

SOFTWARE REQUIREMENTS SPECIFICATION

For

Calendar Scheduler

Prepared by:-

Dhivya

Karunambika

Prithika

Subhashini

Vigneshwaran

Academic Year: 2023-2024

1. Introduction

The Calendar Scheduler System is a software application designed to assist users in managing their schedules, appointments, and events efficiently.

1.1 Purpose

The purpose of a calendar scheduler is to help people manage their time and schedules more effectively. It can be used to track upcoming events, deadlines, and appointments. Calendar schedulers can also be used to set reminders and recurring events, and to share calendars with other people.

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 system is intended for individual users and small to medium-sized organizations.
- It will allow users to create, edit, and manage calendars.
- Users can schedule, view, and edit events with reminders and notifications.
- Integration with external calendars and email services may be considered in future versions.

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

IEEE ->Institute of Electrical and Electronics Engineers

1.5 References

➤ Books

The Book of Google Calendar:

A book that provides a comprehensive overview of the Google Calendar application, including tips and tricks on how to use it effectively.

Power of Google Calendar:

A book that provides an in-depth guide to the Google Calendar application, including advanced features and integrations.

The Microsoft Outlook Calendar Book:

A book that provides a comprehensive overview of the Microsoft Outlook Calendar application, including tips and tricks on how to use it effectively.

The Calendly Playbook:

A book that provides a guide to using the Calendly calendar scheduler application to automate your scheduling process.

➤ Websites

G2 Crowd: A website that provides user reviews and ratings of business software, including calendar scheduler applications.

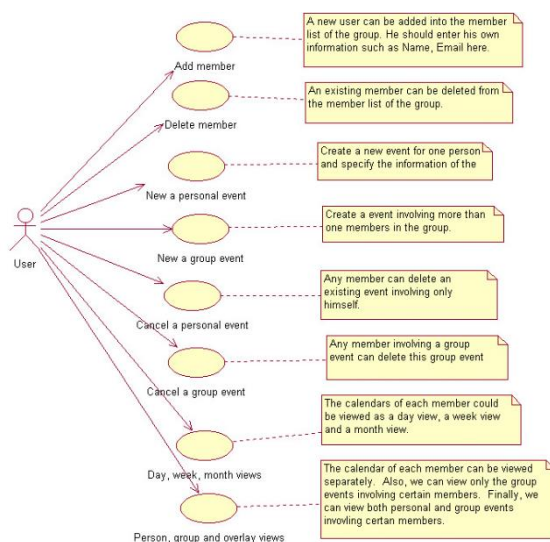
Capterra: A website that provides user reviews and ratings of business software, including calendar scheduler applications.

Software Advice: A website that provides comparisons of different business software applications, including calendar scheduler applications.

2. Overall Descriptions

2.1 Product Perspective

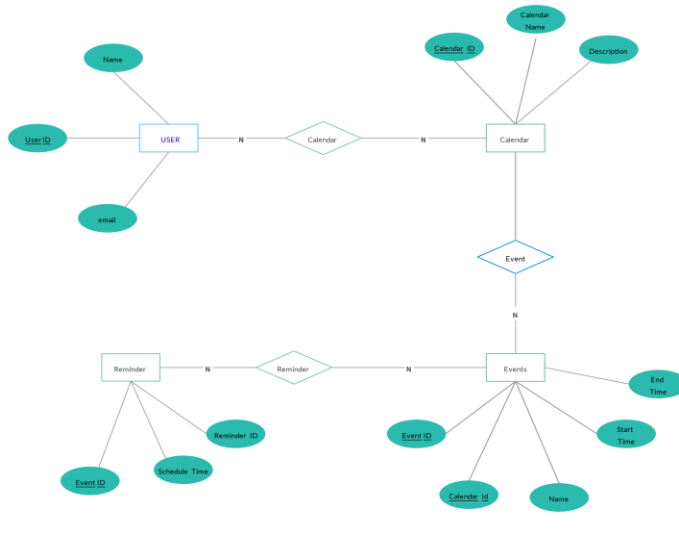
Use Case Diagram of Calendar scheduler



A use case diagram for a calendar scheduler system is a graphical representation of the system's functionality from the perspective of the users. It shows the different types of users (actors) that can interact with the system, and the different actions (use cases) that they can perform.

2.2 Product Function

Entity Relationship Diagram of Library Management System



The ER diagram shows that the User and Calendar entities are connected by a one-to-many relationship. This means that a User can have many Calendars, but a Calendar can only belong to one User. The Calendar and Event entities are also connected by a one-to-many relationship. This means that a Calendar can have many Events, but an Event can only belong to one Calendar.

