Library Management System

Test Plan Document

IS5113: Software Quality Assurance

Group 21

Index Number	Name
20APC4656	DSC Wijesinghe
20APC4554	LH Udani
20APC4672	JM Jarshan
20APC4574	KVSA Dias
20APC4561	KLE Isurika

Table of content

1. Scope of Testing

The scope of this project covers the testing of core functionalities, focusing on functional, performance, and usability aspects of the platform. The following components will be tested in-depth:

I.   Member Registration
Ensure users can register as members with valid data and appropriate validation checks are performed.

Input Data Validation

- Ensure required fields (Name, Email, Username, Password) are filled.
- Test valid and invalid inputs (check for valid email formats, strong password rules).
- Validate uniqueness for usernames or emails to prevent duplicates in the database.

Form Submission

- Successful registration upon entering valid data.
- Check for appropriate error messages when required fields are missing or incorrectly filled.
- Ensure proper redirection or feedback upon successful registration (confirmation email).

Database Interaction

- Verify that valid member data is correctly stored in the database.
- Ensure no incomplete or invalid entries are saved.
- Check for any SQL injection or data breach vulnerabilities.

II.   Member Profile Management
Allow users to view, update, and delete their profiles with correct data validation and security measures.

Profile Viewing:

- Ensure that the user can view their personal profile details (Name, Email, Registration Date).
- Ensure that data retrieval from the database is accurate and complete.
- Implement user-specific access control to prevent unauthorized viewing of profiles.

Profile Editing:

- Validate changes to the profile (updating the email, password, or address).
- Test input validation for edits (valid email format, new password rules).
- Ensure changes are reflected accurately in the database.
- Prevent unauthorized profile modifications by implementing proper authentication.

Profile Deletion:

- Ensure that the member's profile is completely deleted, including associated data (borrowed book history).
- Test confirmation steps to prevent accidental deletions.
- Verify that after deletion, the user can no longer access the system.

III.   Book Management

Ensure that administrators or authorized users can add, edit, view, and delete books while validating data input and managing database interactions efficiently.

Adding New Books:

- Validate all required fields (Title, Author, ISBN, Category).
- Test for correct input validation (valid ISBN formats, non-empty fields).
- Ensure that adding a new book creates a new record in the database.
- Check for edge cases like duplicate ISBN entries or long book titles.

Editing Book Details:

- Ensure that all book details can be updated correctly.
- Validate updates (change in category, author name, or book availability).
- Test database consistency after editing a book (no duplicated records or partial updates).

Viewing Book Records:

- Test the search functionality for locating books by different criteria (Title, Author, ISBN).
- Verify that viewing a book displays all relevant information (description, availability).
- Ensure performance is maintained while searching through large datasets (fast search, pagination).

Deleting Books:

- Ensure that deleting a book removes it from the system and the database.
- Verify that deleted books no longer appear in search results or listings.
- Check for any cascading effects (ensuring no reference to deleted books in member histories).

IV.   Report Handling
Ensure that reports are generated accurately based on real-time data.

2. Tested features

I. Member Registration:

- Validate mandatory fields (e.g., name, email).
- Check for unique usernames or email addresses.
- Password strength validation.
- Success/failure scenarios for form submission.

II. Member Profile Management:

- Editing member profile details.
- Viewing member information.
- Deleting member profiles and handling associated book records.

III. Book Management:

- Adding a new book (including validation of fields : title, author, ISBN).
- Editing book details.
- Viewing and searching books.
- Deleting book records.

IV. Report Handling:

- Generate reports for member activity.

3. Testing approach and strategy

I. Unit Testing

Objective: Test individual components (methods and classes) to ensure they function correctly in isolation.

- **Scope**:
    - Each function, method, or class will be tested individually to ensure correctness.
    - This includes testing components like registration logic, book management functions, and report generation.
- **Tools**:
    - **JUnit**:
        - Used JUnit for writing and executing unit tests.
        - Test different aspects of the platform like validation logic, CRUD operations for books, and member management functionality.
        - Focus on ensuring that functions produce the correct output given a specific input.
- **Examples**:
    - **Member Registration**:
        - Test validation methods to ensure they check for valid input (email format, password strength).
        - Test for success or failure of member registration based on valid or invalid data.
    - **Book Management**:
        - Test methods for adding, editing, and deleting book records, ensuring they perform as expected.
    - **Report Generation**:
        - Test logic behind report creation, such as filtering by member activity or date range.

