# Cart System

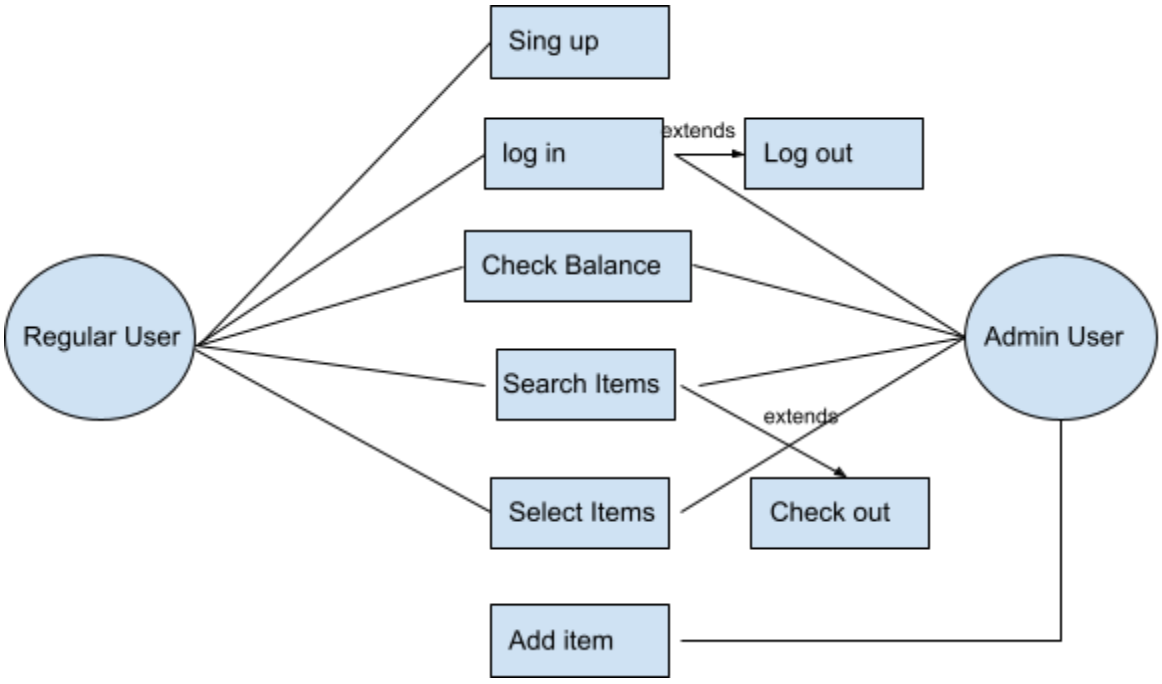
## Introduction

This project involves implementing a simple Cart System in Spring Boot. The system allows users to purchase items and check their previous orders. The system supports two classes of users i.e regular user and admin user. A regular user can sign up and is immediately provided with an account with a balance of $ 100. They can then continue to browse the available items and proceed to add them to a basket/trolley and finally they can purchase the selected items. In addition to these actions, an Admin user can also add items into the system. Adding an item will require the Admin to provide the name of the item, its category, the quantity, link to an image of the item and the price of the item. Note that a regular user does not have permission to add an item.

In this spring boot application, the system does not support authentication or implements a database for the users and the items. The usernames and passwords as well as the items are stored as static variables in the system. A user is implemented as a class and has the following attributes; username, password, list of items in the basket and list of past orders. An item is also implemented as a class and contains the following attributes; item name, item category, price, quantity and link to an image. Below is a use case diagram for the users.

