

SOFTWARE REQUIREMENTS SPECIFICATION

For

To-Do List Application

Prepared by:-

Antony Jude Shaman A

Babiana Roy R

Danush Athithya R K

Vibu Roshin E V

Academic Year: 2023-2024

1. Introduction

1.1. Purpose

The primary purpose of this document is to outline the requirements and specifications for the development of our To-Do List Application using React for the frontend and SpringBoot for the backend. This document serves as a comprehensive guide for project stakeholders, including developers, designers, and business analysts, to understand the scope and functionality of the proposed system.

1.2. Document Conventions

- Entire document should be justified.
- Convention for Main title
 - Font face: Times New Roman
 - Font style: Bold
 - Font Size: 14
- Convention for Sub title
 - Font face: Times New Roman
 - Font style: Bold
 - Font Size: 12
- Convention for body
 - Font face: Times New Roman
 - Font Size: 12

1.3. Scope of Development Project

The scope of this project extends to the creation and development of a dynamic To-Do List Application. The application will provide users with an intuitive platform to manage tasks efficiently, set priorities, and enhance overall productivity through features such as due dates, reminders, and collaboration.

1.4. Definitions, Acronyms, and Abbreviations

- React: A JavaScript library for building user interfaces, commonly used for building the frontend of web applications.
- Java: A widely-used programming language, utilized for building the backend of web applications.

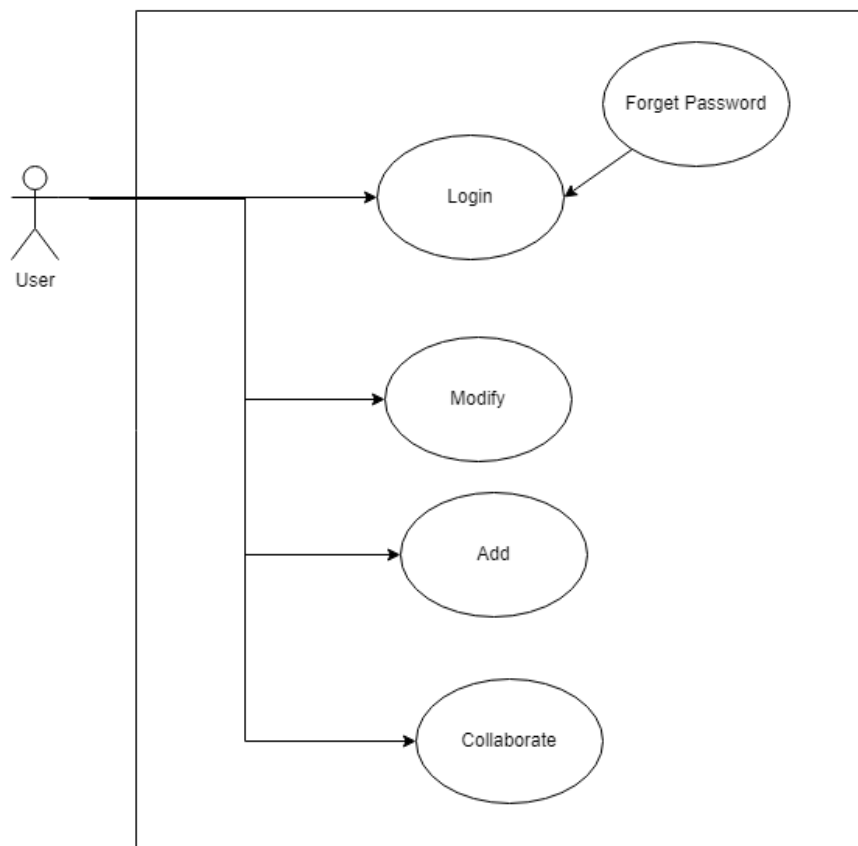
- SpringBoot: A Java-based framework for building web applications, known for simplifying development by providing robust features for creating RESTful APIs and handling business logic.
- ER Diagram: Entity-Relationship Diagram, a visual representation of the data model that shows the relationships between different entities.
- GUI: Graphical User Interface, the visual part of a computer program that users interact with.

2. Overall Descriptions

2.1. Product Perspective

The To-Do List Application is designed to operate as a standalone system, providing users with a centralized and organized platform for efficient task management. It functions independently, allowing users to create, manage, and collaborate on tasks seamlessly. While it operates independently, the application recognizes the importance of integration with external calendars and task management tools to enhance user productivity.

