HW 2

Authors: Claire Rhoda, Genna Quach, Jeremy Escobar, Juan Uriarte and Ricardo Gonzalez.

Design concept: Spring 2022

Table of Contents

1.1 - Executive Summary -

The goal of The Grocery Experience is to help automate the process of checking out so the consumer is able to pick and choose exactly what they want and obtain a total of how much the final cost would be. The goals of The Grocery Experience are:

- To provide a searchable database of all items at a store.
- To provide ease of access to consumers.
- To minimize the amount of time spent calculating.
- To have simplicity that all consumers are able to understand.

1.2 - Requirements (Use cases and UML Class diagram should have all items tied back to a requirement)

The Grocery Experience had the following general requirements:

- A way in which the customer can pick from different categories.
- A way in which the customer can add different items in different quantities to the same shopping cart.
- A way in which the items have an image and a name the customers can pick by.
- A way to search for an available item.
- A way the customer can remove items from the shopping cart.
- A way the customer can view the items in the cart.
- A way the customer can view the total costs and amount of taxes of items selected.

- **Interface design**

  The interface is designed to have the best usability of the product for the customer.

- **Database**

  The database is pre-registered with items in different categories with registered prices and weights.

- **Accessibility**

  The program is designed to not require training and to be easily operated by the customer, delivering the same functionality every time.

2.1 - Use Cases (step-by-step instructions, start including exceptions, not all required yet)

The primary actor is the customer.

The goal of the customer is to be able to pick items by the four available categories, choose the number of specific items, and have the total cost of all items for a transaction to take place.

The precondition of the customer is to have sufficient money for a transaction to take place and to know how to operate their site with a minimal thought process and customer friendly experience.

The main tasks performed by the customer is a welcome screen where they would have to acknowledge by clicking on the screen. After this the customer will be able to choose from the four categories which are food, vegetables, snacks, and miscellaneous. Once the category is chosen, the customer will be able to choose the items being represented with pictures that match the description and the weight of the item. Once the item is chosen, they can choose the number of that specific item they would like to purchase on the shopping list and have the freedom of returning to the main menu where the category list is located.

The extensions that are being considered are a user ID for the customer, implementation of coupons, and a survey that would represent the customer service to the user.

The site will have no variations in the customer's interaction since it's a site that will be easy to use and have the same customer service to everyone despite the number of transactions.

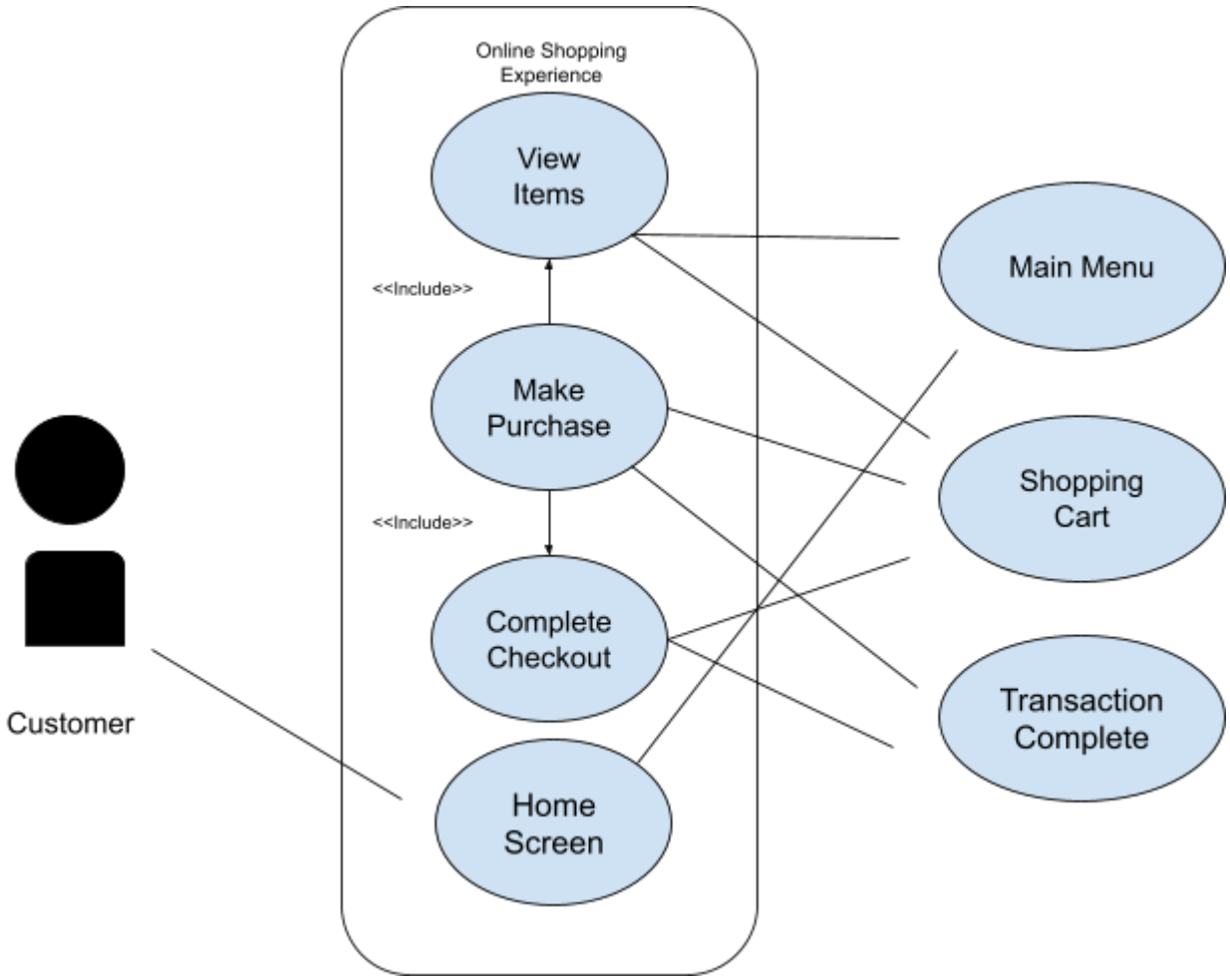
The system information the customer will acquire is a list of items located in the shopping cart that will contain the total number of items, total amount, and the taxes associated with the location of the customer and state.

