COMSATS University, Islamabad Pakistan

Fiesta Virtual Office Assistant

By

ADEENA MASOOD CIIT/SP19-BSE-008/ISB

AREEBA NAVEED CIIT/SP19-BSE-020/ISB

Supervisor

Atique Ahmad Zafar

Bachelor of Science in Software Engineering (2019-2023)

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.

COMSATS University, Islamabad Pakistan

Fiesta Virtual Office Assistant

A project presented to
COMSATS University, Islamabad

In partial fulfillment
of the requirement for the degree of

Bachelor of Science in Software Engineering (2019-2023)

By

ADEENA MASOOD **CIIT/SP19-BSE-008/ISB**
AREEBA NAVEED **CIIT/SP19-BSE-020/ISB**

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 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
ADEENA MASOOD	AREEBA NAVEED
---------------------------	-------------------------

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (SE) “Fiesta Virtual Office” was developed by **AREEBA NAVEED (CIIT/SP19-BSE-020)** and **ADEENA MASOOD (CIIT/SP19-BSE-008)** under the supervision of “Atique Ahmad Zafar” and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Software Engineering.

Supervisor

External Examiner

Head of Department

(Department of Computer Science)

Executive Summary

Fiesta Consultants is a content writing agency that has a team of freelance writers scattered around the country. The company has started to work on Search Engine Optimization and Web development recently. As their business is expanding day-to-day, managing projects and staff data had become difficult. Managing such a large number of records is not a one-man's job. Hiring managers for every team and service needs a lot of effort to train the managers on how the business runs, and money to pay off the manager.

Fiesta Virtual Office will help the firm to automate their office work, which was consuming extra cost and effort on team and record management. The Company provides an immense range of services, and its employees work from remote locations. All the employees are connected using a different application.

The biggest targeted Challenges is task allocation and project tracking. Managing large reserves of data manually is a challenging task. Assigning tasks, maintaining a record of these tasks along with the employee performance manually is very hectic. It will serve as a platform for effective communication, task monitoring, and automating the employee's and project's record management.

We proposed a software system that will manage the tasks, notify the employees about their tasks, and maintain the record. It will engage the employees through the application, so they can feel like working in an actual office. It will serve as a platform for effective communication. It will evaluate the employees on their previous tasks and give recommendations to managers to recruit certain employees for the tasks. The system will reduce the overall cost spent on hiring managers for each team and maintaining the records, instead the work will be automated.

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 Sir Atique Ahmad Zafar. Without his personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are grateful to them for their 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 with the values of honesty & hard work.

Student Name 1 Student Name 2

AREEBA NAVEED ADEENA MASOOD

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

Abbreviations

SRS	Software Require Specification
PC	Personal Computer
FR	Functional Requirement
SDS	Software Design Document
UI	User Interface

Table of Contents

List of Figures

1 Introduction

Fiesta Consultants is a content writing agency that has a team of freelance writers scattered around the country. The company has started to work on Search Engine Optimization and Web development recently. As their business is expanding day-to-day, managing projects and staff data had become difficult. Managing such a large number of records is not a one-man's job. Hiring managers for every team and service needs a lot of effort to train the managers on how the business runs, and money to pay off the manager.

The company needs a software system that can manage things for them, smartly and efficiently. It should connect the employees on a single platform, where it is easy for managers to monitor their teams and communicate with them. A system that will always keep an eye on employees' performance and recommend them, for future projects, based on their past performance. The software would be able to tackle the diverse nature of startups the company supports. It will help the manager in project plans, and maintain a record of the daily tasks, they have completed and the word count of each task.

1.1 Vision Statement

For a writing firm, ***who*** wants to use only one platform to communicate with employees and keep a record of the employees, tasks, and projects, ***the*** Fiesta Virtual Office is a Web Application that will help manage teams, create project plans, monitor teams, and projects, and serve as a medium of communication between teams and the management. ***Unlike*** the previous system where they use Gmail, excel and WhatsApp to assign/submit tasks, record activities, and communicate with teams respectively. ***Our product*** will reduce the time they spend on manually computing the project cost, employees will have to monitor only one application rather than monitoring three applications to connect.

1.2 Related System Analysis/Literature Review

Table 1: Relates System Analysis with Fiesta Virtual Office Assistant

Application Name	Weakness	Proposed Project Solution
Jira [1]	• The free version of Jira can only be used by 8 to 9 people. • It is expensive.	• It is a tailor-made solution. • It does record keeping along with task management.
Teams [2]	• It only provides a communication platform. • It does not help track the progress of projects. • It neither evaluate the employees nor gives recommendations.	• It serves as a medium of communication, along with tracking the projects. • It maintains the record of the tasks and projects. • It evaluates the employees from their previous tasks and recommends them for future ones.
Trello	• It is suitable for smaller projects with small teams. • Task dependencies are not shown in Trello.	• It can accommodate big teams with large projects. • It provides Gantt charts and existing templates to facilitate employees.
Asana	• It lacks the time tracking feature. • Only one user can be assigned to a task at any given time.	• It tracks the time of projects for the evaluation of employees.

1.2 Project Deliverables

List down the project deliverables.

Table 2: Project Deliverables

Document	***Implementation***
Scope proposal	-
SRS	-
SDS	40%
Testing and implementation document	60%
Final implementation document	100%

1.3 System Limitations/Constraints

LI-1: Fiesta Virtual Office is a custom solution and shall only be available for Fiesta Consultants LLC.

LI-2: It does not have an integrated mobile application.

1.4 Tools and Technologies

Table 3: Tools and Technologies

	Tools	**Version**	**Rationale**
Tools And Technologies	Visual Studio Code	1.62	IDE
	Adobe Photoshop	CSC 6	Design Work
	Adobe XD	2021	UI design
	MongoDB Compass	1.30.1	Database queries
	Wamp	3.2.6	Server
	Technology	**Version**	**Rationale**
	MongoDB	5.0.6	Database
	Express	4.17.3	Framework for Node
	React.js	17	Front-end Development
	Node.js	16.14.0	Back-end Development

1.5 Relevance to Course Modules

Table 4: Relevance to Course Modules

COURSE	**RELEVANCE**
Web Technologies	It is a web-based platform built in MERN stack
Database Systems	We are using MongoDB as our database
Software Project Management	SPM is used to manage project using MS project and Jira.
Software Requirement Engineering	This course helped to fetch requirements
Intro to Software Engineering	ISE concepts are used in developing SDS document
Programming Fundamentals	It provides basic programming concepts
Human Computer Interaction	It is used in building UI

2 Problem Definition

Our project will help Fiesta firm to automate their office work, which was consuming extra cost and effort on team and record management.

2.1 Problem Statement

In any company, employee management and monitoring is a big challenge, and if the company has remote employees this challenge increases. Monitoring and connecting with employees in remote locations is very important to check if the assigned duties are performed correctly. In addition to assigning tasks, maintaining a record of these tasks along with the employee performance manually is very hectic. There is a high chance that some of the data is missed or unobserved. If this happens, the performance is compromised, which will in return arise a lot of questions about the company's services. Good performance can increase the business, whereas small negligence that affects performance can take the company to an edge.

For Fiesta Consultants, the daily tasks are assigned to the writers on Gmail. They are informed about the tasks through WhatsApp and they have to inform their manager when the task is completed, and also send it back to the manager via email. With that, both managers and the writers have to keep a constant check on two applications. It diverts their attention from the papers they are writing and thus affects the performance of the work. It is very hard for the writers to get their attention back to where they were when they were interrupted. The compromised performance at this point affects the services of the organization, which can give them a financial loss.

The writers also have to maintain a record of the daily tasks they have completed, and the word count of each task. Similarly, the managers have to maintain a list of daily task at their end to keep a record. It is an extra yet crucial activity because it takes extra time to be developed the task record. The pay is generated when the manager compares the company's task record with the writer's task record. This needs to be automated. Moreover, if the manager has news for all the employees, he/she has to notify each team member through their WhatsApp, they need a professional method to handle the notifications. The problem of using WhatsApp for professional purposes is that your personal information is at risk. If a system can get the writer his/her tasks, without getting the personal information of the team on board, it will be good.

2.2 Problem Solution

The company needs a software system that can keep a record of employees and projects. A dashboard for the employees to perform daily work activities. A system where the organization's daily tasks are managed, and their record is maintained. The system can generate reports for the projects and tasks for the manager and the team member.

A system to help managers stay in touch with the employees through a discussion forum for different teams, arrange meetings, and post important alerts. A software system that can serve as a virtual office for Fiesta Consultants, so the employees at the remote location could feel connected and thoroughly monitored. The managers can make project plans and track their project's progress.

2.3 Objectives of the Proposed System

BO-1: Reduce the cost of management by 40%.
BO-2: Reduce the time spent on preparing excel sheets by 60%.
BO-3: Increase the efficient work-time of employees by 20 minutes per day.
BO-4: Time spent notifying the teams about project progress is minimized by 60%.

2.4 Scope

In the Virtual Office, Staff is registered in the system and have their workspaces. They are further categorized and allotted work on the basis of their experience. Managers form teams to accomplish the tasks they are given and further divide and allot subtasks to the teams. Team members get notification of their allotted task and its submission time. Managers approve the task daily after they are submitted on Virtual Office and the next team member is notified about his respective task. Employees can set up alerts or reminders of their deliverables for their convenience. Employees are evaluated after every project's completion on the basis of task complexity and efficiency. A virtual office also recommends employees for particular tasks on the basis of their previous evaluations and ratings. Templates are provided for the employees to help them accomplish the task efficiently and easily. Templates can also be modified and a record of updated versions is saved. Reports are generated of all projects for evaluation. Moreover, the meeting can be scheduled through the virtual office to communicate with the client or with team members through chat, voice call, or video conferencing. The managers can schedule the meetings and start instant meetings. The respective team members will be notified through reminders about the scheduling of meetings. [3]

2.5 Modules

2.5.1 Module 1: Employee Profiling

FE-1: Create and categorize employee profiles.
FE-2: Add, update, view and delete user profiles.
FE-3: Allow employees to request for change of any information.
FE-4: Assign and change role of the employees.

2.5.2 Module 2: Project Template Management

FE-1: Create, update, view, and delete the project templates.
FE-2: Allow manager/admin to use project templates and dynamically add tasks and subtasks into the project templates.
FE-3: Allow manager/admin to save the changes made to the template, maintaining the record of the time and name of the person who made the changes.

2.5.3 2.5.3 Module 3: Project Management

FE-1: Create, update, view, and delete projects.
FE-2: Categorize the project on basis of type, complexity, budget, time, and the workforce required to complete the project.
FE-3: A dashboard maintaining the manager's ongoing projects and using graphs to show the project completion rate.
FE-4: Add or remove employees in the team.
FE-5: Allow manager/admin to allocate and deallocate a task to the team members, add expected word count, and attach related files.
FE-6: Allow team members to view the assigned tasks and submit the files.
FE-7: Keep a record of the time at which the task was assigned and the time at which the task was submitted.
FE-8: Maintain the record of the writer's word count, by reading the attached file.
FE-9: Add the payment into writer's account, when the task is completed and approved by the Manager/Team lead.
FE-10: Compare the actual word count from the expected one, and notify the manager about the deviation.
FE-11: Evaluate the manager at the end of a project.

2.5.4 Module 4: Payment Management

FE-1: Increment the money to the employee's account when a task is completed successfully. FE-2: Maintain pending payments of employees.
FE-3: Remove the money from pending payments, when the employee is paid. FE-4: Maintain a history of payments for each employee.

2.5.5 Module 5: Task Monitoring and Controlling

FE-1: Notify the employees on daily basis about their tasks and the submission time.
FE-2: Notify the managers when a team member submits the task.
FE-4: Generate reports of completed tasks in a specific duration, this duration will be taken from the employees/managers.
FE-5: Locate the tasks assigned to a team member on the calendar and notify them when the deadline is closed.
FE-6: Set a reminder for important deliverables and deadlines.
FE-7: A notification board where the manager/admin can add and remove formal notifications.

2.5.6 Module 6: Employee Evaluation and recommendation

FE-1: System will evaluate the employee's performance on some performance parameters.
FE-2: Graphs and charts represent the employee performance
FE-3: Make recommendations for future tasks on the basis of their previous performance and experience of the task complexity.

2.5.7 Module 7: Communication

FE-1: Communicate with teams through discussion forums and individual chats.
FE-2: Start instant meetings and add people from within the organization.
FE-3: Schedule meetings between teams and allow the addition of some individuals in the meeting later on.
FE-4: Allow video conferencing and screen sharing in meetings.
FE-5: Notify the team about the upcoming meeting schedule.

2.5.8 Module 8: Onboarding

FE-1: Show newly hired employees the training videos relevant to their roles.
FE-2: Display a list of questions at the end of the video session to evaluate its effectiveness.
FE-3: Show the manager the answers to the video session questions.
FE-4: Add or remove training videos.
FE-5: Allow Managers/admin to assign trainees trail tasks.
FE-6: Allow trainees to upload the task.
FE-7: Allow manager/admin to evaluate the trainee's performance on the task.
FE-8: Update the trainee's status to the employee, when he/she complete the trial period successfully.

2.5.9 Module 9: Help

FE-1: AI chat bot can handle user queries to help them understand the system.

3 Requirement Analysis

3.1 User classes and characteristics

Table 5: User classes and characteristics of Fiesta Virtual Office Assistant

User class	Description
Admin	Admin is controller of the entire system. He manages employees profiles and put announcement for important notifications in the system.
Manager	The Manager is the company's employee who manages ongoing projects in the application. He has access to statistics and graphs for employee performance. He can manage the payments of the employees and assign them tasks. He can also communicate with the team members through video conferencing.
Team Lead	Team Lead is the company's employee who manages the teams assigned to them. The team lead will allocate tasks to the employees and evaluate the solution. He can also communicate with the team members through video conferencing.
Writer	Writer is the company's employee who views the task assigned to him and submits the task within the allocated time. He can also view his tasks history, pending payments, and his own performance.
Trainee	Trainee is the new employee who is going through the hiring process, so he can view the training videos and perform the assigned evaluation tasks.

3.2 Requirement Identifying Technique

3.2.1 Use Case Diagram

3.2.1.1 Use Case Diagram for Admin

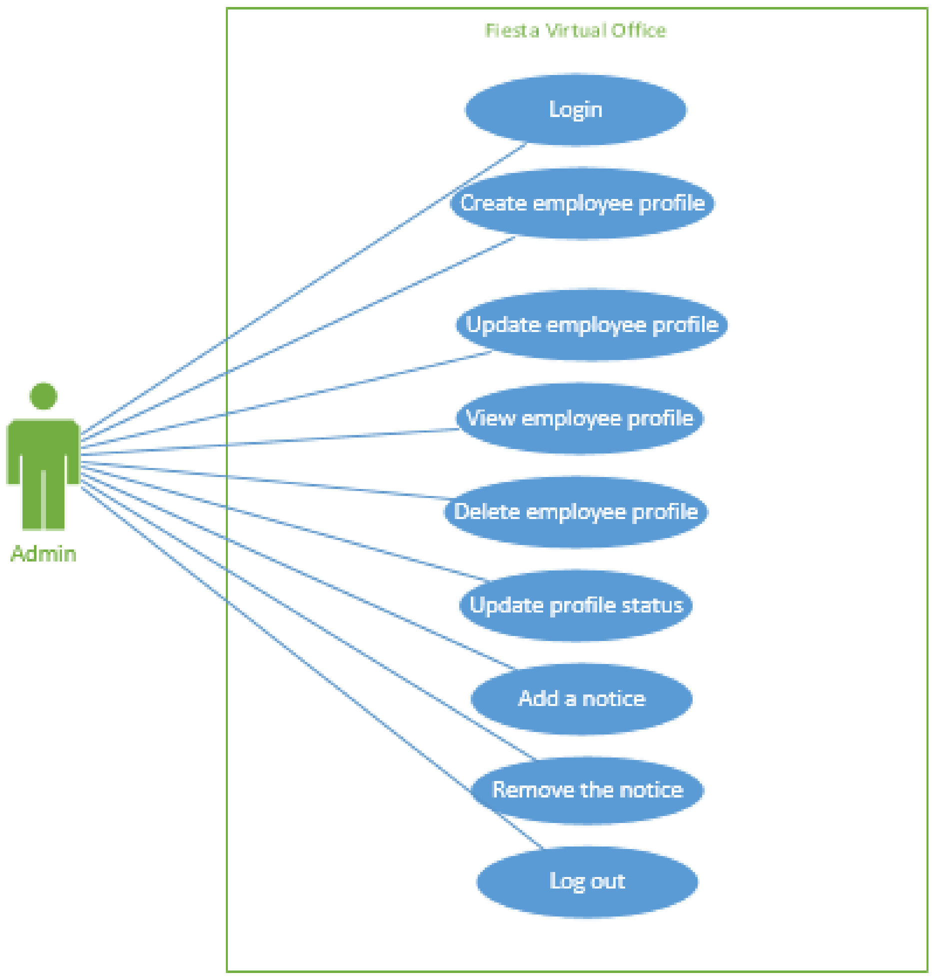


Figure 1: Use case diagram for Admin of Fiesta Virtual Office

3.2.1.2 Use Case Diagram for Writer

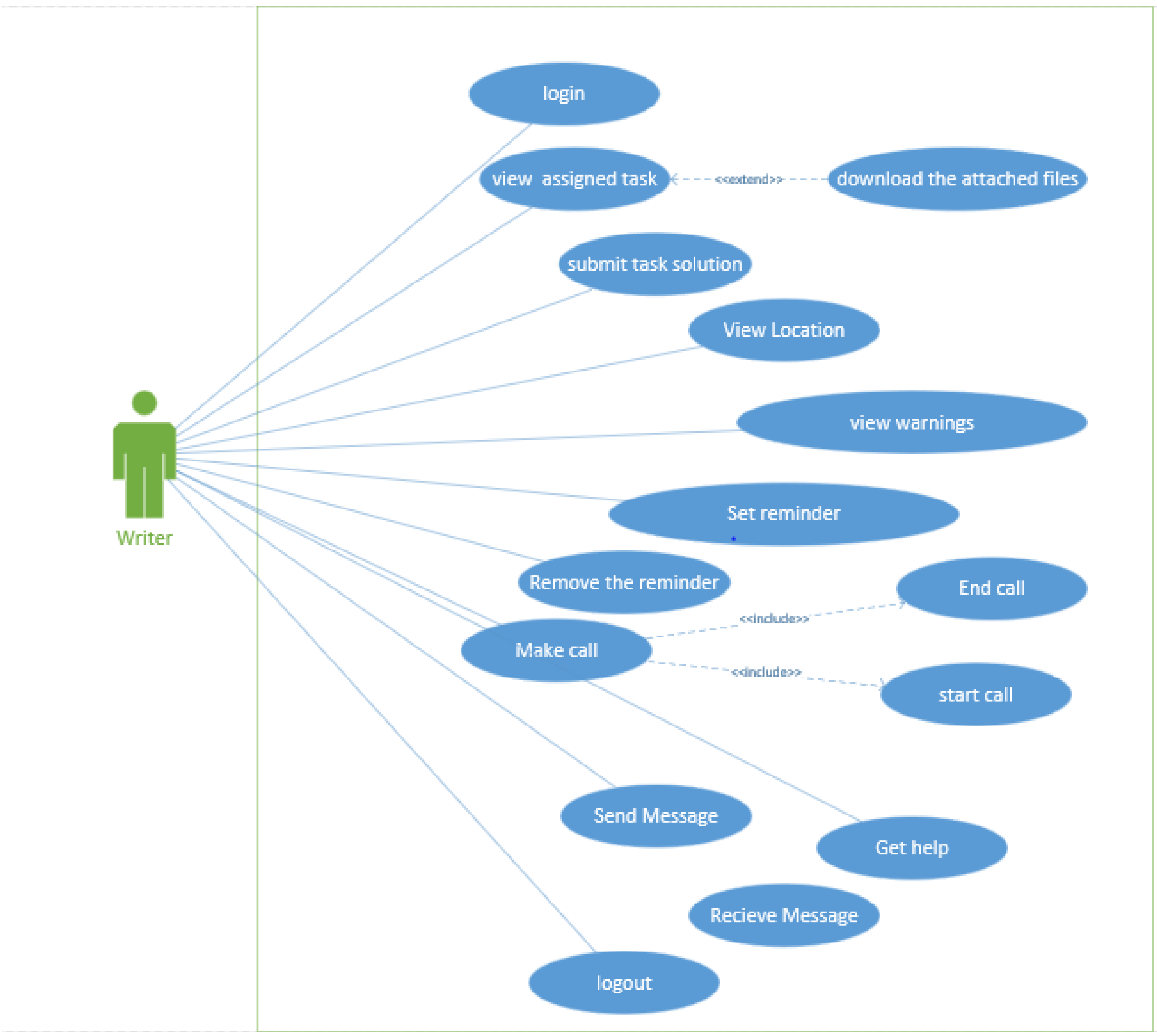


Figure 2: Use case diagram for Writer of Fiesta Virtual Office Assistant

3.2.1.3 Use Case Diagram for Trainee

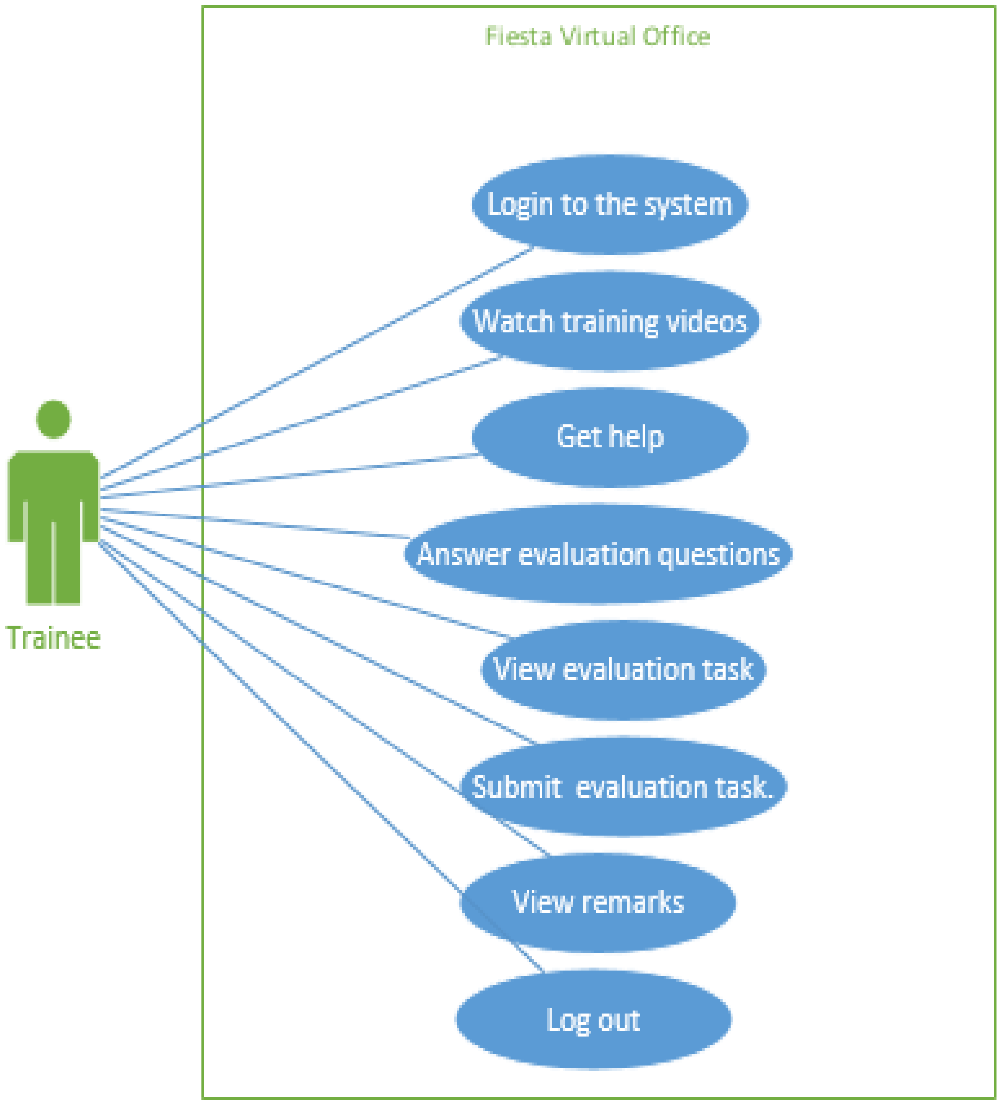


Figure 3: Use case diagram for Trainee of Fiesta Virtual Office Assistant

3.2.1.4 Use Case Diagram for Manager

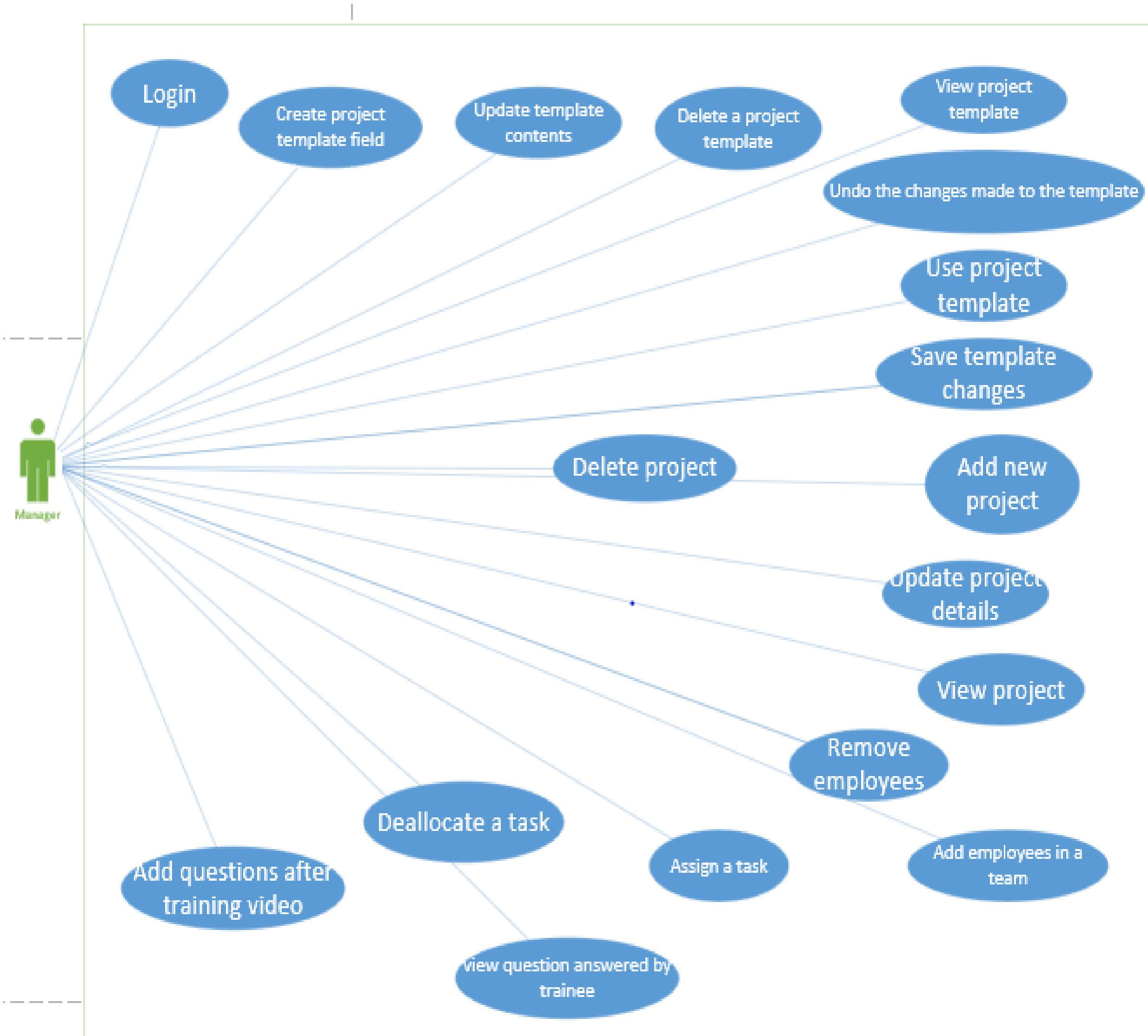


Figure 4: Use case diagram for Manager of Fiesta Virtual Office Assistant - 1

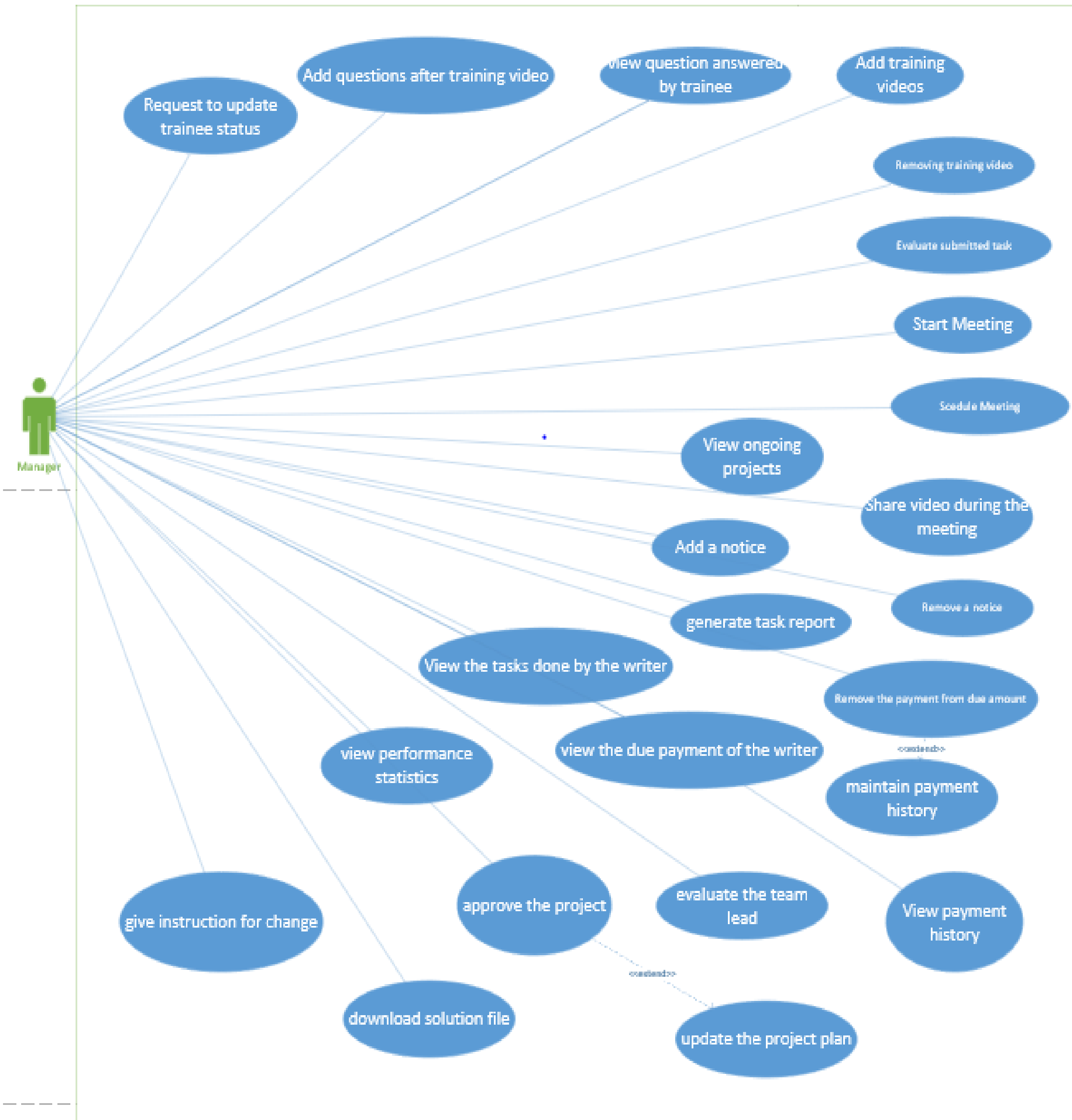


Figure 5: Use case diagram for Manager of Fiesta Virtual Office Assistant – 2

3.2.2 Detailed Use case Diagrams

Following are the detailed tabular usecases of all modules of Fiesta Virtual Office.

3.2.2.1 Module 1: Employee Profiling

The detailed usecases of Employee Profiling are mentioned below.

3.2.2.1.1 Create employee profile

Table 6: Use case for Create Employee Profile

Use case ID:	UC-1.1
Use case Name	Create Employee Profile
Actors	Primary actor: Admin
Description:	The admin fills out the form containing personal information of the employee. He also assign role to the employee. This profile will give employee a username and password through which he can login to the system
Trigger:	The company hired a new employee, who needs to use the application.
Preconditions:	PRE-1. Admin is logged into the system.
Post conditions:	POST-1. Employee profile will be created. POST-2. Record is added into the database
Normal Flow	1. Admin clicks on the employee field on the screen. 2. System displays the list of employees and displays a button to add new employee. 3. Admin clicks on the button and a form appear on the screen. 4. The system ask admin to enter the name, age, email address, phone number, and qualification. 5. The system verifies that the employee profile does not already exists. (See E1) 6. The system ask admin to assign role to the employee, it can be a manager, a team lead, a writer, or a trainee. 7. The system asks admin to enter a username for the employee. 8. The system verifies that the assigned username is unique. (See E2) 9. The system asks admin to enter a password. 10. Admin submit the information to add employee. 11. The system displays the information on the screen as a preview. 12. Admin confirms the information.

