Test Plan Template

Prepared by:

- Sahil Naik
- Sohan Chidrawar
- Supreet Birajdar

(27-10-2023)

TABLE OF CONTENTS

1.0 Introduction

2.0 Objectives and Tasks

- 2.1 Objectives
- 2.2 Tasks

3.0 Scope

4.0 Testing Strategy

- 4.1 Alpha Testing (Unit Testing)
- 4.2 System and Integration Testing
- 4.3 Performance and Stress Testing
- 4.4 User Acceptance Testing
- 4.5 Automated Regression Testing
- 4.6 Beta Testing
- 4.7 Compatibility Testing

5.0 Hardware / Software Requirements

6.0 Test Schedule

7.0 Features to Be Tested

8.0 Features Not to Be Tested

9.0 Resources/Roles & Responsibilities

1.0 INTRODUCTION

This documentation is the review of the platform-based application which was intentionally developed for Windows Users to add books or manage their books records with Author and ISBN number.

1.1 Product Overview

The core product that we present is a system which is used to manage the books along with their title, author name, ISBN and other detailed related to the books. The product is focused to provide ability to add books, update books and even delete books as per relevance of the user.

1.2 Core Functions at a High Level

Our blog site encompasses the following core functions

- Book addition: The ability for individuals to add their choice of book along with the book details that they see fit.
- Book updating: A way for the user to individually update the existing book record and details regarding it.
- Book deletion: Gives the ability for the user to delete the record related to any existing book at any time.

2.0 OBJECTIVES AND TASKS

2.1 Objectives of Test Plan

- Quality Assurance: To ensure the software or application meets the defined quality standards and requirements.
- Functional Validation: To validate that the software functions correctly and fulfills its intended purpose.
- Defect Identification: To identify and document defects, issues, or deviations from expected behavior.
- Risk Mitigation: To mitigate and manage risks associated with software quality and performance.
- Verification and Validation: To verify that the software aligns with requirements and validate that it meets user expectations.

2.2 Tasks of Test Plan

- Testing user registration functionality, including successful book addition and error handling.
- Testing book viewing functionality, including book searches.
- Testing the landing page, including content displayed and navigation.
- Writing and executing test cases and test scripts.
- Reporting and documenting defects, if any.
- Performing regression testing when updates or changes are made.
- Preparing and maintaining test environments.

3.0 SCOPE General

Sr. No.	#Business Requirements (BR)	#Functional Requirements (FR)	#Test Scenarios (TS)	#Testing Approaches /Strategies (TA)
01	Book addition	Book addition form	Successful book addition	**1.Functional Testing:** Verify that a user can add books successfully by providing valid information. **2.Non-Functional Testing:** Test insertion performance under a load to ensure it meets response time requirements.
		Book data validation	Duplicate book data prevention	**1. Functional Testing:** Ensure that book with the same author's name or ISBN cannot be inserted twice. **2.Non-Functional Testing:** Test insertion capacity to ensure it can handle a specified number of concurrent insertions.
02	Search Book	Book search form	Successful book search	**1.Functional Testing:** Verify that users can search book with valid data. **2.Non-Functional Testing:** Test search performance to ensure quick response times.
		Search mechanism	Failed search attempts	**1.Functional Testing:** Test the system's response to multiple failed search attempts. **2.Non-Functional Testing:** Measure the system's response time under a high number of concurrent failed search attempts.
03	Landing Page	Display of curated content	Content loading	**1.Functional Testing:** Ensure that content loads correctly on the home page. **2.Non-Functional Testing:** Test the home page's response time for content retrieval.

This section outlines the scope of our testing efforts, providing a comprehensive overview of what is being tested within our platform-based book management app. It encompasses all the functions, existing interfaces, and the integration of these functions. Our scope extends to ensuring the quality and reliability of book registration, updating, and the book viewing page. To achieve this, we will employ a systematic and structured approach, as outlined below:

Book addition:

Testing the ability of user to add new books, including the creation of new book credentials and viewing them.
Verifying user is able to search and view the books using valid book data.
Examining the prevention of duplicate book data during addition.
Evaluating necessary data provided for adding book
Functional Testing: Test each book addition scenario for correctness.
Non-Functional Testing: Measure performance and capacity under insertion load.

