



**The Islamia University of Bahawalpur**

**Department of information technology**

**SOFTWARE REQUIREMENTS SPECIFICATION  
(SRS DOCUMENT) for event management**

<b>Name:</b>	<b>Muhammad Adeel Jameel</b>
<b>Roll no:</b>	<b>F22BINFT1M01145</b>
<b>Supervisor:</b>	<b>Mam Sara Fareed</b>
<b>Section:</b>	<b>5M</b>
<b>Semester:</b>	<b>7<sup>th</sup></b>
<b>Department:</b>	<b>Information technology</b>

## Table of Contents

Version 1.0 .....	<b>Error! Bookmark not defined.</b>
Revision History .....	1
Application Evaluation History .....	2
Introduction .....	3
Purpose .....	3
Scope .....	4
Major Features and Functionalities: .....	4
Overall description .....	4
Product perspective .....	4
Operating environment .....	5
Design and implementation constraints .....	5
Requirement identifying technique .....	6
Use case diagram .....	6
Use case description .....	10
User Workflow: .....	11
Functional Requirements .....	18
Non-Functional Requirements .....	21
Usability .....	21
• The design should follow accessibility standards (e.g., WCAG 2.1) to ensure usability for all users, including those with disabilities. ....	21
Performance .....	21

## Revision History

Name	Date	Reason for changes	Version

## Application Evaluation History

<b>Comments (by committee)</b> *include the ones given at scope time both in doc and presentation	<b>Action Taken</b>

**Supervised by:**

**Mam Sara Fareed**

Signature\_\_\_\_\_

# Introduction

Event management has become an essential aspect of organizing and coordinating various activities, from corporate events to personal celebrations. An Event Management System (EMS) is designed to streamline the planning, execution, and monitoring of events, ensuring efficiency, transparency, and scalability.

This Software Requirements Specification (SRS) document provides a detailed overview of the EMS, outlining its purpose, scope, functionalities, and technical requirements.

The document aims to serve as a guide for stakeholders, including Events, Venues, User, Event Reports, User Reports and end-users, ensuring a shared understanding of the system's requirements. By addressing both functional and non-functional requirements, this SRS establishes a foundation for the successful design, development, and deployment of the EMS.

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to define and outline The requirements for the development of the Event Management Website. This document acts as a formal agreement between stakeholders, including business owners, developers, designers, and End-users, to ensure a shared understanding of the project's objectives, features, and functionalities.

The SRS serves the following primary purposes:

### **Defining Requirements:**

Clearly identify and document the functional, non-functional, and technical requirements of the Event Management platform.

- **Guiding Development:**

Provide developers and designers with a structured framework to design and implement the system effectively and efficiently.

- **Ensuring Consistency:**

Establish a single source of truth to avoid misunderstandings or Misinterpretations during the Development process.

- **Facilitating Communication:**

Act as a communication tool among stakeholders to ensure all parties are aligned on the scope, priorities and deliverables of the project.

- **Supporting Quality Assurance:**

Assist testers in understanding the system's requirements and Functionality to develop comprehensive test cases for ensuring quality.

- **Planning for Scalability:**

Identify performance and scalability requirements to accommodate future Growth in user base and

system features.

Ultimately, this SRS is designed to ensure the successful development and deployment of a Event Management Website that meets user expectations, supports business objectives, and delivers a seamless online shopping experience.

## **Scope**

This web application the transaction of event management, user management and also provide Automatic generate bill. This APP consist of login page for admin or owner And provide dashboard for accounts detail, reports, product listing. This web Application provide Online Booking for Events without going to the Venue physically.

### **Major Features and Functionalities:**

#### **1. Venue Management:**

Display an organized and visually appealing catalog of Venues with detailed descriptions, images, and pricing.

#### **2. Event Management:**

Enable users to quickly find Events.

#### **3. User Management:**

Manage Users for both public event and private events and confirmation of users.

#### **4. Responsive Design:**

Ensure compatibility across various devices (desktop, tablet, mobile) for a consistent user experience.

#### **5. Admin Panel:**

Allow administrators to manage and generate reports to optimize business performance.

## **Overall description**

### **Product perspective**

The Event Management Website is a standalone system that integrates seamlessly with external Services to create a comprehensive online Booking and organizing event platform. It is designed to cater to a wide Range of users.

## System Interfaces

- **User Interface:**

The website features a responsive and intuitive interface designed for ease of navigation across devices, including desktops, tablets, and smartphones.

- **Admin Panel Interface:**

Provides administrators with the tools to manage product listings, monitor orders, analyze sales, and control website content.

- **API Integration:**

Supports APIs to enable secure communication between the Event Management platform and external systems such as payment gateways and logistics services.

