

Library Management System

Project Introduction

Library management system is building applications in java. This project focuses on how to create a book rental management system project in java. The system is divided into two roles: administrator and user(borrower). This system could be used in local bookstores. The main feature of this system is that all the books available in the bookstore can be borrowed by users and also the borrowed books by users can be returned in the bookstore.

The main features of the administrator include create and reset databases, add users, view users, issue books, view books and return books. The main features of ordinary users include viewing books and borrowing books.

Technology

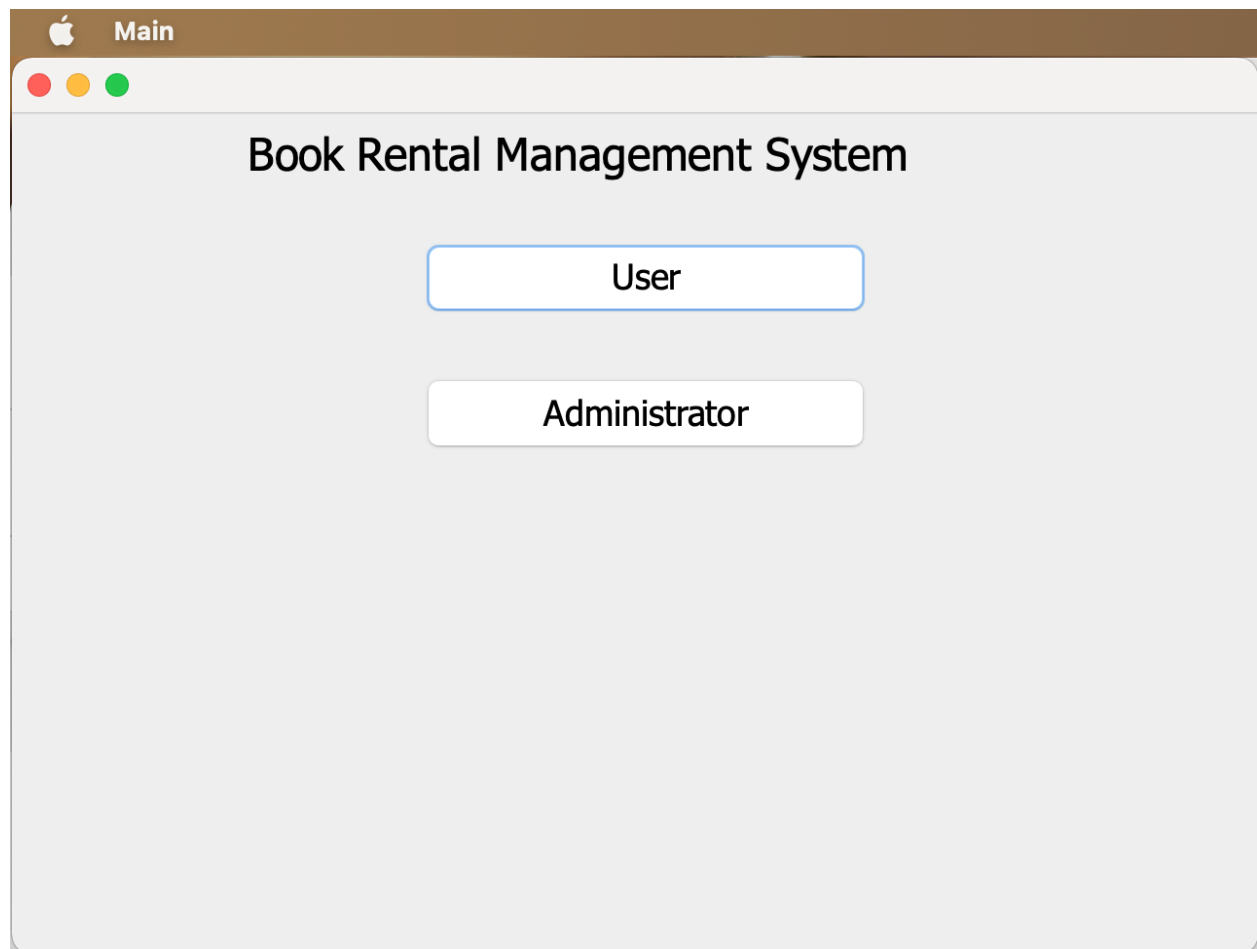
Java, Eclipse IDE, MySQL(open-source relational database management system), MySQL Community Server, MySQL JDBC Connector, Databases, JDBC, GUI, rs2xml.jar(used to display the data in a table format).

