

# WAREHOUSE MANAGEMENT SYSTEM



TEAM ABC

Shannon, Lauren, Jaden, Dan, David & Cayleigh

# TEAM DESCRIPTIONS

## Shannon Maier

- a. Freshman from Atlantic Beach, New York.
- b. I am a Lifeguard during the summers
- c. I am a Computer Science major with a focus in Software Engineering and a Fashion Merchandising Minor.
- d. Member of the Women's Lacrosse Team.

## Lauren Lietzke

- e. From Austin, Texas
- f. Two dogs named Moose and Stormi
- g. I play volleyball
- h. Psychology major with a minor in data science and analytics

# TEAM DESCRIPTIONS CONT.

## Jaden Reasor

- a. Sophomore from Fairfax, VA
- b. I love spending time with my family and dog
- c. Applied Mathematics major with a Computer Science minor
- d. I play volleyball

## Dan Aulbach

- e. Sophomore from Brookline, NH
- f. I have two older brothers and a dog
- g. Communication- Double Concentrated Sports and Journalism with data science minor
- h. Associate Editor For Center Field, Student Sports Publication

# TEAM DESCRIPTIONS CONT.

## David Penafiel

- a. Freshman from Marlboro, New York
- b. I'm majoring in Computer Science
- c. I was born in Ecuador
- d. I played soccer for the USA Youth National Team

## Cayleigh Goberman

- e. Freshman from East Granby, Connecticut
- f. I have a younger sister
- g. My family has a hound dog named Strawberry, two parakeets named Neo and Beetle, and two horses named Red and Parker
- h. Computer Science Major with a concentration in Game Design and Programming

# PROJECT DESCRIPTION

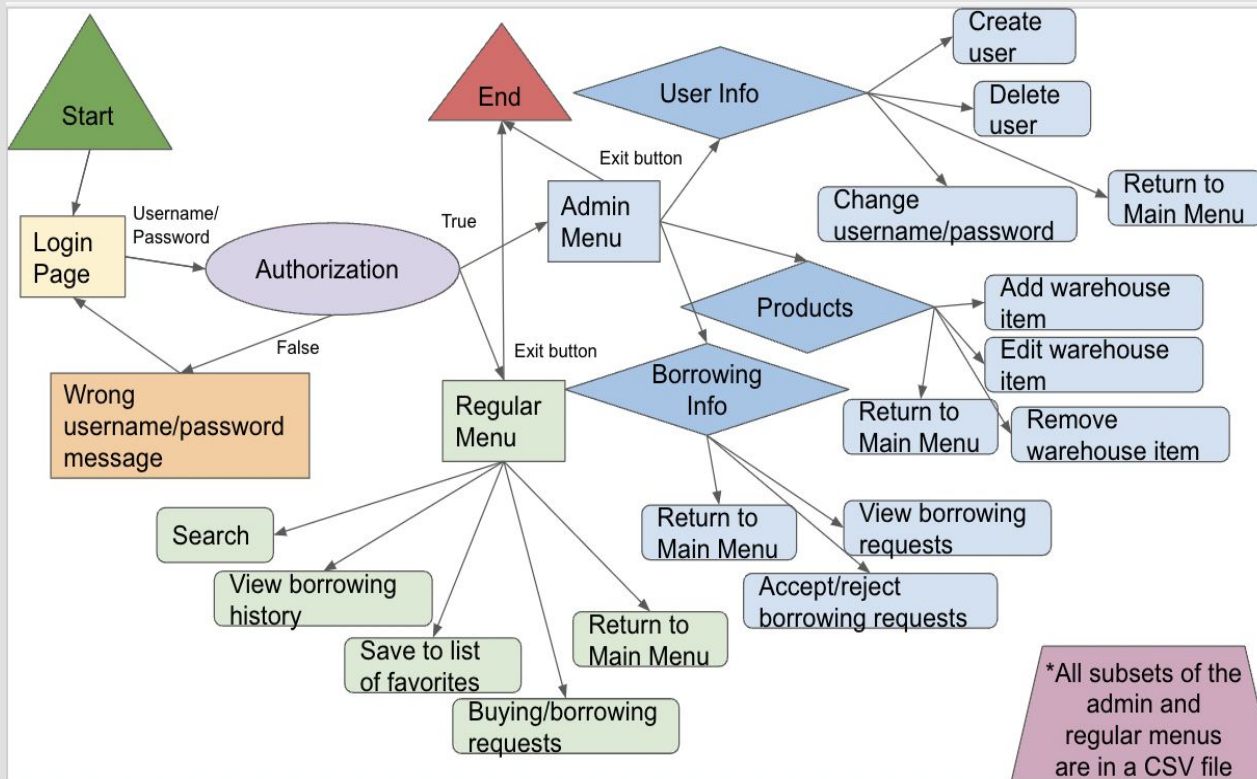
**Summary:** Warehouse management system (WMS) provides an organized way of storing different products and elements in a warehouse. You can consider a library as a warehouse, which maintains books' details and user libraries. A general WMS stores details of name and identification number of products, their store time, the required storage condition, price, weight, height, etc. following this, this system allows guest users to search for different content and request to borrow/buy them.

