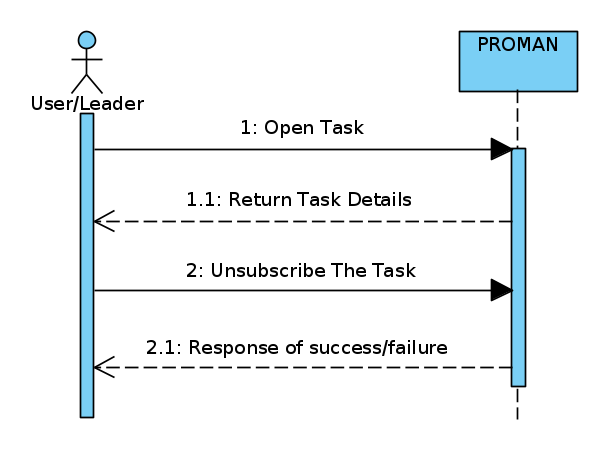


### **2.3.17 Subscribe:**

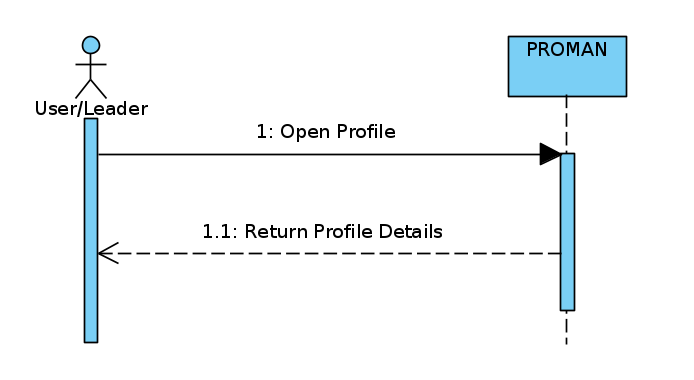


### 

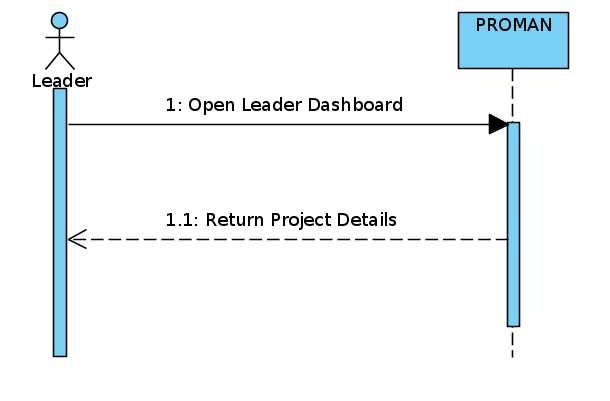
### **2.3.18 Unsubscribe:**



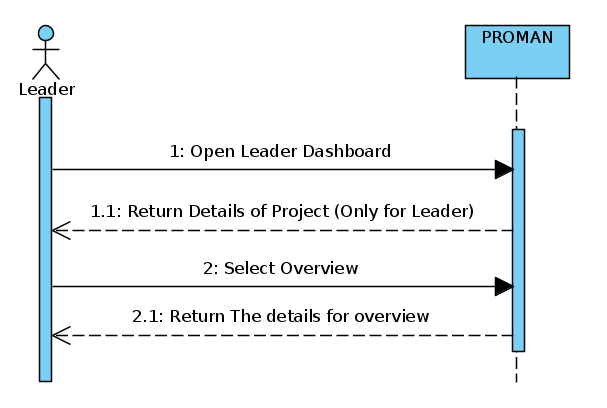
### **2.3.19 Profile:**



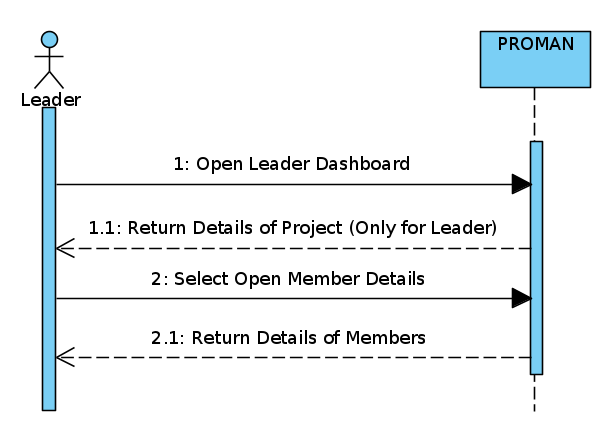
### **2.3.20 Check Project Details:**



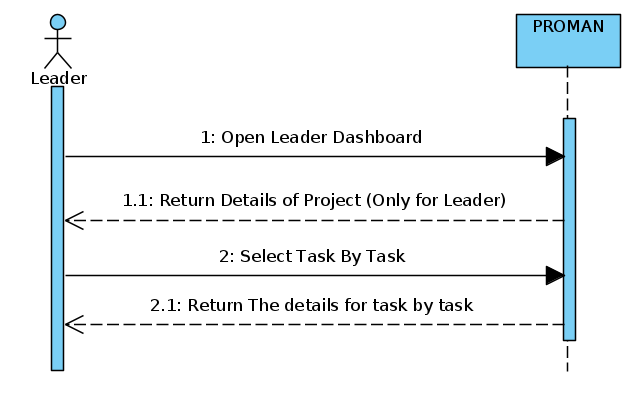
### **2.3.21 Check Overview of Project:**



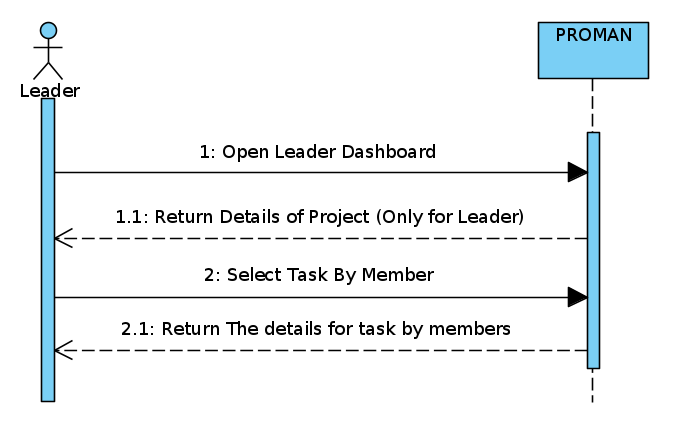
### **2.3.22 Check All Member Details of Project:**



