Test Plan and Test Case Specification

Bookstore management system

Version 6.0

30/8/2023

https://github.com/AbdelRahmanAlmasri/BMS.git

SE

AAU

[TESTING AND IMPLEMENTATION 56](file:///C:\Users\user\OneDrive\Desktop\Bookstore%20Management%20System\Documentation.docx#_Toc3818975)

[4.1 Testing Approach Used 57](file:///C:\Users\user\OneDrive\Desktop\Bookstore%20Management%20System\Documentation.docx#_Toc3818976)

[4.2 Test Cases 59](file:///C:\Users\user\OneDrive\Desktop\Bookstore%20Management%20System\Documentation.docx#_Toc3818977)

[4.3 Implementation Approaches 63](file:///C:\Users\user\OneDrive\Desktop\Bookstore%20Management%20System\Documentation.docx#_Toc3818978)

# Chapter 4

# Testing And

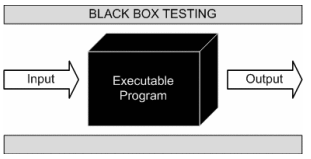
# 4.1 Testing Approach Used

* Black box testing

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as specifications based .

Testing Independent testing team usually perform this type of testing during the software testing life cycle.

This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.



### (Figure 37 : Black Box Testing)

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

* + - Incorrect or missing functions
    - Interface errors
    - Errors in data structures or external database access
    - Behavior or performance errors
    - Initialization and termination errors

## Advantages of Black box Testing:

Tests are done from a user’s point of view and will help in exposing discrepancies in the specifications.

Tester need not know programming languages or how the software has been implemented.

* White box testing

White box testing is a testing technique That examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

White box testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations performed according to the specification. And all internal components have been adequately exercised.

* White box testing techniques:

1. Statement Coverage – This technique is aimed at exercising all programming statements with minimal tests.
2. Branch Coverage – This technique is running a series of tests to ensure that all branches are tested at least once.
3. Path coverage – This technique corresponds to testing all possible paths which means that each statement and branches is covered.

* Advantages of white box Testing:

1. Forces test developer to reason carefully about implementation.
2. Reveals errors in “hidden” code.
3. Sports the code or other issues with respect to best programming practices.

* Gray-box Testing:

Grey-box testing is a testing technique performed with limited information about the internal functionality of the system. Grey-box testers have access to the detailed design information about requirements.

Grey box are generated based on the state based modes, UML diagrams or of the target system.

Grey Box Testing is a technique to test the software product or application with partial knowledge of the internal workings of an application.



### (Figure 38 : Gray Box Testing Testing)

# 4.2 Test Cases

4.2.1 Admin Login Detail

|  |  |  |  |
| --- | --- | --- | --- |
| Username | Admin | Password | Admin |
| Expected Result :   * If fields empty then gives a error for fill up fields * If password or username does not exist then gives error for valid detail. | | | |

4.2.2 Login Detail

|  |  |  |  |
| --- | --- | --- | --- |
| Username | Dhaval | Password | Dhaval |
| Expected Result :   * If fields empty then gives a error for fill up fields * If password or username does not exist then gives error for valid detail. | | | |

4.2.3 Registration Details

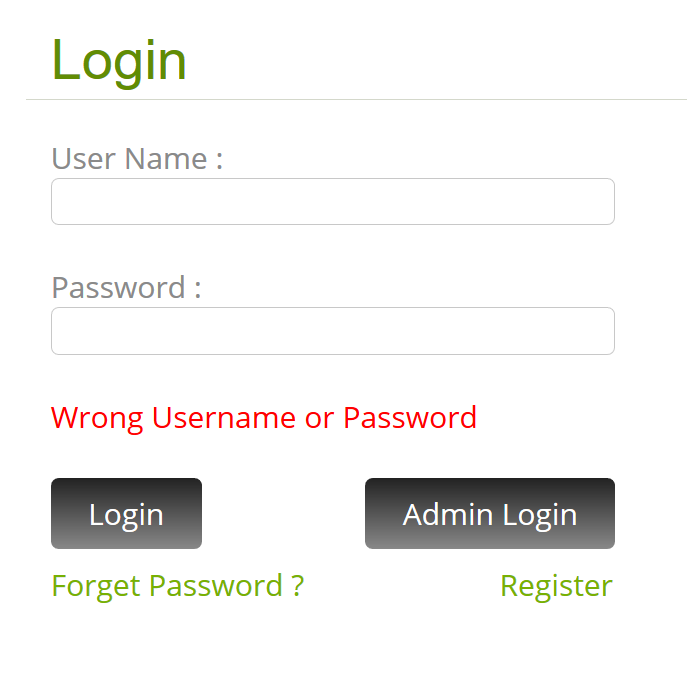
|  |  |  |  |
| --- | --- | --- | --- |
| Username | EMPTY | Password | EMPTY |
| Full Name | EMPTY | Security Answer | EMPTY |
| Expected Result :   * If fields empty then gives a error for fill up fields * If password or username does not exist then gives error for valid detail. * If password is < 8 characters then it will gives error. | | | |

