Introduction to Software Engineering

Requirements Analysis

The student team is required to complete the Software Requirements Specification (SRS) document for the assigned course project, following the attached template.

Software Engineering Department
Faculty of Information and Technology
University of Science

Table of Contents

Software Requirements Specification

Objectives

This document focus on the following topics:

- ✓ Complete the Software Requirements Specification (SRS) document with the following contents:
  - Elaborate on the Problem Statement
  - Overview of Requirements (Functional and Non-Functional), Stakeholders
  - Use Case Model
  - Use Case Specifications
  - Create Prototype and Mockup Diagrams of the System Interface
- ✓ Đọc hiểu tài liệu phân tích yêu cầu.

1 Member Contribution Assessment

ID	Name	Contribution (%)	Signature
23127086	Huỳnh Sĩ Luân	100%	
23127148	Ân Tiến Nguyên An	100%	
23127212	Nguyễn Quang Đăng Khoa	100%	
23127280	Nguyễn Hiền Tuấn Anh	100%	
23127442	Trầm Hữu Nhân	100%	

2 Problem Statement

2.1 Business description

The Online Project Management System is a platform designed to help individuals and organizations efficiently plan, assign, monitor, and collaborate on work. The system allows users to create and manage multiple projects, each consisting of various tasks that can be assigned to specific team members. Its primary goal is to help teams optimize their workflow, track progress in real time, and ensure projects are completed on schedule. Through intuitive interfaces and collaborative tools, the system enhances coordination, transparency, and productivity across all stages of project development.

- **User Management**

The online project management system allows users to create accounts, log in, and participate in multiple projects simultaneously. Each user account contains personal information such as name, email, password, profile image, and assigned role within the organization. Users can update their profiles, change passwords, and enable **Two-Factor Authentication (2FA)** to enhance security. The system also supports notification management so users can receive the latest updates whenever a new task, comment, or status change occurs in any project. Additionally, the system must support **Role-Based Access Control (RBAC).** Users are granted different access permissions based on their role in each project:

  - **Administrator (Admin):** Has global access to system settings and user oversight.
  - **Project Manager:** Has write-access to create projects, assign resources, and modify workflows.
  - **Team Member:** Has restricted access to view tasks and update their own progress.

- **Project Creation and Management**

Users can create new projects by entering detailed information such as name, description, start and end dates, and a list of project members. Once a project is created, the system automatically generates a **Kanban Board** with default columns like **"To Do," "In Progress," and "Done."** The Project Manager can add or remove members, assign permissions for each participant, and track overall progress of the team. Projects can be edited, paused, or closed upon completion. The system also enables users to search for projects quickly by name, manager, or current status. This helps users manage multiple projects efficiently and collaborate more effectively across teams.

- **Task Management**

Within each project, members can create and manage tasks of various types. Each task includes details such as title, description, assignee, deadline, priority level, and current status. Users can attach files, images, or create sub-checklists within tasks for better organization. The system supports direct commenting inside each task, allowing members to @tag teammates for discussion and quick updates. Tasks are visually represented on the Kanban board and can be dragged and dropped between columns to change their status. Users can also filter and search tasks by name, due date, assignee, or priority level to ensure that every project milestone is properly tracked and managed.

- **Progress and Performance Tracking**

The system provides real-time progress tracking to help Project Managers and Team Members easily monitor ongoing activities. Each task displays a progress bar (%) that updates automatically based on its completion status or checklist items. Managers can view **Gantt charts** to observe the overall project timeline, milestones, and task dependencies. Additionally, the system supports **Time tracking** for each member, recording working hours and providing individual **performance analytics** weekly or monthly. When a task nears or exceeds its deadline, the system will automatically **send alerts** to the responsible member to ensure timely completion and maintain project consistency.

- **Reporting and Notifications**

Another key feature of the system is its smart reporting and notification capability. Every action such as creating tasks, changing statuses, or adding comments is recorded in the **Activity Log** for transparency. Users receive real-time notifications via **web sockets and email integration (SMTP)** whenever significant updates occur. The system also generates comprehensive reports summarizing project progress, completed tasks, ongoing tasks, and individual member performance. These reports can be exported as **PDF or Excel/CSV** files for presentation or archival purposes. With detailed reporting and instant notifications, users can maintain full control over project workflows and make timely management decisions.

- **Security and Subscription Plans**

To ensure data security, all system information is encrypted through **HTTPS protocols** and api requests are verified using **JSON Web Tokens (JWT)**. Users can activate **two-factor authentication (2FA)** to strengthen login protection. Passwords are never stored in plain text; they are hashed using strong algorithms (e.g., BCrypt). All project data, attachments, and activity logs are regularly backed up to prevent loss. The platform also provides flexible subscription plans: the **Free plan** targets individuals or small teams (limited to 3 projects), while the **Premium plan** supports organizations with unlimited projects, members, and advanced

features such as Gantt charts, advanced reports, and performance analytics. Subscription upgrades are processed online via secure payment gateways like Momo, PayPal, or credit cards.

2.2 Operating Environment

To ensure broad accessibility and reliability, the system is designed to operate within the following environment:

- **Client-Side (User Interface)**
  - **Web Browsers:** The application is accessible via standard web browsers supporting **HTML5, CSS3, and JavaScript (ES6+)**. It is compatible with Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
  - **Devices Compatibility:** The interface uses **Responsive Web Design (RWD)** to adapt to Desktop, Tablet, and Mobile viewports.
- **Server-Side (Backend Infrastructure)**
  - **Web Server / Reverse Proxy:** The application logic is hosted on a web server such as **Apache HTTP Server** or **Nginx**.
  - **Application Server:** The application logic runs on **Node.js** (LTS Version), utilizing its non-blocking, event-driven architecture to handle high concurrency.
  - **Database Server:** The system requires a **MongoDB** instance (Version 6.0+) configured as a Replica Set to support ACID transactions and high availability.
  - **Third-Party Services:** The environment relies on external availability of:
    - **SMTP Server** (e.g., SendGrid, Mailgun) for email notifications.
    - **Payment Gateway APIs** (Momo, PayPal) for subscription processing.
- **Network Requirements**
  - **Connectivity:** A persistent high-speed internet connection (Broadband, 4G, or 5G) is required for users to access the platform.
- **Firewall/Port Configuration:**
  - **Port 443 (HTTPS):** Must be open for secure API requests.
  - **WebSocket Protocol (WSS):** The network must allow persistent WebSocket connections for real-time updates (avoiding aggressive proxy filtering).

