SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Event Management System**

**Prepared by:-**

*Algotutor Academy*

# Introduction

## Purpose

The primary goal of this document is to delineate the requirements for the development of an Event Management System. The document provides a detailed account of both functional and non-functional requirements, aiming to create a comprehensive platform for efficient event planning, organization, and monitoring.

## 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

## Scope of Development Project

The Event Management System is designed to streamline the processes involved in planning and executing events. It caters to the needs of event organizers, participants, and administrators. The project encompasses features such as event creation, participant registration, real-time monitoring, and post-event analysis.

The project can be easily implemented under various situations. We can add new features as and when we require, making reusability possible as there is flexibility in all the modules.

The language used for developing the project is Java as it is quite advantageous than other languages in terms of performance, tools available, cross platform compatibility, libraries, cost (freely available), and development process.

## 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

## References

* + - Books

 Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson

Software Requirements (Microsoft) Second EditionBy Karl E. Wiegers

Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman

* + - Websites

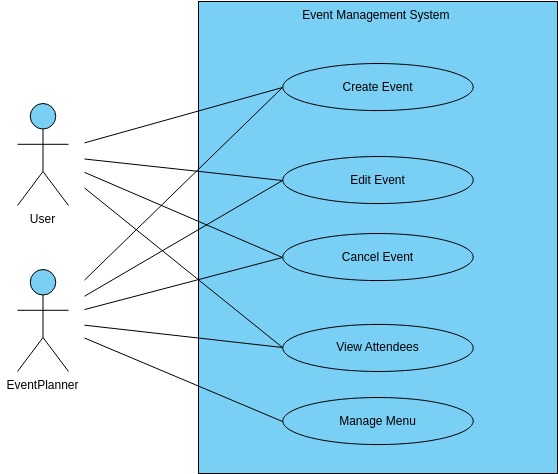
[**http://www.slideshare.net/**](http://www.slideshare.net/)

**https://www.eventbrite.com/**

# Overall Descriptions

## Product Perspective

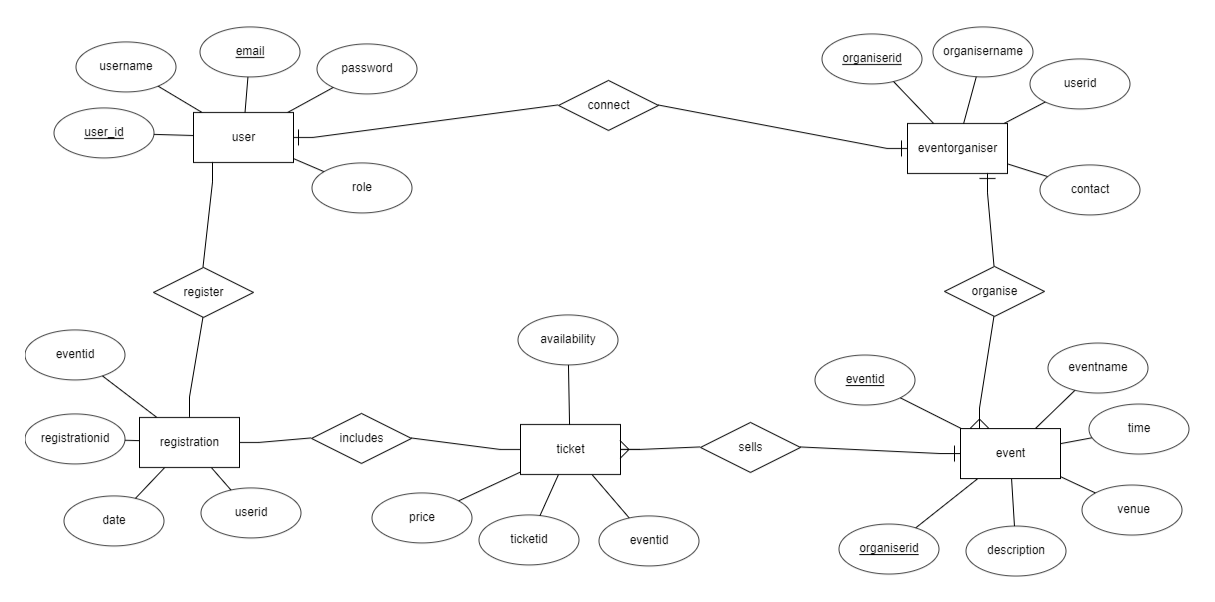
Use Case Diagram of Event management System



The use case diagram illustrates the interactions between system actors (Organizers, Participants, and Administrators) and their respective functionalities within the Event Management System. Use cases include event creation, participant registration, ticket generation, real-time monitoring, and post-event analysis.

## Product Function

Entity Relationship Diagram of Event Management System. The Entity-Relationship (ER) diagram illustrates the fundamental structure of the Event Management System. Entities such as Events, Participants, Organizers, Venues, and Tickets are interconnected to capture the relationships and dependencies within the system.



Entity Relationship Diagram of Event Management System. The Entity-Relationship (ER) diagram illustrates the fundamental structure of the Event Management System. Entities such as Events, Participants, Organizers, Venues, and Tickets are interconnected to capture the relationships and dependencies within the system.