## Operating environment

The Event Management Website is a web-based platform designed to operate in a highly dynamic and user-focused environment. It will be accessible to users across the globe through various devices and internet connections. Below are the key components and details of the environment in which the software will operate:

Example:

### Client-Side:

- Operating Systems: Windows, macOS, Android, iOS, and Linux.
- Web Browsers: Latest versions of Chrome, Firefox, Safari, Microsoft Edge, and Opera.

### Server-Side:

- Web Server: Apache or Xampp for handling HTTP requests.
- Database: Relational databases like MySQL.
- Backend: PHP frameworks for server-side operations.
- Frontend: HTML5, CSS3, JavaScript.

## Design and implementation constraints

The development of the Event Management Website must adhere to several design and implementation constraints that impact the architecture, development process, and overall functionality. These constraints certify that the system meets performance, compatibility, security, and scalability requirements while staying within budget and time limitations.

Example:

### **Responsive Design**

The platform must provide a seamless user experience across multiple devices, including desktops, tablets, and mobile phones, with varying screen sizes.

Implementation of adaptive UI components using frameworks like Bootstrap is mandatory.

### **Scalability**

The system should be able to handle a high volume of concurrent users.

### **Browser Compatibility**

The website must function correctly on the latest versions of popular web browsers such as Google Chrome, Firefox, Safari, and Microsoft Edge.

Support for older browsers may be limited to ensure efficient development and modern features.

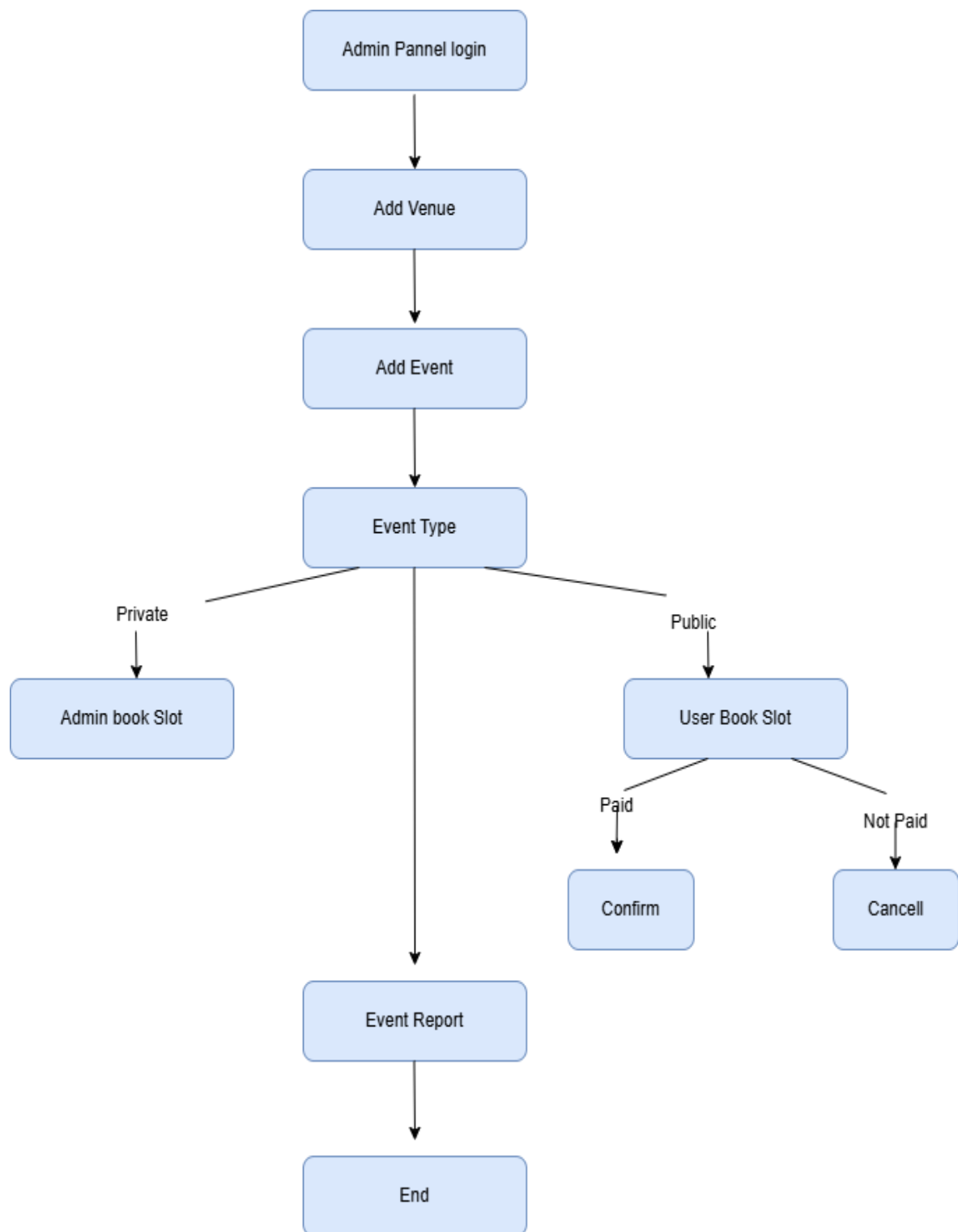
## **Requirement identifying technique**

