Software Requirements Specification

Table of Contents

Chapter 1
INTRODUCTION

1.1 DOCUMENT PURPOSE

A Software Requirements Specification (SRS) is a comprehensive document that outlines the detailed requirements for a software system. The primary purpose of an SRS is to provide a clear and unambiguous description of what the software should do, how it should behave, and the constraints under which it must operate. This SRS provides a complete description of all the functions and specifications of the Todoit.

1.2 PROJECT SCOPE

1.2.1 DESCRIPTION

A Todo list, short for "To-Do list," is a simple and commonly used tool for organizing tasks, assignments, or activities that need to be completed. It serves as a reminder and a way to keep track of what needs to be done, helping individuals or teams stay organized and productive.

1.2.2 BENEFITS

A to-do list offers a structured approach to task management, promoting organization, productivity, and time management. By prioritizing tasks, setting deadlines, and providing a clear roadmap of what needs to be done, to-do lists help individuals stay focused, reduce stress, and achieve their goals. With the flexibility to adjust tasks as needed and the satisfaction of checking off completed items, to-do lists serve as valuable tools for enhancing accountability, improving memory, and fostering collaboration in both personal and professional settings.

1.3 INTENDED AUDIENCE

The SRS document is addressed to:

• Developers

• Testers

• All users of the program

1.4 DOCUMENT CONVENTIONS

The document follows the IEEE formatting requirements with font size 12 used and no special formatting techniques used.

1.5 REFERENCES

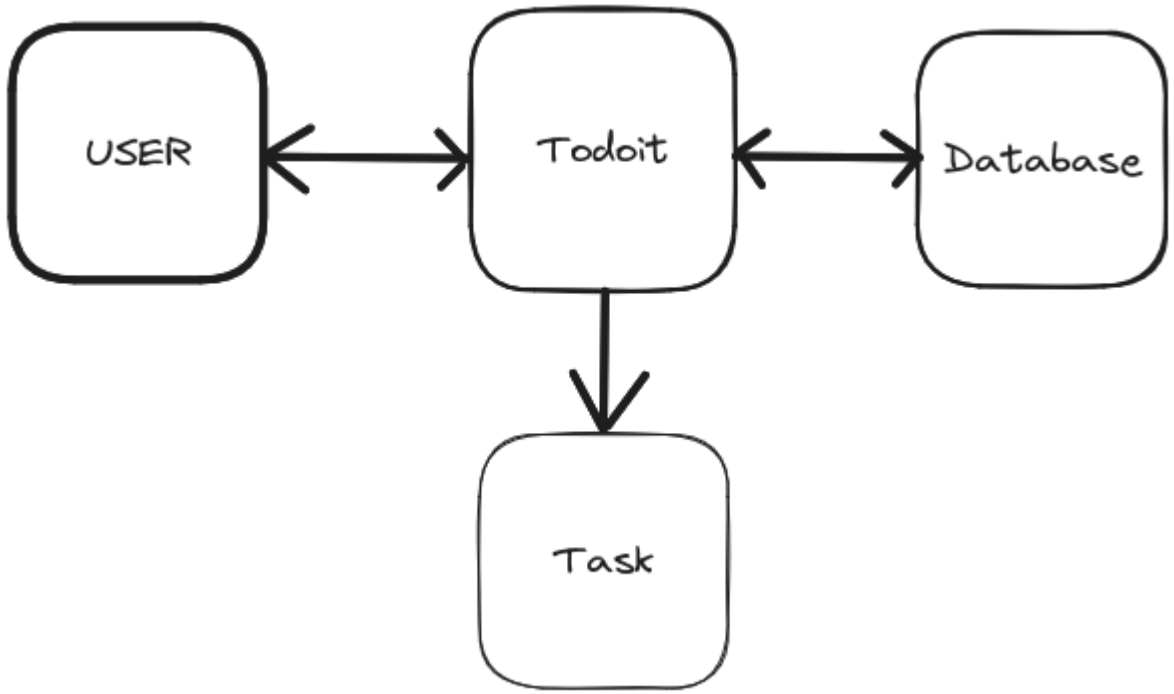
Software Engineering slides provided by lectures

Chapter 2

OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

This software is developed to serve as a pivotal organizational tool tailored to optimize task management and productivity in professional settings. Rooted in user-centric design principles, the emphasis lies in providing a seamless and intuitive user experience, characterized by robust functionality and versatile customization options. Core features encompass task creation, prioritization, and tracking capabilities, complemented by advanced functionalities such as deadline setting, sub-task management, and integration with external calendars or project management tools. Moreover, the product's architecture facilitates seamless collaboration among team members, allowing for shared task lists, real-time updates, and centralized task allocation.



