**Software Requirements Specification (SRS)**

**Project: todo\_app**

Version: 1.0  
Date: August 06, 2025

**1. Introduction**

**1.1 Purpose**

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

**1.2 Scope**

The todo\_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 Modification
* Task Deletion
* Task Due Date Assignment
* Task Reminder Setting
* Task Categorization
* Calendar View
* List View
* Task Completion Marking

**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

**Acceptance Criteria:** A new task can be created with a required title and description.

* Users can optionally set a due date for a task.
* Users can optionally set reminders for a task.
* Each created task is uniquely identifiable.

**Stakeholders:** End Users

**Subtasks:** Implement form for user input of task title and description.

* Implement functionality to set optional due date for tasks.
* Implement functionality to set optional reminders for tasks.
* Store created tasks in a persistent data store.

**3.2 Task Modification [1]**

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

**Constraints:** []

**Priority:** High

**Acceptance Criteria:** A user can edit the title of an existing task.

* A user can edit the description of an existing task.
* A user can change the due date of an existing task.
* A user can add or remove reminders for an existing task.
* A user can change the status of an existing task (e.g., from 'To Do' to 'In Progress' or 'Completed').

**Subtasks:** Implement an edit functionality for task titles.

* Implement an edit functionality for task descriptions.
* Implement an edit functionality for task due dates.
* Implement functionality to add and remove reminders for tasks.
* Implement functionality to change task status.

**3.3 Task Deletion [2]**

**Description:** Users can remove tasks from the to-do list.

**Priority:** High

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

* Deleted tasks are permanently removed from the list and cannot be recovered.
* The application provides a clear visual indication that a task has been deleted.
* Deleted tasks do not affect the order of remaining tasks.

**Constraints:** Task deletion should be performed securely to prevent accidental or malicious data loss.

**Subtasks:** Implement a delete button for each task.

* Confirm task deletion with a pop-up dialog.
* Remove the deleted task from the user interface.
* Update the local storage or database to reflect the task deletion.

**3.4 Task Due Date Assignment [3]**

**Description:** Users can assign due dates to tasks, specifying a target completion time.

**Constraints:** Due date must be a valid date in the future.

* Users should be able to edit assigned due dates for existing tasks.

**Priority:** High

**Acceptance Criteria:** A user can successfully input a due date for a new task.

* A user can view assigned due dates for existing tasks.
* A user can modify the due date of an existing task.

**Subtasks:** Implement date input functionality for new tasks.

* Display due dates for tasks in the task list.
* Allow users to edit existing task due dates.

**3.5 Task Reminder Setting [4]**

**Description:** Users can set reminders for tasks, receiving notifications before the due date.

**Constraints:** []

**Priority:** High

**Acceptance Criteria:** The user can specify a date and time for the reminder.

* The user can choose to receive a notification via email, push notification, or both.
* Reminders should be sent at the specified date and time.
* Users should be able to view and manage their set reminders.
* Users should be able to disable reminders for individual tasks.

**Subtasks:** Implement date and time picker for reminder setting.

* Integrate with notification system (email, push) for reminder delivery.
* Display reminders in a clear and accessible manner within the app.
* Allow users to edit and delete existing reminders.
* Enable users to toggle reminder notifications on/off per task.

**3.6 Task Categorization [5]**

**Description:** Users can categorize tasks using tags or labels, grouping similar tasks together.

**Constraints:** Tags should be alphanumeric and allow spaces.

* Users should be able to apply multiple tags to a single task.
* The system should display a list of available tags for easy selection.

**Priority:** High

**Business Value:** Improved task organization and filtering capabilities, enhancing user productivity.

**Subtasks:** Implement tag creation and management functionality.

* Allow users to assign tags to tasks during creation or editing.
* Enable filtering and searching of tasks by tags.

**3.7 Calendar View [6]**

**Description:** The app provides a calendar view, displaying tasks based on their due dates.

**Constraints:** []

**Priority:** High

**Acceptance Criteria:** Tasks are displayed on the corresponding day in the calendar.

* Users can navigate between days, weeks, and months in the calendar view.
* The calendar view should display tasks with their respective due dates clearly.
* The calendar view should be visually appealing and easy to understand.

**Subtasks:** Implement a calendar widget.

* Fetch tasks from the database and display them on the calendar based on due dates.
* Allow users to select a specific date in the calendar to view tasks for that day.
* Enable navigation between days, weeks, and months using calendar controls.

**3.8 List View [7]**

**Description:** The app displays tasks in a list format, showing task details and status.

**Constraints:** The list should be sortable by due date, priority, or task name.

* The list should allow for filtering tasks by status (e.g., completed, pending).

**Priority:** High

**Business Value:** Provides users with a clear and organized overview of their tasks.

**Acceptance Criteria:** The user can view all tasks in a list format.

* Each task in the list displays its title, description, due date, and status.
* The user can sort the list by due date, priority, or task name.
* The user can filter the list by status (completed, pending).

**Subtasks:** Implement list view display for all tasks.

* Display task title, description, due date, and status.
* Implement sorting functionality by due date, priority, and task name.
* Implement filtering functionality by task status.

**3.9 Task Completion Marking [8]**

**Description:** Users can mark tasks as complete, indicating that the task has been finished.

**Constraints:** The marking of a task as complete should be irreversible.

* Marked complete tasks should visually differentiate from incomplete tasks.

**Priority:** High

**Business Value:** Improves user satisfaction by providing a clear visual indication of task progress.

**Subtasks:** Implement a visual indicator (e.g., checkmark) to show task completion status.

* Allow users to toggle the completion status of a task.
* Persist the completion status of tasks even after the application is closed.

**4. Non-Functional Requirements**

|  |  |
| --- | --- |
| **ID** | **Description** |
| 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. |