Software Requirement Specification

-Project Collaboration Tool

Document ID: ASP-CMS-SRS-02.doc

Author(s): Rosary Abilash M

Version: 0.1

Date: March 25, 2024

1. Abstract

In today's dynamic work environment, effective project management and collaboration are essential for achieving organizational goals and delivering successful outcomes. The project collaboration tool aims to address these challenges by providing a centralized platform for teams to manage projects, collaborate with team members, and streamline communication. This web-based application offers a comprehensive set of features, including user management, project creation and management, document management, time tracking, and communication tools. Users can create new projects, assign tasks, track project progress, and collaborate on project documents and files in real-time. The tool also facilitates communication through real-time messaging, and group discussions, fostering seamless collaboration among team members.

2. Objective and Scope

The objective of the project collaboration tool is to streamline project management processes, foster team collaboration, and enhance productivity by providing essential features such as user management, document sharing, time tracking, and communication tools.

The scope of the product includes the following basic features:

- The Product allow users to register, login, and manage their profiles. Implement role-based access control (RBAC) to assign permissions to users.
- The Source system enables users to create new projects, set goals, and define project details. Provide project management features such as task assignment, deadlines, and progress tracking.
- Allow users to upload, share, and collaborate on project documents and files. Implement version control to track changes and revisions.
- Enable users to track time spent on tasks and projects. Generate timesheets for reporting and analysis purposes.
- Facilitate real-time messaging between team members. Provide options for video calls, file sharing, and group discussions.
- Generate reports on project progress, task completion rates, and user activity. Visualize project data through charts, graphs, and dashboards.
- Allow users to customize project settings, workflows, and notifications. Provide flexibility to adapt the tool to the specific needs of each project and team.

3. Project End Users

The project end users are the individuals or groups who will interact with and benefit from the project collaboration tool.

3.1Administrators

Responsible for managing user accounts, project permissions, and collaboration settings within the system. Have the authority to create, edit, or delete user accounts, assign roles and permissions, and configure system-wide settings. Utilize advanced features such as reporting and analytics to monitor project performance and user activity.

3.2Project Manager

Oversee and coordinate the execution of projects within the collaboration tool. Create and assign tasks, set project milestones, and monitor progress towards project goals. Use communication tools to facilitate collaboration among team members and resolve issues in real-time.

3.3Team Members

Actively participate in project execution by completing assigned tasks, collaborating with teammates, and sharing project-related documents and files. Use the collaboration tool to track time spent on tasks, update task statuses, and communicate progress to project managers and team members.

3.4Clients

Access project-related information and updates through the collaboration tool. Provide feedback, review project deliverables, and communicate with project managers and team members. Monitor project progress and timelines to ensure alignment with business objectives and expectations.

4. Module Description

4.1User Management

User Registration:

Provide a user-friendly registration form where individuals can sign up for the collaboration tool by entering essential information such as their full name, email address, and desired username.

Specify password requirements such as minimum length, inclusion of special characters, and complexity to enhance security and protect user accounts from unauthorized access.

User Login:

Implement authentication mechanisms such as password hashing to securely verify users' identities during the login process. Include a "Remember Me" feature to allow users to stay logged in across sessions, enhancing user convenience.

User Accounts Management:

Allow administrators to create new user accounts, edit existing account details (such as email address or role), and deactivate or suspend accounts when necessary (e.g., for inactive users or security reasons).

Enable administrators to assign roles and permissions to user accounts based on their responsibilities and access needs within the collaboration tool.

4.2Project Creation and Management

Create New Projects:

Offer users a dedicated form or interface where they can initiate the creation of new projects within the collaboration tool. Prompt users to input essential project information such as project name, description, and objectives to provide context and clarity to team members.

Allow users to customize project settings according to their specific requirements, including options for setting project start and end dates, defining project milestones, and specifying project goals.

Assign Tasks:

Enable project managers or team leads to create tasks within each project and assign them to individual team members based on their roles, skills, and availability.

