Bibliotec

Library Management Software

Team 4: Gabriel Peregrino, Rishi Raja, Bolormaa Munkhbold

San Jose State University

CS 157A

Final Report

Professor: Aravind Rokkam

05/3/2024

Table of Contents

1.1 Project Introduction

Bibliotec helps libraries efficiently and accurately manage their books, loans, and patrons. It also helps patrons easily find new books and request holds. Librarians/administrators have their own view which allows them full control to view and manage patrons, books, loans, and holds. The application also allows patrons to view their loan history, view available books to loan, customize their information and view/request their own holds, with a more limited access. With the increasing need for digital solutions, Library Management Systems(LMS) have become essential for libraries to streamline their operations and improve user experience. This project report presents the design and implementation of an app-based LMS that allows librarians to manage their collection, and patrons to easily find and borrow books.

1.2 Objective:

The objective of this project is to design and implement a web-based Library Management System (LMS) that provides efficient and user-friendly management of library resources, including books, patrons, and checkouts. The system will provide features such as user authentication, book management, patron management, loan and return functionality, database transactions, and a responsive navigable user interface. The system is developed using Java, with JavaFX as the UI framework, Gradle as the build tool, and MySQL as the database. The project aims to improve library management and user experience by providing a modern and efficient LMS.

The specific objectives of this project are:

- To design and implement an LMS as a desktop application that provides efficient and user-friendly management of library resources, including books, patrons, and checkouts.
- To provide user authentication and role-based access control, ensuring that only authorized users can access the system's features.
- To provide book management functionality, allowing librarians to add, edit, and delete books from the system.
- To provide book genres and categories, allowing users to easily discover new books based on their preferences.
- To provide patron management functionality, allowing librarians to add, edit, and delete patrons from the system.
- To provide loan and return functionality, allowing patrons to borrow and return books using the system.
- To provide database transactions, ensuring that the system's data is consistent and reliable.
- To be reliable and work without internet connectivity or over a local network
- To document the system's design, implementation, and testing, providing a comprehensive record of the project.

1.3 Project High-Level Design

In Depth Feature Explanation

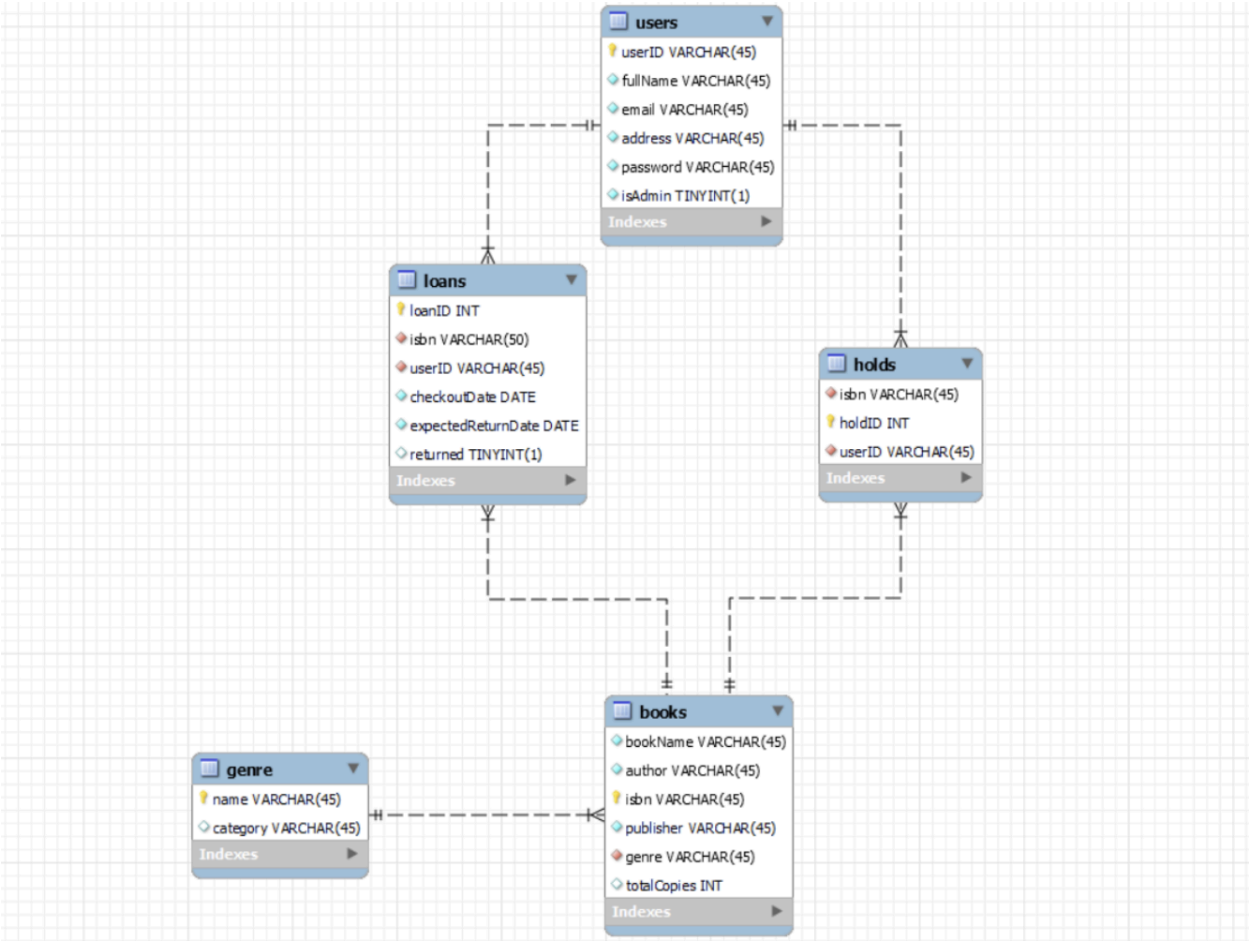
- Patron Registration
    - Upon clicking Register, patrons are prompted to create a username and password, and provide their full name, email, and address.
    - If the Username is already registered, an error message will be displayed, explaining the username has already been taken.
- Patron Login
    - After a Patron logs in using their username and password, they will be able to see the patron view of our UI.
    - They will see 4 tabs: Books, Loans, Holds, and My Profile
        - Books
            - Shows all books in the library including their title, author, isbn, publisher, genre, total copies, and available copies.
            - Allows the patron to create a hold for a book that is already on loan.
        - Loans
            - Shows all the loans the user has ever made.
            - References loans using its LoanID, the books being lent with the ISBN, the checkout date, the return date, and a check box where the admin can confirm if the book has been returned or not.
        - Holds
            - Shows all the holds the user has made.
            - References the hold with a holdID, ISBN of the book, and the date the hold was requested.
            - Holds may also be canceled by the user.
            - Holds cannot be made if there are available books. Instead, loans can be taken out at the library.
        - My Profile
            - Allows the user to change any of their information, as well as change their password.
- Librarian View
    - If in the creation of the user, the attribute “isAdmin” is set to true, the user will be able to access the Librarian View of the UI when logged in.
    - They will see 4 tabs: Books, Patrons, Loans, and Holds.
        - Books
            - Shows all the books in the library just as Patron View.
            - Allows the librarian to add/delete books with buttons at the bottom.
            - Includes another table on the right that will show the loan history of a book when it is selected.
        - Patrons
            - Allows the librarian to view all the patrons who have made an account.

