**SM CINEMA BOOKING SYSTEM**

A Requirement Specification Document Presented to the

Faculty of Datamex College of Saint Adeline, Inc.

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science in Information Technology

By:

Jerusalem, Auldey Glen A.

September 2025

**REQUIREMENT SPECIFICATION DOCUMENT**

**INTRODUCTION**

This Requirements Specification Document defines the comprehensive functional and non-functional requirements for the Cinema Booking System developed for SM Cinema Caloocan Grand Central branch. This document serves as the authoritative guide for system development, establishing clear specifications for all stakeholders involved in the project implementation and future maintenance.

The Cinema Booking System is a web-based application designed to modernize and streamline the movie ticket booking process at SM Cinema Caloocan Grand Central. The system addresses current inefficiencies in seat selection by replacing the existing digital queue-based approach with real-time seat availability and immediate booking confirmation.

The system enables customers to browse current movie listings, view detailed movie information, select showtimes, and reserve specific seats through an intuitive web interface without requiring user account creation. Additionally, the system provides administrative personnel with a comprehensive dashboard for managing movie schedules, monitoring reservation activity, and accessing audit logs for operational oversight.

**Scope of the Requirements Specification**

This document encompasses all requirements for the Cinema Booking System implementation at SM Cinema Grand Central branch, including:

* Complete functional requirements for customer booking operations and administrative management
* Non-functional requirements covering performance, security, usability, and reliability standards
* System constraints, assumptions, and dependencies
* Data modeling and API specifications
* Use case definitions and acceptance criteria

**FUNCTIONAL REQUIREMENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement ID | Description | Priority | Dependencies | Acceptance Criteria |
| FR-001 | Display movie catalog and detailed information | High | None | Shows active movies with all details and showtimes |
| FR-002 | Provide real-time seat selection and booking | High | FR-001 | Interactive seat map with race condition prevention |
| FR-003 | Process booking confirmation and email delivery | High | FR-002 | Mock payment processing with email ticket delivery |
| FR-004 | Implement admin authentication and authorization | High | None | Token-based auth protecting sensitive API endpoints |
| FR-005 | Enable complete movie catalog management | High | FR-004 | CRUD operations for movies including poster uploads |
| FR-006 | Provide showtime scheduling and management | High | FR-005 | Create and manage showtimes with conflict prevention |
| FR-007 | Enable booking oversight and management | Medium | FR-004 | View and search customer bookings through admin panel |
| FR-008 | Implement comprehensive audit logging | Medium | FR-004 | Log all admin actions with viewing capabilities |

Table 1. System Functional Requirements

**NON-FUNCTIONAL REQUIREMENTS**

**Performance**

* System response time shall not exceed 3 seconds for standard operations under normal load
* System shall support up to 100 concurrent users during peak booking periods
* Database queries shall execute within 2 seconds for complex operations
* Real-time seat updates shall reflect within 1 second across all active session

**Usability**

* User interface shall be intuitive and require minimal learning curve for customers
* System shall be fully responsive and functional on desktop and mobile browsers
* Interface shall adapt seamlessly to different screen sizes and orientations
* Error messages shall be clear and provide actionable guidance to users

**Security**

* All API communications shall use HTTPS encryption
* Admin authentication shall use secure token-based authorization
* API endpoints shall implement rate limiting to prevent abuse
* Login attempts shall be throttled to prevent brute force attacks
* Sensitive endpoints shall require proper authentication tokens
* User data shall be protected according to standard privacy practices

**Scalability**

* System architecture shall support increased user load through API optimization
* Database design shall accommodate growing booking and movie data
* Backend API shall handle additional concurrent requests efficiently
* System shall maintain performance standards as data volume increases

**Maintainability**

* Code shall include comprehensive comments and documentation
* API endpoints shall be thoroughly documented with usage examples
* Git commits shall follow descriptive naming conventions
* System shall implement proper logging for debugging and monitoring
* Code structure shall follow established patterns for future enhancements
* Database schema shall be well-documented and version controlled

**USE CASES**

**UC-001: Browse Movie Listings**

Customer visits the cinema booking system to view available movies and their details. The customer can browse through active movie listings, view movie information including posters, descriptions, ratings, and other details to help make their selection decision.