2.2 PRODUCT FUNCTIONS

1. **Task Creation**: Users can easily create new tasks, including titles, descriptions, due dates, and priority levels.

2. **Task Organization**: The ability to categorize tasks into different lists or projects enables users to organize their workload according to their preferences and workflows.
3. **Task Prioritization**: Users can prioritize tasks based on urgency or importance, ensuring that critical activities are addressed promptly.
4. **Deadline Management**: Setting deadlines for tasks helps users stay on track and meet project milestones, with reminders or notifications to alert them of approaching deadlines.
5. **Task Details**: Users can include additional details or notes for each task, such as subtasks, attachments, or comments, to provide context and clarity.
6. **Task Tracking**: The to-do list allows users to track the status of tasks, including completion percentage or status updates, providing a clear overview of progress.

2.3 USERS AND CHARACTERISTICS

Our comprehensive task management solution caters to all user groups, from busy executives to organized students. Its intuitive design allows users to seamlessly manage tasks based on their specific workflow. For high-level professionals, it functions as a virtual assistant, providing reminders for critical meetings and milestones, prioritizing tasks based on company objectives, and ensuring focus on the most impactful work. Students and individuals can leverage its organizational capabilities to create structured plans using simple voice commands. This empowers everyone, from CEOs to teenagers, to achieve enhanced organization and productivity by maintaining constant visibility on their tasks.

2.4 OPERATION ENVIRONMENT

This web-based to-do list operates directly in your web browser, accessible from any device with an internet connection. The user interface you see is built with web technologies like HTML, CSS, and JavaScript. Behind the scenes, a web server handles data storage (using a secure database) and processing. The application communicates securely (via HTTPS) and is designed to scale smoothly for a growing number of users. This approach offers flexibility and accessibility, allowing you to manage your tasks from anywhere.

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

There aren’t any specific design and implementation constraints as long as you have a web browser and a internet connection.

2.6 ASSUMPTIONS AND DEPENDENCIES

Assumptions:

- The user has a basic understanding of how to use a web browser application
- The device the application runs on has a reliable internet connection (for cloud-based versions) or sufficient storage space (for local storage).
- The user is comfortable creating and managing tasks within a digital interface.

Dependencies:

