Software Requirements Specification

for

Theatre Management System

Prepared by

Batch: D

Dev Rai **2024301022** **dev.rai24@spit.ac.in**

Instructor: **Prof. Prasenjit Bhavathankar**

Course: **Software Engineering**

Date: **13th August 2025**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Dev Rai	Initial draft of the SRS document.	13/08/25

1 Introduction

1.1 Document Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the requirements for the **Theatre Management System**. It describes the system's purpose, features, and the constraints under which it will operate. This document will serve as a foundational agreement between the development team and stakeholders, and will be the basis for validating the final delivered system.

1.2 Product Scope

The Theatre Management System is a comprehensive web-based platform designed to streamline the management and booking process for movie theatres. The system will automate key operations including movie and screening management, online ticket booking, seat selection, and secure payment processing.
The primary goals are to enhance the user experience for customers by providing a seamless booking process and to improve operational efficiency for theatre administrators through a centralized management dashboard. Key features include real-time occupancy monitoring, dynamic report generation, and automated booking confirmations with downloadable PDF tickets.

1.3 Intended Audience and Document Overview

This document is intended for the following audiences:

- ***Developers & Design Team:*** *To understand the system's requirements for design, implementation, and testing.*
- ***Project Managers:*** *To oversee the project scope and progress.*
- ***Theatre Administrators (Client):*** *To verify that the system meets their operational needs and to serve as a basis for acceptance testing.*
- ***Customers (End-Users):*** *While not direct readers, their needs are the primary focus of the user-facing requirements.*

This document begins with an overall description of the product perspective, its core functionalities, user characteristics, and operational environment. It then delves into the specific functional and non-functional requirements, followed by a data dictionary in the appendix.

1.4 Definitions, Acronyms and Abbreviations

TMS: Theatre Management System
SRS: Software Requirements Specification
UI: User Interface
GUI: Graphical User Interface
DB: Database
PDF: Portable Document Format
HTTP: Hypertext Transfer Protocol
HTTPS: Hypertext Transfer Protocol Secure
CRUD: Create, Read, Update, Delete

1.5 Document Conventions

Header:
- Font Size: 10
- Font Style: Bold Italic
- Font: Times New Roman

Heading:
- Font Size: 18
- Font Style: Bold
- Font: Times New Roman

Sub Heading:
- Font Size: 14
- Font Style: Bold
- Font: Times New Roman

Content:
- Font Size: 12
- Font: Times New Roman

1.6 References and Acknowledgments

IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications.
Project guidelines and specifications provided by the course instructor.

2 Overall Description

2.1 Product Perspective

The Theatre Management System is a new, self-contained product designed to replace or supplement existing manual or semi-automated booking and management processes in movie theatres. It is a web-based application that will serve as the central hub for all customer and administrative interactions. The system will interface with an external payment gateway for financial transactions and will manage its own databases for movie, customer, and booking information.
A context diagram would show the Theatre Management System as the central process, interacting with external entities such as Customer, Administrator, and a Payment Gateway.

2.2 Product Functionality

The major functions of the TMS are as follows:

- **User Management**: Allows customers to register, log in, and manage their profiles.
- **Movie and Screening Management**: Enables administrators to perform CRUD operations on movie listings and schedule showtimes in different auditoriums (screens).
- **Ticket Booking**: Allows customers to browse movies, select a showtime, choose their seats from an interactive map, and book tickets.
- **Payment Processing**: Securely handles payments for ticket bookings via an integrated payment gateway.
- **Confirmation and Ticketing**: Generates and sends booking confirmations to users and provides an option to download the ticket as a PDF file.
- **Reporting and Analytics**: Provides administrators with dashboards to view theatre occupancy, booking statistics, and financial records.

2.3 Users and Characteristics

1. **Customer (End-User):**
   - Any individual with internet access wishing to book movie tickets.
   - Requires a simple and intuitive UI for browsing movies and completing the booking process in minimal steps.
   - No special training is required; should be familiar with standard web applications.

2. **Theatre Administrator:**
   - An employee of the theatre responsible for managing the system's content and operations.
   - Has privileged access to an administrative dashboard.
   - Responsible for adding/updating movie information, managing show schedules, and monitoring bookings.
   - Requires basic computer literacy and will be provided with documentation on using the admin panel.

2.4 Operating Environment

- **Server-Side:** The system will be deployed on a server running a Linux or Windows operating system with Node.js and a MySQL database installed.
- **Client-Side:** The system shall be accessible via modern web browsers (e.g., Google Chrome, Mozilla Firefox, Safari, Microsoft Edge) on desktops, tablets, and mobile devices.

- **Hardware:** The system will run on a server with at least 2GB RAM and 50GB storage space. Client machines require only an internet connection and a web browser.

