**Software Requirements Specification (SRS)**

**Project: todo app**

Version: 1.0  
Date: August 04, 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 their 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 Management
* Task Scheduling
* Task Categorization
* Deadline Reminders
* Data Synchronization
* System Scalability

**3. Functional Requirements**

**3.1 Task Management [0]**

**Description:** Users can create, edit, and delete tasks.

**Constraints:** []

**Acceptance Criteria:** Users should be able to add new tasks with a title and description.

* Users should be able to modify existing task titles and descriptions.
* Users should be able to delete individual tasks.
* Deleted tasks should be permanently removed from the list.

**Priority:** High

**Category:** Core Functionality

**Affected User Stories:** As a user, I want to create a to-do list so that I can keep track of my tasks.

**3.2 Task Scheduling [1]**

**Description:** Users can assign due dates and set reminders for tasks.

**Application Area:** Task Management

**Priority:** High

**Type:** Functional

**Dependencies:** []

**Constraints:** Due dates must be in a valid date format.

* Reminder notifications should be sent at least 30 minutes before the due date.

**Acceptance Criteria:** Users can input a specific due date for each task.

* Users can set reminder notifications for tasks with customizable time intervals.
* The app should display the due date and reminder notification time clearly to the user.
* Reminder notifications should be delivered to the user via a chosen method (e.g., push notification, email).

**3.3 Task Categorization [2]**

**Description:** Users can organize tasks using categories or tags.

**Priority:** High

**Stakeholders:** User

**Constraints:** Categories should be user-defined and editable.

* A maximum of 10 categories per task is allowed.

**Acceptance Criteria:** Users can create new categories.

* Users can assign multiple categories to a single task.
* Users can view tasks filtered by specific categories.
* Existing categories can be renamed or deleted by the user.

**Related Requirements:** []

**3.4 Deadline Reminders [3]**

**Description:** The app should notify users of upcoming deadlines.

**Application Context:** A simple and intuitive to-do list app designed to help users manage tasks efficiently. The app supports task creation, categorization, deadlines, and reminders.

**Priority:** High

**Stakeholders:** End-Users

**Constraints:** Reminders should be sent at least one hour before the deadline.

* Users should be able to customize reminder settings (frequency, method).

**Acceptance Criteria:** The app should display a notification when a task's deadline is approaching.

* Users should be able to choose from different notification methods (e.g., push notification, email, in-app alert).
* Users should be able to configure the time interval before a deadline when a reminder is sent.
* The notification should include the task title and deadline date/time.

**Related Requirements:** []

**3.5 Data Synchronization [4]**

**Description:** Tasks should sync across devices using cloud storage.

**Application Area:** Task Management

**Priority:** High

**Type:** Functional

**Stakeholders:** User

* System Administrator

**Constraints:** Synchronization should be seamless and occur in the background.

* Data should be encrypted during storage and transmission.

**Acceptance Criteria:** A user can create a task on one device and it should be visible on other synced devices.

* Changes made to a task (e.g., status, deadline) on one device should be reflected on other synced devices.
* Deleted tasks should be removed from all synced devices.

**3.6 System Scalability [5]**

**Description:** The system should support at least 10,000 users simultaneously.

**Constraints:** []

**Acceptance Criteria:** The system should maintain responsiveness and performance under peak load with 10,000 concurrent users.

* User data access and task management operations should remain efficient with a high user base.
* The system architecture should allow for horizontal scaling to accommodate future growth beyond 10,000 users.

**Priority:** High

**Stakeholders:** End Users

* System Administrators

**4. Non-Functional Requirements**

|  |  |
| --- | --- |
| **ID** | **Description** |
| NFR1 | The app 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. |