Software Testing Document

Used Book System

Marya Belanger

TABLE OF CONTENTS

TABLE OF FIGURES

1 TEST PLAN

1.1 Features to be Tested

The tests covered in this document include all the implemented features of the Used Book System (UBS). These include Login, Register, Search for a Book, and Add New Book. We will test some cases with validation testing, to show that the software meets its requirements. The results of these tests should show the expected outputs defined in the requirements document. We will also use defect testing for other cases by entering incorrect inputs to see how the software reacts. For the defect tests, results should show anomalies and inconsistencies that need to be worked on.

1.2 Features not to be Tested

The features not to be tested in this document (aside from those functional requirements that remain unimplemented) are some non-functional requirements. These include availability and performance, which cannot be tested until the website is put online and operates off of an active server, not the localhosts used in development. Also, we will not test security, because the system is incomplete so the security measures that should be implemented before release will not function properly on our unsecured test version.

2 TEST CASES

2.1 Case 1 - Search

2.1.1 Purpose

The purpose of this test case is to ensure the proper functionality of the “Search” feature, by which users search for a book/books. This means the user follows the guidelines of searching by “book title, course title, or course code” or any single keyword, and results are displayed that properly match their search. For this case we will use validation testing.

2.1.2 Inputs

The example inputs we will represent will be partitions from each category the search operates under according to its requirements. As mentioned, one will be “book title”, for which we will use the example of a specific book entitled “C How to Program”. The next partition will represent “course title”, and the sample input will be “Discrete Mathematics”. The next partition will be “course code” for which we will input “CS330”. The last is any keyword, and we will input “data”.

2.1.3 Expected Outputs

A successful output for the “book title” input will return books whose titles are labeled the same as the input text. For “course title”, a successful output should show books for the name of the course which we enter. For “course code”, books should be displayed that are associated with the course whose code we input. For keyword searching, a successful output is one that contains the keyword in either the title, the course name or the course code, or a combination of any of those categories.

2.1.4 Test Procedure

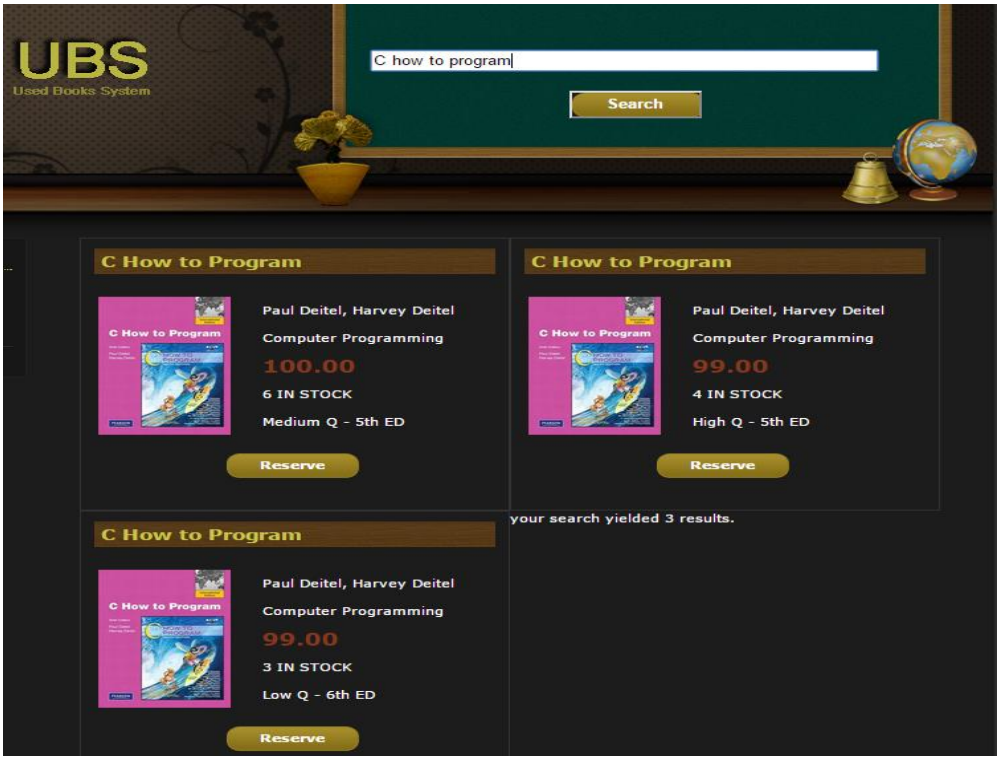

Figure 1: Search "book title"

As shown in Figure 1, inputting "C how to program" and pressing search displays all books with that title. The title can be seen as the top of every division in yellow text on a brown background. The reason three results are shown for one title is that they are different versions of the book; in the fifth line to the right of the cover image you can see different quality levels and editions.

