

Readme file

Project Objective:

The primary objective of this project is to design and implement a robust library management application that facilitates book borrowing and returning for users, alongside providing librarians with the necessary tools to efficiently manage user accounts.

Acknowledgments:

We would like to thank Professor Yashvardhan Sharma and Professor Amit Dua as well as our mentor Sir Shivam Lahoria for their guidance and support during the development of this application.

Features:

This app integrates multiple features aimed at enhancing the overall library experience for its users. These features include:

- Intuitive book search capabilities
- User-friendly borrowing and returning processes
- Personalized account management options
- Comprehensive librarian tools for overseeing user activities and book inventory

By combining these features, the app aims to emulate the traditional library experience while leveraging the convenience and accessibility of modern technology.

a) User Groups:

The application serves two primary user groups:

- Students (Users) : Users manage their accounts and access library resources.
- Librarians: Librarians oversee system administration, manage library resources, and assist users.

b) User Functionalities:

Users can:

1. Register for a library member account
2. Withdraw and return books
3. Search the library catalog by book name, author, or genre
4. View book availability status, due dates, and locations
5. Place holds on unavailable books
6. Renew borrowed books if available
7. View borrowing history
8. Update account information (password, profile)
9. Rate previously issued books

c) Librarian Functionalities:

Librarians can:

1. Search and update the library catalog
2. Add, edit, or delete book entries
3. Accept book returns from users
4. Register new user accounts
5. Edit user details
6. Manage user accounts (lock/unlock) based on overdue fines
7. Set fines for overdue books and manage borrowing limits

d) Main Functionalities:

- *User Login*: Enables users to access library accounts to utilize library services.
 - *Book Issue/Hold*: Allows users to borrow available books and place holds on unavailable ones, ensuring priority access upon availability.
 - *User Management*: Allows librarians to create, edit, and manage user accounts, including registration details.
 - *Catalog Management*: Enables librarians to keep the library catalog up-to-date by adding, editing, or removing book entries, ensuring reliable information for users.
- Rating System*: Allows users to provide ratings on books, enhancing the book selection process for other users.

- *Track Borrowing History*: Provides users with a detailed view of their borrowing history, including their current penalty, due dates, and renewals, aiding in the management of borrowed items.

Technologies Used:

Backend: MySQL

Frontend: Java Swing and AWT

Connection: JDBC

Project Development Steps:

a) Designing ER Model and Relational Schema:

We first created an Entity-Relationship (ER) model encompassing all the entities, their relations, and fields. Next, we translated the ER model into a corresponding relational schema, defining primary keys for each table, functional dependencies, and foreign key constraints.

b) Database Setup and Population:

Using MySQL, we implemented the tables based on the relational schema created earlier. The tables were populated with initial values to simulate a functioning library database.

c) Implementation of Procedures:

Procedures were developed to implement all the desired functionalities of the library management system. These procedures included actions such as book issue/return, user management, catalog management, etc. Detailed Explanation of some major procedure are given below:

1) withdrawBook:

This procedure ensures a smooth borrowing process. Firstly, it checks if the user's account is locked to prevent unauthorized borrowing. Then, it verifies if the requested book exists in the library's inventory and if it's available for borrowing. If the book is available, a new entry is added to the "borrows" table to record the borrowing transaction. Additionally, the details of the

book in the inventory are updated to reflect that it has been issued to the user.

2) returnBook:

When a user returns a book, this procedure is invoked. It checks if the user has indeed borrowed the book they are returning. If the return exceeds the due date, a fine is imposed, and a record is added to the penalty table. After the return, the status is set to pending until a librarian approves of it.

3) placeHold:

This procedure allows users to place holds on books. It verifies if the user's account is unlocked and if the requested book is available for borrowing. If the book is not available, the user can place a hold on it. A record is then inserted into the holds table to indicate that a hold has been successfully placed by the user.

4) addBook:

Librarians use this procedure to add new books or copies of existing ones to the library's inventory. It performs checks to ensure that the book, author, publisher, shelf, and genre exist in their respective tables. If not, new entries are added to corresponding tables. Once the book is successfully added, it is also added to the inventory.

5) approveBookReturn:

This procedure is utilized by librarians to manage return requests from users. It focuses on requests that are in a pending approval state. When invoked, it updates the approval status of all pending return requests to "approved." Additionally, it handles scenarios where other users have placed holds on the returned books. In such cases, the first person to place the hold is identified, and the book whose return is approved is assigned to them. This ensures efficient management of book returns and holds, streamlining the library's operations.

d) Frontend Development using Java Swing:

In the frontend, we utilized Java Swing to design graphical user interfaces (JFrames) . Elements such as dropdown menus, buttons, text fields, etc., were added to the JFrames to create an intuitive user interface.

e) Integration of Backend with Frontend:

The final step involved linking all the procedures created in MySQL with the frontend developed using Java Swing. This was achieved by establishing a connection using JDBC between the MySQL database and the Java application, allowing seamless interaction between the user interface and the underlying database functionalities.

Contributors:

- ADITYA MITTAL- 2022A7PS0569P(FRONTEND,BACKEND,ER-MODEL,RELATIONAL SCHEMA,REPORT)
- PRATYUSH GHOSH- 2022A7PS1292P(BACKEND,ER-MODEL,RELATIONAL SCHEMA,REPORT)
- PARTH KHANDELWAL- 2022A7PS0103P(FRONTEND,BACKEND,PPT,RELATIONAL SCHEMA)
- GYANESH ROY BHOWMIK- 2022A7PS0035P(FRONTEND,RELATIONAL SCHEMA,PPT)