This section describes the requirements identifying technique(s) which further help to derive functional requirements specification. The selection of the technique(s) will depend on the type of project. For instance,

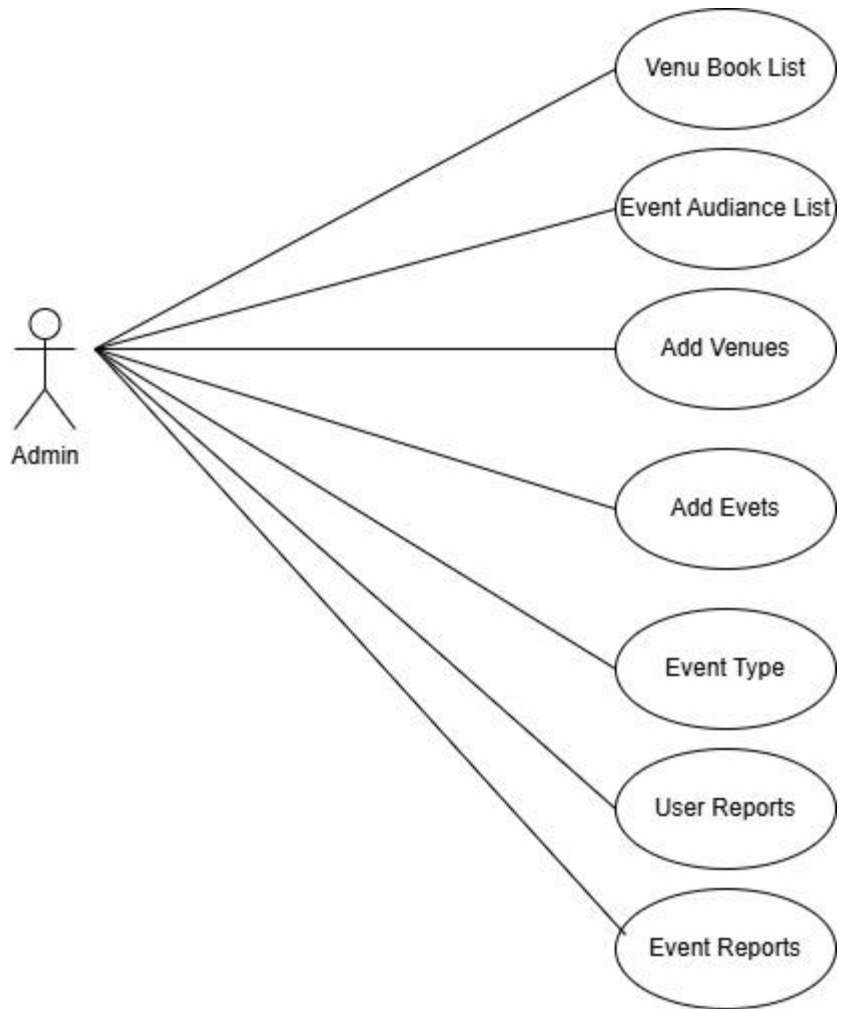
- **Use case** is an effective technique for interactive end-user applications
- **Event- response tables** is for real time system and
- **Story boarding** for graphically intensive applications.

