**Project Abstract**

BC Liquor Stores are currently the largest in-person alcohol stores in British Columbia. However, with new technology coming out, the stores are missing out on a lot of revenue which could be made by creating a web application that delivers their products. Our application is designed to solves such problems:

1. Helping BC Liquor Store expand an online store to increase revenue

2. Bringing convenience to our users: Users are able to order their alcohol with a touch of a button

3. Benefiting our society for decreased DUI rate: Using our App may help reduce the likelihood of people driving to get refills

**Competitive Analysis**

Our competitors such as Uber Eats or DoorDash only provide restaurant liquor delivery service. Usually, these products are overpriced due to profit earned by both liquor providers and restaurant owners. Our app helps customers reduce their expenses by directly ordering their favorite product from the one of the largest liquor retailers in BC. Moreover, BC Liquor Stores’ product is more varied than any restaurant in town. From world-renown brands to local popular brands, no matter wine, beer, spirit, coolers or ciders, all purchases can be easily done with a touch of a button.

**Customer**

Our target customers are ages 19+ liquor lovers who enjoy drinking at home, sensitive to price, and prefer board product selections.

**App Test Information(Iteration Two)**

Test accounts are provided, but you would need to sign up with your own phone number to see the actual SMS sent when a driver has accepted your order.

When checking out, you may fill required fields with test data and proceed using this credit card.

**Card Number: 4242 4242 4242 4242**

**Expiry: 10/22**

**CVV: 222**

Driver Partition: Please accept the location request when prompted.

If there are errors on the driver partition, please clear cache and browsing history and visit with HTTPS protocol.

**Test accounts**

|  |  |  |
| --- | --- | --- |
| type | account | password |
| admin | 2368870654 | admin111 |
| General user | 7785982990 | 123456 |
| driver | 6047263371 | test123 |

**These accounts are provided, but you would need to sign up with your own phone number to see the actual SMS sent when a driver has accepted your order.**

**API used**

* Google Maps API

provide a map feature for drivers.

* BC Liquor Store

Provide alcohol information to clients

* Twilio API

Provide phone and messages features for login and passwords reset

* Stripe API

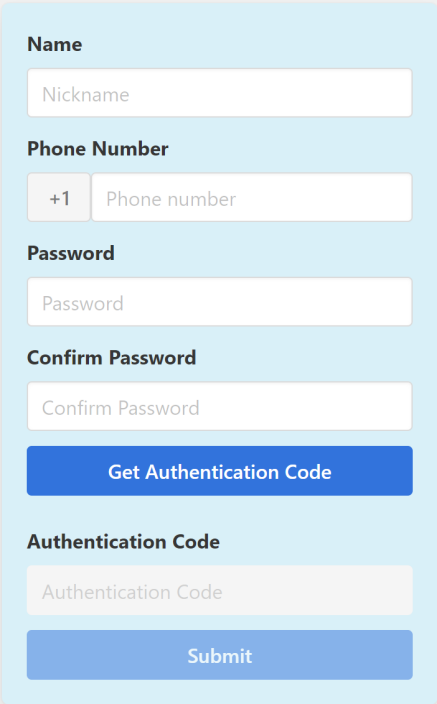
Provide an online payment for clients

**User Stories**

**User Story :** User Sign Up (Iteration 1)

**Name and Actor** : All Users

**Prototype:**



(Sign Up Page )

-----

**Triggers:** As a user, I need to sign up first and then login to the website

**Actions:** After opening the homepage, I click “SignUp” from the navigation bar, then I am redirected to a signUp page with a registration form. I type in my cell phone number and click “get authentication code” button. After I put in the authentication code I get and type in the password, confirm my password and click the submit button to register.

