Software Requirements Specification

for

Capstone Project

(Project Task Manager)

Version 1.1

Prepared by Nima Tshering

IOD Cohort

10-07-2025

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

2. Purpose 1

3. Business Context 1

4. Intended Audience 1

5. Project Scope 2

6. Description 2

7. Product Perspective 2

8. Product Features 3

9. Functional Requirements (User Stories) 3

10. Project Timeline 4

11. Effort Allocation 4

12. Future Work 5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| SRS | 10-07-2025 | Initial Draft | 1.1 |
| SRS | 12-07-2025 | Final Version | 1.2 |

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to clearly define the functional and non-functional requirements for the Project Task Manager application, as identified during the requirements elicitation phase. This document serves as a comprehensive reference for stakeholders, developers, and testers to ensure a shared understanding of the system's features and constraints, enabling successful development, deployment, and maintenance of the application.

## Business Context

This lightweight project management tool to be tailored for small businesses — helping teams stay organized, hit deadlines, and collaborate, without enterprise-level costs.

* + *Small businesses with 5–50 people*
  + *Limited software budgets*
  + *Rely on spreadsheets, emails, or other free tools*

## Intended Audience

This document is intended to serve as a comprehensive guide for the planning, design, development, testing, deployment, and maintenance of the Project Task Manager web application. The primary audience includes:

* **Project Manager**: Planning and overseeing the project timeline, scope, and deliverables.
* **UI/UX Designer:** To understand user expectations and interface needs to create system flow diagrams and user interface designs.
* **Development Team:** To implement features and functionalities based on clearly defined requirements.
* **QA/Testers**: To develop appropriate test cases and ensure the application meets acceptance criteria.
* **DevOps Engineers**: To plan deployment, scalability, and infrastructure needs.
* **Project Clients(evaluator)** track progress, ensure alignment with organizational objectives, and verify that the solution meets defined business needs.

This document may also serve as a point of reference for future maintenance or feature expansion.

## Project Scope

The web-based Project Task Manager will be designed with a modern, intuitive user interface to support efficient task planning, team collaboration, and real-time project tracking for internal teams.

The scope of this project includes: Development of a responsive web application accessible across desktops, tablets, and mobile devices.

Implementation of key project task management features, including:

* Define and document detailed business and system requirements.
* Design system architecture, database schema, and UI wireframes.
* Develop core functionality for task creation, assignment, and status tracking.
* Conduct functional and integration testing to ensure reliability.
* Prepare staging and production environments for deployment.
* Deploy the application to the production environment.
* Provide technical documentation for development, deployment, and maintenance.

Exclude third-party integrations, mobile app development, and advanced analytics in the initial phase

This scope delivers the essential components required to support collaborative project and task management within internal teams, with the flexibility to extend functionality through future sprints in an Agile development environment.

# Description

## Product Perspective

The Project Task Manager is a web-based application designed to help teams plan, track, and manage projects and tasks collaboratively. As a SaaS product, it will be hosted in the cloud and accessible through modern web browsers without the need for local installation, providing users with flexibility, scalability, and ease of access from any location. This application will be developed with a modular, multi-tenant architecture to support multiple organizations or teams under a single platform, while maintaining secure data isolation and user access controls

**Key Characteristics**:

* **SaaS Delivery Model**: Delivered as a cloud-hosted service with subscription-based access, ensuring scalability and centralized maintenance.
* **Multi-Tenant Architecture**: Designed to serve multiple clients or teams with isolated workspaces and user data.
* **Responsive Web Interface**: Optimized for desktop and mobile browsers to support cross-device usability.
* **API-First Approach:** Designed with future integrations in mind (e.g., third-party apps, mobile clients) through RESTful APIs.
* **Secure and Compliant**: Built with industry-standard security practices to protect user data.

This SaaS application is positioned as a lightweight yet powerful alternative to complex enterprise project management tools, offering flexibility for teams of varying sizes and industries

## Product Features

The Project Task Manager SaaS application provides a comprehensive set of functions to support project planning, task management, team collaboration, and progress tracking. Key functions include:

1. **Project Management:**

Create, edit, and archive projects; assign project owners; set project timelines and priorities; and monitor overall project status.

1. **Task Management**

Add, update, assign, prioritize, and categorize tasks; track task status and due dates; and support task comments and attachments.

1. **User Management**

Invite and manage team members; assign roles such as admin, project manager, or contributor; and manage team memberships across projects

1. **Responsive Web Interface**

Ensure usability across devices with a consistent user experience on desktop, tablet, and mobile browsers.

