GenAi: Library Management System

Name: Jake Chapman

Student Email: [21427254@stu.mm.ac.uk](mailto:21427254@stu.mm.ac.uk)

Table of Contents

[1. Introduction 3](#_Toc149854022)

[2. Test Objectives 3](#_Toc149854023)

[3. Scope of Testing 3](#_Toc149854024)

[3.1 In Scope 3](#_Toc149854025)

[3.2 Out of Scope 4](#_Toc149854026)

[4. Test Criteria 4](#_Toc149854027)

[Section One: Book Management 4](#_Toc149854028)

[Section Two: Member Management 5](#_Toc149854029)

[Section Three: Loan Management 5](#_Toc149854030)

[Section Four: Fine Management 5](#_Toc149854031)

[Section Five: Performance 6](#_Toc149854032)

[Section Six: Usability 6](#_Toc149854033)

[5. Test Environment 6](#_Toc149854034)

[Hardware 6](#_Toc149854035)

[Software 7](#_Toc149854036)

[Network 7](#_Toc149854037)

[6. Resource Allocation 7](#_Toc149854038)

[Human Resources 7](#_Toc149854039)

[Hardware Resources 8](#_Toc149854040)

[7. Test Data Requirements 8](#_Toc149854041)

[8.Test Schedule 9](#_Toc149854042)

[9. Testing Strategy 9](#_Toc149854043)

[10.Testing tools 10](#_Toc149854044)

[11. Test Deliverables 11](#_Toc149854045)

[12. Test Responsibilities 11](#_Toc149854046)

[13. Risks and Assumptions 12](#_Toc149854047)

[Risks 12](#_Toc149854048)

[Assumptions 12](#_Toc149854049)

[Contingency Plans 12](#_Toc149854050)

[14.Approvals 13](#_Toc149854051)

[15. Appendices 13](#_Toc149854052)

# 1. Introduction

The system under test is a Library Management System designed for smaller local libraries. Its primary purpose is to automate and streamline various operations, thereby reducing the risk of human error. This includes tasks such as cataloguing books, managing member records, processing loan requests, and calculating fines. The objectives of the test plan are to verify that the product, developed by junior developers, continues to meet, or exceed the established standards that were in place before the introduction of the GenAI model, and to assess the system's capacity to produce robust, secure, and scalable code.

# 2. Test Objectives

**1. Test Objective**: Ensure that the system allows the addition of books with all required attributes, including the generation of new book IDs.

**2. Test Objective:** Confirm that the Library Management System (LMS) enables librarians to efficiently manage member records and validates the format of email addresses.

**3. Test Objective:** Validate that librarians can successfully issue loans and that loan and due dates are automated.

**4. Test Objective:** Verify that the system automates fine calculations based on library-defined conditions and allows fines to be marked as paid and updated in the database.

# 3. Scope of Testing

## 3.1 In Scope

**Book Management:** The tests will validate that librarians can add new books to the system with correct attributes such as: Author, ISBN, Genre. Adding a new book to the system will automatically update the book id number.

**Member Management:** It should validate the librarians can control member records; delete, add, update. In addition, members emails need to be in valid format.

**Loan management:** The procedure will verify that the system can be used by librarians to lend books to members and to automatically calculate loan terms and due dates.

**Fine management:** The process will ensure that fine calculations are automated to the conditionals of the rule and to mark fines as completed.

**Performance:** The response time for queries will be tested to ensure it does not exceed 2 seconds

**Usability:** Testing will make sure that it usable to librarians and if errors are to occur, they should be clear and understandable to the user

## 3.2 Out of Scope

**User Authentication:** The system doesn’t need to check whether if the LMS allows for librarian to login with username and password and block users after three unsuccessful logins as it not implemented.

**Security:** Validation will not be made sure that passwords are encrypted to the right standard and that users do not have access to unauthorised actions as this system is not implemented

**Operating Systems**: Tests shouldn’t be done to validate whether it can run any other operating systems such as link or OS. Only should run on windows 10 or above.

**External Adaptations** – The system doesn’t not need to be tested for adaptations of other library systems. It is developed using Java and SQLite as they are already implemented in the library.

