

# Software Requirements Specification (SRS)

## Table of Contents

1. Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Definitions, Acronyms, and Abbreviations
  - 1.4 References
  - 1.5 Overview
2. Overall Description
  - 2.1 Product Perspective
  - 2.2 Product Features
  - 2.3 User Classes and Characteristics
  - 2.4 Operating Environment
  - 2.5 Design and Implementation Constraints
  - 2.6 Assumptions and Dependencies
3. System Features
  - 3.1 Task Management
  - 3.2 User Management
  - 3.3 Notifications and Collaboration
4. External Interface Requirements
  - 4.1 User Interfaces
  - 4.2 Hardware Interfaces
  - 4.3 Software Interfaces
  - 4.4 Communication Interfaces
5. System Attributes
  - 5.1 Performance Requirements
  - 5.2 Security Requirements
  - 5.3 Usability Requirements
  - 5.4 Reliability and Availability
  - 5.5 Maintainability
  - 5.6 Portability
6. Other Non-Functional Requirements
7. Appendices
  - 7.1 Diagrams and Flowcharts
  - 7.2 Glossary

---

## 1. Introduction

### 1.1 Purpose

This Software Requirements Specification (SRS) provides a detailed description of the functionalities, features, and constraints of the **Task Management System (TMS)**. The

document is intended to serve as a comprehensive guide for developers, testers, and stakeholders involved in the design and implementation of the system. This SRS will be used as a reference for system development and validation against customer expectations.

## 1.2 Scope

The Task Management System (TMS) is a web-based application designed to facilitate project and task management for individuals and teams. It provides a platform to organize tasks, track progress, manage deadlines, assign responsibilities, and enable collaboration across team members. Key features include task creation, task assignment, notifications, collaboration through comments, and real-time task tracking. The system supports integration with calendar tools like Google Calendar, reporting capabilities, and task prioritization.

## 1.3 Definitions, Acronyms, and Abbreviations

- **TMS:** Task Management System
- **API:** Application Programming Interface
- **UI:** User Interface
- **CRUD:** Create, Read, Update, Delete operations
- **Kanban:** A visual workflow management tool, typically a board with task cards
- **REST:** Representational State Transfer, an architectural style for APIs

## 1.4 References

- IEEE Standard 830-1998: Software Requirements Specifications
- UX/UI Design Guidelines
- Web Accessibility Standards (WCAG 2.1)
- [Agile Software Development Practices](#)

## 1.5 Overview

This SRS includes a detailed description of the system's overall design, individual system features, external interface requirements, system attributes, and non-functional requirements. Diagrams and flowcharts are included to illustrate the architecture and flow of operations.

---

# 2. Overall Description

## 2.1 Product Perspective

The Task Management System is an independent, web-based product developed to enhance productivity by allowing users to manage tasks across multiple projects. It follows a **client-server architecture**, where users interact with the front-end UI, and the back-end processes requests through REST APIs. The system integrates external tools such as Google Calendar and email notification services.

### 2.1.1 System Interfaces

- The TMS will interface with cloud-based storage for data persistence.
- It will connect to external services (Google Calendar) via OAuth2 for authentication and integration.

### 2.1.2 User Interfaces

- Users will interact with the system via a responsive web interface (browser-based) designed with HTML5, CSS3, and JavaScript frameworks (e.g., React or Vue.js).

### 2.1.3 Hardware Interfaces

- No specific hardware requirements for end-users; the system is accessible from any device with a web browser.

### 2.1.4 Software Interfaces

- **Operating System:** Windows, macOS, Linux
- **Web Browsers:** Chrome, Firefox, Edge, Safari (latest versions)
- **Third-party APIs:** Google Calendar, Slack (for notifications)

### 2.1.5 Communications Interfaces

- The system will use HTTPS for secure communication between clients and the server.
- RESTful APIs will be used for CRUD operations related to tasks, projects, and users.

## 2.2 Product Features

- **Task Management:** Users can create, assign, and track tasks within projects.
- **Kanban Board:** A visual representation of tasks in different states (e.g., To Do, In Progress, Done).
- **User Collaboration:** Team members can comment on tasks, tag each other, and share files.
- **Notification System:** Real-time notifications for task assignment and updates.
- **Reporting and Analytics:** Generate reports on task completion and user activity.

## 2.3 User Classes and Characteristics

- **Administrator:** Has full control over user management, task settings, and system configurations.
- **Project Manager:** Responsible for creating projects, assigning tasks, and monitoring progress.
- **Team Member:** Can view, update, and comment on tasks assigned to them.
- **Viewer:** Can view project status and task updates but cannot modify tasks.

