LOYOLA-ICAM COLLEGE OF ENGINEERING AND TECHNOLOGY

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

TO-DO LIST

TEAM MEMBERS

JOE SEBASTIAN L(311120104023)

PRASANNA L(311120104045)

PRAVEEN SAMUEL S (311120104046)

VINITH RAJ A (311120104063)

DEPARTMENT OF COMPUTER SCIENCE

Chapter 1

INTRODUCTION

In the fast-paced rhythm of modern life, where demands and responsibilities abound, the humble To-Do List emerges as a beacon of organization and productivity. A To-Do List is more than just a collection of tasks; it is a strategic tool that empowers individuals to structure their time, prioritize activities, and achieve their goals with purpose. Whether scrawled on paper or meticulously managed through digital apps, the To-Do List is a universal companion that transforms chaos into order, providing a roadmap for success in both personal and professional spheres. In this guide, we'll explore the significance of To-Do Lists, unravel the art of crafting effective lists, and delve into the myriad ways this simple yet powerful tool can elevate your daily productivity and bring clarity to the complexities of your tasks. Get ready to unlock the potential of your time and streamline your journey to success with the artful mastery of the To-Do List.

1.1 DOCUMENT PURPOSE

A Software Requirements Specification (SRS) is a document that describes what the software will do and how it will be expected to perform. This Software Requirements Specification provides a complete description of all the functions and specifications of the to-do list. Todo apps are widely used, but different users use the apps differently depending on their particular needs. The main purpose of this smart to-do list is thatit provides ease to users as it helps users to set their tasks or any important meeting to be reminded of by using their voice commands and the algorithm then processes the natural human language into computer understandable language and set the reminder of the particular task for the user. It reduces human efforts of typing and without going through the process of providing every detail to the system, the user can just say the sentence along with date and time and the system will automatically set it for user.

1.2 PROJECT SCOPE

1.2.1 DESCRIPTION

A To-Do List is a systematic and organized inventory of tasks, activities, or objectives that an individual needs to accomplish within a specific timeframe. Whether handwritten on paper, digitally managed through apps, or a combination of both, the To-Do List serves as a practical tool for personal and professional task management. It typically includes details such as task descriptions, deadlines, priorities, and sometimes additional notes. The primary purpose of a To-Do List is to provide a clear and tangible way for individuals to plan, prioritize, and track their progress in order to enhance productivity and achieve their goals efficiently**.**

1.2.2 BENEFITS

A To-Do List enhances productivity and time management by providing organization, prioritization, and focus, reducing stress, fostering accountability, and offering motivation through the systematic tracking and completion of tasks.

1.3 INTENDED STAKEHOLDERS

The SRS document is addressed to:

- Developers who want to extend the program with new features.
- Testers who are interested in discovering possible flaws of the program and want to report them for improvement.
- All users of the program, who are interested in being informed about the capabilities, which to-do list, gives to them.

1.4 DOCUMENT OVERVIEW

The rest of this SRS

- chapter 2 gives a general description of the project. It gives what level of capability is anticipated from the client, some broad requirements while making the product and a few suppositions and conditions that are accepted.
- chapter 3 contains most significant highlights of the project portraying detail de-scription and requirements. It explicitly states the requisites which the product is required to convey. Functional requirements are given in this section. This area is composed essentially for the developers and portrays in specialized terms the functioning of the project, security performance and execution.
- chapter 4 consists of non-functional requirements describing them in detail.
- chapter 5 comprises of database information.

1.5 DEFINITIONS, ACRONYMS AND ABBREVI-ATIONS

To-Do List: A task management system used to organize and prioritize activities, often in digital form for easy access and tracking.

C#: A programming language developed by Microsoft, commonly used for Windows application development.

.NET Library: A collection of code and resources within the .NET framework, facilitating software development.

MS Visual Studio: Microsoft Visual Studio, an integrated development environment (IDE) used for coding, debugging, and deploying applications.

