|  |  |  |
| --- | --- | --- |
| Guillaume Labasse | **Homework 3** | SEG3103 |
| 300132984 | **E-Book Store** | 20/06/2021 |

**Question 1**

**Assumption:** In general, requirements with only one sub-section that serve as additional details are treated as one requirement. I make an exception for F5 and F5.1 which seem distinct enough to me, and that I separate into two requirements.

|  |  |  |
| --- | --- | --- |
| **Req** | **Positive Situation** | **Negative Situation** |
| **F1** | All book information fields are given valid input → book added to catalogue. | Some fields have invalid input (e.g., nothing in the Category, Title and Author fields; Book ID less than 5 characters or more than 8; Cost not a number or lower than 0) → book not added to catalogue. |
| **F2.1** | Non-empty category searched for → books from that category are displayed. | Empty (thus inexistant) category searched for → no books displayed. |
| **F2.2** | Catalogue is not empty → all books from the catalogue are displayed. | Catalogue is empty → no books displayed. |
| **F3** | Desired book is available → book added to order. | Desired book unavailable (e.g., out of stock) → book not added to order. |
| **F4** | Book order is not empty → user can view its contents. | Book order is empty → cannot view the contents. |
| **F5** | Book order list not empty, valid number of copies greater than one given → number of ordered copies updated. *This number can also be increased at the “adding books from catalogue” step.* | Cannot be tested with an empty book order; the feature is not available. With a non-empty order, invalid (e.g., not a number) number of copies given OR item was already removed → number of copies is not updated. |
| **F5.1** | Book currently in order → book removed from order. | Book already removed from order → nothing changes. See Q3, Implementation Issues. |
| **F6.1** | Valid (e.g., not empty, not exceeding stock) order sent → checkout information displayed with tax rate correctly calculated. | Invalid (e.g., empty or exceeding stock) order sent → error message instead of checkout information (no tax rate). See note. |
| **F6.2** | Valid (e.g., not empty, not exceeding stock) order sent → checkout information displayed with handling & shipping fees correctly calculated. | Invalid (e.g., empty or exceeding stock) order sent → error message instead of checkout information (no handling & shipping fees). See note. |
| **F7** | Catalogue is not empty → books can be removed. | Catalogue is empty → no books to remove. See note. |
| **F8** | Valid administrator credentials given → admin logged in | Invalid credentials given → user not logged in as admin. |

**Note**: These are situations where the feature in question *cannot be accessed* in a negative context. There is no way to both use this feature and get a negative result. As I added the concept of “availability” for F3, you may also consider the concept of “undeletability” for a better F7 negative test scenario, but I was unable to come up with acceptable reasons for why a book should be impossible to delete.

**Question 2**

**Use Case 1**: Administrator Sign in

|  |  |
| --- | --- |
| **Main Scenario**  1. Administrator navigates to administrators’ operation page  2. e-BookStore asks for login credentials  3. Administrator enters login credentials  4. e-BookStore presents administrators operations page  **Alternatives**  3. a. Wrong login credentials  3. a. 1. e-BookStore displays invalid credentials error message  3. a. 2. GOTO Step 2 |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2-3-4 | Normal course of events; admin logins with valid credentials. |
| 2 | 1-2-3-3a.1-2\*-3-4 | Admin logins after giving invalid credentials any number of times. |

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: SignInTC1 **Goal**: Test normal course of events for admin login. **Scenario** #: 1 **Setup**: Opened the admin login page; account with name "admin" and password "password" exists. **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User submits username "admin" and password "password" | System validates admin credentials and opens admin session. |   **Success Criteria**: Admin logged in |
| **Test Case**: SignInTC2 **Goal**: Test for successful admin login after a few failures entering credential. **Scenario** #: 2 **Setup**: Opened the admin login page; account with name "admin" and password "password" exists. **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User submits username "blah" | System displays "invalid username and/or password" | | **2** | User submits password username "foo" and password "bar" | System displays "invalid username and/or password" | | **3** | User submits username "admin" and password "password" | System validates admin credentials and opens admin session. |   **Success Criteria**: Admin logged in |

