|  |  |
| --- | --- |
| **Project:** | WhatABook |
| **Course:** | CSD310-312A Database Development and Use |
| **Description:** | Create a program for users to access the WhatABook Database and modify their accounts |
| **Date:** | 17 December 2021 |
| **Version:** | 1.0 |

**TABLE OF CONTENTS**

[**Main Menu Test** 1](#_Toc90650069)

[**Account Menu** 2](#_Toc90650070)

[**Database Tests** 3](#_Toc90650071)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 1 | **Main Menu Test** | | | |
|  | **Test Objective:** Ensure that the main menu operates as expected | **Developer:**  **Date tested:** <yyyy/mm/dd> | **Peer tester:**  **Date tested:** **2021/12/17** | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Test that the menu can handle invalid inputs | Invalid inputs do not crash the program or lead to an unexpected result | <yes/no> | Pass |
| 2 | Access each menu option to ensure that they produce the correct result | Menu will only lead to the expected result with a proper input | <yes/no> | Pass |
| 3 | Examine menu layout and wording | Menu makes sense in layout and wording | <yes/no> | Pass |
| 4 | Run View Books option and examine book list printout | All book data should be displayed in a readable format | <yes/no> | Fail |
| 5 | Run View Store Locations option and examine store print out | All store location data should be displayed in a readable format | <yes/no> | Fail |
| **Comments** | When I ran the test for viewing books, I noticed that the labels of for the books do not match the data that is output. Instead, the Id is where the book name should be and everything else is shifted down one. To Correct this I would look at line 58 of your code and change it to this: print("Book Name: {}\n Author: {}\nDetails {}\n".format(book[1], book[2], book[3])) as it appears you grabbed the data from the wrong part of the array.  I believe that as per the project description we were supposed to include the store hours as part of the store information which is not included in your store location print out | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 2 | **Account Menu** | | | |
|  | **Test Objective:** Ensure that the account menu operates as expected and without errors. | **Developer:**  **Date tested:** <yyyy/mm/dd> | **Peer tester:**  **Date tested:** **2021/12/17** | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Test that the menu can handle invalid inputs | Invalid inputs do not crash the program or lead to an unexpected result | <yes/no> | Pass |
| 2 | Test that the Account Menu can only be accessed with a valid account number | Invalid inputs do not crash the program or lead to an unexpected result | <yes/no> | Pass |
| 3 | Run the View Wishlist option and examine print out | Wishlist data should be accurate to the profile | <yes/no> | Pass |
| 4 | Run the Add Book to Wishlist option and try to add an invalid book\_id. | Program should not allow an invalid book\_id to be added to the wish list and should return to the account menu if the id is invalid | <yes/no> | Fail |
| 5 | Run the Add Book to Wishlist option, add a valid book, and then run the View Wishlist option | The new book should be displayed when the wish list is updated. | <yes/no> | Pass |
| **Comments** | Small nitpick on the account menu but when entering an option, it says “enter a wishlist” which does not make sense from a user perspective. Also, I think you left a test print statement in the add a book section since the SQL command prints out when adding a book  There does not appear to be any input validation on the add a book to wishlist part and the code errors out when an invalid input is entered. Without input checking, your program can be vulnerable to SQL injection attacks which can lead to data leaks or tampering with your database. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test 3 | **Database Tests** | | | |
|  | **Test Objective:** Ensure the program can handle additional data added to the database | **Developer:**  **Date tested:** <yyyy/mm/dd> | **Peer tester:**  **Date tested:** **2021/12/17** | |
| **Step** | **Action** | **Expected results:** | **Developer pass/fail** | **Tester pass/fail** |
| 1 | Run the initialization script in the for the database | Database initialization script runs without error and initializes the required data in the database | <yes/no> | Pass |
| 2 | Add more books to the database and run the View Books, and User Account Wish list commands | Added Books should appear in the book list and should be able to be added to wish lists | <yes/no> | Pass |
| 3 | Add another location to the database and run the View Store Locations option | New location should also be displayed in the output | <yes/no> | Pass |
| 4 | Delete all user wish list data from the database and ensure no data is outputted | User wish list should be blank and with out errors | <yes/no> | Pass |
| 5 | Add more users to the database and ensure their accounts can be accessed | Added user accounts should be able to be used as normal including wish list functions | <yes/no> | Fail |
| 6 | Fill a User’s wish List with all available books and then try to add another | User should not be able to add any more books and no books should be listed to add | <yes/no> | Fail |
| **Comments** | Program does not use the database to find valid user Ids, but it is instead hard coded. These needs fixed as the account menu will not work with more than three customers and will error out with less than 3 customers.  The Wishlist will allow a user to add a book that is not listed in the available books and then creates a duplicate book in the users wishlist since it already exists there. | | | |