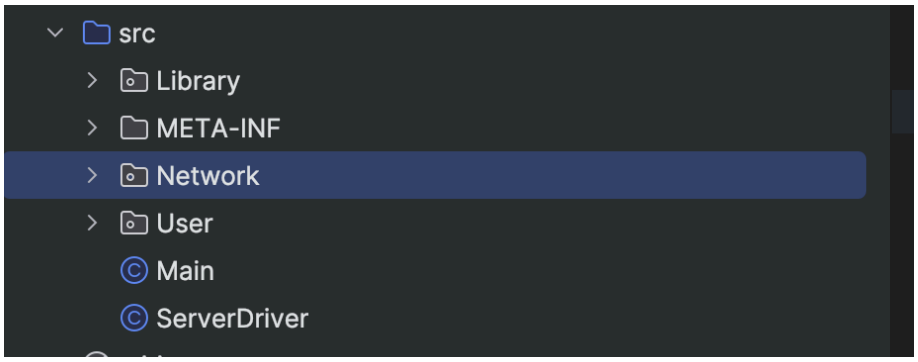
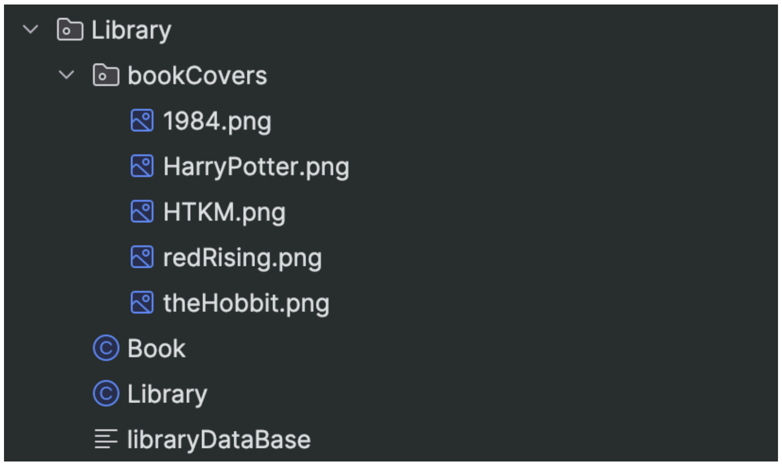
Final Project 6
Library DataBase

Programmers POV:
Structure:
Server Side:



My code is structured into three main parts, library, network, and User. The code is launched using the ServerDriver.java. This file simply initializes my User database, the Library database and initializes the Server and which port to use.

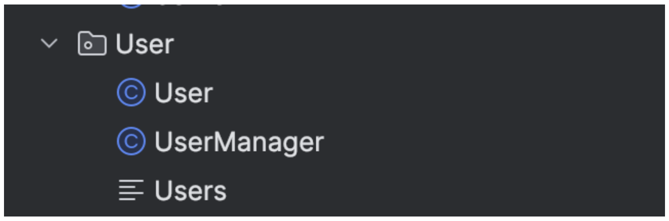


Library Structure:

I have two classes that run my database and keep track of the library. The book class represents any object that can be stored in the library. The library class is a collection of books. At the start of the program the library class reads through the libraryDataBase.txt file and generates a library.

```
"Red Rising","Pierce Brown","Book","2","/bookCovers/redRising.png","A slave of the new world rises up and breaks the chains of society"
```

Each line in the file is structured like so.
“Title”, “Author”,”Type”,”Available”,”image url”,”A description”
In my Library package, it also includes bookCovers which is all of my images.



User Structure:

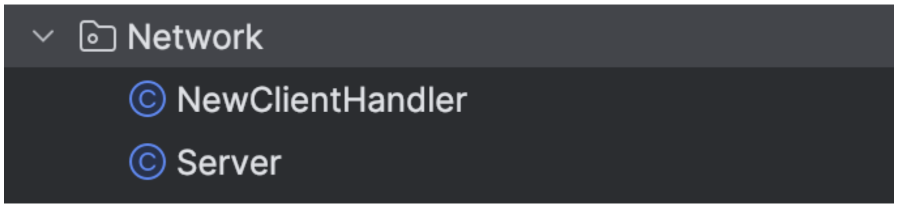
My user structure is very similar to my Library structure. I have a UserManager and a user class. The User class is each person that has access to the database. Each User has a name, password, if they are a librarian, and a list of borrowed items. The UserManager is a collection of Users and is used to organize and create new users. At the start of the program the UserManager reads a from Users.txt and creates a database of users.

```
Simon blue true
```