- When selected, it will show the specific patrons loan history in the table to the right.
- Allows the librarian to delete users as well.

■ Loans
- Allows the librarian to view all the loans made as well as the user ID of the patron who has the loan.
- Librarians are able to add and delete loans as well as check and uncheck the “Returned” box to confirm if the book has been returned or not.

■ Holds
- Allows the librarian to view all the holds made, including the user who made it, the date, and the book placed on hold.
- Allows the librarian to add and delete holds.

1.4 ER Diagram



1.5 Normalization Table

1NF -

- a single cell must not hold more than one value (atomicity)
- there must be a primary key for identification
- no duplicated rows or columns
- each column must have only one value for each row in the table

2NF -

- be in 1NF
- has no partial dependency.

3NF -

- be in 2NF
- have no transitive partial dependency.

Our tables satisfy 1NF as each table has a primary key, no duplicated rows or columns, and each column has only one value for each row in the table.
Our tables also satisfy 2NF as each table has no partial dependencies, which means all the attributes are dependent on the primary key only.
Finally, our tables also satisfy 3NF as each table has no transitive dependencies, which means none of our non-key attributes are dependent on another non-key attribute.

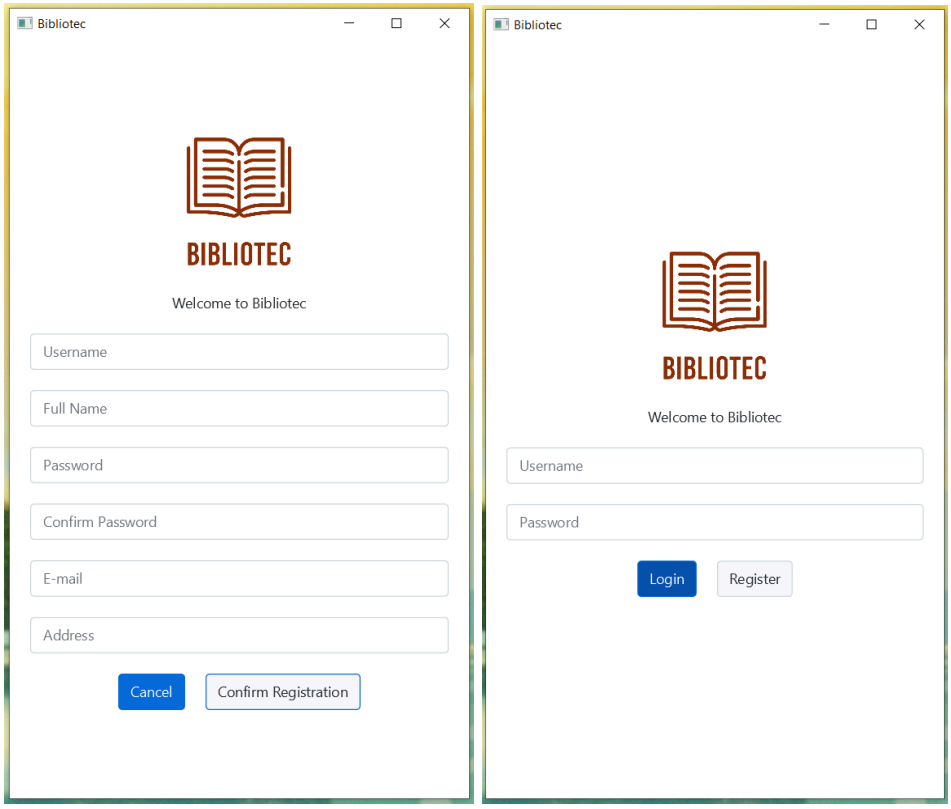
1.6 Challenges and concerns / Results

One of the most significant challenges we faced involved connecting our database to our Java application. As it was our first time creating an application that involved storing our values in a MySQL database, we applied a decent amount of our efforts towards learning how to use the MySQL connector dependency. As well as calling queries within our Java methods to accurately fill and retrieve tables from the database and displaying them onto our UI. Our last biggest challenge was syncing information we added or edited in our database to our UI. However after those first few hurdles, we were able to easily create our CRUD operations for our objects that connected to our database. A few concerns we had were primarily focused on our user security as well as database security. We tackled these concerns by using environmental variables for our MySQL password so it would not be explicitly shown in the connector. Furthermore, we were able to use BCrypt which allowed us to hash passwords for our users and admins.