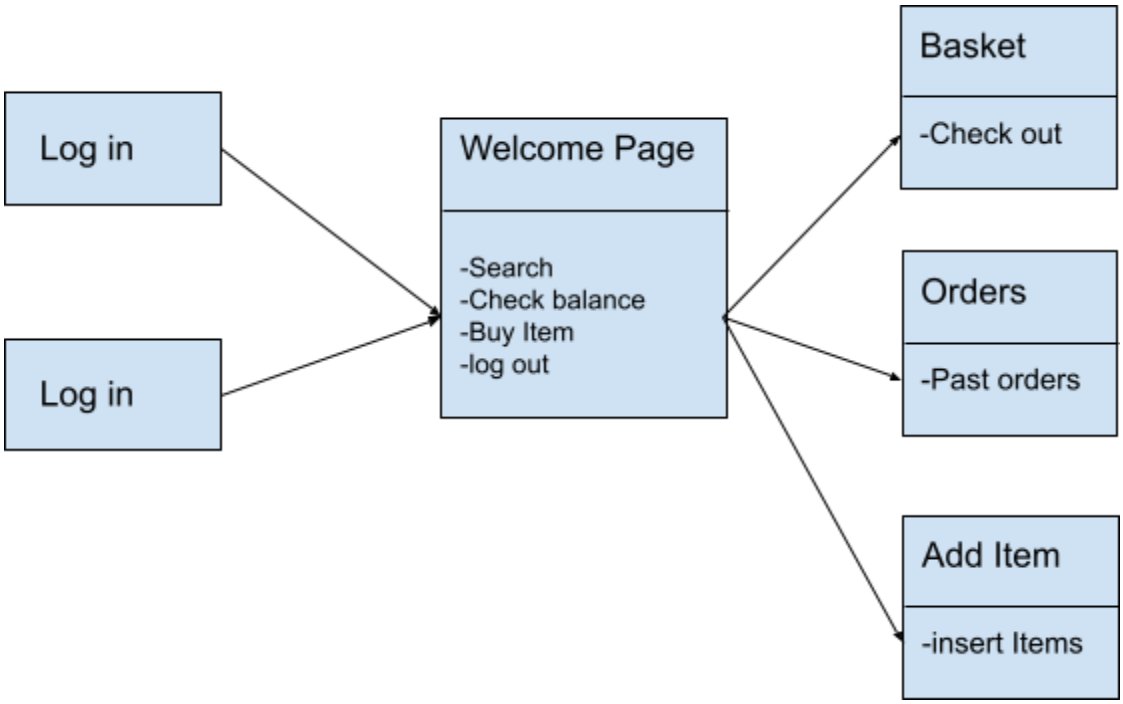
## Use Cases

Below is an image of the use cases for both regular user and admin user.



## Sequence Diagram

Below is a sequence diagram that describes the flow of the system.




## System Analysis

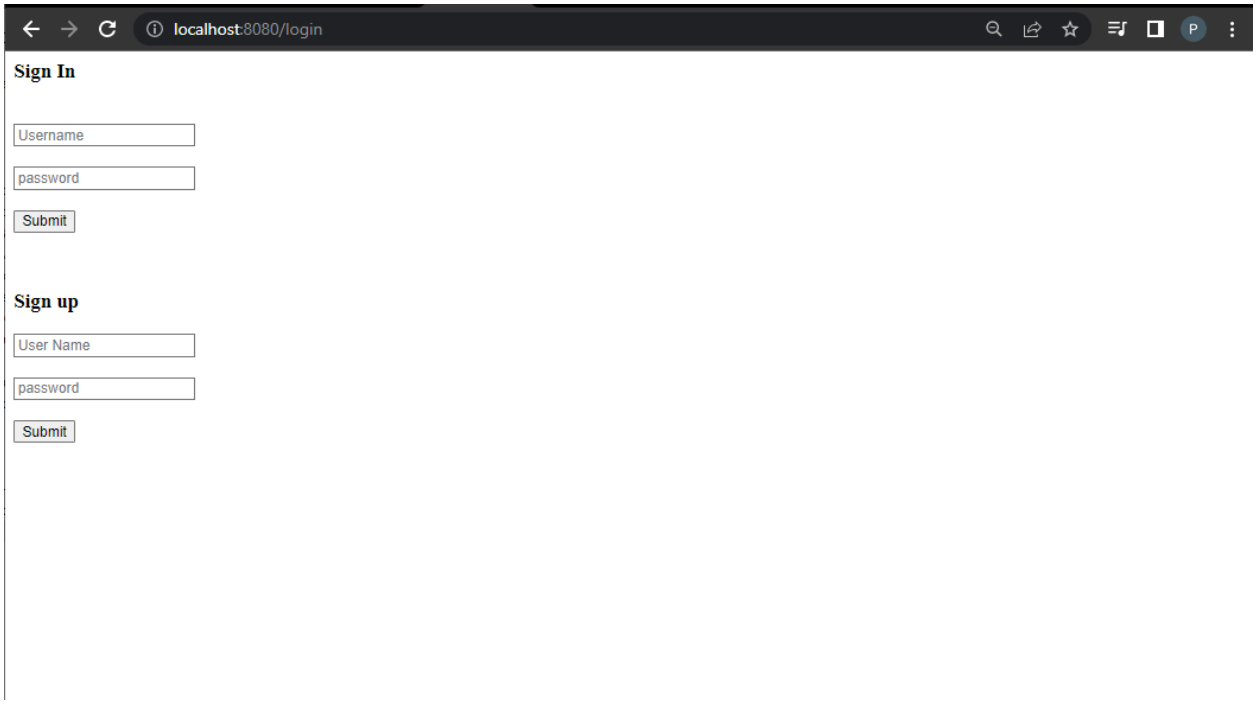
In this section, the various parts of the systems will be explored. Description will be provided for what actions can be performed for each stage or page of the application.