1. Admin user is capable of:
  - a. Having admin user and password for log in (a string of at least 8 characters)
  - b. Changing the admin user and admin password
  - c. Adding a guest user to WMS by creating a new username and password. a guest user is not able to define or remove other users.
  - d. Removing users from WMS by removing their username, password, and corresponding recorded data.
  - e. Adding an item to the warehouse with varied details, such as:
    - i. Type: food, books, cars, etc.
    - ii. Stored time in the warehouse
    - iii. Pick out time from the warehouse
    - iv. ID: each item in your library should have a unique identification number with a specific format
    - v. Name
    - vi. Provider/creator's name
    - vii. Quantities: the number of available items. For instance, item x with a quantity of 2 is a sign of 2 available x items in your warehouse.
    - viii. Place: where the item is stored
    - ix. Price
  - f. Deleting an item from the warehouse
  - g. Editing an item in the warehouse
  - h. Viewing the list of borrowing requests
  - i. Accepting or rejecting a borrowing request

# PROJECT DESCRIPTION CONT.

1. Each user should be able to:
  - a. Search through WMS based on all items' details, such as id, name, and producer.
  - b. Save a list of favorite items
  - c. Request to borrow/buy some items for a specific time. For example, borrowing an item for 3 weeks.
  - d. View the history of borrowed items
2. WMS should be a user-friendly software, such that:
  - a. It shows a welcome page
  - b. It provides a menu of all functions to the user on all pages
  - c. It illustrates the reports in a tabular form. For instance, it displays a well-organized list of the requested items.
  - d. WMS should provide an exit function and thank the user for using this software.
  - e. It shows a warning if:
    - i. The admin user tries to add a new item to the library with an existing ID.
    - ii. If a guest user tries to borrow more than 3 items.
    - iii. A user search request returns null items.

# GRAPHICAL USER INTERFACE



- Warehouse Management Systems Flowchart
- Starts with Login Page
- Ability to return to main menus
- All information goes into CSV files

# GRAPHICAL USER INTERFACE DESIGN



# LOGIN PAGE

The image shows a simple login page layout. At the top center is the title 'Login Page'. Below it, there are two input fields: one labeled 'Username' on the left and one labeled 'Password' on the right. Both fields are represented by light gray rectangles. Below these fields, centered horizontally, is a blue rectangular button with the text 'Login' in white.

- The login page allows the user to access the program by putting in a username and password and clicking the login button.
- The user can access either the Admin Main Menu or the User Main menu.

