

STUDENT MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by:

Rishabh-23BCS10443

Nagesh-23BCS10608

Ansh-23BCS10612

Yash-23BCS10616

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTERSCIENCE&ENGINEERING



Chandigarh University

April ,2025

TABLE OF CONTENTS

List of Figures.....	
CHAPTER1. INTRODUCTION.....	5
Introduction to Project.....	5
Identification of Problem.....	6
CHAPTER2. BACKGROUND STUDY	7
Existing solutions	7
Problem Definition	8
Goals/Objectives	8
CHAPTER3. DESIGN FLOW/PROCESS	9
Evaluation & Selection of Specifications/Features	9
Analysis of Features and finalization subject to constraints	9
Design Flow	12
CHAPTER4. RESULTS ANALYSIS AND VALIDATION	13
Implementation of solution.....	13
ER Model for student management system.....	14
CHAPTER5. CONCLUSION AND FUTURE WORK.....	15
Conclusion	15
Future work.....	

CHAPTER1. INTRODUCTION

- **Introduction to Project :**

The **Library Management System (LMS)** is a robust database solution designed to efficiently manage and organize library operations within educational or public institutions. It offers a centralized platform for handling a wide array of library data, including book inventories, member registrations, issue-return transactions, staff details, and fine management. By automating routine processes and maintaining structured records, the system enhances operational efficiency, minimizes manual work, and ensures the accuracy and availability of information. The LMS supports easy access for librarians, staff, and members, ultimately improving user experience and enabling better resource management and decision-making within the library ecosystem.

- **Identification of Problem**

Educational institutions and public libraries often face considerable challenges in managing library operations manually. These challenges include:

- a) **Redundancy and Inconsistency:** Manual book tracking and member data entry often result in duplicate records and inconsistencies.
- b) **Increased Error Rate:** Human errors in tracking issued/returned books or updating records can lead to data inaccuracy and confusion.
- c) **Time-Consuming Processes:** Manually managing book inventory, issuing, returning, and fine calculation is labor-intensive and inefficient.
- d) **Difficulty in Data Retrieval:** Locating specific book or member details becomes tedious and slow in paper-based or loosely maintained systems.
- e) **Lack of Integration:** Using separate ledgers or tools for inventory, members, and transactions causes fragmentation, making it hard to view complete library activity.

A centralized **Library Management System** addresses these issues by providing an integrated and automated platform. It reduces redundancy, ensures data accuracy, simplifies workflow, and enhances overall productivity and user experience.

CHAPTER 2. BACKGROUND STUDY

- **Existing Solutions**

Currently, several proprietary and open-source Library Management Systems are available in the market. Systems like Koha, Libsys, and Evergreen offer various features to support library operations. However, these solutions often come with certain limitations:

- a) **High Cost:** Proprietary library systems usually involve substantial licensing and maintenance fees, making them less accessible for smaller libraries or institutions.
- b) **Limited Customization:** Many off-the-shelf solutions offer predefined features that may not align with the unique workflows and policies of individual libraries.
- c) **Complex Interfaces:** Some existing systems are not intuitive, requiring extensive training for librarians and staff.
- d) **Scalability Issues:** Open-source platforms may lack the scalability and support needed for expanding institutions or libraries with evolving demands.

Developing a custom Library Management System allows institutions to create a flexible, cost-effective, and user-friendly platform that aligns precisely with their needs, improving operational efficiency and user satisfaction.

Problem Definition

The primary problem is the absence of an integrated and customizable database system that can efficiently manage all aspects of student information in educational institutions. This includes the need to:

- a) Centralize and streamline student and teacher data management
- b) Automate class structures and subject locations
- c) Accurately track attendance
- d) Schedule exams and generate results systematically
- e) Ensure quick data access with minimal errors or duplication

The project aims to design and implement a robust, scalable, and user-friendly student management database system that resolves these challenges and meets the institution's operational needs.

- **Goals/Objectives**

- a) Efficient storage and retrieval of book, member, and staff data
- b) Streamlined book issue and return management
- c) Automated fine calculation and report generation
- d) Real-time tracking of book availability and status
- e) Enhanced user experience through a centralized and user-friendly interface.