**Unit Test(Iteration 2):**

1.SignUp Form Submission

Test Parameters: The new user’s Name, phone password in the post request

Test Assertions: server get post request and sends response

2.Authentication Code Submission

Test Parameters: The new user’s phone number and Authentication code in the post request

Test Assertions: server get post request and sends response

**Functional Test(Iteration 3):**

1. SignUp PageTest:
2. Test Parameters: the Signup component

Assertion: This component is rendered

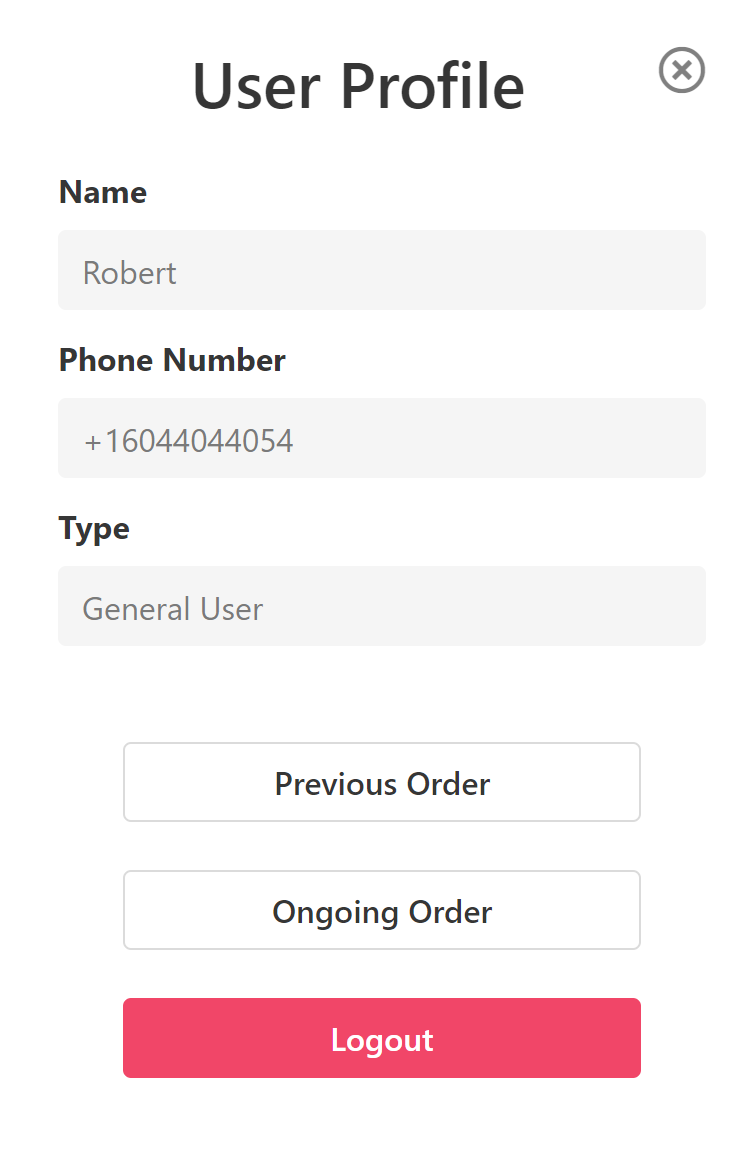
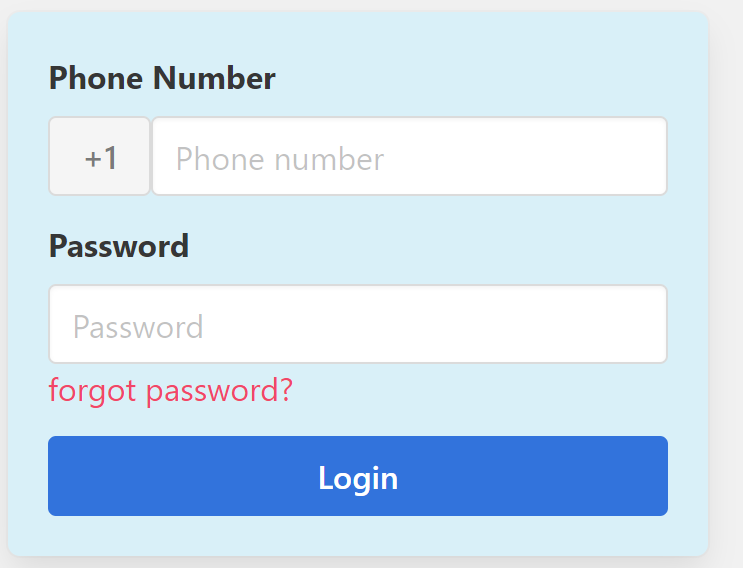
1. Test Parameters: the user’s post request

