**Date: 01/01/2024**

**HCL LIBRARY MANAGEMENT SYSTEM**

1. **Objective/Aim:**

* This project provides a friendly environment to maintain details of books and library members.
* The main purpose of the project is to provide easy use of the system to prepare different reports.
* The system can be used by new or existing users to manage its books, books borrowing, insertion and monitoring.
* It should allow users to get about new edition or addition of books.
* It should ensure the data and privacy of library users.
* It should allow user to search for a particular book.
* It should allow user to issue a book and view all the categories, languages.
* It should allow librarian to check the issued book and check all the books in system.

1. **Scope:**
2. **Within the scope:**

* Functional testing
* External interface
* Login Management
* Account Management
* User Support
* Searching Management

1. **Out of scope:**

* Automation testing
* Non-functional testing
* Stress testing
* Performance testing

1. **Test strategy:**
2. **Level of testing:**

* **Unit testing**: The developer has to validate the code with respect to syntax and semantic.
* **Integration testing**: User is able to integrate with external applications such as Eclipse IDE.
* **System testing:** User should able to perform end to end operations like login into account, addition of new book, search for particular book, issue a book, view book details, get update about books.
* **User acceptance testing:**
  + It should allow user to perform or check with real world scenario.
  + It should fulfill the user’s expectations.

1. **Types of testing:**

* **White box testing**: The validation of code should be done.
* **Black box testing:**

**Functional testing**:

* **Smoke testing**: Development lead need to validate the developed application and check ready for detailed testing.
* **Sanity testing**: Test lead need to check the basis functionalities of software and check application is ready for detailed testing.
* **Retest testing:** Once the defect is closed test engineer should perform testing in order to confirm that problem is fixed.
* **Regression testing**: It verify that modification of a part of code has not caused any server effect in the software.

1. **Test design technique:**

* **White box testing technique:**
* **Statement testing**: Testing all the statements at least once in order to avoid syntax and semantic error.
* **Branch testing:** Test all the branches (if and else) at least once.
* **Black box testing technique**:
* **ECP and BVA:** In the admin’s account creation, username, password we should ECP and BVA.
* **Decision table:** At the time of issuing book and searching book it should show each and every information of book. It should also show the new edition of book if available.
* **State transition table**: It should allow user to perform different end to end operations like adding new employee, viewing books, searching book etc.

1. **Terminologies:**

* **Index:** The process of maintain a record of library resources including books and other materials.
* **Authentication:** It is the process of verifying the identity of user before providing it any access to the library resources.
* **Renewed:** It is used to extend the time period of issued book, if it is not reserved by any other user.
* **Reports**: This gives a detailed information about the usages of library resources and books.

**e. Configuration management:** This tool allow user to store all the project related documents at the one place like repository.

**f. Area planned for automation tool:** N/A (Not applicable)

1. **List of automation tool**: N/A (Not applicable)
2. **Exit and Entry criteria:**

**Entry criteria:**

* Requirement should be approved.
* Application should be ready.
* Test cases should be reviewed.
* Test environment should be ready.
* Test data should be ready.

**Exit criteria:**

* Code should not have any syntax and semantic errors.
* Integration should work fine.
* Application should allow to perform different end to end operations.

1. **Test environment:**

* **Server:** Need a database server.
* **Test tool:** Develop a test tool which can auto generate the test case results.
* **Network:** Set up a LAN gigabit and an internet lines with speed of at least 25 mbps.
* **Computer:** Window 10, RAM 8 GB, CPU 4GHz, Eclipse IDE.
* **MS tools:** Test preparation, test case execution, defect management.

1. **Test deliverable:**

* Requirement
* Test cases
* Test scenario
* Test plan
* Test data
* Test strategy
* Defect report
* RTM

1. **Hiring and training:**

* We need to hire a manual tester with 2+ years’ experience.
* We need to hire a security tester with 6+ years’ experience.
* We need to provide the training on the security feature for all the project employees.
* We need to provide the training on the domain.
* We need to provide the training all the members of team not only fresher.

1. **Risk and Mitigation:**

**Risk:**

* Security breaches and unauthorized access should not allow.
* Technical glitches and connectivity when user is trying to issue a book.
* Not compatible with different browsers like Microsoft edge, Chrome.
* User not able to access the book and library resources.
* Not allow the update and addition of books.

**Contingency:**

* Backup server for uninterrupted access.
* Secure data backups to prevent data loss.
* A strategy for technical glitches and protocol for handling security breaches.
* Having clear communication channels and support system to face challenges.

**Mitigation:**

* Implement strong encryption for data security.
* Regular update and patching software to address vulnerabilities.

1. **Schedule:**

|  |  |
| --- | --- |
| **Activity** | **Duration** |
| Requirement analysis | 25 man hours |
| Test case planning | 20 man hours |
| Test case creation | 20 man hours |
| Test case review | 30 man hours |
| Test case execution | 20 man hours |
| Retest | 30 man hours |
| Regression testing | 25 man hours |
| Test closure | 25 man hours |

1. **Roles and Responsibilities:**

|  |  |
| --- | --- |
| **Roles** | **Responsibilities** |
| Requirement analyst | * Capturing requirements * Preparation of functional document * Review of functional document * Addressing the issue of team |
| Test lead | * Understanding requirement * Preparing of test case * Reviewing of test cases * Preparing of test plan * Reviewing of test plan |
| Test engineer | * Creating detailed test cases * Design specification * Documenting issue or bugs * Provide feedback on testing process |
| Test manger | * Overall planning and coordinating * Ensure the functionality of project * Risk Management * Issue Management |
| Performance Test engineer | * Preparation of test scenario * Review test scenario * Create test scripts using performance * Executing test scripts * Analysis performance results |
| Automation test engineer | * Preparation of test scenario * Review test scenario * Create test scripts using record * Executing test scripts * Analysis test defects |
| Security analyst | * Implement the security practices * Ensure the safety of application |

1. **Approval information:**

|  |  |
| --- | --- |
| **Document** | **Approving roles** |
| Requirement | SME/Stakeholders |
| Test plan | Project manager |
| Test case | Test lead |
| Defects | Developer |
| Final review | QA manager |
| Product release | Product manager |

1. **Assumptions:**

* Reliable internet connection for user.
* Security measure to prevent data loss.
* Compatibility with common browsers.
* Regular system maintenance to ensure smooth functioning.
* User support of troubleshooting issues.
* Operating System compatible.

1. **Test Matrix:**

* Passed test cases percentage (no. of passed test cases/ no. of test cases executed) \*100
* Failed test cases percentage (no. of failed test cases/ no. of test cases executed) \*100
* Fixed defect percentage: (defects fixed/ defects reported) \*100
* Accepted defect percentage: (defects accepted by dev team / total defects reported)
* Defect deferred percentage: (defects differed for future release/ total defects reported)