CHAPTER3. DESIGNFLOW/PROCESS

- **Evaluation & Selection of Specifications/Features**

The project includes the following features:

- a) Member Registration and Management
- b) Book Cataloging and Categorization
- c) Book Issue and Return Tracking
- d) Fine Calculation for Overdue Returns
- e) Real-time Book Availability Status
- f) Staff and Branch Information Management

- **Analysis of Features and Finalization Subject to Constraints**

The Library Management System is designed with a strong emphasis on data accuracy, integrity, and operational efficiency. SQL is used as the backend database language to define, store, and manipulate data structures. During the design phase, the following considerations were made, subject to key database constraints:

- a) **Relational Integrity:** Maintained using PRIMARY KEY and FOREIGN KEY constraints to ensure unique identification and proper linkage between entities like books, members, and transactions.
- b) **Data Normalization:** Up to the third normal form (3NF) was applied to eliminate redundancy and maintain organized, consistent data.
- c) **Field-Level Constraints:** Used NOT NULL, UNIQUE, CHECK, and DEFAULT constraints to validate input and maintain data correctness.
- d) **Scalability and Flexibility:** The schema was designed to support future additions like digital resources, advanced search filters, and analytics features.

These design choices ensure a reliable, scalable, and error-resistant system for managing diverse and dynamic library operations.

- **Design Flow**

Tables were created to represent **Books, Members, Staff, Issue Records, Return Records, and Branches**. Each table is interlinked using appropriate FOREIGN KEY constraints to maintain data relationships. Queries were written to retrieve member activity, track book availability, and calculate fines. CASE statements were used in queries for conditional fine calculation and book status display.

CHAPTER 4. RESULTS ANALYSIS AND VALIDATION

Implementation of Solution:

The system was implemented using **SQL**. Sample data was inserted into all major tables including books, members, staff, issue, and return records. Final SQL queries were written to fetch details like issued books, return dates, and calculate fines for overdue returns.

Example: For member "*Asha Rani*", issued book details were retrieved, along with return status and any applicable fine, using JOIN and CASE statements.

CODE :

-- LibraryBranch Table

```
CREATE TABLE LibraryBranch (  
    BranchID VARCHAR(10) PRIMARY KEY,  
    HeadLibrarianID VARCHAR(10),  
    Location VARCHAR(120),  
    PhoneNumber VARCHAR(15)  
);
```

-- Staff Table

```
CREATE TABLE Staff (  
    StaffID VARCHAR(10) PRIMARY KEY,  
    FullName VARCHAR(60),  
    Role VARCHAR(30),  
    MonthlyPay NUMERIC(10,2),  
    BranchID VARCHAR(10),  
    FOREIGN KEY (BranchID) REFERENCES LibraryBranch(BranchID) ON DELETE SET NULL  
);
```

-- Members Table

```
CREATE TABLE Members (  
    MemberID VARCHAR(10) PRIMARY KEY,  
    Name VARCHAR(60),  
    Address VARCHAR(120),  
    MembershipDate DATE  
);
```

-- LibraryBooks Table

```
CREATE TABLE LibraryBooks (  
    BookCode VARCHAR(25) PRIMARY KEY,
```



```
Title VARCHAR(90),
Genre VARCHAR(40),
DailyRentalRate NUMERIC(10,2),
AvailableStatus VARCHAR(5),
Writer VARCHAR(60),
PublishingHouse VARCHAR(60),
AddedDate DATE
);
```

-- BookIssues Table

```
CREATE TABLE BookIssues (
    IssueCode VARCHAR(10) PRIMARY KEY,
    MemberID VARCHAR(10),
    BookCode VARCHAR(25),
    DateIssued DATE,
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID) ON DELETE CASCADE,
    FOREIGN KEY (BookCode) REFERENCES LibraryBooks(BookCode) ON DELETE CASCADE
);
```

-- BookReturns Table

```
CREATE TABLE BookReturns (
    ReturnCode VARCHAR(10) PRIMARY KEY,
    MemberID VARCHAR(10),
    BookCode VARCHAR(25),
    DateReturned DATE,
    Penalty NUMERIC(6,2),
    FOREIGN KEY (BookCode) REFERENCES LibraryBooks(BookCode) ON DELETE CASCADE
);
```