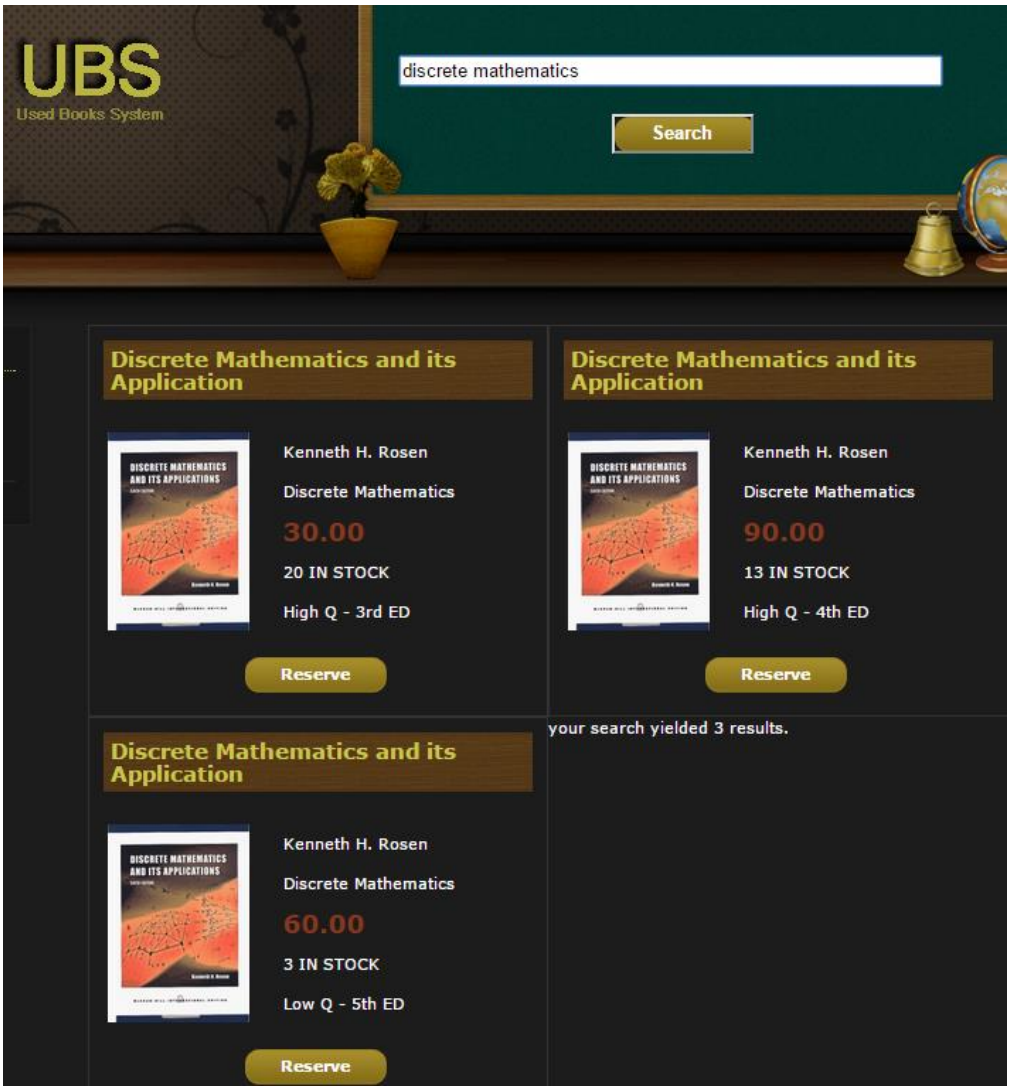

Figure 2: Search "course title"

In Figure 2, it can be seen that searching for the course "Discrete Mathematics" returns books that are associated with that course. The second line down to the right of the cover image, the course associated with the book on display is listed.

