6CS007 – Project and Professionalism

Milestone 3 – Final Year Project Report (Draft)

University Id:	2330520
Group:	L6CG22
Supervisor:	Dipti Gyawali
Submitted by:	Shrota Ghimire
Submitted on:	31st March, 2025

Table of Contents

Table of Figures

Abstract

Task Master helps manage projects by letting team members cooperate more efficiently and communicate better. Managers can generate, manage and remove projects using unique project codes to ensure user access is properly regulated. When a user enters a code, they are assured that the task environment is structured and safe. With Task Master, it is simple to assign tasks, monitor progress and update everyone in real time. It enables role-based access, so that only permitted users can perform some tasks and also includes a system to let users and admins know about any important updates. The system is written in Node.js, HTML and CSS to make it convenient, simple and responsive for use in schools and organizations. By implementing this project, the goal is to raise productivity, facilitate clearer communication and monitor and manage all project activities within a central hub.

1. Introduction

1.2 Project briefing

Organization of time is crucial in increasing efficiency and reducing the effects of moving deadlines that are ever present in any establishment. The concept of a task management system means that the process of tracking tasks is made much easier and less time consuming when it comes to checking the time, priority and overall status of tasks in question for a particular person or a team. For people that work in schools or colleges, where numerous tasks, assignments, and schedules are typical, it creates a possibility to facilitate the work of both administrators and students.

For this reason, I decided to write a simple and effective task management program that could be designed for educational institutions only. In this way, this system of interaction between administrators and students prevents any of the important tasks to be left unnoticed and provides a clear division of the tasks and responsibilities, as well as meeting all the deadlines. It also assists in keeping the work of different parties at par by making communication together with the management of responsibilities protocol easier since it is all in one system.

Especially in schools and college where daily turn has a lot of assignments, tests meetings, and several responsibilities it becomes very necessary to have a systematic way of handling time and activity. Originally, students will be able to organize their work loads, exams, and project due dates in the system; on the other hand, administrators will be able to monitor the schedules of the faculties, the events, and the academic lessons more efficiently. This scheduling makes it easier to avoid scenarios where some individuals fail to complete tasks on time as well as making it easier for every participant to execute their rightful roles thereby making the education system more efficient and much more organized.

1.3 Academic Questions:

- • In what ways does the task management system increase the productivity and efficiency of its users?
- - With the Task Master system, people can complete tasks and projects more efficiently because everything is organized. It enables users to see their assignments, when they are due and their

progress, so they can concentrate on important work. With real-time updates, everyone on the team can monitor any changes right away and mistake-free. Moreover, administrators can easily assign tasks and keep an eye on them and users can cooperate without relying on numerous platforms. Arranging users in groups according to projects, assigning roles and using automated notifications help improve the overall smoothness of the workflow.

- What makes a difference in this system as compared to other systems in the field?

- Uniquely, Task Master requires administrators to create individual project codes that users must have to participate in their projects. Access control is particularly valuable in schools and workplaces where certain projects should only be viewed by specified people. Some task managers use open email opt-ins, but with Task Master, users can only join projects when assigned by the manager. Besides, the system highlights simplicity by featuring necessary features only, making it perfect for use in schools, class projects or group work between a few people.

- What problems do it solve?

- Task Master addresses common issues that arise when people collaborate in teams. Examples of these are unsystematic task organization, missing faster updates and not being able to handle access to projects. Generally, it is not easy to organize communication and tasks among students or groups in schools or projects, so this system steps in by offering a single place for all tasks. Also, access to the project is restricted by using a coding system and admins can control who is involved or leaves the project. Moreover, it brings down the need to organize tasks with the use of external services.

- How does it handle user challenges?

- Task Master assists users by giving them a default interface that is user-friendly and easy to navigate. Tasks are clearly outlined, progress is easy to track, and everything is arranged in an easy-to-understand layout. Within the administration, you can conveniently manage users, assign tasks and keep track of upcoming modifications. Regular users can easily add themselves to tasks by using a project code and they always get direct access to the tasks chosen for them. A notification system and role-based access limits errors, reduces confusion and prevents people from handling the same tasks.

- What notifications or updates do it provide?

- Task Master sends real-time alerts to let users know whenever they are added to a project, assigned work or if the status of their work is updated. As soon as a person joins a project or changes the state of a task, admins are notified instantly. As a result, all stakeholders know what is happening and can take appropriate actions. Users can access the notifications within the application, and they can also store them for later which helps them catch any important updates.

1.4 Aims:

Task management software is designed to offer a framework to enable people to create, schedule and prioritize tasks effectively. This helps in breaking down work, making work specific and defining priorities to ensure that any work done is accompanied by a laid down procedure. As the central location of task information, it reduces confusion and aids in directing the attention to goals and objectives of one or more persons/teams.

1.4 Objectives:

- To promote active, engaging discussion between the teachers and students.
- To help in collaboration among team members.
- To organize, and prioritize, tasks effectively.
- To reduce the risk of missed deadlines.
- To enable better decision-making through task insights.
- To offer real-time updates and timings.
- To boost productivity and efficiency.

1.5 Artefact

1.5.1 Functional Decomposition Diagram

It is an overall picture of a system where the system function is divided into subsystems. (A. H. M. Al-Ahmad, 2020)

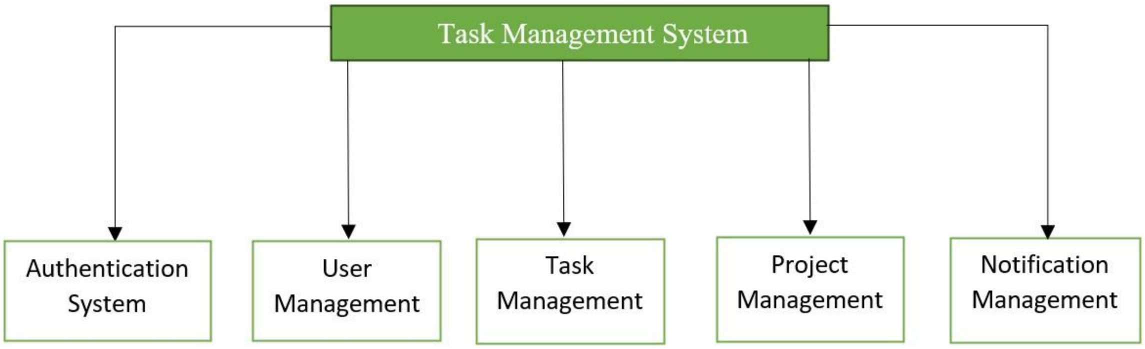


Figure 1: Functional Decomposition Diagram (FDD)

1.5.2 Agile

Agile is an ability to respond to a change and work on building a project towards the end of greatest. This means that Agile software development includes number of frameworks. There are facility management methods like Scrum, Feature-driven Development (FDD), and Extreme Programming. (Beck, K., & Andres, 2004)

1.5.3 Gantt Chart

This is chart employed to describe an activity with the time of the activity in a development of a project. (Kerzner H, 2013)

1.5.4 Product Backlog

Product backlog is a list of tasks for the developers, based on the needs and requirements of the system. (Alliance, 2000)

1.5 Scope and Limitation of the Project

1.5.1 Scope

- The system enables two roles: Admin and User, with differing access levels and functionality.
- Admin can create, assign and manage users within specified projects, via unique codes.
- Users can view, update, and change the status of the tasks assigned to them.
- Task status updates and adding new users to a project can happen in real-time using WebSocket or something similar.

1.5.2 Limitations

- The systems depend heavily on digital literacy, which can be a barrier for users who have little experience using web applications or digital tools.
- Users who do not have adequate typing skills or navigation skills will likely have problems updating task statuses and using the dashboard appropriately.
- Annoyances from lack of face-to-face communication can lead to miscommunication/misunderstandings between the admin and users' understanding of task details or deadlines.
- When a task list is too long and there are too many notifications at once for a user, you risk overwhelming the user and causing them difficulty prioritizing their work.
- Furthermore, some may not always be an active participant regarding updating his or her tasks status, which can impact the reliability and accuracy of the system for the administrators.

1.6 Report Structure

1.6.1 Introduction

This section includes the short project description, the primary objective, and the problem of motivation behind this project. It will also discuss the early ideas of the system on how we will manage and develop the system.

1.6.2 Literature Review

This section will present the original research carried out prior to conceptualizing this system. The research included research papers from various websites discovered in a google search such as academia, IEEE explore, and any similar platform.

1.6.3 Project Methodology

This section emphasizes the project's planning and methods that will be used during all development processes. Additionally, we plan tasks and create a schedule, that will be a chart i.e. a Gantt chart.

1.6.4 Different Technology, Tools and Techniques

This section includes the tools, technologies and techniques that are used while developing the system.

1.6.5 Artefact Designs

This section contains the product backlog of the project (list of tasks of the project), overall diagrams and figures of the whole project and a table representation which is the software requirement specifications (SRS) of the system.

1.6.6 Critical Evaluation of the Report

This section contains an overall evaluation of the report showing what I have learned.

1.6.7 Evidence of Project Management

Here, it includes the log sheet we have written during the meeting we conducted with our supervisor, receiving their feedback and signatures on our work. Also, we will include a complete Gantt chart that outlines our input given daily in a timely manner towards our final year project.

1.6.8 Conclusion

This section outlines the about the learnings, outcomes, future work or possibilities and the experiences throughout the project including both good and bad parts.

1.6.9 References

This section contains all the citations that are given in every section.

2. Literature Review

Task management systems are now widely used in everyday project management because they solve a simple but fundamental problem of how to organize work, assign tasks, and track progress. Most organizations apply them to increase efficiency as well as facilitate and organize work to enhance timely delivery of projects. New technologies such as AI and analytics have enhanced the progression of TMS from mere tasking tools to sensitive project management solutions that in addition to scheduling include automated depredation and self- assessment.

This paper aims to include the literature between 2018 to 2023, to explore the advancement, adoption and significance of TMS tools. For the same reason, it outlines the present state of research in the context of TMS, as well as the future directions that should be considered for further study.

Theme 1: System Development and Design

Paper 1: Task Management System (2022)

May states that this activity management is crucial to improving individual and organizational performance as individuals and organizations face enormous demands in the present world. Developed from traditional paper-based tools of to-do lists and planners, task management systems (TMS) have become much more sophisticated with the aid of digital technology. David Allen's book "Getting things done" signed a turn point of this evolution by presenting a methodology based on collection, categorizations, and clarity of action. Contemporary TMS are now digital workhorses featuring essential functionalities such as creating and modifying a task, advancing and distributing them, attaching and sharing files, adding comments for teamwork, and reporting with analyzing tools for monitoring advancement and searching for failures. The more sophisticated applications may also include time logs, calendars, and scheduling, Kanban boards, Gantt charts for timeline management. They can also be synchronized with other programs such as project management software, CRM and email, to enhance functioning in other divisions. Such features enable and mobilize people and teams, synchronize their activities and optimize their performance. For this reason, although the features of a TMS can range from the needs of an individual and/or an organization, the general idea of its application lies in the optimization of the task management process a support of collaboration. It emphasizes the concept of TMS in the modern world, as a necessity in the growing-pressure and the increasingly globalized world, thus plays a crucial role epigenetically in managing the tasks and hence, unearthing our personal and career goals. (May, 2022)

Paper 2: Employee Task Tracking System (2023)

Sandhya states that the Employee Task Tracking System is a software application that is intended for efficiently assigning and monitoring tasks within an organization and overcomes the drawbacks of using mere papers and spreadsheets for this purpose. They stressed that the modern working environment is characterized by high tempo and that the applications of old models and approaches result in severe problems, such as delays and obscurity. This system, built by employing python and frameworks such as Django, offers a wholly automated method for the management of employee tasks whilst assuring the staff, contractors and employers a real-time view of the ongoing tasks. Python's natural language, suitably supported libraries like TensorFlow for data automation, Pandas for data representation in an efficient manner, NumPy for mathematical computations and Matplotlib for giving visualization are some of the reasons that Python can be apt for the core functionality of the software application. Django is a high-level web application framework for Python designed to enhance web-based applications and Scikit-learn for the specification of machine learning models and other statistical methods for tracking tasks. The system helps the employee and also the managers in that, as per the policy of the company they are able to decide whom to assign a particular task to as well as gives the employee the ability to create as well as assign specific tasks for particular subordinates and also helps the managers track workloads as well as other measurable like identifying crowded areas and most importantly in directing resources most appropriately. In keeping track of tasks, the system helps improve productivity, increase efficiency and improve collaboration, all of which are done using a reliable tool for tracking the tasks across the organization. This development corresponds to the increasing demand for comprehensive and scalable global solutions in the sphere of system design, effective managing of processes, non-transparency and accountability within the companies. (Sandhya, 2023)

Theme 2: User Behavior and Tool Adoption

Paper 3: Unpacking Task Management Tools, Values, and Worker Dynamics (2023)

The degrees and modes of utilizing Task Management Systems (TMS) have been proved to be affected by personality factors and job features. Since knowledge workers carry out knowledge intensive tasks as part of their tasks, they have been faced with the challenge of how best to track and manage their workloads, resulting in the creation of task management tools at one point or the other. It has been noted that users tailor these tools based on their requirements and goals, and the desirable features they find in these tools are defined by personal disposition and work responsibilities. Studies reveal that what are commonly referred to as personal, as in the Big Five personality models influence how employees interact with tools used for task management. For instance, people with high conscientiousness self-ensure they use tools that help them to organize their work or environment, while those with high neuroticism would want features that help to calm them or reduce their stress levels. These characteristics of the job influence an individual demand on the characteristics of the tools: their communicability, adaptability, and collaboration properties. Research has reviewed how various workers employ TMS tools regarding these self-characteristics, with a focus on the tools' complexity/encompassment tradeoff. Some employees go for tools that have a variety of features that can enhance collaboration to do the work, while others opt for simple tools that can track individual assignments. Knowledge of these preferences is relevant for the construction of more appropriate tools for managing tasks that can fit the different requirements of knowledge workers, for better task organization and distribution as well as time and work progress tracking. (Library, 2023)

Theme 3: Systematic Review of Current Tools

Paper 4: The Development of the Task Management Software (2023)

The paper "The Development of the Task Management Software" by Marianna Spezie focus on how software tool, particularly Task Management Software (TMS), can be implemented to support project management in the typical business environment. Chen et al stress that efficient control of tasks is critical for achieving more with time – one of the most significant organizational assets that help to accomplish business goals. They reveal a wide range of approaches, for example, the Eisenhower Matrix for sorting the tasks due to their significance and deadline, and the system called Kanban that deals with the notion of work in progress. The paper in addition underscores that a good TMS should not only help with project and task management across teams but also help with planning through tools like the Gantt chart all the while allowing for parallel working amongst the different team members. With such systems, it is possible for an organization to enhance performance and guarantee achievement of set goals. (Spezie, 2023)

It is important to note that the concept of a Task Management System (TMS) has devolved into a necessary organizational element for enhancing performance and organizing workloads. Studies show that the user personality plays an important factor in determining the instrument choice and the extent of customizing it depends on their exact requirements, the tools and the job demands may vary from simple to complex designs. Such frameworks as the Eisenhower Matrix and Kanban have really been of great help when it comes to organizing tasks where we have very many of them within a given timespan as well as managing the work in progress or as they are called the

WIPs On the other hand, features such as the Gantt chart has made it easier to plan as well as execute a number of activities hand in parallel in different teams. Nevertheless, some research areas need more attention, for instance the combination of AI for predictive task allocation, UI better fit for a broader spectrum of users, and AI readiness for SMEs. Moreover, there is a constant tension of making a tool more complex to fulfil sophisticated needs of users while staying easy enough to be operated by any employee; thus, it remains an important goal to make future advancements that would counter these issues and continue the evolution of TMS as an indispensable tool beneficial for numerous careers.

2.1 Similar System

2.1.1 Trello

Trello, developed by Atlassian, this software provides a way for people and groups to organize their tasks using visual layouts on the web. With this method, tasks are displayed as cards that change places on lists named “To Do,” “In Progress,” and “Done.” Since Trello is easy to use and drag-and-drop task management is built-in, it works for everyone. People can create boards for different projects and each board can be filled with lists having individual cards. A card can have due dates, checklists, labels, attachments, comments and assigned team members. People can also use Power-Ups on the platform to integrate tools such as Slack, Google Drive, Dropbox and calendar apps. Trello is well suited for small and medium teams that seek a simple and easy way to organize regular tasks, projects and everyday duties. Since the tool is designed for mobile phones and offers team collaboration, everyone can view updates and notifications anytime. This tool may not have all the detailed reporting and automatic tools featured in other programs, but it is much simpler to use and easier to understand. While Trello and Task Master handle organizing and assigning tasks, only Task Master includes a unique system that uses project codes for better security and control in similar settings.

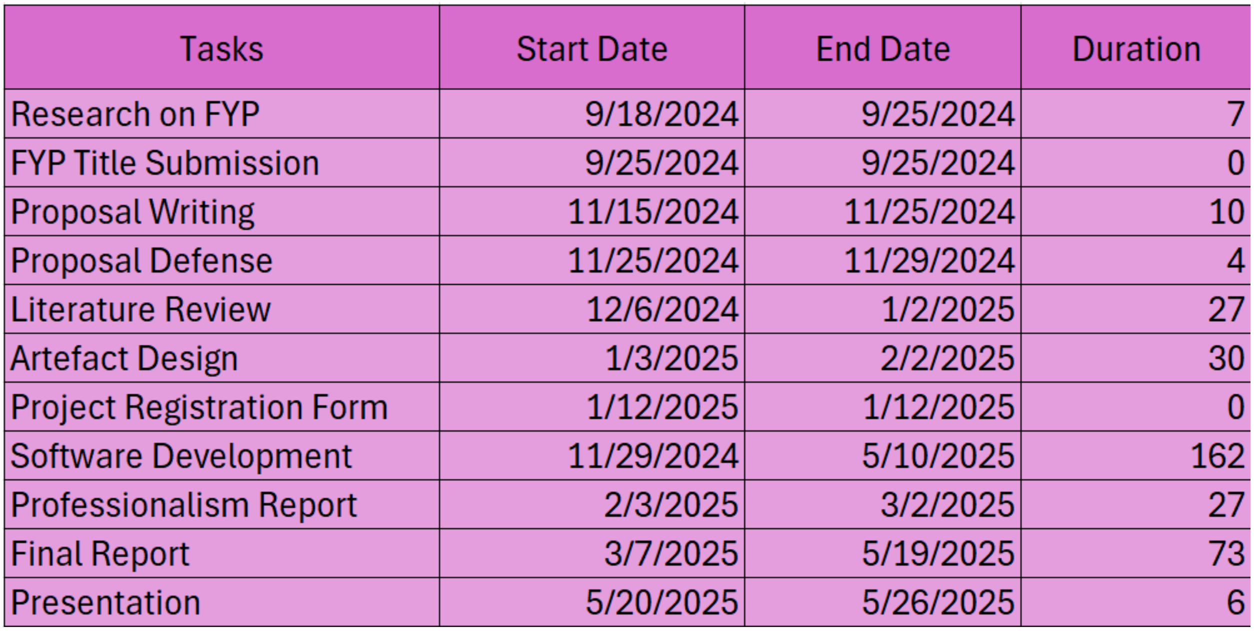
2.1.2 Jira

Jira, another product by Atlassian, helps teams in agile software development to efficiently manage projects and issues. It offers a range of tools you can use to manage planning, monitoring and reporting your projects. Jira gives teams the ability to manage sprints, epics, stories, tasks and bugs using Scrum or Kanban boards. Every task is referred to as an issue and it has spaces for descriptions, who should do the work, its priority level, labels, comments, documents and custom processes. Great flexibility is one of the best things users can create different processes, set up automations, control user rights and design dashboards with a range of graphs and statistics. Because it integrates with Bitbucket, GitHub, Jenkins and CI/CD pipelines, Jira provides software teams with insights throughout their development cycle. You can also produce burndown charts, sprint velocity and cumulative flow diagrams to easily track how the team is performing as time goes by. Jira may have more features, but it takes longer to learn than Trello and is usually chosen by teams and organizations handling projects that need advanced process control. Jira has a comparable focus on roles and managing tasks as Task Master, but it excels at scaling in size and working with third-party tools. Knowing it could be used in educational or small-project workplaces, Task Master centers on straightforwardness and includes project codes for admins and simple ways to take care of users.

3. Project Methodology

The project uses the Agile methodology because it makes things easy to understand, adaptable and authorizes splitting the task into smaller parts. Therefore, it can build the system slowly, test it at every stage and change as required. Agile allows changes and doesn’t need everything to be finalized at the beginning. If an issue occurs during development when using the Waterfall method, it becomes difficult to correct. This model works well, especially for large and risky projects, but it requires a lot more time and preparation. The agile approach is simple and proved helpful because it is efficient in solving problems and open to making changes as needed. The project had to demonstrate our development weekly to the supervisor, receive their feedback on what we did and make necessary changes. Agile allows for constant updates and feedback. Furthermore, the structure of our final year project split things into stages, aligning with Agile approach. We drew a wireframe of the system before any actual building began. As a result, developers could imagine the system from a user’s perspective and catch problems early in the development process. By doing this, all the details aligned according to the plan.

3.1 Plan / Schedule



Tasks	Start Date	End Date	Duration
Research on FYP	9/18/2024	9/25/2024	7
FYP Title Submission	9/25/2024	9/25/2024	0
Proposal Writing	11/15/2024	11/25/2024	10
Proposal Defense	11/25/2024	11/29/2024	4
Literature Review	12/6/2024	1/2/2025	27
Artefact Design	1/3/2025	2/2/2025	30
Project Registration Form	1/12/2025	1/12/2025	0
Software Development	11/29/2024	5/10/2025	162
Professionalism Report	2/3/2025	3/2/2025	27
Final Report	3/7/2025	5/19/2025	73
Presentation	5/20/2025	5/26/2025	6

Figure 2: Plan and Scheule

3.2 Gantt Chart

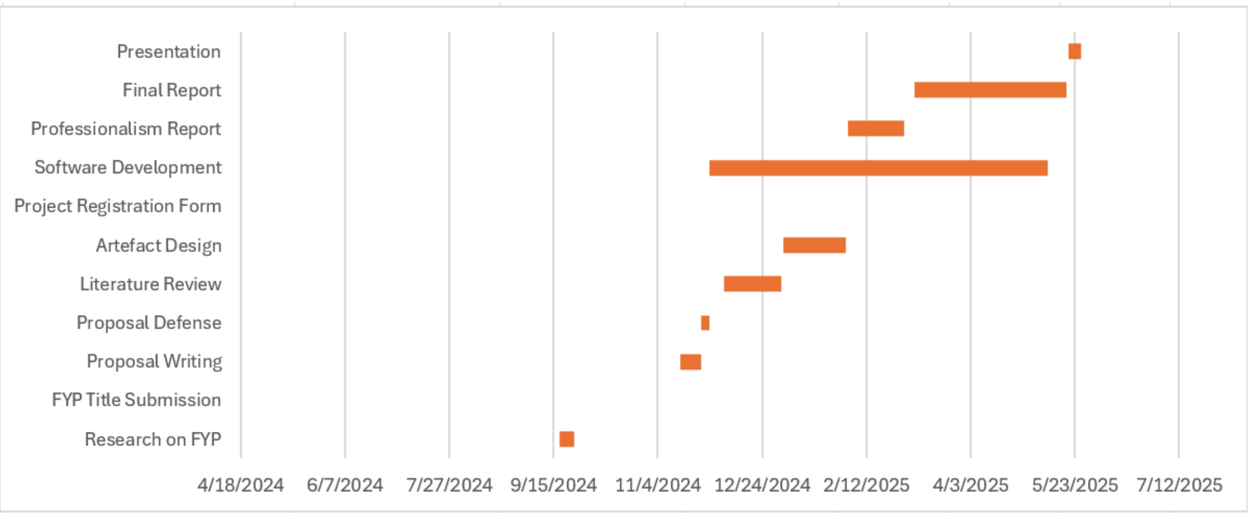


Figure 3: Gantt Chart

4. Technologies and Tools

4.1 Frontend

4.1.1 HTML

HTML (HyperText Markup Language) is the language used in developing contents on web services. It forms the fundamental structure of a webpage by having elements (tags) that mark headings, paragraphs, links, images, tables and many others.

4.1.2 CSS

CSS (Cascading Style Sheets) is the stylesheet language which serves the purpose of describing how HTML created webpage looks like. It enables the developers to dictate on the locations, hues, types, and basically the feel and look of a website.

4.1.3 JavaScript

JavaScript is a dynamism friendly, supple, and efficient language that occupies the space of an average web page's script language mainly used to construct web pages interactivities, functionalities, and behaviors. jQuery is used in conjunction with HTML (markup), and CSS (styling) to design and develop contemporary web-based applications.

4.2 Backend

4.2.1 Node JS

Node.js is an open source, cross platform JavaScript environment that lets developers run JavaScript code on the server side. Classically JavaScript is being implemented on the browsers alone, but Node.js jumps into the backend which allows full stack development in JavaScript only.

4.2.2 XAMPP

XAMPP is an open source, cross platform local server software development environment aimed at deploying different software applications. This is done by including important components such as Apache Web Server, MySQL database, PHP and Perl, making the package very viable for web developers.

4.3 Tools

4.3.1 Visual Studio Code

It is also called as VS Code and is a light weighted, open-source and an efficient source code editor developed by Microsoft. It is lightweight and made for constructing and developing contemporary web site applications and software projects. VS Code works on various programming languages and offers many extensions, settings, and adaptability to several operations.

4.3.2 GitHub

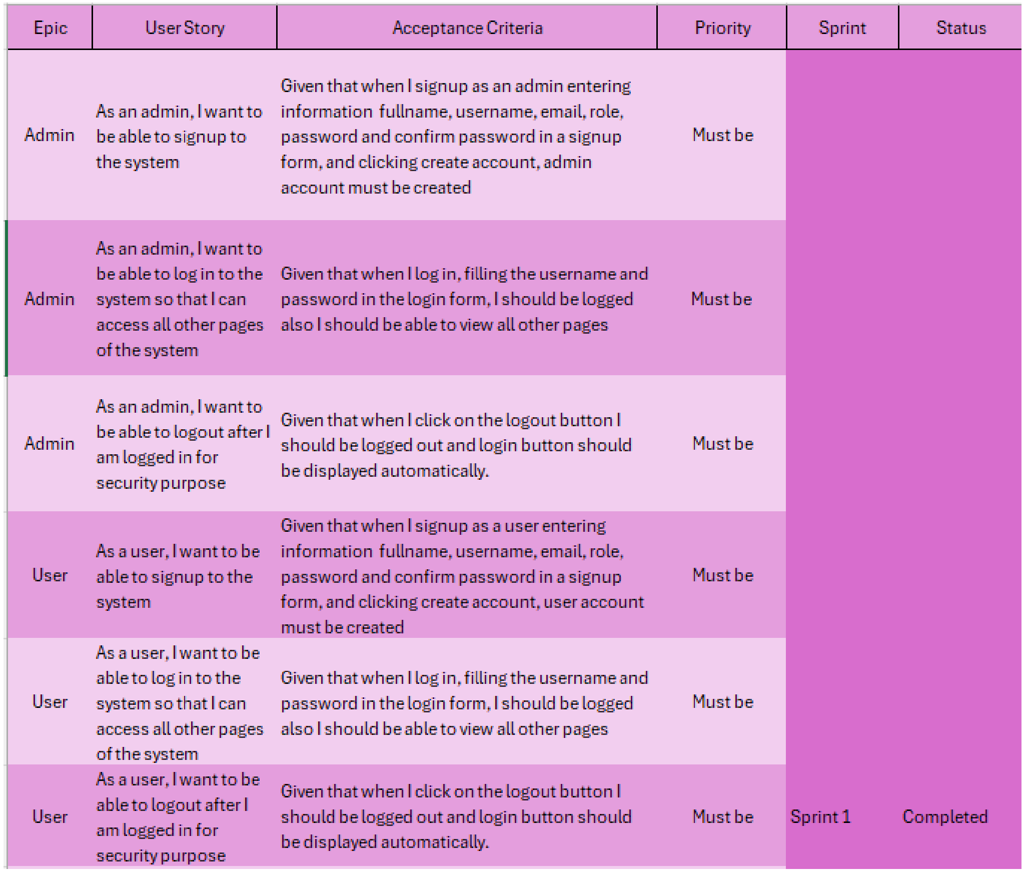
On GitHub, individuals save their codes, collaborate with others and keep track of changes. This allows different members to team up, share development work and track and control various versions of their code.

4.3.2 Figma

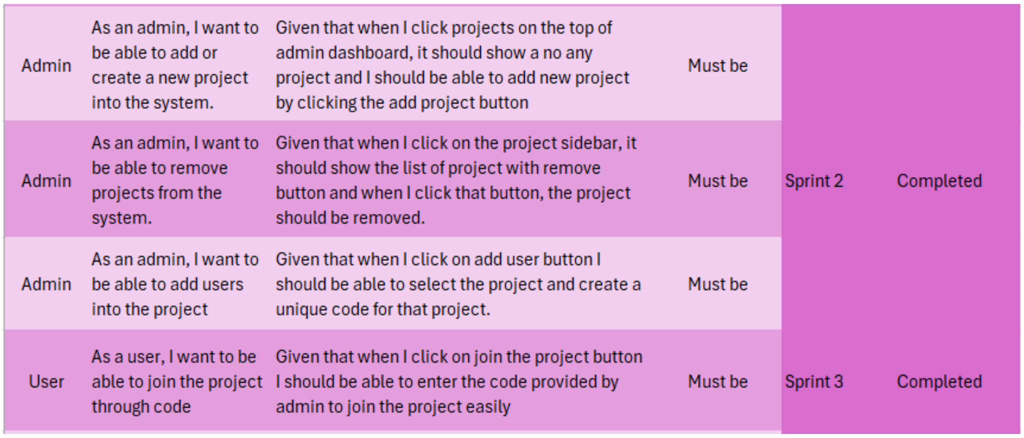
Figma is software used to design the layout of apps or websites. Designers can create and share their user interfaces, as well as collaborate live on the platform through their browser.