SQL: Structured Query Language, a domain-specific language for managing and querying relational databases.

OS: Operating System, software that manages computer hardware and provides services for other software.

Ecosystem: The interconnected environment within which the To-Do List project operates, including compatible platforms and devices.

User Interface (UI): The visual or interactive elements through which users interact with the To-Do List application.

Voice Commands: Spoken instructions enabling users to interact with the To-Do List project by adding, editing, or deleting tasks.

Local Database: A database stored on the user's device, ensuring data is kept locally rather than on a remote server.

Cross-Platform Compatibility: The ability of the To-Do List application to function seamlessly across different operating systems or devices.

Alarm Tone: A sound signal used to notify users of upcoming tasks or deadlines within the To-Do List.

Smart Categorization: Intelligent grouping of tasks based on predefined categories, enhancing organization and efficiency.

Guest Account: An account providing limited access to the To-Do List project without full user privileges.

Copyright: Legal protection for the original work, indicating exclusive rights to its use and distribution.

.NET Framework Classes: Reusable, object-oriented classes provided by the .NET framework to aid in software development.

Free Software: Software that is freely available for use, modification, and distribution.

Efficiency: The ability of the To-Do List project to perform tasks with minimal resource consumption and optimal speed.

Usability: The ease with which users can interact with and navigate the To-Do List application for effective task management.

Operational: Pertaining to the functionality and usage of the To-Do List software, often involving user registration and access control.

Performance Requirements: Specifications related to the speed, efficiency, and resource usage of the To-Do List application.

Security Requirements: Specifications ensuring the protection of user data and the To-Do List project from unauthorized access.

Software Quality Attributes: Characteristics such as efficiency, usability, and ecosystem support contributing to the overall quality of the To-Do List application.

1.6 DOCUMENT CONVENTIONS

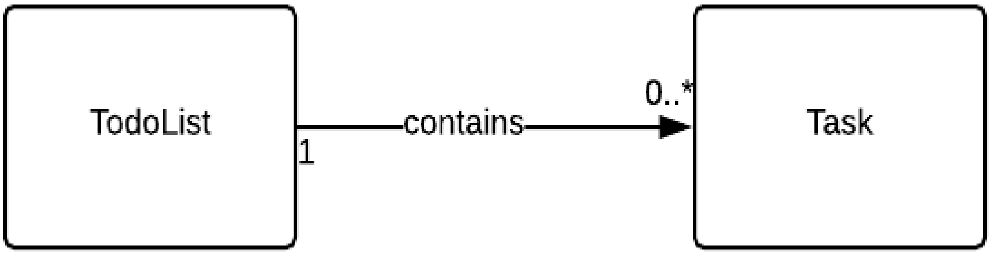
In general this document follows the IEEE formatting requirements. font size 12 is used throughout the document for text. Document text is single spaced .no special formatting techniques are used.

Chapter 2

OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

A To-Do List is a versatile and indispensable tool that serves as a user-centric solution for effective task management. It provides an intuitive interface for individuals to input, organize, and prioritize their tasks, fostering a seamless and efficient user experience. Features such as categorization, deadlines, and reminders enhance customization, ensuring that the product caters to diverse user needs and preferences. Additionally, integrations with other tools and platforms contribute to a holistic approach, allowing users to synchronize their task management across various aspects of their digital lives. The ongoing evolution of To-Do List products often involves user feedback loops, iterative improvements, and advancements in user interface design, reflecting a commitment to enhancing user satisfaction and overall productivity.



2.2 PRODUCT FUNCTIONALITY

The functionality of a To-Do List product is designed to streamline task management and enhance productivity. Key features include:

Task Entry: Users can easily add and input tasks into the To-Do List, including details such as task names, descriptions, and relevant information.

Prioritization: The product allows users to assign priorities to tasks, helping them focus on what needs immediate attention or is of higher importance.