2.3 Design and Implementation Constraints

The development and deployment of the system must adhere to specific technical and organizational constraints:

- **Technological Constraints**
  - **Architectural Pattern:** The system follows the standard 3-Tier Architecture pattern to ensure separation of concerns:

        - **Presentation Layer: A ReactJS Single Page Application (SPA)** built with **Vite and TailwindCSS**. It handles Client-Side Rendering (CSR) and consumes data via RESTful APIs.
        - **Business Layer: A Node.js server** using the Express.js framework. It processes **API routing, middleware validation, and business rules**. It must utilize **Socket.io** to push real-time updates (Event-Driven behavior) to clients for the Kanban board and Chat.
        - **Data Layer: A MongoDB database** that stores data in **JSON-like documents**, using Mongoose for schema enforcement. The system utilizes a **document-oriented model** (via **Mongoose ORM**) to handle flexible data structures for Projects and Tasks, while ensuring high write speeds for real-time collaboration.
    - **Asset Management & Storage:**
        - **Separation of Concerns:** Large binary files (user avatars, task attachments, PDFs) must **not** be stored directly in the MongoDB database.
        - **Storage Strategy:** The system must utilize an **Object Storage** strategy (e.g., AWS S3, Cloudinary, or a dedicated local uploads directory). The database will only store the file metadata and a reference URL.
    - **Database Performance:**
        - **Indexing:** To ensure fast retrieval of tasks, the database must implement **Compound Indices** on frequently queried fields (e.g., ProjectID + Status) and **Text Indexes** for content searching.
- **Security and Authentication Standards**
    - **Data Transmission:** All data transmission must be encrypted via **HTTPS** protocols (TLS 1.2+).
    - **Authentication:** The system must implement stateless authentication using **JSON Web Tokens (JWT)**.
        - ***Access Token****:* Short lifespan (e.g., 15 mins), stored in memory/client.
        - ***Refresh Token****:* Long lifespan (e.g., 7 days), stored securely in an **HTTP-Only, Secure Cookie** to prevent XSS attacks.
    - **Access Control:** The system must enforce **Role-Based Access Control (RBAC)** at two levels:
        - **API Route Level:** Middleware to block unauthorized endpoints.
        - **Database Level:** Checks to ensure a user belongs to the project they are trying to edit, require a role to do the specific actions.

  - **Input Validation:** Mandatory validation and sanitization of all API inputs must be implemented (using libraries like Joi or express-validator) to prevent **NoSQL Injection** and **Cross-Site Scripting (XSS)**.
  - **Sensitive Data: P**asswords must never be stored in plain text. They must be salted and hashed using a strong algorithm like **BCrypt**.
- **Business Logic Constraints**
  - **Subscription Enforcement:**
  - **Resource Limits:** The system must enforce hard limits based on the user's plan. *Constraint:* Free Plan users attempting to create a 4th project must receive a 403 Forbidden error.
  - **Payment Compliance:**
    - **PCI-DSS:** The system must **not** store credit card numbers locally. All payment processing must be offloaded to secure third-party APIs (PayPal, Momo).
- **Operational and Usability Constraints**
  - **User Interface:** The UI must be designed using standard **UX heuristics**, featuring an intuitive **drag-and-drop** interface for task management with minimal navigation depth.
  - **Version Control:** Source code must be managed using **Git**, adhering to conventional commit message standards.
  - **Reliability:** The system architecture must support **automated daily backups** of the database to ensure data recovery capability.
  - **Documentation Standards:**
    - The requirements specification adheres to **IEEE 830** standards.
    - Codebase must be documented using **JSDoc**.
    - API endpoints must be defined and testable using **Swagger/OpenAPI**.

3 Requirements Overview

3.1 Stakeholders

STT	Stakeholder	Description
1	Project Manager	The primary user is responsible for initiating projects, adding team members, assigning roles, and overseeing the overall workflow.
2	Team Member	Users assigned specific projects. They work on tasks, move them across the Kanban board, comment, upload attachments, and log their working hours.
3	Individual Users	Create accounts, manage tasks, track personal progress.
4	Organization / Premium Subscribers	Uses paid subscription plan for full system features. Typically, it includes multiple teams working collaboratively.
5	System Administrator	The super-admin responsible for maintaining the platform, managing the subscription tiers, and ensuring system uptime and security.
6	Payment Gateway	External systems (Momo, PayPal, Credit Card providers) that process transactions for Premium subscription upgrades.
7	Email/Notification Services	Sends system alerts, 2FA codes, and project notifications. Integrates with SMTP or cloud-based messaging services.

3.2 Requirements

3.2.1. Functional Requirements Specification

A. User Management

- FR-1: The system shall allow users to register via email and password.
- FR-2: The system allow users to log in using email and password.
- FR-3: The system shall support two-factor authentication (2FA) for login security.

- FR-4: The system shall allow users to update their profile information and change passwords.
- FR-5: The system shall assign roles (Admin, Project Manager, Team Member) with different permission levels.
- FR-6: The system allow users to manage notifications and choose which alerts to receive.
- FR-7: The system shall allow Admin to manage user accounts (activate, deactivate, or delete).

B. Project Management

- FR-8: The system allow users to create new projects with name, description, start/end dates, and member list.
- FR-9: The system shall automatically generate a Kanban board for each project.
- FR-10: The system shall allow Project Manager to add, remove, or update project members.
- FR-11: The system allow users to edit project details or close a project when completed.
- FR-12: The system shall allow users to search projects by name, manager, or status.

C. Task Management (Kanban)

- FR-13: Members shall be able to create tasks containing title, description, assignee, deadline, priority, and status.
- FR-14: The system shall allow tasks to include file attachments, images, and sub-checklists.
- FR-15: Users shall be able to comment on tasks and tag (@mention) other team members.
- FR-16: The system shall support a drag-and-drop interface to move tasks between Kanban columns.
- FR-17: The system shall allow users to filter and search tasks by name, priority, due date, or assignee.

D. Progress & Performance Tracking

- FR-18: The system shall display a progress bar (%) for each task based on checklist completion or status.
- FR-19: (Premium Feature) The system shall generate Gantt charts to visualize timelines and dependencies.
- FR-20: The system shall allow users to log working hours for specific tasks.

- FR-21: (Premium Feature) The system shall provide individual performance analytics (weekly/monthly).
- FR-22: The system must send automatic alerts to assignees when a task deadline is approaching or overdue.

E. Reporting & Notifications

- FR-23: The system shall maintain an Activity Log recording all actions (Task creation, status changes, comments).
- FR-24: The system must send real-time notifications via Web and Email for significant updates.
- FR-25: (Premium Feature) The system shall generate comprehensive reports (Project progress, Completed/Ongoing tasks) exportable to PDF or Excel.