5. Artefact Designs

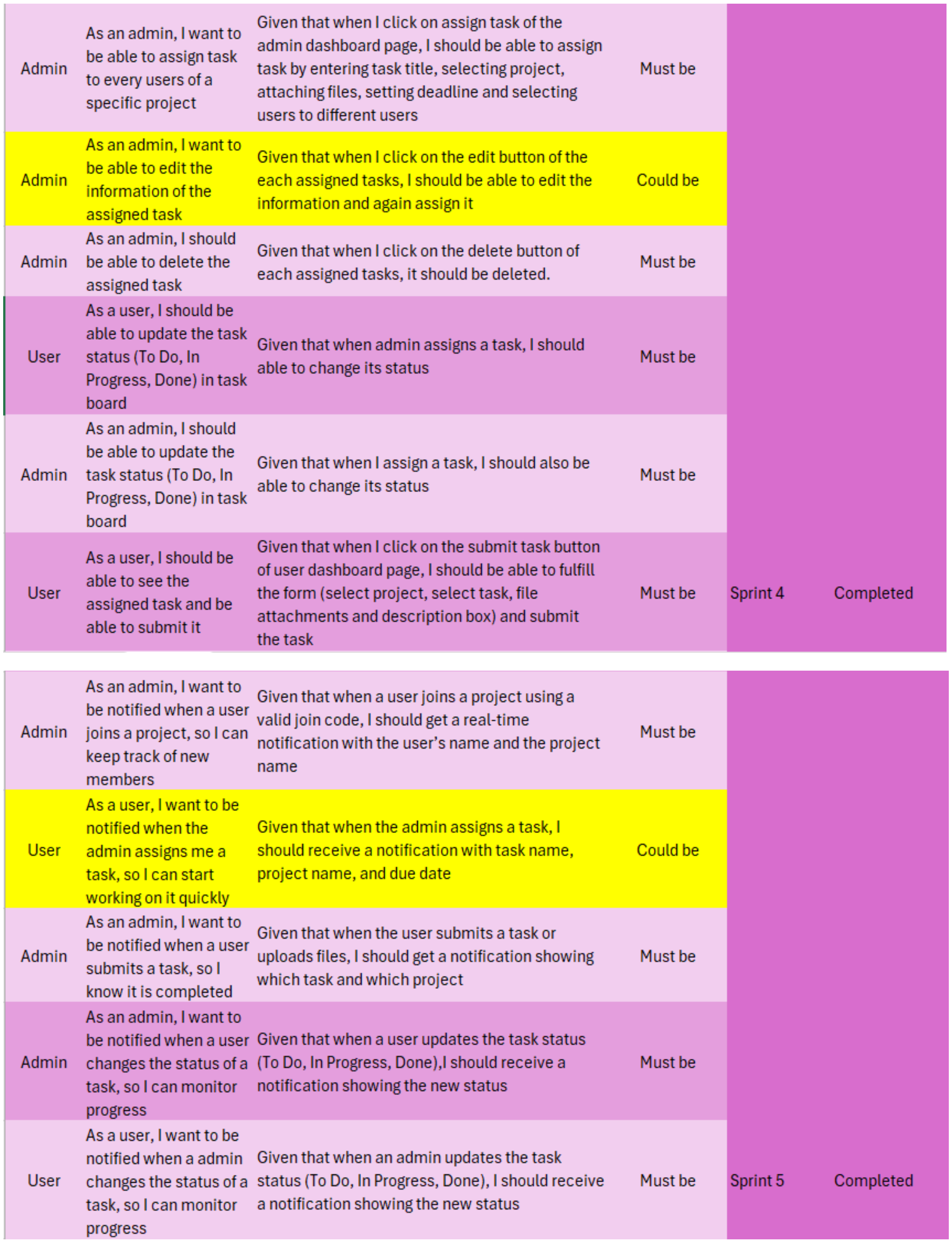
5.1 Product Backlog



Epic	User Story	Acceptance Criteria	Priority	Sprint	Status
Admin	As an admin, I want to be able to signup to the system	Given that when I signup as an admin entering information fullname, username, email, role, password and confirm password in a signup form, and clicking create account, admin account must be created	Must be	Sprint 1	Completed
Admin	As an admin, I want to be able to log in to the system so that I can access all other pages of the system	Given that when I log in, filling the username and password in the login form, I should be logged also I should be able to view all other pages	Must be		
Admin	As an admin, I want to be able to logout after I am logged in for security purpose	Given that when I click on the logout button I should be logged out and login button should be displayed automatically.	Must be		
User	As a user, I want to be able to signup to the system	Given that when I signup as a user entering information fullname, username, email, role, password and confirm password in a signup form, and clicking create account, user account must be created	Must be		
User	As a user, I want to be able to log in to the system so that I can access all other pages of the system	Given that when I log in, filling the username and password in the login form, I should be logged also I should be able to view all other pages	Must be		
User	As a user, I want to be able to logout after I am logged in for security purpose	Given that when I click on the logout button I should be logged out and login button should be displayed automatically.	Must be		



Epic	User Story	Acceptance Criteria	Priority	Sprint	Status
Admin	As an admin, I want to be able to add or create a new project into the system.	Given that when I click projects on the top of admin dashboard, it should show a no any project and I should be able to add new project by clicking the add project button	Must be	Sprint 2	Completed
Admin	As an admin, I want to be able to remove projects from the system.	Given that when I click on the project sidebar, it should show the list of project with remove button and when I click that button, the project should be removed.	Must be		
Admin	As an admin, I want to be able to add users into the project	Given that when I click on add user button I should be able to select the project and create a unique code for that project.	Must be		
User	As a user, I want to be able to join the project through code	Given that when I click on join the project button I should be able to enter the code provided by admin to join the project easily	Must be	Sprint 3	Completed



Admin	As an admin, I want to be able to assign task to every users of a specific project	Given that when I click on assign task of the admin dashboard page, I should be able to assign task by entering task title, selecting project, attaching files, setting deadline and selecting users to different users	Must be		
Admin	As an admin, I want to be able to edit the information of the assigned task	Given that when I click on the edit button of the each assigned tasks, I should be able to edit the information and again assign it	Could be		
Admin	As an admin, I should be able to delete the assigned task	Given that when I click on the delete button of each assigned tasks, it should be deleted.	Must be		
User	As a user, I should be able to update the task status (To Do, In Progress, Done) in task board	Given that when admin assigns a task, I should able to change its status	Must be		
Admin	As an admin, I should be able to update the task status (To Do, In Progress, Done) in task board	Given that when I assign a task, I should also be able to change its status	Must be		
User	As a user, I should be able to see the assigned task and be able to submit it	Given that when I click on the submit task button of user dashboard page, I should be able to fulfill the form (select project, select task, file attachments and description box) and submit the task	Must be	Sprint 4	Completed
Admin	As an admin, I want to be notified when a user joins a project, so I can keep track of new members	Given that when a user joins a project using a valid join code, I should get a real-time notification with the user's name and the project name	Must be		
User	As a user, I want to be notified when the admin assigns me a task, so I can start working on it quickly	Given that when the admin assigns a task, I should receive a notification with task name, project name, and due date	Could be		
Admin	As an admin, I want to be notified when a user submits a task, so I know it is completed	Given that when the user submits a task or uploads files, I should get a notification showing which task and which project	Must be		
Admin	As an admin, I want to be notified when a user changes the status of a task, so I can monitor progress	Given that when a user updates the task status (To Do, In Progress, Done),I should receive a notification showing the new status	Must be		
User	As a user, I want to be notified when a admin changes the status of a task, so I can monitor progress	Given that when an admin updates the task status (To Do, In Progress, Done), I should receive a notification showing the new status	Must be	Sprint 5	Completed

Figure 4: Product Backlog

5.2 Software Requirement Specifications (SRS)

Legend:

AMS – Authentication Management System

PMS – Project Management System

UMS – User Management System

TMS – Task Management System

NMS – Notification Management System

Requirement Types:

FR – Functional Requirement

NFR – Non-Functional Requirement

UR – Usability Requirement

5.3 Subsystem 1 – Authentication Management System

This document outlines the artifact design for the Authentication Subsystem as part of the final year project. User authentication gets secure management through this subsystem, so it protects authorized access to the Task Management System.

This subsystem provides users with secure identification procedures which offer both flexibility and defensive integrity. User registration functionality and login authentication services and role-based access control capabilities will be managed by this subsystem to realize data security and system integrity across all platform elements.

5.3.1 System Architecture and Design

- Activity Diagram

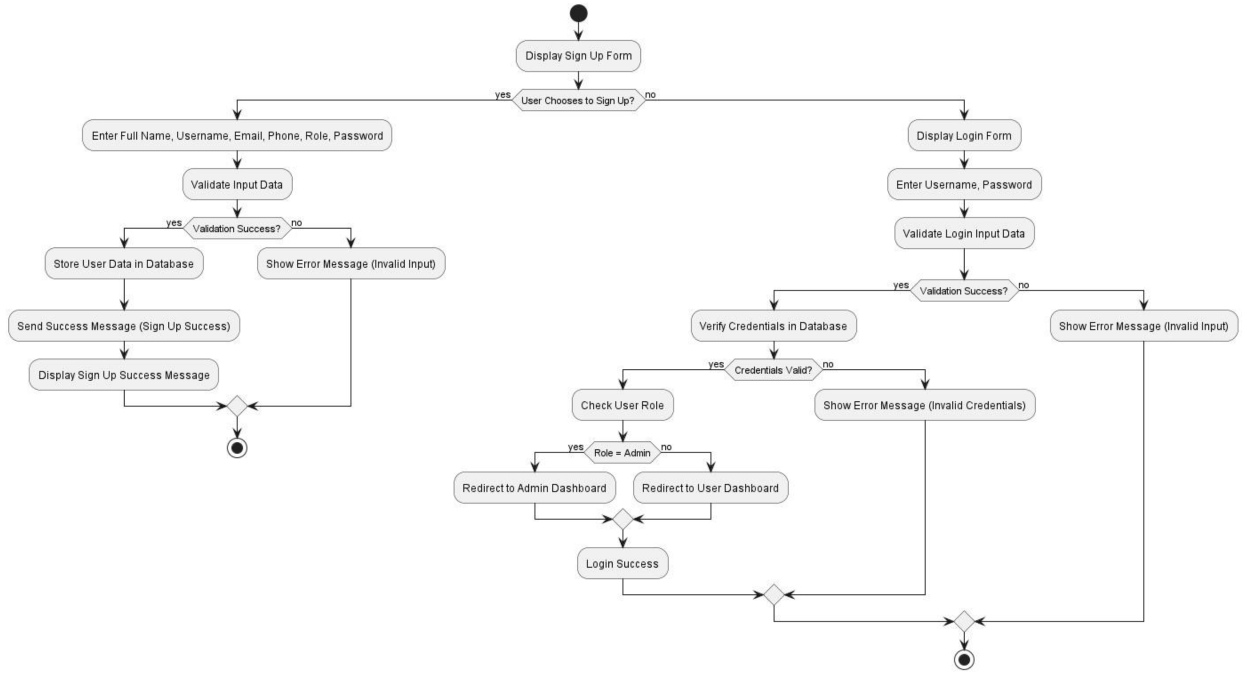

Display Sign Up Form
yes
User Chooses to Sign Up?
no
Enter Full Name, Username, Email, Phone, Role, Password
Validate Input Data
yes
Validation Success?
no
Store User Data in Database
Show Error Message (Invalid Input)
Send Success Message (Sign Up Success)
Display Sign Up Success Message
Display Login Form
Enter Username, Password
Validate Login Input Data
yes
Validation Success?
no
Verify Credentials in Database
Show Error Message (Invalid Input)
yes
Credentials Valid?
no
Check User Role
Show Error Message (Invalid Credentials)
yes
Role = Admin
no
Redirect to Admin Dashboard
Redirect to User Dashboard
Login Success

Figure 5: Activity Diagram of AMS

- Use case Diagram

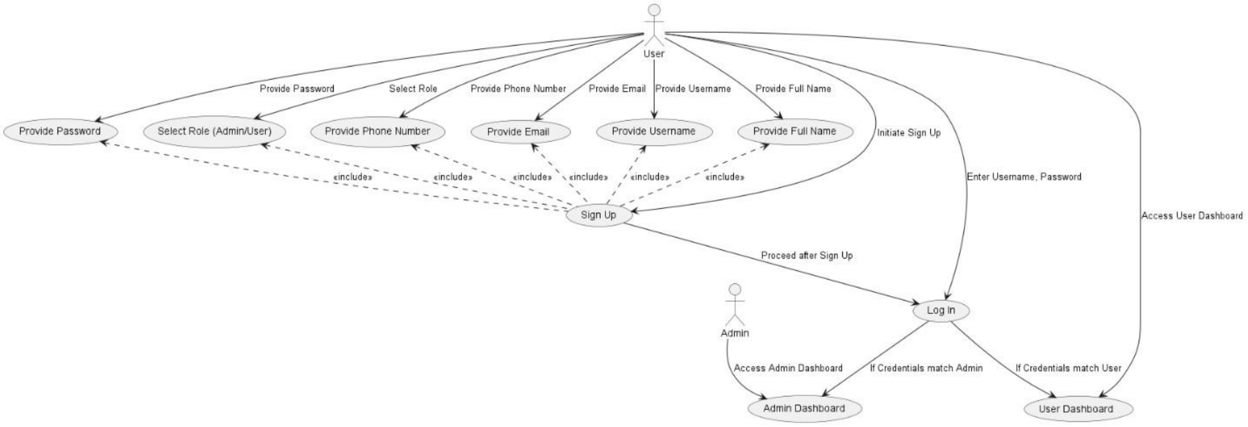


Figure 6: Use Case Diagram of AMS

- Entity Relationship Diagram (ERD)

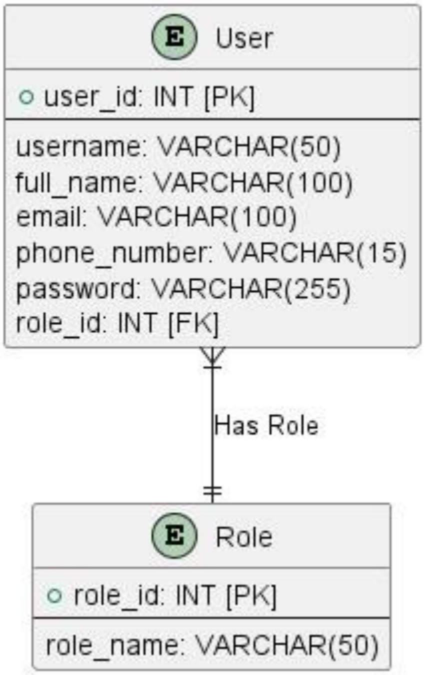


Figure 7: ER Diagram of AMS

- Class Diagram

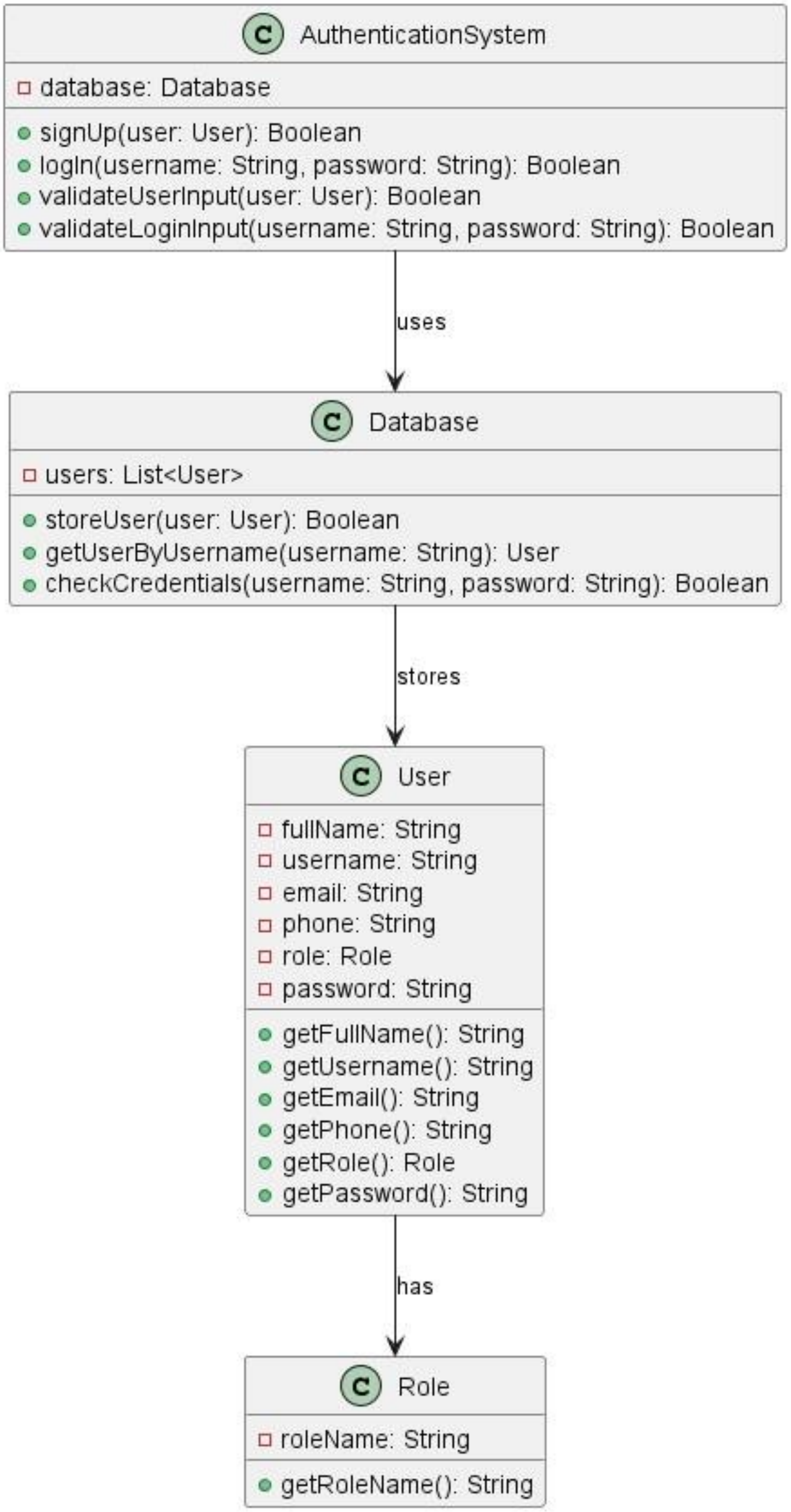


Figure 8: Class Diagram of AMS

Figure 6: Class Diagram

- Sequence Diagram

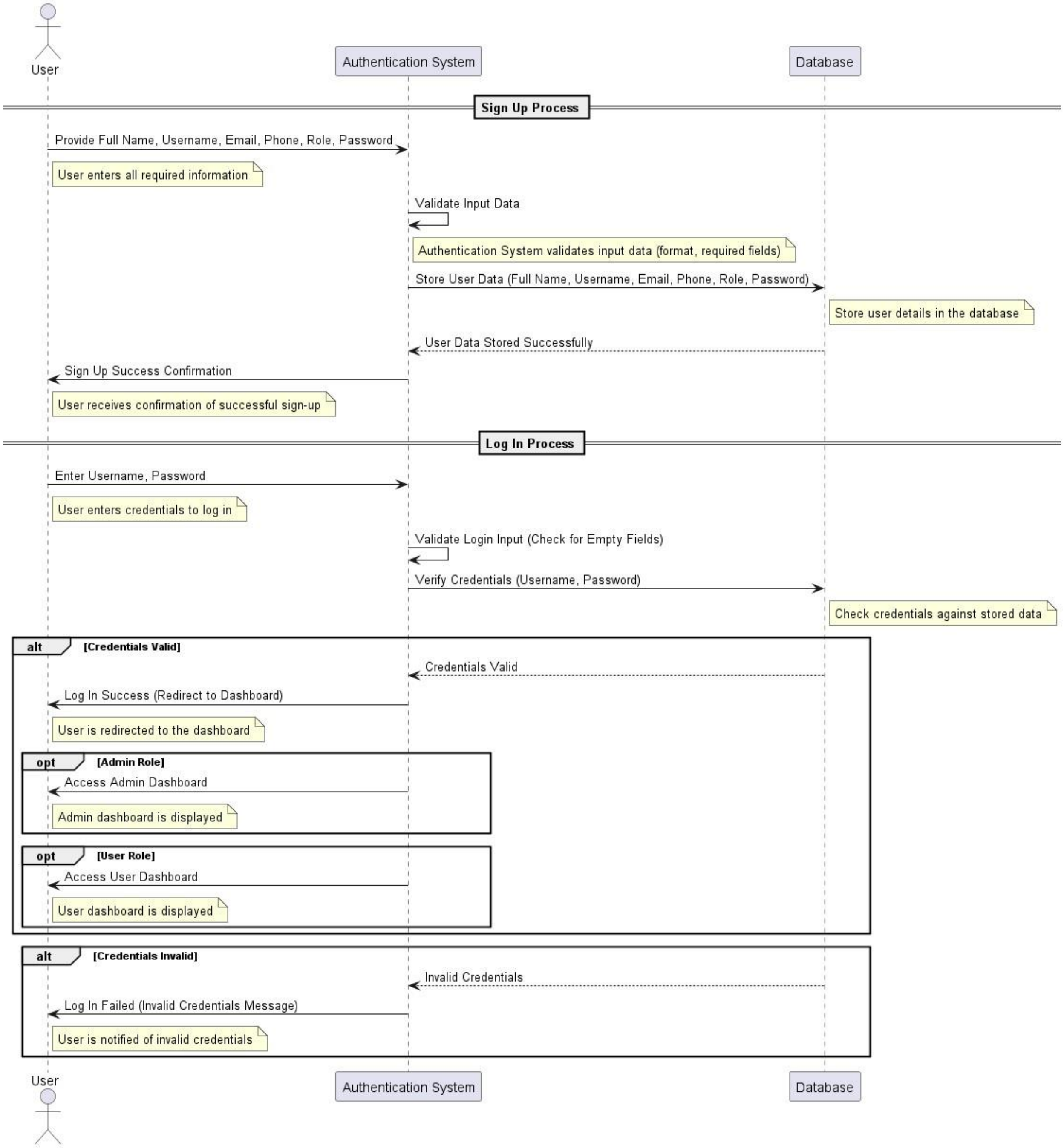
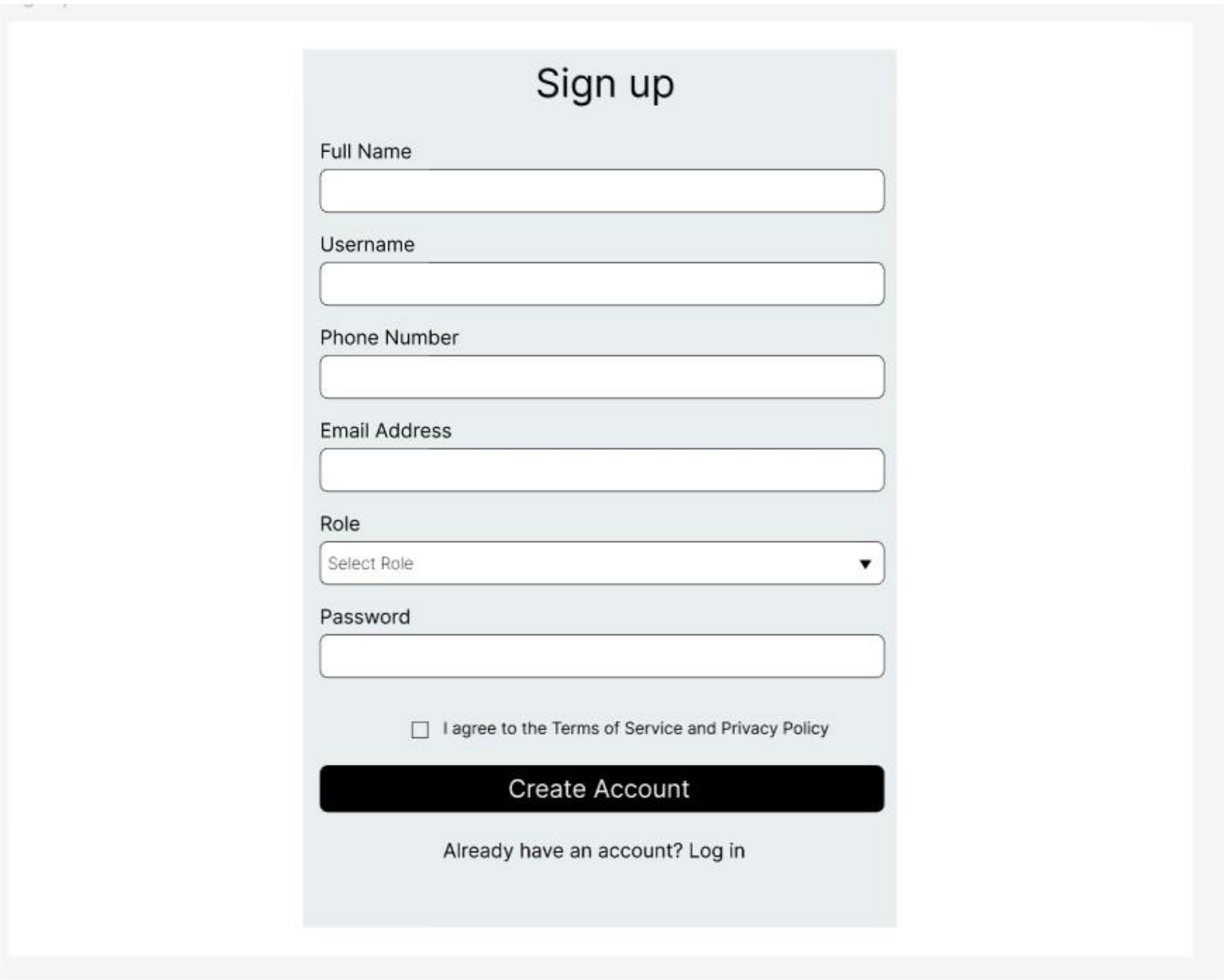

User
Authentication System
Database
Sign Up Process
Provide Full Name, Username, Email, Phone, Role, Password
User enters all required information
Validate Input Data
Authentication System validates input data (format, required fields)
Store User Data (Full Name, Username, Email, Phone, Role, Password)
Store user details in the database
User Data Stored Successfully
Sign Up Success Confirmation
User receives confirmation of successful sign-up
Log In Process
Enter Username, Password
User enters credentials to log in
Validate Login Input (Check for Empty Fields)
Verify Credentials (Username, Password)
Check credentials against stored data
alt
[Credentials Valid]
Credentials Valid
Log In Success (Redirect to Dashboard)
User is redirected to the dashboard
opt
[Admin Role]
Access Admin Dashboard
Admin dashboard is displayed
opt
[User Role]
Access User Dashboard
User dashboard is displayed
alt
[Credentials Invalid]
Invalid Credentials
Log In Failed (Invalid Credentials Message)
User is notified of invalid credentials
User
Authentication System
Database

Figure 9: Sequence Diagram of AMS

- **Wireframes**

- **Sign Up**



Sign up

Full Name

Username

Phone Number

Email Address

Role

Select Role ▼

Password

□ I agree to the Terms of Service and Privacy Policy

Create Account

Already have an account? Log in

Figure 10: Signup wireframe

- **Log In**

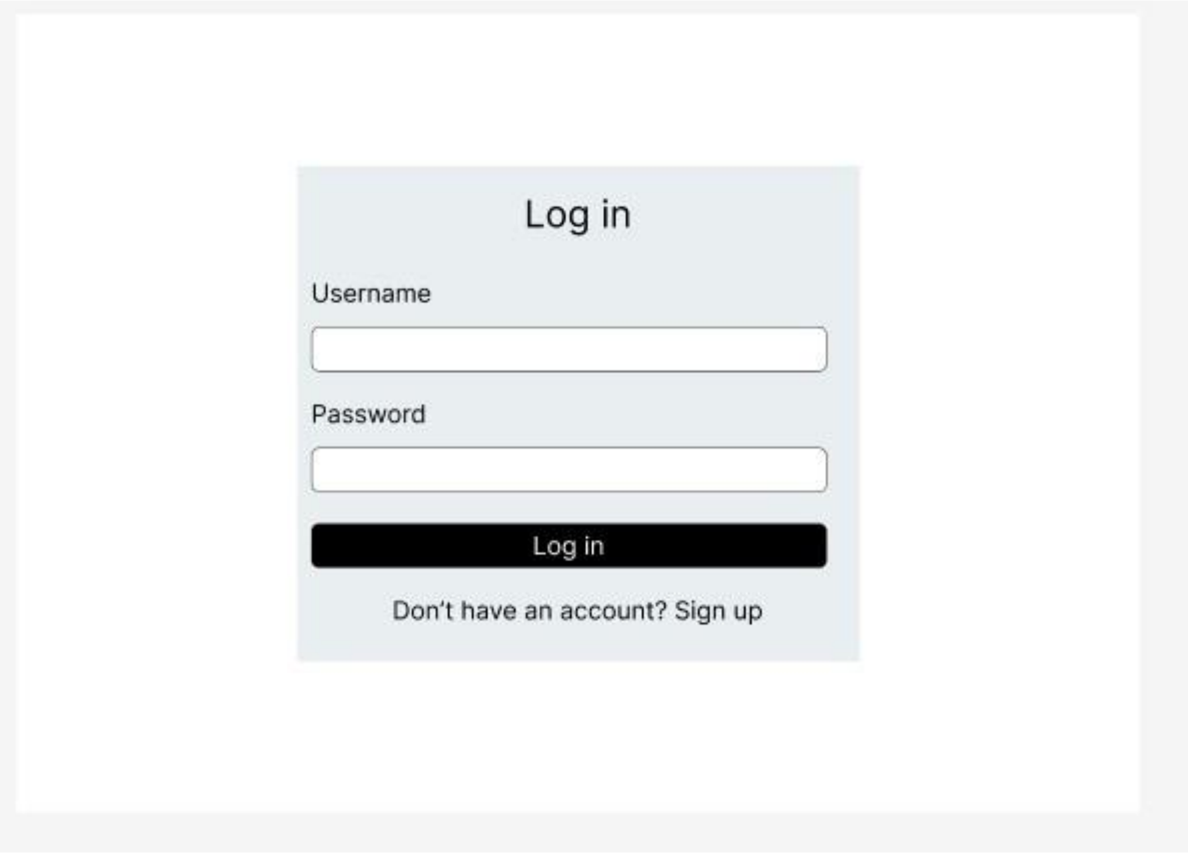


Figure 11: Log in wireframe

- Software Requirement Specifications (SRS)

Request No.	Requirement Description	Use Case	Moscow Prioritization
AMS-FR-1	Admin and User must be able to signup securely.	Signup	Must have
AMS-FR-2	Admin and User must be able to log in with correct credentials.	Login	Must have
AMS-FR-3	Admin and User must be able to log out securely.	Log out	Must have
AMS-NFR-4	Password must be stored securely (hashed).	Security	Must have

Table 1: SRS of AMS

5.3.2 Critical Features and Implementation

- Key Functionalities

Feature 1: User Sign-up

Through the Sign-Up interface users can establish new accounts through submitting their account information. Users who need system access must register while the application maintains the secure storage of their information.

Implementation:

- Users need to complete a registration form containing full name, username, email, phone number, password and their designated role. Users complete a form which contains mandatory input validations to verify both mandatory field entry and proper data formatting.
- The Authentication System class processes user input before validating it for accuracy then stores user information in the database.
- Security tests verify that passwords fulfill requirements regarding minimum length and complexity and confirm uniqueness of both usernames and emails addresses.
- The system must perform an encryption process when storing passwords into its database to enhance safety.
- The implementation of bcrypt hashing serves as the standard approach to secure store passwords.

Feature 2: User Login

Through Login functionality users can access the system by providing their username along with their password. The system authenticates user credentials by checking them against its database then enables access according to user roles that include Admin or User.

Implementation:

- Users need to input their username together with password on a login form.

- Both required fields must have content before continuing validation tests.
- The Authentication System executes a search against the Database for users matching the given username.
- A match check occurs for the user's stored hashed password versus their supplied password which they provide after our system hashed it. Successful authorization results in system access.
- The system reviews the assigned role of the active user. The system sends regular Users toward the User Dashboard however it directs Admins straight to the Admin Dashboard.
- When credentials fail authentication, the system shows an error message which prompts users to try again.
- A secure password verification algorithm (bcrypt) helps compare hashes stored in the system with entered passwords.