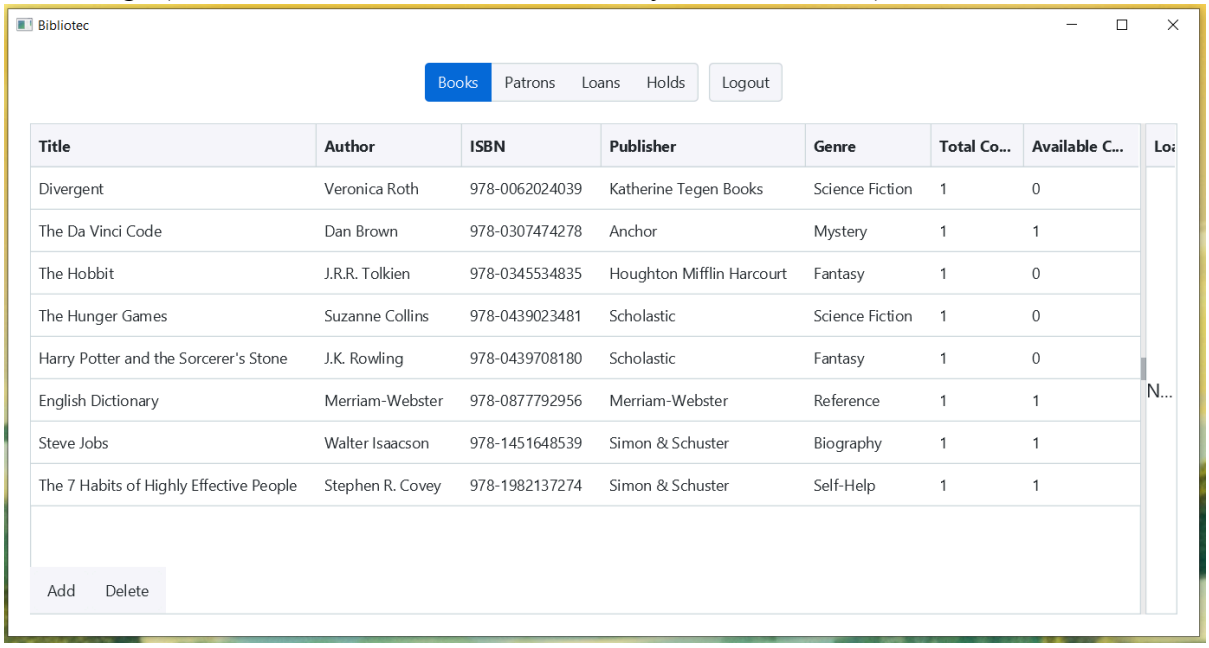
In terms of our results, I believe that we managed to achieve all of the objectives we stated in our proposal. We were able to implement User Authentication, Book Management, Patron Management, Checkouts and Returns, Database Transactions, and a Responsive Web Interface as stated in our proposal. We also added extra features such as the ability for users to request a hold for a book that is already out on loan. For our tech stack we did end up straying from our proposal, such as using JavaFX instead of SpringBoot and we also did not have to use React.

1.7 Screenshot of the application with all the views

Login Page & Register Page

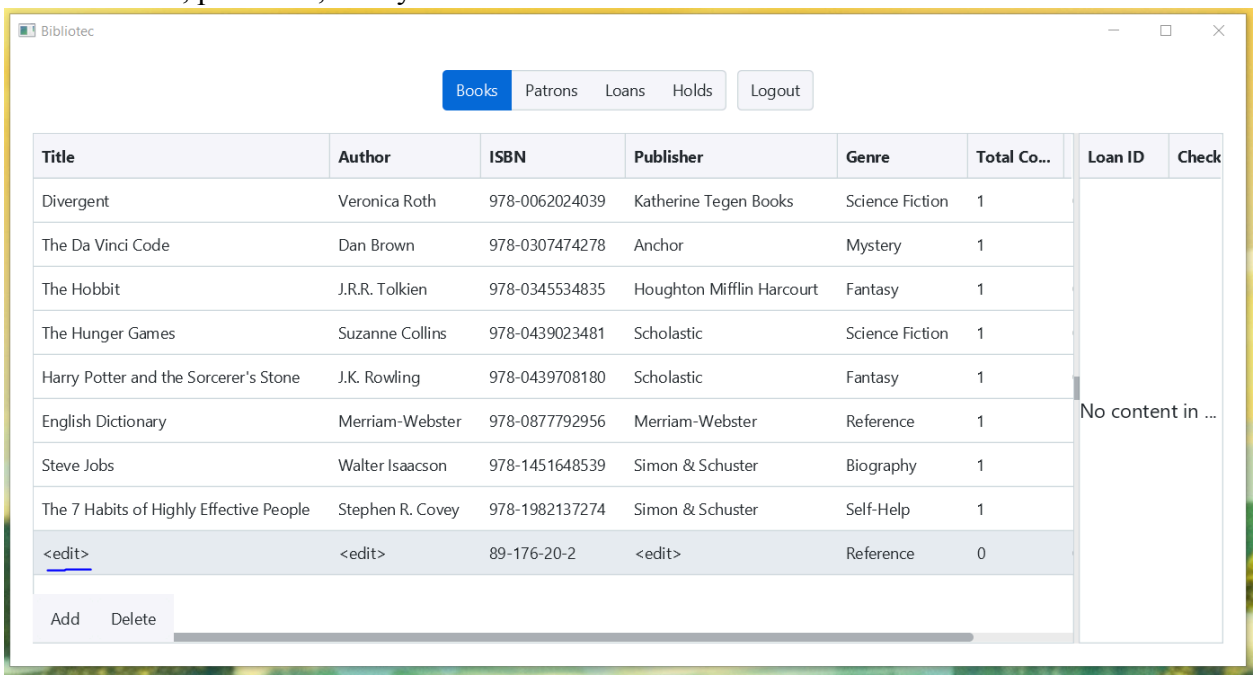


Admin View - (Consists of Add/Delete functionality for all tables)
Books Page (select on a book to see the loan history on the left table)

