

Software Requirements Specification (SRS)

Event Booking System

Software Requirements Specification (SRS)	1
Event Booking System.....	1
1. Introduction.....	2
1.1 Purpose.....	2
1.2 Document Conventions.....	2
1.3 Intended Audience and Reading Suggestions.....	2
1.4 Project Scope.....	2
1.5 References.....	2
2. Overall Description.....	3
2.1 Product Perspective.....	3
2.2 Product Features.....	3
2.3 User Classes and Characteristics.....	3
2.4 Operating Environment.....	3
2.5 Design and Implementation Constraints.....	3
2.6 Assumptions and Dependencies.....	3
3. System Features.....	4
3.1 Functional Requirements.....	4
3.1.1 User Accounts & Authentication.....	4
3.1.2 Event Management.....	4
3.1.3 Event Discovery.....	4
3.1.4 Ticket Booking.....	4
3.1.5 Payments.....	4
3.1.6 My Tickets.....	4
3.1.7 Organizer Dashboard.....	4
3.1.8 Admin Management.....	5
3.1.9 Advanced Features (Phase 2).....	5
4. External Interface Requirements.....	5
4.1 User Interfaces.....	5
4.2 Hardware Interfaces.....	5
4.3 Software Interfaces.....	5
4.4 Communications Interfaces.....	5
5. Nonfunctional Requirements.....	5
5.1 Performance Requirements.....	5
5.2 Safety Requirements.....	6
5.3 Security Requirements.....	6
6. Appendix.....	6

1. Introduction

1.1 Purpose

This document provides a detailed description of the Software Requirements Specification (SRS) for the **Event Booking System**. It defines functional and nonfunctional requirements to guide developers, testers, project managers, and stakeholders throughout the system development lifecycle.

1.2 Document Conventions

- **Bold text** indicates main features or roles
- *Italic text* highlights important terms
- Numbered lists represent system requirements
- "Shall" is used to indicate mandatory requirements

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Software developers
- System analysts
- Project managers
- Testers
- Academic evaluators.

1.4 Project Scope

The Event Booking System is a web-based platform that allows users to discover events, book tickets, and make payments online. Event organizers can create and manage events, while administrators oversee system operations, approvals, and security.

1.5 References

- IEEE 830 / IEEE 29148 SRS Standard
- REST API Design Guidelines
- JWT Authentication Documentation

2. Overall Description

2.1 Product Perspective

The Event Booking System is a standalone web application that may integrate with external payment gateways. It follows a client-server architecture with RESTful APIs and role-based access control.

2.2 Product Features

- User authentication and role management
- Event creation and approval
- Event search and discovery
- Ticket booking and payment processing
- Organizer and admin dashboards
- Advanced features such as ticket resale and gift cards

2.3 User Classes and Characteristics

1. **Customer**
 - Browse and search events
 - Book and pay for tickets
 - View purchased tickets
2. **Event Organizer**
 - Create and manage events
 - View ticket sales and attendees
 - Monitor revenue
3. **Admin**
 - Manage users
 - Approve or block events
 - Monitor system activity

2.4 Operating Environment

- Web browsers (Chrome, Firefox, Edge)
- Backend server (Java/Spring Boot or similar)
- Database server (MySQL/PostgreSQL)
- Internet-based deployment

2.5 Design and Implementation Constraints

- JWT-based authentication must be used
- Secure payment handling is mandatory
- Role-based access control is required
- Compliance with academic project guidelines

2.6 Assumptions and Dependencies

- Users have internet access
- Payment gateway availability
- QR code generation library availability

3. System Features

3.1 Functional Requirements

3.1.1 User Accounts & Authentication

- The system shall allow users to register and log in
- The system shall use JWT for authentication
- The system shall support Customer, Organizer, and Admin roles
- Users shall manage their profiles

3.1.2 Event Management

- Organizers shall create, update, and delete events
- Events shall include title, description, venue, date/time, category, ticket price, and quantity
- Admin shall approve or reject events

3.1.3 Event Discovery

- Users shall browse all available events
- Users shall search and filter events by keyword, category, location, and date
- The system shall support pagination

3.1.4 Ticket Booking

- Users shall select ticket quantity
- The system shall check ticket availability in real time
- Booking statuses shall include Pending, Confirmed, and Cancelled
- Each ticket shall have a unique ID (QR-ready)

3.1.5 Payments

- The system shall process secure payments
- Payment statuses shall include Pending, Successful, and Failed
- Payments shall be linked to bookings
- Mock or local payment gateways shall be supported

3.1.6 My Tickets

- Users shall view purchased tickets
- Users shall see booking and payment status
- Users shall access event details

3.1.7 Organizer Dashboard

- Organizers shall manage their events
- Organizers shall view ticket sales and attendee lists
- Organizers shall view revenue summaries

3.1.8 Admin Management

- Admin shall manage users
- Admin shall approve or block events
- Admin shall monitor bookings and payments

3.1.9 Advanced Features (Phase 2)

- Users shall resell tickets with ownership verification
- Ticket ownership shall be transferable
- Users shall purchase and redeem gift cards
- Gift card balance and expiry shall be tracked

4. External Interface Requirements

4.1 User Interfaces

- Responsive web-based UI
- Separate dashboards for Customer, Organizer, and Admin
- Accessible and user-friendly design

4.2 Hardware Interfaces

- No specific hardware requirements
- Optional QR code scanners for ticket validation

4.3 Software Interfaces

- Payment gateway APIs
- Email/SMS notification services (optional)
- QR code generation library

4.4 Communications Interfaces

- HTTPS protocol for secure communication
- RESTful API communication

5. Nonfunctional Requirements

5.1 Performance Requirements

- System shall support concurrent users
- Search results shall load within acceptable response times

5.2 Safety Requirements

- Data loss prevention through regular backups
- Graceful error handling

5.3 Security Requirements

- Secure authentication using JWT
- Encrypted data transmission (HTTPS)
- Role-based access control
- Protection against common web vulnerabilities

6. Appendix

A. Abbreviations

- JWT: JSON Web Token
- SRS: Software Requirements Specification
- UI: User Interface

B. Future Enhancements

- Mobile application support
- Event recommendations
- Analytics and reporting dashboards