PROJECT PART 2: DATABASE REQUIREMENTS

February 23, 2025

INTRODUCTION

Project Overview

The purpose of this database project is to develop an SQL-based Library Management System (LMS) for managing books, magazines, borrowers, transactions, and overdue tracking. The LMS will streamline library operations by replacing paper-based systems with a modular, user-friendly database that enhances eDiciency and generates insightful reports.

Scope

The LMS will manage diDerent types of library materials, including books (physical and digital) and magazines. It will support client management, borrowing rules, and overdue tracking. The system will enforce borrowing restrictions based on membership types, allow reservations, and generate reports. However, it will not include physical checkout systems (e.g., barcode scanners or RFID tags) or integration with third-party digital content services.

Glossary

- ***Library Management System (LMS):*** *An architecture created to manage library assets, member enrollments, and transaction operations.*
- ***Overdue Monitoring:*** *A device that tracks and records overdue books and calculates corresponding fees.*
- ***Membership Type:*** *A classification of customers that identifies the borrower ability and charge terms for these customers (for example, normal, student, old person).*
- ***Reservation:*** *A feature allowing clients to place holds on borrowed items.*
- ***Notifications:*** *Automated alerts for due dates, overdue items, and reserved book availability.*

STAKEHOLDERS

The stakeholders have a very critical role to play in the development and success of the Library Management System. We must identify these key user groups and ensure that the database is curated to meet the needs of all users, from the end-users to administrators as well as developers. This section will highlight the primary, administrative and indirect stakeholders that are involved in the development of the Library Management System, elaborating on the roles, responsibilities, and system expectations.

1. Primary Stakeholders: End - Users

The primary stakeholders are the users that directly interact with the LMS and use it for borrowing, returning, and interacting directly with library resources.

Library Patrons

- ***Role:*** *The people who use the library resources, borrow books and content, return resources and reserve library materials.*
- ***Responsibilities:***
    - *Find the required resources the person needs according to their requirements.*
    - *Search for books, digital content, magazines which are available or able to be reserved.*
    - *Borrow items and return items according to the policies.*
    - *Pay penalties for overdue items or policy encroachment.*
    - *Receive any notifications and outreach regarding borrowed materials return, overdue items, reserved books.*
- ***System Needs:***
    - *User oriented interface for borrowers to be able to browse inventory.*
    - *Notifications for incoming due dates and availability of reserved resources.*
    - *Secure client access to account.*

Library Staff

- ***Role:*** *Oversee the daily library management operations and assist the clients with needs such as borrowing and resource management.*
- ***Responsibilities:***
    - *Check items in/out for clients.*
    - *Assist with membership registration and client records.*

- *Monitor policies, check late/overdue resources and apple penalties.*
- *Process and manage reserved resources for clients.*

- ***System Needs:***
  - *Administrative access in order to manipulate the client records.*
  - *Ability to create reports on items that are late/overdue with fines.*
  - *Efficient searching and filtering abilities to find books and resources effectively.*

2. Administration Stakeholders

These stakeholders will oversee the policies in place at the library, maintain the integrity of the user system and make sure the client data is secure.

Library Administrator

- ***Role:*** *Manage the system scale configurations and maintain the policies of the library.*
- ***Responsibilities:***
  - *Define and enforce policies, monitor and update resource use limits and fees.*
  - *Generate the reports on library resource usage, overdue items and financial summaries in given context.*
  - *Maintain system integrity by enforcing user roles and access.*
- ***System Needs:***
  - *Role assignment, given administrative access features based on roles for the different staff based on levels and client engagement.*
  - *Reporting tools for tracking resource allocation to clients, fines and reserved items.*
  - *Data security measures to make sure data is not lost or corrupted.*

Database Developers and Support Team

- ***Role:*** *Design, implement and maintain the database.*
- ***Responsibilities:***
  - *Make sure that the database performance is sound, secure, and scalable.*
  - *Provide technical support and bug fixes to database issues.*
  - *Continuously optimize the database.*
  - *Make efficient SQL queries for fast and efficient data retrieval.*
- ***System Needs:***

- *Scalable database architecture that can handle an increasing number of users and data.*
- *Secure authentication policies and encryption for the user data to prevent data leaks.*

3. Indirect Stakeholders

The stakeholders that will not directly interact with the library system but has influence on the policies and development of the database.

