Yaşar University

Spring, 2024 - 2025

SE 2232 - Software System Analysis

Final Project Report: Software Requirements Specifications Document (SRS)

Student Name:	**Arda Gürkan**
Student No:	**23070006096**
Department Name:	**Software Engineering**
Course Section No:	**Section-2**

This template is prepared based on the IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998).

Table of Contents (Do not change the Section Names!)

1 Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the requirements and functionalities of the desktop application MyLibrary. This application is developed for users who want to manage the books they have read or wish to read.

Scope

The software product to be developed is named **MyLibrary**.

This application is a Java-based desktop system that allows users to manage their personal book collections. It provides functionalities such as adding, deleting, viewing, editing, and rating books. Users can also track reading statuses, search for authors, view favorite books and authors, and get notified about upcoming releases.

The main goal of the application is to help users organize and keep track of their reading habits effectively, while also applying software system analysis principles and improving the developer's practical skills in Java, database integration, and GUI design.

1.2 Definitions, acronyms, and abbreviations

GUI Graphical User Interface the visual interface used by the user to interact with the system
IDE Integrated Development Environment the tool used to write and run the code in this project NetBeans is used
MyLibrary the name of the desktop application developed in this project
MySQL the relational database management system used to store and manage application data
JFrame a Java Swing component used to create application windows
User Type 1 admin user who has access to all functionalities in the application
User Type 2 regular user who has limited access to some functions
BookId unique identifier for each book in the database
AuthorId unique identifier for each author in the database
Read Status integer value that shows the reading status of a book 1 read 2 not read 3 wish to read
SRS Software Requirements Specification the document that describes the requirements and design of the system

1.3 References

SE 2232 Software System Analysis course slides
Final Project Specification Document provided by Asst Prof Dr Deniz Özsoyeller
Java Swing official documentation by Oracle
MySQL official documentation

NetBeans IDE official user guide
Visual Paradigm official modeling tool documentation

1.4 Overview

This subsection explains the structure and contents of the Software Requirements Specification document
The document begins with an introduction and continues with the overall system description and constraints
It then presents detailed functional and nonfunctional requirements of the system
Use case analysis class diagrams sequence diagrams and data flow diagrams are included to support the design process
Finally the document includes GUI design details and system attributes to provide a complete understanding of how the application will function and be developed

2 Design and Implementation Constraints

The application must be developed using Java programming language
The graphical user interface must be built using Java Swing components
NetBeans IDE is used as the development environment
MySQL is used as the relational database management system
The system must use JDBC for database connectivity
UML diagrams must be created using Visual Paradigm Community Edition
The application must run on desktop computers with Java Runtime Environment installed
There is no integration with external applications or online APIs required

3 Specific Requirements

3.1 Functional Requirements

3.1.1 Function 1 – Add Book

- The system shall allow the user to add a new book by entering all book attributes using text fields in the GUI.
- The system shall retrieve the authorId from the authors table if the author already exists, or insert a new author if not.
- The system shall use the retrieved or newly generated authorId when inserting the book into the books table.
- The system shall automatically assign and increment the bookId for each new book added.

3.1.2 Function 2 – Delete Book

- The system shall allow the user to delete a book by entering its bookId.

- The system shall remove the corresponding row from the books table.
- If the author of the deleted book has no remaining books in the database, the system shall also delete that author from the authors table.
- The system shall display a confirmation message when deletion is successful.

3.1.3 Function 3 – Display Book Information

- The system shall allow the user to display all details of a specific book by entering its bookId.
- The system shall retrieve the book's information from the books table and display it in the GUI.

3.1.4 Function 4 – Search Author

- The system shall allow the user to search for an author by entering the author's name and surname.
- The system shall retrieve and display the author's information from the authors table if a match is found.

3.1.5 Function 5 – Update Book Information

- The system shall allow the user to update an existing book's details.
- After retrieving the book information by bookId, the user shall be able to modify and save the changes.
- The system shall update the books table with the new values.

3.1.6 Function 6 – Display Favorite Books

- The system shall display all books read by the user that have a rating of 4 or 5.
- The system shall retrieve these records from the books table based on the read and rating values.

3.1.7 Function 7 – Display Favorite Authors

- The system shall identify and display authors who have at least 3 books in the user's library.
- The system shall count the number of books per author and determine if the threshold is met.

3.1.8 Function 8 – Display Unread Books

- The system shall display all books marked as not read (read = 2) by the user.
- The system shall retrieve and show all relevant details of these books.

3.1.9 Function 9 – Notify Upcoming Wishlist Books

- The system shall, upon application startup, notify the user of books that are on the wishlist

  and have a releaseDate within the next 7 days.

3.1.10 Function 10 – Display Book Cover Image

- The system shall display the image of a book's cover based on the file path stored in the cover attribute.
- The system shall retrieve the path using the bookId and load the corresponding image.

3.2 Performance Requirements

The system shall support up to 2 simultaneous users operating the application on different computers without performance loss
The application shall efficiently handle up to 1000 book records and 500 author entries stored in the MySQL database
The system shall respond to all user interactions and database operations such as add update delete or view within 1 second under normal usage conditions

3.3 Software System Attributes

Reliability the system shall perform all core functions without failure for at least 2 continuous hours of use

Availability the system shall be available for use on any computer that has Java and MySQL properly installed

Security the system shall require a valid username and password for login to prevent unauthorized access

Maintainability the codebase shall be modular and well documented to allow future improvements or debugging with minimal effort

Usability the user interface shall be designed to be simple and intuitive so that a new user can perform basic actions without training

3.4 Use Case Analysis

3.4.1 Actors

Admin - A user who has full access to the application.

User - A limited-access user who can only perform a subset of the operations.

3.4.2 Scenarios

Scenario 1 View Unread Books
The user logs into the system and selects the option to view unread books
The system filters books where the read status is 2 and displays the list with all details

Scenario 2 Search Author
The user selects the author search function from the menu and enters the author's name
The system retrieves and displays the author's full name surname and website if a match is found

3.4.3 Use Case Forms

3.4.3.1 Complete Use Case Form 1

Use Case Name Add Book
Participating Actors Admin
Description Admin adds a new book and its author if the author does not exist
Trigger Admin clicks on the Add Book button
Preconditions Admin must be logged into the system
Normal Course
1 Admin opens the Add Book form
2 Enters book and author details
3 System checks if author exists
4 If not adds new author
5 Adds the book using correct authorId
6 Displays success message
Postconditions The book and if needed the author are saved in the database
Exceptions Required fields are empty or database connection fails

3.4.3.2 Complete Use Case Form 2

Use Case Name Delete Book
Participating Actors Admin
Description Admin deletes a book using its bookId
Trigger Admin clicks on the Delete Book button
Preconditions Admin must be logged into the system
Normal Course
1 Admin enters the bookId to delete
2 System deletes the book from the database
3 Checks if the corresponding author has other books
4 If not deletes the author
5 Displays confirmation message
Postconditions Book and possibly author are removed from the database
Exceptions BookId not found or database error

3.4.3.3 Complete Use Case Form 3

Use Case Name View Favorite Books
Participating Actors User
Description User views a list of books that are read and rated 4 or 5
Trigger User selects the View Favorite Books option
Preconditions User must be logged in
Normal Course
1 User selects the favorite books option
2 System filters books with read status 1 and rating 4 or 5
3 Displays list of matching books
Postconditions Favorite book list is shown to the user
Exceptions No favorite books found or database error

3.4.3.4 Basic Use Case Forms

Use Case Name View Book
Participating Actors Admin User
Description User views the details of a book using its ID
Trigger User clicks on View Book
Preconditions User must be logged in

Use Case Name Search Author
Participating Actors Admin User
Description User searches for an author by name
Trigger User clicks on Search Author
Preconditions User must be logged in

Use Case Name Update Book
Participating Actors Admin
Description Admin updates book details after viewing it
Trigger Admin clicks on Update Book
Preconditions Admin must be logged in