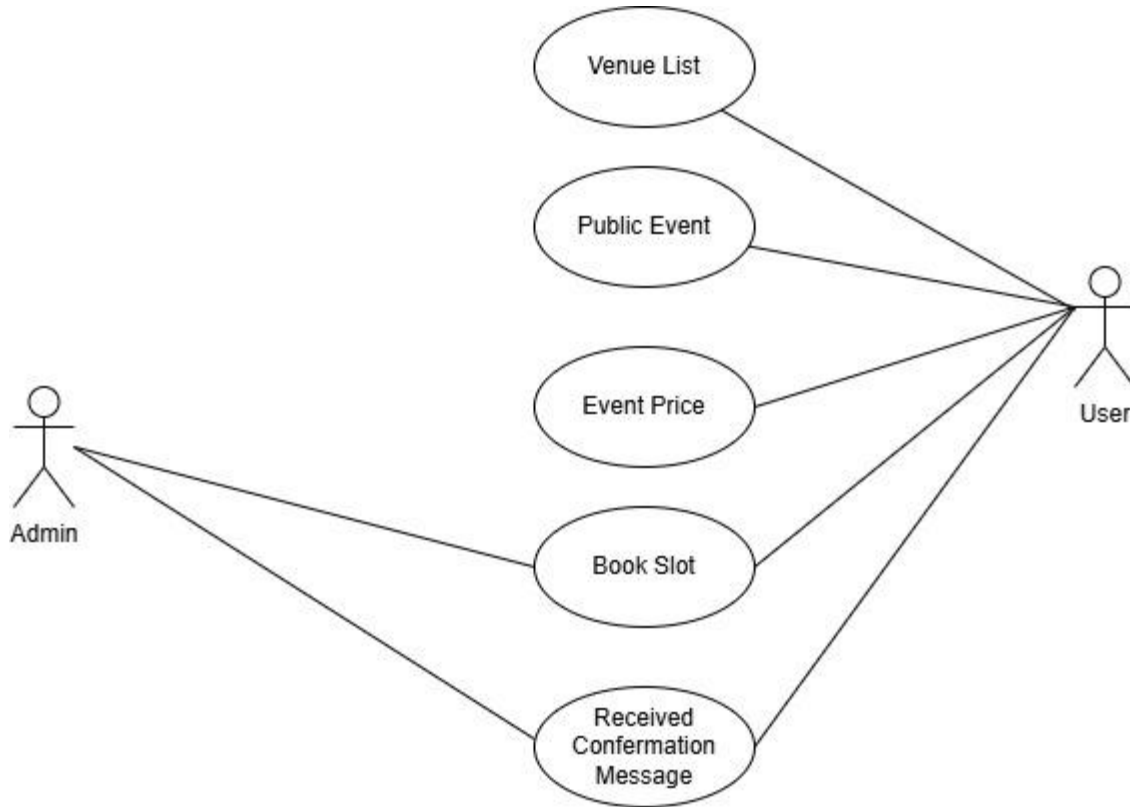
### **Use case diagram**











## Use case description

It represents the process a customer goes through, from registration to order delivery submission. Here's a breakdown of the steps shown:

1. **Admin Panel Login:**  
The admin logs into the system to access the backend panel.
2. **Add Venue:**  
The admin adds venue details to the system.
3. **Add Event:**  
The admin registers an event, providing details such as the event type, date, and time.
4. **Event Type:**  
The admin specifies the type of event (e.g., wedding, conference, birthday, etc.).
5. **Admin Book Slot:**  
The admin reserves a slot for an event if required.
6. **Event Report:**  
After the event, the admin generates or views a report summarizing event details, performance, or outcomes.
7. **End:**  
The admin completes the workflow.

## User Workflow:

1. **User Book Slot:**  
Users access the system to book an event slot.
2. **Confirm or Cancel:**  
Users confirm their booking or cancel the reservation as per their needs.
3. **Event Report:**  
Users may have access to view event reports, depending on the system's permissions.
4. **End:**  
The user completes their interaction with the system.

**table Show the detail use case template**

<b>Use Case ID:</b>	UC-1
<b>Use Case Name:</b>	Create Event
<b>Actors:</b>	<b>Primary Actor:</b> Event Organizer <b>Secondary Actors:</b> System Admin
<b>Description:</b>	This use case allows event organizers to create events by entering event details, setting event categories, and availability.
<b>Trigger:</b>	The Event Organizer initiates the event creation process by clicking the "Create Event" button in the dashboard.
<b>Preconditions:</b>	PRE-1: The Event Organizer must be logged in. PRE-2: The Event must be approved by the system admin. PRE-3: Event must be public for receiving inquiries.
<b>Postconditions:</b>	POST-1: The event is stored in the database with a status of "Active." POST-2: Notifications are sent to followers of the organizer about the new event. POST-3: Event details are displayed in the search results.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The Event Organizer clicks "Create Event" in the dashboard.</li> <li>2. The system displays a form to enter event details (title, description, date, time, location, ticket categories, etc.).</li> <li>3. User fills out the form.</li> <li>4. Organizer selects ticketing options (price, quantity, categories, early-bird discounts).</li> <li>5. The system validates the entered details and confirms the creation of the event.</li> <li>6. The event is marked as "Active" and made live on the platform.</li> </ol>

<b>Alternative Flows:</b>	<p><b>AF-1: Incomplete Event Details</b></p> <ol style="list-style-type: none"> <li>1. If the organizer misses mandatory fields, the system displays an error message and highlights the missing fields.</li> <li>2. Organizer corrects the information and resubmits the form.</li> </ol> <p><b>AF-2: Duplicate Event Check</b></p> <ol style="list-style-type: none"> <li>1. The system checks for similar events created by the same organizer (based on title and date).</li> <li>2a. If a duplicate is found, the system prompts the organizer to modify the details.</li> <li>2b. If no duplicates are found, the system proceeds with event creation</li> </ol>
<b>Exceptions:</b>	<p><b>EX-1: Payment Gateway Configuration Missing</b></p> <ol style="list-style-type: none"> <li>1. The system detects that the payment gateway is not configured.</li> <li>2a. The organizer is prompted to configure payment details.</li> <li>2b. If the organizer cancels, the use case is terminated.</li> </ol> <p><b>EX-2: Event Creation Timeout</b></p> <ol style="list-style-type: none"> <li>1. If the session times out, unsaved changes are lost, and the organizer must restart the process.</li> </ol>
<b>Business Rules</b>	<p>BR-1: Events cannot be scheduled for past dates.</p> <p>BR-2: Event inquiry cannot be full filled without incomplete details.</p>
<b>Assumptions:</b>	<p>A Event must be verified by the Admin.</p> <ol style="list-style-type: none"> <li>1.Admin have the necessary permissions to create events.</li> <li>2.Users are familiar with event inquiry workflows.</li> </ol>

