**Software Requirements**

**Specification**

**for**

**EVENT MANAGEMENT SYSTEM**

**Version 1.0 approved**

**Prepared by**

INIKA P S-23CSR087

JOSHNA K-23CSR094

KATHIR M-23CSR104

**KONGU ENGINEERING COLLEGE**

**20.02.2025**

**Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.**

**Software Requirements Specification for <Project> Page ii**

# Table of Contents

[Table of Contents ii](#_Toc8033)

[Revision History ii](#_Toc8034)

[1. Introduction 1](#_Toc8035)

[1.1 Purpose 1](#_Toc8036)

[1.2 Document Conventions 1](#_Toc8037)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc8038)

[1.4 Product Scope 1](#_Toc8039)

[1.5 References 1](#_Toc8040)

[2. Overall Description 2](#_Toc8041)

[2.1 Product Perspective 2](#_Toc8042)

[2.2 Product Functions 2](#_Toc8043)

[2.3 User Classes and Characteristics 2](#_Toc8044)

[2.4 Operating Environment 2](#_Toc8045)

[2.5 Design and Implementation Constraints 2](#_Toc8046)

[2.6 User Documentation 2](#_Toc8047)

[2.7 Assumptions and Dependencies 3](#_Toc8048)

[3. External Interface Requirements 3](#_Toc8049)

[3.1 User Interfaces 3](#_Toc8050)

[3.2 Hardware Interfaces 3](#_Toc8051)

[3.3 Software Interfaces 3](#_Toc8052)

[3.4 Communications Interfaces 3](#_Toc8053)

[4. System Features 4](#_Toc8054)

[4.1 System Feature 1 4](#_Toc8055)

[4.2 System Feature 2 (and so on) 4](#_Toc8056)

[5. Other Nonfunctional Requirements 4](#_Toc8057)

[5.1 Performance Requirements 4](#_Toc8058)

[5.2 Safety Requirements 5](#_Toc8059)

[5.3 Security Requirements 5](#_Toc8060)

[5.4 Software Quality Attributes 5](#_Toc8061)

[5.5 Business Rules 5](#_Toc8062)

[6. Other Requirements 5](#_Toc8063)

[Appendix A: Glossary 5](#_Toc8064)

[Appendix B: Analysis Models 5](#_Toc8065)

[Appendix C: To Be Determined List 6](#_Toc8066)

# 1. Introduction

## 1.1 Purpose

## The purpose of this document is to define the functional, non-functional, and technical requirements for an Event Management Website. The system will allow users to plan, organize, and manage events efficiently.

## 1.2 Document Conventions

The Event Management Website will enable users to:

* Create and manage events
* Register and book tickets for events
* Provide event-related information
* Handle payments and confirmations
* Send notifications and reminders

## 1.3 Intended Audience and Reading Suggestions

This document is intended for developers, testers, project managers, and stakeholders involved in the development of the Event Management Website.

## 1.4 Product Scope

The **Event Management Website** is a web-based platform designed to simplify event planning and management. The system will provide features for event organizers to create, edit, and manage events while allowing attendees to register and book tickets. The website will support secure payments, real-time notifications, and user engagement through feedback and reviews. The key benefits include:

* **Efficient Event Planning:** Streamlined event creation and management process.
* **User-friendly Registration:** Simple attendee registration and ticket booking.
* **Secure Transactions:** Safe and seamless payment processing.
* **Automated Notifications:** Email and SMS reminders for attendees.
* **Scalability:** Capable of handling large-scale events with high user traffic.

## 1.5 References

## Capterra Blog - Event Management Software

## TechRadar - Event Management Software

# 2. Overall Description

## 2.1 Product Perspective

The Event Management Website will be a standalone web application that provides seamless event creation and management services.

## 2.2 Product Functions

* User Registration and Authentication
* Event Creation and Management
* Ticket Booking System
* Payment Processing
* Email and SMS Notifications
* User Dashboard for Event Tracking
* Reviews and Feedback System

## 2.3 User Classes and Characteristics

* Event Organizers: Require tools to manage events.
* Attendees: Need an easy interface for event discovery and ticket booking.
* Admins: Require control over system users and events.

## 2.4 Operating Environment

* Web-based system accessible from modern browsers.
* Cloud-hosted backend with secure APIs.

## 2.5 Design and Implementation Constraints

* The system must be web-based and responsive.
* Payments must be processed securely.
* Event details should be available in real-time.

## 2.6 User Documentation

* Online help and tutorials.
* FAQ section.

## 2.7 Assumptions and Dependencies

* Users have internet access.
* Third-party services for payments and notifications function correctly.

# 3. External Interface Requirements

**3.1 User Registration and Authentication**

* Users should be able to sign up and log in using email/password.
* OAuth integration for social login (Google, Facebook, etc.).

**3.2 Event Management**

* Event organizers can create, edit, and delete events.
* Event details should include title, date, location, description, and images.

**3.3 Ticket Booking System**

* Users can book tickets online.
* Generate and send e-tickets via email.

**3.4 Payment Processing**

* Secure payment gateway integration.
* Refund and cancellation policies should be handled.

**3.5 Notifications**

* Email and SMS alerts for registration, payment confirmation, and reminders.

**3.6 Reviews and Feedback**

* Attendees can submit reviews and ratings for events.

# 4. Non-Functional Requirements

**4.1 Performance Requirements**

* The system should handle at least 1000 concurrent users.
* Page load time should be under 3 seconds.

**4.2 Security Requirements**

* User passwords must be encrypted.
* Secure authentication mechanisms (e.g., OAuth, JWT).
* PCI-DSS compliance for payment processing.

**4.3 Usability Requirements**

* The UI should be intuitive and responsive.
* The website should be accessible to users with disabilities.

**4.4 Availability and Reliability**

* 99.9% uptime should be maintained.
* Automated backups must be performed daily.

**5. System Architecture**

* The system will follow a **three-tier architecture**:
  1. **Frontend:** React/Angular (User Interface)
  2. **Backend:** Node.js/Python (Business Logic & APIs)
  3. **Database:** MySQL/PostgreSQL (Data Storage)
* Cloud hosting using AWS/Azure for scalability.

**6. External Interfaces**

**6.1 User Interface**

* Web-based application compatible with modern browsers.

**6.2 Hardware Interfaces**

* Runs on any device with an internet connection.

**6.3 Software Interfaces**

* Payment Gateway (Stripe/PayPal)
* Email Service (SendGrid/Mailgun)
* SMS Service (Twilio)

**7. Assumptions and Dependencies**

* Users have internet access.
* Third-party services (payment gateway, email/SMS providers) function correctly.

**8. Appendices**

* Sample UI wireframes
* Flow diagrams for event booking and payments