Test Plan

for

Bookshop Automation Software

Version 1.0 approved

Prepared by Team: MAD

Aashray Tandon 22CS30002

Deetosh Kumar Kuila 22CS30021

Mayash Nayak 22CS30064

Indian Institute of Technology, Kharagpur

9 April 2024

Contents

1 Test Plan Identifier

- Test Plan for Bookstore Automation Software (BAS) - Version 1.0

2 References

- Software Requirements Specifications for Bookshop Automation Software.
- Test plan outline (IEEE 829 FORMAT)

3 Introduction

This document is a high-level overview defining testing strategy for the Book Automation Software. This document will only address those functions which were described in the SRS document submitted earlier. The primary focus of this plan is to ensure that all the various components of this web application are working as expected in a wide variety of situations. Testing criteria under the white box, black box, and system-testing paradigm will be utilized. Throughout the testing process, the test documentation specifications described in the IEEE Standard 829-1983 for Software Test Documentation will be applied.

4 Test Items

The following is a list of the items to be tested

- Book query and availability
- Book request
- Cart management
- Sales management
- Inventory management
- User management
- Reports and Statistics generation

5 Software Risk Issues

The following is a list of items that are a potential risk factor to project success

A. Incorrect and Inconsistent Database Entries

Having incorrect or inconsistent database entries in the database will result in fatal errors in the web app, as the web application will be unable to determine the correct records for updating as well as show incorrect results to the users. Inconsistent database entries might be a result of wrong linking database models or failure of update functions (add, delete, create, modify). Incorrect database entries might be a result of wrong data fed by the user. In order to tackle these problems, the software development team needs to put in adequate checking functions in place as well as test the update functions and linking of the models in the database.

B. Third Party Integrations

While leveraging third-party software can expedite development and add functionality, it introduces several potential risks. Firstly, compatibility of the third-party software with the web application poses a risk. If the third-party software is not compatible it can lead to the web application having unpredictable behaviour or system instability. Moreover, changes or discontinuation of support from third-party vendors could necessitate costly rewrites or replacements of affected components. To mitigate these risks, thorough evaluation and testing of third-party software, contingency planning for potential disruptions is necessary.

C. Network Requirements

The web application should not crash and should be responsive under a huge load. The web application should be able to connect over LAN, and should be accessible to all the users connected to it. The LAN network should not block any third-party software being used.
The web application should be able to work concurrently and manage multiple users simultaneously. Therefore, the web application needs to be tested against all these possibilities in order to ensure smooth functioning.

6 Features To Be Tested

These are the features that are to be tested from the user's perspective. This primarily consists of user-facing features that are handled by a graphical web page that allows privileged access to the internal server operations.

A. Book Query and Availability (Priority M)

- Verify the system's ability to retrieve and display book details on the basis of author name / title.
- Validate the responses for both existing and non-existing books.
- Validate whether the system shows the correct details of the respective books as

per the database.

B. Book Request (Priority L)

- Verify the submission and processing of new book requests.
- Test request submission for additional copies of available books. Make sure that the requested quantity is more than the quantity available.

C. Cart Management (Priority H)

- Verify adding books to the cart making sure that the quantity added is less equal the current stock available.
- Verify removing books from the cart.
- Test deletion of abandoned carts.
- Test sending a summary of cart items via email.

D. Sales Management (Priority M)

- Verify completion of book purchases and bill generation.
- Verify sending the bill to the user via email.
- Test updating inventory records upon completion of the transaction.
- Validate generation of sales graph and revenue analysis (filter sales by date or book).

E. Inventory Management (Priority H)

- Verify inventory updates after transactions and stock arrivals.
- Test calculation and identification of books under the threshold value.

F. User Management (Priority H)

- Verify user authentication and segregation of different types of the user based on customer, manager and employee.
- Test customer registration process.
- Validate creation of separate carts for different customers.

7 Features Not to Be Tested

The following is a list of the areas that will not be specifically addressed.

A. Structure Of Models and Database

The structure of the models such as fields created won't be tested as they are implicitly checked by Django. Nor, the linking between different tables in the database will be checked as they are implicitly made by Django.

8 Testing Approach

A. Unit Testing

The unit test cases shall be designed to test the validity of the program's correctness. The following are the functions that are going to be tested

- search_authors()
- search_books()
- book_details()
- request_book()
- send_procure_book()
- cart()
- remove_from_cart()
- add_to_cart()
- send_email()
- proceed_to_buy()
- register_view()
- login_view()
- logout_view()
- revenue_graph() in SalesAdmin
- queryset() in ThresholdFIlter

The unit testing is implemented in a white box testing fashion as we are testing directly within the application at the source code level.

B. General Interface Testing

The general interface testing shall cover the testing of the web applications GUI. For the testing purposes manual testing will be employed. The following process will be followed

- The manual tester will open the web application on different browsers to check for compatibility issues.
- To check whether there are any broken URLs or not, the Django test framework will be employed.

The general interface testing is implemented in a grey box testing fashion as we are testing some of the functions at the source code while for others, we are only looking at the final output with the internal works abstracted.

C. Performance Testing

The performance testing shall cover the response of the web application under high user load and poor connectivity issues. For the testing purposes manual testing will be

employed. The following process will be followed

- The manual tester will check whether the web applications is functional under low-bandwidth situations.
- The manual testers shall check whether the web application is responsive under heavy user load.

The performance testing is implanted in a black box testing fashion as we are testing the features by only evaluating the final output with the internal works abstracted from us.

D. Measures and Metrics

The following information will be collected by the Development team during the Unit testing process.

- Errors and bugs in the system
- Web page loading time
- Response under heavy user load (defined by more than 20 users using the system at a given point of time)
- Potential user interface complications like broken links, convoluted menus and submenus.
- Failure of any third-party applications.

9 Item Pass Fail Criteria

A. Item Pass Criteria

- All the unit tests for the features to be tested should be successfully working. This implies that there should be no test case that should not be working. Any test case which does not work means that the software has bugs and cannot be delivered to the client.
- The web application should load on the latest versions of widely used web browsers like Mozilla Firefox, Google Chrome and Internet Explorer. The web application should be able to handle at least 20 users at a given point of time as well as should be able to load web pages within 5 seconds.

B. Item Fail Criteria

If the web application does not fulfil all the requirements mentioned in the item pass criteria, then the software will not be approved and the software development team needs to take appropriate actions and plan to debug the software.

10 Suspension Criteria and Resumption Requirements

The testing needs to be suspended if the following cases occur

A. Failure to load the web page of different subsystems

The testing will be suspended if the web page related to the concerned sub-system does not load. This is necessary as tests related to that particular sub-system cannot be proceed to further levels.

B. Failure of Priority H subsystems

The testing will be suspended if any of the priority H sub system fails as the web application is dependent on the success of all the priority H sub system and priority H subsystems are dependent on each other.

The testing will only resume if the above cases are not encountered or addressed if encountered.

11 Test Deliverables

- Test Plan
- Test Cases
- Bug Report
- Error Logs

12 Schedule

- The testing will occur after the development process is complete. The developers will meet every day for one week until the testing is complete.

13 Glossary

Term	*Definition*
Threshold value	The quantity of books sold over a period of two weeks + a constant number (20)
Customer	A person who is querying for the book in the bookstore with the motivation of buying it.
Manager	A person who has the super user access of the software and can view the sales statistics

	as well as order for new books, alter the inventory, list of vendors and other information.
Employee	A person who only has access to alter the inventory of the bookstore.
BAS	Bookshop Automation Software
SRS	Software Requirement Specification