Deadline Setting:

Allow task creators to set deadlines or due dates for each task to establish clear timelines and expectations for completion.

Task Progress Tracking:

Provide tools and interfaces for team members to update task statuses, mark tasks as completed, and provide progress updates in real-time.

Milestone Monitoring:

Implement features to define project milestones or checkpoints and track progress towards achieving these milestones to ensure that the project stays on track and aligned with its objectives.

Visual Task Boards:

Provide visual representations of tasks and their statuses using Gantt charts to enhance visibility and facilitate project monitoring and management.

4.3Document Management

Upload, Share, and Collaborate:

Provide users with a centralized repository where they can upload and store documents and files relevant to the project, including design mockups, reports, spreadsheets, presentations, and more.

Allow users to control access to documents by setting permissions and sharing settings, specifying who can view, edit, or delete documents within the collaboration tool.

Version Control:

Implement version control mechanisms to track changes and revisions made to documents over time, maintaining a history of edits and updates.

Provide users with access to a detailed revision history that includes information such as the date and time of each revision, the user who made the changes, and a summary of the modifications.

4.4Time Tracking

Track Time Spent on Tasks and Projects:

Provide users with an intuitive interface where they can easily log the time spent on individual tasks or projects. Allow users to enter time manually by specifying the start and end times or duration of each task. Allow users to add notes or descriptions to time entries to provide context and detail about the work performed, enhancing clarity and transparency.

Generate Timesheets for Reporting:

Automatically generate timesheets based on the logged time entries for each user, providing an overview of their activities and contributions.

Include detailed information in timesheets such as task descriptions, dates, durations, and project affiliations, to understand how time was allocated and spent.

4.5Communication Tools

Provide a user-friendly chat interface that allows team members to communicate in real-time. Allow users to send direct messages to each other for private conversations, fostering collaboration and information exchange. Enable users to attach files and documents directly within chat conversations, making it easy to share important resources and information.

5. Functional and Non-Functional Requirements

5.1Functional Requirements

Registration: Users shall provide necessary information such as full name, email address, and password during registration. The system shall validate user-provided information and ensure that email addresses are unique.

Authentication: Users shall be authenticated securely using methods such as password hashing and salting.

Administrator Controls: Administrators shall have the ability to create, edit, and delete user accounts.

Project Details: Users shall specify project details including name, description, start date, end date, and objectives.

Task Assignment: Project managers shall assign tasks to team members, specifying task descriptions, deadlines, priorities, and dependencies.

Progress Tracking: Users shall be able to update task statuses, mark tasks as complete, and provide progress updates.

Upload and Share: Users shall upload documents and files to the system, specifying relevant project folders or categories.

Collaboration: Users shall collaborate on documents in real-time, with simultaneous editing capabilities and version history tracking.

Access Control: The system shall enforce access control policies, restricting access to sensitive documents based on user roles and permissions.

Time Entry: Users shall log time spent on tasks manually, entering start and end times or durations.

Timesheet Generation: The system shall generate timesheets automatically based on logged time entries, grouping entries by project, task, or user.

Real-Time Messaging: Users shall send and receive messages in real-time through a chat interface, supporting individual and group conversations.

5.2Non-Functional Requirements

The system shall respond to user interactions within milliseconds, ensuring a smooth and responsive user experience. The system shall support concurrent access by multiple users without performance degradation, even during peak usage periods. The system shall be scalable to accommodate increasing numbers of users, projects, and documents without compromising performance. The system shall implement automated backup mechanisms to ensure data integrity and recovery in the event of system failures or data loss incidents. The system shall be resilient to network outages and hardware failures, with failover mechanisms in place to minimize service disruptions.

