Software Requirements Specification (SRS)

Project: Tasks App

Version: 1.0

Date: August 24, 2025

# 1. Introduction

## 1.1 Purpose

To streamline task management and enhance productivity.

## 1.2 Scope

- task creation

- task prioritization

- task scheduling

- task completion tracking

## 1.3 Intended Audience

- students

- professionals

# 2. System Overview

A mobile application designed to help users manage their tasks efficiently.

# 3. Functional Requirements

## 3.1 User Registration [0]

**Description:**

The application should allow users to create new accounts by providing their email address, password, and other relevant information.

**Priority:**

High

**Subtasks:**

- Implement email and password input fields with validation.

- Store user data securely in a database.

- Generate a unique user ID for each new account.

- Send a confirmation email to the user with a link to verify their email address.

## 3.2 User Login [1]

**Description:**

The application should allow registered users to log in securely using their email address and password.

**Priority:**

High

**Subtasks:**

- Implement email and password input fields.

- Validate email and password format.

- Hash passwords securely before storing.

- Authenticate users against the database.

- Handle successful and unsuccessful login attempts.

- Implement password recovery functionality.

## 3.3 Task Creation [2]

**Description:**

The application should allow users to create new tasks with customizable details such as title, description, due date, and priority.

**Priority:**

High

**Subtasks:**

- Add a new task entry form with fields for title, description, due date, and priority

- Validate user input for required fields and data types

- Allow users to set due dates and priorities using intuitive interfaces

- Persist newly created tasks in a persistent storage mechanism

## 3.4 Task Viewing [3]

**Description:**

The application should display a list of all tasks created by the user, allowing them to view task details and progress.

**Priority:**

High

**Subtasks:**

- Display a list of all user-created tasks with relevant details such as title, due date, and status.

- Provide an option to filter tasks by status (e.g., completed, pending, overdue).

- Allow users to view detailed information about a task by tapping on it.

- Display the task progress visually (e.g., a progress bar) based on completion percentage.

## 3.5 Task Editing [4]

**Description:**

The application should allow users to modify existing tasks by updating their details, such as title, description, due date, and priority.

**Priority:**

High

**Subtasks:**

- Allow users to change the title of an existing task.

- Enable users to modify the description of an existing task.

- Provide functionality for users to adjust the due date of an existing task.

- Allow users to update the priority level of an existing task.

- Ensure changes are reflected in the task list and other relevant views.

## 3.6 Task Deletion [5]

**Description:**

The application should allow users to delete tasks they no longer need.

**Priority:**

High

**Subtasks:**

- Implement a delete button for each task

- Confirm deletion with a user prompt

- Permanently remove the task from the database

- Update the task list to reflect the deletion

## 3.7 Task Prioritization [6]

**Description:**

The application should allow users to prioritize tasks, indicating their urgency and importance.

**Priority:**

High

**Subtasks:**

- Implement a priority level system (e.g., High, Medium, Low) for tasks.

- Allow users to assign priority levels to individual tasks.

- Display tasks in the user interface sorted by priority level.

- Provide visual cues (e.g., color-coding) to indicate task priority.

## 3.8 Task Reminders [7]

**Description:**

The application should send timely reminders to users about upcoming tasks, based on their due dates and preferences.

**Priority:**

High

**Subtasks:**

- Send reminders based on task due dates.

- Allow users to customize reminder times and frequencies.

- Integrate with device notifications for timely alerts.

- Provide options for snoozing or dismissing reminders.

## 3.9 Task Completion [8]

**Description:**

The application should allow users to mark tasks as completed, indicating their successful completion.

**Priority:**

High

**Subtasks:**

- Display a 'Mark as Complete' button for each task.

- Update the task status to 'Completed' when the button is clicked.

- Visually differentiate completed tasks from incomplete tasks.

- Allow users to undo the completion of a task.

## 3.10 Search Functionality [9]

**Description:**

The application should allow users to search for tasks based on keywords or criteria such as due date, priority, or tags.

**Priority:**

High

**Subtasks:**

- Implement keyword search functionality for task titles and descriptions.

- Allow filtering tasks by due date (past, today, upcoming).

- Enable filtering tasks by priority level (high, medium, low).

- Support searching and filtering by task tags.

## 3.11 Task Filtering [10]

**Description:**

The application should allow users to filter tasks based on various criteria, such as due date, priority, status, or tags.

**Priority:**

High

**Subtasks:**

- Implement filtering by due date (Today, This week, This month, Custom date range)

- Implement filtering by priority (High, Medium, Low)

- Implement filtering by task status (Completed, Pending, Overdue)

- Implement filtering by tags

## 3.12 Task Categorization [11]

**Description:**

The application should allow users to categorize tasks into different groups or categories for better organization.

**Priority:**

High

**Subtasks:**

- Implement a system for creating custom task categories.

- Allow users to assign existing tasks to specific categories.

- Display tasks categorized by their assigned categories.

- Enable users to filter and search tasks by category.

## 3.13 Cloud Synchronization [12]

**Description:**

The application should synchronize user data across multiple devices, ensuring data consistency and accessibility.

**Priority:**

High

**Subtasks:**

- Implement a cloud storage solution for user data.

- Develop a robust synchronization mechanism that tracks changes and updates data across devices.

- Design a system for handling conflicts that may arise during data synchronization.

## 3.14 Data Security [13]

**Description:**

The application should implement robust security measures to protect user data from unauthorized access and breaches.

**Priority:**

High

**Subtasks:**

- Implement user authentication with strong password requirements and multi-factor authentication options.

- Encrypt sensitive user data, such as usernames, passwords, and task details, both in transit and at rest.

- Regularly update security protocols and libraries to address emerging threats.

- Implement secure data storage practices, including access controls and data masking.

- Conduct regular security audits and penetration testing to identify vulnerabilities.

## 3.15 User Profile Management [14]

**Description:**

The application should allow users to manage their profiles, including updating their information and changing their preferences.

**Priority:**

High

**Subtasks:**

- Implement user registration functionality

- Allow users to update their name, email address, and other profile details

- Enable users to change their password

- Provide options for users to customize their profile settings, such as notification preferences and theme selection

## 3.16 Notification System [15]

**Description:**

The application should provide notifications to users about important events, such as task deadlines, reminders, or new messages.

**Priority:**

High

**Subtasks:**

- Implement push notifications for task deadlines

- Implement in-app notification badges for new messages

- Allow users to customize notification settings (frequency, sound, vibration)

# 4. Non-Functional Requirements

**NFR-1**: The application should be responsive and provide quick task completion feedback.

**NFR-2**: The application should be secure and protect user data from unauthorized access.

**NFR-3**: The application should have a user-friendly interface that is easy to navigate and understand.

**NFR-4**: The application should be compatible with different mobile devices and operating systems.

**NFR-5**: The application should be reliable and function correctly under varying network conditions.

**NFR-6**: The application should allow for offline task management capabilities.

**NFR-7**: The application should provide efficient task prioritization and organization features.

**NFR-8**: The application should allow for easy task sharing and collaboration with other users.

**NFR-9**: The application should have a notification system to remind users of upcoming tasks and deadlines.

**NFR-10**: The application should be scalable to accommodate a growing number of users and tasks.