**Use Case 2**: Administrator Log Out

|  |  |
| --- | --- |
| **Main Scenario**  1. Administrator selects to Sign out  2. e-BookStore acknowledges that Administrator has Signed out **Added** 3.User goes back to login screen |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2-3 | Normal course of events. Administrator logs out. |

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: LogOutTC1 **Goal**: Test the main course of events for logging out. **Scenario** #: 1 **Setup**: Administrator is logged in on the home page **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User clicks "Sign out" button | System logs out user and redirects them to login screen. |   **Success Criteria**: Administrator is logged out. |

**Use Case 3**: Add Book

|  |  |
| --- | --- |
| **Main Scenario**  1. Administrator enters the book information  2. e-BookStore acknowledge the book addition to the catalogue  **Alternatives**  1. a. Wrong or missing information  1. a. 1. e-BookStore displays the pertinent error messages  1. a. 2. GOTO Step 1  1. b. Duplicated book id  1. b. 1. e-BookStore displays a duplicated book id error message  1. b. 2. GOTO Step 1 |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2 | Normal course of events. Book added. |
| 2 | 1-1a.1-1\*-2 | Book added after invalid information rejected any number of times. |
| 3 | 1-1b.1-1\*-2 | Book added after invalid (duplicate) ID rejected any number of times. |
| 4 | 1-1a.1-1-1b.1-1-2 | Book added after invalid information rejected + invalid ID rejected once. |

*Additional scenarios can be added for repeated failures due to* ***both*** *reasons but those would add a lot more work on my side for little actual utility.*

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: AddBookTC1 **Goal**: Test normal course of events for adding books. **Scenario** #: 1 **Setup**: Admin is logged in; valid information entered. **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | Admin clicks Add button | System adds the book to the catalogue. |   **Success Criteria**: Book added to catalogue. |
| **Test Case**: AddBookTC2 **Goal**: Test adding a book after entering invalid information. **Scenario** #: 2 **Setup**: Admin is logged in; no information entered. **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | Admin clicks Add button | System displays all validation errors. | | **2** | Admin enters "z" in all fields & clicks Add | System displays "invalid id" and "invalid cost" errors. | | **3** | Admin enters "azaza" as ID and 1 as Cost & clicks Add | System adds the book to the catalogue. |   **Success Criteria**: Book added to catalogue. |
| **Test Case**: AddBookTC3 **Goal**: Test adding a book after entering duplicate IDs (already in catalogue). **Scenario** #: 3 **Setup**: Admin is logged in; valid information entered with ID "hall001" **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | Admin clicks Add button | System gives "id already exists" error | | **2** | Admin enters "bbbbbb" ID & clicks Add | System adds the book to the catalogue. |   **Success Criteria**: Book added to catalogue. |
| **Test Case**: AddBookTC4 **Goal**: Test adding a book after entering invalid information and duplicate ID. **Scenario** #: 4 **Setup**: Admin is logged in; no information entered. **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | Admin clicks Add button | System displays all validation errors. | | **2** | Admin enters "a" in all fields, "hall001" as ID and 1 as cost & clicks Add | System gives "id already exists" error | | **3** | Admin enters "ccccccc" as ID & clicks add | System adds the book to the catalogue. |   **Success Criteria**: Book added to catalogue. |

**Use Case 4**: Browse Catalogue

|  |  |
| --- | --- |
| **Main Scenario**  1. User specifies a category upon which a catalogue search is to be performed  2. e-BookStore shows a list of books satisfying the selected category  **Alternatives**  1. a. No category was specified  1. a. 1. e-BookStore shows a list of all the books in the catalogue  1. b. No book satisfying the searched category is found  1. b. 1. e-BookStore displays a no matching book found message |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2 | Normal course of events. Category displayed. |
| 2 | 1-1a.1 | Full catalogue displayed. |
| 3 | 1-1b.1 | No books displayed. |

