**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

**DESIGN DOCUMENT**

**INTRODUCTION**

**Purpose of Document**

This design document provides detailed technical specifications for the Library Management System (LMS) being developed for Datamex College of Saint Adeline Valenzuela Branch. It serves as a blueprint for developers,, defining the system architecture, design patterns, data structures, and implementation guidelines.

**Overview of the Software**

The Library Management System is a desktop application developed using Visual Basic 10 (VB.NET) designed to modernize and streamline library operations at Datamex College of Saint Adeline Valenzuela Branch. The system will replace manual, paper-based processes with a digital solution that provides:

* Focus book inventory management
* Administrative dashboard for librarians
* Course/strand-specific book categorization

**Scope**

The technical design coverage including desktop architecture, local database design, Windows Forms UI, security and deployment specifications.

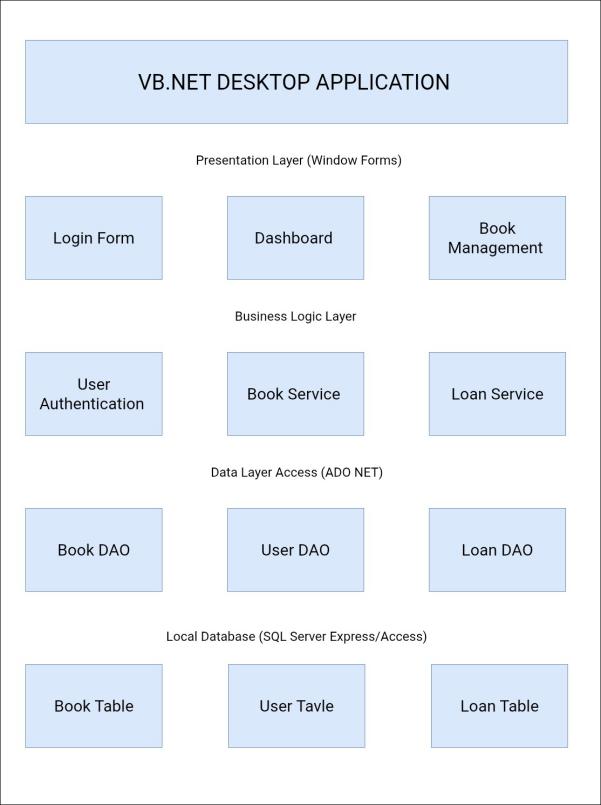
This document covers the complete technical design of the LMS, including:

* Desktop application architecture and form interactions
* Local database design and data models
* Windows Forms user interface specifications
* Security considerations for desktop applications
* Local database integration requirements
* Application deployment architecture
* Performance and memory management requirements

**Exclusions**

* Digital book content (e-books)
* Student self-service access
* Multi-branch functionality (Valenzuela Branch only)
* Network-based multi-user concurrent access

**SYSTEM ARCHITECTURE**

**Overview of the System Architecture:** Desktop Application with Local Database

*Figure 1*: Desktop Application with Local Database

**High-level components and their interactions**

Three-tier structure: presentation layer, business logic

components and data access layer.

**Presentation Layer Components (Windows Forms)**

* **Login Form**: User authentication interface
* **Main Dashboard Form**: Central navigation and statistics display
* **Book Management Forms**: Add, edit, search, and view books
* **Borrowing Interface Forms**: Loan processing and management
* **Settings Form**: System configuration and user management

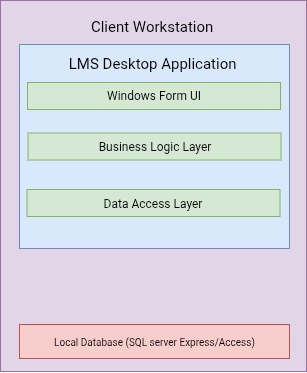
**Business Logic Components**

* Book Management Service: CRUD operations for book inventory
* Report Generation Service: Data analysis and report creation
* Validation Service: Data input validation and business rules
* Configuration Manager: Application settings management

**Data Access Layer Components**

* **Database Connection Manager**: Connection pooling and management
* **Book Data Access Object**: Book-related database operations
* **User Data Access Object**: User account database operations