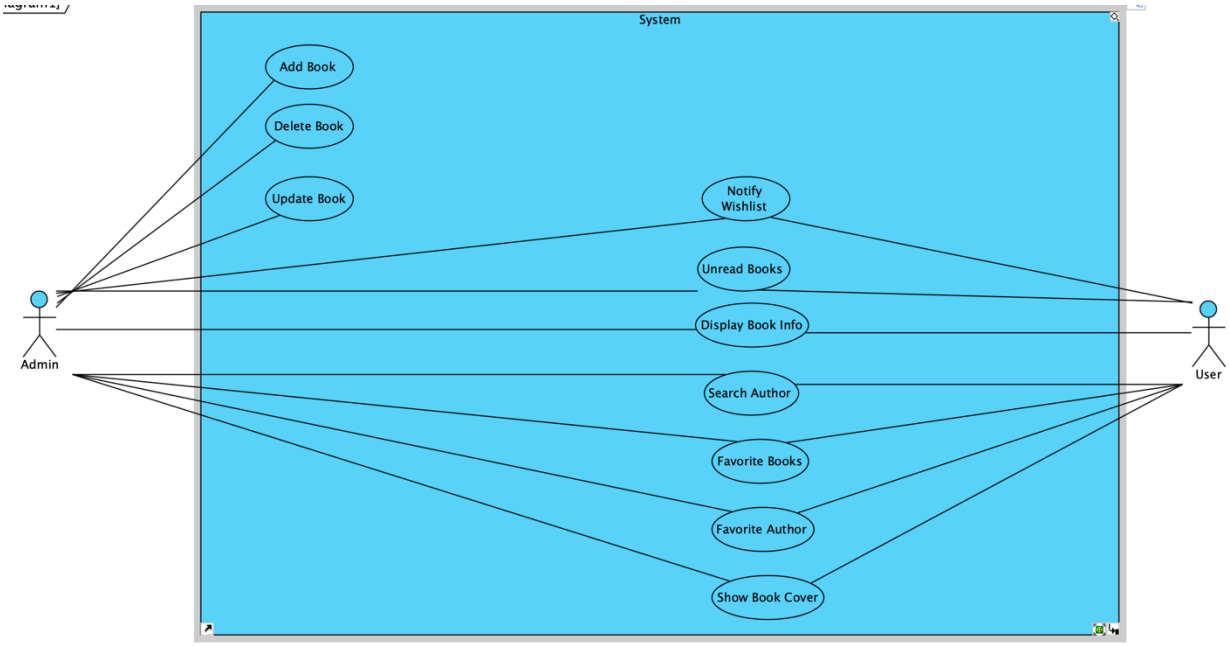
Use Case Name Show Cover Image
Participating Actors Admin User
Description User views the image of a book cover by entering the book ID
Trigger User clicks on Show Cover
Preconditions User must be logged in

Use Case Name Notify Upcoming Releases
Participating Actors Admin User
Description System shows upcoming books from the wish list with a release date within 7 days
Trigger Application startup
Preconditions User must be logged in

3.4.4 Relationships among Actors and Use Cases

Admin actor is connected to all use cases in the system including adding updating deleting and viewing books as well as managing authors and accessing user specific features
User actor is connected only to limited use cases including viewing books searching authors checking favorite books and authors viewing unread or wish to read books seeing book covers and receiving upcoming release notifications
Both Admin and User actors share access to common viewing operations such as View Book Search Author and Show Cover Image
Only Admin has access to sensitive data modification operations like Add Book Delete Book and Update Book
The use case diagram also reflects these differences by connecting each actor only to the relevant use cases they are allowed to access

