

TeamSync

Enhanced Remote Task Management System

Business Requirement Document

Karthick Sankar and Nasih P S

September 3, 2024

Contents

1	Introduction	2
1.1	Project Overview	2
1.2	Objective	2
1.3	Target Audience	2
1.4	Key Features	2
2	Business Goals	2
2.1	Improve Coordination	2
2.2	Support Multiple Time Zones	2
2.3	Increase Accountability	3
2.4	Enhance Productivity	3
2.5	Support Growth	3
3	Scope	3
3.1	In Scope	3
3.2	Out of Scope	3
4	Stakeholders	3
4.1	Business Stakeholders	3
4.2	User Needs	4

1 Introduction

1.1 Project Overview

The TeamSync project aims to develop a web-based application designed to enhance team coordination and task management for remote teams across various time zones. By addressing the unique challenges of remote work, TeamSync will offer features for managing projects, assigning tasks, tracking progress, and receiving notifications, thereby improving productivity and collaboration.

1.2 Objective

The objective of TeamSync is to create a robust task management platform that supports transparent communication and efficient project management, facilitating seamless coordination among remote teams despite geographical and time zone differences.

1.3 Target Audience

The primary users of TeamSync include remote teams, project managers, and organizations with a global workforce. The platform is designed to address the needs of these users by providing tools for effective project and task management.

1.4 Key Features

- Project creation and management
- Task assignment and tracking
- Time zone management
- Notification system
- User profile management

2 Business Goals

2.1 Improve Coordination

Enhance team coordination by providing a transparent and unified task management system that facilitates effective communication and alignment on project objectives.

2.2 Support Multiple Time Zones

Enable teams to manage tasks and projects across different time zones, ensuring that work is synchronized and deadlines are met despite geographical differences.

2.3 Increase Accountability

Promote accountability by offering visibility into task assignments, progress tracking, and deadlines, ensuring that team members are responsible for their contributions and deadlines.

2.4 Enhance Productivity

Improve overall productivity by streamlining task management processes, reducing delays caused by time zone differences, and providing timely notifications and updates.

2.5 Support Growth

Design the platform to support scalability, accommodating an increasing number of users and projects as the organization grows.

3 Scope

3.1 In Scope

- **User Registration and Login:** Secure access to the platform through account creation and authentication.
- **Project Creation and Management:** Tools for setting up and overseeing projects, including task assignment and progress monitoring.
- **Task Assignment and Tracking:** Features for assigning tasks, tracking their status, and managing deadlines.
- **Time Zone Management:** Handling of time zone differences to ensure accurate scheduling and coordination.
- **Notification System:** Automated alerts for task updates, assignments, and approaching deadlines.

3.2 Out of Scope

- **Video Conferencing Integration:** The platform will not include video conferencing features.
- **Advanced Analytics and Reporting:** Excludes advanced analytics and reporting capabilities beyond basic tracking.

4 Stakeholders

4.1 Business Stakeholders

- **Remote Teams:** Users who will directly benefit from enhanced project and task management capabilities.

- **Project Managers:** Individuals responsible for overseeing project execution and ensuring alignment with business goals.
- **Organizational Leaders:** Executives and decision-makers who will use the platform's insights to drive strategic decisions and monitor team performance.

4.2 User Needs

- **Remote Team Members:** Need for efficient task management and coordination tools that address time zone challenges.
- **Project Managers:** Require comprehensive project oversight features and progress tracking.
- **Organizational Leaders:** Seek transparency and accountability in project execution to support strategic planning and decision-making.

TeamSync

Enhanced Remote Task Management System

Functional and Non-Functional Document

Karthick Sankar and Nasih P S

September 3, 2024

1 Introduction

Project Title: Remote Work Task Management System

Objective: To create a web-based application to improve team coordination and task management for remote teams across various time zones.

2 Scope

2.1 In Scope

- User registration and login.
- Project creation and management.
- Task assignment and tracking.
- Time zone management.
- Notification system for task updates and deadlines.

2.2 Out of Scope

- Video conferencing integration.
- Advanced analytics and reporting.

