SOFTWARE REQUIREMENT SPECIFICATION

1. Introduction

1.1 Purpose

The purpose of this SRS document is to provide a comprehensive overview of the "Project Management Tool." This document outlines the requirements, features, and specifications for the development and implementation of this software project.

1.2 Scope

The "Project Management Tool" is intended to be a robust and user-friendly software solution for managing projects efficiently. It is designed to streamline project planning, execution, monitoring, and reporting processes. The tool will target both individual users and teams, allowing them to plan, track, and collaborate on projects effectively.

1.3 Document Conventions

Throughout this document, certain conventions are used to enhance clarity and understanding:

- **Bold Text:** Indicates section headings and subheadings.
- `(Parentheses):` Provides additional explanations or context.
- `...` Signifies that more detailed information will follow.

1.4 Intended Audience

This SRS document is primarily intended for the following audiences:

- **Development Team:** This document serves as a guide for the development team responsible for designing, coding, and testing the "Project Management Tool."
- **Project Stakeholders:** This includes project managers, team leads, and end-users who need a clear understanding of the software's requirements and

capabilities.

- **Quality Assurance Team:** QA engineers will use this document to create test cases and validate that the software meets specified requirements.

1.5 References

This section provides a list of documents, standards, and external sources referenced during the preparation of this SRS. These references may include:

- **Project Proposal:** The original project proposal or description that outlines the high-level goals and objectives of the "Project Management Tool."
- **Relevant Standards:** Any industry or company-specific standards that must be followed during development.
- **User Feedback:** Feedback and input from potential end-users or stakeholders that influenced the project's scope and features.

2. Overall Description

2.1 Product Perspective

The "Project Management Tool" is positioned as a standalone software application that provides comprehensive project management capabilities. It is not intended to be integrated directly with other software systems but may offer import/export functionalities to facilitate data exchange with popular project management software.

2.2 Product Features

The "Project Management Tool" will encompass a wide range of features to meet the needs of project managers and team members. Key features include:

- **Project Planning:** Users can create and define project plans, set milestones, and assign tasks to team members.
- **Task Management:** Detailed task lists with priorities, due dates, and dependencies to track task progress.
- **Collaboration:** Real-time collaboration features, such as chat and comments on tasks, documents, and project timelines.
- **Resource Allocation:** Tools to allocate and manage resources, including team members and materials.
- **Time Tracking:** Functionality for tracking time spent on tasks and projects.

- **Reporting:** Comprehensive reporting and analytics capabilities to monitor project progress and identify bottlenecks.
- **Notifications:** Automated notifications for task updates, deadlines, and project status changes.
- **Integration:** Possibility to integrate with popular third-party tools and services like calendars and email.

2.3 User Classes and Characteristics

The "Project Management Tool" will cater to various user classes, including:

- **Project Managers:** Responsible for planning, executing, and monitoring projects. They require full access to all features for project management and reporting.
- **Team Members:** Collaborate on tasks and projects, primarily using the tool for task management, progress tracking, and communication.
- **Administrators:** Manage user access, configure settings, and ensure the tool operates smoothly.

2.4 Operating Environment

The "Project Management Tool" is designed to be a web-based application, accessible via modern web browsers (e.g., Google Chrome, Mozilla Firefox, Microsoft Edge) on multiple platforms, including Windows, macOS, and Linux. The backend will be hosted on robust and scalable server infrastructure.

2.5 Design and Implementation Constraints

- **Tech Stack:** The development of the "Project Management Tool" must adhere to the constraint that Java is a core component of the tech stack.
- **Data Security:** The tool must comply with data security and privacy regulations, implementing encryption and access control mechanisms.
- **Scalability:** The architecture should be scalable to accommodate an increasing number of users and projects.
- **Cross-Browser Compatibility:** The user interface should be designed to work seamlessly across various web browsers.

2.6 User Documentation

Comprehensive user documentation, including user guides, video tutorials, and a knowledge base, will be provided to help users understand and make the most of the "Project Management Tool."

2.7 Assumptions and Dependencies

- The project assumes that users have access to modern web browsers and an internet connection.
- Dependencies on third-party libraries and services may exist for certain features (e.g., authentication via third-party providers).

