**SM CINEMA BOOKING SYSTEM**

A Testing Documentation 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.

October 2025

**TESTING DOCUMENTATION**

**INTRODUCTION**

The testing phase was conducted to verify that the Web-Based Cinema Booking System for SM Cinema Grand Central functions correctly and meets the requirements outlined in the project proposal. This documentation presents the systematic approach taken to evaluate system functionality, identify defects, and confirm readiness for deployment.

The primary objectives of this testing process were to validate core booking functionalities, verify data integrity during reservations, assess system responsiveness across different devices, confirm administrative control features, and ensure the interface works properly on common browsers. Each objective was addressed through structured test scenarios designed to simulate real-world usage patterns.

The scope of testing encompasses all major system components deployed in the production environment. This includes the customer-facing interface for movie browsing and seat reservation, the administrative dashboard for schedule management and monitoring, the backend API handling business logic and data operations, database interactions through Supabase PostgreSQL, and media delivery through Cloudinary.

The testing does not cover payment gateway integration, SMS notification services, or mobile application features, as these were excluded from the project scope. Additionally, load testing beyond typical concurrent user scenarios and security penetration testing were not performed due to resource constraints.

**TESTING ENVIRONMENT**

**Hardware Specifications**

Testing was conducted on a personal computer with the following specifications:

* Processor: Intel Core i3 (8th Generation)
* Memory: 4GB DDR4 RAM
* Storage: 256GB SSD
* Display: 1366x768 resolution monitor
* Network: Stable WiFi connection with average download speed of 25 Mbps

Additional mobile device testing was performed using:

* Smartphone: Android device with 6.5-inch display
* Tablet: iPad with 10.2-inch display

**Software Requirements**

The following software tools and platforms were used throughout the testing process:

**Operating Systems:**

* Windows 10 (Primary testing environment)
* Android 12 (Mobile testing)
* iOS 15 (Tablet testing)

**Web Browsers:**

* Google Chrome Version 117
* Mozilla Firefox Version 118
* Microsoft Edge Version 117
* Safari Version 16 (iOS testing only)

**Development and Testing Tools:**

* Visual Studio Code for reviewing code and logs
* Browser Developer Tools for inspecting network requests and console logs

**Deployment Platforms:**

* Vercel (Frontend hosting)
* Render (Backend hosting)
* Supabase (Database management)
* Upstash Redis (Caching service)
* Cloudinary (Media storage)

**Test Data**

Sample movie data was created specifically for testing purposes. This included five fictional movie entries with varying genres, ratings, and showtimes spanning different dates and times. Theater layout data consisted of a standard configuration with 10 rows and 12 seats per row, totaling 120 seats per screening. Sample reservation data was generated to pre-populate certain screenings with existing bookings, allowing for testing of availability checks and concurrent booking situations.

**TESTING METHODOLOGY**

The testing approach combined multiple methodologies to provide thorough coverage of system functionality. Black-box testing formed the primary strategy, where the system was evaluated from an end-user perspective without requiring knowledge of the internal code structure. This approach focused on verifying that inputs produced expected outputs and that user interactions resulted in correct system behaviors.

Functional testing was conducted to verify that each feature performed according to specifications. This included testing individual components such as the movie listing page, seat selection interface, booking confirmation process, and administrative dashboard. Each function was tested in isolation before proceeding to integration testing that verified proper interaction between components.

Manual testing was employed throughout, as the project timeline and resources did not allow for automated testing framework implementation. Test cases were executed systematically, with results documented in structured formats for later analysis. No specialized testing frameworks were used beyond the browser's built-in developer tools.

Testing criteria were established based on functional requirements outlined in the project proposal. A test case was considered successful if it produced the expected output, maintained data integrity, displayed appropriate error messages when necessary, and completed within reasonable time frames. Failed test cases were logged with detailed reproduction steps for debugging purposes.