3 Stakeholders

- **Project Members:** Users who can view and update tasks.
- **Project Leaders:** Users who can create projects, assign tasks, and monitor progress.
- **Admins:** Users who can manage user accounts, view all projects and tasks, and configure system settings.

4 Functional Requirements

4.1 User Registration and Authentication

- Users should be able to register an account with basic details such as name, email, password, and time zone.
- Users should be able to log in using their registered credentials.

4.2 User Roles and Permissions

- There should be two user roles: Admin and User.
- Admins should have the ability to manage user accounts, projects, and system settings.
- Users can create and manage their own projects and participate in projects they are invited to.

4.3 Profile Management

- Users should be able to view and update their profile details, including name, email, password, and time zone.
- Users should be able to change their passwords.

4.4 Project Management

- Users (Project Leaders) should be able to create new projects.
- Users (Project Leaders) should be able to invite other registered users to their projects.

4.5 Task Management

- Project leaders should be able to create, assign, and manage tasks within a project.
- Tasks should be assigned with deadlines, and the system should automatically adjust deadlines based on the assignee's time zone.
- Task progress should be trackable through status columns (to-do, in-progress, testing, completed).
- Team members should be able to update the status of their tasks.

4.6 Dashboard

- The project dashboard should display an overview of the project, including all tasks, their statuses, and the members involved.
- Users should be able to filter tasks based on status, due date, or assigned member.

4.7 Notification System

- Users should receive notifications when tasks are assigned to them.
- Project leaders should receive notifications when tasks are completed by team members.
- Users should receive reminders when a task is approaching its due date.

4.8 Time Zone Management

- Users should be able to set their time zone in their profile.
- Deadlines should be automatically converted to the user's local time zone.

4.9 Asynchronous Communication

- Notifications should be delivered asynchronously, ensuring timely updates without requiring the user to refresh the page.
- A communication log should be maintained for tracking all notifications.

4.10 Admin Features

- Admins should have the ability to view and manage all projects and users.
- Admins should be able to delete or deactivate user accounts and projects.

5 Non-Functional Requirements

5.1 Performance

- The application should load within 2-3 seconds for the majority of user interactions.
- The system should handle up to 1000 concurrent users without performance degradation.

5.2 Scalability

- The application should be able to scale horizontally to accommodate increasing numbers of users and projects.

5.3 Security

- Passwords should be stored securely using encryption.
- User sessions should be managed securely, with tokens expiring after a period of inactivity.
- Role-based access control should be implemented to ensure users can only access data they are permitted to see.

5.4 Availability

- The system should be available 99.9% of the time, with minimal downtime for maintenance.
- A backup and recovery system should be in place to restore data in case of failures.

5.5 Usability

- The user interface should be intuitive and easy to navigate, with consistent design across all pages.
- The application should be accessible, following standard accessibility guidelines.

5.6 Maintainability

- The codebase should be modular and well-documented to allow for easy maintenance and updates.
- The system should be designed with clear separation of concerns between frontend and backend components.

5.7 Compatibility

- The application should be compatible with modern web browsers (Chrome, Firefox, Safari, Edge).
- The system should be responsive and work seamlessly on both desktop and mobile devices.

6 Design

6.1 High Level Design Document

6.1.1 Architectural Overview

The architectural overview describes the major components and their interactions in the system. **Architectural Diagram Details:**

- **Web Client:** This is the user interface through which users interact with the system. It could be a web browser-based application or a mobile app.
- **Security + API Gateway:** This component acts as a central point of control for all incoming requests. It handles security measures such as authentication and authorization, ensuring that only authorized users can access the system. Additionally, it functions as an API gateway, routing requests to the appropriate services based on their endpoints.
- **Project Service:** This service is responsible for managing project-related data and operations. It might handle tasks like creating, updating, and deleting projects, assigning tasks to team members, and tracking project progress.
- **Notification Service:** This service handles sending notifications to users, such as email alerts or push notifications. It could be triggered by events like project updates, task deadlines, or new messages.
- **SQL:** This represents the database where the application stores its data. It is likely a relational database like MySQL, PostgreSQL, or SQL Server, used to persist information about projects, users, tasks, and notifications.