Categorization: Users can categorize tasks based on different criteria such as project, context, or type, providing an organized structure for better clarity.

Deadline Setting: The ability to set deadlines for tasks helps users establish timelines and ensures timely completion of time-sensitive activities.

Reminders: The product often includes reminder functionalities, alerting users of upcoming tasks or deadlines through notifications.

Notes and Details: Users can attach additional notes, subtasks, or relevant details to each task, providing comprehensive information and context.

Cross-Platform Sync: Many To-Do List products offer synchronization across devices, allowing users to access and update their tasks seamlessly from various platforms.

Collaboration: In team settings, collaborative features enable users to share and collaborate on tasks, enhancing communication and coordination.

Progress Tracking: Users can track the progress of tasks, mark them as completed, and gain a visual representation of their accomplishments over time.

Integration: To-Do List products often integrate with other productivity tools and calendars, facilitating a cohesive user experience and avoiding duplication of effort.

Customization: Users can customize the appearance and settings of their To-Do List to align with their preferences, making the product adaptable to individual workflows.

Search and Filter: Robust search and filtering options allow users to quickly locate specific tasks or categories within their lists, saving time and improving efficiency.

2.3 USERS AND CHARACTERSTICS

To-Do Lists cater to a diverse range of users, including professionals, students, entrepreneurs, and parents, offering organizational focus, prioritization, flexibility, reminder mechanisms, accessibility across platforms, customization options, a motivational aspect through task completion, and collaborative features for team-oriented individuals.

2.4 OPERATION ENVIRONMENT

The operational environment of a To-Do List encompasses various platforms, devices, and settings where users engage with and manage their task lists, reflecting a versatile and adaptable tool for diverse contexts.

1. Digital Platforms: To-Do Lists operate seamlessly on digital platforms, including desktop applications, web browsers, and mobile apps, providing users with accessibility across a range of devices.

2. Mobile Devices: With the prevalence of smartphones and tablets, To-Do Lists are commonly accessed and updated on the go, allowing users to manage tasks from virtually anywhere.

3. Web Browsers: Online To-Do List platforms enable users to access and modify their task lists through web browsers, promoting flexibility and

convenience.

4. Desktop Applications: Installed software on desktop computers provides users with a dedicated interface for managing To-Do Lists, often offering additional features and customization options.

5. Cloud Services: To-Do Lists often integrate with cloud services, allowing users to sync their task data across multiple devices and ensuring real-time updates and accessibility.

6. Collaborative Environments: To-Do Lists operate effectively in collaborative settings, where teams can collectively manage tasks, share responsibilities, and communicate within a shared workspace.

7. Offline Functionality: Many To-Do List applications provide offline functionality, enabling users to access and update their task lists even when not connected to the internet.

8. Integrations: To-Do Lists seamlessly integrate with other productivity tools and applications, such as calendars and project management platforms, creating a cohesive operational ecosystem for users.

9. Customization Features: Users operate within personalized environments as To-Do Lists offer customization features, allowing them to tailor the appearance, categorization, and notification settings according to individual preferences.

10. Cross-Platform Compatibility: To-Do Lists maintain operational consistency across different operating systems, ensuring a uniform user experience whether on Windows, macOS, Android, or iOS.

Minimum System Requirement

Operating System:

For Desktop Applications:
Compatible with Windows 7 or later, macOS 10.12 or later, and popular Linux distributions.

For Mobile Apps:
Compatible with iOS 11 or later for Apple devices and Android 5.0 (Lollipop) or later for Android devices.

Processor:

For Desktop Applications:
A modern dual-core processor (e.g., Intel Core i3 or equivalent).

For Mobile Apps:
A processor compatible with the mobile device's operating system requirements.
Memory (RAM):

For Desktop Applications:
2 GB RAM or higher.

For Mobile Apps:
1 GB RAM or higher.

Storage:

For Desktop Applications:
At least 100 MB of available disk space.