## User Classes and Characteristics

The Event Management System caters to three primary user classes: Organizers, Participants, and Administrators.

**Role: Organizers:**

* Characteristics: Experienced individuals responsible for planning, creating, and managing events.
* Features:
* Create and publish events.
* Customize event details, such as date, time, and location.
* Generate tickets and set pricing.
* Monitor real-time participant registration.
* Analyze post-event statistics.

**Role: Participants:**

* Characteristics: End-users attending events with the following features.
* Features:
* Browse and register for events.
* Purchase tickets.
* Receive event details and updates.
* Provide feedback post-event.

**Role: Administrators:**

* Characteristics: System administrators with the authority to oversee and manage the entire Event Management System.
* Features:
* Monitor system health and performance.
* Resolve issues and provide support.
* Manage user accounts and permissions.

## Operating Environment

The product will be operating in windows environment. The Event Management System 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.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

## 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
    - Users have consistent internet access for seamless event participation.
* Participants are familiar with standard web browsers.
* The system is compatible with laptops, tablets, and smartphones.
* Users maintain the confidentiality of login credentials.
* Event data accuracy depends on truthful participant responses.
* 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
    - User access depends on valid login credentials.
    - Relies on contemporary web technologies.
    - Dependency on established security measures.
    - Depends on routine maintenance for optimal performance.

## 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

## Data Requirement

In the Event Management System, inputs consist of user-generated queries such as event creation, participant registration, and ticket purchase. The system processes these queries to provide outputs, including event details, participant lists, and post-event analysis.

# External Interface Requirement

## GUI

The graphical user interface (GUI) of the Event Management System is designed to be intuitive and user-friendly.

Login Screen:

Secure login interface with fields for username and password.

Dashboard:

Upon login, users encounter an intuitive dashboard displaying options like creating events, accessing event details, and viewing post-event statistics.

Event Creation Interface:

Organizers experience a user-friendly interface for creating events, specifying details such as date, time, venue, and ticket pricing.

Ticket Generation Screen:

A dedicated screen allows organizers to generate tickets, set pricing, and manage ticket inventory.

Event Selection Screen:

Participants encounter a screen for browsing and selecting events, with options to register and purchase tickets.

Real-time Monitoring:

Organizers access a real-time monitoring interface during events, providing live participant registration updates.

Post-Event Analysis Interface:

Organizers and administrators have access to an interface for analyzing post-event statistics, including participant feedback and attendance.

User Account Interface:

All users have access to a user account interface to manage personal details, preferences, and view event history.

Navigation and Responsiveness:

The entire GUI ensures smooth navigation, responsiveness across devices, and adherence to accessibility standards for an inclusive user experience.

System Features

# System Features

* Comprehensive Event Management:

The system empowers organizers to create, customize, and manage events, fostering a dynamic and interactive platform.

* Seamless User Experience:

Participants experience a seamless and user-friendly interface for accessing events, registering, purchasing tickets, and providing feedback.

* Real-time Monitoring:

Organizers can monitor events in real-time, gaining immediate insights into participant engagement and performance.

* Post-Event Analysis:

Organizers and administrators can analyze post-event statistics, including participant feedback and attendance, facilitating future event planning.

# Other Non-functional Requirements

## Performance Requirement

* The system is designed to handle simultaneous events with a large number of participants, ensuring smooth and responsive interactions.
* Response times for user interactions, such as participant registration and ticket purchase, should be within 2 seconds under normal system load.
* The system should support at least 100 concurrent events without degradation in performance.

## Safety Requirement

* Regular security audits and updates will be conducted to identify and address potential vulnerabilities.
* In the event of unexpected system disruptions, automated backups will be in place to prevent data loss and facilitate prompt recovery.

## Security Requirement

* + - System will use secured database
    - Role-based access controls ensure that users have appropriate privileges, maintaining data integrity.
    - Encrypted credentials are required for secure access, preventing unauthorized entry.
    - Robust logging mechanisms are in place to record system activities, enabling effective monitoring and auditing.

## Requirement attributes

* + - There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
    - The project should be open source
    - The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
    - The user be able to easily download and install the system

## Business Rules

The Event Management System adheres to essential business rules, ensuring a secure and efficient event planning and execution environment. These rules dictate user authentication protocols, secure data storage, and timely response rates for user interactions. The system strictly enforces policies regarding event creation, participant access, and accurate representation of event data, fostering a reliable and trustworthy platform for both organizers and participants.

## User Requirement

User requirements for the Event Management System emphasize a seamless experience. Users expect the platform to be intuitive, responsive, and accessible 24/7. The system should support simultaneous events with real-time monitoring, ensuring an engaging and efficient event management environment for both organizers and participants.

The admin provides certain facilities to the users in the form of:

* Backup and Recovery
* Forgot Password
* Collaborative Events
* Regular server maintenance and updates

# Other Requirements

## Data and Category Requirement

The project requires a robust data management system, encompassing event details, participant information, ticket data, feedback, and user preferences. Categories include organizer-created events, participant registration, and post-event analysis. This ensures efficient storage, retrieval, and analysis, facilitating seamless interactions and personalized feedback within the Event Management System.

## 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;

## 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

## 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 ‘Event’, ‘Participant’ and ‘Organizer’ are the most important classes which are related to other classes.

