BOOK STORE PROJECT

TEST PLAN DOCUMENT

## **Document Approval History**

|  |  |  |
| --- | --- | --- |
| Prepared by | Nguyen Huu Hoang Kiet |  |

## **Document Revision Histories**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Date*** | ***Version*** | ***Description*** | ***Author*** |
| 23/03/2025 | v.1.0 | * Init the test plan | Kiet |
| 05/04/2025 | v.1.1 | * Update the schedule and estimation for test plan for Book Store Project | Kiet |
| 14/04/2025 | V.1.2 | * Finalize test scope and testing types | Kiet |

# **Table of Contents**

[Document Approval History 1](#_8hyfeafkbznt)

[Document Revision Histories 1](#_3znysh7)

[**Table of Contents 2**](#_astnvu7f249w)

[**1. Overview 2**](#_147n2zr)

[**2. Test Item(Phase 1) 2**](#_b5lp0zycj3on)

[**3. Test Approach 4**](#_m3qsaajea0yi)

[3.1. Define Scope of Testing 4](#_go2i10iv6asn)

[3.1.1. In the scope 4](#_93cwz8pvgg0s)

[3.1.2. Out of scope 5](#_5au6ajrfjkxt)

[3.2. Identify Testing Type: 5](#_6obp05p5mib7)

[3.2.1. System Testing 5](#_1d0fhbqfx5xc)

[3.2.1.1. Functional Testing 5](#_nhtjp97hpdb7)

[3.2.1.2. Regression Testing 5](#_expwbv5wclz4)

[3.2.1.3. Exploratory Testing 6](#_3cw7upb8a0yc)

[3.2.1.4. Usability Testing 6](#_mshcs4ec7cd8)

[3.2.1.5. Installation Testing 6](#_ldgpc83etmyp)

[3.2.1.6. Deployment Testing 6](#_vezcyh4ryjy9)

[3.2.1.7. User acceptance testing 7](#_bi1p9kxwc803)

[3.2.1.8. Compatibility testing 7](#_kvvrt8ovx4n6)

[3.2.1.9. Smoke Testing 7](#_sbiltfsy5eqm)

[- This testing type involves testing the critical functionality of the application to ensure that it's stable and ready for further testing. 7](#_l0bkj0f1ft7t)

[3.2.2. API Testing 7](#_1dpa8nuakwks)

[3.2.3. Load Testing 7](#_s2izjejqtjdv)

[3.2.4. End to End testing 7](#_mwjhgmaq9076)

[3.2.5. Security testing 8](#_mwjhgmaq9076)

[3.3. Identify the Testing Type in each environment: 8](#_d9oid7sqw8nq)

[5. Exploratory Testing 8](#_4rd5jo2ytl0z)

[8. API Testing 8](#_pfyetgc252h9)

[Integration Testing 8](#_pfyetgc252h9)

[Integration Testing 8](#_apgscrohgru5)

[Integration Testing 9](#_rix32k4rg9vo)

[**4. Test Objective 9**](#_ola5cud603st)

[**5. Test Strategy 9**](#_ge2rjqkzz35m)

[5.1. Test Procedure Description 9](#_j2sim626f4ex)

[5.2. Test Case 11](#_yadgt0k928ui)

[5.3. Test Reporting 11](#_ew3eipif5eul)

[5.4. Defect Classification 11](#_k7okwelmivv9)

[**6. Testing Tools 12**](#_gpdwwf5jpg4y)

[**7. Platforms and Browsers Testing 12**](#_ls2oqyw527sy)

[7.1. Platform Compatibility Testing (for mobile app) 12](#_fhxjbp6ra0q1)

7.2. Browser Compatibility Testing (for Ecommerce Web) 12

[7.3. Screen size and Devices: 12](#_evk5zis5eux)

[**8. Test Criteria 13**](#_12sgpsj9l4ni)

[8.1. Entry & Exit Criteria 13](#_3as4poj)

[8.1.1.Entry Criteria 13](#_6i777caijjvl)

[8.1.2.Exit Criteria 13](#_l6xdctee7crt)

[8.2.Suspension Criteria & Resumption Requirements 13](#_1pxezwc)

[8.3. Acceptance Criteria 14](#_nujeh93xoupp)

[**9. Testing Task & Estimation 14**](#_nl4ck92ijc8q)

[**10. Resource Planning 15**](#_cpt8bz24yddo)

[**11. Acknowledgement 15**](#_23ckvvd)

# 

# 

# **1. Overview**

* This document outlines the testing approach to validate the acceptability of the **Book Store Project** software prior to production deployment. It details the testing scope, strategies, tools, tasks, and resources required to ensure the system meets functional and non-functional requirements.
* Reference the documentation: the following are the references for this test plan:

|  |  |
| --- | --- |
| **The documented purpose** | **The link** |
| Test Case | CR236 "Export all carrier choices" |
| System Design Specification | Internal Document |
| API | Internal Document |