<b>Use Case ID:</b>	UC-002
<b>Use Case Name:</b>	User Login
<b>Actors:</b>	<b>Primary Actor:</b> User <b>Secondary Actor:</b> System
<b>Description:</b>	A registered user logs into the event management platform by providing valid credentials (email and password) or by using a third-party authentication provider.
<b>Trigger</b>	The user initiates the login process by clicking the "Login" button on the platform.
<b>Preconditions</b>	PRE-1: The user must already be registered on the platform. PRE-2: The system must be operational and accessible.
<b>Postconditions</b>	POST-1: The user is authenticated and granted access to their account dashboard. POST-2: A session is created for the user.
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. The user clicks the "Login" button on the platform.</li> <li>2. The system displays the login form, requesting the user's email and password.</li> <li>3. The user enters their credentials and submits the form.</li> <li>4. The system validates the credentials.</li> <li>5. If the credentials are valid, the user is redirected to their account dashboard.</li> </ol>
<b>Alternative Flows</b>	<b>AF-1: Forgot Password</b> <ol style="list-style-type: none"> <li>1. The user clicks the "Forgot Password" link.</li> <li>2. The system redirects the user to the password recovery process.</li> </ol> <b>AF-2: Third-Party Login</b> <ol style="list-style-type: none"> <li>1. The user selects a third-party provider (e.g., Google, Facebook).</li> <li>2. The system redirects the user to the third-party authentication page.</li> <li>3. The user completes authentication with the provider.</li> <li>4. The system receives the authentication token and logs the user into the platform.</li> </ol>
<b>Exceptions</b>	<b>EX-1: Invalid Credentials</b> <ol style="list-style-type: none"> <li>1. The system detects invalid credentials.</li> <li>2. The system displays an error message and prompts the user to re-enter their credentials.</li> </ol> <b>EX-2: Account Locked</b> <ol style="list-style-type: none"> <li>1. The user's account is locked after multiple failed login attempts.</li> <li>2. The system notifies the user and prompts them to reset their password.</li> </ol>
<b>Business Rules</b>	BR-1: Users must enter valid email addresses and passwords to log in. - BR-2: After 5 consecutive failed login attempts, the system locks the account for 15 minutes. - BR-3: Third-party logins must comply with OAuth 2.0 standards.
<b>Assumptions</b>	- Users have access to their registered email address. - Users are familiar with the login process and their credentials. - The system's authentication services are operational.

<b>Use Case ID</b>	UC-003
<b>Use Case Name</b>	Search Event
<b>Actors</b>	<b>Primary Actor:</b> User <b>Secondary Actor:</b> System
<b>Description</b>	A user searches for events on the platform using keywords, filters, or categories to find events matching their preferences.
<b>Trigger</b>	The user initiates the search by entering keywords or selecting filters on the platform.
<b>Preconditions</b>	PRE-1: The system must have events available in the database. PRE-2: The user must have access to the platform.
<b>Postconditions</b>	POST-1: A list of events matching the search criteria is displayed to the user. POST-2: The search results page is updated with the relevant data.
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. The user navigates to the search bar or filter section of the platform.</li> <li>2. The user enters keywords (e.g., "music concert") or selects specific filters (e.g., date, location, event type).</li> <li>3. The system processes the input and queries the database.</li> <li>4. The system displays a list of events matching the criteria, including relevant details like name, date, location, and a brief description.</li> <li>5. The user can click on an event to view its detailed page.</li> </ol>
<b>Alternative Flows</b>	<p><b>AF-1: No Results Found</b></p> <ol style="list-style-type: none"> <li>1. The system detects that no events match the search criteria.</li> <li>2. The system displays a "No events found" message and suggests adjusting the filters or keywords.</li> </ol> <p><b>AF-2: Filtered Search</b></p> <ol style="list-style-type: none"> <li>1. The user applies multiple filters (e.g., "Concerts in New York on January 15th").</li> <li>2. The system updates the results dynamically based on the applied filters.</li> </ol>
<b>Exceptions</b>	<p><b>EX-1: Invalid Search Input</b></p> <ol style="list-style-type: none"> <li>1. The system detects invalid or unsupported input (e.g., special characters only).</li> <li>2. The system displays an error message and prompts the user to revise the search criteria</li> </ol> <p><b>EX-2: Network Error</b></p> <ol style="list-style-type: none"> <li>1. A network issue prevents the system from fetching search results.</li> <li>2. The system displays a "Network error" message and suggests retrying later.</li> </ol>
<b>Business Rules</b>	<p>BR-1: Search results must be sorted by relevance or date by default.</p> <p>BR-2: Users can filter by categories such as date, location, event type, or price.</p> <p>BR-3: Search results must include at least a brief description, image, and link to the event details page.</p>
<b>Assumptions</b>	<ol style="list-style-type: none"> <li>1. Users have a general idea of the type of event they are searching for.</li> <li>2. The platform's search and filtering functionalities are operational.</li> <li>3. Event data is well-organized in the database.</li> </ol>



