

DISCOREAD – FINAL PROJECT DOCUMENTATION

Name: Oumar Ouonogo

Neptun Code: JV3LWP

Application Name: DiscoRead - Digital Library System

Github Repository: [OInfinity/DiscoRead: JavaFX Digital Library App with SQLite + Google Books API](#)

1. Introduction

DiscoRead is a digital library application developed in Java, using JavaFX for the UI and SQLite for local database persistence.

The application allows users to add, search, edit and delete books while also integrating the Google Books API for online book lookup.

Users can attach PDF files, open them externally, and view book details interactively.

The system was created to help organize reading materials, improve accessibility, and provide a modern digital bookshelf experience.

2. System Modules & Libraries Used

Component	Version	Purpose
Java (JDK)	17	Core language used for logic, OOP structure, controller classes, model handling.
JavaFX	20+	Graphical User Interface, scenes, buttons, tables, forms.
SQLite + JDBC Driver	Built-in	Local persistent storage database for books + users.
Gson JSON Parser	Google Library	Converts Google Books API JSON → Java Book objects.
HTTP / URLConnection	Java Standard Networking	Sends API requests + receives online book metadata.
Maven	Latest	Project build, dependency management, packaging.
Git + GitHub	Git 2.52.0	Version control, project backup, collaboration.
JUnit 5	5.10.0	For testing DAO, login, and API logic (if applied).

3. Use-Case Documentation

Every use-case in this document contains two comparison screenshots, one *before* the action and one *after*, followed by an explanation describing the process and outcome.

3.1 User Registration and Login

```
RegisterController.java
UserDAO.java
User.java
register-view.fxml
```



Explanation

The user enters a new Username & Password and submits the form.

Details are stored in SQLite and the new user is allowed to log in immediately.

LoginController.java
UserDAO.java
login-view.fxml



Explanation:

User enters credentials → system validates through database → access granted.

Authentication ensures controlled access and data privacy.

3.3 Add a Book Manually

AddBookController.java
BookDAO.java
Book.java
add-book-view.fxml

The screenshot shows the 'Add New Book' form in the DiscoRead Library application. The form has a blue header bar with the title 'Add New Book'. Below the header, there are several input fields for book information: a large empty text box, Author Name, Publication Year, Genre / Category, ISBN, Library Location (Shelf/Rack), Short Description, Cover Image URL, and Google Books URL (Optional). At the bottom of the form, there are three buttons: 'No PDF selected', 'Choose PDF', and 'Search Google Books Online'. An orange arrow points to the 'Choose PDF' button.

[illegible]

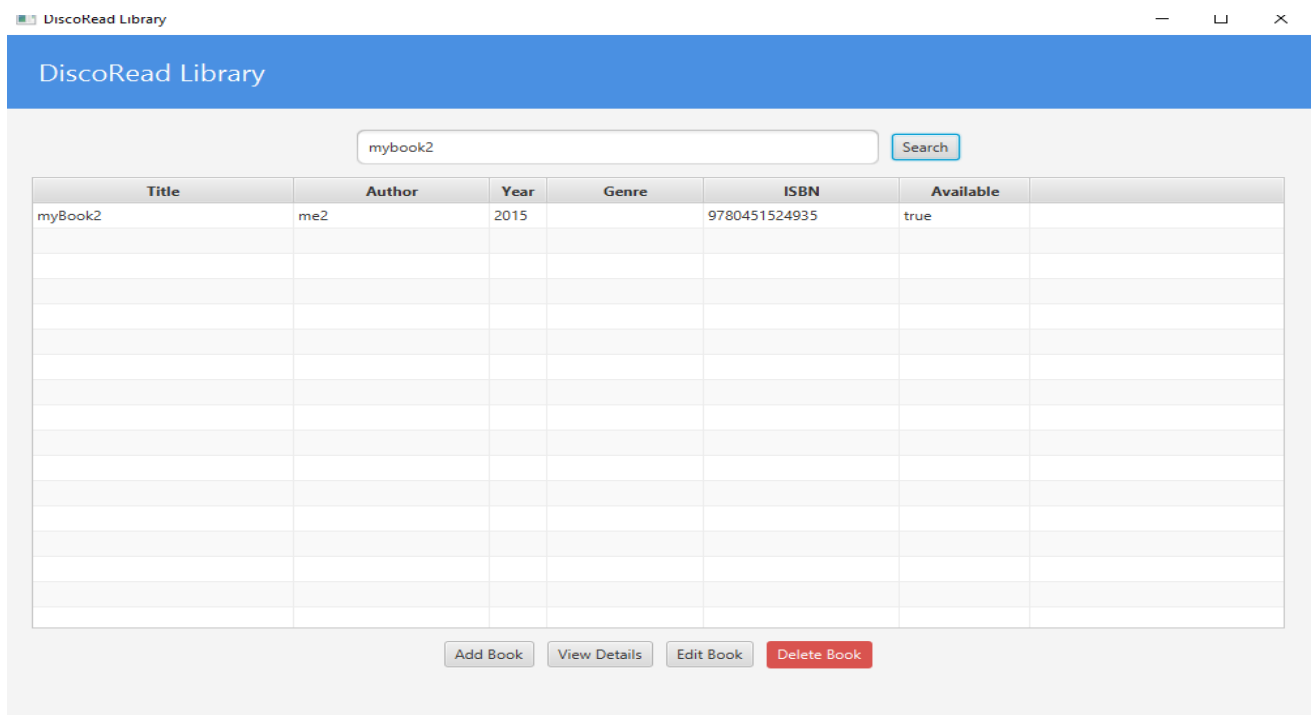
Explanation:

The user enters book details manually including title, author and genre. On save, the book is inserted into SQLite and displayed in the main list.

3.4 Search for a Book

MainController.java
BookDAO.java
main-view.fxml

[illegible]



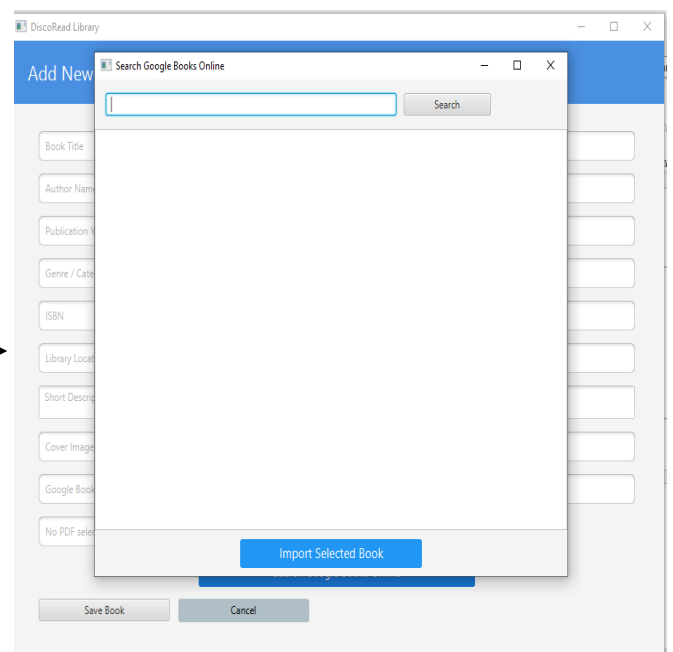
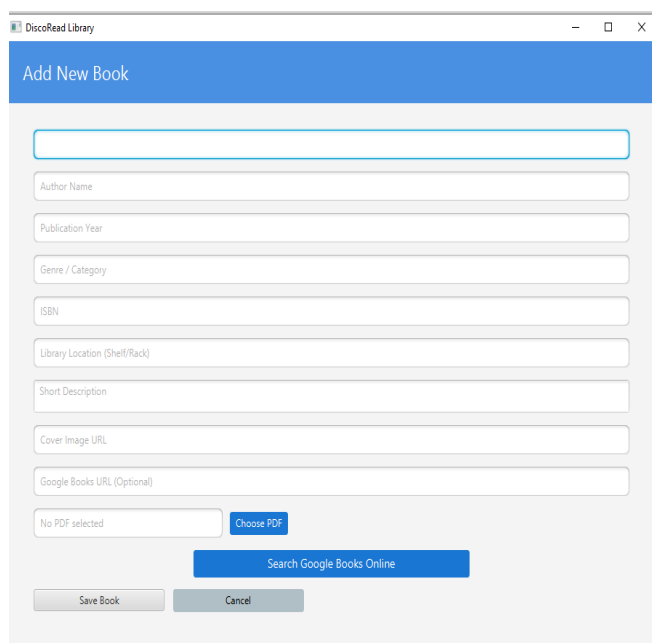
Explanation:

Typing in the search bar filters results instantly by Title

Search uses dynamic filtering for fast and responsive results.