Key Activities

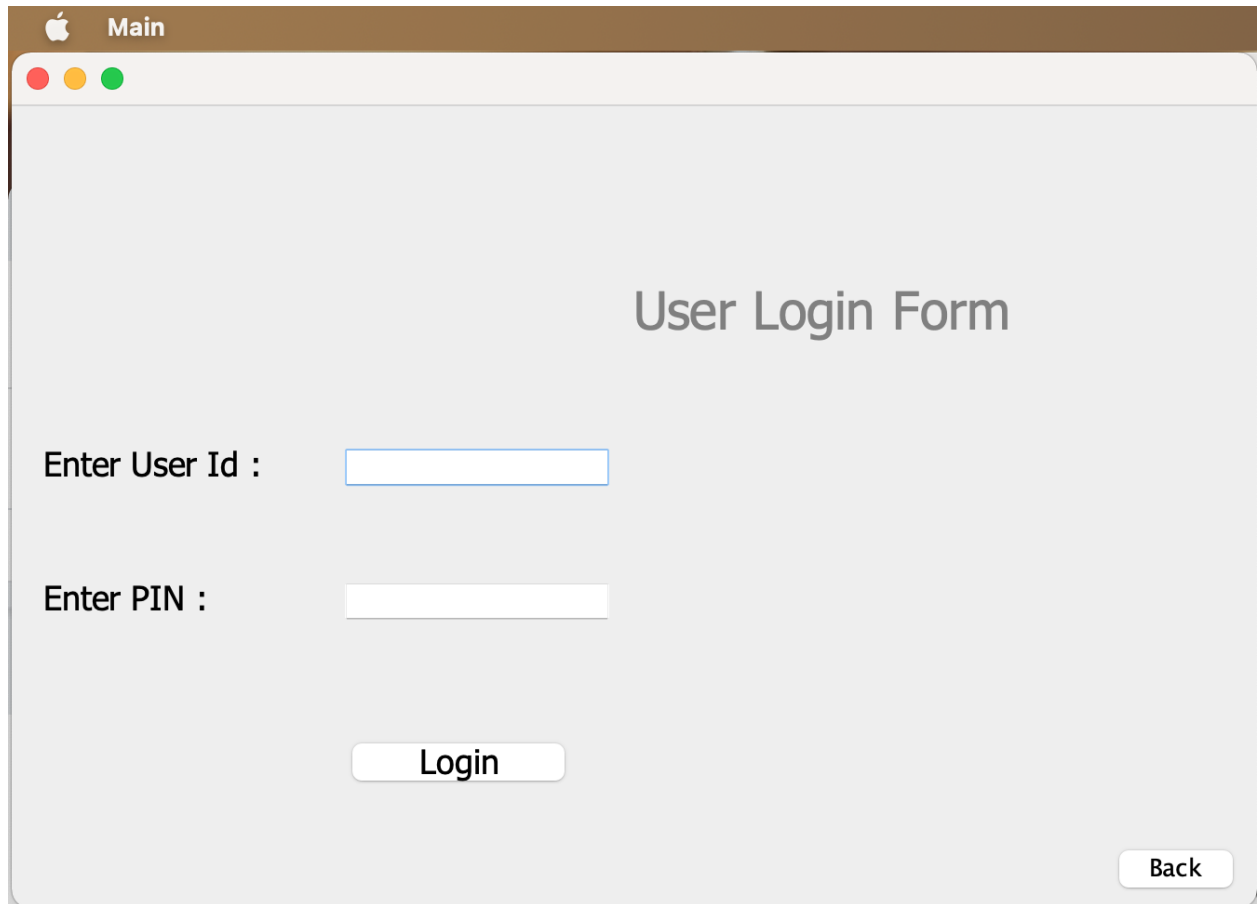
1. Download MySQL
2. Import the rs2xml.jar and JDBC connector JAR into the project
3. Use Swing to create GUI(graphical user interface) components such as scroll bars, buttons, dialog boxes.
4. Connect the MySQL database to the GUI
5. Create the database, tables and data into these tables
6. Create a Login class for admin and create a Login class for users with Runnable interface to implement multi-threading.
7. Connect, create and reset database.
8. Create User Menu functions.
9. Create Admin Menu functions.

Design and Development of the Project with Image Results

1. Main. java
Run Main.java



Click User,



A screenshot of a macOS-style window titled "Main" with a brown title bar and standard red, yellow, and green window control buttons. The window has a light gray background and contains a "User Login Form". The form includes two input fields: "Enter User Id :" and "Enter PIN :". Below these fields is a "Login" button. In the bottom right corner of the window is a "Back" button.

Main

User Login Form

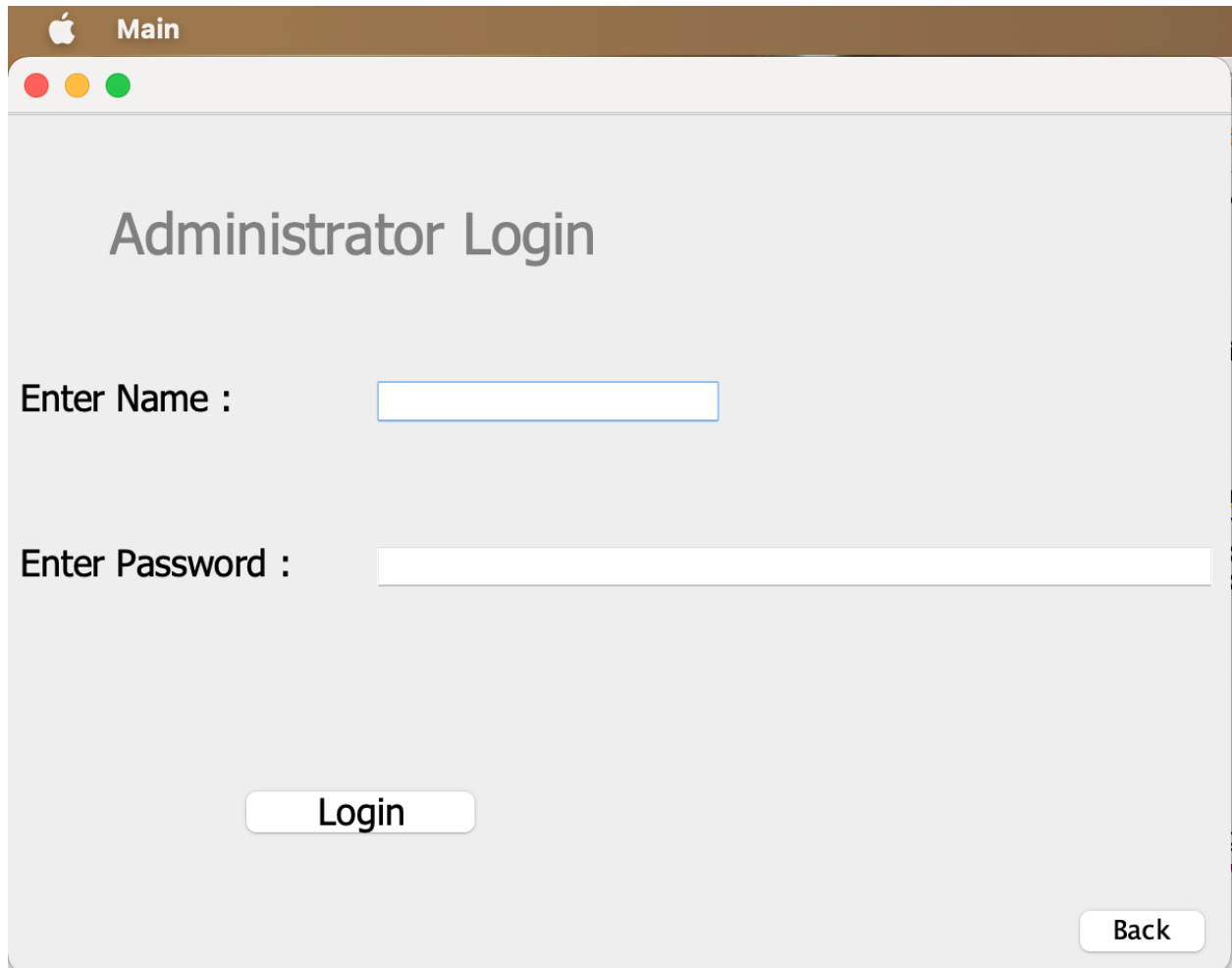
Enter User Id :