**TEST CASES**

The following table documents the systematic testing of core system functionalities. Each test case represents a specific scenario designed to verify expected system behavior.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Description | Test Steps | Expected Output | Actual Output | Status | Remarks |
| TC001 | View movie listings | 1.Navigate to homepage 2. Observe displayed movies | Movies display with posters and information | Movies displayed correctly | Pass | N/A |
| TC002 | Access movie details | 1. Click on a movie card  2. View details page | Movie information and showtimes appear | Details displayed properly | Pass | N/A |
| TC003 | Select showtime | 1. Navigate to movie details  2. Click on a showtime | Seat selection page loads | Redirected to seat selection correctly | Pass | N/A |
| TC004 | Mobile responsiveness | 1. Access from mobile device  2. Navigate booking processimplement caching strategies to reduce redundant requests | Layout adapts to screen size | Interface adjusted properly | Pass | N/A |
| TC005 | Admin login valid | 1. Go to admin URL  2. Enter correct credentials  3. Click login | Access granted to dashboard | Successfully logged in | Pass | N/A |
| TC006 | Admin login invalid | 1. Go to admin URL  2. Enter wrong password  3. Click login | Error message appears | Login denied with error | Pass | N/A |
| TC007 | Access admin without login | 1. Navigate directly to admin dashboard URL without logging in | Redirected to login page | Redirected to login successfully | Pass | Authentication required |
| TC008 | Access protected frontend routes | 1. Try to access admin-only frontend paths without authentication | Access denied or redirected | Prevented from accessing protected routes | Pass | Authentication hook working |
| TC009 | Add new movie | 1. Login to admin  2. Click "Add Movie"  3. Fill required fields  4. Save | Movie added to system | Movie appeared on homepage Pass | Pass | N/A |
| TC010 | Edit movie information | 1. Select movie in admin  2. Modify details  3. Save | Changes reflected on frontend | Updated information displayed | Pass | N/A |
| TC011 | Delete movie | 1. Select movie in admin  2. Delete entry  3. Confirm | Movie removed from system | Movie no longer visible | Pass | N/A |

Table 1. Systematic testing of core system functionalities.

**BUG TRACKING & ISSUE LOG**

The following issues were identified during testing and development. All reported bugs have been resolved and closed before final deployment.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bug ID | Description | Severity | Reported By | Status | Resolution |
| B001 | Inactive movies still displayed on homepage | High | Auldey Jerusalem | Closed | Updated movie card rendering logic to filter out inactive movies from homepage display |
| B002 | Scroll position not resetting when navigating between pages | Low | Auldey Jerusalem | Closed | Implemented scroll-to-top functionality on route changes for better user experience |
| B003 | Theme reverts to light mode on admin page refresh | Low | Auldey Jerusalem | Closed | Fixed theme persistence in admin pages by properly storing theme preference |
| B004 | Slow initial page load after inactivity | Medium | Auldey Jerusalem | Closed | Expected behavior on Render free tier; cold start delay documented as platform limitation |
| T B005 | Seat selection didn't register on slow connections | High | Auldey Jerusalem | Closed | Added loading indicator and improved click event handling for better responsiveness |
| B006 | Mobile keyboard covered form input fields | Medium | Auldey Jerusalem | Closed | Adjusted viewport settings and form positioning for mobile devices |
| B007 | Concurrent bookings briefly showed inconsistency | High | Auldey Jerusalem | Closed | Implemented database transaction locking to prevent race conditions |
| B008 | Movie poster displayed slowly on initial load | Low | Auldey Jerusalem | Closed | Configured Cloudinary optimization settings for faster image delivery |

Table 2. Bug Report and Resolution Summary

All identified bugs were addressed and resolved during the development and testing phases. The issues ranged from critical functionality problems, such as concurrent booking inconsistencies and inactive movie filtering, to minor user experience improvements like scroll behavior and theme persistence. The systematic approach to bug tracking and resolution ensured that the system achieved a stable state before final deployment. No critical or high-severity issues remain open, and the system is considered ready for operational use.

**USER ACCEPTANCE TESTING**

User acceptance testing focused on evaluating the system from the perspective of actual end users. Since this is an academic project, formal UAT with real cinema customers was not feasible. Instead, testing scenarios simulated typical user journeys and gathered feedback from peers and faculty advisors.