Alternative Flows:	**Writer's word count addition** In step 6 of normal flow, 1. If admin select writer as a role for the employee, the system extends the form. 2. The system ask admin to enter writer's daily word count. 3. Return to the step 7 of normal flow. **Change the entered information** In step 11 of normal flow, 1. If the admin finds some typing error in the entered information, he clicks on edit. 2. The system will display the already filled information. 3. The admin corrects the invalid information. 4. Return to step 9 of the normal flow.
Exceptions	**E1. Employee is already registered in the system.** 1. System informs the admin that the employee already exists. 2. If the admin cancels the profile creation, use case terminates. 3. Else the admin is taken back to enter unique information. 4. Return to the step 5 of normal flow. **E2. Username already exists.** 1. System informs the admin that username already exists. 2. The admin changes the username for the employee. 3. Return to the step 8 of normal flow.
Business Rules	BR-1. Only admin can create employee profiles.
Assumptions:	N/A

3.2.2.1.2 Edit Employee Profile

Table 7: Use case for Edit employee profile

Use case ID:	UC-1.2
Use case Name	Edit Employee Profile
Actors	Primary actor: Admin
Description:	The information of the selected employee is displayed in an editable form. The admin can change the desired information on the form.
Trigger:	The employee requests for the change of information.
Preconditions:	PRE-1. Admin is logged into the system.
Post conditions:	POST-1. Employee profile is edited. POST-2. Information is updated in the database.
Normal Flow	1. Admin clicks on the employee field on the screen. 2. The system displays a list of all the employees. 3. Admin selects the specific employee. 4. The system displays the employee's information. 5. Admin click on the button to edit the information. 6. The system generates an editable form which contains the old information. 7. The admin changes the information he/she wants to update. 8. The admin submits the form. 9. The system displays the preview of the information before it is actually submitted. 10. The admin confirms to save the changes.
Alternative Flows:	**Filter search using search bar** In step 3 of normal flow, 1. The admin can use search bar to find the employee. 2. Return to the step 4 of normal flow. **Change the information** In step 9 of normal flow, 1. If the admin finds some typing error in the entered information, he clicks on edit. 2. The system will display the already filled information. 3. The admin corrects the invalid information. 4. Return to step 9 of the normal flow.
Exceptions	N/A
Business Rules	BR-1. Username cannot be edited BR-2. Only admin can edit the profile

3.2.2.1.3 Delete Employee Profile

Table 8: Use case for Delete Employee

Use case ID:	UC-1.3
Use case Name	Delete Employee Profile
Actors	Primary Actor: Admin
Description:	Admin can delete the employee profile from the company's data.
Trigger:	The admin wants to delete employee's profile
Preconditions:	PRE-1. Admin is logged into the system.
Post conditions:	POST-1. Employee profile is deleted.
Normal Flow	1. Admin clicks on the employee field on the screen. 2. The system displays a list of all the employees. 3. Admin select a specific employee. 4. The system displays the employee's information. 5. Admin clicks on the delete button to remove the record. 6. The system ask admin to enter the password. 7. Admin enters the password. 8. System validates admin password. (See E1)
Alternative Flows:	**Filter search using search bar** In step 3 of normal flow, 1. The admin can use search bar to find the employee. 2. Return to the step 4 of normal flow.
Exceptions	**E1. Wrong password** In step 8 of normal flow, 1. If the admin enters the wrong password, the system displays wrong password. 2. The system ask admin to re-enter password in order to delete the profile. 3. The use case terminates
Business Rules	BR-1. Only admin can delete an employee profile.

3.2.2.1.4 Upgrade Profile Status

Table 9: Use case for Upgrade Profile

Use case ID:	UC-1.4
Use case Name	Upgrade Profile Status
Actors	Primary Actor: Admin
Description:	Admin can upgrade the trainee’s profile to writer status so the tasks they performed are added for payment and the writer can get daily tasks.
Trigger:	Manager sends request to admin to upgrade trainee’s profile
Preconditions:	PRE-1. Admin is logged into the system. PRE-2. Admin received a request to upgrade trainee’s profile
Post conditions:	POST-1. Trainee’s profile is upgraded to writer.
Normal Flow	1. Admin opens the employee field on the main screen. 2. Admin opens the requests section. 3. Admin views the upgrade request sent by the manager. 4. Admin clicks on the request. 5. The system displays the trainee’s profile. 6. Admin clicks on upgrade profile to writer. 7. The system changes the role “trainee” to “writer. 8. The system shifts the trail tasks to writer’s task details where tasks with due payment are kept. 9. The system displays a message that the profile is upgraded successfully.
Alternative Flows:	**Open upgrade request from notification** In step 1 of normal flow, 1. The admin opens the notification section. 2. Admin clicks on the upgrade request sent by the manager. 3. Return to the step 3 of normal flow.
Exceptions	N/A
Business Rules	BR-1. Only admin can make changes to employee’s profile.

3.2.2.1.5 Sign in

Table 10: Use case for Sign In

Use case ID:	UC-1.5
Use case Name	Sign in
Actors	Primary Actor: Writer/Manager/Team Lead/Trainee
Description:	The employees can login to the system using the credentials assigned to them by the admin.
Trigger:	The employees need to perform their respective tasks.
Preconditions:	PRE-1. The employee profile is created by the admin.
Post conditions:	POST-2. The employee successfully sign in the system.
Normal Flow	1. Employee enters the username. 2. Employee enters the password 3. The system validates the username and password. (See E1, E2) 4. User is logged in to the system.
Alternative Flows:	N/A
Exceptions	**E1. Wrong username** In step 3 of normal flow, if the employ enters the wrong user name: 1. The system shall display an error message “username doesn’t exist”. 2. Return to step 1 of normal flow. **E2. Wrong password** In the step 3 of normal flow, if the employee enters the wrong password: 1. The system shall display an error message “incorrect password. 2. Return to step 1 of normal flow.
Business Rules	BR-1. Lock the employee profile for 30 minutes after 5 unsuccessful login attempts.

3.2.2.1.6 Change Profile

Table 11: Use case for Change Profile

Use case ID:	UC-1.6
Use case Name	Change Profile
Actors	Primary Actor: Writer/Manager/Team Lead/Trainee
Description:	The employees can change the password for login.
Trigger:	The employees needs to change the password
Preconditions:	PRE-1. The employee profile is created by the admin. PRE-2. The employee is logged into the system.
Post conditions:	POST-2. The employee successfully changes the password.
Normal Flow	1. The employee will go onto view profile button in the system. 2. The employee clicks on change password. 3. The system asks user to enter old password. 4. The employee enters the old password. 5. The system verifies the old password. (See E1) 6. The employee enters new password. 7. The employee confirms new password. 8. The employee clicks on reset password button.
Alternative Flows:	N/A
Exceptions	**E1. Wrong old password** In the step 5 of normal flow, if the employee enters the wrong password: 1. The system shall display an error message “incorrect password. 2. Return to step 3 of normal flow.
Business Rules	N/A

3.2.2.2 Module 2: Project Template Management

The detailed usecases of Project Template Management are mentioned below.

3.2.2.2.1 Create Project template field

Table 12: Use case for Create Project Template

Use case ID	UC-2.1
Use case Name	Create Project template field
Actors	Primary Actor: Manager
Description	This use case allows Manager to create a new template for project to that will help employees in future.
Trigger	Manager wants an addition in templates present for the projects.
Precondition	PRE-1. Manager is logged into the system
Postcondition	POST-1. A new template will be created in the template list.
Normal Flow	1. Manager opens the project template field from dashboard. 2. Manager prompt at the add template to the system. 3. The system displays a form to enter project template details. 4. Manager enters the template name, creation date, type. 3. System prompts the manager to enter details of that template. 6. System saves the information of that template. 7. System prompts a message “Template entered successfully”.
Alternative Flow	N/A
Exception	**E1. No empty field** In step 5 of normal flow, if the user forgets to fill any field: 1. The system prompt user to “enter required information”. 2. Use case resume from step 4 of normal flow.
Business Rules	N/A

3.2.2.2.2 *Update Template contents*

Table 13: Use case for Update template

Use case ID	UC-2.2
Use case Name	Update Template contents
Actors	Primary Actor: Manager
Description	This use case allows manager to update templates details.
Trigger	Manager needs to change the templates existing details.
Precondition	PRE-1. Manager is logged into the system PRE-2. Template is already present in the system.
Postcondition	POST-1. Manager will have successfully edited template name.
Normal Flow	1. Manager clicks on the template field from the main screen. 2. The system shall display the list of all templates. 3. The manager clicks on the template to be modified. 4. The system displays the details of the template. 5. The manager clicks on edit details. 6. The system will display an editable form with pre-field template details. 7. The manager changes the desired information. 8. The manager clicks on save changes. 9. The system prompts a message "Template details modified successfully".
Alternative Flow	**Search project from search bar** In step 2 of normal flow, if the user wants to filter the projects: 1. Manager clicks on the search bar. 2. Manager enters the name or any keyword in the project name. 3. The system will filter the search. 4. Use case resumes from step 3 of normal flow.
Exceptions	N/A
Business Rules	N/A

3.2.2.2.3 Delete a Project Template

Table 14: Use case for Delete Project

Use case ID	UC-2.3
Use case Name	Delete a Project Template
Actors	Primary Actor: Manager
Description	This use case allows manager to delete a project template.
Trigger	Manager wants to delete a project template.
Precondition	PRE-1. Manager is logged into the system. PRE-2. Template is already registered.
Postcondition	POST-1. Template will be deleted from templates list.
Normal Flow	1. Manager clicks on the projects template field on the screen. 2. The system displays a list of all the project templates. 3. Manager selects a specific project template that needs to be deleted. 4. The system displays the project template's information. 5. Manager clicks on the delete button to remove the record. 6. The system prompts a message "Project deleted successfully".
Alternative Flow	**Filter search using search bar** In step 3 of normal flow, 1. The manager can use search bar to find the project template. 2. Return to the step 4 of normal flow.
Exceptions	N/A
Business Rules	N/A

3.2.2.2.4 Use Project Template

Table 15: Use case for Use Project Template

Use case ID	UC-2.4
Use case Name	Use Project Template
Actors	Primary Actor: Manager
Description	This use case allows manager to use an existing project template as a template.
Trigger	Manager clicks on "Use Existing Template" button.
Precondition	PRE-1. Manager is logged into the system PRE-2. Template is already present in system.
Postcondition	POST-1. A new template will be created in the system.
Normal Flow	1. Manager access the template details from dashboard. 2. Manager will tap on edit template contents. 3. Manager generates request to the system to generate a copy of selected template and open it in editable form. 4. Manager will modify the template according to his needs. 5. Manager will save the changes 6. System will prompt a message "New Template is added".
Alternative Flow	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.2.5 Undo the changes

Table 16: Use case for Undo the changes

Use case ID	UC-2.5
Use case Name	Undo the changes
Actors	Primary Actor: Manager
Description	This use case allows manager to Undo the changes made to the template
Trigger	Manager wants to undo the changes made to the template.
Precondition	PRE-1. Manager is logged into the system PRE-2. Template is already registered.
Postcondition	POST-1. The changes made to the project template will be removed
Normal Flow	1. Manager access the template details from dashboard. 2. Manager will tap on edit template contents. 3. Manager will tap at undo changes made to the template. 4. System will fetch the last edited version of the template from the database. 5. Manager will save the changes 6. System will prompt a message “Changes from the Template are removed”.
Alternative Flow	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.3 Module 3: Project Management

The detailed usecases of Project Management are mentioned below.

3.2.2.3.1 Add New Project

Table 17: Use case for Add new Project

Use case ID	UC-3.1
Use case Name	Add New Project
Actors	Primary Actor: Manager
Description	This use case allows Manager to create a new project in the system.
Trigger	A new project is received by the company and needs to be registered in the system.
Precondition	PRE-1. Manager is logged into the system
Postcondition	New project details are entered in the system.
Normal Flow	1. Manager clicks on the projects field from the main screen. 2. Manager clicks on add project. 3. The system displays a form to enter project details. 4. Manager enters the project name, creation date, type, complexity, budget. 5. Manager clicks on submit button. (See E1) 6. System prompts the message "Project added successfully".
Alternative Flow	N/A
Exceptions	**E1. No empty field** In step 5 of normal flow, if the user forgets to fill any field: 1. The system prompt user to "enter required information". 2. Use case resume from step 4 of normal flow.
Business Rules	N/A

3.2.2.3.2 Update Project Details

Table 18: Use case for Update Project Details

Use case ID	UC-3.2
Use case Name	Update Project Details
Actors	Primary Actor: Manager
Description	Manager can update the project details after the data is saved in the system.
Trigger	Project details need to be modified.
Precondition	PRE-1. Manager is logged into the system PRE-2. The project details are already entered in the system,
Postcondition	POST-1. The project details are updated in the database
Normal Flow	1. Manager clicks on the projects field from the main screen. 2. The system shall display the list of all projects. 3. The manager clicks on the project to be modified. 4. The system displays the details of the project. 5. The manager clicks of edit details. 6. The system will display an editable form with pre-field project details. 7. The manager changes the desired information. 8. The manager clicks on save changes. 9. The system prompts a message “Project details modified successfully”.
Alternative Flow	**Search project from search bar** In step 2 of normal flow, if the user wants to filter the projects: 1. Manager clicks on the search bar. 2. Manager enters the name or any keyword in the project name. 3. The system will filter the search. 4. Use case resumes from step 3 of normal flow.
Exceptions	N/A
Business Rules	N/A

3.2.2.3.3 Delete Project

Table 19: Delete Project

Use case ID	UC-3.3
Use case Name	Delete Project
Actors	Primary Actor: Manager
Description	Manager can delete project from company's data.
Trigger	Manager wants to delete project from company's data.
Precondition	PRE-1. Manager is logged into the system
Postcondition	POST-1. Project is deleted successfully.
Normal Flow	1. Manager clicks on the projects field on the screen. 2. The system displays a list of all the projects. 3. Manager selects a specific project. 4. The system displays the project's information. 5. Manager clicks on the delete button to remove the record. 6. The system prompts a message "Project deleted successfully".
Alternative Flow	**Filter search using search bar** In step 3 of normal flow, 1. The manager can use search bar to find the project. 2. Return to the step 4 of normal flow.
Exceptions	N/A
Business Rules	N/A

3.2.2.3.4 View ongoing Project

Table 20: Use case for View Ongoing Projects

Use case ID	UC-3.4
Use case Name	View ongoing Project
Actors	Primary Actor: Manager
Description	This use case allows Manager to view all ongoing project in the system and see their progress.
Trigger	Manager wants to see the details or progress of the project.
Precondition	PRE-1. Manager is logged into the system. PRE-2. Project should be registered in the system. PRE-3. Manager should be registered in the system.
Postcondition	POST-1. Project details gets displayed on the system.
Normal Flow	1. Manager prompts at the view projects. 2. System opens the window that shows the list of all ongoing projects. 3. Manager will select the project to view. 4. Manager will request the system to display the projects details. 5. System will open the desired projects details.
Alternative Flow	N/A
Exceptions	**E1 Project not Found** 2a. In step 2 of the normal flow, if the system cannot find the project 3. System prompts the user “Project not found!”. 4. Use Case terminates.
Business Rules	N/A

3.2.2.3.5 Add employees in a team

Table 21: Use case for Add employees in a team

Use case ID	UC-3.5
Use case Name	Add employees in a team
Actors	Primary Actor: Manager
Description	Manager adds employees in a team to work on a project.
Trigger	Newly entered Project needs a team to start working on it.
Precondition	PRE-1. Manager is logged into the system PRE-2. Project is already created. PRE-3. Project details exist in the database.
Postcondition	Employees are assigned to the team to work.
Normal Flow	1. Manager clicks on team details from home screen. 2. Manager clicks on to "assign team members to the team". 3. Manager views all unassigned employees details and check their profiles. 4. Manager selects the employees suitable for the team. 5. System updates the team members details. 6. System prompts a message that "Employee added to team successfully".
Alternative Flow	**Search Employee from Search Bar** In step 3 of normal flow, if there are no unassigned employees 1. The manager selects the desired employee for the team. 2. The manager can use search bar to find the project. 3. The system will check schedule of that employee. 4. If employees working hours are less than 4hrs per day than system prompts a message to manager that "Employee can be selected" 5. Manager adds the employee into the team. 6. System prompts a message that "Employee added to Projects team successfully".
Exceptions	N/A
Business Rules	N/A

3.2.2.3.6 Assign Task to Team members

Table 22: Use case for Assign task to Team

Use case ID	UC-3.6
Use case Name	Assign Task to Team members
Actors	Primary Actor: Manager
Description	This use case allows Manager to assign tasks to the team members.
Trigger	Project needs to be completed so the team members should be assigned tasks to get the work completed in time.
Precondition	PRE-1. Manager is logged into the system PRE-2. Project is created in system. PRE-3. Team members are allocated to the team.
Postcondition	POST-1. Employees in respective team are assigned with their tasks.
Normal Flow	1. Manager clicks on the view the team. 2. System shows the employees in the respective team. 3. Manager clicks the team member whom they want to assign the task. 4. Manager selects the task which they want to assign to the respective team member. 5. Manager clicks assign task. 6. System prompts a message “Task assigned successfully.”
Alternative Flow	**Filter search using search bar** In step 3 of normal flow, 1. The manager can use search bar to find the task. 2. Return to the step 4 of normal flow.
Exceptions	N/A
Business Rules	N/A

3.2.2.3.7 Deallocate a task to team members

Table 23: Use case for Deallocate a task

Use case ID	UC-3.7
Use case Name	Deallocate a task to team members
Actors	Primary Actor: Manager
Description	Manager thinks that the employee is not working efficiently, or this work is not in his domain so he thinks of deallocating the task of the team member.
Trigger	Task needs to be deallocated.
Precondition	PRE-1. Manager is logged into the system PRE-2. Employee must have been enrolled in a team. PRE-3. Employee must have been assigned a task before.
Postcondition	POST-1. Team members task gets deallocated.
Normal Flow	1. Manager clicks to view teams with their respective employees and tasks allocated. 2. The system displays options to the manager to select team member. 3. The manager clicks at "Delete Task".(E1) 4. System deletes the task already assigned to the team member. 5. Manager clicks to saves the change. 6. System prompts a message "Task is deallocated".
Alternative Flow	N/A
Exceptions	**E1. Employee with no task** 3a. In step 3 of the normal flow, if the manager selects team member that has not been allocated a task before. 1. System will prompt the manager to select the team member again. 2. Manager selects the team member. 3. Use case resumes on step 4 of normal flow.
Business Rules	N/A

3.2.2.3.8 Remove employees from team

Table 24: Use case for Remove employees from team

Use case ID	UC-3.8
Use case Name	Remove employees from team
Actors	Primary Actor: Manager
Description	Manager thinks that the employee is not settling in the team or this team is not in his domain so he thinks of removing the employee from the team.
Trigger	Manager wants to remove the employee from the team.
Precondition	PRE-1. Manager is logged into the system PRE-2. Employee must be enrolled in a team.
Postcondition	POST-1. Employee has been removed from the system.
Normal Flow	1. Manager opens the employees field from main screen. 2. Manager clicks to view teams with their respective employees. 3. The system displays options to the manager to select team member. 4. The manager clicks on "remove employee" to remove the desired employee from team. 5. System removes the employee already assigned to the team. 6. System prompts a message "Employee Removes successfluly"
Alternative Flow	N/A
Exceptions	**E1. Employee not found** 3a. In step 3 of the normal flow, if the manager selects team member that has not been enrolled in a team before. 1. System will prompt the manager to select the employee again. 2. Manager selects the employee. 3. Use case resumes on step 4 of normal flow.
Business Rules	N/A

3.2.2.3.9 View Questions answered

Table 25: Use case for View Questions

Use case ID	UC-3.9
Use case Name	View Questions answered
Actors	Primary Actor: Manager
Description	This use case allows manager to view the questions answered by the trainee.
Trigger	Manager needs to view the questions answered by the trainee.
Precondition	PRE-1. Manager is logged into the system PRE-2. Trainee has submitted the answers.
Postcondition	POST-1. Manager has seen the answers submitted by Trainee.
Normal Flow	1. Manager opens trainee credentials from home screen. 2. Manager clicks on Evaluation results. 3. System shows a list of trainees that have submitted their evaluation tasks. 4. Manager clicks on trainee to see his answers. 5. System opens the form with answered questions.
Alternative Flow	N/A
Exceptions	**E1 No answered Questions Found** In step 2 of normal flow, there might not be any answered questions. For such case. 1. System displays the manager “No answered Questions”. 2. Use case terminates.
Business Rules	N/A

3.2.2.3.10 Add Questions

Table 26: Use case for Add Questions

Use case ID	UC-3.10
Use case Name	Add Questions
Actors	Primary Actor: Manager
Description	Manager adds the questions after training videos to check the understanding of a trainee and evaluate them.
Trigger	Questions needs to be added for trainee employees for their evaluation.
Precondition	PRE-1. Manager is logged into the system. PRE-2. Trainee has completed watching his training videos.
Postcondition	POST-1. Manager has added the questions for the trainees.
Normal Flow	1. Manager opens trainee credentials from home screen. 2. Manager clicks on Add Evaluation tasks. 3. System opens a form for the manager to add questions. 4. Manager enters the questions and saves the form. 5. System prompts a message “Evaluation Task added successfully”.
Alternative Flow	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.4 Module 4: Payment Management

The detailed usecases of Payment Management are mentioned below.

3.2.2.4.1 Remove the payment from due amount

Table 27: Use case for Remove Payments

Use case ID	UC-4.1
Use case Name	Remove the payment from due amount
Actors	Primary Actor: Manager
Description	Manager removes the due payment from history after the payment has been made to the employees.
Trigger	Payment has been made to the employee.
Precondition	PRE-1. Payment is made to the employee. PRE-2. Employee’s payment data is stored in system.
Postcondition	POST-1. Due payments are removed from the payment history.
Normal Flow	1. Manager clicks the payment tab from home screen. 2. Manager clicks on the Due payments tab. 3. System opens a list of employees with their payments due. 4. Manager removes the payments that have been made. 5. Manager saves the information.
Alternative Flow	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.5 Module 5: Task Monitoring and Controlling

The detailed usecases of Task Monitoring and Controlling are mentioned below.

3.2.2.5.1 Add a Notice

Table 28: Use case for Add Notice

Use case ID:	UC-5.1
Use case Name	Add a notice
Actors	Primary Actor: Admin/Manager
Description:	If there is an important information or announcement that needs to be understood by all the employees, admin adds it on the notice board. This noticeboard is visible to all the employees.
Trigger:	The admin/manager wants to an important announcement.
Preconditions:	PRE-1. Admin/Manager is logged into the system.
Post conditions:	POST-2. Announcement is visible on the notice board.
Normal Flow	1. Admin/Manager clicks on the noticeboard option from the home menu. 2. The system takes admin to the notice board preview. 3. Admin/Manager clicks on add new notice option. 4. The system displays a form. 5. Admin/Manager enters header for the announcement. (See E1) 6. Admin/Manager enters the detail of the announcement. 7. Admin/Manager press the submit button. 8. The new notice is visible on the notice board.
Alternative Flows:	N/A
Exceptions	**E1. Header Field is empty** In the step 5 of normal flow, if the Manager/Admin leaves the header empty: 1. The system shall display an error message “Header cannot be empty”. 2. Return to step 4 of normal flow.
Business Rules	N/A

3.2.2.5.2 Remove a Notice

Table 29: Use case for Remove Notice

Use case ID:	UC-5.2
Use case Name	Remove a notice
Actors	Primary Actor: Admin/Manager
Description:	If the information is out-to-date the admin/manager can remove it from the noticeboard
Trigger:	The admin/manager wants to remove a notice from the noticeboard.
Preconditions:	PRE-1. Admin/Manager is logged into the system.
Post conditions:	POST-2. Announcement is removed from the notice board.
Normal Flow	1. Admin/Manager clicks on the noticeboard option from the home menu. 2. The system takes admin to the notice board preview. 3. The system displays a delete button next to the each announcement. 4. Admin/Manager clicks on the delete button of the announcement that needs to be removed. 5. The system displays a confirmation screen on the window. 6. Admin/Manager confirms the deletion of announcement. 7. The notice is removed from the noticeboard.
Alternative Flows:	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.5.3 Add Reminder on Calendar

Table 30: Use case for Add Reminder

Use case ID:	UC-5.3
Use case Name	Add reminder on calendar
Actors	Primary Actor: Writer/Manager
Description:	The writer/manager can set a reminder on the calendar for an important deliverable.
Trigger:	The trainee clicks on “attach solution file”.
Preconditions:	PRE-1. Manager/Writer is logged into the system.
Post conditions:	POST-1. Reminder is set on the calendar. POST-2. Writer/Manager is notified on the day.
Normal Flow	1. Writer/Manager open the calendar from the dashboard. 2. The system displays the calendar. 3. Writer/Manager clicks on set a reminder. 4. The system displays a form to set a reminder. 5. Writer/Manager enters the name and description for the reminder. 6. Writer/Manager sets a date on which the reminder is set. (See E1) 7. The system displays a message that reminder is set.
Alternative Flows:	N/A
Exceptions	**E1. Not a valid date** In step 6 of normal flow, if the selected date had pasted already: 1. The system will display an error message to select a valid date. 2. Writer/Manager will select a valid date. 3. Return to step 7 of normal flow.
Business Rules	N/A

3.2.2.5.4 Remove Reminder

Table 31: Use case for Remove Reminder

Use case ID:	UC-5.4
Use case Name	Remove reminder
Actors	Primary Actor: Writer/Manager
Description:	The writer/manager can remove the reminder from the calendar.
Trigger:	Writer/Manager clicks on the reminder.
Preconditions:	PRE-1. Manager/Writer is logged into the system. PRE-2. Reminder was already set.
Post conditions:	POST-1. Reminder is removed from the calendar.
Normal Flow	1. Writer/Manager open the calendar from the dashboard. 2. The system displays the calendar and the marked dates. 3. Writer/Manager clicks on date when the reminder was set. 4. The system displays details of the reminder. 5. Writer/Manager click on delete reminder. 6. The system displays a confirmation message. 7. Writer/Manager confirm the changes. 8. The system removes the reminder from the calendar.
Alternative Flows:	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.5.5 View Assigned Tasks

Table 32: Use case for View Assigned Tasks

Use case ID:	UC-5.5
Use case Name	View assigned task
Actors	Primary Actor: Writer
Description:	The trainee views the task assigned to him/her by the manager.
Trigger:	The trainee clicks on the task on the task window.
Preconditions:	PRE-1. Writer is logged into the system. PRE-2. Manager has assigned trainee an evaluation task. PRE-3. Writer is notified about the assignment.
Post conditions:	POST-1. Writer has viewed the evaluation task. POST-2. Writer start working on the assigned task.
Normal Flow	1. The system displays the evaluation task on the main screen. 2. The writer clicks on the task. 2. The system displays the task description, allocated time, required word count, and the attached files. 3. The writer reads the task description. 4. The writer downloads the attached file.
Alternative Flows:	**Open from notification** In step 1 of normal flow, 1. Trainee opens the notification. 2. The system displays the word count, and submission time of the task along with the task name. 3. Return to step 3 of normal flow.
Exceptions	N/A.
Business Rules	BR-1. Trainee task evaluation is displayed in his performance statistics.

3.2.2.5.6 Submit task solution

Table 33: Use case for Submit task solution

Use case ID:	UC-5.6
Use case Name	Submit task solution
Actors	Primary Actor: Writer
Description:	The writer has completed the assigned task and now he wants to submit the task solution.
Trigger:	The writer clicks on “attach solution file”.
Preconditions:	PRE-1. Writer is logged into the system. PRE-2. Writer viewed the assigned task. PRE-3. Manager assigned trainee an evaluation task.
Post conditions:	POST-1. Trainee submits the evaluation task. POST-2. The solution file is sent to the Manger/Team Lead.
Normal Flow	1. The writer select the solution file from the system. 2. The writer submits the file by clicking on submit button. (See E1) 3. The system reads the file and calculate its word count. 4. The system checks the difference between required word count and actual word count. 5. The system sends word count deviation to the manager. 6. The system sends the solution file to the manager.
Alternative Flows:	**Select multiple files** In step 1 of normal flow, 1. The writer can also select multiple files. 1. Return to step 5 of normal flow.
Exceptions	**E1. No attached file** In step 2 of normal flow, if the writer forgets to attach the file: 1. The system displays an error message that “No file is selected”. 2. Return to step 1 of normal flow.
Business Rules	BR-1. The submission is allowed even after the allocated time has passed. BR-2. The system will not accept a file greater than 15 MB.

3.2.2.6 Module 6: Employee Evaluation and recommendation

The detailed usecases of Employee Evaluation and recommendation are mentioned below.

3.2.2.6.1 Evaluate team lead

Table 34: Use case for Evaluate teamlead

Use case ID	UC-6.1
Use case Name	Evaluate team lead
Actors	Primary Actor: Manager
Description	Manager evaluates the team leaders to ensure that the teams are doing fine.
Trigger	Team lead needs to be evaluated.
Precondition	PRE-1. Manager is logged into the system.
Postcondition	POST-1. Manager has evaluated the team lead.
Normal Flow	1. Manager access the team leads profile from home screen. 2. Manager clicks on the performance charts. 3. System shows a window with pie charts that shows the performance of the team lead based on their teams work.
Alternative Flow	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.6.2 Evaluate Manager

Table 35: Use case for Evaluate Manager

Use case ID	UC-6.1
Use case Name	Evaluate Manager
Actors	Primary Actor: Manager
Description	Admin evaluates the managers to ensure that the projects are going fine.
Trigger	Team lead needs to be evaluated.
Precondition	PRE-1. Admin is logged into the system.
Postcondition	POST-1. Admin has evaluated the team lead.
Normal Flow	1. Admin access the team leads profile from home screen. 2. Admin clicks on the performance charts. 3. System shows a window with pie charts that shows the performance of the manager based on their teams work.
Alternative Flow	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.7 Module 7: Communication

The detailed usecases of Communication are mentioned below.

3.2.2.7.1 Start Meeting

Table 36: Use case for Start Meeting

Use case ID	UC-7.1
Use case Name	Start Meeting
Actors	Primary Actor: Manager
Description	This use case describes how manager can start a meeting where he can communicate with the employees.
Trigger	Manager wants to do a meeting with employees.
Precondition	PRE-1. Manager is logged into the system
Postcondition	POST-1. Manager will successfully be able to start a meeting
Normal Flow	1.Manager access the meeting tab from home screen. 2.Manager clicks on the start meeting tab. 3. System appears a window for manager to enter Date and Time for the meeting. 3.System will initiate a meeting where other employees can join in for the meeting. 4.Employees will be sent notification showing “The meeting has started”.
Alternative Flow	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.7.2 Schedule Meeting

Table 37: Use case for Schedule Meetings

Use case ID	UC-7.2
Use case Name	Schedule Meeting
Actors	Primary Actor: Manager
Description	This use case describes how manager can schedule a meeting where he can communicate with the employees.
Trigger	Manager wants to schedule a meeting with employees for future.
Precondition	PRE-1. Manager is logged into the system
Postcondition	POST-1. Manager will successfully be able to schedule a meeting.
Normal Flow	1. Manager access the meeting tab from homescreen. 2. Manager clicks on the schedule a meeting tab. 3. System will prompt a window asking for time and date for the meeting. 4. System will initiate a meeting where other employees can join in for the meeting. 5. Employees will be sent notification showing “The meeting has started”.
Alternative Flow	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.7.3 Join Meeting

Table 38: Use case for Join meetings

Use case ID:	UC-7.3
Use case Name	Join meeting
Actors	Primary Actor: Writers
Description:	The writer can join the meeting with the manager or team lead.
Trigger:	The writer clicks on join meeting button.
Preconditions:	PRE-1. The meeting is scheduled. PRE-2: The meeting is started
Post conditions:	POST-2. Writer joins the meeting
Normal Flow	1. A notification will appear on the notification panel of the writer that a meeting has started. 2. The writer clicks on the notification. 3. The system will show a window displaying meeting details. 4. The writer will click on join meeting to join the call.
Alternative Flows:	N/A
Exceptions	N/A
Business Rules	N/A

3.2.2.7.4 Message in Teams

Table 39: Use case for Message in Teams

Use case ID:	UC-7.4
Use case Name	Message in teams
Actors	Primary Actor: Writer/ Team Lead/ Manager
Description:	The employees can write a message in the text area to send it to their team.
Trigger:	The employee wants to send a direct message.
Preconditions:	PRE-1. Employee is logged into the system.
Post conditions:	POST-1. Employee has sent the message. POST-2. The team members receive the message.
Normal Flow	1. The employee opens the team chat. 2. The employee writes a message in the text area. 3. The employee can attach any relevant file or photo. 4. The employee clicks on the send button. 5. The message is sent to all team members. 6. The system will send notification to the team members.
Alternative Flows:	N/A
Exceptions	N/A
Business Rules	BR-1. The employees will not have individual chats.

3.2.2.8 Module 8: Onboarding

The detailed usecases of Onboarding are mentioned below.

3.2.2.8.1 Watch training videos

Table 40: Use case for Watch training videos

Use case ID:	UC-8.1
Use case Name	Watch training videos
Actors	Primary Actor: Trainee
Description:	The trainee can see some videos that are necessary for him/her to start working in the company.
Trigger:	The trainee need to watch the training video.
Preconditions:	PRE-1. Trainee is logged into the system.
Post conditions:	POST-2. Trainee has watched the training videos
Normal Flow	1. Trainee will select watch training videos from the dashboard. 2. The system will display him only the videos manager selected for the job requirement. 3. The trainee will open the link to the video. 4. The system will play the video through YouTube. 5. Trainee can watch the video.
Alternative Flows:	N/A
Exceptions	N/A
Business Rules	BR-1. After watching the video, trainee must perform a quiz to validate his understanding.

3.2.2.8.2 Answer Evaluation Questions

Table 41: Use case for Answer questions

Use case ID:	UC-8.2
Use case Name	Answer evaluation question
Actors	Primary Actor: Trainee Secondary Actor: Manager
Description:	After watching the training videos, the trainee evaluates his understanding with the help of a short quiz.
Trigger:	The trainee clicks on “Take quiz button” after watching the video.
Preconditions:	PRE-1. Trainee is logged into the system. PRE-2. Trainee has watched the training videos.
Post conditions:	POST-1. Trainee has taken the quiz. POST-2. Trainee views his quiz score. POST-3. The quiz results are send to the manager.
Normal Flow	2. The system displays a form containing question and MCQ’s based on the requirement of the video. (See E1) 2. The trainee answers the questions. 3. The trainee submits the form. 4. The system validates the answers. 5. The system calculates the MCQ’s marks and display the score on the screen. 6. The system sends the answers to the manager.
Alternative Flows:	N/A
Exceptions	**E1. Time up** In step 1 of normal flow, when the allocated time for the quiz is finished: 1. The system displays a message “Time is up”. 2. The use case terminates.
Business Rules	BR-1. Trainee can take quiz only once. BR-2. The time field is open, manager can set in on their own.
Assumptions:	Trainee has watched the video completely.

3.2.2.8.3 View Evaluation Task

Table 42: Use case for View Evaluation questions

Use case ID:	UC-8.3
Use case Name	View evaluation task
Actors	Primary Actor: Trainee
Description:	The trainee views the evaluation task assigned to him by the manager.
Trigger:	The trainee clicks on the task on the tasks window.
Preconditions:	PRE-1. Trainee is logged into the system. PRE-2. Manager has assigned trainee an evaluation task. PRE-3. Trainee is notified about the assignment.
Post conditions:	POST-1. Trainee has viewed the evaluation task. POST-2. Trainee start working on the assigned task.
Normal Flow	5. Trainee opens the notification. 6. The system displays the task description, allocated time, required word count, and the attached files. 7. The trainee reads the task description. 8. The trainee downloads the attached file.
Alternative Flows:	In step 1 of normal flow, 1. The system displays the evaluation task on the main screen. 2. The system displays the word count, and submission time of the task along with the task name. 3. Return to step 3 of normal flow.
Exceptions	N/A.
Business Rules	BR-1. Trainee task evaluation is displayed in his performance statistics.

