SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

## School of Computing

## SRM IST, Kattankulathur – 603 203

**Course Code: 18CSC206J**

**Course Name: Software Engineering and Project Management**

| | |
|---|---|
| **Experiment No** | 10 |
| **Title of Experiment** | Developing a Testing Framework/User Interface |
| **Name of the candidate** | Shaurya Singh Srinet |
| **Team Members** | Shounak Chandra, Parth Galhotra |
| **Register Number** | RA2111032010006, RA2111032010026, RA2111032010029 |
| **Date of Experiment** | 30.3.23 |

## Mark Split Up

| S. No | Description | Maximum Mark | Mark Obtained |
|---|---|---|---|
| 1 | Exercise | 5 | |
| 2 | Viva | 5 | |
| **Total** | | **10** | |

**Staff Signature with date**

**Aim:**

To develop the testing framework and/or user interface framework for the Library Management System.

**Team Members:**

| S No | Register No | Name | Role |
|---|---|---|---|
| 1 | RA2111032010006 | Shaurya Singh Srinet | Rep/Member |
| 2 | RA2111032010026 | Shounak Chandra | Member |
| 3 | RA2111032010029 | Parth Galhotra | Member |

**Executive Summary:**

This Module is used to manage the list of books that have been stored in database. It displays a tabular view of all the books owned by the library i.e the User. It also Gives the option to Delete A certain book from the database. A search bar may be added for additional user convenience in the same page. The module to Issue and return books has also been integrated with the current one which allows the user to issue and return books. It also tells us about the availability of the books in their current status and enables us to distinguish between them. The functionality to add books is better left as another module due to its dependencies and requirements. Further background design and QOL features may also be added to stylize the page if time permits.

## Test Plan

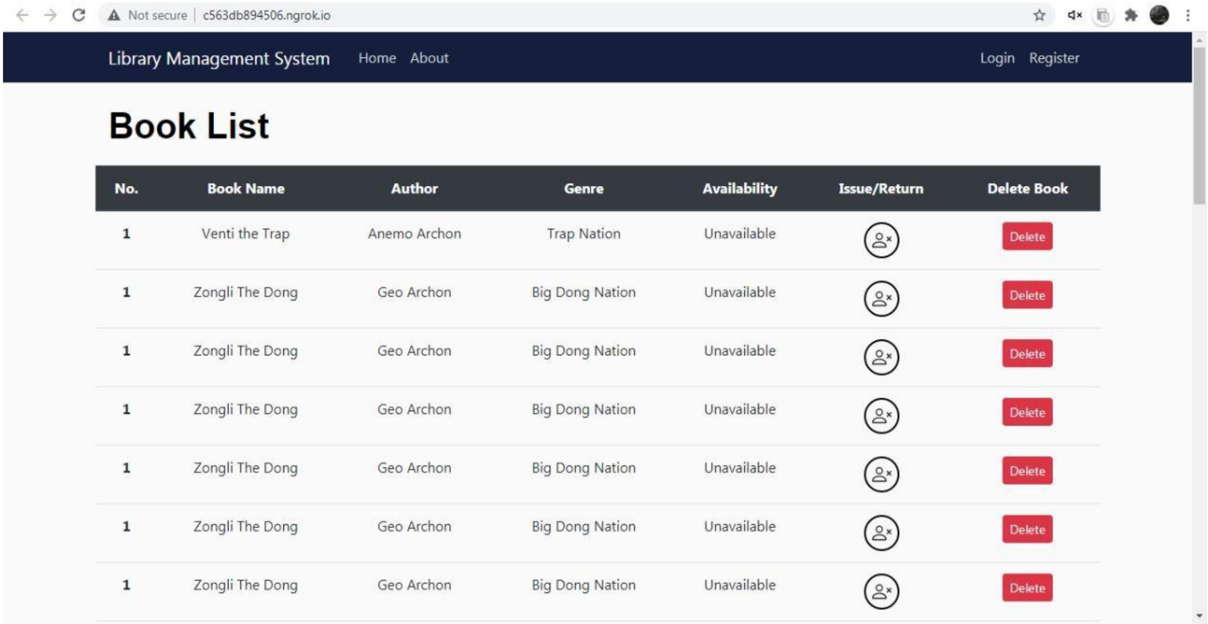
### Scope of Testing

The scope of testing for an online library management system project includes creating distinct product users based on their roles and permissions, authenticating users at their login, providing a list of books that users can borrow, and providing facilities to reserve books that are available. The system should also provide a status page for all users to view books reserved by them and allow them to cancel reservations made earlier. Additionally, it should provide a status page for all users to view books borrowed by them, their individual due dates, and their individual penalties if any. The system should also provide an interface for users to view and edit their own profile, adjust account settings such as passwords, and reset their password in case they forget it. Staff should be provided with an interface to add or delete books.

**Types of Testing, Methodology, Tools**

| Category | Methodology | Tools Required |
|---|---|---|
| Functional Requirements | Use case testing to ensure all required functionality is working. | Test management tools (e.g. TestRail), Test automation tools (e.g. Selenium), Debugging tools (e.g. Visual Studio Debugger) |
| Non-functional Requirements | Performance testing to ensure the system can handle expected loads. | Load testing tools (e.g. Apache JMeter), Security testing tools (e.g. OWASP ZAP), Usability testing tools (e.g. UserTesting.com) |
| User Interface Testing | Manual testing to ensure the user interface is intuitive and easy to use. | Documentation tools (e.g. Confluence), Screen capture tools (e.g. Snagit), Word templates for test case documentation. |
| Integration Testing | API testing to ensure that API calls to external systems work correctly. | Programming languages (e.g. Java), Frameworks (e.g. Spring), Libraries (e.g. Apache HttpClient), Databases (e.g. MySQL), Tools (e.g. Postman) |
| Category | Methodology | Tools Required |
| Acceptance Testing | User acceptance testing to ensure that the system meets user requirements. | Collaboration tools (e.g. Microsoft Teams), Test management tools (e.g. Zephyr), Test automation tools (e.g. Selenium WebDriver) |
| Book Listing and Availability | Automated testing to ensure that the book listing and availability are accurate and up-to-date. | Test automation tools (e.g. Selenium WebDriver), Programming languages (e.g. Python), Libraries (e.g. BeautifulSoup) |
| Issue and Return of Books | End-to-end testing to ensure that the system can properly issue and return books. | Test management tools (e.g. TestRail), Test automation tools (e.g. Selenium), Debugging tools (e.g. Visual Studio Debugger) |

## UI Design Example



Book List

| No. | Book Name | Author | Genre | Availability | Issue/Return | Delete Book |
|---|---|---|---|---|---|---|
| 1 | Venti the Trap | Anemo Archon | Trap Nation | Unavailable | | Delete |
| 1 | Zongli The Dong | Geo Archon | Big Dong Nation | Unavailable | | Delete |
| 1 | Zongli The Dong | Geo Archon | Big Dong Nation | Unavailable | | Delete |
| 1 | Zongli The Dong | Geo Archon | Big Dong Nation | Unavailable | | Delete |
| 1 | Zongli The Dong | Geo Archon | Big Dong Nation | Unavailable | | Delete |
| 1 | Zongli The Dong | Geo Archon | Big Dong Nation | Unavailable | | Delete |
| 1 | Zongli The Dong | Geo Archon | Big Dong Nation | Unavailable | | Delete |

## Result

Thus, the testing framework/user interface framework has been created for the Library Management System.