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: BrowseTC1 **Goal**: Test the normal course of events for searching **Scenario** #: 1 **Setup**: AddBookTC1 **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User searches for AddBookTC1's book category | System displays the book added in AddBookTC1. |   **Success Criteria**: Books displayed |
| **Test Case**: BrowseTC2 **Goal**: Test the normal course of events for displaying the full catalogue **Scenario** #: 2 **Setup**: None **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User clicks search. | System displays all books in the catalogue. |   **Success Criteria**: All books displayed |
| **Test Case**: BrowseTC3 **Goal**: Test for searching an empty/non-existent category **Scenario** #: 3 **Setup**: None **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User searches for category "foobar" | A "no item matching category" error message is displayed. |   **Success Criteria**: No books displayed |

**Use Case 5:** Remove Book

|  |  |
| --- | --- |
| **Main Scenario**  1. Administrator selects to remove a book  2. e-BookStore shows the list of books without the removed book |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2 | Normal course of events. Book removed. |

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: RemoveBookTC1 **Goal**: Test the normal course of events for removing books **Scenario** #: 1 **Setup**: Logged in as admin; non-empty category searched for **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | Admin clicks Delete button | System removes book from list & catalogue. |   **Success Criteria**: Given book removed from catalogue |

**Use Case 6**: Order Book

|  |  |
| --- | --- |
| **Main Scenario**  1. Customer selects a book to add to his/her books order  2. e-BookStore add the book to the books order  **Alternatives**  1. a. The selected book is already in the customer’s books order  1. a. 1. e-BookStore increments the number of copies of the book in the books order |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2 | Normal course of events. Book added to order. |
| 2 | 1-1a.1 | Number of copies in order for selected book incremented. |

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: OrderBookTC1 **Goal**: Test the normal course of events for adding books to an order **Scenario** #: 1 **Setup**: Not logged in as admin, book list displayed **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User clicks "add to cart" for book ID'd "hall001" | System adds book with ID "hall001" to order. |   **Success Criteria**: Book added to order |
| **Test Case**: OrderBookTC2 **Goal**: Test adding multiple books to an order from the search result list **Scenario** #: 2 **Setup**: Not logged in as admin, book list displayed **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User clicks "add to cart" for book ID'd " hall001" twoce | System adds book with ID "hall001" to order a second time. |   **Success Criteria**: Book added to order twice |

**Use Case 7**: View Books Order

|  |  |
| --- | --- |
| **Main Scenario**  1. Customer selects to view his/her books order  2. e-BookStore shows the books order information |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2 | Normal course of events. Book order displayed. |

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: ViewOrderTC1 **Goal**: Test the normal course of events for viewing a book order **Scenario** #: 1 **Setup**: OrderBookTC1 **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User clicks the cart icon | System redirects user to the order page. |   **Success Criteria**: Book order displayed (that being one copy of the hall001 book). |

**Use Case 8**: Update Books Order

|  |  |
| --- | --- |
| **Main Scenario**  1. Customer specifies a number of copies for an ordered book  2. e-BookStore updates the total cost for the book  **Alternatives**  1. a. New number of copies is less or equal to 0  1. a. 1. e-BookStore updates the total cost for the book to 0  1. a. 2. e-BookStore removes the book from the books order |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2 | Normal course of events. Number of copies updated for selected book. |
| 2 | 1-1a.1-1a.2 | Total cost for book set to 0, book removed from the order table. |

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: UpdateOrderTC1 **Goal**: Test the normal course of events for updating the number of copies in an order (n > 0) **Scenario** #: 1 **Setup**: OrderBookTC1; ViewOrderTC1  **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User writes "3" in number field & clicks Update | System updates total cost and shows "$119.85". |   **Success Criteria**: Order is updated |
| **Test Case**: UpdateOrderTC2 **Goal**: Test deleting a book from an order by updating its copy number to 0 **Scenario** #: 2 **Setup**: OrderBookTC1; ViewOrderTC1  **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User writes "0" in number field & clicks Update | System updates total cost to 0 and removes book from list. |   **Success Criteria**: Order list is emptied |

