Software Requirements Specification (SRS)

Project: odfas

Version: 1.0

Date: August 11, 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 Task Creation [0]

**Description:**

Users can add new tasks to their to-do list.

**Priority:**

High

**Subtasks:**

- Display an input field for task title.

- Allow users to set task description (optional).

- Provide options for task categorization (e.g., work, personal).

- Enable setting task due dates and reminders.

- Save new tasks to persistent storage.

## 3.2 Task Editing [1]

**Description:**

Users can modify existing task details, including content, due dates, and reminders.

**Priority:**

High

**Subtasks:**

- Allow users to change the task content.

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

- Provide an option to modify the reminder settings for a task.

## 3.3 Task Deletion [2]

**Description:**

Users can remove tasks from their to-do list.

**Priority:**

High

**Subtasks:**

- Implement functionality to delete individual tasks.

- Delete tasks from the database permanently.

- Provide visual confirmation to the user upon task deletion.

## 3.4 Due Date Assignment [3]

**Description:**

Users can set due dates for tasks.

**Priority:**

High

**Subtasks:**

- Display a calendar interface for selecting due dates.

- Allow users to input custom due dates.

- Validate due date format and ensure it is in the future.

- Store the due date along with the task information.

- Display the due date prominently in the task view.

## 3.5 Reminder Setting [4]

**Description:**

Users can set reminders for tasks.

**Priority:**

High

**Subtasks:**

- Allow users to specify reminder time and date

- Implement different reminder notification methods (e.g., push notification, email, in-app alert)

- Enable users to set recurring reminders for tasks

- Provide a clear and concise display of upcoming reminders within the app

## 3.6 Task Categorization [5]

**Description:**

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

**Priority:**

High

**Subtasks:**

- Implement a system for creating new task categories.

- Allow users to assign existing tasks to categories.

- Display tasks categorized by their assigned tags.

- Provide search functionality based on task categories.

## 3.7 Deadline Notification [6]

**Description:**

The app notifies users of upcoming deadlines.

**Priority:**

High

**Subtasks:**

- Send push notifications for upcoming deadlines.

- Implement email notifications for missed deadlines.

- Allow users to customize notification preferences (frequency, method).

## 3.8 Task Completion Marking [7]

**Description:**

Users can mark tasks as complete.

**Priority:**

High

**Subtasks:**

- Implement UI element to mark a task as complete.

- Update task status in the database upon marking as complete.

- Display completed tasks differently (e.g., strikethrough, grayed out).

- Allow users to un-mark a task as complete.

## 3.9 Cross-Device Synchronization [8]

**Description:**

Tasks are synced across multiple devices using cloud storage.

**Priority:**

High

**Subtasks:**

- Implement a cloud storage solution for task data.

- Develop a mechanism for authenticating users and syncing data securely.

- Ensure seamless synchronization of task creation, updates, and deletions.

- Handle conflicts that may arise from concurrent modifications on different devices.

# 4. Non-Functional Requirements

**NFR1**: The application 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.