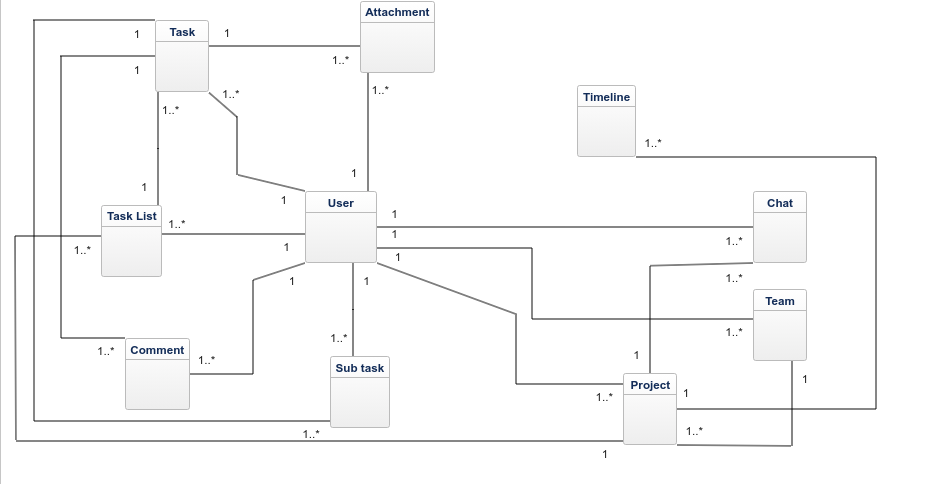
### **2.3.23 Check Task By Task Details:**



### **2.3.24 Check Task by Member Details:**

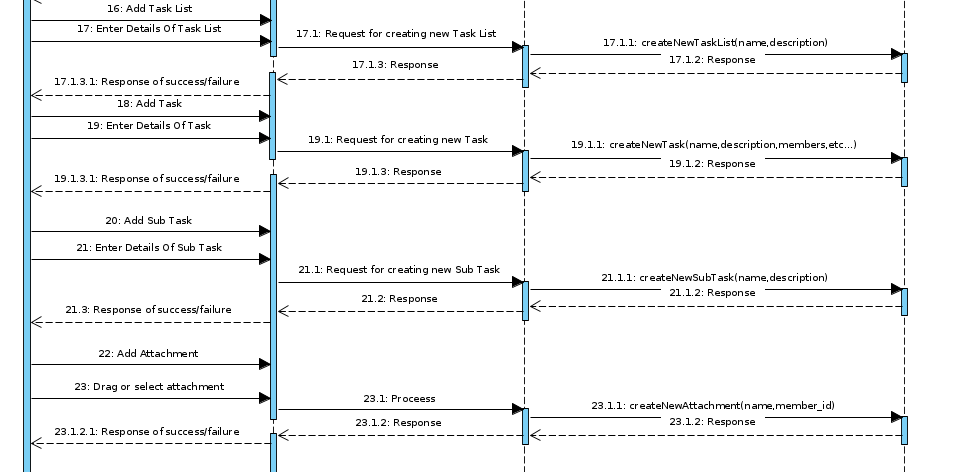
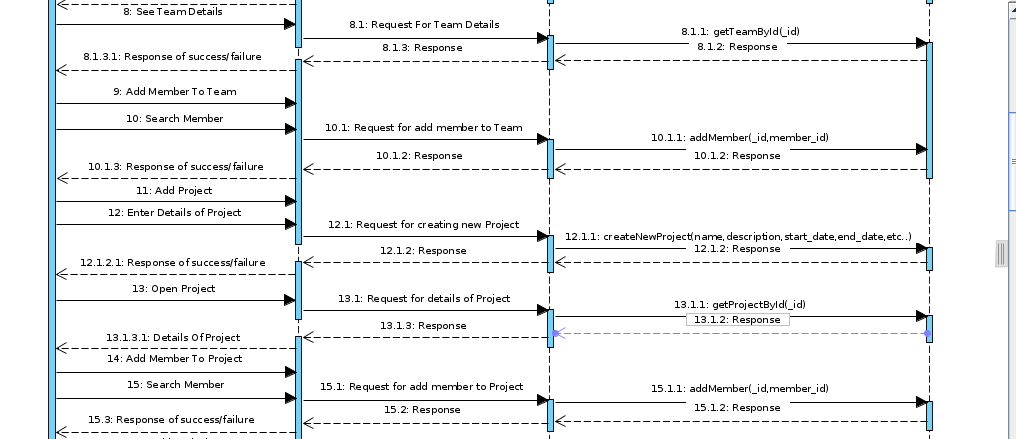
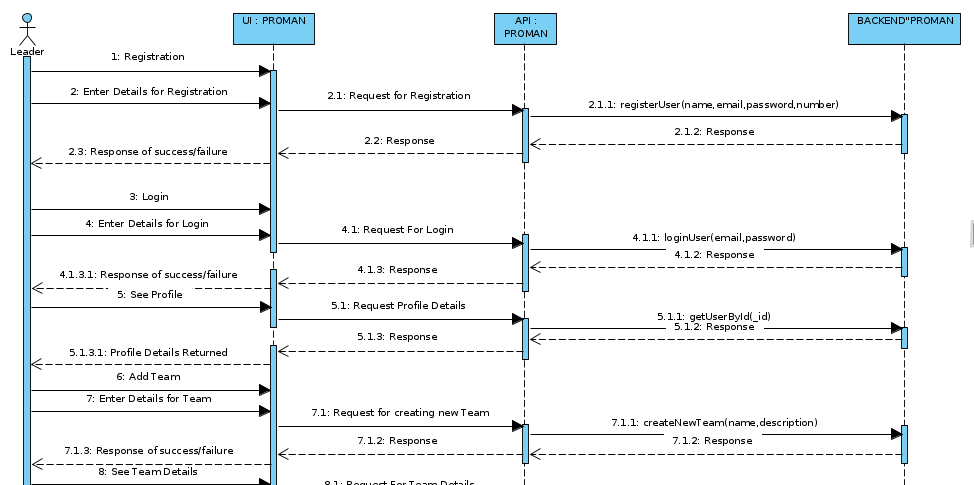