Enter PIN :

Login

Back

Back, click Administrator,



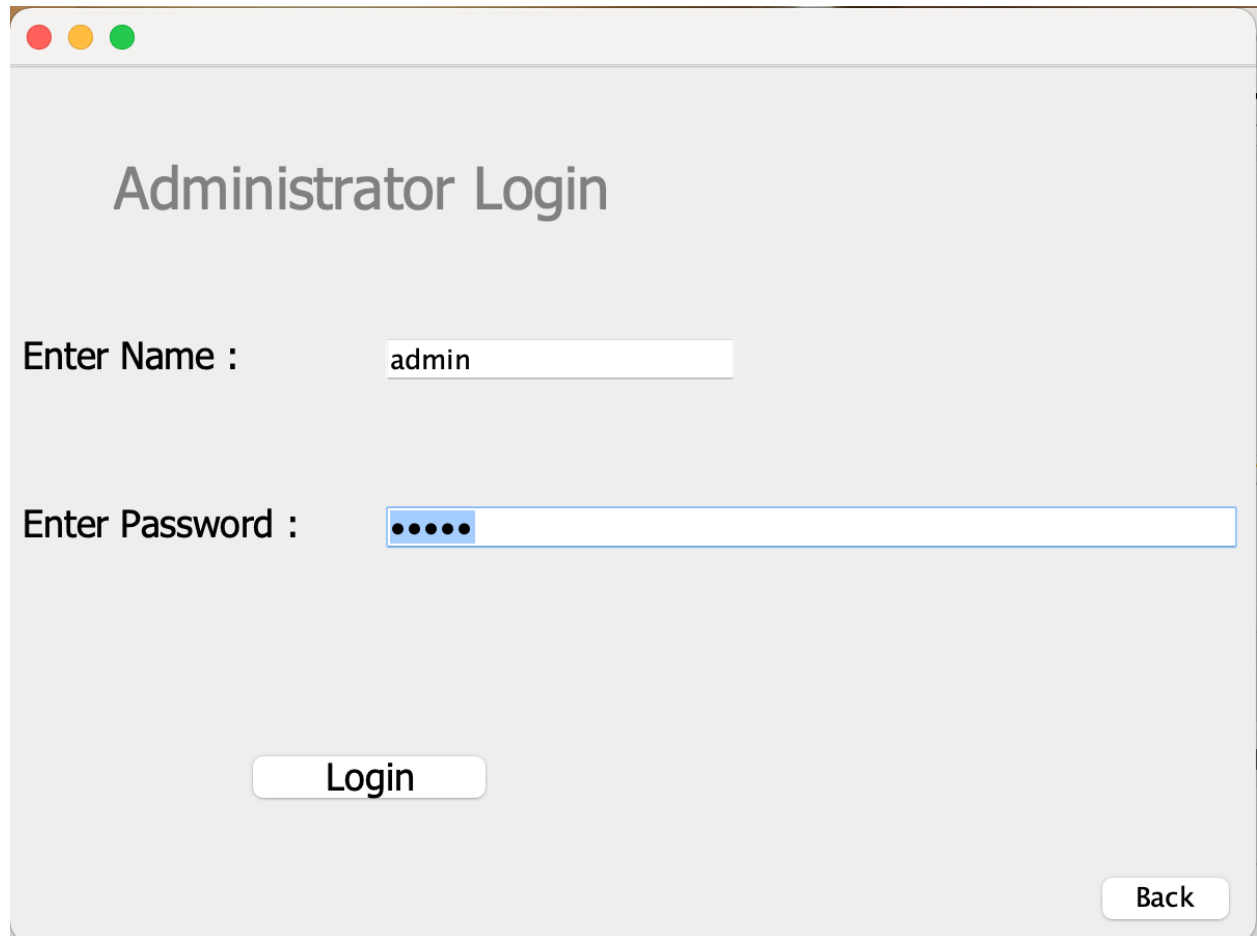
A screenshot of a Java Swing window titled "Main" with an Apple logo icon. The window has a light gray background and rounded corners. It contains the following elements:

- Title Bar:** A brown bar at the top with an Apple logo and the text "Main". Below it are three colored window control buttons (red, yellow, green).
- Header:** The text "Administrator Login" is centered in a large, dark gray font.
- Form Fields:**
 - The label "Enter Name :" is followed by a small rectangular text input field.
 - The label "Enter Password :" is followed by a longer rectangular text input field.
- Buttons:**
 - A "Login" button is centered below the input fields.
 - A "Back" button is located in the bottom right corner.