The customer would not be able to inform changes in the external environment.

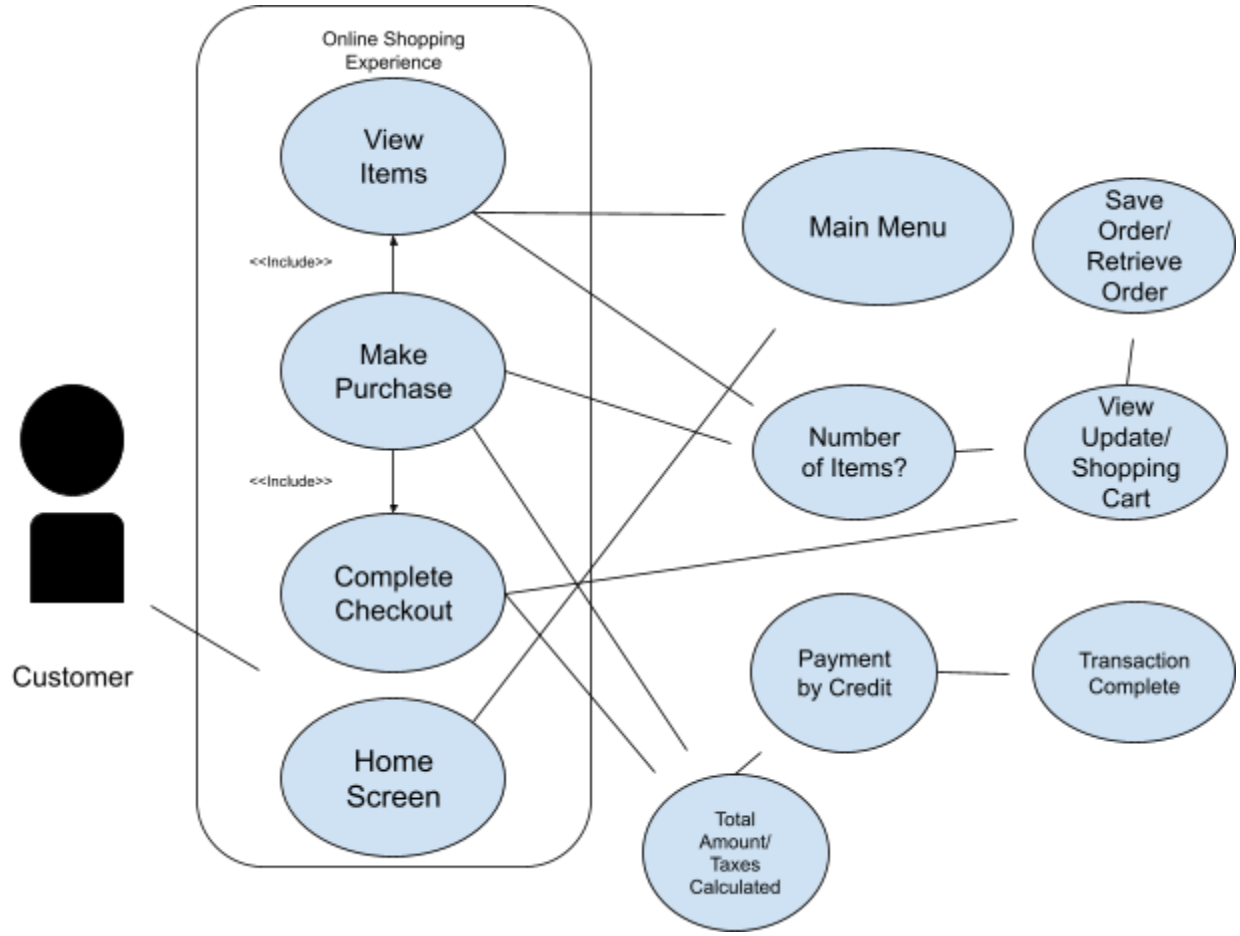
The information that is desirable to the customer is the total amount of the transaction, the number of items, and the list of items that can be purchased.