## **2.4 Domain Model**

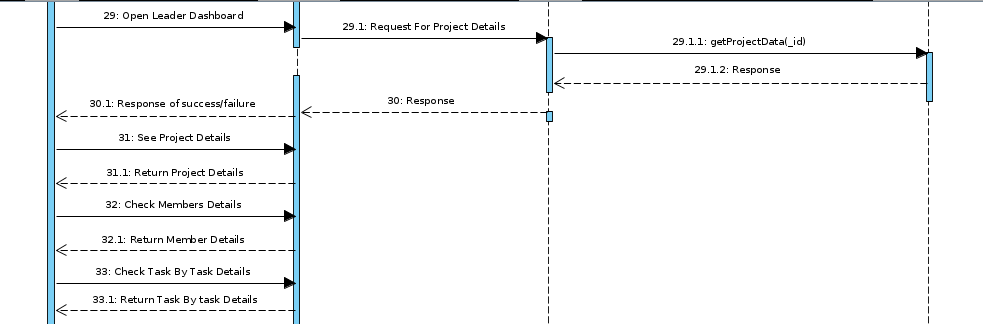
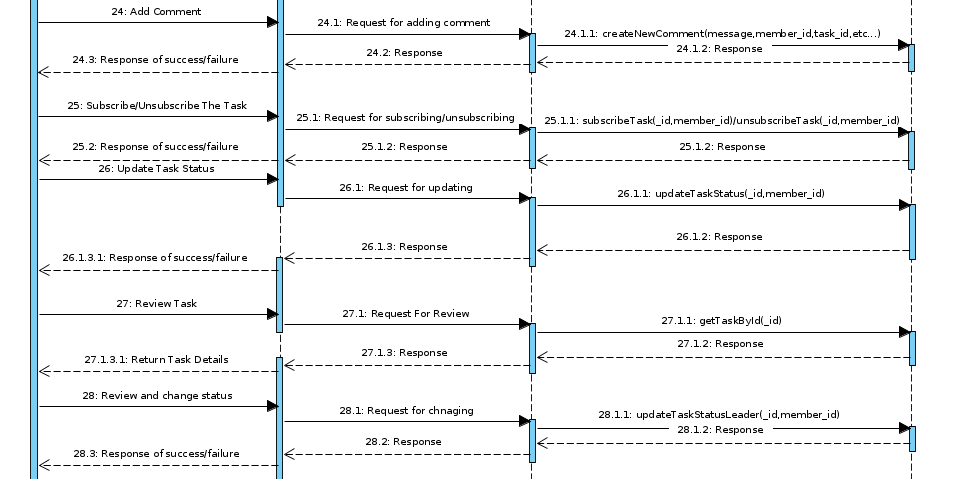


3. System Design:

## **3.1 System Sequence Diagram: (Leader)**



# 



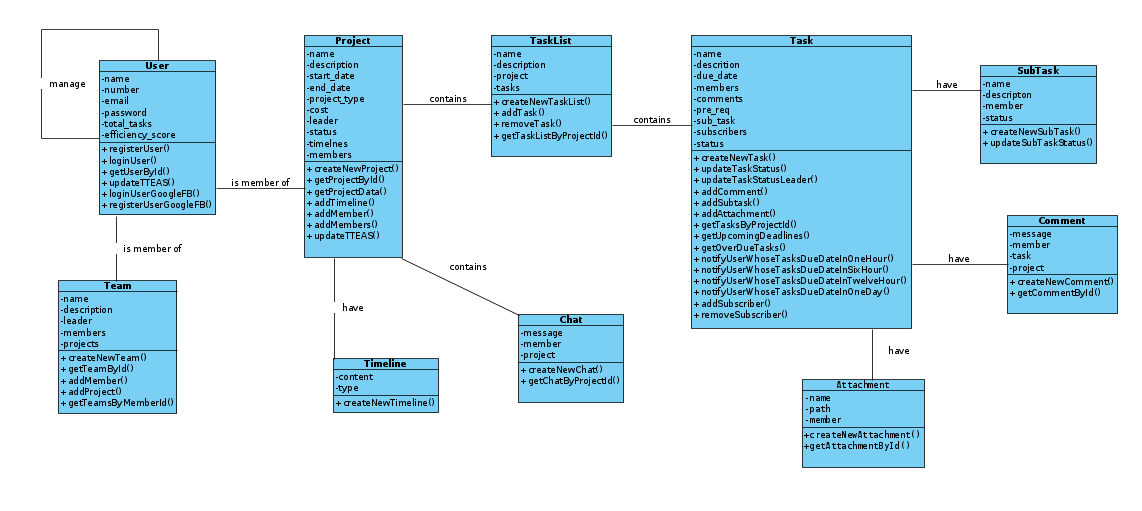
### **3.2 System Sequence Diagram: (User)**

### 

## **3.4 Alternate to ERD:**

Because we are using Mongo DB

## **3.5 Class Diagram**



# 4. Implementation:

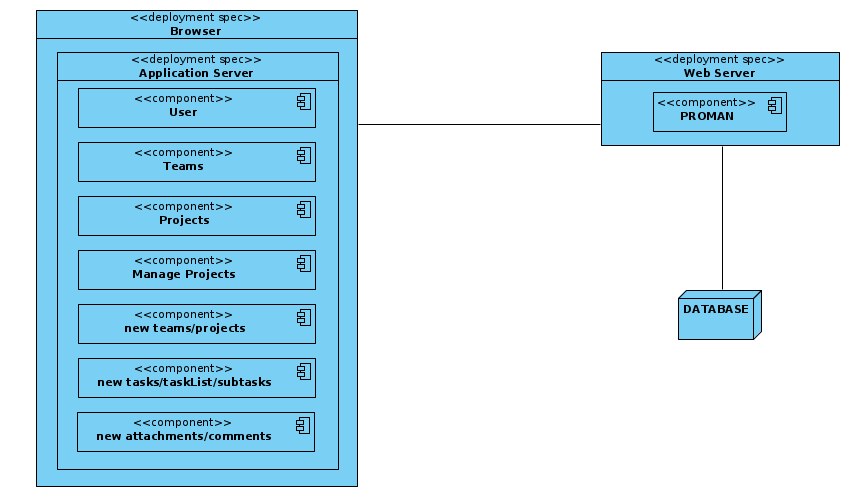
Implementation is usually considered as the coding of the system which is going to be developed. We will be discussing component diagrams and deployment diagram in this chapter**.**

## **4.1 Functional Features of the Model:**

Our system includes the following features:

* User friendly Interface
* Easy to use
* Fully Responsive
* Totally reliable
* Fully Secure

# 4.2 Deployment Diagram.



# 5. System Testing:

Software testing is an activity which is usually done at the end of development process but it can be done before the development process on the artifacts which can be different documents. System testing is basically used to identify the issues which we are having in our system that is checking and comparing the expected outputs with the outputs generated by the system. There are two major types of testing which are as follows:

## **5.1 Black Box Testing:**

Black box testing is a type of testing in which we consider the behavior of the system without having any concerns with the internal functionality of a software or a code. We just go for the output and match it with the expected one and then concludes the result on the basis of that comparison.

