

**Instructor: Ms. Itrat Fatima**

**Course Name :Introduction to Database System**

**Section No: 02**

**Majid khan Bse241078**

**Khuzaima fariq Bse241078**

**Waleed babar Bse241061**

**Library Management System**

[**Library Management System** 1](#_Toc201550538)

[Project Requirements 2](#_Toc201550539)

[**Introduction** 2](#_Toc201550540)

[**User Roles** 2](#_Toc201550541)

[**Functional Requirements** 3](#_Toc201550542)

[**Non-Functional Requirements** 3](#_Toc201550543)

[**User Stories (Scenarios)** 3](#_Toc201550544)

[Entities, Attributes, and Relationships 4](#_Toc201550545)

[**1. User** 4](#_Toc201550546)

[**2. Librarian** 4](#_Toc201550547)

[**3. Book** 5](#_Toc201550548)

[**4. Transaction** 5](#_Toc201550549)

[**Relationships Between Entities** 5](#_Toc201550550)

[ERD: 6](#_Toc201550551)

[Relational Data Model (RDM) 7](#_Toc201550552)

[Tools and Techniques used in your project. 8](#_Toc201550553)

[Implementation Details 8](#_Toc201550554)

9

# Project Requirements

**Introduction**

This project is about designing and developing a Library Management System (LMS) that will help manage the day-to-day operations of a library. The idea is to reduce manual work, make the system more efficient, and provide a simple and secure way to manage books, users, and transactions like issuing and returning books.

Project Objectives

* To automate the process of issuing and returning books.
* To keep track of books available in the library.
* To allow students and faculty to search books easily.
* To calculate fines for late returns automatically.
* To provide separate roles for Admin, Librarian, and Students with different access.

**User Roles**

There are three main types of users in the system:

1. Admin
   * Can add/edit/delete users and librarians.
   * Can view all transactions and reports.
   * Manages system settings and roles.
2. Librarian
   * Manages books (add, update, delete).
   * Issues and returns books to users.
   * Keeps track of due dates and fines.
3. Student/User
   * Searches for books by title, author, or category.
   * Requests to borrow books.
   * Views issued books, due dates, and fines.

**Functional Requirements**

1. User registration and secure login system.
2. Admin can manage user roles and librarian accounts.
3. Librarian can manage book records (add/edit/delete).
4. Users can search for books using various filters.
5. Books can be issued to users with a due date.
6. When books are returned, the system records the return date.
7. If a book is returned late, a fine is automatically calculated.
8. Reports can be generated for:
   * Books issued
   * Overdue books
   * Inventory status
   * Usage statistics

**Non-Functional Requirements**

* The system should handle at least 50 users at the same time.
* Only authorized users can access certain parts of the system.
* The system should be available all the time (24/7).
* Data must be backed up regularly to avoid loss.
* Interface should be clean and user-friendly.
* The system should be easy to maintain and update in the future.

**User Stories (Scenarios)**

* As a student, I want to search for books by title or author, so I can find what I need easily.
* As a librarian, I want to manage book transactions, so I can issue and return books without confusion.
* As an admin, I want to see reports about library activity, so I can keep track of everything going on.

**Scope of the Project**

**In Scope:**

* Managing books and users
* Issuing and returning books
* Fine calculation for late returns
* Reporting system for usage, overdue, and inventory

**Out of Scope:**

* Online book reading
* Payment gateway for fines
* Multi-language support

**Technologies**

* Frontend: HTML, CSS, JavaScript
* Backend: PHP
* Database: MySQL

# Entities, Attributes, and Relationships

**1. User**

Represents a general system user (student or faculty member).

Attributes:

* UserID (Primary Key)
* Name
* Email
* Password
* Role *(e.g., Student or Faculty)*

**2. Librarian**

Represents a librarian who handles book transactions.

Attributes:

* LibrarianID (Primary Key)
* Name
* Email
* Phone
* ShiftTime

**3. Book**

Represents the book records available in the library.

Attributes:

* BookID (Primary Key)
* Title
* Author
* ISBN
* Category
* Status *(Available / Issued)*

**4. Transaction**

Represents each time a book is issued or returned.

Attributes:

* TransactionID (Primary Key)
* UserID (Foreign Key – refers to User)
* BookID (Foreign Key – refers to Book)
* LibrarianID (Foreign Key – refers to Librarian)
* IssueDate
* DueDate
* ReturnDate
* Fine

**Relationships Between Entities**

| Relationship | Description |
| --- | --- |
| User —< Transaction | A single user can have many transactions (i.e., borrow many books over time). |
| Book —< Transaction | A single book can appear in many transactions (issued multiple times). |
| Librarian —< Transaction | A librarian handles many transactions (issues and returns books to/from users). |
| **ERD:** Description: C:\Users\malik\AppData\Local\Packages\5319275A.WhatsAppDesktop_cv1g1gvanyjgm\TempState\10C72A9D42DD07A028EE910F7854DA5D\WhatsApp Image 2025-05-19 at 21.18.36_4ac0cf4a.jpg Relational Data Model (RDM) **User ( UserID PRIMARY KEY, Name,Email, Password,Role )**  Librarian (LibrarianID PRIMARY KEY, Name, Email,Phone, ShiftTime)  Book ( BookID PRIMARY KEY,Title, Author,ISBN,Category,Status -- (Available / Issued))  Transaction ( TransactionID PRIMARY KEY,UserID FOREIGN KEY REFERENCES User(UserID), BookID FOREIGN KEY REFERENCES Book(BookID),LibrarianID FOREIGN KEY REFERENCES Librarian(LibrarianID),IssueDate,DueDate,ReturnDate,Fine) Tools and Techniques used in your project. **Frontend:** HTML, CSS, JavaScript   * **Backend:** PHP * **Database:** MySQL * **Development Platform:** VS Code / XAMPP  Implementation Details This project was developed using simple web technologies to manage library operations like book issue/return, fine calculation, and user roles.  **1. Login System**   * Users (Admin, Librarian, Student) can register and log in. * Login redirects users to their respective dashboards. * Role-based access is used to control features.   **2. Book Management**   * Librarians can add, edit, or delete books. * Book data includes title, author, ISBN, and status (Available/Issued). * Admins can also manage books.   **3. Search Module**   * Students can search books by title, author, or category. * Only available books are shown to users.   **4. Issue/Return Books**   * Librarians can issue books with a due date. * On return, the system checks the return date and updates book status.   **5. Fine Calculation**   * Late returns are fined (e.g., Rs. 10 per day). * Fine is shown during return and saved in transaction history.   **6. Reports**   * Admin can see reports for:   + Overdue books   + Total issued/returned books   + Inventory status |  |