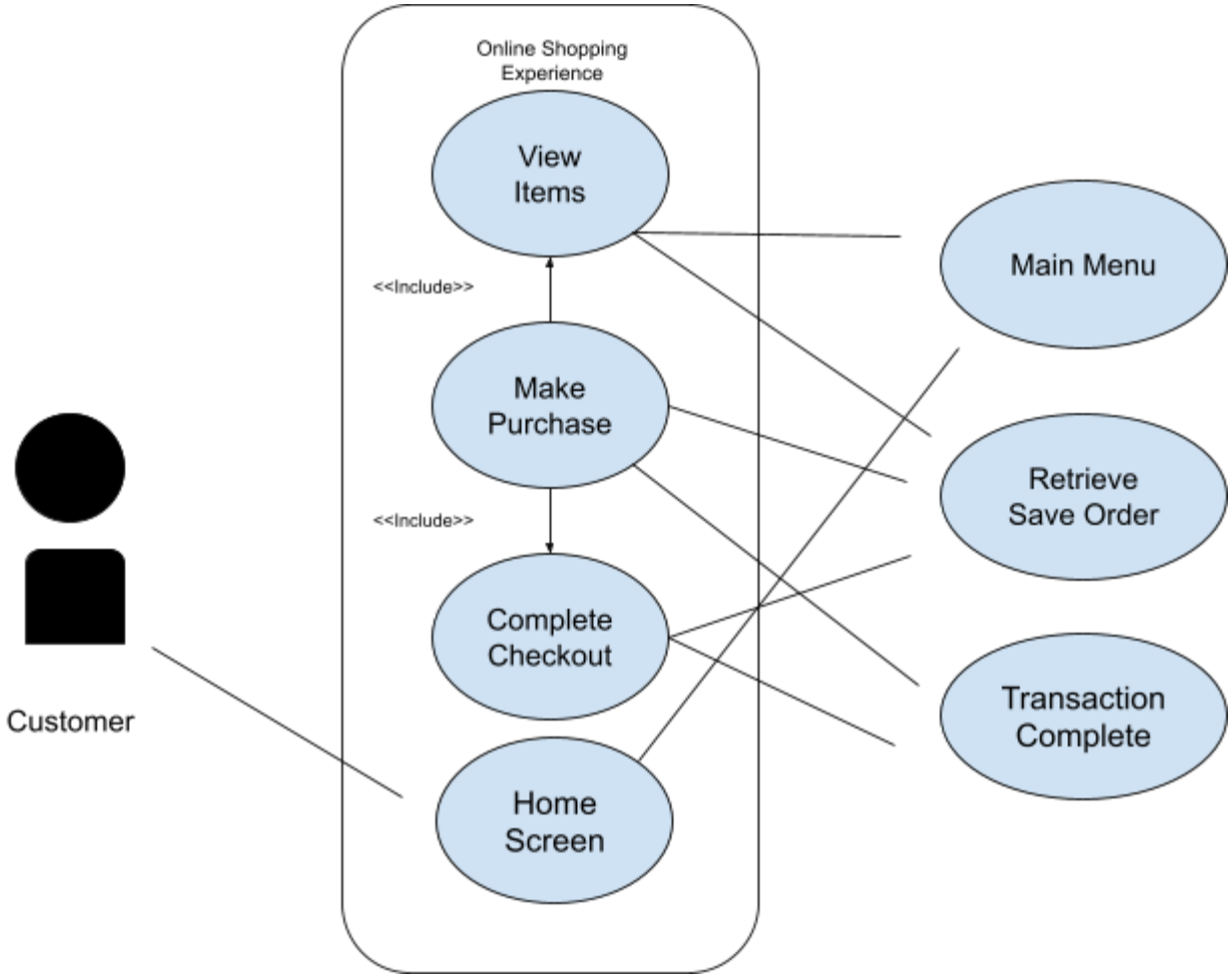
The customer will be informed of unexpected changes because our goal is to have the best customer experience and have them in control of the future expectations they would like the site to be in.

