Software Requirements Specification

for

Calendar Scheduler

Version 1.0 approved

Prepared by

R.Kalaiyarasi

J.Christy Evangline

A.Monika

S.V.Niveditha

06/10/2023

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this SRS document is to define the requirements for the Calendar Scheduler project. This document serves as a foundation for the design and development of the software.

1.2 Project Scope

The Calendar Scheduler is a web-based application that allows users to create, manage, and share events and appointments. It aims to simplify scheduling and time management.

1.3 Definitions, Acronyms, and Abbreviations

- *SRS: Software Requirements Specification*
- *UI: User Interface*
- *ERD: Entity Relationship Diagram*

1.4 Overview

This Software Requirements Specification (SRS) outlines the requirements for the Calendar Scheduler project. The Calendar Scheduler is a web-based application designed to facilitate event management, scheduling, and collaboration. This document provides a comprehensive understanding of the project's purpose, scope, and key functionalities.

The SRS is structured to cater to various stakeholders, including developers, project managers, testers, marketing staff, users, and documentation writers. It includes sections detailing functional requirements, system features, external interfaces, non-functional requirements, and other essential aspects of the system.

By following this document, readers will gain insights into the project's objectives, system features, and quality attributes. The SRS serves as a foundation for development, testing, and project management activities, ensuring that the Calendar Scheduler meets the expectations and needs of its users and stakeholders. Readers are encouraged to explore the specific sections that align with their roles and interests to comprehend the project's intricacies thoroughly.

1.5 References

IEEE SoftwareIEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications," IEEE, New York, 1998 Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications," IEEE, New York, 1998.

2. Overall Description

2.1 Product Perspective

The "Calendar Scheduler" is a standalone application that can be used on its own or integrated with other calendar or project management tools. It is not a component of a larger system, but it can interact with other systems through its synchronization feature.

2.2 Product Features

The main functions of the "Calendar Scheduler" application include:

***Creating events:** Users can create new events on the calendar.*

***Viewing events:** Users can view the details of existing events.*

***Editing events:** Users can change the details of existing events.*

***Deleting events:** Users can remove existing events from the calendar.*

***Setting reminders:** Users can set reminders for upcoming events.*

***Sharing events:** Users can share events with others through email or a shared calendar.*
***Synchronizing with other applications:** Users can synchronize their calendar data with other applications.*

2.3 User Classes and Characteristics

There are three classes of users who will interact with the "Calendar Scheduler" application:

***Individual Users:** These are private users who use the application for personal time management. They need to manage their personal schedules and tasks.*

***Team Users:** These users use the application within a team context. They need to coordinate their tasks and meetings with their team members.*

***Organizational Users:** These users use the application across an entire organization. They need to schedule and track company-wide events.*

2.4 Operating Environment

The “Calendar Scheduler” application will operate on various platforms, including Windows, macOS, Linux, iOS, and Android. It will be a cross-platform application that can run on desktop computers, laptops, tablets, and smartphones.

2.5 Design and Implementation Constraints

The “Calendar Scheduler” application must comply with all applicable privacy and data protection laws. It must also be compatible with the latest versions of the operating systems mentioned above. The application must be designed to be user-friendly and intuitive to use.

2.6 User Documentation

User documentation will be provided to help users understand how to use the “Calendar Scheduler” application. The documentation will include a user manual that explains how to use each feature of the application, online help that provides answers to common questions, and video tutorials that demonstrate how to use the application.

2.7 Assumptions and Dependencies

It is assumed that users have basic computer skills and can navigate the user interface of the application. The application depends on the device’s operating system to function correctly. It also depends on the availability of an internet connection for the synchronization feature.

3. System Features

This section provides a detailed description of each system feature and its functional requirements. For each feature, include:

A brief description
The inputs and outputs related to the feature
The functionality of the feature
The user interaction related to the feature
Any error conditions that could arise during the use of the feature

3.1 System Feature 1: Creating Events

3.1.1 Description and Priority

Description: This feature allows users to create new events on the calendar by providing event details such as title, date, time, location (optional), and description (optional).
Priority: High

3.1.2 Stimulus/Response Sequences

- ○ *User selects the "Create Event" option.*
- ○ *User inputs event details (title, date, time, optional location, and optional description).*
- ○ *System validates the input data.*
- ○ *If input is valid, the system adds the new event to the calendar.*

○ *If input is invalid or incomplete, the system provides an error message and prompts the user to correct the input.*

3.1.3 Functional Requirements

REQ-1: The system must provide a user interface with a "Create Event" option.
REQ-2: Users must be able to input event details, including a title, date, and time.
REQ-3: Users should have the option to provide a location and description for the event, but these fields are optional.
REQ-4: The system must validate the input data, ensuring that the date and time are in the correct format and that the date is in the future.
REQ-5: If the input data is valid, the system must add the new event to the calendar, and the event details should be displayed.
REQ-6: If the input data is invalid (e.g., an invalid date format or a past date), the system should display an error message and prompt the user to correct the input.