<b>Use Case ID:</b>	UC-004
<b>Use Case Name:</b>	Register User
<b>Actors:</b>	<b>Primary Actor:</b> User <b>Secondary Actor:</b> System
<b>Description:</b>	A user registers on the event management platform by providing necessary personal details, creating a secure password, and verifying their email address.
<b>Trigger:</b>	The user initiates the registration process by clicking the "Sign Up" button on the platform.
<b>Preconditions:</b>	PRE-1: The user must have a valid email address. PRE-2: The system must be operational.
<b>Postconditions:</b>	POST-1: The user's account is created and stored in the database. POST-2: The user receives a verification email.
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1. The user clicks the "Sign Up" button on the platform.</li> <li>2. The system displays a registration form asking for details such as name, email, password, and contact information.</li> <li>3. The user fills out the form and submits it.</li> <li>4. The system validates the entered details (e.g., email format, password strength).</li> <li>5. If the details are valid, the system creates the user account and sends a verification email.</li> <li>6. The user clicks the verification link in the email.</li> <li>7. The system verifies the link and activates the user's account.</li> </ol>
<b>Alternative Flows:</b> [Alternative Flow 1 – Not in Network]	<b>AF-1: Invalid Email Address</b> <ol style="list-style-type: none"> <li>1. The system detects an invalid email format.</li> <li>2. The system displays an error message and prompts the user to correct the email.</li> </ol> <b>AF-2: Weak Password</b> <ol style="list-style-type: none"> <li>1. The system detects that the password does not meet security requirements.</li> <li>2. The system displays a message with password requirements (e.g., at least 8 characters, includes uppercase, lowercase, and a number).</li> </ol>
<b>Exceptions:</b>	<b>EX-1: Email Already Registered</b> <ol style="list-style-type: none"> <li>1. The system detects that the entered email is already registered.</li> <li>2a. If the user chooses "Forgot Password," redirect them to the password recovery flow.</li> <li>2b. If the user cancels, the use case is terminated.</li> </ol> <b>EX-2: Email Verification Link Expired</b> <ol style="list-style-type: none"> <li>1. The user clicks an expired verification link.</li> <li>2. The system prompts the user to request a new verification email.</li> </ol>
<b>Business Rules</b>	BR-1: Passwords must be at least 8 characters long and include uppercase, lowercase, and numeric characters. BR-2: Users must verify their email address within 24 hours to activate their account.
<b>Assumptions:</b>	<ol style="list-style-type: none"> <li>1. Users have a valid email address and access to the verification email.</li> <li>2. Users have basic knowledge of filling out online registration forms.</li> </ol>

<b>Use Case ID:</b>	US-005
<b>Use Case Name</b>	Send Notification
<b>Actors</b>	<b>Primary Actor:</b> System <b>Secondary Actor:</b> User (Event Organizer or Attendee)
<b>Description</b>	The system sends notifications to users regarding event updates, reminders, cancellations, or other relevant information.
<b>Trigger</b>	A predefined condition, such as a change in event details, a scheduled reminder time, or user action, initiates the notification.
<b>Preconditions</b>	PRE-1: The user must have opted-in to receive notifications. PRE-2: The system must have valid contact details (e.g., email, phone)..
<b>Postconditions</b>	POST-1: The notification is successfully delivered to the user. POST-2: The user is informed about the event-related update.
<b>Normal Flow</b>	1. The system identifies a notification trigger (e.g., event update, reminder). 2. The system determines the target audience for the notification. 3. The system composes the notification message, including relevant details such as event name, time, and updates. 4. The system sends the notification via the selected medium (e.g., email, SMS, app notification). 5. The user receives the notification and takes appropriate action (e.g., reviewing event details).
<b>Alternative Flows</b>	- <b>AF-1: Notification Medium Not Available</b> 1. The selected notification medium (e.g., SMS) is unavailable. 2. The system retries using an alternate medium (e.g., email). - <b>AF-2: User Opted Out</b> 1. The system detects that the user has opted out of notifications. 2. The system skips sending the notification to that user.
<b>Exceptions</b>	- <b>EX-1: Delivery Failure</b> 1. The notification fails to reach the user due to incorrect contact details or network issues. 2a. The system logs the failure and retries. 2b. If retries fail, the system notifies the administrator.
<b>Business Rules</b>	- BR-1: Notifications must respect user preferences (e.g., frequency, medium). - BR-2: Critical notifications (e.g., event cancellations) must be prioritized and sent immediately. - BR-3: Notifications must comply with data protection regulations (e.g., GDPR, CAN-SPAM).
<b>Assumptions</b>	- Users have provided valid contact information. - The system's notification feature is integrated with reliable delivery services (e.g., email servers, SMS gateways). - Users will check and act on received notifications.