Each line looks like this

“name “ “password” “if Librarian”

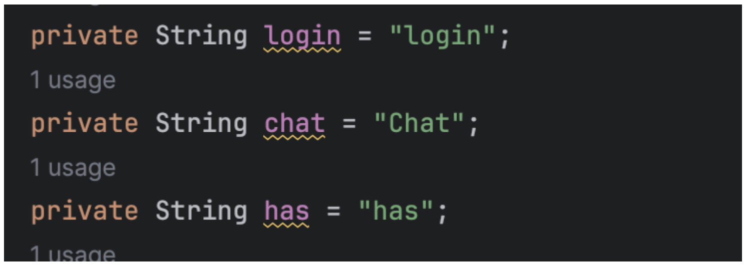
The librarian field is optional. If it is not there/ specified it defaults to false.



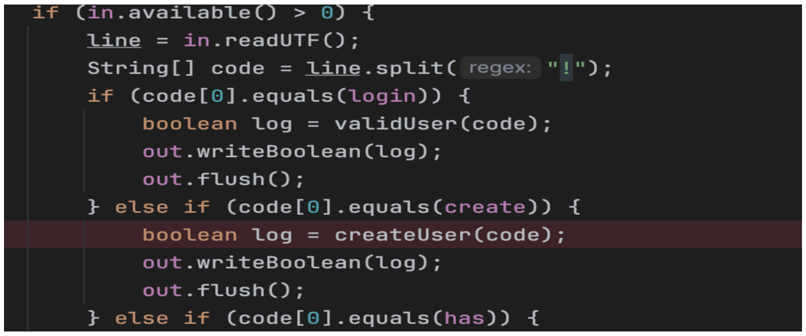
Network Structure:

After the creation of the library and user database my program creates a new Server. This class creates a new ServerSocket and then runs a infinite loop constantly looking for new clients. When a new client is found, it creates a NewClientHandler thread and starts it.

The new client handler deals with every request from the client.



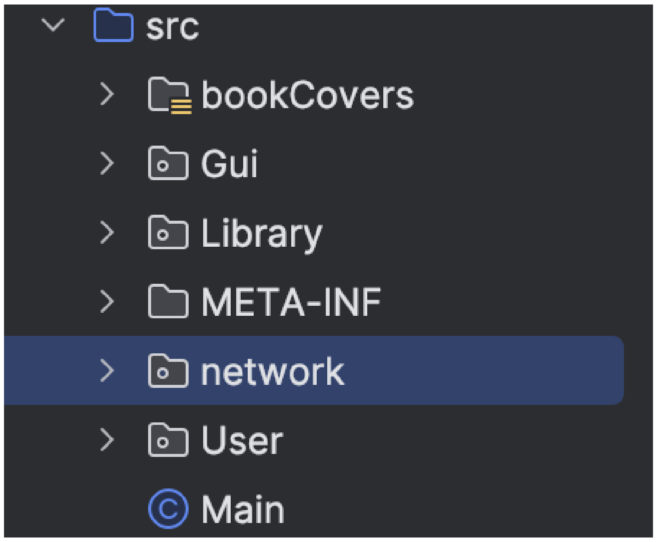
```
private String login = "login";
1 usage
private String chat = "Chat";
1 usage
private String has = "has";
1 usage
```



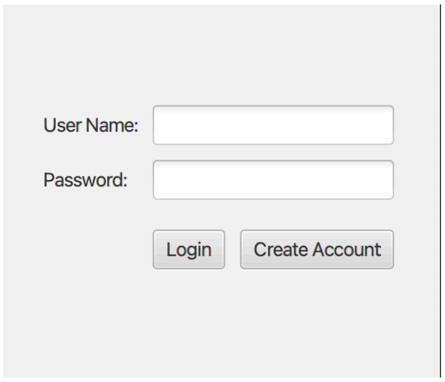
```
if (in.available() > 0) {
    line = in.readUTF();
    String[] code = line.split( regex: "!");
    if (code[0].equals(login)) {
        boolean log = validUser(code);
        out.writeBoolean(log);
        out.flush();
    } else if (code[0].equals(create)) {
        boolean log = createUser(code);
        out.writeBoolean(log);
        out.flush();
    } else if (code[0].equals(has)) {
```