**DEPLOYMENT ARCHITECTURE**

**Architecture Type:** Standalone Desktop Application

*Figure 2*: Standalone Desktop Application

**Deployment Requirements**

* Windows Operating System (Windows 7 or later)
* .NET Framework 4.0 or later
* SQL Server Express for database
* Minimum 2GB RAM, 500MB disk space

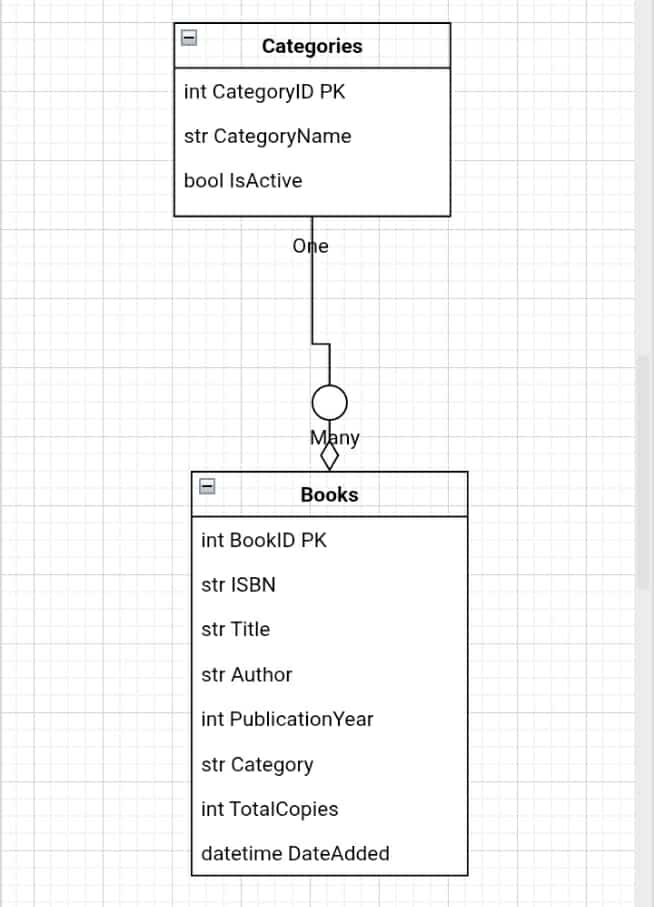
**DATABASE DESIGN**

Local database structure supporting book inventory, borrower management, and user

authentication

**Entity-relationship diagram (ERD)**

Visual mapping of Books and Categories table relationships in the local database

****

*Figure 3*: Entity-relationship Diagram

**DESCRIPTION OF DATABASE TABLES, FIELDS, AND RELATIONSHIPS**

**Books Table**

Book information and inventory data storage with ISBN, title, author, and copy tracking

**Purpose:** Store book information and inventory data

BookID (INT, Primary Key, Auto-increment)

ISBN (VARCHAR(13), Unique)

Title (VARCHAR(255))

Author (VARCHAR(255))

Category (VARCHAR(100))

TotalCopies (INT)

DateAdded (DATETIME)

AddedBy (INT, Foreign Key to Users)

**Borrowers Table**

Student and faculty borrower information management system

**Purpose:** Store borrower information

BorrowerID (INT, Primary Key, Auto-increment)

StudentID (VARCHAR(20))

FullName (VARCHAR(255))

ContactNumber (VARCHAR(15))

Email (VARCHAR(255))

BorrowerType (VARCHAR(20): ‘Student’, ‘Faculty’, ‘Staff’)

**Users Table**

Librarian and administrative account storage for system access control

**Purpose:** Store librarian and admin account information

UserID (INT, Primary Key, Auto-increment)

Username (VARCHAR(50), Unique)

FullName (VARCHAR(255))

