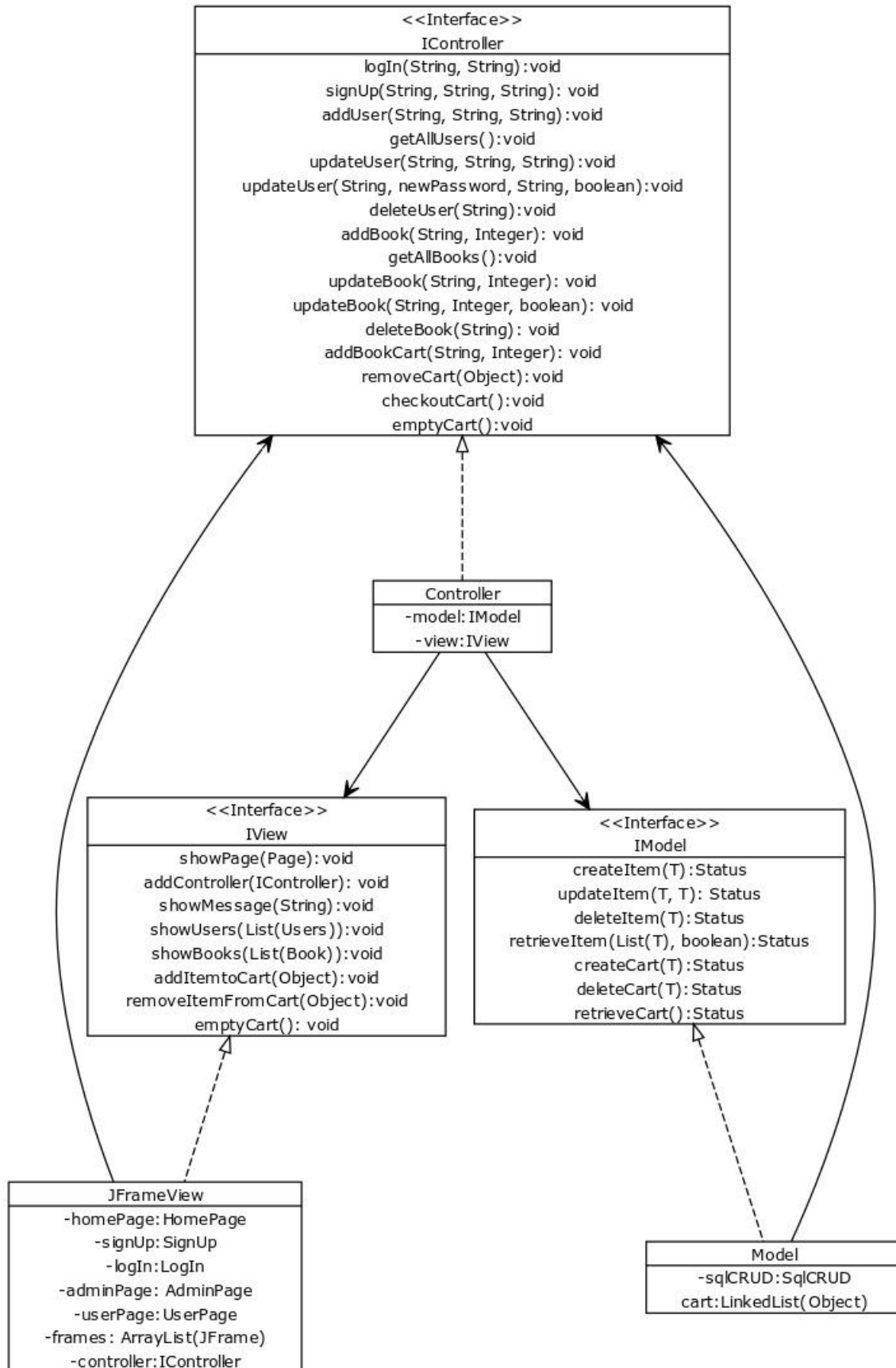# Book Inventory System Documentation

Introduction

The goal of this project is to build an inventory system that comes with a relational database and a user interface.

The program is built using Java. In addition, I used this opportunity to explore Java Reflection, several design patterns (i.e.,, Singleton, Facade and Factory patterns), the MVC architecture, and implemented Object Relational Mapping.