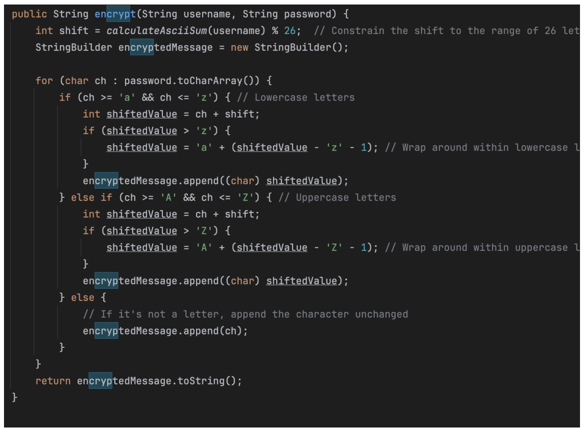
NewClientHandler:

My NewClientHandler works by constantly waiting for a new String from the client. As a string is passed to the server it is then split up into an array. Each string passed starts with a code word that tells the server what is being requested.
Ex: line = login!Simon!password
Each word is separated with an “!”. There is a total of 11 different commands that the client can ask the server to perform.



Client:
My client is structure almost the same way but it also includes a GUI class.
My client side is started from the GUI class. Here is creates a library field and then calls my Gui constructor. Here is where you can change the IP address of the Client. As the program opens it first prompts you for a username and password or the option to create a user. All users passwords on the client side are stored with an encryption.
There username is converted into its ascii value and then their password is shifted to the right that value. The letters are shifted but stay in a range from ‘a’ to ‘z’ and ‘A’ to ‘Z’. Once the passwords are sent to the Server they are decrypted using the same method.

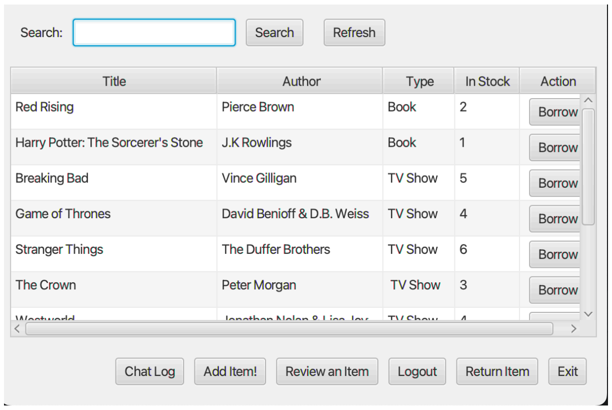




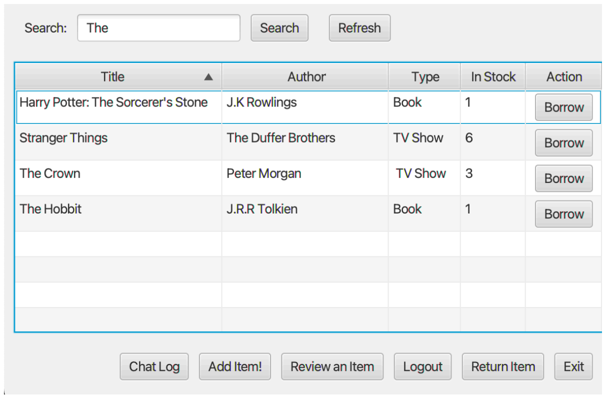
```java
public String encrypt(String username, String password) {
    int shift = calculateAsciiSum(username) % 26;  // Constrain the shift to the range of 26 let
    StringBuilder encryptedMessage = new StringBuilder();

    for (char ch : password.toCharArray()) {
        if (ch >= 'a' && ch <= 'z') { // Lowercase letters
            int shiftedValue = ch + shift;
            if (shiftedValue > 'z') {
                shiftedValue = 'a' + (shiftedValue - 'z' - 1); // Wrap around within lowercase l
            }
            encryptedMessage.append((char) shiftedValue);
        } else if (ch >= 'A' && ch <= 'Z') { // Uppercase letters
            int shiftedValue = ch + shift;
            if (shiftedValue > 'Z') {
                shiftedValue = 'A' + (shiftedValue - 'Z' - 1); // Wrap around within uppercase l
            }
            encryptedMessage.append((char) shiftedValue);
        } else {
            // If it's not a letter, append the character unchanged
            encryptedMessage.append(ch);
        }
    }
    return encryptedMessage.toString();
}
```

After you log in or create a user, the program sends a request to the server and downloads the library database from the server.