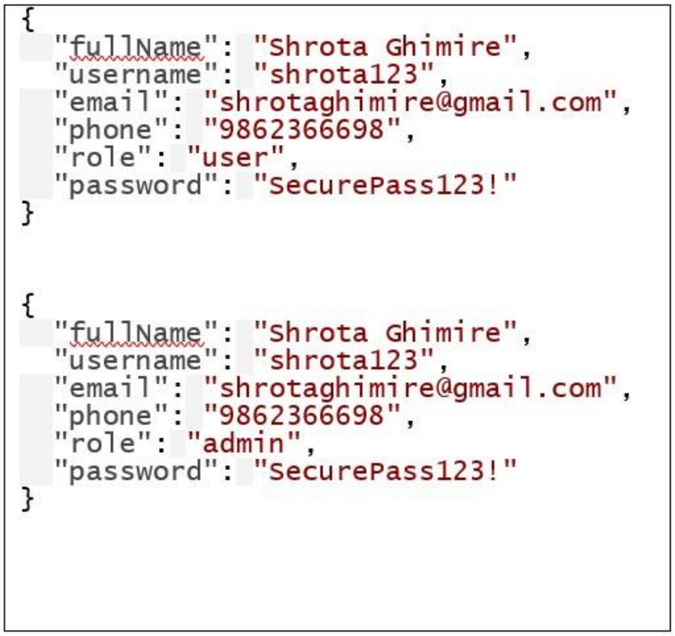
- API and Database Implementation

API Endpoints

Through its RESTful API system, the authentication structure handles both new user registration alongside user login operations as well as access restriction functions. Below is a structured list of the key API endpoints:

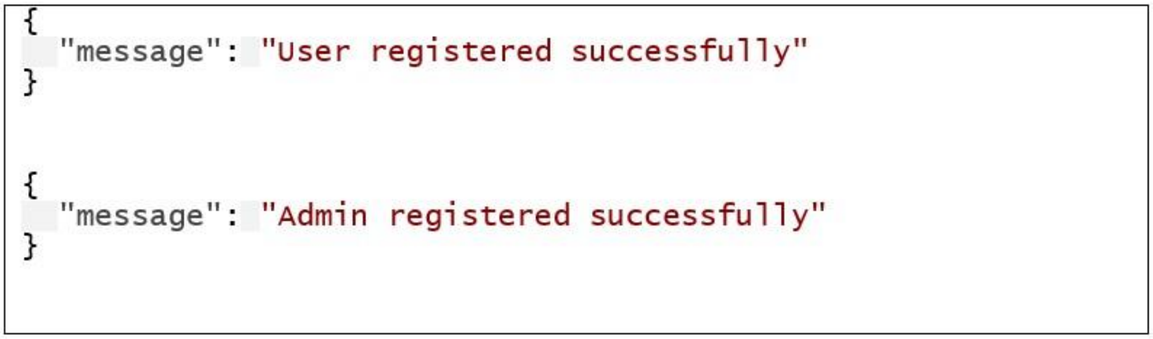
• **User Signup**

- Endpoint: POST/api/signup
- Description: Allow new users to register an account.
- Request Body (JSON):

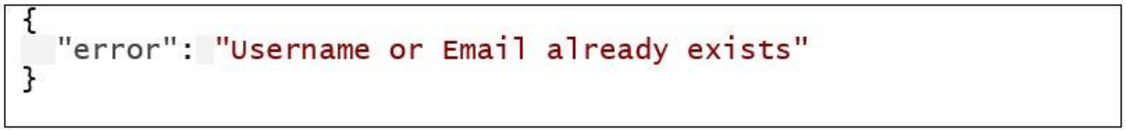


```json
{
  "fullName": "Shrota Ghimire",
  "username": "shrota123",
  "email": "shrotaghimire@gmail.com",
  "phone": "9862366698",
  "role": "user",
  "password": "SecurePass123!"
}


{
  "fullName": "Shrota Ghimire",
  "username": "shrota123",
  "email": "shrotaghimire@gmail.com",
  "phone": "9862366698",
  "role": "admin",
  "password": "SecurePass123!"
}
```

- Response:



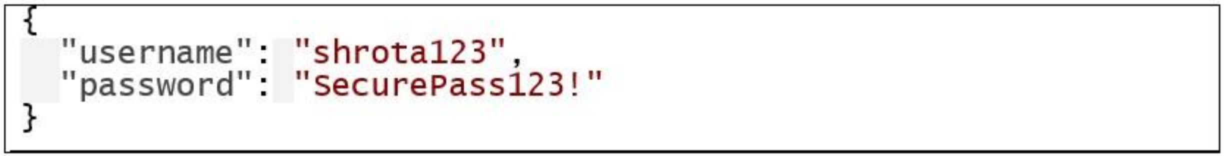
```
{
  "message": "User registered successfully"
}


{
  "message": "Admin registered successfully"
}
```



```
{
  "error": "Username or Email already exists"
}
```

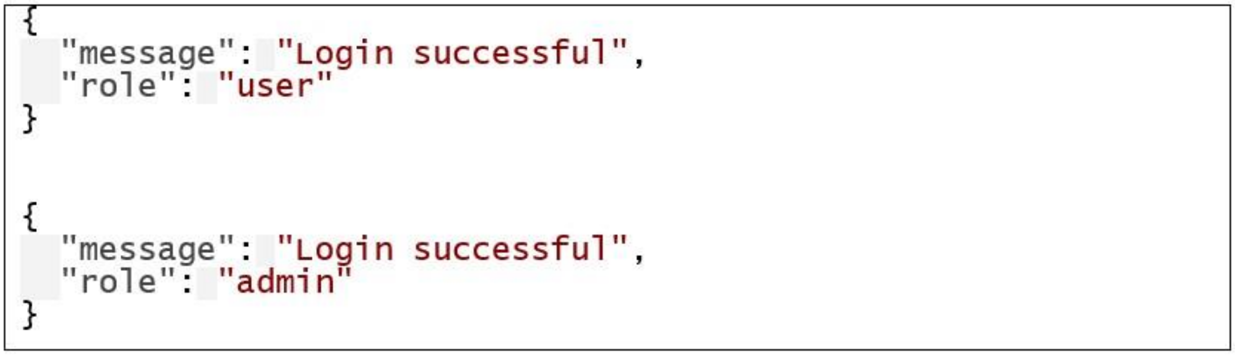
• **User Login**

- Endpoint: POST /api/login
- Description: Authenticates a user or admin based on username and password.
- Request Body (JSON):

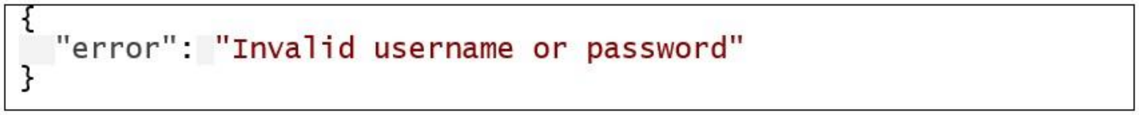


```
{
  "username": "shrota123",
  "password": "SecurePass123!"
}
```

- Response:

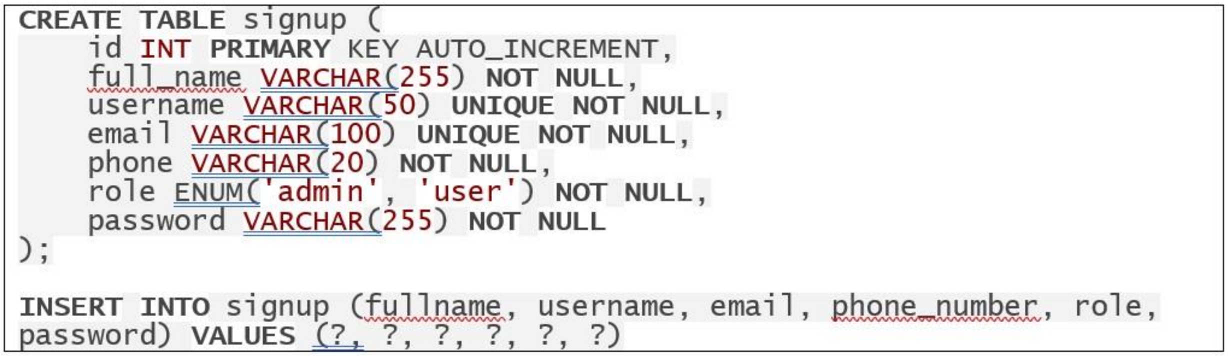


```
{
  "message": "Login successful",
  "role": "user"
}


{
  "message": "Login successful",
  "role": "admin"
}
```



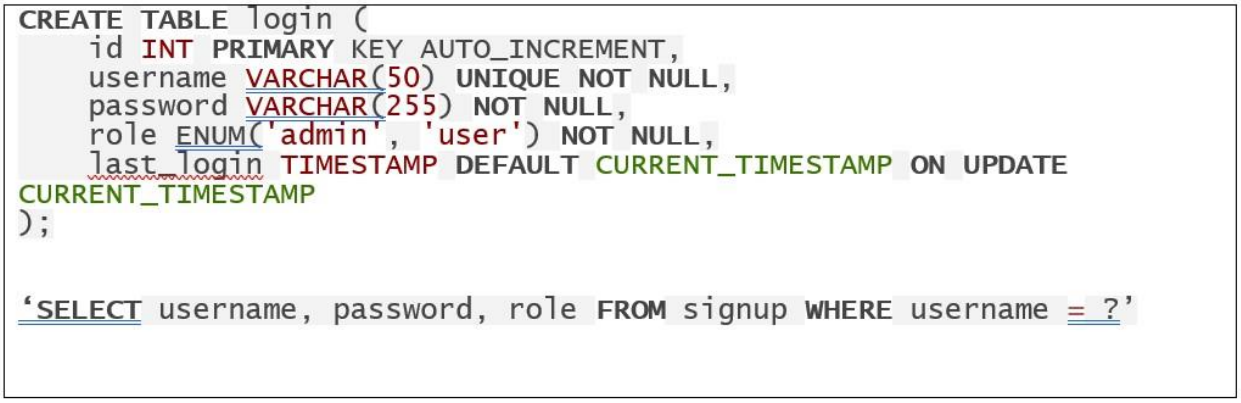
```
{
  "error": "Invalid username or password"
}
```

- **Database Schema**



```
CREATE TABLE signup (
    id INT PRIMARY KEY AUTO_INCREMENT,
    full_name VARCHAR(255) NOT NULL,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    phone VARCHAR(20) NOT NULL,
    role ENUM('admin', 'user') NOT NULL,
    password VARCHAR(255) NOT NULL
);

INSERT INTO signup (fullname, username, email, phone_number, role,
password) VALUES (?, ?, ?, ?, ?, ?)
```

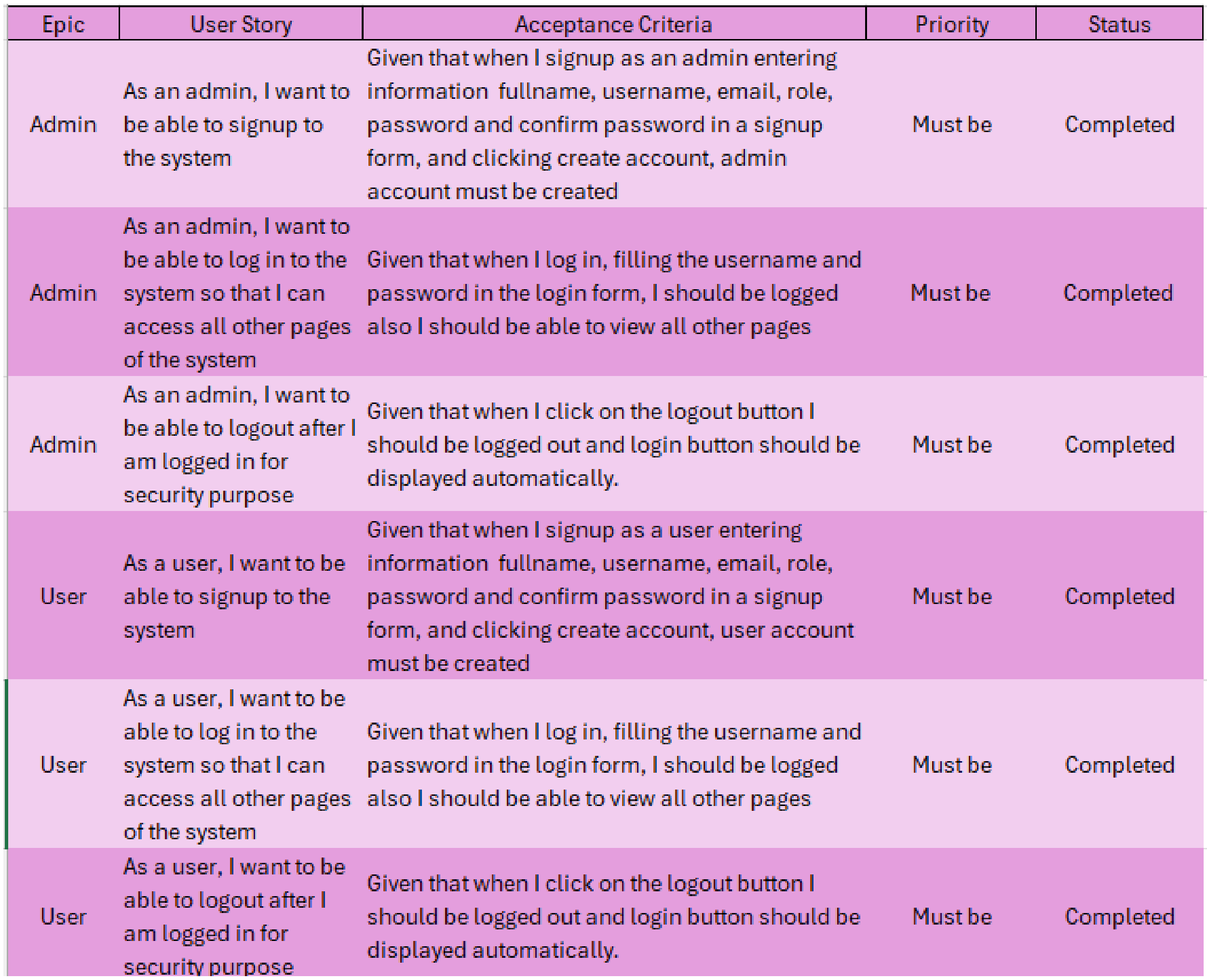
Figure 15: Signup Database Schema



```
CREATE TABLE login (
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    role ENUM('admin', 'user') NOT NULL,
    last_login TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);


‘SELECT username, password, role FROM signup WHERE username = ?’
```

5.3.3 Development Process

- Sprint Backlog



Epic	User Story	Acceptance Criteria	Priority	Status
Admin	As an admin, I want to be able to signup to the system	Given that when I signup as an admin entering information fullname, username, email, role, password and confirm password in a signup form, and clicking create account, admin account must be created	Must be	Completed
Admin	As an admin, I want to be able to log in to the system so that I can access all other pages of the system	Given that when I log in, filling the username and password in the login form, I should be logged also I should be able to view all other pages	Must be	Completed
Admin	As an admin, I want to be able to logout after I am logged in for security purpose	Given that when I click on the logout button I should be logged out and login button should be displayed automatically.	Must be	Completed
User	As a user, I want to be able to signup to the system	Given that when I signup as a user entering information fullname, username, email, role, password and confirm password in a signup form, and clicking create account, user account must be created	Must be	Completed
User	As a user, I want to be able to log in to the system so that I can access all other pages of the system	Given that when I log in, filling the username and password in the login form, I should be logged also I should be able to view all other pages	Must be	Completed
User	As a user, I want to be able to logout after I am logged in for security purpose	Given that when I click on the logout button I should be logged out and login button should be displayed automatically.	Must be	Completed

Figure 12: Sprint backlog of AMS

5.3.4 Execution Details

The development approach implemented systematic workflow beginning with frontend layout construction which then integrated backend application programming interfaces.

- **Frontend Development**

The central priority at first was to assemble an engaged user interface that interacted with users. The frontend used HTML and CSS along with JS to become a fast and modular development environment. Frontend components received integration treatment to work with form validation alongside error handling functionality as well as API integration. I moved on to creating application responsiveness while sustaining a natural user experience after completing the UI stage.

- **Backend Development**

The backend development started with Node.js after frontend development finished. A set of APIs was designed to manage user authentication at different levels through login and signup and role-based authentication services. The database architecture received architectural patterns to store credentials alongside valuable operational information. User data management demands dependable performance which led to MySQL selection.

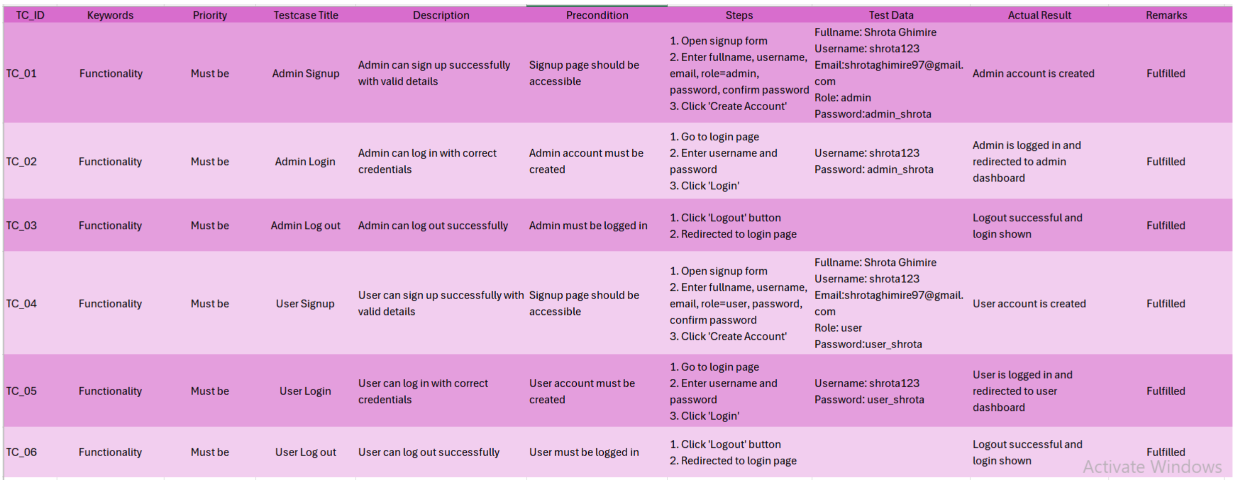
- **Integration**

Integration of frontend design with backend APIs followed development completion of frontend and backend components independently. The authentication system performed requests at the frontend which were transmitted to the backend functions.

5.3.5 Testing Approach

Testing stands as an essential developmental component for verifying that every system component performs to specifications. I have done black box testing in my project, below is the testing approach.

- **Black Box Testing**



TC_ID	Keywords	Priority	Testcase Title	Description	Precondition	Steps	Test Data	Actual Result	Remarks
TC_01	Functionality	Must be	Admin Signup	Admin can sign up successfully with valid details	Signup page should be accessible	1. Open signup form 2. Enter fullname, username, email, role=admin, password, confirm password 3. Click 'Create Account'	Fullname: Shrota Ghimire Username: shrota123 Email:shrotaghimire97@gmail.com Role: admin Password:admin_shrota	Admin account is created	Fulfilled
TC_02	Functionality	Must be	Admin Login	Admin can log in with correct credentials	Admin account must be created	1. Go to login page 2. Enter username and password 3. Click 'Login'	Username: shrota123 Password: admin_shrota	Admin is logged in and redirected to admin dashboard	Fulfilled
TC_03	Functionality	Must be	Admin Log out	Admin can log out successfully	Admin must be logged in	1. Click 'Logout' button 2. Redirected to login page		Logout successful and login shown	Fulfilled
TC_04	Functionality	Must be	User Signup	User can sign up successfully with valid details	Signup page should be accessible	1. Open signup form 2. Enter fullname, username, email, role=user, password, confirm password 3. Click 'Create Account'	Fullname: Shrota Ghimire Username: shrota123 Email:shrotaghimire97@gmail.com Role: user Password:user_shrota	User account is created	Fulfilled
TC_05	Functionality	Must be	User Login	User can log in with correct credentials	User account must be created	1. Go to login page 2. Enter username and password 3. Click 'Login'	Username: shrota123 Password: user_shrota	User is logged in and redirected to user dashboard	Fulfilled
TC_06	Functionality	Must be	User Log out	User can log out successfully	User must be logged in	1. Click 'Logout' button 2. Redirected to login page		Logout successful and login shown	Fulfilled

Figure 13: Test Case of AMS

5.4 Subsystem 2 – Project Management System

This document presents an outline of the artifact design of the Project Management System within the Task Master final year project. Only an administrator can use the Project Management section to create, add or remove projects in the Task Master system. This means that daily users cannot adjust the details of the project list in the system. A project is added by the admin to the system so that people can create their accounts and work on the tasks inside. If a project is no longer needed or used, the administrator can get rid of it to maintain an up-to-date website. With this approach, it is easier to monitor the project, avoid unwanted changes and offer users only those projects that are authorized by the group. This ensures that the admin manages the system safely and properly.

5.4.1 System Architecture and Design

- Activity Diagram

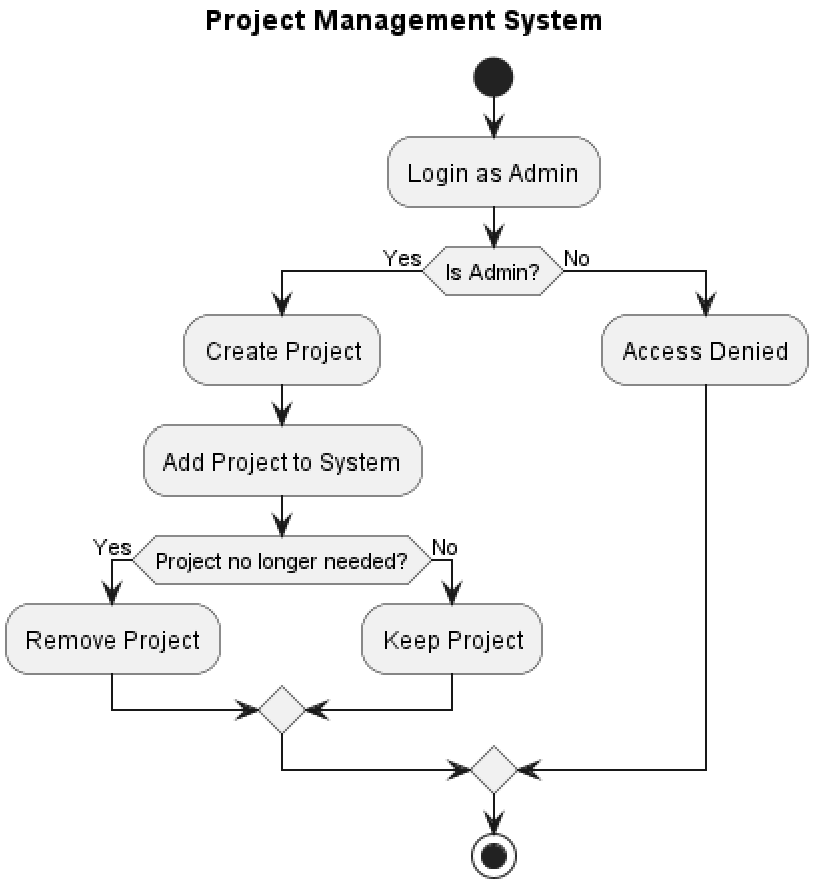


Figure 14: Activity Diagram of PMS

- Use Case Diagram

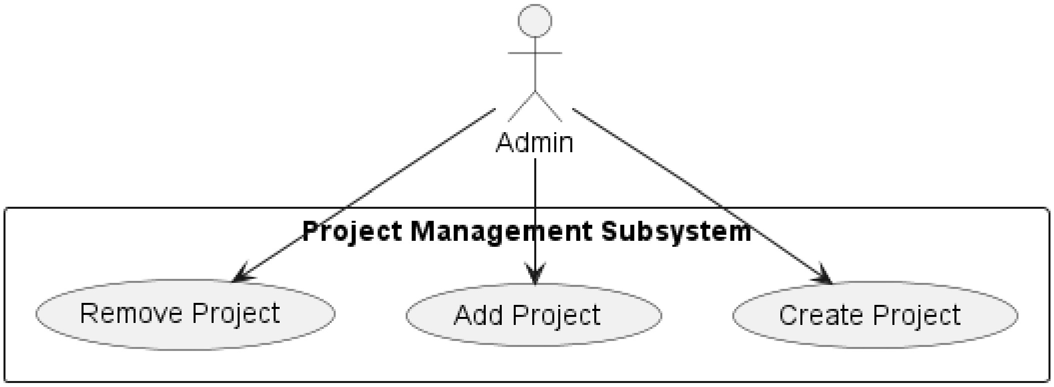


Figure 15: Use Case Diagram of PMS

- Entity Relationship Diagram

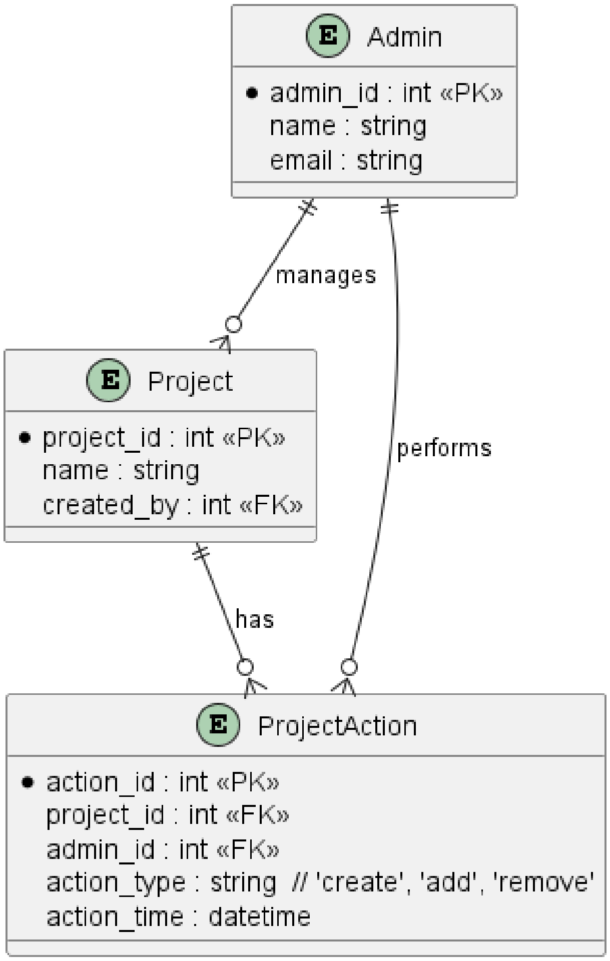


Figure 16: ER Diagram of PMS

- **Class Diagram**

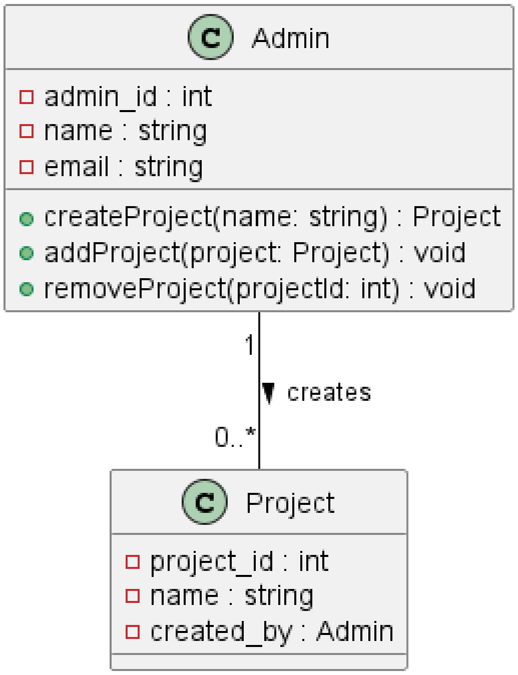


Figure 17: Class Diagram of PMS

- Sequence Diagram

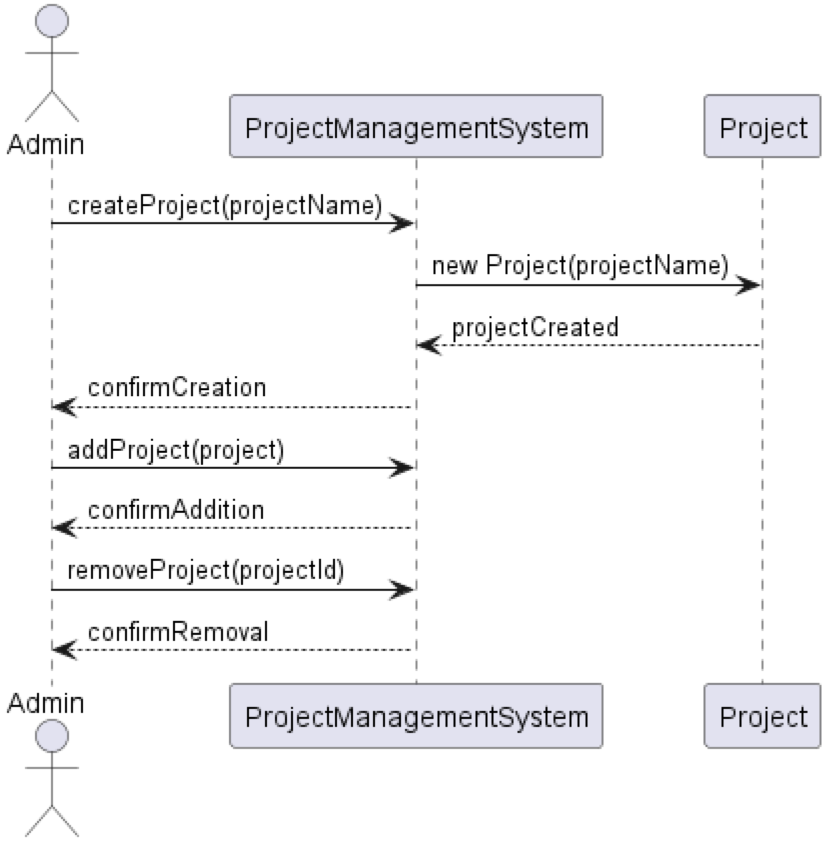


Figure 18: Sequence Diagram of PMS

- Wireframes

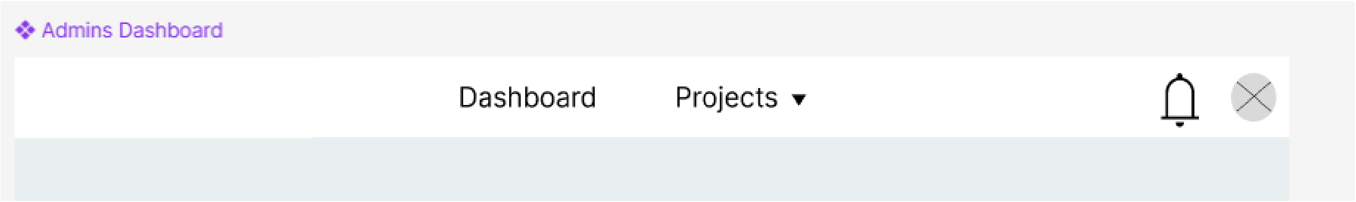


Figure 19: Wireframe of PMS

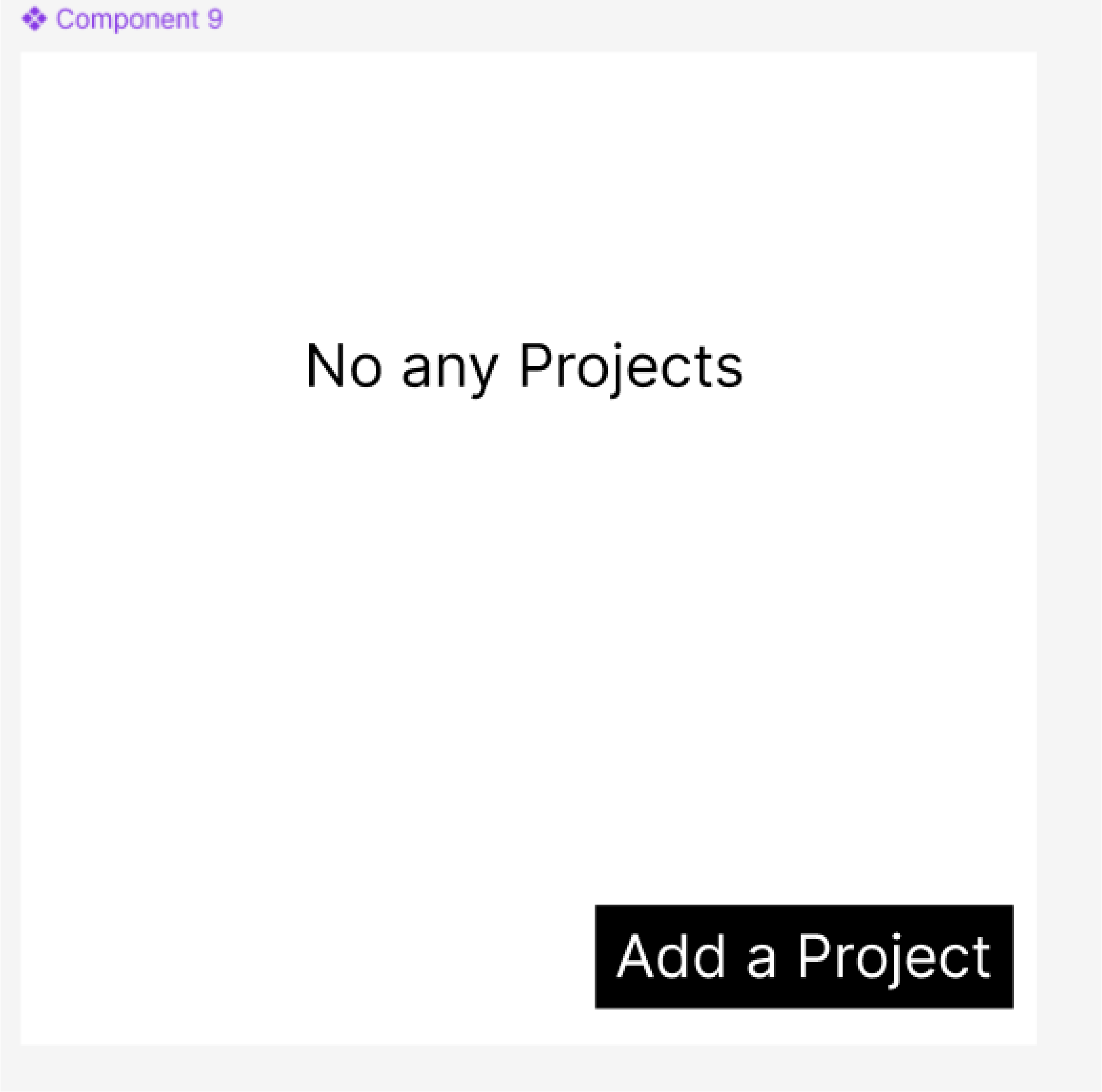


Figure 20: Wireframe of PMS

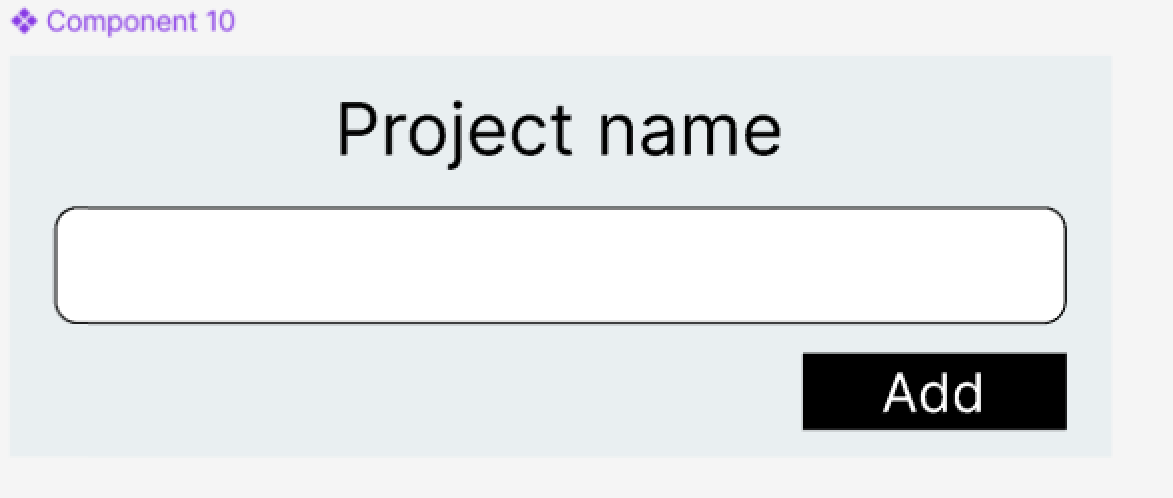


Figure 21: Wireframe of PMS

- Software Requirement Specifications (SRS)

Request No.	Requirement Description	Use Case	Moscow Prioritization
PMS-FR-1	Admin must be able to create or add a new project entering the project name.	Create and add project	Must have
PMS-FR-2	Admin must be able to view the existing project.	View Projects	Must have
PMS-FR-3	Admin must be able to delete any projects.	Delete Projects	Must have
PMS-UR-4	The system should ask for confirmation before deleting any projects.	Delete Confirmation	Must have

5.4.2 Critical Features and Functionalities

- Key Functionalities

Feature 1: Project Creation

Using the Project Creation interface administrators generate new projects by supplying project name. The system establishes unique project codes for security purposes during controlled access administration.