F. Security & Subscription Management

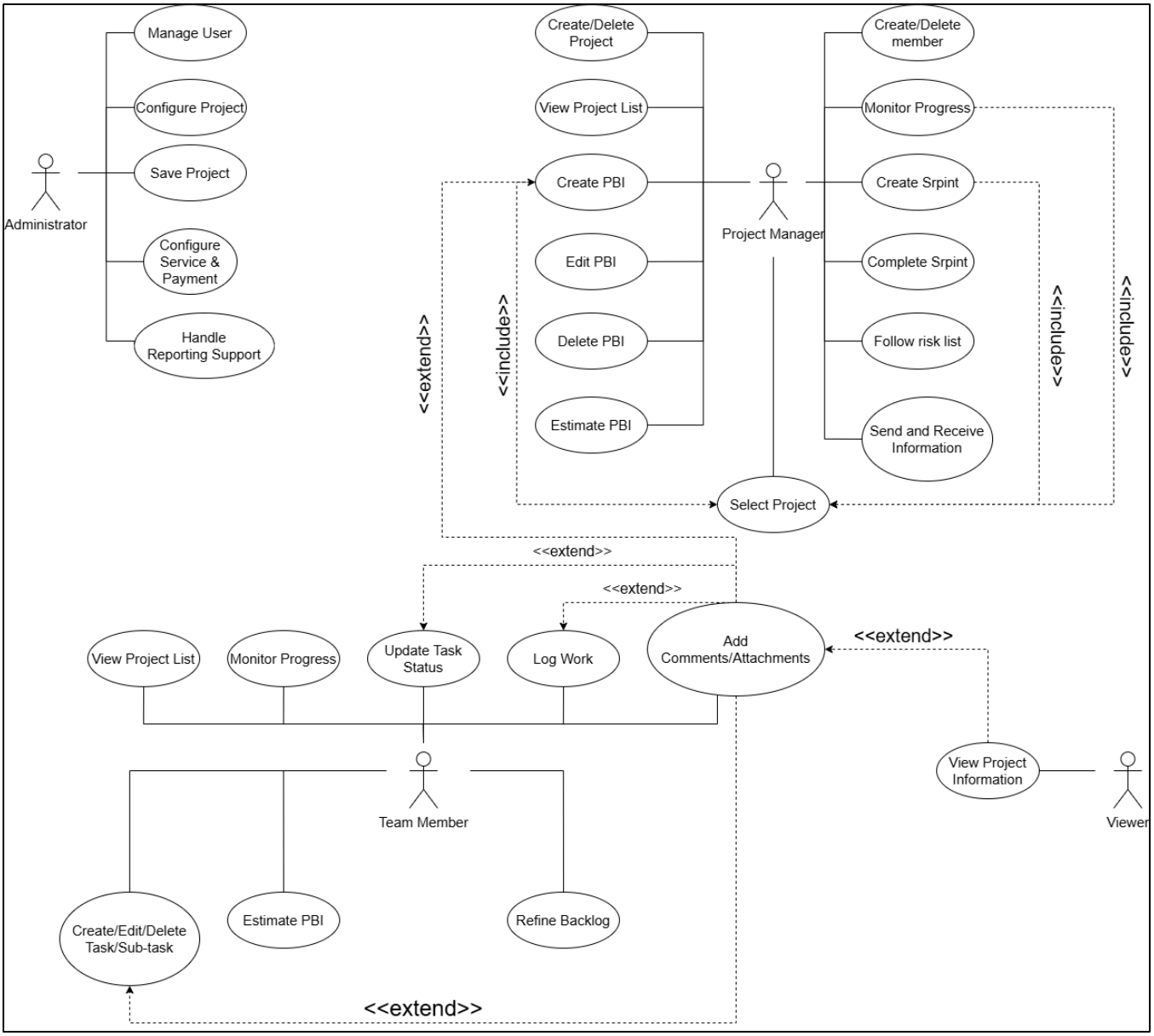
- FR-28: The system shall allow users to choose between Free and Premium plans.
- FR-29: The system shall process payments via Momo, PayPal, or credit cards.
- FR-30: The system shall limit Free plan users to 3 active projects.
- FR-31: The system shall provide Premium features such as unlimited projects, advanced analytics, and Gantt charts.

3.2.2. Non-Functional Requirements Specification

- NFR-01 (Security): All data transmission must be encrypted using HTTPS protocols.
- NFR-02 (Security): User authentication must be stateless and verified using JWT (JSON Web Tokens).
- NFR-03 (Reliability): The system must perform regular automatic backups of all project data, attachments, and activity logs.
- NFR-04 (Performance): The system must support real-time updates (via WebSocket or similar technology) so that changes to the Kanban board are instantly visible to all active users without refreshing.
- NFR-05 (Usability): The user interface must be intuitive, supporting drag-and-drop interactions for task management.
- NFR-06 (Scalability): The database design must support the specific limitations of Free plans vs. the unlimited nature of Premium plans without performance degradation.
- NFR-07 (Compatibility): The reports generated must be compatible with standard PDF readers and Microsoft Excel.

4 Requirements Analysis

4.1 Use Case model



4.2 Use Case Specification

4.2.1. Use Case 1

Use case ID	*UC-ADMIN-001*
Use Case	Manage User
Brief Description	Admin creates, updates, or deactivates user accounts in the system.
Actor	Administrator
Pre-Condition	Admin is logged in.
Result	User information is updated in the database.
Main Scenario	1. Admin navigates to User Management. 2. System lists existing users. 3. Admin selects "Create New" or selects a user to "Edit/Delete". 4. Admin inputs/modifies details and roles. 5. Admin clicks Save. 6. System validates and persists data.
Alternatice Scenarios	Email already exists: System shows a "Duplicate Email" error.
Non-Functional Constraints	Passwords must be encrypted.

Use case ID	*UC-ADMIN-002*
Use Case	Configure Project
Brief Description	Admin creates, updates, Setup technical parameters, workflows, and global settings for a project. or deactivates user accounts in the system.
Actor	Administrator
Pre-Condition	Project must exist.
Result	Project configuration is applied.
Main Scenario	1. Admin selects the target project.

	2. Admin modifies settings (Workflow, Task Types, Permissions). 3. Admin saves configuration. 4. System applies changes to the project scope.
Alternatice Scenarios	N/A
Non-Functional Constraints	Data consistency must be maintained.

Use case ID	***UC-ADMIN-003***
Use Case	Handle Reporting Support
Brief Description	The Administrator reviews system reports regarding errors/issues or handles support tickets submitted by users.
Actor	Administrator
Pre-Condition	There are pending support tickets or system reports generated.
Result	The issue is reviewed, and a resolution or response is recorded.
Main Scenario	1. Admin opens the "Support & Reporting" center. 2. System displays a list of reported issues or tickets. 3. Admin selects a specific item to review details. 4. Admin takes action (e.g., Send Reply, Mark as Resolved, Escalate to Dev Team). 5. System updates the ticket status and notifies the reporter.
Alternatice Scenarios	N/A
Non-Functional Constraints	Response time for critical system reports should be prioritized in the UI.

Use case ID	***UC-ADMIN-004***
Use Case	Configure Service & Payment
Brief Description	The Administrator manages the system's subscription plans, billing

	information, and integration with third-party services.
Actor	Administrator
Pre-Condition	Admin has valid payment credentials or API keys.
Result	Subscription plan is updated or external services are connected.
Main Scenario	1. Admin accesses the "Billing & Services" dashboard. 2. Admin selects an action (e.g., Upgrade Plan, Update Credit Card, Configure API Key). 3. Admin enters the required financial or technical details. 4. Admin submits the form. 5. System verifies with the external payment gateway or service provider. 6. System updates the service status.
Alternatice Scenarios	Payment Failed: Gateway rejects card -> System notifies Admin and retains current plan status.
Non-Functional Constraints	All payment information must be processed via secure channels (SSL/TLS) and comply with PCI-DSS standards.