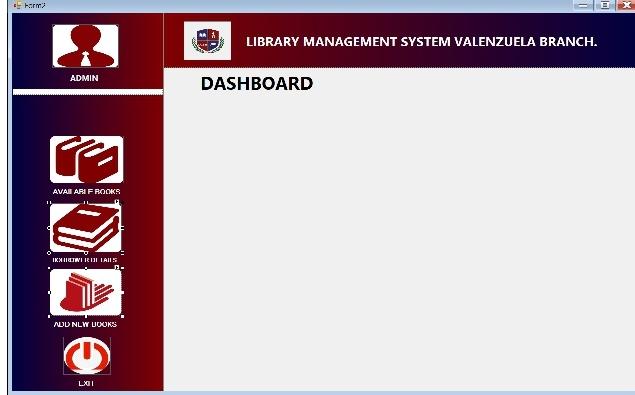
Role (VARCHAR(20): ‘Admin’, ‘Librarian’)

Email (VARCHAR(255))

**USER INTERFACE DESIGN**

**Main Dashboard Interface**

Based on the provided interface design, the main dashboard follows a sidebar navigation pattern:

*Figure 4*: Dashboard Interface

**Layout Structure**

Fixed sidebar navigation with main content area for function-specific displays

* **Left Sidebar (200px width):** Navigation menu with user profile and main functions
* **Header Bar:** Application title and branch identification
* **Main Content Area:** Dynamic content display based on selected function

**Navigation Elements**

Organized menu system providing access to all library management functions

**ADMIN Profile Section**

* User avatar/icon display
* User role indicator
* Quick access to user settings

**Primary Navigation Menu**

* **Dashboard** – Main statistics and overview
* **Manage E-Books** – Book inventory management (renamed from web version)
* **Borrowers Details** – Borrower information management
* **Add New Books** – New book entry functionality
* **Exit** – Application termination

**Login Form**

**Dimensions**: 400x300 pixels, centered

**Controls**

* Username TextBox (300x30)
* Password TextBox (300x30, masked)
* Login Button (100x35)
* Cancel Button (100x35)
* Logo/Institution branding

**Dimensions**: 1024x768 pixels (resizable)

**Layout**

* MenuStrip with File, Edit, View, Tools, Help menus
* Left Panel (200px) for navigation buttons
* Main Panel for content display
* StatusStrip at bottom for system information

**Add/Edit Book Form**:

* ISBN TextBox with validation
* Title TextBox (required)
* Author TextBox (required)
* Publication Year NumericUpDown
* Copies NumericUpDown
* Save, Cancel, Clear buttons

**Borrowing Forms**

**Borrow Book Form**

* Borrower search functionality
* Book search and selection
* DateTimePicker

**Design Principles and Usability Considerations**

Visual consistency, hierarchy, accessibility, and efficiency standards for library staff

workflow

**Visual Design Principles**

* **Consistency**: Uniform color scheme, fonts, and button styles
* **Hierarchy**: Clear information hierarchy with headings and sections
* **Accessibility**: High contrast colors, readable fonts (minimum 10pt)
* **Efficiency**: Keyboard shortcuts for common operations

**Color Scheme**

* **Primary**: Dark Red/Maroon (#8B0000) – matches school branding
* **Secondary**: White (FFFFFF) for main content areas
* **Accent**: Light Gray (F5F5F5) for form backgrounds
* **Text**: Black (000000) for readability

**Navigation Design**

* Icon-based navigation with text labels
* Breadcrumb navigation for sub-forms
* Quick access buttons on main dashboard
* Context menus for right-click operations

**Usability Features**

* Auto complete for frequent searches
* Input validation with immediate feedback
* Confirmation dialogs for destructive operations
* Keyboard navigation support
* Tool tips for complex functions

**COMPONENT DESIGN**

**Key System Components/Modules**

Login, Book Management, and supporting service modules for complete library operations

**Login Module**

User authentication and session management using LoginManager and User classes

**Purpose**: Handle user login and session management

**Location**: Login namespace

|  |
| --- |
| Public Class LoginManager  Public Function LogoutUser() As Boolean  Public Property CurrentUser As User  End Class  Public Class User  Public Property UserID As Integer  Public Property FullName As String  End Class |

**Book Management Module**

Complete book lifecycle management through BookService and Book classes with CRUD operations