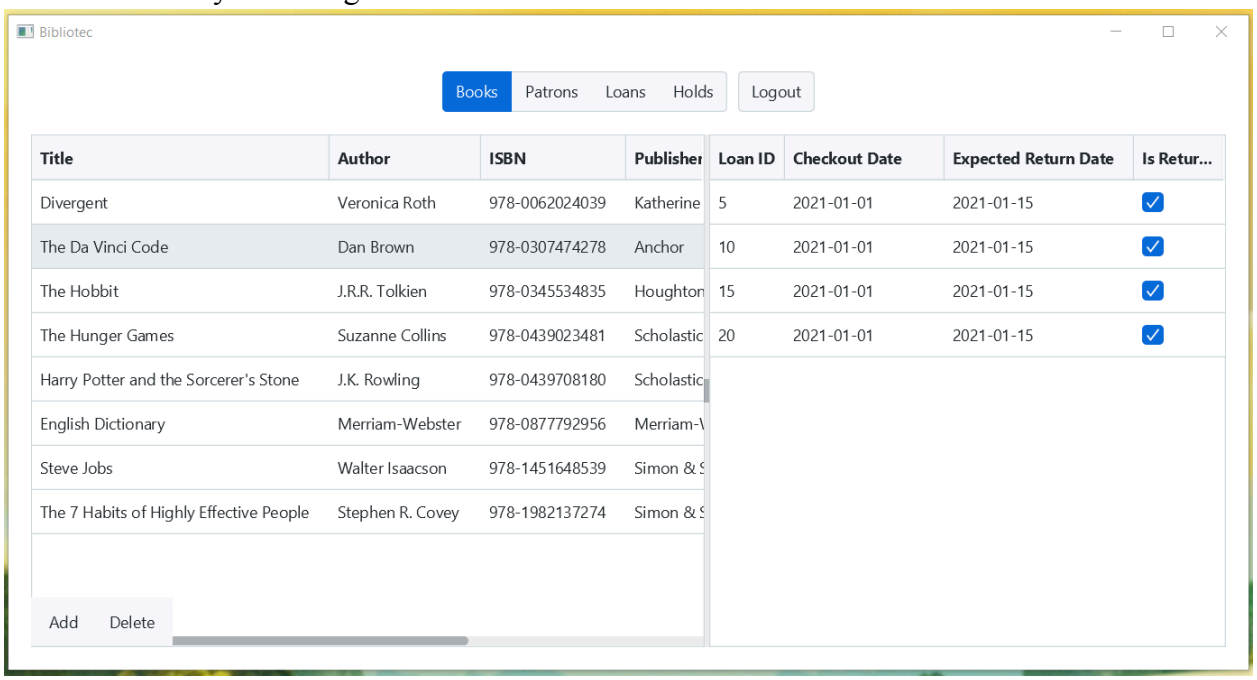


Title	Author	ISBN	Publisher	Genre	Total Co...	Available C...	Loa
Divergent	Veronica Roth	978-0062024039	Katherine Tegen Books	Science Fiction	1	0	
The Da Vinci Code	Dan Brown	978-0307474278	Anchor	Mystery	1	1	
The Hobbit	J.R.R. Tolkien	978-0345534835	Houghton Mifflin Harcourt	Fantasy	1	0	
The Hunger Games	Suzanne Collins	978-0439023481	Scholastic	Science Fiction	1	0	
Harry Potter and the Sorcerer's Stone	J.K. Rowling	978-0439708180	Scholastic	Fantasy	1	0	
English Dictionary	Merriam-Webster	978-0877792956	Merriam-Webster	Reference	1	1	N...
Steve Jobs	Walter Isaacson	978-1451648539	Simon & Schuster	Biography	1	1	
The 7 Habits of Highly Effective People	Stephen R. Covey	978-1982137274	Simon & Schuster	Self-Help	1	1	

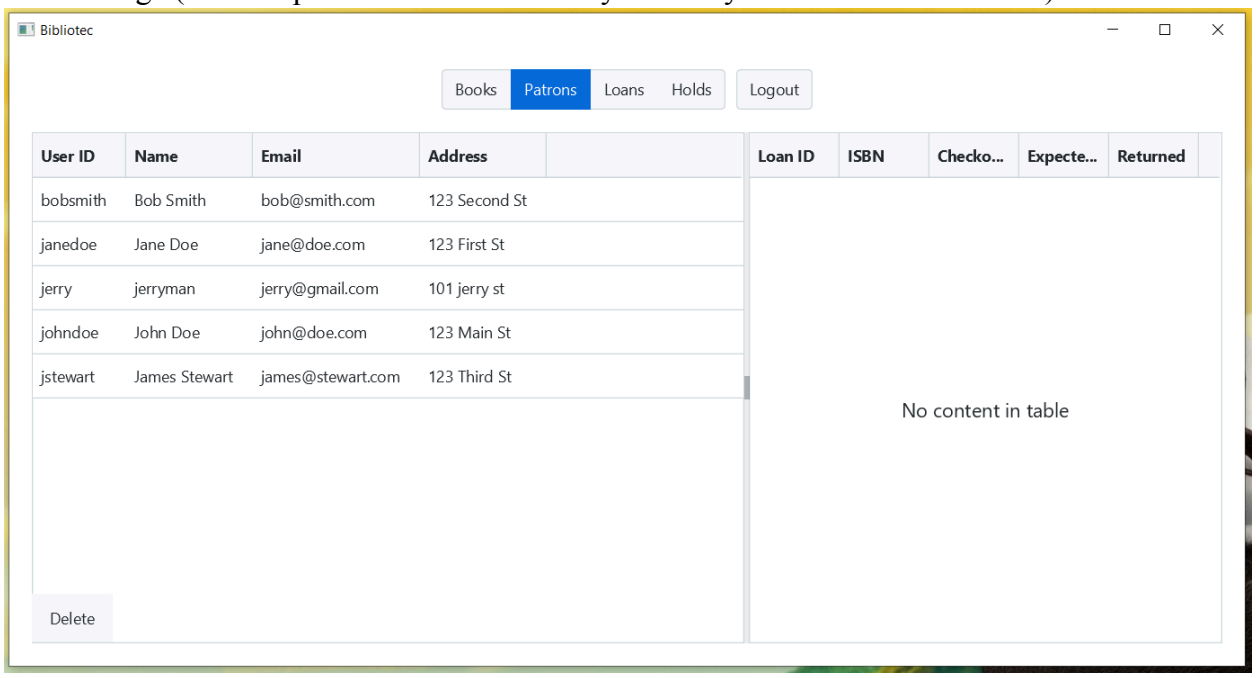
To add a book, press add, then you will be able to edit the information for each column



