Software Requirements Specification (SRS)

Project: Todo App

Version: 1.0

Date: August 24, 2025

# 1. Introduction

## 1.1 Purpose

To help users manage their tasks efficiently.

## 1.2 Scope

- Task management

- Project planning

## 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. The app supports task creation, categorization, deadlines, and reminders.

# 3. Functional Requirements

## 3.1 User Authentication [0]

**Description:**

Users can create accounts, log in, and securely access their data.

**Priority:**

High

**Subtasks:**

- Implement user registration functionality

- Implement user login functionality with password hashing

- Implement password recovery functionality

- Ensure secure storage of user data

## 3.2 Task Creation [1]

**Description:**

Users can add new tasks with details such as title, description, due date, and reminders.

**Priority:**

High

**Subtasks:**

- Allow users to input a task title

- Enable users to add a description for the task

- Provide a date picker for setting due dates

- Implement a reminder system with configurable time and frequency

- Validate user input for required fields

## 3.3 Task Editing [2]

**Description:**

Users can modify existing tasks, including updating details, deadlines, and reminders.

**Priority:**

High

**Subtasks:**

- Allow users to change the task title.

- Enable users to modify the task description.

- Provide an interface to update task deadlines.

- Enable users to add, edit, or delete task reminders.

## 3.4 Task Deletion [3]

**Description:**

Users can remove completed or unwanted tasks from their list.

**Priority:**

High

**Subtasks:**

- Implement a delete button next to each task.

- Confirm deletion with a dialog box.

- Delete the task from the database.

- Update the UI to reflect the deleted task.

## 3.5 Task Categorization [4]

**Description:**

Users can organize tasks into categories or tags for easier management and filtering.

**Priority:**

High

**Subtasks:**

- Implement a system for assigning categories to tasks.

- Display tasks categorized by their assigned categories.

- Allow users to filter tasks by specific categories.

- Provide a visual representation of task categories (e.g., color-coded tags).

## 3.6 Due Date and Reminder Management [5]

**Description:**

Users can set due dates and reminders for tasks, with customizable notification options.

**Priority:**

High

**Subtasks:**

- Allow users to input due dates for tasks.

- Enable users to set reminders for tasks with customizable time intervals (e.g., 1 hour, 1 day, 1 week) before the due date.

- Provide options for notification methods (e.g., email, push notification, in-app alert).

- Allow users to customize notification content (e.g., task details, due date, reminder time).

## 3.7 Task Prioritization [6]

**Description:**

Users can prioritize tasks, marking them as high, medium, or low priority for better organization.

**Priority:**

High

**Subtasks:**

- Implement UI to select priority level (High, Medium, Low)

- Store priority information with each task in the database

- Display tasks sorted by priority

- Allow users to change task priority

## 3.8 Task Viewing [7]

**Description:**

Users can view tasks in various formats, such as a list, calendar, or Kanban board.

**Priority:**

High

**Subtasks:**

- Display tasks as a list with details such as title, description, due date, and status.

- Implement a calendar view to visualize tasks based on their due dates.

- Develop a Kanban board view to organize tasks by priority or status.

## 3.9 Task Completion Marking [8]

**Description:**

Users can mark tasks as completed, indicating their successful completion.

**Priority:**

High

**Subtasks:**

- Display a clear visual indicator to mark tasks as completed.

- Implement functionality to toggle task completion status.

- Update task display to reflect completed status (e.g., strikethrough text, color change).

## 3.10 Cloud Synchronization [9]

**Description:**

Tasks and user data are synchronized across multiple devices using cloud storage.

**Priority:**

High

**Subtasks:**

- Implement user authentication and secure data storage in the cloud.

- Develop API endpoints for syncing tasks and user data.

- Ensure efficient data transfer and minimize bandwidth usage.

- Implement conflict resolution strategies for concurrent updates.

## 3.11 User Profile Management [10]

**Description:**

Users can view and edit their profile information, including name, email, and settings.

**Priority:**

High

**Subtasks:**

- Display user profile information (name, email, etc.)

- Allow users to edit their profile information

- Provide options to change password

- Implement settings management for user preferences

## 3.12 Search Functionality [11]

**Description:**

Users can search for tasks by keyword, category, or due date.

**Priority:**

High

**Subtasks:**

- Implement keyword search across all task fields (title, description, category)

- Enable category-based filtering for task search results

- Allow searching by due date range

- Display search results with relevant sorting and pagination options

## 3.13 Reporting and Analytics [12]

**Description:**

Users can view reports and analytics on their task completion rates and progress.

**Priority:**

High

**Subtasks:**

- Display total number of tasks completed vs. incomplete

- Show progress over time (e.g., weekly, monthly) for task completion

- Generate reports on task completion rates for different categories

- Allow users to filter reports by date range, category, or status

# 4. Non-Functional Requirements

**NFR-1**: The app should be responsive and mobile-friendly.

**NFR-2**: Tasks should sync across devices using cloud storage.

**NFR-3**: The system should support at least 10,000 users simultaneously.