2.2 - Use Case Diagram

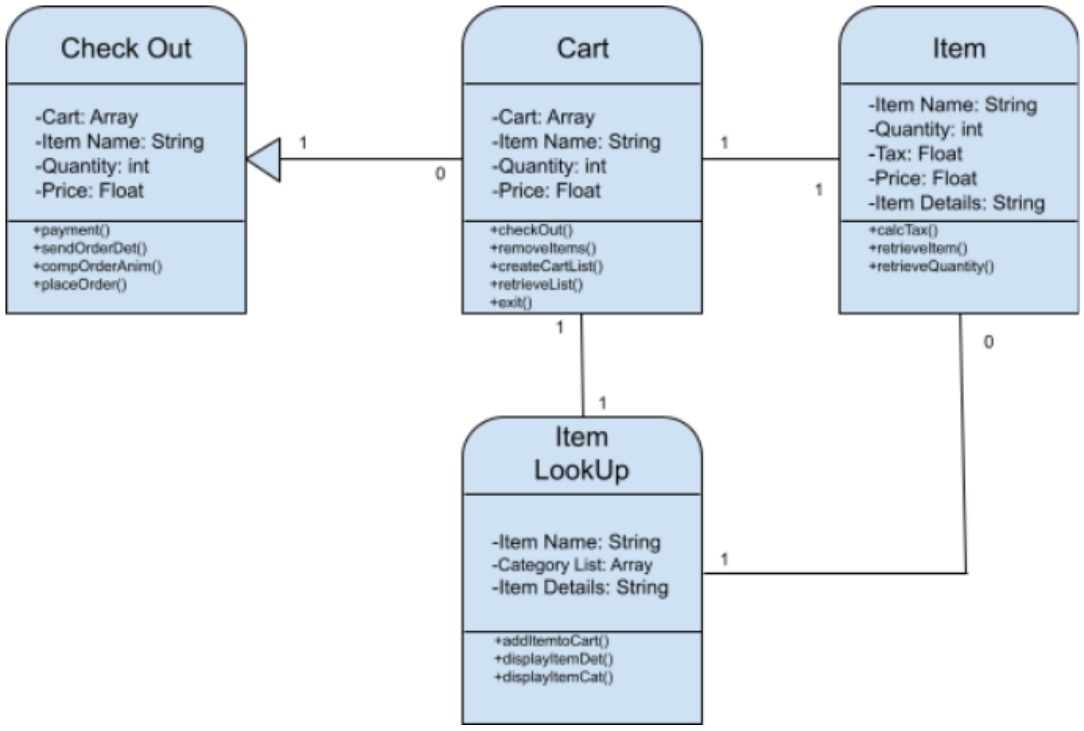


2.3 - Activity Diagram (at least 2 use cases, aim for biggest/most frequent)




Online Shopping Experience
View Items
<<Include>>
Make Purchase
<<Include>>
Complete Checkout
Home Screen
Main Menu
Retrieve Save Order
Transaction Complete
Customer

2.4 - UML Class Diagram with connections , relationship multiplicity, and attributes



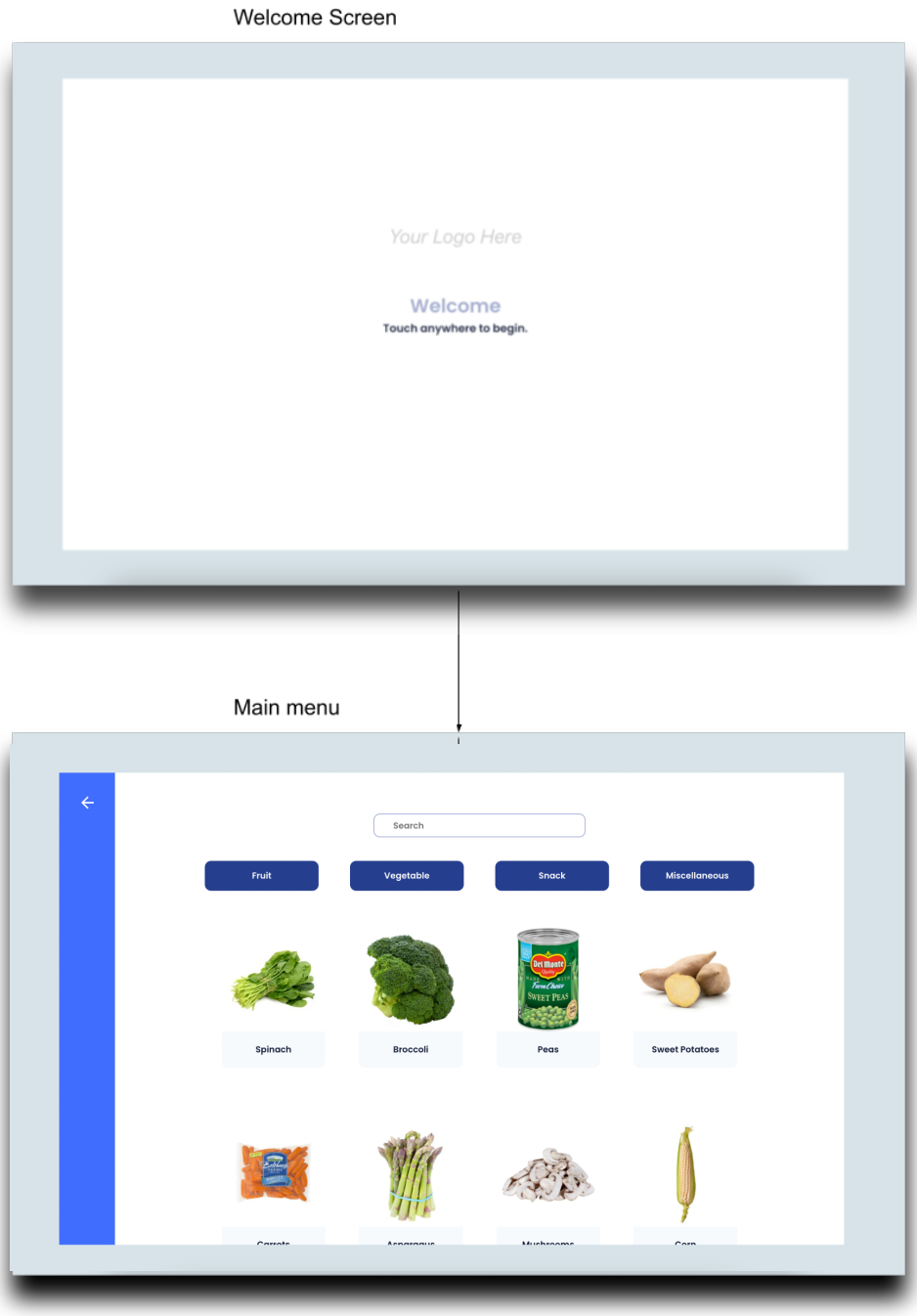
2.5 - Operating environment and dependencies (what software/components are you using?

