

# PL3

## CINEMA PROJECT



## **1. Introduction**

This Testing Strategy document describes the overall approach, scope, objectives, and methodologies used to test the Cinema Application. The goal of this strategy is to ensure that the system functions correctly, meets user requirements, and maintains high quality in terms of reliability, performance, and maintainability.

---

## **2. Purpose of Testing Strategy**

- The purpose of this testing strategy is to:
    - Define how testing activities will be planned and executed
    - Ensure that all critical features of the Cinema Application are thoroughly tested
    - Reduce the risk of defects in production
    - Provide confidence in system quality before deployment
- 

## **3. Scope of Testing**

### ***3.1 In-Scope***

- The following components are included in the testing scope:
  - User Interface (UI)
  - Ticket booking and cancellation process
  - Business logic and validation rules
  - Database operations (read/write)
  - Configuration handling
  - Error handling and exception management

### ***3.2 Out-of-Scope***

- The following are not covered by this testing strategy:
  - Testing of third-party database engines
  - Operating system-level testing
  - Network infrastructure testing

---

## **4. Testing Levels**

### ***4.1 Unit Testing***

- Unit testing focuses on testing individual modules and functions in isolation.
- **Modules Covered:**
  - Domain models
  - Core business logic
  - Ticket system operations
- **Objective:**
  - Verify correctness of functions
  - Validate business rules
  - Detect defects early in development
- Unit tests are automated and executed frequently during development.

---

### ***4.2 Integration Testing***

- Integration testing verifies interactions between multiple modules.
- **Integration Areas:**
  - Core Logic with Database Layer
  - Ticket System with Core Logic
- **Objective:**
  - Ensure correct data flow between components
  - Identify interface defects
  - Validate error handling during integration

---

### ***4.3 System Testing***

- System testing evaluates the complete application as a whole.
- **Scenarios Tested:**
  - Successful ticket booking
  - Ticket cancellation

- Viewing available seats
  - Handling invalid inputs
  - **Objective:**
    - Validate end-to-end functionality
    - Ensure the system meets functional requirements
- 

#### *4.4 User Interface Testing*

- UI testing ensures that the system is user-friendly and behaves as expected.
  - **Focus Areas:**
    - Input validation
    - Navigation flow
    - Error and success messages
    - Overall usability
- 

### **5. Non-Functional Testing**

#### *5.1 Performance Testing*

- Performance testing checks the system response under normal and heavy loads.
  - **Objectives:**
    - Measure response time
    - Validate system stability during multiple requests
- 

#### *5.2 Security Testing*

- Security testing ensures that the system protects data and prevents unauthorized actions.
- **Focus Areas:**
  - Secure ticket access
  - Protection against invalid operations
  - Safe database interactions

---

### ***5.3 Reliability Testing***

- Reliability testing verifies that the system performs consistently over time.
  - **Objective:**
    - Ensure no data loss or corruption
    - Maintain stable behavior during repeated operations
- 

### **6. Test Data Management**

- Mock data is used for unit testing
  - A separate test database is used for integration and system testing
  - Production data is never used during testing
- 

### **7. Test Execution Strategy**

- Testing activities follow this sequence:
    1. Unit testing
    2. Integration testing
    3. System testing
    4. Regression testing after changes
  - Defects are logged, reviewed, fixed, and retested.
- 

### **8. Entry and Exit Criteria**

#### ***8.1 Entry Criteria***

- Application code is compiled successfully
- Test environment is ready
- Test cases are reviewed and approved

#### ***8.2 Exit Criteria***

- All critical test cases have passed
- No high-severity defects remain open
- Test results are documented

---

## **9. Conclusion**

- This testing strategy ensures that the Cinema Application is tested thoroughly and systematically. By applying multiple testing levels and focusing on both functional and non-functional requirements, the quality and reliability of the system can be confidently assured before deployment.