SAMPLE DATA:

-- Insert Branches

```
INSERT INTO LibraryBranch VALUES ('LB01', 'L201', 'Sector 10, Noida', '+911234567890');
INSERT INTO LibraryBranch VALUES ('LB02', 'L202', 'Jayanagar, Bengaluru', '+919812345678');
```

-- Insert Staff

```
INSERT INTO Staff VALUES ('S101', 'Anjali Mehta', 'Head Librarian', 58000.00, 'LB01');
INSERT INTO Staff VALUES ('S102', 'Rohit Das', 'Assistant', 32000.00, 'LB02');
```

-- Insert Members

```
INSERT INTO Members VALUES ('M001', 'Tanya Singh', 'Sector 10, Noida', '2022-03-01');
```

```
INSERT INTO Members VALUES ('M002', 'Karan Joshi', 'Jayanagar, Bengaluru', '2023-07-11');
```

```
-- Insert Books
```

```
INSERT INTO LibraryBooks VALUES ('BK101', 'Sapiens', 'History', 8.00, 'Yes', 'Yuval Noah Harari', 'Harper', '2021-08-20');
```

```
INSERT INTO LibraryBooks VALUES ('BK102', '1984', 'Dystopian', 6.50, 'Yes', 'George Orwell', 'Penguin', '2020-05-15');
```

```
-- Insert Book Issues
```

```
INSERT INTO BookIssues VALUES ('ISS001', 'M001', 'BK101', '2024-02-10');
```

```
-- Insert Book Returns
```

```
INSERT INTO BookReturns VALUES ('RET001', 'M001', 'BK101', '2024-03-01', 0.00);
```

```
-- Another Issue
```

```
INSERT INTO BookIssues VALUES ('ISS002', 'M002', 'BK102', '2024-04-10');
```

CREATING A VIEW :

```
CREATE VIEW MemberActivity AS
```

```
SELECT
```

```
    M.MemberID,  
    M.Name AS MemberName,  
    B.Title AS BookTitle,  
    B.Genre,  
    B.Writer,  
    BI.DateIssued,  
    BR.DateReturned,  
    BR.Penalty
```

```
FROM Members M
```

```
LEFT JOIN BookIssues BI ON M.MemberID = BI.MemberID
```

```
LEFT JOIN BookReturns BR ON M.MemberID = BR.MemberID AND BI.BookCode =  
BR.BookCode
```

```
LEFT JOIN LibraryBooks B ON B.BookCode = BI.BookCode;
```

