Software Requirements Document (SRD) for To-Do List Software

1. Introduction

1.1 Purpose

The purpose of this SRD is to define the functional and non-functional requirements for the development of a simple to-do list software.

1.2 Scope

This document covers the requirements for a to-do list software that allows users to create, manage, and organize their tasks.

1.3 Definitions, Acronyms, and Abbreviations

- **To-Do Item**: A task or activity that a user needs to complete.
- **SRD**: Software Requirements Document.

1.4 References

- IEEE Standard 830-1998 - IEEE Recommended Practice for Software Requirements Specifications

1.5 Overview

This document is structured as follows:

1. Introduction
2. Overall Description
3. Specific Requirements
    - Functional Requirements
    - Non-Functional Requirements
    - External Interface Requirements
4. Constraints and Assumptions
5. Appendices

2. Overall Description

2.1 Product Perspective

The to-do list software is a standalone application that operates on both web and mobile platforms.

2.2 Product Functions

The main functions of the to-do list software include creating, editing, deleting, and organizing to-do items, as well as setting due dates and receiving notifications.

2.3 User Characteristics

The intended users are individuals who need to manage and organize their tasks.

2.4 Constraints

- The system must comply with data protection regulations and ensure the security of user data.
- The system must operate within the allocated budget.

2.5 Assumptions and Dependencies

- The system is assumed to operate in a stable network environment and rely on cloud-based data storage.
- The development team will have access to the necessary development tools and technologies.

3. Specific Requirements

3.1 Functional Requirements

FR-1: Create New To-Do Item

- **Use-Case Description**:
  - **Use Case ID**: UC-1
  - **Use Case Name**: Create New To-Do Item
  - **Actors**: User
  - **Preconditions**: The user is logged into the system.
  - **Main Flow**:
    1. The user selects the option to create a new to-do item.
    2. The system prompts the user to enter a title and description.
    3. The user enters the title (compulsory, max 120 characters) and description (optional, max 2400 characters).
    4. The user submits the new to-do item.
    5. The system saves the to-do item and confirms the creation.
  - **Postconditions**: The new to-do item is saved and displayed in the user's to-do list.
  - **Alternate Flows**:
    - **AF-1**: If the user does not enter a title, the system displays an error message and prompts the user to enter a title.
    - **AF-2**: If the user enters a title longer than 120 characters, the system displays an error message and prompts the user to shorten the title.

3.2 Non-Functional Requirements

NFR-1: Performance

- **Description**: The system shall respond to user actions (e.g., adding, editing, viewing tasks) within 1 second under normal operating conditions.

4. External Interface Requirements

EIR-1: REST API for To-Do Items

- **Description**: The system shall provide a REST API for managing to-do items, including endpoints for creating, editing, deleting, and retrieving to-do items.

EIR-2: Integrating OS calendar

- **Description**: The system shall integrate the OS calendar to achieve the setting of due dates and reminder.

5. Constraints and Assumptions

CA-1: Data Storage

- **Description**: The system shall store all to-do list data in a cloud-based database.

6. Appendices

Appendix A: Glossary

- **To-Do Item**: A task or activity that a user needs to complete.

Appendix B: Analysis Models

- **UML Diagrams**: Include UML diagrams illustrating the system architecture and interactions.

Appendix C: Issues List

- **Issue 1**: Determine the specific cloud service provider for data storage.