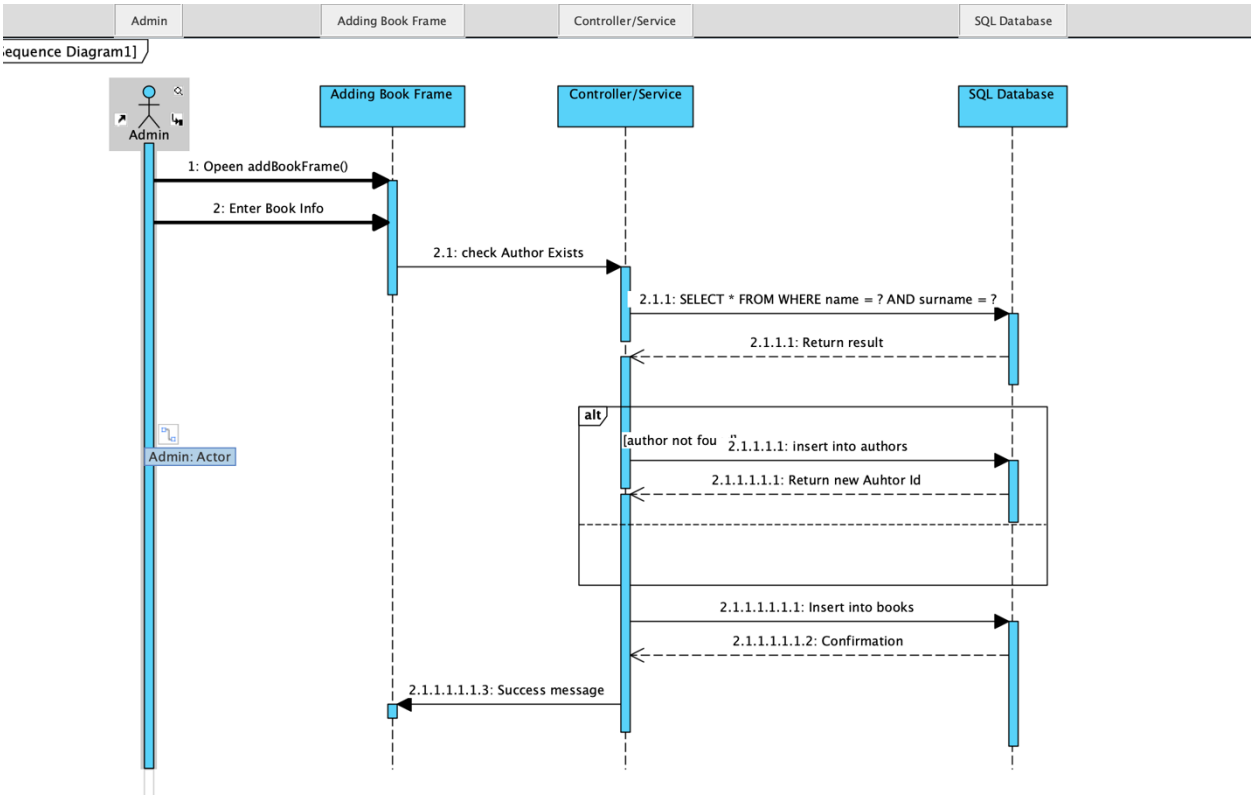
3.4.5 Use Case Diagram



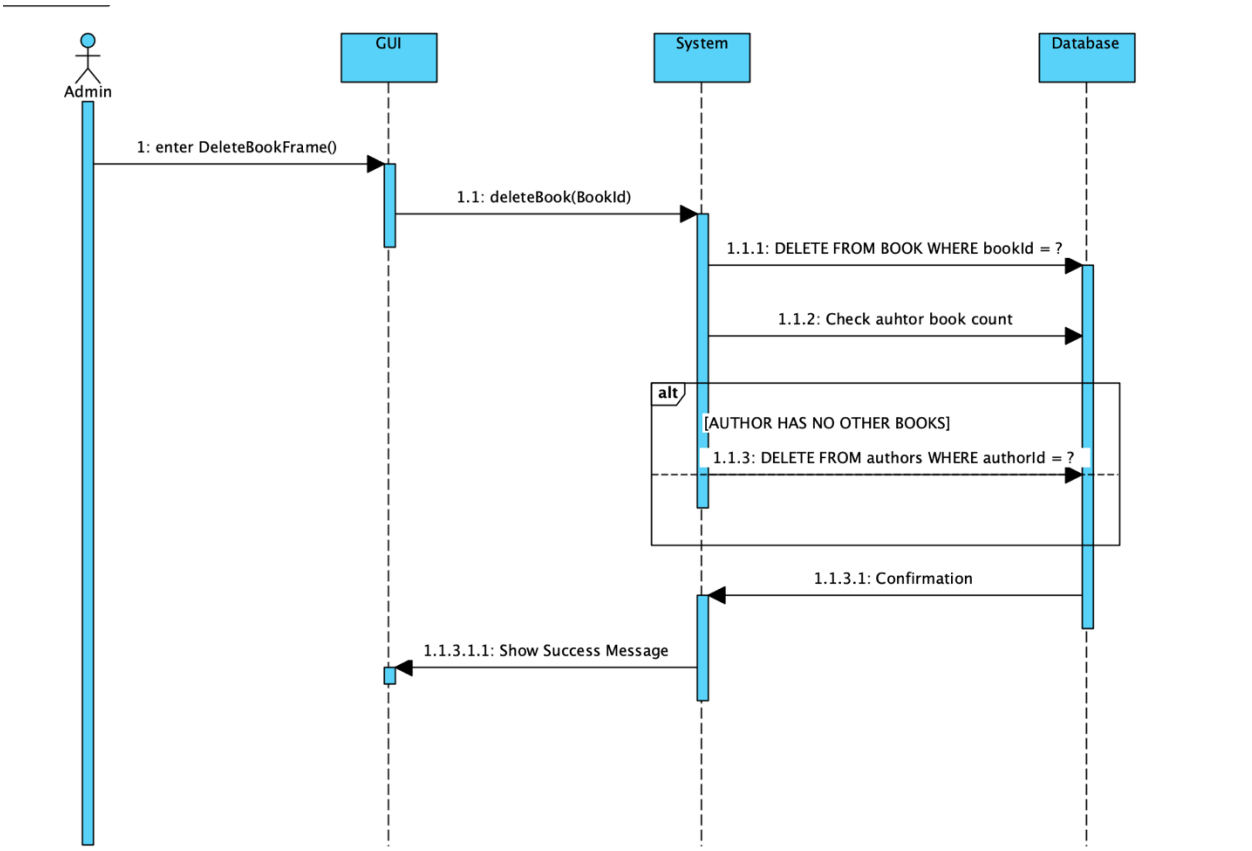
The use case diagram above shows the interactions between two actors, **Admin** and **User**, and the system. The **Admin**actor has access to all functionalities, including adding, deleting, and updating books, as well as all viewing and searching operations. The **User** actor has limited access and can perform only viewing-related actions such as displaying book information, searching authors, checking unread and favorite books, viewing book covers, and receiving wish list notifications. Shared use cases are connected to both actors, reflecting the roles and permissions defined in the system.

4 Behavioral Models

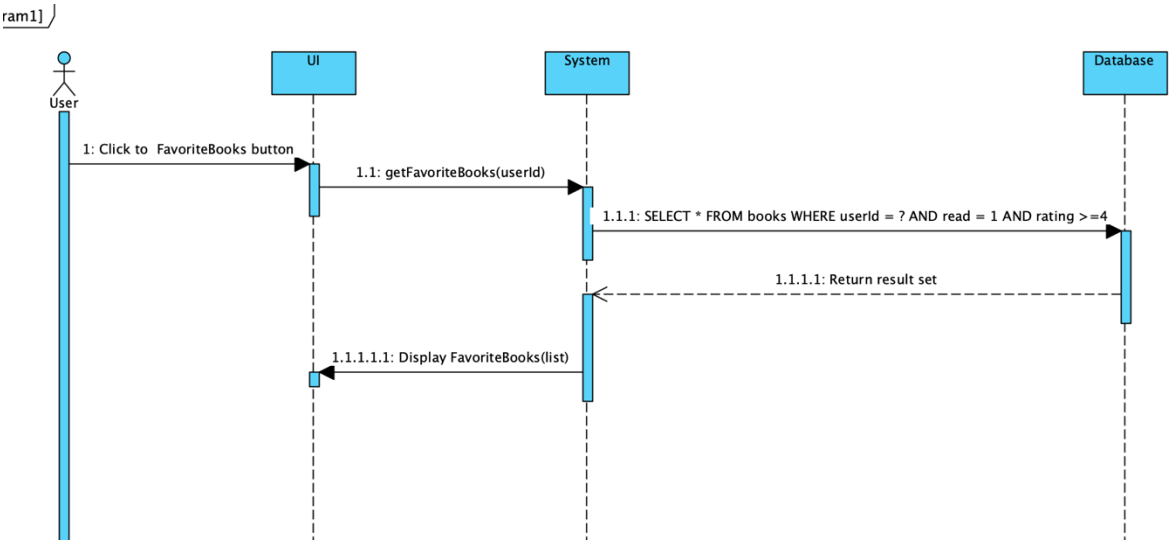
4.1 Sequence Diagram



ADDING A BOOK



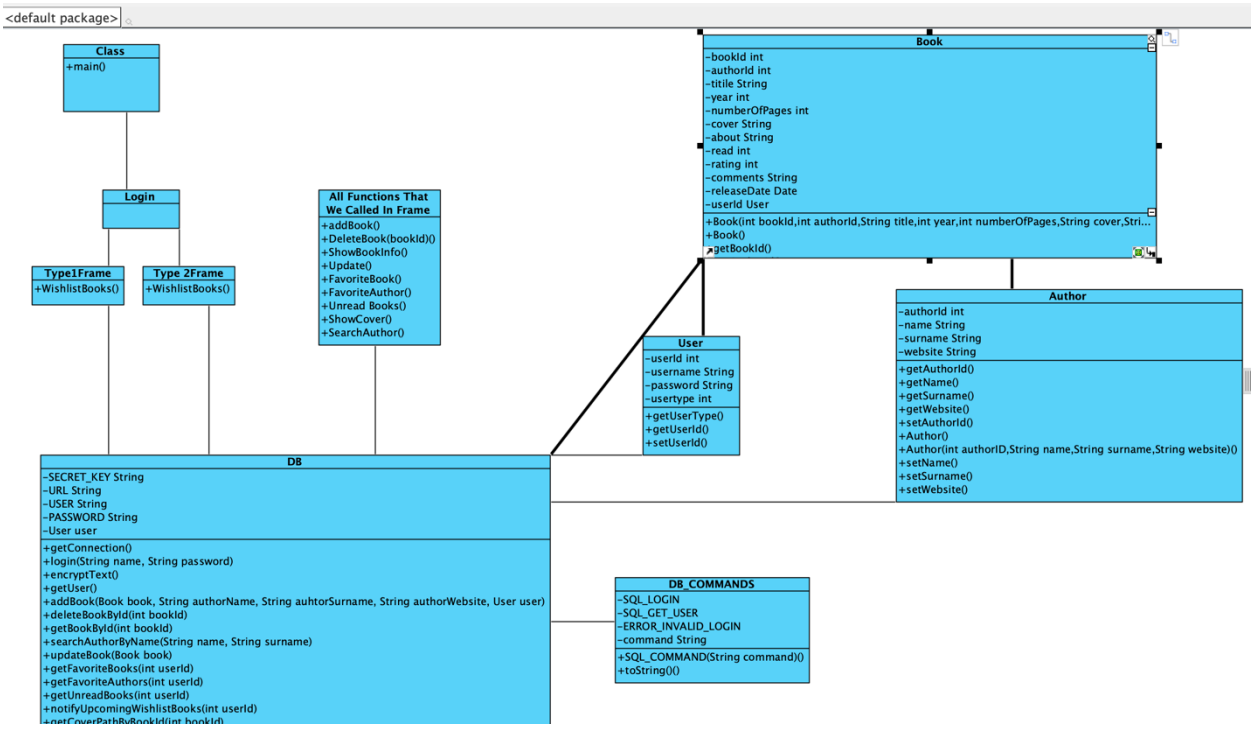
DELETE A BOOK



DISPLAY FAVORITE BOOKS

5 Structural Models

5.1 Class Diagram



This class diagram shows the main structure of the MyLibrary application. It includes the classes for user login, GUI frames, database operations, and data models.