## **5.2 White Box Testing:**

White box testing is a type of testing in which we have the code as well, where we can check all the internal and well as external behavior of the system in a very detailed manner and often this type of testing is also called as Open testing,

## 

## **5.3 Test Cases:**

### **5.3.1 Register:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-01 |
| **Functional Area/Module** | Register |
| **Action to Perform** | 1. User starts the application. 2. Clicks on register button. 3. Enters name, email, password and other personal information. 4. Clicks on register button. 5. Finish |
| **Purpose** | To register a user. |
| **Prerequisites** | Application should be up and running. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | User registered to system successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.2 Login:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-02 |
| **Functional Area/Module** | Login |
| **Action to Perform** | 1. User starts the application. 2. Clicks on Login button. 3. Enters his credentials. 4. Clicks on login button. 5. Finish |
| **Purpose** | To create a project. |
| **Prerequisites** | Application should be up and running. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | User logged in successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.3 Add Project:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-03 |
| **Functional Area/Module** | Add Project |
| **Action to Perform** | 1. User starts the application. 2. Clicks on Login button. 3. Enters his credentials. 4. Clicks on login button. 5. Finish |
| **Purpose** | To create a project. |
| **Prerequisites** | Application should be up and running. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Project created successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.4 Add Team:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-04 |
| **Functional Area/Module** | Add Team |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks on add team. 4. Enter team information. 5. Click on create button. 6. Finish |
| **Purpose** | To create a team. |
| **Prerequisites** | Application should be up and running. User should be logged in. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Team added successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.5 Add Members to Team:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-05 |
| **Functional Area/Module** | Add Members to Team |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks Add members to team  button from sidebar. 4. Click on add member and Enter his email to search. 5. Click on add button. 6. Finish. |
| **Purpose** | To add members to a team. |
| **Prerequisites** | Application should be running.  User should be registered and logged in.  User you want to add must exists and should not be a part of team. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Member added to team successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.6 Add Task List:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-06 |
| **Functional Area/Module** | Add task list |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks on existing project. 4. Clicks on add task list. 5. Provide its name and description. 6. Hit on create button. 7. Finish. |
| **Purpose** | To add task list to projects. |
| **Prerequisites** | Application should be up and running.User should be registered and logged in.  Project should be existed and you must be the leader. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Task list created successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.7 Add Task:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-07 |
| **Functional Area/Module** | Add task |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks on existing project. 4. Clicks on add task. 5. Enter its name and other detail. 6. Click on create button. 7. Finish. |
| **Purpose** | To add task to project. |
| **Prerequisites** | Application should be up and running.User should be registered and logged in.  Project should be existed and you must be the leader. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Task added successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.8 Add Sub Task:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-08 |
| **Functional Area/Module** | Add Subtask |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks on existing project. 4. Clicks on existing task. 5. Click on add subtask button. 6. Enter its name and description. 7. Click on create button. 8. Finish. |
| **Purpose** | To add Sub task to projects. |
| **Prerequisites** | Application should be up and running. User should be registered and logged in.  Project and task must be existed. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Sub Task added successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.9 Add Members to Project:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-09 |
| **Functional Area/Module** | Add members to project |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks on existing project. 4. Clicks on add member button. 5. Enter its email. 6. Click on add. 7. Finish. |
| **Purpose** | To add task list to projects. |
| **Prerequisites** | Application must be running.  User must be registered and logged in.  Project must be existed and you must be the leader. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Member added to project successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.10 Add Comment:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-10 |
| **Functional Area/Module** | Add comment |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks on existing project. 4. Clicks on existing task. 5. Enter its comment. 6. Click on post button. 7. Finish. |
| **Purpose** | To add comments on tasks. |
| **Prerequisites** | Application should be up and running.User should be registered and logged in.  Project and task should be existed. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Comment posted successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.11 Add Attachment:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-11 |
| **Functional Area/Module** | Add attachment |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks on existing project. 4. Clicks on existing task. 5. Click on attachment button or drag and drop file to add attachment. 6. Finish. |
| **Purpose** | To add attachment in task. |
| **Prerequisites** | Application should be up and running. User should be registered and logged in.  Project and task should be existed. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Attachment posted successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.12 Subscribe Task:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-12 |
| **Functional Area/Module** | Subscribe Task |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks on existing project. 4. Clicks on existing task. 5. Press Subscribe button. 6. Finish. |
| **Purpose** | To get notifications about a tasks. |
| **Prerequisites** | Application should be up and running. User should be registered and logged in.  Project and task should be existed.  User should not be a part of that task. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Task Subscribed successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.13 Unsubscribe task:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-13 |
| **Functional Area/Module** | Unsubscribe Task |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. Clicks on existing project. 4. Clicks on existing task. 5. Press Unsubscribe button. 6. Finish. |
| **Purpose** | To stop getting notifications about a tasks. |
| **Prerequisites** | Application must be running.  User must be registered and logged in.  Project and task must be existed.  User must be subscribed to that task. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Task Unsubscribed successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.14 See Profile:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-14 |
| **Functional Area/Module** | See Profile |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. User clicks on avatar. 4. Then clicks on profile button. 5. Finish. |
| **Purpose** | To add comments on tasks. |
| **Prerequisites** | Application should be up and running. User should be registered and logged in. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Profile loaded successfully. |
| **Comment:** Test passed successfully. | |

## **5.3.15 Review Task:**

|  |  |
| --- | --- |
| **Test Case ID** | TC-15 |
| **Functional Area/Module** | Review Task |
| **Action to Perform** | 1. User starts the application. 2. User logs in by entering right  credentials. 3. User opens the project. 4. User opens task. 5. User clicks on push button. 6. Finish. |
| **Purpose** | To push task for Leader’s review. |
| **Prerequisites** | Application should be up and running. User should be registered and logged in.  Project and task should be existed. |
| **Test Case Engineer** | Kamran Nazir |
| **Environment** | Windows 10, Chrome browser |
| **Expected Result(s)** | Task is pushed for review successfully. |
| **Comment:** Test passed successfully. | |