4.2.4 Order Details

|  |  |  |  |
| --- | --- | --- | --- |
| Full Name | Address | Contact Number | EMPTY |
| Expected Result :   * If fields empty then gives a error for fill up fields * If contact number is not Numeric then gives error | | | |

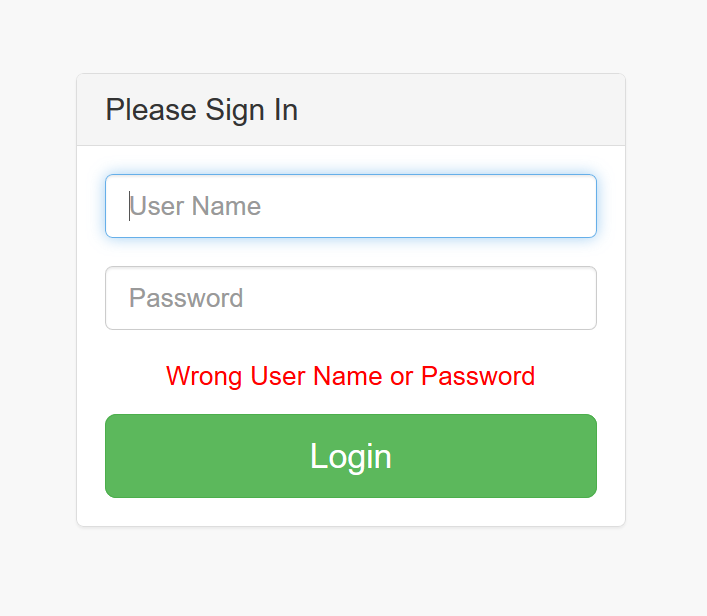
### Screen-Shots

1. User Login



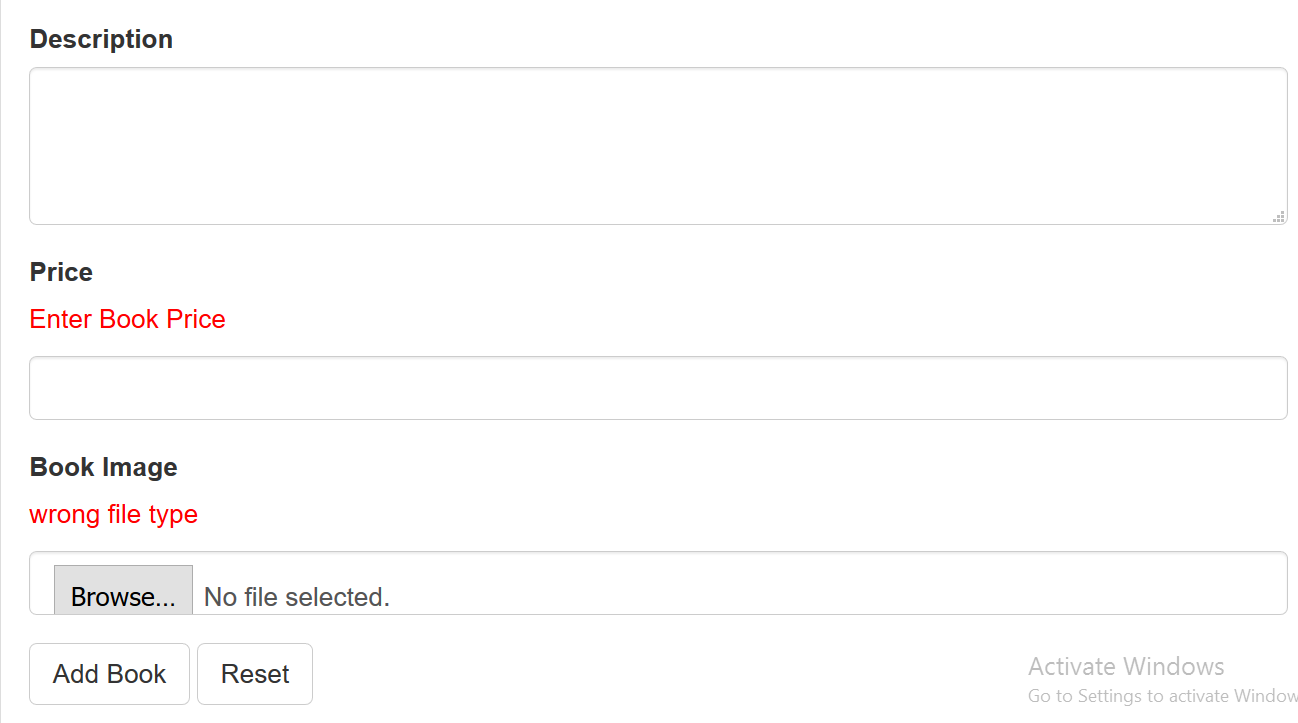
### (Figure 39 : Test Cases 1)