2. AdministratorLogin.java

Create a Login class for admin with Runnable interface to implement multi-threading. Runnable is an interface, which provides the method run. Threads are implementations and use Runnable to call the method run().

Enter name "admin" and password "admin"



A screenshot of a Java Swing window titled "Administrator Login". The window has a light gray background and a standard macOS-style title bar with red, yellow, and green buttons. The title "Administrator Login" is centered at the top in a large, dark gray font. Below the title, there are two input fields. The first is labeled "Enter Name :" and contains the text "admin". The second is labeled "Enter Password :" and contains five black dots, indicating a password field. Below the password field, there is a "Login" button. In the bottom right corner, there is a "Back" button.

Administrator Login

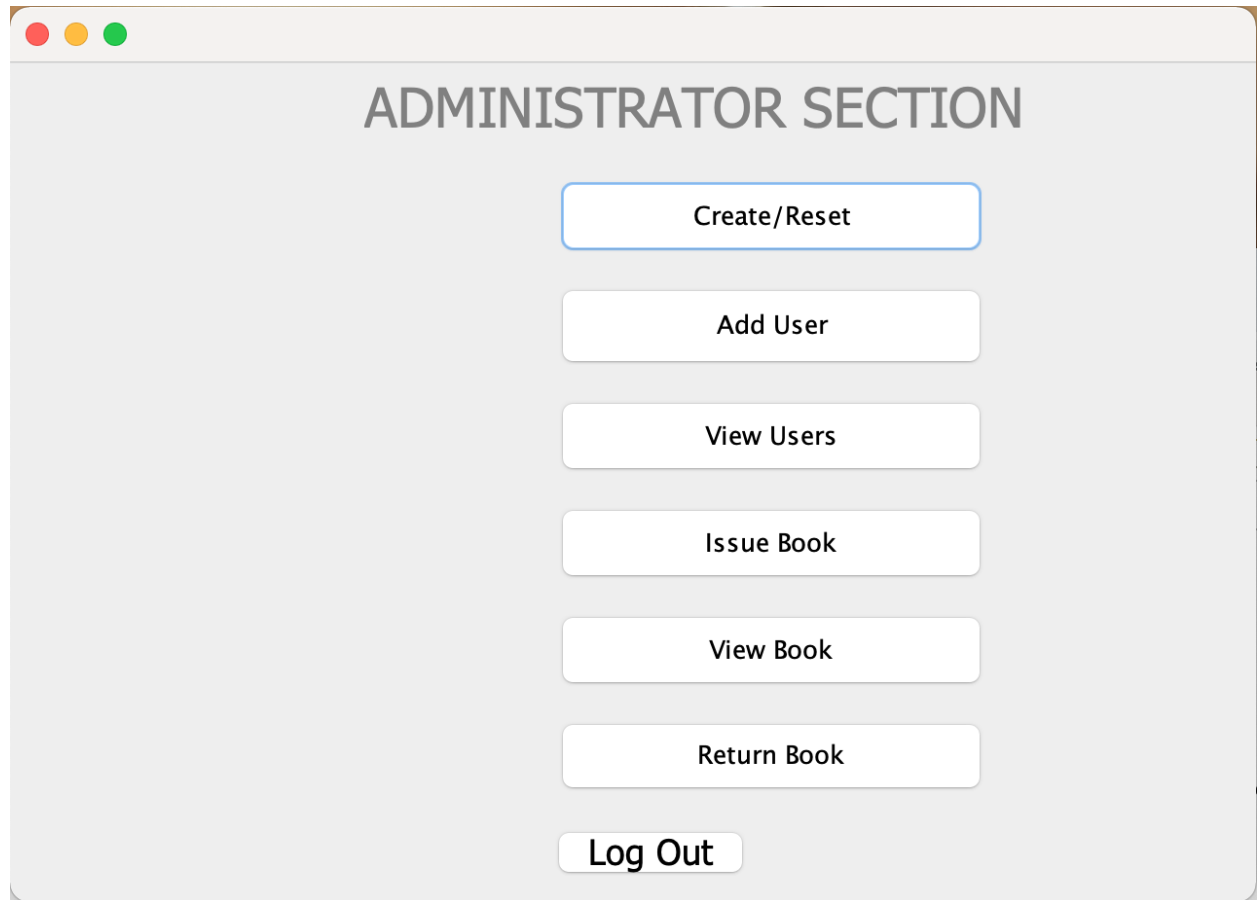
Enter Name :

Enter Password :

Login

Back

If successful, the AdminSuccess.java will automatically run.
And you will see this,