# 4. Test Criteria

**General Entry Criteria for All Sections:**

The Library Management System (LMS) is operational and properly configured in the test environment.

*Entry and exit criteria numbers do not always match to each other (number to number.) Please refer to the appendices 2) at the end of the document these will help you understand and give a clear understanding of what the entry and exit criteria will look like.*

## Section One: Book Management

**Entry Criteria**:

1. Ensure that BookID, Title, Author, and Status values exist in the SQLite database schema in the LMS.

2. Ensure that the search system for books is working

3. ISBN Validation system works

**Exit Criteria:**

1. Librarians can add a book to the system with the correct criteria.

2. The system generates a unique BookID for a new book entry.

3. Validate that the auto-increment feature works when adding a book.

4. Can see all the books in the system

## Section Two: Member Management

**Entry Criteria:**

1. Ensure that MemberID, FirstName, LastName, and dateJoined values exist in the database schema.

2. Check that the email verification works correctly.

**Exit Criteria:**

1. Librarians can successfully control member records in all formats.

2. Emails with correct abbreviations will be added to the system, while incorrect abbreviations will prompt the user.

## Section Three: Loan Management

**Entry Criteria:**

1. Ensure that the LMS has valid librarian and member accounts set up in the test environment.

2. Confirm that books are available and in the correct status (e.g., not already on loan) in the system.

3. Verify that system settings for loan management are properly configured.

**Exit Criteria:**

1. The system should automatically set the loan duration and calculate due dates accurately based on predefined rules.

2. Librarians should be able to issue loans to members successfully.

## Section Four: Fine Management

**Entry Criteria:**

1. Confirm that the loan management and date tracking functionalities are working correctly, as fine calculations depend on loan due dates.

2. Verify that the system settings for fine management, including the specified fine rules, are properly configured.

**Exit Criteria:**

1. The system should automatically calculate fines for late returns based on the defined rules, including a £1 fine for 1 to 7 days late, £5 for 8 to 14 days late, and £1 additional for each day after 14 days.

2. Librarians should be able to mark fines as paid, and the system should update the database, accordingly, reflecting the updated payment status.

## Section Five: Performance

**Entry Criteria:**

1. The database is populated with representative data that simulates real usage.

2. Test environment resources (hardware, network) are available and functioning.

3. No performance-enhancing or caching mechanisms are enabled for the LMS.

**Exit Criteria:**

1. Response time for queries should be measured during testing.

2. Response time for queries should not exceed 2 seconds.

## Section Six: Usability

**Entry Criteria:**

1. Training materials for librarians are available.

**Exit Criteria:**

1. Usability testing is performed to validate that the system is user-friendly and requires minimal training.

2. Error messages should be clear, instructive, and localized.

# 5. Test Environment

## Hardware

**Storage:** Preserve the database and LMS files in no less than 12 GB of accessible storage space to avoid storage problem.

**Graphics:** A separate graphics card with sufficient video memory is highly recommended if JavaFX components require a great deal of graphics processing power. Integrated graphics are acceptable for basic LMS operations.

**Processor (CPU):** To meet the benchmarks for Java and JavaFX, use a suitable multi-core processor. “1) *Intel Pentium 4, Intel Centrino, Intel Xeon, or Intel Core Duo (or compatible) 1.8 GHz minimum”* Make sure your CPU can handle SQLite's minimum requirements, or more.

**Memory (RAM):** For JavaFX to function, the system must have at least 1)“*512 MB of RAM, 2 GB is advised.”* With Java and SQLite, a minimum of 4 GB of RAM is advised for best performance.

## Software

**Operating System:** Ensure that the test environment uses Windows 10 or a more recent version.

**System Architecture:** Implement a three-tier architecture, utilizing JavaFX for the User Interface, Java for Core Services, and SQLite for the database.

**User Categories and Roles:** Establish a hierarchy of user roles to control feature access, including roles for librarians, administrators, and various member types (junior and adult).

**JDK:** 1)*“JDK 6 Update 13 minimum (*[*JDK 6 Update 14*](https://www.oracle.com/java/technologies/javase-downloads.html)*recommended).”* For JavaFX

## Network

**Connection and Internet:** Ensure that the system is connected to the LAN system so it can cooperate with the SQLite database.