Use case ID	***UC-ADMIN-005***
Use Case	Save Project
Brief Description	The Administrator manually triggers a save of the project state, often used for creating backups, snapshots, or archiving a finished project.
Actor	Administrator
Pre-Condition	Project is active and selected.
Result	A snapshot of the project data is saved or the project is archived.
Main Scenario	1. Admin navigates to the Project Maintenance area. 2. Admin selects "Save Snapshot" or "Archive Project". 3. System prompts for confirmation or a version name. 4. Admin confirms. 5. System processes the data and stores a secure copy/archive.
Alternatice Scenarios	Storage Full: If system storage is insufficient -> Display error "Insufficient storage space".

Non-Functional Constraints	Data integrity must be guaranteed during the save process.

Use case ID	***UC-PM-001***
Use Case	View Project List
Brief Description	The Project Manager views a list of all projects they are assigned to or own.
Actor	Project Manager
Pre-Condition	User is logged in.
Result	A list of projects is displayed.
Main Scenario	1. PM navigates to the "Projects" dashboard. 2. System retrieves projects linked to the PM's account. 3. System displays the list (Name, Status, Role).
Alternatice Scenarios	No Projects: System displays "No projects found" and prompts to create one.
Non-Functional Constraints	Pagination required if projects > 20.

Use case ID	***UC-PM-002***
Use Case	Create/Delete Project
Brief Description	The PM initializes a new project workspace or removes an obsolete one.
Actor	Project Manager
Pre-Condition	User is logged in.
Result	Project is created or deleted.
Main Scenario	**(Create)** 1. PM clicks "New Project". 2. PM enters Name, Description, Key. 3. PM clicks Save. 4. System creates the project.

Alternatice Scenarios	**(Delete)** 1. PM selects a project in the list. 2. PM clicks "Delete". 3. System asks for confirmation. 4. PM confirms. 5. System soft-deletes the project.
Non-Functional Constraints	Project Key must be unique.

Use case ID	***UC-PM-003***
Use Case	Select Project
Brief Description	The PM selects a specific project context to work within. This is an <<include>> dependency for many other cases. (Create PBI; Create Sprint).
Actor	Project Manager
Pre-Condition	"View Project List" has been executed.
Result	The specific project workspace is loaded.
Main Scenario	1. PM clicks on a specific project card/link from the list. 2. System loads project-specific data (Backlog, Sprints, Members).
Alternatice Scenarios	**(Delete)** 1. PM selects a project in the list. 2. PM clicks "Delete". 3. System asks for confirmation. 4. PM confirms. 5. System soft-deletes the project.
Non-Functional Constraints	Project Key must be unique.

Use case ID	***UC-PM-004***
Use Case	**Create/Delete Member**

Brief Description	The PM manages the project team by adding new users or removing existing ones.
Actor	Project Manager
Pre-Condition	A project is selected.
Result	Member list is updated.
Main Scenario	**(Add)** 1. PM goes to "Team Members". 2. PM clicks "Add Member". 3. PM searches by email and selects role. 4. System adds user to project.
Alternatice Scenarios	**(Remove)** 1. PM finds a member in the list. 2. PM clicks "Remove". 3. System revokes project access for that user.
Non-Functional Constraints	N/A

Use case ID	***UC-PM-005***
Use Case	Create PBI
Brief Description	PM adds a new item (Story, Bug, Task) to the Product Backlog.
Actor	Project Manager
Pre-Condition	Project is selected (<<include>> Select Project).
Result	New PBI is added to the bottom of the backlog.
Main Scenario	1. PM navigates to "Backlog". 2. PM clicks "Create Item". 3. PM enters Summary and Type. 4. System saves the item.
Alternatice Scenarios	N/A

Non-Functional Constraints	N/A

Use case ID	***UC-PM-006***
Use Case	Edit PBI
Brief Description	PM modifies the details of an existing backlog item.
Actor	Project Manager
Pre-Condition	PBI exists in the backlog.
Result	PBI details are updated.
Main Scenario	1. PM clicks on a PBI. 2. System opens detail view. 3. PM updates Description, Priority, or Acceptance Criteria. 4. System saves changes.
Alternatice Scenarios	N/A
Non-Functional Constraints	Concurrent editing handling (optimistic locking).

Use case ID	***UC-PM-007***
Use Case	Delete PBI
Brief Description	PM removes an item from the backlog.
Actor	Project Manager
Pre-Condition	PBI exists in the backlog.
Result	PBI is removed.
Main Scenario	1. PM selects a PBI. 2. PM selects "Delete" option. 3. System prompts confirmation. 4. PM confirms.

	5. System removes PBI.
Alternatice Scenarios	N/A
Non-Functional Constraints	N/A

Use case ID	***UC-PM-008***
Use Case	Estimate PBI
Brief Description	PM assigns complexity points or time estimates to a PBI.
Actor	Project Manager
Pre-Condition	PBI exists in the backlog.
Result	PBI has an estimate value (e.g., Story Points).
Main Scenario	1. PM opens PBI details. 2. PM clicks on "Estimate" field. 3. PM inputs value (e.g., 5 SP). 4. System saves value.
Alternatice Scenarios	N/A
Non-Functional Constraints	Only numeric values or valid Fibonacci sequence allowed.

Use case ID	***UC-PM-009***
Use Case	Create Sprint
Brief Description	PM defines a new sprint cycle.
Actor	Project Manager
Pre-Condition	Project is selected (<<include>> Select Project).
Result	An empty Sprint container is created.
Main Scenario	1. PM clicks "Create Sprint" on the Backlog board.

	2. System creates "Sprint [N]". 3. PM edits Sprint dates and Goal. 4. PM drags PBIs into the Sprint.
Alternatice Scenarios	N/A
Non-Functional Constraints	N/A

Use case ID	***UC-PM-0010***
Use Case	Complete Sprint
Brief Description	PM closes the current active sprint.
Actor	Project Manager
Pre-Condition	Active sprint exists and end date has arrived (or PM forces close).
Result	Sprint is closed; unfinished tasks are moved.
Main Scenario	1. PM clicks "Complete Sprint". 2. System shows summary of completed vs. incomplete issues. 3. PM chooses destination for incomplete issues (Backlog or New Sprint). 4. System archives the sprint.
Alternatice Scenarios	N/A
Non-Functional Constraints	Trigger report generation upon completion.

Use case ID	***UC-PM-0011***
Use Case	Monitor Progress
Brief Description	PM tracks project health via charts (Burndown, Velocity).
Actor	Project Manager
Pre-Condition	Project is selected (<<include>> Select Project).

Result	Project reports are displayed.
Main Scenario	1. PM clicks "Reports". 2. PM selects report type (e.g., Burndown Chart). 3. System renders chart based on current data.
Alternatice Scenarios	N/A
Non-Functional Constraints	Data visualization must be responsive.