Implementation:

- A project creation form requires all admins to enter the project name alongside its description with additional optional settings.
- Project Management System class receives the submitted information which it then validates and establishes permanent storage of this data in the database.
- A new project code automatically generates itself for every new project to enable user addition.
- Duplicate project names are prevented by this system across all projects within the same organization for maintaining clear identification.
- The security checks for project details stop unauthorized users from making unauthorized modifications.
- Administrative personnel have full control to refresh and erase projects through the system for enhanced project management capabilities.

- API and Database Implementation

API Endpoints

The Project Management module operates via the RESTful API system to execute project creation. The API system contains multiple structured endpoints which function as follows:

- **Project Creation**

- Endpoint: POST/api/add-project
- Description: Allow admins to create new projects
- Request Body (JSON):

```
{
   "name": "Task Management System",
  "adminId": 1,
  "project_key": ebf5eb18
}
```

- Response:

{

```
  "message": "Project created successfully"
}
```

- **Deleting Project**

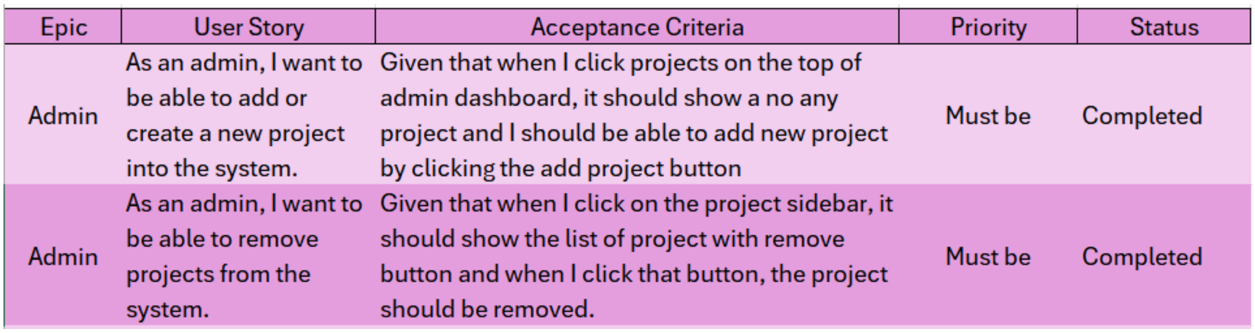
- Endpoint: DELETE/api/project/projectKey
- Description: Allows an admin to remove project.

- **Database Schema**

```
CREATE TABLE `project` (
  `id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `admin_id` int(11) DEFAULT NULL,
  `project_key` varchar(10) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `invite_code` varchar(10) DEFAULT NULL,
  `invite_expires_at` datetime DEFAULT NULL
)
```

5.4.3 Development Process

- Sprint Backlog



Epic	User Story	Acceptance Criteria	Priority	Status
Admin	As an admin, I want to be able to add or create a new project into the system.	Given that when I click projects on the top of admin dashboard, it should show a no any project and I should be able to add new project by clicking the add project button	Must be	Completed
Admin	As an admin, I want to be able to remove projects from the system.	Given that when I click on the project sidebar, it should show the list of project with remove button and when I click that button, the project should be removed.	Must be	Completed

Figure 22: Sprint backlog of PMS

5.4.4 Execution Details

I executed this work through Agile methodology to create and optimize separate elements in the system. The development approach implemented systematic workflow beginning with frontend layout construction which then integrated backend application programming interfaces.

- **Frontend Development**

Development begins with creating an attractive user interface system that lets users communicate with the system. A fast and modular development environment will be built from HTML and CSS and JS components at the front end. The frontend components need to integrate with form validation as well as error handling functions and API integration services. The developers will build application responsiveness before focusing on maintaining user experience naturalness after finishing the UI development stage.

- **Backend Development**

The Node.js backend development launches following completion of frontend development. The system will require APIs to perform adding user and removing through user management. The database architecture requires architectural patterns to keep operational information and credentials in the storage system. MySQL will be selected because performing data management on users demands dependable system performance.

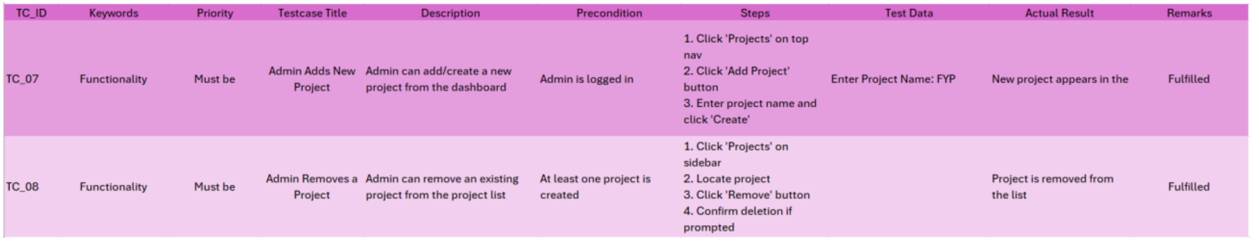
- **Integration**

The front-end design integration with backend APIs will commence after the independent development of both front-end and back-end components. The user management executes frontend requests which are transmitted to backend operations.

5.4.5 Testing Approach

Testing stands as an essential developmental component for verifying that all system components perform to specifications. Research-based testing extends across unit testing and integration testing and user testing for complete system examination.

- Black Box Testing



TC_ID	Keywords	Priority	Testcase Title	Description	Precondition	Steps	Test Data	Actual Result	Remarks
TC_07	Functionality	Must be	Admin Adds New Project	Admin can add/create a new project from the dashboard	Admin is logged in	1. Click 'Projects' on top nav 2. Click 'Add Project' button 3. Enter project name and click 'Create'	Enter Project Name: FYP	New project appears in the	Fulfilled
TC_08	Functionality	Must be	Admin Removes a Project	Admin can remove an existing project from the project list	At least one project is created	1. Click 'Projects' on sidebar 2. Locate project 3. Click 'Remove' button 4. Confirm deletion if prompted		Project is removed from the list	Fulfilled

Figure 23: Test case of PMS

5.5 Subsystem 3 – User Management System

With this, administrators can control access to various projects. The admin introduces unique codes for each project to ensure its safety. Users are given the codes, and they can use them to join the correct projects securely. If a user types in the proper code, they become involved in the project.

The admin can remove any user from a project, when necessary, this allows the project to remain neat and secure. The system allows an admin to quickly disable the account if someone should not be able to access it anymore. All in all, this system allows admins to handle who can use the project, safeguard its information and exclude unauthorized users.

5.5.1 System Architecture and Design

- Activity Diagram

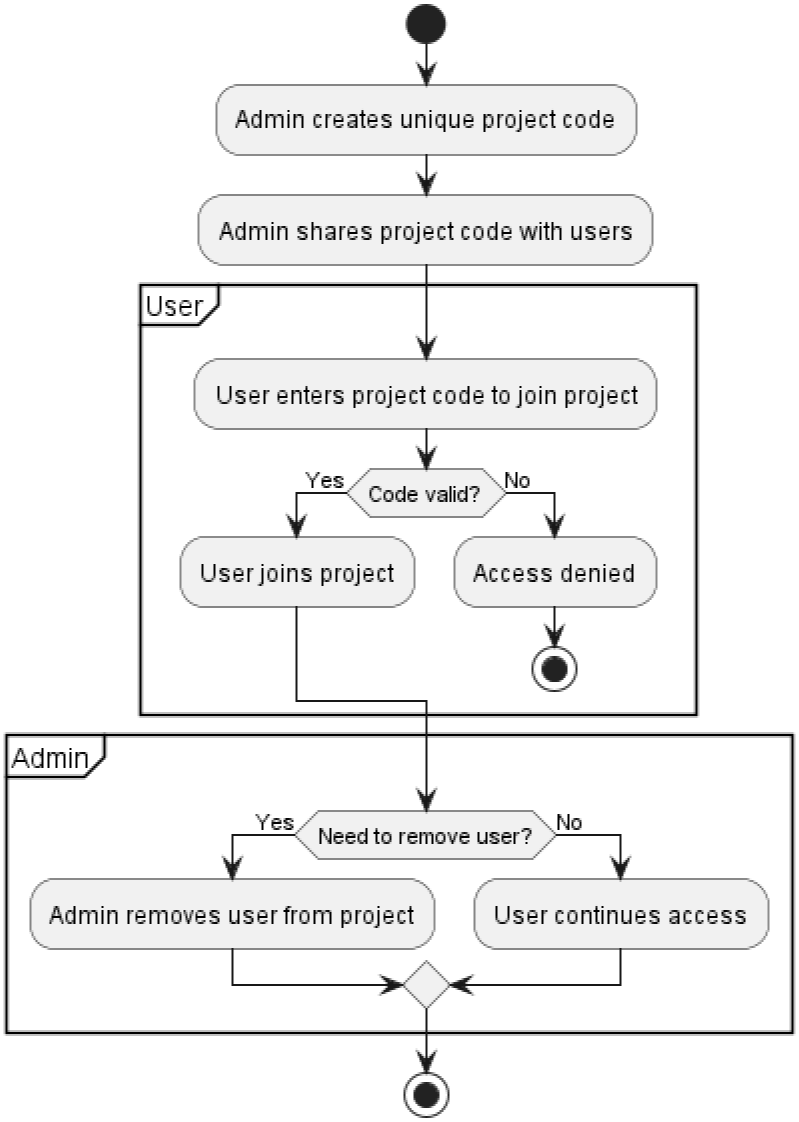

Admin creates unique project code
Admin shares project code with users
User
User enters project code to join project
Code valid?
Yes
No
User joins project
Access denied
Admin
Need to remove user?
Yes
No
Admin removes user from project
User continues access

Figure 24: Activity Diagram of UMS

- **Use Case Diagram**

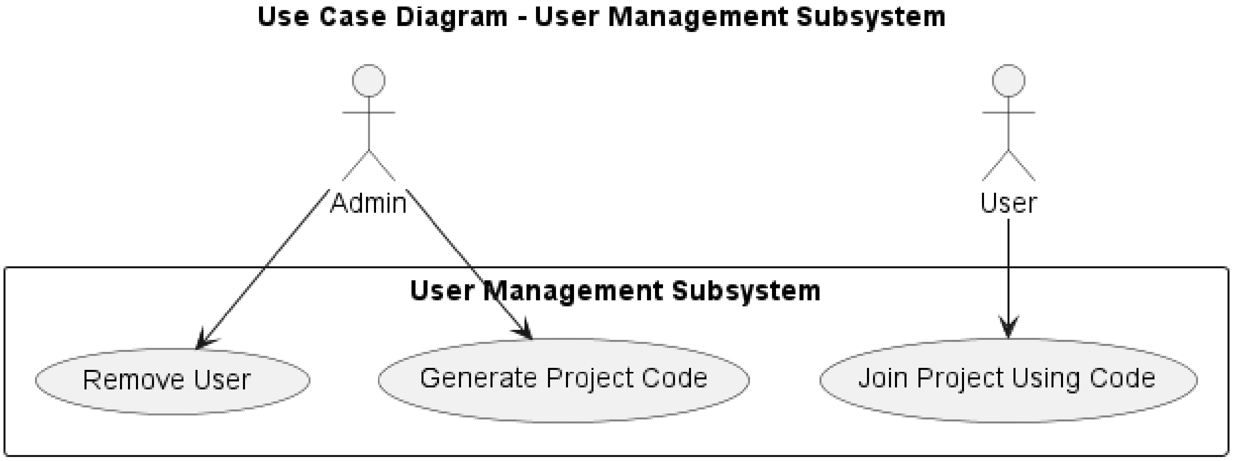

Use Case Diagram - User Management Subsystem
Admin
User
User Management Subsystem
Remove User
Generate Project Code
Join Project Using Code

Figure 25: Use case Diagram of UMS

- **Entity Relationship Diagram (ERD)**

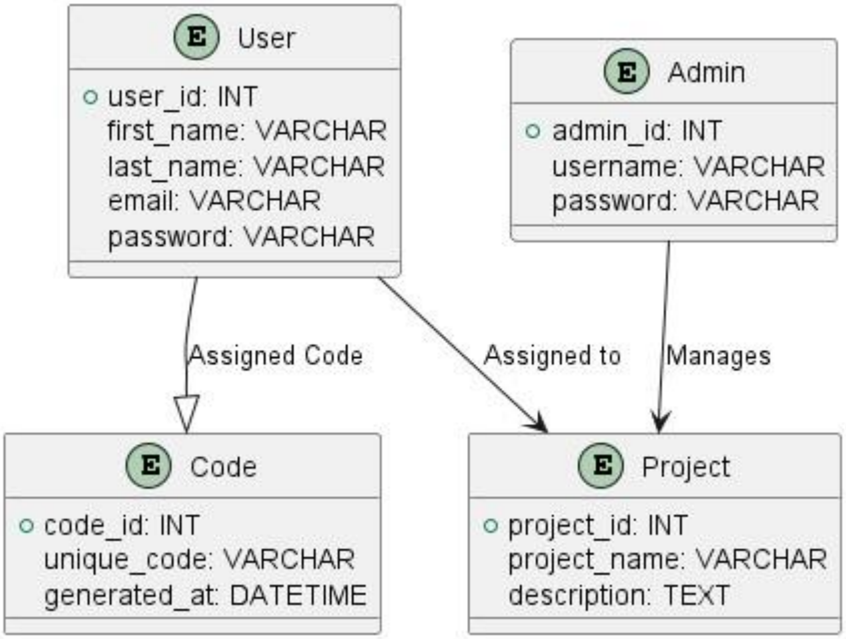


Figure 26: ERD Diagram of UMS

- **Class Diagram**

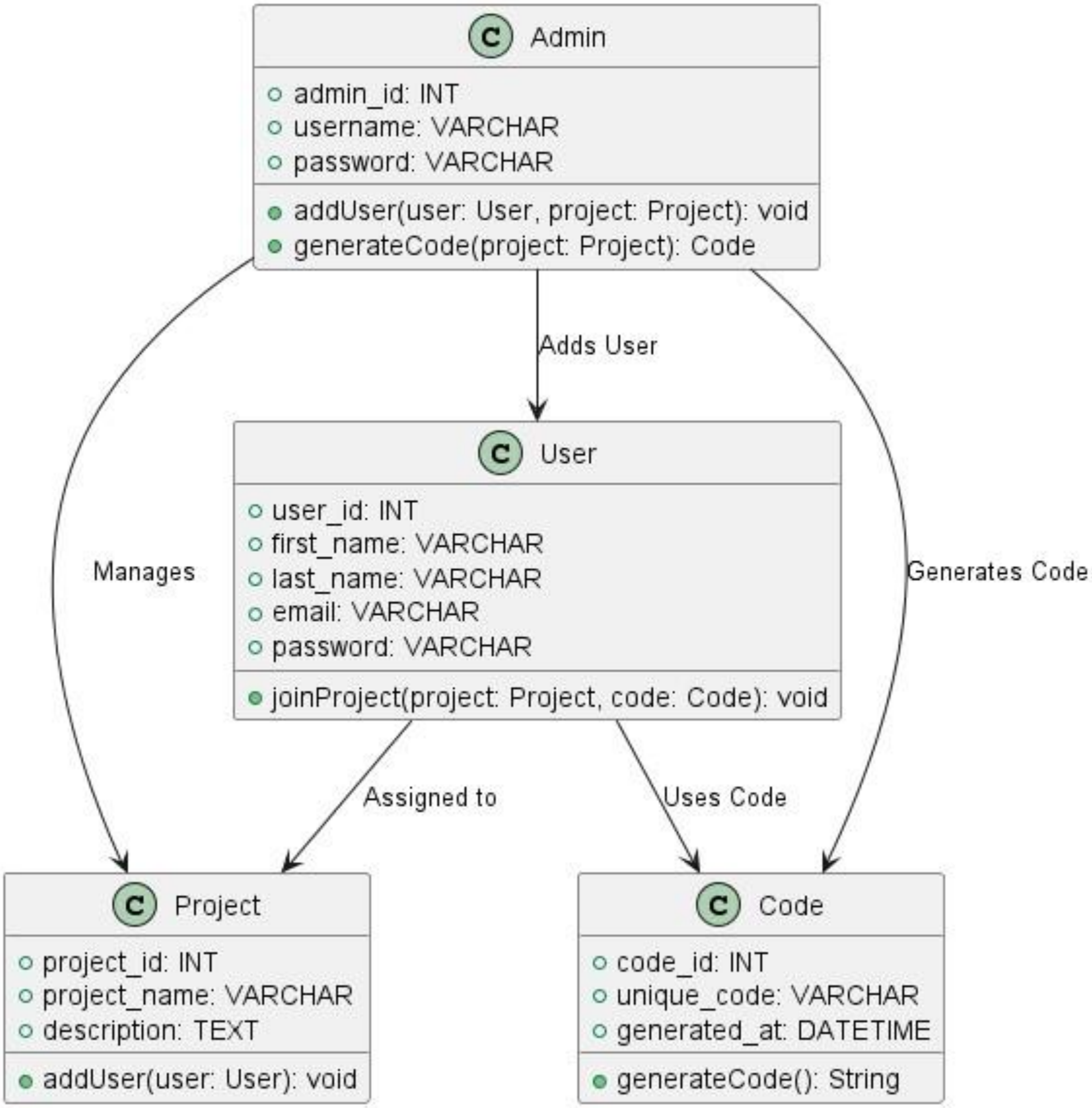


Figure 27: Class Diagram of UMS

- **Sequence Diagram**

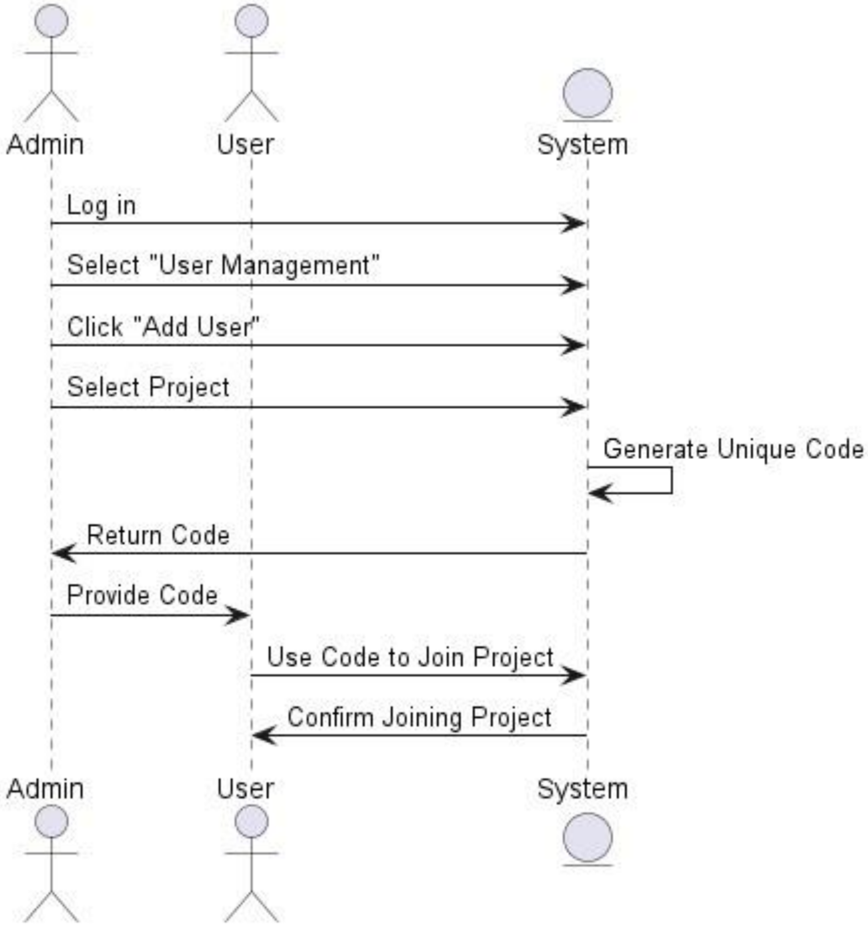
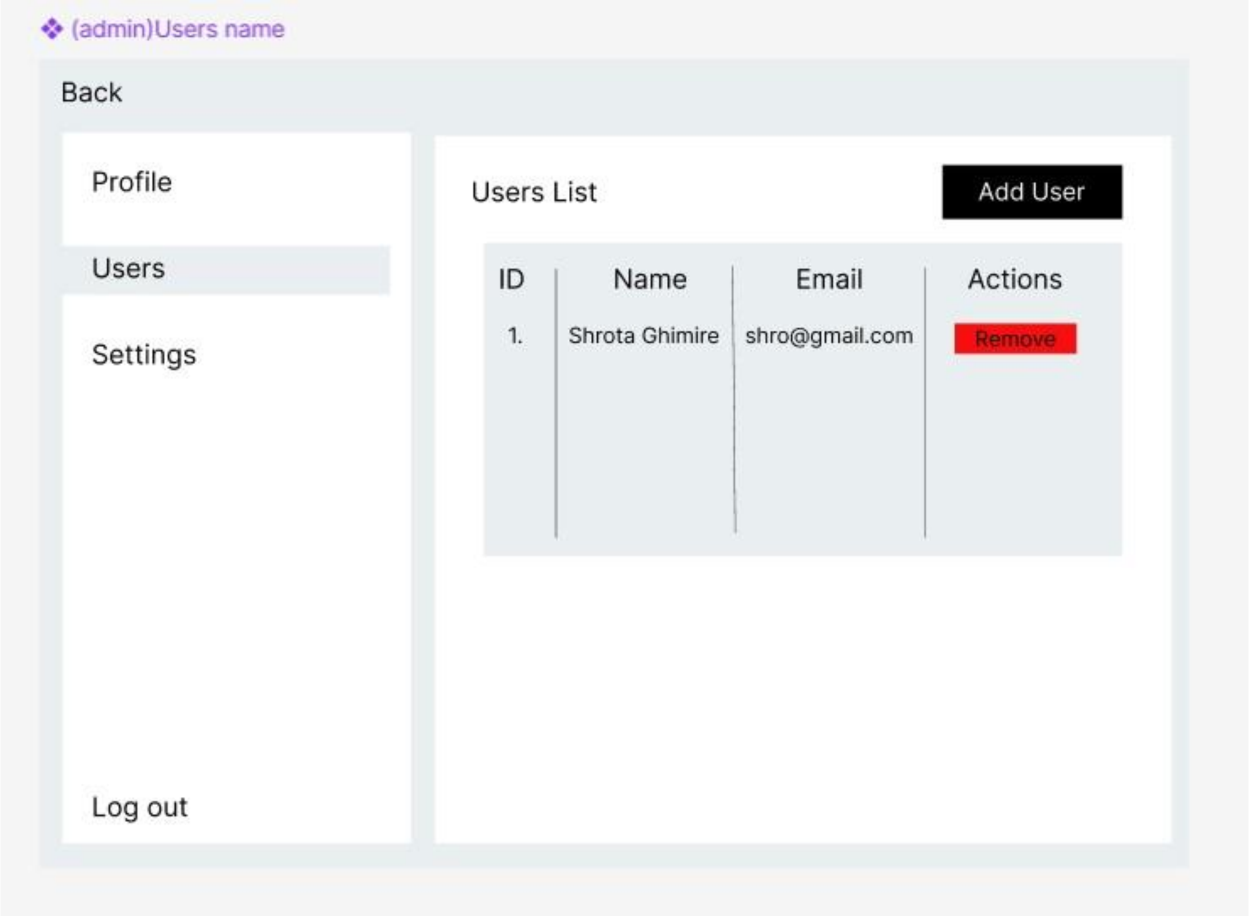


Figure 28: Sequence Diagram of UMS

- Wireframes

- **Adding and Removing User**



ID	Name	Email	Actions
1.	Shrota Ghimire	shro@gmail.com	Remove

Figure 29: Wireframe of UMS

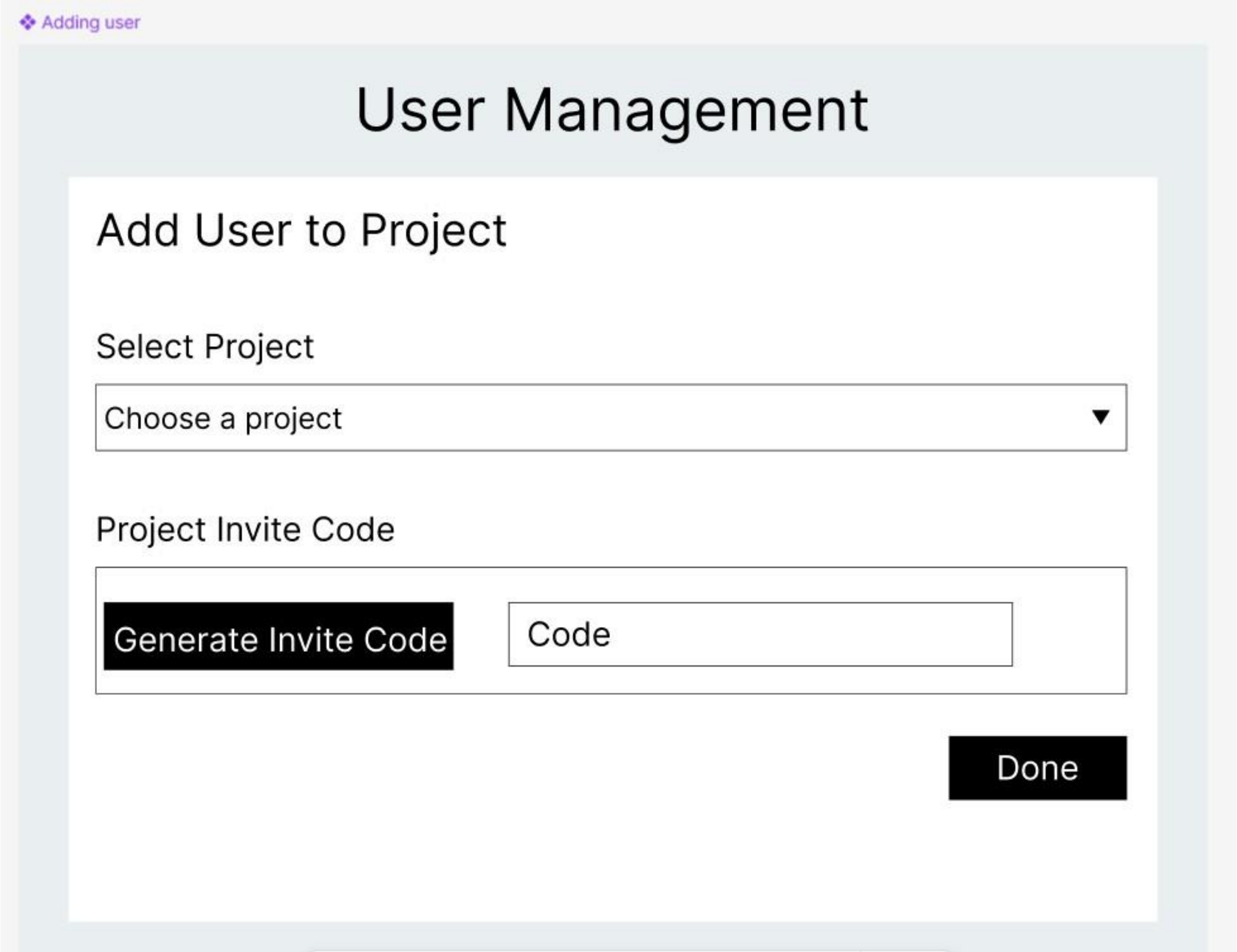


Figure 30: Wireframe of UMS

- Software Requirements Specification (SRS)

Request No.	Request Description	Use Case	Moscow Prioritization
UMS-FR-1	Admin must be able to create a unique code for every project.	Generate Code	Must have
UMS-NFR-2	The code must be unique, secure and difficult to guess.	Security	Must have

UMS-FR-3	Users must be able to join the project via code provided by admin.	Join Project	Must have
UMS-UR-4	When a user joins a project, it should throw a successful message.	Confirmation message	Must have
UMS-UR-5	When a user joins a project and the code is invalid it should throw an error message.	Error message	Must have

5.5.2 Critical Features and Implementation

- Key Functionalities

Feature 1: Add User

Through the system administrators obtain the capability to invite users to project through code generation. Staff can choose a project to code generation while the resultant code becomes available to platform users. The system enables users to join projects through a specified code which provides both controlled and secure entry.