**Actors:** Customer

**Preconditions:** System is operational and contains active movie listings

**Postconditions:** Customer has viewed available movies and can proceed to showtime selection

**Alternate Flows:**

* If no active movies are available, system displays appropriate message
* If movie images fail to load, system shows placeholder images
* Customer may exit without selecting any movie

**UC-002: Select Showtime**

After browsing movies, customer selects a specific movie to view available showtimes. The system displays all scheduled showtimes for the selected movie, organized by date and time, showing available seat counts for each showing.

**Actors:** Customer

**Preconditions:** Customer has selected a movie from the listings

**Postconditions:** Customer has selected a specific showtime and can proceed to seat selection

**Alternate Flows:**

* If no showtimes are available for selected movie, system informs customer
* If showtime is fully booked, system indicates no availability
* Customer may return to movie listings to select different movie

**UC-003: Select Seats and Complete Booking**

Customer views the theater seat map for their selected showtime, selects desired seats from available options, provides contact information, and completes the booking process with mock payment. The system processes the booking, generates confirmation, and sends email ticket.

**Actors:** Customer, System

**Preconditions:** Customer has selected a showtime with available seats

**Postconditions:** Booking is confirmed, seats are reserved, and email ticket is sent

**Alternate Flows:**

* If selected seats become unavailable during selection, system updates display and prompts reselection
* If email address is invalid, system requests correction
* If mock payment processing fails, system displays error and allows retry
* Customer may cancel booking process at any step before confirmation

**UC-004: Admin Authentication**

Administrative staff accesses the admin dashboard by providing valid credentials. The system validates credentials, generates authentication token, and grants access to administrative functions

**Actors:** Admin User

**Preconditions:** Admin has valid username and password credentials

**Postconditions:** Admin is authenticated and can access administrative functions

**Alternate Flows:**

* If credentials are invalid, system displays error message and logs failed attempt
* If multiple failed attempts occur, system may temporarily lock access
* If session expires, admin must re-authenticate to continue

**UC-005: Manage Movie Catalog**

Authenticated admin performs CRUD operations on the movie catalog including adding new movies with all required information, updating existing movie details, uploading poster images, and activating or deactivating movies for customer viewing.

**Actors:** Admin User

**Preconditions:** Admin is authenticated and has movie management permissions

**Postconditions:** Movie catalog is updated according to admin actions, changes are reflected in customer view

**Alternate Flows:**

* If required movie information is missing, system prevents saving and displays validation errors
* If poster image upload fails, system allows retry or continuation without image
* If movie deletion is attempted with existing bookings, system may prevent deletion or require confirmation

**UC-006: Manage Showtimes**

Authenticated admin creates and manages movie showtimes by selecting movies, setting dates and times, and assigning theater capacity. The system prevents scheduling conflicts and updates showtime availability for customer booking.

**Actors:** Admin User

**Preconditions:** Admin is authenticated and active movies exist in catalog

**Postconditions:** Showtimes are created or modified and available for customer booking

**Alternate Flows:**

* If scheduling conflict exists for same theater time slot, system prevents creation and suggests alternatives
* If showtime modification affects existing bookings, system may require admin confirmation
* Admin may cancel showtimes with appropriate system warnings about affected bookings

**UC-007: View Booking Management**

Authenticated admin views and manages customer bookings through administrative dashboard, including searching bookings by various criteria, viewing booking details, and monitoring reservation activity for operational oversight.

