Software Requirements Specification (SRS)

Project: Todo App

Version: 1.0

Date: August 08, 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.

The app allows users to add, update, and delete tasks, set due dates and reminders, organize tasks using categories or tags, and view tasks in a calendar or list view.

Users can mark tasks as complete and the app notifies them of upcoming deadlines.

# 3. Functional Requirements

## 3.1 Task Creation [0]

**Description:**

Users can create new tasks with a title, description, and optional due date and reminder.

**Priority:**

High

**Subtasks:**

- Implement a form for users to input task details (title, description)

- Allow users to optionally set a due date for the task

- Enable users to set reminders for tasks

- Validate input fields for required data and format

- Persist newly created tasks in the database

## 3.2 Task Editing [1]

**Description:**

Users can modify existing tasks by updating their title, description, due date, reminder, or status.

**Priority:**

High

**Subtasks:**

- Implement functionality to edit a task's title.

- Implement functionality to edit a task's description.

- Implement functionality to edit a task's due date.

- Implement functionality to edit a task's reminder.

- Implement functionality to change a task's status (e.g., from pending to completed).

## 3.3 Task Deletion [2]

**Description:**

Users can remove tasks from the to-do list permanently.

**Priority:**

High

**Subtasks:**

- Implement functionality to delete individual tasks.

- Ensure deleted tasks are removed from all views (list, calendar).

- Confirm deletion with user before permanently removing the task.

- Provide an undo functionality to recover deleted tasks within a specified time frame.

## 3.4 Task Scheduling [3]

**Description:**

Users can set due dates and reminders for individual tasks.

**Priority:**

High

**Subtasks:**

- Implement functionality to set due dates for tasks

- Allow users to specify reminder times for tasks

- Enable users to choose reminder frequency (e.g., daily, weekly)

- Ensure reminders are delivered through a chosen notification method (e.g., push notification, email)

- Display upcoming task reminders in a user-friendly interface

## 3.5 Task Categorization [4]

**Description:**

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

**Priority:**

High

**Subtasks:**

- Implement a mechanism for users to create and assign categories/tags to tasks.

- Allow users to search and filter tasks based on assigned categories/tags.

- Display tasks categorized by tags on the task list view.

- Provide an option to view all tasks regardless of category/tag.

## 3.6 List View Display [5]

**Description:**

The app displays tasks in a user-friendly list view, allowing for easy navigation and task management.

**Priority:**

High

**Subtasks:**

- Display tasks in a clear and concise format, including title, due date, category, and completion status.

- Allow users to sort tasks by various criteria, such as due date, category, or creation date.

- Enable users to filter tasks based on specific criteria, such as due date, category, or completion status.

- Provide options to view tasks in ascending or descending order within each sort criteria.

## 3.7 Calendar View Display [6]

**Description:**

The app provides a calendar view to visualize tasks and deadlines.

**Priority:**

High

**Subtasks:**

- Display tasks as events on the calendar

- Color-code events based on task category or priority

- Allow users to view tasks for a specific day, week, or month

- Show task deadlines as reminders on the calendar

- Enable users to click on a calendar event to view task details

## 3.8 Task Completion [7]

**Description:**

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

**Priority:**

High

**Subtasks:**

- Display a clear visual indicator to mark a task as complete.

- Implement functionality to toggle task completion status.

- Update task status in the database upon completion.

- Filter tasks based on completion status (completed/incomplete).

## 3.9 Deadline Notification [8]

**Description:**

The app sends notifications to users about upcoming deadlines for tasks.

**Priority:**

High

**Subtasks:**

- Send push notifications for tasks due within the next 24 hours.

- Allow users to customize notification frequency and time.

- Integrate with a reliable notification service (e.g., Firebase Cloud Messaging, AWS SNS).

- Display notification content with task title, deadline date, and relevant category/tag.

## 3.10 Task Synchronization [9]

**Description:**

Users can synchronize their tasks across multiple devices using cloud storage.

**Priority:**

High

**Subtasks:**

- Implement a cloud storage solution for task data.

- Develop a mechanism for detecting changes in task data and triggering synchronization.

- Ensure data integrity during synchronization.

- Provide users with feedback on the synchronization process.

# 4. Non-Functional Requirements

**NFR-1**: The application 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.