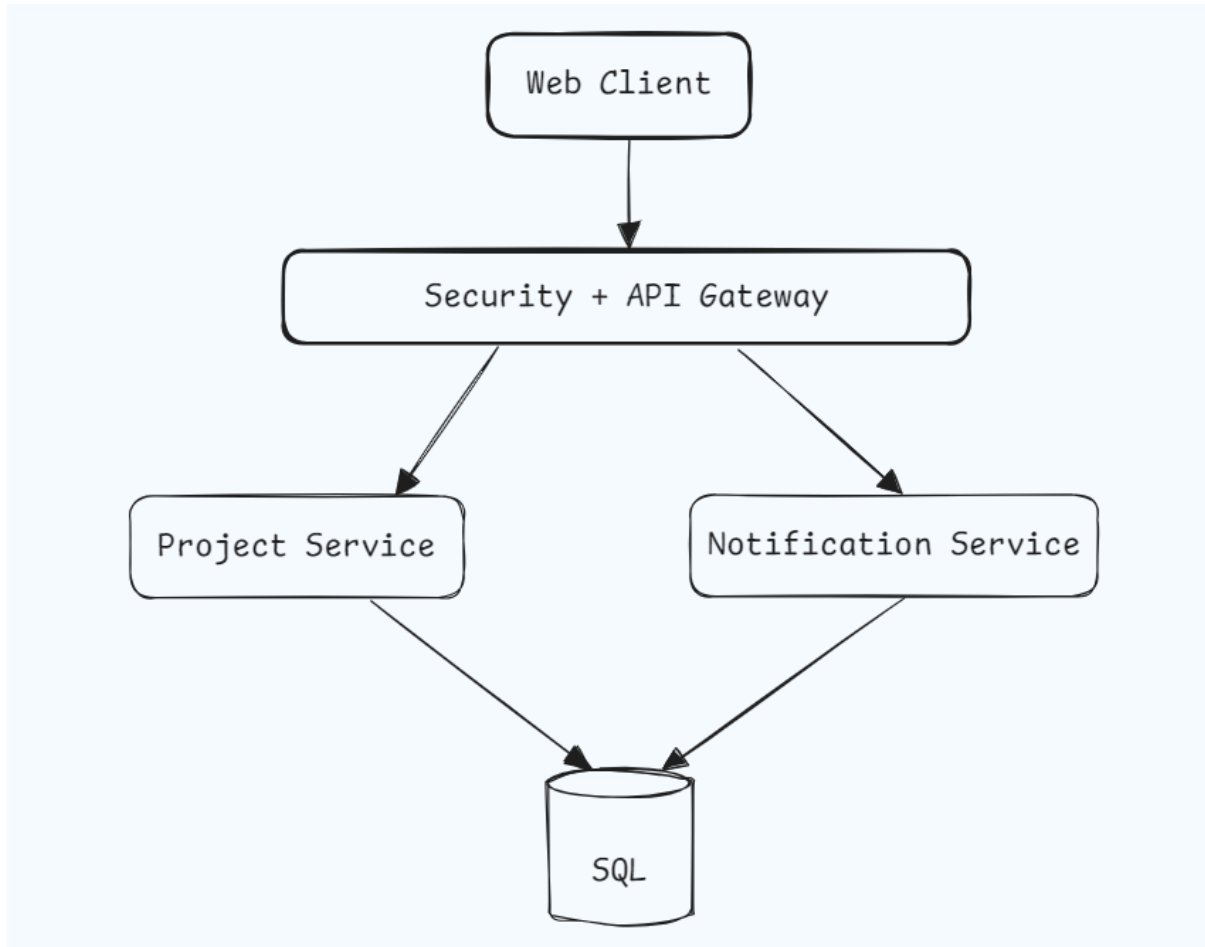


Figure 1: Architectural Overview Diagram

6.1.2 Functional Specifications

Functional specifications detail how components interact and process requests. **Sequence Diagram Details:**

- **User Interaction:** The user initiates a request, such as logging in, creating a project, or updating a task.
- **API Gateway Handling:** The API gateway processes the request, performing authentication and authorization checks. It then forwards the request to the appropriate microservice.
- **JWT Generation:** If the request involves user authentication, the API gateway generates a JSON Web Token (JWT) for the user, which is used to verify their identity for subsequent requests.
- **Microservices Processing:** The microservices (Project Service, Notification Service, etc.) handle the request according to their responsibilities. They interact with the database to retrieve or update data as necessary.
- **Response Forwarding:** Once the microservice has processed the request, it sends the response back to the API gateway, which then forwards it to the user.