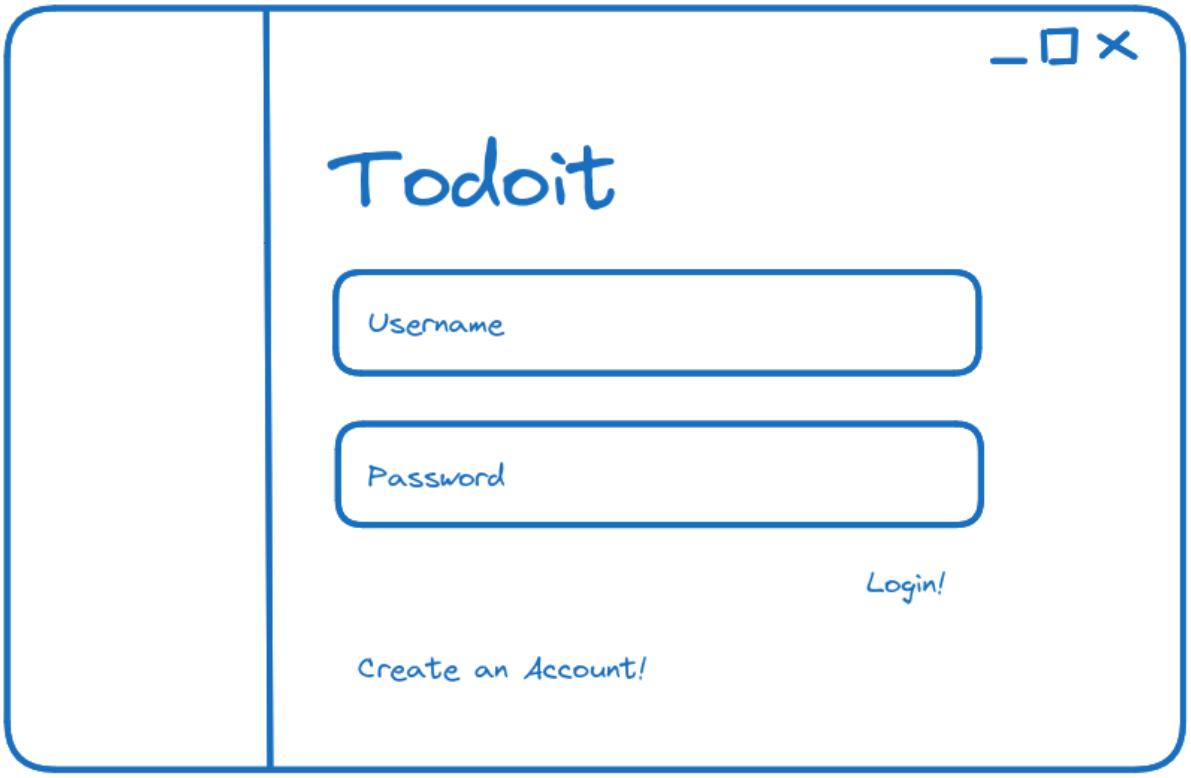
- For cloud-based versions, a stable internet connection is necessary for data storage, synchronization, and access across devices.