**Result:**

|  |  |
| --- | --- |
| **Test Case Number** | **Results** |
| **TC-01** | **Passed** |
| **TC-02** | **Passed** |
| **TC-03** | **Passed** |
| **TC-04** | **Passed** |
| **TC-05** | **Passed** |
| **TC-06** | **Passed** |
| **TC-07** | **Passed** |
| **TC-08** | **Passed** |
| **TC-09** | **Passed** |
| **TC-10** | **Passed** |
| **TC-11** | **Passed** |
| **TC-12** | **Passed** |
| **TC-13** | **Passed** |
| **TC-14** | **Passed** |
| **TC-15** | **Passed** |

# 

# Chapter No. 6

# Conclusion

## **6. Conclusion:**

PROMAN is a web based application for users to manage their projects while working remotely. This application provides best features to manage the normal projects and software projects as well and can estimate cost as well. The Application “PROMAN” is developed by both Group members with a lot of effort and hard work. Beside that we were thinking about deploying our project in real life environment so people can get benefit from it and also we can then find what else is necessary and should be added which is helpful for the people. Because our main purpose is to give people a very friendly environment with a good number of services so they can do their work in a very easy manner.

# Appendix-A

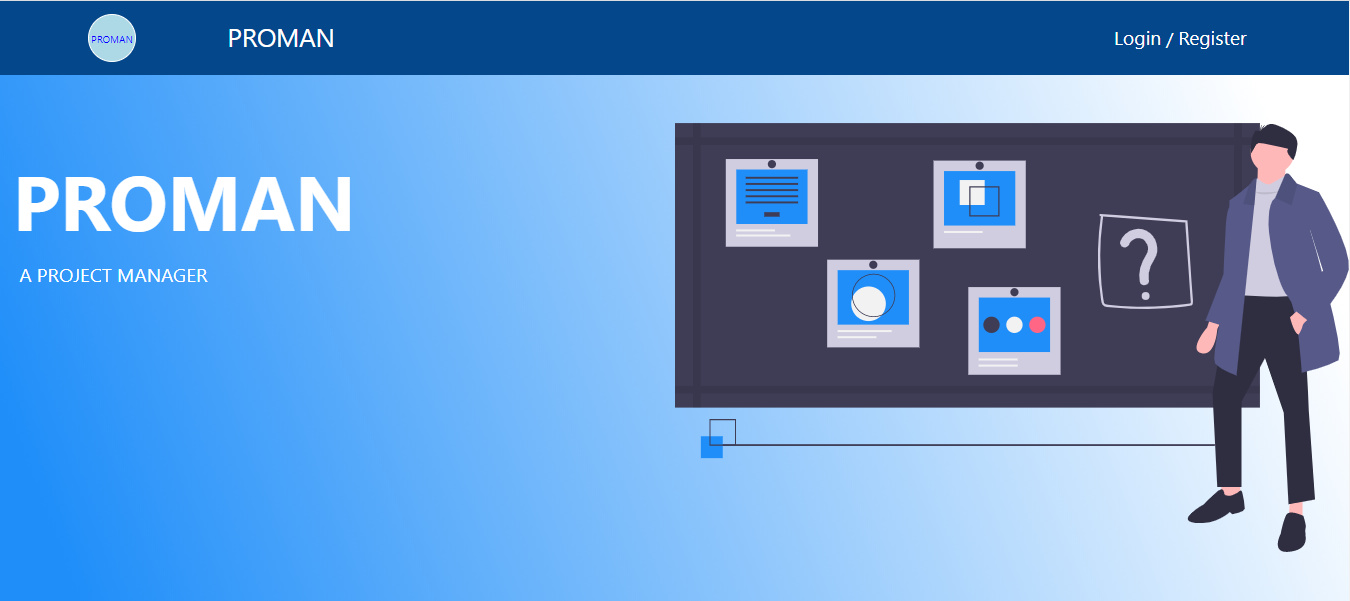
# User Manual

## **A. User manual:**

This user manual or user guide is especially designed for our users so they can use our system easily and efficiently. They can take help from this user manual and can complete their training without any major issue as this manual contains almost each and everything which is necessary for user to use the system. Steps to use our application are as follows:

## **A.1 Public View:**

Main page of our application is Homepage is this what people see when they open our application without signing in.

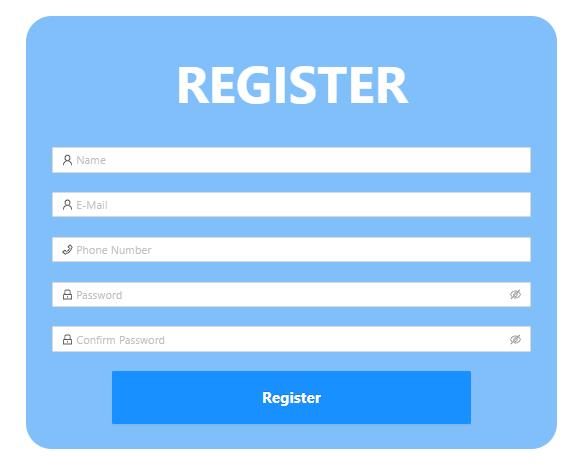


* In homepage, user can find the following options:

1. User can click on register button to get registered in our application.
2. User can press login button and can log into our system using his credentials.

## **A.2 Register:**

If someone is using our application for the very first time, he needs to register himself in our system. For that, he has to enter his credentials and by pressing register button, he will be registered into our system and will be able to use our system.

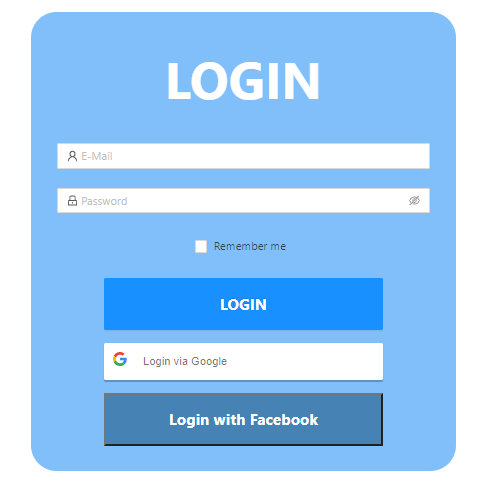


* Do the following procedure to get registered in our application.