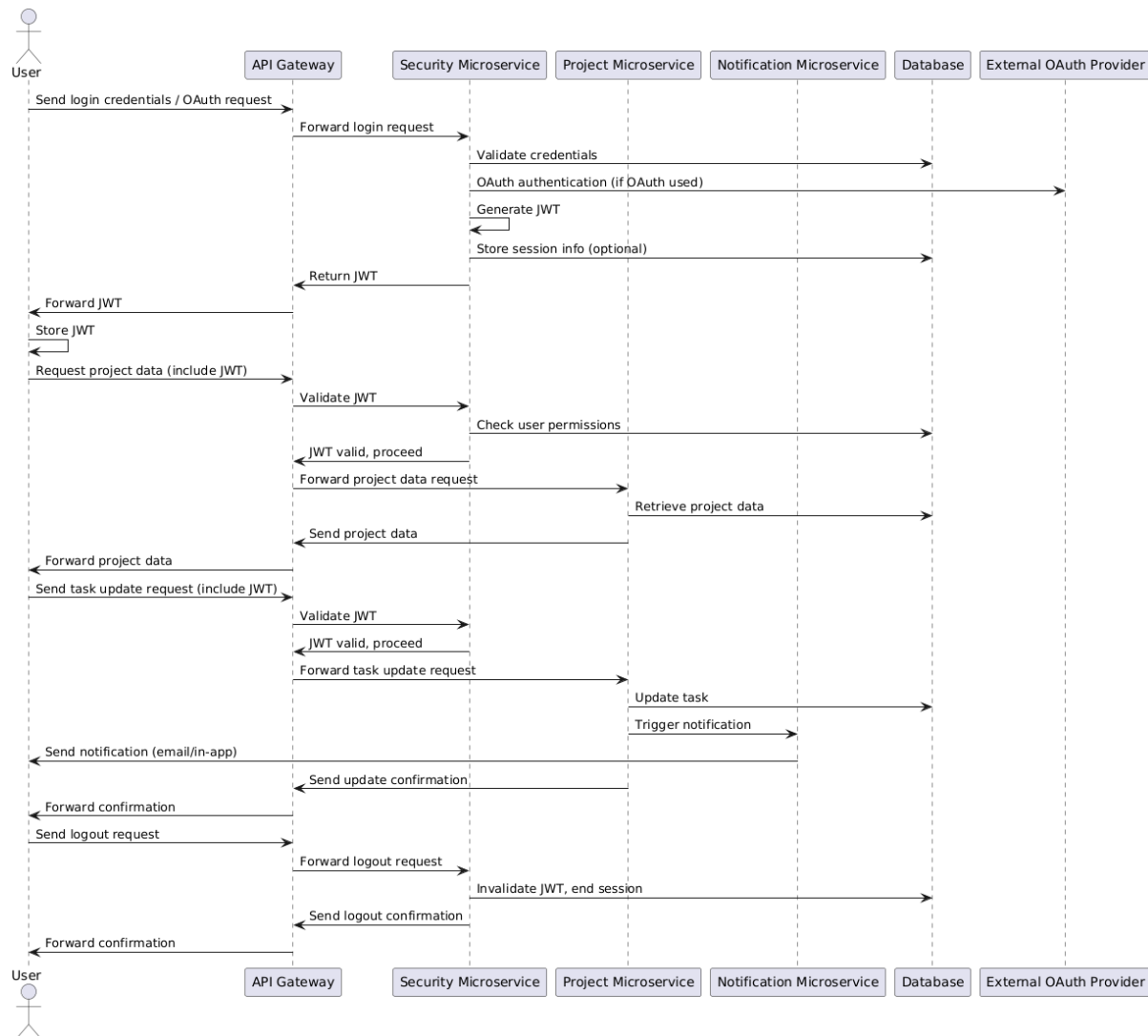


Figure 2: Sequence Diagram

6.2 Low Level Design Document

6.2.1 Class Diagram

The class diagram illustrates the key classes and their relationships within the system.