The operating environment of the project is on the web. It is a web application that utilizes html, css, and javascript. It utilizes React, a javascript library, to both create a clean U.I. for the user to interact with and the backend that allows easy access to the inventory. The web application was actively made on Create to help visualize the changes made. The service will be hosted on AWS using a S3 bucket and will contain a C++ backend in order to handle the user login and authentication.

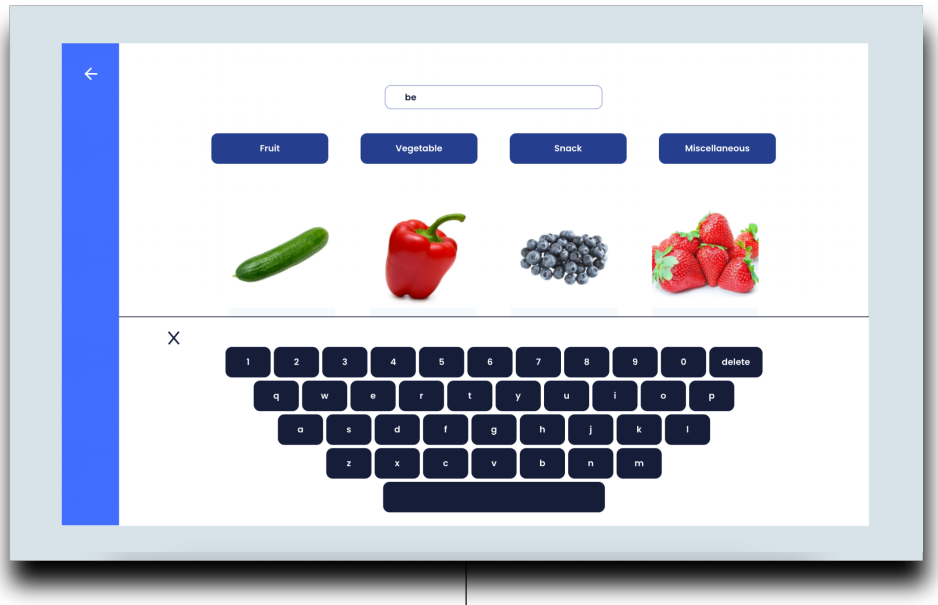
2.5.1 - If your project is available remotely, how can it be accessed?

