

Library Management System using DAO

Objective

In this assignment, you will create a Library Management System using Java and the Data Access Object (DAO) pattern. The system will allow users to manage books and borrowers, as well as track borrowed books. You will interact with a database using JDBC.

Assignment Instructions

1. PROJECT OVERVIEW

You will develop a console-based Java application that performs the following tasks:

- ✓ Manage Books – Add, update, delete, and list books.
- ✓ Manage Borrowers – Add, update, delete, and list borrowers.
- ✓ Manage Borrowing Transactions – Allow borrowers to borrow and return books.

2. DATABASE SETUP

Create an SQLite database with the following tables:

- books
 - id (auto increment pk) , title, author, available_copies
- borrowers
 - id (auto increment pk), name, email
- borrowed_books
 - id (auto increment pk), book_id, borrower_id, borrow_date, return_date

You can either use IntelliJ to help with these tables, or another database program such as DataGrip, HeidiSQL, etc

3. IMPLEMENT DAO CLASSES

You must create separate DAO classes to interact with the database.

BookDAO.java

- addBook(Book book) – Adds a new book.
- updateBook(Book book) – Updates book details.
- deleteBook(int bookId) – Deletes a book.
- getAllBooks() – Retrieves all books.

BorrowerDAO.java

- addBorrower(Borrower borrower) – Adds a new borrower.
- updateBorrower(Borrower borrower) – Updates borrower details.

- deleteBorrower(int borrowerId) – Deletes a borrower.
- getAllBorrowers() – Retrieves all borrowers.

BorrowedBookDAO.java

- borrowBook(int bookId, int borrowerId) – Records a borrowed book and decreases available copies.
- returnBook(int bookId, int borrowerId) – Marks a book as returned and increases available copies.
- getAllBorrowedBooks() – Shows all borrowed books.

4. IMPLEMENT THE APPLICATION LOGIC

Create a LibraryService.java class to handle business logic. This class should:

- ✓ Call DAO methods for book and borrower management.
- ✓ Ensure books can only be borrowed if copies are available.
- ✓ Prevent borrowing the same book multiple times without returning it first.

5. USER INTERACTION (MAIN.JAVA)

The Main.java file should provide a simple menu-driven console interface for users to interact with the system.

Example Menu:

```
Welcome to the Library Management System!
1. Add a new book
2. List all books
3. Update book details
4. Delete a book
5. Add a new borrower
6. List all borrowers
7. Borrow a book
8. Return a book
9. View all borrowed books
0. Exit
Enter your choice:
```

Grading Breakdown

Criteria	Points
DAO implementation (CRUD)	40

Database interaction (JDBC)	20
Borrowing/Returning logic	20
Complete Application	10
Packages	10