3.2.2.8.3 Submit Evaluation Task

Table 43: Use case for Submit task solutions

Use case ID:	UC-8.4
Use case Name	Submit evaluation task
Actors	Primary Actor: Trainee
Description:	The trainee has completed the task and now he wants to submit the task solution.
Trigger:	The trainee clicks on “attach solution file”.
Preconditions:	PRE-1. Trainee is logged into the system. PRE-2. Trainee viewed the assigned task. PRE-3. Manager assigned trainee an evaluation task.
Post conditions:	POST-1. Trainee submits the evaluation task. POST-2. The solution file is sent to the Manger/Team Lead.
Normal Flow	7. The trainee selectss the solution file from the system. 8. The trainee submits the file by clicking on submit button. (See E1) 9. The system reads the file and calculate its word count. 10. The system checks the difference between required word count and actual word count. 11. The system sends word count deviation to the manager. 12. The system sends the solution file to the manager.
Alternative Flows:	**Select multiple files** In step 1 of normal flow, 3. The trainee can also select multiple files. 1. Return to step 5 of normal flow.
Exceptions	**E1. No attached file** In step 2 of normal flow, if the trainee forgets to attach the file: 1. The system displays an error message that “No file is selected”. 2. Return to step 1 of normal flow.
Business Rules	BR-1.The submission is allowed even after the allocated time has passed. BR-2. The system will not accept a file greater than 15 MB.

3.2.2.9 Module 9: Help

The detailed usecases of Help are mentioned below.

3.2.2.9.1 Get Help

Table 44: Use case for Get Help

Use case ID:	UC-9
Use case Name	Get Help
Actors	Primary Actor: Trainee/Manager/Writer/Team Lead
Description:	The employees new to the system can get help form the AI chat bot to understand the system.
Trigger:	The trainee clicks on “?” icon on the window
Preconditions:	PRE-1. Employee is logged into the system.
Post conditions:	POST-1. Employee gets the answer from the chat bot.
Normal Flow	1. The “?” expands into a small chat window. 2. The employee enters the query in the text area. 3. The system responds with answer to the query. 4. The employee closes the window.
Alternative Flows:	N/A
Exceptions	N/A
Business Rules	BR-1. The submission is allowed even after the allocated time has passed. BR-2. The system will not accept a file greater than 15 MB.

3.3 Functional Requirements

3.3.1 Functional Requirement for Module 1: Employee Profiling

Following are the Functional Requirements for Employee Profiling.

3.3.1.1 Create Profile Personal Information

Table 45: Functional Requirement for Create Personal Profile

Identifier	FR-1.1
Title	Create Profile Personal Information
Requirement	The system shall ask admin to enter name, age, email address, qualification, and contact number in the profile creation.
Source	Student
Rationale	The company wants to have the record of its employee.
Business Rule	BR-1. Only admin can create the employee profiles.
Dependencies	N/A
Priority	High

3.3.1.2 Personal Information Verification

Table 46: Functional Requirement for Personal Profile Verification

Identifier	FR-1.2
Title	Personal Information Verification
Requirement	When the admin enters the personal information of the employee, the system shall check the full name and phone number
Source	Student
Rationale	It will help the admin not to register a user twice
Business Rule	BR-2. The system shall contain unique record of employee
Dependencies	FR-1.1
Priority	High

3.3.1.3 Personal Information Verification Failed

Table 47: Functional Requirement for Personal Profile Verification failed

Identifier	FR-1.3
Title	Personal Information Verification Failed
Requirement	If the personal information of an employee already exists in the system, the system shall display “Record already exists”.
Source	Student
Rationale	It verifies that the system has unique record of employees
Business Rule	The full name and phone number should not match with any of the record in the database.
Dependencies	FR-1.1, FR-1.2
Priority	High

3.3.1.4 Enter Employee Role

Table 48: Functional Requirement for Enter employee role

Identifier	FR-1.4
Title	Enter Employee Role
Requirement	The system shall ask the admin to select a role for the employee.
Source	Student
Rationale	The employees have different roles.
Business Rule	The employee can be a manager, team lead, writer, trainee
Dependencies	FR-1.1, FR-1.2, FR-1.3
Priority	High

3.3.1.5 Employee Role Writer

Table 49: Functional Requirement for employee role writer

Identifier	FR-1.5
Title	Employee Role Writer
Requirement	If the selected role for the employee is writer, the system will ask the admin to enter word count for the employee.
Source	Client
Rationale	The writer word count is necessary for daily task allocation.
Business Rule	The writers in the company have daily word count which must be completed within working hours
Dependencies	FR-1.4
Priority	High

3.3.1.6 Assign Login Credentials

Table 50: Functional Requirement for Assign Credentials

Identifier	FR-1.6
Title	Assign Login Credentials
Requirement	The system shall ask admin to enter the username and password for the new employee.
Source	Student
Rationale	Employee requires login credentials to use the system
Business Rule	Every employee should have a unique username.
Dependencies	FR-1.2, FR-1.4
Priority	High

3.3.1.7 Log in

Table 51: Functional Requirement for Login

Identifier	FR-1.7
Title	Log in
Requirement	The employee can enter the assigned username and password to log into the system
Source	Student
Rationale	The registered employees can access the system to perform their tasks.
Business Rule	Only registered employees can access the system
Dependencies	FR-1.6
Priority	High

3.3.1.8 Log in verification

Table 52: Functional Requirement for Login verification

Identifier	FR-1.8
Title	Log in verification
Requirement	The system verifies the username and password entered by the employee before he/she is given system access.
Source	Student
Rationale	The registered employees can access the system to perform their tasks.
Business Rule	N/A
Dependencies	FR-1.6, FR-1.8
Priority	High

3.3.1.9 Log in Verification Failed

Table 53: Functional Requirement for Login verification failed

Identifier	FR-1.9
Title	Log in Verification Failed
Requirement	If the employee enters wrong username or password, the system will display “Invalid login, Try Again”
Source	Student
Rationale	To only login authorized people into the system.
Business Rule	N/A
Dependencies	FR-1.6, FR-1.7, FR-1.8
Priority	High

3.3.1.10 Change Password

Table 54: Functional Requirement for Change Password

Identifier	FR-1.10
Title	Change Password
Requirement	The system shall allow employees to change the password of their account.
Source	Student
Rationale	It will allow them to take security measure if they feel someone knows their password.
Business Rule	N/A
Dependencies	FR-1.6
Priority	High

3.3.1.11 View list of employees

Table 55: Functional Requirement for View List of employees

Identifier	FR-1.11
Title	View List of employees
Requirement	The system shall display a list of all the employees registered in the system.
Source	Student
Rationale	The admin can keep a record of people working in the company.
Business Rule	N/A
Dependencies	FR-1.1, FR-1.4, FR-1.6
Priority	High

3.3.1.12 Select employee profile

Table 56: Functional Requirement for Select profile

Identifier	FR-1.12
Title	Select
Requirement	The system will allow admin to select the employee whose information needs to be modified.
Source	Student
Rationale	To change the information of a particular employee.
Business Rule	Only admin can modify the employee information.
Dependencies	FR-1.1, FR-1.4, FR-1.6, FR-1.11
Priority	High

3.3.1.13 Change employee information

Table 57: Functional Requirement for Change employee information

Identifier	FR-1.13
Title	Change employee information
Requirement	The system will allow admin to change the employee information
Source	Student
Rationale	The system must have up-to-date information about its employees.
Business Rule	Username cannot be changed.
Dependencies	FR-1.12
Priority	High

3.3.1.14 Employee Information Preview

Table 58: Functional Requirement for Emplouee information

Identifier	FR-1.14
Title	Employee Information Preview
Requirement	The system will display all the information to the admin as a preview, so he can check that correct information is entered.
Source	Student
Rationale	The system must have up-to-date information about its employees.
Business Rule	N/A
Dependencies	FR-1.13
Priority	High

3.3.1.15 Profile Contents Modified Notification

Table 59: Functional Requirement for Profile Contents Modified Notification

Identifier	FR-1.15
Title	Profile Contents Modified Notification
Requirement	When the admin changes the employee information, the system will display a message "Contents modified successfully".
Source	Student
Rationale	To ensure the manager that contents are modified.
Business Rule	Only admin can modify the employee information.
Dependencies	FR-1.14
Priority	High

3.3.1.16 Change Password Notification

Table 60: Functional Requirement for Change Password Notification

Identifier	FR-1.16
Title	Change Password Notification
Requirement	When an employee changes the password of his/her account, the system will send a notification to the admin "password is modified".
Source	Student
Rationale	The system must have up-to-date information about its employees.
Business Rule	Only admin can modify the employee information.
Dependencies	FR-1.10
Priority	High

3.3.1.17 Delete Employee Profile

Table 61: Functional Requirement for Delete Employee Profile

Identifier	FR-1.17
Title	Delete Employee Profile
Requirement	The system will allow admin to delete the employee information from the system
Source	Student
Rationale	To remove the incorrect record from the system.
Business Rule	Only admin can delete the employee profiles.
Dependencies	FR-1.1, FR-1.4,FR-1.6
Priority	High

3.3.1.18 Search Employee

Table 62: Functional Requirement for Search Employee

Identifier	FR-1.18
Title	Search Employee
Requirement	The system will contain a search bar from admin to search an employee.
Source	Student
Rationale	To reduce the time spent on finding a record from an extended list.
Business Rule	N/A
Dependencies	FR-1.11
Priority	High

3.3.1.19 Request for profile Upgrade

Table 63: Functional Requirement for Profile Upgrade

Identifier	FR-1.19
Title	Request For Profile Upgrade
Requirement	The system will allow the manager to request for profile upgrade of trainee to writer status.
Source	Client
Rationale	When the trainees work is approved by the manager, his status needs to be updated to a writer.
Business Rule	To complete onboarding process the employee profile status must be a writer.
Dependencies	N/A
Priority	High

3.3.1.20 Profile status upgrade

Table 64: Functional Requirement for Profile Status

Identifier	FR-1.20
Title	Profile Status Upgrade
Requirement	The system shall allow admin to upgrade the profile of a trainee to writer status.
Source	Client
Rationale	When the trainees work is approved by the manager, his status needs to be updated to a writer.
Business Rule	To complete onboarding process the employee profile status must be a writer.
Dependencies	FR-1.19
Priority	High

3.3.2 Functional Requirement for Module 2: Project Template Management

Following are the Functional Requirements for Project Template Management.

1.1.1.1 Create project template

Table 65: Functional Requirement for Create Project Template

Identifier	FR- 2.1
Title	Create project template
Requirement	The system shall ask manager to enter Project template name, type and date of creation in template creation window.
Source	Student
Rationale	The organizations needs ready-to-use templates to make employees work easy
Business Rule	Only manager can make project templates.
Dependencies	N/A
Priority	High

3.3.2.1 Project template details verification

Table 66: Functional Requirement for Template Details

Identifier	FR- 2.2
Title	Project template details verification
Requirement	When the manager enters the project template details, the system shall for any left-out spaces.
Source	Student
Rationale	It will help the manager to not register a template with ambiguity.
Business Rule	The system shall contain authentic data.
Dependencies	FR-2.1
Priority	High

3.3.2.2 Project template details verification failed

Table 67: Functional Requirement for Template details verification

Identifier	FR- 2.3
Title	Project template details verification failed
Requirement	If the manager has left any detail to that needs to be filled, the system shall display “Fill all details”.
Source	Student
Rationale	It will help manager to register template with authentic data.
Business Rule	BR-1. The system shall contain authentic record of templates.
Dependencies	FR-2.1, FR-2.2
Priority	High

3.3.2.3 Template details update

Table 68: Functional Requirement for Template Details Verification

Identifier	FR- 2.4
Title	Template details update
Requirement	The system shall allow manager to edit the details of the template.
Source	Student
Rationale	It will allow manager to keep the template details up to date.
Business Rule	BR-1. Template ID cannot be updated.
Dependencies	N/A
Priority	High

3.3.2.4 Template details update failed

Table 69: Functional Requirement for Template details verification failed

Identifier	FR- 2.5
Title	Template details update failed
Requirement	If the manager doesn't saves the updated details of template, the system shall display a message "Details not updated".
Source	Student
Rationale	It will allow manager to keep the template details up to date.
Business Rule	Template ID cannot be updated.
Dependencies	FR-2.4
Priority	High

3.3.2.5 Select Template

Table 70: Functional Requirement for Select template

Identifier	FR- 2.6
Title	Select Template
Requirement	The system shall allow manager to select the template that needs to be updated.
Source	Student
Rationale	To change the information of a template.
Business Rule	Only one template can be selected at a time.
Dependencies	N/A
Priority	Medium

3.3.2.6 Search template

Table 71: Functional Requirement for Search Template

Identifier	FR- 2.7
Title	Search template
Requirement	The system shall contain a search bar or manager to search for templates.
Source	Student
Rationale	To reduce time spent on finding a record from extended lists.
Business Rule	BR-1. Only manager can make changes in the template details.
Dependencies	FR- 2.7
Priority	Medium

3.3.2.7 Search template failed

Table 72: Functional Requirement for Search Teamplate Failed

Identifier	FR- 2.8
Title	Search template failed
Requirement	If system doesn't found searched template name, the system shall display a message "template not found" .
Source	Student
Rationale	To reduce time spent on finding a record from extended lists.
Business Rule	BR-1. Only manager can make changes in the template details.
Dependencies	FR- 2.7
Priority	High

3.3.2.8 Delete template

Table 73: Functional Requirement for Delete Template

Identifier	FR- 2.9
Title	Delete template
Requirement	The system shall allow manager to delete the templates information from the system.
Source	Student
Rationale	To remove the data that is of further no use.
Business Rule	N/A
Dependencies	FR-2.4
Priority	High

3.3.2.9 View Template

Table 74: Functional Requirement for View Template

Identifier	FR- 2.10
Title	View Template
Requirement	The system shall display list of all templates registered in the system.
Source	Student
Rationale	Manager can store a record of templates.
Business Rule	Already registered templates can be viewed.
Dependencies	FR-2.1
Priority	High

3.3.2.10 Undo Changes

Table 75: Functional Requirement for Undo Changes

Identifier	FR- 2.11
Title	Undo Changes
Requirement	The system shall allow the manager to undo the changes made before.
Source	Student
Rationale	The changes made to the template details must be deleted.
Business Rule	Only manager can make changes in the template details.
Dependencies	FR-2.4 ,FR-2.5
Priority	High

3.3.2.11 Template Contents Updated Notification

Table 76: Functional Requirement for Template content

Identifier	FR- 2.12
Title	Template Contents Updated Notification
Requirement	The system shall display a message “Contents Updated successfully” after the template has been updated by the manager.
Source	Student
Rationale	This will ensure that the changes are saved.
Business Rule	N/A
Dependencies	FR-2.4, FR-2.5
Priority	High

3.3.2.12 Make template from Existing template

Table 77: Functional Requirement for make template of existing template

Identifier	FR- 2.13
Title	Make template from existing template
Requirement	The system shall allow manager to edit existing template in system to make a new one.
Source	Student
Rationale	To save time if both templates have almost same features.
Business Rule	N/A
Dependencies	FR-2.1
Priority	High

3.3.3 Functional Requirement for Module 3: Project Management

Following are the Functional Requirements for Project Management.

3.3.3.1 Create Writing Team

Table 78: Functional Requirement for Create Writers team

Identifier	FR-3.1
Title	Create Writing Team
Requirement	Manager shall be able to create a team in the system.
Source	Student
Rationale	Group the same niche writers in one place so similar tasks could be assigned to them.
Business Rule	Similar tasks are assigned to writers in a group.
Dependencies	N/A
Priority	High

3.3.3.2 Add Members in Team

Table 79: Functional Requirement for Add members in team

Identifier	FR-3.2
Title	Add Members in Team
Requirement	Manager shall be able to add members in the team.
Source	Student
Rationale	New writer or employee must be assigned to a group.
Business Rule	One employee can be in more than one team.
Dependencies	FR-3.1
Priority	High

3.3.3.3 Remove Members from Team

Table 80: Functional Requirement for remove members from team

Identifier	FR-3.3
Title	Remove Member from Team
Requirement	Manager shall be able to remove member from the team
Source	Student
Rationale	Writer needs to be removed from a team.
Business Rule	N/A
Dependencies	FR-3.1, FR-3.2
Priority	High

3.3.3.4 Assign Task

Table 81: Functional Requirement for Assign task

Identifier	FR-3.4
Title	Assign Task
Requirement	The system shall allow manager or team lead to add task name, task description, allocated time, word count, and attached files for task assignment.
Source	Client
Rationale	It informs the writer/trainee about the assigned task and its details
Business Rule	Tasks are assigned to writers/trainee on daily basis.
Dependencies	N/A
Priority	High

3.3.3.5 View Assigned Task

Table 82: Functional Requirement for view task

Identifier	FR-3.5
Title	View Assigned Task
Requirement	The system shall display writer/trainee the assigned task, task description, allocated time, word count, and attached files.
Source	Client
Rationale	It informs the writer/trainee about the assigned task and its details
Business Rule	Tasks are assigned to writer/trainee on daily basis.
Dependencies	FR-3.4
Priority	High

3.3.3.6 Download task file

Table 83: Functional Requirement for Download Task

Identifier	FR-3.6
Title	Download task file
Requirement	The system shall allow writer/trainee to download the attached files.
Source	Student
Rationale	It helps the user to start working on the given assignment.
Business Rule	N/A
Dependencies	FR-3.4, FR-3.5
Priority	High

3.3.3.7 Submit Task File

Table 84: Functional Requirement for Submit task

Identifier	FR-3.7
Title	Submit Task File
Requirement	The system shall allow writer/trainee to submit the files.
Source	Client
Rationale	It will help the writer to send the solution file to the Team Lead/Manager.
Business Rule	The file size should not exceed 15 MB.
Dependencies	FR-3.4, 3.5
Priority	High

3.3.3.8 Submit Multiple Files

Table 85: Functional Requirement for Submit Multiple files

Identifier	FR-3.8
Title	Submit Multiple Files
Requirement	The system shall allow writer/trainee to submit multiple files.
Source	Student
Rationale	The writer can add more than 1 file if needed.
Business Rule	The file size should not exceed 15 MB.
Dependencies	FR-3.4, FR-3.5
Priority	High

3.3.3.9 Task Submission Failed

Table 86: Functional Requirement for Task Submission failed

Identifier	FR-3.9
Title	Task Submission Failed
Requirement	If the writer/trainee has not selected the file, the system will not allow him/her to submit the form.
Source	Supervisor
Rationale	It will not allow the submission of empty files.
Business Rule	N/A
Dependencies	FR-3.7, FR-3.8
Priority	High

3.3.3.10 Read File Word Count

Table 87: Functional Requirement for Read file word count

Identifier	FR-3.10
Title	Read File Word Count
Requirement	The system will read the files word count using the text editor.
Source	Client
Rationale	It allows system to compare allocated word count and actual word count.
Business Rule	N/A
Dependencies	FR-3.7, FR-3.8
Priority	High

3.3.3.11 Send Submitted File

Table 88: Functional Requirement for Send submit file

Identifier	FR-3.11
Title	Send Submitted File
Requirement	The system will send the submitted file to the manager/team lead
Source	Student
Rationale	It allows manager/team lead to download check the solution file
Business Rule	N/A
Dependencies	FR-3.7, FR-3.8
Priority	High

3.3.3.12 Notify Word Count Deviation

Table 89: Functional Requirement for Notify word count

Identifier	FR-3.12
Title	Notify Word Count Deviation
Requirement	The system shall notify manager about the difference between allocated word count and actual word count.
Source	Client
Rationale	It helps to team lead/manager to check the quality of the work
Business Rule	The word count deviation should not be greater than 100.
Dependencies	FR-3.10
Priority	High

3.3.3.13 Approve Submitted Task

Table 90: Functional Requirement for Approve Submitted Task

Identifier	FR-3.13
Title	Approve Submitted Task
Requirement	The system shall allow manager to approve the submitted task.
Source	Student
Rationale	It removes a task from writer's task screen.
Business Rule	N/A
Dependencies	FR-3.11, FR-3.12
Priority	High

3.3.3.14 Move Task to Task History

Table 91: Functional Requirement for Move task to history

Identifier	FR-3.14
Title	Move Task to Task History
Requirement	When the manager approves the task, the task is added to the task history.
Source	Supervisor
Rationale	The task is removed from writer's task screen and is added to the task history.
Business Rule	N/A
Dependencies	FR-3.13
Priority	High

3.3.3.15 View Task History Writer

Table 92: Functional Requirement for View task history

Identifier	FR-3.15
Title	View Task History Writer
Requirement	The system shall allow writer to view his/her task history
Source	Supervisor
Rationale	Task history maintain a record of all the tasks submitted by the writer.
Business Rule	N/A
Dependencies	N/A
Priority	High

3.3.3.16 Deallocate a Task

Table 93: Functional Requirement for Deallocate task

Identifier	FR-3.16
Title	Deallocate a task
Requirement	Team lead shall be able to deallocate a task given to the writer.
Source	Student
Rationale	The wrong task is assigned to the writer.
Business Rule	N/A
Dependencies	FR-3.4
Priority	High

3.3.3.17 Task Submission After Deadline

Table 94: Functional Requirement for Task submission after deadline

Identifier	FR-3.17
Title	Task Submission After Deadline
Requirement	Writer/Trainee shall be able to submit the task even after the submission time has passed.
Source	Student
Rationale	It helps to tackle late submissions.
Business Rule	Allocated time is the approximate time required to complete the task.
Dependencies	FR-3.4
Priority	High

3.3.3.18 View Task History Manager

Table 95: Functional Requirement for View task history

Identifier	FR-3.18
Title	View Task History Manager
Requirement	The system shall allow manager to view task history of all employees.
Source	Supervisor
Rationale	It maintains the record of the tasks completed by the writer.
Business Rule	Writers are paid on the word count of completed tasks.
Dependencies	N/A
Priority	High

3.3.4 Functional Requirement for Module 4: Payment Management

Following are the Functional Requirements for Payment Management.

3.3.4.1 Maintain payments records

Table 96: Functional Requirement for Maintain payment records

Identifier	FR-4.1
Title	Maintain Payment Records
Requirement	The system shall allow manager to have payment records of all employees.
Source	Client
Rationale	It will allow the manager to have all the data stored.
Business Rule	Payment is generated on the basis of completed tasks.
Dependencies	N/A
Priority	High

3.3.4.2 Remove payment records

Table 97: Functional Requirement for Remove payment records

Identifier	FR-4.2
Title	Remove Payment Records
Requirement	The system shall allow manager to remove the payments that are given to the employees from Due payments.
Source	Client
Rationale	It will allow to have a clean record of payments.
Business Rule	N/A
Dependencies	FR-4.1
Priority	High

3.3.4.3 Payment history

Table 98: Functional Requirement for Payment history

Identifier	FR-4.3
Title	Payment history
Requirement	The system shall allow manager to see the pending payments of employees.
Source	Student
Rationale	It will allow to have a clean record of payments.
Business Rule	Record of the payments must be kept in the database.
Dependencies	FR-4.1
Priority	High

3.3.5 Functional Requirement for Module 5: Task Monitoring and Controlling

Following are the Functional Requirements for Task Monitoring and Controlling.

3.3.5.1 Add An Announcement

Table 99: Functional Requirement for Add announcement

Identifier	FR-5.1
Title	Add An Announcement
Requirement	The system shall allow admin/manager to add header and detail of the announcement.
Source	Student
Rationale	It will allow all system users to view important announcement that should be known by the employee.
Business Rule	Notification must contain the header.
Dependencies	N/A
Priority	High

3.3.5.2 View notice board

Table 100: Functional Requirement for View Notice Board

Identifier	FR-5.2
Title	View Notice Board
Requirement	The system shall allow all the users to view the notice board.
Source	Client
Rationale	It will allow all system users to view important announcement that should be known by the employee.
Business Rule	The announcement must be visible to all employees.
Dependencies	FR-5.1
Priority	High

3.3.5.3 Remove the announcement

Table 101: Functional Requirement for Remove announcement

Identifier	FR-5.3
Title	Remove the announcement
Requirement	The system shall allow admin/manager to remove the announcement from notice board.
Source	Client
Rationale	It will help remove any information that is no longer required or that can create confusion.
Business Rule	Only admin/Manager can remove the announcement form the notice board.
Dependencies	FR-5.1, FR-5.2
Priority	High

3.3.5.4 Remove Announcement Notification

Table 102: Functional Requirement for Remove announcement notification

Identifier	FR-5.4
Title	Remove Announcement Notification
Requirement	The system shall display a message when the information is removed from the noticeboard.
Source	Student
Rationale	It will ensure admin/manager that announcement is removed from the noticeboard.
Business Rule	N/A
Dependencies	FR-5.3
Priority	High

3.3.5.5 View Calendar

Table 103: Functional Requirement for View Calender

Identifier	FR-5.5
Title	View Calendar
Requirement	The system shall allow the employees to have a calendar where they can set reminders for important meetings and deliverables.
Source	Client
Rationale	To set reminders for important meetings and deliverables.
Business Rule	All employees will have a personal calendar.
Dependencies	N/A
Priority	Medium

3.3.5.6 Add Reminder

Table 104: Functional Requirement for Add reminder

Identifier	FR-5.6
Title	Add Reminder
Requirement	The system shall allow employees to set a reminder on the calendar
Source	Client
Rationale	To notify about important deliverable and meetings.
Business Rule	1. Date should not be older than the current date.
Dependencies	N/A
Priority	High

3.3.5.7 Set Reminder Invalid Date

Table 105: Functional Requirement for Set reminder invalid date

Identifier	FR-5.7
Title	Set Reminder Invalid Date
Requirement	The system shall display an error message “Invalid Date” if the selected date has already past.
Source	Supervisor
Rationale	To set a reminder for a valid date.
Business Rule	N/A
Dependencies	FR-5.6
Priority	High

3.3.5.8 Notification for Reminder

Table 106: Functional Requirement for Notification for reminder

Identifier	FR-5.8
Title	Notification For Reminder
Requirement	The system shall notify the employee on the date for which the reminder was set.
Source	Client
Rationale	To notify about important deliverable and meetings.
Business Rule	The system shall be notified as soon as the date changes.
Dependencies	FR-5.6
Priority	High

3.3.5.9 Remove Reminder

Table 107: Functional Requirement for Remove reminder

Identifier	FR-5.9
Title	Remove Reminder
Requirement	The system shall allow the employee to remove a reminder.
Source	Client
Rationale	To remove a reminder that is no longer needed.
Business Rule	N/A
Dependencies	FR-5.6
Priority	High

3.3.5.10 Dim Highlight from Past Reminder

Table 108: Functional Requirement for Dim highlight

Identifier	FR-5.10
Title	Dim Highlight from Past Reminder
Requirement	The system shall diminish the reminder when the date has passed.
Source	Student
Rationale	To maintain the highlight of upcoming reminders.
Business Rule	N.A
Dependencies	FR-5.8
Priority	High

3.3.5.11 Generate Reports

Table 109: Functional Requirement for Generate reports

Identifier	FR-5.11
Title	Generate Reports
Requirement	The system shall allow writer/manager to download the report of the task for a specified time.
Source	Supervisor
Rationale	The writers/manager can download the list of tasks completed in the specific time for record.
Business Rule	Supported download formats are PDF, Word, and Excel sheets.
Dependencies	FR-3.15, FR-3.18
Priority	High

3.3.6 Functional Requirement for Module 6: Employee Evaluation and Recommendation

Following are the Functional Requirements for Employee Evaluation and Recommendation.

3.3.6.1 Manager evaluates Team lead

Table 110: Functional Requirement for Manager evaluates team lead

Identifier	FR-6.1
Title	Manager evaluates Team lead
Requirement	The system shall allow manager to evaluate the Team leads by the performance of their teams.
Source	Student
Rationale	It will allow manager to evaluate team leads leadership skills.
Business Rule	Team leads must be evaluated for their leadership skills.
Dependencies	N/A
Priority	High

3.3.6.2 Manager evaluates Trainee

Table 111: Functional Requirement for Manager evaluates trainee

Identifier	FR-6.2
Title	Manager evaluates Trainee
Requirement	The system shall allow manager to evaluate the Trainee by evaluating their tasks.
Source	Student
Rationale	It will allow manager to evaluate trainees .
Business Rule	Trainees must be evaluated on the basis of their submitted tasks.
Dependencies	FR-3.7
Priority	High

3.3.6.3 Manger evaluates Writer

Table 112: Functional Requirement for Manager evaluated Writer

Identifier	FR-6.3
Title	Manager evaluates Writer
Requirement	The system shall allow manager to evaluate the Writer by evaluating his word count per day with time
Source	Student
Rationale	It will allow manager to evaluate writer.
Business Rule	Writers are evaluated on the basis of word count per day with time.
Dependencies	FR-3.7
Priority	High

3.3.6.4 View Employees Performance

Table 113: Functional Requirement for View employees performance

Identifier	FR-6.4
Title	View Employees Performance
Requirement	The system shall allow manager to view employees performance shown by Gant charts and pie charts.
Source	Student
Rationale	It will allow manager to view employees evaluations.
Business Rule	Track performance of the employees.
Dependencies	FR-4.8
Priority	High

3.3.7 Functional Requirement for Module 7: Communication

Following are the Functional Requirements for Communication.

3.3.7.1 Write Message in Team

Table 114: Functional Requirement for Write message in team

Dependencies	FR-7.1
Priority	Write Message In Team
Dependencies	The system shall allow employees to send message in the team.
Priority	Student
Dependencies	It allows team members to communicate with each other.
Priority	Employee can only communicate with the team members.
Dependencies	N/A
Priority	High

3.3.7.2 Attach File/Photo with Message

Table 115: Functional Requirement for Attach file with message

Identifier	FR-7.2
Title	Attach File/Photo with Message
Requirement	The system shall allow employee to attach file and photo in the message.
Source	Student
Rationale	It help to send attachment in the chat.
Business Rule	N/A
Dependencies	FR-7.1
Priority	High

3.3.7.3 Notify Team Members

Table 116: Functional Requirement for Notify team members

Identifier	FR-7.3
Title	Notify Team Members
Requirement	When a team member sends a message, they system will send notification to all other team members.
Source	Supervisor
Rationale	It helps to alert team members about a message.
Business Rule	The employee must be in the team to receive the message.
Dependencies	FR-7.1, FR-7.2
Priority	High

3.3.7.4 Schedule Meeting

Table 117: Functional Requirement for Schedule meetings

Identifier	FR-7.4
Title	Schedule Meeting
Requirement	The system shall allow managers to schedule a meeting.
Source	Student
Rationale	It helps people know the meeting time and agenda before it is actually held.
Business Rule	Only manager/team lead can schedule a meeting
Dependencies	N/A
Priority	High

3.3.7.5 Start Meeting

Table 118: Functional Requirement for Start Meetings

Identifier	FR-7.5
Title	Start Meeting
Requirement	The system shall allow manager to start a meeting.
Source	Student
Rationale	It help to hold important discussion sessions.
Business Rule	Only manager/team lead can start a meeting
Dependencies	N/A
Priority	High

3.3.7.6 Meeting Notification

Table 119: Functional Requirement for Meeting notifications

Identifier	FR-7.6
Title	Meeting Notification
Requirement	The system shall send the meeting notification to all the team members.
Source	Student
Rationale	It helps all team members to join the meeting.
Business Rule	The employee must be in the team to receive meeting notification.
Dependencies	FR-7.4, FR-7.5
Priority	High

3.3.7.7 Join Meeting

Table 120: Functional Requirement for Join meeting

Identifier	FR-7.7
Title	Join Meeting
Requirement	The system shall allow team members to join the meeting.
Source	Supervisor
Rationale	It will allow employees to attend an online meeting
Business Rule	N/A
Dependencies	FR-7.4, FR-7.5, FR-7.6
Priority	High

3.3.7.8 Share Screen

Table 121: Functional Requirement for Share screen

Identifier	FR-7.8
Title	Share Screen
Requirement	The system shall allow participants to share screen.
Source	Student
Rationale	It helps to present some important information
Business Rule	N/A
Dependencies	FR-7.7
Priority	High

3.3.7.9 Video Sharing

Table 122: Functional Requirement for Video sharing

Identifier	FR-7.9
Title	Video sharing
Requirement	The system shall allow meeting participants to share their video.
Source	Student
Rationale	To help user turn on the camera.
Business Rule	N/A
Dependencies	FR-7.7
Priority	High

3.3.7.10 Leave meeting

Table 123: Functional Requirement for Leave meeting

Identifier	FR-7.10
Title	Leave meeting
Requirement	The system shall allow participants to leave the meeting.
Source	Student
Rationale	To leave the meeting when discussion is over
Business Rule	N/A
Dependencies	FR-7.7
Priority	High

3.3.8 Functional Requirement for Module 8: Onboarding

Following are the Functional Requirements for Onboarding.

3.3.8.1 Add training videos

Table 124: Functional Requirement for Add training videos

Identifier	FR-8.1
Title	Add training videos
Requirement	Manager shall be able to attach a YouTube video link for the trainees in the training video section.
Source	Student
Rationale	It will help to provide the best source in order to learn a concept.
Business Rule	N/A
Dependencies	N/A
Priority	High

3.3.8.2 Watch Training Videos

Table 125: Functional Requirement for Watch training videos

Identifier	FR-8.2
Title	Watch Training Videos
Requirement	Trainee shall be able to watch training videos through the YouTube link.
Source	Client
Rationale	It will help to develop the skillset required to work in the company.
Business Rule	The trainee must complete minimum 3 days trail period before he/she is upgraded to writer status.
Dependencies	FR-8.1
Priority	High

3.3.8.3 Make Quiz for Training Video

Table 126: Functional Requirement for Make quiz

Identifier	FR-8.3
Title	Make Quiz for Training Video
Requirement	Manager shall be able to make a quiz for the training video that will contain questions and text fields to answer the question.
Source	Student
Rationale	It will help to evaluate trainee understanding.
Business Rule	The quiz will have 5 questions.
Dependencies	FR-8.1
Priority	High

3.3.8.4 Allocate Time for Quiz

Table 127: Functional Requirement for Allocate time for quiz

Identifier	FR-8.4
Title	Allocate Time for Quiz
Requirement	The system shall allow manager to assign a time for the quiz.
Source	Student
Rationale	It will help trainee to complete quiz in a certain amount of time.
Business Rule	Quiz time can be allocated based on the complexity of questions.
Dependencies	FR-8.3
Priority	High

3.3.8.5 Take Quiz

Table 128: Functional Requirement for Take Quiz

Identifier	FR-8.5
Title	Take Quiz
Requirement	The system shall allow trainee to take a quiz.
Source	Student
Rationale	It will help to evaluate trainee understanding.
Business Rule	N/A
Dependencies	FR-8.3, FR-8.4
Priority	High

3.3.8.6 Monitor Quiz Time

Table 129: Functional Requirement for Monitor Quiz Time

Identifier	FR-8.6
Title	Monitor Quiz Time
Requirement	The system shall submit the quiz when the time is up.
Source	Supervisor
Rationale	It will allow trainee to complete the quiz before time.
Business Rule	N/A
Dependencies	FR-8.3, FR-8.5
Priority	High

3.3.8.7 Submit Quiz

Table 130: Functional Requirement for Submit Quiz

Identifier	FR-8.7
Title	Submit Quiz
Requirement	The system shall allow trainee to submit the quiz.
Source	Student
Rationale	It will allow manager to evaluate trainee's understanding.
Business Rule	The trainee can only submit the form once.
Dependencies	FR-8.5, FR-8.6
Priority	High

3.3.8.8 Evaluate Answered Question

Table 131: Functional Requirement for Evaluate Answers

Identifier	FR-8.8
Title	Evaluate Answered Question
Requirement	The system shall allow manager to evaluate the answers by giving score for each question.
Source	Student
Rationale	It will allow manager to evaluate trainee's understanding.
Business Rule	Empty answer field are replaced with text "Not answered"
Dependencies	FR-8.3, FR-8.7
Priority	High

3.3.8.9 View Quiz Score

Table 132: Functional Requirement for View Quiz Score

Identifier	FR-8.9
Title	View Quiz Score
Requirement	The system shall allow trainee to view the score of the quiz.
Source	Supervisor
Rationale	It will allow trainee to view his performance.
Business Rule	N/A
Dependencies	FR-8.8
Priority	High

3.3.9 Functional Requirement for Module 9: Help

Following are the Functional Requirements for Help.

