Software Requirements Specification (SRS)

Project: Todo App

Version: 1.0

Date: August 16, 2025

# 1. Introduction

## 1.1 Purpose

To help users manage tasks efficiently.

## 1.2 Scope

- Task management

- Personal task management

- Work task tracking

## 1.3 Intended Audience

- Students

- Professionals

- General users

# 2. System Overview

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

# 3. Functional Requirements

## 3.1 User Account Management [0]

**Description:**

Users can create, edit, and delete their accounts.

**Priority:**

High

**Subtasks:**

- Implement user registration functionality

- Allow users to modify their profile information

- Enable users to reset their passwords

- Provide a secure mechanism for deleting user accounts

## 3.2 Task Creation [1]

**Description:**

Users can add new tasks with descriptions, due dates, and reminders.

**Priority:**

High

**Subtasks:**

- Allow users to input task title and description.

- Enable users to set due dates for tasks.

- Implement a reminder feature for tasks.

- Validate user input for date format and required fields.

## 3.3 Task Editing [2]

**Description:**

Users can modify existing tasks, including updates to descriptions, due dates, and reminders.

**Priority:**

High

**Subtasks:**

- Allow users to change the task description.

- Enable users to modify the due date of a task.

- Provide an option to set or modify task reminders.

## 3.4 Task Deletion [3]

**Description:**

Users can remove tasks from their list.

**Priority:**

High

**Subtasks:**

- Implement a delete button for each task.

- Confirm deletion with a user prompt.

- Remove the task from the local storage/database.

- Update the UI to reflect the deleted task.

## 3.5 Task Categorization [4]

**Description:**

Users can group tasks into categories or assign tags for better organization.

**Priority:**

High

**Subtasks:**

- Implement functionality to create new task categories.

- Allow users to assign existing or create new tags to tasks.

- Display tasks based on selected categories or tags.

- Enable filtering and searching tasks by categories and tags.

## 3.6 Task Prioritization [5]

**Description:**

Users can prioritize tasks based on importance or urgency.

**Priority:**

High

**Subtasks:**

- Implement a priority level selection for each task (e.g., High, Medium, Low)

- Visually distinguish tasks based on their priority level in the task list

- Allow users to reorder tasks based on priority

- Option to set default priority for new tasks

## 3.7 Deadline and Reminder Management [6]

**Description:**

Users can set due dates and reminders for tasks.

**Priority:**

High

**Subtasks:**

- Implement functionality to allow users to set due dates for tasks.

- Develop a system for sending reminders to users based on set due dates and times.

- Allow users to customize reminder frequency and notification methods (e.g., email, push notification).

## 3.8 Task Completion Tracking [7]

**Description:**

Users can mark tasks as completed to track progress.

**Priority:**

High

**Subtasks:**

- Display a visual indicator for completed tasks.

- Allow users to toggle task completion status.

- Update task status in the database.

- Provide a progress summary based on completed tasks.

## 3.9 Task Views [8]

**Description:**

Users can view tasks in different formats, such as a list or calendar.

**Priority:**

High

**Subtasks:**

- Implement a list view to display tasks chronologically

- Develop a calendar view to display tasks by date

## 3.10 Device Synchronization [9]

**Description:**

Tasks are synced across multiple devices using cloud storage.

**Priority:**

High

**Subtasks:**

- Implement task data storage in a cloud database.

- Develop a real-time synchronization mechanism using WebSockets or a similar technology.

- Implement encryption for task data at rest and in transit.

- Design a mechanism for handling network interruptions and ensuring data consistency.

## 3.11 User Authentication [10]

**Description:**

Users can securely log in to the application using their credentials.

**Priority:**

High

**Subtasks:**

- Implement user registration functionality.

- Implement login functionality with password validation.

- Store user credentials securely.

- Implement password recovery functionality.

# 4. Non-Functional Requirements

**nfrs\_1**: The application should be responsive and mobile-friendly.

**nfrs\_2**: Tasks should sync across devices using cloud storage.

**nfrs\_3**: The system should support at least 10,000 users simultaneously.