**Cloud Backup**: Assumptions of the system that there is cloud backup however these features are not implemented so would out of scope for testing.

# 6. Resource Allocation

## Human Resources

**Software Developers:**

* **Role:** Collaborate to fix identified defects, including database-related issues.
* **Responsibilities:** Review and address defect reports, perform database setup and maintenance, optimize queries, and manage data backup, create UI.

John Smith - Automated Tester [Responsible for Automated Testing] (30 hours per week) – Conduct in helping the test if they have limited resources.

Peter Frost – UI designer [Responsible for UI design] (20 hours per week)

* **Training Requirements:** Proficiency in the Java and JavaFX codebase, familiarity with the test plan, and database management tools such as SQLite.

**Test Team:**

* **Role:** Design, plan, execute, and report on the tests.
* **Responsibilities:** Create and execute test cases, report defects, and compile test documentation. Sarah Miller - Test Analyst [Responsible for Test Execution] (35 hours per week) – Will need approval for over time.
* **Training Requirements**: In-depth understanding of the Library Management System (LMS) and the testing environment, Java, and JUnit

**Librarians (End Users):**

* **Role:** Participate in usability testing.
* **Responsibilities**: Evaluate the LMS user interface, provide feedback on user experience, and suggest improvements. Emily Davis - Librarian [Responsible for Usability Testing] (3 hours per week) – Will only be allowed this time.
* **Training Requirements:** Familiarity with library operations and basic computer skills.

## Hardware Resources

**Network Infrastructure:**

* Role: Ensure network connectivity within the test environment
* Responsibility: Connecting servers and workstations

**Workstations:**

* Role: Ensure can connect with LMS
* Responsibility: Can be used by test team, developers, and librarians for their given roles

**Database Server:**

* Role: Host/connect database for the LMS
* Responsibility: Ensure all the data can be correctly managed and stored

# 7. Test Data Requirements

**Storage of Test Data:** All test data will be stored in an SQLite database. Test data will be directly inserted into the SQLite database used by the application.

**Access and Security:** Only authorised personnel will have access to the test data. To protect private data, security and privacy protocols will be followed. Personal data will be anonymized regardless of authorization, guaranteeing that it is only used for testing.

**Types of Test Data:**

1. Book Management
2. Book Information
3. Loan Details
4. Fine Details
5. Member Management

# 8.Test Schedule

Testing begins after all the wireframes have all been conducted and signed off by the project manager. Bugs and testing will be ongoing throughout the process ensuring it meets the needs of the stakeholder’s requirements and the users of the system.

|  |  |  |
| --- | --- | --- |
| Testing Type | Timeline | Comments |
| Test Plan | Nov 15 – Nov 30 | Set up test environment |
| Unit Testing | Nov 30 – Dec 15 | Individual components work |
| Integration Testing | Dec 15 – Dec 24 / Jan 2 – Jan 7 | Check components work – account for Christmas holidays |
| System Testing | Jan 8 – Jan 21 | Test system performance |
| Performance Testing | Jan 21 – Jan 31 | Stress test for various conditions |
| Bug fixing and Re-Testing | Ongoing checks | Does not specific dates this is ongoing thought the process |
| Final Check | Feb 1 – Feb 11 | Check everything is ready to be deployed |

# 9. Testing Strategy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Types of testing | Approach Description | Methodologies and techniques | Tools and Frameworks | Order Of testing |
| Unit testing | Testing smaller units of code such as methods or function | Automated tests for efficiency. Also use mock objects to conduct test insolation to pinpoint issues easier | JUnit | First |
| Integration Testing | Bottom-up testing such as adding or removing books\ | Testing using “drivers” for higher modules. When low level components are ready before higher level logic | Junit | Second |
| System testing | Check the entire system works meeting the requirements of the stakeholders | Black box testing for requirement’s testing | JDBC/SQLite | third |
| User Acceptance testing | Validate with users/librarians that the product meets there needs | Use beta / final versions to improve the systems constantly | JavaFX | Final/Ongoing |

# 

# 10.Testing tools

**Unit Testing**

**Tool:** Junit

**License:** Open source

**Justification:** The requirement of the system mentions that the system will be tested using. Furthermore, it will be best suited for testing individual components. Could use testNG as backup however this out of scope because it’s not described.