### 1. Login and Sign up Page

From the root of the application a user is delivered to a sign in and login page. In this page, an existing user can sign in while a new user can create an account. To sign in, the user provides a username and password. The system checks if a user with the provided username exists in the system and then proceeds to match the provided password with the saved password. If the username does not exist or the password does not match the password in the system, the app redirects the user to the same login page and provides a message indicating that the username or password is incorrect. Otherwise if the password and username checks out, the user is delivered to the welcome page.

For signing up, the user is only required to provide a username and password. If the username is already taken i.e there exists a user with the same username, the user is redirected to the same sign up page and an error message indicating that the username has been taken is provided. Otherwise a new user is created and provided with an initial balance of $100 and the user is directed to the welcome page.

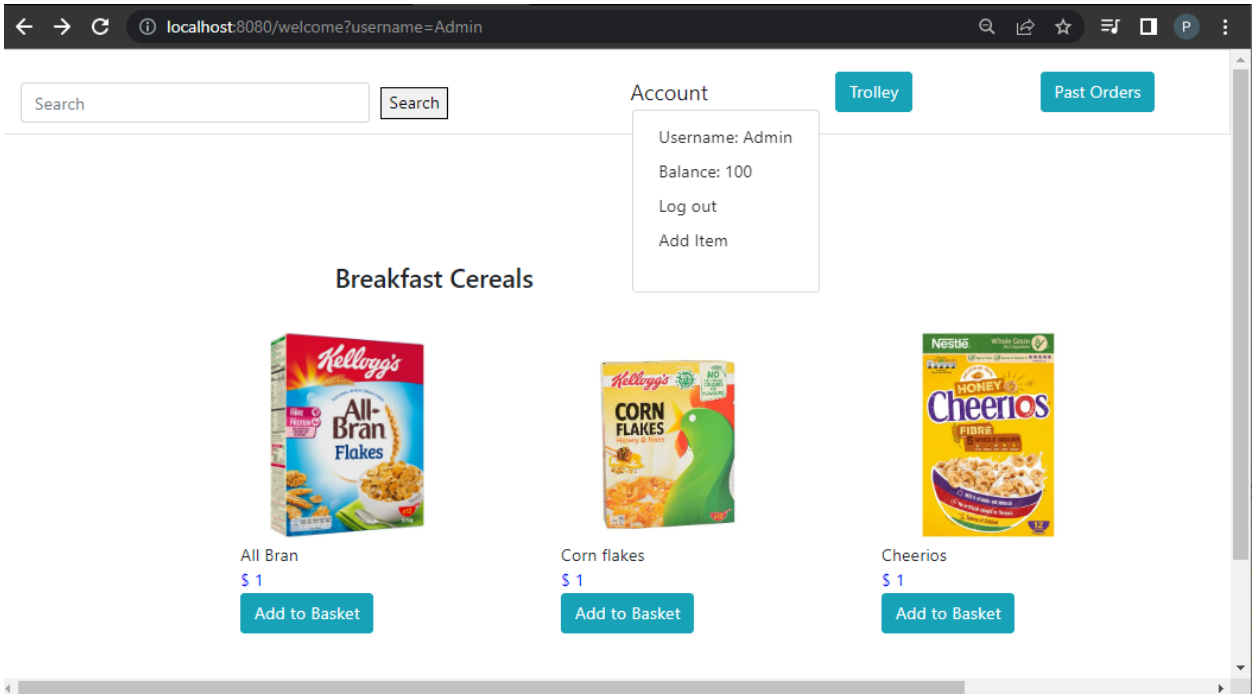
Below is an image of the sign in and sign up page.