Chapter 3

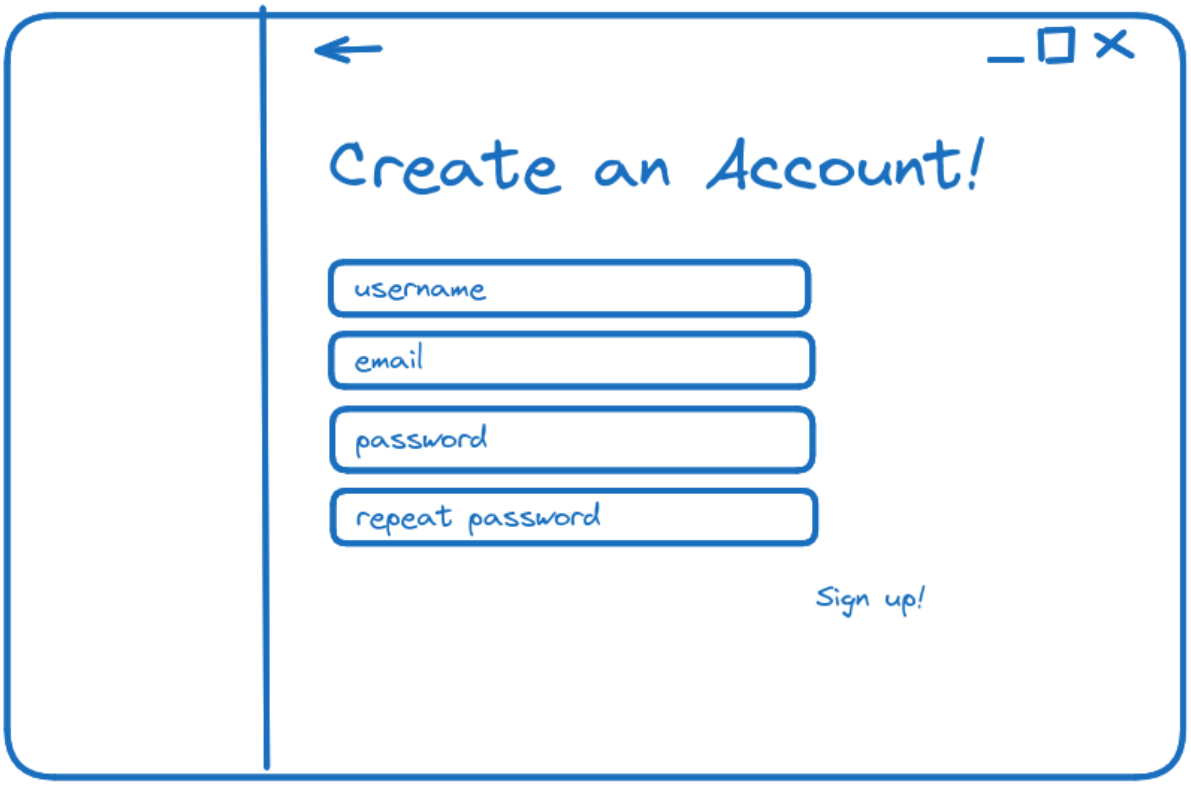
EXTERNAL INTERFACE REQUIREMENTS

3.1 USER INTERFACES

- Must have login screen



- Must have register page



3.2 HARDWARE INTERFACES
3.3 SOFTWARE INTERFACES

Not a lot is needed for software interface as all it need is a web browser

3.4 COMMUNICATION INTERFACES

Todoit will be both locally available and cloud-based as data will be stored locally when the user is offline and synced into the database when the user is online.

Chapter 4

SYSTEM FEATURES
4.1 TASK

Description: There are many features regarding the task in the to do list. Features in this category are about creating, editing and managing tasks.

Priority: Extremely High - 5/5 (Core functionality of the to-do list app)

(a) NEW TASK

USER REQUIREMENT

- User will add a task. User can set a task “meeting” on “3rd March 2024” at “3:00 pm” or Task name, date and time.

SYSTEM REQUIREMENT

- The system shall store the entered task title securely in the application's database.
- The system may optionally store the entered task description in the database.
- The system shall provide a clear button or action for users to save the newly created task.
- Upon saving, the system shall store the task details (title, optional description, and any additional features used) in the database.
- The system shall update the user interface to display the newly created task within the user's to-do list.

(b) EDIT TASK

User Requirements:

- Users can access an "Edit Task" option for existing tasks in their to-do list. (This could be achieved by tapping/clicking on the task or an "Edit" button.)
- Upon selecting "Edit Task," a dedicated edit page should open, allowing users to modify the task details.
- The edit page should provide functionalities to change:
    - Task name (text field)
    - Due date and/or time (optional - calendar or time picker)
- Users can save the edited task.

System Requirements:

- The system shall identify the selected task and display its current details on the edit page.
- The system shall allow users to modify the task name and optionally set a due date/time.
- The system shall implement validation to prevent saving tasks with an empty name. (REQ-v: Edited task cannot be saved if the new name is empty.)
- The system shall display an error message if a task is not selected before choosing "Edit Task." (REQ-vi: If there is not a task selected before user chooses edit task option, the task cannot be saved.)

- Upon successful editing and saving, the system shall update the modified task details in the database. (REQ-vi: The edited information will be updated in the database.)

(c) DELETE TASK

User Requirements:

- Users can select a task from their to-do list.
- Users can initiate task deletion using a dedicated "Delete Task" option (e.g., button, menu item).

System Requirements:

- The system shall identify the selected task for deletion.
- The system shall prompt the user for confirmation before permanently deleting the task. This can be a pop-up message asking "Are you sure you want to delete this task?"
- Based on user confirmation:
    - If the user confirms deletion (positive answer), the system shall delete the selected task from the database.
    - If the user cancels deletion (negative answer), the system shall maintain the current task list without deletion.

4.2 VIEW

(a) COMPLETED TASK

USER REQUIREMENT

- User will be able to view their completed tasks by choosing “View: Show completed tasks”.

SYSTEM REQUIREMENT

- User can choose the options to view tasks
- User can select the option of view completed tasks
- The system must show any task that is completed in the task list

(b) INCOMPLETE TASKS

USER REQUIREMENT

- User will be able to view their incomplete tasks

SYSTEM REQUIREMENT

- User can choose the options to view tasks
- User can select the option of view incomplete tasks
- The system must show any task that is incomplete in the task list

4.3 TASK PRIORITIZATION AND CATEGORIZATION

Description: This feature automatically categorizes user-created tasks, aiming to simplify task management and improve organization.