6.2.2 Entity-Relationship Diagram (ERD)

The ERD depicts the database schema and relationships between entities. **ER Diagram Details:**

- **User to Project:** One-to-many relationship, where one user can lead multiple projects.
- **Project to Task:** One-to-many relationship, where one project can have multiple tasks.
- **User to Task:** Many-to-many relationship, where multiple users can be assigned to multiple tasks.

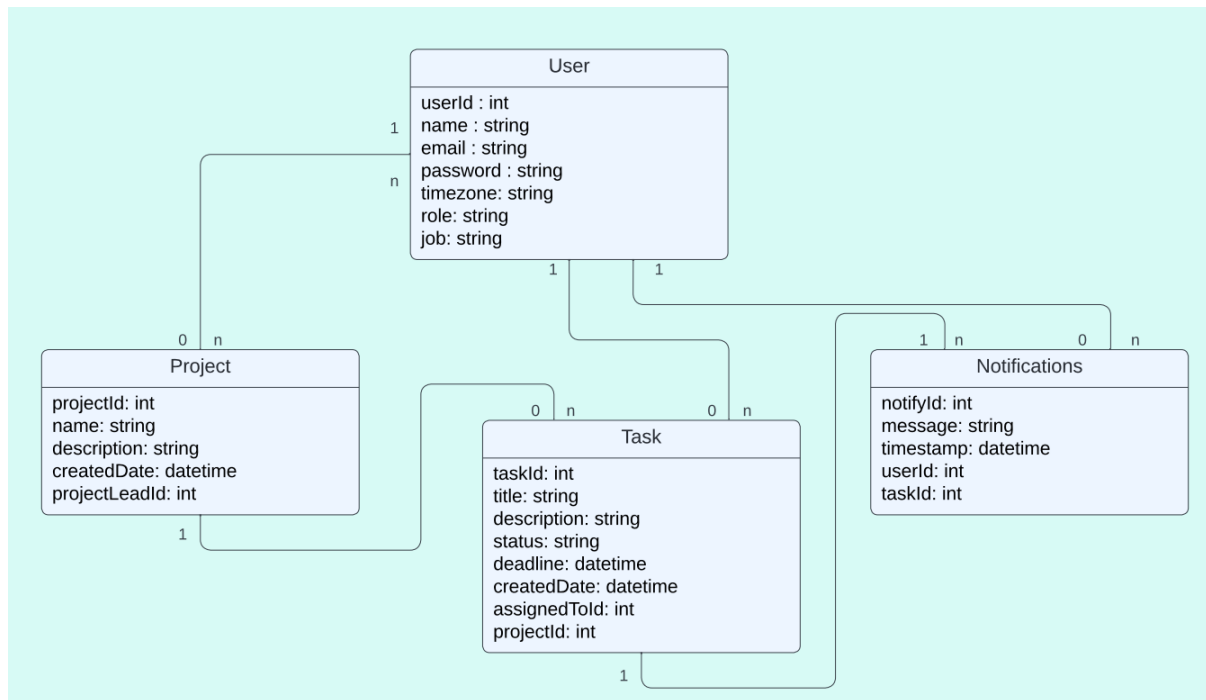


Figure 3: Class Diagram

- **Task to Notification:** One-to-many relationship, where one task can have multiple notifications associated with it.