3.3.9.1 Get Help

Table 133: Functional Requirement for Get Help

Identifier	FR-9.1
Title	Get Help
Requirement	The system shall allow employee to get help in using the system through an AI chat bot.
Source	Supervisor
Rationale	It helps new employees to learn the system use.
Business Rule	Chat bot help will make the system more interactive.
Dependencies	N/A
Priority	High

3.4 Non-Functional Requirements

3.4.1 Reliability

REL-1: The web application shall be available 24/7
REL-2: The system shall retrieve the data from database correctly 85% of the time without crashing.
REL-4: The system shall be able to successfully handle wrong input 85% of the time.

3.4.2 Usability

USE-1: The system shall display an error message in case of any exception.
USE-2: The system shall display loading icon when the system is busy or fetching data.
USE-3: The system shall allow users to navigate the complete application with as few instructions as possible.
USE-4: The buttons used in the system shall be of large size so that user can interact with them easily.
USE-5: The buttons and icons used in the system shall convey an easily understandable meaning of their functionality.
USE-6: Theme, fonts, icons, and language used should be consistent throughout the system to increase readability and understandability.

.

3.4.3 Performance

PER-1: The system shall display search result within 3 seconds.
PER-2: The web application shall respond to any trigger within 1 second.
PER-3: The system shall generate report successfully within 1 minute.
PER-4: The system shall accommodate 100 users at the peak usage time.
PER-5: The web application shall load the login screen within 3 seconds when the user starts the application.

3.4.4 Security

SEC-1: The system shall login users with valid credentials only.
SEC-1: The system shall not give access to the roles that are not assigned.
SEC-1: The system shall verify security through JWT.

3.5 External Interface Requirements

3.5.1 User Interfaces Requirements

The user interface requirements are selected by keeping in mind, the 10 usability heuristics [4], and some important concepts of human computer interaction

UI-1: The system shall have a self-descriptive interface.
UI-2: The system shall perform a task in minimum 5-6 steps.
UI-3: The system shall have a consistent layout.
UI-4: The system shall keep users informed about their position in the system.
UI-5: The system shall have a navigation bar to use usability.
UI-6: The system shall have high contrast of colors.

3.5.2 Software interfaces

SI-1: The system shall interact with the zoom API to support video conferencing.
SI-2: The system shall interact with Chat bot API to help user understand the system.
SI-3: The system shall retrieve data from MongoDB server.

3.5.3 Hardware interfaces

HI-1: The system will turn on the camera for video conferencing.

3.5.4 Communications interfaces

CI-1: The system shall send an email to the admin to request for any change in employees profile's information.

4 Design and Architecture

4.1 Architectural Design

4.1.1 Box and Line Diagram

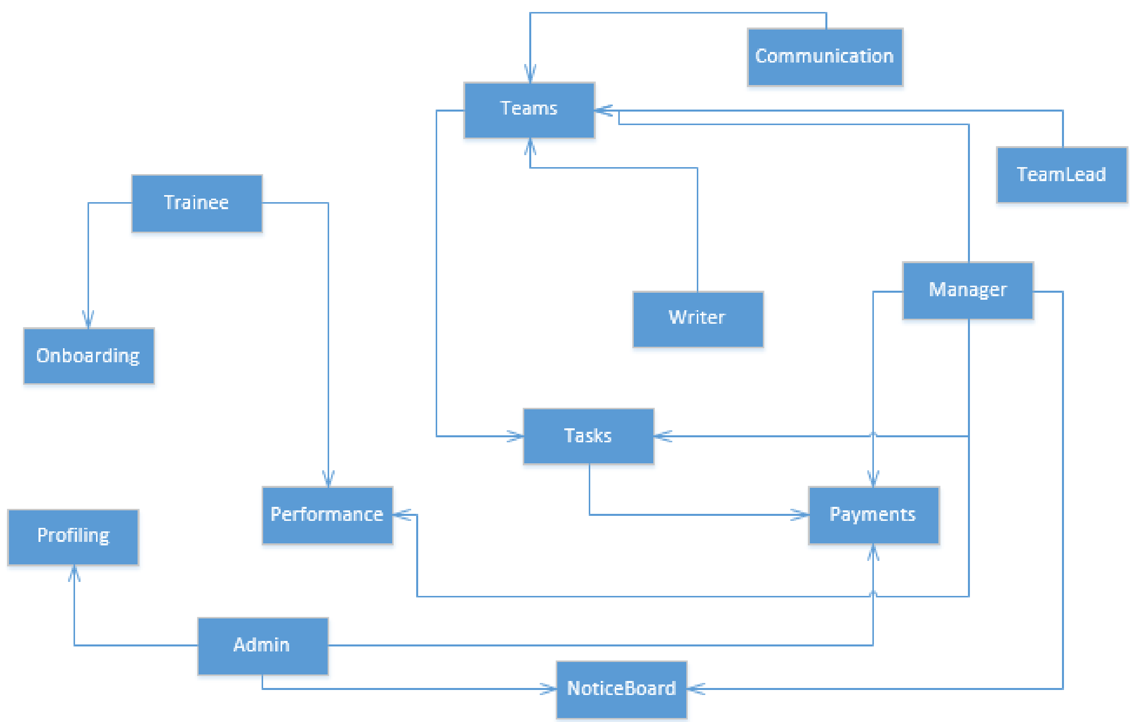


Figure 6: Box and Line Diagram

4.1.2 Architecture Diagram

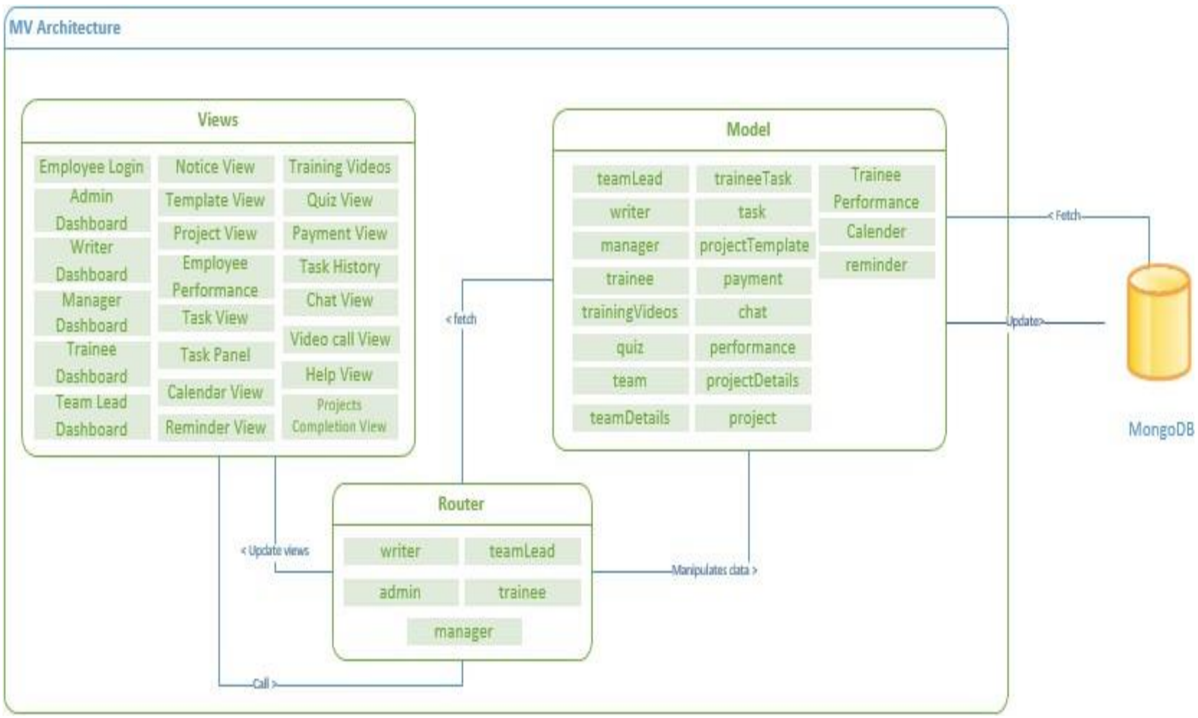


Figure 7: Architecture Diagram

4.2 Design Models

The design model contains the activity diagrams, class diagram, and sequence diagram of FVOA to help in the development of frontend and backend of the web application.

4.2.1 Activity Diagrams

4.2.1.1 View Employee Performance

Manager opens the employee panel and click on the employee. The system shows graphs and statistics of the employee based on his/her performance on the task. If the employee is new the performance graph will show "employee not evaluated".

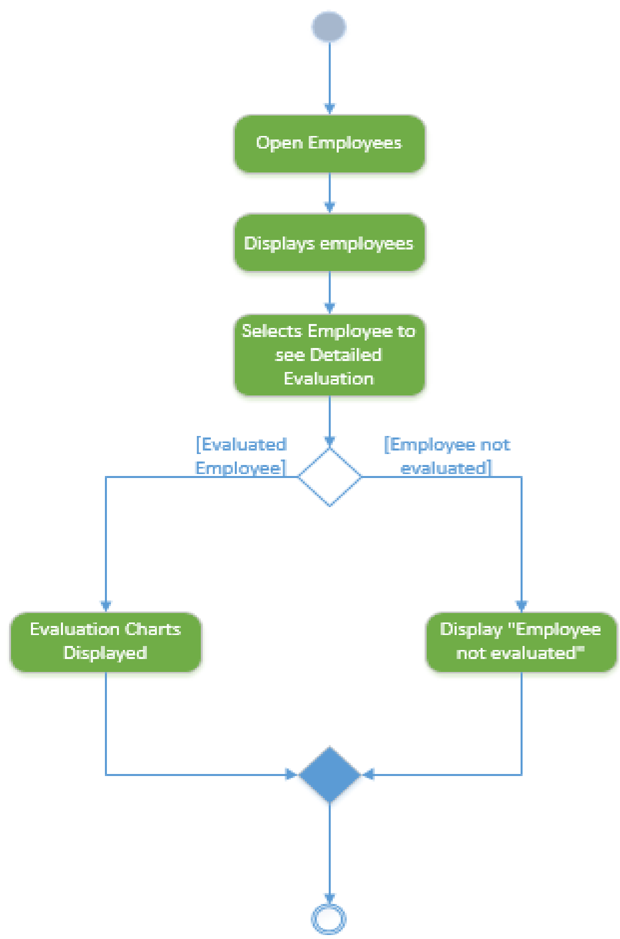


Figure 8: Activity diagram for View Employees

4.2.1.2 Create Employee Profile

Admin opens the employee panel and selects to add a new employee. The system displays a form to enter the details and verifies so a unique employee data is being added. When the admin selects the role if the role is writer, as for word count. The admin then adds a unique username and password for the writer and profile is created successfully.

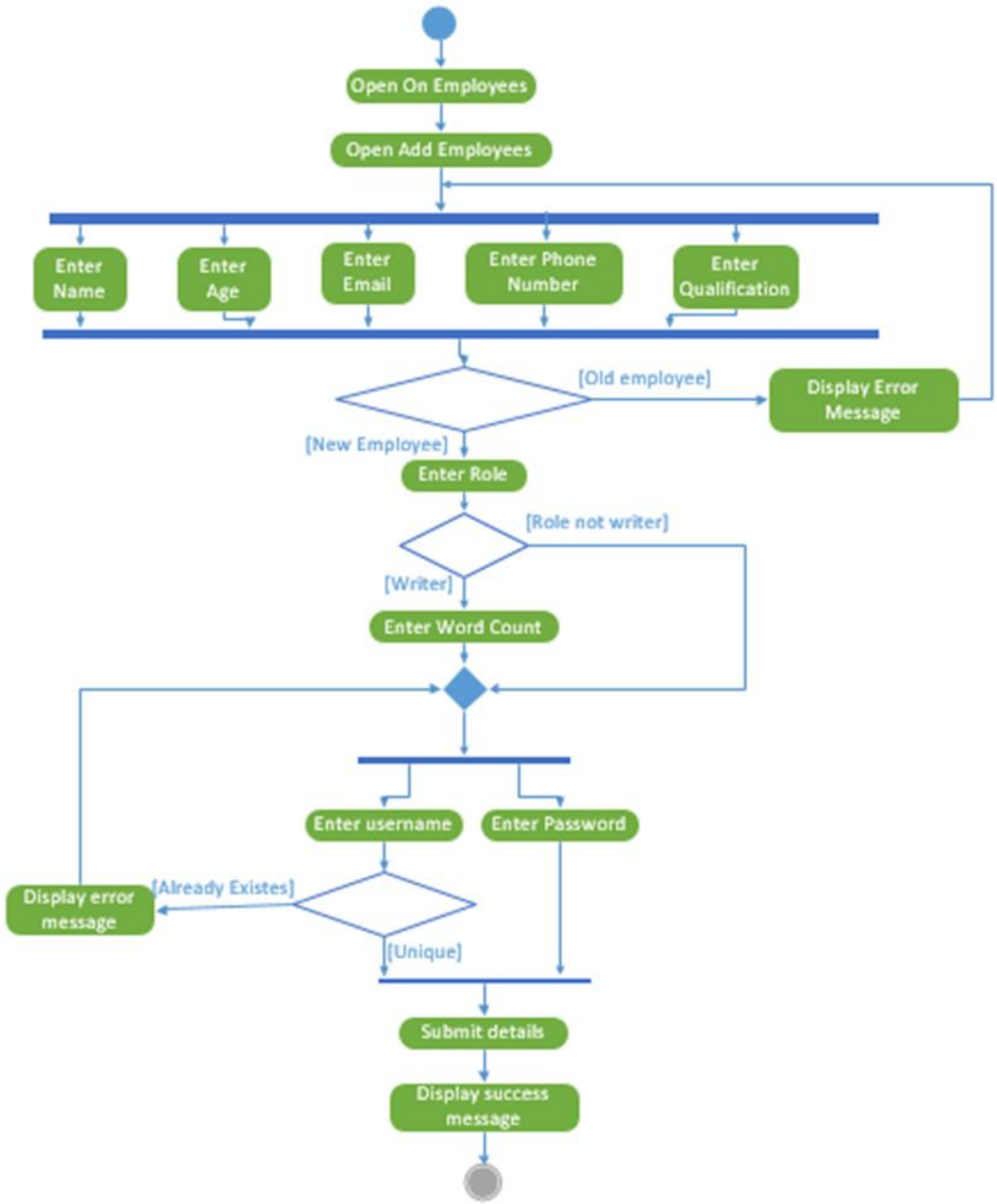


Figure 9: Activity diagram for Create employee profile

4.2.1.3 Upgrade Profile

When the manager request for the up gradation of trainee profile to writer or any information needs to be edited. The employee is selected and the desired information is edited.

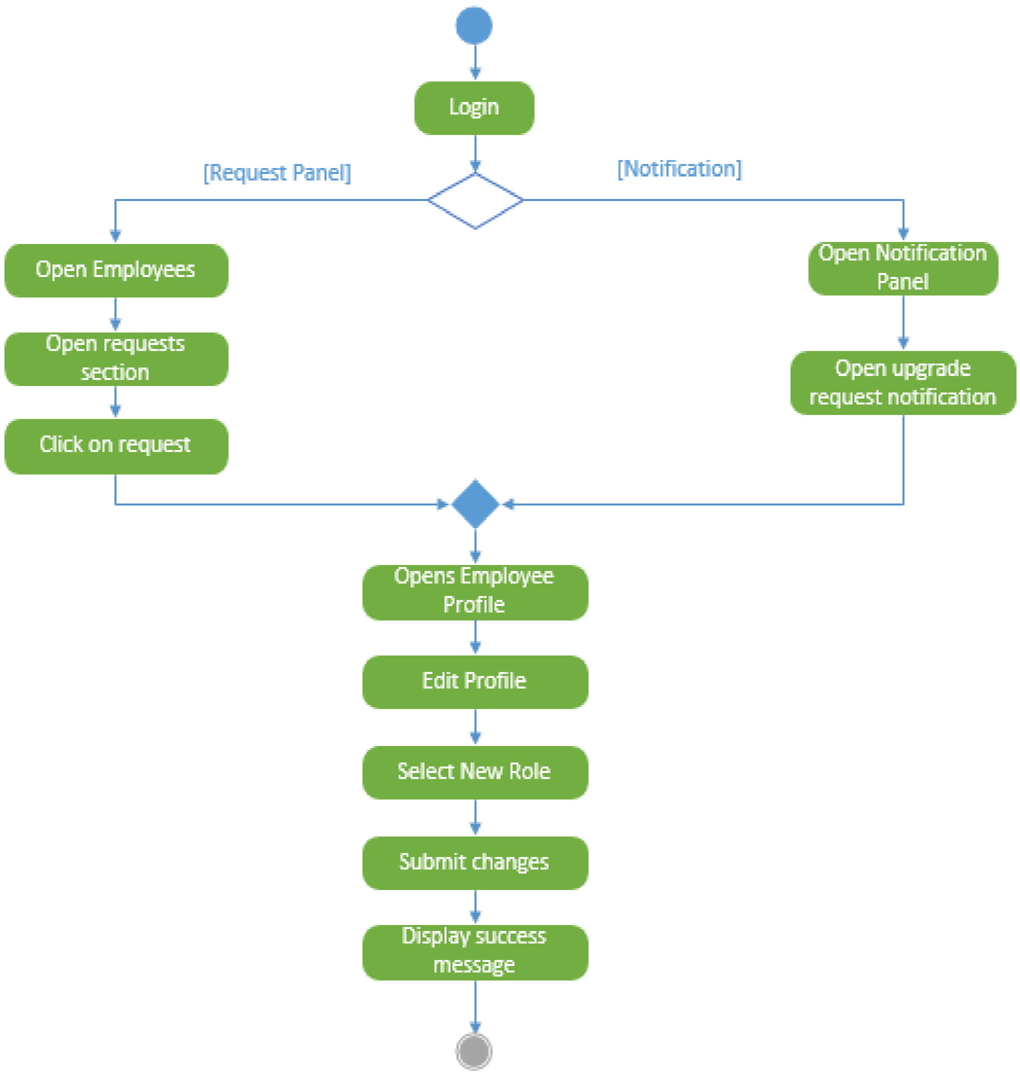


Figure 10: Activity diagram for Update profile

4.2.1.4 Sign In

User selects the role through which he/she wants to sign into the system. They then enter the credentials (username and password). The system validates the information and the user is logged into the system.

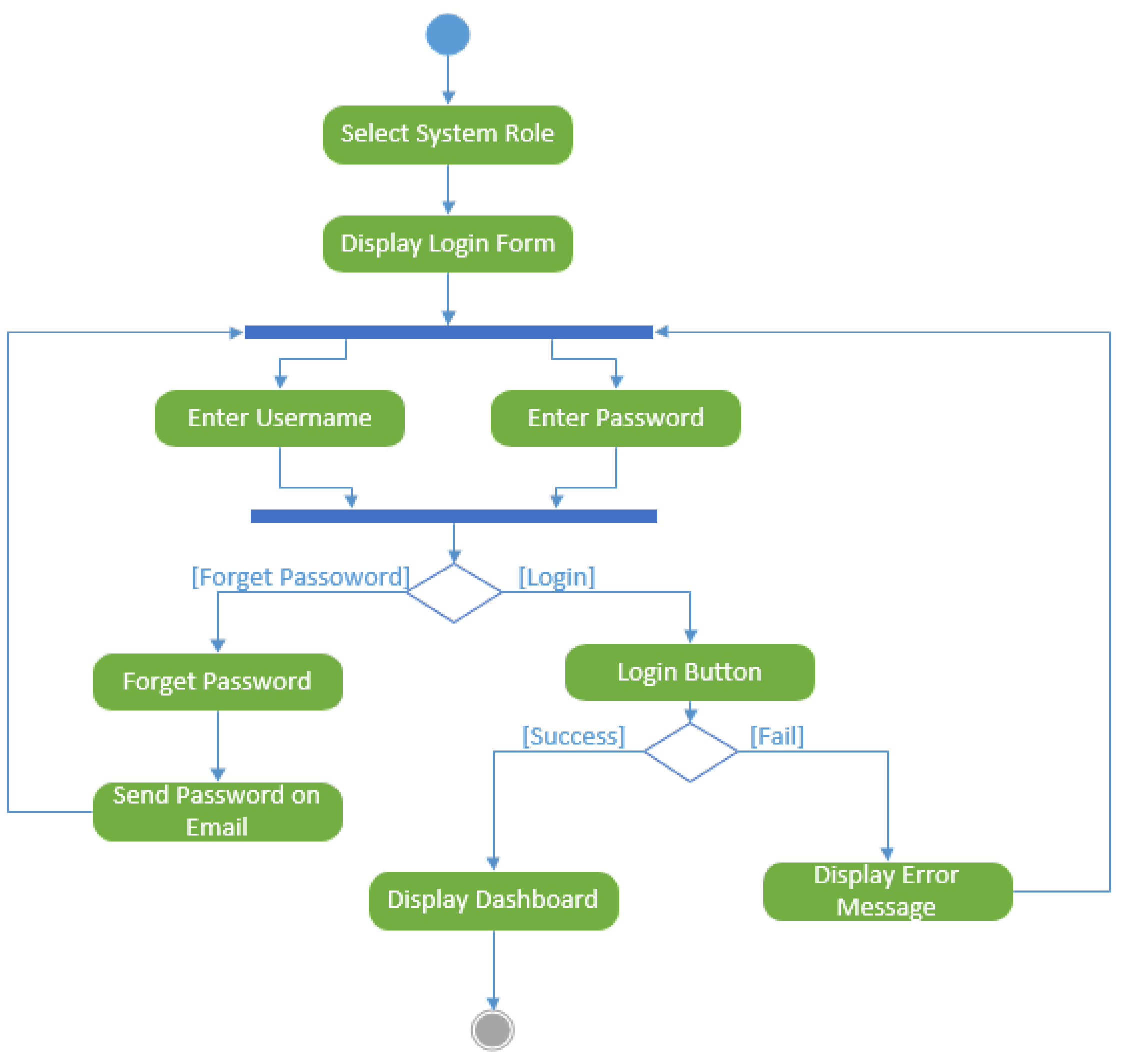


Figure 11: Activity Diagram for Signin

4.2.1.5 Add Project Template

To create a new project template, the manager opens Template panel. He/she selects add a new template and fills out the name for the template system takes the name of the manager and the date at which the template is created and create a new template area.

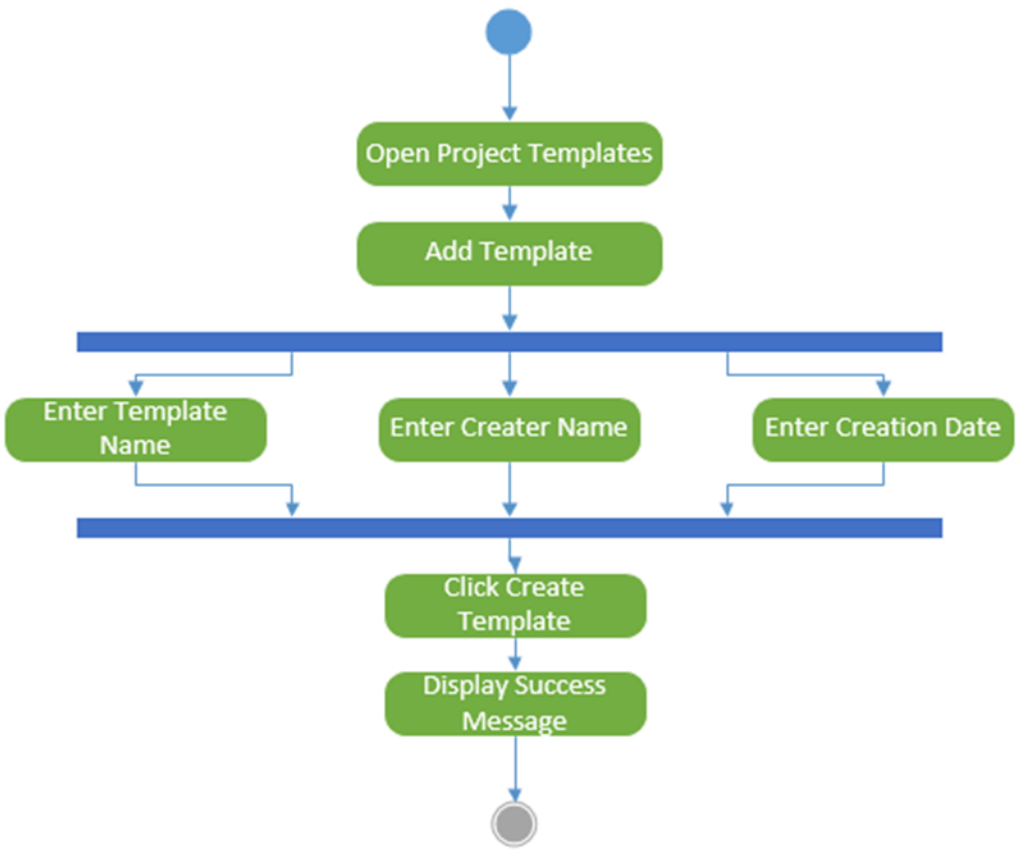


Figure 12: Activity Diagram for Add project template

4.2.1.6 Edit Template Contents

To edit the template contents, the manager selects the template from the templates panel. The can add a new task or add a sub-task to the existing tasks. The process is completed by giving the task a name, start-date, end-date, and dependency in case of sub-task.

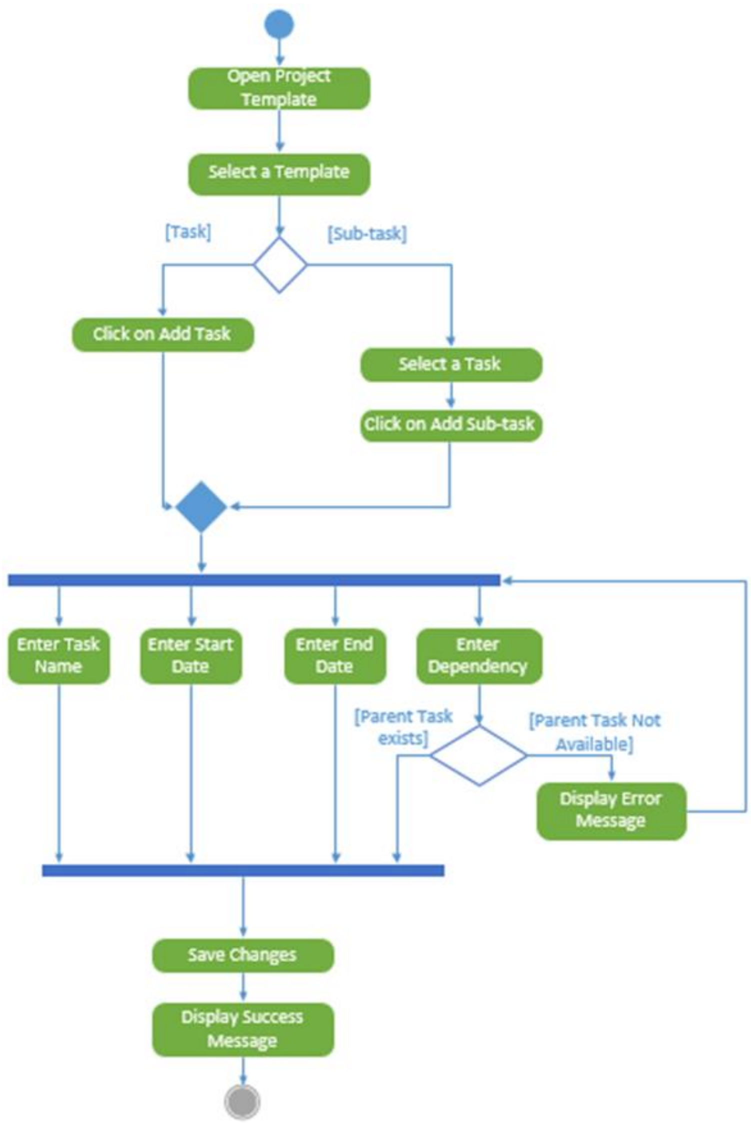


Figure 13: Activity Diagram for Edit template

4.2.1.7 Add Project Details

Manager opens the projects panel from where he goes to add a new project. The system displays a form to add the project details. If the fields are empty the user is not able to submit the project details, otherwise the detailed are added successfully.

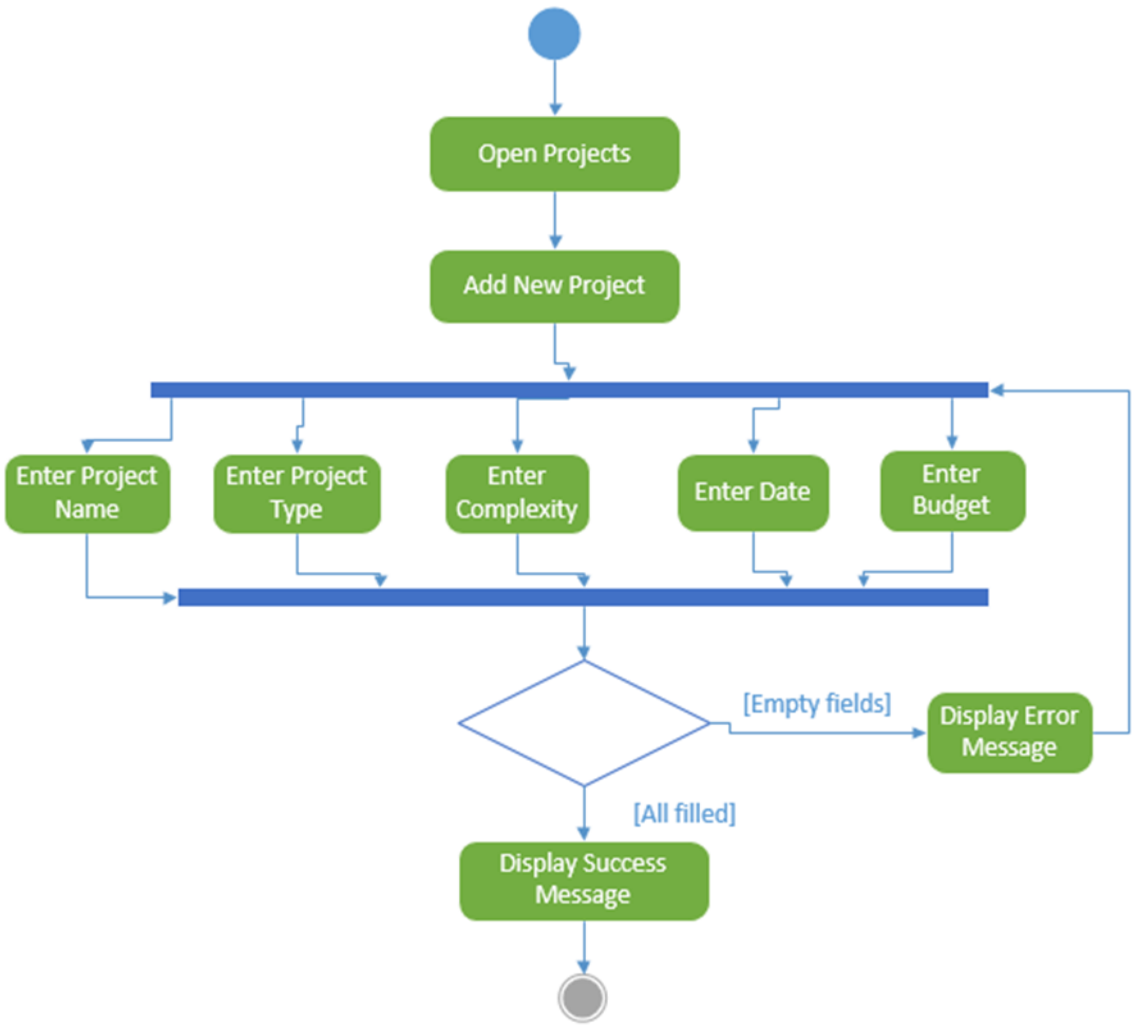


Figure 14: Activity Diagram for Add project Details

4.2.1.8 View Projects

Manager can view the projects in the project panel. The project list is displayed to the manager. The details of the project will include the writers working on the project. The manager can filter out the ongoing projects to check their completion rate. The completed projects can also be viewed by changing the filter on the projects.

