

CSE302/ICE305: Database Systems (Section No.6) [Summer-25]

Project Report

**Project Name**: Library Management System.

Project URL/Link: <https://github.com/2023-1-50-008-ops/library-management-system>

## Submitted by:

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student Name** | **Contribution Percentage** |
| 2023-1-50-008 | Kazi Ratul | 35% |
| 2022-3-50-031 | Fatima Tuz Zohura | 35% |
| 2020-2-50-011 | Mobasshera Noushin | 30% |
|  |  |  |
|  |  |  |

*.*

# Project Description

This project is about designing and implementing a Library Management System for East West University. The system will allow efficient management of library operations such as maintaining records of books, users, borrowing/returning of books, and tracking overdue items. It simulates how a real university library functions but in a structured, digital database format.

**Real-World Scenario:**

In a real-world university library, thousands of books, journals, and digital resources need to be maintained. Students and faculty members regularly borrow and return these resources. The librarian must track availability, manage user records, calculate fines for late returns, and generate useful reports. This project simulates such a university library environment, where books, members, and transactions are systematically stored and managed in a database to reduce manual effort and errors.

**Main Purpose & Data Handled by the System:**

* To simplify and automate library operations.
* To keep accurate records of books, members, and borrowing history.
* To reduce paperwork and improve efficiency.
* To provide quick access to book availability and member details.

**Types of Data the System Will Handle:**

**User Data:** Student ID, Name, Email, Role (admin/member).

**Member Records**: Member ID, Contact details, linked user account.

**Book Records**: Book ID, Title, Author, Publisher, Category, Availability status.

**Borrow/Return Transactions**: Borrow ID, Book ID, Member ID, Borrow date, Due date, Return date, Fines.

**Admin Data**: Admin credentials, ability to manage books and member records.

.(Insert ,Update, Delete)

# 

# 2.Key Features

# User Authentication: Separate login for admin (librarian) and users (students).

# Book Management: Add, update, search, and delete books.

# User Management: Search book ,Borrow a book ,Return a book, Show all book.

# Issuing and Returning Books:Track issued books, due dates, and returns.

# Fine Calculation: Automatic fine calculation for late returns.

# Search Functionality: Search for books by title, author, or category.

# Reports: Generate reports (issued/returned books, overdue books, etc

.

# Database Design

## E-R Model

## (Entities and Attributes)

##  Users

## Attributes: user\_id (PK), username, password, role (admin or member)

## Description: Represents system users (admins and members).

##  Members

## Attributes: member\_id (PK), user\_id (FK), name, email, phone, join\_date

## Relationship: Each member is linked to a user account.

## Type: One-to-One (1:1) with Users (since user\_id is unique per member)

##  Books

## Attributes: book\_id (PK), title, author, publisher, year, available\_copies

## Description: Represents books in the library.

##  Borrow

## Attributes: borrow\_id (PK), member\_id (FK), book\_id (FK), borrow\_date, return\_date

## Relationship:

## Member to Borrow → One-to-Many (1 member can borrow multiple books)

## Book to Borrow → One-to-Many (1 book can be borrowed multiple times)

## Deletion Rule: Cascades applied

| Relationship | Type | Notes |
| --- | --- | --- |
| Users → Members | 1:1 | Each user can be linked to one member |
| Members → Borrow | 1:N | A member can borrow many books |
| Books → Borrow | 1:N | A book can be borrowed multiple times |

## 

## 

## It shows the main entities of a library system and how they are related. There are four entities: Users, Members, Books, and Borrow. Each entity has attributes, and the relationships between them are shown with 1:1 or 1:N connections. For example, one user can be one member, one member can borrow many books, and one book can be borrowed many times. This diagram is mainly for conceptual understanding of the system’s structure.

## Relational Data Model

**Converted Tables from ERD**

**1. Users Table**

Primary Key (PK): user\_id

Attributes:

* user\_id (PK)
* username
* password
* role

**2. Members Table**

Primary Key (PK): member\_id

Foreign Key (FK): user\_id → references Users(user\_id)

Attributes:

* member\_id (PK)
* user\_id (FK)
* name
* email
* join\_date

**3. Books Table**

Primary Key (PK): book\_id

Attributes:

* book\_id (PK)
* title
* author
* publisher
* year
* available\_copies

**4. Borrow Table**

Primary Key (PK): borrow\_id

Foreign Keys (FK):