Design Documentation

The UML diagram is shown on the next page. Note that the diagram only shows the core part of the program, namely the MVC interfaces. A more detailed UML diagram can be found here.

```
                        <<Interface>>
                          IController
                    logIn(String, String):void
                   signUp(String, String, String): void
                  addUser(String, String, String):void
                        getAllUsers():void
                 updateUser(String, String, String):void
          updateUser(String, newPassword, String, boolean):void
                     deleteUser(String):void
                   addBook(String, Integer): void
                       getAllBooks():void
                   updateBook(String, Integer): void
                updateBook(String, Integer, boolean): void
                     deleteBook(String): void
                  addBookCart(String, Integer): void
                    removeCart(Object):void
                     checkoutCart():void
                      emptyCart():void
```

```
            Controller
          -model:IModel
          -view:IView
```

```
        <<Interface>>                        <<Interface>>
           IView                                IModel
   showPage(Page):void              createItem(T):Status
   addController(IController): void  updateItem(T, T): Status
   showMessage(String):void         deleteItem(T): Status
   showUsers(List(Users)):void      retrieveItem(List(T), boolean):Status
   showBooks(List(Book)):void       createCart(T):Status
   addItemtoCart(Object):void       deleteCart(T):Status
   removeItemFromCart(Object):void  retrieveCart():Status
   emptyCart(): void
```

```
        JFrameView                            Model
   -homePage:HomePage                  -sqlCRUD:SqlCRUD
     -signUp:SignUp                   cart:LinkedList(Object)
     -logIn:LogIn
   -adminPage: AdminPage
    -userPage:UserPage
   -frames: ArrayList(JFrame)
    -controller:IController
```

How to Use The Program

Add the H2 Database Engine JAR

Add the H2 Database Engine JAR (*h2-2.0.202.jar*) to your classpath.
On the other hand, you could also use the "Add External Archives.." option on Eclipse.
1. Right click on the project folder → Build Path → Add External Archives.. → Choose the JAR file → Open

Running the Program

Run the *Driver.java* File. It can be found in the *src/inventory* folder. You will see the Home Page popping up.

Home Page
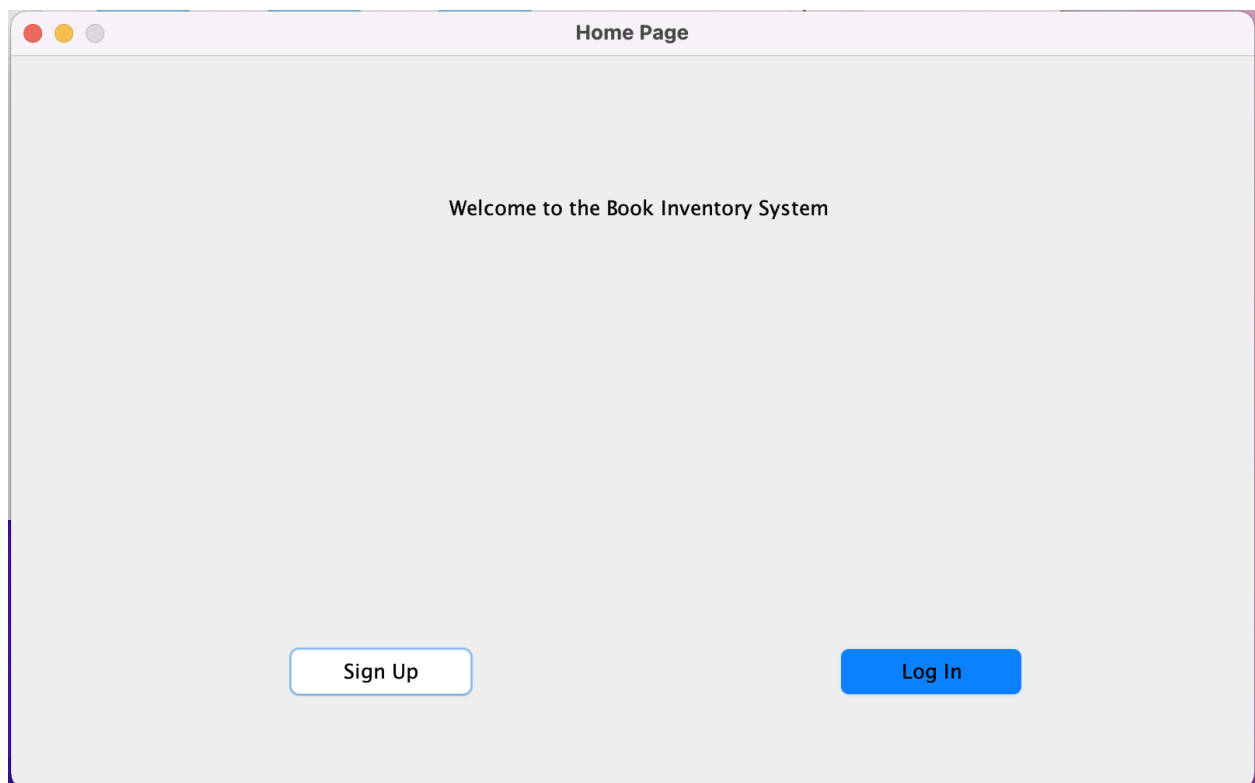
Click on "Sign Up" to sign up for an account. Click on "Log In" to log in to an account.

