**How to Run and Test Librarian Module:**

**Navigating the Librarian Module:**

Select the Librarian radio button option to redirect to a sign in page, which prompts the user for a username and password.

Valid Librarian login data can be found in the resources/admin/ Librarians.json file.

A successful login will redirect the user to a welcome Librarian page which displays three options that the user can select.

**Add Book:**

If the add book option is selected, the user will be prompted to fill a form with the book’s information. All fields must have a valid value, otherwise an error message will display, with a prompt of valid inputs when the register button is pressed. The ISBN value must be unique. If the book is successfully added, a success message will display.

The user now has the option to go back or add another book.

**View Books:**

If the View Books option is selected, the system will display a list of currently added books in the system with all their information except borrowed/waiting lists. A go back button at the bottom of the screen allows the user to return to the options menu for Librarian.

**Delete Books:**

If the Delete Books option is selected, the system will display a list of currently added books in the system with all their information except borrowed/waiting lists, with a delete button alongside each book. If the book is successfully deleted, a success message will display. The user now has the option to go back or delete another book.

**Testing the Librarian Module:**

There are four testing classes in the LibrarianTest.java file. One for each important functionality of the Librarian class.

**librarianLoginTest**:

Run this test to try the login functionality for an Librarian. This class contains two tests.

**testLibrarianLoginSuccess**:

This test checks for a successful login.

**testLibrarianLoginFail**:

This test checks for a failed login.

**viewBooksLibrarianTest**:

Run this test to try the getBooks method of the Librarian class. This class contains two tests.

**testGetAllBooksSuccess**:

This test checks for the books array returned.

**testGetBookSuccess**:

This test checks for one of the book object returned.

**addBooksLibrarianTest**:

Run this test to try the addBook method of the Librarian class. This class contains two tests.

**testSuccessBookAdd**:

This test checks for the “success” value returned, once book is added.

**testUnsuccessfulBookAdd**:

This test checks for the “book exists” value returned, if the book is not added.

**validateIDBookAdd**:

This test checks for values in the field that are other than numbers.

**validateNameBookAdd**:

This test checks for values in the field that are other than alphabets.

**validateAuthorBookAdd**:

This test checks for values in the field that are other than alphabets.

**validatePagesBookAdd**:

This test checks for values in the field that are other than numbers.

**validateStockBookAdd**:

This test checks for values in the field that are other than numbers.

**returnBooksLibrarianTest**:

Run this test to try the returnBook method of the Librarian class. This class contains two tests.

**testSuccessBookReturn**:

This test checks for the true value, once book is returned.

**testUnsuccessfulBookReturn**:

This test checks for the false value returned, if the book is not returned.

**borrowBooksLibrarianTest**:

Run this test to try the borrowBook method of the Librarian class. This class contains two tests.

**testSuccessBookBorrow**:

This test checks for the true value, once book is borrowed.

**testUnsuccessfulBookBorrow**:

This test checks for the false value returned, if the student is in waiting list.

**deleteBookLibrarianTest**:

Run this test to try the deleteBook method of the Librarian class. This class contains one tests.

**testDeleteSuccess**:

This test checks for a true value returned, if the deletion was executed.