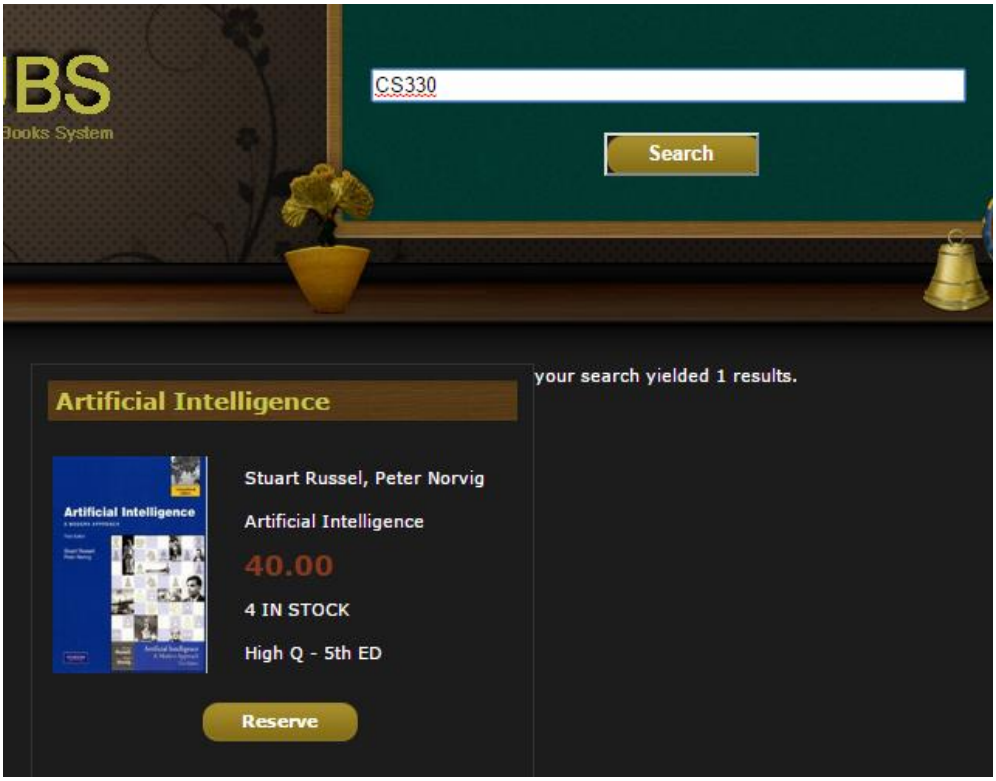

Figure 3: Search "course code"

As seen in Figure 3, inputting a course code into the search bar successfully displays all books associated with that course. The course code itself is not displayed with the book information, but rather it is processed internally through the database.

Figure 4: Search with keyword

Figure 4 shows that using a keyword to search does as expected; books are displayed that contain the keyword “data” in both/either the book title and/or the course title.

2.2 Case 2 - Register

2.2.1 Purpose

To test the “Register” feature of the system, we will use defect testing. This is in order to highlight its unresponsiveness to certain inconsistencies. We will test the condition where a user tries to register with an ID that belongs to another user. Since the ID is the defining attribute (or

primary key) of each user's information in the database, registering two users with the same ID should not be allowed.

2.2.2 Inputs

The sample input will be one where the ID field is identical to one already existing in the database. Checking the database, it is shown that the ID number "1200000" is already ascribed to a user, so our sample input will include "1200000" as the ID. The rest of the input fields are inconsequential to the results of this test; users can have the same name, password and phone number. More than one user cannot, however, have the same email, but that is unimportant to our test because it is not a matter of integrity as the email attribute is not a primary key.

2.2.3 Expected Outputs

The output that should result from this defect test is that the new user trying to register should not be entered into the system. The information should not be saved if the user tries to claim an already-existing ID as their own. Additionally, the user should be alerted that their ID must be unique, and that the ID they are trying to enter already exists in the system, so they should try registering again with a different ID.