Sign Up

Sign Up an account. The userName has to be unique or you will fail signing up an account. You can hover your mouse over the Password field to unhide the password. You will be redirected to the Login page upon signing up successfully.

Log In

Log in to your account. Hover your mouse over the Password field to unhide the password you typed in. You will be led to an Admin Page if you were an administrator; else, you will be led to a User Page. You can hover your mouse over the Password field to unhide the password.

User Page

Add To Cart

To Add a book to the cart, double click on any row of the table, then "Add To Cart".

Search

You could also search for the book of interest.

Multiple inputs from the search text field can be separated by a blank space. The first input is used to filter the first table column, the second input for the second table column, and so forth. Multiple inputs/search criteria are chained together using the AND clause.

Inputs/Search criteria with incorrect regular expression syntax is ignored. Besides, the search criteria is set to be case **in**sensitive.

Table Properties and Functionalities

The table **refreshes** itself and the search bar is **reset** to be empty every time a book is added to the cart. The following shows that books "The Pets" and "Sprint" are no longer showing up on the table because they are already added to the cart.

User Page

Bookname

Book Year

Add To Cart

Book Name	Published Year
Change Your Brain	2004
Never Eat Alone	2009
Never Split The Difference	2003
The Good Psychopath Guide to Succ...	2009
Winning Body Language	2017
The Charisma Edge	2012
The 7 Habits	1998
How to lift depression	1999
Not Enough Hours	2007
Applications of NLP	2009
The Sorted Guide	2021

Cart

Log Out
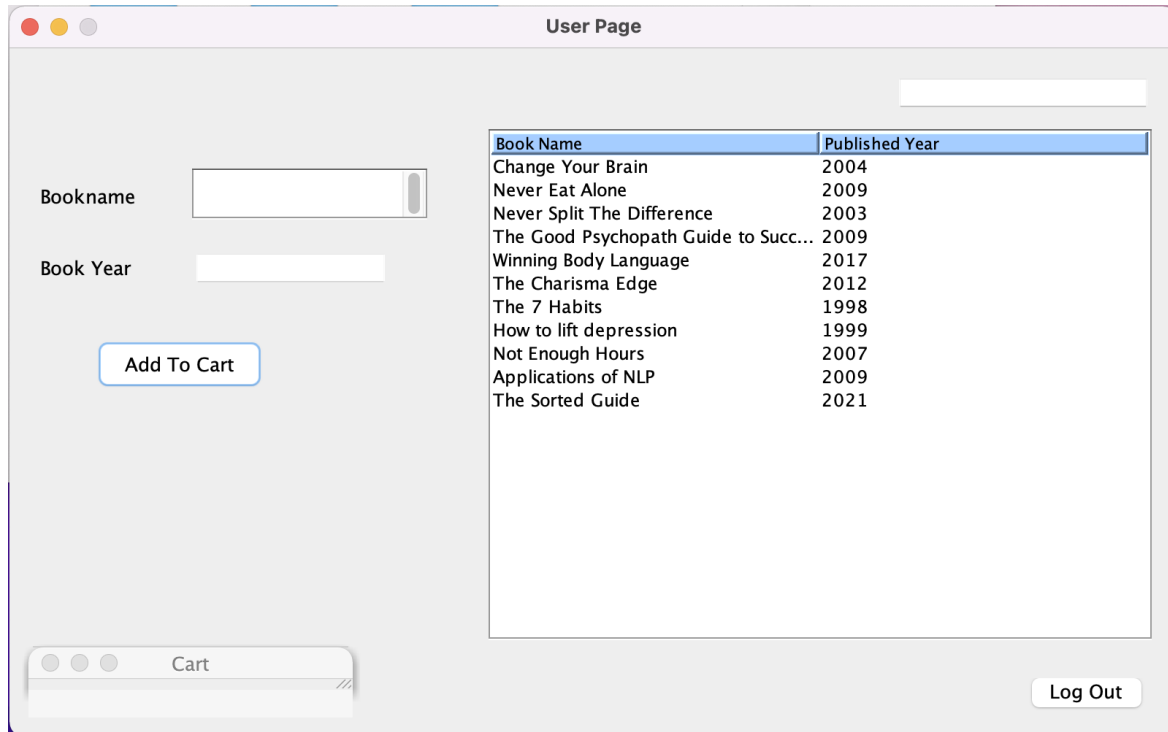
You could click on the table header to sort the table, and adjust the table column size as well.

