bookmyshow

Software Requirements Specification (SRS) document for bookmyshow
(Online ticket booking system)

Group Members:

Jeevan Reddy T 2023BCS-069

David S 2023BCS-066

Sameer Y 2023BCS-073

TABLE OF CONTENTS

1. Introduction

1.A. Purpose

The purpose of this document is to specify the software requirements for the development of an online ticket booking platform similar to BookMyShow. This document will help in the design, development, testing, verification, and deployment of the platform.

1.B. Scope

This online ticket booking platform aims to provide users with a seamless experience for booking tickets for movies, concerts, and other events. It will include features such as user accounts, event listings, ticket booking, payment integration, seat selection, notifications, reviews and ratings, and settings related to user preferences and security.

2. Overall Description

2.A. Product Perspective

This online ticket booking platform will be developed as a separate web and mobile application, providing users with a smooth experience across different devices. It will integrate with third-party payment gateways and event organizers for seamless ticket booking and management. The platform will rely on a backend service to handle user authentication, data storage, and communication.

2.B. Product Features

- **User Registration and Profiles Management:** Users can create, deactivate, or delete their accounts, set up profiles with personal details, preferences, and payment information.
- **Event Search and Booking:** Users can search for movies, concerts, and events, view details, and book tickets.
- **Seat Selection:** Users can choose their preferred seats from an interactive seating layout before booking.
- **Notifications:** Users receive notifications for upcoming events, ticket confirmations, and special offers.
- **Movie and Event Reviews:** Users can rate and review movies and events, helping others make informed decisions.
- **E-Tickets and QR Codes:** Users receive digital tickets with QR codes for easy check-in at venues.
- **Secure Payments:** Multiple payment options, including credit/debit cards, UPI, and wallets, ensure a safe transaction process.
- **Cancellation and Refund Policy:** Users can cancel tickets based on event policies and receive refunds as applicable.
- **Personalized Recommendations:** The platform suggests movies and events based on user preferences and history.

- **Customer Support:** Users can access 24/7 customer support for queries and issue resolution.

2.C. User Classes

This platform can be used by 5 different user groups:

- **Individual Users:** Can browse events, book tickets, make payments, download e-tickets, and manage bookings.
- **Event Organizers:** Can create and manage event listings, set ticket prices, track sales, and analyze event performance.
- **Cinema/Theater (Venue) Owners:** Can manage seating layouts, oversee bookings, and coordinate logistics with organizers.
- **Customer Support Staff:** Can assist with booking issues, cancellations, refunds, and customer queries.
- **Administrators:** Can manage user accounts, moderate events, configure system settings, monitor security, and oversee platform maintenance.

2.D. Operating Environment

This platform can be accessed on various devices, including smartphones, tablets, laptops, and PCs. It will be compatible with operating systems such as Windows, macOS, Android, and iOS.

2.E. Design and Implementation Constraints

2.E.1. Technological Stack:

- **Backend:** Node.js for server-side development
- **Database:**

  1.MongoDB is great for **unstructured** or **semi-structured data** (e.g., user preferences, movie details, event metadata).

  2.MySQL is ideal for **structured** data that requires **ACID compliance** (e.g., transactions, payments, and user accounts).

- **Frontend:** React.js for building the web interface
- **Development Tools:** Visual Studio Code, Postman
- **Diagram Drawing Tools:** Draw.IO, Visio, Figma

2.E.2. Security:

This platform ensures user authentication, data encryption, and access control measures to prevent unauthorized access.

The platform must comply with GDPR and other relevant data protection regulations for user privacy.

2.F. User Documentation

2.F.1. Getting Started:

- **User Registration:** Steps to create an account on the platform.
- **Logging In:** Instructions for logging into the platform using credentials.

2.F.2. Interface Overview:

- **Home Page:** Overview of the main dashboard where users can browse movies, concerts, and events.
- **Profile:** Explanation of the user profile page, where users can manage bookings, preferences, and payment methods.
- **Notifications:** Description of the notifications section showing updates on bookings, offers, and event reminders.
- **Explore:** Overview of the explore section for discovering trending movies, upcoming events, and special screenings.

2.F.3. Features:

- **Ticket Booking:** Instructions for searching, selecting, and booking tickets for movies and events.
- **Seat Selection:** Guide on choosing seats from the interactive seating layout.
- **E-Tickets:** Explanation of digital tickets and QR codes for hassle-free check-ins.
- **Cancellations and Refunds:** Steps to cancel a booking and understand refund policies.