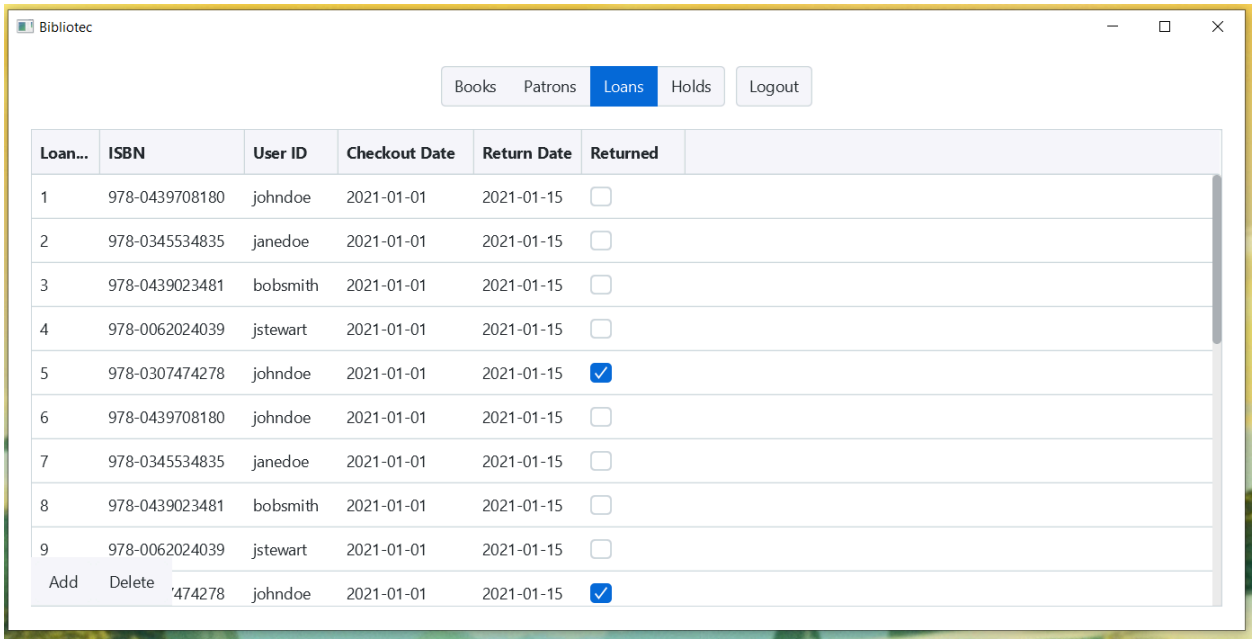
Book loan history on the right table



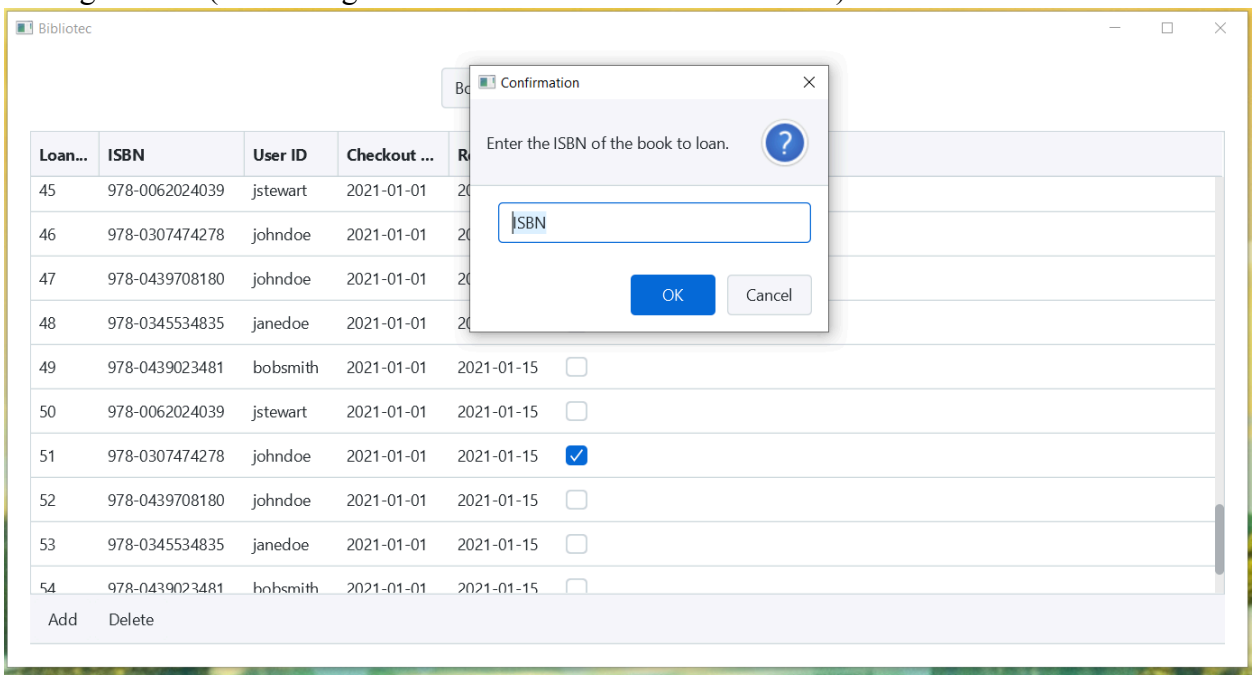
Patrons Page (Select a patron to view loans they currently have/have made before)



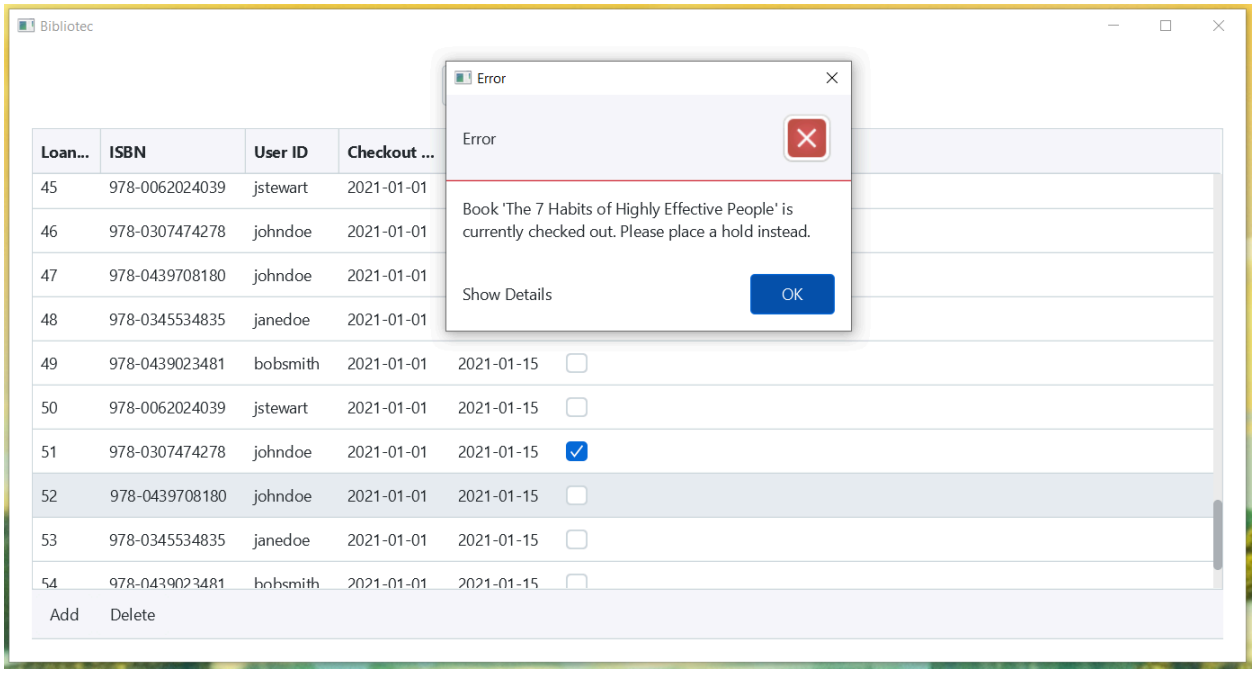
Loan Page (Shows all loan history)