# **2. Test Item**

|  |  |
| --- | --- |
| **Site** | **Module / Function** |
| Web App | **Authentication & Authorization**: Login, Role-based access, Logout. **Book Management**: Add, Update, Delete, Search Books. **Inventory Management**: Update Book Quantity. **Ordering**: Add to Cart, Process Orders, Payment Methods. **Reports**: Revenue and Frequency Report. **Rules Management**: Update System Rules. |
| **API** | Authentication Book Management  Inventory Ordering Reports Rules Management |

# **3. Test Approach**

## **3.1. Define Scope of Testing**

### **3.1.1. In the scope**

* Functional Testing: Validate all modules (auth, book management, inventory, ordering, reports, rules management) against requirements.
* Integration Testing: Ensure seamless interaction between modules and APIs.
* API Testing: Verify API endpoints for correct responses, error handling, and performance.
* System Testing: Test the entire system end-to-end to ensure business flows work as expected.
* Load Testing (web and app). For Performance testing, based on capacity sizing defined by the customer, the system will be set up, run, checked, and review script load test with different scenarios. E.g average number of transactions per hour during peak hours: 2000-2500 transactions per hour → The application need to handle at least this threshold
* Security Testing: Validate authentication, authorization, and data protection mechanisms.
* Usability Testing: Ensure the web application is user-friendly and intuitive.
* Regression Testing: Re-test after defect fixes to ensure no new issues are introduced.

### **3.1.2. Out of scope**

* Performance testing (e.g., load and stress testing) is not included in this phase.
* Web application testing.
* Localization and internationalization testing.

## **3.2. Identify Testing Type:**

### **3.2.1. System Testing**

* System Testing is testing that is performed on a complete system to verify if it works as expected once all the modules or components are integrated.

#### **3.2.1.1. Functional Testing**

* Functional Testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input and verifying the output against the Functional requirements.

#### 

#### **3.2.1.2. Regression Testing**

* Re-testing after fixes or modifications of the software or its environment.
* Regression Tests will occur at least once throughout the testing phase to ensure that changes don’t impact any functionality that was already validated.
* This is to ensure that end-to-end functions and the business flow of the system are working as expected.

#### **3.2.1.3. Exploratory Testing**

* This type of testing aims to cover scenarios that are not confined within the requirements defined but rather on the tester’s exploratory skills to navigate around the system.
* This also includes **Negative Testing** wherein test scenarios written are beyond the functionalities specified in the Requirements Specification documents to ensure that the system doesn’t do what it is NOT supposed to do.

#### **3.2.1.4. Usability Testing**

* Usability Testing - Testing the usability of the application, mainly focusing on testing the ease of use, flexibility, and friendliness of the product.

#### **3.2.1.5. User acceptance testing**

* User acceptance testing (UAT) is a type of software testing that is performed to ensure that a software application meets the requirements of its intended users. UAT is typically conducted in the final stages of the software development life cycle, and it involves testing the application in a simulated or real-world environment to verify its functionality, usability, and overall quality.
* UAT is performed by end-users, stakeholders, or a designated testing team who are representative of the intended users of the application. The purpose of UAT is to identify any issues or errors that may have been missed during the development and testing phases of the software application. UAT helps to ensure that the application meets the needs and expectations of the end-users and stakeholders, and it helps to increase user satisfaction and adoption of the application.

#### **3.2.1.6. Integration Testing**

* Integration testing is a type of software testing that focuses on verifying that different components or modules of an application work together correctly. The purpose of integration testing is to detect and resolve issues that may arise when different parts of the system are integrated and interact with each other.

### **3.2.2. API Testing**

* API testing is used to determine whether APIs return the correct response (in the expected format) for a broad range of feasible requests, react properly to [edge cases](https://en.wikipedia.org/wiki/Edge_cases) such as failures and unexpected/extreme inputs, and deliver responses in an [acceptable amount of time](https://en.wikipedia.org/wiki/Service-level_agreement).

### **3.2.3. Load Testing**

* Testing is performed to determine system's behavior under both normal and anticipated peak load conditions. It helps to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation.

### **3.2.4. Stress Testing**

* Stress testing is a type of software testing that involves evaluating the performance of a system or application under extreme conditions. The goal of stress testing is to identify the maximum capacity of the system or application, and to determine how it performs under heavy load or adverse conditions.

### **3.2.5. End to End testing**

* End to End (E2E) testing is a software testing method to test the application flow from start to finish. It is implemented end-to-end in real-life scenarios such as the application's communication with hardware, networks, databases, API and other applications.

