A Proposed Library Management System at Navotas City Library

A Maintenance 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

**MAINTENANCE DOCUMENTATION**

**INTRODUCTION**

The Library Management System (LMS) is a desktop-based application developed using Visual Basic 2010 (VB.NET) and Microsoft SQL Server as its database backend. The system is designed to assist librarians and administrative staff in efficiently managing essential library operations such as book cataloging, borrower registration, book lending and returns, overdue tracking, and report generation. It ensures accurate record-keeping, easy data retrieval, and smooth daily operations within the library.

Regular maintenance of the Library Management System is essential to ensure that it continues to operate efficiently and reliably after deployment. Maintenance activities help sustain optimal performance, prevent system errors or downtime, and protect against potential data loss or corruption. It also supports long-term scalability, allowing the system to adapt to changes such as the addition of new features, database growth, or updates in the software environment.

The scope of system maintenance includes:

* Software Updates: Enhancing existing modules or adding new features such as advanced search filters, automated notifications, or improved reporting tools.
* Bug Fixes: Identifying and resolving errors encountered by users to maintain system stability and functionality.
* Database Optimization: Ensuring efficient performance of book and borrower data queries while maintaining data accuracy and integrity.
* Security Patches: Strengthening data protection measures to safeguard sensitive records, such as borrower information and transaction logs.
* Backup and Recovery: Implementing regular data backups and providing reliable recovery procedures in case of system failure or data corruption.

**MAINTENANCE PLAN**

The maintenance strategy for the Library Management System (LMS) is proactive and preventive, ensuring that potential issues are detected and resolved before they can affect day-to-day library operations. The IT and development teams conduct regular system monitoring, testing, and database optimization to preserve the system’s reliability, performance, and data integrity.

Maintenance activities are structured around four main types — Corrective, Adaptive, Perfective, and Preventive Maintenance — to cover all aspects of system improvement and stability.

Types of Maintenance

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **Example Scenario** |
| **Corrective Maintenance** | Fixing identified bugs or errors that interrupt or affect normal system operations. | A borrowed book record fails to update the borrower’s account due to an incorrect SQL statement; the issue is identified and fixed. |
| **Adaptive Maintenance** | Modifying the system to accommodate new operational or technical requirements. | Updating the database connection settings when the library’s server is upgraded or moved to a new machine. |
| **Perfective Maintenance** | Improving system performance, usability, or efficiency based on user feedback. | Enhancing the book search function to include filters by author, category, or publication year; improving the user interface with Guna Framework components. |
| **Preventive Maintenance** | Conducting regular updates, checks, and maintenance tasks to prevent future issues. | Performing weekly database backups, optimizing indexes in SQL Server, and reviewing system logs for potential errors or warnings. |

**Table 1: Maintenance Plan**

**BACKUP & RECOVERY PLAN**

**Backup Strategy**

The Library Management System (LMS) employs both manual and automated backup procedures to safeguard library records against data corruption, accidental deletion, or hardware failure. Backups ensure that information such as borrower records, book inventories, and transaction histories are protected and easily restorable.

**Backup Types:**

* Full Backup: A complete copy of the LibraryDatabase is created weekly using SQL Server Management Studio (SSMS).
* Incremental Backup: (Optional enhancement) Saves only modified data daily to reduce backup size and time.
* System Backup: The LMS includes a “Backup Database” button in the Admin Panel, which saves the database .bak file to C:\LibrarySystem\Backups\.

**Backup Procedures**

* Open the Backup Form within the Library Management System or use SQL Server Management Studio (SSMS).
* Select the backup destination folder (e.g., C:\LibrarySystem\Backups\).
* Click Start Backup to generate the .bak file.
* Verify the backup file’s timestamp and confirm successful completion.
* Copy the backup file to an external drive or cloud storage (e.g., OneDrive or Google Drive) for additional protection.

