SRS for To Do List - Lincy A, Sofia Rani K, Suchithra J M, Susmitha B M

SOFTWARE REQUIREMENTS SPECIFICATION

TO-DO LIST

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this section is to articulate the fundamental objectives of this Software Requirements Specification (SRS) document. It serves as the compass that guides all stakeholders, including developers, project managers, and users, by offering a clear understanding of why this document is essential.

The primary goal of this section is to establish a unified, comprehensive reference for the TO-DO List project. By defining the document's purpose, we ensure that everyone involved comprehends the scope and goals of the project, facilitating effective communication, decision-making, and successful project execution. The SRS document acts as a foundational document that outlines the key functionalities, requirements, and specifications necessary for the development, testing, and deployment of the TO-DO List application, thus providing a blueprint for the project's success.

1.2 Document Conventions

This section outlines the specific conventions and standards employed throughout this Software Requirements Specification (SRS) document to ensure clarity, consistency, and ease of comprehension. These conventions are critical in maintaining uniformity in documentation practices, which aids in better communication among project stakeholders.

Document conventions include consistent formatting guidelines, such as font styles and sizes, section numbering, and a clear hierarchy of headings and subheadings. Additionally, it defines a standardized glossary of terms and abbreviations used within the document, ensuring that technical jargon is clearly defined for all readers, regardless of their level of familiarity with the subject matter. These conventions create a structured and accessible document that promotes efficient collaboration and understanding among project team members.

1.3 Scope of Development Project

This section elucidates the overall breadth and boundaries of the TO-DO List project, offering a comprehensive view of what the project aims to achieve and what falls outside its purview. It delineates the project's objectives, features, and functionalities, clarifying the specific goals that the development team seeks to accomplish.

The scope of the TO-DO List project encompasses the creation of a versatile task management application, tailored for individual users and collaborative teams. Its core functionalities include task creation, organization, prioritization, and categorization. While the application facilitates efficient task management, it does not extend to advanced project management capabilities, ensuring a clear distinction of its capabilities. By defining the scope, this section guides the project team, stakeholders, and users, providing a shared understanding of the project's boundaries and objectives.

1.4 Definitions, Acronyms and Abbreviations

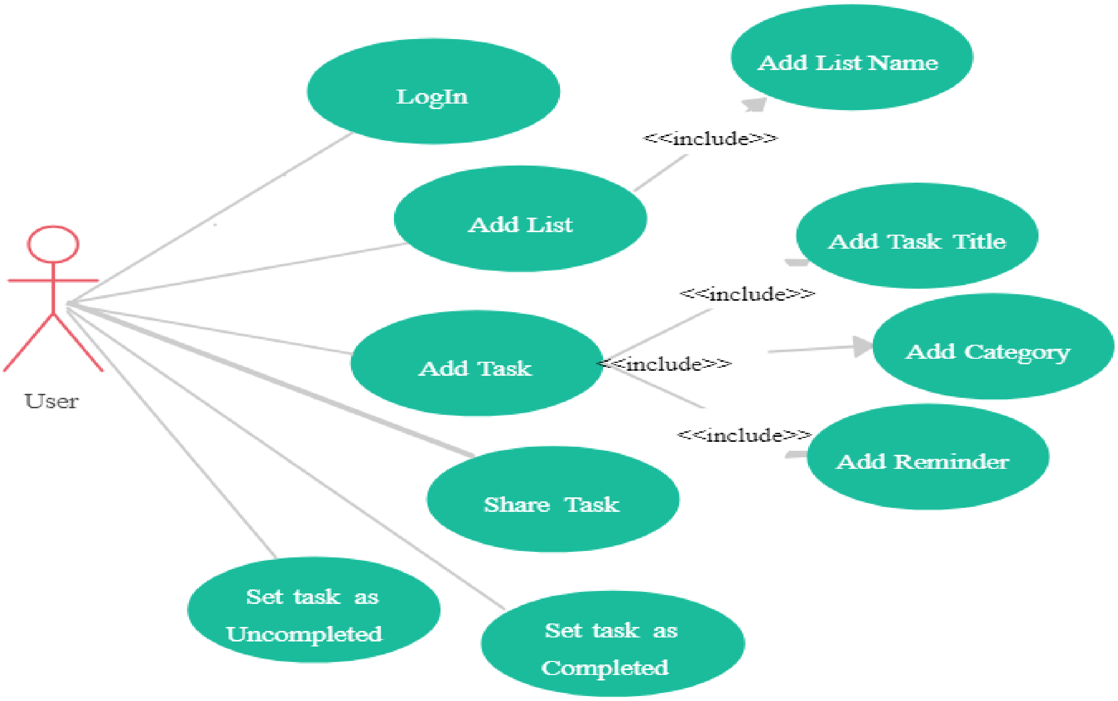
JAVA -> platform independence
SQL-> Structured query Language
ER-> Entity Relationship
UML -> Unified Modeling Language
IDE-> Integrated Development Environment
SRS-> Software Requirement Specification
ISBN -> International Standard Book Number

