



SCHOOL OF COMPUTER SCIENCES

ACADEMIC SESSION: 2025/2026

CSE442 SOFTWARE TESTING

SOFTWARE TESTING PROJECT

MILESTONE 1: TEST PLAN REPORT

SMART EVENT AUTOMATED TICKETING (SEAT) SYSTEM

STUDENT NAME	MATRIC NUMBER
NOR SHAFIQA ATIRAH BINTI MOHD ZAIDI	164211
SARAH NUR DINIE BINTI MOHD AZUDDIN	161720
NUR ALIA FARISHA BINTI MOHD SOFIAN	163778
NUR RABIATUL ADAWIYAH HUSNA BINTI ABDUL HAFIZ	162081

SUBMISSION DATE:

27TH APRIL 2026 (WEEK 7) / MONDAY

TABLE OF CONTENTS

1.0 Introduction.....	3
1.1 Purpose.....	3
1.2 Scope.....	3
2.0 Test Items.....	4
2.1 Test Traceability Matrix.....	4
3.0 Features To Be Tested.....	6
4.0 Features Not To Be Tested.....	7
5.0 Test Approach.....	8
5.1 Test design techniques for white-box testing.....	8
5.2 Test design techniques for black-box testing.....	9
6.0 Item Pass/Fail Criteria.....	10
7.0 Entry Criteria.....	11
8.0 Exit Criteria.....	12

1.0 Introduction

1.1 Purpose

The primary purpose of this test plan is to establish a structured framework for validating the Smart Event Automated Ticketing (SEAT) system. This document supports the following objectives:

- 1) To define the testing procedures required to validate the Smart Event Automated Ticketing (SEAT) system across all development stages.
- 2) To ensure that all system functionalities such as event registration, payment processing, QR ticket generation, and attendance tracking operate correctly.
- 3) To determine the environment, tools, and testing techniques needed to conduct unit, integration and system testing.

1.2 Scope

The scope of testing covers the functional validation of internally developed components of the SEAT system, specifically testing will be conducted on the 18 core features outlined in the features list from **F001 to F018**. Testing will be performed at three levels namely **unit testing**, **integration testing**, and **system testing** to ensure that individual components, module interactions, and the complete system workflow operate as intended. This test plan adopts a combination of **white-box** and **black-box** testing approaches. White-box testing will be used to verify internal logic, control flow and code structure while black-box testing will focus on validating system functionality against specified requirements. The primary goal is to ensure the system satisfies its criteria and operates dependably under anticipated circumstances.

Exclusions: This test strategy excludes extensive non-functional testing like performance testing, advanced security, and infrastructure stress testing. Since they are outside the scope of validating the internally built system logic, these components are not included.

2.0 Test Items

The Smart Event Automated Ticketing (SEAT) system's test items include all system modules, features and supporting documentation that specify expected system behaviour.

The SEAT system Software Requirement Specification (SRS) is used as the main reference document for deriving test cases. This document determines both functional and non-functional requirements, providing as the baseline for verification and validation during the testing phase.

Furthermore, the test items also cover all key system features such as user administration, event handling, payment processing, ticket generation, notification services and reporting functionality. Each feature listed in Section 2.1 below is a test item that will be verified and validated through relevant testing levels, including Unit Testing, Integration Testing, and System Testing.

The testing technique ensures that all key modules function properly and accurately both independently and as an integrated system, as well as that they meet all the requirements that are stated in the SRS.

2.1 Test Traceability Matrix

Feature ID	Risk Level	Function Description	Source of Function
F001	High	System access and application initialization (web application startup)	SRS
F002	High	User registration for student, staff, and organizer account creation	SRS
F003	High	User authentication including login validation and session management	SRS
F004	Medium	User account management (admin activation or deactivation of user accounts)	SRS
F005	Medium	Update of user profile information	SRS

F006	Medium	Browse and filter event listings	SRS
F007	High	Event registration including form submission and validation	SRS
F008	High	Internal simulated payment processing workflow	SRS
F009	High	Transaction status update after internal payment completion	SRS
F010	High	QR-based digital ticket generation	SRS
F011	Medium	Retrieval of ticket details after successful registration	SRS
F012	Medium	Viewing and categorization of event tickets based on status (upcoming, cancelled, completed)	SRS
F013	Medium	Notification delivery (email confirmation, in-app announcements, and updates)	SRS
F014	Medium	Event creation functionality (organizer module)	SRS
F015	Medium	Event management including updating or deleting event details	SRS
F016	High	QR ticket scanning and attendance tracking	SRS
F017	Low	View analytics dashboard for event statistics and performance overview	SRS
F018	Low	Exporting reports in PDF and CSV formats	SRS