Use case ID	***UC-PM-0012***
Use Case	Follow Risk List
Brief Description	PM views and updates the risk register for the project.
Actor	Project Manager
Pre-Condition	Project is selected.
Result	Risk status is updated.
Main Scenario	1. PM opens "Risk Management" tab. 2. System lists identified risks. 3. PM updates status (e.g., Mitigated, Occurred) or adds new risk. 4. System saves changes.
Alternatice Scenarios	N/A
Non-Functional Constraints	N/A

Use case ID	***UC-PM-0013***
Use Case	Send and Receive Information
Brief Description	PM communicates with stakeholders or the system (Notifications / Messages).
Actor	Project Manager

Pre-Condition	User is logged in.
Result	Message sent or notification read.
Main Scenario	1. PM clicks on "Notifications/Inbox". 2. System displays messages from system or members. 3. PM reads or replies to a message. 4. System sends the reply.
Alternatice Scenarios	N/A
Non-Functional Constraints	Real-time notification delivery (WebSockets).

Use case ID	***UC-TM-001***
Use Case	View Project List
Brief Description	The Team Member views the list of projects they have been assigned to.
Actor	Team Member
Pre-Condition	User is logged in.
Result	A list of accessible projects is displayed.
Main Scenario	1. Team Member navigates to the "Projects" dashboard. 2. System filters and retrieves projects. 3. System displays the list (Project Name, Key, Lead).
Alternatice Scenarios	No Assignments: System displays "You are not assigned to any projects."
Non-Functional Constraints	List should load < 2 seconds.

Use case ID	***UC-TM-002***
Use Case	Monitor Progress
Brief Description	The Team Member views the project status, sprint burndown charts,

	or velocity charts to understand team performance.
Actor	Team Member
Pre-Condition	User is inside a specific project.
Result	Progress charts and metrics are displayed.
Main Scenario	1. Team Member clicks on the "Reports" or "Board" tab. 2. System calculates metrics based on current task status. 3. System renders the Sprint Board or Burndown Chart.
Alternatice Scenarios	N/A
Non-Functional Constraints	Data must reflect real-time updates.

Use case ID	***UC-TM-003***
Use Case	Update Task Status
Brief Description	The Team Member changes the workflow status of a task (e.g., from "To Do" to "In Progress").
Actor	Team Member
Pre-Condition	Task exists.
Result	Task status is updated in the database.
Main Scenario	1. Team Member selects a task on the Kanban board. 2. Team Member drags the task to the next column (or changes status via dropdown). 3. System validates the workflow transition. 4. System updates the status. *(Extension Point: Add Comments/Attachments)*
Alternatice Scenarios	Transition Not Allowed: System blocks the move and shows "Invalid Transition" error.
Non-Functional Constraints	N/A

Use case ID	*UC-TM-004*
Use Case	Log Work
Brief Description	The Team Member records the time spent working on a specific task.
Actor	Team Member
Pre-Condition	Task exists.
Result	Work log is saved, and remaining estimate is updated.
Main Scenario	1. Team Member opens a task. 2. Team Member clicks "Log Work". 3. Team Member inputs "Time Spent" (e.g., 2h) and "Date Started". 4. Team Member clicks Save. 5. System deducts time from "Remaining Estimate". *(Extension Point: Add Comments/Attachments)*
Alternatice Scenarios	N/A
Non-Functional Constraints	Input format validation (e.g., '1h 30m').

Use case ID	*UC-TM-005*
Use Case	Create/Edit/Delete Task/Sub-task
Brief Description	The Team Member manages the breakdown of work items under a PBI.
Actor	Team Member
Pre-Condition	A PBI exists.
Result	Task/Sub-task is created, modified, or removed.
Main Scenario	**(Create)** 1. Team Member selects a parent PBI.

	2. Team Member clicks "Create Sub-task". 3. Team Member enters Summary, Assignee, and Estimate. 4. Team Member saves the task. *(Extension Point: Add Comments/Attachments)*
Alternatice Scenarios	**(Delete)** User selects task -> Clicks Delete -> Confirms -> System removes task.
Non-Functional Constraints	N/A

Use case ID	***UC-TM-006***
Use Case	Estimate PBI
Brief Description	The Team Member provides an estimation (Story Points) for a PBI during planning.
Actor	Team Member
Pre-Condition	A PBI exists.
Result	Estimation value is saved.
Main Scenario	1. Team Member views a PBI. 2. Team Member clicks the "Estimate" field. 3. Team Member selects a value from the sequence (e.g., 1, 2, 3, 5, 8). 4. System updates the PBI.
Alternatice Scenarios	N/A
Non-Functional Constraints	N/A

Use case ID	***UC-TM-007***
Use Case	Refine Backlog

Brief Description	The Team Member reviews backlog items, adds technical details, or splits items to prepare them for future sprints.
Actor	Team Member
Pre-Condition	Access to Product Backlog.
Result	Backlog items are detailed and ready for planning.
Main Scenario	1. Team Member opens the Backlog view. 2. Team Member selects a PBI. 3. Team Member adds "Technical Description" or "Acceptance Criteria". 4. Team Member saves changes.
Alternatice Scenarios	N/A
Non-Functional Constraints	N/A

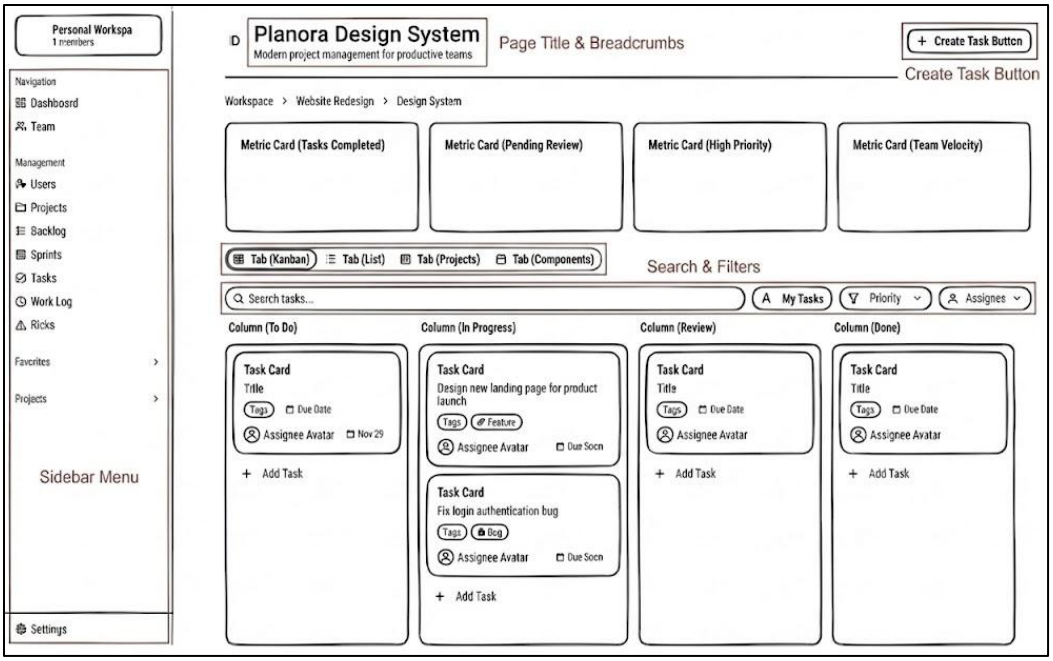
Use case ID	***UC-TM-008***
Use Case	Add Comments/Attachments
Brief Description	An extension use case that allows adding context (text or files) to various objects (Tasks, Logs, PBIs).
Actor	Team Member
Pre-Condition	User is executing a base use case (e.g., Log Work, Create Task, Update Status, View Project Info).
Result	Comment or file is appended to the object.
Main Scenario	1. User clicks the "Comment" or "Attachment" icon within the active form/view. 2. User enters text or uploads a file. 3. User clicks "Add". 4. System saves the entry and timestamp.
Alternatice Scenarios	Upload Fail: File size too large -> System shows error.
Non-Functional Constraints	Max file size 10MB. Allowed types: PNG, JPG, PDF, DOCX.