3. Specific Requirements

In this section, we will detail the specific functional and non-functional requirements of the "Project Management Tool." These requirements serve as a comprehensive guide for the development team, ensuring that the software meets the needs and expectations of its users.

3.1 Functional Requirements

3.1.1 Requirement 1 - User Registration and Authentication

- **Description:** Users must be able to register for an account and log in securely to access the application.
- **Inputs:** User-provided registration details (e.g., name, email, password).
- **Processing:** User data validation, authentication, and token-based session management.
- **Outputs:** Confirmation of successful registration or login.

3.1.2 Requirement 2 - Project Creation

- **Description:** Authenticated users should be able to create new projects.
- **Inputs:** Project name, description, start date, end date, and project settings.
- **Processing:** Validation of project details, creation of a new project in the database.
- **Outputs:** Confirmation of successful project creation.

3.1.3 Requirement 3 - Task Management

- **Description:** Users can create, update, and delete tasks within a project.
- **Inputs:** Task details, such as task name, description, priority, due date, and assignment.
- **Processing:** Validation of task data, task assignment, and task status updates.
- **Outputs:** Task creation and update confirmations.

3.1.4 Requirement 4 - Collaboration Tools

- **Description:** Users can collaborate on tasks and projects by sharing comments, attachments, and updates.
- **Inputs:** User comments, uploaded files, and task/project updates.
- **Processing:** Storing comments and attachments, real-time updates for project members.
- **Outputs:** Displaying comments and attached files.

3.1.5 Requirement 5 - Reporting and Analytics

- **Description:** Users can generate reports and access analytics to track project progress.
- **Inputs:** Selection of reporting criteria and analytics parameters.
- **Processing:** Data aggregation, generation of graphical reports.
- **Outputs:** Visual reports and analytics dashboards.

3.2 Non-functional Requirements

3.2.1 Performance

- **Performance Requirement 1:** The application must load within 3 seconds on standard internet connections.
- **Performance Requirement 2:** The system should support up to 1,000 simultaneous users without significant performance degradation.

3.2.2 Security

- **Security Requirement 1:** User passwords must be stored securely using industry-standard encryption algorithms.

- **Security Requirement 2:** Access to sensitive project data should be controlled through role-based access control (RBAC).

3.2.3 Usability

- **Usability Requirement 1:** The user interface (UI) should be intuitive, with clear navigation and user-friendly design.
- **Usability Requirement 2:** User testing should be conducted to ensure that common user tasks are easily achievable.

3.3 User Interface Requirements

3.3.1 UI Design

- **UI Requirement 1:** The UI should follow a responsive design approach to adapt to various screen sizes and devices.
- **UI Requirement 2:** The colour scheme and layout should be visually appealing and consistent.

3.3.2 Navigation

- **Navigation Requirement 1:** The application should have a top-level menu for easy access to key features.
- **Navigation Requirement 2:** Breadcrumb navigation should be available for users to track their location within the application.

3.4 Database Requirements

3.4.1 Data Models

- **Database Requirement 1:** The system should maintain a relational database to store user accounts, projects, tasks, and related data.
- **Database Requirement 2:** Proper indexing and normalisation should be implemented to ensure efficient data retrieval.

3.4.2 Data Storage

- **Database Requirement 3:** Files and attachments should be stored securely and efficiently, with appropriate backup and redundancy measures in place.

3.5 External Interface Requirements

3.5.1 APIs

- **API Requirement 1:** The application should expose RESTful APIs for integration with third-party services.
- **API Requirement 2:** API documentation should be provided for external developers.

3.5.2 Integrations

- **Integration Requirement 1:** The system should support integration with common calendar applications (e.g., Google Calendar) for task scheduling.
- **Integration Requirement 2:** Integration with email services (e.g., SMTP) for notifications and communication.

4. System Features

In this section, we will detail the key features and functionalities of the "Project Management Tool." These features are essential to meeting the needs of project managers, team members, and administrators using the software.