Implementation:

- Use of the "Add User" action by an admin leads to the selection of a preferred project.
- The system creates a distinct alphanumeric identification code which belongs to the project.
- The user performs entry of the code to prove authorization and becomes part of the project database.
- The database system adds the user to the project user list as part of its automatic database process.

Feature 2: Remove User

Active administrators possess the ability to eliminate users from projects which leads to terminated access permissions. Project data protection happens through this system since it eliminates unauthorized users' access.

Implementation:

- The administrator selects a project before viewing a list that displays allocated users.
- The system includes a removal option for all users within its interface.
- The database gets modified with access revocation after the system gets confirmation for removal.

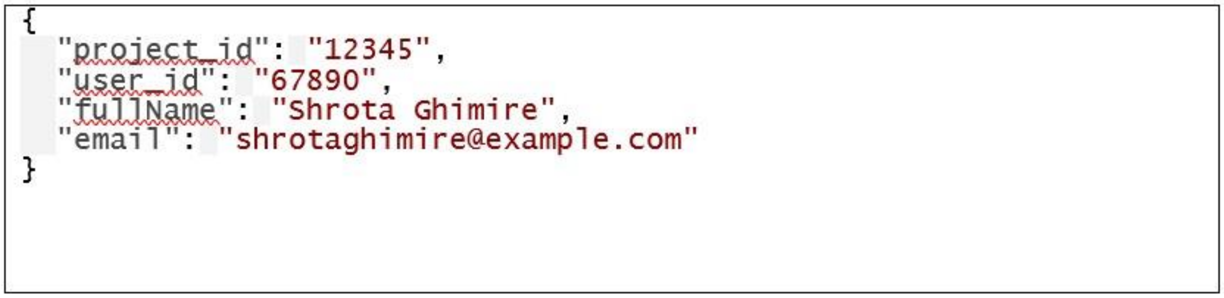
5.5.3 API and Database Implementation

- API Endpoints

Through its RESTful API system, the authentication structure handles both add user alongside remove operations. Below is a structured list of the key API endpoints:

• **Add User:**

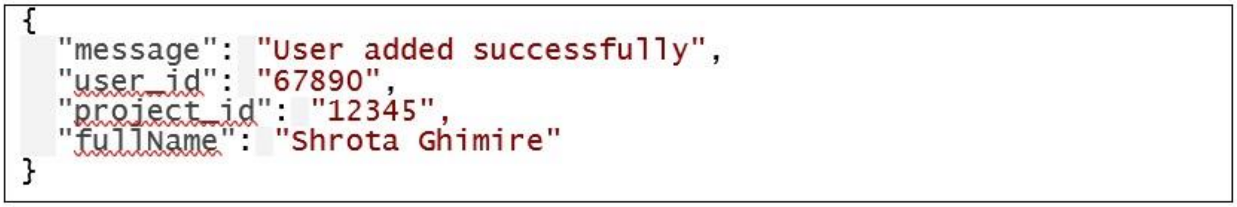
- Endpoint: POST /api/addUser - Description: Add users to the project - Request Body (JSON):



```
{
  "project_id": "12345",
  "user_id": "67890",
  "fullName": "Shrota Ghimire",
  "email": "shrotaghimire@example.com"
}
```

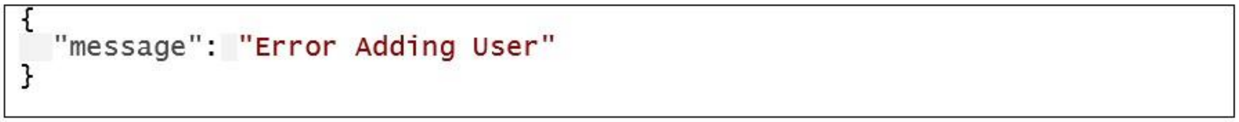
Figure 24: Add User - Request Body

- Response:



```
{
  "message": "User added successfully",
  "user_id": "67890",
  "project_id": "12345",
  "fullName": "Shrota Ghimire"
}
```

Figure 25: Success Message



```
{
  "message": "Error Adding User"
}
```

Figure 26: Error Message

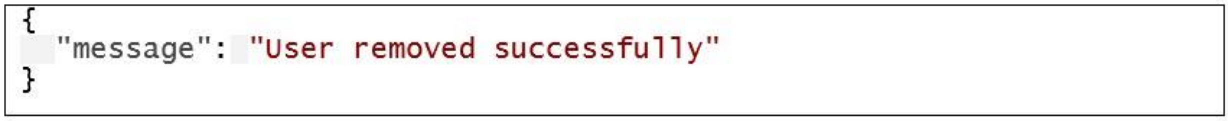
• Remove User:

- Endpoint: POST /api/removeUser
- Description: Removing users from the project
- Request Body (JSON):

```
{
  "project_id": "12345",
  "user_id": "67890"
}
```

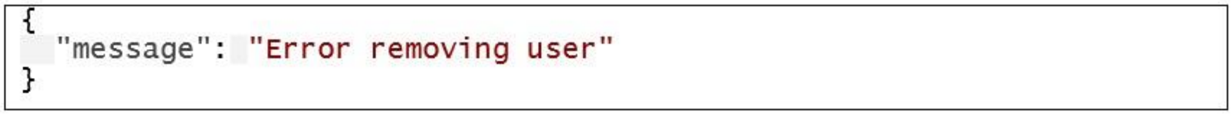
Figure 27: Remove User – Request Body

- Response:



```
{
  "message": "User removed successfully"
}
```

Figure 28: Success Message

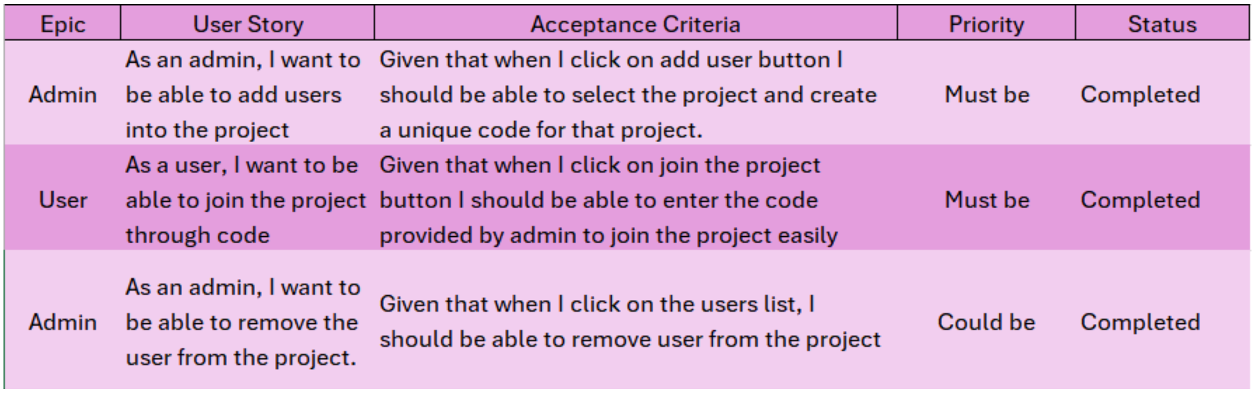


```
{
  "message": "Error removing user"
}
```

Figure 29: Error message

5.5.4 Development Process

- Sprint Backlog



Epic	User Story	Acceptance Criteria	Priority	Status
Admin	As an admin, I want to be able to add users into the project	Given that when I click on add user button I should be able to select the project and create a unique code for that project.	Must be	Completed
User	As a user, I want to be able to join the project through code	Given that when I click on join the project button I should be able to enter the code provided by admin to join the project easily	Must be	Completed
Admin	As an admin, I want to be able to remove the user from the project.	Given that when I click on the users list, I should be able to remove user from the project	Could be	Completed

Figure 31: Sprint Backlog of UMS

- Execution Details

I executed this work through Agile methodology to create and optimize separate elements in the system. The development approach implemented systematic workflow beginning with frontend layout construction which then integrated backend application programming interfaces.

- **Frontend Development**

Development begins with creating an attractive user interface system that lets users communicate with the system. A fast and modular development environment will be built from HTML and CSS and JS components at the front end. The frontend components need to integrate with form validation as well as error handling functions and API integration services. The developers will build application responsiveness before focusing on maintaining user experience naturalness after finishing the UI development stage.

- **Backend Development**

The Node.js backend development launches following completion of frontend development. The system will require APIs to perform adding user and removing through user management. The database architecture requires architectural patterns to keep operational information and credentials in the storage system. MySQL will be selected because performing data management on users demands dependable system performance.

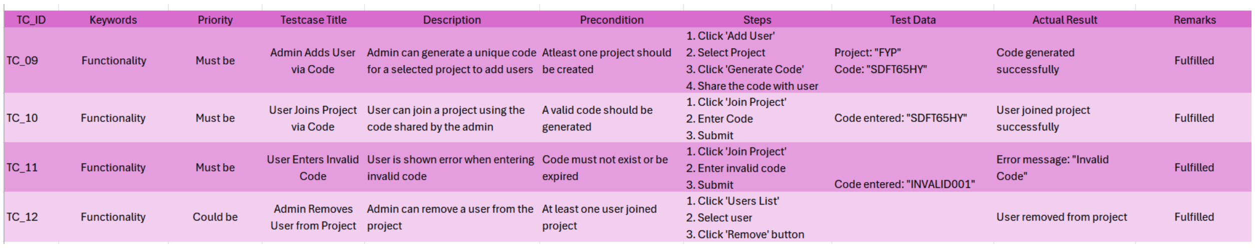
- **Integration**

The front-end design integration with backend APIs will commence after the independent development of both front-end and back-end components. The user management executes frontend requests which are transmitted to backend operations.

5.5.5 Testing Approach

Testing stands as an essential developmental component for verifying that every system component performs to specifications. Research-based testing extends across unit testing and integration testing and user testing for complete system examination.

- **Black Box Testing**



TC_ID	Keywords	Priority	Testcase Title	Description	Precondition	Steps	Test Data	Actual Result	Remarks
TC_09	Functionality	Must be	Admin Adds User via Code	Admin can generate a unique code for a selected project to add users	Atleast one project should be created	1. Click 'Add User' 2. Select Project 3. Click 'Generate Code' 4. Share the code with user	Project: "FYP" Code: "SDFT65HY"	Code generated successfully	Fulfilled
TC_10	Functionality	Must be	User Joins Project via Code	User can join a project using the code shared by the admin	A valid code should be generated	1. Click 'Join Project' 2. Enter Code 3. Submit	Code entered: "SDFT65HY"	User joined project successfully	Fulfilled
TC_11	Functionality	Must be	User Enters Invalid Code	User is shown error when entering invalid code	Code must not exist or be expired	1. Click 'Join Project' 2. Enter invalid code 3. Submit	Code entered: "INVALID001"	Error message: "Invalid Code"	Fulfilled
TC_12	Functionality	Could be	Admin Removes User from Project	Admin can remove a user from the project	At least one user joined project	1. Click 'Users List' 2. Select user 3. Click 'Remove' button		User removed from project	Fulfilled

Figure 32: Test case of UMS

5.6 Subsystem 4 – Task Management System

This document presents the designed structure for the Task Management Subsystem within the final year project framework. The subsystem provides tracked secure management of task assignments to support systematic execution within the Task Management System.

The subsystem enables administrators to follow procedure-based tasks distribution that maintains both clarity and distribution accountability. The subsystem will handle the entire process of task generation and assignment and performance monitoring to optimize project workflow and organization. Admins can select their projects through the system before they create tasks that get distributed to users for smooth collaborative tracking across every component of the platform.

5.6.1 System Architecture and Design

- Activity Diagram

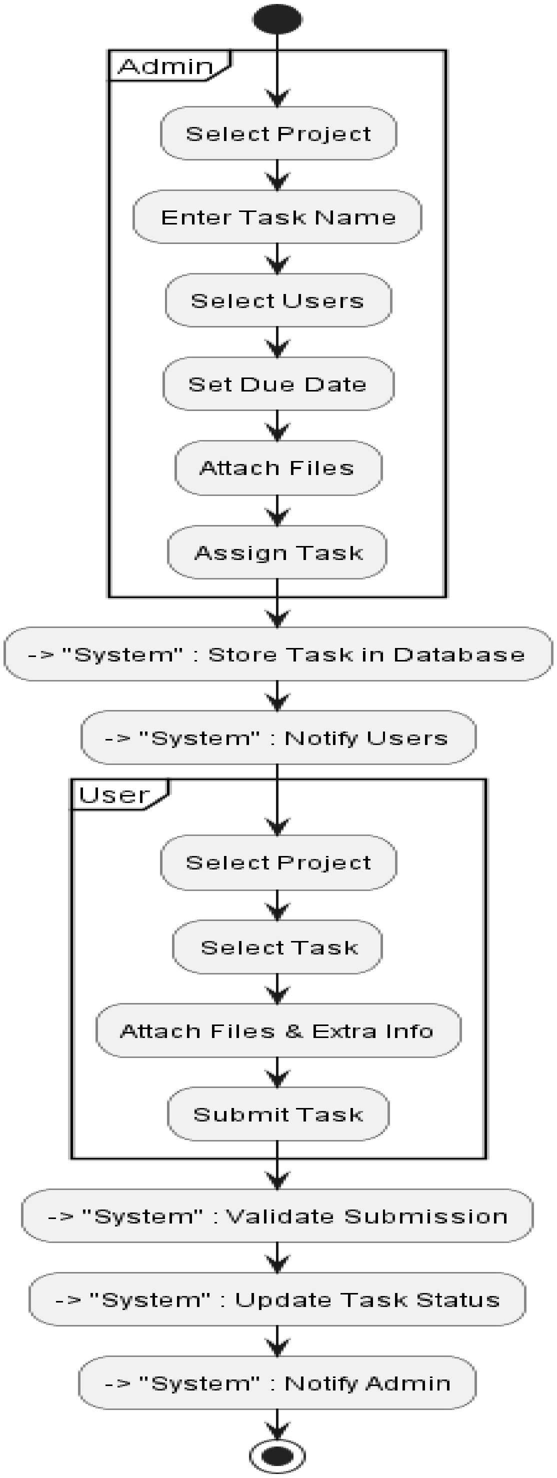

Admin
Select Project
Enter Task Name
Select Users
Set Due Date
Attach Files
Assign Task
-> "System" : Store Task in Database
-> "System" : Notify Users
User
Select Project
Select Task
Attach Files & Extra Info
Submit Task
-> "System" : Validate Submission
-> "System" : Update Task Status
-> "System" : Notify Admin

Figure 33: Activity Diagram of TMS

- Use Case Diagram

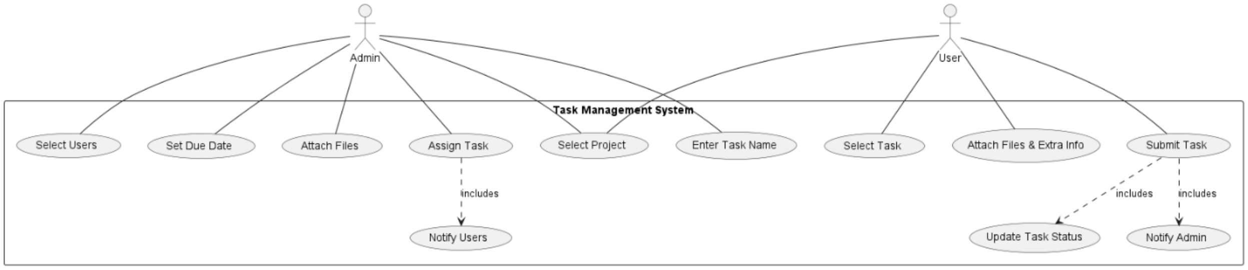


Figure 34: Use Case Diagram of TMS

- Entity Relationship Diagram (ERD)

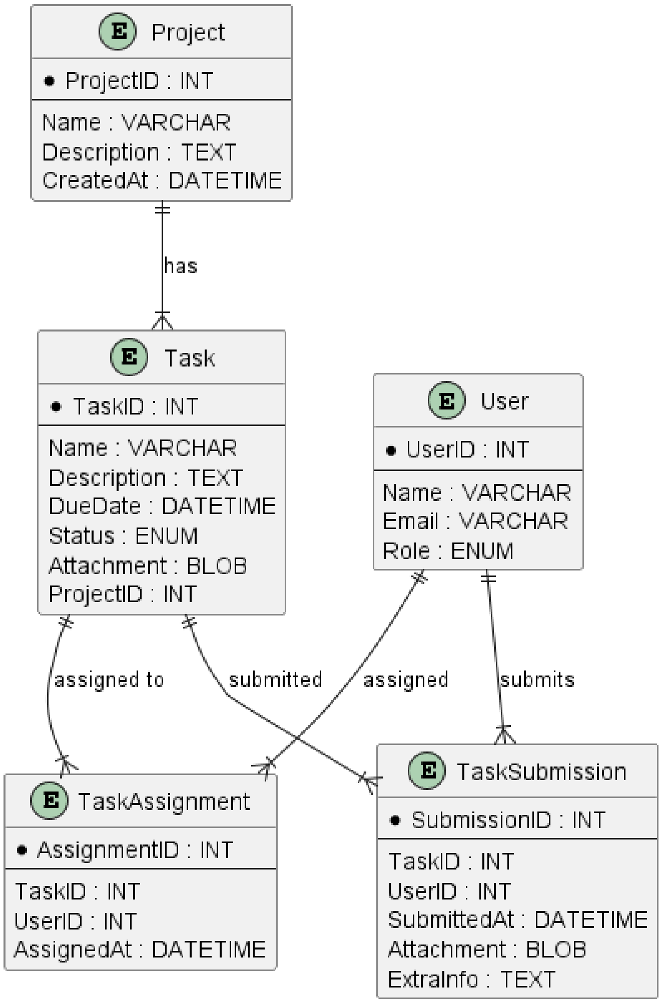


Figure 35: ER Diagram of TMS

- Class Diagram

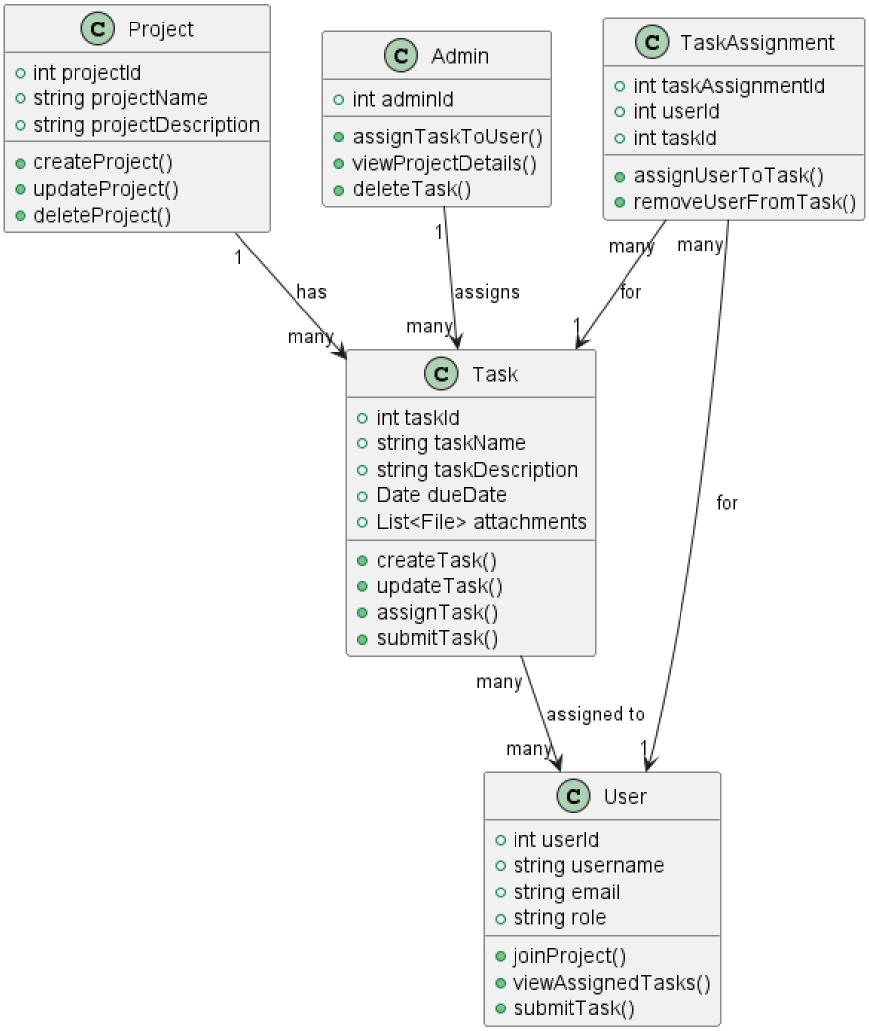


Figure 36: Class Diagram of TMS

- Sequence diagram

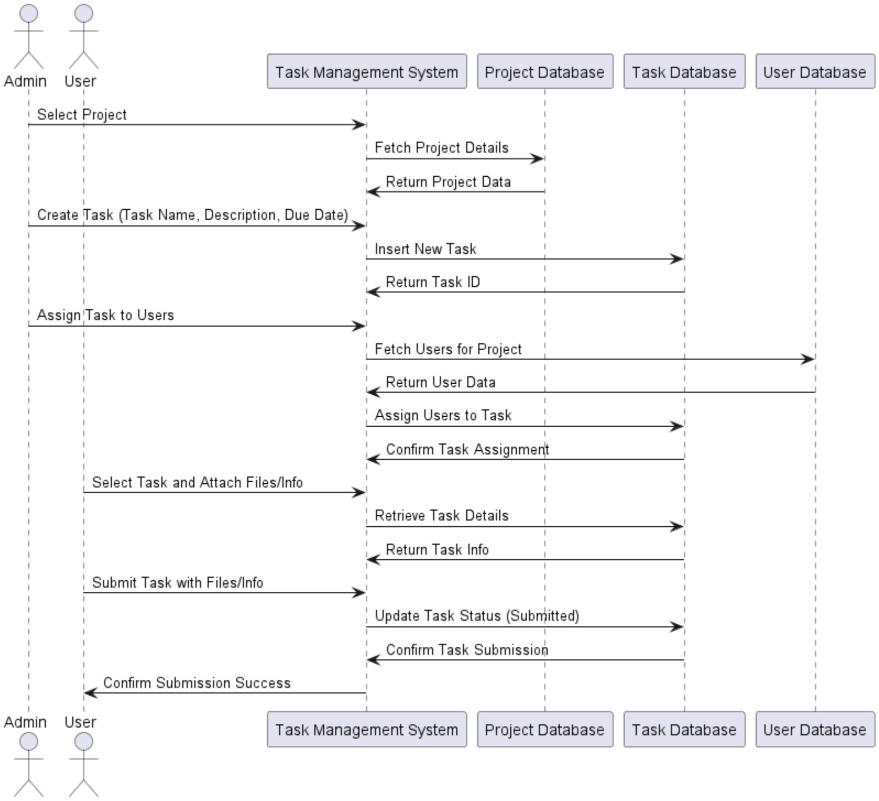


Figure 37: Sequence Diagram of TMS

- Wireframes

- **Assign Tasks**

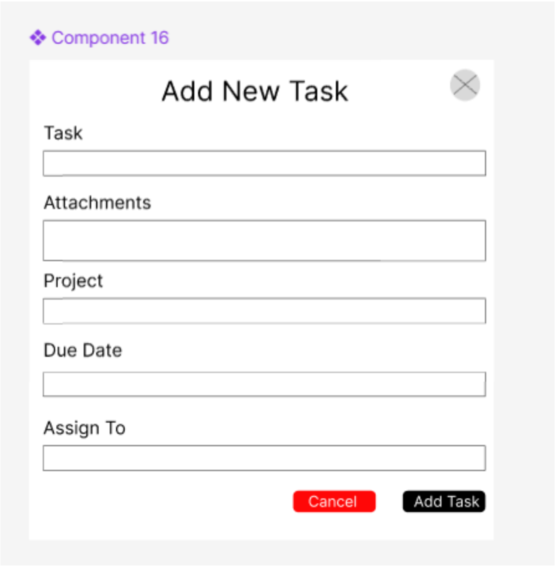


Figure 38: Wireframe of TMS

- **Submit Task**

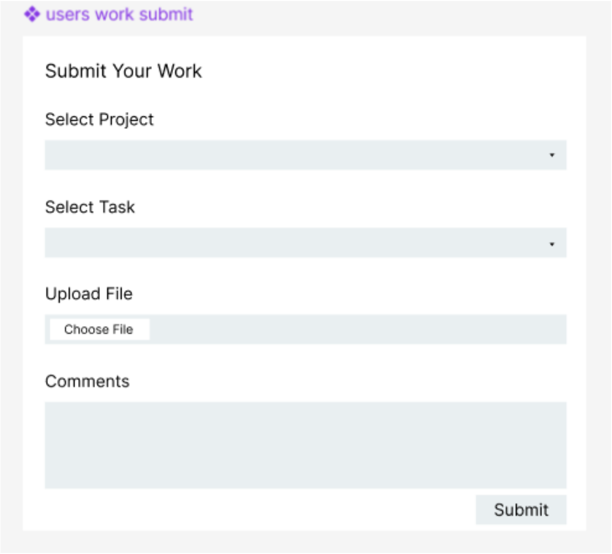


Figure 39: Wireframe of TMS

- **Task Status Board**

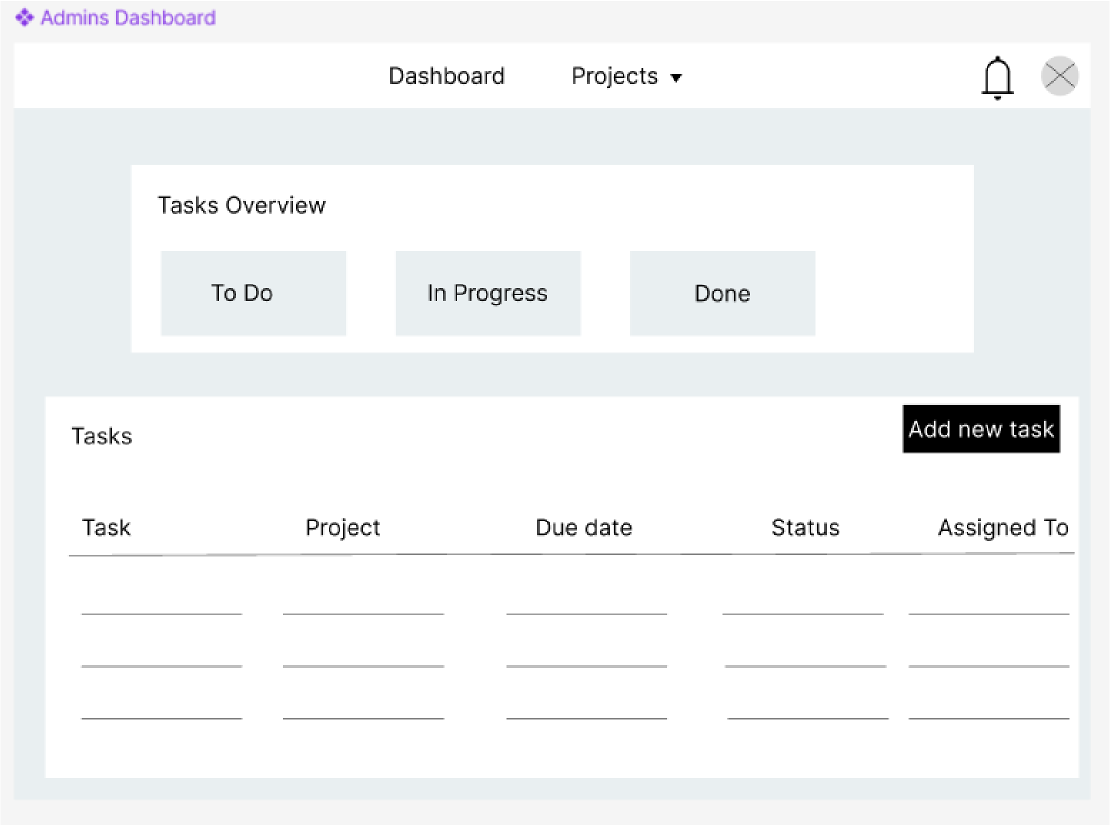


Figure 40: Wireframe of TMS

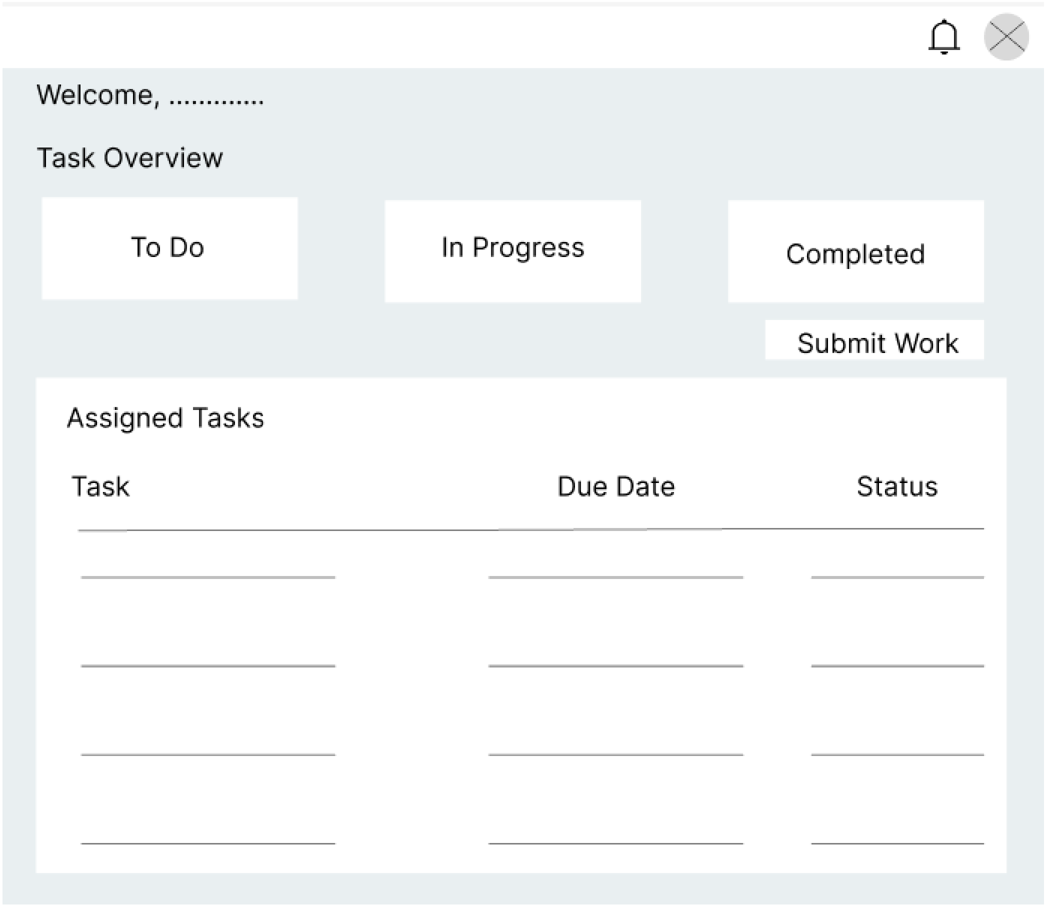


Figure 41: Wireframe of TMS

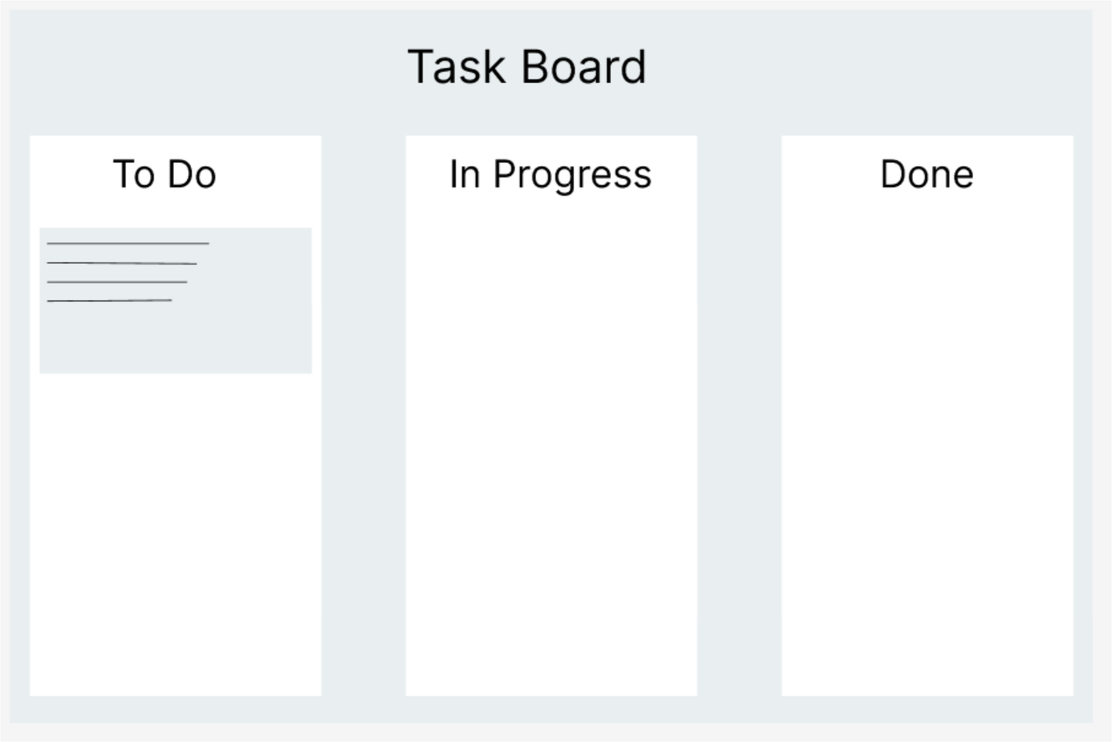


Figure 42: Wireframe of TMS

- **Software Requirements Specifications (SRS)**

Request No.	Request Description	Use Case	Moscow Prioritization
TMS-FR-1	Admin must be able to assign a task to every user who has joined the project.	Assign task	Must have
TMS-UR-2	When assigning a task, admin must be able to mention title, select project, attach file, set deadline and select user in a task assigning form.	Filling up form	Must have
TMS-UR-3	Users must be able to view the tasks assigned to them in the tasks section.	View tasks	Must have

TMS-FR-4	Users must be able to submit a task.	Submit task	Must have
TMS-UR-5	When submitting a task, the user must be able to fill in the form which contains select project, select task, file attachments and description box.	Filling up form	Must have
TMS-FR-6	When admin assigns a task, it should be shown on a task board of both admin and user.	Tasks on task board	Must have
TMS-FR-7	Both admin and user must be able to update the task board status (To Do, In Progress, Done).	Update task status	Must have
TMS-FR-8	Admin must be able to delete the assigned task.	Delete task	Must have
TMS-UR-9	After deleting it, it should show a confirmation message.	Confirmation message	Could have

5.6.2 Critical Features and Implementation

- Key Functionalities

- **Feature 1: Assigning Task to User**

Through the Assigning Task to User functionality the administrator allocates jobs to team members working within selected projects, so they receive appropriate work assignments. Project management efficiency heavily depends on this system because it enables proper task distribution across users.