2.3 User Classes and Characteristics

User classes:

Individuals: Individuals use calendar schedulers to manage their personal calendars, schedule appointments, and keep track of events.

Businesses: Businesses use calendar schedulers to manage their team's schedules, schedule meetings, and book appointments with clients.

Organizations: Organizations use calendar schedulers to manage their events, such as conferences, workshops, and webinars.

User characteristics:

Level of experience: Calendar schedulers can be used by people of all experience levels, from beginners to experts. Some users may only need to use basic features, such as creating events and viewing their calendars. Other users may need to use more advanced features, such as scheduling recurring events,

sharing calendars with others, and integrating with other software applications.

Industry: Calendar schedulers are used in a variety of industries, including healthcare, education, technology, and sales.

Role: Calendar schedulers can be used by people in a variety of roles, such as executives, managers, administrative assistants, and customer representatives.

Additional characteristics:

Device usage: Some users may prefer to use calendar schedulers on their desktop computers, while others may prefer to use them on their mobile devices.

Integration needs: Some users may need to integrate their calendar scheduler with other software applications, such as CRM systems and project management tools.

Security requirements: Some users may need to use a calendar scheduler that offers strong security features, such as encryption and two-factor authentication.

When designing a calendar scheduler, it is important to consider the needs of the different user classes and their characteristics. This will help to ensure that the scheduler is easy to use and meets the needs of a wide range of users.

2.4 Operating Environment

The product will be operating in windows environment. The Calendar Scheduler is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

2.5 Assumptions and Dependencies

The assumptions are:-

- The coding should be error free
 - The system should be user-friendly so that it is easy to use for the users
 - The information of all users, events and scheduling must be stored in a database that is accessible by the website
 - The system should have more storage capacity and provide fast access to the database
 - The system should provide search facility and support quick transactions
 - The Calendar Scheduler is running 24 hours a day
 - Users may access from any computer that has Internet browsing capabilities and Internet connection.
- Users must have their correct usernames and passwords to enter into their online accounts and do actions

The dependencies are:-

- The specific hardware and software due to which the product will be run
- On the basis of listing requirements and specification the project will be developed and run
- The end users (admin) should have proper understanding of the product
- The system should have the general report stored
- The information of all the users must be stored in a database that is accessible by the Calendar Scheduler.
- Any update regarding the event from the calendar is to be recorded to the database and the data entered should be correct

2.6 Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows 98, Windows XP

Language: Java Runtime Environment, Net beans 7.0.1 (front end)

Database: MS SQL Server (back end)

Hardware Configuration:- Processor:

Pentium(R)Dual-core CPU Hard Disk:

40GB

RAM: 256 MB or more

2.7 Data Requirement

Creating a calendar scheduler application involves handling various types of data to ensure the application functions smoothly and effectively. Below, I'll outline the essential data requirements for a calendar scheduler application:

User Profiles: Store information about each user, such as name, email, username, and password.

User Preferences: Allow users to customize settings like time zone, notification preferences, and default calendar views.

User Roles: Assign roles such as regular user, administrator, or manager, depending on the application's functionality.

Events: Store event details, including title, description, location, start date and time, end date and time, and recurrence rules (if applicable).

Calendars: Users should be able to create multiple calendars to organize events, and you need to store calendar names, colors, and ownership information.

Invitations and Attendees: Manage invitations for events and store information about attendees, including their responses (accepted, declined, tentative).

Reminders and Notifications: Store preferences for event reminders and notifications, including methods (e.g., email, push notification).

Scheduling Data:

Availability: Store users' availability information, including working hours and blackout periods when they are not available for scheduling.

Busy/Free Status: Keep track of users' availability status to prevent scheduling conflicts.

Resource Booking (optional): If your application supports booking resources like meeting rooms or equipment, you'll need data for these resources.

Time Zones and Localization:

Time Zones: Store time zone information for users and events to ensure accurate scheduling and display of time.

3. External Interface Requirement

3.1 GUI

Calendar View:

- Display a monthly or weekly calendar view to help users visualize their schedule.
- Use color-coding or different visual cues to distinguish between different events or categories.
- Include navigation controls (e.g., arrows or dropdowns) to switch between months or weeks.

Event Creation and Editing:

- Provide a user-friendly interface for adding and editing events.
- Use a pop-up dialog or a separate event form with fields for event title, date, time, location, and any additional details.
- Include options for setting event recurrence (e.g., daily, weekly, monthly) and reminders.
- Allow users to assign colors or tags to events for easy identification.