4.1 Feature 1 - Project Planning

- **Description:** Project Planning is a core feature that allows users to create and manage project plans effectively.
- **Functionality:**
  - **Create Projects:** Users can initiate new projects, providing project details such as project name, description, start date, end date, and project settings.
  - **Define Milestones:** Within each project, users can define milestones that mark significant achievements or stages in the project's lifecycle.
  - **Task Assignment:** Assign tasks to team members, specifying responsibilities, due dates, and task priorities.
  - **Dependencies:** Users can establish task dependencies to ensure that certain tasks cannot start until others are completed.
- **Benefits:** This feature enables project managers to lay out project timelines, allocate tasks, and establish dependencies, ensuring a structured and organised approach to project execution.

4.2 Feature 2 - Task Management

- **Description:** Task Management is a critical aspect of the "Project Management Tool" that enables users to create, update, and monitor tasks within projects.
- **Functionality:**
    - **Create Tasks:** Users can add new tasks to a project, including task name, description, due date, and priority.
    - **Update Tasks:** Task details can be edited, including changing due dates, priorities, and task assignments.
    - **Task Status:** Users can mark tasks as completed or in-progress, providing visibility into task progress.
    - **Task Comments:** Collaboration is facilitated through the ability to add comments and attachments to tasks.
- **Benefits:** This feature simplifies task tracking, ensures accountability, and promotes effective communication among team members.

4.3 Feature 3 - Collaboration Tools

- **Description:** Collaboration Tools are essential for real-time communication and information sharing among project team members.
- **Functionality:**
    - **Comments:** Users can leave comments on tasks, projects, and milestones, facilitating discussions and updates.
    - **Attachments:** Files and documents can be attached to tasks and projects, ensuring relevant materials are readily accessible.
    - **Real-Time Updates:** Users receive notifications and updates when there are changes or comments on tasks and projects.
- **Benefits:** Collaboration tools enhance teamwork and streamline project communication, reducing the need for external communication platforms.

4.4 Feature 4 - Reporting and Analytics

- **Description:** Reporting and Analytics empower users to monitor project progress and make data-driven decisions.
- **Functionality:**
    - **Report Generation:** Users can generate various reports, including task status reports, project timelines, and resource allocation reports.

    - **Data Visualisation:** Data is presented in the form of charts, graphs, and visual dashboards for easy interpretation.
    - **Customisation:** Users can customise report parameters and set filters to focus on specific project aspects.
- **Benefits:** Reporting and Analytics provide project managers and stakeholders with insights into project performance, allowing for timely adjustments and optimisations.

4.5 Feature 5 - Notifications

- **Description:** Notifications keep users informed about important project updates, deadlines, and task assignments.
- **Functionality:**
    - **Email Notifications:** Users receive email notifications for task assignments, upcoming deadlines, and mentions in comments.
    - **In-App Notifications:** Users see real-time notifications within the application for task updates and new comments.
- **Benefits:** Notifications ensure that users stay informed and engaged with project activities, reducing the risk of missed deadlines or critical information.

5. External Interface Requirements

In this section, we will detail the various external interfaces that the "Project Management Tool" may interact with, including user interfaces (UI), application programming interfaces (APIs), hardware interfaces, and integrations with third-party services.

5.1 User Interfaces

5.1.1 Login

- **Description:** The Login UI is the initial point of interaction for users accessing the application.
- **Requirements:**
    - **Username and Password:** Users must provide a valid username (email) and password for authentication.
    - **Forgot Password:** An option for users to reset their password via email.

    - **Remember Me:** An option to remember user credentials for convenience.
- **Design Considerations:** The UI should be user-friendly, responsive, and include appropriate error handling for login failures.

5.1.2 Dashboard

- **Description:** The Dashboard UI serves as the central hub for users to view project summaries, task lists, and project analytics.
- **Requirements:**
    - **Project Overview:** A summary of all ongoing projects and their status.
    - **Task Lists:** Display of task lists, categorised by project, with due dates and priorities.
    - **Analytics Widgets:** Visual representations of project analytics, such as charts and graphs.
- **Design Considerations:** The Dashboard should provide an intuitive and customisable interface, allowing users to tailor their view to their preferences.

5.2 APIs

5.2.1 Authentication API

- **Description:** The Authentication API provides endpoints for user registration, login, and session management.
- **Requirements:**
    - **User Registration:** Endpoint to create new user accounts with validation checks.
    - **User Login:** Endpoint for user authentication and token generation.
    - **Session Management:** APIs to manage user sessions, including token validation and expiration.
- **Authentication Protocol:** The API should use secure authentication protocols, such as OAuth 2.0, to ensure data security.