Implementation:

- Through the dedicated interface of the Task Management System the admin performs task assignments. Using this interface the administrator can choose from a collection of accessible projects. The administrator selects a project then the system displays an accessible list of pending tasks that will be allocated to users.

- Through the selected project interface, the admin has the power to make new tasks as well as select from existing tasks. After selecting the task, the admin can enter a task name together with a description and due date followed by entering additional information such as priority levels and attachments which users will see when they get assigned tasks.
- The administrator can choose users belonging to the project team following the selection of a task to receive assignment. Multiple users can get assigned to work on the same task due to the flexible user selection method present in the system.
- Through the system users together with admin personnel have access to monitor current task statuses. The system enables administrators to check the statuses of tasks included in their assignments based on pending, current work, or completed conditions. Tasks accessible through the system provide users with task information and both progress tracking functionality and completion report capabilities.
- The system enables administrators to make updates to assigned tasks when alterations happen in project structure or due dates and priority requirements. The application shows instant updates on user task lists to maintain project alignment between team members and project goals and time expectations.
- Through the designated access system, authorized admins alone can perform task assignments. The system allows users to view only their assigned tasks while providing them necessary access to work on those responsibilities.

- **Feature 2: Submit Task**

Users can effectively track project work progress through the Submitting Task by using a User feature that enables them to submit completed tasks within project structures. Project task tracking relies on this functionality for completion verification because users and administrative staff need to monitor progress to meet project deadlines.

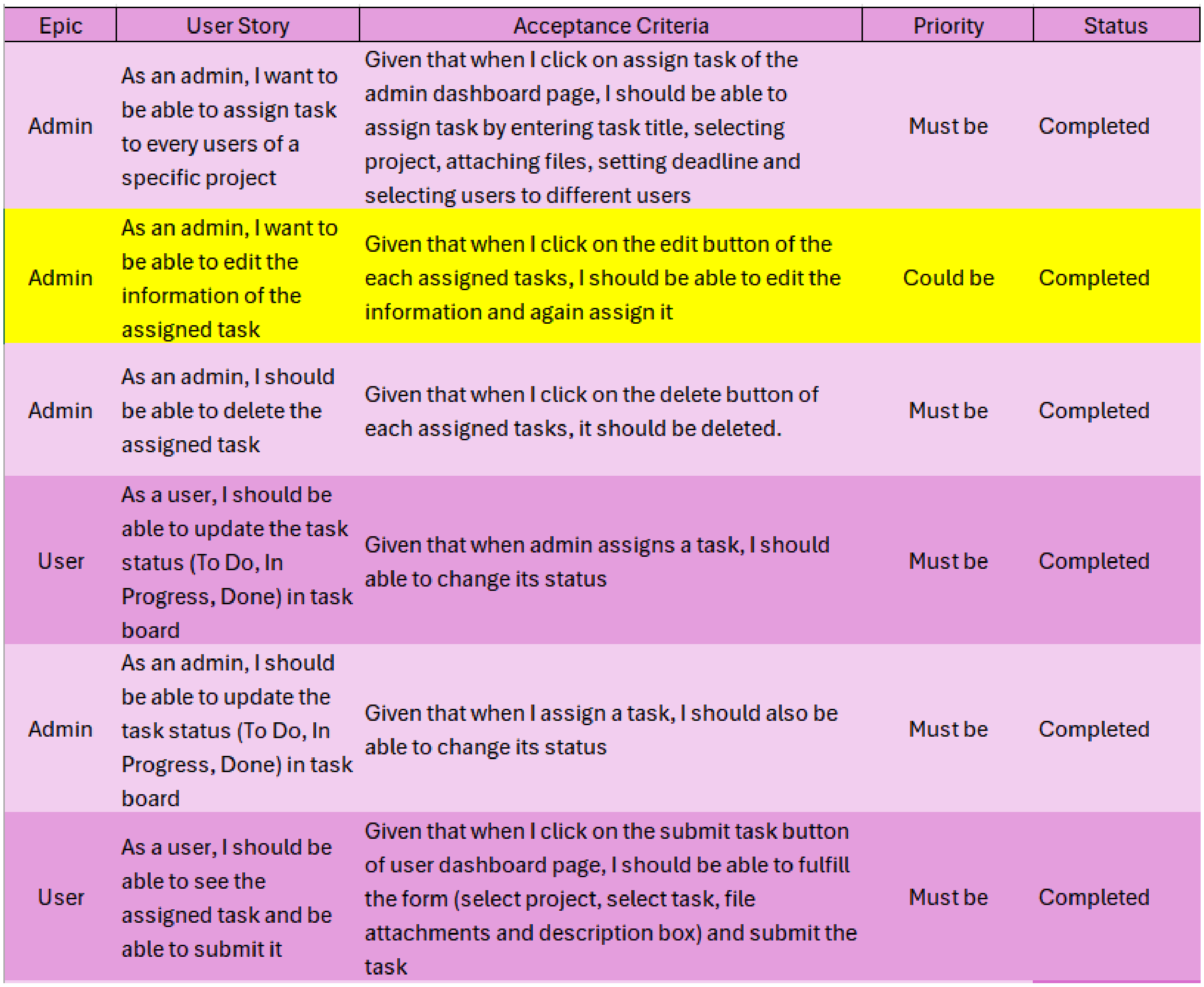
Implementation:

- The User Interface enables users to navigate their tasks using an intuitive dashboard in the Task Management System. A dashboard displays tasks to users who see the task name along with description and due date information and additional relevant details including attachments or task-related files.

- The system enables users to complete their work by enabling them to mark tasks as finished through the application. Users will locate the “Submit Task” button in the interface to trigger the submission process after they complete their assignments. The system allows task submission exclusively for users who received assigned tasks to prevent any unauthorized submissions.
- Users maintain the ability to attach necessary files during their task submission process when they use the system's attachment feature while adding additional information. An upload attachment feature becomes available to users prior to finishing their submission through the system. Users possess the capability to enter supplementary information through the task submission form that enhances the understanding of their work.
- Users gain access to track the status of all their submitted tasks within the system. The system moves tasks to the “Done” section in the user dashboard when the admin approves or marks them as finished work.

5.6.3 Development Process

- Sprint Backlog



Epic	User Story	Acceptance Criteria	Priority	Status
Admin	As an admin, I want to be able to assign task to every users of a specific project	Given that when I click on assign task of the admin dashboard page, I should be able to assign task by entering task title, selecting project, attaching files, setting deadline and selecting users to different users	Must be	Completed
Admin	As an admin, I want to be able to edit the information of the assigned task	Given that when I click on the edit button of the each assigned tasks, I should be able to edit the information and again assign it	Could be	Completed
Admin	As an admin, I should be able to delete the assigned task	Given that when I click on the delete button of each assigned tasks, it should be deleted.	Must be	Completed
User	As a user, I should be able to update the task status (To Do, In Progress, Done) in task board	Given that when admin assigns a task, I should able to change its status	Must be	Completed
Admin	As an admin, I should be able to update the task status (To Do, In Progress, Done) in task board	Given that when I assign a task, I should also be able to change its status	Must be	Completed
User	As a user, I should be able to see the assigned task and be able to submit it	Given that when I click on the submit task button of user dashboard page, I should be able to fulfill the form (select project, select task, file attachments and description box) and submit the task	Must be	Completed

Figure 43: Sprint Backlog of TMS

- Execution Details

I executed this work through Agile methodology to create and optimize separate elements in the system. The development approach implemented systematic workflow beginning with frontend layout construction which then integrated backend application programming interfaces.

• Frontend Development

Development begins with creating an attractive user interface system that lets users communicate with the system. A fast and modular development environment will be built from HTML and CSS and JS components at the front end. The frontend components need to integrate with form validation as well as error handling functions and API integration services. The developers will build application responsiveness before focusing on maintaining user experience naturalness after finishing the UI development stage.

- **Backend Development**

The Node.js backend development launches following completion of frontend development. The system will require APIs to perform adding user and removing through user management. The database architecture requires architectural patterns to keep operational information and credentials in the storage system. MySQL will be selected because performing data management on users demand dependable system performance.

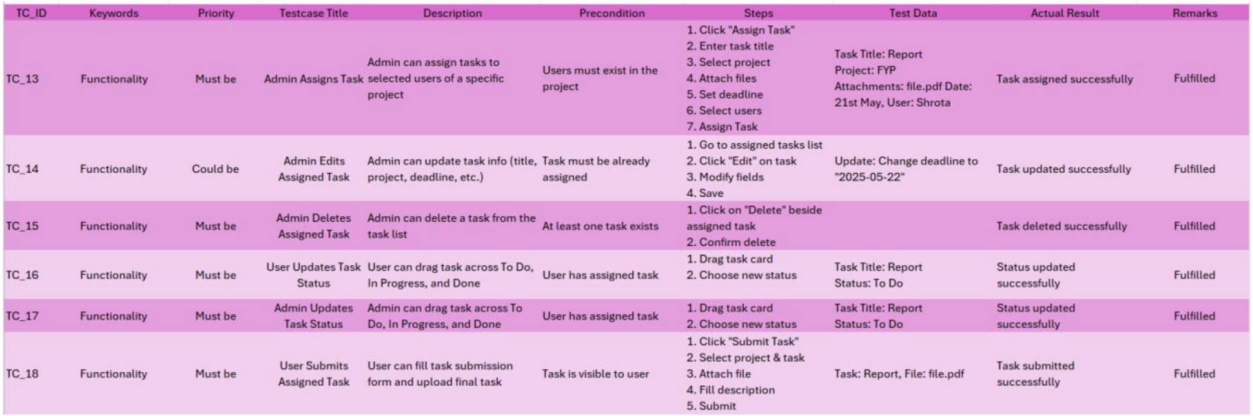
- **Integration**

The front-end design integration with backend APIs will commence after the independent development of both front-end and back-end components. The user management executes frontend requests which are transmitted to backend operations.

5.6.4 Testing Approach

Testing stands as an essential developmental component for verifying that all system components perform to specifications. Research-based testing extends across unit testing and integration testing and user testing for complete system examination.

- Black Box Testing



TC_ID	Keywords	Priority	Testcase Title	Description	Precondition	Steps	Test Data	Actual Result	Remarks
TC_13	Functionality	Must be	Admin Assigns Task	Admin can assign tasks to selected users of a specific project	Users must exist in the project	1. Click "Assign Task" 2. Enter task title 3. Select project 4. Attach files 5. Set deadline 6. Select users 7. Assign Task	Task Title: Report Project: FYP Attachments: file.pdf Date: 21st May, User: Shrota	Task assigned successfully	Fulfilled
TC_14	Functionality	Could be	Admin Edits Assigned Task	Admin can update task info (title, project, deadline, etc.)	Task must be already assigned	1. Go to assigned tasks list 2. Click "Edit" on task 3. Modify fields 4. Save	Update: Change deadline to "2025-05-22"	Task updated successfully	Fulfilled
TC_15	Functionality	Must be	Admin Deletes Assigned Task	Admin can delete a task from the task list	At least one task exists	1. Click on "Delete" beside assigned task 2. Confirm delete		Task deleted successfully	Fulfilled
TC_16	Functionality	Must be	User Updates Task Status	User can drag task across To Do, In Progress, and Done	User has assigned task	1. Drag task card 2. Choose new status	Task Title: Report Status: To Do	Status updated successfully	Fulfilled
TC_17	Functionality	Must be	Admin Updates Task Status	Admin can drag task across To Do, In Progress, and Done	User has assigned task	1. Drag task card 2. Choose new status	Task Title: Report Status: To Do	Status updated successfully	Fulfilled
TC_18	Functionality	Must be	User Submits Assigned Task	User can fill task submission form and upload final task	Task is visible to user	1. Click "Submit Task" 2. Select project & task 3. Attach file 4. Fill description 5. Submit	Task: Report, File: file.pdf	Task submitted successfully	Fulfilled

Figure 44: Test Case of TMS

5.7 Subsystem 5 – Notification Management

The Notification Management Subsystem artifact design serves as part of the final year project according to this document. Through the Notification Management subsystem users and admins achieve efficient communication through real-time updates about project and task activities that exist in the Task Management System.

This subsystem delivers automatic alerts about major system occurrences to maintain constant awareness among all users and administrators. The admin receives an alert whenever user adds to join the project. The system sends notification alerts to users about their assigned tasks through the assigned details provided by the admin. After a user finishes their task assignment the system automatically informs the administrator for review work. The system maintains a real-time track of task progress that generates alerts which specify To-Do, In Progress and Completed states. Effective notification management from this subsystem improves both project collaboration and enhances user accountability that leads to successful platform execution.

5.7.1 System Architecture and Design

- Activity Diagram

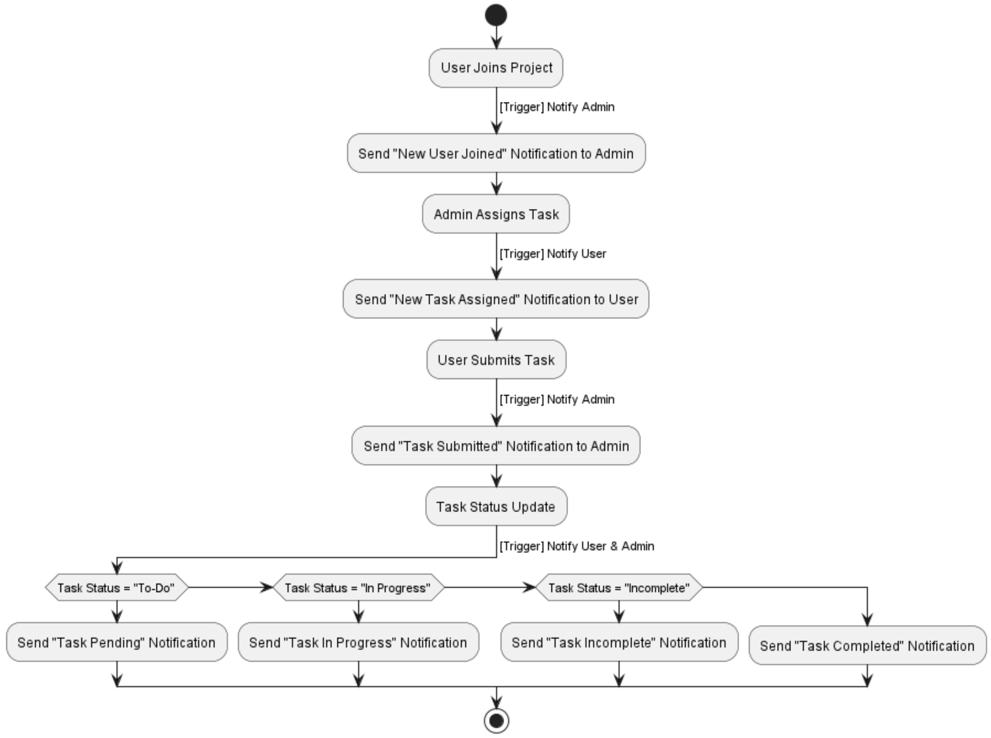

User Joins Project
[Trigger] Notify Admin
Send "New User Joined" Notification to Admin
Admin Assigns Task
[Trigger] Notify User
Send "New Task Assigned" Notification to User
User Submits Task
[Trigger] Notify Admin
Send "Task Submitted" Notification to Admin
Task Status Update
[Trigger] Notify User & Admin
Task Status = "To-Do"
Task Status = "In Progress"
Task Status = "Incomplete"
Send "Task Pending" Notification
Send "Task In Progress" Notification
Send "Task Incomplete" Notification
Send "Task Completed" Notification

Figure 45: Activity Diagram of NMS

- Use Case Diagram

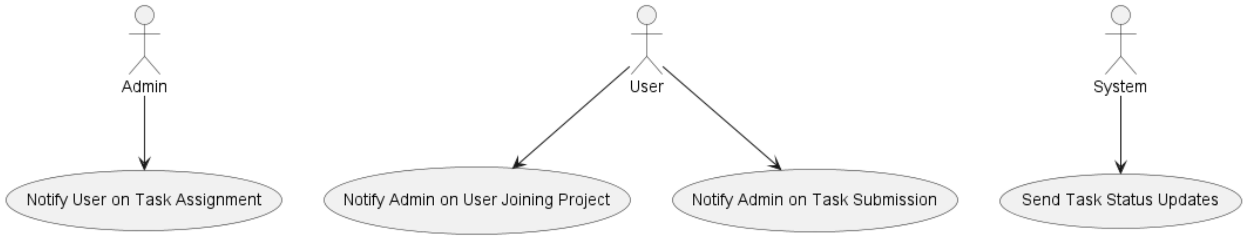

Admin
User
System
Notify User on Task Assignment
Notify Admin on User Joining Project
Notify Admin on Task Submission
Send Task Status Updates

Figure 46: Use Case Diagram of NMS

- Entity Relationship Diagram

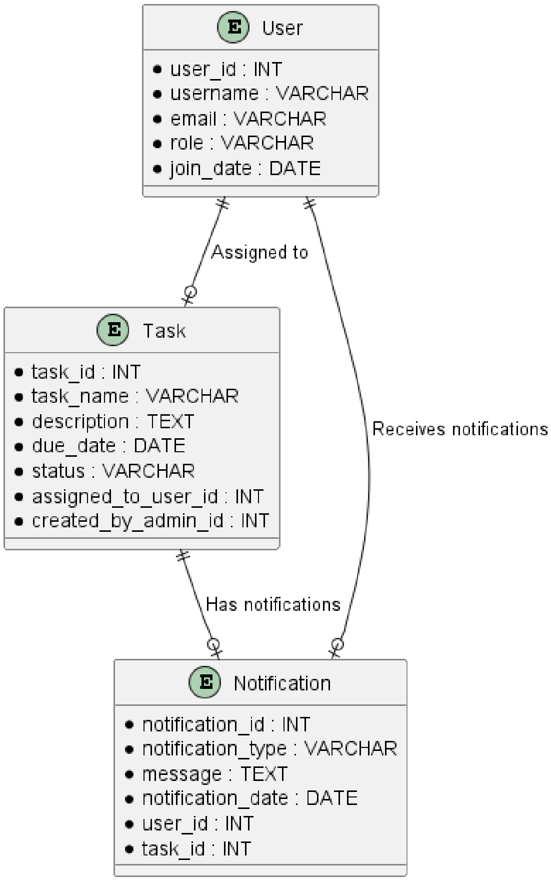


Figure 47: ER Diagram of NMS

- Class Diagram

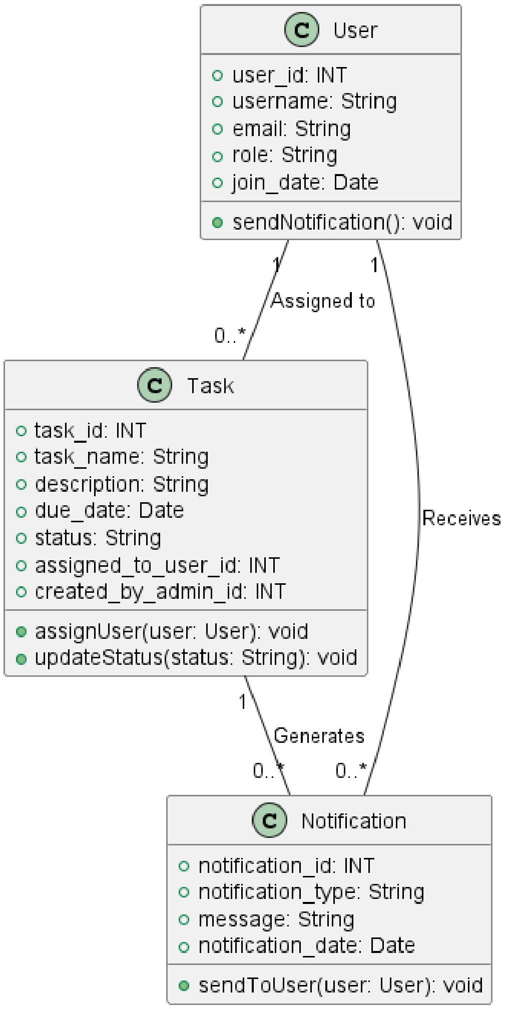


Figure 48: Class Diagram of NMS

- Sequence Diagram

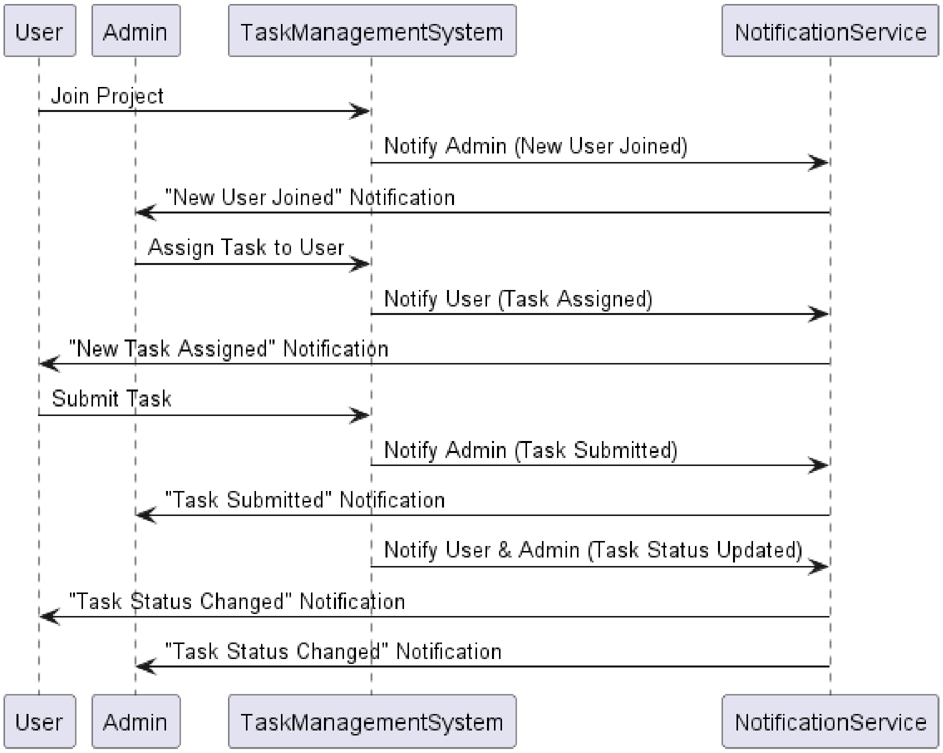
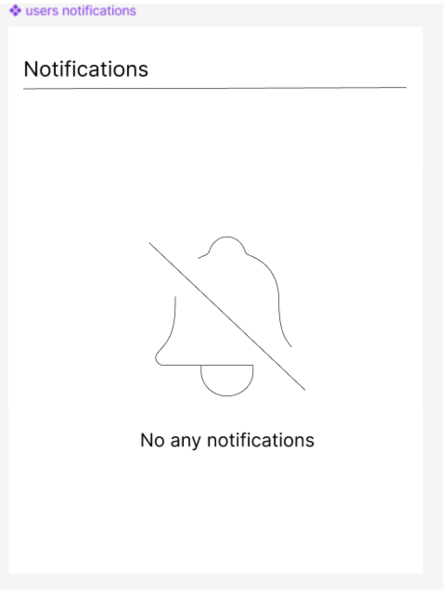

User
Admin
TaskManagementSystem
NotificationService
Join Project
Notify Admin (New User Joined)
"New User Joined" Notification
Assign Task to User
Notify User (Task Assigned)
"New Task Assigned" Notification
Submit Task
Notify Admin (Task Submitted)
"Task Submitted" Notification
Notify User & Admin (Task Status Updated)
"Task Status Changed" Notification
"Task Status Changed" Notification
User
Admin
TaskManagementSystem
NotificationService

Figure 49: Sequence Diagram of NMS

- Wireframes

- **Users Notification**

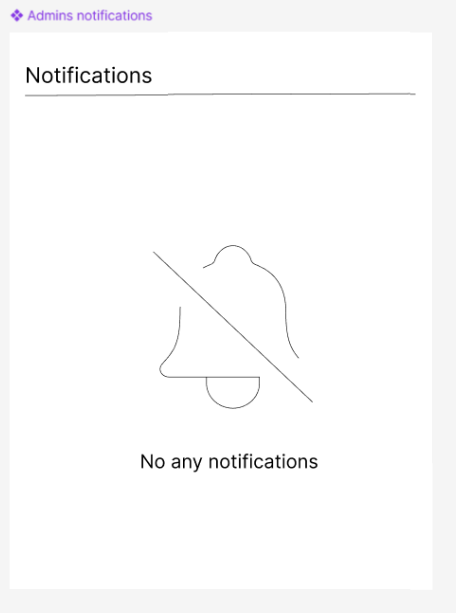


Notifications

No any notifications

Figure 50: Wireframe of NMS

- **Admins Notification**



Admins notifications

Notifications

No any notifications

Figure 51: Wireframe of NMS

- **Software Requirements Specifications (SRS)**

Request No.	Request Description	Use Case	Moscow Prioritization
NFS-FR-1	When a user joins a project via code, admin must be notified.	Notify joining project	Must have
NFS-FR-2	When an admin assigns a task to a user, the user must be notified.	Notify assigned task	Must have
NFS-FR-3	When an admin or a user changes the task status (To Do, In Progress, Done), admin and the user must be notified.	Notify task status update	Must have
NFS-FR-4	When a user submits their task, the admin must be notified.	Notify submitted task	Could have
NFS-UR-5	When a new notification arrives, it should update the badge count in the notification bar.	Update the badge count	Must have

NFS-UR-6	When the deadline arrives, the user must be notified.	Notify the deadline	Could have

5.7.2 Critical Features and Implementation

- Key Functionalities

Feature 1: Notification System

Inside the Notification System users alongside administrators receive immediate time-based event updates which enhances the communication features in Task Management System.

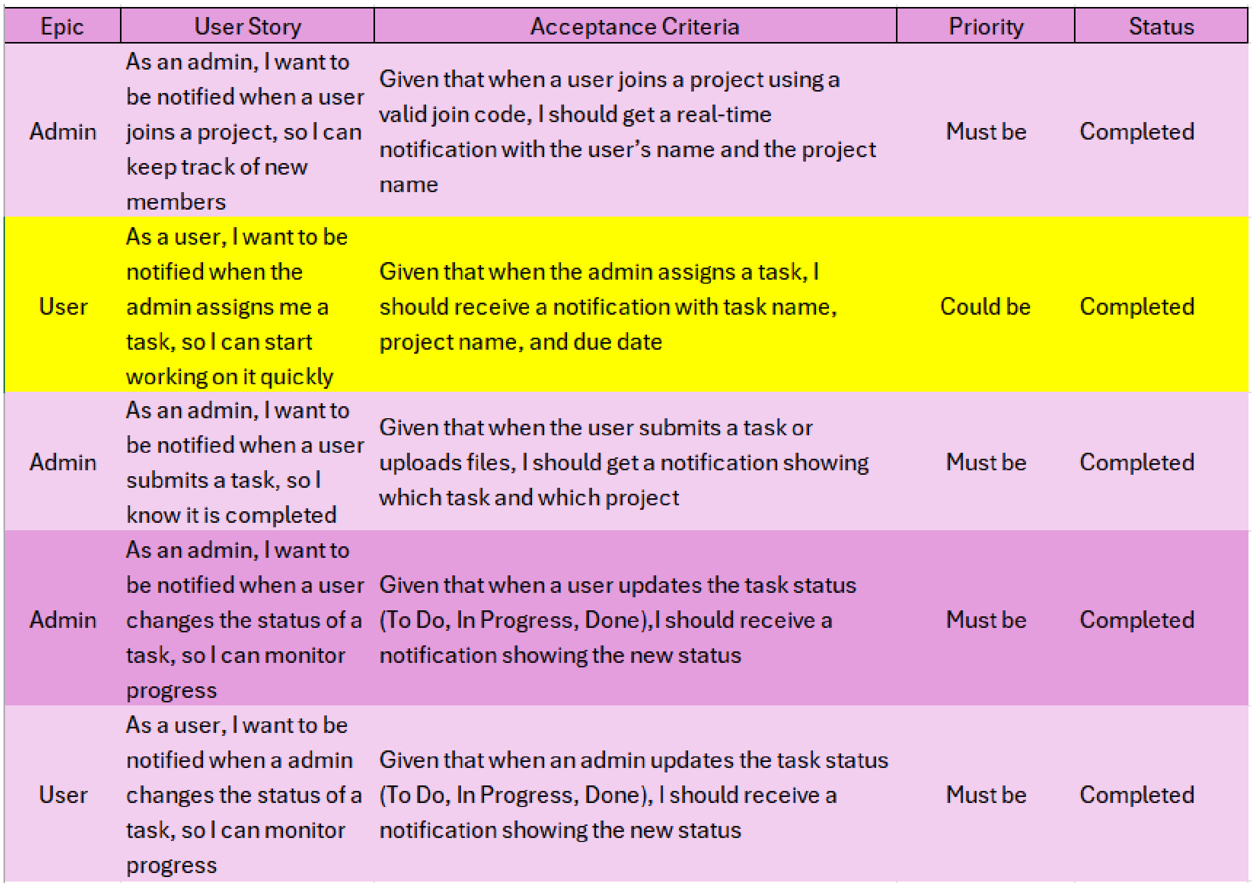
Implementation:

- The system creates automatic alerts for important events that take place throughout the system.
- Upon joining a project, the system automatically notifies the administrator.
- The system notification sends the designated task to both user and administrator.
- The user task submission triggers an automatic notification to reach the administrator.
- All system events that trigger task changes from To-Do to In Progress and to Completed generate instant alerts for both admins and users.
- The system operates with an event-driven management system which delivers immediate notifications after actions happen.
- The application maintains notification data in database tables which provide users with record of all message exchanges.
- Users can choose from real-time alerts and dashboard messages and email notifications which enable them to access updates according to their preference.
- The Notification Service class functions to process and distribute notifications whenever events take place.
- The user dashboard uses API integration to display notifications which results in a seamless interface.