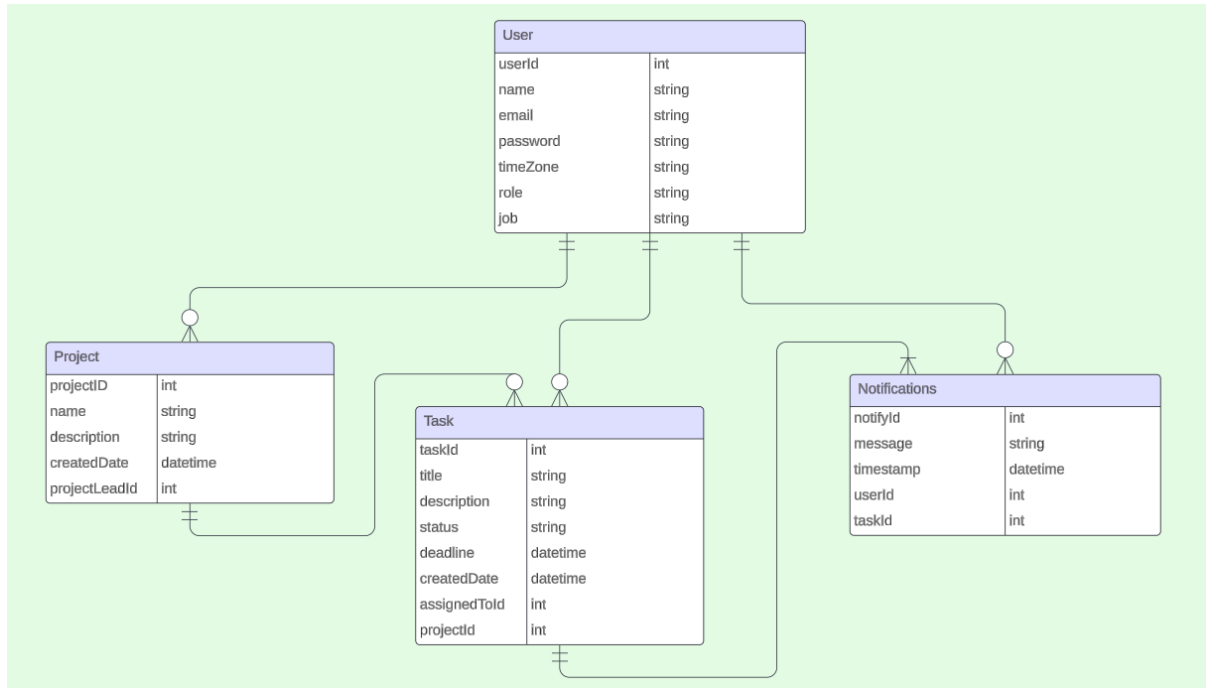


Figure 4: Entity-Relationship Diagram

6.2.3 Use Case Diagram

The use case diagram shows the interactions between users and the system. **Use Case Diagram Details:**

- **Account Management:** Includes functionalities for user registration, login, and profile management.
- **User Management:** Encompasses tasks related to managing user roles and permissions.
- **Project Management:** Involves creating and managing projects and their associated tasks.
- **Task Management:** Covers creating, assigning, and tracking tasks within projects.
- **Notifications:** Handles sending notifications related to task updates and deadlines.

