**A Proposed Library Management System at Navotas City Library**

A Testing Documentation Presented to the

Faculty of Datamex College of Saint Adeline, Inc.

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science in Information Technology

Prepared by:

Caldit, Peter Joel M.

September 2025

**TESTING DOCUMENTATION**

**INTRODUCTION**

The purpose of this testing phase is to carefully evaluate the Library Management System (LMS) to ensure that it functions efficiently before being deployed for actual use. This phase is crucial because it helps identify and resolve potential issues that could affect the system’s performance, reliability, and usability. Through thorough testing, the goal is to deliver a stable and user-friendly system that meets the needs of both librarians and library users.

The main objectives of this testing” process are to verify that all features of the system such as book registration, book borrowing and returning, user management, and record tracking operate as intended. It also aims to confirm that the system enforces library rules correctly, such as preventing multiple checkouts of the same book, calculating fines for overdue returns accurately, and storing all book and user information securely. Another key objective is to ensure that the system’s user interface is intuitive and easy to navigate, allowing staff to perform daily operations smoothly and efficiently.

The scope of this testing phase covers all major functional areas of the Library Management System. This includes testing the accuracy of book records, validating borrowing and returning processes, confirming that transaction histories are properly recorded, and ensuring that reports generate accurate data. Additionally, the system’s performance, error handling, and data security features are evaluated to ensure they meet expected standards. The testing scope does not include integrations with online library databases, mobile access, or future system extensions not yet implemented.

Through this comprehensive testing process, the aim is to identify defects, reduce the risk of system failures, and enhance overall user satisfaction. The results of this testing phase will provide valuable insights for further improvements, guide necessary refinements, and ensure that the system is fully prepared for real-world use. By addressing issues early, the library can achieve more efficient operations, accurate record management, and improved service to its users once the system is officially deployed.

**TESTING ENVIRONMENT**

**Hardware Specifications**

The testing of the Library Management System (LMS) was conducted on standard desktop and laptop computers to simulate typical library usage conditions. The specifications of the devices used are as follows:

* Processor: Intel Core i3 or higher
* RAM: 8 GB or more
* Storage: 256 GB SSD or larger
* Operating System: Windows 10 or Windows 11 (64-bit)

These specifications ensured smooth performance during testing, especially when handling multiple records and transactions simultaneously.

**Software Requirements**

The software environment used for testing includes all tools and platforms required for the system to operate correctly:

* Database Management: Microsoft SQL Server Management Studio (SSMS) 2019
* Development Environment: Visual Basic 2010 (VB.NET)
* Operating System: Windows 10 / Windows 11
* Additional Libraries/Frameworks: .NET Framework 4.5, Guna.UI2 WinForms (for modern and responsive UI components)

These software tools were used to design, deploy, and test the system, ensuring that all modules interact properly and perform the intended functions reliably.

**Test Data**

Sample data was created to simulate real-world library operations and verify the system’s functionality. The test data includes:

* Books: Titles, authors, publication years, ISBN numbers, categories (fiction, non-fiction, academic, etc.), and availability status.
* Library Users: Names, ID numbers, contact details, membership types (student, faculty, guest), and borrowing limits.
* Transactions: Book borrow and return records with various dates, due dates, and fines for overdue returns to test the accuracy of tracking and fine calculation.

Using this test data, all major operations such as adding new books, registering users, borrowing and returning books, and generating transaction reports were verified to ensure the system functions accurately and efficiently under realistic conditions.

**TESTING METHODOLOGY**

The system was tested primarily using User Acceptance Testing (UAT) to ensure that the Library Management System (LMS) meets its functional requirements and provides an intuitive experience for its intended users—librarians, staff, and library members. Real users and testers performed tasks that replicate day-to-day library operations, such as adding new books, registering borrowers, borrowing and returning books, and generating reports.

**Testing Tools and Frameworks:**

The system was tested manually through the live application interface developed in Visual Basic 2010.

Testing relied on observation, verification of outputs, and comparison of actual results against expected outcomes. No automated testing tools were required, as the system operates offline and focuses on user interactions within the desktop environment.

**Test Cases and Criteria:**

Test cases were carefully designed to cover all essential features and common workflows within the system. Each test case includes a clear task description, step-by-step execution guide, expected result, and success criteria.

A test is considered successful if the system:

* Executes the intended action correctly.
* Displays accurate and complete information.
* Handles errors or invalid inputs gracefully without freezing or crashing.
* Provides a responsive and user-friendly interface for all operations.

The test cases focus on core functionalities such as:

* Book Management: Adding, updating, and deleting book records.
* User Management: Registering new members, updating profiles, and managing user records.
* Borrowing and Returning: Checking out books, validating due dates, and recording returned items.
* Fine Calculation: Automatically computing penalties for overdue books.
* Reports and Records: Viewing transaction history, borrower lists, and inventory summaries.

By following these testing criteria, the process ensures that the Library Management System is functional, reliable, and ready for real-world deployment in a library setting.