2.2.4 Test Procedure

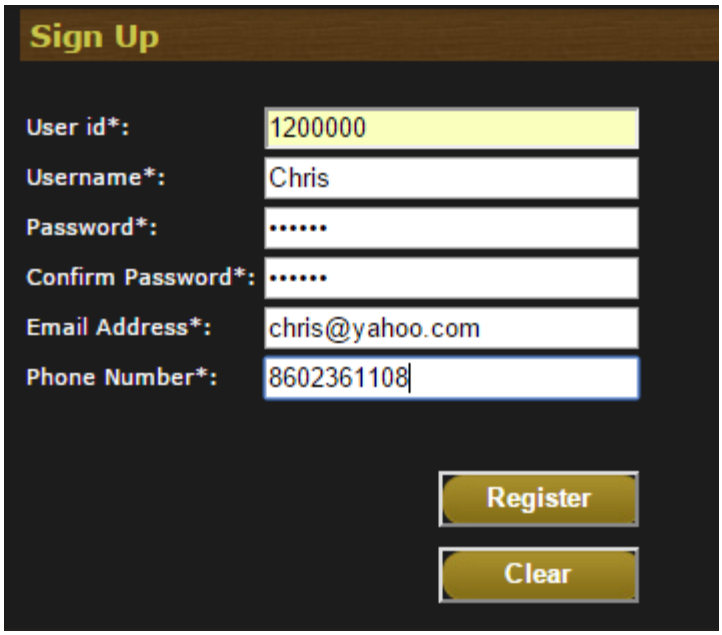

Figure 5: Register

In Figure 5 the sample input of User id "1200000", plus some irrelevant inputs for Username, Password, etc., are shown. Pressing "Register" should deliver some alert that this input is not being accepted.

Figure 6: After Registering

USER_ID	USER_NAME	PASSWORD	ROLE	EMAIL	PHONE NO
0	nada	00c66aaf5f2c3f49946f15c1ad2ea0d3	clerk	kk@gmail.com	14246
1200000	yara	e10adc3949ba59abbe56e057f20f883e	student	yara@gmail.com	0
11120053	nedaa	00c66aaf5f2c3f49946f15c1ad2ea0d3	student	nedaahadi94@gmail.com	0
12120349	marya	e10adc3949ba59abbe56e057f20f883e	student	m@gmail.com	0

Figure 7: Database (no changes made)

After pressing "Register", we can identify a defect. The user is not alerted that their registration is not confirmed. Figure 6 is the view the user will see after incorrectly registering. There is no way for the user to know if their registration failed of succeeded. It is possible that the user may now think that they have registered properly and that they now posses an account, but this is incorrect. It is necessary for the user to be alerted that their transaction was not completed.

Also evident from Figure 7 is that there was, in fact, no change made to the database and the user was not added to the system. However, this is just because of the database maintaining its own integrity. In reality, this should be handled by the system because letting the database secure its own integrity is not safe.

2.3 Case 3 - Login

2.3.1 Purpose

This test case is a validation test that will prove that both students and clerks can login successfully and their views will be displayed appropriately. The test will show that clerks will not have access to student views, and that, more importantly, students will not have access to clerk views. The test will show the correct login of users, student and clerk, already existing in the system (Fig.7).

2.3.2 Inputs

For testing the clerk login, the input values will be that of clerk with User id “0” and Password “123456”. For testing the student login, the input values will be that of student with User id “12120349” and password “123456”. As mentioned before, these are pre-existing users to ensure that the login will be successful.

2.3.3 Expected Outputs

The expected outputs of this test case are successful ones; namely, that the clerk will be directed to the clerk view and be able to access all his functionalities, and that the student will be directed to the student view, and be able to access all his functionalities. The output is also expected to not allow one type of user (clerk or student) to have access to the other’s views.

2.3.4 Testing Procedure

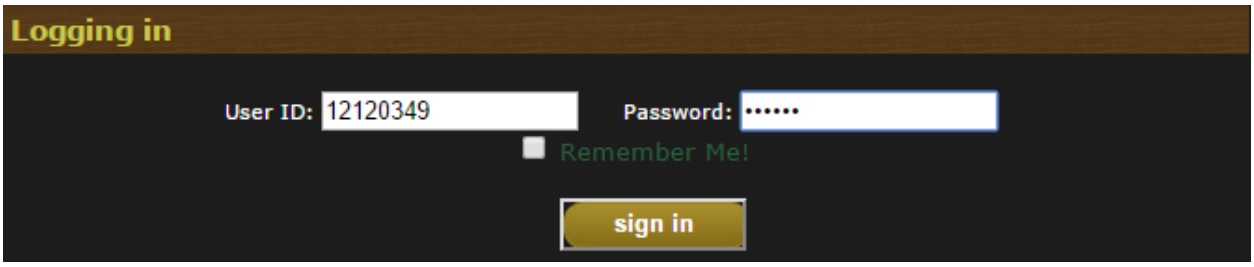

Figure 8: Login as student

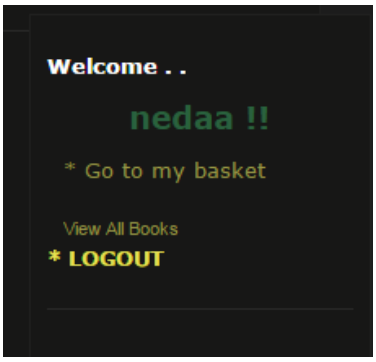

Figure 9: Student view

Figures 8 and 9 show the successive login and view of a student user. After they entered their information, and it was valid and existed in the system, they were directed back to the homepage. The home page for each view looks the same, save for the left option bar. From here, students can access their basket, view all the books, and logout. The fact that this option bar is showing means the login was successful.