2. Overall Description

2.1 Product Perspective

In this section, we provide an in-depth overview of the TO-DO List application's place within the broader context of software systems. It elucidates how the TO-DO List system interacts with other software components or systems, emphasizing its role and significance.

The TO-DO List application stands as a standalone product, accessible through web browsers and mobile devices. While it can operate independently, it can also integrate with external systems via APIs, allowing for features like calendar synchronization and task collaborations with third-party applications. Understanding the product's perspective is crucial for comprehending how it fits into the digital ecosystem and how it can enhance users' task management experiences while seamlessly integrating with other software solutions they might use.



2.2 Product Function

This section delves into the core functions and capabilities of the TO-DO List application, providing a detailed breakdown of what users can expect when using the software. It outlines the primary features and functionalities that the application offers to fulfill users' needs effectively.
The TO-DO List application empowers users to create, edit, delete, categorize, and prioritize tasks efficiently. Users can customize their task lists, set due dates, receive reminders, and collaborate with others on shared tasks. It offers a user-friendly interface for seamless navigation and intuitive task management. By detailing these functions, this section offers a comprehensive understanding of the application's capabilities, helping both users and developers envision how the system will address their task management requirements.

2.3 User Classes and Characteristics

This section identifies and profiles the various user classes or roles within the TO-DO List application and elucidates their unique characteristics and responsibilities. It offers insights into the diversity of users who will interact with the system, shaping a more nuanced understanding of their distinct needs and expectations.
The user classes for the TO-DO List application primarily consist of regular users and administrators. Regular users possess the ability to manage their personal tasks, customize settings, and collaborate on shared tasks. Administrators, on the other hand, hold additional privileges, including user account management, category administration, and overseeing system-level configurations. Understanding these user classes and their characteristics is instrumental in tailoring the application's features and permissions to cater to the varying requirements and roles within the user base.

2.4 Operating Environment

This section outlines the specific environments in which the TO-DO List application will operate, ensuring that users and developers have a clear understanding of the technical prerequisites for using the software effectively.
The TO-DO List application is designed to be versatile, accommodating a wide range of operating environments. It is compatible with major web browsers such as Chrome, Firefox, and Safari, ensuring accessibility across different platforms. Additionally, the application supports popular operating systems like Windows, macOS, iOS, and Android, guaranteeing a seamless user experience on desktops and mobile devices. With a responsive design, it adapts gracefully to varying screen sizes, ensuring usability and performance across diverse operating environments. This information is vital for users and developers to ensure a smooth and optimized experience.

2.5 Assumptions and Dependencies

This section articulates the foundational assumptions made during the planning and development of the TO-DO List application. It also highlights the external dependencies on which the project relies, ensuring that all stakeholders are aware of these critical factors.

Assumptions encompass several aspects, including the presumption of a stable internet connection for real-time synchronization of tasks across devices. Additionally, it assumes that users will have access to compatible web browsers and mobile devices. Dependencies extend to third-party libraries and services integrated into the application, such as UI components and cloud-based storage. Recognizing these assumptions and dependencies is crucial for effective project planning, risk management, and resource allocation throughout the software development lifecycle.

2.6 Requirements - Hardware and Software Configuration

This section delineates the specific hardware and software prerequisites necessary to run the TO-DO List application seamlessly. It offers precise guidance to users and administrators regarding the technical specifications required for optimal system performance.

The TO-DO List application has modest hardware requirements, making it accessible to a broad user base. Users can access the application using standard personal computers, laptops, or mobile devices with internet connectivity. Compatible web browsers, including but not limited to Chrome, Firefox, and Safari, are required for accessing the web-based application. For mobile devices, dedicated apps may be available for various platforms. Understanding these hardware and software configurations ensures users can experience the application as intended, and administrators can make informed decisions regarding deployment and support.

2.7 Data Requirements

This section elaborates on the types of data that the TO-DO List application will store, manage, and interact with. It provides a comprehensive understanding of the data structures and entities crucial to the system's functionality.