**Purpose**: Handle all book-related operations

**Location**: BookManagement namespace

Key Classes:

|  |
| --- |
| Public Class BookService  Public Function AddBook(book As Book) As Boolean  Public Function UpdateBook(book As Book) As Boolean  Public Function DeleteBook(bookId As Integer) As Boolean  Public Function SearchBooks(criteria As SearchCriteria) As List(Of Book)  Public Function GetBookById(bookId As Integer) As Book  End Class  Public Class Book  Public Property BookID As Integer  Public Property ISBN As String  Public Property Title As String  Public Property Author As String  Public Property Category As String  Public Property TotalCopies As Integer  Public Property AvailableCopies As Integer  End Class |

**Component Communication Flow**

**Book Addition Process**

1. User fills AddBookForm
2. Form calls BookService.AddNewBook()
3. BookService validates data
4. BookService calls BookRepository.Add()
5. Repository executes SQL INSERT
6. Result propagated back to UI
7. UI displays success/error message

**Error Handling Strategy**

Comprehensive error management with custom ErrorCode enumeration and centralized exception handling

|  |
| --- |
| Public Class ErrorHandler  Public Shared Sub LogError(ex As Exception, context As String)  Public Shared Sub DisplayError(message As String, title As String)  Public Shared Function HandleDatabaseError(ex As Exception) As String  End Class |

**Error Handling Levels**

* **Data Layer**: SQL exceptions, connection errors
* **Service Layer**: Business rule violations, validation errors
* **Presentation Layer**: User input errors, display formatting issues

**Data Sources**

* Manual user input (librarian)
* Local database files
* Configuration files

**Processing Logic**

* Input validation and sanitization
* Business rule enforcement
* Database CRUD operations
* Report generation calculations

**Data Destinations**

* Local database storage
* Screen display output
* Printed reports
* Log file

**Performance Requirements**

* Application startup: < 5 seconds
* Form loading: < 2 seconds
* Database queries: < 1 second
* Report generation: < 10 seconds

**Performance Testing Plan**

* Load testing with maximum expected records (100 books)
* Memory usage monitoring during extended use
* Response time measurement for critical operations

**Error Handling and Logging**

Comprehensive error management with custom ErrorCode enumeration and centralized exception handling

**Error Handling Strategy**

|  |
| --- |
| Public Enum ErrorCode  None = 0  DatabaseConnection = 1001  ValidationFailed = 1002  BookNotFound = 1003  UserNotAuthorized = 1004  End Enum  Public Class ErrorHandler  Public Shared Sub HandleError(ex As Exception, errorCode As ErrorCode)  LogError(ex, errorCode)  ShowUserFriendlyMessage(errorCode)  End Sub  End Class |

**Logging Specifications**

Daily log files with 30-day retention in application Logs directory using standard .NET logging levels

* **Log Location:** Application folder/Logs directory
* **Log Rotation:** Daily log files, 30-day retention
* **Log Levels:** Error, Warning, Information, Debug

**Third-Party Integrations**

Self-contained system using only .NET Framework libraries, Windows Forms, and bundled SQL Server Express

**Integrated Components**

* **No External APIs**: System operates independently without internet connectivity
* **.NET Framework Libraries**: Built-in data access and UI components
* **Database Engine**: SQL Server Express (bundled with application)

**Integration Points**

* **SQL Lite**: Native database connectivity
* **Windows Forms**: Standard UI framework

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Changes** | **Author** | **Date** |
| 1.0 | Initial Web Design | Team | August 27th 2025 |
| 1.1 | Converted to VB.NET desktop application | Team | August 28th 2025 |
| 2.1 | Description and format | team | September 14th  , 2025 |

**APPENDIX**

Microsoft. (n.d.). SQL Server Express LocalDB reference (SQL Server). MicrosoftLearn.

Wikipedia contributors. (2023, July 25). SQL Server Express. In Wikipedia.

Wikipedia contributors. (2023, August 2). Windows Forms. In Wikipedia.

Mescus Inc. (2023, March 15). The definitive guide to WinForms controls. Medium.

Stack Exchange contributors. (2012, July 25). How to properly structure a project in WinForm?. Software Engineering Stack Exchange.