Drag-and-Drop Functionality:

- Implement drag-and-drop functionality to enable users to easily move events to different dates or times.
- Highlight the target date or time slot when dragging an event, making it clear where the event will be placed when dropped.

User-Friendly Navigation:

- Include navigation elements such as date picker controls, today's date shortcut, and a search button
- Provide a side panel or a toolbar with quick access to views like day, week, month, or agenda.
- Add buttons for creating new events, deleting events, and toggling between different calendars if the application supports multiple calendars.template.

4. System Features

Event Scheduling and Management:

- ☐ Allow users to easily schedule and manage events, appointments, and tasks. Users should be able to specify event details, including title, date, time, duration, location, and description.

User Collaboration and Sharing:

- ☐ Support collaboration by allowing users to share their calendars with others, whether it's for work, family, or group projects

Cross-Platform Accessibility and Integration:

- ☐ Ensure the calendar scheduler is accessible across multiple platforms, including web, mobile (iOS and Android), and desktop applications.

5. Other Non-functional Requirements

5.1 Performance Requirement

- o **Response Time:**

Define acceptable response times for common actions, such as opening the application, adding or editing events, and navigating between dates or views.

Aim for near-instantaneous responses to user interactions to minimize waiting times.

- o **Scalability:**

Design the system to handle a scalable number of users and events. Specify the maximum number of events and users the system should support simultaneously.

Implement load balancing and resource scaling mechanisms to ensure consistent performance as user and event data grow.

- o **Concurrency and Concurrency Handling:**

Specify how the system should handle concurrent access to events and calendars by multiple users. Ensure that conflicts are resolved gracefully.

Implement locking mechanisms or optimistic concurrency control to prevent data corruption in cases of simultaneous edits or updates.

5.2 Safety Requirement

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

5.3 Security Requirement

- System will use secured database
- Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
- System will have different types of users and every user has access constraints
- Proper user authentication should be provided
- No one should be able to hack users' password
- Users must be provided with regular reminders ,notifications.

5.4 Requirement attributes

The calendar scheduler system should have robust requirements in terms of responsiveness, scalability, data retrieval speed, fault tolerance, and mobile accessibility. It must offer near-instantaneous response times for common user actions, such as event creation and navigation, while supporting a scalable number of users and events. The system should efficiently retrieve event data and implement fault tolerance mechanisms to ensure continuous availability. Furthermore, it should prioritize mobile accessibility to provide a responsive experience on various devices and screen sizes.

5.5 Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols.

5.6 User Requirement

User requirements for a calendar scheduler include a user-friendly interface for easily scheduling and managing events, support for various views (daily, weekly, monthly), customizable event details, options for setting reminders and recurring events, cross-platform accessibility (web, mobile, desktop), synchronization with popular calendar services, sharing and collaboration features, the ability to categorize or color-code events, and efficient performance with quick response times for common actions.

The website provides certain facilities to the users in the form of:-

- Backup and Recovery
- Forgot Password
- Data migration i.e. whenever user registers for the first time then the data is stored in the server
- Data replication i.e. if the data is lost in one branch, it is still stored with the server
- Auto Recovery i.e. frequently auto saving the information
- Maintaining files i.e. File Organization
- The server must be maintained regularly and it has to be updated from time to time

6. Other Requirements

6.1 Data and Category Requirement

The calendar scheduler system must efficiently manage and store a wide range of data, including event details (titles, dates, times, durations, locations, descriptions), user profiles, and calendar categories. Events should support various attributes such as recurrence patterns, reminders, and participant lists. The system should also allow users to categorize events by assigning labels or colors for easy differentiation. Furthermore, it must support the creation and management of multiple calendars to organize events based on context or purpose. Data should be securely stored, with redundancy and backup mechanisms in place to ensure data integrity and availability.

6.2 Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance, Perspective, Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

6.3 Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

- Administrator: A login id representing a user with user administration privileges to the software
- User: A general login id assigned to most users
- Client: Intended users for the software
- SQL: Structured Query Language; used to retrieve information from a database
- SQL Server: A server used to store data in an organized format
- Layer: Represents a section of the project
- User Interface Layer: The section of the assignment referring to what the user interacts with directly
- Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
- Data Storage Layer: The section of the assignment referring to where all data is recorded
- Use Case: A broad level diagram of the project showing a basic overview
- Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system's cases, their attributes, and the relationships between the classes
- Interface: Something used to communicate across different mediums
- Unique Key: Used to differentiate entries in a database

6.4 Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes' structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes

which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here 'Member', 'Group' and 'EventList' are the most important classes which are related to other classes.