Stakeholder project information is maintained through this deployment which increases the efficiency of task management and workflow processes in the Task Management System.

5.7.3 Development Process

- Sprint Backlog



Epic	User Story	Acceptance Criteria	Priority	Status
Admin	As an admin, I want to be notified when a user joins a project, so I can keep track of new members	Given that when a user joins a project using a valid join code, I should get a real-time notification with the user's name and the project name	Must be	Completed
User	As a user, I want to be notified when the admin assigns me a task, so I can start working on it quickly	Given that when the admin assigns a task, I should receive a notification with task name, project name, and due date	Could be	Completed
Admin	As an admin, I want to be notified when a user submits a task, so I know it is completed	Given that when the user submits a task or uploads files, I should get a notification showing which task and which project	Must be	Completed
Admin	As an admin, I want to be notified when a user changes the status of a task, so I can monitor progress	Given that when a user updates the task status (To Do, In Progress, Done),I should receive a notification showing the new status	Must be	Completed
User	As a user, I want to be notified when a admin changes the status of a task, so I can monitor progress	Given that when an admin updates the task status (To Do, In Progress, Done), I should receive a notification showing the new status	Must be	Completed

Figure 52: Sprint Backlog of NMS

- Execution Details

I executed this work through Agile methodology to create and optimize separate elements in the system. The development approach implemented systematic workflow beginning with frontend layout construction which then integrated backend application programming interfaces.

- **Frontend Development**

Development begins with creating an attractive user interface system that lets users communicate with the system. A fast and modular development environment will be built from HTML and CSS and JS components at the front end. The frontend components need to integrate with form validation as well as error handling functions and API integration services. The developers will build application responsiveness before focusing on maintaining user experience naturalness after finishing the UI development stage.

- **Backend Development**

The Node.js backend development launches following completion of frontend development. The system will require APIs to perform sending alerts through notification management. The database architecture requires architectural patterns to keep operational information and credentials in the storage system. MySQL will be selected because performing data management on users demands dependable system performance.

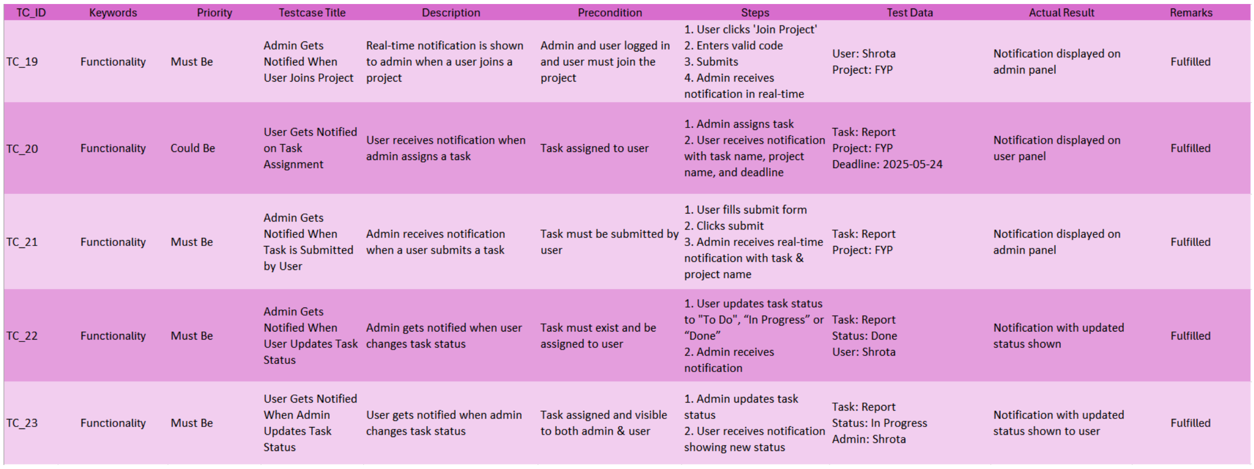
- **Integration**

The front-end design integration with backend APIs will commence after the independent development of both front-end and back-end components. The notification management executes frontend requests which are transmitted to backend operations.

5.7.4 Testing Approach

Testing stands as an essential developmental component for verifying that every system component performs to specifications. Research-based testing extends across unit testing and integration testing and user testing for complete system examination.

- Black Box Testing



TC_ID	Keywords	Priority	Testcase Title	Description	Precondition	Steps	Test Data	Actual Result	Remarks
TC_19	Functionality	Must Be	Admin Gets Notified When User Joins Project	Real-time notification is shown to admin when a user joins a project	Admin and user logged in and user must join the project	1. User clicks 'Join Project' 2. Enters valid code 3. Submits 4. Admin receives notification in real-time	User: Shrota Project: FYP	Notification displayed on admin panel	Fulfilled
TC_20	Functionality	Could Be	User Gets Notified on Task Assignment	User receives notification when admin assigns a task	Task assigned to user	1. Admin assigns task 2. User receives notification with task name, project name, and deadline	Task: Report Project: FYP Deadline: 2025-05-24	Notification displayed on user panel	Fulfilled
TC_21	Functionality	Must Be	Admin Gets Notified When Task is Submitted by User	Admin receives notification when a user submits a task	Task must be submitted by user	1. User fills submit form 2. Clicks submit 3. Admin receives real-time notification with task & project name	Task: Report Project: FYP	Notification displayed on admin panel	Fulfilled
TC_22	Functionality	Must Be	Admin Gets Notified When User Updates Task Status	Admin gets notified when user changes task status	Task must exist and be assigned to user	1. User updates task status to "To Do", “In Progress” or “Done” 2. Admin receives notification	Task: Report Status: Done User: Shrota	Notification with updated status shown	Fulfilled
TC_23	Functionality	Must Be	User Gets Notified When Admin Updates Task Status	User gets notified when admin changes task status	Task assigned and visible to both admin & user	1. Admin updates task status 2. User receives notification showing new status	Task: Report Status: In Progress Admin: Shrota	Notification with updated status shown to user	Fulfilled

Figure 53: Test Case of NMS

6. Conclusion

I was able to successfully build a fully working and responsive task management website called Task Master using the knowledge and skills I gained during my college years. The main technologies I used for this project were Node.js, HTML, and CSS. While working on this system, I faced several issues, especially technical errors and bugs. But I managed to solve most of them by reading online documentation, using resources like W3Schools.

One of the biggest challenges I faced during this project was time management. I had to work on weekly assignments, monthly milestone submissions, and other coursework at the same time. Even though it was tough, I managed it by working extra hours, especially close to the deadlines, which helped me submit everything on time.

The design and layout (UI/UX) of the system turned out just the way I had planned—clean, simple, and easy to use. I was able to achieve all the main goals of the project. The system allows admins to create projects, add or remove users, and manage tasks easily. Admins can generate special codes for each project, which users can use to join that project securely. If needed, admins can also remove users from the project. This helps keep the system organized and safe.

The system also includes task updates, user roles, and real-time management features to help teams work together smoothly. It's especially useful for students' projects who need a proper platform to assign, track, and complete tasks without confusion.

In this report, I have included everything related to my project, such as the planning methods I followed, the diagrams I created to explain how the system works, and the tools and techniques I used during development. I also created wireframes, test cases, and a product backlog to keep the project on track. These helped me organize my work and complete each part step by step.

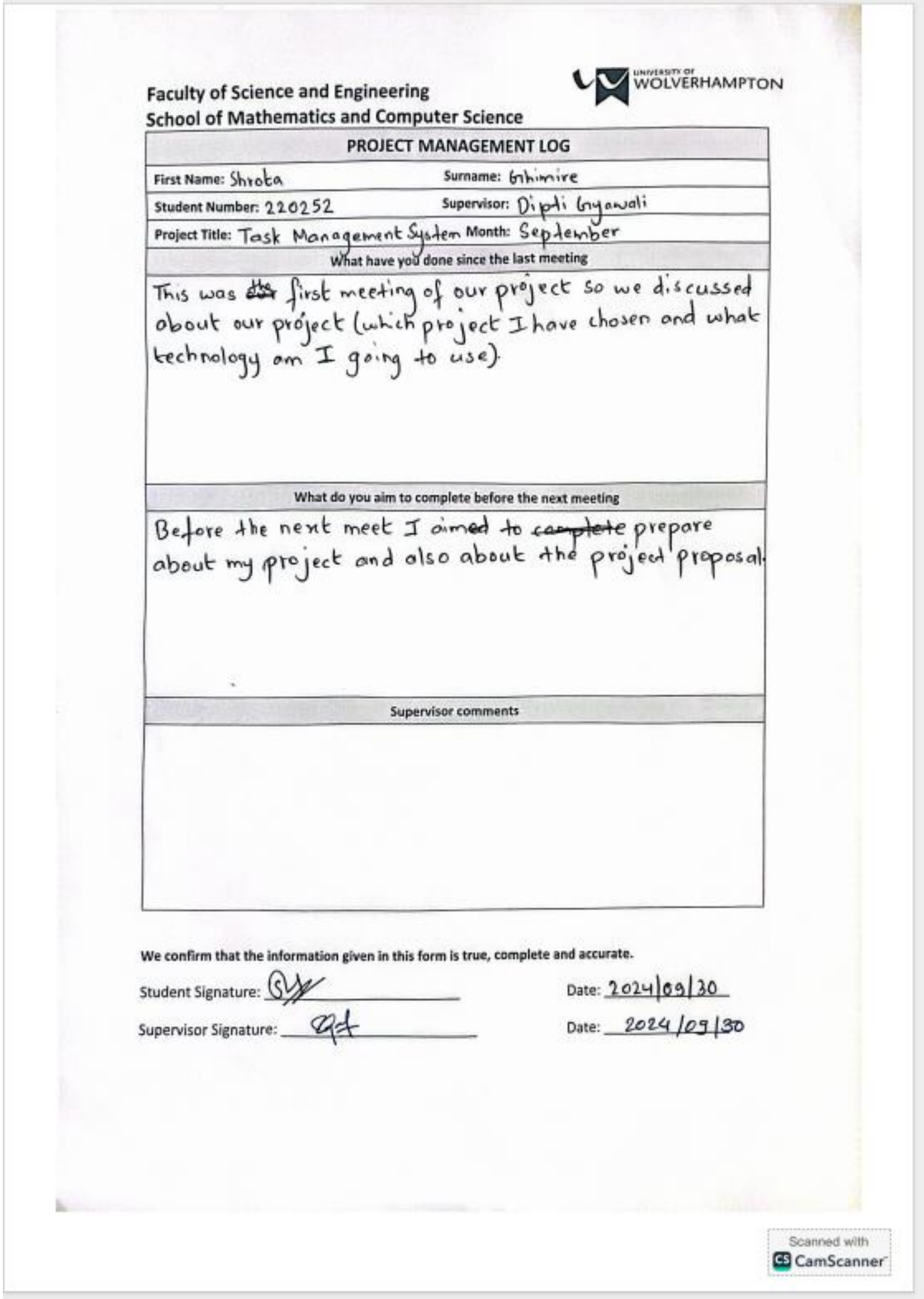
Working on this project has been a great learning experience. It helped me improve my coding skills, time management, and problem-solving ability. I am proud of what I have built, and I believe Task Master can be a helpful tool for anyone who wants to manage tasks and projects in a better, more organized way.

7. Critical Evaluation of the Report

Looking back at the report for my project **Task Master**, I believe it successfully communicates the overall journey of developing a complete task management system from start to finish. The report includes a comprehensive breakdown of the system's features, objectives, and design structure in a way that is both clear and informative. One of the strongest parts of the report is how I was able to explain the purpose and functionality of each subsystem, especially project creation, user access control, task management, and real-time updates. I took care to represent each functional and non-functional requirement properly, using diagrams such as use case, activity, ER, class, and sequence diagrams to give a visual understanding of how the system works. These diagrams helped clarify the architecture and workflow of the system. In terms of findings and process, I found that integrating real-time updates and secure project access were two of the most useful solutions for improving team coordination and user management. Through this process, I learned how important backend logic, frontend design, and database planning are in making a system responsive and functional. The final product met the goals I had set at the beginning of the project, and I was happy with the outcome, both in terms of performance and usability. My planning and time management were tested throughout, especially with ongoing academic responsibilities like assignments and milestone submissions, but I used a sprint-based development approach to stay organized. I also documented my work regularly, which made it easier to prepare the final report. I relied on quality sources such as the official Node.js documentation, W3Schools, troubleshoot issues and improve features. These helped me stay on track and overcome technical obstacles. Writing the report also helped me reflect on my growth—both technically and personally. I developed skills in web development using Node.js, HTML, and CSS, and improved my ability to explain technical concepts in written form. More importantly, I became more disciplined and confident in my ability to manage a full-scale project independently. This experience has added to my professional development by strengthening my problem-solving skills, technical abilities, and time management. Personally, it taught me resilience and the value of consistent effort. In conclusion, while there were challenges during the report preparation and system development, I believe the final report effectively showcases my learning, achievements, and the practical application of the skills I've acquired throughout my academic journey.

8. Evidence of Project Management

8.1 Log Sheets



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG	
First Name: Shrota	Surname: Ghimire
Student Number: 220252	Supervisor: Dipti Gyawali
Project Title: Task Management System	Month: September

What have you done since the last meeting

This was ~~our~~ the first meeting of our project so we discussed about our project (which project I have chosen and what technology am I going to use).

What do you aim to complete before the next meeting

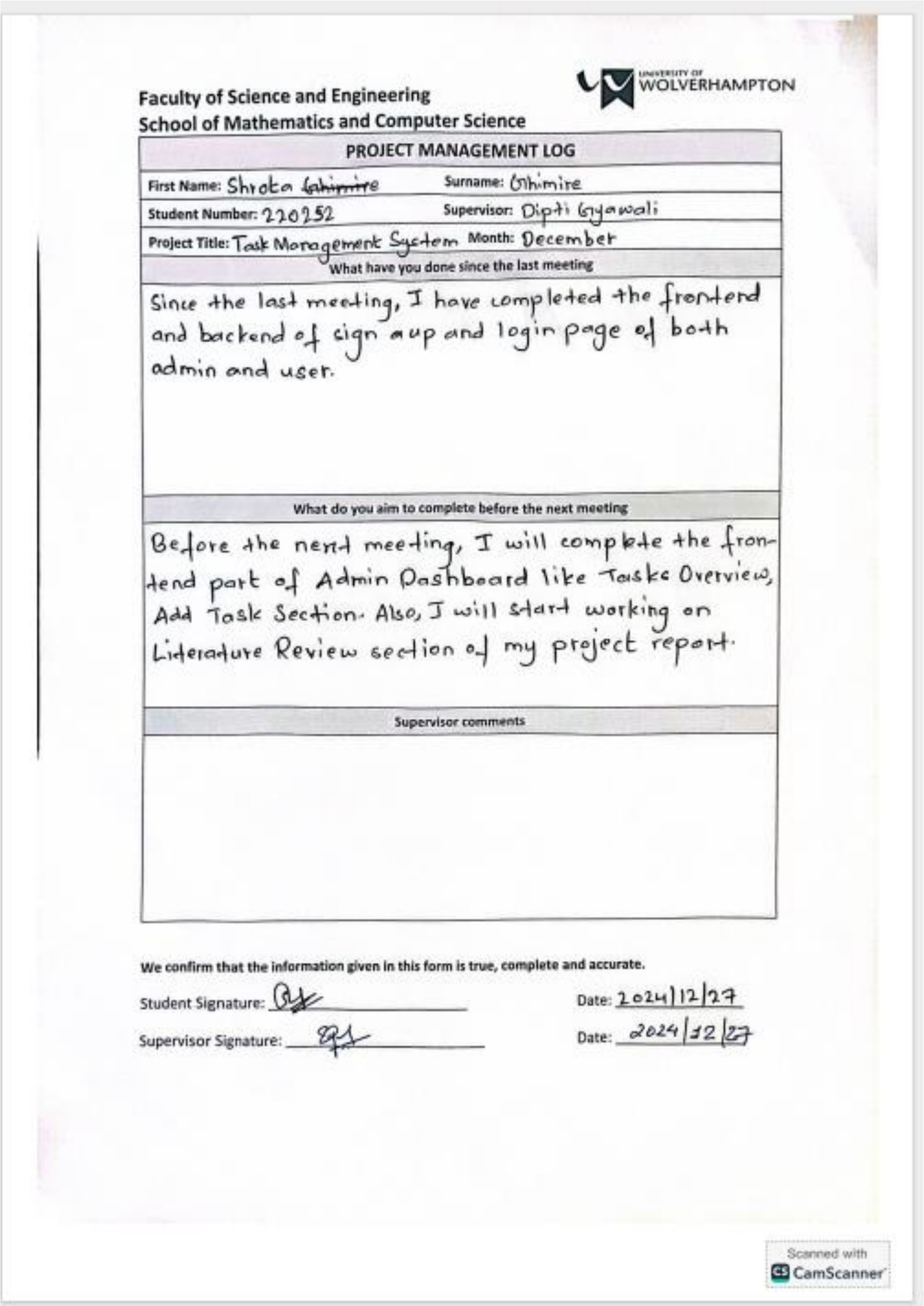
Before the next meet I aimed to ~~complete~~ prepare about my project and also about the project proposal.

Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: ____________ Date: 2024/09/30

Supervisor Signature: ____________ Date: 2024/09/30



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shrota ~~Ghimire~~ Surname: Ghimire

Student Number: 220252 Supervisor: Dipti Gyawali

Project Title: Task Management System Month: December

What have you done since the last meeting

Since the last meeting, I have completed the frontend and backend of sign up and login page of both admin and user.

What do you aim to complete before the next meeting

Before the next meeting, I will complete the frontend part of Admin Dashboard like Tasks Overview, Add Task Section. Also, I will start working on Literature Review section of my project report.

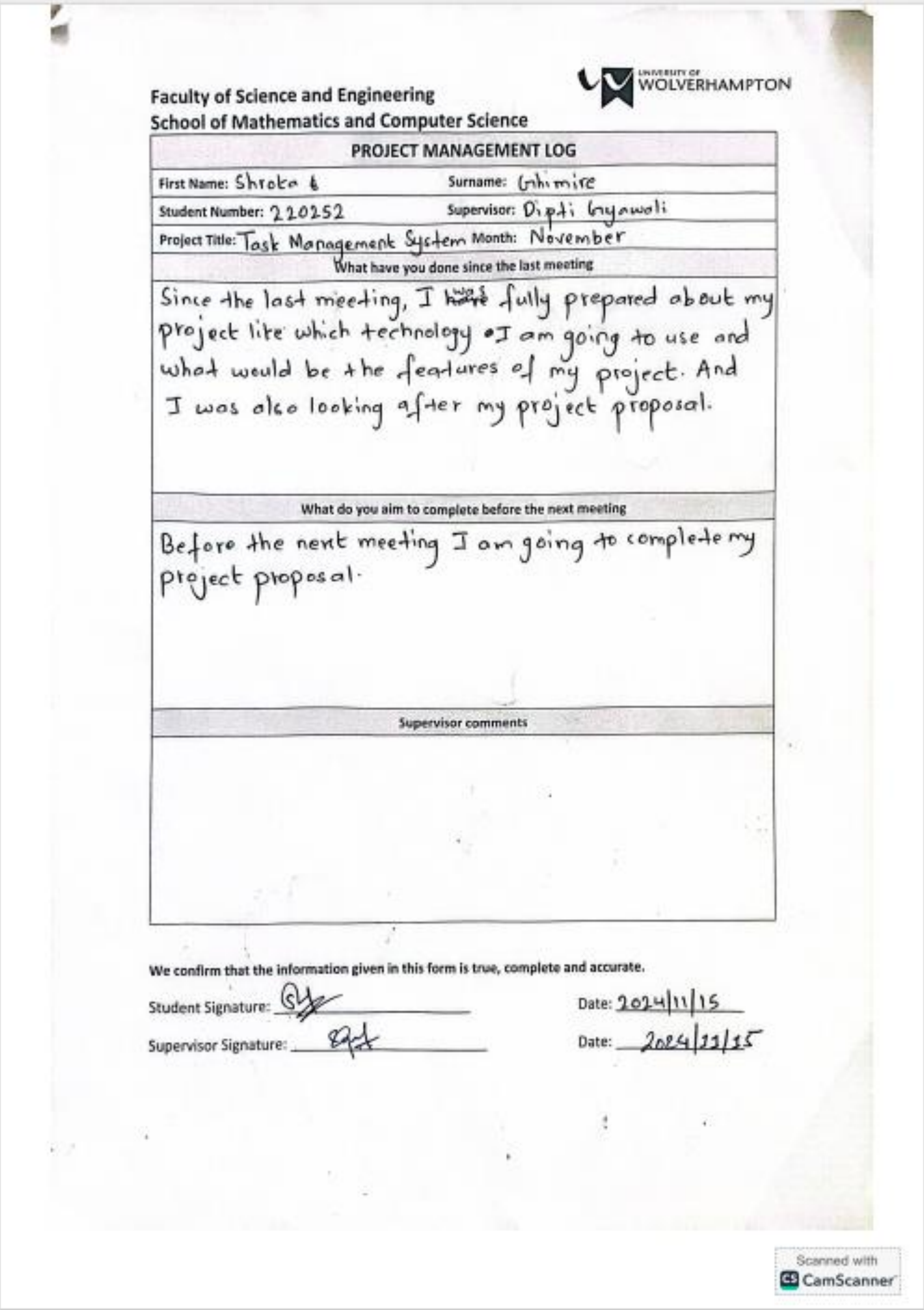
Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: 2024/12/27

Supervisor Signature: Date: 2024/12/27

Scanned with CamScanner



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shroka b	Surname: Ghimire
Student Number: 220252	Supervisor: Dipti Gyawali
Project Title: Task Management System	Month: November

What have you done since the last meeting

Since the last meeting, I ~~have~~ was fully prepared about my project like which technology I am going to use and what would be the features of my project. And I was also looking after my project proposal.

What do you aim to complete before the next meeting

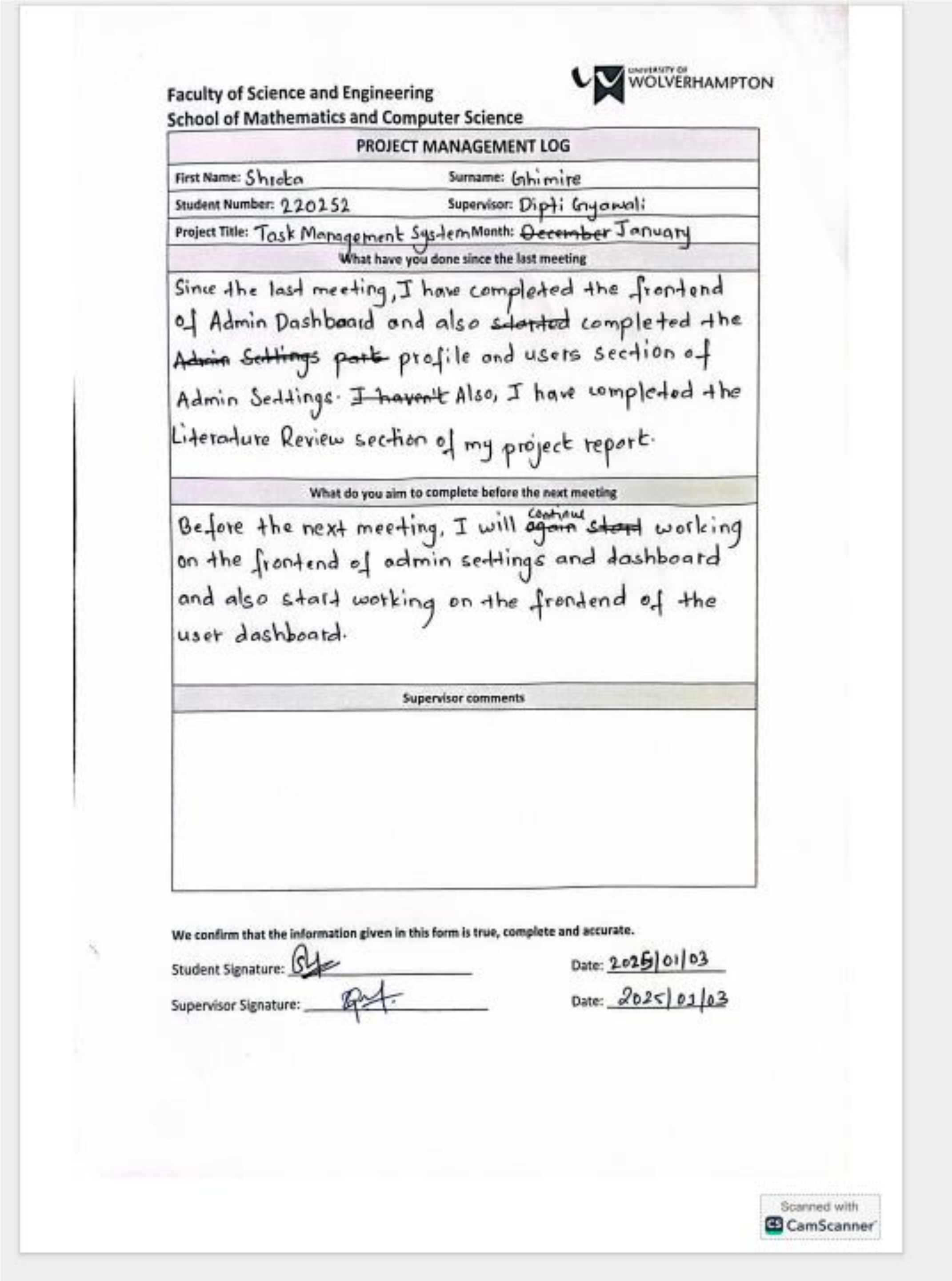
Before the next meeting I am going to complete my project proposal.

Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: ______ Date: 2024/11/15

Supervisor Signature: ______ Date: 2024/11/15



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shreta Surname: Ghimire

Student Number: 220252 Supervisor: Dipti Gyawali

Project Title: Task Management System Month: ~~December~~ January

What have you done since the last meeting

Since the last meeting, I have completed the frontend of Admin Dashboard and also ~~started~~ completed the ~~Admin Settings part~~ profile and users section of Admin Seddings. ~~I haven't~~ Also, I have completed the Literature Review section of my project report.

What do you aim to complete before the next meeting

Before the next meeting, I will ~~again start~~ continue working on the frontend of admin settings and dashboard and also start working on the frontend of the user dashboard.

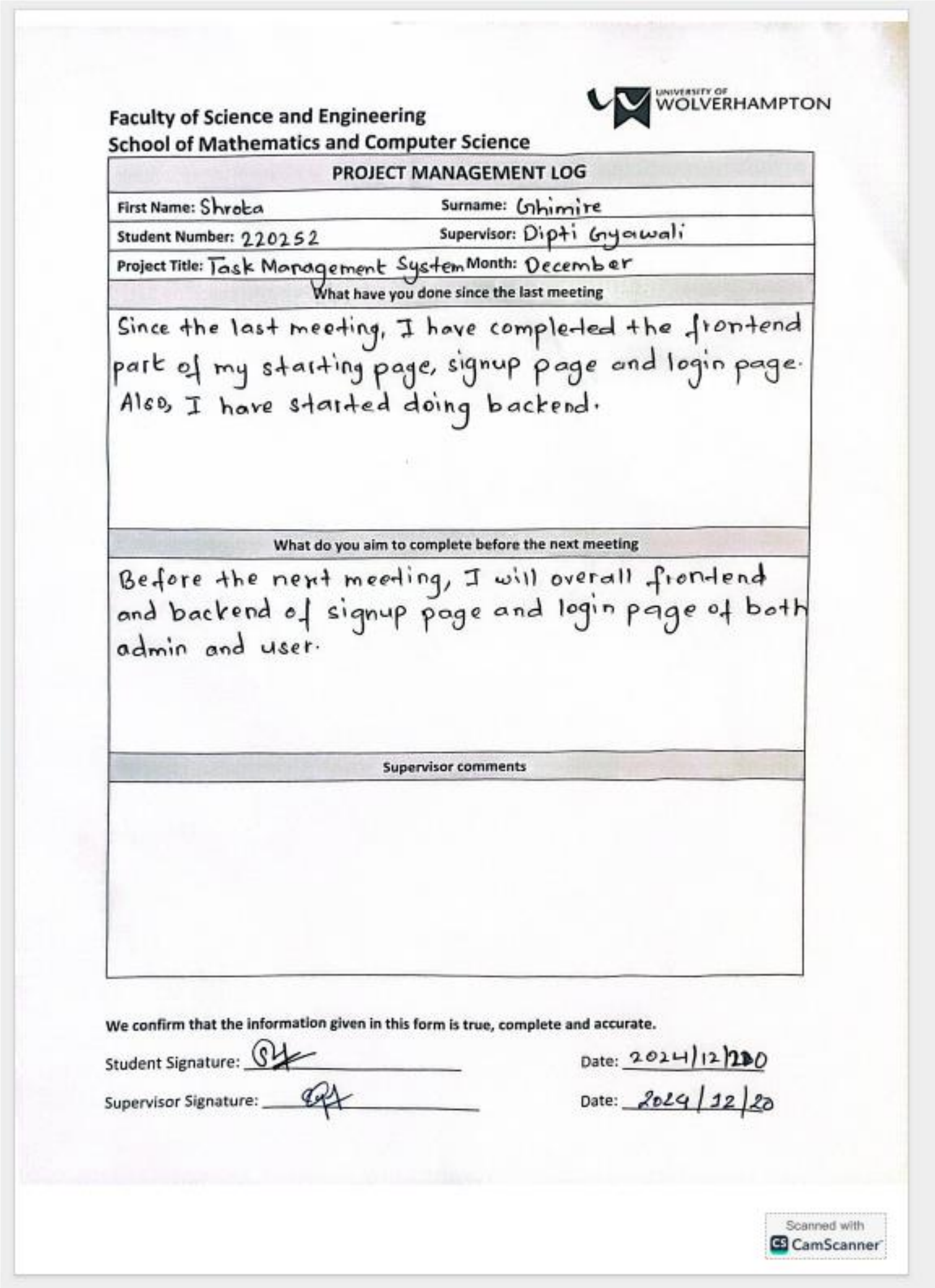
Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: 2025/01/03

Supervisor Signature: Date: 2025/01/03

Scanned with CamScanner



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shroka	Surname: Ghimire
Student Number: 220252	Supervisor: Dipti Gyawali
Project Title: Task Management System	Month: December

What have you done since the last meeting

Since the last meeting, I have completed the frontend part of my starting page, signup page and login page. Also, I have started doing backend.

What do you aim to complete before the next meeting

Before the next meeting, I will overall frontend and backend of signup page and login page of both admin and user.