3.5 Google Books Online Import

GoogleSearchController.java
GoogleBooksService.java
google-search-view.fxml



Explanation:

User searches online, selects a book, and details auto-fill instantly.

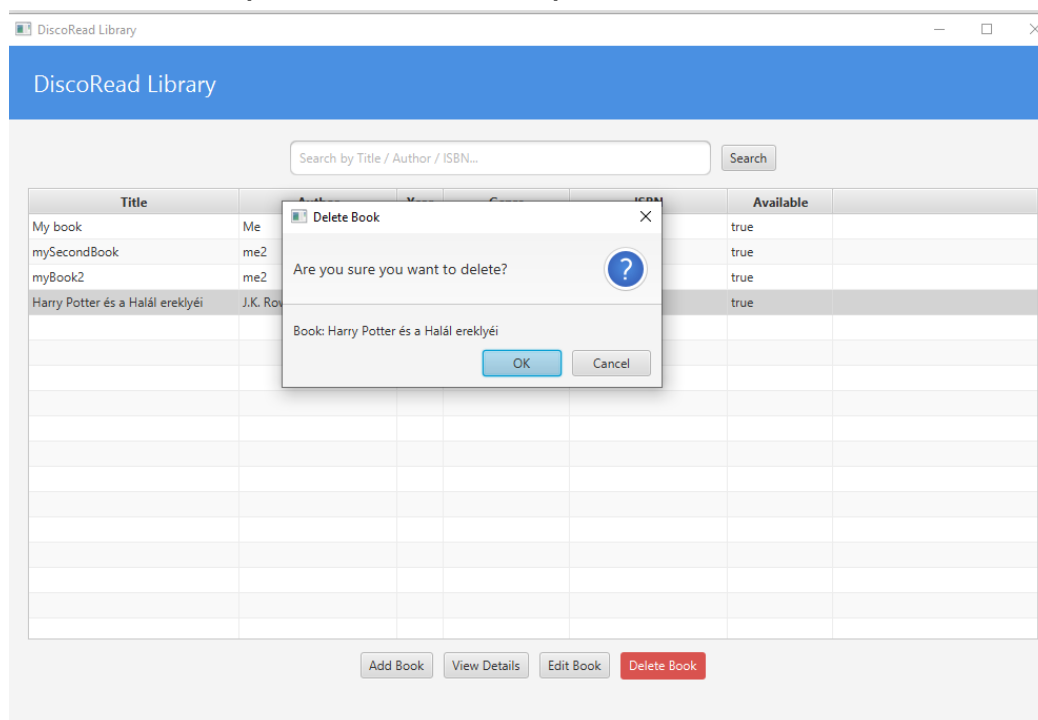
Reduces manual entry time and increases accuracy.

Explanation:

The user opens details → updates fields → saves changes.

SQLite updates and UI refreshes immediately.