6. Lower-Level Design

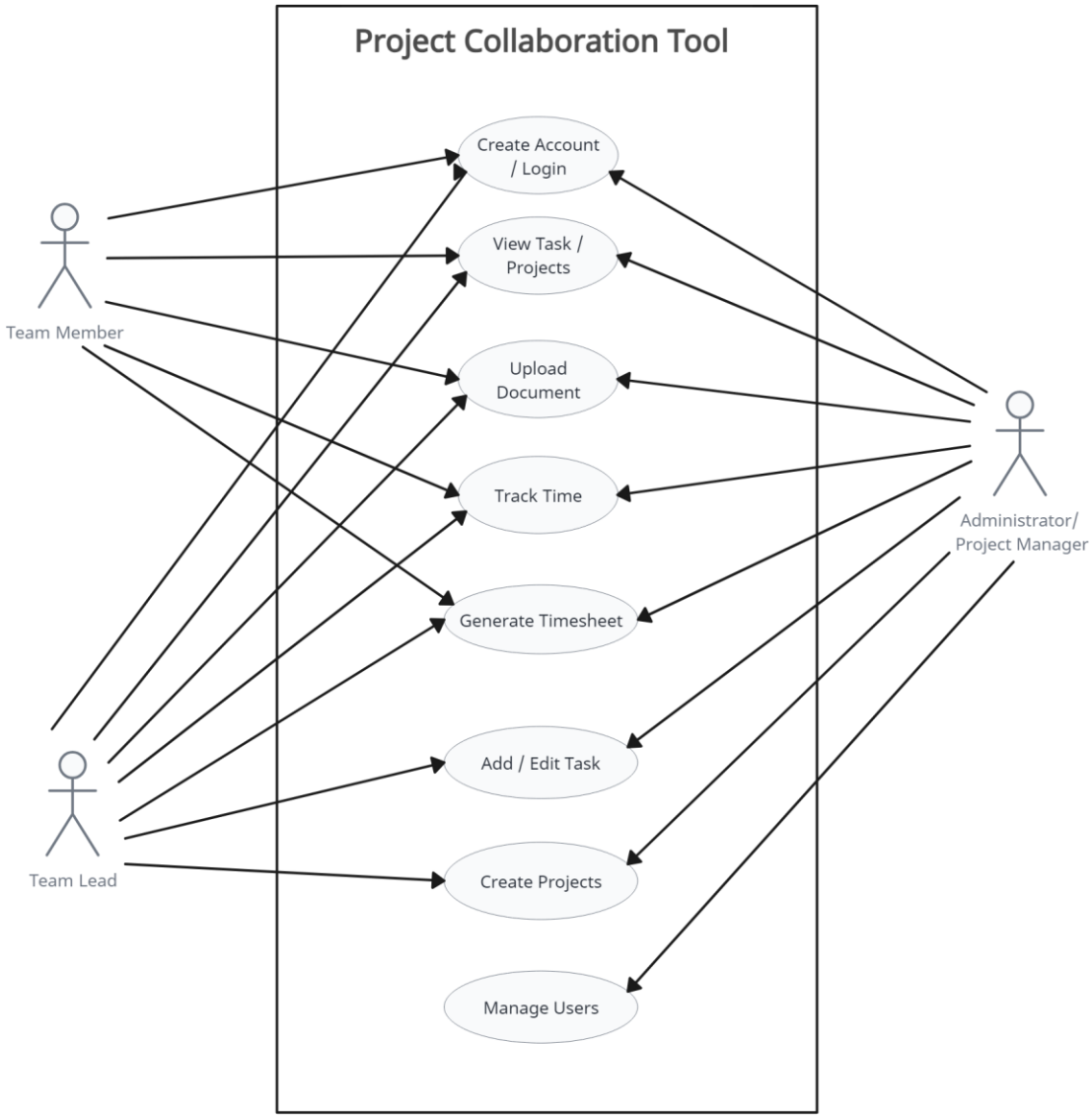
The lower-level design of the project collaboration tool encompasses detailed architectural and implementation considerations for both the frontend and backend components. On the frontend, a component-based architecture is adopted, for asynchronous data retrieval from the backend. The backend architecture follows a modular approach, with separate modules for user management, project management, and document management, among others. Middleware functions handle request processing, including authentication, authorization, and error handling, to facilitate database interactions and ensure data consistency. The database schema is designed to represent entities like users, projects, and tasks, with normalized tables and appropriate relationships. Authentication and authorization mechanisms are implemented using session cookies for user authentication, while role-based access control (RBAC) allows fine-grained control over user permissions. Modules like project management and document management include controllers, routes, and service functions to handle CRUD operations, business logic, and validation. Real-time messaging are enabled through WebSocket connections and WebRTC technology, with support for individual and group chats and peer-to-peer communication. Time tracking and reporting functionalities leverage dynamic timesheet generation and SQL queries or aggregation pipelines for insights into project timelines and resource utilization.