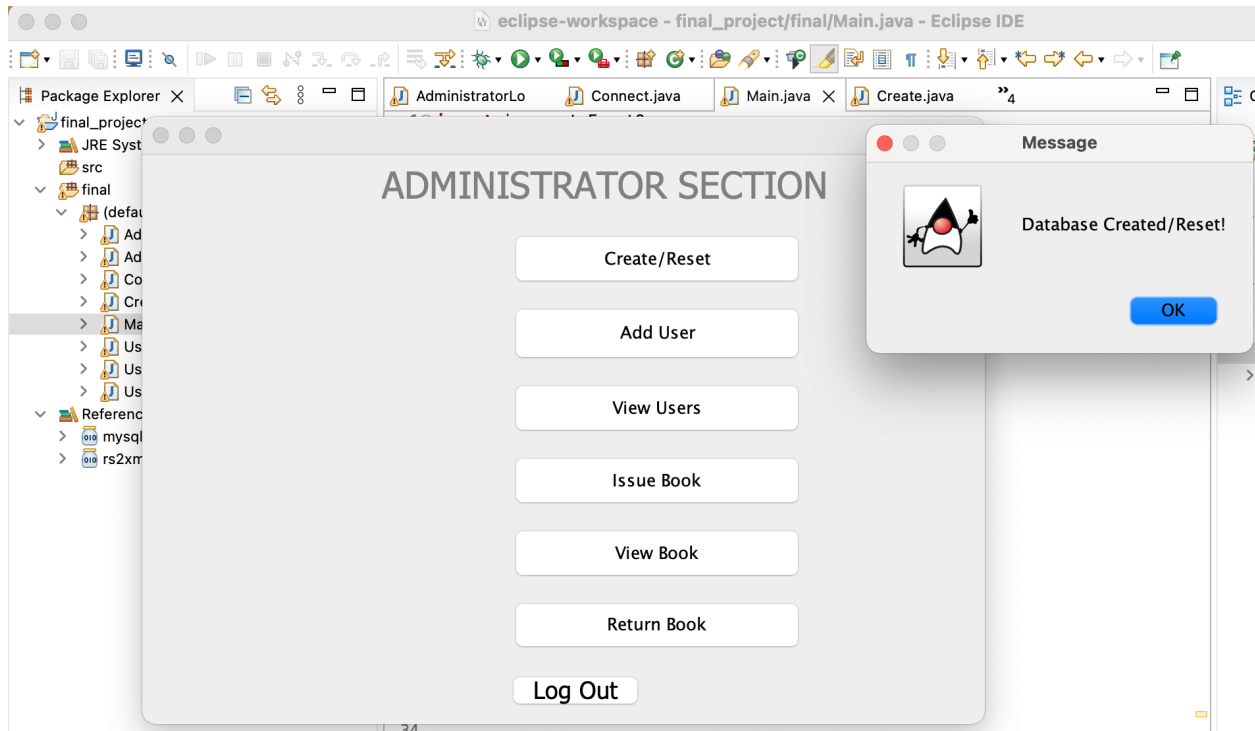


3. AdminSuccess.java

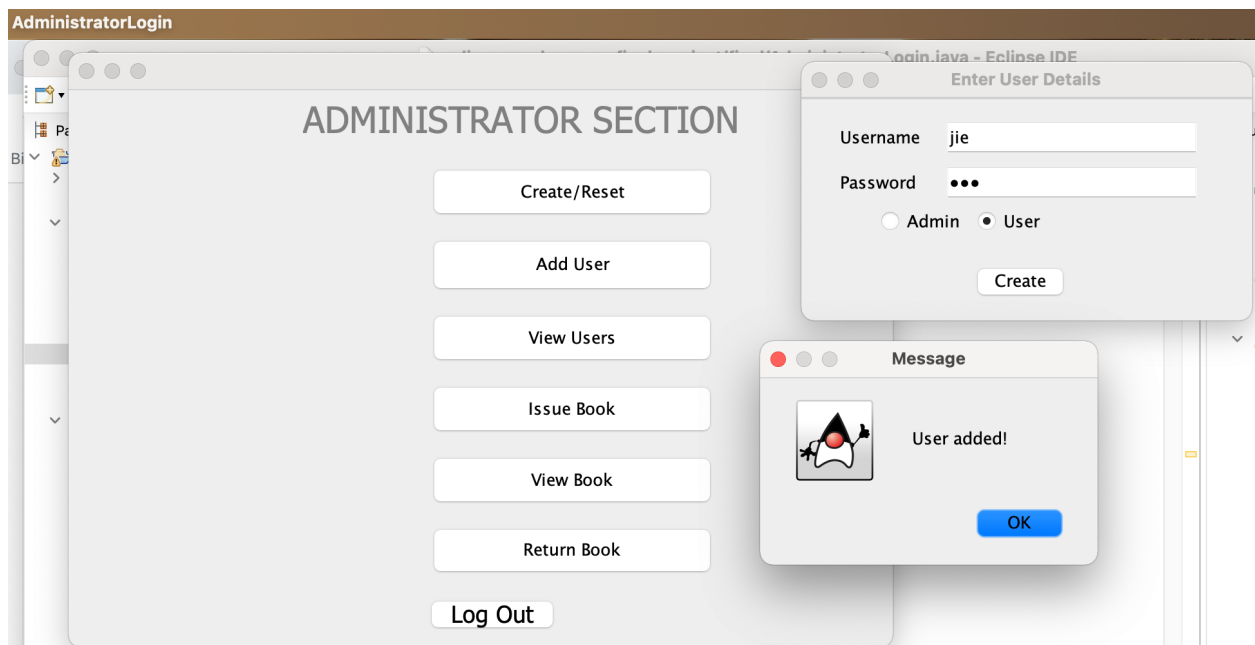
Create/Reset:

Connect.java is used to connect the database to the GUI. We are connecting our MySQL database with the username "root" and password "abc123##" to our application. Once the application is connected to the database, then we create or reset the database.