## 2.4 Operating Environment

The system will be hosted on a cloud platform (AWS or Google Cloud) and accessed via web browsers. The environment will support horizontal scaling for high concurrency.

## 2.5 Design and Implementation Constraints

- **Security:** All user data must be encrypted at rest and in transit.
- **Compliance:** The system must comply with GDPR regulations for handling user data.
- **Scalability:** The system must support up to 10,000 active users concurrently.

## 2.6 Assumptions and Dependencies

- Users are expected to have stable internet access and use modern browsers.
  - The system depends on third-party services (Google Calendar, OAuth2) for integration.
  - Database performance must be optimized for scalability.
- 

# 3. System Features

## 3.1 Task Management

### 3.1.1 Description and Priority

This feature allows users to create tasks, assign them to team members, set deadlines, and update task statuses. It is a **critical feature** for the system.

### 3.1.2 Stimulus/Response Sequences

- **Stimulus:** A user creates a new task.
- **Response:** The task is added to the task list and displayed on the project's Kanban board.
- **Stimulus:** A user updates the task status (e.g., moves it to "In Progress").
- **Response:** The system updates the task's status and triggers a notification to the assigned user.

### 3.1.3 Functional Requirements

- **REQ-1:** The system shall allow users to create, edit, and delete tasks.
- **REQ-2:** The system shall allow users to assign tasks to multiple users.
- **REQ-3:** The system shall display tasks on a Kanban board with status updates.
- **REQ-4:** The system shall allow users to set task priorities (Low, Medium, High).

## 3.2 User Management

### 3.2.1 Description and Priority

This feature enables administrators to manage user roles and permissions. It is of **high priority** to ensure proper system access control.

### 3.2.2 Functional Requirements

- **REQ-5:** The system shall allow admins to create, update, and delete user accounts.
- **REQ-6:** The system shall allow admins to assign roles (e.g., Admin, Manager, Member).
- **REQ-7:** The system shall authenticate users via email/password.

### **3.3 Notifications and Collaboration**

#### **3.3.1 Description and Priority**

The system provides real-time notifications for task updates and supports team collaboration through task comments.

#### **3.3.2 Functional Requirements**

- **REQ-8:** The system shall notify users when they are assigned a task.
  - **REQ-9:** The system shall allow users to comment on tasks.
  - **REQ-10:** The system shall provide real-time updates on the Kanban board for task changes.
- 

## **4. External Interface Requirements**

### **4.1 User Interfaces**

- **UI-1:** The web interface shall allow users to view tasks in different states (To Do, In Progress, Done).
- **UI-2:** The system shall provide a task details modal for updating or commenting on tasks.

### **4.2 Hardware Interfaces**

- No specialized hardware is required for users; any device with a modern web browser will suffice.

### **4.3 Software Interfaces**

- **SI-1:** The system shall integrate with Google Calendar for task scheduling via the Google API.
- **SI-2:** The system shall provide a REST API for external tool integration.

### **4.4 Communication Interfaces**

- The system will use HTTPS for all data transmission between the server and client.
- 

## **5. System Attributes**

## 5.1 Performance Requirements

- The system shall handle up to 10,000 concurrent users.
- The system shall respond to task creation or updates within 2 seconds.

## 5.2 Security Requirements

- User passwords shall be hashed and stored using a secure hashing algorithm (e.g., bcrypt).
- All data transmissions shall use SSL/TLS for encryption.

## 5.3 Usability Requirements

- The system shall be designed with a simple, intuitive UI that requires no more than 30 minutes of user training.
- The interface shall support both desktop and mobile views.

## 5.4 Reliability and Availability

- The system shall have 99.9% uptime, with backup data stored daily in the cloud.

## 5.5 Maintainability

- The codebase shall follow modular development practices to ensure maintainability.
- The system shall support automated testing to facilitate continuous integration.

## 5.6 Portability

- The system shall be deployable on any cloud platform, including AWS and Google Cloud.
- It shall be platform-independent and accessible through any web browser.

---

# 6. Other Non-Functional Requirements

- **Backup and Recovery:** The system shall automatically back up the database daily and store backups for at least 30 days.
  - **Audit Logs:** The system shall maintain a log of all user actions for auditing purposes.
- 

# 7. Appendices

## 7.1 Diagrams and Flowcharts

- Use Case Diagram: To be added
- System Architecture Diagram: To be added

## 7.2 Glossary

- **CRUD:** Operations for Create, Read, Update, Delete
- **OAuth2:** An open-standard authorization protocol often used for granting access to third-party services