**Actors:** Admin User

**Preconditions:** Admin is authenticated and booking data exists

**Postconditions:** Admin has reviewed booking information as needed

**Alternate Flows:**

* If no bookings match search criteria, system displays appropriate message
* If booking details are corrupted, system displays available information with error indication

**UC-008: View Audit Logs**

Authenticated admin accesses comprehensive audit logs to review all administrative actions, login attempts, and system changes. Admin can filter logs by date range, action type, or user to monitor system usage and maintain operational accountability.

**Actors:** Admin User

**Preconditions:** Admin is authenticated and audit logging is active

**Postconditions:** Admin has reviewed audit information as required

**Alternate Flows:**

* If no audit entries match filter criteria, system displays empty results message
* If audit log data is extensive, system provides pagination for manageable viewing

**DATA REQUIREMENTS**

**Data Entities**

The Cinema Booking System manages the following data entities:

* **Movie** - Contains information about films available for screening
* **Showtime** - Represents scheduled screenings of movies at specific times
* **Booking** - Records customer reservations and ticket purchases
* **Customer** - Stores contact information for booking purposes
* **Theater** - Represents the physical screening room and seating configuration
* **Admin User** - Contains administrative staff login credentials and permissions
* **Audit Log** - Records all administrative actions for system accountability

|  |  |  |
| --- | --- | --- |
| **Entity** | **Attribute** | **Description** |
| Movie | title | Movie title/name |
| Movie | description | Detailed movie synopsis |
| Movie | genre | Movie category/genre |
| Movie | genre\_detail | Additional genre information |
| Movie | age\_rating | Content rating classification |
| Movie | age\_rating\_detail | Detailed rating explanation |
| Movie | poster | Movie poster image |
| Movie | duration | Runtime in minutes |
| Movie | release\_date | Official release date |
| Movie | language | Primary language of the film |
| Movie | trailer\_url | Link to movie trailer |
| Movie | is\_active | Status indicator for public visibility |
| Showtime | showtime\_id | Unique identifier |
| Showtime | movie\_id | Reference to associated movie |
| Showtime | theater\_id | Reference to screening theater |
| Showtime | show\_date | Date of screening |
| Showtime | show\_time | Time of screening |
| Showtime | available\_seats | Number of seats remaining |
| Showtime | total\_seats | Total theater capacity |
| Booking | booking\_id | Unique booking identifier |
| Booking | showtime\_id | Reference to selected showtime |
| Booking | customer\_name | Customer's full name |
| Booking | customer\_email | Customer's email address |
| Booking | selected\_seats | List of reserved seat numbers |
| Booking | booking\_reference | Unique confirmation code |
| Booking | booking\_timestamp | Date and time of booking |
| Booking | booking\_status  Table 2. Database Schema Specification | Current status of reservation |

|  |  |  |
| --- | --- | --- |
| Entity | Attribute | Description |
| Booking | booking\_timestamp | Date and time of booking |
| Booking | booking\_status | Current status of reservation |
| Theater | theater\_id | Unique theater identifier |
| Theater | theater\_name | Name or number of theater |
| Theater | total\_seats | Maximum seating capacity |
| Theater | seat\_layout | Configuration of seat arrangement |
| Admin User | admin\_id | Unique administrator identifier |
| Admin User | username | Login username |
| Admin User | password\_hash | Encrypted password |
| Admin User | is\_active | Account status indicator |
| Admin User | last\_login | Timestamp of most recent login |
| Audit Log | log\_id | Unique log entry identifier |
| Audit Log | admin\_id | Reference to admin user who performed action |
| Audit Log | action\_type | Category of action performed |
| Audit Log | action\_description | Detailed description of action |
| Audit Log | timestamp | Date and time of action |
| Audit Log | affected\_entity | Type of data modified |
| Audit Log | entity\_id | Identifier of modified record |