Book searching:

Testing the user ability to able to search the books with valid book data.
Functional Testing: Confirm that existing books can be viewed by user by entering valid book data
Non-Functional Testing: Assess search performance, particularly under load conditions.

Book Viewing Page:

Testing the display of books content on the landing page.
Verifying the functionality of the navigation menu.
Evaluating the effectiveness of the search functionality.
Functional Testing: Confirm the correct display of content and navigation.
Non-Functional Testing: Measure the responsiveness and performance of content loading and search results.

4.0 TESTING STRATEGY

In this section, we outline the overall approach to testing, specifying the approach for each major group of features within our platform-based book management app. We detail the major activities, techniques, and tools that will ensure comprehensive testing of the designated feature groups. The testing approach is described in sufficient detail to identify the key testing tasks and estimate the time required for each one.

Book insertion:

- Approach: We will adopt a structured approach to validate book insertion, covering both functional and non-functional aspects.
- Major Activities:
  Test data validation, including ISBN formats, Author's name, and title uniqueness.
  Verify book insertion form submission.
  Ensure that duplicate book data cannot be registered.

Book searching:

- Approach: We will take a methodical approach to validate book insertion functionality, encompassing both successful and failed book addition attempts.
- Major Activities:
  Test successful book insertion with correct details.
  Assess the response to failed book details attempts.

Landing Page:

- Approach: We will employ a systematic approach to validate the landing page, ensuring its content, navigation functionalities are thoroughly tested.
- Major Activities:
  Test the correct display of curated content.
  Measure the responsiveness and performance of content loading

Approach	Type of Testing	Manual Testing	Automated Testing	Tools/APIs/Libraries
Standard Testing (Functional Testing)	Unit Testing	Yes	Yes	Python-UnitTest (Software/IDE based testing package provided within the Python Environment)
	Integration Testing	Yes	Yes	
	System Testing	Yes	Yes	
	Regression Testing	No	Yes	
	Acceptance Testing	No	No	

4.1 Unit Testing

Definition:

Unit testing is the initial phase of the testing process, focusing on isolating and verifying the correctness of individual program components or units. The minimum degree of comprehensiveness desired is to ensure that each unit is tested thoroughly to identify and rectify any defects. The comprehensiveness of unit testing will be judged by measuring code coverage, ensuring that every statement and branch condition is executed at least once. Completion criteria include a low error frequency, where the number of defects detected should be minimized.

Participants:

- Developers : Supreet Birajdar, Sahil Naik, Sohan Chidrawar.
- Quality Assurance (QA) team members responsible for reviewing and validating unit test scripts.

Methodology:

- **Test Script Development:** Developers will be responsible for writing unit test scripts for their respective code units. These test scripts will be written to validate the correctness of the individual code components.
- **Test Execution:** Unit tests will be executed as a part of the development process. Whenever a code unit is developed or modified, the corresponding unit tests will be executed immediately to detect and address defects promptly.
- **Code Coverage Analysis:** Code coverage tools will be used to track the execution of code statements and branch conditions. The goal is to ensure that every part of the code is exercised at least once.

- **Defect Identification and Resolution:** If any defects are identified during unit testing, they will be reported and addressed promptly. The unit tests will be re-executed after defect resolution to ensure correctness.