Table 1.0: *Test Traceability Matrix*

3.0 Features To Be Tested

Feature ID	Description	Test Level
F001	System access and application initialization (web application startup)	System
F002	User registration for student, staff, and organizer account creation	Unit, Integration
F003	User authentication including login validation and session management	Unit, Integration
F004	User account management (admin activation or deactivation of user accounts)	System
F005	Update of user profile information	Unit
F006	Browse and filter event listings	System
F007	Event registration including form submission and validation	Integration
F008	Internal simulated payment processing workflow	Integration, System
F009	Transaction status update after internal payment completion	Integration
F010	QR-based digital ticket generation	Integration
F011	Retrieval of ticket details after successful registration	System
F012	Viewing and categorization of event tickets based on status (upcoming, cancelled, completed)	System
F013	Notification delivery (email confirmation, in-app announcements, and updates)	Integration
F014	Event creation functionality (organizer module)	Unit, Integration

F015	Event management including updating or deleting event details	System
F016	QR ticket scanning and attendance tracking	Integration, System
F017	View analytics dashboard for event statistics and performance overview	System
F018	Exporting reports in PDF and CSV formats	System

Table 2.0 : Features To be Tested

4.0 Features Not To Be Tested

Aside from the features listed in Section 2.0 , other aspects of the SEAT system are outside the scope of this test plan. This test plan focuses on functional and system-level validation of internally developed components.

This includes:

I. Email (SMTP) service infrastructure

The system supports email notifications such as registration confirmation and password reset via SMTP services. However, email delivery performance and server reliability are not evaluated.

II. QR API server

The system integrates a QR API server for QR code generation. This test plan only verifies API communication and QR code retrieval, while external server performance is not evaluated.

III. Hosting and server infrastructure

Server performance, uptime, hardware specifications, and hosting configuration are not included in this test plan as they are managed externally.

IV. Network conditions and connectivity

External factors such as internet speed, network stability, and user connectivity are not within the scope of this test plan.

V. **Advanced security testing**

Penetration testing, vulnerability assessment, and other advanced security evaluations are not included, as this test plan focuses on functional validation.

5.0 Test Approach

In order to guarantee both code-level accuracy and requirement fulfillment, the Smart Event Automated Ticketing (SEAT) system is tested using a multi-level approach. White-box and black-box testing methods will be combined to test the strategy.

5.1 Test design techniques for white-box testing

White-box testing takes place at the unit and integration level to verify that the PHP and JavaScript components' internal logic is acceptable. The testing focuses on making sure that all potential paths are evaluated and that the computer code runs appropriately.

The following white-box testing techniques will be used:

- I. **Statement Coverage:** Ensuring every line of code in critical modules such as processPayment.php and generateTicket.php is executed at least once to identify unreachable or dead code.
- II. **Branch Coverage:** Testing every decision point in the code, such as if-else statements in login authentication (F003), ensuring both True and False branches are exercised to achieve 100% branch coverage.
- III. **Path Coverage:** Tests selected independent execution paths in critical modules, especially user authentication (F003), payment processing (F008), and transaction status updates (F009), to detect hidden logical errors.

5.2 Test design techniques for black-box testing

Black-box testing is carried out at the system and integration levels to verify the SEAT system's functioning based on specifications without taking internal code structure into account.

The following black-box techniques will be used:

- I. **Equivalence Partitioning:** Dividing input data into valid and invalid partitions. For example, testing event capacity with valid integers vs. negative numbers or text.
- II. **Boundary Value Analysis:** Testing the limits of input fields. For example, testing the character limits for event titles or the minimum/maximum price for a paid ticket.
- III. **Decision Table Testing:** Used for complex logic like F012 (Ticket Categorization), where multiple conditions including event date passed, payment status and user attendance will determine the ticket status.
- IV. **Use Case Testing:** Executing end-to-end scenarios based on the "Organizer", "Staff" and "Student" roles, such as the full flow from event creation to attendee check-in (F014 to F016).
- V. **Error Guessing:** Based on common web vulnerabilities, testing for improper form submissions or unauthorized access to the Admin Dashboard (F004).