5.2.2 Integration with Other Tools

- **Description:** Integration APIs allow the "Project Management Tool" to communicate with third-party applications.

- **Requirements:**
    - **Calendar Integration:** API endpoints to synchronise project timelines and task deadlines with external calendar applications (e.g., Google Calendar).
    - **Email Integration:** APIs to send notifications and updates to users via email services (e.g., SMTP).
    - **Task Import/Export:** APIs for importing and exporting tasks and project data in standard formats (e.g., CSV, JSON).
- **Documentation:** Comprehensive API documentation should be provided to external developers for integration purposes.

5.3 Hardware Interfaces

5.3.1 Supported Browsers

- **Description:** The "Project Management Tool" should be accessible via web browsers on multiple platforms.
- **Requirements:**
    - **Supported Browsers:** Compatibility with popular web browsers, including but not limited to Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
    - **Browser Version Support:** Support for current and recent versions of supported browsers.
- **Cross-Browser Testing:** Regular testing and optimisation for each supported browser.

5.3.2 Device Compatibility

- **Description:** The tool should be accessible on a variety of devices, including desktops, laptops, tablets, and mobile phones.
- **Requirements:**
    - **Responsive Design:** The user interface should adapt to different screen sizes and orientations.
    - **Mobile App Compatibility:** Consideration for developing mobile apps for Android and iOS platforms.

6. Appendices

In this section, we include additional information and reference materials that supplement the main body of the SRS document. Appendices provide detailed documentation, diagrams, and other resources to enhance understanding and assist stakeholders, developers, and testers in the implementation and validation processes.

6.1 Glossary

The Glossary provides a list of terms, acronyms, and definitions used throughout the SRS document. This is essential for ensuring a common understanding among all stakeholders. Some key terms and their definitions may include:

- **SRS (Software Requirements Specification):** A formal document that outlines the software's functional and non-functional requirements.
- **API (Application Programming Interface):** A set of rules and protocols for building and interacting with software applications.
- **UI (User Interface):** The point of interaction between a user and a computer program, including screens, menus, buttons, and visual elements.
- **OAuth 2.0:** An open standard for authorisation often used in secure authentication protocols.

6.2 Use Case Diagrams

Use Case Diagrams visually represent interactions between system components and external actors (users or other systems). These diagrams help illustrate how the "Project Management Tool" is expected to be used. Examples of use cases may include:

- **User Registration:** A diagram showing the steps involved when a user registers for an account.
- **Task Creation:** Diagrams illustrating how users create and manage tasks within the application.
- **Project Reporting:** Diagrams outlining the steps for generating project reports.

6.3 Data Flow Diagrams

Data Flow Diagrams (DFDs) provide a visual representation of data flow within the system. They help describe how data moves through the application and how it is processed. Key DFDs may include:

- **User Authentication Flow:** Illustrating how user authentication requests are processed.
- **Task Assignment Flow:** Visualising how task assignments are made and updated.
- **Reporting Data Flow:** Showing how data is collected and processed to generate project reports.

6.4 User Guides

User Guides offer detailed instructions on how to use the "Project Management Tool." These guides are crucial for helping users understand and make the most of the software's features. User Guides may be presented as written documents or interactive tutorials and could include sections such as:

- **Getting Started:** Step-by-step instructions for new users on creating an account and logging in.
- **Project Management:** Guides on creating and managing projects, tasks, and milestones.
- **Collaboration:** Instructions for using collaboration tools like comments and attachments.
- **Reporting and Analytics:** How to generate and interpret project reports and analytics.

6.5 Test Plans

Test Plans outline the strategies and procedures for testing the "Project Management Tool." These plans detail test cases, scenarios, and criteria for validating that the software meets its specified requirements. Test Plans may include:

- **Functional Testing:** Test cases for verifying that each functional requirement behaves as expected.
- **Performance Testing:** Strategies for evaluating the system's performance under various loads and conditions.
- **Security Testing:** Procedures for testing the application's security measures, including authentication and data protection.