The main() method starts the program and opens the Login screen. After login, the user is directed to either Type1Frameor Type2Frame depending on their user type. Both frames can access the WishlistBooks() function.

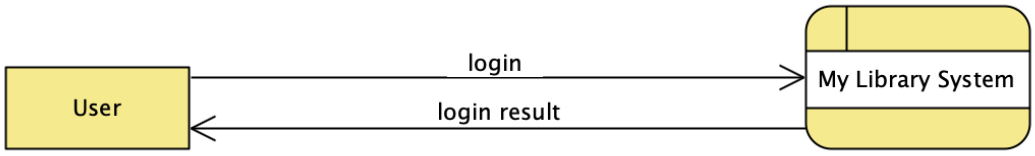
All the main functions (like addBook(), deleteBook(), update(), searchAuthor()) are handled by the DB class. This class connects to the database and runs all the queries. It also keeps the logged-in user as a field.

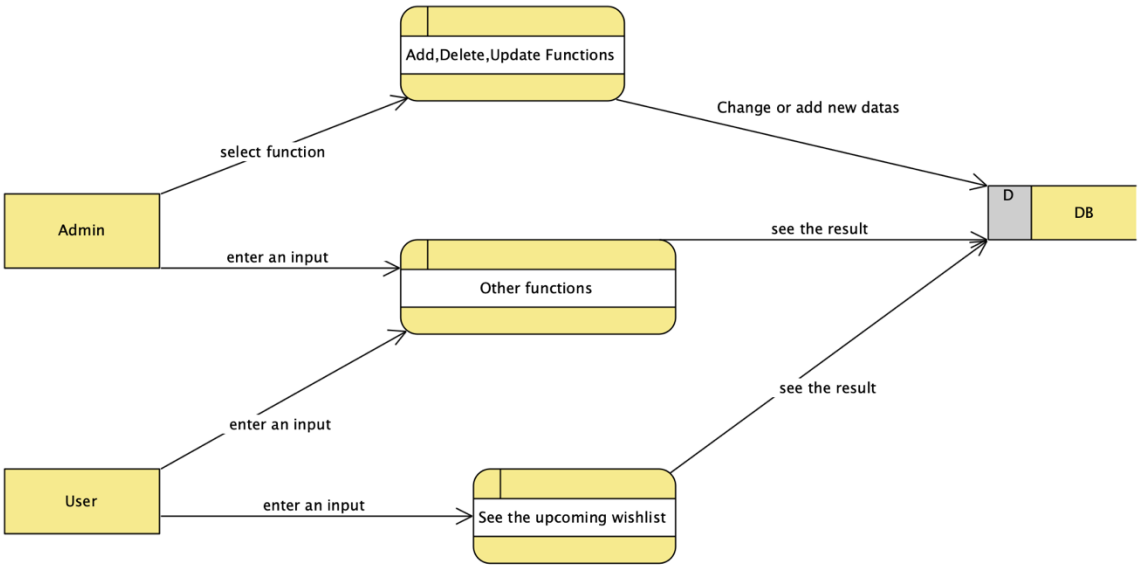
The Book, Author, and User classes are data models. Each book belongs to one user and one author. The DB_COMMANDSclass includes constant SQL commands used by the DB class.

This diagram helps show how the GUI, logic, and data parts of the system are connected and how they work together.

6 Process Modeling

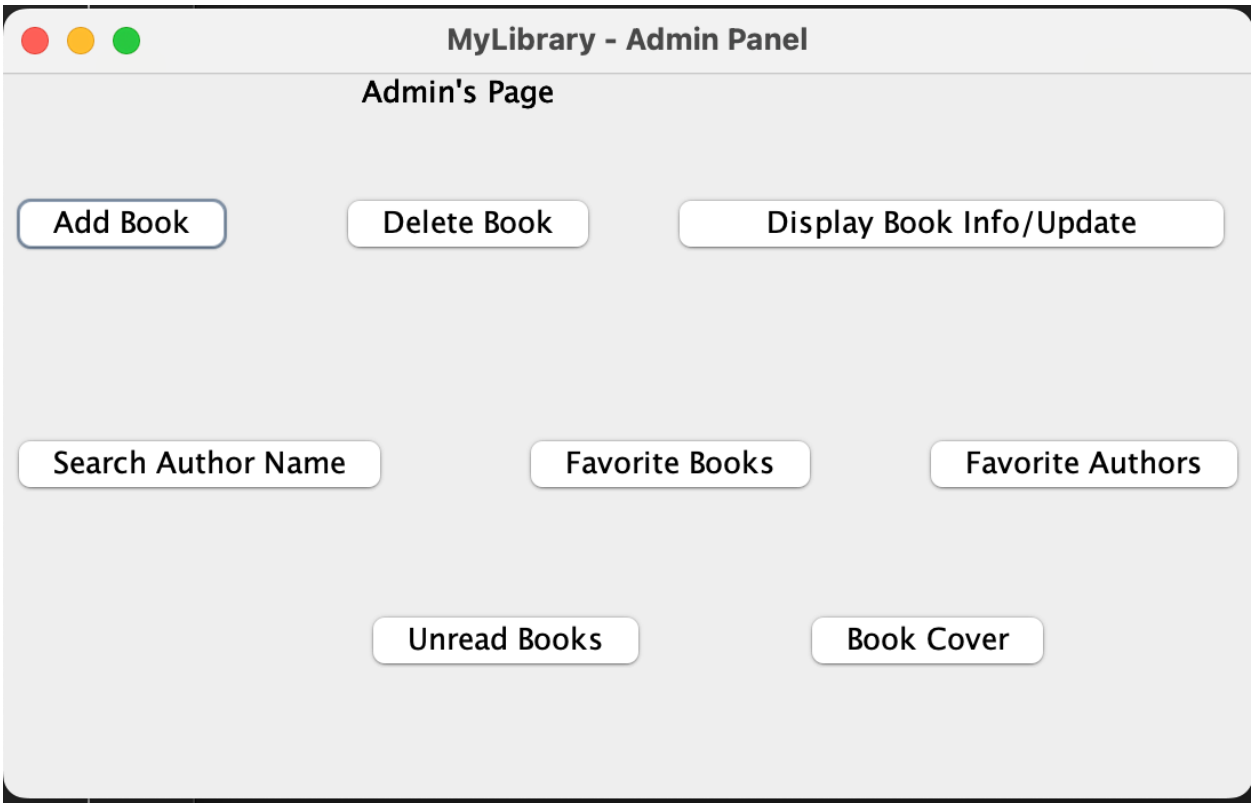
6.1 Data Flow Diagram (DFD)



