Documentation

1. **SRS**

﻿*Table of Contents*

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. Overall Description

2.1 Product Perspective 2.2 Product Functions

2.3 User Characteristics 2.4 General Constraints

2.5 Assumptions and Dependencies

3. Specific Requirements

﻿ 3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2. Functional Requirements

3.3 Performance Requirements

3.4 Design Constraints

3.5 Attributes

1. Introduction

1.1 Purpose

The purpose of this Windows Forms task management application is to provide a user-friendly platform for managing tasks and projects within a single project environment. It allows employees to create tasks, assign and track their progress, and collaborate on subtasks.

The application also includes features such as user registration and login, search functionality, a backlog for deadline-expired tasks, and a commenting system for subtasks.

* 1. Scope

The scope of this application is limited to managing tasks within a single project. It does not support multiple projects. Each employee represents a user account and can assign each other specific tasks. The application facilitates task creation, assignment, and tracking for individual employees.

Additionally, it allows users to manage subtasks, log hours worked, and keep a record of completed subtasks. The application's focus is on streamlining task management and enhancing collaboration among project team members.

1.3 Definitions, Acronyms, and Abbreviations

* Task: A unit of work assigned to an employee within the project.
* Subtask: A sub-division of a task that represents a smaller unit of work.
* Backlog: A list of tasks that have missed their deadline and require attention.
* User Registration: The process of creating a new user account within the application.
* User Login: The process of accessing the application with a registered user account.
* IHelpU: A dialog which help the employee to count working hours in a day.

1.4 References

No specific external references are required for the development of this application. However, the development team may refer to the SQLite documentation for proper usage and integration of the SQLite database within the application. The team will also follow best practices and guidelines for Windows Forms application development to ensure a robust and user-friendly system.

1.5 Overview

This SRS document provides a comprehensive description of the requirements and functionalities of the Windows Forms task management application. It covers the specific requirements, user interfaces, performance expectations, design constraints, attributes, and other relevant details to guide the development process effectively.

1. Overall Description

2.1 Product Perspective

The application functions as a standalone task management system, allowing users to create, assign, and track tasks within a single project. It operates independently of any external systems or dependencies. The application's user interfaces are designed to offer a cohesive and intuitive experience, enabling users to create, assign, and monitor tasks and subtasks without the need for internet connectivity.

The Windows Forms task management application operates as a standalone software solution, providing an integrated environment for task management within a single project. It utilizes an SQLite database for efficient data storage and retrieval.

2.2 Product Functions

The application includes various functions, such as user registration and login, task and subtask creation, task assignment to employees, subtask commenting for tracking progress, a search bar for quickly locating tasks, and a backlog feature to prioritize deadline-expired tasks.

Additionally, it provides a profile page for each employee to manage their personal information and settings. The helper dialog helps employees track their working hours and provides a convenient time management tool.

2.3 User Characteristics

The intended users of the application are employees working on a specific project. These users may possess varying levels of technical expertise, but the application is designed to accommodate users with basic computer skills. The user interfaces prioritize ease of use, with intuitive controls and clear navigation, ensuring that employees can seamlessly interact with the application to manage their assigned tasks and subtasks efficiently.

2.4 General Constraints

The Windows Forms task management application has certain constraints and requirements. It is designed to run on Windows operating systems and utilizes the SQLite database for local storage and retrieval of data. The application does not rely on internet connectivity, making it suitable for both online and offline usage. While the application's primary focus is on task and subtask management within a single project, it provides a comprehensive set of features to meet the project team's needs effectively.

2.5 Assumptions and Dependencies

The application assumes that users have access to a Windows operating system that supports the required .NET Framework version for Windows Forms development. It further assumes the presence of SQLite as a reliable and performant database engine for storing user accounts, tasks, subtasks, and related information securely. The application does not have external dependencies on remote servers or online services, allowing users to work with their tasks and subtasks seamlessly, regardless of internet connectivity.

1. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The main screens include an "All-Tasks" page, where employees can view and interact with all the tasks, an "All-Employees" page that displays a list of all registered employees, a profile page for each employee to manage their personal information and settings, and a helper dialog to assist employees in counting working hours accurately. There is also a “Backlog” page where employees can see and search expired tasks in case they want to assign them. In addition, each task and subtask have specific pages where all the details are presented.

The user interfaces of the application should be intuitive, visually appealing, and provide an efficient workflow for task management. The user interfaces should prioritize ease of use and responsiveness to enhance the overall user experience.

3.1.2 Hardware Interfaces

The Windows Forms task management application does not have any specific hardware interface requirements. It is designed to run on standard desktop or laptop computers with compatible Windows operating systems, requiring no additional hardware peripherals for its normal functionality.

3.1.3 Software Interfaces

The application utilizes the SQLite database management system for data storage and retrieval. It interacts with the SQLite database using appropriate data access libraries and APIs to ensure seamless integration. The software interfaces should be implemented in a way that allows efficient and secure communication between the application and the SQLite database.

These interfaces include the utilization of Windows Forms controls, event handling mechanisms, and graphical user interface (GUI) components to provide a rich and interactive user experience. The application relies on software interfaces to capture user input, handle user interactions, and update the display accordingly. By leveraging these software interfaces, the application ensures an intuitive user interface that enables users to efficiently create tasks, assign them to employees, manage subtasks, and track progress within the application's graphical environment.

3.1.4 Communication Interfaces

As the application operates as a standalone system and does not rely on internet connectivity, there are no specific communication interfaces required. All data operations and interactions are handled locally within the application, ensuring that employees can manage tasks and subtasks without the need for network communication.

