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	VARCHAR(13) NOT NULL, VARCHAR(255), YEAR, etc..	Stores book details (physical or digital)	RIMARY KEY (ISBN), unique ISBN, Availability could be an ENUM.
DigitalMedia	MediaID (PK), Title, Creator, MediaType, Availability	INT UNSIGNED AUTO_INCREMENT, VARCHAR(255), etc..	Tracks digital media items (e.g., eBooks, audiobooks, DVDs)	PRIMARY KEY (MediaID), unique ID, Availability could be an ENUM.
Magazine	IssueID (PK), Title, IssueNumber, PublicationDate, Status	INT UNSIGNED AUTO_INCREMENT, VARCHAR(255), DATE	Manages magazines	PRIMARY KEY (IssueID), unique ID, Status could be an ENUM.
Member	MemberID (PK), Name, Contact, TypeID (FK), AccountStatus	INT UNSIGNED AUTO_INCREMENT, VARCHAR(255), etc.	Stores patron info (name, address/email, membership details)	PRIMARY KEY (MemberID), references MembershipType.TypeID
MembershipType	TypeID (PK), TypeName, BorrowLimit, LateFeeRate	INT UNSIGNED AUTO_INCREMENT, VARCHAR(50), etc..	Defines membership categories (Student, Regular, Senior) & associated borrowing rules	RIMARY KEY (TypeID), unique name, BorrowLimit e.g. TINYINT.
Loan	LoanID (PK), MemberID (FK), ItemID, ItemType, LoanDate,	INT UNSIGNED AUTO_INCREMENT, DATE, etc.	Records borrowing transactions; references correct item	PRIMARY KEY (LoanID), references MemberID, references FineID.

	DueDate, ReturnDate, FineID (FK)		via (ItemID, ItemType)	
Fine	FineID (PK), LoanID (FK), Amount, PaymentStatus	INT UNSIGNED AUTO_INCREMENT, DECIMAL(6,2), etc.	Holds fines for overdue items.	PRIMARY KEY (FineID), references LoanID, PaymentStatus could be an ENUM.
Reservation	ReservationID (PK), MemberID (FK), ItemID, ItemType, RequestDate, Status	INT UNSIGNED AUTO_INCREMENT, DATE, etc.	Tracks reservations on any item (book, media, magazine)	PRIMARY KEY (ReservationID), references MemberID.
Librarian	LibrarianID (PK), Name, Contact, Role	INT UNSIGNED AUTO_INCREMENT, VARCHAR(255), etc.	Stores librarian/staff info	PRIMARY KEY (LibrarianID), Role could be ENUM('Admin','Staff').