7. Higher Level Design

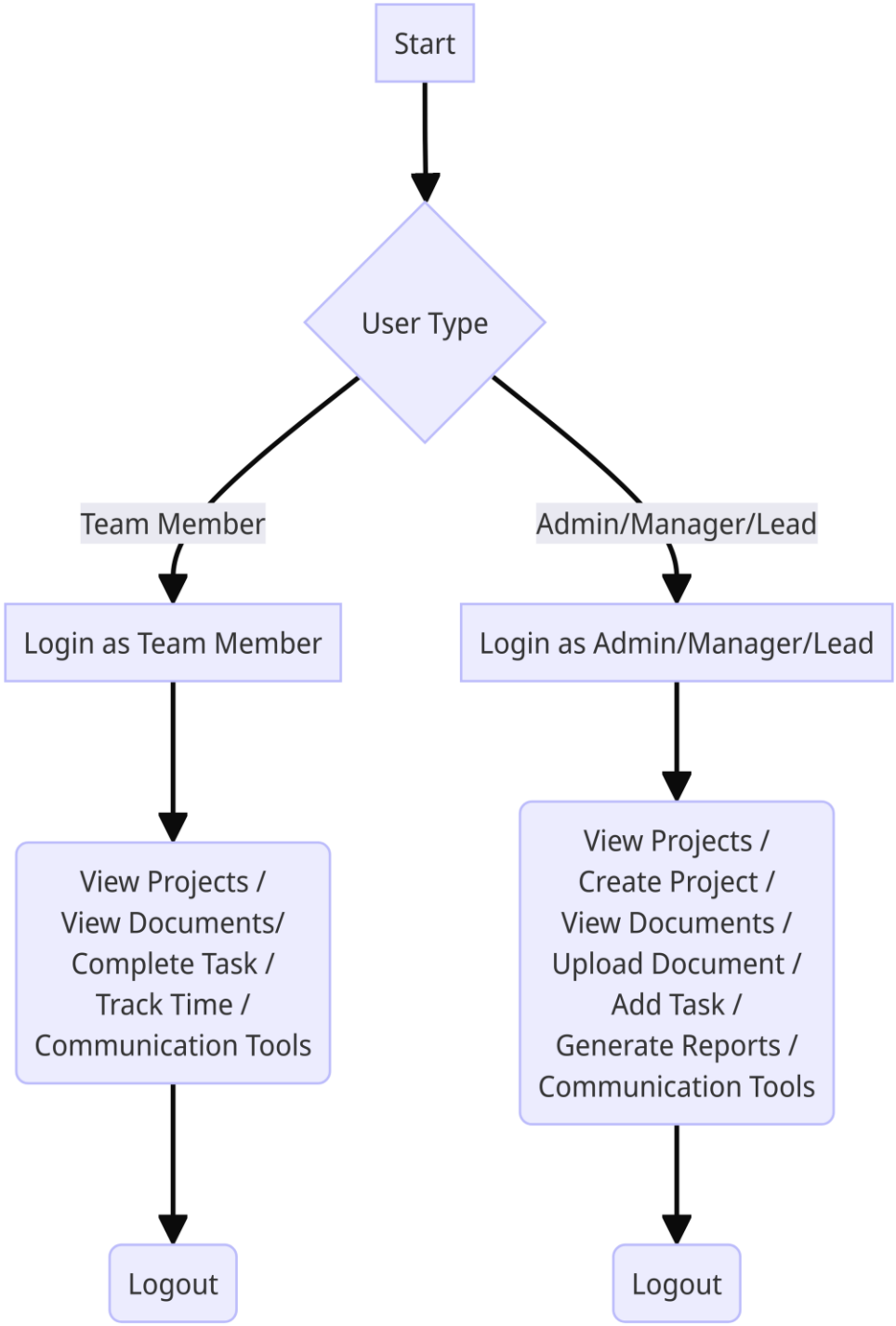
The higher-level design of the project collaboration tool is structured around a client-server architecture, leveraging modern web technologies for the frontend and a scalable backend server. The frontend is developed using frameworks like HTML, CSS, React.js, Angular.js, providing a dynamic user interface for seamless interaction. On the backend, a robust technology stack such as Node.js, Python Django, or Ruby on Rails is utilized to create RESTful APIs for communication with the frontend. Data is stored in a relational database management system (RDBMS) such as PostgreSQL or MySQL, with a well-defined schema encompassing tables for user accounts, projects, tasks, documents, and time entries. User authentication is ensured through industry-standard protocols like OAuth 2.0 or JSON Web Tokens (JWT), coupled with role-based access control (RBAC) mechanisms to manage permissions effectively. Modules for project management, document management, communication tools, time tracking, and reporting are seamlessly integrated, employing algorithms such as critical path method (CPM) for task management. Real-time messaging, and group discussions are facilitated using WebRTC or WebSocket protocols, while notifications and alerts keep users informed of relevant updates. Scalability, performance, and security are prioritized, with mechanisms for horizontal scaling, caching, encryption, and regular security audits in place to ensure a robust and reliable system architecture.