User Page

Bookname

Book Year

Add To Cart

Book Name ▲	Published Year
Applications of NLP	2009
Change Your Brain	2004
How to lift depression	1999
Never Eat Alone	2009
Never Split The Difference	2003
Not Enough Hours	2007
The 7 Habits	1998
The Charisma Edge	2012
The Good Psychopath Guide to Success	2009
The Sorted Guide	2021
Winning Body Language	2017

Cart

Log Out
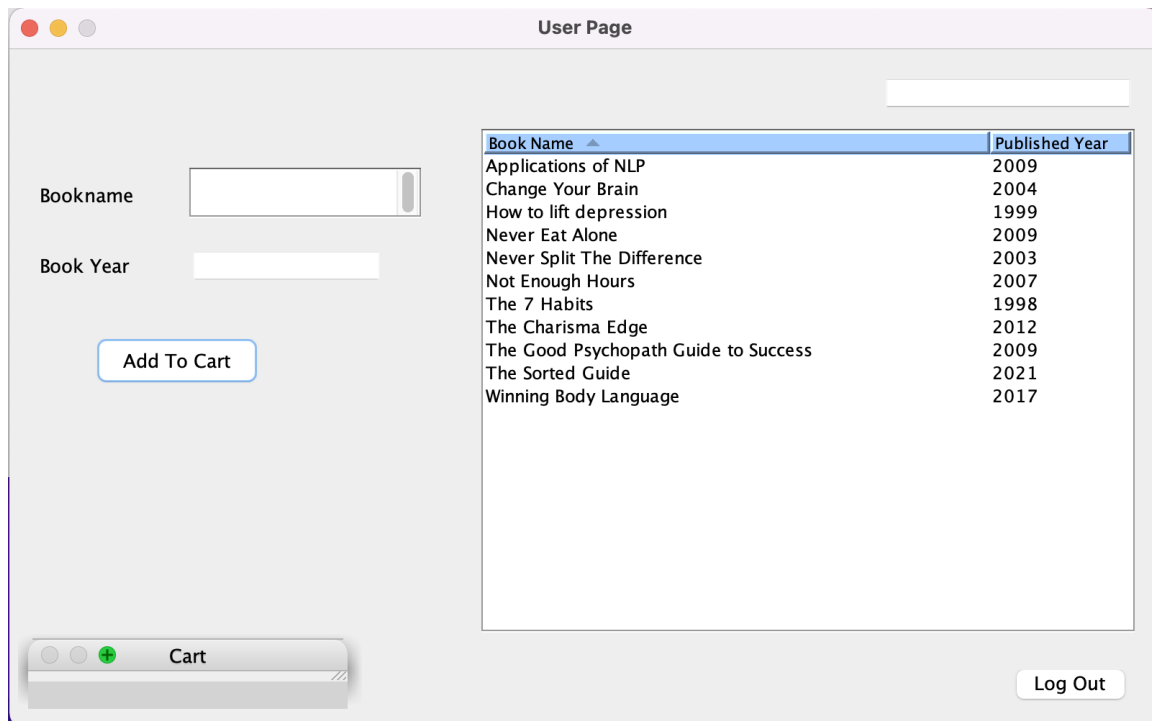
Cart

There is a floating window named **Cart** at the bottom left corner. You can drag and move the Cart window as you wish.

