

# **SOFTWARE REQUIREMENT SPECIFICATIONS**

Prepared By

Geoffrey Lazer

Mugilan R

Lino Xavier Ignatius L

Nanci Arokia M R

# **SRS FOR TO-DO LIST Website**

## **1. Introduction**

### **1.1 Purpose**

This document outlines the functional and non-functional requirements for developing a task management application utilizing React and Spring Boot. The goal is to create an efficient and user-friendly to-do list app that facilitates task organization and completion.

### **1.2 Document Convention**

The entire document adheres to justified formatting. The main title uses Times New Roman font, bold style, and size 14. Subtitles follow the Times New Roman font, bold style, and size 12. The body text is in Times New Roman with a font size of 12.

### **1.3 Scope of Development Project**

The project involves designing and implementing a responsive to-do list application with interactive features. The app targets users seeking an organized tool for managing tasks and to increase productivity.

### **1.4 Definitions, Acronyms, and Abbreviations**

- SRS: Software Requirements Specification
- GUI: Graphical User Interface

## **2. Overall Descriptions**

### **2.1 Product Perspective**

The to-do list app functions as a standalone application, interacting with users through web browsers. It aims to provide an efficient task management solution with a user-friendly interface.

### **2.2 Product Function**

The app will offer the following functionalities:

- Task Management: Users can add, edit, and mark tasks as completed.
- Category Organization: Tasks can be categorized for better organization.
- User Authentication: Users can create accounts and log in securely.
- Notifications: Optional reminders for upcoming tasks.

## **2.3 User Classes and Characteristics**

- User: Anyone accessing the to-do list app.
- Admin: The individual with additional privileges for task management and app settings.

## **2.4 Operating Environment**

The app should be accessible from common web browsers and be responsive on desktops, tablets, and mobile phones.

## **2.5 Assumptions and Dependencies**

- The app will be hosted on a reliable server.
- Users have access to the internet for real-time updates and synchronization.

## **2.6 Requirements**

- The app should be developed using React for the front end and Spring Boot for the back end.
- Integration with a database for storing user tasks and settings.

## **2.7 Data Requirement**

User tasks and app settings will be stored in a secure database.

## **3. External Interface Requirement**

### **3.1 GUI**

The GUI should be modern, intuitive, and include:

- Task List: Clearly presented list of tasks with options for categorization.
- Navigation Bar: Easily accessible sections (Tasks, Categories, Settings).
- User Profile: Account information and settings.

## **4. System Features**

- Task Management: Add, edit, and mark tasks as complete.
- Category Organization: Categorize tasks for better management.
- User Authentication: Secure user accounts and login functionality.
- Notifications: Optional reminders for upcoming tasks.

## 5. Non-Functional Requirements

### 5.1 Performance Requirement

The app should load within 3 seconds for optimal user experience.

### 5.2 Safety Requirement

User authentication information should be encrypted for security.

### 5.3 Security

Implement secure coding practices to prevent common vulnerabilities.

### 5.4 Requirement Attributes

- Admin should easily update the app through a user-friendly interface.
- Ensure cross-browser compatibility.

### 5.5 Business Rules

- The app design should focus on simplicity and ease of use.

### 5.6 User Requirements

- Users should find it easy to manage tasks and navigate between sections.
- User authentication and account management should be straightforward

## 6. Class Diagram

