Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this document is to describe the functional and non-functional requirements of the **Movie Ticket Booking System (MVP)**.
This system allows users to browse movies, view showtimes, book seats, and manage bookings, while administrators manage movies, halls, schedules, and system data.

This SRS serves as a reference for developers, instructors, and evaluators to understand system behavior and scope.

1.2 Scope

The Movie Ticket Booking System is a web-based application designed to manage cinema operations. The system supports multiple user roles including **Guest**, **Registered User**, and **Admin**.
Core functionalities include movie browsing, ticket booking, seat selection, payment processing (conceptual in MVP), and administrative management.

The current version focuses on a **Minimum Viable Product (MVP)** with extensibility for future enhancements.

1.3 Definitions, Acronyms, and Abbreviations

- **MVP**: Minimum Viable Product
- **PK**: Primary Key
- **FK**: Foreign Key
- **CRUD**: Create, Read, Update, Delete
- **RBAC**: Role-Based Access Control

---

2. Overall Description

2.1 Product Perspective

The system follows a **multi-role architecture** where access and permissions are determined by user roles.
It is designed using a relational database structure with well-defined entity relationships to ensure data integrity and scalability.

2.2 Product Functions

The system provides the following high-level functions:

- Browse movies (currently showing and upcoming)
- View movie details and showtimes
- Search and filter movies
- Book tickets and select seats
- Manage bookings and user profiles
- Administrative management of movies, halls, seats, and schedules
- Generate reports and analytics (Admin)

2.3 User Classes and Characteristics

User Type	Description
Guest	Unauthenticated user who can browse movies and view details
Registered User	Authenticated customer who can book tickets and manage bookings
Admin	System administrator with full management privileges

2.4 Operating Environment

- Web-based application
- Backend: MVC / Spring Boot (or similar)
- Database: Relational Database (MySQL / Oracle / PostgreSQL)
- Client: Web browser

2.5 Design Constraints

- Role-based access control must be enforced
- Seats cannot be double-booked for the same showtime
- Database normalization must be maintained
- MVP scope limits advanced features like refunds and external payment gateways

---

3. System Roles and Permissions

3.1 Guest (Unauthenticated User)

- Browse movies (currently showing & upcoming)
- View movie details
- View showtimes
- Search and filter movies
- View cinema information

3.2 Registered User (Customer)

Includes all Guest permissions, plus:

- Register and login
- Book tickets
- Select seats
- Make payments (logical flow)
- View and download digital tickets
- View booking history
- Cancel bookings within allowed time
- Update profile information
- Rate and review movies (optional)

3.3 Admin

Includes all User permissions, plus:

- Add, edit, and delete movies
- Manage movie schedules and showtimes
- Manage cinema halls and seat configurations
- View all bookings and reservations

- Generate reports (sales, occupancy)
- Manage user accounts
- Set ticket prices
- Manage genres and categories
- View dashboard analytics

---

4. Functional Requirements

FR1: User Registration and Authentication

- The system shall allow users to register using email and password.
- The system shall authenticate users during login.
- The system shall assign roles to users.

FR2: Movie Management

- The system shall allow admins to add, update, and delete movies.
- Each movie shall include title, description, duration, poster, and status.

FR3: Showtime Management

- The system shall allow admins to create showtimes for movies.
- Each showtime shall be linked to one movie and one hall.
- Each showtime shall have date, time, and ticket price.

FR4: Hall and Seat Management

- The system shall allow admins to manage cinema halls.
- Each hall shall contain multiple seats.
- Seats shall be uniquely identified by row and seat number.

FR5: Ticket Booking

- Registered users shall be able to book tickets for a showtime.
- Users shall select one or more available seats.
- The system shall prevent double booking of seats for the same showtime.

FR6: Booking Management

- The system shall generate a booking record per transaction.
- Users shall view booking history.
- Users shall cancel bookings within allowed constraints.

FR7: Booking-Seat Mapping

- The system shall maintain a many-to-many relationship between bookings and seats.
- A junction table (BookingSeat) shall be used.
- BookingSeat shall include showtime_id to ensure seat reusability.

FR8: Reporting and Analytics

- The system shall allow admins to view booking statistics.
- The system shall generate sales and occupancy reports.

---

5. Data Requirements (Entities)

Core Entities (MVP)

1. **User** (id, name, email, password, role)
2. **Movie** (id, title, description, duration, poster_url, status)
3. **Hall** (id, name, capacity)
4. **Seat** (id, hall_id, row, seat_number, type)
5. **Showtime** (id, movie_id, hall_id, show_date, show_time, price)
6. **Booking** (id, user_id, showtime_id, total_price, status)
7. **BookingSeat** (booking_id, seat_id, showtime_id)

---

6. Entity Relationships

- User → Booking (One-to-Many)
- Movie → Showtime (One-to-Many)
- Hall → Showtime (One-to-Many)
- Hall → Seat (One-to-Many)

- Showtime → Booking (One-to-Many)
- Booking ↔ Seat (Many-to-Many via BookingSeat)

This design ensures:

- No double booking
- Seat reuse across different showtimes
- Accurate booking history
- Scalable scheduling system

---

7. Non-Functional Requirements

7.1 Performance

- The system shall support multiple concurrent users.
- Seat availability checks shall be real-time.

7.2 Security

- Passwords shall be encrypted.
- Role-based authorization shall be enforced.
- Users shall access only permitted features.

7.3 Reliability

- The system shall ensure data consistency.
- Transactions shall be atomic and rollback on failure.

7.4 Usability

- The system shall have a simple and intuitive UI.
- Booking steps shall be clear and user-friendly.

7.5 Scalability

- The system shall support future extensions such as payments, reviews, and promotions.

---

8. Assumptions and Dependencies

Assumptions

- Users have internet access.
- Admins manage all cinema-related data.

Dependencies

- Database Management System
- Backend Framework (MVC / Spring Boot)
- Web Frontend

---

9. Future Enhancements

- Online payment gateway integration
- Loyalty programs
- Advanced seat pricing
- Movie reviews and ratings
- Mobile application support

---

10. Conclusion

The Movie Ticket Booking System (MVP) provides a structured and scalable solution for cinema management and ticket booking. The system is designed with clear separation of roles, normalized data structures, and extensibility for future growth, making it suitable for academic and real-world applications.