Use case ID	*UC-VIEWER-001*
Use Case	**View Project Information**
Brief Description	The Viewer (stakeholder/guest) accesses project details in read-only mode.
Actor	Viewer
Pre-Condition	User is executing a base use case (e.g., Log Work, Create Task, Update Status, View Project Info).
Result	Comment or file is appended to the object.
Main Scenario	1. User clicks the "Comment" or "Attachment" icon within the active form/view. 2. User enters text or uploads a file. 3. User clicks "Add". 4. System saves the entry and timestamp.
Alternatice Scenarios	Upload Fail: File size too large -> System shows error.
Non-Functional Constraints	Max file size 10MB. Allowed types: PNG, JPG, PDF, DOCX.

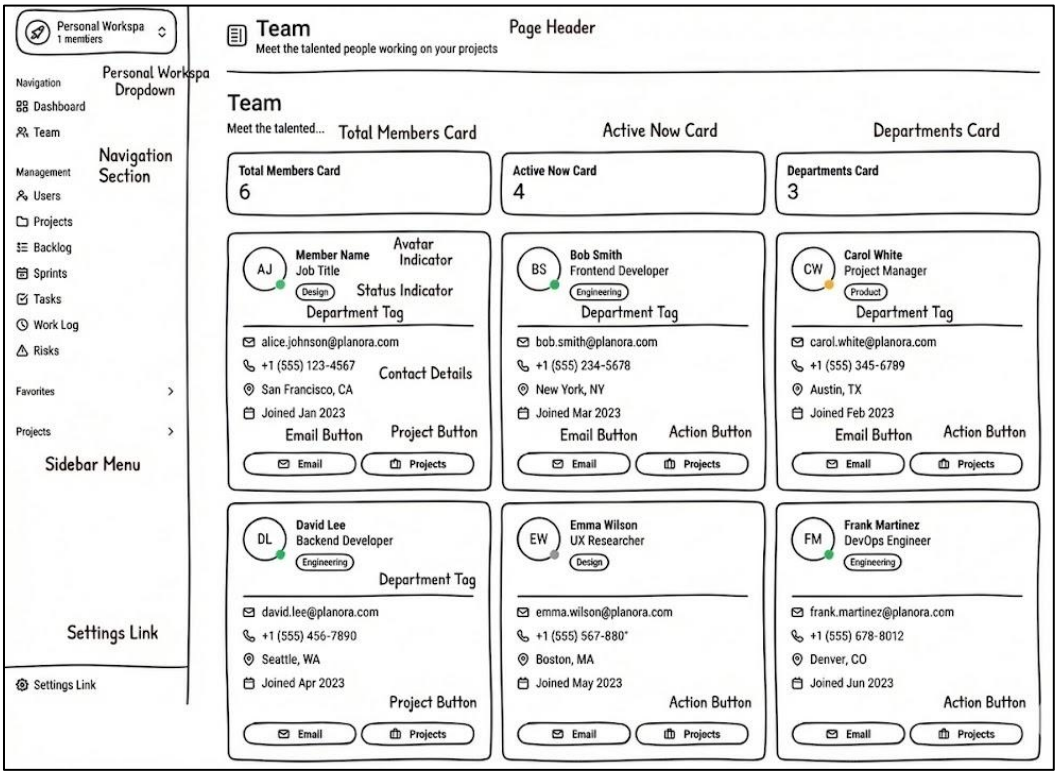
5 Prototype/Mockup

5.1 Wireframes

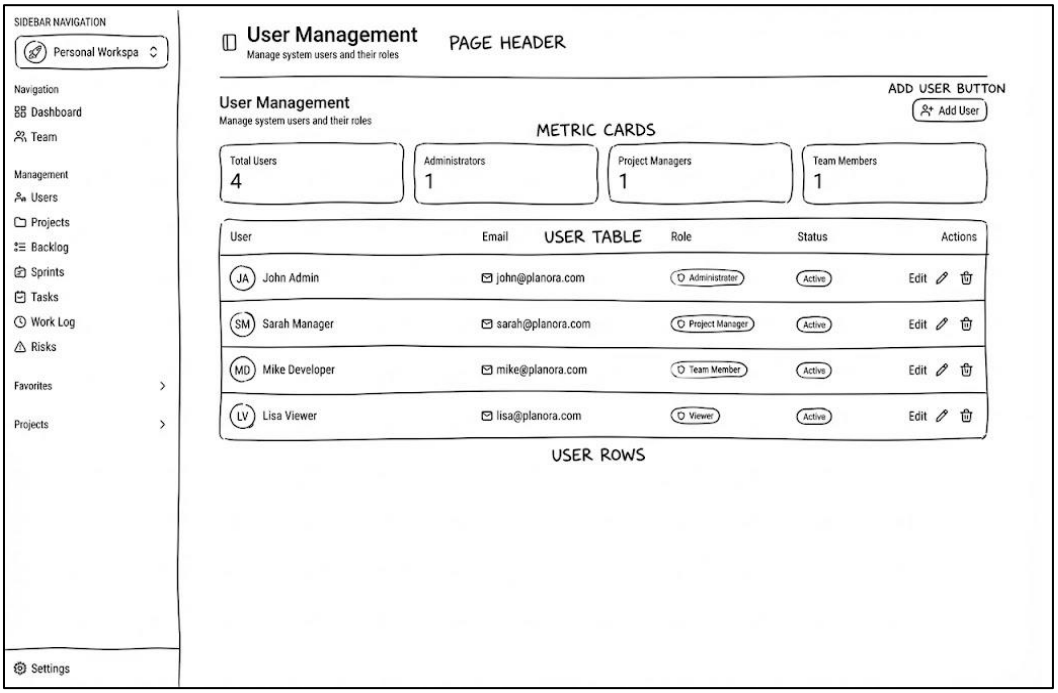
Login Screen



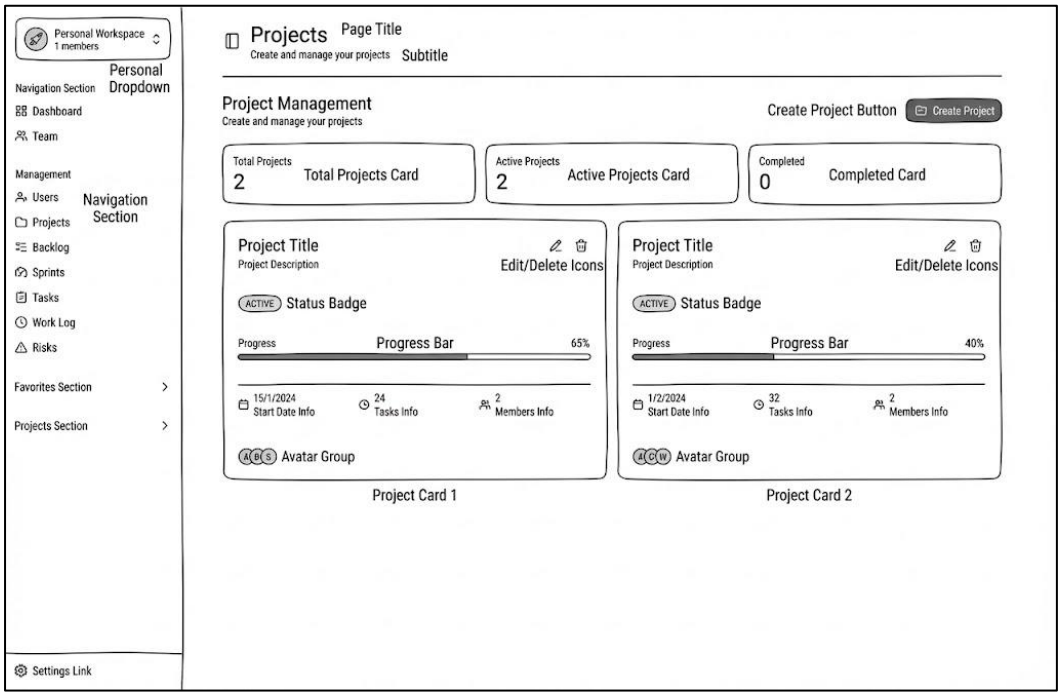
