SOFTWARE REQUIREMENTS SPECIFICATION

for

Event Management System

Prepared By :

S.kavinadhithan

P.Ravishankar

S.Keerthana subisri

1.Introduction:

1.1 Purpose

The purpose of this document is to provide a comprehensive overview of the requirements for the Event Management System.

1.2 Scope

The Event Management System is designed to facilitate the planning, organization, and management of events. It includes features for event creation, attendee management, schedule planning, and reporting.

2.Overall description:

2.1 Product Perspective

The EMS will be a standalone system interacting with databases to store event-related information. It will integrate with external systems for features such as payment processing and email notifications.

2.2 Product Features

Event Creation and Management

Attendee Registration and Management

Schedule Planning

Venue Management

Reporting and Analytics

Payment Processing

2.3 User Classes and Characteristics

Administrator: Responsible for system configuration, user management, and overall system supervision.

Event Organizer: Manages the planning and details of specific events.

Attendee: Registers for events, manages personal information.

2.4 Operating Environment

The system should be compatible with major web browsers (Chrome, Firefox, Safari) and support common operating systems (Windows, macOS, Linux).

3. System Features:

3.1 Event Creation and Management

Create a new event with details (name, date, location, etc.).

Edit and update event information.

Cancel or reschedule events.

3.2 Attendee Registration and Management

Allow attendees to register for events.

Manage attendee information and registrations.

3.3 Schedule Planning

Create and manage event schedules.

Notify attendees of schedule changes.

3.4 Venue Management

Maintain a database of available venues.

Assign venues to events.

3.5 Reporting and Analytics

Generate reports on attendance, revenue, etc.

Provide analytics for event success and improvement.

3.6 Payment Processing

Integrate with a payment gateway for ticket sales.

Provide secure payment options.

4. External Interface Requirements

4.1 User Interfaces:

Describe the interfaces for administrators, event organizers, and attendees.

4.2 Hardware Interfaces:

Specify any hardware requirements, if applicable.

4.3 Software Interfaces:

List and describe any third-party software or APIs the system will interact with.

4.4 Communication Interfaces:

Describe communication protocols and methods used by the system.

A Software Requirements Specification (SRS) document for an Event Management System outlines the functional and non-functional requirements of the software. Below is a template you can use as a starting point for creating an SRS for an Event Management System. Keep in mind that this is a general template, and you may need to customize it based on the specific needs of your project.

1. Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive overview of the requirements for the Event Management System.

1.2 Scope

The Event Management System is designed to facilitate the planning, organization, and management of events. It includes features for event creation, attendee management, schedule planning, and reporting.

1.3 Definitions, Acronyms, and Abbreviations

EMS: Event Management System

CRUD: Create, Read, Update, Delete

2. Overall Description

2.1 Product Perspective

The EMS will be a standalone system interacting with databases to store event-related information. It will integrate with external systems for features such as payment processing and email notifications.

2.2 Product Features

Event Creation and Management

Attendee Registration and Management

Schedule Planning

Venue Management

Reporting and Analytics

Payment Processing

2.3 User Classes and Characteristics

Administrator: Responsible for system configuration, user management, and overall system supervision.

Event Organizer: Manages the planning and details of specific events.

Attendee: Registers for events, manages personal information.

2.4 Operating Environment

The system should be compatible with major web browsers (Chrome, Firefox, Safari) and support common operating systems (Windows, macOS, Linux).

2.5 Design and Implementation Constraints

The system should be developed using [programming language] and [framework].

Database: [Specify database system].

Security: The system should implement encryption for sensitive data and have user authentication and authorization mechanisms.

3. System Features

3.1 Event Creation and Management

Sub-feature 1: Create a new event with details (name, date, location, etc.).

Sub-feature 2: Edit and update event information.

Sub-feature 3: Cancel or reschedule events.

3.2 Attendee Registration and Management

Sub-feature 1: Allow attendees to register for events.

Sub-feature 2: Manage attendee information and registrations.

3.3 Schedule Planning

Sub-feature 1: Create and manage event schedules.

Sub-feature 2: Notify attendees of schedule changes.

3.4 Venue Management

Sub-feature 1: Maintain a database of available venues.

Sub-feature 2: Assign venues to events.

3.5 Reporting and Analytics

Sub-feature 1: Generate reports on attendance, revenue, etc.

Sub-feature 2: Provide analytics for event success and improvement.

3.6 Payment Processing

Sub-feature 1: Integrate with a payment gateway for ticket sales.

Sub-feature 2: Provide secure payment options.

4. External Interface Requirements

4.1 User Interfaces

Describe the interfaces for administrators, event organizers, and attendees.

4.2 Hardware Interfaces

Specify any hardware requirements, if applicable.

4.3 Software Interfaces

List and describe any third-party software or APIs the system will interact with.

4.4 Communication Interfaces

Describe communication protocols and methods used by the system.

5. Non-functional Requirements

5.1 Security Requirements

User data should be encrypted during transmission.

Access to sensitive information should be restricted based on user roles.

5.2 Reliability and Availability

The system should have an uptime of at least [Z]%.

Data backups should be performed regularly.

5.3 Maintainability

The system should be easy to maintain and update.

Provide documentation for system administration and troubleshooting.