**TEST CASES**

The table below summarizes the User Acceptance Testing (UAT) conducted for the Library Management System (LMS). It lists each test case with its ID, description, steps, expected and actual results, status, and remarks. This allows testers and stakeholders to quickly verify which core functionalities—such as book management, user registration, borrowing and returning books, and report generation—were tested and whether they performed as expected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Test Steps** | **Expected Output** | **Actual Output** | **Status** | **Remarks** |
| **UAT001** | Add a new book record | 1. Open Book Management module  2. Enter book details (title, author, ISBN, category)  3. Click “Save” | Book record saved and displayed in the list | Book record saved and visible | Pass | N/A |
| **UAT002** | Register a new library user | 1. Open User Registration form 2. Enter user details (name, ID, contact info) 3. Click “Submit” | User record created and displayed in member list | User record created successfully | Pass | N/A |
| **UAT003** | Borrow a book | 1. Select registered user  2. Choose available book  3. Click “Borrow” | Borrowing record created; book marked as “Borrowed” | Book marked as “Borrowed”; transaction recorded | Pass | N/A |
| **UAT004** | Return a borrowed book | 1. Select user with borrowed book  2. Choose book to return 3. Click “Return” | Book status updated to “Available”; fine calculated if overdue | Book returned successfully; fine computed correctly | Pass | N/A |
| **UAT005** | Calculate overdue fine | 1. Borrow a book and set return date in past  2. Return the book  3. Verify fine amount | System displays correct fine based on number of overdue days | Fine displayed accurately | Pass | N/A |
| **UAT006** | Search for a book | 1. Enter title or author name in search box  2. Click “Search” | Matching book(s) displayed in results | Correct search results displayed | Pass | N/A |
| **UAT007** | Generate report | 1. Open Reports module  2. Select “Borrowed Books” report 3. Click “Generate” | Report displays accurate and complete data | Report generated accurately | Pass | N/A |

**Table 1: Test Cases for Library Management System**

**BUG TRACKING AND ISSUE LOG**

During the testing phase, all bugs and issues encountered in the Library Management System (LMS) were carefully documented, categorized, and tracked to ensure they were resolved before final deployment. Each issue was assigned a unique Bug ID, a detailed description, severity level, the person who reported it, its current status, and the corresponding resolution or action taken.

Bugs were classified based on their severity as Critical, High, Medium, or Low, depending on their impact on the system’s functionality, usability, or performance.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bug ID** | **Description** | **Severity** | **Reported By** | **Status** | **Resolution** |
| **B001** | Application freezes when adding multiple books quickly | High | Tester | Open | Optimize database insert handling |
| **B002** | Fine calculation gives incorrect result when due date is missing | Medium | Tester | In Progress | Add date validation before calculation |
| **B003** | Book search fails to display results for lowercase queries | Low | Tester | Open | Improve search filter to be case-insensitive |
| **B004** | User registration form accepts incomplete data | High | Tester | Fixed | Added required field validation |
| **B005** | Returned book does not immediately update availability status | Medium | Tester | In Progress | Refresh book list after return transaction |

**Table 2: Bug Tracking and Issue Log**

This bug tracking process ensured that all identified problems were systematically logged, reviewed, and prioritized based on their severity. By maintaining clear documentation, the development team was able to address critical issues efficiently, enhance overall system stability, and deliver a reliable and user-friendly Library Management System.

**USER ACCEPTANCE TESTING**

User Acceptance Testing (UAT) was conducted to validate that the Library Management System (LMS) meets the needs and expectations of its end-users, including librarians, library assistants, and administrative staff. The testing process focused on real-world scenarios that the system is expected to handle during daily library operations.

Key test scenarios included adding and updating book records, registering new library members with complete and accurate information, borrowing and returning books, calculating overdue fines, searching for books by title, author, or category, and generating various reports such as borrowed books and overdue lists.

During testing, librarians and staff actively interacted with the system to ensure that all workflows were intuitive, efficient, and correctly implemented. Users reported that the interface was easy to navigate and that most functions performed smoothly without significant errors.

Feedback from UAT participants included suggestions such as adding clearer notifications for invalid inputs, improving the responsiveness of the book search feature, and automatically refreshing lists after a transaction is completed. These recommendations were recorded for consideration in future updates.

Overall, the User Acceptance Testing confirmed that the Library Management System is functional, reliable, and ready for deployment, with only minor enhancements needed to further improve user experience and operational efficiency

**CONCLUSION AND RECOMMENDATIONS**

The testing phase of the Library Management System (LMS) demonstrated that the application is stable, reliable, and meets all defined functional requirements. All major features including book registration, user management, borrowing and returning of books, fine calculation, report generation, and search functionality—were thoroughly tested and confirmed to work as intended. The system successfully handled both valid and invalid inputs, updated records accurately, and maintained proper tracking of all transactions and user activities.

Key observations from the testing process indicated that the system’s core functionalities performed consistently, database operations were handled efficiently, and the user interface provided a smooth and intuitive experience for librarians and staff. Minor issues, such as slight delays when loading large datasets and search filtering inconsistencies, were identified and properly documented for resolution.

Based on the testing results, several recommendations are suggested to further improve the system’s performance and usability:

* Enhance input validation across all forms to minimize user entry errors.
* Optimize search and report generation features to handle larger data volumes more efficiently.
* Improve notification and error messages for clearer communication with users.
* Conduct regular maintenance and database backups to preserve data integrity and prevent data loss.
* Provide basic user training for librarians and staff to ensure familiarity with all features and proper system usage.

Overall, the Library Management System is ready for deployment. Implementing these recommendations will further enhance its efficiency, reliability, and user satisfaction, ensuring smooth and organized management of library operations in real-world use