For Mobile Apps:
The app size plus additional space for data storage. This can vary but typically ranges from 20 MB to 100 MB.

Graphics:

For Desktop Applications:
A graphics card and monitor that support a resolution of at least 1024x768.

For Mobile Apps:
Compatible with the device's screen resolution.

Web Browser:

If the To-Do List application is web-based, it should be compatible with modern web browsers like Chrome, Firefox, Safari, or Edge.

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

In the intricate development landscape of a To-Do List application, a myriad of design and implementation constraints shapes its functionality and user experience. With the imperative of cross-platform usability, the application must harmonize seamlessly across diverse environments, spanning desktops, web browsers, and mobile devices, each demanding meticulous attention to user interface intricacies. A crucial constraint lies in the necessity for offline functionality, compelling developers to engineer a system that not only caters to users in connectivity-challenged scenarios but also gracefully syncs data once a connection is reestablished.

Accessibility considerations further intricately weave into the design fabric, demanding the application be attuned to diverse user needs, from screen reader compatibility to keyboard navigation, ensuring an inclusive experience for all users, regardless of abilities. As the To-Do List often contains sensitive information, a formidable constraint emerges—security and privacy must be paramount, dictating the implementation of robust measures such as encryption and secure authentication protocols.In the interconnected landscape of digital productivity tools, integration constraints emerge, necessitating a design that fluidly integrates with other platforms, calendars, and project management tools, facilitating a cohesive workflow for users. Scalability becomes a pivotal constraint, anticipating the potential expansion of user base and tasks, mandating a robust infrastructure and data storage system to ensure optimal performance even as the demands on the system increase.

Users' desire for customization introduces a constraint that steers the design towards flexibility, allowing users to tailor the application's appearance and functionality to suit their preferences. The collaborative nature of work introduces a constraint in the form of collaboration features, demanding an implementation that supports real-time updates, shared lists, and nuanced user permissions for seamless teamwork.

Ensuring a consistent and intuitive user interface across various devices presents a multifaceted challenge, requiring a design that gracefully adapts to different screen sizes while maintaining a cohesive visual identity. Resource constraints on mobile devices add another layer of complexity, compelling developers to optimize the application for performance and battery efficiency.

Crucially, regulatory compliance constitutes a non-negotiable constraint, demanding adherence to data protection and privacy regulations through features like encrypted storage and user consent mechanisms. Navigating these intricate design and implementation constraints is not merely a technical endeavor but a commitment to crafting a To-Do List application that not only meets user expectations but also stands resilient in the dynamic and demanding landscape of digital productivity.

2.6 USER DOCUMENTATION

There's not such user documentation needed for this product as this product is prettysimple and everyone of every age group can easily hold a grasp on it. Adding the taskyou wish to work on and provide the deadline is all you want to do for this product towork. The rest will be handled by the ToDo program. All you need to know is what taskyou should be reminded of and when and the ToDo assured everything for you.

2.7 ASSUMPTIONS AND DEPENDENCIES

Assumptions and dependencies inherent in a To-Do List application include the expectation that users will consistently engage with the tool, assuming their accessibility to reliable internet for synchronization features. Compatibility across various devices is assumed, prompting the design to prioritize cross-device responsiveness. Trust in the application's data privacy is assumed, necessitating robust security measures. User understanding of task prioritization is assumed, influencing the inclusion of intuitive features for assigning and comprehending priorities. Collaboration intent is assumed, leading to the integration of collaborative features such as shared lists. Regular updates and support are expected, as users rely on continuous improvements for functionality and compatibility. Assumptions about user preferences drive the provision of customization options, while the assumption of integration interest influences support for external tool connectivity. Notification comprehension is assumed, guiding the implementation of a timely and user-friendly notification system. Lastly, adherence to regulations is expected, necessitating ongoing monitoring. .