**Use Case 9**: Check Out

|  |  |
| --- | --- |
| **Main Scenario**  1. Customer select to checkout  2. e-BookStore processes the books order and display the information |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2 | Normal course of events. Checkout page displayed. |

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: CheckOutTC1 **Goal**: Test the normal course of action for checking out. **Scenario** #: 1 **Setup**: OrderBookTC1; ViewOrderTC1; UpdateOrderTC1 **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User clicks "Proceed to Checkout" | System displays checkout page. |   **Success Criteria**: Checkout page displayed; total check shown is $13.54 |

**Use Case 10**: Select Language

|  |  |
| --- | --- |
| **Main Scenario**  1. Customer select a language  2. e-BookStore displays information in the selected language |  |

**Scenario Set:**

|  |  |  |
| --- | --- | --- |
| **ID** | **Event Sequence** | **Scenario Description** |
| 1 | 1-2 | Normal course of events. Language changed. |

**Abstract Test Case Set:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case**: SelectLanguageTC1 **Goal**: Test the normal course of events for changing languages. **Scenario** #: 1 **Setup**: Page loaded in English **Course of test case**   |  |  |  | | --- | --- | --- | |  | **External Event** | **Reaction** | | **1** | User clicks language selector and selects "French" | System reloads active page with all text content translated to French. |   **Success Criteria**: Language changed to French |

**Question 3**

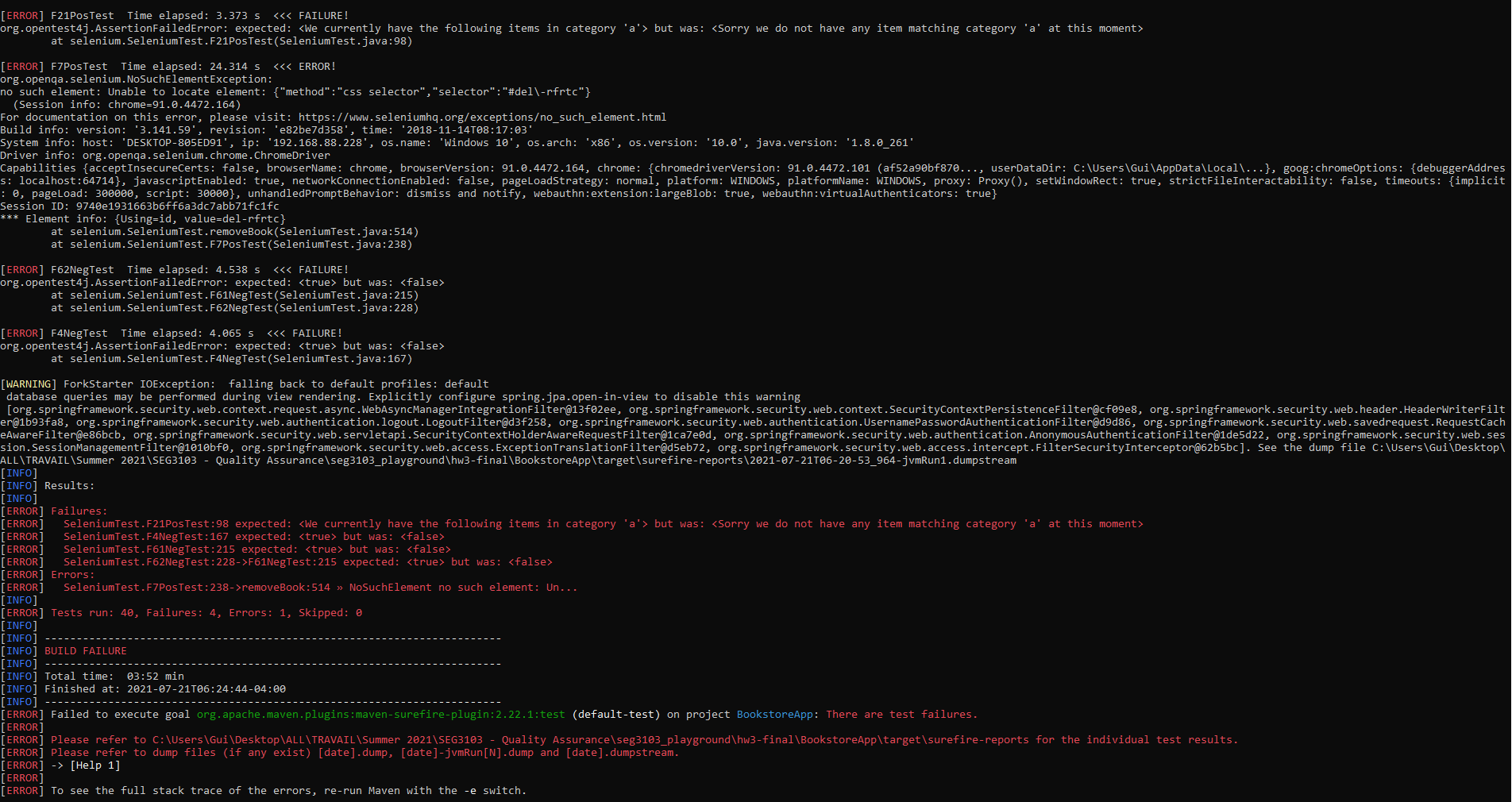
Please find all my tests in BookStoreApp/src/test/java/selenium/SeleniumTest.java.