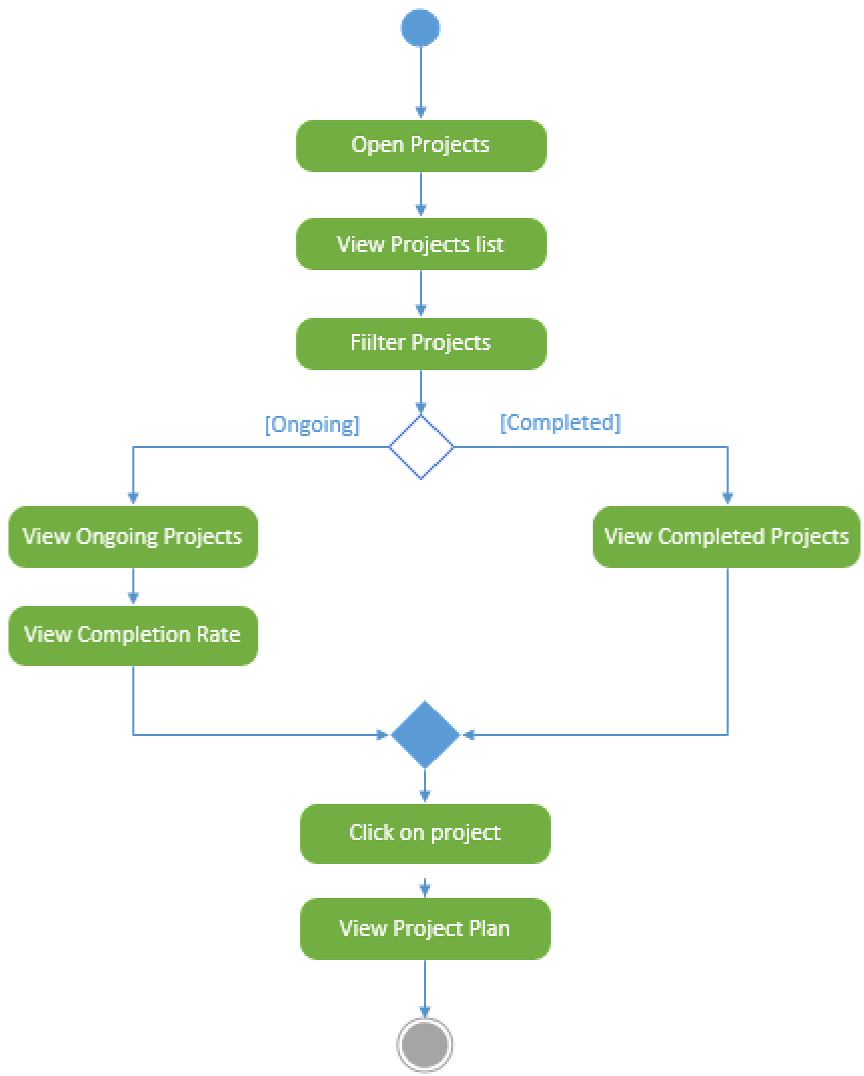


Figure 15: Activity diagram for View projects

4.2.1.9 Submit Task

The writer opens the assigned task from the task panel and attach the file where the see the option to choose file. The system checks for the file to be below 10MB then the file is uploaded.

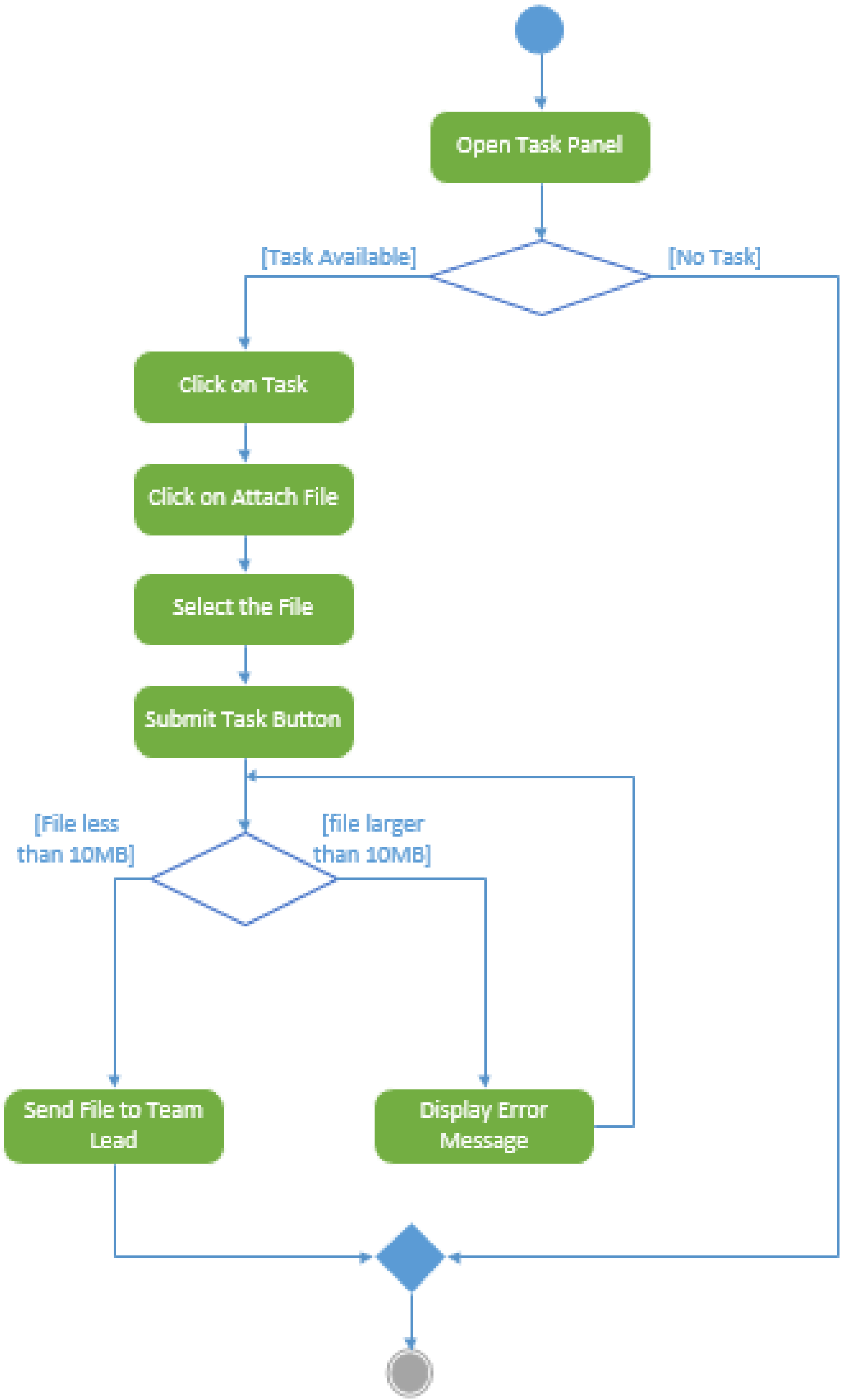


Figure 16: Activity diagram for Submit task

4.2.1.10 Evaluate Task

The Manager/Team Lead opens the task panel on their side to view the tasks solutions submitted by the writers. The task solution contains the word count of the file, submission time, and file. The manager downloads the file and checks the quality of the task. If it is acceptable, the writer is notified and task is added.

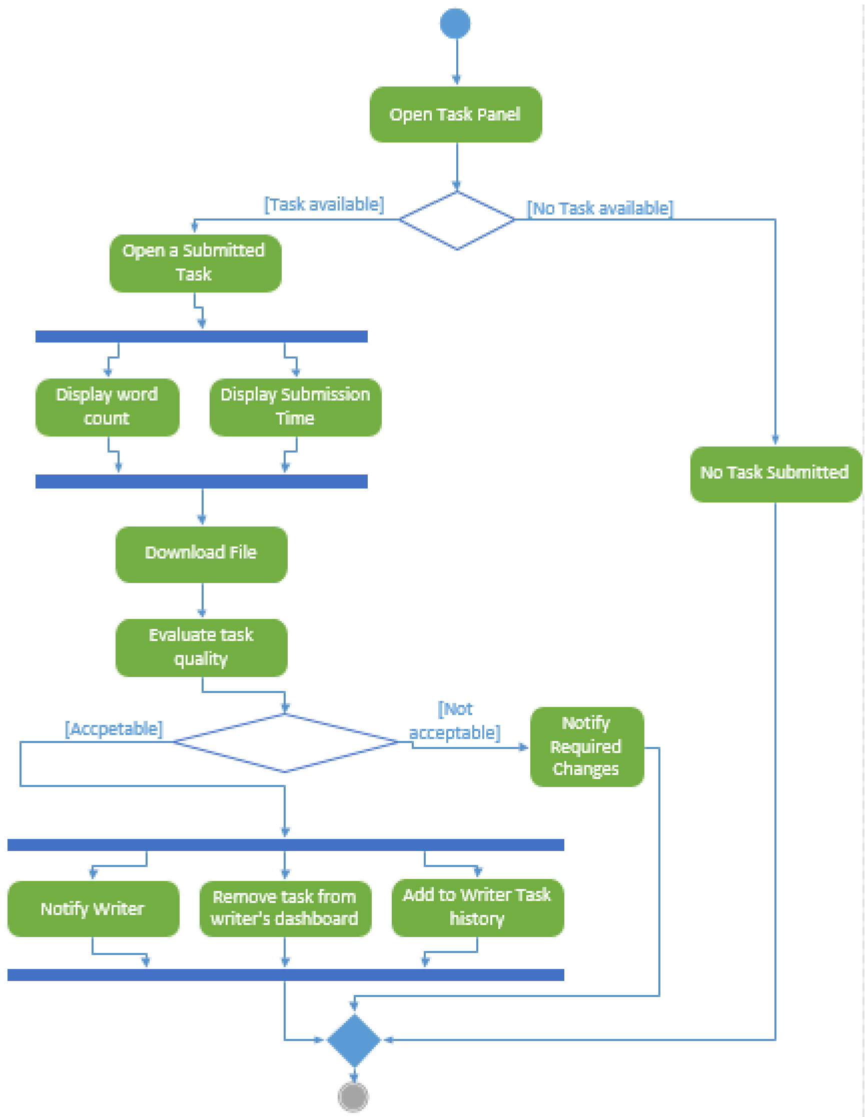


Figure 17: Activity diagram for Evaluate task

4.2.1.11 Remove a Payment

Manager and admin can remove the payments from the payment panel. First the filter out due payments. When the due payment are displayed they can pay and employee and manually change the due status to paid. The system will ask the manager/admin for password for authentication. After successful authentication, it will be moved to the “paid” record.

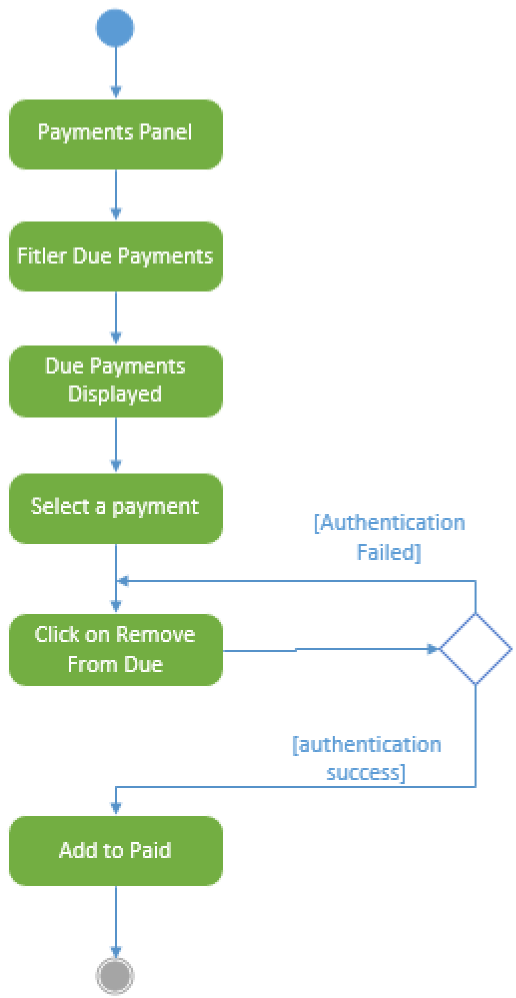


Figure 18: Activity diagram for Remove a payment

4.2.1.12 Add Notice on Notice Board

The admin/Manger views the notice board and clicks on add notice. They add header and description of the notice. The system checks that the header is not empty. If it is empty the system stops the user on the same page, otherwise the notice is added on the noticeboard.

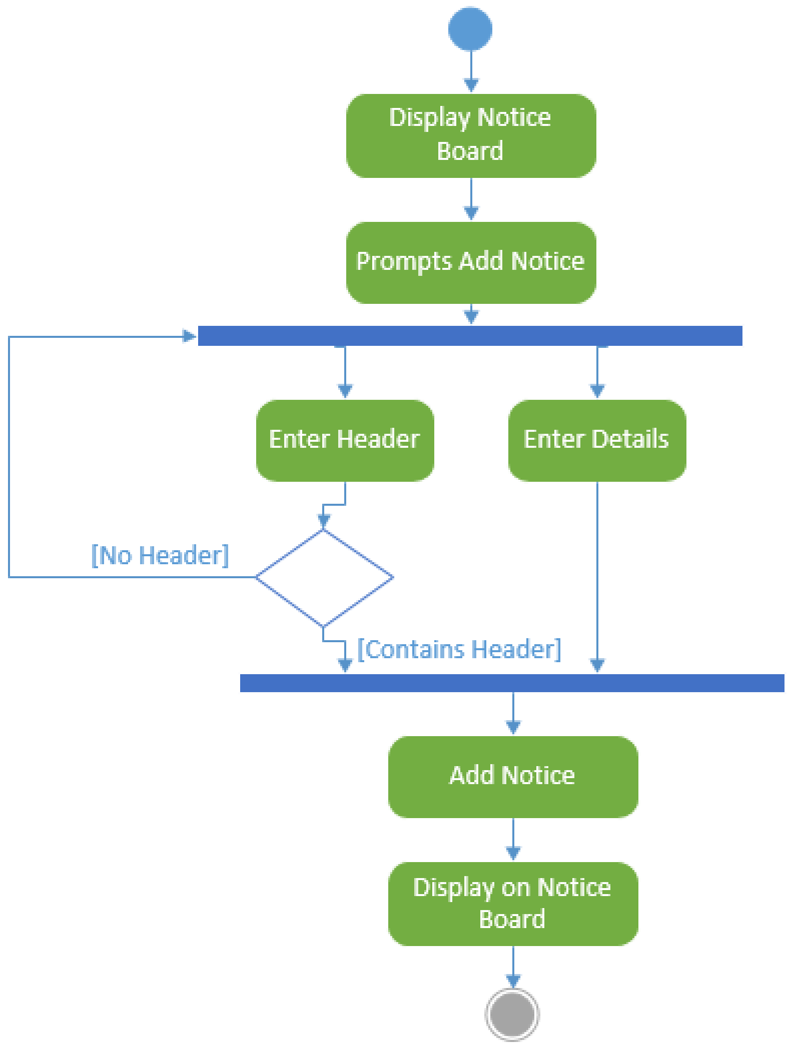


Figure 19: Activity diagram for Add notice on noticeboard

4.2.1.13 Add Reminder

The Manger/TeamLead/Writer opens the calendar and then goes to adding a new reminder. The system displays a form to select a date and title for the reminder. The user can also add the time for the reminder, it is optional. The user gets error message if date and title is not entered, otherwise the user is

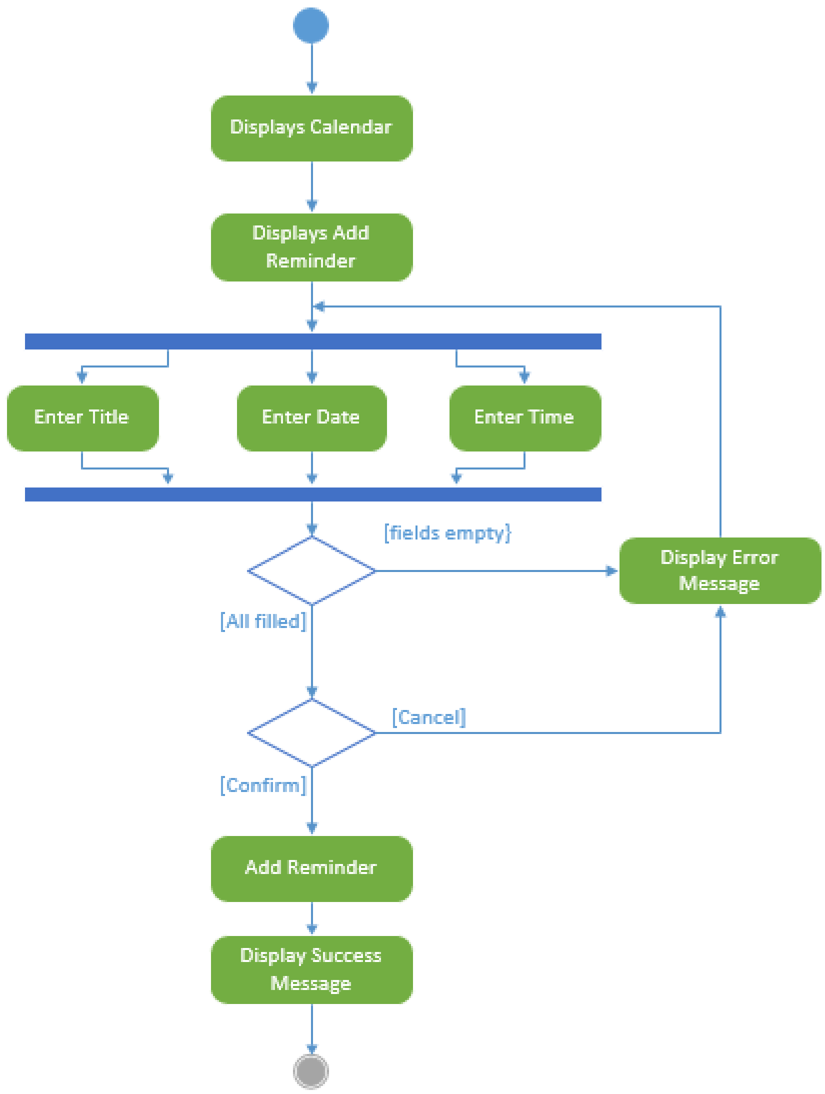


Figure 20: Activity diagram for Add reminder

asked to confirm the reminder.

4.2.1.14 Watch Training Videos

Trainees click on the training videos panel form the dashboard, the system displays a list of all the available videos. The trainee then selects the desired video and play it. It trainee can control the video through the control buttons.

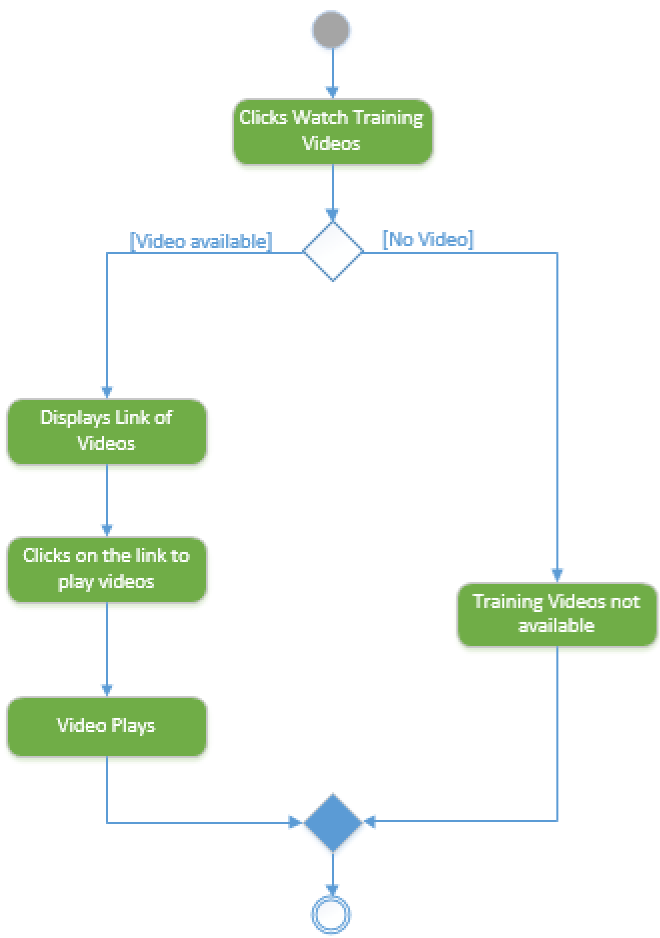


Figure 21: Activity diagram for Watch training videos

4.2.1.15 Help

The system users select the "?" from the panel which will open a chat window. The user can add the queries their and chatbot will reply to the user queries.

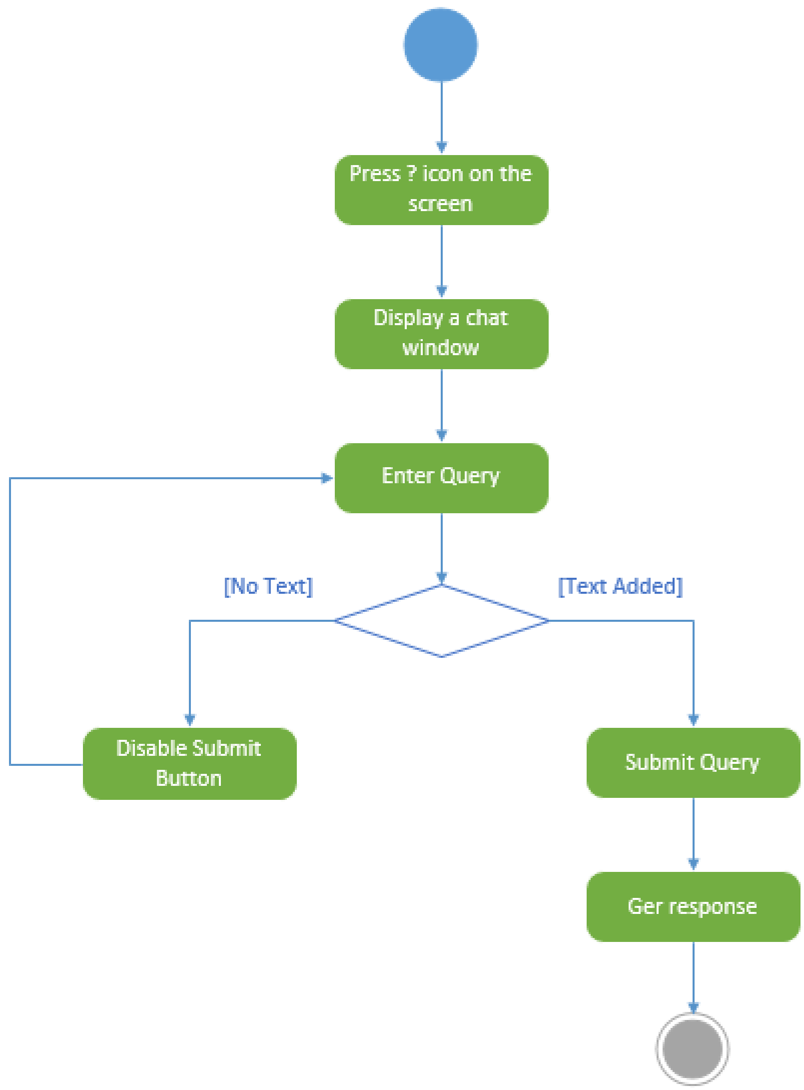


Figure 22: Activity diagram for Get Help

4.2.2 Class Diagram

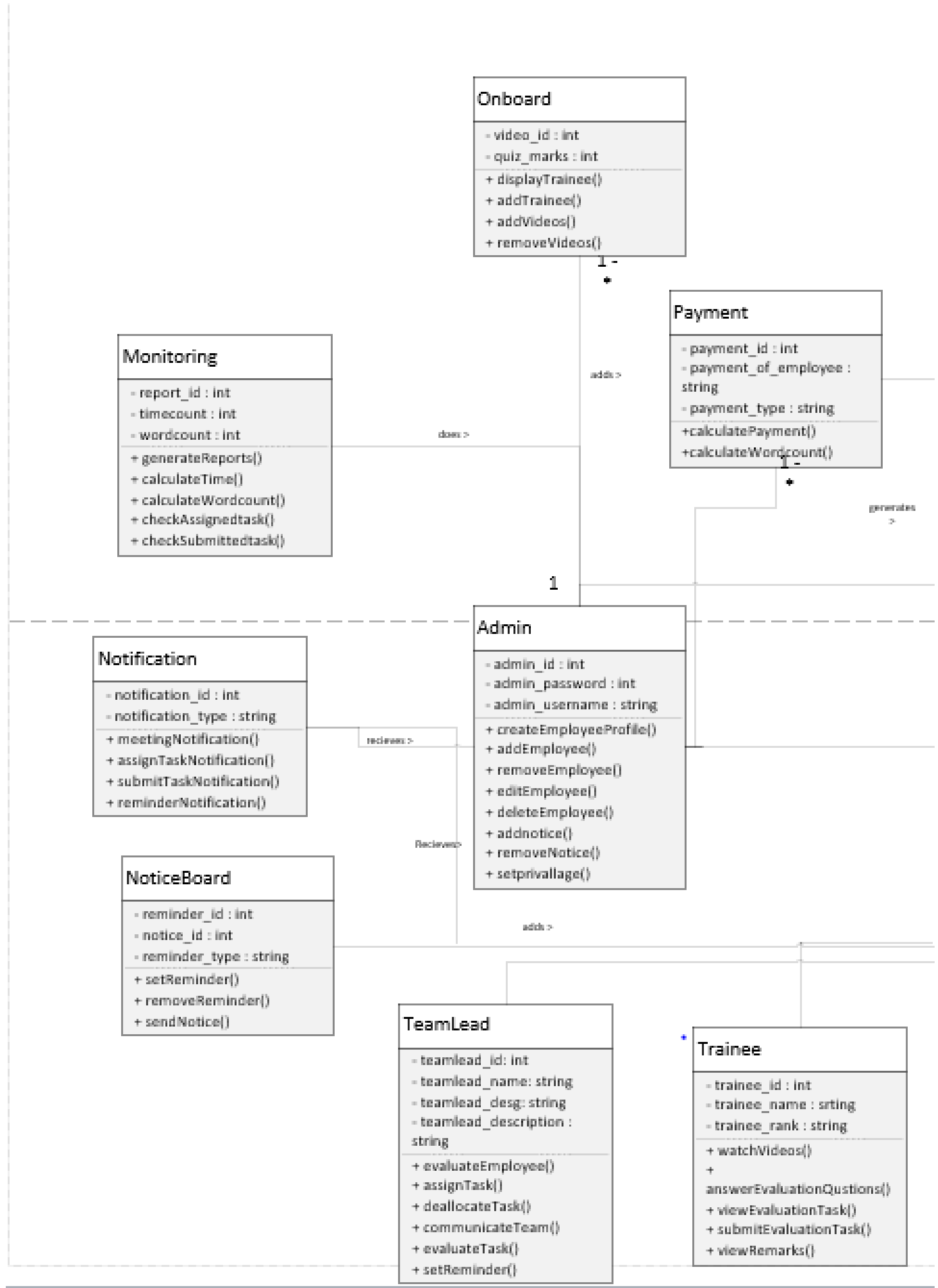


Figure 23: Class Diagram for FVOA - 1

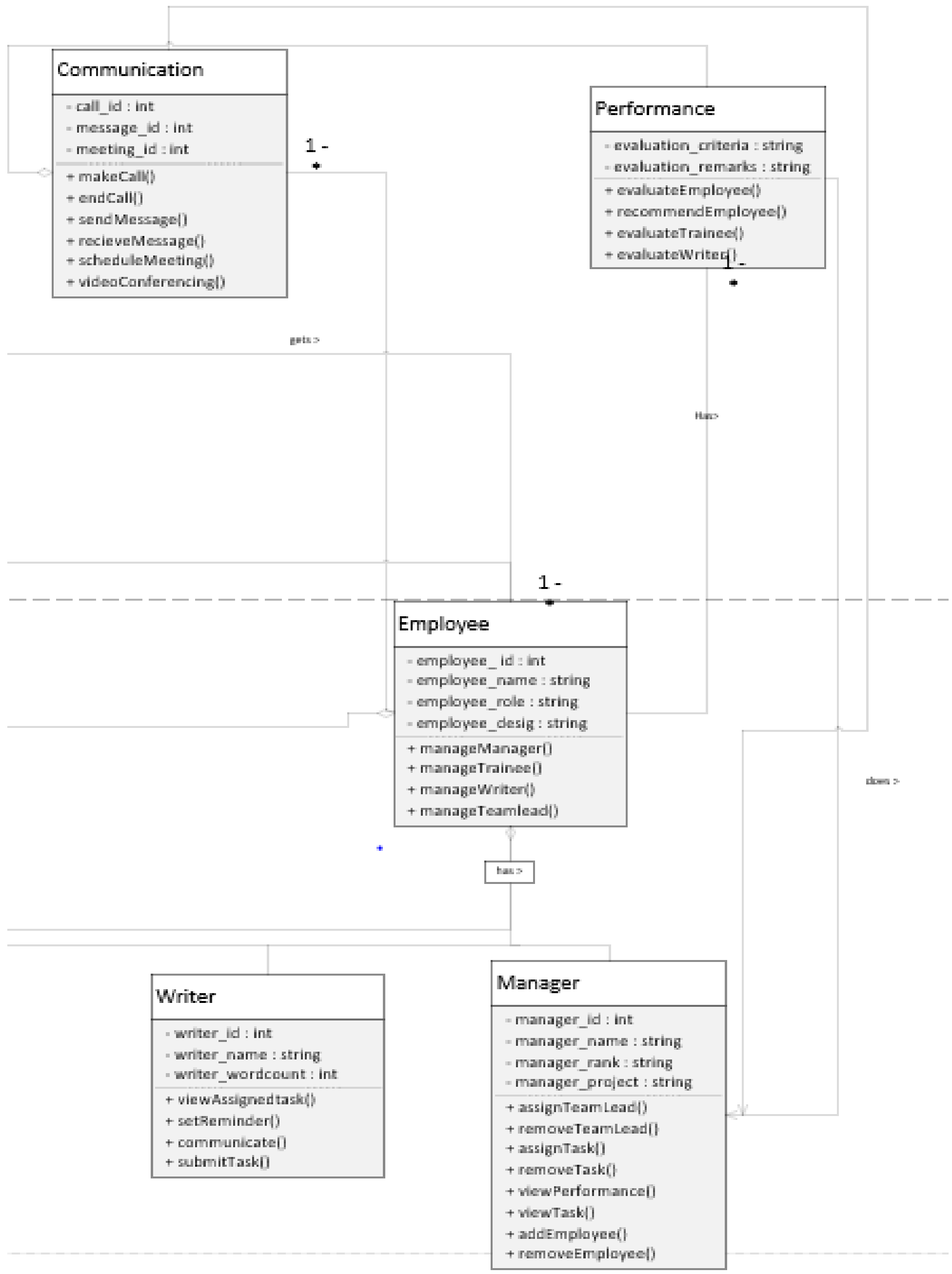


Figure 24: Class Diagram for FVOA - 2

4.2.3 Sequence Diagram

4.2.3.1 Edit Template Contents

To edit the template contents, the manager selects the template from the templates panel. The can add a new task or add a sub-task to the existing tasks. The process is completed by giving the task a name, start-date, end-date, and dependency in case of sub-task.

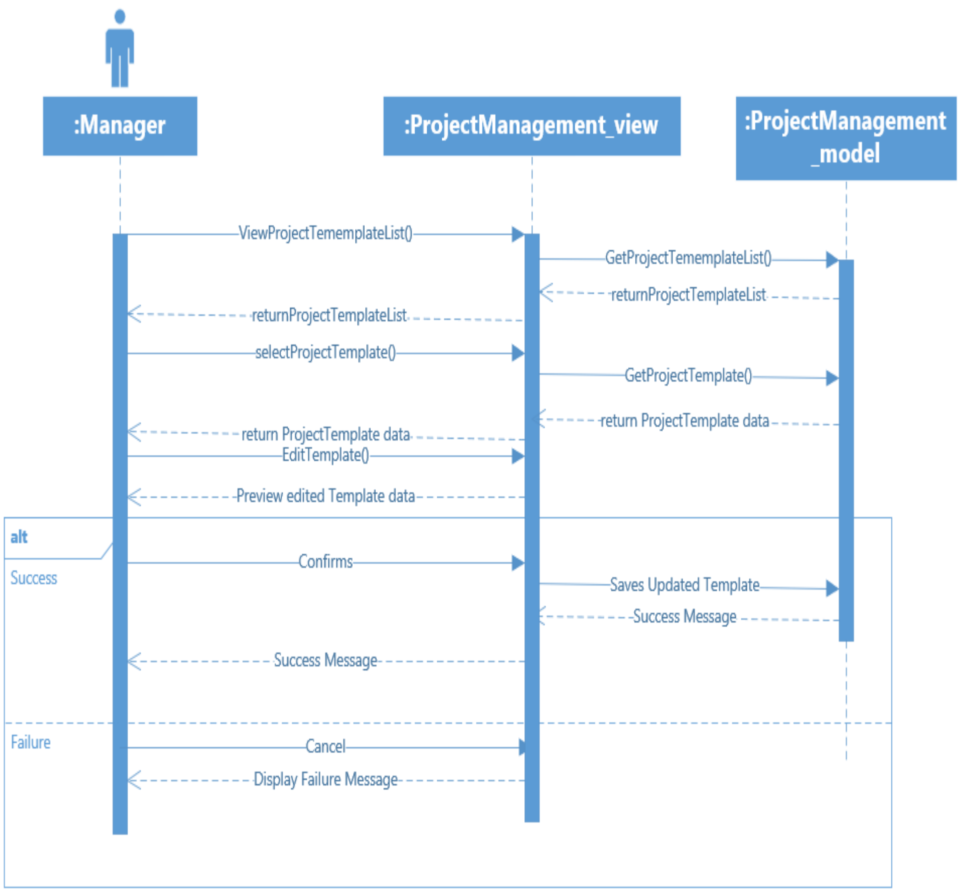


Figure 25: Sequence Diagram for Edit Template Contents

4.2.3.1 Assign Task to Writer

Manager assigns task to the writer by opening Writer's profile from Writer's list. Then the system checks the task count of the writer. If the task count is less than 5 then the system will prompt a window to manager to assign task to the writer, otherwise the system will prompt an error message.