**Frequency**

* Full Backup: Weekly (every Friday)
* Manual Backup: Before any major update or system modification

**Storage Locations**

* Local Path: C:\LibrarySystem\Backups\
* Cloud Backup: OneDrive\LibrarySystem\Backups\ (optional)

**Recovery Steps**

* Launch SQL Server Management Studio (SSMS).
* Right-click Databases → select Restore Database.
* Choose the most recent .bak file from the backup directory
* Verify the database name and file paths, then click OK to start restoration.
* Update the VB.NET connection string in the system to link with the restored database.
* Test the restored system by logging in and confirming that book, borrower, and transaction records are intact

**PERFORMANCE MONITORING**

System performance for the Library Management System (LMS) is continuously observed and optimized to ensure efficient access to library records and smooth daily operations. Regular monitoring and database tuning are performed to maintain stability and responsiveness.

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Description** | **Threshold** | **Monitoring Tool** |
| **Server Uptime** | Measures the percentage of time the system is available and running without interruption. | 99.9% | SQL Server Agent Logs |
| **Response Time** | Time taken for main forms (e.g., Dashboard, Book Management, Borrow/Return Modules) to load and respond. | < 2 seconds | Windows Performance Monitor |
| **Error Rate** | Percentage of failed transactions, database errors, or runtime exceptions detected in logs. | < 1% | System Error Logs |
| **Database Size Growth** | Tracks monthly database growth to manage storage capacity and prevent slowdowns. | < 5% monthly | SQL Server Management Studio (SSMS) Reports |
| **Backup Success Rate** | Ensures successful creation and verification of .bak files during scheduled backups. | 100% | Backup Form Log File |

**Table 4: Performance Monitoring**

**SECURITY MEASURES**

Security is essential to protect library records, user accounts, and transaction data from unauthorized access or loss. The Library Management System (LMS) implements several security controls to ensure data integrity and confidentiality.

**User Authentication:**

The login module validates librarian and staff credentials against securely stored, encrypted values in the database.

**Database Security:**

The SQL Server database uses authentication with limited privileges, allowing only authorized users to read or modify specific tables (e.g., Books, Borrowers, Transactions).

**Encryption:**

Sensitive information such as user passwords and account details are encrypted using SHA-256 or AES algorithms before storage.

**Access Control:**

System access is role-based—administrators, librarians, and staff have permissions only for their assigned tasks (e.g., book management, borrowing transactions, report generation).

**Audit Trail:**

The system automatically logs user activities, including adding or deleting books, managing borrowers, and recording borrow/return transactions, for accountability and monitoring.

**Firewall & Antivirus:**

The system host is secured using Windows Defender and the institution’s network firewall to prevent malware, intrusion, or unauthorized network access.

**DOCUMENTATION UPDATES**

All modifications or improvements made to the Library Management System (LMS), including code adjustments, database updates, or configuration changes, must be properly documented to maintain accuracy and consistency. The Documentation Lead or assigned librarian updates the following records after every major system revision or release:

**Technical Documentation** – Updated to reflect changes in the system structure, installation procedures, and maintenance guidelines.

**User Manual** – Revised to include modifications in the user interface, new features, and updated instructions for book management, borrowing, and returning processes.

**Changelog** – Records all version updates, bug fixes, feature enhancements, and performance improvements for reference and tracking.

Regular documentation updates ensure that users and administrators can easily understand and adapt to any system improvements or adjustments.

**CONCLUSION & RECOMMENDATIONS**

The maintenance phase ensures that the Library Management System (LMS) remains efficient, reliable, and secure throughout its operational lifespan. The structured maintenance plan includes regular database backups, bug fixes, security monitoring, and performance optimization to ensure smooth library operations. Continuous maintenance also guarantees that users — including librarians, staff, and students — experience minimal downtime and accurate record management.

**Recommendations**

* Implement automated backup scheduling using Windows Task Scheduler to prevent data loss.
* Integrate email or in-system notifications for overdue books, upcoming due dates, and backup completions.
* Enhance role-based access control (RBAC) to ensure that only authorized users can perform administrative tasks.
* Conduct annual system audits to verify data integrity, identify unused records, and optimize database performance.
* Consider migrating future versions to a web-based or cloud platform to enable remote access and improve scalability.