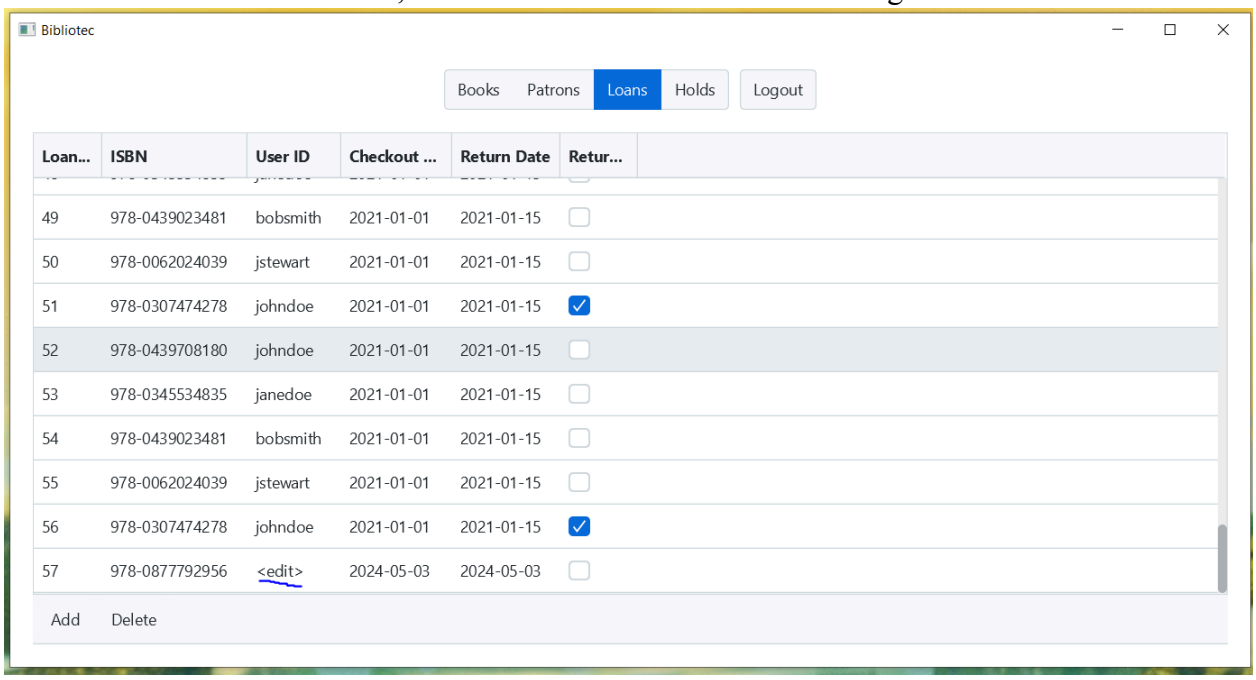
Adding a loan - (Will also give an error for an non-existent ISBN)



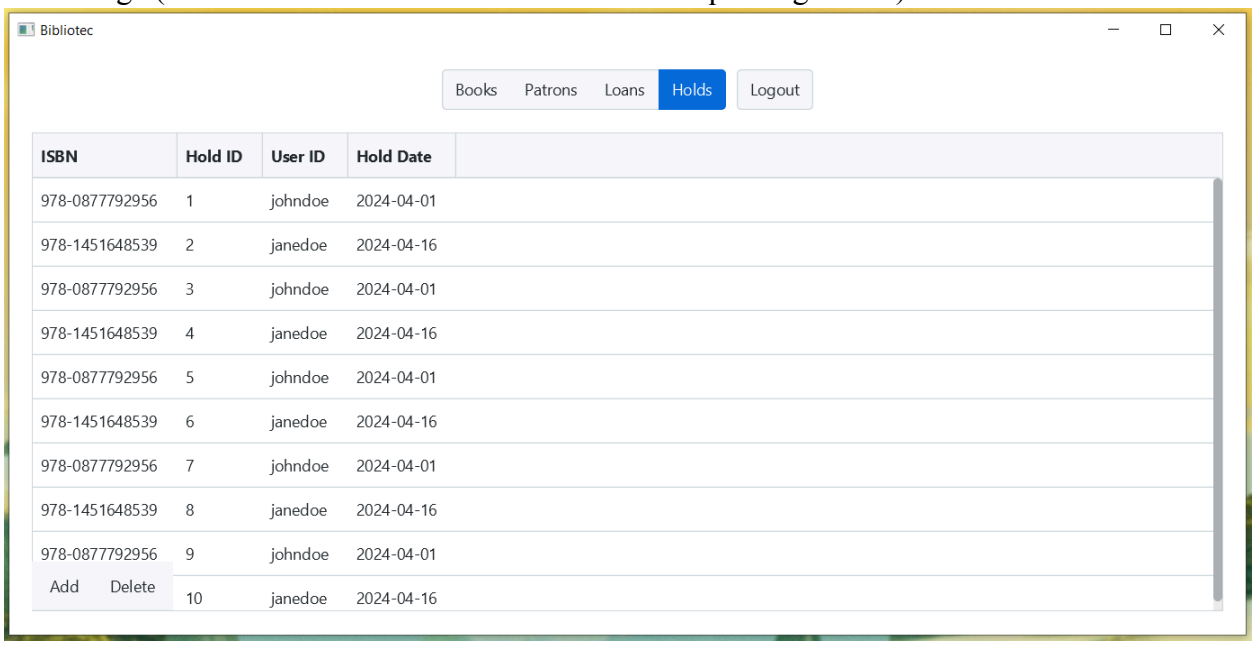
If you attempt to add a loan for an ISBN that is out on loan you will get an error message -