All test-case tests use the same name they were given in Q2; functional positive and negative tests are also named appropriately (i.e., "F4PosTest" for the positive test situation for F4). Due to Java method naming rules, commas have been removed ("2.1" becomes "21").

**Implementation Issues:**

* **Test Interdependence**: To cut down on time, I do not reset the environment in between each test. This comes with some disadvantages: it is possible that runs running first will result in later tests failing (such as a test that deletes all books in the catalogue). To prevent this, I try to make sure most tests "clean up after themselves", e.g., they destroy anything they added or add back anything they removed. Some tests that create new books are exception to this; they are written in a way that allows them to be used by other tests to avoid having to repeat myself. Tests that depend on these types of tests make sure their activity will not have an effect on the test they are using. While I make do make use of calling previously-written tests several times (which, my apologies, does make the code harder to read), I make it a point to separate Functional Requirements (Q1) and Use Case (Q2) tests. I will admit I am not sure that my approach to Q3 was optimal, as it did turn into a sort of "debugging hell" near the end.
* **F3PosTest**: Verification via asserts cannot be done here without first checking the order list, which would make F3PosTest the exact same as F4PosTest. I assume it is acceptable not to have verification in this instance (we're only checking that this doesn't crash the app). The same logic applies to **OrderBookTC1** & **OrderBookTC2**.
* **F3NegTest**: Cannot be implemented as the app has no concept of "availability" for books. It is not possible to both use this feature and get a negative result.
* **F4NegTest**: I make the assumption that the order page, especially the button to checkout, should not be displayed for an empty order. As this is not the case in our app, this is an expected test failure.
* **F51NegTest:** According to Use Case 8, changing the number of copies to 0 for a book in an order should remove it from the table. This is not correctly implemented. If it were, it would render a F5.1 negative test scenario impossible: there would be no scenario where using this feature would lead to a negative result. I make the assumption that it is acceptable in this instance to test for the negative test scenario that *becomes possible* due to this incorrect implementation. Similarly, my **F51PosTest** implementation is not actually what we would want; we wouldn't be able to check for the price being reduced to $0 if the item was removed from the table. This also applies to **UpdateOrderTC2**.
* **F61NegTest**: This is another expected test failure resulting from the app letting us view and checkout an empty order.

**Running the Tests:**



**Final Tally**: 4 failures, 1 error  
**Expected**: 3 failures (F4, F6.1 & F6.2 negative scenarios)

There is an unexpected failure and an unexpected error with tests F21PosTest and F7PosTest, respectfully. After a long, long night of debugging my test suite, I think this is a good time to call it quits. I tried to fix these two errors extensively, but fixes only created additional failures and would have taken a lot more time to resolve. Had I approached this question in a better way, perhaps these types of inter-dependent errors would not have arisen.