Chapter 3: SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 USER INTERFACES

The To-Do List application will greet users with a home page featuring a prominent button to proceed. Upon interaction, users will be directed to a screen allowing them to add tasks using voice commands, schedule tasks, or delete them.

3.1.2 HARDWARE INTERFACES

The application's interaction with hardware is facilitated through C# and .NET Library, creating a seamless bridge between user instructions and system processing. Compatibility is optimized for Windows machines, leveraging the Microsoft ecosystem.

3.1.3 SOFTWARE INTERFACES

The To-Do List operates efficiently on Windows machines using MS Visual Studio. The system utilizes a local database, prioritizing simplicity for personal and professional use. While Windows is the preferred OS, the application can run on Linux or MacOS.

3.1.4 COMMUNICATION INTERFACE

The To-Do project is designed for local availability, with no involvement of cloud services or cross-platform support.

3.2 FUNCTIONAL REQUIREMENTS

The To-Do List application's functional requirements are categorized into tasks, views, and smart categorization, providing users with a comprehensive task management system.

TASKS

NEW TASK

User Requirement: Users can add tasks manually or through voice commands, setting names, dates, times, and reminders.
System Requirements: Fields must be completed, and tasks stored in the database.

EDIT TASK

User Requirement: Users can edit task names, dates, or days.
System Requirements: Edited information is updated in the database.

DELETE TASK

User Requirement: Users can delete tasks.
System Requirements: Confirmation prompts for task deletion.

VIEW

COMPLETED TASKS

User Requirement: Users can view completed tasks.
System Requirements: Completed tasks are displayed.

INCOMPLETED TASKS

User Requirement: Users can view incomplete tasks.
System Requirements: Pending tasks are displayed.

SMART CATEGORIZATION

CATEGORY ASSIGNMENT

Function: Tasks are intelligently categorized for better organization.
Description: Tasks are divided into predefined categories, facilitating easy management for users.