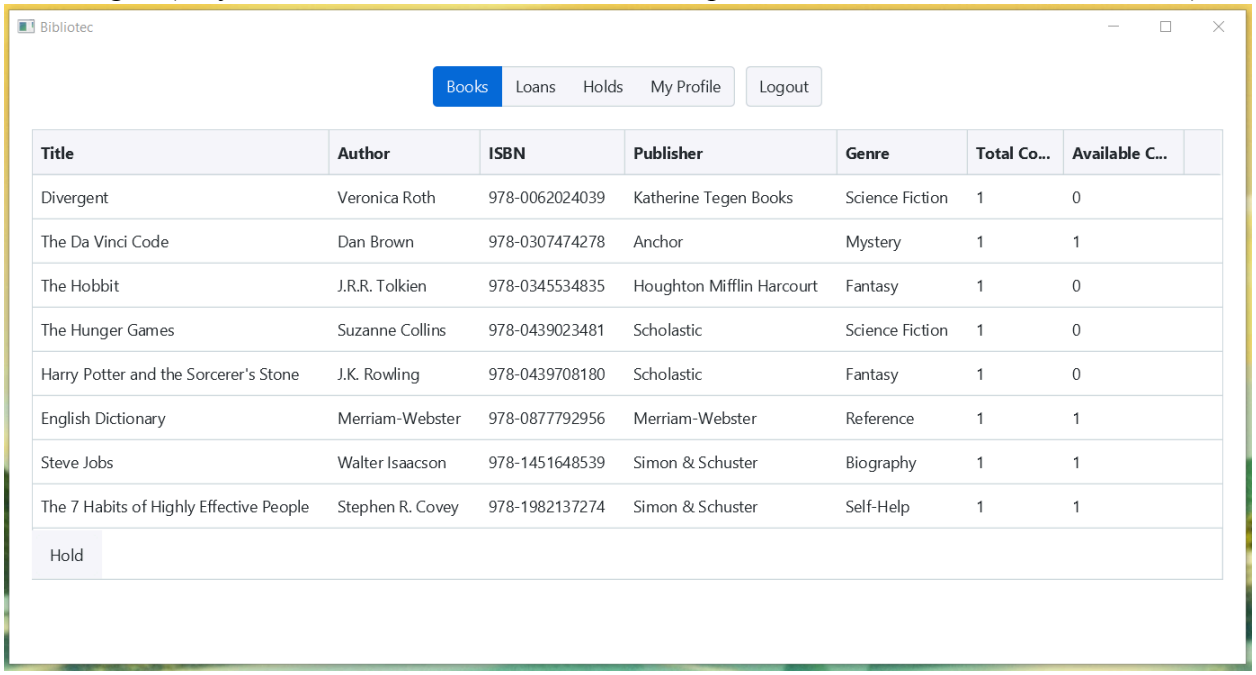
Once the loan has been created, the admin can edit the user ID to assign the loan.



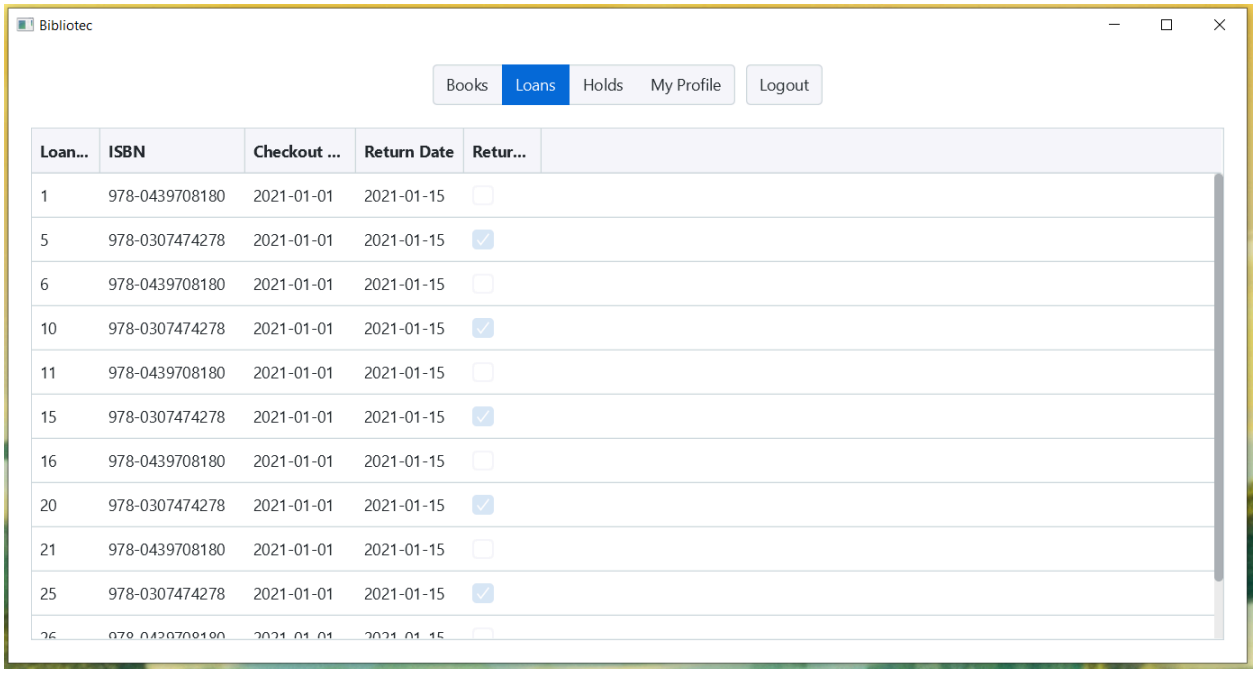
Holds Page (shows all holds for books and the users requesting a hold)