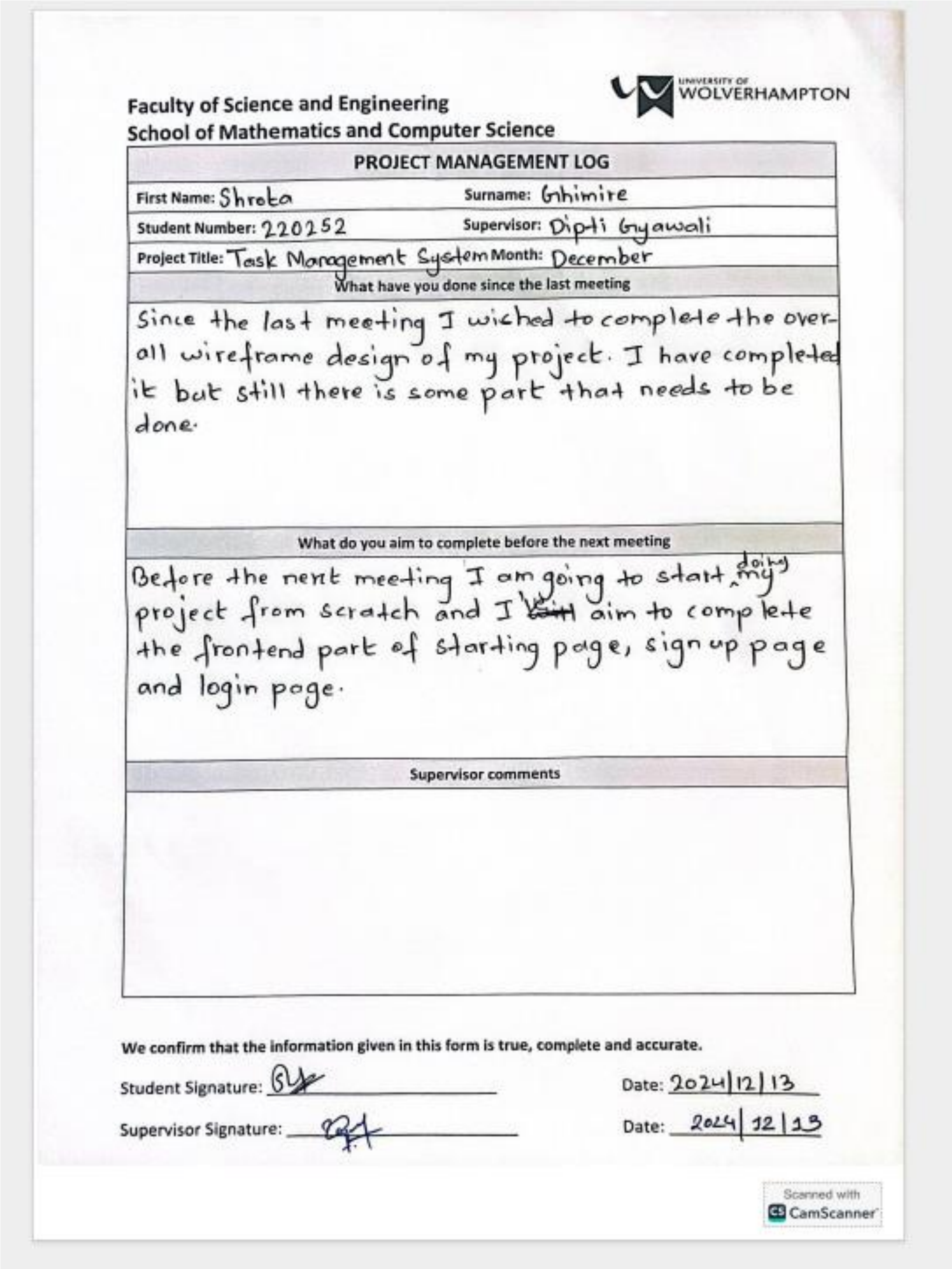
Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: 2024/12/20

Supervisor Signature: Date: 2024/12/20

Scanned with CamScanner



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shreka	Surname: Ghimire
Student Number: 220252	Supervisor: Dipti Gyawali
Project Title: Task Management System	Month: December

What have you done since the last meeting

Since the last meeting I wiched to complete the overall wireframe design of my project. I have completed it but still there is some part that needs to be done.

What do you aim to complete before the next meeting

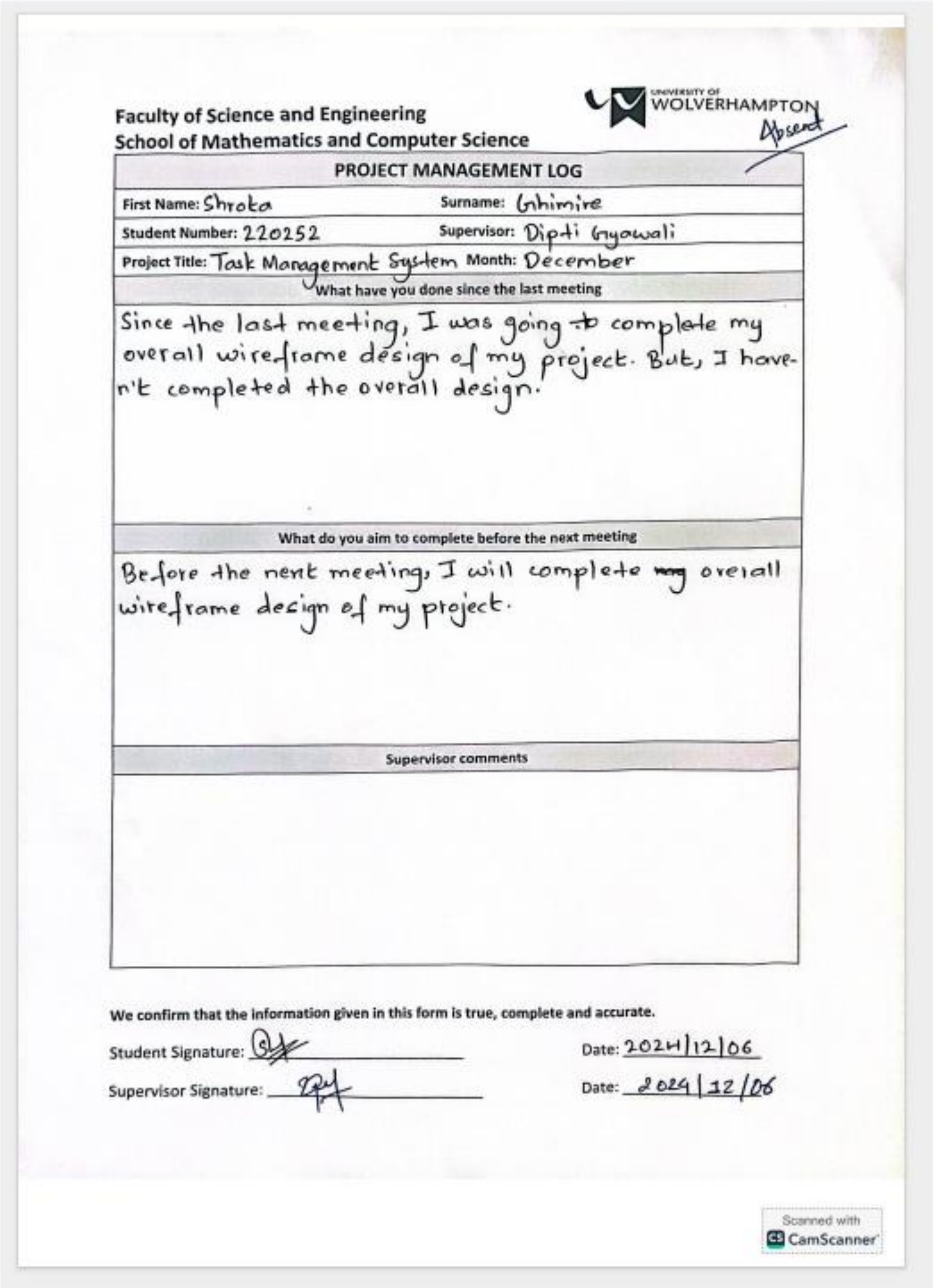
Before the next meeting I am going to start doing my project from scratch and I ~~will~~ aim to complete the frontend part of starting page, sign up page and login page.

Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: ______ Date: 2024|12|13

Supervisor Signature: ______ Date: 2024|12|13



Faculty of Science and Engineering
School of Mathematics and Computer Science

Absent

PROJECT MANAGEMENT LOG

First Name: Shroka Surname: Ghimire

Student Number: 220252 Supervisor: Dipti Gyawali

Project Title: Task Management System Month: December

What have you done since the last meeting

Since the last meeting, I was going to complete my overall wireframe design of my project. But, I haven't completed the overall design.

What do you aim to complete before the next meeting

Before the next meeting, I will complete ~~my~~ overall wireframe design of my project.

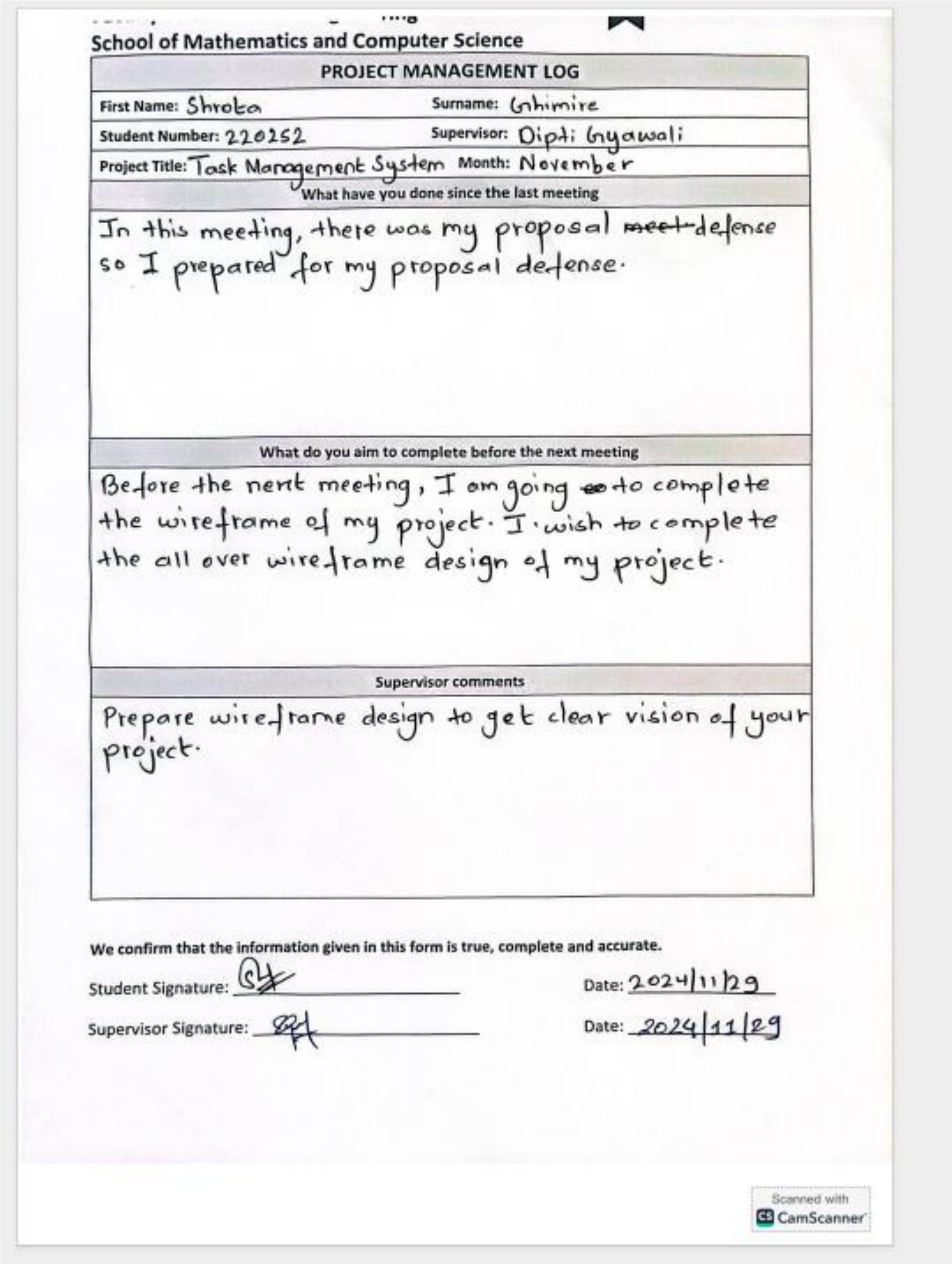
Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: 2024/12/06

Supervisor Signature: Date: 2024/12/06

Scanned with CamScanner



School of Mathematics and Computer Science

PROJECT MANAGEMENT LOG

First Name: Shroka	Surname: Ghimire
Student Number: 220252	Supervisor: Dipti Gyawali
Project Title: Task Management System	Month: November

What have you done since the last meeting

In this meeting, there was my proposal ~~meet~~ defence so I prepared for my proposal defense.

What do you aim to complete before the next meeting

Before the next meeting, I am going ~~co~~ to complete the wireframe of my project. I wish to complete the all over wireframe design of my project.

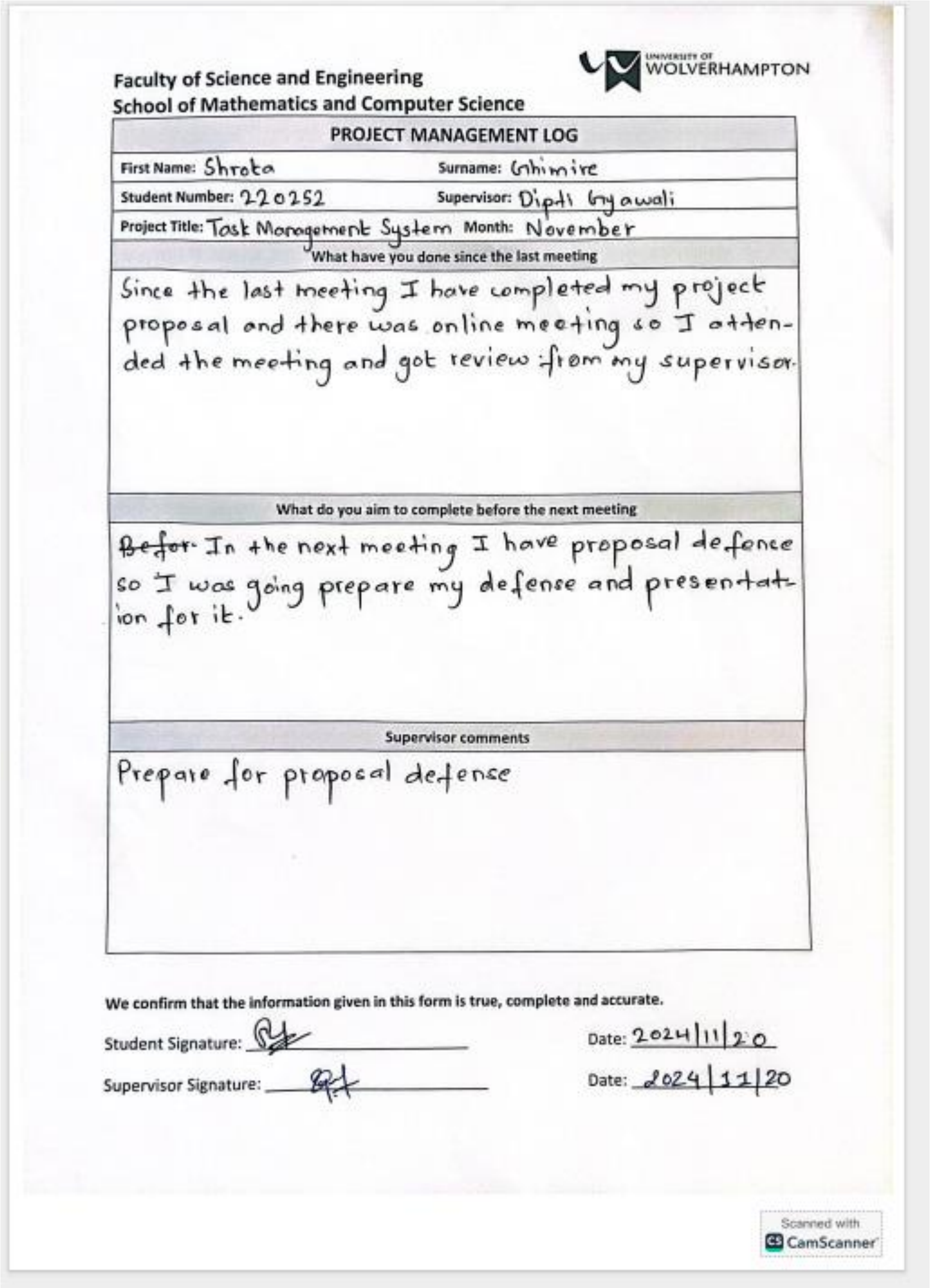
Supervisor comments

Prepare wireframe design to get clear vision of your project.

We confirm that the information given in this form is true, complete and accurate.

Student Signature: ____________ Date: 2024/11/29

Supervisor Signature: ____________ Date: 2024/11/29



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shrota	Surname: Ghimire
Student Number: 220252	Supervisor: Dipti Gyawali
Project Title: Task Management System	Month: November

What have you done since the last meeting

Since the last meeting I have completed my project proposal and there was online meeting so I attended the meeting and got review from my supervisor.

What do you aim to complete before the next meeting

~~Befor~~ In the next meeting I have proposal defence so I was going prepare my defense and presentation for it.

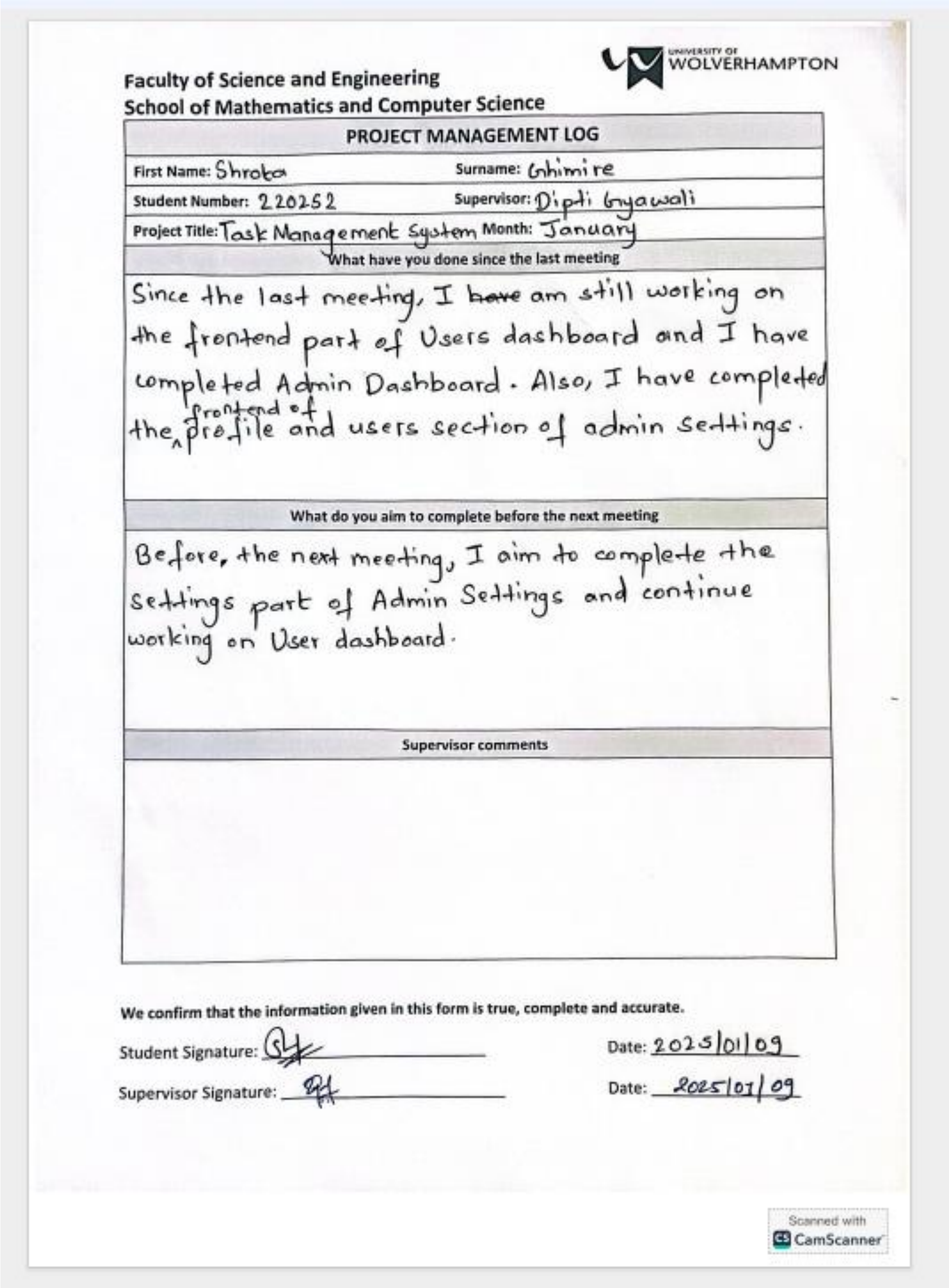
Supervisor comments

Prepare for proposal defense

We confirm that the information given in this form is true, complete and accurate.

Student Signature: ______________ Date: 2024|11|20

Supervisor Signature: ______________ Date: 2024|11|20



UNIVERSITY OF WOLVERHAMPTON

Faculty of Science and Engineering
School of Mathematics and Computer Science

PROJECT MANAGEMENT LOG

First Name: Shroba	Surname: Ghimire
Student Number: 220252	Supervisor: Dipti Gyawali
Project Title: Task Management System	Month: January

What have you done since the last meeting

Since the last meeting, I ~~have~~ am still working on the frontend part of Users dashboard and I have completed Admin Dashboard. Also, I have completed the frontend of profile and users section of admin settings.

What do you aim to complete before the next meeting

Before, the next meeting, I aim to complete the settings part of Admin Settings and continue working on User dashboard.

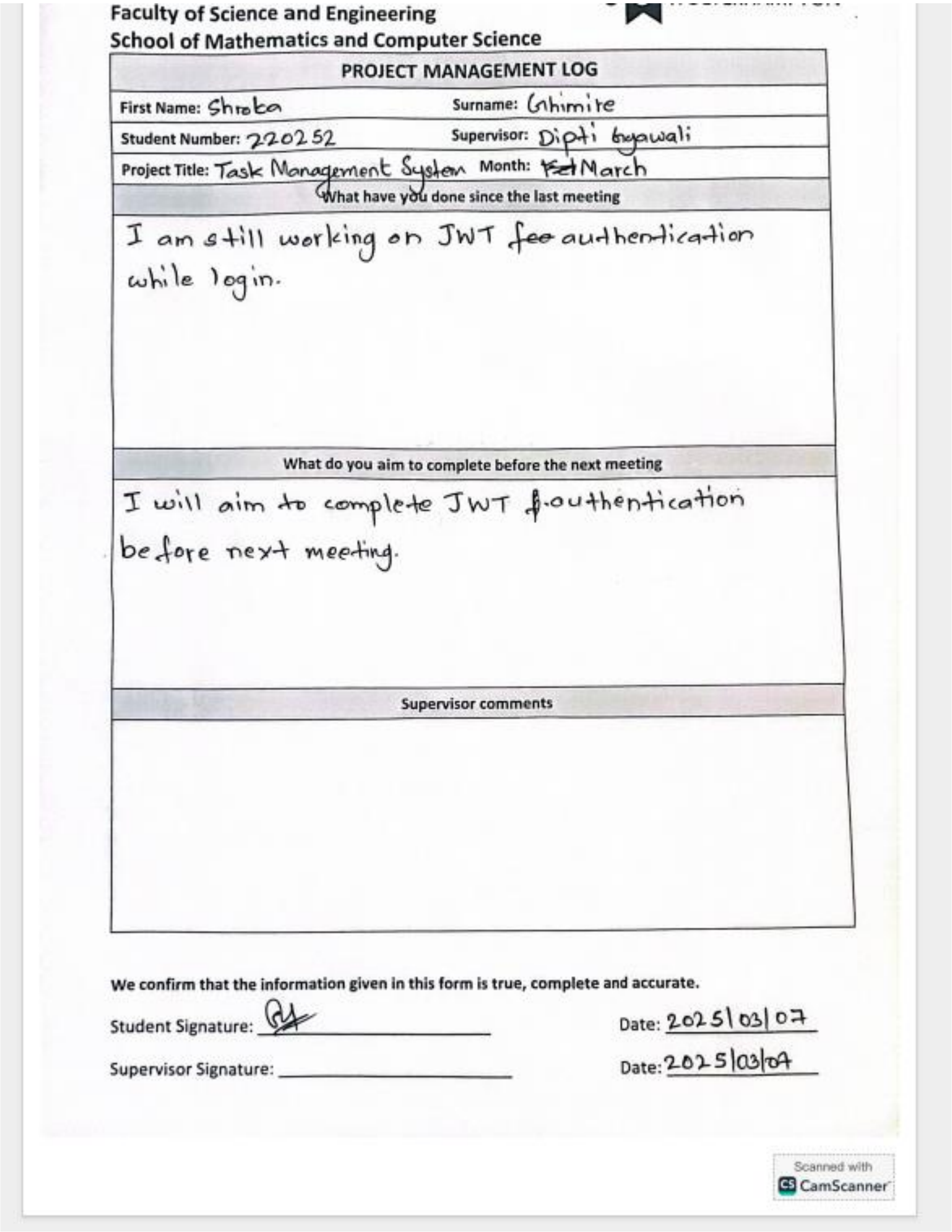
Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: ____________ Date: 2025/01/09

Supervisor Signature: ____________ Date: 2025/01/09

Scanned with CamScanner



Faculty of Science and Engineering
School of Mathematics and Computer Science

PROJECT MANAGEMENT LOG

First Name: Shroka Surname: Ghimire

Student Number: 220252 Supervisor: Dipti Gyawali

Project Title: Task Management System Month: ~~Feb~~ March

What have you done since the last meeting

I am still working on JWT ~~fee~~ authentication while login.

What do you aim to complete before the next meeting

I will aim to complete JWT ~~f~~.outhentication before next meeting.

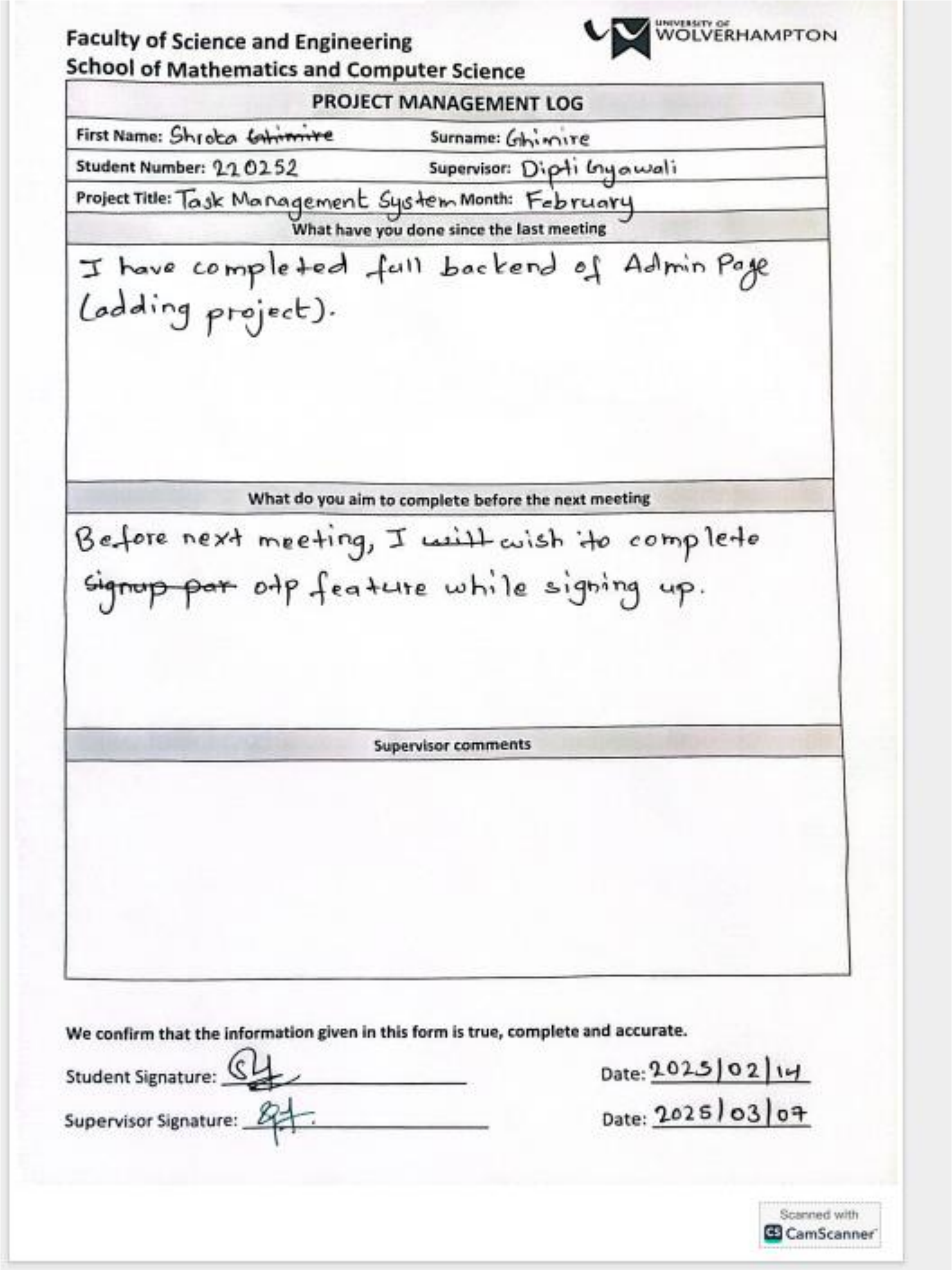
Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: ____________________ Date: 2025|03|07

Supervisor Signature: ____________________ Date: 2025|03|07

Scanned with CamScanner



Faculty of Science and Engineering
School of Mathematics and Computer Science

UNIVERSITY OF WOLVERHAMPTON

PROJECT MANAGEMENT LOG

First Name: Shroka ~~Ghimire~~ Surname: Ghimire

Student Number: 220252 Supervisor: Dipti Gyawali

Project Title: Task Management System Month: February

What have you done since the last meeting

I have completed full backend of Admin Page (adding project).

What do you aim to complete before the next meeting

Before next meeting, I ~~will~~ wish to complete ~~signup par~~ otp feature while signing up.

Supervisor comments

We confirm that the information given in this form is true, complete and accurate.

Student Signature: Date: 2025/02/14

Supervisor Signature: Date: 2025/03/07

9. References

A. H. M. Al-Ahmad, 2020. A Case of Feature Driven Development.

Alliance, A., 2000. *What is a product backlog.* s.l.:s.n.

Anon., 2010. *JavaScript.* s.l.:s.n.

Anon., 2015. *HTML Explanation.* s.l.:s.n.

Anon., 2015. *VS Code.* s.l.:s.n.

Anon., 2015. *XAMPP.* s.l.:s.n.

Anon., 2020. *Node js.* s.l.:s.n.

Anon., n.d. *CSS.* s.l.:s.n.

Beck, K., & Andres, 2004. *Extreme Programming Explained.* 2nd ed. s.l.:s.n.

Fowler, M, 2003. *UML.* s.l.:s.n.

J., B., 2016. *Task Management in Education.* s.l.:s.n.

Kerzner H, 2013. *Project Management.* s.l.:s.n.

Pichler, R, 2010. *Management.* s.l.:s.n.

Library, A. D., 2023. Unpacking Task Management Tools, Values, and Worker Dynamics.

May, S., 2022. Task Management System.

Sandhya, K., 2023. Employee Task Tracking System, 5(7).

Spezie, M., 2023. The Development of Task Management Software .