II. Integration Testing

Objective: Test the integration of different modules to ensure that they work together as expected.

- **Scope**:
    - Test the interaction between various components, such as linking **Member Management** with **Book Management** and ensuring that data flows correctly between modules.
    - Verify that modules communicate and integrate properly with external dependencies (databases or APIs).
- **Tools**:
    - **Mockito**:
        - Used Mockito to mock external dependencies during integration tests (e.g., database interactions, HTTP calls).
        - This allows testing integration without relying on external systems or live databases.

- **Examples**:
  - **Member and Book Management**:
    - Ensure that when a member borrows a book, the book's status updates correctly and is reflected in the member's profile.
    - Test the integration of profile deletion and ensure that related borrowed books are handled appropriately.
  - **Report Generation**:
    - Test the interaction between book and member data when generating reports. For example, borrowed books should appear correctly under the corresponding

III.   UI Testing

Objective: Ensure the user interface behaves as expected and all user interactions function properly.

- **Scope**:
  - Test the user interface, including form submissions, navigation between pages, and display of error messages.
  - Verify that the UI reflects changes in the system (e.g., books being added or deleted).

- **Tools**:
  - **Selenium**:
    - Used Selenium to automate testing of the user interface.
    - Perform tests on user interactions, such as submitting forms, navigating to different sections, and verifying that UI components display the correct data.
- **Examples**:
  - **Form Submissions**:
    - Used Selenium to automate the process of filling out and submitting registration forms.
    - Verify that after submitting valid data, the correct success message appears and the user is redirected to the appropriate page.
  - **Navigation**:
    - Test the UI for smooth transitions between pages, such as navigating from the home page to the member profile or book list.

IV.   System Testing

Objective: Ensure the e-library management platform functions correctly when all components are integrated.

- **Scope**:
- Test critical features: **Member Registration**, **Profile Management**, **Book Management**, and **Report Generation**.
- Validate end-to-end workflows and ensure the system handles normal and edge cases.

- **Approach**:
  - Simulate user actions (admin and members) to test functionalities and workflows.
- **Examples**:
  - **Member Registration**: Test adding new members and check for proper data saving and login.
  - **Profile Management**: Verify editing, viewing, and deleting member profiles.
  - **Book Management**: Ensure books can be added, edited, viewed, and deleted.
  - **Report Generation**: Confirm that reports for members and books generate accurately and filter correctly.

4. Testing tools

- **JUnit**:

Use Case: Validate the functionality of core features, like registration logic, book management methods, and report generation, ensuring they behave as expected in isolation.

- **Selenium**:

Use Case: Automate tests that verify user interactions with the e-library management platform, ensuring that the registration forms, book management pages, and report generation work correctly on the front end.

5. Roles and responsibilities

Roles	Responsibility	Group Member
Frontend Development	Implements the user interface for the e-library platform, Design and implement the UI for registration, book management, and reports generate.	KVSA Dias KLE Isurika LH Udani
Backend Development	Responsible for designing and implementing the key features, **Member Registration**, **Member Profile Management**, **Book Management**, **Report Generation.**	JM Jarshan DSC Wijesinghe
Testing	Writing test cases and implement testing activities, ensuring the platform meets functional, integration, and performance standards.	JM Jarshan DSC Wijesinghe LH Udani KVSA Dias KLE Isurika
Documentation	Creates and maintains comprehensive documentation for the project, including design, user guides, and testing results. Test plan document, Test Execution Report, Code Coverage Report, **Test Documentation**, Final Project Report and presentation.	KVSA Dias LH Udani KLE Isurika JM Jarshan DSC Wijesinghe

6. Project Timeline

I. Gantt Chart

Gantt Chart

Task	Weeks	Time duration (Weekly)														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Requirement gathering & analysis	1	■														
System Design	2		■	■												
Implementation	10			■	■	■	■	■	■	■	■	■	■			
Documentation	2			■	■											
Learning testing technologies	4				■	■	■	■								
Test case Implementation	5							■	■	■	■	■				
Testing	3											■	■	■		
Final Report	2														■	■