## Functional Requirements (User Stories)

1. **User Registration and Authentication**
   * As a new user, I want to register an account using my email and password, so that I can securely access the application.
   * As a registered user, I want to log in and log out securely, so that I can protect my data and sessions.
2. **User Management**
   * As an **admin**, I want to manage users, so that I can create, update, delete and view users.
3. **Project Management**
   * As a project manager, I want to create, edit, and delete projects, so that I can organize work effectively.
   * As a project manager, I want to add tasks to the project and set project deadlines, so that my team understands the project timelines.
4. **Task Management**
   * As a user, I want to create tasks for the project with title, description, due date, so that work items are clearly defined.
   * As a user, I want to update task status (e.g., At-Risk, In-Progress, Done), so that progress is tracked accurately.
   * As a user, I want to add comments to tasks, so that I can communicate with team members about task details.

# Project Timeline

The project will start on 7th July and conclude by 22nd July.

|  |  |
| --- | --- |
| Timeline | Activities |
| 7th July 12 July | Requirement finalization, environment setup, user and team management, and authentication features development |
| 13th July – 17th July | Core functionality development including project and task management, dashboard, and notifications |
| 18th July – 20th July | Testing, bug fixing, UI refinements, and preparation for presentation |
| 21st July | Review and rehearsal |
| 22nd July | Presentation and feedback session |

# Effort Allocation

|  |  |  |
| --- | --- | --- |
| Responsibility Areas | Estimated Effort (%) | Description |
| Requirement Analysis | 15% | Finalizing requirements and planning |
| Design & Development | 50% | UI/UX design, frontend and backend development |
| Testing & QA | 20% | Manual testing, bug fixes, and quality assurance |
| Presentation Prep | 15% | Preparing demos and documentation |

# Future Work

While the initial release of the Project Task Manager will focus on core features necessary for basic project and task tracking, several enhancements are planned for future iterations to improve functionality, user experience, and scalability. These include:

* **Advanced Reporting and Analytics**: Generation of project performance reports, user productivity metrics, and timeline-based analytics for better decision-making.
* **Integration with Third-Party Tools:** Support for integrating with platforms like Stripe Payment gateway
* **File Attachments and Document Management:** Ability to upload and manage documents or files associated with tasks and projects.
* **Custom Workflow Support**: Allowing teams to define custom task stages, labels, and workflows to match specific project needs.
* **Role-Based Access Control (RBAC):** Enhanced security and permissions, with fine-grained control over who can view, edit, or delete specific content using clerk authentication
* **Mobile Application Development** of a mobile version for iOS and Android to allow on-the-go task and project management.

# References

1. **Next.js 15+**
   * *Official Docs: https://nextjs.org/docs*
   * *App Router Overview:* [*https://nextjs.org/docs/app/building-your-application/routing*](https://nextjs.org/docs/app/building-your-application/routing)
2. **Middleware in Next.js**
   * *Middleware Guide: https://nextjs.org/docs/app/building-your-application/routing/middleware*
3. **JOSE Library (JWT Authentication)**
   * *Official Docs:* [*https://jose.dev/*](https://jose.dev/)
   * *GitHub Repository:* [*https://github.com/panva/jose*](https://github.com/panva/jose)
4. **Drizzle ORM**
   * *Official Docs: https://orm.drizzle.team*
   * *GitHub:* [*https://github.com/drizzle-team/drizzle-orm*](https://github.com/drizzle-team/drizzle-orm)
5. **MySQL**
   * *Official Site:* [*https://www.mysql.com/*](https://www.mysql.com/)
   * *Developer Docs:* [*https://dev.mysql.com/doc/*](https://dev.mysql.com/doc/)
6. **Tailwind CSS**
   * *Official Docs:* [*https://tailwindcss.com/docs*](https://tailwindcss.com/docs)
7. **ShadCN UI**
   * *Component Library: https://ui.shadcn.com*
   * *Based on Radix UI:* [*https://www.radix-ui.com/*](https://www.radix-ui.com/)
8. **MVC Architecture**
   * *MDN Glossary:* [*https://developer.mozilla.org/en-US/docs/Glossary/MVC*](https://developer.mozilla.org/en-US/docs/Glossary/MVC)
9. **ChatGPT (OpenAI)**
   * *Assisted in* ***code suggestions****,*
   * *Helped with* ***clarifying MVC pattern for NextJS****, and*
   * *Learn more:* [*https://openai.com/chatgpt*](https://openai.com/chatgpt)