# ADMIN MAIN MENU

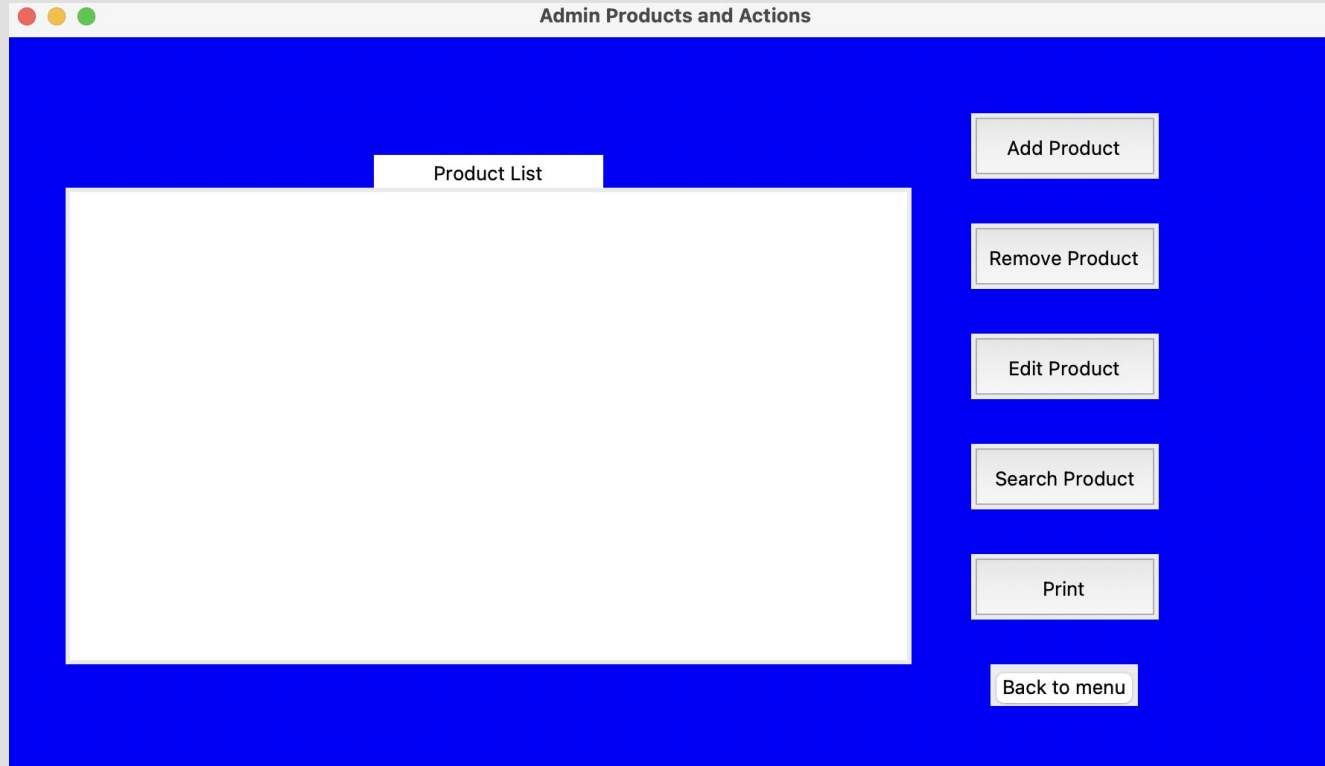


Mainly a page to navigate to other Admin pages

## Buttons:

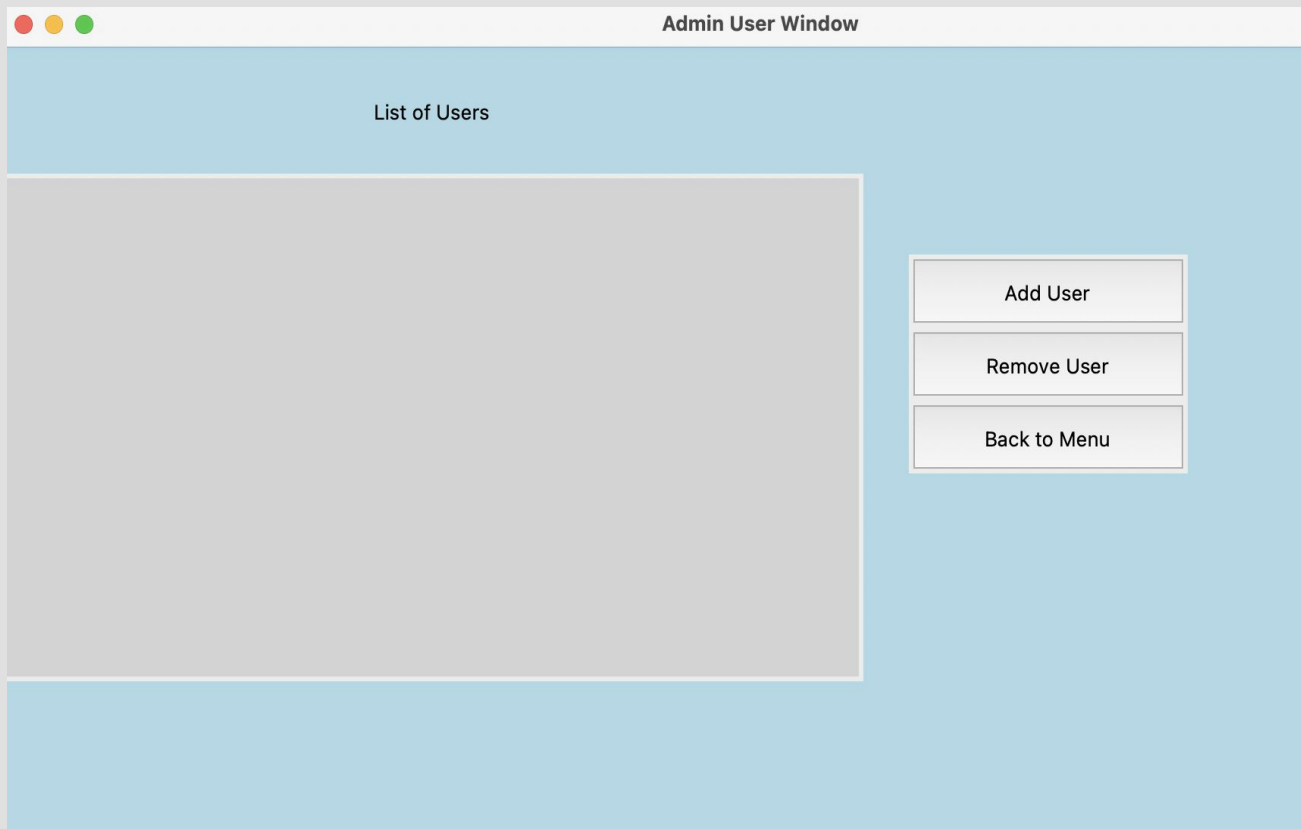
- Navigate to a list of products and actions the admin can perform with products
- Navigate to a page to view user requests to borrow products
- Navigate to a page to view list of users and user actions
- Navigate to a page to change to Admin username and password
- Log out of the Admin Account