3.2 Functional Requirements

The functional requirements of the application include:

* User Registration and Login: The application should provide a user registration feature that allows employees to create their accounts with unique usernames and passwords. It should also enable registered users to log in securely using their credentials.
* Task and Subtask Creation: Employees should be able to create new tasks, providing relevant details such as task name, description, and deadline. Additionally, they should have the ability to create subtasks and associate them with the corresponding tasks.
* Task Assignment: The application should allow task assignment to individual employees, ensuring that a task is assigned to only one employee at a time.
* Subtask Commenting and Time Tracking: Employees should be able to comment on subtasks, providing updates on progress, logging hours worked, and maintaining a record of activities performed. This functionality helps in collaboration and tracking of task completion.
* Search Functionality: The application should include a search bar that enables users to search for specific tasks based on task names or keywords, facilitating efficient task discovery and retrieval.
* Backlog for Deadline-Expired Tasks: The application should provide a dedicated backlog section to highlight tasks that have missed their deadlines, allowing users to prioritize and address them accordingly.

These functional requirements are essential for creating a comprehensive task management system that empowers employees to effectively manage their tasks and collaborate within the project environment.

3.3 Performance Requirements

The performance requirements of the Windows Forms task management application focus on ensuring efficient and responsive behavior, providing a smooth user experience. These requirements include:

* Responsiveness: The application should respond promptly to user interactions, such as creating tasks, assigning subtasks, and updating task status.
* Loading Time: The application should load quickly, allowing users to access their tasks and subtasks without significant delays.
* Data Retrieval: The retrieval of task and subtask data should be fast and efficient, even when dealing with a large number of records.
* Search Performance: The search functionality should deliver search results in a timely manner, enabling users to find specific tasks quickly.
* Scalability: The application should be designed to handle a growing number of users, tasks, and subtasks without sacrificing performance.

These performance requirements aim to ensure that the application operates smoothly and provides a responsive experience to users, regardless of the size of the task and subtask dataset.

3.4 Design Constraints

The design constraints of the application are influenced by the use of Windows Forms and the SQLite database. These constraints include:

* Windows Forms Design Guidelines: The application should adhere to established design principles and guidelines for Windows Forms applications, ensuring a consistent and familiar user interface across different screens and controls.
* UI Responsiveness: The user interfaces should be designed to be highly responsive, minimizing lag and delays in user interactions.
* SQLite Integration: The application should properly integrate with the SQLite database, leveraging its features for efficient data storage, retrieval, and management.
* Platform Compatibility: The application should be compatible with Windows operating systems, supporting the required version of the .NET Framework and ensuring seamless operation within the targeted Windows environments.

These design constraints provide a framework for developing a user-friendly and well-integrated task management application while considering the limitations and requirements of the underlying technologies.

3.5 Attributes

The attributes of the Windows Forms task management application encompass additional characteristics that contribute to its overall functionality and usability. These attributes include:

* Security: The application should ensure the security and confidentiality of user data by implementing appropriate measures such as data encryption and secure user authentication.
* Error Handling: The application should have robust error handling mechanisms in place to detect and handle exceptions, providing informative error messages to users when necessary.
* Accessibility: The application should strive to meet accessibility standards, ensuring that individuals with disabilities can access and use the application effectively.
* Localization: The application should support localization to accommodate users from different regions by providing translations of user interfaces and content into multiple languages.
* Backup and Recovery: The application should include provisions for data backup and recovery, allowing users to restore their data in case of accidental loss or system failures.

These attributes enhance the overall quality and usability of the application, ensuring that it meets users' expectations and industry standards.

1. The usage of the program from the perspective of a non-programming user

* User Registration: The user begins by creating an account within the application by providing their necessary information, such as name, email address, and a secure password.
* Logging In: Once registered, the user can log in to the application using their chosen credentials. The login process verifies the user's identity and grants access to the application's features.
* Task Creation: After logging in, the user can create new tasks by navigating to the appropriate section or using the provided options. The user fills in the task details, such as the task name, description, and deadline. They can also add any necessary subtasks associated with the task.
* Task Assignment: The user can assign the task to themselves or another employee working on the project. This ensures that each task is assigned to a specific person responsible for its completion.
* Subtask Management: As the user works on a task, they can create subtasks associated with it. The user can add comments to subtasks, logging the hours worked or providing updates on the progress. This feature helps keep track of the work done and facilitates collaboration.
* Task Monitoring: The user can access the "All-Tasks" page, where they can view all the tasks assigned to them. This page provides an overview of task statuses, deadlines, and any associated subtasks. The user can also use the search bar to find specific tasks based on names or keywords.
* Profile Management: The user can access their profile page to manage their personal information, update their password, and customize their preferences within the application.
* Helper Dialog: To assist with time tracking, the user can utilize the helper dialog provided by the application. This dialog helps employees accurately count their working hours and maintain a record of activities performed.
* Subtask Commenting: Users can add comments to individual subtasks associated with a task. These comments serve as a communication tool, allowing users to provide updates on their progress, discuss any challenges or obstacles, or share relevant information related to the subtask.
* Time Tracking: Users can use comments to log the hours worked on a particular subtask. By including the amount of time spent, users can accurately track their working hours and maintain a record of their activities within the application. This feature aids in monitoring individual progress and helps project managers or supervisors assess overall task completion.

1. UML Diagrams (See UMLs folder for all diagrams)
2. Class Diagrams

* Persistence
* View
* Presenter
* Whole Project

1. Activity Diagrams

* Status Change
* Add Comment to Subtask
* Add Subtask to Task

1. Sequence Diagrams

* Log In
* Search Task

1. Utility of Use-Case

* Registration
* Add Task/Subtask
* Delete Task

1. Application States Screenshots (See Screenshots folder)

* Log In
* Log In Exception
* Sign Up
* Email Registration Exception
* Profile
* Tasks Page
* Create Task
* Subtask
* Users Page
* IHelpU