**2. Welcome Page**

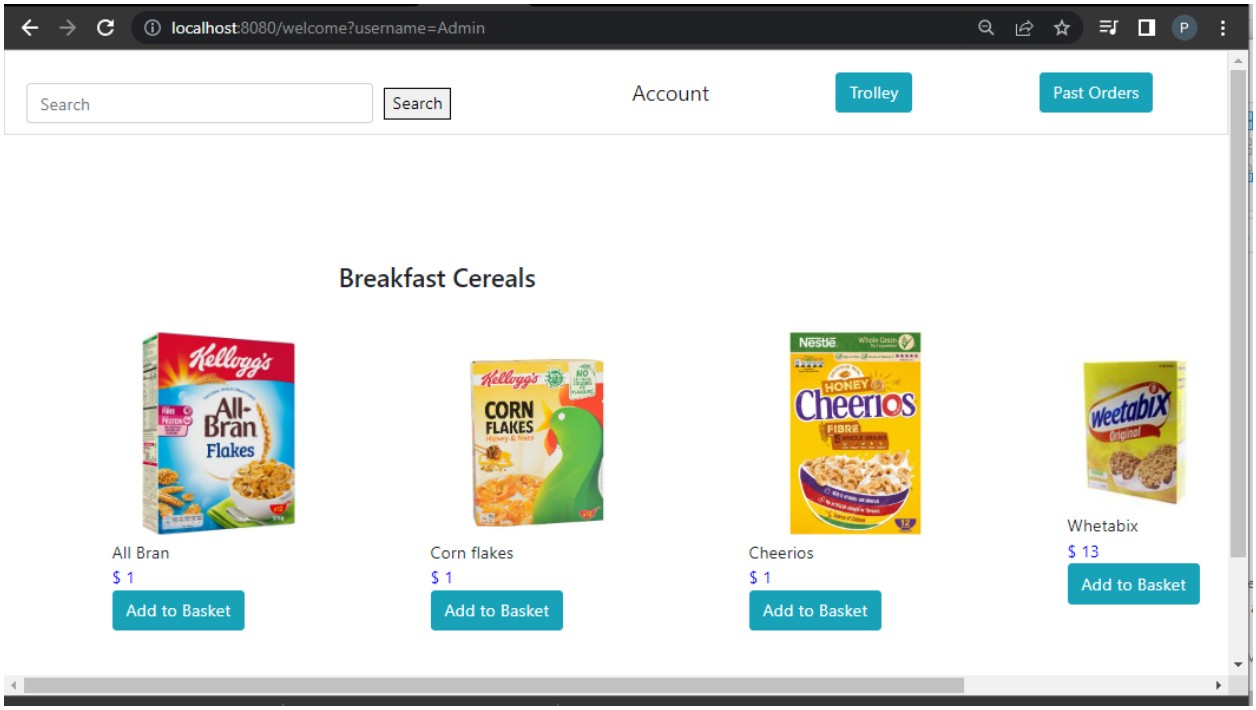
The welcome page provides most of the functionality of the system. The body of the page lays out all the available items with their names and prices. In the nav bar, there is an option to search for items. Users can either enter the name of the item or the category of the item. Once the search button is clicked, the system selects all the items that either have the same category or name provided by the user. The page is then refreshed and the searched items are displayed.