# ADMIN PRODUCT PAGE



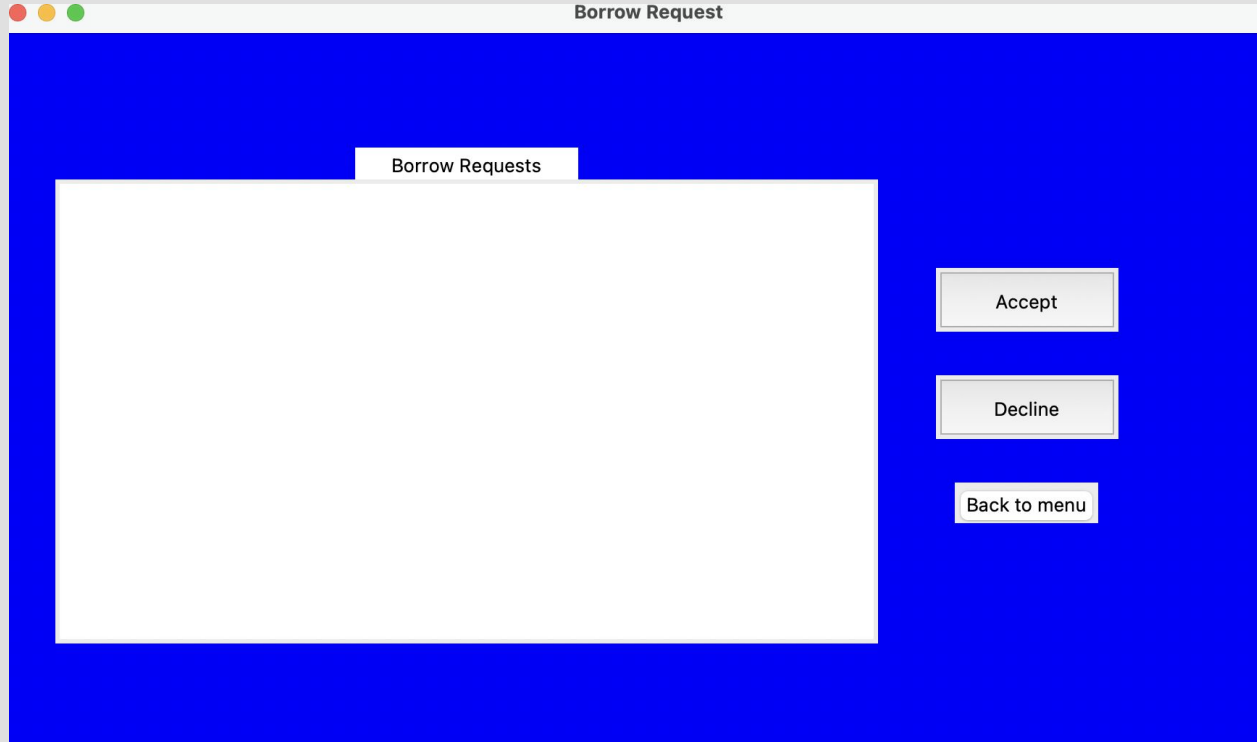
- Page where products can get added, deleted, edited, and search