https://groceryexperience.s3.amazonaws.com/index.html

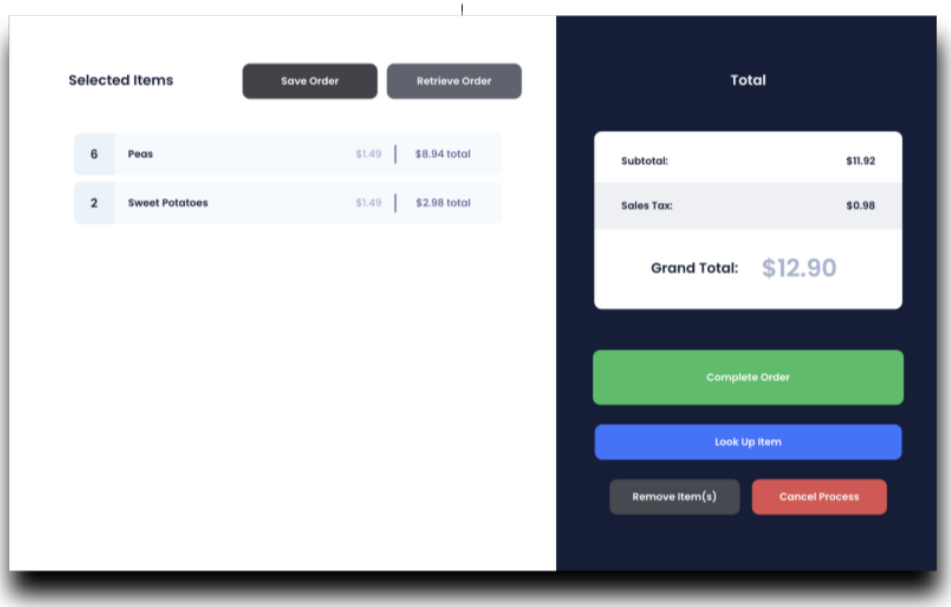
2.6 - Mockup UI screens



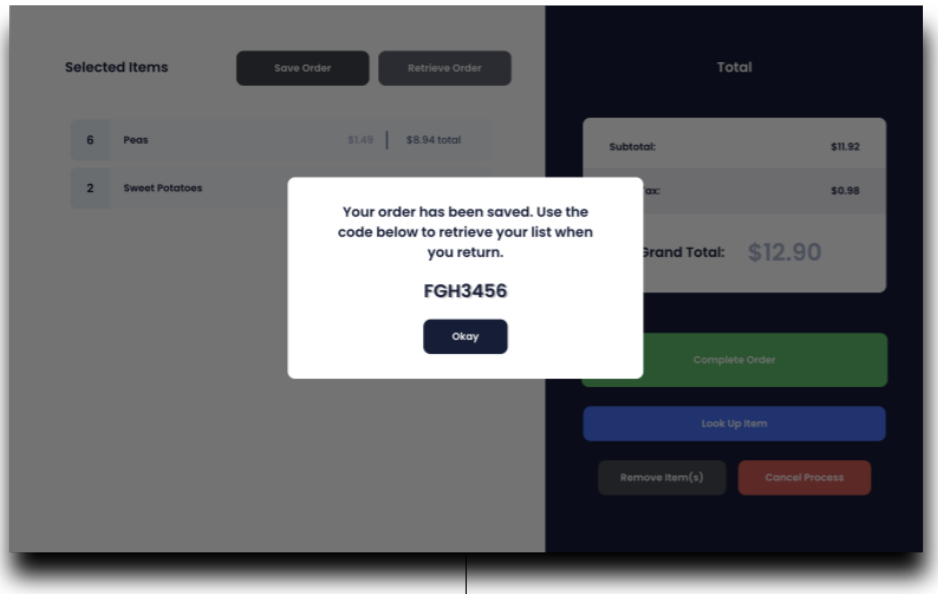
Search option



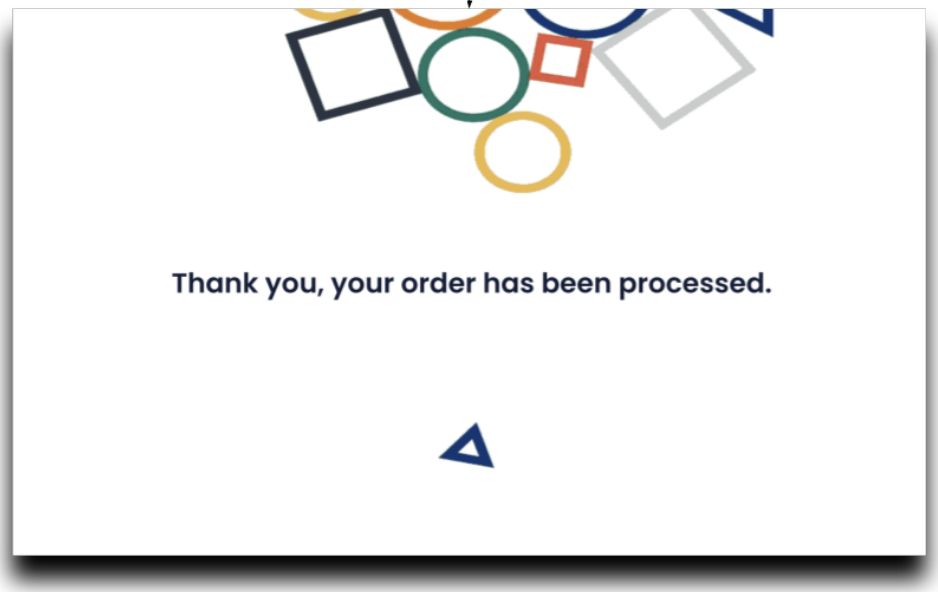
Order screen



Saved order screen



Completed order



2.7 - Risk Table

Risk	Probability	Impact	RMMM
Technical Problems	0.1	5	Company will lose revenue and trust of our Customers; Rapid Code implementation will be required
Risk of Fraud	0.2	5	Website will only allow unique passwords and usernames to make an account; Website will be held in high security standards for the safety of our Customers.
Out of Stock Items	0.2	5	Code can be implemented to track the number of items in stock to prevent Customers from buying those specific items
Unhappy Customers utilizing the Website	0.2	1	Company will only provide satisfaction to those that are confident enough to purchase items through a software; It is important to have the software accessibility and functionality as easy as possible
Customer dissatisfaction on the physicality of the item	0.2	2	Potential damaged Items can have the Customers look at other options; The Grocery Experience will have strict rules on what Items are available for purchase based on the quality

3.1 - Developer documentation

The program we used for the development of this program was visual code on a Macintosh. The process was using react.js and a live window in an internet browser.

