**Software Requirements Specification (SRS)**

**Project: my app**

Version: 1.0  
Date: August 04, 2025

**1. Introduction**

**1.1 Purpose**

This document outlines the functional and non-functional requirements for the my app system. The primary purpose of this project is To help users manage their tasks efficiently.

**1.2 Scope**

The my app system will provide functionalities for:

* Task Management
* Project Planning

**1.3 Intended Audience**

This document is intended for the following stakeholders:

* Students
* Professionals
* General users
* Project Managers
* Developers
* QA/Testers

**2. System Overview**

A simple and intuitive to-do list app designed to help users manage tasks efficiently.

The system will consist of the following primary functional areas:

* Task Creation
* Task Editing
* Task Deletion
* Task Due Dates and Reminders
* Task Categorization
* Task View Options
* Task Completion
* Task Synchronization
* User Authentication and Data Security
* Scalability and Performance

**3. Functional Requirements**

**3.1 Task Creation [0]**

**Description:** Users can create new tasks with a title, description, and optional due date and reminders.

**Priority:** High

**Type:** Functional

**Stakeholders:** End Users

**Constraints:** Task title must be at least 3 characters long.

* Due date format should be YYYY-MM-DD.

**Acceptance Criteria:** A user can input a title for a new task.

* A user can input a description for a new task.
* A user can optionally set a due date for a new task.
* A user can optionally set a reminder for a new task.
* The created task is saved and displayed in the task list.

**3.2 Task Editing [1]**

**Description:** Users can modify existing tasks by updating their title, description, due date, reminders, or category.

**Constraints:** []

**Acceptance Criteria:** Users can change the title of an existing task.

* Users can modify the description of an existing task.
* Users can update the due date of an existing task.
* Users can add, remove, or modify reminders for an existing task.
* Users can change the category of an existing task.

**Priority:** High

**Related Requirements:** []

**Business Value:** Enables users to keep their task information up-to-date and organized.

**Stakeholders:** End Users

**Type:** Functional

**3.3 Task Deletion [2]**

**Description:** Users can remove tasks from the task list.

**Application:** To-Do List App

**Priority:** High

**Type:** Functional

**Stakeholders:** User

**Constraints:** Deletion should be permanent.

* Deleted tasks should not be recoverable through undo/redo.

**Acceptance Criteria:** A user can delete a single task from the task list.

* A user can delete multiple tasks from the task list.
* Deleted tasks are removed from all views (list, calendar, etc.).
* Confirmation dialog should be displayed before deleting a task.
* The user should be informed that the deletion is permanent.

**3.4 Task Due Dates and Reminders [3]**

**Description:** Users can set due dates for tasks and receive notifications when the due date approaches.

**Application Area:** Task Management

**Priority:** High

**Stakeholders:** End User

* Product Owner

**Constraints:** Reminder notifications should be customizable (frequency, time of day).

**Acceptance Criteria:** Users can input a specific date and time for a task's due date.

* The app should send a notification to the user a configurable time before the due date.
* Users can choose to receive multiple reminders for a single task (e.g., 1 day before, 3 hours before, etc.).
* The notification should clearly indicate the task title and due date.
* Users can disable reminders for specific tasks or globally.

**3.5 Task Categorization [4]**

**Description:** Users can categorize tasks into groups or projects for better organization.

**Priority:** High

**Type:** Functional

**Stakeholders:** End Users

**Constraints:** Categories must be user-defined and customizable.

* There should be no limit to the number of categories a user can create.

**Acceptance Criteria:** Users can create new categories with descriptive names.

* Tasks can be assigned to multiple categories.
* The application should display tasks grouped by category.
* Category filters should allow users to view tasks based on specific categories.

**3.6 Task View Options [5]**

**Description:** The app should allow users to view their tasks in a list format or a calendar view.

**Constraints:** []

**Acceptance Criteria:** Users should be able to switch between list and calendar view.

* List view should display tasks chronologically or by priority.
* Calendar view should display tasks as events with color-coding based on category.
* Users should be able to filter tasks in both views by category, due date, or status.

**Priority:** High

**Stakeholders:** End Users

**User Stories:** As a user, I want to see my tasks in a list format so I can quickly scan through them.

**3.7 Task Completion [6]**

**Description:** Users can mark tasks as complete to indicate they have been finished.

**Constraints:** []

**Acceptance Criteria:** A user should be able to mark a task as completed.

* Completed tasks should visually differentiate from incomplete tasks.
* Marking a task as completed should update its status in the application's database.

**Priority:** High

**Dependencies:** []

**Related Requirements:** []

**3.8 Task Synchronization [7]**

**Description:** The app should provide a mechanism for users to sync their task data across multiple devices.

**Constraints:** Synchronization should occur seamlessly in the background.

* Data should be synchronized securely.

**Acceptance Criteria:** Users can log into their account on a different device and access their complete task list.

* Changes made to tasks on one device should be reflected on all other synced devices.
* The app should indicate the status of the synchronization process (e.g., syncing, synced, error).

**Priority:** High

**Stakeholders:** All users

**Dependencies:** []

**Related Requirements:** []

**3.9 User Authentication and Data Security [8]**

**Description:** The app should ensure secure user authentication and data protection.

**Priority:** High

**Type:** Functional

**Affected\_ Components:** User Management System

* Data Storage
* Network Communication

**Constraints:** Authentication must comply with industry-standard security practices.

* Data encryption must be implemented at rest and in transit.

**Acceptance Criteria:** Users must be able to create secure accounts with unique usernames and passwords.

* Password complexity requirements must be enforced (e.g., minimum length, special characters).
* Two-factor authentication should be optional for enhanced security.
* User data (tasks, categories, deadlines, etc.) must be encrypted using strong algorithms.
* Data transmitted between the app and the server must be encrypted using secure protocols (e.g., HTTPS).

**3.10 Scalability and Performance [9]**

**Description:** The app should be responsive and performant, even with a large number of users and tasks.

**Constraints:** The app should maintain a maximum response time of 2 seconds for any user action, regardless of the number of concurrent users or tasks.

* The app should be able to handle at least 100,000 tasks per user without significant performance degradation.

**Acceptance Criteria:** The task list should load and display all tasks within 2 seconds, even with 10,000 tasks.

* Adding a new task should not take more than 0.5 seconds.
* Searching for tasks should return results within 1 second, even with 1,000 search results.
* Deleting a task should be completed within 0.2 seconds.

**Priority:** High

**Category:** System Performance

**4. Non-Functional Requirements**

|  |  |
| --- | --- |
| **ID** | **Description** |
| NFR1 | The application should be responsive and mobile-friendly. |
| NFR2 | Tasks should sync across devices using cloud storage. |
| NFR3 | The system should support at least 10,000 users simultaneously. |