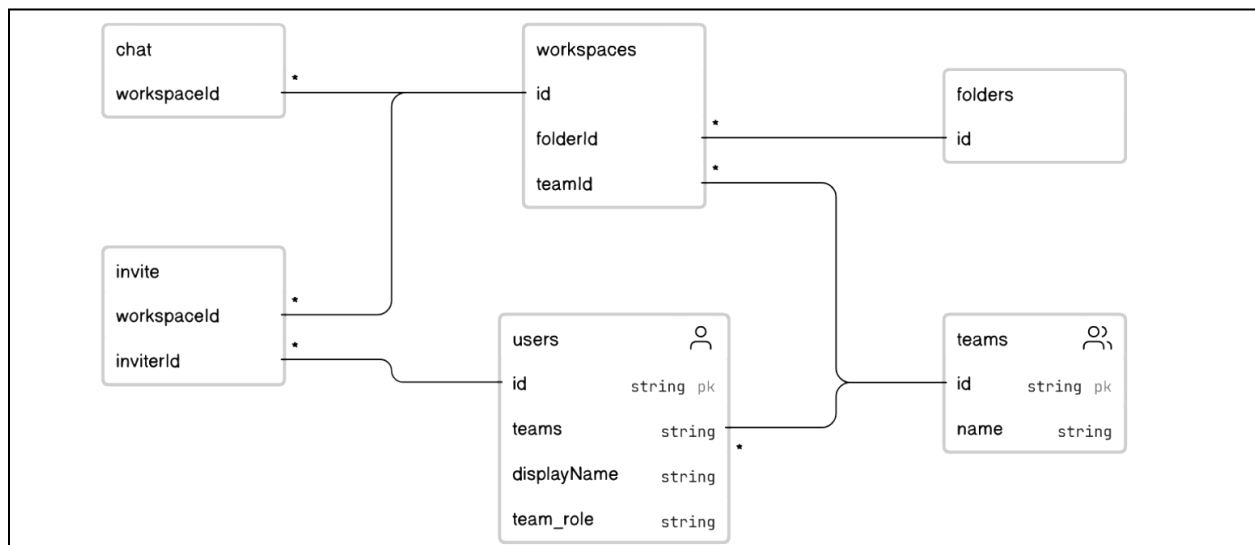
Use Case Diagram for To-Do List Application:



2.2. Product Function

The core functionality of the To-Do List Application centers around efficient task management, enabling users to organize and prioritize their activities seamlessly. The application facilitates the creation, editing, and deletion of tasks, with each task having essential details such as a title, description, due date, and priority level. Tasks can be organized into categories or projects, offering users flexible organization options. An Entity-Relationship (ER) diagram provides a visual representation of the main data entities and their relationships within the To-Do List Application, ensuring a clear understanding of the underlying data structure that supports robust task management.

ER Diagram for E-Commerce Website:



2.3. User Classes and Characteristics

The E-Commerce website will accommodate two primary user classes, each with distinct characteristics and functionalities:

- **Users:**
 - Can register an account with a unique username and password, providing a personalized experience.
 - Can input and manage personal information relevant to task management preferences.
 - Can log in securely to access the application's features.
 - Can create, edit, and delete tasks, specifying details such as task titles, descriptions, due dates, and priority levels.
 - Can view their task history and monitor progress over time, enhancing productivity insights.

- **Administrators:**

- Can add, edit, or remove user accounts, ensuring the integrity of the user database.
- Have control over task listings and categories. They can add, edit, or remove tasks as needed.
- Should manage the backend database, ensuring accurate records of tasks and user activities.
- Should monitor and maintain the overall functionality of the application, addressing any issues that may arise during operation.

2.4. Operating Environment

The To-Do List Application is designed to operate within a versatile computing environment, ensuring accessibility for a wide user base. The following details outline the supported operating systems, web browsers, and hardware configuration:

- Supported Operating Systems: Windows 10, Windows 11 and Ubuntu.
- Web Browser Compatibility: Microsoft Edge, Google Chrome and Mozilla Firefox.
- Hardware Configuration: 40 GB Hard Disk, Basic Input and Output Devices.