3.7 Delete Book (ID-based safe delete)

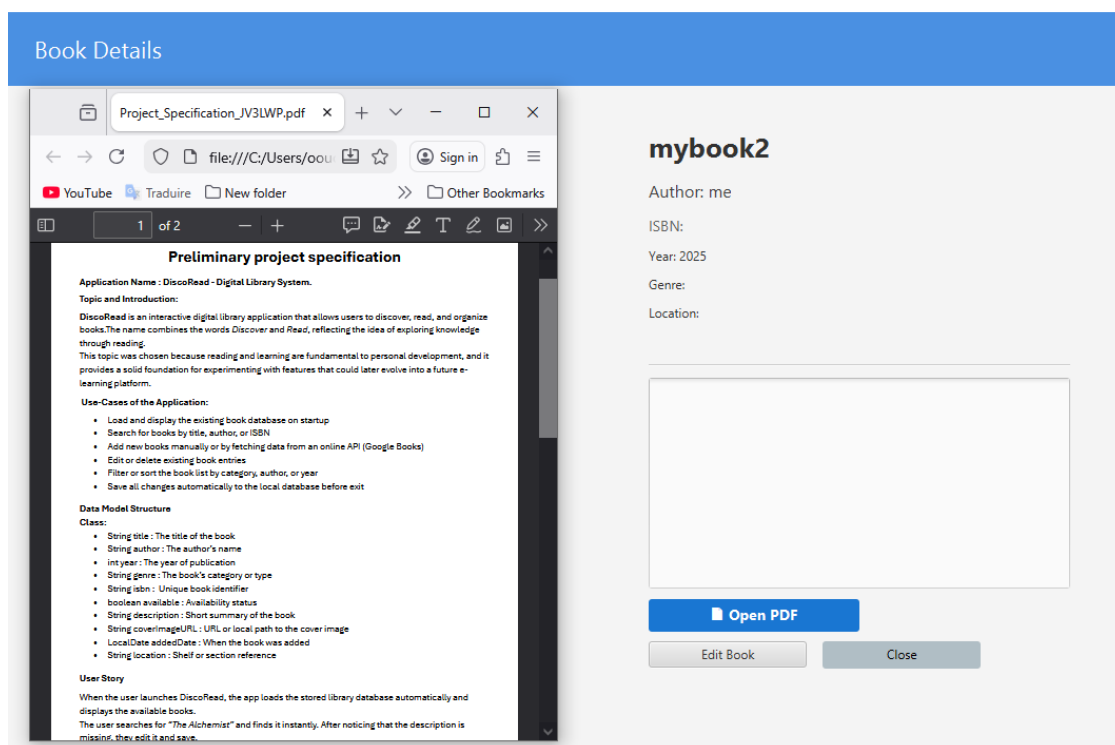


MainController.java
BookDAO.java

Explanation:

Selected book is deleted permanently from SQLite using unique ID.

3.8 PDF Upload & Open

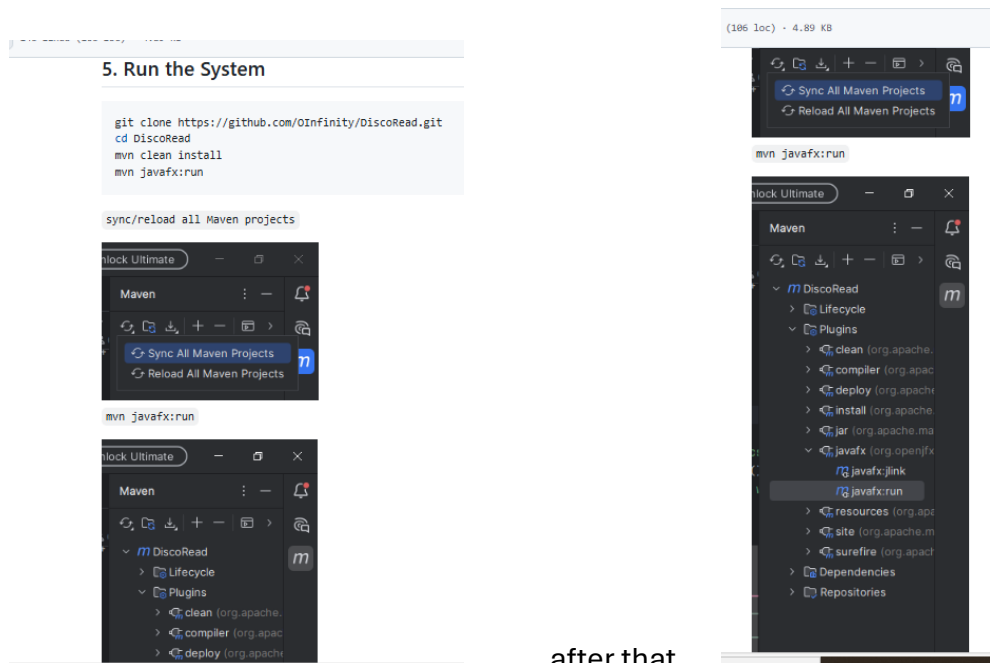


PDFImportService.java
BookDetailsController.java
book-details-view.fxml

Explanation:

User selects a PDF → stored locally → one-click access to open.

Allows reading resources directly from the system.



after that

4. Project Summary

DiscoRead successfully implements a digital library with user authentication, online book importing, local storage and PDF support.

It allowed practical application of JavaFX GUI design, database integration and API communication. The system is efficient, scalable and open for future upgrades including cloud-sync and in-app PDF viewing.

Overall, the project achieved its purpose and demonstrates a full working software solution.

5. Course Summary

Through this course, I strengthened my skills in Java, MVC structure, UI development and database persistence.

I also learned API integration, JSON processing, version control and debugging with real GUI workflow. The project improved my ability to design complete applications from backend to interface. It was both challenging and enjoyable, and prepared me for larger real-world development.