8. Diagrams

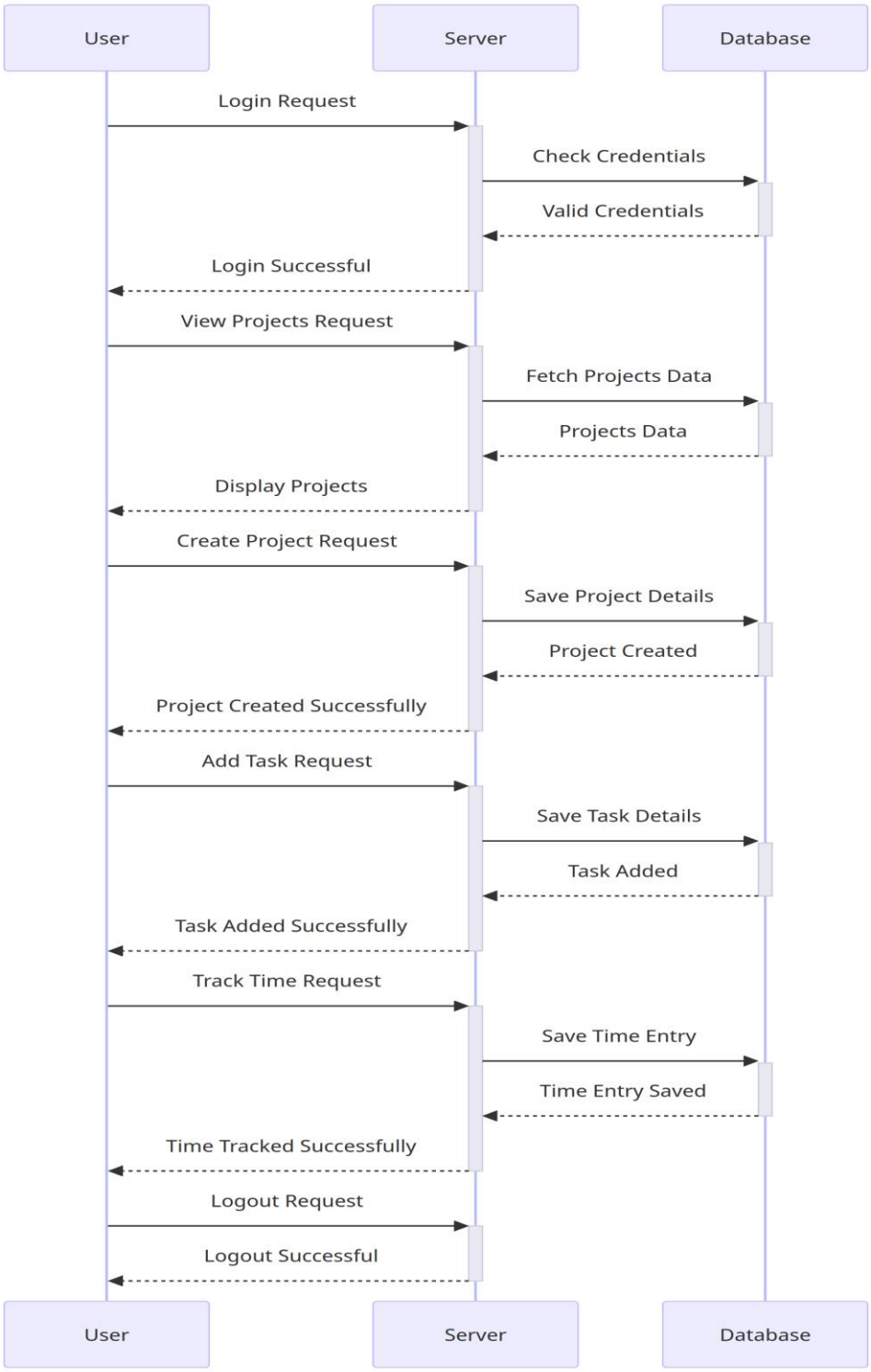
8.1 Use Case Diagram



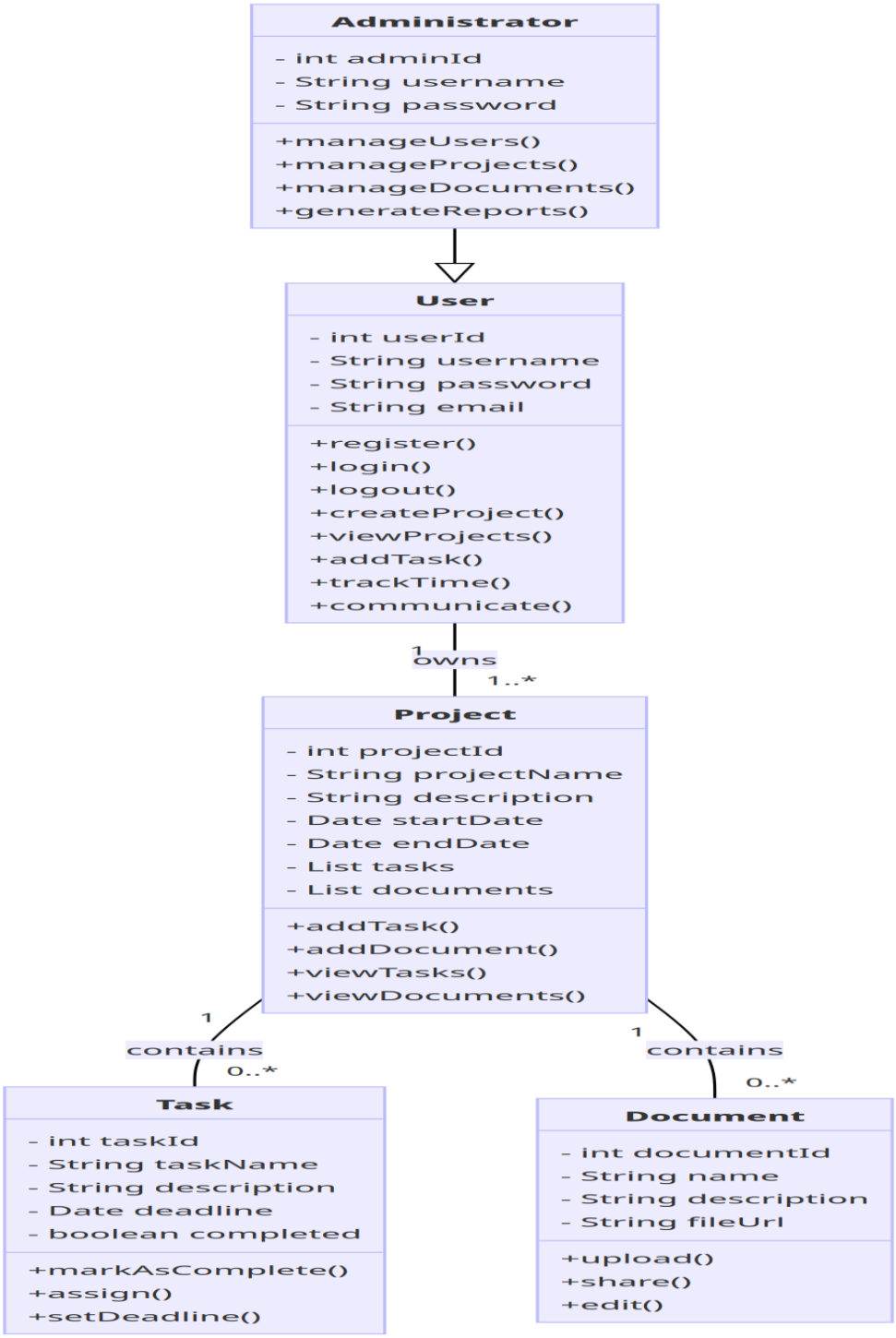
8.2Flow Chart



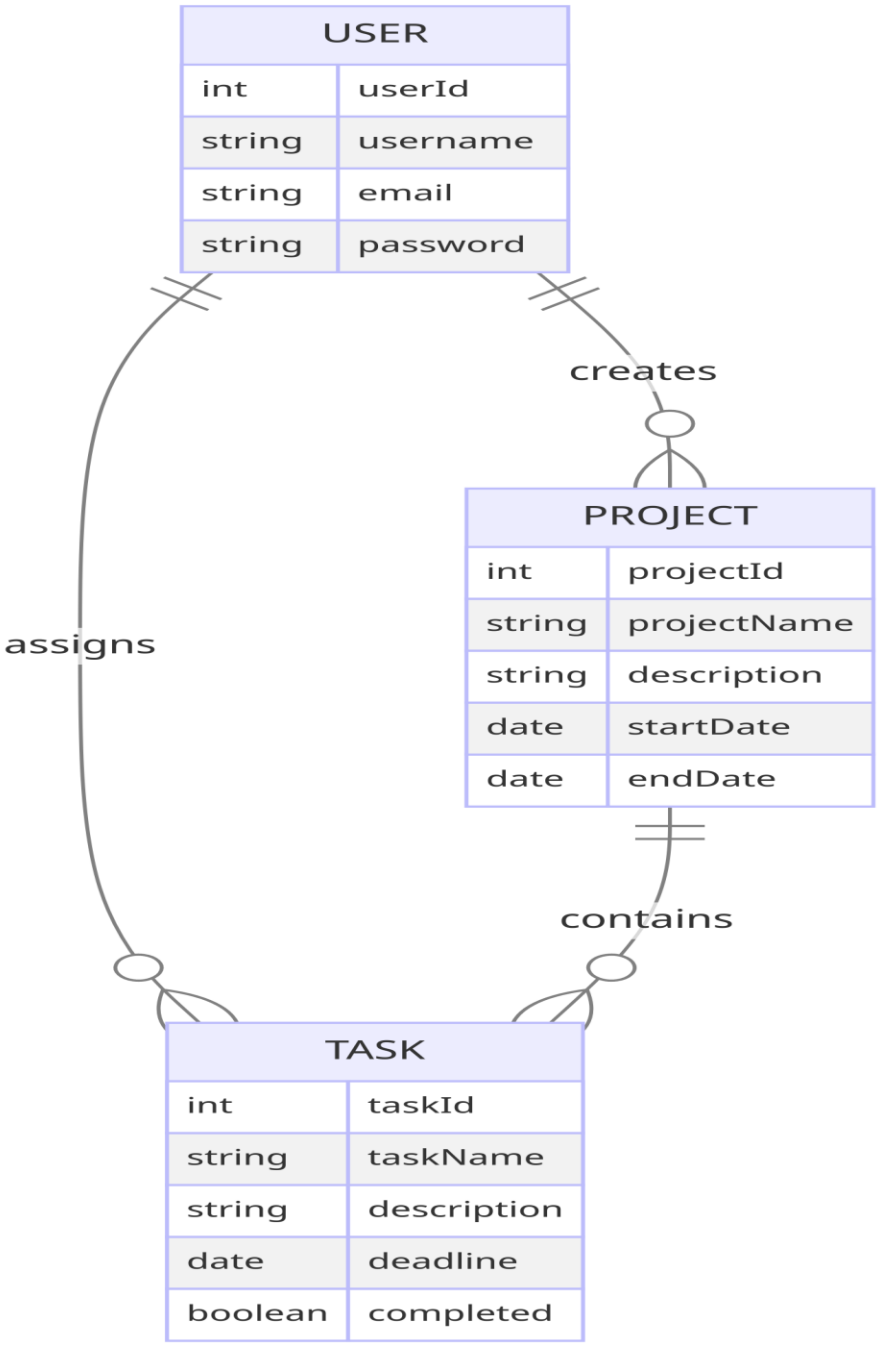
8.3Sequence Diagram



8.4Class Diagram



9. ER Diagram


USER
int userId
string username
string email
string password
creates
PROJECT
int projectId
string projectName
string description
date startDate
date endDate
assigns
contains
TASK
int taskId
string taskName
string description
date deadline
boolean completed

10. Test Cases

The Test Plan is derived from the Requirements, Functional Specifications and Detailed Design Specification.

Test Case	Test Purpose	Test Condition	Expected Outcome	Actual Result
