**LIBRARY MANAGEMENT SYSTEM FOR DATAMEX COLLEGE OF SAINT ADELINE VALENZUELA BRANCH**

Submitted by:

Ladia, Anthony

Junsay, Rjhay

Hondrado, Acezzel ann

Marabe, Uvert

Submitted to:

Mr. Gabriel Thomas Torneros

**REQUIREMENTS SPECIFICATION**

**INTRODUCTION**

**Purpose of the Document**

This Requirements Specification Document defines the functional and non-functional requirements for the Library Management System (LMS) to be developed for Datamex College of Saint Adeline Valenzuela Branch. This document is for system design, development, and testing phases, ensuring all stakeholders have a clear understanding of what the system will accomplish.

The document is intended for:

* Development team members
* System administrators
* Book Management
* Requirements and Purpose

**Overview of the Software System**

The Library Management System is a desktop application developed using Visual Basic 10 (VB.NET) designed to help the library operations at Datamex College of Saint Adeline Valenzuela Branch.

Key System Capabilities:

* Managing the book inventory and catalog information
* Search functionality for categorized books
* Administrative dashboard for librarian operations
* Reduced manual workload and minimized human errors

Target Users:

Primary: Librarians, Borrowers, and Faculty

Secondary: Library administrators

**Scope of the Requirements Specification**

Included in Scope:

* Desktop Library Management System application for local use only
* Basic book catalog management (add, edit, delete, search, view books)
* Simple and manual book categorization system
* Basic reporting features
* Local database storage using Microsoft Access or SQL Server Express

Excluded from Scope:

* Digital copies of books (e-book functionality)
* Student direct access to the system
* Integration with other branches of Datamex College
* Mobile application development
* Online payment processing
* Integration with external library systems
* Automated notification systems (email/SMS)

System Boundaries:

* The system will operate on a single computer workstation
* Application will use local database storage
* No network connectivity required
* Manual operation by librarians only

**FUNCTIONAL REQUIREMENTS**

**Book Catalog Management**

LMS-001: The system shall allow add new books to the catalog.

Priority: High

Dependencies: None

* Form includes fields for: Title, Author, ISBN, Category, Publication Year, Quantity
* System validates that required fields are filled
* System assigns a unique Book ID automatically

LMS-002: The system shall provide functionality to edit the book from catalog.

Priority: Medium

Dependencies: LMS-001

* Users can modify all book details except Book ID
* Changes are saved immediately to the database

LMS-003: The system shall allow the users to remove books from the catalog.

Priority: Medium

Dependencies: LMS-001

* System shows confirmation message before deletion
* Books are permanently removed from database

**Book Search and Browse**

LMS-004: The system shall provide comprehensive search functionality to locate books to catalogs.

Priority: High

Dependencies: LMS-001

* Search by: Title, Author, ISBN, or Category
* Search results display matching books with all details

LMS-005: The system shall organize books by categories.

Priority: High

Dependencies: LMS-001

* Users can filter books by selecting a category
* Common categories include: Fiction, Non-Fiction, Reference, Textbooks, etc.

**Dashboard and Reporting**

LMS-006: The system shall provide a basic reports about books and catalogs.

Priority: Low

Dependencies: LMS-001, LMS-002

* The system shall provide basic reports about the library collection
* Reports include: Total books count, Books by category, List of all books
* Reports can be printed or saved as text files

**NON-FUNCTIONAL REQUIREMENTS**

**Performance**

|  |  |
| --- | --- |
| LMS 007 | System shall respond to user actions within 2 seconds. |
| LMS 008 | System shall handle up to 500 book records without performance issues |
| LMS 009 | Database operations shall complete within 3 seconds |

**Usability**

|  |  |
| --- | --- |
| LMS 010 | Interface shall be simple and easy to use with minimal training |
| LMS 011 | Error messages shall be displayed in simple, understandable language |

**Reliability**

|  |  |
| --- | --- |
| LMS 012 | : System shall save all changes immediately to prevent data loss. |
| LMS 013 | : System shall handle database errors with appropriate messages. |
| LMS 014 | : System shall maintain data integrity during all operations |

.

**Security**

|  |  |
| --- | --- |
| LMS 015 | Check that only valid data is entered in text boxes. |
| LMS 016 | Not allow saving books with the same ISBN number. |
| LMS 017 | Prevent empty required fields from being saved. |

**Scalability**

|  |  |
| --- | --- |
| LMS 018 | Load book lists quickly even with many books |
| LMS 019 | Allow easy addition of new book categories |

**Maintainability**

|  |  |
| --- | --- |
| LMS 020 | Code shall have simple comments explaining what each button and function does |
| LMS 021 | Show helpful error messages when something goes wrong |

*Table 1*: Non-Functional Requirements

**USE CASES**

**Book Management Use Cases**

|  |  |
| --- | --- |
| Use Case ID | UCLMS-001 |
| Use Case Name | Add New Book |
| Description | A librarian adds a new book to the library catalog with complete information |
| Actors | Librarian, System |
| Preconditions | A librarian adds a new book to the library catalog with complete information |
| Postconditions | New book is saved in database with unique BookID and appears in catalog |
| Alternate Flows | Duplicate ISBN: System alerts user and prevents duplicate entry  Missing required fields: System highlights missing information and prevents submission  Database error: System displays error message and retains entered data |