Process development of program:

- React.js tools
- Browsers
- AWS to host server
- Visual Studios Code on Macintosh computer
- Online debugging (view live website)

Common process:

- Used knowledge gained in CS courses to build programs.
- Macintosh operating system or Windows 10+
- Knowledge of javascript to debug code
- Knowledge of React.js to update website
- Access images online or take photos of “The Grocery Experience”.

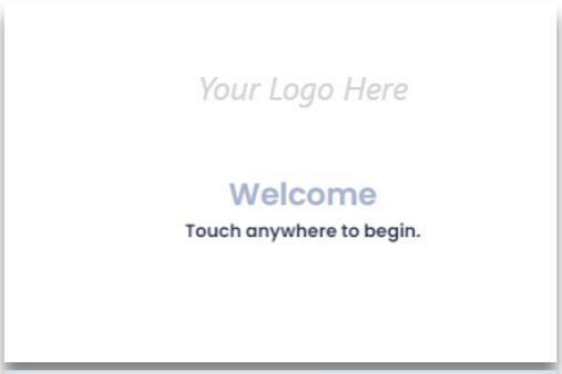
Future development program:

- Using React.js in future development
- Using updated versions of Visual studios
- Updated versions of Mac or Windows.
- Updated versions of React.js to work with
- Expanding on pages in the website
- Expanding on inventory options based on clients needs
- Growing the server to accommodate larger demand online
- Reaching out to new potential clients and advertising program

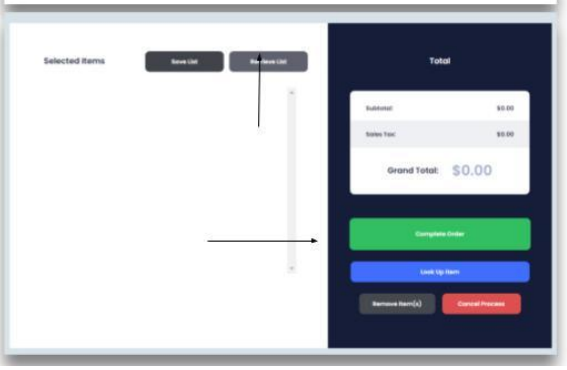
3.2 - User documentation - Notes for a first-time user

User Guide for first time site users.

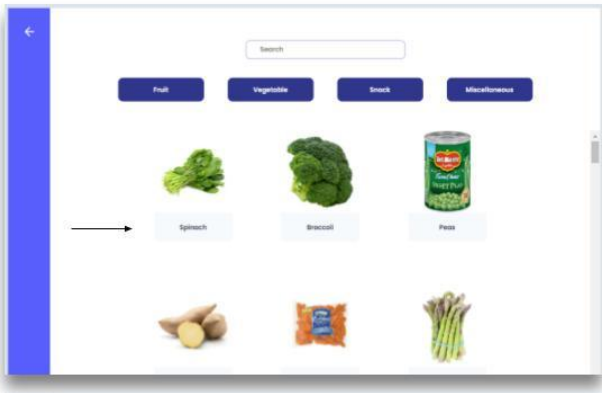
1 **Step one:** touch or click with Mouse to enter website.



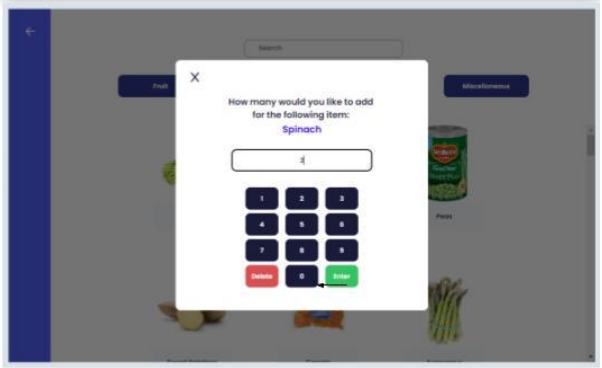
2 **Step 2:** Click on “Look up Item” or “Retrieve list” to load saved cart



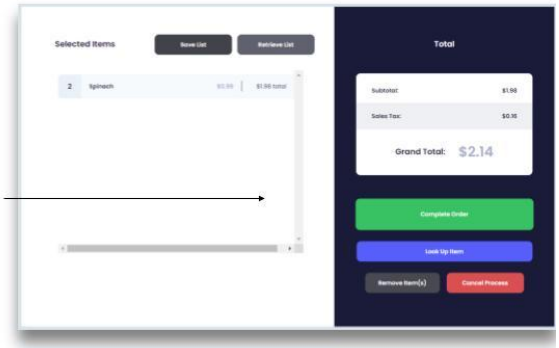
3 **Step three:** Select and item or Category (i.e. Fruit) to begin order.



