|  |
| --- |
| Project Part 2: Database requirements |
| February 23, 2025 |

# Introduction

## Project Overview

|  |  |
| --- | --- |
|  | The Library Management System (LMS) is designed to manage a small library’s collection of books, digital media, and magazines. It will facilitate borrowing, returning, and reserving items while tracking client memberships and fees. The system aims to improve library efficiency and provide insightful reports on borrowing trends, overdue items, and membership activity. |

## Scope

|  |  |
| --- | --- |
|  | The project covers the creation of a relational database to store and manage library resources, membership, borrowing rules, and transactions. It includes:   * *Managing books, digital media, and magazines.* * *Tracking borrowing and returns.* * *Enforcing borrowing limits and late fees.* * *Generating various reports for library administration.* * *Implementing client and staff roles with appropriate access levels.* |

## Glossary

|  |  |
| --- | --- |
|  | * ***LMS****: Library Management System* * ***ISBN****: International Standard Book Number* * ***Borrowing Limit****: The maximum number of items a member can borrow.* * ***Late Fee****: A penalty charged for overdue items.* * ***Client****: A registered library member* * ***Administrator****: Library staff responsible for managing the system* |

# Stakeholders

* ***Library Clients****: End-users who borrow and return items.*
* ***Library Staff****: Employees responsible for managing the inventory and client records.*
* ***Administrators****: Users with full system access to enforce rules and generate reports.*
* ***Database Developers****: Team members responsible for designing, implementing, and maintaining the system.*

# Requirements

## Functional Requirements

|  |  |
| --- | --- |
|  | 1. ***User Management***    * *Register new clients with unique IDs, names, contact details, and membership types.*    * *Maintain a record of membership status and borrowing privileges.* 2. ***Item Management***    * *Store book and media details (title, author, ISBN, publication year, genre, availability status).*    * *Store magazine details (title, issue number, publication date, availability status).* 3. ***Borrowing & Returning***    * *Track borrowing transactions with timestamps and responsible clients.*    * *Enforce borrowing limits based on membership type.*    * *Automatically calculate late fees for overdue items.* 4. ***Reservations & Notifications***    * *Allow clients to reserve borrowed items.*    * *Notify clients of upcoming due dates and overdue items.*    * *Notify clients when reserved items become available.* 5. ***Reporting & Queries***    * *Generate reports on:*      + *Book availability*      + *Membership activity*      + *Borrowing trends*      + *Fine calculations*      + *Most borrowed books/authors*      + *Overdue items and fines*    * *Identify frequent borrowers and analyze borrowing behavior.* |

## Data Entities

|  |  |
| --- | --- |
| Badge Tick1 with solid fill | * ***Clients*** *(ClientID, Name, ContactInfo, MembershipType, AccountStatus)* * ***Books*** *(BookID, Title, Author, ISBN, PublicationYear, Genre, Availability)* * ***Magazines*** *(MagazineID, Title, IssueNumber, PublicationDate, Availability)* * ***DigitalMedia*** *(MediaID, Title, Creator, Type, Availability)* * ***BorrowingRecords*** *(TransactionID, ClientID, ItemID, BorrowDate, DueDate, ReturnDate, LateFee)* * ***Reservations*** *(ReservationID, ClientID, ItemID, RequestDate, NotificationStatus)* |

# Requirements

## Database Management System

|  |  |
| --- | --- |
|  | MariaDB (on EECS servers) or MySQL (local/cloud-based) |

## Hardware Requirements

|  |  |
| --- | --- |
|  | * *Server with at least 4GB RAM and 100GB storage.* * *Backup storage for daily snapshot.* |

**Software Requirements**

* *SQL-based database engine*
* *SQL client tools for queries and administration*
* *GitHub for version control and repository management*