**Customer Booking Scenarios:**

Several test scenarios replicated common customer behaviors. The first scenario involved a user browsing available movies, selecting a film of interest, choosing an appropriate showtime, and completing a reservation. This end-to-end flow was tested multiple times with different movie selections and timeslots. Feedback indicated that the process was straightforward and required no external guidance. Users appreciated the clear confirmation message displayed after successful booking.

Another scenario tested users who changed their minds during booking. Testers selected seats, went to the booking form, navigated back to change selections, and completed the booking with different seats. The system handled this workflow correctly, maintaining a smooth experience throughout.

A third scenario examined how users responded to unavailable seats. Testers attempted to select reserved seats and observed the visual feedback. The gray coloring effectively communicated unavailability without requiring text explanations.

**Administrative Task Scenarios:**

Administrative testing involved common cinema management tasks. The first scenario covered adding a new movie to the system, including uploading a poster and creating showtimes. The admin interface handled this workflow efficiently, though feedback suggested bulk showtime creation would improve efficiency for movies with many screenings.

The second scenario tested updating existing movie information. This task was completed successfully through the admin panel, with changes reflected immediately on the frontend. Administrators found the interface straightforward for routine content updates. The third scenario involved reviewing booking activity and monitoring reservation status. The admin interface provided adequate functionality for tracking bookings, with reservation records displaying customer information and selected seats. This allowed staff to manage theater capacity effectively.

**Feedback Summary:**

Overall feedback from testers was positive. The customer interface was described as clean and easy to navigate. The seat selection visualization worked well, and color coding clearly distinguished between available, selected, and reserved seats. The on-screen confirmation after booking provided clear acknowledgment of successful reservations.

Suggested improvements included adding search functionality for movies, implementing a confirmation dialog before finalizing bookings to prevent accidental submissions, and providing a printable summary of booking details. The administrative interface was functional but basic, with suggestions for batch operations and more detailed reporting noted as potential future enhancements.

**CONCLUSION & RECOMMENDATIONS**

The testing phase confirmed that the Web-Based Cinema Booking System meets the core functional requirements established in the project proposal. All critical features performed as expected across multiple browsers and devices.

**Key Observations:**

The system successfully handles the booking workflow from start to finish. Users can browse movies, view details, select seats in real time, and complete reservations with on-screen confirmation. The interface responds well to user interactions and provides clear feedback throughout the process.

Database integrity remained stable throughout testing, even under scenarios involving concurrent seat selections. The system effectively prevents double-booking situations through proper transaction handling. Reservation data persisted correctly across sessions, and administrative modifications reflected immediately on the customer interface.

The deployment on cloud platforms proved reliable within the constraints of free-tier services. While cold start delays occasionally resulted in slower initial load times, subsequent requests performed acceptably. Cross-browser compatibility testing revealed no significant issues, and responsive design successfully adapted the interface for mobile devices.

The administrative dashboard provides sufficient functionality for basic cinema management tasks. Staff can add and modify movie schedules, upload promotional images, and monitor reservation activity through a centralized interface. While more advanced features could enhance administrative efficiency, the current implementation meets the project requirements for operational management.

**Recommendations for Future Improvements:**

Several enhancements could improve the system's functionality. Implementing payment processing would transform the system from a reservation platform to a complete booking solution. Adding user accounts would enable customers to view their booking history and manage multiple reservations over time.

Expanding notification capabilities would provide customers with better confirmation of their reservations through multiple channels. Enhanced reporting in the administrative dashboard would provide better insights into booking patterns, popular showtimes, and theater occupancy rates.

Adding support for multiple theater locations would make the system scalable for cinema chains operating across various venues. Implementing accessibility features such as screen reader support and keyboard navigation would make the system usable for a broader audience.

Performance optimization through advanced caching strategies and database query refinement would reduce response times during peak usage periods. Implementing automated system monitoring would help identify and address potential issues before they impact users.