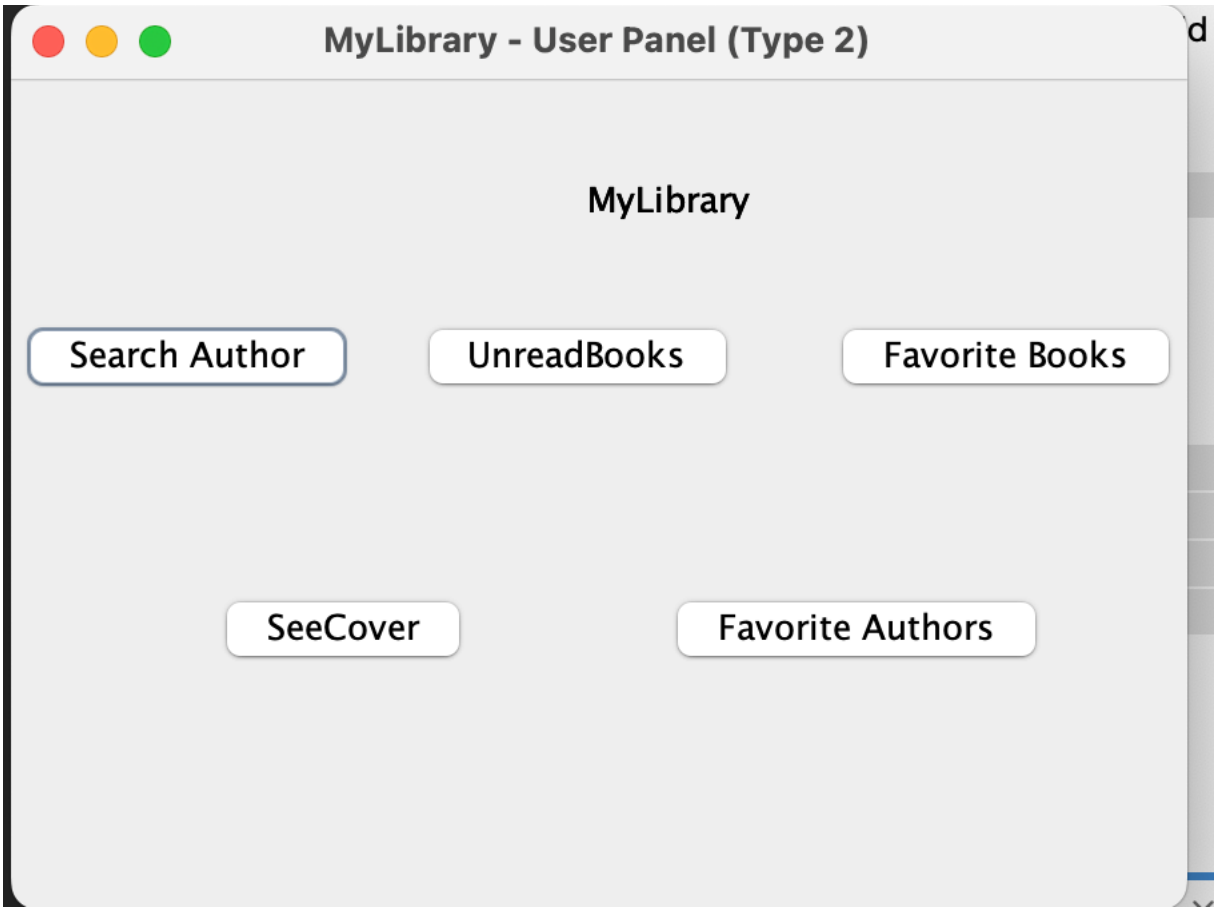


7 Graphical User Interface(s) (GUIs)

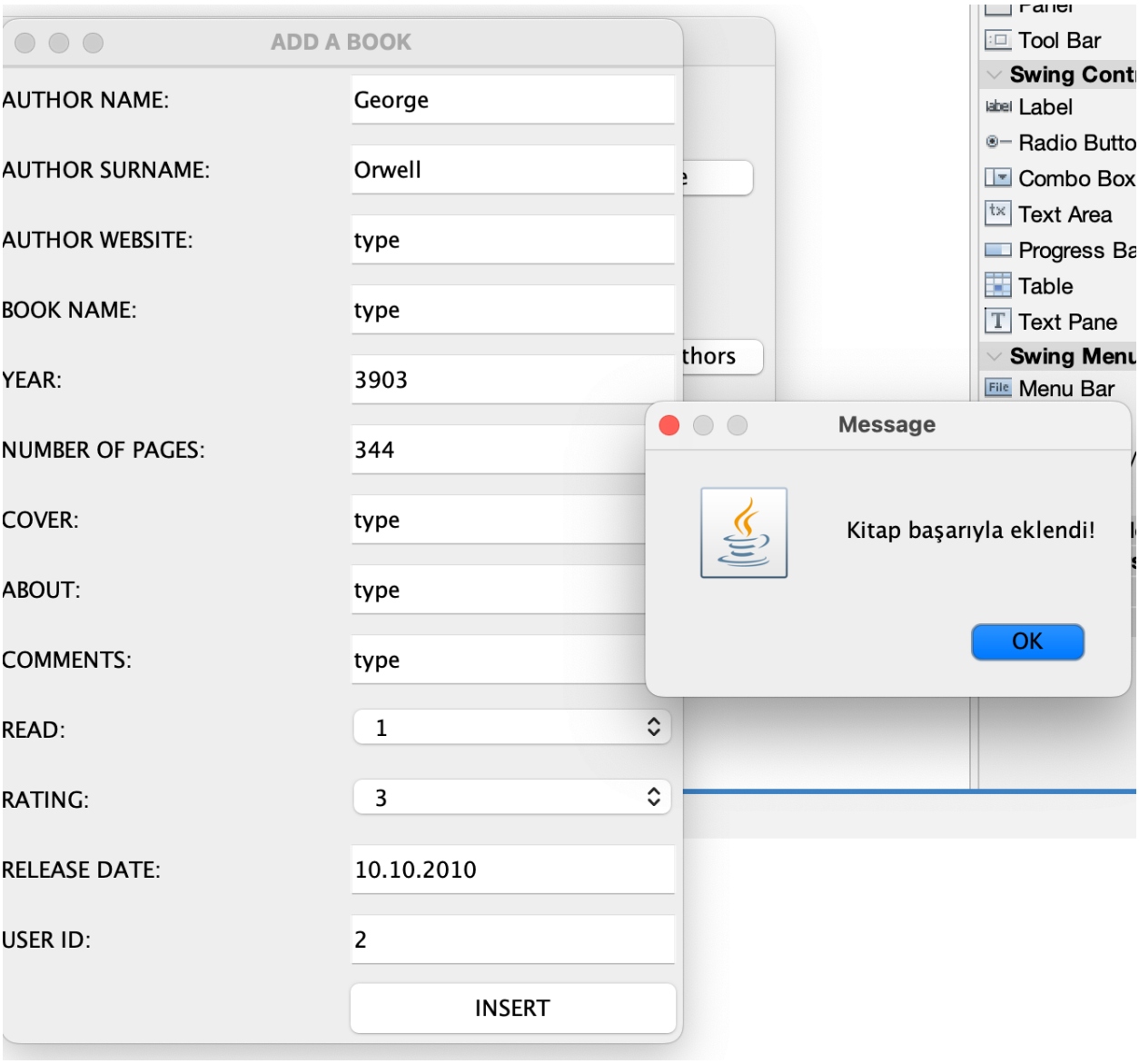
LOGIN FRAME AND ITS ERROR MESSAGE.



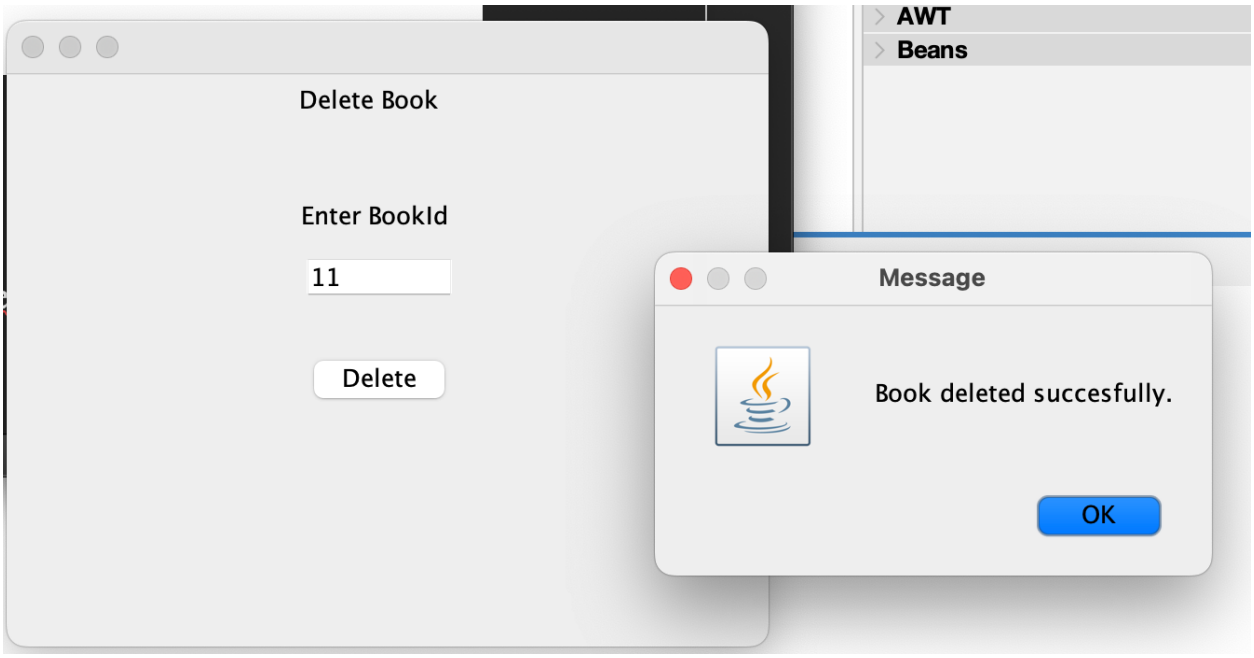
TYPE1 MAINFRAME.



