**Library Management System**

**Introduction**

The Library Management System is a Java-based application developed to manage books, authors, and book-author relationships in a library. It uses JavaFX for the user interface and MySQL for the database backend.

**System Requirements**

* **Java Development Kit (JDK)**: Version 11 or higher
* **JavaFX**: Included with JDK or installed separately
* **MySQL**: Version 5.7 or higher
* **Maven**: For dependency management and build

**Architecture**

The system follows a simple MVC (Model-View-Controller) pattern:

* **Model**: Represents the database entities (Book, Author, BookAuthor)
* **View**: JavaFX FXML files for the UI (AddBookView.fxml, MainDashboardView.fxml, BookDialog.fxml, EditBookView.fxml,BookListView.fxml)
* **Controller**: Java classes that handle user input and update the model and view (AddBookController, MainDashboardController)

**Installation and Setup**

**Prerequisites**

1. Install JDK 11 or higher.
2. Install MySQL and create a database named librarydb.
3. Set up Maven for dependency management.

**Steps**

1. **Clone the Repository**

git clone < https://github.com/amponsemmichael/backend-development-training/tree/master/Week-1\_oop-%20Databases%20and%20Data%20Structure\_Algorithm >

1. **Set Up the Database**
   * Execute the following SQL script to create the necessary tables:

sql

Copy code

CREATE TABLE book (

id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

category\_id INT,

publication\_date DATE,

copies\_owned INT

);

CREATE TABLE author (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL

);

CREATE TABLE book\_author (

book\_id INT,

author\_id INT,

PRIMARY KEY (book\_id, author\_id),

FOREIGN KEY (book\_id) REFERENCES book(id),

FOREIGN KEY (author\_id) REFERENCES author(id)

);

1. **Configure Application Properties**
   * Update database connection settings in src/main/resources/

db.url=jdbc:mysql://localhost:3306/librarydb

db.username=root

db.password=yourpassword

1. **Build the Project**
2. **Run the Application**

**Database Schema**

**Implemented Tables**

* **Book**
  + id: INT, Primary Key, Auto Increment
  + title: VARCHAR(255), Not Null
  + category\_id: INT
  + publication\_date: DATE
  + copies\_owned: INT
* **Author**
  + id: INT, Primary Key, Auto Increment
  + name: VARCHAR(255), Not Null
* **BookAuthor**
  + book\_id: INT, Primary Key, Foreign Key to book(id)
  + author\_id: INT, Primary Key, Foreign Key to author(id)

**Usage**

**Main Dashboard**

* **Add Book**: Allows the user to add a new book to the library.
* **View Books**: Displays a list of all books in the library.
* **Add Author**: Allows the user to add a new author.
* **View Authors**: Displays a list of all authors.
* **Delete Book**: Allows the user to delete from the library database

**Adding a Book**

* Navigate to the Add Book screen.
* Fill in the book details (title, category, publication date, copies owned).
* Click "Save" to add the book to the database.

**Adding an Author**

* Navigate to the Add Author screen.
* Fill in the author details (name).
* Click "Save" to add the author to the database.

**Deleting a Book**

* Click on **delete** confirm from the prompt. Click on cancel to stop.

**Edit a Book**

* Click on the Edit button from the **maindashboard**.
* Fill in the book details (title, category, publication date, copies owned).
* Click "Save" to change the book details in the database or cancel to stop the edit action

**Controllers and Views**

**Working Controllers**

* **AddBookController.java**
  + Handles logic for adding a new book.
  + Interacts with the database to save book details.
  + Edit a book delete
  + Delete a book from the library’s database
* **MainDashboardController.java**
  + Manages the main dashboard view.
  + Provides navigation between different views (add book, view books, edit book details, delete book).

**Views**

* **AddBookView.fxml**
  + FXML file for the add book interface.
  + Includes input fields for book details and a save button.
* **MainDashboardView.fxml**
  + FXML file for the main dashboard interface.
  + Includes navigation buttons for different functionalities.

**Troubleshooting**

**Common Issues**

* **Location is not set Error**:
  + Ensure the path to the FXML file is correct in the controller.
  + Verify the FXML file exists in the src/main/resources directory.
* **Database Connection Issues**:
  + Check the database URL, username, and password in the application properties file.
  + Ensure MySQL is running and the librarydb database exists.

**Future Enhancements**

* **User Authentication**: Implement login functionality for different user roles (admin, librarian, member).
* **Enhanced Search**: Add advanced search capabilities for books and authors.
* **Reporting**: Generate reports for books borrowed, overdue books, etc.
* **UI Improvements**: Enhance the user interface for better usability.
* **Implement other functionalities on MainDashBoard** such as Manage Members, Manage Reservation, Manage Fine Payments, Manage Loans and Fines
* **Add images to the books**