# 7. Test Plan

## 7.1 Purpose

This section defines the verification and validation plan for the Movie Theater Ticketing System. The goal of the test plan is to ensure that all functional and non-functional requirements specified in this document are correctly implemented, reliable, and fulfill user expectations. The testing process will include unit, integration, and system testing, covering both functional workflows (ticket purchase, cancellation, scheduling) and non-functional qualities (performance, reliability, and security).

## 7.2 Verification Approach

The verification strategy focuses on three granularities of testing, as discussed in class:

1. Unit Testing:  
 - Scope: Individual classes and methods such as Seat, Order, and Payment.  
 - Purpose: Verify that internal logic functions correctly and handles invalid inputs or edge cases.  
 - Example focus: ensuring that Seat.updateAvailability() changes seat status correctly or that Payment.authorizePaymentInformation() rejects invalid payment data.

2. Integration Testing:  
 - Scope: Interactions between subsystems (e.g., Order & Checkout, Payment Orchestrator, Notification Service).  
 - Purpose: Ensure smooth communication between components, such as confirming that a successful payment triggers ticket creation and notification sending.  
 - Example focus: confirming that order status updates correctly propagate to the database and notification system.

3. System Testing:  
 - Scope: The complete application as deployed in the test environment.  
 - Purpose: Validate real-world user workflows across all modules, including UI, server, database, and external integrations.  
 - Example focus: end-to-end purchase of a ticket, cancellation within the refund policy, or manager scheduling a new screening.

## 7.3 Test Coverage

The following major system features will be covered through the test plan:

|  |  |  |
| --- | --- | --- |
| System Component | Feature / Behavior Tested | Test Type |
| Seat Management | Seat selection, hold, release logic | Unit |
| Order Management | Order creation, status transitions | Unit |
| Payment Processing | Payment authorization and refund handling | Unit / Integration |
| Notification System | Sending of email confirmations | Integration |
| User Roles & Policies | Access restrictions, policy enforcement | Integration |
| End-to-End Flow | Full purchase, cancel, reschedule | System |
| Security / Access | Unauthorized actions blocked | System |
| Performance | Seat hold timer, response time | System |

## 7.4 Responsibilities

|  |  |  |
| --- | --- | --- |
| Role | Member | Responsibility |
| Design Verification / Update | Matthew Gallenberger | Verify that design and diagrams reflect all modules tested; update if new classes are added. |
| Test Cases & Coverage | Ryan Schindler | Define and document 10 Excel test cases across all levels (unit, integration, system). |
| Test Plan & Repository Management | Nikola Milosavljevic | Write this Test Plan section, manage GitHub uploads, maintain timeline, ensure all members commit. |

Each team member must contribute at least one commit to the GitHub repository before submission to receive full credit.

## 7.5 Testing Tools and Environment

- Environment: Local web server with mock payment provider (sandbox mode).  
- Database: Azure SQL (test instance).  
- Testing Frameworks:  
 - Unit Testing: JUnit / PyTest  
 - Integration Testing: Postman, Mock APIs  
 - System Testing: Manual workflow validation via browser  
- Automation / CI: GitHub Actions for automatic testing on push events.

## 7.6 Test Deliverables

1. Test Plan Document (this section)  
2. Excel Test Case Sheet – testCasesSample.xlsx, containing:  
 - 2 Unit test cases  
 - 2 Integration test cases  
 - 2 System test cases  
 - 4 additional exploratory tests  
3. GitHub Repository – Updated with /docs/test-plan folder containing:  
 - TestPlan\_Assignment3.pdf  
 - testCases.xlsx  
 - Updated SRS\_v2.3.docx with this new section and GitHub link.

## 7.7 Testing Timeline

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Activity | Responsible | Deliverable |
| Oct 13–14 | Prepare and document test plan | Nikola | Test Plan draft |
| Oct 15–16 | Write and review Excel test cases | Ryan | Excel test sheet |
| Oct 17 | Verify architecture & diagram consistency | Matthew | Diagram verification |
| Oct 18 | Group review and merge into SRS | All | SRS\_v2.3 final version |
| Oct 19 | Push files to GitHub and validate commits | Nikola | Final GitHub folder link |

## 7.8 GitHub Repository Link

All testing materials will be available in the following repository under /docs/test-plan/:  
  
👉 https://github.com/KypleeFearz/CS250\_Group7  
  
Each team member must make at least one commit prior to submission. The link above should be included in Canvas as part of the official submission.

## 7.9 Approval

This section and its accompanying files form part of the SRS version 2.3 submission.  
Approved by:  
- Matthew Gallenberger  
- Ryan Schindler  
- Nikola Milosavljevic  
  
Date: October 19, 2025