*Table 2*: Add New Book

|  |  |
| --- | --- |
| Use Case ID | UCLMS-002 |
| Use Case Name | Search Books |
| Description | A librarian searches for specific books in the catalog using various search criteria |
| Actors | Librarian, System |
| Preconditions | System is running and catalog contains book records |
| Postconditions | Search results are displayed showing matching books with their details |
| Alternate Flows | No results found: System displays "no results" message with search suggestions  Multiple results: System displays paginated results with sorting options |

*Table 3*: Search Books

|  |  |
| --- | --- |
| Use Case ID | UCLMS-003 |
| Use Case Name | Edit Book Information |
| Description | A librarian modifies existing book details in the catalog |
| Actors | Librarian, System |
| Preconditions | System is running, book exists in catalog, and book is selected for editing |
| Postconditions | Book information is updated in database and changes are visible in catalog |
| Alternate Flows | Validation Error: System highlights invalid fields and shows error “Please check the highlighted fields”  Update Failed: System shows error “Could not update book information” and keeps original data |

*Table 4*: Edit Book Information

|  |  |
| --- | --- |
| Use Case ID | UCLMS-004 |
| Use Case Name | Delete Book Information |
| Description | A librarian removes a book from the catalog permanently |
| Actors | Librarian, System |
| Preconditions | System is running, book exists in catalog, and book is selected for deletion |
| Postconditions | A librarian removes a book from the catalog permanently |
| Alternate Flows | User Cancels: User clicks “No” on confirmation dialog and book remains in catalog  Failed: System shows error “Could not delete book” and book remains in catalog  Database Error: System shows error “Database error occurred, please try again” |

*Table 5*: Delete Book Information

**Administrative Use Case**

|  |  |
| --- | --- |
| Use Case ID | UCLMS-005 |
| Use Case Name | View Dashboard |
| Description | Dashboard is displayed with current statistics and all navigation options are available |
| Actors | Librarian, System |
| Preconditions | System is running and database connection is established |
| Postconditions | Dashboard is displayed with current statistics and all navigation options are available |
| Alternate Flows | Data loading error: System displays cached data with refresh option  Slow loading: System displays loading indicator and loads data progressively |

*Table 6*: View Dashboard

**DATA REQUIREMENTS**

**Data Entities**

**Book Entity**

**Attributes**

* Book ID (Primary Key, Auto-increment integer)
* Title (Text, 255 characters, Required)
* Author (Text, 255 characters, Required)
* ISBN (Text, 20 characters, Optional)
* Category (Text, 100 characters, Required)
* Publication Year (Integer, Optional)
* Quantity (Integer, Required, Default: 1)
* Date Added (Date/Time, Auto-generated)

**User Entity**

**Attributes**

* User ID (Primary Key, Auto-generated)
* Full Name (Text, 255 characters)

**ASSUMPTIONS AND CONSTRAINTS**

**Assumptions**

* + Single user operation (no concurrent access needed)
  + Small to medium library collection (under 10,000 books)
  + Basic computer literacy of library staff
  + Microsoft Access is available on target computer

**Constraints**

* + No network functionality required
  + No user authentication needed
  + Simple interface design for ease of use
  + Use only basic VB.NET 2010 features
  + No external dependencies beyond .NET Framework 4.0

**GLOSSARY**

**Terms and Definitions**

**API (Application Programming Interface)**: A set of protocols and tools for building software applications that allows different programs to communicate with each other.

**Backup**: A copy of data stored separately from the original to protect against data loss and enable recovery in case of system failure.

**Catalog**: A comprehensive database of all books and materials available in the library, including their details and availability status.

**Database**: An organized collection of structured information stored electronically for efficient retrieval and management.

**Due Date**: The date by which a borrowed book must be returned to avoid penalty charges.

**ISBN**: International Standard Book Number, unique identifier for published books that must be unique in the system

**LMS (Library Management System):** A software application designed to manage library operations including cataloging, circulation, and administration.

**Overdue**: The status of a borrowed book that has not been returned by its due date.

**Primary Key**: A unique identifier for each record in a database table that ensures no duplicate entries exist.

**Query**: Database operation to search and retrieve specific information

**Session**: The period during which a user is actively logged into the system, typically terminated by logout or timeout.

**SQL (Structured Query Language)**: A programming language designed for managing and manipulating relational databases.

**Stakeholder**: Any individual or group with an interest in or influence over the project, including users, sponsors, and development team members.

**VB.NET (Visual Basic .NET):** A programming language and development environment used for creating Windows desktop applications on the .NET Framework platform**.**

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Purpose/Changes | Author | Date |
| 0.0 | Initial Draft - Nothing | Team | August 27th , 2025 |
| 1.0 | Change related Web-related to VB.NET | Team | August 28th , 2025 |
| 2.0 | Removed any related advanced mechanism and functions and minimized the item should be put in system | Team | September 4th, 2025 |

**APPENDIX**

GeeksforGeeks. (2023, May 8). *Library Management System*. In *Software Engineering*. GeeksforGeeks.

Wikipedia contributors. (2023, August 1). *Sharable Content Object Reference Model*. In *Wikipedia*.

Wikipedia contributors. (2023, July 27). *Koha (software)*. In *Wikipedia*.

Wikipedia contributors. (2023, July 31). *Integrated library system*. In *Wikipedia*.

Duy Tan University. (2022). *Software requirement specification document*. Scribd.

SourceCodester. (2019, February 22). *Library management system using VB.NET and MySQL*. SourceCodester.