**Integration Testing**

**Tool:** Junit

**License:** Open source

**Justification:** One of the tools described in the brief. This can also be used for testing the interactions between the classes and components in the system. Could you postman as another tool to test the API integration with responses in the data, however this out of scope as it is not described.

**System Testing**

**Tool:** JavaFX, JDBC and SQLITE

**License:** Open source

**Justification:** JavaFX Used for UI and can be part of system testing especially GUI related task. For testing the database, the requirements of the system include that the database will be tested using JDBC it is also used to check interaction in Java applications.

**UAT**

**Tool:** JavaFX

**License:** Open source

**Justification:** Ensure that the end users can validate the systems usability and functionality

# 11. Test Deliverables

**Test Cases**

**Identification:** Should be labelled with a correlating number pattern or name – number pattern or name would relate to the test script code that is for

**Format:** Documented in a spreadsheet clearly identifying the test cases

**Test Environment Setup Instructions**

**Identification:** “Test Environment setup instructions”

**Format and medium:** PDF or word document

**Version Control:** Should just be a shared document – edits to setup should only available testers and developers.

**Test Scripts**

**Identification:** Description of what the automation test is testing (e.g., add books) and correlating number pattern for a test case

**Format:** Will be written in java and tested in Junit. Will be stored in version-controlled repository.

**Version Control:** Will be stored using git and GitHub to identify progress and easily identify problems.

**Test Plan Document**

**Identification:** “Library Management System Test Plan”

**Format:** Word or PDF which is easily accessible for others to see

**Version Control:** Using a shared document or version control such as word allowing to track progression.

# 

# 12. Test Responsibilities

**John Smith, Automated Tester (30 hours per week)-** is responsible for conducting automated testing and assisting the test process when resources are limited.

**Peter Frost, UI Designer (Ongoing)** - is responsible for co-operating with librarians and the testers to ensure that the GUI is to the standard of the requirement’s/librarians and of the system.

**Sarah Miller, Test Analyst (35 hours per week)-**is responsible for test execution and may require overtime approval.

**Emily Davis, Librarian (3 hours per week)-** is responsible for usability testing and is allocated a specific amount of time.

**Raj Patel, Project Manager (Ongoing)** – Responsible for signing most important features after it has been tested and developed and signing off the finished product ready for the customer.

No external contractors are needed for in the system and the budget does not account for this.

# 13. Risks and Assumptions

## Risks

**User Interaction:** Users such as the librarians may struggle with managing the system if the training has not been adequate leading to adhering of test schedules.

**Performance:** Performance-related issues with the LMS, like sluggish response times or system crashes, could affect the testing schedule and user experience.

**Database:** That the database may experience crashes or slow responses leading to slower book management testing for example

## Assumptions

According to the brief the **assumptions** are of the library management system are:

1. *“All librarians have basic computer skills and can navigate the Windows operating system.”*
2. *“The local library has a reliable internet connection for software updates and cloud backup. “*

## Contingency Plans

Ensure that all assumptions and risks of the systems are adequately accounted for.

1. Librarians address any concerns they have with the UI of the system and have the minimum required hands-on time with it
2. Assumptions of the system that there is cloud backup however these features are not implemented so would out of scope for testing. So could not create a contingency plan for cloud/data backup.
3. Ensure the system meets the Non-functional requirements of the system to account for response times.

# 14.Approvals

The approval of the test plan will be documented with:

1. Version Number – Allows to show amendments made.
2. Date
3. Role of the user who proposed this element to be signed off.
4. Name of the person signing of the proposal
5. Role of the person who signs off.
6. Date of signature,

**Raj Patel (Stake Holder)** - Has the most authority within the team as project manager. Any significant approvals should go through Raj especially the final product.

**John Smith** – Creates the automated testing the results will be documented and sent to Raj Patel to ensure it meets the requirements.

**Margaret Jones** – Copywriter will report to Raj Patel again for any approval to be signed off.

**Peter Frost** – UI Designer for the LMS will have to report to Raj Patel for an approval to signed.

# 15. Appendices

1) https://www.oracle.com/java/technologies/javafx/system-requirements.html

2)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**