Click on the "+" button to expand/minimize it. Expanding the cart shows that we have the two books added into the cart. You could scroll the cart if there's many items in it.
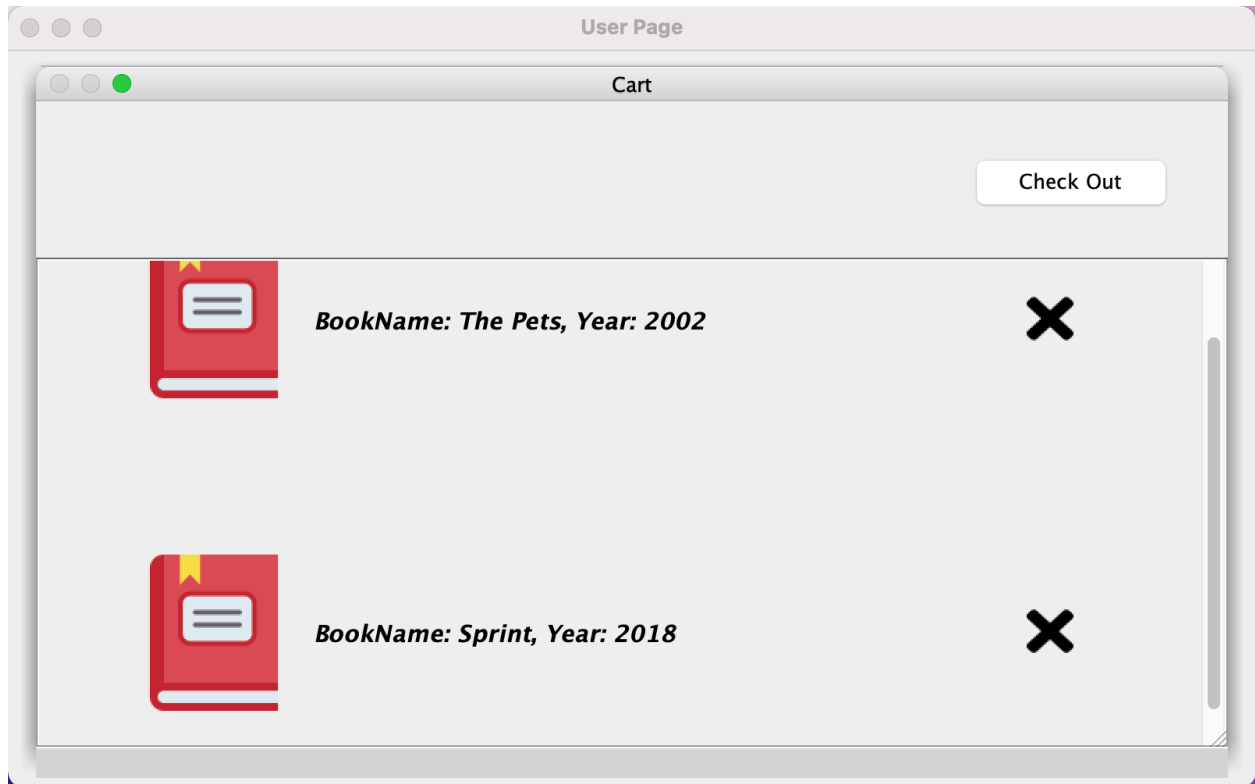
Clicking on the "X" button will remove the item from the cart and add it back to the Book table. Removing all items from the cart will make the cart minimize itself.