School/University/Organization

- ***Role:*** *The governing body that influences the policies for library system operations.*
- ***Responsibilities:***
  - *Provides resource allocation for the system maintenance and upgrades.*
  - *Establish the policies for data protection and accessibility.*
  - *Manage changes regarding borrowing limits and fees.*
- ***System Needs:***
  - *Reports on library usage to measure resource allocation and funding.*
  - *Adhere to the institutional standards on data privacy and security.*

Government or Public Library Authority

- ***Role:*** *This entity regulates that libraries follow local laws and regulations in accordance with the set policies.*
- ***Responsibilities:***
  - *Monitor the adherence of the system with public service regulations and data protection laws.*
  - *Allocate any funding and resources for system improvements.*
  - *Able to measure community engagement and library service effectiveness.*
- ***System Needs:***
  - *Report generation features that help analyze public library services efficiency.*

REQUIREMENTS

Functional Requirements

1. User Administration

- ***Role-Based Access Control (RBAC)***
    - ***Patrons****: Can browse/search library items, borrow, reserve, and pay fines.*
    - ***Librarians****: Perform patron actions plus add/update/delete library items, manage members, fines, and generate reports.*
- ***Authentication***
    - *Each user (patron or librarian) must have a **unique username**.*
    - *All passwords must be **hashed** and **secured** (e.g., using MySQL's secure password functions or within the application layer).*

2. Data Entry & Maintenance

- ***Create** new records for books, magazines, digital media, and members.*
- ***Read** records for quick lookups.*
- ***Update** attributes (e.g., availability, membership details, contact info).*
- ***Delete** items no longer in circulation or memberships that are canceled/expired.*
- *Use **referential integrity** in MySQL (foreign keys in InnoDB) to maintain data consistency.*

3. Borrowing & Returning

- ***Loans****: Track which member borrowed an item, the loan date, and due date.*
- ***Overdue Handling:***
    - *Automatic **fine calculation** for overdue items based on membership type's late fee rate.*
    - *Mark the item as "Overdue" or "Lost" if not returned by a certain threshold.*
- ***Returns****: Update the return date, calculate any fines, and change availability to "Available" again.*

4. Reservations

- *If an item is checked out, patrons can **reserve** it.*
- ***Notification** system alerts the patron when the item is available.*
- ***Expiration****: Unclaimed reservations expire after a configurable time.*

5. Notifications

- ***Due Date Reminders*** *(e.g., email notices, text messages).*
- ***Overdue Alerts****: Inform members of overdue items and fines.*
- ***Reservation Availability****: Prompt the member when a reserved item is ready for pickup.*

6. Report Generation

- ***Inventory Report****: All items, their availability status, and usage statistics.*
- ***Overdue Report****: Which items are overdue, how long, and associated fines.*
- ***Borrowing Trends****: Which items/genres/members are most active.*
- ***Fine Calculation Summary****: Summaries of who owes fines and which are paid.*
- ***Reservation Report****: Which items are reserved, by whom, and the status of each reservation.*
- ***Member Activity****: A history of what each member borrowed, returned, and reserved.*

Data Entities

Entity	Attributes	Example MySQL Data Types	Description	Constraints
Book	ISBN (PK), Title, Author, Genre, PubYear, Availability	ISBN: VARCHAR(13) NOT NULL, Title: VARCHAR(255), PubYear: YEAR	Stores book details (physical or digital).	PRIMARY KEY (ISBN), unique ISBN, Availability could be an ENUM.
DigitalMedia	MediaID (PK), Title, Creator, MediaType, Availability	MediaID: INT UNSIGNED AUTO_INCREMENT, Title: VARCHAR(255)	Tracks digital media items (e.g., eBooks, audiobooks, DVDs).	PRIMARY KEY (MediaID), unique ID, Availability could be an ENUM.
Magazine	IssueID (PK), Title, IssueNumber,	IssueID: INT UNSIGNED AUTO_INCREMENT, Title:	Manages magazines.	PRIMARY KEY (IssueID), unique ID, Status could be an ENUM.