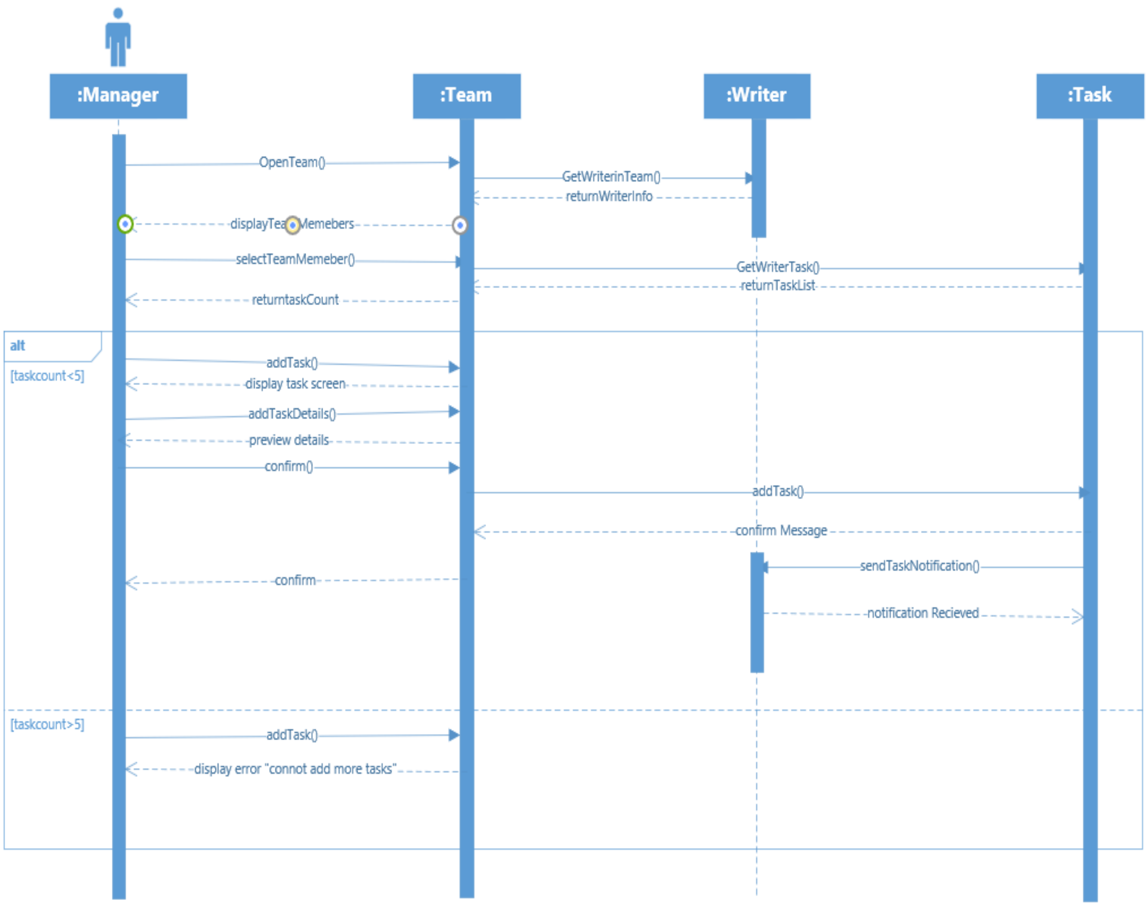


Figure 26: Sequence Diagram for Assign task to writer

4.2.3.1 Add Project

Manager opens the projects panel from where he goes to add a new project. The system displays a form to add the project details. If the fields are empty the user is not able to submit the project details, otherwise the detailed are added successfully.

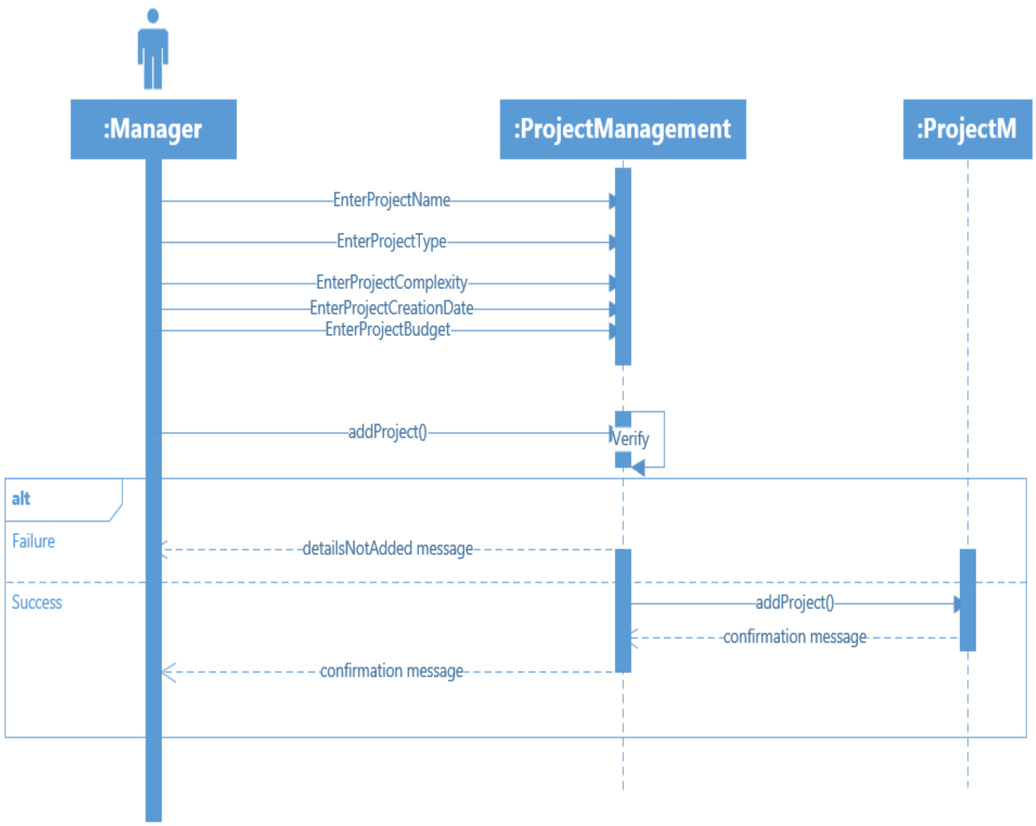


Figure 27: Sequence Diagram for Add project

4.2.3.1 View Projects

Manager can view the projects in the project panel. The project list is displayed to the manager. The details of the project will include the writers working on the project. The manager can filter out the ongoing projects to check their completion rate. The completed projects can also be viewed by changing the filter on the projects.

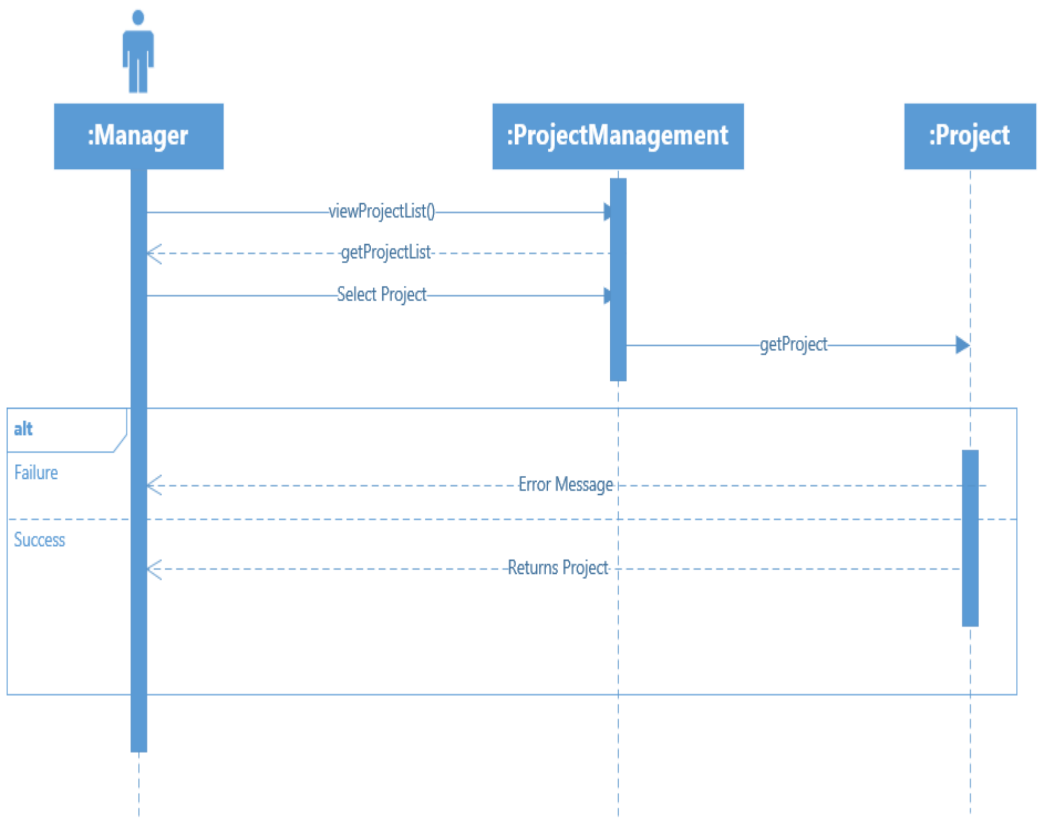


Figure 28: Sequence Diagram for View Projects

4.2.3.1 Sign In

User selects the role through which he/she wants to sign into the system. They then enter the credentials (username and password). The system validates the information, and the user is logged into the system.

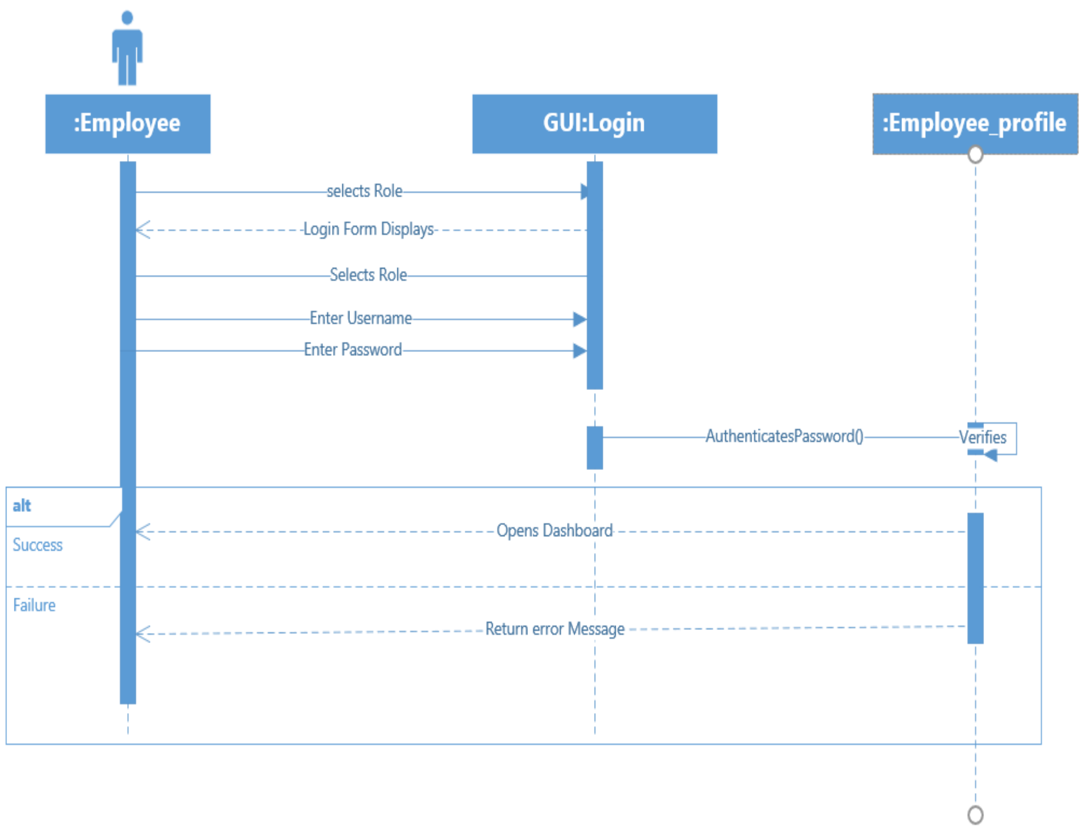


Figure 29: Sequence Diagram for Signin

4.2.3.1 Evaluate Task

The Manager/Team Lead opens the task panel on their side to view the tasks solutions submitted by the writers. The task solution contains the word count of the file, submission time, and file. The manager downloads the file and checks the quality of the task. If it is acceptable, the writer is notified and task is added into the task history.

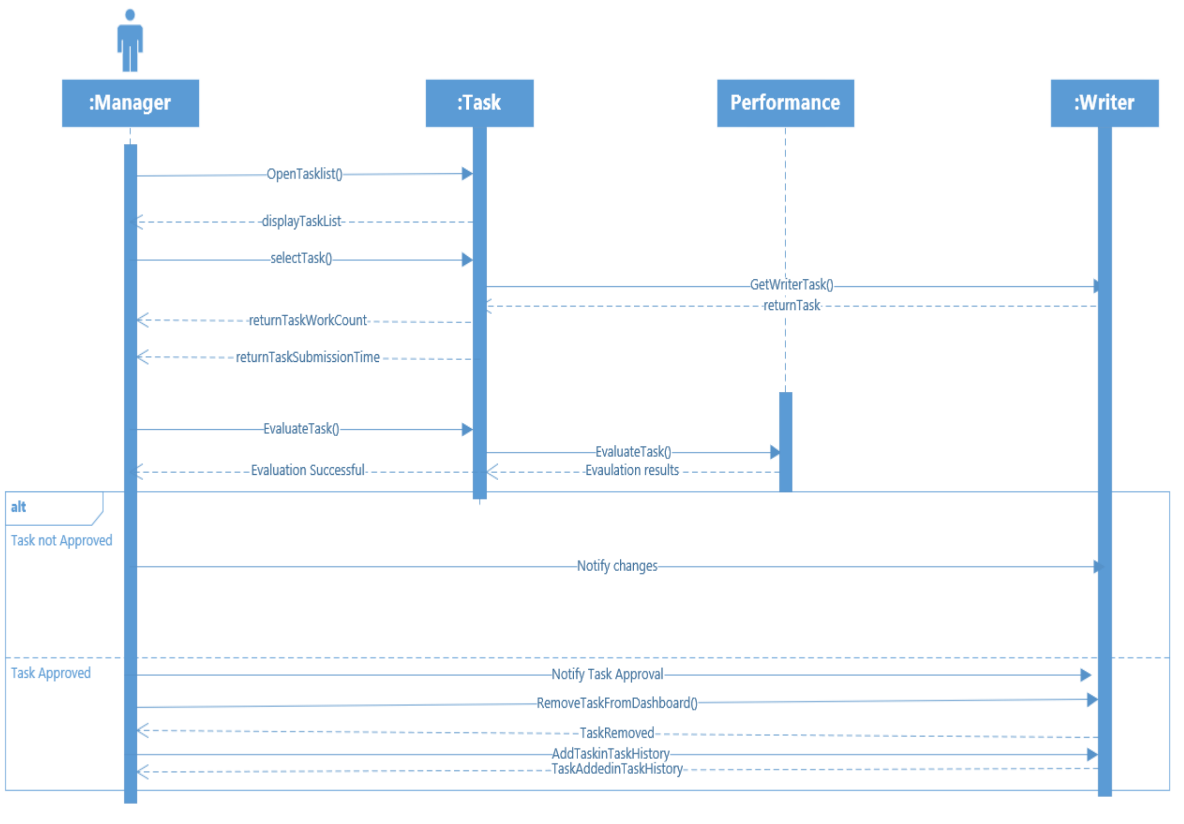


Figure 30: Sequence Diagram for Evaluate Task

4.2.3.1 Create Project Template

To create a new project template, the manager opens Template panel. He/she selects add a new template and fills out the name for the template system takes the name of the manager and the date at which the template is created and create a new template area.

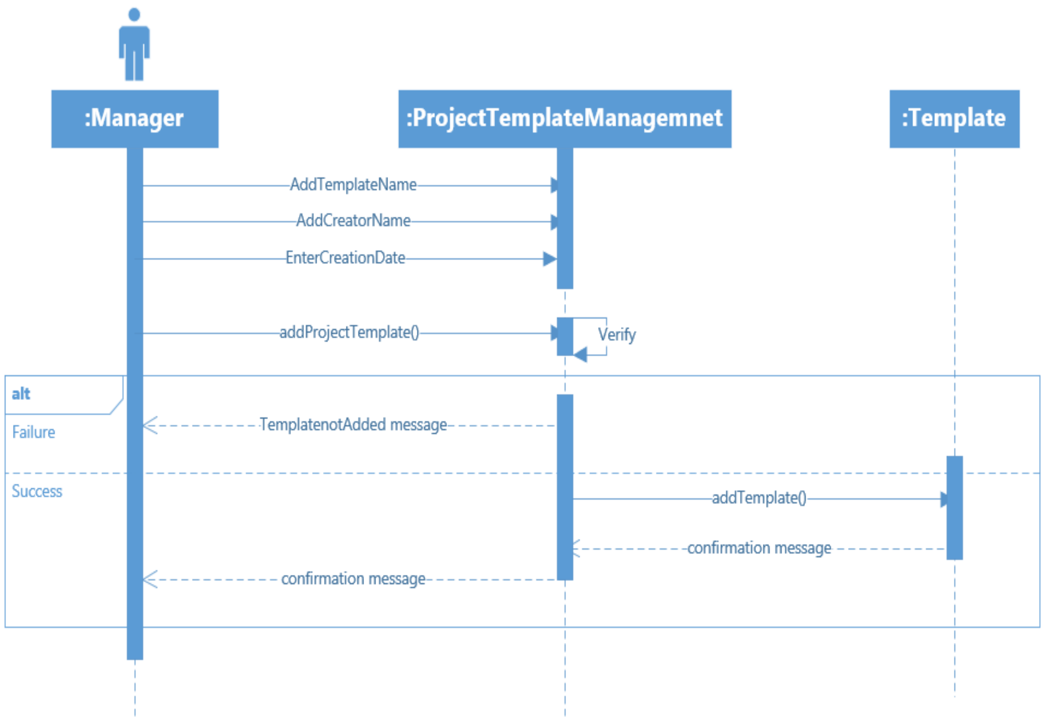


Figure 31: Sequence Diagram for Create Project template

4.2.3.1 Submit Task

The writer opens the assigned task from the task panel and attach the file where the see the option to choose file. The system checks for the file to be below 10MB then the file is uploaded.

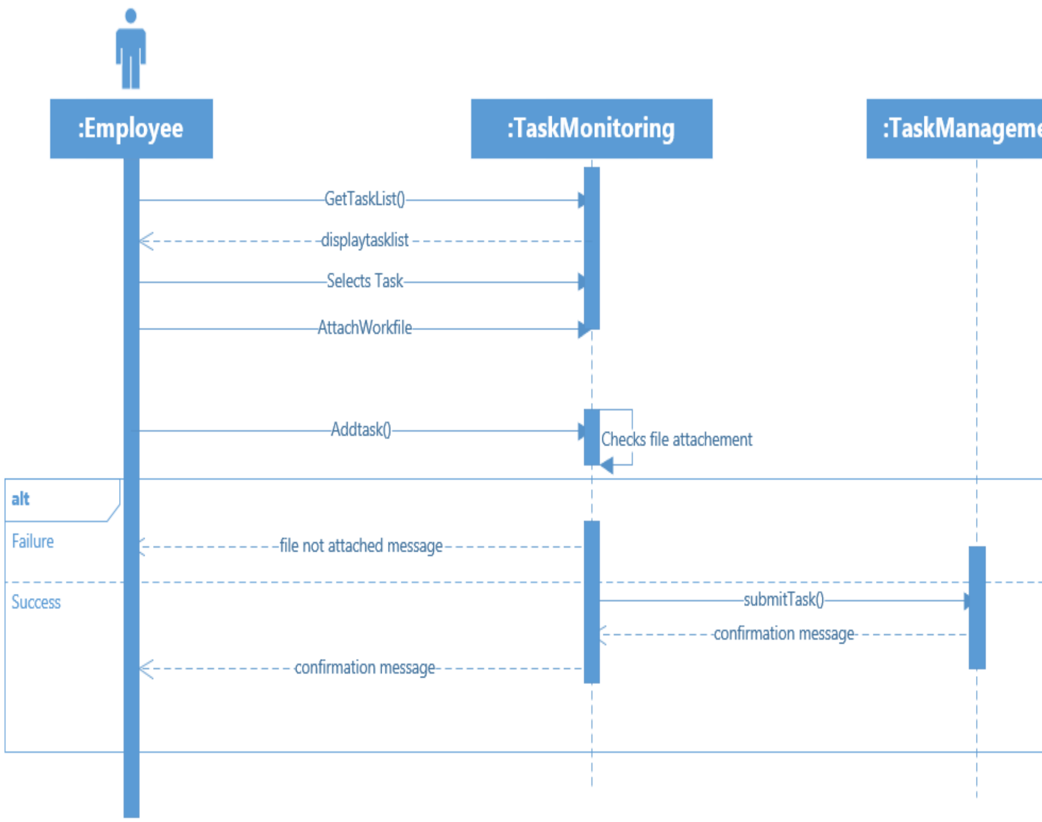


Figure 32: Sequence Diagram for submit task

4.2.3.1 Get Help

The system users select the "?" from the panel which will open a chat window. The user can add the queries their and chatbot will reply to the user queries.

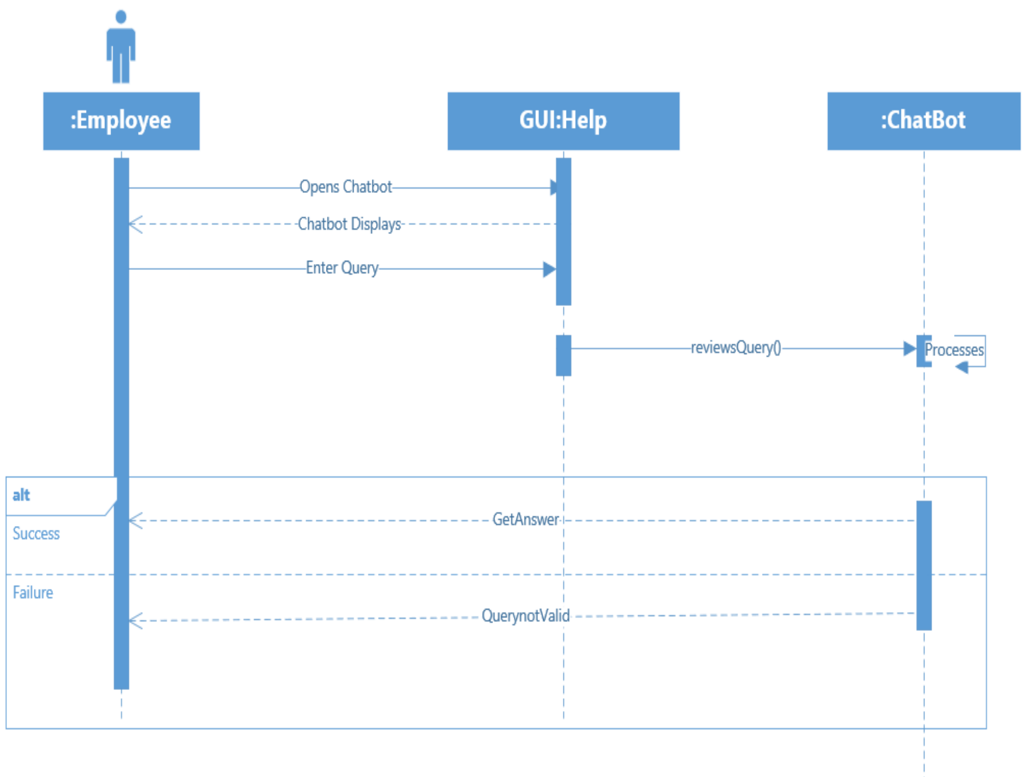


Figure 33: Sequence Diagram for Get Help

4.3 Data Design

Data design consist of a complete entity-relationship diagram of FVOA and the individual schemas for each mongoDB collection.

The following schemas are written in the NodeJS syntax for mongoose.

4.3.1 Manager Schema

```
var managerSchema = new schema({
	managerID: {type: int, required: true, autoincrement}
	manager_name: {type: String, required: true},
	manager_qualification: {type: String, required: true},
	manager_age: {type: String, required: true},
	manager_email: {type:String, required: true},
	manager_phone: {type:String, required: true},
	manager_username: {type:String, required: true},
	manager_password: {type:String, default: “fiesta123” }
});
```

4.3.2 TeamLead Schema

```
var teamLeadSchema = new schema({
	teamLeadID: {type: int, required: true, autoincrement}
	tl_name: {type: String, required: true},
	tl_qualification: {type: String, required: true},
	tl_age: {type: String, required: true},
	tl_email: {type:String, required: true},
	tl_phone: {type:String, required: true},
	tl_username: {type:String, required: true},
	tl_password: {type:String, default: “fiesta123” }
	managerID: {type: int, ref: ‘Manager’}
});
```

4.3.3 Writer Schema

```
var writerSchema = new schema({
        writerID: {type: int, required: true, autoincrement}
        writer_name: {type: String, required: true},
        writer _qualification: {type: String, required: true},
        writer _age: {type: String, required: true},
        writer _email: {type:String, required: true},
        writer _phone: {type:String, required: true},
        writer _username: {type:String, required: true},
        writer _password: {type:String, default: “fiesta123” }
        managerID: {type: int, ref: ‘Manager’}
        teamLeadID: {type: int, ref: ‘Teamlead’}

});
```

4.3.4 Trainee Schema

```
var traineeSchema = new schema({
        traineeID: {type: int, required: true, autoincrement}
        trainee_name: {type: String, required: true},
        trainee _qualification: {type: String, required: true},
        trainee _age: {type: String, required: true},
        trainee _email: {type:String, required: true},
        trainee _phone: {type:String, required: true},
        trainee _username: {type:String, required: true},
        trainee _password: {type:String, default: “fiesta123” }
        managerID: {type: int, ref: ‘Manager’}

});
```

4.3.5 Team Schema

```
var teamSchema = new schema({
        teamID: {type:int, required:true},
        team_name: {type:String, required:true},
        team_description: {type:String},
        creater: {type: int, ref: ‘Manager’}

});
```

4.3.6 TeamDetails Schema

```
var teamDetailsSchema = new schema({
        teamDetailsID: {type:int, required:true},
        teamID:{type: int, ref: ‘Team},
        teamLead:{type: int, ref: ‘Teamlead’},
        writer:{[ wid: {type: int, ref: ‘Writer’} ]}
```

```
});
```

4.3.7 Chat Schema

```
var chatSchema = new schema({
	chatID: {type:int, required: true},
	teamID: {type:int, ref: ‘Team’},
	senderID: {type:int, ref: ‘Employee’},
	send_date: {type:String, required: true},
	file: {type: String},
	message: {type: String}
}):
```

4.3.8 Training Video Schema

```
var trainingVideoSchema = new schema ({
	traineeVideoID: {type: int, required: true},
	video_name: {type:String, required: true},
	video_link: {type:String, required: true},
	quizID: {type:int, ref: ‘Quiz’},
});
```

4.3.9 Quiz Schema

```
var quizSchema = new schema ({
	quizID: {type: int, required: true},
	quiz_name: {type:String, required: true},
	allocated_time: {type:String, required: true},
	traineeID: {type:String, required: true},
	questions:{[ {type:int, ref: ‘Questions’}]},
});
```

4.3.10 Trainee Performance Schema

```
var traineePerformanceSchema = new schema ({
	tperID: {type: int, required: true},
	traineeID:[ {type:int, ref: ‘Trainee’},
	taskID:{[ {type:int, ref: ‘Ttask’}]},
	quizID:{[ {type:int, ref: ‘Quiz’}]},
});
```

4.3.11 Calender Schema

```
var calenderSchema = new schema ({
	calendarID: {type: int, required: true},
	userID: {type:int, ref: ‘Employee’},
```

```
});
```

4.3.12 Reminder Schema

```
var reminderSchema = new schema ({
        reminderID:  {type: int, required: true},
        reminder_title: {type:String, required: true},
        reminder_desc: {type:String, required: true},
        reminder_date: {type:String, required: true},
        reminder_time: {type:String, required: true},
        calenderID: {type:int, ref: 'Calender'},
});
```

4.3.13 Project Template Schema

```
var templateSchema = new schema ({
        templateID: {type: int, required: true},
        template_title: {type:String, required: true},
        template_creater: {type:int, ref: 'Manager'},
});
```

4.3.14 Project Schema

```
var projectSchema = new schema ({
        projectID: {type: int, required: true},
        project_name: {type:String, required: true},
        project_date: {type:String, required: true},
        project_type: {type:String, required: true},
        project_complexity: {type:String, required: true},
        project_budget: {type:String, required: true},
        project_completion: {type:int, default: 0}
        project_templateID: {type:int, ref: 'Template'},
});
```

4.3.15 ProjectDetails Schema

```
var projectDetailsSchema = new schema ({
        projectDetailsID: {type: int, required: true},
        projectID: {type:int, ref: 'Project' },
        writerID: {[ wid: {type:int, ref: 'Writer'}]}
        template_creater: {type:int, ref: 'Manager'},
});
```

4.4 Data Dictionary

- ***Admin***

Fields	Types	Description
Full Name	String	Super Admin's Full name
Email	String	Super Admin's Email
Password	String	Password for security

- ***Manager***

Fields	Types	Description
Full Name	String	Manager's Full name
Email	String	Manager's Email
Password	String	Password for security
Qualification	String	Qualification of Manager

- ***Trainee***

Fields	Types	Description
Full Name	String	Trainee's Full name
Email	String	Trainee's Email
Password	String	Password for security
Qualification	String	Qualification of Trainee

- ***Writer***

Fields	Types	Description
Full Name	String	Writer's Full name
Email	String	Writer's Email
Password	String	Password for security
Qualification	String	Qualification of Writer

- ***Employee***

Fields	Types	Description
Full Name	String	Employee full name
Designation	String	Designation of employee in the company.
Email	String	Email of the employee.
Department	String	Department of the employee where he works.
Attendance	Array	Employee’s attendance recorded here.

- ***Reminder***

Fields	Types	Description
Title	String	Title of reminder
Attachment	String	Link of attached project or task with reminder.

- ***Report***

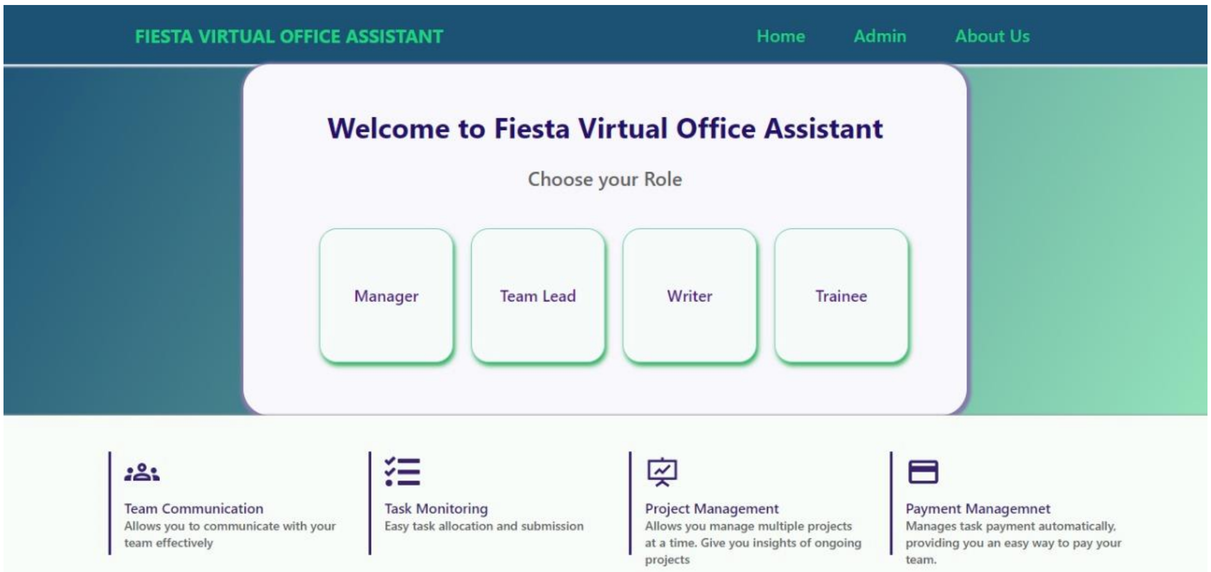
Fields	Types	Description
Title	String	Title of the report provided by Manager.
Admin	Object	Reference of object
File	Bytes	Word file uploaded by Manager

4.5 Human Interface Design

4.5.1 Screen Images

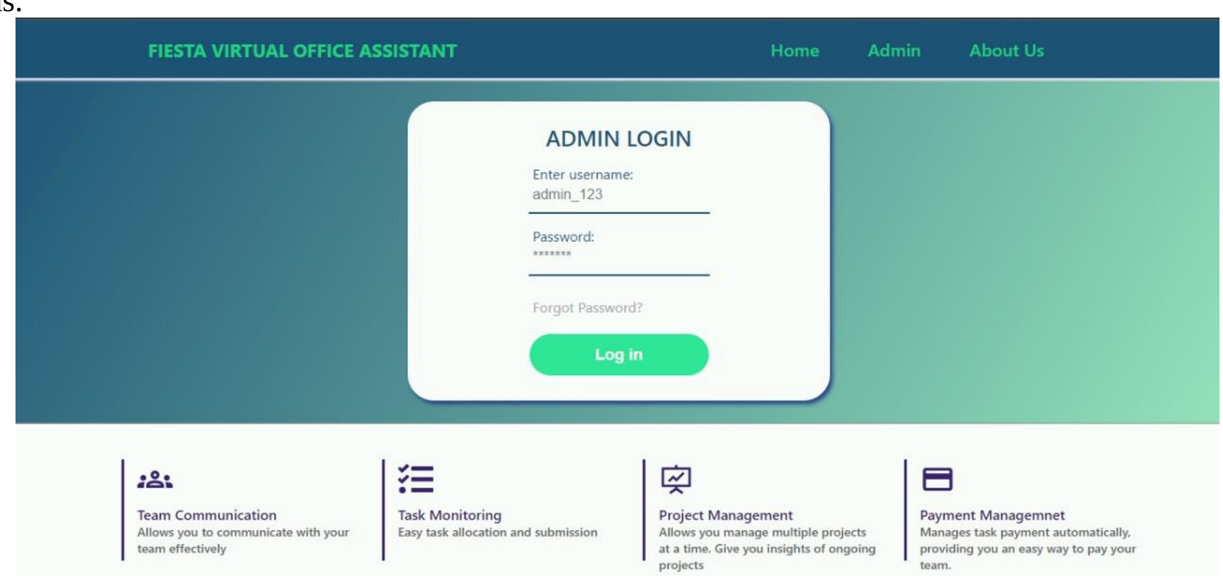
4.5.1.1 Landing Page

This mockup shows the arrival screen to the employees from where they can select their role and sign in to their profiles.