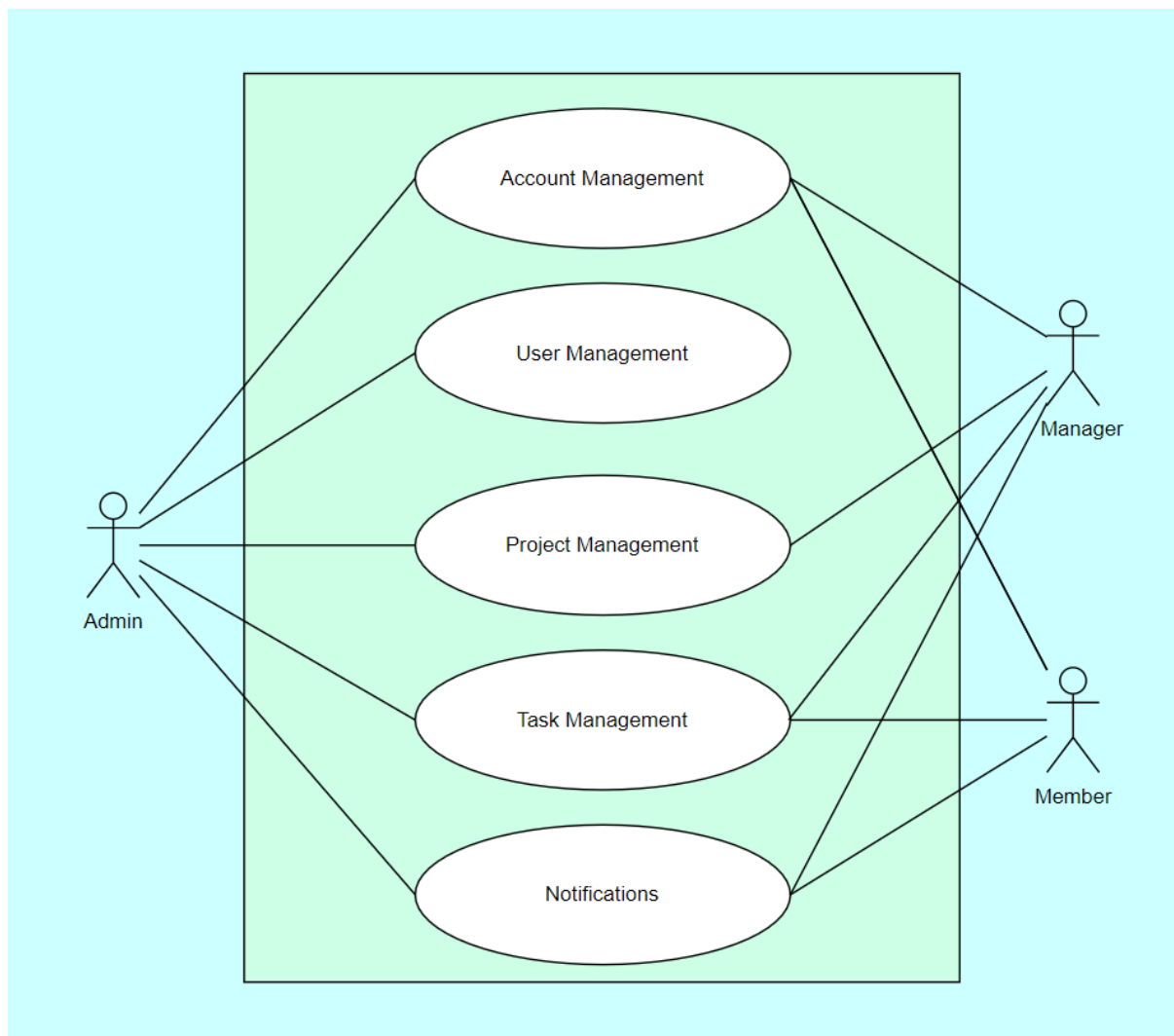


Figure 5: Use Case Diagram

TeamSync - Enhanced Remote Task Management System

Wireframe prototype:

<https://www.figma.com/proto/9VhTJwYB7tbo1pTFm27THM/CS?node-id=0-1&t=HilbDOvSsYjVdsms-1>

Register User

First name

Last name

Mobile

Email

Country

Register

Sign up with Google

Task Manager

Signin with Google

Login

Username

Password

Login

Register

Create Project +

Notifications

You got assigned

```
#mem1 completed
task
```

You got assigned

```
#mem1 completed
task
```

Add Task

User X
Time Zone

User X
Time Zone

The diagram illustrates a Kanban board layout with four columns, each representing a different stage of a task's lifecycle. The columns are separated by vertical lines. Each column has a header with a colored circle and a label. Below the headers, tasks are represented by rounded rectangles, each containing the text 'Task 1' and a trash icon.

- ToDO** (Red circle): Contains three tasks.
- OnGoing** (Yellow circle): Contains two tasks.
- Testing** (Blue circle): Contains one task.
- Completed** (Green circle): Contains one task.

At the bottom right of the diagram, there is a 'Save' button.

Save

Project name

Add Task

Task Name

Add Description

Assign People

Due

Start

End

Calendar

Save

Create Project

Name

Description

Time Zone Converter

Date

dd-mm-yyyy

From Location

To Location

Convert

Sample Data: xxxxxx

Edit Profile

Username

Password

Confirm new password

Set Timezone

Upgrade plan