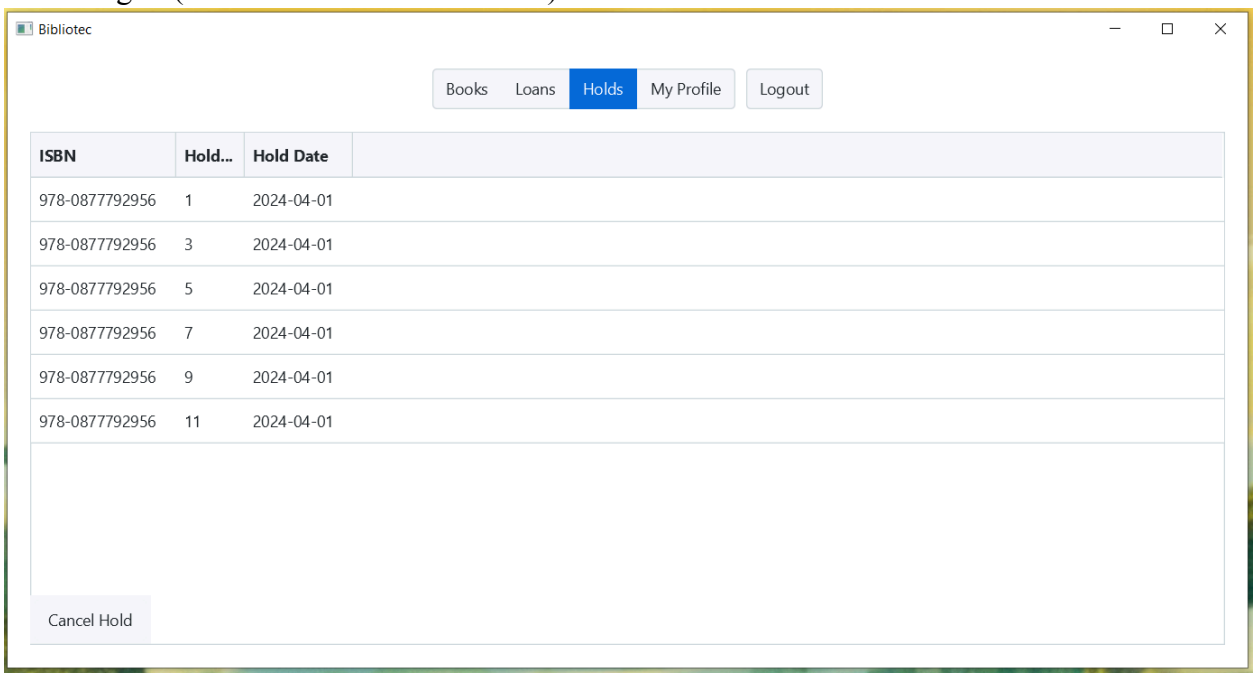
Patron View - (View only, however can add/cancel hold for themself)
Book Page - (only shows books, as well as available copies, can also create a hold for a book)



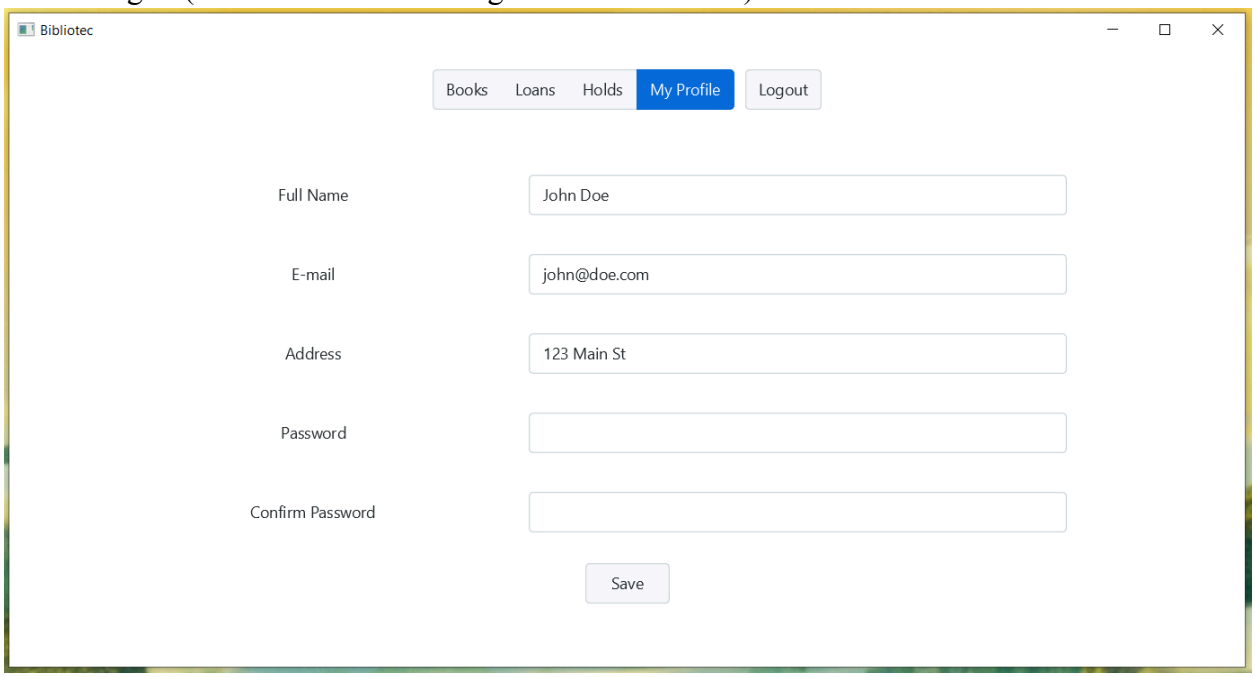
Loans Page - (shows the loans the user has)



Holds Page - (shows the holds the user has)



Profile Page - (allows the user to change their information)



1.8 Contribution/Work done by each team member

Gabriel Peregrino: I worked on finalizing our project proposal, creating the schema for our database, creating the ER diagram, figuring normalization for our tables and connecting our MySQL database with the MySQL connector for all our DatabaseAccess methods. Implementing our methods for adding/removing/getting patrons, books, loans, holds. Communicating mostly with Rishi to make sure our methods and tables matched with the UI. I also was able to implement BCrypt for hashing our user passwords. I also learned how to use JavaFX and was able to help create the user profile editor with Rishi's help. Finally, I was able to work on the final report adding the references, screenshots of the application, normalization, high level design, and results.

Rishi Raja: I set up the project structure with Gradle and JavaFX, and wrote most of the UI and logic for communication. Work on the UI included the login and registration window, displaying/editing the tables, validating user input, and handling/displaying errors. Since this is a desktop application, there isn't a traditional "backend", but I worked on the features that are typically associated with a backend like handling user sessions and authentication. I also worked with Gabriel to create and normalize our database tables (including deciding on attributes) as well as using the MySQL connector to connect to our database. I also worked on debugging various issues we encountered. I wrote the README.md file with instructions on how to build and run the project.

Bolormaa Munkhbold: I assisted in creating presentation slides, giving an idea of ERD, and a project report. I used design and communication skills to create informative and visually appealing slides, an accurate ERD, and a comprehensive report that summarized the project's goals, methods, and outcomes. I worked collaboratively with the team to ensure that the deliverables were fine and make sure everything was on time.

1.9 References

Libraries Used:

JavaFX - https://openjfx.io
- UI Framework

AtlantaFX - https://github.com/mkpaz/atlantafx
- UI Styling

ControlsFX - https://github.com/controlsfx/controlsfx
- UI controls for JavaFX

MySQL Connector J - https://dev.mysql.com/downloads/connector/j
- Connecting UI to Database

MyBatis - https://mybatis.org/mybatis-3/
- Running SQL script files

1.10 Github Repository: https://github.com/gabeP232/Bibliotec.git

Instructions to build and run the project are included in the README.md file of the repository.