SAMPLE QUERIES

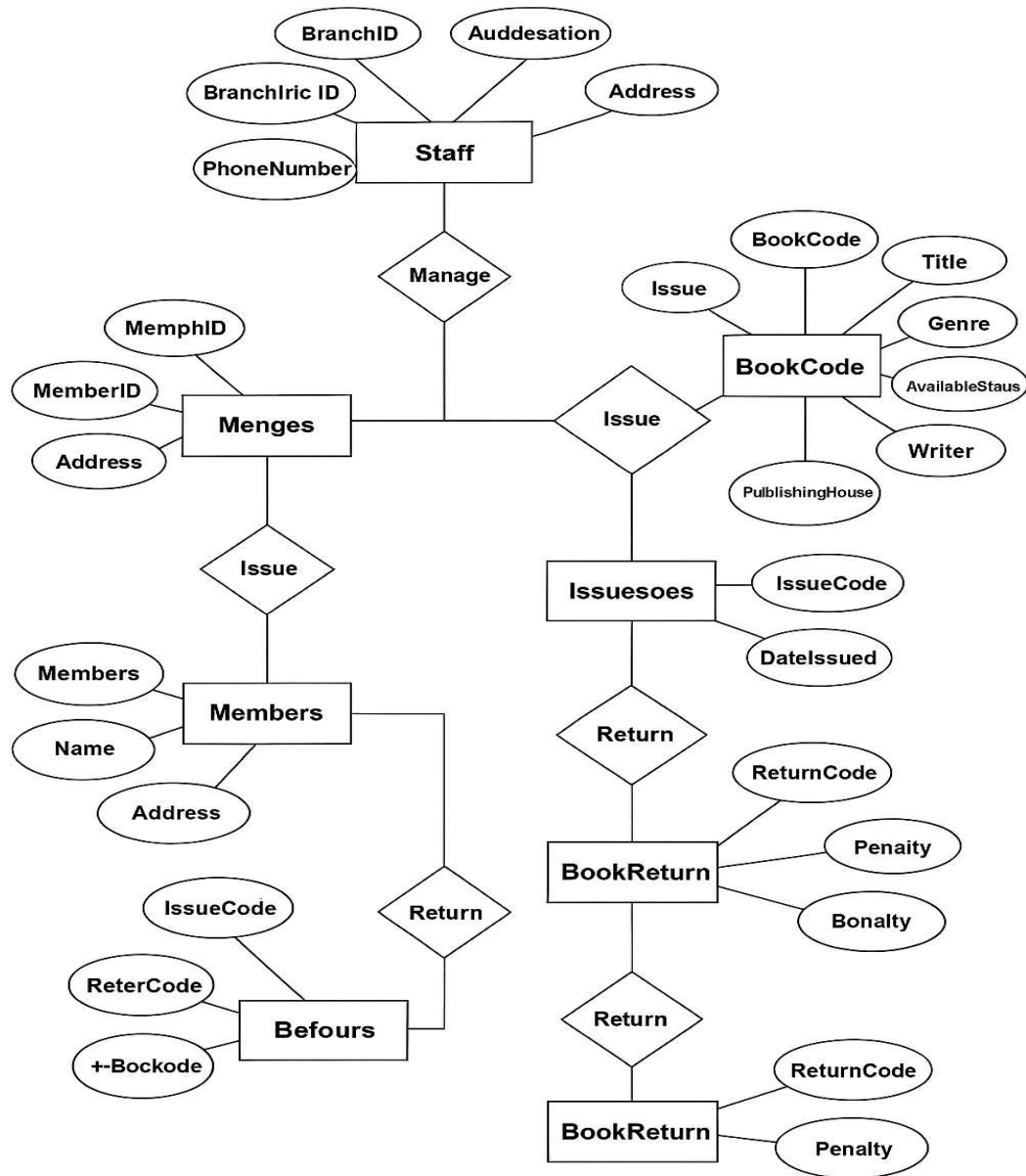
```
SELECT * FROM LibraryBranch;
```

```
SELECT * FROM Staff;
```

```
SELECT * FROM Members;
```

```
SELECT * FROM LibraryBooks;  
SELECT * FROM BookIssues;  
SELECT * FROM BookReturns;  
SELECT * FROM MemberActivity;
```

ER MODEL



ER Diagram for Library Management System

CHAPTER 5. CONCLUSION AND FUTURE WORK

- **Conclusion**

The Library Management System project effectively demonstrates the advantages of using structured databases to manage library operations efficiently. By leveraging SQL for data definition and manipulation, the system ensures:

- a) **Efficient Data Management:** All book and member records are organized systematically for easy access and updates.
- b) **Accurate Fine and Status Handling:** SQL logic, including CASE statements, allows precise calculation of fines and tracking of book status.
- c) **Data Integrity and Security:** Use of PRIMARY KEY and FOREIGN KEY constraints maintains relational integrity and prevents data inconsistencies.
- d) **Reduced Redundancy:** Data normalization eliminates duplication, ensuring clean and scalable database design.

Overall, this project provides a strong, extensible foundation for a reliable library system that can evolve with future requirements.

- **Future Work**

To enhance the usability and functionality of the Library Management System in real-world applications, the following improvements are proposed:

- a) **Integration with a Webor Desktop Interface:**
Develop a user-friendly front-end application that allows librarians and members to interact with the system through an intuitive interface without requiring SQL knowledge.
- b) **Automated SMS/Email Alerts:**
Implement a notification system to send due date reminders, return alerts, and fine notices to members automatically, improving communication.
- c) **Analytics Dashboard:**
Introduce a dashboard with data visualizations for book circulation trends, most issued books, member activity, and overdue patterns to support informed decision-making.
- d) **Role-Based Access Control(RBAC):**
Add a secure login system with defined access levels for librarians, members, and administrators to ensure data privacy and prevent unauthorized actions.