Dashboard screen



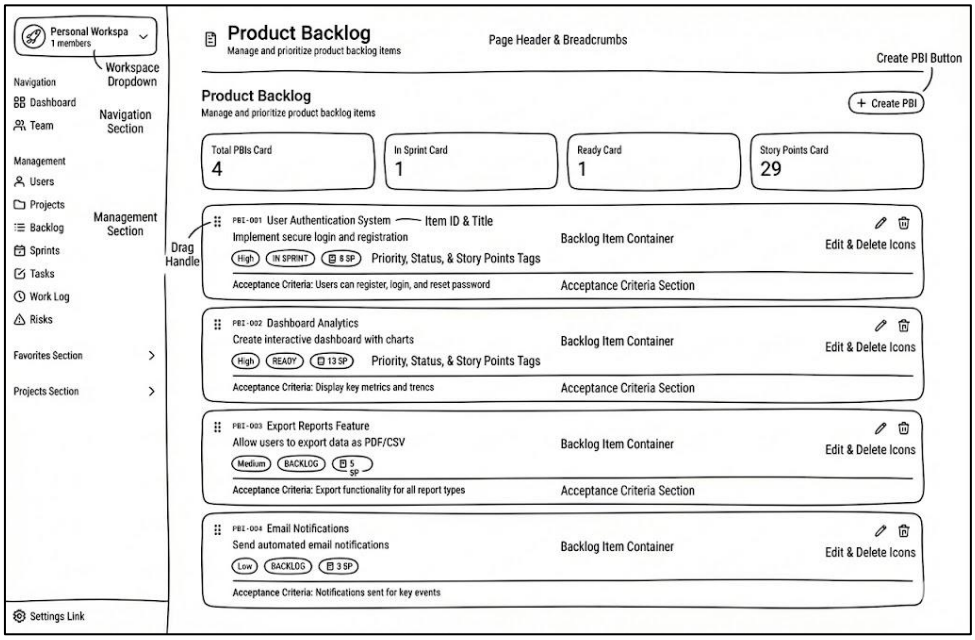
Team screen



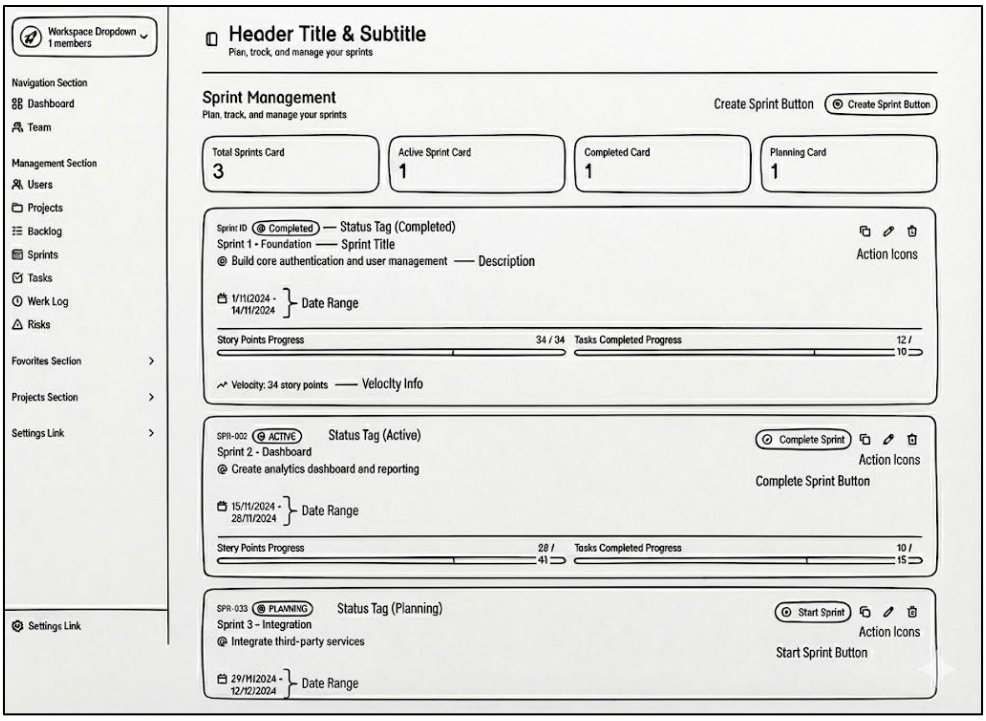
User Mangement screen



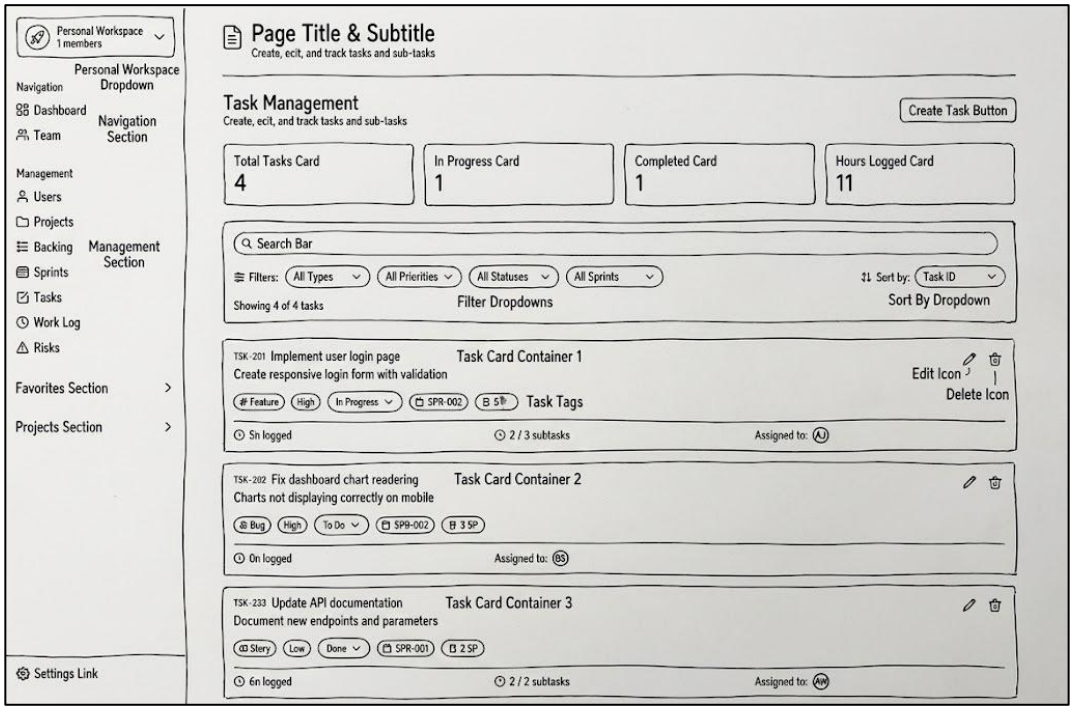
Project screen



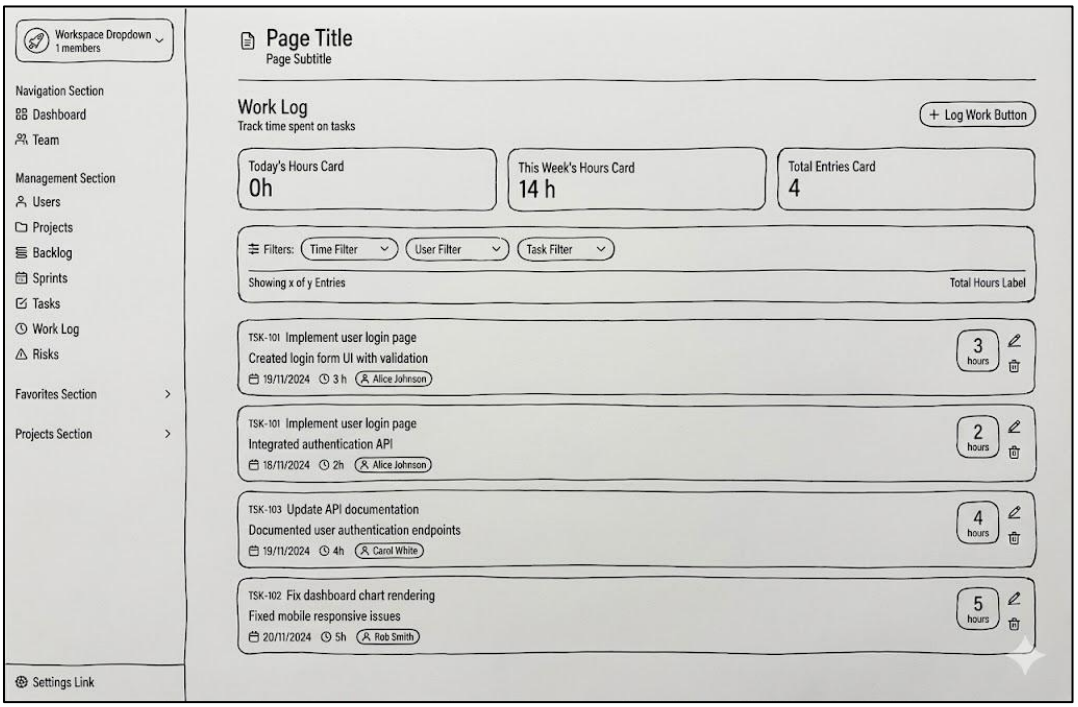
Product Backlog screen



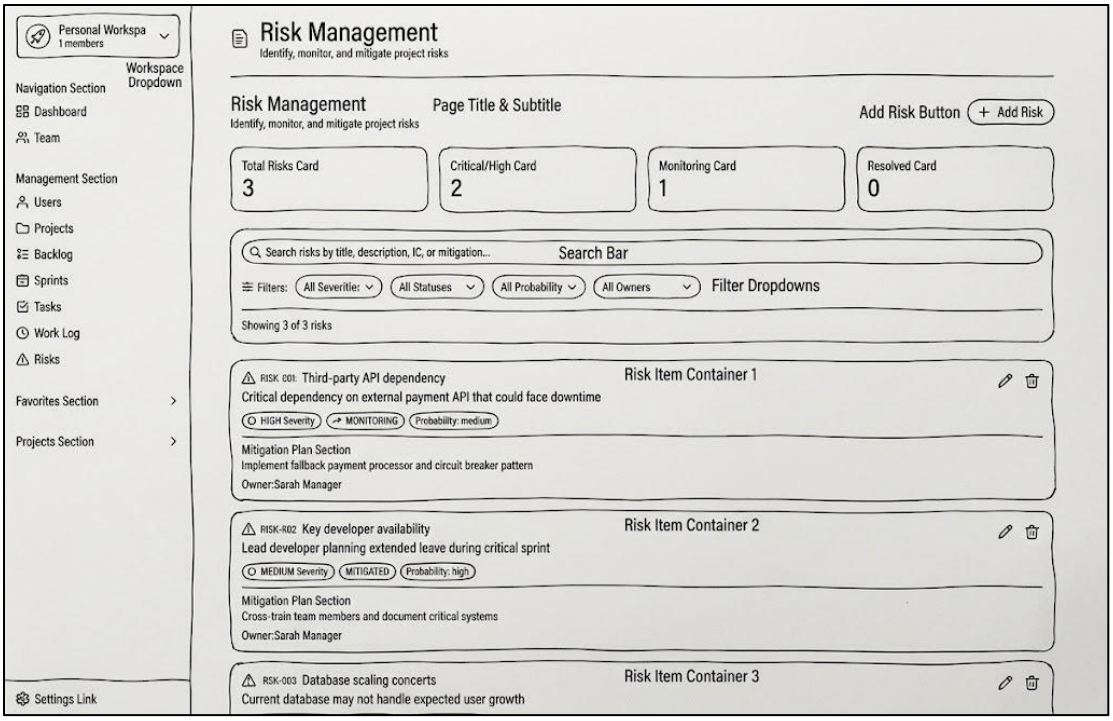
Sprint screen



Task Management screen



Work Log screen



Risk Management screen

Popups

5.2 Prototype

Link: UI prototype