<b>Use Case ID:</b>	US-006
<b>Use Case Name</b>	Access Admin Dashboard
<b>Actors</b>	<b>Primary Actor:</b> Admin
<b>Description</b>	The admin accesses a centralized dashboard to manage events, users, notifications, and system configurations.
<b>Trigger</b>	The admin logs into the system and navigates to the dashboard.
<b>Preconditions</b>	PRE-1: The admin must have valid credentials. PRE-2: The admin role must have the necessary permissions to access the dashboard.
<b>Postconditions</b>	POST-1: Admin actions (e.g., adding or editing events) are successfully recorded in the system. POST-2: Dashboard updates reflect recent changes.
<b>Normal Flow</b>	<ol style="list-style-type: none"> <li>1. The admin logs into the system using their credentials.</li> <li>2. The system verifies the admin's role and permissions.</li> <li>3. The admin navigates to the dashboard.</li> <li>4. The system displays an overview of system metrics, such as active events, registered users, and notifications.</li> <li>5. The admin performs management actions, such as: Viewing and editing event details. Adding or removing users. Sending bulk notifications. Monitoring system performance logs.</li> <li>6. The system confirms the successful completion of each action.</li> </ol>
<b>Alternative Flows</b>	<b>AF-1: Insufficient Permissions</b> <ol style="list-style-type: none"> <li>1. The system detects that the admin lacks necessary permissions for a specific action.</li> <li>2. The system displays an error message and prevents the action.</li> </ol> <b>AF-2: Invalid Admin Credentials</b> <ol style="list-style-type: none"> <li>1. The admin enters invalid credentials.</li> <li>2. The system displays an error and prompts the admin to retry.</li> </ol>
<b>Exceptions</b>	<b>EX-1: System Downtime</b> <ol style="list-style-type: none"> <li>1. The system is temporarily unavailable.</li> <li>2. The admin receives a downtime notification and an estimated recovery time.</li> </ol> <b>EX-2: Data Synchronization Error</b> <ol style="list-style-type: none"> <li>1. Changes made by the admin fail to sync with the database.</li> <li>2. The system logs the error and notifies technical support.</li> </ol>
<b>Business Rules</b>	BR-1: Only users with the admin role can access the admin dashboard. BR-2: All admin actions must be logged for audit purposes. BR-3: Sensitive actions (e.g., deleting a user) require confirmation.
<b>Assumptions</b>	The admin has basic knowledge of navigating the dashboard. <ol style="list-style-type: none"> <li>1. The admin role is predefined with appropriate permissions.</li> <li>2. The system provides real-time updates on dashboard metrics.</li> </ol>

## Functional Requirements

The Event Management System (EMS) has several functional requirements to ensure smooth and efficient operations for both administrators and users. Firstly, the system must allow the admin to log in securely to access the admin panel using valid credentials. Once logged in, the admin should be able to add venue details such as name, location, capacity, and availability, as well as register new events by specifying details like event name, description, date, time, duration, and associated venue.

For users, the system must provide functionality to search for available event slots and book them based on their preferences. Users should also have the ability to confirm or cancel their bookings, with proper notifications being sent to both the admin and the user.

The EMS should maintain data consistency to prevent double-booking of venues or time slots and provide an intuitive, user-friendly interface for all stakeholders. It must log all actions performed by admins and users for auditing and troubleshooting purposes.