### **3.2.6. Security testing**

* Security Testing is a type of Software Testing that uncovers vulnerabilities, threats, risks in a software application and prevents malicious attacks from intruders. The purpose of Security Tests is to identify all possible loopholes and weaknesses of the software system which might result in a loss of information, revenue, repute at the hands of the employees or outsiders of the Organization.

# **4. Test Objective**

* Verify that the Book Store Project’s **functionality** works as expected without errors in a real-world environment.
* Ensure external interfaces (e.g., UI, APIs) meet user and technical requirements.
* Validate the **usability** and **reliability** of the web application.
* Achieve 100% test **coverage** for critical functionalities and at least 90% overall test case pass rate.

# **5. Test Strategy**

### **5.1. Test Procedure Description**

|  |  |
| --- | --- |
| **Procedure** | **Description/Definition** |
| 1. **Write Test Case** | Create test cases based on requirements (e.g., CR236). Cover functional, integration, and edge cases.   * End to end test cases * Core Products related test cases |
| 1. **Test Case Review** | QC Lead reviews test cases, provides feedback, and ensures alignment with requirements. |
| 1. **Prepare Test Suite** | Consolidate test cases into suites for execution, organized by module. |
| 1. **Prepare Test Environment** | QC Team prior to test execution shall prepare the following:   * Prepare test environment and test ware * Obtain test suite / assigned test cases, and any needed reference documents / configuration guides * Set up test tracking processes * Set logging and archiving processes * Set up or obtain test input data. |
| 1. **Perform Tests** | Execute tests on the software by following what is stated in the Test Plan and Test Case. |
| 1. **Evaluate Results** | Evaluate and record results of each test on the provided Pass/Fail/Remarks column of the Test Case. |
| 1. **Log Test Activities and Defects** | Log testing activities, defects, and corrections using the appropriate forms  **Determine Severity of Defects**  Confirm severity of defects by referring to the Severity Levels as discussed in the **“Defect Classification”** section. |
| **8. Assign a Developer to Fix Defects** | QC Lead and QC member shall assign fixing of the found defects to the designated developer by specifying on the Defects Tracking System used. |
| **9.Assign a Developer to Fix Defects** | The developer assigned fixes the problem. |
| **10. Test Reporting** | QC Lead updates the reportfor the test executed and evaluation gathered. |

### **5.2. Test Case**

Test Case scenarios will be created using Microsoft Excel

### **5.3. Test Reporting**

Test Reports will be created using Microsoft Excel with the following data:

1. Throughout the duration of testing - Test Progress, Test Results, and Defect Summary Report will be produced daily. Several exceptions when:

* 4-Critical, and 3-High Severity defects are uncovered during testing
* Environment related issues that will impact testing schedule
* Any other external/internal incidence that will impact testing schedule

For these instances above, immediate reporting and escalation will be made by the QC Lead to PM, Dev and any other stakeholders.

1. Completion of System Test Phase - System Test Report will be produced at the end of the testing. It will contain details of the test runs and defects.
   * It records the summary of tests, test outcome, problems encountered and the appropriate follow up actions that were taken during the System Acceptance Test.
   * It serves to inform management on the successful completion of the System Test.

### **5.4. Defect Classification**

Defects found during the testing phase will be classified as follows

* 4 - Critical

This is the most critical and it must be the testing team’s top priority to ensure that there shouldn’t be any bug of this severity in the product. Example: some error leads to the system to shut down.

* 3 - High

This level is not as critical compared to the higher level of severity but occurrence of this type of bugs will have great effect on the overall functionality of the system.

* 2 - Medium

These are defects that affect a particular functionality for which there are existing alternatives for bypassing the problem.

* 1 - Low

May be considered as a “cosmetic” severity level in which the primary concern is the cosmetic appearance of the system to the user. May also include suggestions, or recommendations for better implementation (which does not concern functionality).

# **6. Testing Tools**

* 100% of testing tasks will be done manually using ***MS Office*** for test case writing and development, test status reporting, defect reporting, and test reporting.
* **Pytest** and **Pytest-BDD** for automated unit and behavior-driven tests.
* Defects will be logged and tracked using **Taiga**
* SBSG APIs will be executed using ***Postman.***

# **7. Platforms and Browsers Testing**

## **7.1. Platform Compatibility Testing**

|  |  |
| --- | --- |
| **Platform** | **Module** |
| Web | All web app features |