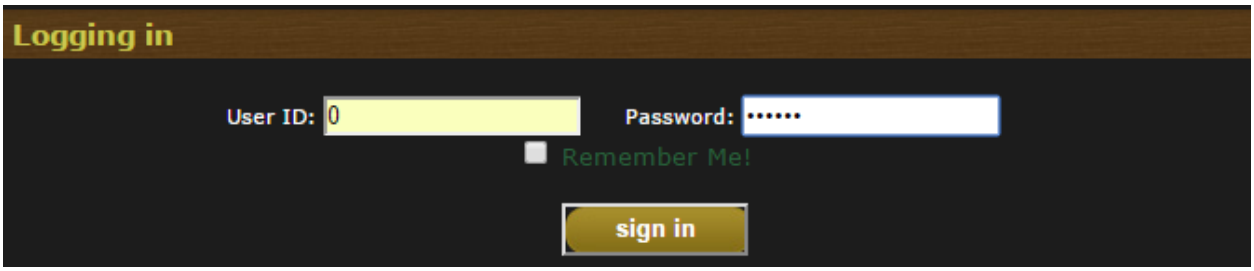

Figure 10: Login as clerk

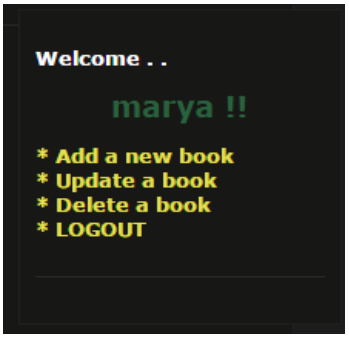

Figure 11: Clerk view

The login and view are the same for the clerk. The clerk is also directed back to the homepage after logging in, and the clerk also has his own unique option bar on the left side of the page. Here the clerk can add a new book to the system, update a book already in the system, or delete a book. The fact that this view is available for the clerk means the login was successful.

2.4 Case 4 - Add New Book

2.4.1 Purpose

Defect testing will be perform on this test case. In this feature the clerk has to enter the book info manually even for inserting the book cover. Our system assumes that the clerk has a folder full of images that he will then use to enter the cover image. For entering the book image, he has to enter the image name with its extension. The test will show the output of entering a picture that doesn't exist in the pictures folder.

2.4.2 Inputs

As explained above, the input is a picture that doesn't exist in the image folder. So the input is "1453" for the ISBN, "World Wide Web How to Program" for the title, "deitel" for the author, "cs373.jpg" for the cover image, "cs373" for course code, "web Engineering" for the course name, "2" for quantity, "100" for price, and "High Q - 3 Ed" for the description.

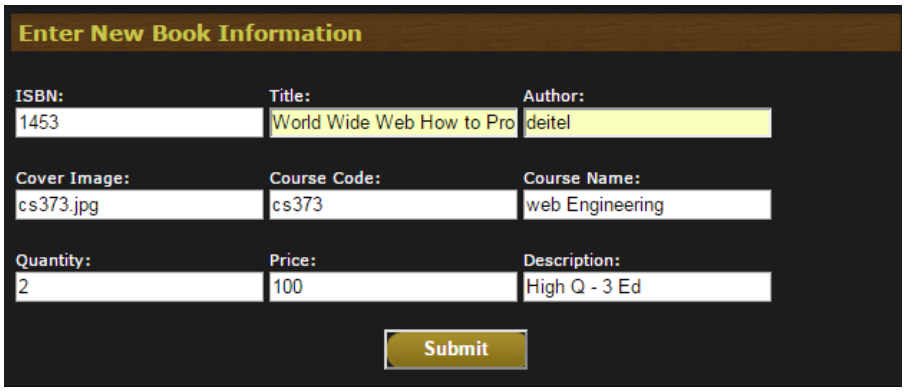

Figure 12: adding a new book

2.4.3 Expected Outputs

The output for this defect testing should be a new book entered to the database with its complete information so that when a student searches, nothing should be missing from the search result. To insure the completeness of the information, the clerk should choose the book cover rather than writing its name.

2.4.4 Testing Procedure

Figure 13: Search Result

The expected output for this input is not a successful one since no image will appear if anyone searches for this book.