User Registration	To verify that a user can successfully register in the system.	User provides valid registration details (username, email, password).	User account is created successfully, and the user is redirected to the login page with a success message.	User account is created, and the system displays a success message prompting the user to login.
User Login	To verify that a registered user can log in to the system.	User provides valid login credentials (username/email and password).	User is authenticated and redirected to the main dashboard.	User is successfully authenticated and directed to the main dashboard.
Create Project	To verify that a user can create a new project.	User is logged in and provides valid project details (project name, description, start date, end date).	A new project is created and added to the user's list of projects.	The project is successfully created and appears in the user's project list.

Add Task to Project	To verify that a user can add a task to a project.	User is logged in and navigates to a specific project, then provides valid task details (task name, description, deadline).	The task is added to the project's task list.	The task is successfully added to the project's task list.
Track Time on Task	To verify that a user can track time spent on a task.	User is logged in and selects a task to track time for, then provides valid time tracking details (start time, end time).	The system records the time spent on the task and updates the time tracking logs.	Time spent on the task is accurately recorded and reflected in the time tracking logs.
Upload Document to Project	To verify that a user can upload a document to a project.	User is logged in and navigates to a specific project, then provides a valid document file to upload.	The document is successfully uploaded and appears in the project's document list.	The document is successfully uploaded and visible in the project's document list.
Manage User Accounts	To verify that an administrator can manage user accounts.	Administrator is logged in and accesses the user management section, then performs actions such as creating, editing, or deactivating user accounts.	Administrator can successfully create, edit, or deactivate user accounts as needed.	Administrator is able to perform user account management actions without any errors