The TO-DO List application will manage diverse data elements, including user profiles, task details (such as titles, descriptions, due dates, and priorities), and user preferences or settings. User data will be securely stored and organized to ensure efficient retrieval and manipulation. Additionally, the system may employ data synchronization mechanisms for real-time updates across devices. This section ensures that stakeholders are aware of the data handled by the application, contributing to data integrity, privacy, and system functionality considerations during development and maintenance.

3. External Interface Requirements

3.1 GUI

This section provides an in-depth description of the Graphical User Interface (GUI) design for the TO-DO List application. It outlines the visual and interactive components that users will encounter when interacting with the software, offering a comprehensive overview of the user interface.

The GUI of the TO-DO List application will feature an intuitive and user-friendly design. It will include elements such as task lists, task creation forms, filtering options, and a dashboard for task organization. The interface will prioritize ease of use and accessibility, ensuring that users can efficiently manage tasks, set priorities, and access relevant features. Detailed mockups, wireframes, or design guidelines may be included to provide a visual representation of the application's appearance and layout, aiding developers and designers in implementing the GUI effectively.

4. System Features

This section provides a comprehensive overview of the key features and functionalities that the TO-DO List application will offer to users. These features are the core components that empower users to efficiently manage their tasks and enhance their productivity.

Task Creation and Management: Users can easily create new tasks, specifying titles, descriptions, due dates, and priority levels. Existing tasks can be edited or marked as completed.

Categorization: The application allows users to categorize tasks into groups or categories, providing better organization and clarity.

Priority Setting: Users can assign priority levels to tasks, helping them focus on the most important and time-sensitive activities.

Due Date Reminders: The system offers due date reminders and notifications, ensuring that users don't miss important deadlines.

User Account Management: Users can create and manage their accounts, customize settings, and maintain personalized profiles.

Task Sharing: Collaboration features enable users to share tasks with colleagues, friends, or family members, fostering teamwork and coordination.

Search and Filtering: Robust search and filtering options help users quickly locate tasks based on keywords, categories, or due dates.

Cross-Platform Synchronization: The application ensures that tasks remain synchronized across multiple devices and platforms, allowing users to access their to-do lists anytime, anywhere.

Data Security: Stringent data security measures, including encryption, protect user data and privacy.

User-Friendly Interface: The application boasts an intuitive and visually appealing user interface, making task management simple and enjoyable.

These features collectively empower users to streamline their task management processes, boost productivity, and maintain efficient organization in their personal and professional lives. By providing these functionalities, the TO-DO List application aims to become an indispensable tool for individuals and teams seeking effective task management and collaboration solutions.

5. Non-Functional Requirements

5.1 Performance Requirements

Performance requirements are essential to ensure that the TO-DO List application operates smoothly and efficiently, meeting user expectations for responsiveness and speed. These requirements define the performance benchmarks and criteria that the system must achieve:

Response Time: The application should respond to user interactions within 2 seconds, providing a seamless experience.

Concurrent Users: The system must support at least 1000 concurrent users without a significant degradation in performance.

Data Load Time: Task lists, even with large data sets, should load within 3 seconds.

Data Synchronization: Synchronization across devices should occur in real-time or with a delay of no more than 10 seconds.

Uptime: The system should aim for 99.9% uptime, ensuring accessibility and reliability.

Meeting these performance requirements is crucial to delivering a high-quality user experience, enabling users to efficiently manage their tasks without frustration due to delays or downtime. Performance testing and monitoring will be essential to verify and maintain these benchmarks throughout the application's lifecycle.

5.2 Safety Requirements

Safety requirements are integral to ensuring the secure operation of the TO-DO List application, safeguarding user data and privacy. These requirements outline measures and safeguards to mitigate potential risks and vulnerabilities:

Data Encryption: All data transmission and storage must employ strong encryption protocols (e.g., TLS) to protect user information from unauthorized access.

Regular Backups: The system should perform automatic regular backups of user data to prevent data loss due to unforeseen events or technical issues.

Access Control: User authentication and authorization mechanisms should be robust, ensuring that users can only access and modify their own tasks and data.

Privacy Compliance: The application should adhere to relevant data privacy regulations (e.g., GDPR) and maintain transparent data handling practices.

Security Updates: Regular security updates and patches must be applied to mitigate vulnerabilities and potential security threats.
Incident Response: Establish protocols for incident detection and response to address any security breaches promptly.
These safety requirements prioritize the protection of user data and maintain a secure environment for task management. By adhering to these measures, the TO-DO List application ensures that user information remains confidential and the system operates with a high level of integrity and trustworthiness.

