**Software Requirements Specification (SRS) for Personal Task Manager Application**

**1. Introduction**

**1.1 Purpose of the Document**

The purpose of this document is to define the software requirements for a Personal Task Manager Application. It will serve as a reference for developers, designers, and stakeholders to ensure that the application meets the intended objectives.

**1.2 Scope of the Application**

The Personal Task Manager Application is designed to help users efficiently manage their daily tasks. Key features include user authentication, task categorization, reminders, and customizable user preferences. The application will be developed using Python, Streamlit, and SQLite3 for database management. It will cater to individual users seeking an easy-to-use tool for personal task management.

**1.3 Intended Audience**

This document is intended for:

* **Developers**: To understand functional and non-functional requirements.
* **Designers**: To create user interface designs based on specified requirements.
* **Stakeholders**: To verify that the application meets business objectives.
* **Testers**: To create test cases based on the requirements.

**2. Overall Description**

**2.1 Product Perspective**

The Personal Task Manager Application is a standalone system with no dependencies on external systems. It will provide a simple yet powerful interface for users to manage their tasks.

**2.2 Product Functions**

* **User Authentication**: Secure login and registration.
* **Task Management**: Add, edit, delete, and view tasks.
* **Task Categorization**: Assign tags and priorities to tasks.
* **Reminders**: Set reminders for tasks.
* **User Preferences**: Customize application settings.

**2.3 User Classes and Characteristics**

* **General Users**: Individuals managing personal tasks. They require an intuitive and responsive interface with secure data storage.
* **Administrators**: Responsible for maintaining the system, ensuring updates, and troubleshooting.

**3. Functional Requirements**

**3.1 User Authentication**

* **Login**: Users can log in using their email and password.
* **Registration**: Users can register by providing basic details.

**3.2 Task Management**

* **Add Task**: Users can create tasks with a title, description, due date, and priority.
* **Edit Task**: Users can modify existing tasks.
* **Delete Task**: Users can remove tasks permanently.
* **View Tasks**: Users can view tasks in a list or calendar view.

**3.3 Task Categorization**

* **Tags**: Users can assign tags to tasks for better organization.
* **Priorities**: Tasks can be categorized as high, medium, or low priority.

**3.4 Reminders**

* **Set Reminders**: Users can set reminders for specific tasks.

**4. Non-Functional Requirements**

**4.1 Performance Requirements**

* The application should load the dashboard within 2 seconds.
* It should handle up to 10,000 tasks per user without performance degradation.

**4.2 Security Requirements**

* Passwords must be encrypted using industry-standard hashing algorithms.
* All user data should be stored securely in the SQLite database.

**4.3 Usability Requirements**

* The application should follow intuitive design principles.
* The interface should be accessible to users with minimal technical expertise.

**5. User Interface Design**

**5.1 Description of Main Screens**

* **Dashboard**: Displays an overview of tasks and upcoming reminders.
* **Task View**: Provides detailed information about a specific task with options to edit or delete.

**5.2 Navigation Flow**

* Login -> Dashboard -> Task View -> Add/Edit/Delete Task -> Settings.

**6. Data Management**

**6.1 Database Schema**

* **Users Table**:
  + id (Primary Key)
  + email
  + password
  + preferences
* **Tasks Table**:
  + id (Primary Key)
  + user\_id (Foreign Key)
  + title
  + description
  + due\_date
  + priority
  + tags
* **Categories Table**:
  + id (Primary Key)
  + name

**6.2 Data Storage and Retrieval Processes**

* Tasks will be stored in SQLite and retrieved based on user queries (e.g., filters by tags or due date).

**7. External Interfaces**

**7.1 APIs or Third-Party Services**

* **Streamlit Cloud**: For deployment of application.

**7.2 Integration Points**

* The application does not integrate with other applications initially.

**8. Assumptions and Constraints**

**8.1 Technology Stack Limitations**

* The application is built using Python, Streamlit, and SQLite, which limits scalability for large-scale deployments.

**8.2 User Environment Considerations**

* Users are expected to have a modern web browser to access the application.

**9. Dependencies**

**9.1 Required Software Libraries or Frameworks**

* **Python 3.12.7**
* **Streamlit**: For building the user interface.
* **SQLite3**: For database management.

This document outlines the core requirements for the development of the Personal Task Manager Application. Further updates will be made as the project progresses.