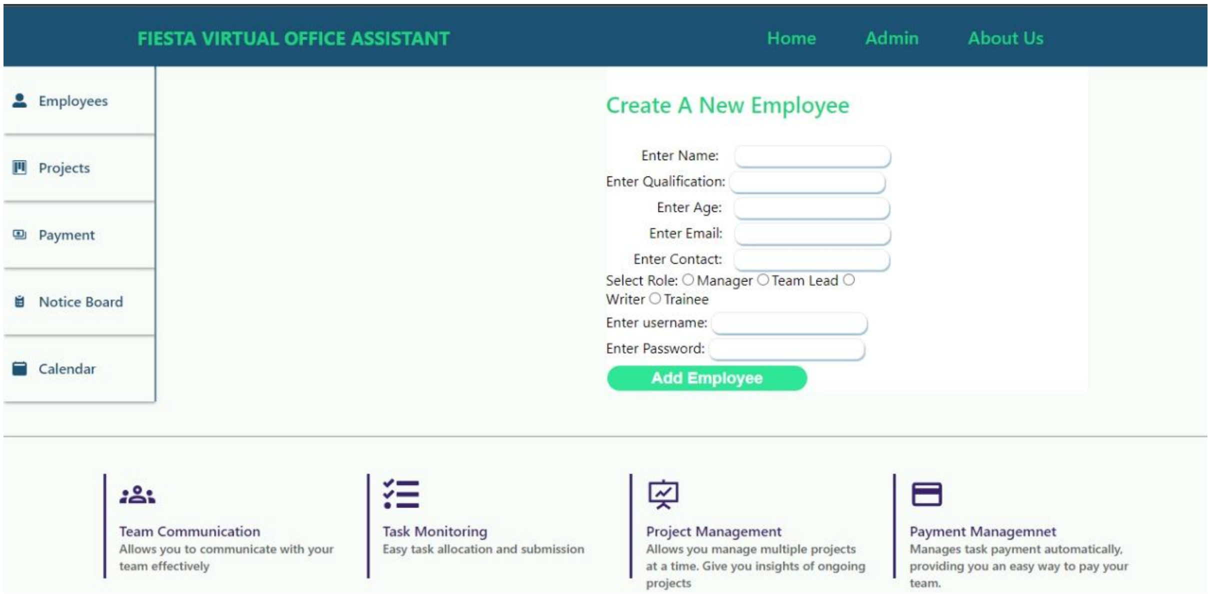


4.5.1.2 Admin Login Page

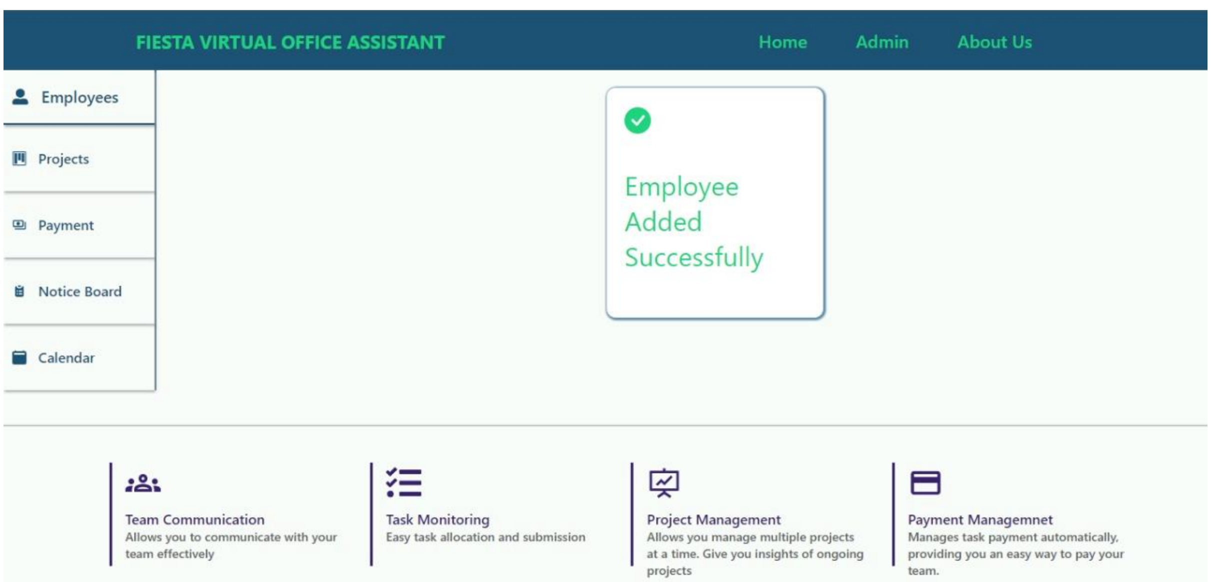
This mockup shows the Sign in screen where the admins can enter their username and password to enter their dashboards.



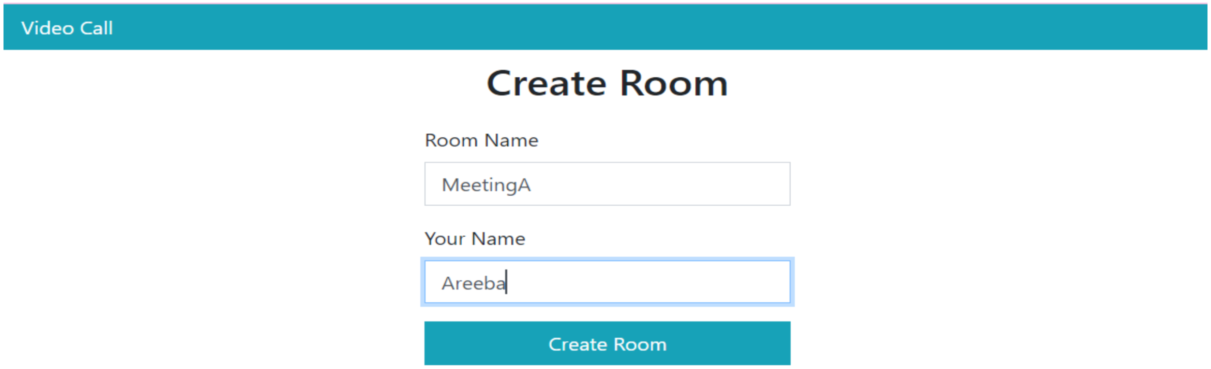
4.5.1.3 Create New Employee



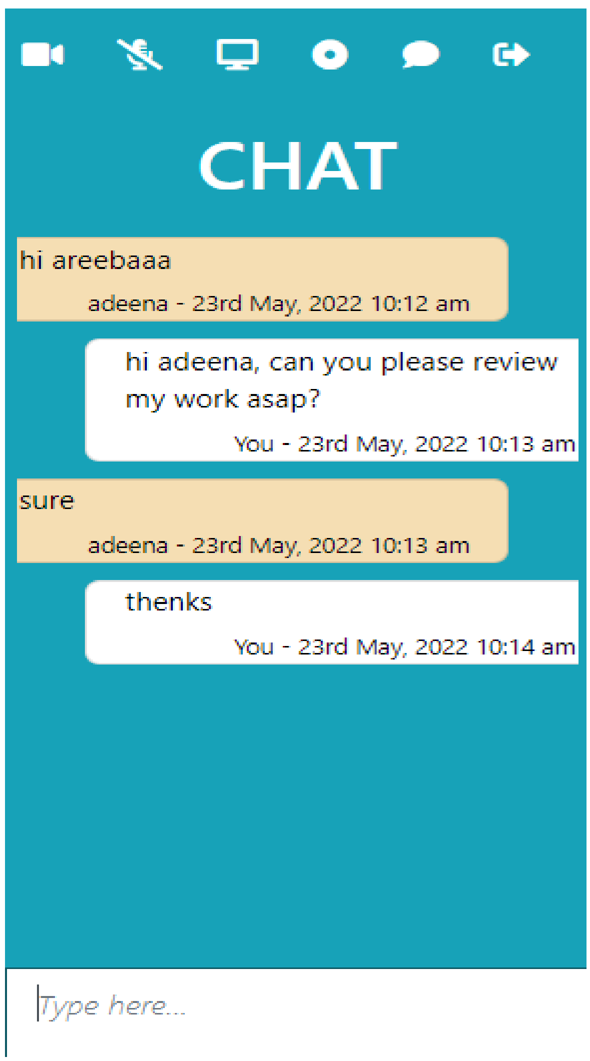
4.5.1.4 Employee Added Successfully



4.5.1.4.1 Create Room



4.5.1.5 Chat



5 Implementation

5.1 Algorithm

Algorithm 1 : Employee dashboard

```
use useHistory for navigation use use React and useState for tracking
use useSelector and useDispatch for using redux functionality

creating employee by name EmployeeCreation
employee has credentials named employeeData ,emp_name, emp_qualification, emp_age, emp_email,
emp_phone, emp_username', emp_password;

useing styles.css for styling useing icons to use in future
useing classes deleteEmployee, getEmployee ,useEffect  for fetching data
const Employees using dispatch const employees = useSelector((state) state.employees);
console.log(employees);
const navigate = useHistory(); const onProfile = (id) =     {
navigate.push(`/admin/employeedetails/${id}`);
}
const onDelete = (id) =     { dispatch(deleteEmployee(id));
}
const onAddEmployee = () =     { navigate.push(`admin/addemployee`);
}
useEffect(()=     { dispatch(getEmployee());
}, []);

return (
className="empinfo_container"
style={{display: 'flex', justifyContent: 'flex-end', alignItems: 'center', alignText: 'center' }}
style={{textAlign: 'center'}}     Employees List</h1
button className="btn_add_emp" onClick={onAddEmployee}     Add New Employee</button

{employees.map((employee) =     (
  key={employee._id} className="employee_record"
<di iv style={{display: 'flex', justifyContent: 'space-between'}}
  style={{width: '80%', justifyContent: 'flex-start !USEANT', textAlign:'left
!useant',alignItems:'flex-end' }}
 className="emp_name"     {employee.emp_name}

  style={{width: '20%', display: 'flex', flexDirection:'row'}}
  style={{width: '30%', marginTop: '15%' ,justifyContent: 'right' , textAlign: 'right'}}
button className="emp_btn" onClick={()=     onDelete(employee._id)}
    AiFillDelete/

  style={{width: '50%', justifyContent: 'flex-end', textAlign: 'right', marginTop: '17%', marginRight: '5%'}}
<button className="emp_profile_btn" onClick={() =     {
console.log(employee._id) onProfile(employee._id);
} // <Employee employee={employee} setCurrentId={setCurrentId} /
}
    Profile</button
```

```
  className="emp_details_container" style={{display: 'flex', justifyContent: 'space-around',
marginBottom: '2%'}}
  style={{display: 'flex', width: '45%', justifyContent: 'space-evenly'}}
  className="emp_title"    Email:
  className="emp_info"    {employee.emp_email}

  style={{display: 'flex', width: '50%', justifyContent: 'space-evenly'}}
  className="emp_title" Phone Number:
  className="emp_info" {employee.emp_phone}


export default Employees;
```

Algorithm 2: Admin Controller

```
  use mongoose from "mongoose";
  use Employee from models and Notice from models

use variable getAdmin = (req, res)   { send request to('This is admin!');
}

use variable createEmployee = async (req, res)   { variable employee = req.body;
variable newEmployee = new Employee(employee); try{
await newEmployee.save(); res.status(201).json(newEmployee) show ('Employee added');
catch (error handling
res.status(409).json({ show error message});

use variable getEmployee = async (req, res)   { try{
variable employees = await Employee.find(); console.log(employees); res.status(200).json(employees);
} catch (error){
res.status(404).json({show error message})
}
}
use variable updateEmployee = async (req, res)   { variable { id } = req.params;
variable emp = req.body; if(!mongoose.Types.ObjectId.isValid(id))
return res.status(404).send('No employee with that id');
variable updateEmployee = await Employee.findByIdAndUpdate(id, emp, {new: true} );
res.json(updateEmployee);
}
use variable getEmployeeDetails= async(req, res) { variable {id}= req.params;
console.log(id) try{
variable data= await Employee.findById(id);

res.status(200).json(data)
}catch(error){default Employees;
res.status(404).json({show error message})

}

}

use variable deleteEmployee = async (req, res)   { variable { id } = req.params;
if(!mongoose.Types.ObjectId.isValid(id)) return res.status(404).send("No user with this id"); await
Employee.findByIdAndRemove(id);
res.json({message: "User deleted successfully"});

}
```

```
//noticeboard
use variable createNotice = async (req, res)   { variable notice = req.body;
variable newNotice = new Notice(notice); try{
await newNotice.save(); res.status(201).json(newNotice); console.log('Notice added');
} catch (error){
res.status(409).json({ show error message});
}
}

use variable getNotices = async (req, res)   { try{
variable notices = await Notice.find(); console.log(notices); res.status(200).json(notices);
} catch (error){
res.status(404).json({show error message})
}
}

use variable deleteNotice = async (req, res)   { variable { id } = req.params;
if(!mongoose.Types.ObjectId.isValid(id)) return res.status(404).send("No user with this id"); await
Notice.findByIdAndRemove(id);
res.json({message: "User deleted successfully"});

}
```

5.2 External APIs/SDKs

No external APIs or SDKs used till now in our project.

5.3 User Interface

5.3.1.1 Login Screen

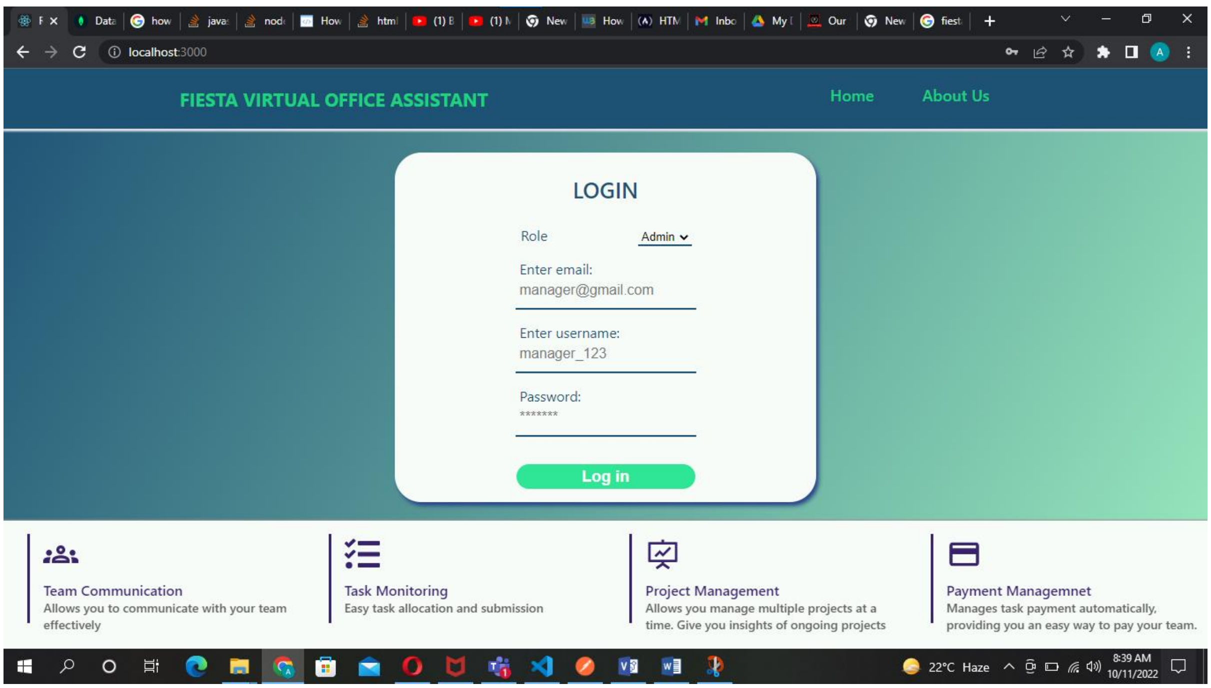


Figure 34: UI screen Login for of FVOA

5.3.1.2 *About us*

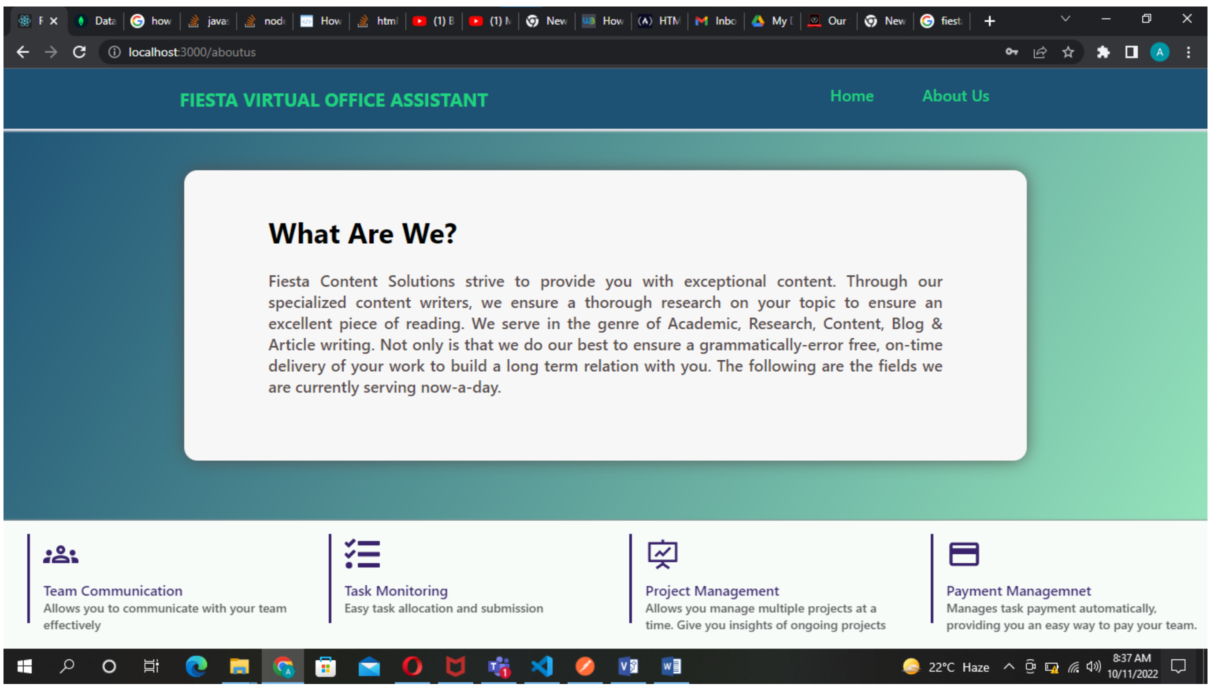


Figure 35: UI screen for Aboutus of FVOA

5.3.1.3 Create employee

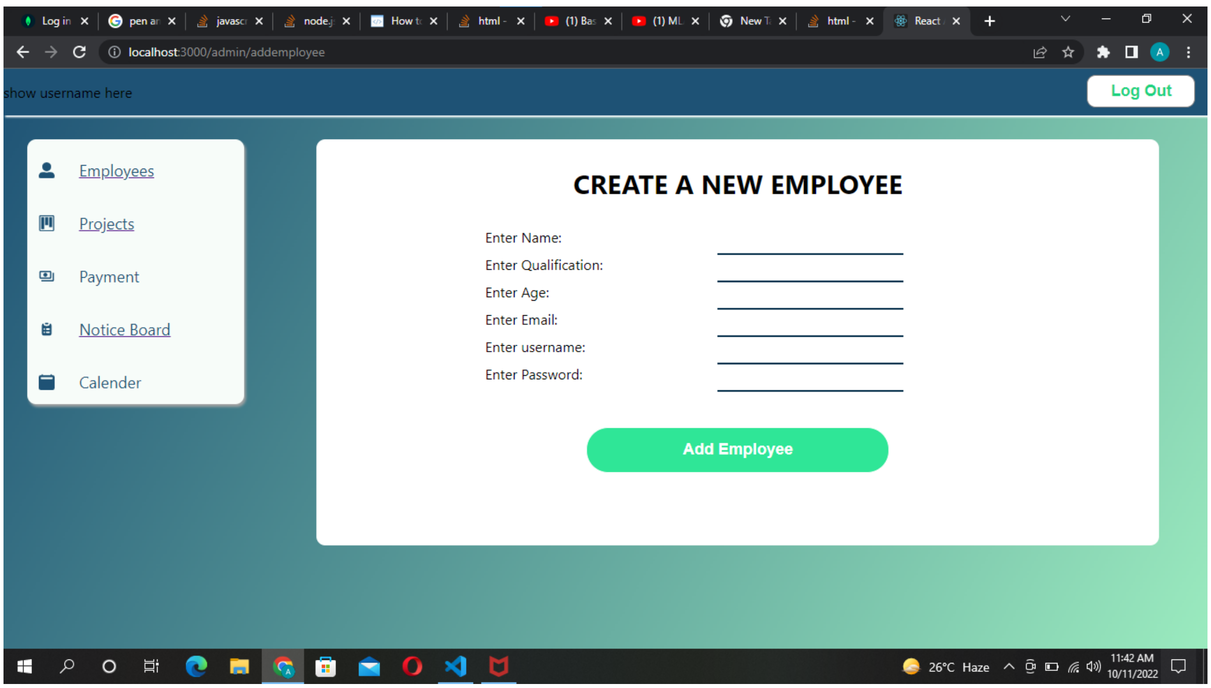


Figure 36: UI screen for create employee of FVOA

5.3.1.4 Employees list

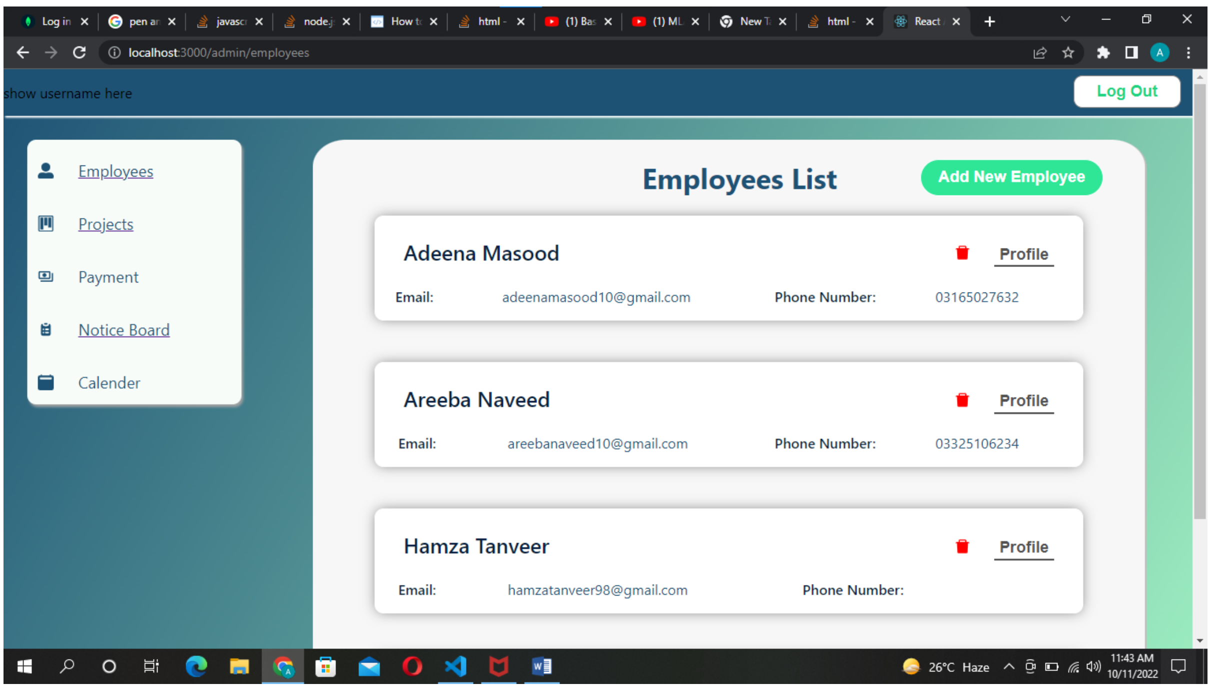


Figure 37: UI screen for employee list of FVOA

5.3.1.5 Notice board

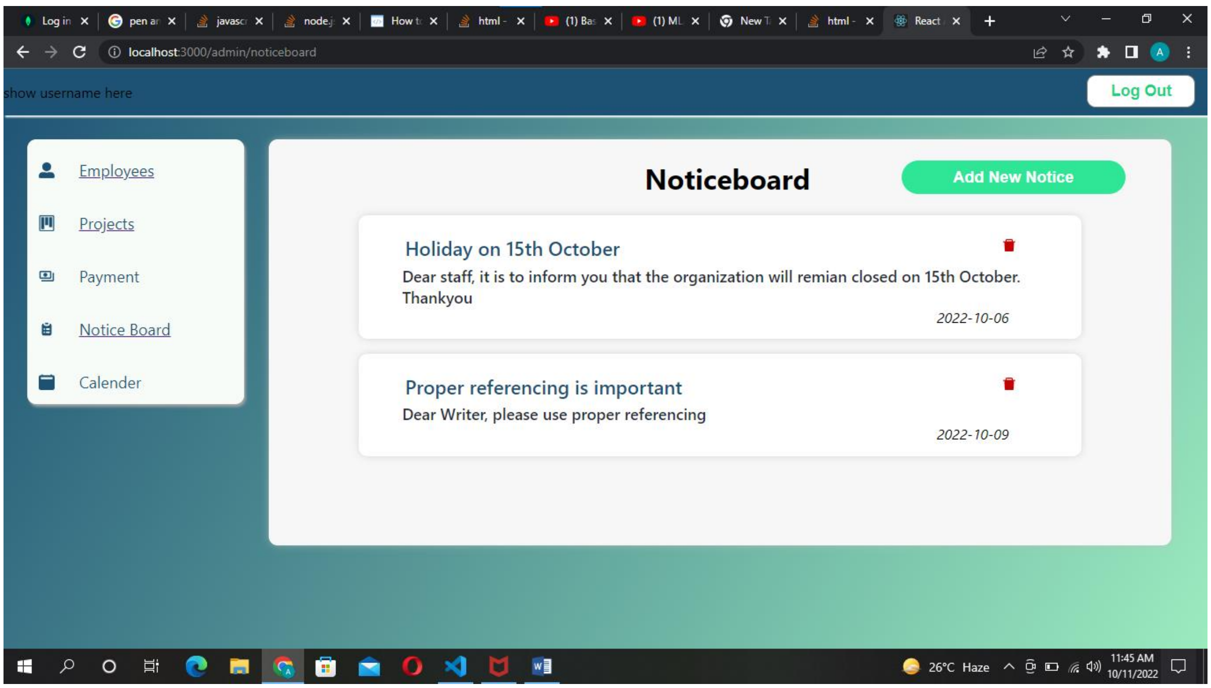


Figure 38: UI screen for Notice Board of FVOA

5.4 Deployment

This project will be deployed and hosted on Heroku. Heroku is a cloud platform as a service supporting several programming languages. One of the first cloud platforms.

6 Testing and Evaluation

6.1 Unit Testing

Unit Testing 1: Sign up as new user
Testing Objective: To ensure the sign-up form is working properly.

Table 134: Unit Testing 01

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check if the name text field takes text	Areeba Naveed	It receives the text shows in text box	Pass
2	Check if user entered email exists	Email: areeba19.gmail.com	Shows email is valid message	Pass
3	Check if user entered email is invalid	Email Abc2@@gmil.cpm	Shows error message	Pass
4	Check if password matches with confirm password	223565	Shows password matches	Pass
5	Check if password does not match with confirm password	@12@35	Shows error message	Pass
6	Check if sign up button works	Sign up button	Redirects to confirmation page	Pass

Unit Testing 2: Login as employee with valid and invalid credentials
Testing Objective: To ensure the login form is working correctly with valid and invalid credentials/inputs when registered as Employee.

Table 135: Unit Testing 02

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the email field of login to validate that it takes proper email	Email: areebanaveed19@gmail.com	Validates email address and moves cursor to next textbox	Pass
2	Check the email field of login to validate that it displays error message.	Email: areebanaveed19.gmail.com	Highlights field and displays error message	Pass
3	Check the field of password if it takes proper password	Qwerty123	Validates password and hides it	Pass
4	Check if it shows error message on wrong password	112233	It shows error popup	Pass
5	Check if login button works	Login button	It validates credentials and redirects to next page	Pass

Unit Testing 3: Login as Manager with valid and invalid credentials
Testing Objective: To ensure the login form is working correctly with valid and invalid credentials/inputs when registered as Manager.

Table 136: Unit Testing 03

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the email field of login to validate that it takes proper email	Email: areebanaveed19@gmail.com	Validates email address and moves cursor to next textbox	Pass
2	Check the email field of login to validate that it displays error message.	Email: areebanaveed19.gmail.com	Highlights field and displays error message	Pass
3	Check the field of password if it takes proper password	Qwerty123	Validates password and hides it	Pass
4	Check if it shows error message on wrong password	112233	It shows error popup	Pass
5	Check if login button works	Login button	It validates credentials and redirects to next page	Pass

Unit Testing 4: Register as Trainee

Testing Objective: To ensure the registration form is working correctly when registering as a trainee.

Table 137: Unit Testing 04

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the email field of registration to validate that it takes proper email	Email: abc@gmail.com	Validates email address	Pass
2	Check the email address and validate that it is unique and not already registered.	Email: abc@gmail.com	Moves to next field if email is unique otherwise error message is displayed	Pass
3	Check the description field and validate that the trainee can add his description	Description: I have 5 months' experience	The description is added successfully	Pass
4	Check the degrees dropdown and validate that the trainee can select his degrees	Degrees: BA, Diploma	The degrees are added successfully	Pass

Unit Testing 5: Login with valid and invalid credentials
Testing Objective: To ensure the login form is working correctly with valid and invalid credentials/inputs.

Table 138: Unit Testing 05

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the email field of login to validate that it takes proper email	Email: abc@gmail.com	Validates email address and moves cursor to next textbox	Pass
2	Check the email field of login to validate that it displays error message.	Email: abc@gmail.com	Highlights field and displays error message	Pass
3	Check and validate that the password is displayed as steric and not plain text.	Password: ****	Password is displayed as steric and not plain text	Pass

Unit Testing 6: Update employee profile with new credentials
Testing Objective: To ensure the input fields of update employee profile are working correctly.

Table 139: Unit Testing 06

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the email field of update profile to validate that it takes proper email	Email: xyz@gmail.com	Validates email address and moves cursor to next textbox	Pass
2	Check the email field of update profile to validate that it is unique.	Email: abc@gmail.com	Moves to next field if email is unique otherwise error message is displayed	Pass
3	Check if current password field shows password as steric and not plain text	Current Password: ****	Password is displayed as steric and not plain text	Pass
4	Ensure if current password is matched with the saved password	Current Password: ****	Password is matched with the saved password database. If correct, the user can add new password otherwise error is displayed	Pass
5	Check and validate that the new password field displays steric and not plain text	New password: ****	Password is displayed as steric and not plain text	Pass

Unit Testing 7: Update trainee profile with new credentials
Testing Objective: To ensure the input fields of update Trainee profile are working. correctly.

Table 140: Unit Testing 07

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the email field of update profile to validate that it takes proper email	Email: xyz@gmail.com	Validates email address and moves cursor to next textbox	Pass
2	Check the email field of update profile to validate that it is unique.	Email: abc@gmail.com	Moves to next field if email is unique otherwise error message is displayed	Pass
3	Check if current password field shows password as steric and not plain text	Current Password: ****	Password is displayed as steric and not plain text	Pass
4	Ensure if current password is matched with the saved password	Current Password: ****	Password is matched with the saved password database. If correct, the user can add new password otherwise error is displayed	Pass
5	Check and validate that the new password field displays steric and not plain text	New password: ****	Password is displayed as steric and not plain text	Pass

Unit Testing 8: Update profile with new credentials
Testing Objective: To ensure the input fields of update manager profile are working correctly.

Table 141: Unit Testing 08

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the email field of update profile to validate that it takes proper email	Email: xyz@gmail.com	Validates email address and moves cursor to next textbox	Pass
2	Check the email field of update profile to validate that it is unique.	Email: abc@gmail.com	Moves to next field if email is unique otherwise error message is displayed	Pass
3	Check if current password field shows password as steric and not plain text	Current Password: ****	Password is displayed as steric and not plain text	Pass
4	Ensure if current password is matched with the saved password	Current Password: ****	Password is matched with the saved password database. If correct, the user can add new password otherwise error is displayed	Pass
5	Check and validate that the new password field displays steric and not plain text	New password: ****	Password is displayed as steric and not plain text	Pass

Unit Testing 9: View Employee Credentials
Testing Objective: To ensure the Employee details are shown correctly when button is pressed

Table 142: Unit Testing 09

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the "View Employee Details" button is working correctly	Press "View Employee Details" button	Employee details are shown correctly after pressing the button	Pass

Unit Testing 10: View Manager details
Testing Objective: To ensure the Manager details are shown correctly when button is pressed.

Table 143: Unit Testing 10

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the "View Manager Details" button is working correctly	Press "View Manager Details" button	Manager details are shown correctly after pressing the button	Pass

Unit Testing 11: View Trainee details
Testing Objective: To ensure the Trainee details are shown correctly when button is pressed

Table 144: Unit Testing 11

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the "View Trainee Details" button is working correctly	Press "View Trainee Details" button	Trainee details are shown correctly after pressing the button	Pass

Unit Testing 12: View Writer details
Testing Objective: To ensure the Writer details are shown correctly when button is pressed

Table 145: Unit Testing 12

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the "View Writer Details" button is working correctly	Press "View Writer Details" button	Writer details are shown correctly after pressing the button	Default

Unit Testing 13: View project details screen
Testing Objective: To ensure the product details are displayed correctly.

Table 146: Unit Testing 13

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the "View Project Details" button is working correctly	Press "View project Details" button	Project details are displayed correctly	Pass

Unit Testing 14: View Task details screen
Testing Objective: To ensure the product details are displayed correctly.

Table 147: Unit Testing 14

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the “View Task Details” button is working correctly	Press “View Task Details” button	Task details are displayed correctly	Pass

Unit Testing 15: View Calendar details screen
Testing Objective: To ensure the calendar details are displayed correctly.

Table 148: Unit Testing 15

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the “Calendar” click button is working correctly	Press “Calendar " button	Calendar is displayed correctly	Pass

Unit Testing 16: Add Calendar details
Testing Objective: To ensure the calendar details added correctly.

Table 149: Unit Testing 16

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check the “Add Reminder” button is working.	Press “Add Reminder " button	Reminder is added successfully.	Pass
2	Check the “Day” button is working.	Press “Day " button	Day wise schedule opens.	Pass
3	Check the “Month” button is working.	Press “Month" button	Month wise schedule opens.	Pass
4	Check “Add Agenda” Button is working.	Press “Agenda” button.	Add Agenda form pops up.	Pass

Unit Testing 17: Edit and update Calendar
Testing Objective: To ensure that the Calendar can easily change different fields.

Table 150: Unit Testing 17

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check that all fields of calender are changeable when edit button activated	Click the edit button on calender	All fields become editable and changes can be made	Pass
2	Validate all the fields of the calender after editing and save the changes to database.	Save button is clicked	All fields are valid according to desire format and data saved	Pass

Unit Testing 18: Edit and update Employee profile
Testing Objective: To ensure that the Employee can easily change different fields of their Profile.

Table 151: Unit Testing 18

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check that all fields of profile are changeable when edit button activated	Click the edit button on profile	All fields become editable and changes can be made	Pass
2	Validate all the fields of the profile after editing and save the changes to database.	Save button is clicked	All fields are valid according to desire format and data saved	Pass

Unit Testing 19: Employee logout from device.
Testing Objective: To ensure that the Employee can easily logout from the device anytime.