**Table 6 Show the functional requirement template**

<b>Identifier</b>	FR-001
<b>Title</b>	Create Event
<b>Requirement</b>	<b>User Perspective:</b> "The event organizer shall be able to create an event by providing event details such as title, description, date, time, location, and ticket information." <b>System Perspective:</b> "Upon submission of the event creation form, the system shall validate the input, save the event details in the database, and display a confirmation message."
<b>Source</b>	Feedback from event organizers and system design requirements.
<b>Rationale</b>	To allow organizers to list their events on the platform for attendees to view and register.
<b>Business Rule (if required)</b>	Event dates must not conflict with existing events for the same organizer. The system must validate all mandatory fields before saving the event details.
<b>Dependencies</b>	FR-003 (Search Event),
<b>Priority</b>	High

**Table 6 Show the functional requirement template**

<b>Identifier</b>	FR-002
<b>Title</b>	User Registration
<b>Requirement</b>	<p><b>User perspective:</b> "The user shall be able to register by providing their name, email, password, and phone number."</p> <p><b>System perspective:</b> "When the registration form is submitted, the system shall validate the data and create a new user record in the database."</p>
<b>Source</b>	End-user feedback during requirement elicitation sessions.
<b>Rationale</b>	To allow new users to access the platform's features and manage their account information.
<b>Business Rule (if required)</b>	The system must validate that email addresses are unique and passwords meet security standards.
<b>Dependencies</b>	FR-002 (User Login).
<b>Priority</b>	High

**Table 6 Show the functional requirement template**

<b>Identifier</b>	FR-003
<b>Title</b>	Search Event
<b>Requirement</b>	<p><b>User Perspective:</b> "The user shall be able to search for events by providing keywords, event categories, location, date, or organizer name."</p> <p><b>System Perspective:</b> "When the user submits a search query, the system shall filter the events in the database based on the provided criteria and display the matching results."</p>
<b>Source</b>	User feedback and competitive analysis.
<b>Rationale</b>	To allow users to quickly find events of interest and improve the platform's usability.
<b>Business Rule (if required)</b>	<p>The system must provide relevant results within 2 seconds of a query submission.</p> <p>- The system should prioritize events based on relevance and recency.</p>
<b>Dependencies</b>	FR-001 (Create Event)
<b>Priority</b>	High

**Table 6 Show the functional requirement template**

<b>Identifier</b>	FR-004
<b>Title</b>	Send Notification
<b>Requirement</b>	<p><b>User Perspective:</b> "The system shall notify users about important updates, reminders, or changes related to their events (e.g., registration confirmation, event updates, or cancellations)."</p> <p><b>System Perspective:</b> "Upon triggering an event-related update, the system shall send notifications via email, SMS, or push notifications to all relevant users."</p>
<b>Source</b>	Stakeholder discussions and user experience enhancement goals.
<b>Rationale</b>	To ensure users are informed promptly about critical information regarding events they are attending or organizing.
<b>Business Rule (if required)</b>	Notifications must be sent within 1 minute of an event trigger. The system must provide options for users to opt-in or opt-out of specific types of notifications.
<b>Dependencies</b>	FR-001 (Create Event), FR-003 (Search Event),
<b>Priority</b>	High

**Table 6 Show the functional requirement template**

<b>Identifier</b>	FR-005
<b>Title</b>	User Login
<b>Requirement</b>	<p><b>User Perspective:</b> "The user shall be able to log into the system by providing a valid username and password."</p> <p><b>System Perspective:</b> "When a user submits a login request with a valid username and password, the system shall authenticate the user and grant access to their account, redirecting them to the user dashboard or the appropriate landing page."</p>
<b>Source</b>	System requirements based on user authentication needs.
<b>Rationale</b>	To provide secure access to the platform for registered users.
<b>Business Rule (if required)</b>	The system must enforce strong password policies (e.g., minimum length, inclusion of special characters). The system must lock accounts after 5 failed login attempts and notify the user via email.
<b>Dependencies</b>	FR-008 (User Registration)
<b>Priority</b>	High

**Table 6 Show the functional requirement template**

<b>Identifier</b>	FR-006
<b>Title</b>	Access Admin Dashboard
<b>Requirement</b>	<p><b>User Perspective:</b> "The system administrator shall be able to access the admin dashboard to manage events, view system analytics, manage users, and configure system settings."</p> <p><b>System Perspective:</b> "When the system administrator logs in, the system shall grant access to the admin dashboard and display a user interface with all necessary administrative controls and analytics."</p>
<b>Source</b>	System design requirements and admin user feedback.
<b>Rationale</b>	To allow the system administrator to manage the platform, monitor activities, and perform system maintenance tasks.
<b>Business Rule (if required)</b>	<p>Only users with admin roles can access the admin dashboard.</p> <p>Admin users must be authenticated before accessing the dashboard.</p> <p>The system must log all admin actions for auditing purposes.</p>
<b>Dependencies</b>	FR-001 (create Events).
<b>Priority</b>	High

## Non-Functional Requirements

### Usability

- The system must have a user-friendly interface for both admins and users, with intuitive navigation and minimal training requirements.
- The design should follow accessibility standards (e.g., WCAG 2.1) to ensure usability for all users, including those with disabilities.

### Performance

- The system must handle at least 500 concurrent users without performance degradation.
- All user actions, such as booking a slot or confirming a reservation, must be processed within 3 seconds.