Table 3. Database Schema Specification

**Relationships**

* Movie to Showtime: One Movie can have multiple Showtimes (One-to-Many)
* Showtime to Booking: One Showtime can have multiple Bookings (One-to-Many)
* Theater to Showtime: One Theater can host multiple Showtimes (One-to-Many)
* Customer to Booking: One Customer can make multiple Bookings (One-to-Many)
* Admin User to Audit Log: One Admin User can generate multiple Audit Log entries (One-to-Many)
* Booking to Customer: Each Booking contains Customer contact information stored per booking

**ASSUMPTIONS AND CONSTRAINTS**

**Assumptions**

The development of the Cinema Booking System is based on several key assumptions. SM Cinema Caloocan Grand Central will provide comprehensive movie catalog information including titles, descriptions, ratings, and promotional materials in digital format suitable for system integration.

The cinema management will supply accurate theater seating configurations and capacity information to enable proper seat mapping functionality.

Users accessing the system are assumed to have stable internet connectivity sufficient for real-time seat selection and booking operations. Customers will have access to valid email addresses for receiving ticket confirmations and booking receipts. The target user base is expected to be familiar with basic web browsing and online form completion processes.

The cinema operates during standard business hours with predictable peak booking periods that can be accommodated by the system's performance specifications. Staff members responsible for administrative functions will have adequate computer literacy to operate the management dashboard effectively. Email delivery services will remain operational and reliable for sending automated booking confirmations to customers.

**Constraints**

The Cinema Booking System development faces several operational and technical constraints. The project operates under a strict two-month development timeline which limits the scope of features and extensive testing phases. Development resources are restricted to a single developer, requiring efficient project management and focused feature prioritization.

Technology implementation is constrained to React frontend framework and Django REST Framework for backend development, limiting architectural flexibility but ensuring consistency with established development expertise. The system scope is deliberately limited to SM Cinema Caloocan Grand Central branch only, excluding multi-location functionality or complex theater chain integration capabilities.

Payment processing functionality is excluded from the implementation scope, with the system designed to simulate payment completion rather than integrate with actual payment gateways. The system will not include SMS notification capabilities, limiting communication channels to email-based confirmations only.

Budget constraints restrict the use of premium third-party services or advanced hosting solutions, requiring the selection of cost-effective development and deployment tools. The system must operate within standard web browser limitations and cannot require specialized software installations on user devices.

**GLOSSARY**

**Terms and Definitions:**

**Admin User** - Authorized cinema staff member with administrative privileges to manage movies, showtimes, and system operations through the administrative dashboard.

**Audit Log** - Systematic record of all administrative actions performed within the system, including timestamps, user identification, and detailed action descriptions for accountability purposes.

**Booking Reference** - Unique alphanumeric identifier generated for each completed reservation, used for customer confirmation and administrative tracking purposes.

**Customer** - Individual user who accesses the system to browse movies, select showtimes, and complete ticket bookings without requiring account registration.

**Mock Payment** - Simulated payment processing functionality that mimics real transaction flow without actual monetary exchange or payment gateway integration.

**Race Condition** - Technical scenario where multiple users simultaneously attempt to select the same seats, potentially causing booking conflicts without proper real-time update mechanisms.

**Real-time Updates** - System capability to immediately reflect seat availability changes across all active user sessions to prevent booking conflicts and ensure accurate information display.

**Seat Map** - Visual representation of theater seating layout showing individual seat positions, availability status, and selection options for customer booking interface.

**Showtime** - Scheduled screening of a specific movie at a designated date, time, and theater location with defined seating capacity and availability.

**Token-based Authentication** - Security mechanism using generated tokens to verify administrative user identity and authorize access to protected system functions and API endpoints.

**REVISION HISTORY**

|  |  |
| --- | --- |
| Date | Description of Changes |
| August 26, 2025 | Initial creation of Requirements Specification Document with complete functional and non-functional requirements, use cases, and data specifications. |
| September 10, 2025 | Document formatting improvements and table structure refinements. |

Table 3. Document Revision History