Table 152: Unit Testing 19

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Employee can logout of device when done using the services.	Employee click logout icon on side drawer and click OK on confirmation message screen	Employee logout successfully.	Pass
2	Employee cannot logout of device when done using the services if they cancel the logout operation	Employee click logout icon on side drawer and click Cancel on confirmation message screen	Employee could not logout.	Pass

Unit Testing 20: Manager logout from device.
Testing Objective: To ensure that the Manager can easily logout from the device anytime.

Table 153: Unit Testing 20

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Manager can logout of device when done using the services.	Manager click logout icon on side drawer and click OK on confirmation message screen	Manager logout successfully.	Pass
2	Manager cannot logout of device when done using the services if they cancel the logout operation	Manager click logout icon on side drawer and click Cancel on confirmation message screen	Manager could not logout.	Pass

Unit Testing 21: Trainee logout from device.
Testing Objective: To ensure that the Trainee can easily logout from the device anytime.

Table 154: Unit Testing 21

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Trainee can logout of device when done using the services.	Trainee click logout icon on side drawer and click OK on confirmation message screen	Trainee logout successfully.	Pass
2	Trainee cannot logout of device when done using the services if they cancel the logout operation	Trainee click logout icon on side drawer and click Cancel on confirmation message screen	Trainee could not logout.	Pass

Unit Testing 22: File uploading is working with progress bar.
Testing Objective: To ensure that the file uploading is working and progress bar updates accordingly.

Table 155: Unit Testing 22

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check that the Upload file button is working.	User clicks on the upload file button.	The upload file button gets clicked.	Pass
2	Check that the select file from device window pops up.	User browses and selects the file from the device.	The selected file is pinned.	Pass
3	Check that the file selected starts uploading.	The pinned file should start to upload.	The pinned file starts to upload.	Pass
4	Checks that the progress bar starts to update as file gets uploaded.	The file starts uploading.	The file is uploaded.	Pass
5	Checks that the file uploaded tag pops up.	User waits for the pop up to show.	The file uploaded tag pops up.	Pass

Unit Testing 23: File downloading is working with progress bar.
Testing Objective: To ensure that the file downloading is working and progress bar updates accordingly.

Table 156: Unit Testing 23

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check that the Download file button is working.	User clicks on the Download file button.	The Download file button gets clicked.	Pass
2	Check that the select file from updated files.	User browses and selects the file from the uploaded files.	The selected file is pinned.	Pass
3	Check that the file selected starts downloading.	The pinned file should start to download.	The pinned file starts to download.	Pass
4	Checks that the progress bar starts to update as file gets downloaded.	The file starts downloading.	The file is downloaded.	Pass
5	Checks that the file download tag pops up.	User waits for the pop up to show.	The file downloaded tag pops up.	Pass

Unit Testing 24: Task evaluations.
Testing Objective: To ensure that the task evaluations sent are sent correctly.

Table 157: Unit Testing 24

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Task evaluations are sent correctly to database.	Files of evaluations are sent to database.	The files are uploaded in database.	Pass
2	Results of evaluations are shown in graphs.	Graphs are generated on the basis of evaluation results.	The graphs are generated.	Pass

Unit Testing 25: View Reviews of employees
Testing Objective: To ensure that the employees reviews can be seen.

Table 158: Unit Testing 25

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check that the employees reviews can be seen	Click on the view employee evaluations.	The employees ratings are shown in form of stars.	Default

Unit Testing 26: Gant charts formation
Testing Objective: To ensure that the gant charts are formed correctly.

Table 159: Unit Testing 26

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Check if gant charts formed are correctly based on task.	Tasks are added in the projects.	Gant chart updates when the task is added in the project.	Default

6.2 Functional Testing

The functional testing will take place after the unit testing. In this functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements.

Functional Testing 1: Sign Up with Valid and Invalid Credentials and redirected to login page
Testing Objective: To ensure the signup form is working correctly with valid and invalid inputs and user is redirect to SignIn page.

Table 160: Functional Testing 01

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if user can signup into the system successfully with valid inputs and redirect to the signIn page.	Email: abc@gmail.com Name: Areeba NAveed Password: qwerty123456	Validates the input fields on signup page and successfully redirect to the signIn page.	Input data is valid and user is redirect to signIn page after successful signup.	Pass
2.	Check if user cannot signup into the system successfully with invalid inputs and error message is display.	Email: abc.gmail.com Name: Areeba Naveed Password: 123456	Validates the input fields on signup page, display and highlight the error message.	Input data is invalid and user display an error message,	Pass

Functional Testing 2: Login with Employee Credentials.
Objective: To ensure that the correct page with the correct navigation bar is loaded.

Table 161: Functional Testing 02

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if user can login into the system successfully with valid credentials and "Login success" message display	Email: abc@gmail.com Password: areeba123456	Validates email address and password, user logged in and "Login successful" message display.	User Logged in and success message display.	Pass
2.	Check if user cannot login into the system with invalid credentials and "Login failed" message display	Email: abc@gmail.com Password: 123456	Validates email address and password, user login failed and "Login failed" message display.	User cannot Login and error message display.	Pass

Functional Testing 3: Login with Trainee Credentials.
Objective: To ensure that the correct page with the correct navigation bar is loaded.

Table 162: Functional Testing 03

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if user can login into the system successfully with valid credentials and "Login success" message display	Email: abc@gmail.com Password: areeba123456	Validates email address and password, user logged in and "Login successful" message display.	User Logged in and success message display.	Pass
2.	Check if user cannot login into the system with invalid credentials and "Login failed" message display	Email: abc@gmail.com Password: 123456	Validates email address and password, user login failed and "Login failed" message display.	User cannot Login and error message display.	Pass

Functional Testing 4: Login with writer Credentials.
Objective: To ensure that the correct page with the correct navigation bar is loaded.

Table 163: Functional Testing 04

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if user can login into the system successfully with valid credentials and "Login success" message display.	Email: abc@gmail.com Password: areeba123456	Validates email address and password, user logged in and "Login successful" message display.	User Logged in and success message display.	Default
2.	Check if user cannot login into the system with invalid credentials and "Login failed" message display	Email: abc@gmail.com Password: 123456	Validates email address and password, user login failed and "Login failed" message display.	User cannot Login and error message display.	Default

Functional Testing 5: Login with Manager Credentials.
Objective: To ensure that the correct page with the correct navigation bar is loaded.

Table 164: Functional Testing 05

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if user can login into the system successfully with valid credentials and "Login success" message display	Email: abc@gmail.com Password: areeba123456	Validates email address and password, user logged in and "Login successful" message display.	User Logged in and success message display.	Pass
2.	Check if user cannot login into the system with invalid credentials and "Login failed" message display	Email: abc@gmail.com Password: 123456	Validates email address and password, user login failed and "Login failed" message display.	User cannot Login and error message display.	Pass

Functional Testing 6: Login with Admin Credentials.
Objective: To ensure that the correct page with the correct navigation bar is loaded.

Table 165: Functional Testing 06

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if user can login into the system successfully with valid credentials and “Login success” message display	Email: abc@gmail.com Password: areeba123456	Validates email address and password, user logged in and “Login successful” message display.	User Logged in and success message display.	Pass
2.	Check if user cannot login into the system with invalid credentials and “Login failed” message display	Email: abc@gmail.com Password: 123456	Validates email address and password, user login failed and “Login failed” message display.	User cannot Login and error message display.	Pass

Functional Testing 7: Logout as Admin

Objective: To ensure that the admin is logged out from its respective accounts

Table 166: Functional Testing 07

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Admin tries to logout	NIL	Alert message ’Are you sure you want to Logout?’	Alert message ’Are you sure you want to Logout?’	Pass
2.	Verify After logout tries to re-login with the same or different account it’s allowing or not	NIL	Main page for the admin is loaded	Main page for the admin is loaded	Pass

Functional Testing 8: Logout as Trainee

Objective: To ensure that the Trainee is logged out from its respective accounts

Table 167: Functional Testing 08

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Trainee tries to logout	NIL	Alert message ’Are you sure you want to Logout?’	Alert message ’Are you sure you want to Logout?’	Pass
2.	Verify After logout tries to re-login with the same or different account it’s allowing or not	NIL	Main page for the admin is loaded	Main page for the Trainee is loaded	Pass

Functional Testing 9: Logout as Manager

Objective: To ensure that the Manager is logged out from its respective accounts

Table 168: Functional Testing 09

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Manager tries to logout	NIL	Alert message ’Are you sure you want to Logout?’	Alert message ’Are you sure you want to Logout?’	Pass
2.	Verify After logout tries to re-login with the same or different account it’s allowing or not	NIL	Main page for the admin is loaded	Main page for the Manager is loaded	Pass

Functional Testing 10: Logout as Writer

Objective: To ensure that the Writer is logged out from its respective accounts

Table 169: Functional Testing 10

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Writer tries to logout	NIL	Alert message ’Are you sure you want to Logout?’	Alert message ’Are you sure you want to Logout?’	Pass
2.	Verify After logout tries to re-login with the same or different account it’s allowing or not	NIL	Main page for the admin is loaded	Main page for the Writer is loaded	Pass

Functional Testing 11: Trainee view profile
Objective: To ensure that the trainee can view their respective profile information

Table 170: Functional Testing 11

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify that system redirecting user to valid page or not when user click on view profile	Nil	Successfully redirects to the Trainee profile page	Successfully redirects to the Trainee profile page	Pass

Functional Testing 12: Employee view profile
Objective: To ensure that the employee can view their respective profile information

Table 171: Functional Testing 12

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify that system redirecting user to valid page or not when user click on view profile	Nil	Successfully redirects to the Employee profile page	Successfully redirects to the Employee profile page	Pass

Functional Testing 13: Writer view profile

Objective: To ensure that the writer can view their respective profile information

Table 172: Functional Testing 13

No.	**Test case/Test script**	**Attribute and value**	**Expected result**	**Actual Result**	**Result**
1	Verify that system redirecting user to valid page or not when user click on view profile	Nil	Successfully redirects to the writer profile page	Successfully redirects to the writer profile page	Pass

Functional Testing 14: Employee edit profile

Objective: To ensure that the employee can edit their profile information

Table 173: Functional Testing 14

No.	**Test case/Test script**	**Attribute and value**	**Expected result**	**Actual Result**	**Result**
1	Check that employee be able to update profile details	Username: Areeba Naveed Email: aya19@gmail.com Age : 31 Qualification: CS	Successfully updates the profile	Successfully updates the profile	Pass
2.	Check that updated details showing or not when user refresh the page	Nil	Successfully displays the updated profile	Successfully displays the updated profile	Pass
3.	If the registered employee enters invalid information while editing their profile	Username: noor Email: mahnoor.gmail.com	Display warnings and prompt the user to enter correct credentials.	Display warnings and prompt the employee to enter correct credentials.	Pass

Functional Testing 15: Trainee edit profile

Objective: To ensure that the Trainee can edit their profile information

Table 174: Functional Testing 15

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that Trainee be able to update profile details	Username: Areeba Naveed Email: aya19@gmail.com Age : 31 Qualification: CS	Successfully updates the profile	Successfully updates the profile	Pass
2.	Check that updated details showing or not when user refresh the page	Nil	Successfully displays the updated profile	Successfully displays the updated profile	Pass

Functional Testing 16: Manager edit profile
Objective: To ensure that the manager can edit their profile information

Table 175: Functional Testing 16

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that manager be able to update profile details	Username: Areeba Naveed Email: aya19@gmail.com Age : 31 Qualification: CS	Successfully updates the profile	Successfully updates the profile	Pass
2.	Check that updated details showing or not when user refresh the page	Nil	Successfully displays the updated profile	Successfully displays the updated profile	Pass
3.	If the registered employee enters invalid information while editing their profile	Username: noor Email: mahnoor.gmail.com	Display warnings and prompt the user to enter correct credentials.	Display warnings and prompt the manager to enter correct credentials.	Pass

Functional Testing 17: Admin deletes profile
Objective: To ensure that the admin can delete their profile with correct credentials

Table 176: Functional Testing 17

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check if the account is deleted by entering incorrect password after clicking on the Delete account button	Password: (incorrect)12345	Display 'Password is incorrect. Try Again'	Display 'Password is incorrect. Try Again'	Pass
2.	Check if the account is deleted by entering correct password after clicking on the Delete account button	Password: 1234567	Account successfully deleted and redirects to the student login page	Account successfully deleted and redirects to the student login page	Pass

Functional Testing 18: Trainee uploads work
Objective: To ensure that the trainee can upload files

Table 177: Functional Testing 18

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check Trainee is able to view upload files	NIL	Page is displayed	Page is displayed	Default
2.	Check if the Trainee can upload files.	NIL	Trainee selects file and uploads.	Selected file is uploaded.	Default
3.	Check if the Trainee can save the displayed report	Name: report id	The system shall prompt the Trainee to enter complete description of file.	The system prompts the trainee to enter complete description of file.	Default

Functional Testing 19: Writer uploads work

Objective: To ensure that the writer can upload files

Table 178: Functional Testing 19

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check writer is able to view upload files	NIL	Page is displayed	Page is displayed	Default
2.	Check if the writer can upload files.	NIL	Writer selects file and uploads.	Selected file is uploaded.	Default
3.	Check if the writer can save the displayed report	Name: report id	The system shall prompt the writer to enter complete description of file.	The system prompts the writer to enter complete description of file.	Default

Functional Testing 20: Manager uploads reports
Objective: To ensure that the manager can upload files

Table 179: Functional Testing 20

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check manager is able to view upload files	NIL	Page is displayed	Page is displayed	Pass
2.	Check if the manager can upload files.	NIL	Manager selects file and uploads.	Selected file is uploaded.	Pass
3.	Check if the manager can save the displayed report	Report of: Team member name	The system shall prompt the manager to enter complete description of file.	The system prompts the manager to enter complete description of file.	Pass

Functional Testing 21: Teamlead view teams reports
Objective: To ensure that the team lead can view reports of the selected employee.

Table 180: Functional Testing 21

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check if the Teamlead can view list of employees once clicks on the “Reports” button.	Nil	Displays the list of students.	Displays the list of employe es.	Pass
2	Check that the Teamlead can view report of the selected employee.	Nil	Display report of the selected employee.	Display report of the selected employ ee.	Pass
3	Check that the Teamlead can view employee by search.	Search bar: Talha	Display the employee information onclick.	Displays the employee information onclick.	Pass

Functional Testing 22: Admin View employees

Objective: To ensure that the admin can view the information of all the employees.

Table 181: Functional Testing 22

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify that the system redirects admin to list of employees once "View employees" button is clicked.	Nil	Display the list of all the employees	Displays the list of all the employees	Pass
2	Check that the system displays the information of the selected employee.	Nil	Display information of the selected employee.	Display information of the selected employee.	Pass
3	Check that the system displays the employee searched by the admin.	Name= employee12	Display the searched employees information.	Display the searched employees information.	Pass
4	Check the system's response when the searched employees is not found.	Name=xyz	Display message employee not found.	Displays message employees not found.	Pass

Functional Testing 23: Admin Add Employee
Objective: To ensure that the admin can give access to Employees requests

Table 182: Functional Testing 23

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify that the system displays the list of Employee	Nil	Display the list of all the Employee requests	Display the list of all the Employee requests	Pass
2	Check that the system grants access to the selected Employee	Nil	Grant access to the selected Employee and remove the employee from the requests list	Grant access to the selected Employee and remove the employee from the requests list.	Pass
3	Check that the system rejects access to the selected Employee	Nil	Reject access to the selected Employee and remove the Employee from the requests list	Reject access to the selected Employee and remove the Employee from the requests list	Pass

Functional Testing 24: Check Calendar functionality
Objective: To ensure that the calendar is working properly.

Table 183: Functional Testing 24

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check if calendar is displaying properly.	User clicks on the “Calendar” button from sidebar.	The calendar is displayed.	The calendar is displayed.	Pass
2	Check if day wise schedule is displayed.	User clicks on the “Day” button from sidebar	The day wise calendar is displayed.	The day wise calendar is displayed.	Pass
3	Check if month wise schedule is displayed.	User clicks on the “Month” button from sidebar	The month wise calendar is displayed.	The month wise calendar is displayed.	Pass
4	Check if agenda can be added.	User clicks on the “agenda” button.	The add agenda window pops up.	The add agenda window pops up.	Pass

Functional Testing 25: File uploading is working with progress bar
Objective: To ensure that the file uploading is working and progress bar updates accordingly.

Table 184: Functional Testing 25

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that the upload file button is working.	User clicks on the upload file button.	The upload file button gets clicked	The upload file button gets clicked	Pass
2	Check that the select file from device window pops up.	User browses and selects the file from the device.	The selected file is pinned.	The selected file is pinned.	Pass
3	Check that the file selected starts uploading.	The pinned file should start to upload.	The pinned file starts to upload.	The pinned file starts to upload.	Pass
4	Checks that the progress bar starts to update as file gets uploaded.	The file starts uploading.	The file is uploaded.	The file is uploaded.	Pass
5	Checks that the file uploaded tag pops up.	User waits for the pop up to show.	The file uploaded tag pops up.	The file uploaded tag pops up.	Pass

Functional Testing 26: File downloading is working with progress bar.
Objective: To ensure that the file downloading is working and progress bar updates accordingly.

Table 185: Functional Testing 26

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that the Download file button is working.	User clicks on the Download file button.	The Download file button gets clicked.	The Download file button gets clicked.	Pass
2	Check that the select file from updated files.	User browses and selects the file from the uploaded files.	The selected file is pinned.	The selected file is pinned.	Pass
3	Check that the file selected starts downloading.	The pinned file should start to download.	The pinned file starts to download.	The pinned file starts to download.	Pass
4	Checks that the progress bar starts to update as file gets downloaded.	The file starts downloading.	The file is downloaded.	The file is downloaded.	Pass
5	Checks that the file download tag pops up.	User waits for the pop up to show.	The file downloaded tag pops up.	The file downloaded tag pops up.	Pass

Functional Testing 27: Checking Task Evaluations

Objective: To ensure that the task evaluations sent are sent correctly.

Table 186: Functional Testing 27

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Task evaluations are sent correctly to database.	Files of evaluations are sent to database.	The files are uploaded in database.	The files are uploaded in database.	Pass
2	Results of evaluations are shown in graphs.	Graphs are generated on the basis of evaluation results.	The graphs are generated.	The graphs are generated.	Pass

Functional Testing 28: View reviews of employees

Objective: To ensure that the employees reviews can be seen.

Table 187: Functional Testing 28

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that the employees reviews can be seen	Click on the view employee evaluations.	The employees ratings are shown in form of stars.	The employees ratings are shown in form of stars.	Pass

Functional Testing 29: Gantt charts formation

Objective: To ensure that the gant charts are formed correctly.

Table 188: Functional Testing 29

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check if gant charts formed are correctly based on task.	Tasks are added in the projects.	Gant chart updates when the task is added in the project.	Gant chart updates when the task is added in the project.	Pass

6.3 Business Rules Testing

Business Testing 1: Choose email for account registration
Testing Objective: To ensure that email chosen is valid

Conditions:

1. Email should follow the given pattern
2. No two users can have the same email

Table 189: Business Rules Testing 01

Conditional	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Email to follow pattern	**True**	**True**	**False**	**False**
No two users can have same username	**True**	**False**	**True**	**False**
Actions/Outcome				
Email is selected	**True**	**False**	**False**	**False**

Business Testing 2: Choose password for account registration
Testing Objective: To ensure that password chosen is valid

Conditions:

1. Password should follow be alphanumeric
2. Password should be minimum 8 characters long

Table 190: Business Rules Testing 02

Conditional	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Password alphanumeric	**True**	**True**	**False**	**False**
Minimum length is 8	**True**	**False**	**True**	**False**
Actions/Outcome				
Password is valid	**True**	**False**	**False**	**False**

Business Testing 3: Login into the account
Testing Objective: To ensure that password chosen is valid

Conditions:

1. Password should be valid
2. Email to be valid
3. Credentials should belong to an existing user

Table 191: Business Rules Testing 03

Condition	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**	**Rule 6**	**Rule 7**	**Rule 8**
Password is valid	**True**	**True**	**True**	**True**	**False**	**False**	**False**	**False**
Email is valid	**True**	**True**	**False**	**False**	**True**	**True**	**False**	**False**
Credential belong to an existing user	**True**	**False**	**False**	**True**	**True**	**False**	**True**	**False**
Action/Outcome								
Login Successful	**True**	**False**	**False**	**False**	**False**	**False**	**False**	**False**

Business Testing 4: The system will not accept a file greater than 15 MB.
Testing Objective: To ensure that file chosen is valid

Conditions:

1. The file should be less than 15 MB.

Table 192: Business Rules Testing 04

Conditional	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Max size is 15mb	**True**	**True**	**False**	**False**
Minimum size is 1mb	**True**	**False**	**True**	**False**
Actions/Outcome				
File is valid	**True**	**False**	**False**	**False**

6.4 Integration Testing

Integration Testing 1: Login as employee with Valid credential and redirected to dashboard
Testing Objective: To ensure the login form is working correctly with valid and invalid credentials/inputs and dashboard is displayed on successful login

Table 193: Integration Testing 01

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if employee can login into the system successfully with valid credentials and redirects to Dashboard screen,	Email: abc@gmail.com Password: areeba123456	Validates email address and password, user logged in and redirected to Dashboard screen	User Logged in and redirected to Dashboard screen.	Pass
2.	Check if employee cannot login into the system with invalid credentials and stays on login screen	Email: abc@gmail.com Password: 123456	Validates email address and password, user login failed and "Login failed" message display and user stays on login screen	User cannot Login and error message display and not redirected to dashboard screen.	Pass

Integration Testing 2: Login as manager with Valid credential and redirected to dashboard
Testing Objective: To ensure the login form is working correctly with valid and invalid credentials/inputs and dashboard is displayed on successful login

Table 194: Integration Testing 02

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if manager can login into the system successfully with valid credentials and redirects to Dashboard Screen.	Email: abc@gmail.com Password: areeba123456	Validates email address and password, user logged in and redirected to Dashboard screen.	User Logged in and redirected to Dashboard screen.	Pass
2.	Check if manager cannot login into the System with invalid credentials and stays on login screen.	Email: abc@gmail.com Password: 123456	Validates email address and password, user login failed and “Login failed” message display and user stays on login screen	User cannot Login and error message display and not redirected to dashboard screen.	Pass

Integration Testing 3: Login as Trainee with Valid credential and redirected to dashboard
Testing Objective: To ensure the login form is working correctly with valid and invalid credentials/inputs and dashboard is displayed on successful login

Table 195: Integration Testing 03

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if Trainee can login into the system successfully with valid credentials and redirects to Dashboard screen,	Email: abc@gmail.com Password: areeba123456	Validates email address and password, user logged in and redirected to Dashboard screen	User Logged in and redirected to Dashboard screen.	Pass
2.	Check if Trainee cannot login into the system with invalid credentials and stays on login screen	Email: abc@gmail.com Password: 123456	Validates email address and password, user login failed and "Login failed" message display and user stays on login screen	User cannot Login and error message display and not redirected to dashboard screen.	Pass

Integration Testing 4: Login as Writer with Valid credential and redirected to dashboard
Testing Objective: To ensure the login form is working correctly with valid and invalid credentials/inputs and dashboard is displayed on successful login

Table 196: Integration Testing 04

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check if Writer can login into the system successfully with valid credentials and redirects to Dashboard screen,	Email: abc@gmail.com Password: areeba123456	Validates email address and password, user logged in and redirected to Dashboard screen	User Logged in and redirected to Dashboard screen.	Default
2.	Check if Writer cannot login into the system with invalid credentials and stays on login screen	Email: abc@gmail.com Password: 123456	Validates email address an d password, user login failed and “Login failed” message display and user stays on login screen	User cannot Login and error message display and not redirected to dashboard screen.	Default

Integration Testing 5: User logout from and redirected to login page
Testing Objective: To ensure that the user can easily logout from current login session after finishing their work and redirected to login page

Table 197: Integration Testing 05

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	To ensure that user can easily logout and redirected to login screen	User logout from the session	User successfully logout from current session and redirected to login page	Logout successful and user is redirected to login page	Pass
2.	To ensure that user cannot logout and do not redirect to login screen if they cancel the logout operation	User didn't logout from the current session.	User could not logout from the current session and stays on home screen.	User cancel the logout operation and stays on Home screen.	Pass

Integration Testing 06: Employee can edit their Profile information.
Testing Objective: To ensure that the employee can edit their profile information.

Table 198: Integration Testing 06

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that employee be able to update profile details	Username: Areeba Naveed Email: aya19@gmail.com Age : 31 Qualification: CS	Successfully updates the profile	Successfully updates the profile	Pass
2.	Check that updated details showing or not when user refresh the page	Nil	Successfully displays the updated profile	Successfully displays the updated profile	Pass
3.	If the registered employee enters invalid information while editing their profile	Username: noor Email: mahnoor.gmail.com	Display warnings and prompt the user to enter correct credentials.	Display warnings and prompt the employee to enter correct credentials.	Pass

Integration Testing 07: Trainee edit profile

Testing Objective: To ensure that the Trainee can edit their profile information

Table 199: Integration Testing 07

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that Trainee be able to update profile details	Username: Areeba Naveed Email: a@gmail.com Qualification: CS	Successfully updates the profile	Successfully updates the profile	Pass
2.	Check that updated details showing or not when user refresh the page	Nil	Successfully displays the updated profile	Successfully displays the updated profile	Pass

Integration Testing 08: Manager edit profile

Testing Objective: To ensure that the manager can edit their profile information

Table 200: Integration Testing 08

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that manager be able to update profile details	Username: Areeba Naveed Email: aya19@gmail.com Age : 31 Qualification: CS	Successfully updates the profile	Successfully updates the profile	Pass
2.	Check that updated details showing or not when user refresh the page	Nil	Successfully displays the updated profile	Successfully displays the updated profile	Pass
3.	If the registered employee enters invalid information while editing their profile	Username: noor Email: mahnoor.gmail.com	Display warnings and prompt the user to enter correct credentials.	Display warnings and prompt the manager to enter correct credentials.	Pass

Integration Testing 09: Admin deletes profile
Testing Objective: To ensure that the admin can delete their profile with correct credentials

Table 201: Integration Testing 09

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check if the account is deleted by entering incorrect password after clicking on the Delete account button	Password: (incorrect)12345	Display 'Password is incorrect. Try Again'	Display 'Password is incorrect. Try Again'	Pass
2.	Check if the account is deleted by entering correct password after clicking on the Delete account button	Password: 1234567	Account successfully deleted and redirects to the student login page	Account successfully deleted and redirects to the student login page	Pass

Integration Testing 10: Trainee uploads work
Testing Objective: To ensure that the trainee can upload files

Table 202: Integration Testing 10

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check Trainee is able to view upload files	NIL	Page is displayed	Page is displayed	Default
2.	Check if the Trainee can upload files.	NIL	Trainee selects file and uploads.	Selected file is uploaded.	Default
3.	Check if the Trainee can save the displayed report	Name: report id	The system shall prompt the Trainee to enter complete description of file.	The system prompts the trainee to enter complete description of file.	Default

Integration Testing 11: Writer uploads work
Testing Objective: To ensure that the writer can upload files

Table 203: Integration Testing 11

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check writer is able to view upload files	NIL	Page is displayed	Page is displayed	Default
2.	Check if the writer can upload files.	NIL	Writer selects file and uploads.	Selected file is uploaded.	Default
3.	Check if the writer can save the displayed report	Name: report id	The system shall prompt the writer to enter complete description of file.	The system prompts the writer to enter complete description of file.	Default

Integration Testing 12: Manager uploads reports
Testing Objective: To ensure that the manager can upload files

Table 204: Integration Testing 12

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Check manager is able to view upload files	NIL	Page is displayed	Page is displayed	Pass
2.	Check if the manager can upload files.	NIL	Manager selects file and uploads.	Selected file is uploaded.	Pass
3.	Check if the manager can save the displayed report	Report of: Team member name	The system shall prompt the manager to enter complete description of file.	The system prompts the manager to enter complete description of file.	Pass

Integration Testing 13: Teamlead view teams reports
Testing Objective: To ensure that the team lead can view reports of the selected employee.

Table 205: Integration Testing 13

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check if the Teamlead can view list of employees once clicks on the "Reports" button.	Nil	Displays the list of students.	Displays the list of employees.	Pass
2	Check that the Teamlead can view report of the selected employee.	Nil	Display report of the selected employee.	Display report of the selected employ ee.	Pass
3	Check that the Teamlead can view employee by search.	Searc h: Talha	Display the employee information onclick.	Displays the employee information onclick.	Pass

Integration Testing 14: Admin View employees
Testing Objective: To ensure that the admin can view the information of all the employees.

Table 206: Integration Testing 14

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify that the system redirects admin to list of employees once “View employees” button is clicked.	Nil	Display the list of all the employees	Displays the list of all the employees	Pass
2	Check that the system displays the information of the selected employee.	Nil	Display information of the selected employee.	Display information of the selected employee.	Pass
3	Check that the system displays the employee searched by the admin.	Name= employee12	Display the searched employees information.	Display the searched employees information.	Pass
4	Check the system’s response when the searched employees is not found.	Name=xyz	Display message employee not found.	Displays message employees not found.	Pass

Integration Testing 15: Admin Add Employee

Testing Objective: To ensure that the admin can give access to Employees requests

Table 207: Integration Testing 15

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify that the system displays the list of Employee	Nil	Display the list of all the Employee requests	Display the list of all the Employee requests	Pass
2	Check that the system grants access to the selected Employee	Nil	Grant access to the selected Employee and remove the employee from the requests list	Grant access to the selected Employee and remove the employee from the requests list.	Pass
3	Check that the system rejects access to the selected Employee	Nil	Reject access to the selected Employee and remove the Employee from the requests list	Reject access to the selected Employee and remove the Employee from the requests list	Pass

Integration Testing 16: Check Calendar functionality

Testing Objective: To ensure that the calendar is working properly.

Table 208: Integration Testing 16

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check if calendar is displaying properly.	User clicks on the "Calendar" button from sidebar.	The calendar is displayed.	The calendar is displayed.	Pass
2	Check if day wise schedule is displayed.	User clicks on the "Day" button from sidebar	The day wise calendar is displayed.	The day wise calendar is displayed.	Pass
3	Check if month wise schedule is displayed.	User clicks on the "Month" button from sidebar	The month wise calendar is displayed.	The month wise calendar is displayed.	Pass
4	Check if agenda can be added.	User clicks on the "agenda" button.	The add agenda window pops up.	The add agenda window pops up.	Pass

Integration Testing 17: File uploading is working with progress bar
Testing Objective: To ensure that the file uploading is working and progress bar updates accordingly.

Table 209: Integration Testing 17

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that the upload file button is working.	User clicks on the upload file button.	The upload file button gets clicked	The upload file button gets clicked	Pass
2	Check that the select file from device window pops up.	User browses and selects the file from the device.	The selected file is pinned.	The selected file is pinned.	Pass
3	Check that the file selected starts uploading.	The pinned file should start to upload.	The pinned file starts to upload.	The pinned file starts to upload.	Pass
4	Checks that the progress bar starts to update as file gets uploaded.	The file starts uploading.	The file is uploaded.	The file is uploaded.	Pass
5	Checks that the file uploaded tag pops up.	User waits for the pop up to show.	The file uploaded tag pops up.	The file uploaded tag pops up.	Pass

Integration Testing 18: File downloading is working with progress bar.
Testing Objective: To ensure that the file downloading is working and progress bar updates accordingly.

Table 210: Integration Testing 18

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that the Download file button is working.	User clicks on the Download file button.	The Download file button gets clicked.	The Download file button gets clicked.	Pass
2	Check that the select file from updated files.	User browses and selects the file from the uploaded files.	The selected file is pinned.	The selected file is pinned.	Pass
3	Check that the file selected starts downloading.	The pinned file should start to download.	The pinned file starts to download.	The pinned file starts to download.	Pass
4	Checks that the progress bar starts to update as file gets downloaded.	The file starts downloading.	The file is downloaded.	The file is downloaded.	Pass
5	Checks that the file download tag pops up.	User waits for the pop up to show.	The file downloaded tag pops up.	The file downloaded tag pops up.	Pass

Integration Testing 19: Checking Task Evaluations

Testing Objective: To ensure that the task evaluations sent are sent correctly.

Table 211: Integration Testing 19

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Task evaluations are sent correctly to database.	Files of evaluations are sent to database.	The files are uploaded in database.	The files are uploaded in database.	Pass
2	Results of evaluations are shown in graphs.	Graphs are generated on the basis of evaluation results.	The graphs are generated.	The graphs are generated.	Pass

Integration Testing 20: View reviews of employees

Testing Objective: To ensure that the employees reviews can be seen.

Table 212: Integration Testing 20

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check that the employees reviews can be seen	Click on the view employee evaluations.	The employees ratings are shown in form of stars.	The employees ratings are shown in form of stars.	Pass
2	Check if employee reviews can be updated.	Employee evaluation is updated by admin.	The employee ranking should be updated.	The employee ranking is updated.	Pass

Integration Testing 21: Gantt charts formation

Testing Objective: To ensure that the gant charts are formed correctly.

Table 213: Functional Testing 29

No.	Test case/Test script	Attribute and value	Expected result	Actual Result	Result
1	Check if gant charts formed are correctly based on task.	Tasks are added in the projects.	Gant chart updates when the task is added in the project.	Gant chart updates when the task is added in the project.	Pass
2	Check if critical path is shown correctly	Tasks are added in the project.	Gant chart should show a critical path.		Default
3	Check if tasks can be edited in Gant chart.	Task is selected	Edit option should be available.		Default

7 Conclusion and Future Work

This chapter concludes the project and highlights future work.

7.1 Conclusion

In the end, we would just like to say that we hope that this project will be beneficial for public and organization. We tried to solve a genuine problem of security in today's world and introduce a cost efficient and digitalize system of surveillance.

Secondly, this project helped in our growth as well as we learned some new technologies and concepts like CNN models, MERN stack for development.

7.2 Future Work

In future, first we intend to improve the UI of our mobile application and web portal and make it more user friendly. Moreover, we have invested so much time and are now emotionally attached with it so we would like it to succeed.

We believe that our project is highly scalable and can help organizations from different culture for the better security of their offices and business. This project can used for enhanced monitoring of employees and their tasks.

We will be adding features like gant charts, project management tools and improved monitoring techniques for tasks and employees. This is a product, which can change the security perspective of many organizations, and we will ask our clients permission to sell it to further organizations too.

8 References

[1] "Jira," 2021. [Online]. Available: www.atlassian.com/software/jira.

[2] "Teams," 2022. [Online]. Available: https://www.microsoft.com/en-ww/microsoft- teams.

[3] Schmidt, "In Learner and User Experience Research.," in *In Learner and User Experience Research.* , Retrieved from Methods of user- centered design and evaluation for learning designers., 2021.

[4] J. Nielsen, "10 Usability Heuristics for User Interface Design," 15 November 2020. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/.

[5] " In Learner and User Experience Research. Retrieved from Methods of user- centered design and evaluation for learning designers.," 2020.