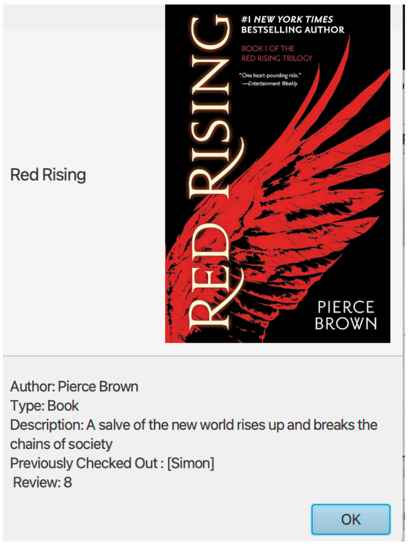


Title	Author	Type	In Stock	Action
Red Rising	Pierce Brown	Book	2	Borrow
Harry Potter: The Sorcerer's Stone	J.K Rowlings	Book	1	Borrow
Breaking Bad	Vince Gilligan	TV Show	5	Borrow
Game of Thrones	David Benioff & D.B. Weiss	TV Show	4	Borrow
Stranger Things	The Duffer Brothers	TV Show	6	Borrow
The Crown	Peter Morgan	TV Show	3	Borrow
Westworld	Jonathan Nolan & Lisa Joy	TV Show	4	



Title ▲	Author	Type	In Stock	Action
Harry Potter: The Sorcerer's Stone	J.K Rowlings	Book	1	Borrow
Stranger Things	The Duffer Brothers	TV Show	6	Borrow
The Crown	Peter Morgan	TV Show	3	Borrow
The Hobbit	J.R.R Tolkien	Book	1	Borrow

Here you can manual scroll through the listed options or you have two options to search through the database. You can use the top search bar to find anything you are looking for or you can click each category. It will rearrange the items by alphabetical order or high-low/low- high for the In Stock field.


Red Rising
#1 NEW YORK TIMES BESTSELLING AUTHOR
BOOK I OF THE RED RISING TRILOGY
RED RISING
PIERCE BROWN
OK

Author: Pierce Brown
Type: Book
Description: A salve of the new world rises up and breaks the chains of society
Previously Checked Out : [Simon]
Review: 8

When you click on any item in the library, a page will open with a photo and a short description of the item. This includes a review out of 10 and who previously checked it out.

Features:

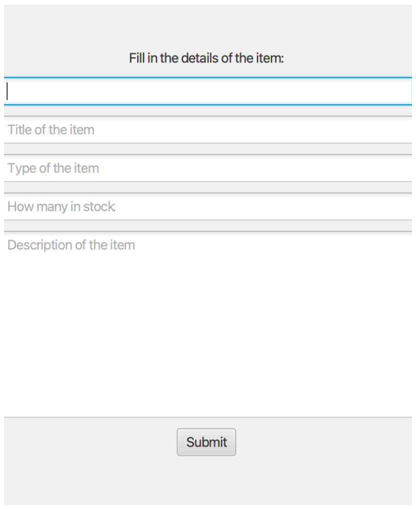
As a user you have three options.

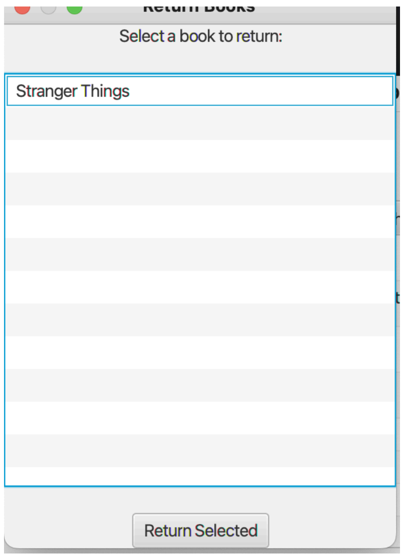
Borrow: You can borrow any item in the library as long as there is one available and you do not already have it checked out.

Return: Here you can return any item that you have already checked out. This allows you to recheck out an item you have already checked out.

Review:You can also leave a review. You enter the books name and a rating from 1-10.

As a librarian you have two additional options.
Chat log: You have a chat log where you can communicate with any other librarian logged in. The chat updates in real time.
Add Item: As a librarian you also have the option to add an item.





Adding a book: **Returning a book:**

Once you are done you can either logout and log into another account or simply quit the application.

References:
https://www.w3schools.com/
https://docs.oracle.com/javase/tutorial/java/data/strings.html
https://www.geeksforgeeks.org/
redit.com
Youtube.com - Arthur Spirin/ Ryi Snow