1. Fill out the whole information in the above figure and click on register button to get yourself register on our application.
2. Or you can register directly via Google or Facebook by entering your credentials.

## **A.3 Login:**

After the user is registered, he can now enter his credentials to log into our system.



* Do the following procedure for logging in in our application.

1. Enter your credentials and click on login button and you will be redirected to the next page.
2. Or you can login directly via Google or Facebook by entering your credentials.

## **A.4 Add new Project:**

User can create new project using this Add new Project page.

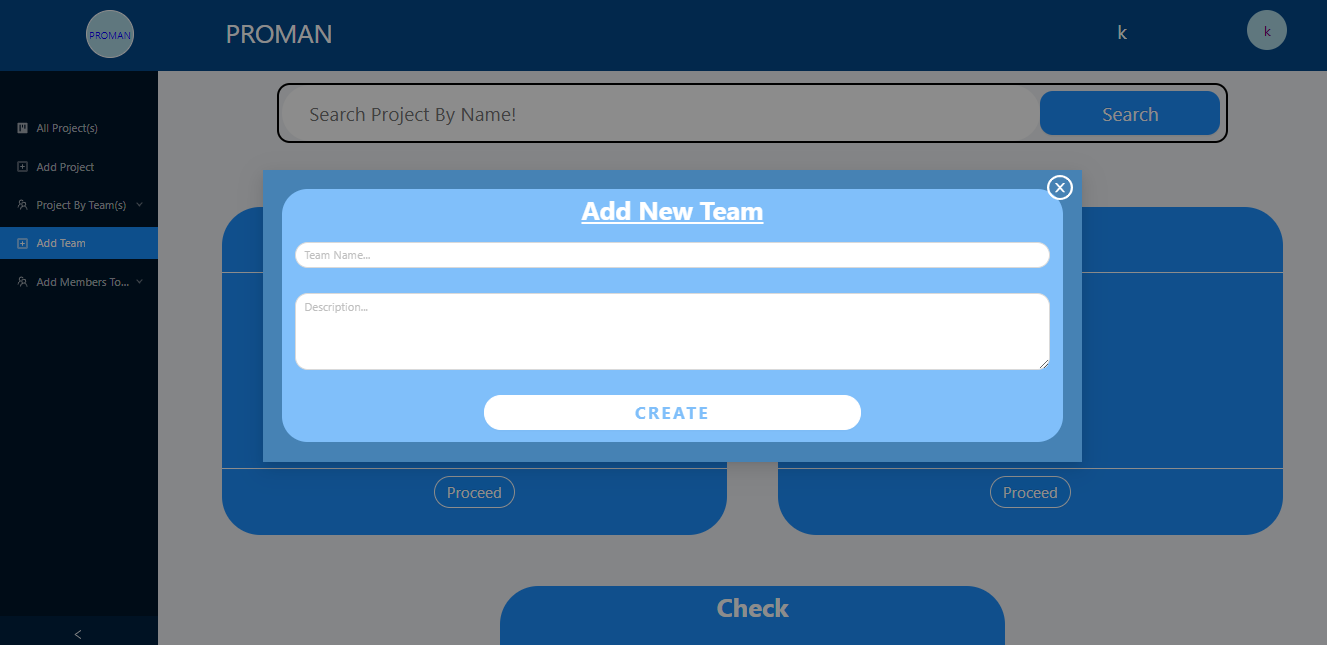


Do the following procedure for Adding project in our application.

1. After logging in you will see this main dashboard page, from there click on add   
   project.
2. Enter details and click on create button and project will be created.

## **A.5 Add new Project:**

User can create new team using this Add new team page.

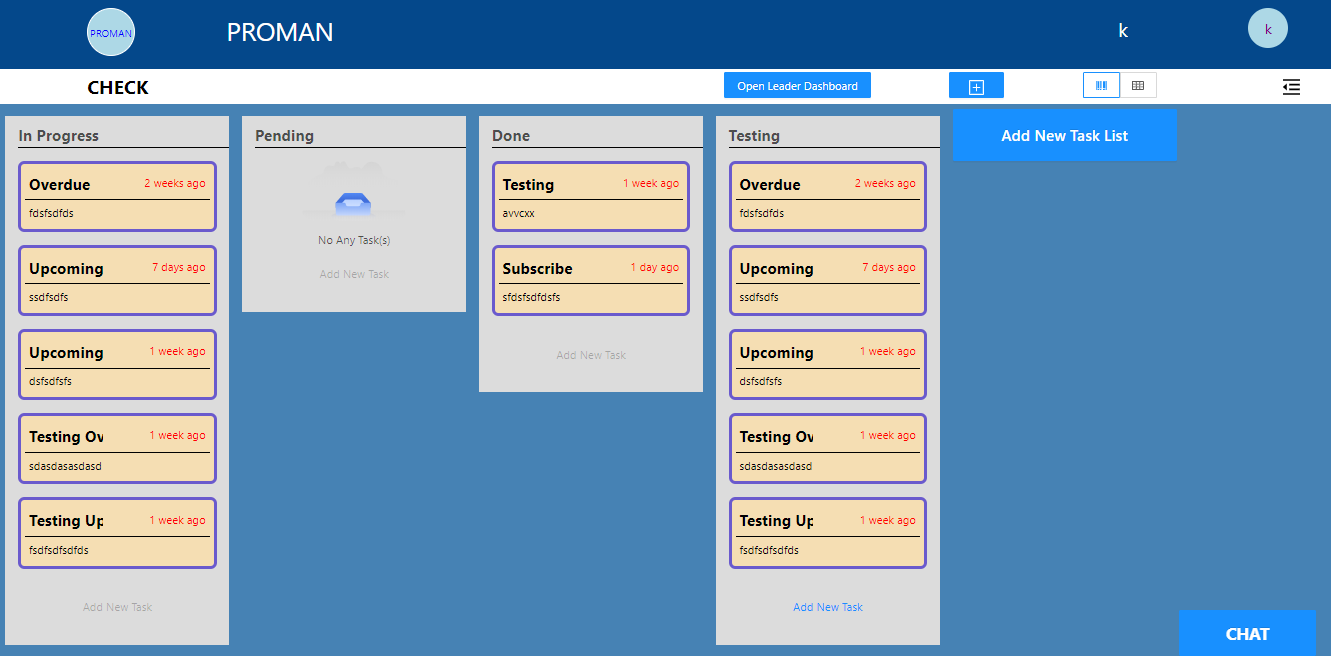


* Do the following procedure for Adding Team in our application.