2.5 Design and Implementation Constraints

- The backend must be developed using Node.js and the Express.js framework.
- The database must be MySQL.
- All payment transactions must be logged using a MySQL trigger for audit purposes.
- The system must use a modular architecture to ensure scalability and maintainability.
- The system must be web-browser enabled and use HTTPS for secure data transfer.
- The system must integrate with a third-party library for PDF generation.

2.6 User Documentation

User documentation will include:

- An online **Help/FAQ** section for customers to resolve common queries regarding booking, payments, and cancellations.
- A **System Administrator Guide** detailing how to manage movies, screenings, view reports, and perform other administrative tasks.

2.7 Assumptions and Dependencies

- *Assumptions:*
  - *Users have a stable and active internet connection.*
  - *Users are familiar with basic web browsing and online payment procedures.*

- *Dependencies:*
  - *The system is dependent on a third-party payment gateway service for processing transactions.*
  - *The system depends on a third-party library for generating PDF tickets.*
  - *The accuracy of movie details (e.g., synopsis, runtime, rating) is dependent on the data entered by the Administrator.*

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

i. Registration/Login Window:
- User: Customer, Administrator.
- Properties: Provides fields for email/username and password. Includes links for "Login", "Register", and "Forgot Password". On successful login, it redirects the user to their respective dashboard (Customer or Admin).

ii. Customer Homepage:
- User: Customer.
- Properties: Displays a list of currently running and upcoming movies with posters and titles. Provides search and filter functionality (e.g., by genre, language). Each movie listing links to a details page.

iii. Movie Details & Showtime Selection Page:
- User: Customer.
- Properties: Shows detailed information about a selected movie (synopsis, cast, duration, trailer). Lists all available showtimes for the upcoming days. Allows the user to select a showtime to proceed with booking.

iv. Seat Selection Page:
- User: Customer.
- Properties: Displays an interactive, graphical map of the theatre's seating layout for the selected showtime. Available, booked, and selected seats are color-coded. Users can click on available seats to select them. Displays the total price in real-time.

v. Administrator Dashboard:
- User: Administrator.
- Properties: A secure portal that provides access to all management functions: Movie Management, Screening Management, Booking Reports, and Payment Logs. Displays key metrics like daily revenue and occupancy rates.

3.1.2 Hardware Interfaces

As a web-based application, the Theatre Management System does not interface directly with most hardware. Instead, it relies on the abstractions provided by the operating system and the client's web browser. The key hardware interactions are as follows:

- **User Input Devices:** Users will interact with the system's web interface using standard hardware such as a keyboard, mouse, or touchscreen (on mobile devices and tablets). These devices are used to input data into forms (e.g., login credentials, search queries) and trigger actions (e.g., clicking buttons to select seats or confirm bookings).
- **Display Devices**: The system's graphical user interface will be rendered on the user's monitor or mobile device screen. The software's primary output is visual information in the form of HTML web pages.
- **Server Hardware:** The backend application resides on a server and interacts with the server's hard drive/SSD for storing application code, log files, and other assets. It also utilizes the server's Network Interface Card (NIC) to receive requests from and send responses to clients over the internet.
- **Peripheral Devices (Optional):**

- **QR Code Scanner:** At the theatre entrance, a hardware QR code scanner may be used to validate e-tickets. The scanner will read the QR code from a customer's printed or mobile ticket, decode the booking ID, and send it to a specific API endpoint of the TMS for real-time validation.
- **Ticket Printer:** For walk-in customers or administrative purposes, the system will support printing booking confirmations. The application will generate a printer-friendly HTML or PDF page, which can be sent to a standard or thermal printer via the browser's print functionality.

3.1.3 Software Interfaces

The Theatre Management System will run on a server and interact with several other software components to perform its functions.

- **Operating System (OS):** The application is designed to be OS-agnostic and will run as a server process on a mainstream operating system such as Linux (e.g., Ubuntu, CentOS) or Windows Server. The OS will provide essential services, including:
  - Managing the Node.js runtime process.
  - Providing access to the file system for logging and configuration file management.
  - Handling low-level network communications by managing TCP/IP sockets on the designated HTTP/HTTPS port (e.g., 80/443).
- **Database**: The system will connect to a MySQL (version 8.0 or later) database server to persist all application data. The mysql2 Node.js library will be used to manage the connection pool and execute SQL queries for all CRUD (Create, Read, Update, Delete) operations on user, movie, booking, and payment data.
- **APIs and Libraries:**
  - Payment Gateway: The system will integrate with an external payment gateway (e.g., Stripe, PayPal, Razorpay) for processing credit/debit card and other online payments. This interface involves redirecting the user to the gateway's secure payment page and receiving a callback/webhook with the transaction status.
  - PDF Generation Library: A Node.js library such as pdfkit or html-pdf will be used to dynamically generate PDF tickets for customers upon successful booking.
  - Email Service: The system will use an SMTP service or a dedicated email API (e.g., SendGrid, Mailgun) via a library like Nodemailer to send transactional emails, such as booking confirmations and password reset links.