2.F.4. Privacy and Security:

- **Account Privacy:** Instructions for customizing privacy settings related to personal information and transaction history.
- **Data Encryption:** Information on how user data, including payment details, is encrypted for security.
- **Reporting Issues:** Guidance on reporting fraudulent activities or incorrect bookings.

2.F.5. Support and Help:

- **Contact Information:** Details for reaching customer support.
- **Help Articles and FAQs:** Links to troubleshooting guides and frequently asked questions (FAQs).

3. Specific Requirements

3.A. Functional Requirements:

3.A.1.User Registration and Profile Management:

- Users can create accounts using email and phone verification.

- Users can securely log in using credentials.
- Users can edit profile details and preferences.
- Users can deactivate or delete accounts with authentication.

3.A.2. Event Browsing and Booking:

- Users can browse available events, movie shows, and concerts.
- Users can filter events by location, genre, and date.
- Users can view event details, including ratings and reviews.

3.A.3. Seat Selection:

- Users can select seats based on real-time availability.
- Users can view seat pricing before confirmation.

3.A.4. Payment and Transactions:

- Users can complete bookings using multiple payment methods (credit/debit card, UPI (paytm, Phonpe, googlepay), etc).
- Users receive booking confirmations and e-tickets post-payment.

3.A.5. Notifications and Alerts:

- Users get reminders for upcoming events.
- Users receive alerts for special offers and discounts.

3.A.6. Search Functionality:

- Users can search for movies, events, and venues.

3.A.7. Ticket Cancellation and Refunds:

- Users can cancel tickets as per platform policies.
- Users can track refund status in their accounts.

3.A.8. Privacy Settings:

- Users can edit privacy settings for account security.
- Users can manage notifications and alerts preferences.

3.B. External Interface Requirements

3.B.1. User Interface

The user interface of the platform will be user-friendly, featuring options for browsing movies, selecting showtimes, booking tickets, and making payments. It will be designed for accessibility across various devices with a responsive and intuitive layout.

3.B.2. Hardware Interface

This platform will be accessible on standard hardware components such as smartphones, tablets, laptops, and desktop computers. It will leverage device features such as GPS for location-based recommendations and barcode/QR code scanning for ticket validation.

3.B.3. Software Interface

The platform will integrate with operating systems such as Android, Windows, macOS, and web browsers like Chrome, Firefox, and Microsoft Edge. It will use APIs and SDKs for payment gateways, location services, and third-party ticketing systems to enhance the user experience.

3.C. Non-Functional Requirements

3.C.1. Performance

- Response time for searching movies, booking tickets, and processing payments should be within specific limits.
- The platform must maintain an uptime of 99.99%, ensuring minimal downtime and service disruptions.
- How it will be achieved:
    1. **Load Balancing**: Utilize a load balancer (e.g., AWS Elastic Load Balancer) to distribute user traffic across multiple backend servers, helping maintain performance during traffic spikes.
    2. **Caching**: Implement caching mechanisms (e.g., Redis, Memcached) to reduce server load by storing frequently accessed data in memory, speeding up response times for common requests.

3.C.2. Reliability

- The platform should operate reliably without downtime, especially during peak booking hours.
- Data integrity and consistency should be maintained across all transactions.

3.C.3. Security

- All user data (including passwords and payment details) will be encrypted using **AES-256** encryption.
- How it will be achieved:
    - **Data-at-Rest Encryption**: Use AES-256 encryption to securely store sensitive data such as user passwords and payment information in the database. This ensures that any sensitive data is encrypted both in the database and in backup storage.

3.C.4. Usability

- The platform must be easy to use with a responsive design for all devices.
- **How it will be achieved**:
    - **Responsive Web Design**: The front-end will be built using responsive design techniques (e.g., Flexbox, CSS Grid) and frameworks like Bootstrap or Material-UI to ensure the platform is accessible on any device (desktop, tablet, mobile).
    - **Continuous Improvements**: Regular usability testing will be conducted with users to identify pain points in the user journey and continuously improve the interface.

3.C.5. Scalability

- The system should scale to handle an increasing number of users and events.
- **How it will be achieved**:
    - **Horizontal Scaling**: The application will be designed to scale horizontally by adding more server instances to meet the demand. For databases, horizontal scaling can be achieved using sharding (splitting data across multiple database servers) or read replicas to distribute load.
    - **Cloud-Native Infrastructure**: Leverage cloud platforms (AWS, GCP, Azure) with services that can auto-scale and expand as the number of users or events grows. This ensures seamless scaling of both the application & database components.
    - **Asynchronous Processing**: Offload non-critical tasks such as notifications, reports, or processing large media files to background jobs using a task queue (e.g., **AWS SQS, Celery**) to keep the system responsive under a high load.