## **7.2. Browser Compatibility Testing (for Ecommerce Web)**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Device/Platform** | **Browser** | **Module** |
| 1 | Windows | Chrome | All Member Portal Web (desktop) features |
| 2 | Mobile/Android | Chrome | All Member Portal Web (mobile) features |
| 3 | Mobile/iOS | Safari | All Member Portal Web (mobile) features |
| 4 | Windows | Firefox | All Member Portal Web (desktop) features |
| 5 | MacOS | Safari | All Member Portal Web (desktop) features |
| 6 | MacOS | Chrome | All Member Portal Web (desktop) features |
| 7 | Mobile/Android | Chrome | All Member Portal Web (mobile) features |
| 8 | Mobile/iOS | Chrome | All Member Portal Web (mobile) features |

## **7.3. Screen size and Devices:**

* iOS 11 - > iOS 15
* 375\*667 (iP6, iP7 IOS 14.7.1)
* 414\*736 (iP6+, ip7+ IOS 15.2)
* 375\*812 (iPhone X, Xs)
* 414\*896 (iP XS Max isOS 15.6)
* 360\*780 (iPhone 12 mini iOS 15)
* 2048\*1536(iPad 5th generation IOS 15.5)
* Android: 6.0 - > 14
* 412\*846 (SS Note8)
* 360\*780 (Vivo Pro Android 11)
* 414×896 (Oppo A55) - Android 11
* (Redmi Note 11) - Android 11
* 412x846 ( A03 Samsung ) - Android 11
* 720\*1612 (Oppo A57)
* 385\*854 (Sam Sung A03s)
* 384\*857 (Sam Sung A13)

# **8. Test Criteria**

### **8.1. Entry & Exit Criteria**

#### **8.1.1. Entry Criteria**

Below are the required items that need to be furnished to commence the activities for the System Testing.

* Beta release of the Book Store Project.
* Signed-off requirements (CR236).
* Signed-off system design document.
* Test environment setup completed.

#### **8.1.2. Exit Criteria**

These criteria listed will conclude the completion of System Testing:

* Book Store Project (Official Release)
* Test Plan (Updated)
* Test Case (Executed 100% coverage)
* Test Report (98%+ pass rate 55/56 passed)

### **8.2. Suspension Criteria & Resumption Requirements**

The system testing shall be suspended with the following criteria:

* 4-Critical severity defect is reported during testing.
* 3-High severity defects are reported impacting testing of more than 50% of the test cases scheduled for test execution.

Testing will be resumed when the Sev 3 and 4 defects are fixed and retested to be working.

## 

### **8.3. Acceptance Criteria**

The system shall be deemed to have passed System Testing if the following conditions are satisfied:

* 100% of the test cases are executed.
* At least 90% of the test cases passed, i.e. achieved 98.21% per test-case.xlsx.
* All 4-Critical, and 3-High severity defects are resolved.
* At least 70% of 2-Medium and 1-Low severity defects are resolved. Any unresolved defects are acceptable to the users, marked as deferred and documented.

# **9. Testing Task & Estimation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Task** | **Work Days** | **Schedule** | | **Responsible** |
|  |  |  | **From** | **To** |  |
| **1** | **System Test Planning** | 10 | 20 Mar | 10 Apr | **QC Lead** |
| **2** | **Test Case Writing** | **30** | **15 Mar** | 14 Apr | **QC Team** |
| **2.1** | **Authentication & Authorization** | 8 | 23 Mar | 30 Mar | QC Team |
| **2.2** | **Book Management** | 10 | 31 Apr | 09 Apr | QC Team |
| **2.3** | **Inventory, Ordering, Reports, Rules** | 5 | 09 Apr | 13 Apr | QC Lead |
| **3** | **Test Execution – System Testing** | 26 | 13 Mar | 02 Jun | **QC Team** |
| **3.1** | **Authentication & Authorization** | 8 | 23 Mar | 30 Mar | QC Team |
| **3.2** | **Book Management** | 10 | 31 Mar | 09 Mar | QC Team |
| **3.3** | **Inventory, Ordering, Reports, Rules** | 5 | 09 Apr | 13 Apr | QC Team |
| **4** | **System Test Reporting** | **1** | **14 Apr** | **14 Apr** | **QC Lead** |
| **5** | **UAT** | **1** | **14 Apr** | **14 Apr** | **SBSG** |
| **6** | **Prod testing** | **1** | **14 Apr** | **14 Apr** | QC Team |

# **10. Resource Planning**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibility** |
| Hoang Kiet | QC Lead (App + Web) | - Manage testing, review requirements, define test strategy, and allocate resources.  - Execute tests, log results, and report defects for auth and book management.  - Execute tests, log results, and report defects for inventory and ordering.  - Execute tests, log results, and report defects for reports and rules management. |

# **11. Acknowledgement**

|  |  |  |
| --- | --- | --- |
| **CLIENT APPROVAL** | | |
| **Approved By:** | **Position:** | **Date:** |
| *[Pls. sign over printed name]* |  |  |