5.3 Security

Security is paramount in the TO-DO List application to safeguard user data and maintain user trust. The security measures and protocols are designed to protect against various threats and vulnerabilities:
User Authentication: Implement secure user authentication mechanisms, such as username-password combinations, two-factor authentication (2FA), or biometric authentication, to ensure only authorized users access the system.
Role-Based Access Control (RBAC): Assign specific roles and permissions to users, restricting their access to sensitive features or data based on their roles within the application.
Data Encryption: Employ robust encryption (e.g., AES) for data at rest and during transmission to prevent unauthorized access to user data.
Security Auditing: Monitor and log user activities to detect and respond to suspicious behavior or security incidents.
Regular Security Testing: Conduct regular security assessments, including penetration testing and vulnerability scanning, to identify and address potential weaknesses.
Security Updates: Stay vigilant in applying security patches and updates to keep the system protected against emerging threats.
By integrating these security measures, the TO-DO List application ensures the confidentiality, integrity, and availability of user data while providing a secure environment for task management and collaboration.

5.4 Requirements Attributes

Requirements attributes are essential for categorizing and managing the various requirements within the TO-DO List project. These attributes provide additional information about each requirement, making it easier for stakeholders to prioritize, track, and assess their importance and stability:
Priority: Each requirement is assigned a priority level, such as high, medium, or low, indicating its relative importance to the project's success.

Stability: Requirements may be classified as stable or subject to change. Stable requirements are less likely to undergo significant alterations during the project's development lifecycle.
Status: The status of each requirement is tracked, whether it's in the planning, implementation, testing, or completed phase, providing visibility into the development progress.
Ownership: Assigning ownership to requirements clarifies responsibility, ensuring that specific team members or stakeholders are accountable for their implementation.
Dependencies: Identify any dependencies between requirements, highlighting where one requirement relies on the completion of another.
By defining these attributes, the TO-DO List project can effectively manage, prioritize, and communicate requirements, streamlining the development process and facilitating collaboration among team members and stakeholders.

5.5 Business Rules

Business rules within the TO-DO List application are a set of directives that govern the behavior, processes, and interactions of the system to align with specific business or operational objectives. These rules define how the application operates and how users can engage with it:
Task Ownership: A user can only modify or delete tasks they've created to maintain data integrity and accountability.
Access Control: Administrators have the authority to manage user accounts and system configurations, while regular users are restricted from these functions.
Privacy: User data must remain confidential and should not be shared or accessible to unauthorized individuals or entities.
Task Sharing: Rules govern the process of sharing tasks, ensuring that users can collaborate effectively while respecting privacy and permissions.
Task Completion: Business rules dictate when a task can be marked as completed, typically when all required actions or criteria are met.
These business rules are fundamental for maintaining order, consistency, and security within the TO-DO List application, ensuring that it aligns with the intended operational and user requirements.

5.6 User Requirements

User requirements articulate the specific expectations and needs of the individuals who will interact with the TO-DO List application. These requirements reflect the functionalities and features users anticipate to effectively manage their tasks and enhance productivity:
Cross-Platform Accessibility: Users expect seamless access to their task lists across various platforms, including web browsers, desktop applications, and mobile devices.
Intuitive Interface: The user interface should be user-friendly, with clear navigation, easy task creation, and efficient organization features.

Real-Time Synchronization: Users require real-time synchronization of tasks across devices to maintain consistency and access their up-to-date task lists.

Task Sharing: Collaboration features should enable users to share tasks effortlessly with colleagues, friends, or family members.