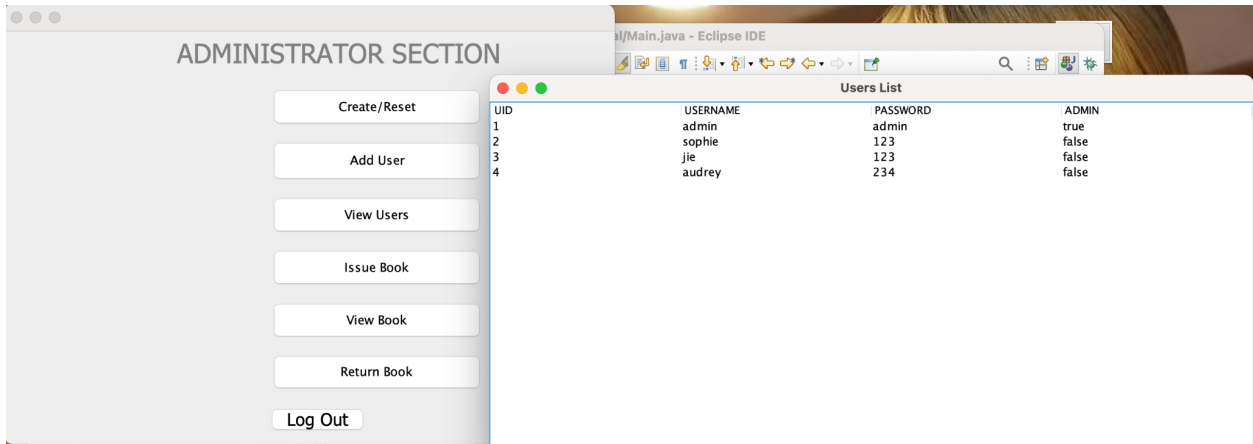
Create.java is used to create the database, tables and add data into these tables. We use SQL statements to achieve this.



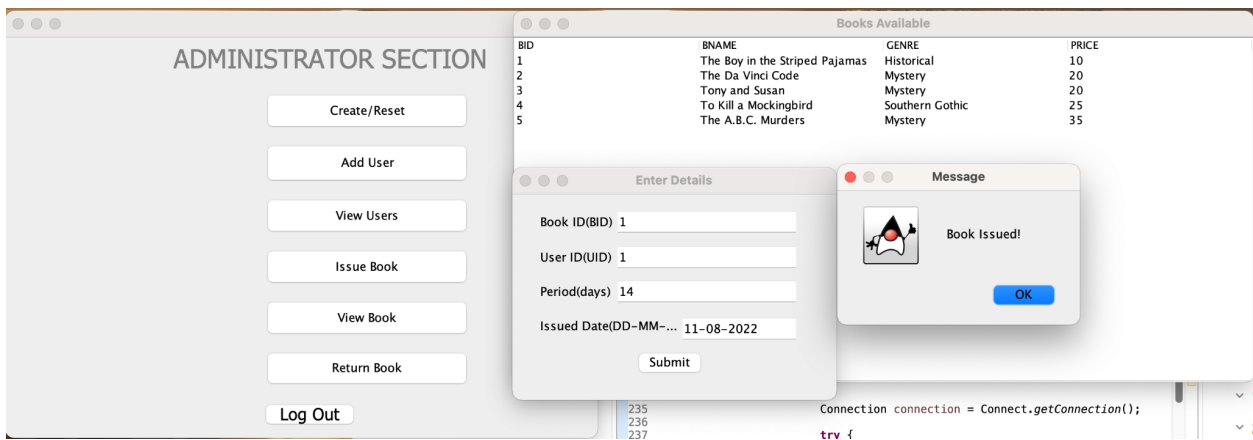
Add User:



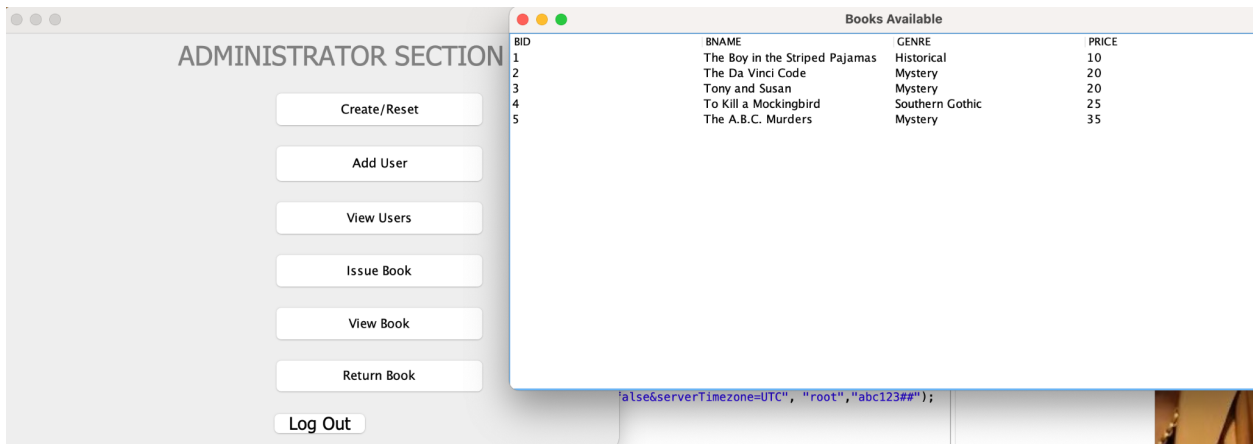
View Users:



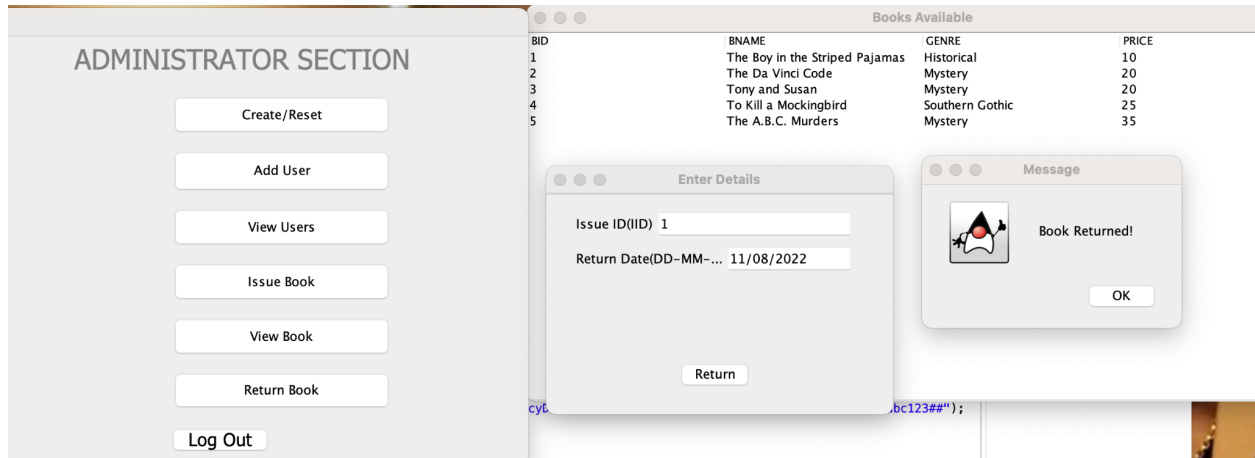
Issue Book:



View Book:



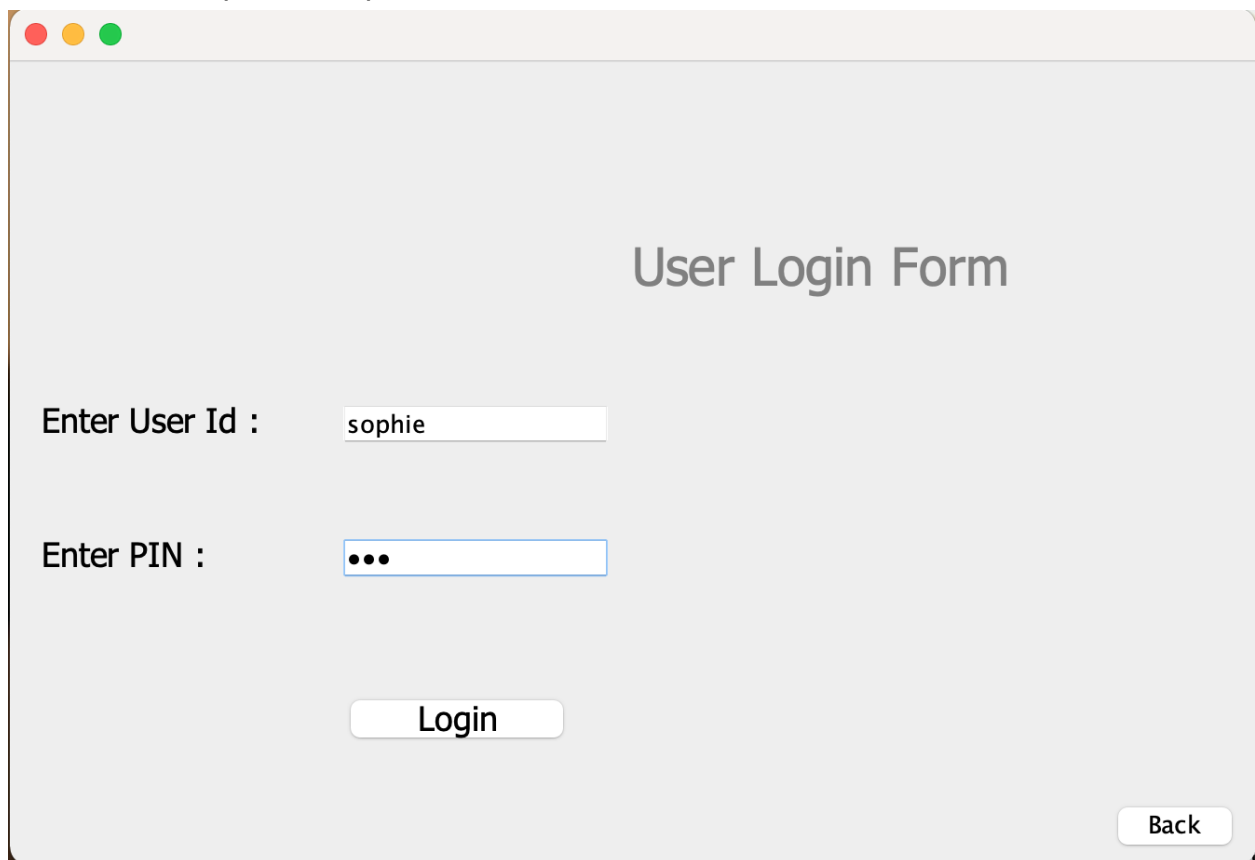
Return Book:



4. UserLogin.java

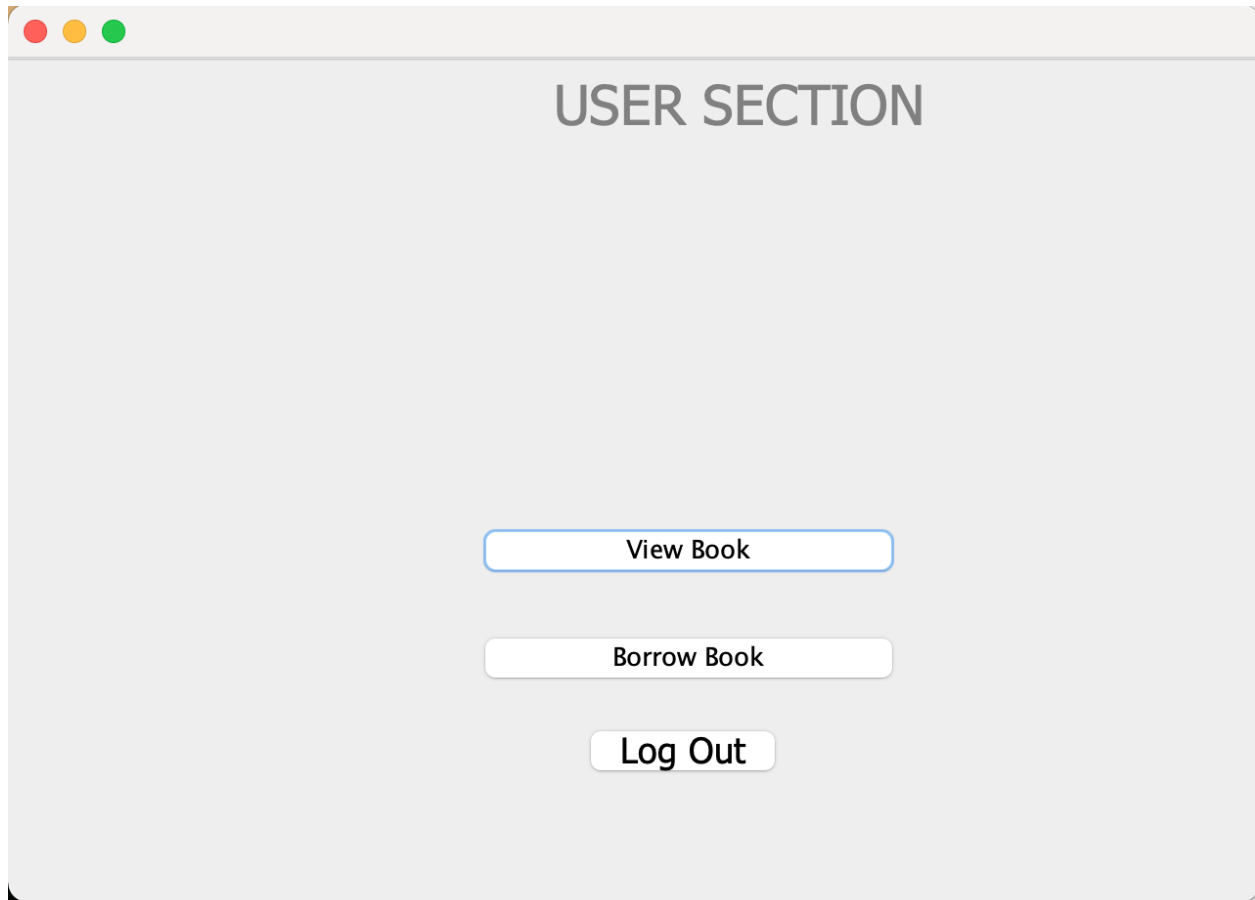
Create a Login class for users with Runnable interface to implement multi-threading. Runnable is an interface, which provides the method run. Threads are implementations and use Runnable to call the method run().

Enter name "sophie" and password "123"



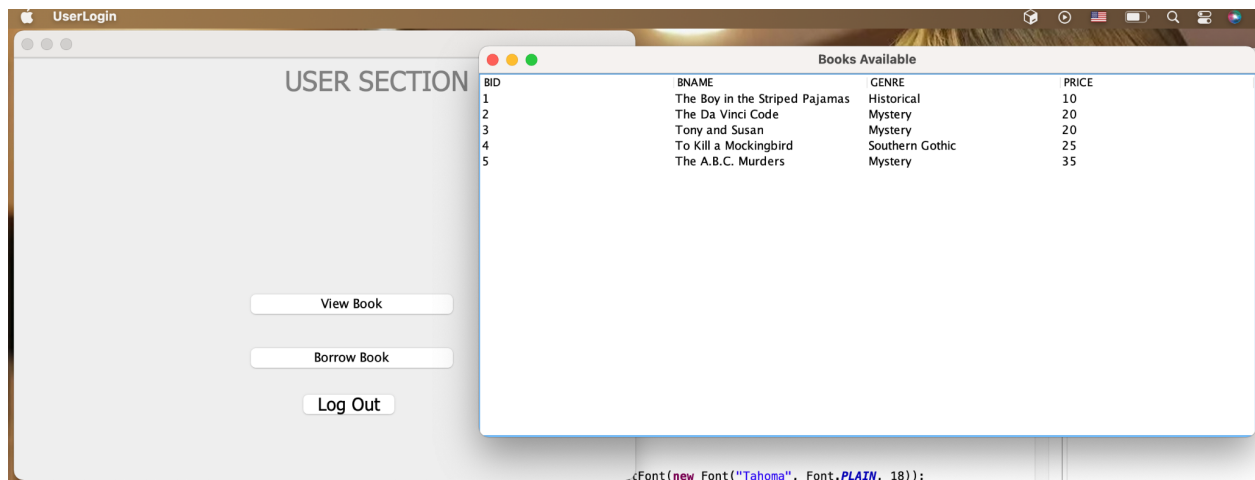
If successful, the UserSuccess.java will automatically run.

And you will see this,



5. UserSuccess.java

View Book:



Borrow Book: user sophie with UID 2 borrow the book with BID 2