	Publication Date, Status	VARCHAR(255), PublicationDate: DATE		
Member	MemberID (PK), Name, Contact, TypeID (FK), AccountStatus	MemberID: INT UNSIGNED AUTO_INCREMENT, Name: VARCHAR(255)	Stores patron info (name, address/email, membership details).	PRIMARY KEY (MemberID), references MembershipType.TypeID.
Membership Type	TypeID (PK), TypeName, BorrowLimit, LateFeeRate	TypeID: INT UNSIGNED AUTO_INCREMENT, TypeName: VARCHAR(50)	Defines membership categories (Student, Regular, Senior) & associated borrowing limits.	PRIMARY KEY (TypeID), BorrowLimit could be TINYINT.
Loan	LoanID (PK), MemberID (FK), ItemID, ItemType, LoanDate, DueDate, ReturnDate, FineID (FK)	LoanID: INT UNSIGNED AUTO_INCREMENT, LoanDate: DATE	Records borrowing transactions & references correct items.	PRIMARY KEY (LoanID), references MemberID & FineID.
Fine	FineID (PK), LoanID (FK), Amount, PaymentStatus	FineID: INT UNSIGNED AUTO_INCREMENT, Amount: DECIMAL(6,2)	Holds fines for overdue items.	PRIMARY KEY (FineID), references LoanID, PaymentStatus could be an ENUM.
Reservation	ReservationID (PK), MemberID (FK), ItemID, ItemType,	ReservationID: INT UNSIGNED AUTO_INCREMENT, RequestDate: DATE	Tracks reservations on any item (book, media, magazine).	PRIMARY KEY (ReservationID), references MemberID.

	RequestDate, Status			
Librarian	LibrarianID (PK), Name, Contact, Role	LibrarianID: INT UNSIGNED AUTO_INCREMENT, Name: VARCHAR(255)	Stores librarian/staff info.	PRIMARY KEY (LibrarianID), Role could be ENUM('Admin', 'Staff').

Non-Functional Requirements

1. Performance

- *The library database should support at least N concurrent users, where N scales with MariaDB's storage and query handling capacity.*
- *Response times should be optimized to ensure quick retrieval of information, ideally within X milliseconds for common queries.*

2. Reliability

- *The system should ensure 99.9% uptime with minimal service interruptions.*
- *Data integrity must be maintained to prevent data loss or corruption.*

3. Scalability

- *The system should support increasing numbers of users and items without a significant drop in performance.*
- *It should be designed to accommodate future expansion, including new library branches and additional media types.*

4. Usability

- *The user interface should be intuitive and accessible for both library staff and clients.*
- *The system should comply with WCAG 2.1 accessibility standards.*

5. Security

- *User authentication should be required to access and modify records.*
- *Role-based access control (RBAC) should be implemented to restrict unauthorized actions.*

- *Data encryption should be applied to sensitive information such as user contact details.*

6. Maintainability

- *The system should be modular, allowing easy updates and bug fixes.*
- *Code should follow standardized documentation practices for ease of maintenance.*

7. Efficiency

- *Queries should be optimized to retrieve results in the shortest time possible.*
- *Resource usage (CPU, memory, storage) should be minimized to ensure cost-effective operation.*

8. Portability

- *The system should be deployable on multiple platforms, including Windows, macOS, and Linux servers.*
- *It should be accessible via desktop and mobile devices through a web interface.*

9. Reusability

- *Components such as authentication modules, reporting tools, and database schemas should be reusable across similar applications.*
- *API endpoints should be structured to support future integration with other library systems or third-party applications.*

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements

- *Within the scope of this project, the EECS cycle servers will be the host site of the database. This will be a standalone system and should suffice for the requirements of our Library Management System.*
- *This system will be able to be run on a system of 8GB of memory and 100GB of storage space.*

Software Requirements

- *The Database Management System will be run through MySQL and MariaDB.*
- *Linux will be the operating system to access the EECS servers, Windows and MacOS will be used for development.*

- *Visual Studio Code will be the main development tool and GitHub will be used for version control.*