**Calendar Scheduler- SOFTWARE REQUIREMENT SPECIFICATIONS**

**1.Introduction**

1.1 Purpose

1.2 Scope

**2.Functional Requirements**

2.1 User Registration and Authentication

2.2 Profile management

2.3 Portfolio creation and Management

2.4 Viewing Portfolios

2.5 Contact Information

2.6 Search and Filter

2.7 Commenting and Feedback

**3.Non-Functional Requirements**

3.1 Performance

3.2 Security

3.3 Scalability

3.4 Usability

3.5 Backup and Recovery

3.6 Legal and compliance

**4.Constraints**

4.1 The website must be developed using a specific technology stack (e.g., HTML/CSS, JavaScript, MySQL,).

4.2 It should be compatible with modern web browsers.

4.3.It should be platform Independent.

**5.Assumptions and Dependencies**

5.1 Assumptions

5.2 Dependencies

**6. Appendix**

6.1 Glossary

6.2 References

1. **INTRODUCTION**

1.1. Purpose

The purpose of the document is to define the software requirements for the development of a calendar scheduler application using Java that allows users to efficiently manage and schedule events, promoting better time management and organization.

1.2. Scope

The scope of a calendar scheduler is to provide users with a digital tool for managing events, appointments, and tasks efficiently. It includes features such as event creation, editing, and deletion, user authentication, notification reminders, and a user-friendly interface for visualizing and organizing scheduled activities

1. **FUNCTIONAL REQUIREMENTS**

2.1. User Registration and Authentication:

Users should be able to create accounts, log in securely, and maintain personalized calendars tied to their accounts.

2.2Event Creation and Editing:

Users must be able to create, modify, and delete events, specifying details like title, date, time, location, and descriptions.

2.3. Viewing and Navigating Calendars:

Users should easily switch between different calendar views (day, week, month) and navigate to specific dates to visualize their events.

2.4. User-Friendly Interface:

The application must offer an intuitive and responsive user interface with features like drag-and-drop event creation, resizing, and color-coding for categorization.

2.5. Notifications and Reminders:

Users should receive timely notifications and reminders for upcoming events through email, in-app notifications, or SMS.

2.6. Sharing and Collaboration:

Users may share calendars with others, enabling collaborative scheduling, and define permissions for viewing and editing events.

2.7. Search and Filtering:

The ability to search for specific events and filter them based on criteria such as date, location, or event type is essential for efficient event management.

1. **NON-FUNCTIONAL REQUIREMENTS**

3.1. Performance

The website should load quickly and efficiently. It should be responsive and work well on various devices and screen sizes.

3.2 Security

User data, including passwords, must be securely stored and transmitted. Protect against common web vulnerabilities (e.g., SQL injection, XSS).

Implement regular security audits and updates.

3.3 Scalability

The system should be able to handle a growing number of users and portfolios.

3.4 Usability

The website should have an intuitive and user-friendly interface.

Accessibility features should be implemented to ensure inclusivity.

3.5 Backup and Recovery

Regular data backups should be performed. A disaster recovery plan should be in place.

3.6 Legal and Compliance

Ensure compliance with relevant data protection regulations (e.g., GDPR).

Users should agree to terms of service and privacy policy during registration.

1. **CONSTRAINTS**

The website must be developed using a specific technology stack (e.g., HTML/CSS, JavaScript, PHP, MySQL). It should be compatible with modern web browsers.

1. **ASSUMPTIONS AND DEPENDINCIES**

5.1. Assumptions:

users will actively engage with the calendar scheduler, creating, editing, and managing events regularly. Users are assumed to have reliable internet access or network connectivity to use the application and sync their calendar data.

5.2. Dependencies

Application relies on Java technologies for the server-side logic. The application requires a web server (e.g., Apache Tomcat) to deploy and host the server-side code.

1. **APPENDIX**

6.1 Glossary

1.Compliance-conformity in fulfilling official requirements

2. Inclusivity- the fact of including all types of people, things or ideas and treating them all fairly and equally

6.2 References

Scheduler-<https://www.researchgate.net/publication/221518959_What_a_to-do_studies_of_task_management_towards_the_design_of_a_personal_task_list_manager?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19>

Event Scheduler -https://doi.org/10.22214/ijraset.2022.42289