6.0 Item Pass/Fail Criteria

The Smart Event Automated Ticketing (SEAT) system must fulfill the following user and system requirements for each test item to be labelled PASS:

1. All test cases related with each functionality (F001-F018) must be conducted and successfully completed and passed without any critical faults/defects.
2. All features must comply with the requirements specified in the Software Requirements Specification (SRS).
3. System access and authentication (F001-F003) must operate appropriately and correctly with valid inputs and securely reject the invalid or unauthorized access attempts.
4. All user-related features (F002, F004, F005) must create, update and manage user data accurately to prevent data loss or inconsistency.
5. Event-related functionalities (F006, F007, F014, F015) must display, create, update and handle the event information correctly with proper validation.
6. Online payment processing and transaction validation must be completed securely and successfully with a 0% mistake rate of monetary calculations, and all completed and successful transactions must be appropriately documented in the system.
7. The digital ticket generation must be created with a valid and unique QR code for each successful registration and the ticket must be easily retrieved by the user.
8. For the ticket viewing and classification (F011 & F012) must display correct and up-to-date ticket information.
9. Notification services must pass on email confirmations, reminders or updates without faults or duplication.
10. QR code scanning and attendance tracking should update the attendance data in real-time, without delay or redundancy.
11. The analytics and reporting tools must generate reliable data and export files (PDF/CSV) without missing information or file corruption.
12. The system must not encounter critical system failures for example, system crashes, deadlocks or data loss while in operation.
13. System response time must be within acceptable limits.
14. Test item will be considered FAILED if:
 - a. One or more crucial test cases failed.
 - b. The output does not match with the expected results specified in SRS.

- c. Improper processing, system failures occur or data inconsistency.
- d. Payment, security or core features are compromised

In summary, all test items must completely comply with the SRS requirements before the SEAT system to be accepted.

7.0 Entry Criteria

The following conditions must be met before the testing phase for the Smart Event Automated Ticketing (SEAT) system can commence:

- I. **System Requirements:** All the functional and non-functional requirements are clearly defined, consistent and approved.
- II. **Development Completion:** The source code for all the modules in the SEAT system must be fully developed and integrated. The database in the system, which uses MYSQL as the database, must be configured with all necessary tables, such as the User table, Registration table, Ticket table, etc., and must be accessible. No critical syntax or runtime errors exist, and the developer has completed basic code review.
- III. **System Deployment and Environment Readiness:** The frontend and backend code must be successfully connected and fully deployed. This includes the setups of the local server environment, PHP Backend functionality, and the MYSQL database.
- IV. **Test Documentation Approval:** All white-box and black-box test cases targeting unit, integration and system levels must be clearly defined, reviewed and approved. Test cases for Unit, Integration, and System testing are documented, expected results and pass/fail criteria are clearly defined
- V. **Test Data Availability:** Sufficient mock data must be populated within the database. This includes mock data such as student accounts, organiser profiles and event details that must be created and ready to be used.
- VI. **Resource and Availability:** The testing team members are assigned and available. The testing schedule and milestones are defined and agreed upon. Also, all necessary documents are accessible.

8.0 Exit Criteria

The testing phase for Smart Event Automated Ticket (SEAT) system is completed and ready for closure only when the following conditions have been met:

- I. **Test Execution Completion:** All planned test cases in the test plan have been executed and all test results are properly recorded and documented in the final report.
- II. **Defect Resolution:** All major and critical defects have been fixed and successfully retested or formally accepted by the stakeholders if unresolved. Specifically, there must be zero critical failures in secure user authentication, online payment processing, QR code/access link ticket generation, and real-time attendance updates. Meanwhile, the remaining defects are low severity and do not affect the core system functionality have to be recorded and documented as accepted risks.
- III. **System Functionality:** System meets the defined functional and non-functional requirements. The end-to-end system workflows work smoothly from start to finish and must operate without failure
- IV. **Reporting and Deliverables:** All test logs, failure reporting, defect analysis and corrective action taken must be fully compiled, documented, reviewed and finalised.
- V. **Stakeholders Approval:** Testing results are reviewed and approved by the stakeholders and the system meets the defined acceptance criteria.