# ADMIN USER WINDOW



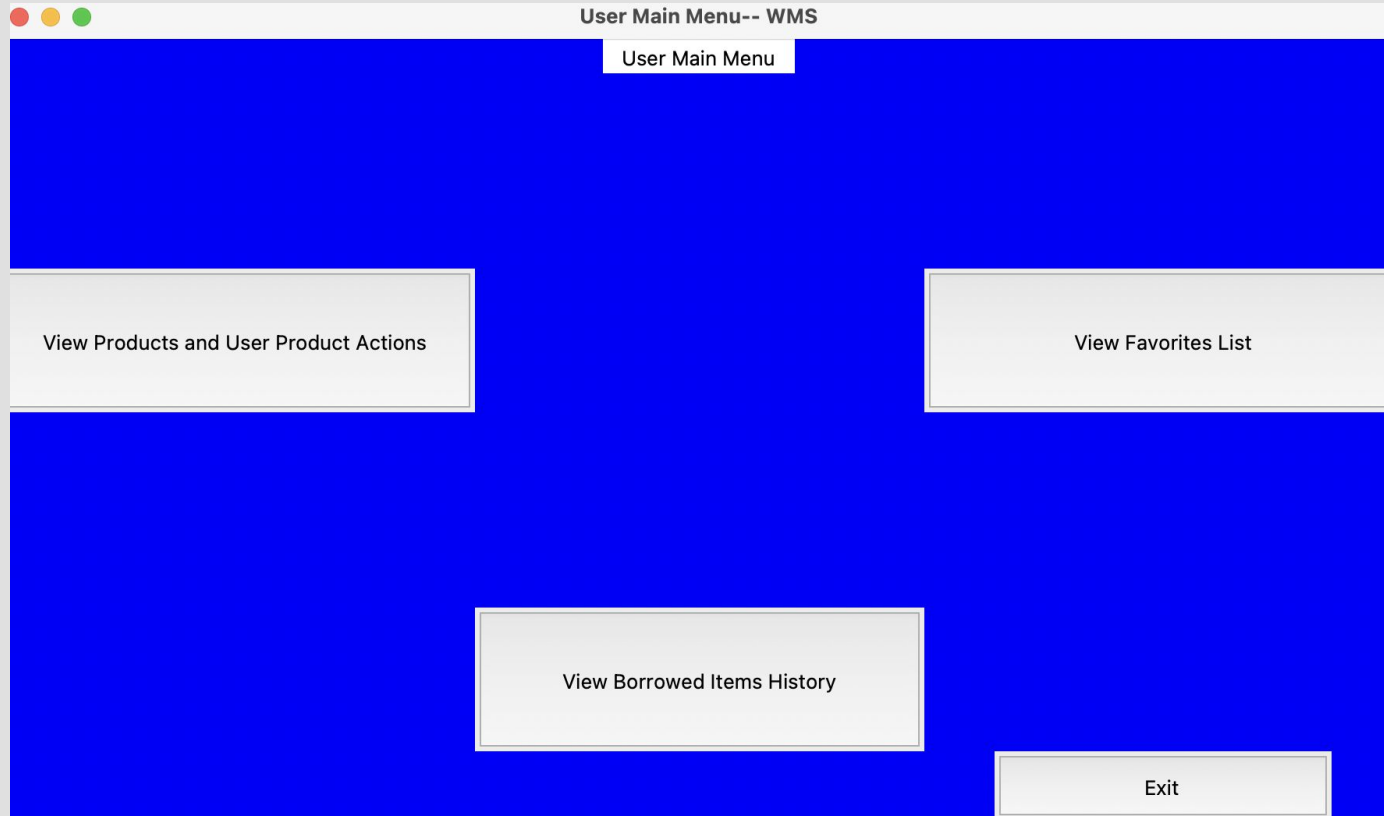
- This Page allows the Admin to add a user, remove a user, and have complete access to go back to the main menu.