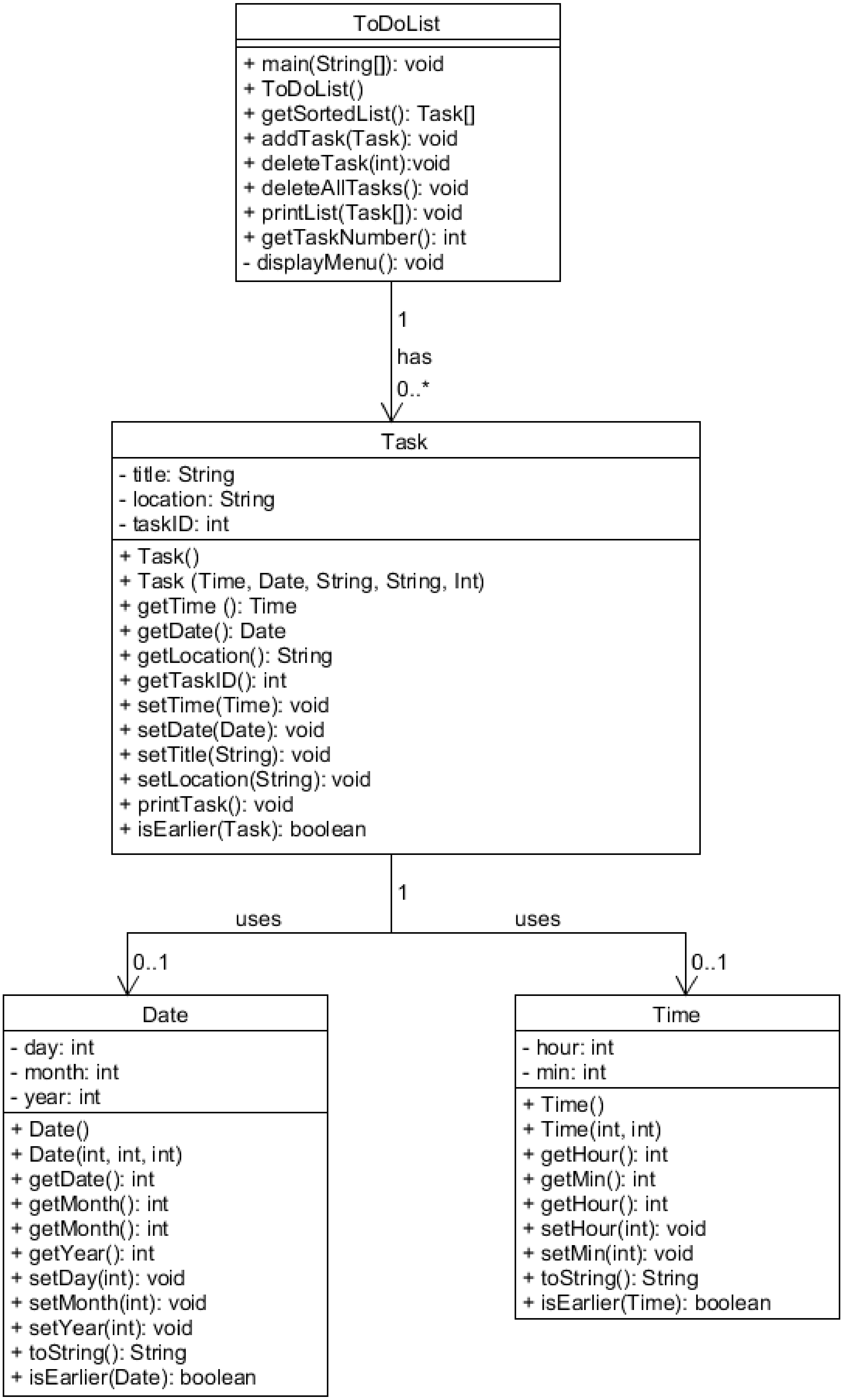
Privacy and Security: Users demand robust data security measures to protect their task-related information and maintain privacy.

Customization: Users appreciate the ability to customize task categories, priorities, and reminders to suit their individual preferences and needs.

Meeting these user requirements is pivotal in creating an application that caters to user expectations, ensuring a positive user experience and increased adoption of the TO-DO List system.

6. Class Diagram

A class diagram will offer a visual representation of the TO-DO List system's key classes, attributes, and relationships. It will assist developers in comprehending the software's architecture and structural components.


ToDoList
+ main(String[]): void
+ ToDoList()
+ getSortedList(): Task[]
+ addTask(Task): void
+ deleteTask(int):void
+ deleteAllTasks(): void
+ printList(Task[]): void
+ getTaskNumber(): int
- displayMenu(): void
1
has
0..*
Task
- title: String
- location: String
- taskID: int
+ Task()
+ Task (Time, Date, String, String, Int)
+ getTime (): Time
+ getDate(): Date
+ getLocation(): String
+ getTaskID(): int
+ setTime(Time): void
+ setDate(Date): void
+ setTitle(String): void
+ setLocation(String): void
+ printTask(): void
+ isEarlier(Task): boolean
1
uses
uses
0..1
0..1
Date
- day: int
- month: int
- year: int
+ Date()
+ Date(int, int, int)
+ getDate(): int
+ getMonth(): int
+ getMonth(): int
+ getYear(): int
+ setDay(int): void
+ setMonth(int): void
+ setYear(int): void
+ toString(): String
+ isEarlier(Date): boolean
Time
- hour: int
- min: int
+ Time()
+ Time(int, int)
+ getHour(): int
+ getMin(): int
+ getHour(): int
+ setHour(int): void
+ setMin(int): void
+ toString(): String
+ isEarlier(Time): boolean