**TMS Software Requirements Specification (SRS)**

**Project Title:** Task Management System (TMS)  
**Version:** 1.1  
**Prepared By:**   
**Date:**

**1. Introduction**

**1.1 Purpose**

The purpose of the Task Management System (TMS) is to provide a robust platform for managing tasks efficiently. It caters to individuals and teams by streamlining task creation, assignment, progress tracking, and reporting.

**1.2 Scope**

The TMS will be a web-based application aimed at managing tasks and workflows for projects. The system will:

* Allow users to create, update, assign, and delete tasks.
* Provide role-based access control.
* Support task categorization, prioritization, and deadlines.
* Generate analytics and reports for task performance.

Target users include:

1. Teams managing agile projects.
2. Individual professionals seeking productivity tools.

**1.3 Definitions, Acronyms, and Abbreviations**

* **TMS:** Task Management System
* **API:** Application Programming Interface
* **CRUD:** Create, Read, Update, Delete
* **JWT:** JSON Web Token

**1.4 Overview**

This document outlines the functional and non-functional requirements for the Task Management System, including user roles, system features, and technical constraints.

**2. System Overview**

The system will consist of the following modules:

* **User Management:** Manage users, roles, and permissions.
* **Task Management:** CRUD operations for tasks.
* **Notifications:** Send reminders for deadlines.
* **Analytics and Reporting:** Generate task completion statistics.
* **Integration:** Provide RESTful APIs for integration with third-party tools.

**3. Functional Requirements**

**3.1 User Management**

* **FR-1:** Users can add, update, and remove users.
* **FR-2:** Users must authenticate using email/password with JWT-based authentication.
* **FR-3:** Support roles such as Admin, Manager, and Employee with defined permissions.

**3.2 Task Management**

* **FR-4:** Users can create tasks with the following attributes:
  + Title
  + Description
  + Priority (Low, Medium, High)
  + Deadline
  + Assignee
* **FR-5:** Users can update task status (e.g., Pending, In Progress, Completed).
* **FR-6:** Support bulk operations on tasks (e.g., delete multiple tasks).
* **FR-7:** Provide task search and filtering capabilities (e.g., by status, priority).

**3.3 Notifications**

* **FR-8:** Notify users of task deadlines via email.
* **FR-9:** Allow users to set custom reminders.

**3.4 Analytics and Reporting**

* **FR-10:** Provide dashboards for:
  + Task completion rates.
  + Pending tasks by priority.
  + Tasks assigned per user.
* **FR-11:** Generate exportable reports (PDF/CSV).

**3.5 Integration**

* **FR-12:** Provide RESTful APIs for external integrations.
* **FR-13:** Support GitHub integration for issue tracking.

**4. Non-Functional Requirements**

**4.1 Performance**

* **NFR-1:** The system must support up to 500 concurrent users.
* **NFR-2:** API response time must not exceed 2 seconds under peak load.

**4.2 Scalability**

* **NFR-3:** The system must scale horizontally to handle increasing user demands.

**4.3 Usability**

* **NFR-4:** The UI must be intuitive and user-friendly, following modern design principles.
* **NFR-5:** The system must support localization for English and French.

**4.4 Security**

* **NFR-6:** Implement secure authentication using JWT.
* **NFR-7:** All user data must be encrypted in transit and at rest.

**4.5 Maintainability**

* **NFR-8:** Codebase must follow standard design principles (e.g., MVC, DRY).
* **NFR-9:** Automated test coverage must exceed 95% for critical components.

**4.6 Compliance**

* **NFR-10:** Comply with GDPR for user data management.

**5. External Interfaces**

**5.1 User Interface**

* Web-based interface using React.js.

**5.2 API**

* RESTful APIs for task management, authentication, and reporting.

**5.3 Database**

* MySQL/MariaDB for data storage.

**5.4 Third-Party Integration**

* Integration with GitHub.

**6. Constraints**

* The project must be completed within 1 month.

**7. Assumptions**

* Users will have basic technical knowledge to interact with the system.
* Internet access is available for all users.

**8. Appendix**

* **Technologies Used:** Spring Boot, React.js, MySQL, MariaDB.
* **Tools:** IntelliJ IDEA, VSCode, Jenkins, JaCoCo, Maven.