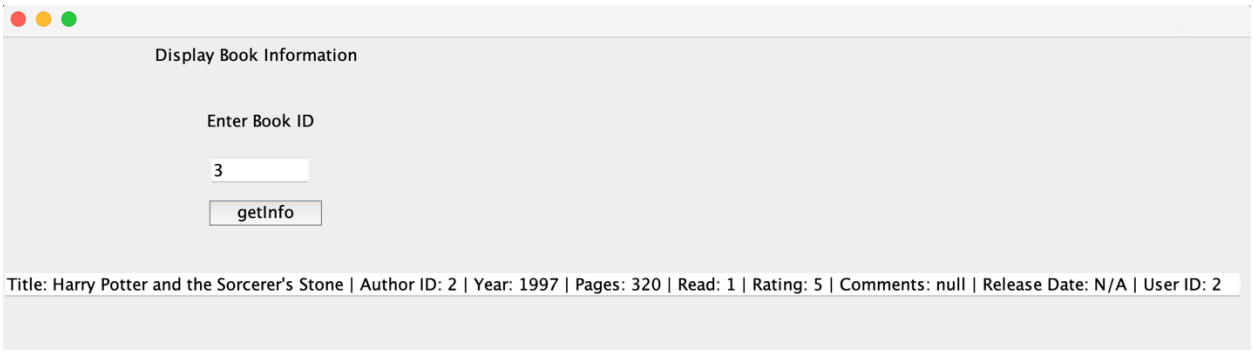
TYPE 2 MAIN FRAME.



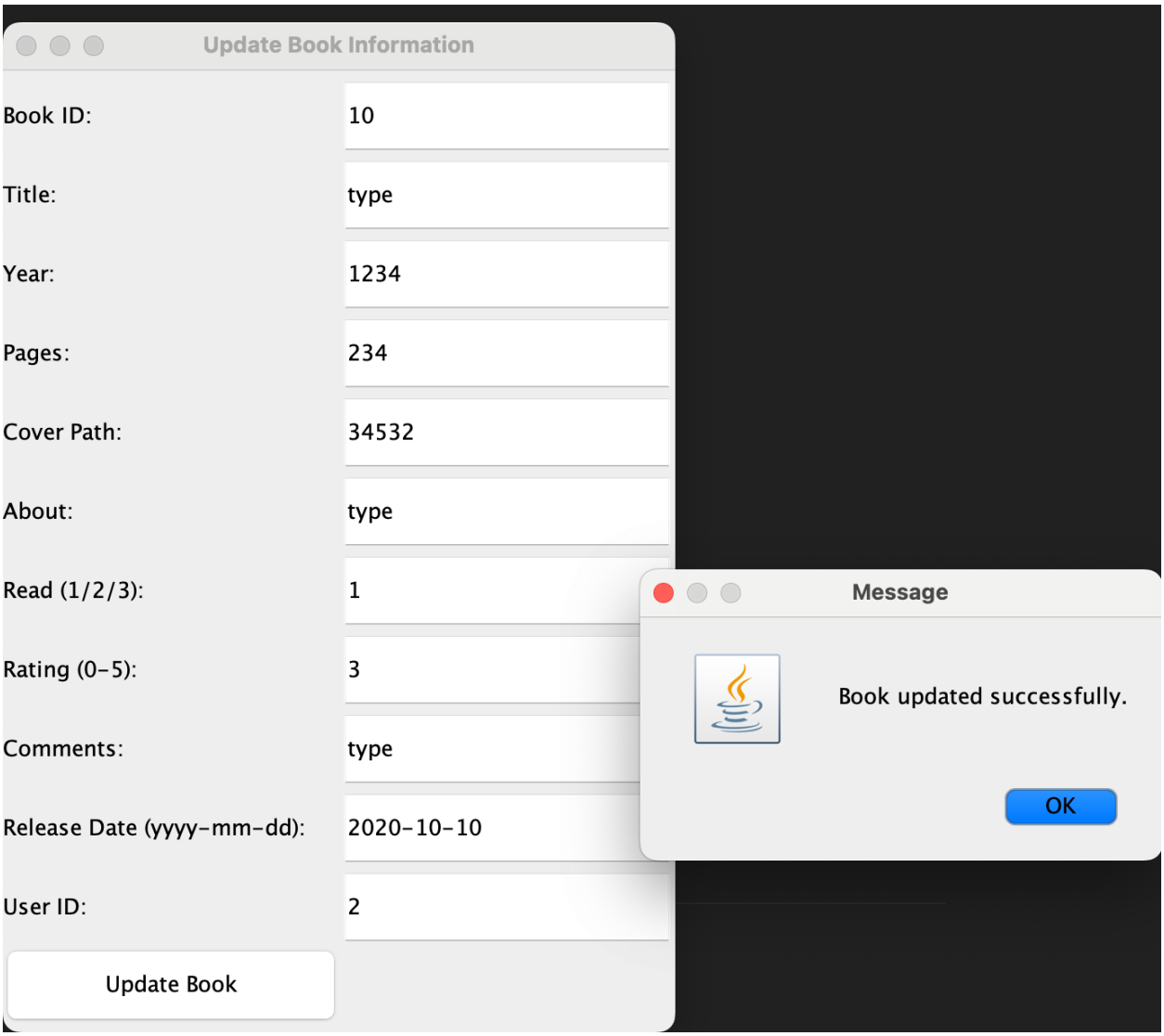
ADD BOOK AND CONFIRMATION.



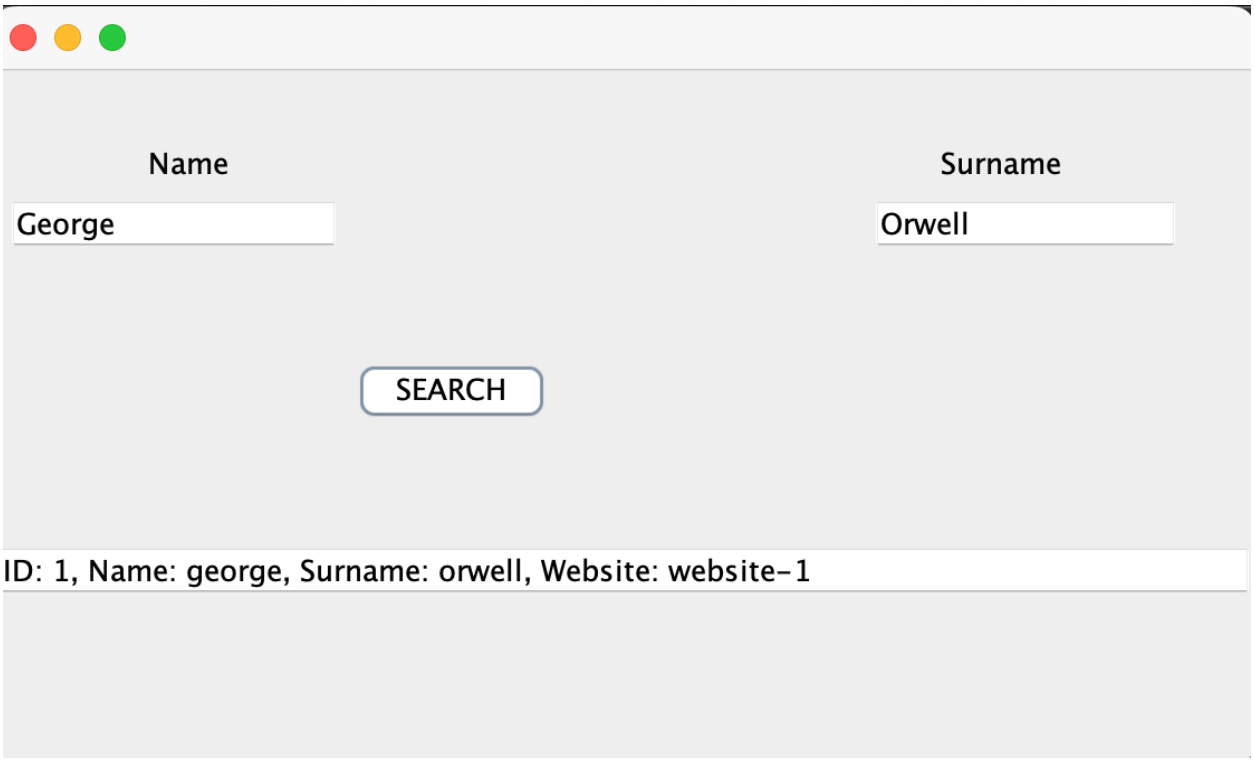
DELETE BOOK AND CONFIRMATION.