# ADMIN BORROW REQUEST WINDOW



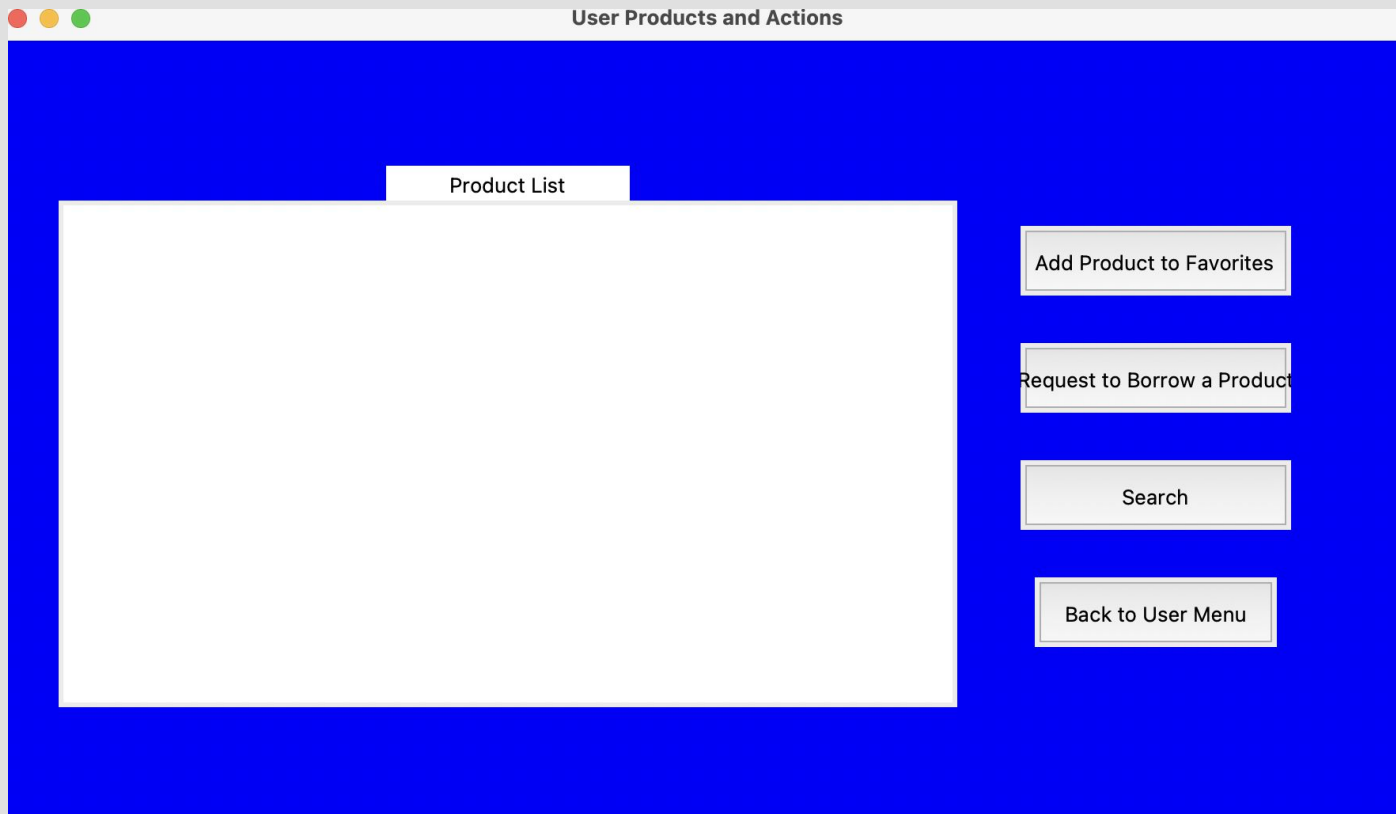
- Administrator can accept or decline any borrow request made by a user
- Two pop-up windows to do this

# USER MAIN MENU



- Navigates User to Three possible menus, all stored in CSV files and read with CSV "r" function, each with pop up windows.
- Tkinter Specifics: Blue Frame, Four buttons(btn), Window Title, Pack()
- If Statement(s)(Checks for CSV file)

# USER PRODUCT PAGE



- Opens pop up windows to perform functions
- Favoriting items and borrowing requests stored in csv files

# DATA STORAGE

```
with open('users.csv', "a", newline = '') as file:  
    csvWriter = csv.writer(file)  
    csvWriter.writerow([name,username,password])
```

```
with open('borrowrequest.csv', 'w') as file:  
    writer = csv.writer(file)  
    writer.writerows(lines)
```

```
with open('AdminLogin.csv', "w",) as file:  
    csvWriter = csv.writer(file)  
    csvWriter.writerow([AdminUser, AdminPass])
```

```
with open ('accepts.csv', "a", newline = "") as file:  
    csvWriter = csv.writer(file)  
    csvWriter.writerow([name, ID, length])
```

```
with open('warehouse.csv', 'w') as file:  
    writer = csv.writer(file)  
    writer.writerows(lines)
```

```
with open ('favorites.csv', "a", newline = "") as file:  
    csvWriter = csv.writer(file)  
    csvWriter.writerow([row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7]])  
    texFavorite.insert(tk.END, row[0]+ " - " + row[1] + " - " + row[2] + " - " + row[3] + " - "  
        + row[4] + " - " + row[5] + " - " + row[6] + " - " + row[7] + '\n')
```