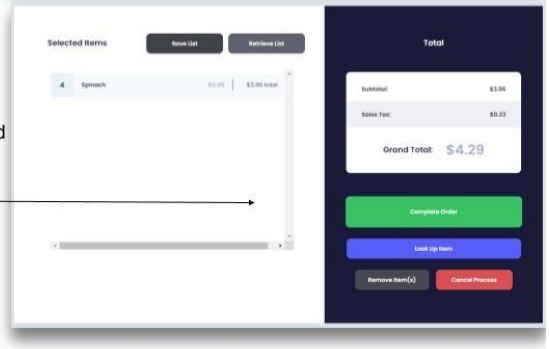
4 **Step four:** Enter the quantity of the amount of the selected item and then press enter.



5 **Step five:** After pressing “enter” in step four you will be taken to cart. repeat **step 2** to add more items.

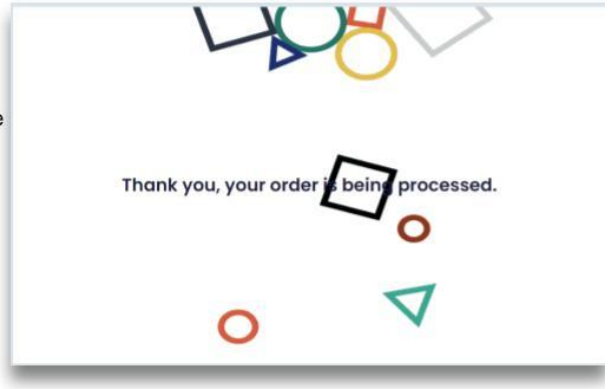


6 **Step six:** If no more items are needed Then click complete order to Finish your order.



7 **Step seven:** Click on place order to Complete payment process

8 **Step eight:** order is placed and you are returned to welcome screen.



3.3 - Version Control/github link

- https://github.com/jge162/362-Group-Project.git

4.1 - What changes did you make to your project over the course of the class? Did any of your requirements have to be changed, added, or removed as you worked through the process?

One of the biggest changes we made to our project over the course of the class was the addition of category buttons in the “Look Up Item” screen. The addition of these buttons introduced a simpler way for our users to break down the scope of items they may be looking for into four categories: fruit, vegetables, snack, miscellaneous.

One thing we removed near the end of the project was requiring users to make accounts before creating their lists. We realized this created many unnecessary steps in an application we wanted to make as simple as possible for all users.

4.2 - What were some of the technical challenges you had to work out? What did you learn about any new technologies you used in your project?

A technical challenge that we had to work out was figuring out a way to send data from the “Look Up Item” page to the “Home” page. The “Look Up Item” page is responsible for displaying grocery items for users to select from and the “Home” page is responsible for showing the total price of all the chosen items added up with tax. If we were to use HTML to code our website, it would be much more difficult to fix this problem of transferring selected item data from one page to another.

This is why we decided to use React as the coding language because of React’s ability to “react” or update when a change has been detected. We learned this reason alone is why React is a leading, popular code in today’s technological world and why so many people are no longer using HTML.

4.3 - Were there any decisions made early in the process that led to later difficulties? How did you resolve those issues? What would you have done differently, knowing what you know now?

At the beginning layout of designing our application, we wanted to add a feature that would make it stand out from other existing grocery related projects. We decided on the addition of a feature that would allow users to save and retrieve a grocery list of their previously added items from their last visit on the application. This led to difficulties because the data (item, item count) being saved had to be stored in the database in a

way that would allow for easy extraction when it came time for users to retrieve their grocery list. We found the best way to fix this problem was to generate a passcode for the user to use upon their return to the application.

This passcode, which is a randomly generated mix of 7 letters/numbers, is stored with the list of their groceries and it also acts as a unique id for efficient database storage. Knowing what we know now, we probably would have used a better system for storing large amounts of data pertaining to both the items in the saved grocery lists and all the data that relates to each individual grocery item. Things quickly became hard to organize when data started piling up.

4.4 - Suppose the class wasn't ending and ending the scope of your project. What is a feature or use case that you would prioritize next? Is there a desired feature or use case that suits your project, but would require a large amount of reworking in order to implement?

Something our group talked about adding would be the feature of allowing users to ship their grocery items to their address. Though it sounds like a simple addition, it would require a great amount of extra steps in our project. First thing to point out is that there would need to be a great amount of new information to add. Employees of the specific grocery store would need to be assigned to prepare these orders. All data related to that alone would need to be tracked effectively.

We would also require users to create an account for important data such as shipping addresses, emails, phone numbers, etc. Because we are dealing with the information of people's personal data, we would also need to increase the security of how this kind of data is encrypted and stored in our database. The addition of this feature would be an incredible detail to see in our project, but there is no doubt that it would take some time to perfect this component before adding it to the final product.