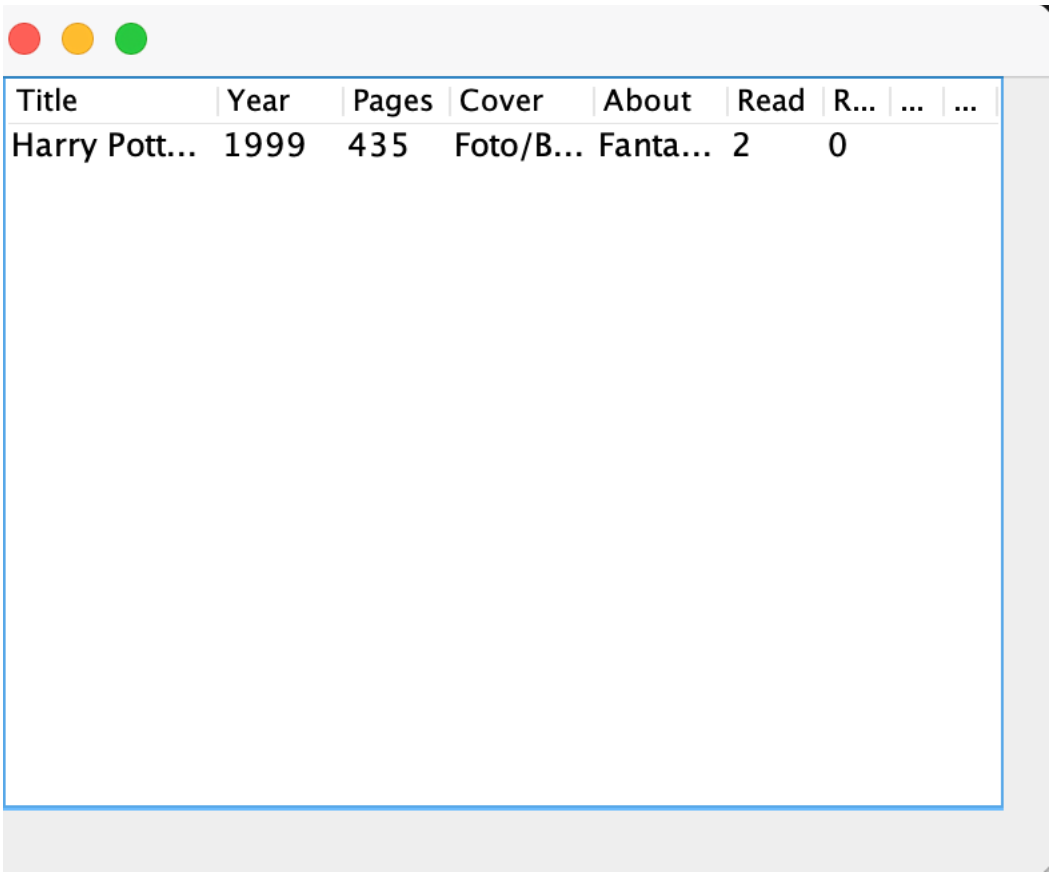
DISPLAY BOOK INFO.



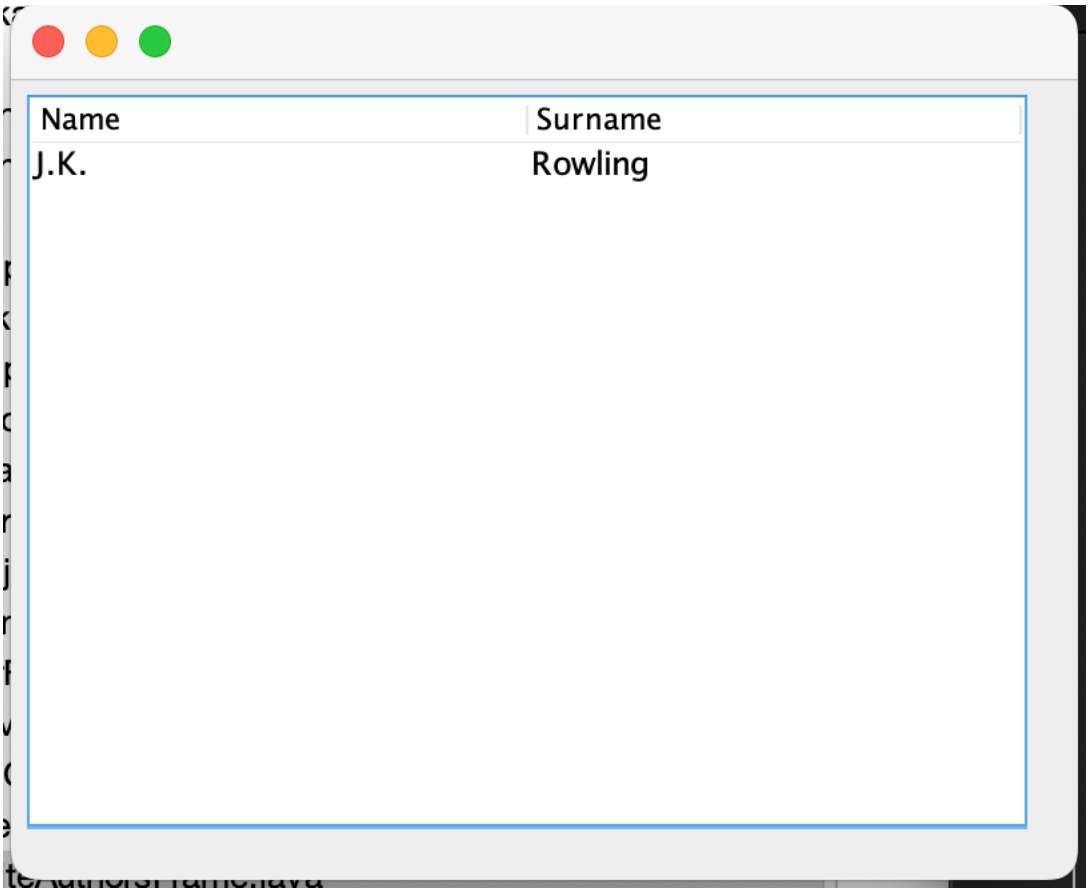
BOOK UPDATE AND CONFIRMATION.