1. After logging in you will see this main dashboard page, from there click on add   
   team.
2. Enter details and click on create button and new team will be created.

## **A.6 Project Overview:**

User can see the details about project in the following page.

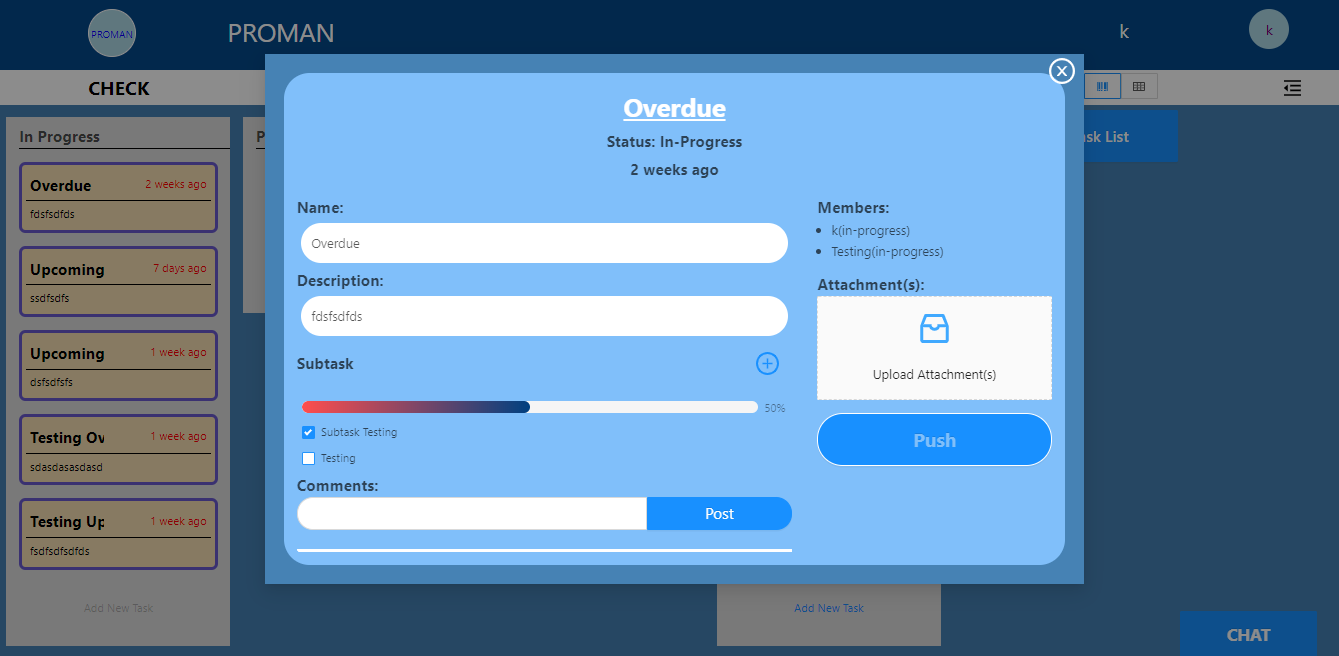


* Do the following procedure to see project information.

1. From dashboard page click on proceed button on the project and you will be   
   redirected to this page where you have all information about project.

## **A.7 Task Overview:**

User can see task detail using following page.

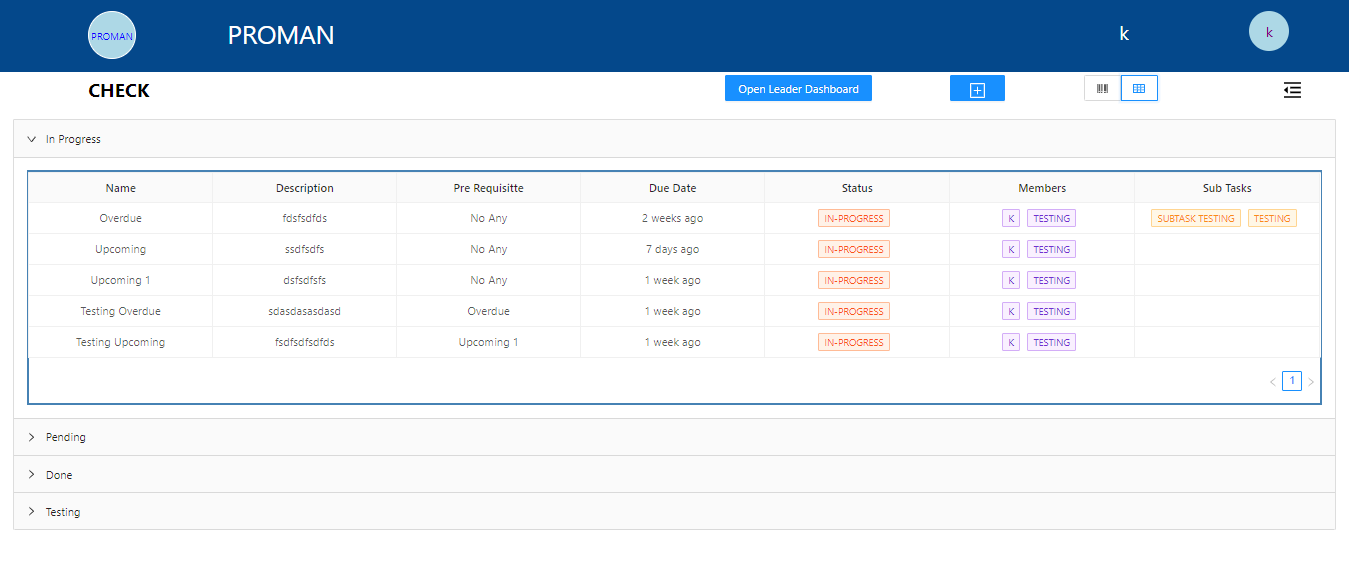


Do the following procedure to see task information.

1. From dashboard page click on proceed button on the project and you will be   
   redirected to this page where you have all information about project, you can click on any task to see its information.