MODULE/FUNCTIONALITY NAME:	Book management Application
UNIT/CLASS:	Addition, Updating, Deletion, Search
CREATED BY:	Sahil Naik
DATE OF CREATION:	16/10/2023
DATE OF REVIEW:	20/10/2023

TEST CASE ID	TEST UNIT/CLASS	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
1	Book collection	User is able to view their books added	None	1. User is on the landing page. 2. User can navigate through the menu 3. User can view the book	Valid User book collection	User is able to view the books present in their collection	Books collection is visible	User is unable to view any books	Fail
		User searches for the book – Valid Data	Book with this data must exist	1. User is on the landing page. 2. User can navigate through the menu 3. User can search the book	Valid User book data	User is able to search the books present in their collection	Books searched is visible	User is able to view the searched book	Pass
2	Add book Module	User adds book without ISBN	No book with that ISBN exists	1. User navigates to the add book section 2. Add details about the book without ISBN 3. Clicks on add book	Books collection dataset	Error message saying that ISBN of book is required is displayed	All data related to that book is cleared	User is able to add the book even without the ISBN number	Fail

	Add book Module	User adds book with ISBN	No book with that ISBN exists	1. User navigates to the add book section 2. Add details about the book with ISBN 3. Clicks on add book	Books collection dataset	Message saying book added successfully is displayed	All data related to that book is cleared	User is able to add the book even with the ISBN number	Pass
3	Update book	Updating the existing book entry	Book data and entry already exists	1. User is able to view the existing book 2. User updates the book entry data	Existing book collection data	User is able to update the existing book collection data	User is navigated back to all books view	User is able to update the existing book collectio n data	Pass

4.2 System and Integration Testing

Definition:

System and Integration Testing for our project involves validating the interactions between different modules and components in the context of the entire platform-based book management application. System Testing ensures that the system as a whole function correctly and meets the specified requirements. Integration Testing focuses on verifying that individual units or components can seamlessly work together as an integrated system.

Participants:

- QA Team: Responsible for planning, designing, and executing system and integration tests.
- Developers: Collaborating with the QA team to address integration issues and defects.

Methodology:

4.3 Performance and Stress Testing

Definition:

Performance and Stress Testing for our project involves evaluating the system's capability to handle a specific workload and its responsiveness under varying conditions. Performance testing ensures that the system functions efficiently within its

specified parameters, while stress testing assesses the system's behavior when subjected to loads beyond its designed capacity. In the context of our book management system, performance testing focuses on optimizing response times and resource usage, while stress testing examines how the system behaves under extreme conditions, such as high user usage.

Participants:

- QA Team: Responsible for planning, designing, and executing performance and stress tests.
- Developers: Collaborating with the QA team to address performance-related issues and optimizations.

Methodology:

Describe how Performance & Stress testing will be conducted. Who will write the test scripts for the testing, what would be sequence of events of Performance & Stress Testing, and how will the testing activity take place?

4.4 User Acceptance Testing

Definition:

User Acceptance Testing (UAT) for our project is the final phase of testing, conducted to confirm that the web-based blog site meets the requirements and is ready for operational use. During UAT, end-users, who are the intended customers, will evaluate the system to ensure it aligns with the initial project requirements and satisfies their needs.

Participants:

- End-Users: These are the actual users or customers who will use the management system. They are responsible for validating that the system meets their requirements.
- QA Team: Supreet Birajdar, Sahil Naik, Sohan Chidrawar.

Methodology:

User Acceptance Testing (UAT) is a critical phase of testing that involves end-users to ensure that the software meets their requirements and works as expected. Here's a methodology for conducting UAT in your project:

- **Test Script Development**: Test scripts will be developed by the testing team in collaboration with key stakeholders, including actual users or representatives of the target user group. These scripts will define specific test cases and the expected outcomes.
- **User Involvement:** End-users or representatives from the user group will actively participate in the UAT process. They will review the test scripts and provide feedback to ensure that test cases align with their expectations and real-world usage.
- **Test Execution:** During UAT, end-users will execute the prepared test scripts, following the specified sequence of events. They will interact with the system as they would in real-life scenarios, including book insertion, content creation, updating and deletion.
- **Validation of Business Requirements:** Testers will validate whether the system meets the defined business requirements. This includes ensuring that user is able to add new books with details related to it and update it later.
- **Usability and User Experience:** UAT will focus on assessing the usability and user experience aspects. Testers will evaluate the platform's friendliness, ease of navigation, and overall satisfaction with the application’s layout and functionality.
- **Identification of Defects:** If any defects or issues are identified during UAT, they will be documented and reported. The testing team, in collaboration with developers, will work to resolve these issues.
- **Regression Testing:** After defect resolution, a round of regression testing may be performed to ensure that fixes haven't introduced new problems or affected existing functionality.
- **Final Approval:** Once UAT is successfully completed, the end-users will provide their final approval or sign-off on the system. This indicates that they find the software suitable for its intended purpose.