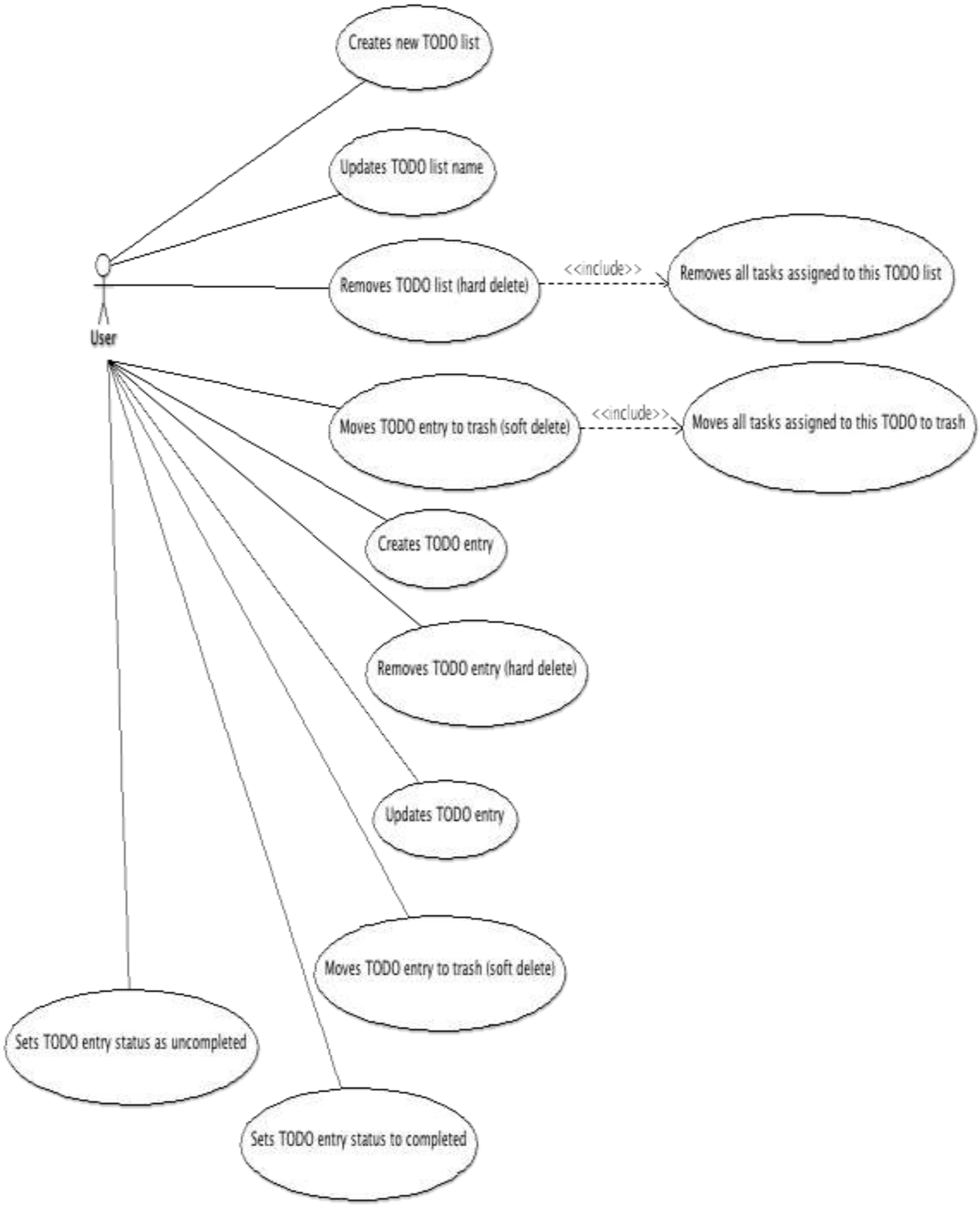
ALARM TONE

User Requirement: Users can select alarm tones.
System Requirements: The system reminds users of tasks with the selected alarm tone.

3.3 BEHAVIOR REQUIREMENTS

3.3.1 USE CASE VIEW

The primary user can add, edit, view, and delete using voice commands or manual input. Tasks are intelligently categorized, enhancing organization. tasks


Creates new TODO list
Updates TODO list name
User
Removes TODO list (hard delete)
<<include>>
Removes all tasks assigned to this TODO list
Moves TODO entry to trash (soft delete)
<<include>>
Moves all tasks assigned to this TODO to trash
Creates TODO entry
Removes TODO entry (hard delete)
Updates TODO entry
Moves TODO entry to trash (soft delete)
Sets TODO entry status as uncompleted
Sets TODO entry status to completed

Chapter 4: OTHER NON-FUNCTIONAL REQUIREMENTS

4.1 PERFORMANCE REQUIREMENTS

The To-Do List application is designed with minimal memory, disk space, and processing power requirements. Voice command functionality relies on a functional microphone. Users can edit or delete any task, completed or not.

4.2 SAFETY AND SECURITY REQUIREMENTS

The program encourages public use through free distribution.
Files include copyright information.
Database security is ensured through username and password protection.
.NET framework classes are open for user access.

4.3 SOFTWARE QUALITY ATTRIBUTES

4.3.1 EFFICIENCY

The program focuses on providing users with ease and functionality at no cost, ensuring minimal resource consumption.

4.3.2 USABILITY

The software is user-friendly, catering to both professional employees and students, offering diverse usage scenarios.

4.3.3 ECOSYSTEM

The application provides a robust ecosystem, allowing cross-platform functionality across Windows PCs and mobile devices.

4.3.4 OPERATIONAL

The software requires user registration for full operational access, with no guest accounts allowed.

Chapter 5: DATABASE REQUIREMENTS

The To-Do List application utilizes a local SQL database, categorized into tables such as Pending Tasks and Completed Tasks. Entities are further divided based on task categories. The lightweight database system ensures compatibility with various PCs without the need for a dedicated server.