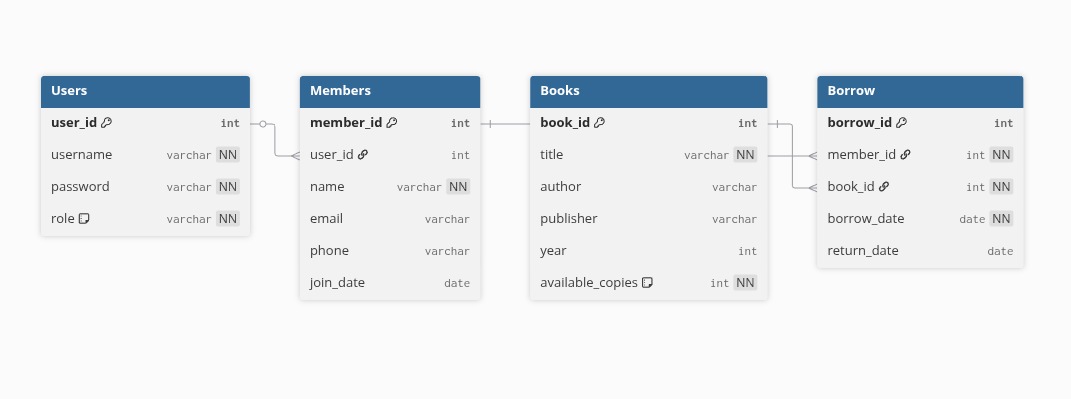
member\_id → references Members(member\_id)

book\_id → references Books(book\_id)

Attributes:

* borrow\_id (PK)
* member\_id (FK)
* book\_id (FK)
* return\_date

**Relationships in Table Form**

* One User → One Member (1:1 via user\_id)
* One Member → Many Borrow (1:N via member\_id)
* One Book → Many Borrow (1:N via book\_id)

It shows the same design but in table format with actual database details. Each entity from the ERD is converted into a table with columns. Primary Keys (PK) and Foreign Keys (FK) are clearly marked, and data types like varchar, int, and date are added. For example, the Borrow table has borrow\_id as the primary key and member\_id and book\_id as foreign keys. This schema is more technical and ready for implementation in SQL

# Tools & Technologies Used

# MySQL / SQL – Online Editor

# Git, GitHub

# Draw.io

# ChatGPT.

# Diagram-GPT

# 

# 4.Role Assignment

1**. Defined User Roles**

In this system, there are two types of users:

* Admin
* Member (User)

**2. Role Descriptions and Responsibilities**

**For-Admin**

* Has full control of the library system.
* Can add, update, or delete books from the catalog.
* Can manage users, including adding new members and removing inactive ones.
* Can view borrowing records of all members.
* Ensures the system is properly maintained and up to date.

**Member (User)**

* Can log in with their unique account.
* Can search and view books available in the library.
* Can borrow books and view their own borrowing history.
* Can return books, after which the record is updated.
* Limited to their own account; they cannot modify other members’ data or the system’s structur4

# 5.GUI Screenshot( What My Project Can Do)

# Create table of ( Users(admin+member), Members( Linked to user) , Books, Borrow Records.

# 

# 

# 

# 

# 

# 

# 

# 

# Example question for Admin vs users

# Show all users & their roles

# 

# Now For the Admin ( What he can do)

# \*Admin can add a new book ,

# 

# \*Admin can update book copies

# 

# \*.Delete a book from the library

# 

# \*View all members and their details

# 

# Now For Users

# \*Show all available books,

# 

# \*Search books by author name

# 

# \*Borrow a book (insert record into Borrow table)

# 

# \*Return a book (update return\_date)

# 

# Conclusion

The East West University Library Management System project provided valuable hands-on experience in designing and implementing a real-world database system. Through this project, I learned how to translate practical library operations—such as managing books, tracking users, handling borrowing/returning, and calculating fines—into structured database tables and relationships.

One of the main challenges faced was ensuring proper relationships among entities, particularly in maintaining referential integrity between users, members, books, and borrow records. Designing the ER diagram and converting it into a relational schema required careful thought to avoid redundancy and maintain normalization. Another challenge was handling role-based access, where admins have full control, while members are restricted to basic borrowing and search operations.

This project not only strengthened my SQL and database design skills but also gave me insights into how digital systems can simplify manual tasks and improve efficiency.

For future improvements, the system could be enhanced by:

Implementing a fine calculation module that automatically updates overdue penalties.

Adding a book reservation system for users.Introducing search optimization with filters by category, author, or year.Integrating the database with a front-end interface (such as a web or mobile app) for easier real-world usage.

Overall, this project successfully demonstrates how a structured database system can simulate and improve a real university library environment.

# 

# References

# Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Database System Concepts (7th Edition). McGraw-Hill Education.

# Core textbook for database design, relational model, SQL queries.

# Date, C. J. (2004). An Introduction to Database Systems (8th Edition). Addison-Wesley.

# Widely used book for understanding relational databases.

# Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems (3rd Edition). McGraw-Hill.

# Good for ER diagrams, relational schema, and normalization.

# MySQL Documentation. MySQL 8.0 Reference Manual. Oracle Corporation.

# Available at: https://dev.mysql.com/doc/

# Draw.io – Online Diagramming Tool for ER Diagrams.

# Available at: https://app.diagrams.net/

* ChatGPT (OpenAI) – Used for guidance and explanation during project development.

END