4.5 Automated Regression Testing

Definition:

Automated Regression Testing for our project is the practice of selectively retesting the web-based blog site to verify that recent modifications or updates have not inadvertently introduced defects or negatively affected the system's existing functionality. The primary goal of regression testing is to ensure that the system continues to function as specified in the requirements, even after changes have been made.

Participants:

- QA Team: Responsible for planning, designing, and executing automated regression tests.
- Developers: Supreet Birajdar, Sahil Naik, Sohan Chidrawar.

4.6 Beta Testing

Participants:

- QA Team: Responsible for planning, designing, and executing automated regression tests.
- Developers: Collaborating with the QA team to address and correct any defects found during regression testing.

Methodology: A beta test is the second phase of software testing in which a sampling of the intended audience tries the product out.

4.7 Compatibility Testing:

Definition: Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices

Participants:

- QA Team: Responsible for planning, designing, and executing automated regression tests.
- Developers: Supreet Birajdar, Sahil Naik, Sohan Chidrawar .

Methodology: Compatibility testing ensures software or hardware works correctly across different environments and configurations. It involves defining the scope, creating test cases, executing tests on various platforms, identifying and resolving issues, and validating fixes. Results are documented and, after successful testing, the product is released. Ongoing monitoring is essential to adapt to changing technologies and user needs.

5.0 HARDWARE / SOFTWARE REQUIREMENTS:

Processor : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
Ram : 8.00 GB (7.87 GB usable)
IDE : Eclipse IDE 4.21
Selenium version : Python 3.0
Browser : Chrome Version 118.0.5993

6.0 TEST SCHEDULE

Task Name	Start Date	Finish Date	Effort Estimation	Comments
Test Planning	16/10/2023	16/10/2023	30 Mins	Completed
Review Requirements documents				
Create initial test estimates				
Functional testing – Register	16/10/2023	16/10/2023	1 hour	Completed
Functional testing – login	17/10/2023	17/10/2023	30 Mins	Completed
Functional testing -home page	17/10/2023	17/10/2023	1 hour	Completed
System testing	18/10/2023	18/10/2023	20 Mins	Completed
Regression testing	19/10/2023	19/10/2023	1 hour	Completed
User Acceptance Testing	19/10/2023	19/10/2023	30 Mins	Completed
Resolution of final defects and final build testing	20/10/2023	20/10/2023	1 hour	Completed
Report Generation	21/10/2023	21/10/2023	30 minutes	Completed

7.0 FEATURES TO BE TESTED

7.1 Book operation and Authentication:

- Book insertion with valid data
- Book insertion with invalid data
- Book updating with valid credentials
- Book search with invalid credentials

7.2 Navigation and Usability:

- Navigating through the site menus
- User interface responsiveness and compatibility with different systems.

7.3 Compatibility Testing:

- Testing compatibility with various operating systems.

8.0 FEATURES NOT TO BE TESTED

8.1 External Integrations:

- Feature: Integration with a third-party service (e.g., payment gateway).
- Reason: Third-party services are tested separately, and this specific integration is covered by their own testing.

9.0 RESOURCES/ROLES & RESPONSIBILITIES

Team member details and work distribution

Sr No	Name	Roles and Responsibilities
1	Sahil Naik	Tester / Developer
2	Sohan Chidrawar	Tester / Developer
3	Supreet Birajdar	Tester / Developer