1. Admin Login



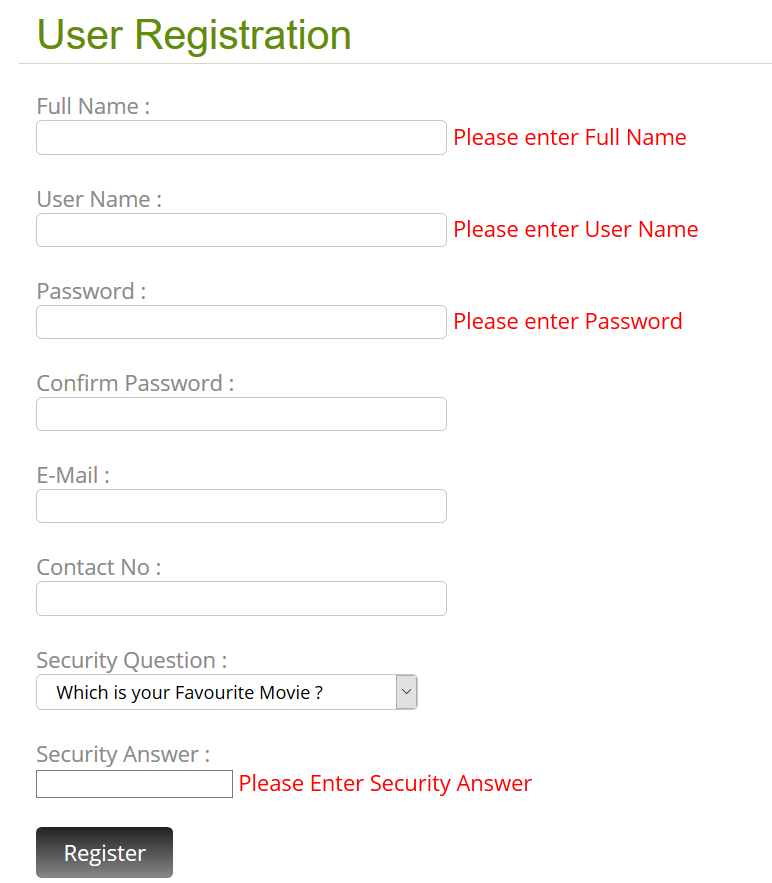
### (Figure 40 : Text Cases 2)

1. Add Book



### (Figure 41 : Text Cases 3)

1. User Registration



### (Figure 42 : Text Cases 4)

# Implementation approaches

Far the biggest challenge encountered was time constraints. Implementation takes an extraordinary amount of time and a large amount of coordination. Scheduling project meetings around every group member’s schedule has been nearly impossible. Many of the group members were unable to devote the amount of focus that the implementation stage required. Both the former and the latter problem may be more of an issue in the academic environment where the priorities of the different group members are skewed in a variety of directions. Another issue that cropped up was knowledge of the PHP programming. At least two of the four group members were unfamiliar with PHP Swing API, which is php primary user interface package. Again, this may not be as much of an issue in software engineering outside the academic arena.

One of tools we found very useful, in situations where member responsibilities need to be hashed out, is the responsibility matrix. It has really been the only tool that has allowed us to continue making progress. Everyone is assigned a task, and everyone is held accountable for the completion of their assigned task. It also allows us to track tasks that need to be done. The responsibility matrix has proven to be an invaluable tool in the software engineering process.