3.1.4 Communications Interfaces

The communication for the Theatre Management System will primarily be facilitated through standard web protocols to ensure broad compatibility and security. The major communication standards are outlined below.

The primary communication protocol between the client (user's web browser) and the server will be the Hypertext Transfer Protocol (HTTP). All user interactions, data submissions, and page requests will be encapsulated within HTTP requests and responses. For dynamic, client-side interactions that do not require a full page reload, data will be exchanged between the frontend and backend API endpoints using the JSON (JavaScript Object Notation) format.

For security, all communications will be mandated to occur over HTTPS (HTTP Secure). This will ensure that the entire data stream between the client and server, including sensitive information like user credentials and session tokens, is encrypted. The system will rely on Transport Layer Security (TLS 1.2 or higher) as the encryption standard. Furthermore, for sending automated booking confirmations and other notifications, the system will communicate with an external mail server using the Simple Mail Transfer Protocol (SMTP).

3.2 Functional Requirements

3.2.1. Customer Registration

- Input: Customer name, email address, phone number, password.
- Output: Creation of a new customer account in the database. A success message is displayed.
- Description: The system validates the input data. The email must be unique. The password must be securely hashed before being stored.

3.2.2. Browse and View Movies

- Input: User navigates to the homepage.
- Output: A list of available movies.
- Description: The system retrieves and displays all movies marked as 'active' from the database.

3.2.3. Select Seats and Book Tickets

- Input: User selects a movie, a showtime, and a number of seats from the seat map.
- Output: The selected seats are temporarily locked. User is redirected to the payment page.
- Description: The system checks the real-time availability of the selected seats. If available, it locks them for a short duration (e.g., 10 minutes) to allow the user to complete the payment.

3.2.4. Process Payment

- Input: User's payment details (handled by the payment gateway interface).
- Output: A payment confirmation or failure message.
- Description: The system redirects the user to the payment gateway. Upon return, it processes the gateway's response. If successful, it confirms the booking. If failed, it releases the locked seats.

3.2.5. Generate PDF Ticket

- Input: A successful booking confirmation.
- Output: A downloadable PDF file of the ticket.
- Description: After a successful payment, the system generates a unique booking ID and creates a booking record. A PDF ticket is generated containing the movie name, showtime, seat numbers, a QR code (optional), and the booking ID. The booking is also confirmed via email.

3.2.6. Admin: Manage Movies

- Input: Movie title, synopsis, poster image, duration, genre, etc.
- Output: The movie is added, updated, or deleted from the database.
- Description: The administrator can perform full CRUD operations on the movie listings through a dedicated interface on the admin dashboard.

3.2.7. Admin: Manage Screenings

- Input: A selected movie, a theatre screen (auditorium), date, and time.
- Output: A new screening is scheduled and becomes visible to customers.
- Description: The administrator can schedule, update, or cancel showtimes. The system prevents scheduling conflicts for the same screen.

3.2.8. Admin: View Reports

- Input: Admin selects a report type (e.g., occupancy, revenue) and a date range.
- Output: A visual and tabular representation of the requested data.
- Description: The system queries the booking and payment logs to generate reports, helping the administrator track performance.

4 Other Non-functional Requirements

4.1 Performance Requirements

- The system response time for any user-facing page load shall be less than 2 seconds under normal network conditions.
- The seat availability check and locking mechanism must complete within 1 second to prevent race conditions.
- The system shall support at least 100 concurrent users performing booking operations without significant degradation in performance.

4.2 Safety and Security Requirements

- All user passwords must be stored in a hashed and salted format (e.g., using bcrypt).
- The system must enforce role-based access control. Only authenticated administrators shall have access to the admin dashboard.
- All data transmission containing sensitive information (passwords, payment details) must be encrypted using HTTPS.
- The system must include protection against common web vulnerabilities like SQL Injection and Cross-Site Scripting (XSS).

4.3 Software Quality Attributes

- Reliability: The system should have an uptime of 99.5%. The payment and booking process must be transactional; in case of any failure during booking creation after a successful payment, the system should have a recovery mechanism or notify the admin for manual reconciliation.
- Usability: The customer-facing interface must be intuitive, self-explanatory, and require minimal clicks to complete a booking. The admin dashboard shall be well-organized and easy to navigate.
- Maintainability: The modular architecture will allow developers to update or fix individual components (e.g., payment module, PDF module) with minimal impact on the rest of the system. Code must be well-documented.
- Scalability: The system should be designed to handle an increase in the number of movies, screens, and users by scaling the server resources horizontally or vertically.