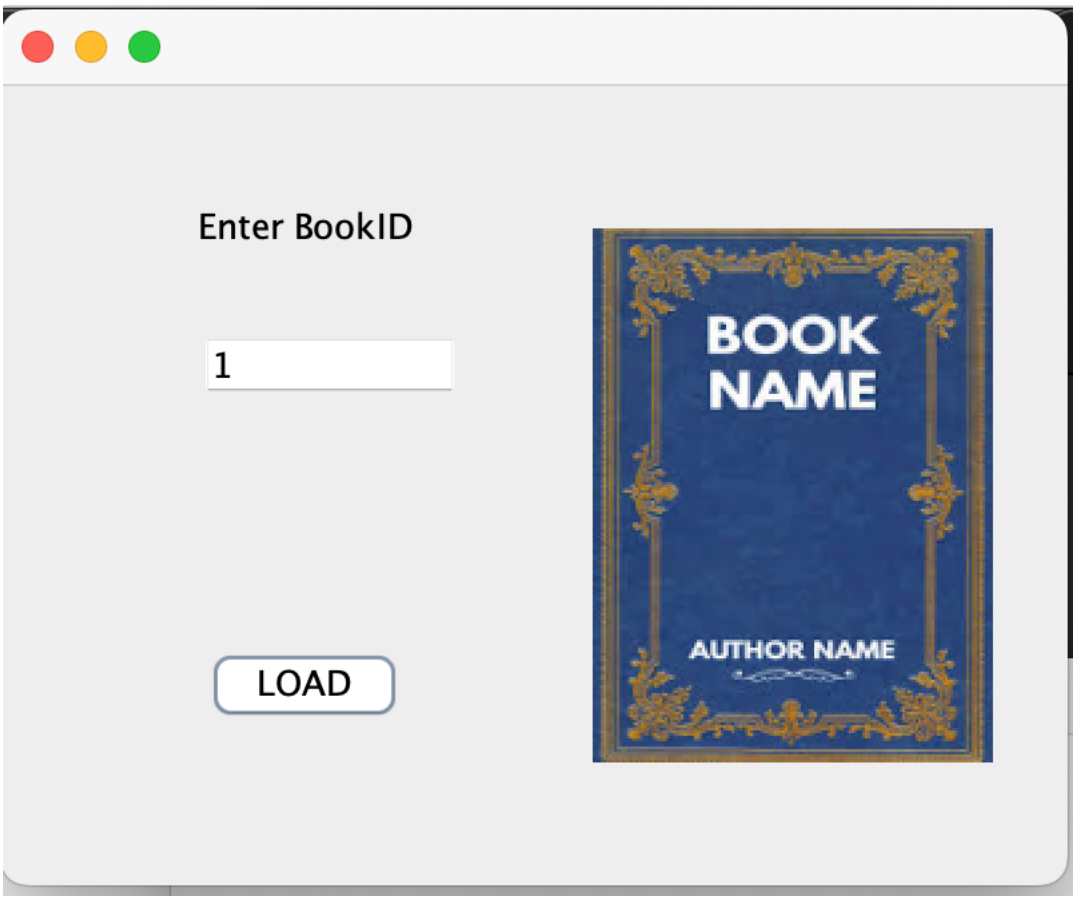
SEARCH AUTHOR.



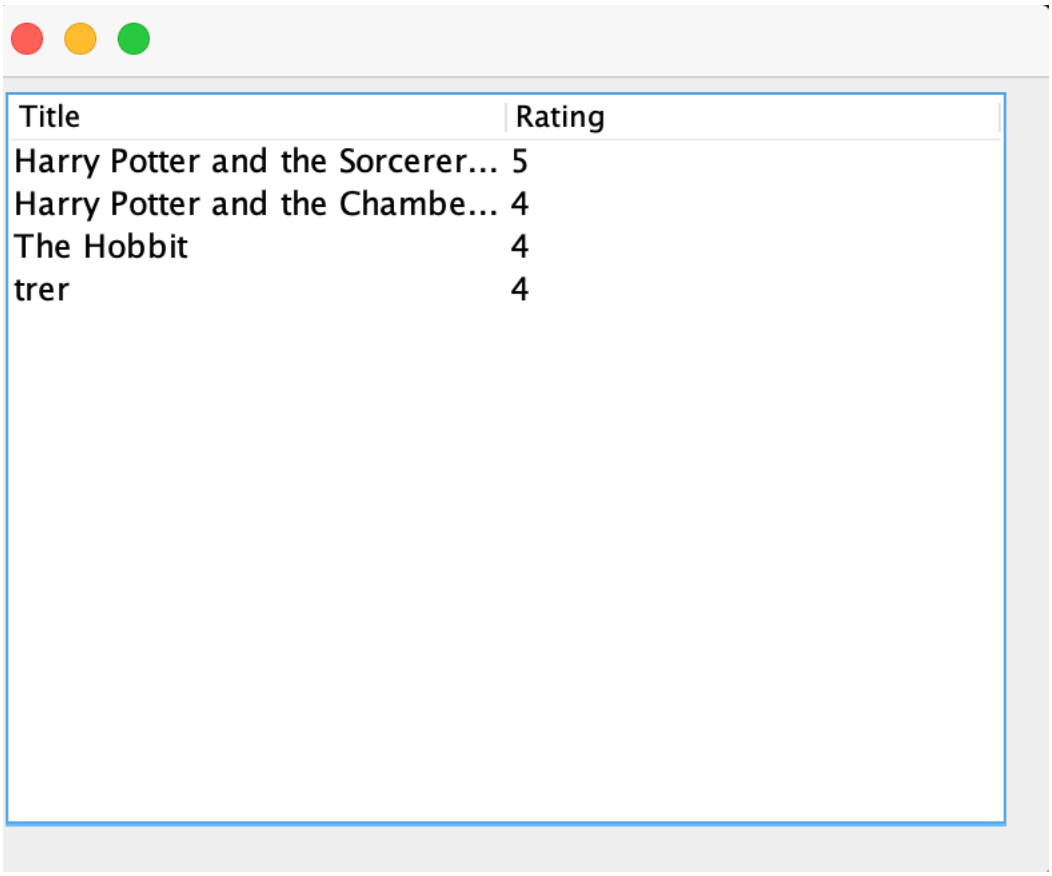
UNREAD BOOKS.



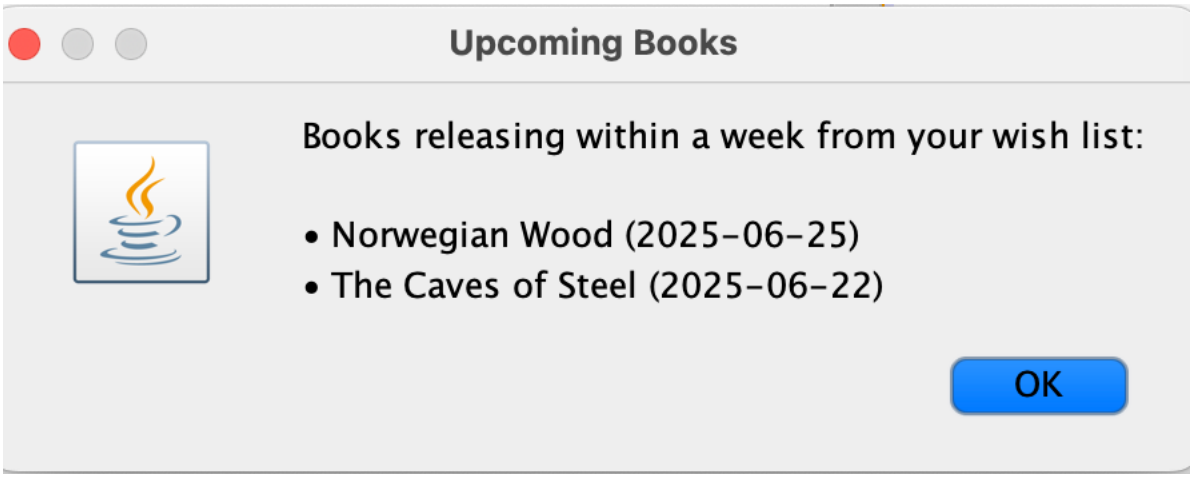
FAVORITE AUTHORS.



BOOK COVER FRAME.



FAVORITE BOOKS.



UPCOMING BOOKS.

8 Conclusion and Future Work

The project helped us improve our coding and database skills. We also learned how to build a user-friendly interface.

In the future, we can add new features like:

User registration,

Book recommendations,

Multi-language support.