Click on the "Check Out" button to check out the current cart. Items that are checked out from the cart are no longer available in the table and the inventory system. The cart will minimize itself if the checkout was successful.
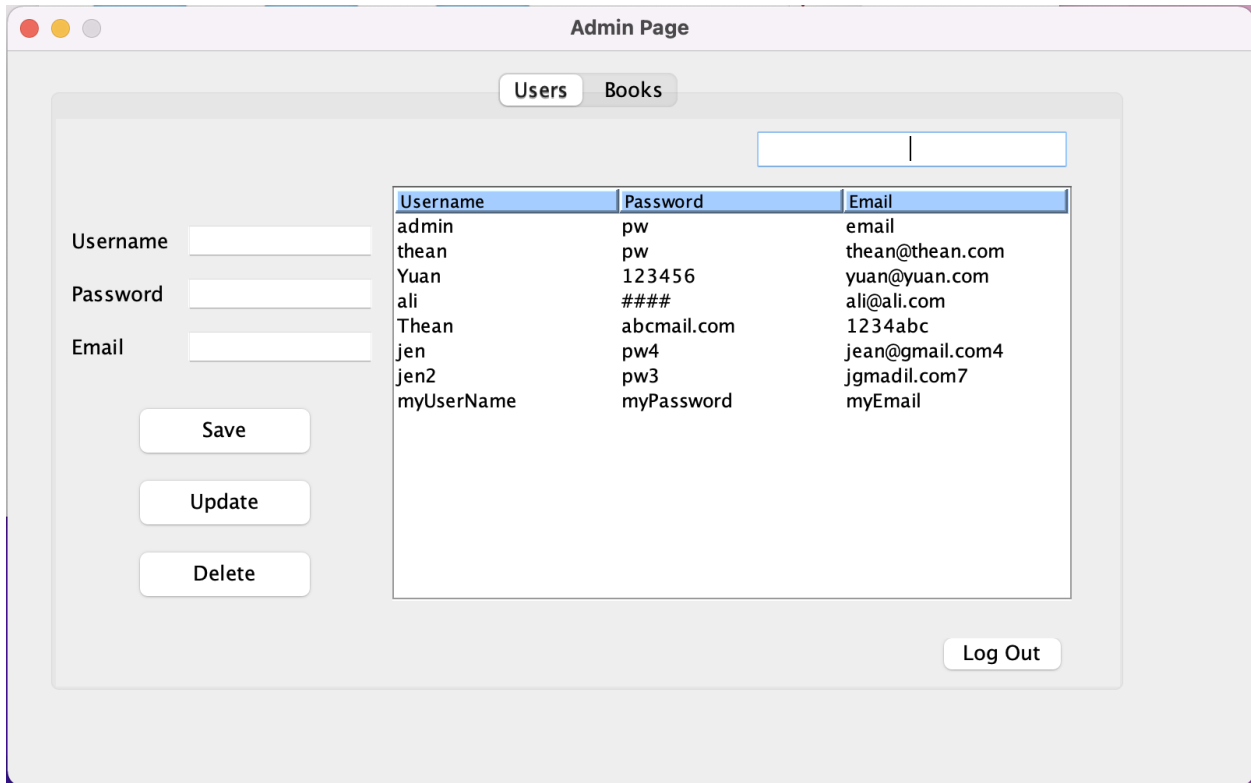
Log Out

Click on the "Log Out" button on the User Page to log out from your account. Logging out will clear the cart (cart is reinitialized per session) and move the Cart floating window back to the lower left corner.

You will be redirected to the Home Page.

Admin Page

By default, there is only one admin for the system with the username and password of "admin" and "pw" respectively.

There are two tabs in the Admin Page: Users and Books.

The User Page has the similar Search functionality as the ones in the User Page.

The admin is allowed to perform the create/save, update and delete (CUD) operations onto the data. Note that you are not allowed to create/save records with the same ID (i.e., same username or book name) and not allowed to update records that do not exist.

CUD using Text Fields

You could click on a row on the table to select a user/book, and populate them to the text fields on the right. You could then make changes to the text fields and perform the CUD operations. The table will refresh itself once an operation is performed.

Update using the Table

The table has the similar sorting, resizing column and searching functionalities as the one in the [User Page](). In addition, it also allows the admin to update records on the table straight away. Note that the ID columns (i.e.,the UserName and Book Name columns) are not editable.
To edit, double click on the table cell of interest, make changes onto it, and press enter.

Once that is done, you will only be able to press on the Update button.

You are allowed to update multiple cells at once before updating all the records in a batch. The table will refresh itself once the update operation is performed.

Log Out

Changes made on the table will be discarded if you log out before making updates. You will be redirected to the [Home Page](#).

Program Testing

1) Test the product in 2 main OS (Windows and Linux) using different machines.
    a) Pass
2) Test the Signup/login page to make sure that only the correct username/password combination is allowed.
    a) Pass. The system notifies the user if errors occurred.
3) Test the Search/View Page -- making sure the correct products show up. Do the similar thing to the Cart page.
    a) Pass for the UserPage and AdminPage.
    b) Didn't implement the search functionality on the Cart page so unable to test.
4) Test whether the inventory counts still tie up properly after a Product is checked out.
    a) Pass
5) Test if it stops users from checking out once the Product of interest is no longer available.
    a) Pass. The system notifies the user about the failure and asks the user to remove the item from the cart manually.
6) Make sure that normal, non-admin users are not allowed to interact directly (such as deleting/ creating/updating) with the Products inventory.
    a) Pass
7) Load lots of Product to the system. Would the system become too slow?
    a) Pass. Loaded 10000 books to the system and the system wasn't lagged.