3.2 System Feature 2 Viewing Events

3.2.1 Description and Priority

Description: This feature allows users to view the details of existing events on the calendar.
Priority: Medium

3.2.2 Stimulus/Response Sequences

- ○ *User navigates to the calendar.*
- ○ *User selects an existing event.*
- ○ *System displays the details of the selected event.*

3.2.3 Functional Requirements

REQ-1: The system must provide a user-friendly calendar interface.
REQ-2: Users should be able to select an existing event on the calendar.
REQ-3: The system should display the details of the selected event, including the title, date, time, location, and description.

4. External Interface Requirements

4.1 User Interfaces

The user interface of the "Calendar Scheduler" application will be graphical and intuitive. It will include a calendar view that displays the events on the calendar, an event creation form that allows users to create new events, and settings options that allow users to customize the application.

4.2 Hardware Interfaces

The "Calendar Scheduler" application does not require any specific hardware interfaces. It will run on any device that meets the minimum system requirements.

4.3 Software Interfaces

The "Calendar Scheduler" application will interface with the device's operating system and any applications with which it synchronizes. The application will use standard APIs to interact with these systems.

4.4 Communications Interfaces

The "Calendar Scheduler" application will use standard internet protocols for synchronization and event sharing. The application will encrypt all data transmissions to ensure data security.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The "Calendar Scheduler" application should respond quickly to user inputs and should be able to handle a large number of events without performance degradation. The application should also synchronize data with other applications quickly and efficiently.

5.2 Safety Requirements

The "Calendar Scheduler" application should handle user data securely. The application should not allow unauthorized access to the data and should provide features for users to control who can view and edit their events.

5.3 Security Requirements

The “Calendar Scheduler” application should encrypt all sensitive data, such as user passwords and event details. The application should require user authentication for access and should provide features for users to manage their account security.

5.4 Software Quality Attributes

The Calendar Scheduler aims to deliver a high-quality software product that meets the expectations of both customers and developers. Several software quality attributes are of significant importance:

5.4.1 Usability:

Priority: High

Description:

The

application

should be

user-friendly

and intuitive,

allowing users

to efficiently

create,

manage, and

share events.

Users should

find it easy to

navigate and

interact with

the system.

This will be

measured

through user

testing and

feedback.

5.4.2 Reliability:

Priority: High
Description: The system should be dependable and stable, ensuring that events and data are accurately stored and displayed. It should have a low probability of crashes or data loss. Reliability will be measured through system uptime and error tracking.

5.4.3 Security:

Priority: High
Description: Data security and user privacy are of utmost importance. The application should protect user data and ensure secure transmission when synchronizing with other applications.

5.4.4 Maintainability:

Priority: Medium
Description: The codebase should be well-structured and modular, making it easier for developers to maintain, extend, and update the software. This will be evaluated based on code maintainability metrics.

5.4.5 Interoperability:

Priority: Medium
Description: The application should be able to synchronize with various calendar and project management tools, promoting interoperability. It should be compatible with standard data formats. Successful synchronization and data exchange will be a key measure.

5.4.6 Flexibility:

Priority: Medium
Description: The system should accommodate user preferences for customization and event categorization. Users should have the flexibility to adapt the application to their specific needs.

5.4.7 Availability:

Priority: Medium
Description: The system should be available to users with minimal downtime. Any scheduled maintenance should be communicated in advance, and the application should have a high uptime rate.

5.4.8 Testability:

Priority: Low

Description:

The application should be designed with testability in mind, allowing for efficient testing of different features. This involves creating clear test cases and providing logs for troubleshooting.

6. Other Requirements

The “Calendar Scheduler” application should be easy to install and uninstall. The application should provide clear error messages and should recover gracefully from errors. The application should also provide features for users to customize the look and feel of the application

Appendix A: Glossary

- *SRS (Software Requirements Specification): A document that outlines the functional and non-functional requirements for a software project, serving as a foundation for its design and development.*
- *UI (User Interface): The graphical and interactive elements of the software that users interact with to use the application.*
- *ERD (Entity Relationship Diagram): A visual representation of the relationships between entities or data in a database.*
- *API (Application Programming Interface): A set of rules and protocols that allow different software applications to communicate and interact with each other.*

Appendix B: Analysis Models

This appendix may include visual representations of the system, such as data flow diagrams, class diagrams, or entity-relationship diagrams, to provide a better understanding of the system's structure and relationships. These models would be specific to the Calendar Scheduler project and are created based on the project's requirements and design.

Appendix C: Issues List

The Issues List for the Calendar Scheduler project is a dynamic document that tracks various open issues and concerns throughout the development process. It's crucial for ensuring that all aspects of the SRS are properly addressed. Here are a few sample entries:

Issue ID	*Description*	*Status*
Issue-1	*Clarify data synchronization requirements*	*In Progress*
Issue-2	*Confirm the specific security protocols*	*Open*
Issue-3	*Define the process for handling user feedback*	*Pending*
Issue-4	*Decide on the default timezone setting*	*Open*