Benefits:

- **Improved Organization:** Tasks are automatically grouped by category, allowing users to easily see all related tasks in one place.
- **Reduced Manual Work:** Users don't need to spend time manually assigning categories to each task.
- **Increased Efficiency:** Easier to locate specific tasks and track progress within each category.

System Requirements:

- REQ-1: The system shall utilize a machine learning model or rule-based algorithm to analyze the task title, description, and potentially other relevant data (e.g., due date, keywords).
- REQ-2: Based on the analysis (REQ-1), the system shall automatically assign a pre-defined category label to each task (e.g., "School," "Work," "Errands," "Personal").
- REQ-3: The system shall store the assigned category along with the task details in the database.

Identifier	Priority	Requirement
R1	5	The app shall allow users to create accounts
R2	2	The app should allow login with Google accounts
R3	2	The system shall allow users to reset their forgotten passwords
R4	5	Users can add tasks
R5	4	Start time and end time for tasks

R6	4	Tasks can be edited
R7	5	Data will be stored locally when the user is offline and synced into the database when the user is online
R8	1	Users can set avatar
R9	4	The app will allow users to create new tasks
R10	3	Tasks can be deleted
R11	3	Filter and search for tasks
R12	3	Task priority

Chapter 5

OTHER NON-FUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

- **Page Load Time:** The application shall be responsive and usable on various devices (desktop, tablet, mobile). The application shall load and display tasks quickly, even with a large number of tasks.
- **Task Addition/Editing:** Adding, editing, or deleting tasks should be reflected in the UI with minimal delay.
- **Data Synchronization:** For multi-device usage, data synchronization between devices (e.g., phone, computer) should be efficient and avoid long wait times.

5.2 SAFETY REQUIREMENTS

- **Error Handling:** The system should gracefully handle unexpected errors (e.g., network issues) and provide informative messages to users.
- **Data Validation:** User input should be validated to prevent invalid data from being stored (e.g., empty task titles, incorrect due dates).
- I**nput Sanitization:** User input should be sanitized to prevent security vulnerabilities like cross-site scripting (XSS) attacks.

5.3 SECURITY REQUIREMENTS

- **User Authentication:** Implement secure user authentication mechanisms (e.g., password hashing, two-factor authentication) to protect user accounts.
- **Data Encryption:** Sensitive user data, like task details, should be encrypted at rest and in transit to ensure confidentiality.
- **Authorization:** Control access to user data based on user roles and permissions. Only authorized users should be able to view or modify specific tasks.
- **Regular Security Updates:** The web application and server software should be kept up-to-date with the latest security patches to address potential vulnerabilities.

5.4 SOFTWARE QUALITY ATTRIBUTES

- **Usability:** The user interface should be intuitive and easy to navigate, even for first-time users.
- **Reliability:** The application should function reliably with minimal downtime or errors.
- **Scalability:** The system should be able to handle an increasing number of users and tasks without significant performance degradation.
- **Maintainability:** The code should be well-documented and easy to maintain for future updates and bug fixes

5.5 BUSINESS RULES

- **Task Ownership:** Define who can create, edit, or delete tasks. This could be based on individual users or teams.
- **Task Sharing (Optional):** If applicable, define rules for sharing tasks with other users and setting access levels (view only, edit access).
- **Data Retention:** Determine how long completed tasks or user data will be stored before deletion (based on user preferences, legal requirements).
- **Privacy:** Clearly outline how user data is collected, stored, and used within the application.

Identifier	Priority	Requirement
R13	3	The application shall be responsive and usable on various devices (desktop, tablet, mobile).
R14	3	The application shall ensure data security and privacy for user tasks.
R15	4	The application shall function reliably and be available to users most of the time (high uptime).
R16	3	The application shall provide a user interface that is intuitive and easy to learn.
R17	2	The application shall load and display tasks quickly, even with a large number of tasks.
R18	3	System maintenance should be done regularly
R19	4	Synchronized data
R20	3	Logging and monitoring of the system and application must be in place.