2.5. Assumptions and Dependencies

- **Assumptions:**

- The To-Do List Application assumes that users will have reliable internet access for seamless interaction with the application, including task creation, editing, and collaboration.
- Strict adherence to industry best practices, including coding standards, quality assurance, and robust security measures, is assumed throughout the development process.
- Users are expected to access the application using modern web browsers, including but not limited to Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox, for an optimal user experience.

- **Dependencies:**

- The core technology stack, including React for the frontend, and SpringBoot for the backend, is essential for the proper functioning of the To-Do List Application.
- The application is dependent on compatibility with modern web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari, to ensure a seamless user experience.

- The application assumes users will have hardware configurations meeting the specified requirements, including a 40 GB hard disk, a 15” color monitor, and a 122-key keyboard.
- Dependencies on adherence to industry best practices, including coding standards, quality assurance, and robust security measures, are essential for the successful development and operation of the To-Do List Application.

2.6. Software Requirements

- **Frontend:** HTML, CSS, Javascript, React
- **Backend:** Java, SpringBoot
- **Operating System:** Windows 10, Windows 11, Ubuntu
- **Database:** MySQL

2.7. Data Requirement

- **Input Data:**
 - Task details, including description, due date, and priority.
 - User account information.
 - Collaboration Data.
- **Output Data:**
 - User IDs and Profiles.
 - Task History.
 - Real-time Updates.

3. External Interface Requirement

3.1. GUI

The To-Do List Application boasts a user interface meticulously crafted for an intuitive and user-friendly experience. With a clean and modern design, the interface presents a visually appealing and clutter-free environment, ensuring a seamless interaction for users. Its responsive layout adapts seamlessly to various screen sizes and devices, allowing users to manage their tasks effortlessly on desktop or mobile platforms. Navigation menus are intuitively structured, facilitating easy access to different sections of the application. Users benefit from customization options, enabling personalization of their task management experience, including preferences for organization and visual elements.

4. System Features

- Users can effortlessly create tasks and provide detailed information, including descriptions, due dates, and priority levels.
- Tasks can be organized into categories or projects, allowing users to structure and manage their to-do lists efficiently.
- User registration enables account creation, providing users with personalized profiles to manage tasks and preferences.
- The application supports real-time collaboration, allowing users to update and synchronize task information seamlessly.
- Users can prioritize tasks based on urgency or importance, ensuring a clear focus on critical activities.
- The application maintains a history of completed tasks and provides analytical insights into task completion patterns and productivity.
- Users receive timely notifications for upcoming tasks, ensuring they stay informed and organized.

5. Other Non-Functional Requirements

5.1. Performance Requirement

- Targeting under 2 seconds for most pages to ensure a responsive and efficient user experience.
- The system should be scalable to handle increased user traffic during peak periods without compromising performance.
- Implementation of robust error handling mechanisms and user-friendly error messages for effective issue resolution.
- Efficient utilization of server resources, including memory and CPU, to support concurrent user interactions without degradation in performance.

5.2. Safety Requirements

- Regular data backups to prevent data loss and ensure data integrity.
- Ensuring system reliability to minimize downtime and provide a stable user experience.
- Development of a comprehensive disaster recovery plan to address system failures and ensure continuity.

5.3. Security Requirement

- Implementation of robust user authentication mechanisms to ensure secure access to the system.
- Use of data encryption techniques to protect sensitive information from unauthorized access.
- Implementation of RBAC to ensure that users and administrators have appropriate access privileges.
- Deployment of a firewall and intrusion detection system to safeguard against cyber threats and attacks.

5.4. Requirement Attributes

- Implementation of good version control practices for effective code management.
- Adoption of productive development methodologies to streamline the development process.
- Ensuring compliance with accessibility standards to create an inclusive user experience.
- The system should be easy to download and install for users, promoting accessibility.
- Implementation of quality assurance processes, including testing and code reviews, to ensure a reliable and high-quality system.

5.5. Business Rules

- Defining rules for creating, editing, and completing tasks.
- Specifying guidelines for user interactions and collaboration on tasks.

5.6. User Requirements

- Meeting customer requirements for creating, editing, and organizing tasks.
- Providing user manuals and online help to assist users in navigating the website.

6. Other Requirements

6.1. Class Diagram