## **A.8 Project overview in table format:**

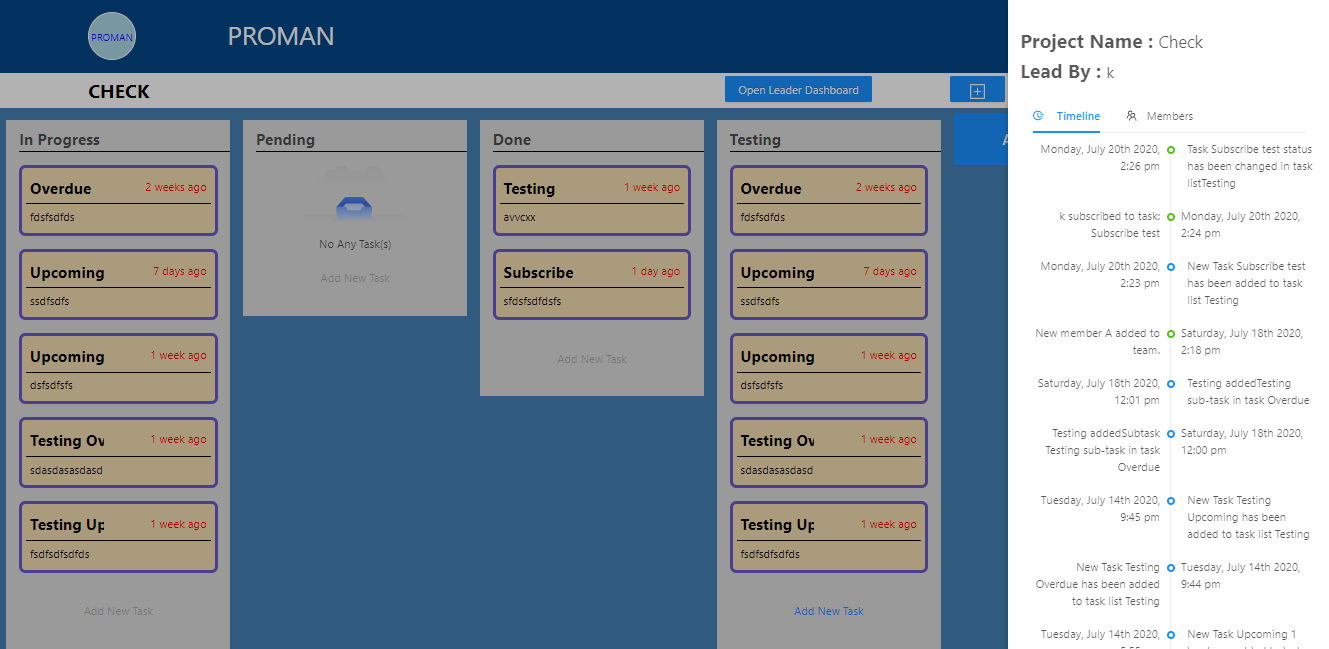
User can view the information about project in the table format as well which is easier to understand.



* Do the following procedure to see the detailed information in table view.

1. From dashboard page click on proceed button on the project and you will be   
   redirected to project information page.
2. From there click on table view button which is right bottom of navbar to see the information in table view.

**A.9 Timeline:**   
 It is just like a notification bar in which User can view the information about what happened to tasks and who did what.



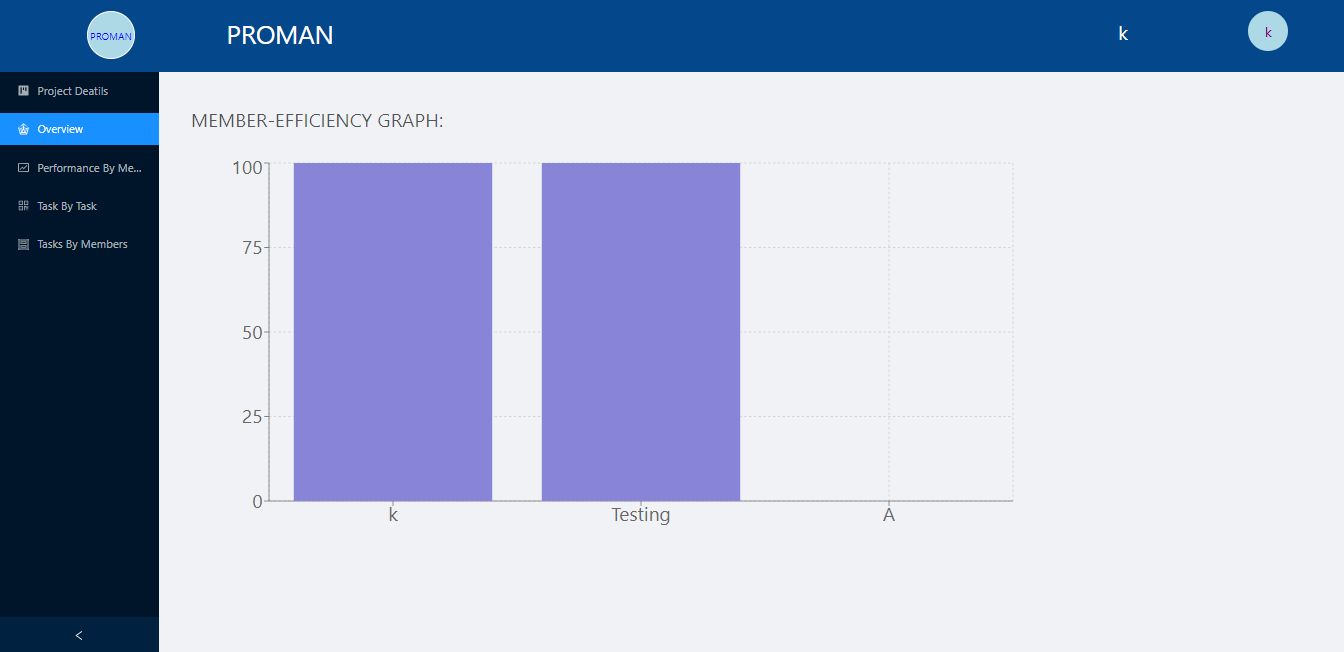
* Do the following procedure to see project information.

1. From dashboard page click on proceed button on the project and you will be   
   redirected to this page where you have all information about project.
2. From there you can click on timeline button which is at right corner below the taskbar to see the recent notifications about project.

## 

## **A.10 Leader Dashboard:**

This is basically an additional tab for Leader where he can get a detailed overview of project.



* Do the following procedure to see project information.

1. From dashboard page click on proceed button on the project and you will be   
   redirected to the project information page.
2. From there you can click leader dashboard button to see this information but for that purpose, you must be the leader of that project.