The nav bar also contains an account option. This is a tag that implements a drop down. The drop down list provides the username of the current user, their balance and option to log out. In addition, for admin users, there is an option to add items. When the admin user selects the add item option, a modal window is launched that allows the admin user to enter the specifics of the item they would like to add. Once the item is added, the page is refreshed and the new item is displayed in the main body. Below is an image of the nav bar and account options.




The nav bar also contains a trolley and past orders button. These buttons will lead the user to a page displaying items in the basket and past orders accordingly.

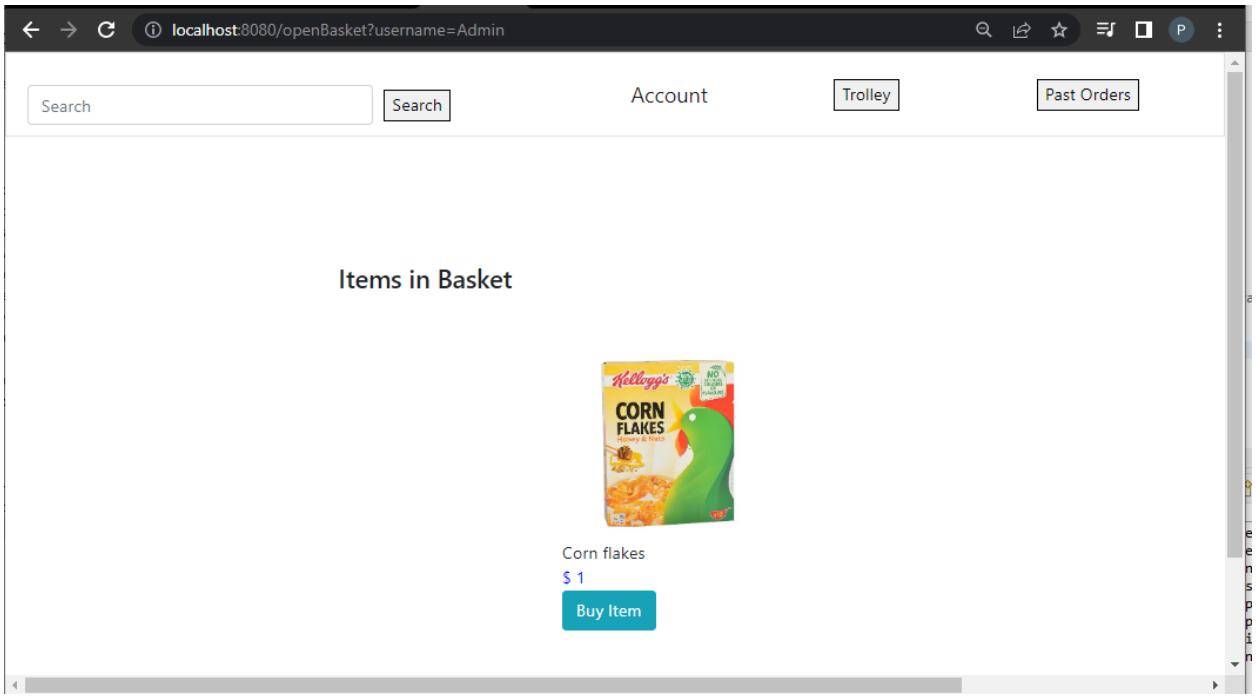
In the main body, the items are displayed with their images, price and name of the item. They are also accompanied with a buy item button. When a user clicks on this button, the item is added to the basket.
Below is an image of the welcome page.




**3. Trolley/Basket page**

This page displays all the items in the users basket. For each item the user is provided with an option to buy the item. Once the user buys the item, it is removed from the basket and added to the list of past orders. The user account balance is also updated accordingly to the price of the item bought. The page also refreshes to reflect the changes.

Below is an image of the basket page.



**4. Past orders page**
This is similar to the basket page with the exception of the buy item button. The page simply displays all the items the user has bought in the past.

## Conclusion

In conclusion, Spring Boot framework has proven quite efficient in developing this system. A major problem during this project was setting up the environment for spring boot to compile and run. Biggest issue was ensuring all the dependencies have been included appropriately. However once the application was running, adding the various components went about much quickly. It would be interesting to learn how to implement the database and various for a production type system.