Assertion: The post request containing user’s authentication code

**User Story: Login** (Iteration 1)

**Name and Actor**:All Users

**Prototype:**



(Login Page and User Profile)

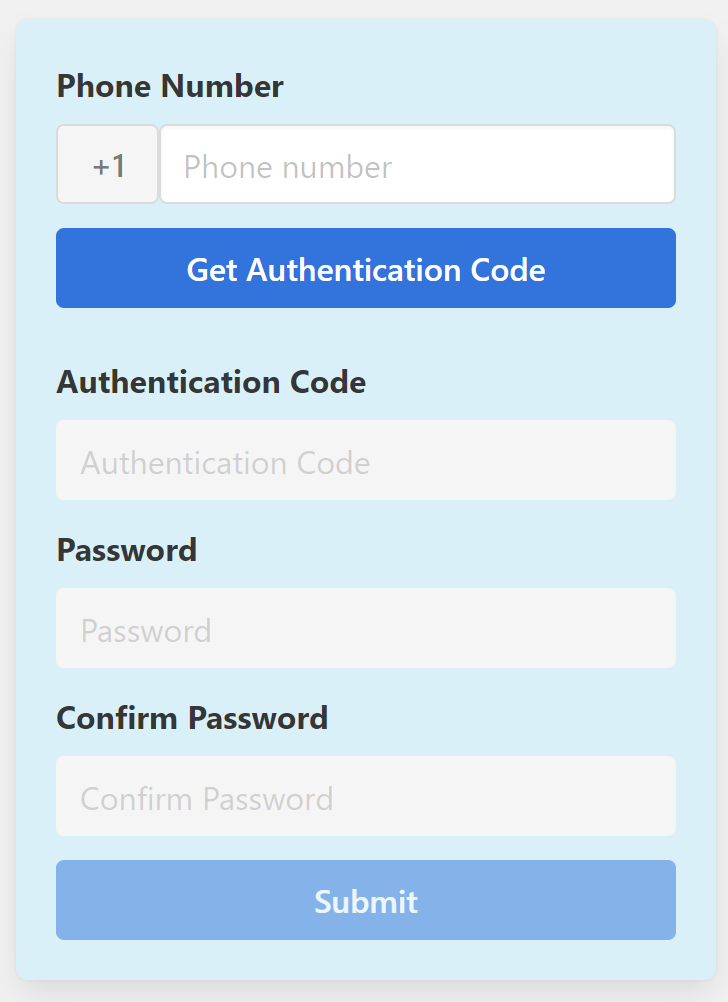
**Triggers:** As an general user, I need to login to the system.

**Actions:** I clicked the “Login” from the navigation bar, then I am redirected to a login page with a login form. I type in my cell phone number and password and click the “submit” button to login. If the password is not correct, I see a warning message: “The password is not correct, please try it again” If the phone number and password is correct, I will be redirected to the home page. I can see my account type on the user profile and my status as “Login”. “Log Out” should show up on my user profile. After clicking “Log Out”, I changed back to the original not login status.

**User Story:** Reset a password when user forgot(Iteration 1)

**Name and Actor**:All Users

**Prototype:**



-----

**Triggers:** As a regular user, I need to sign in my account but I tried several times, I still cannot log in because of the wrong password. After I tried several times to log into my account, I decided to reset my password.

**Actions:**

1. I click on “[forgot password?](http://bcliquorstore.herokuapp.com/ForgoPw)” on the login Page.

2. In order to reset my password, I need to put in my phone number and get an authentication code.

3. After the system checks my authentication code I input my new password

4. I am redirected to the login page and now I am able to use my new password to login to my account

**Unit Test(Iteration 2):**

Test Parameters: The user’s phone number and password in the post request

Test Assertions:server gets post request and sends response

**Functional Test(Iteration 3):**

1. Login Page Render Test:

Test Parameters: the login component render

Assertion: This component is rendered

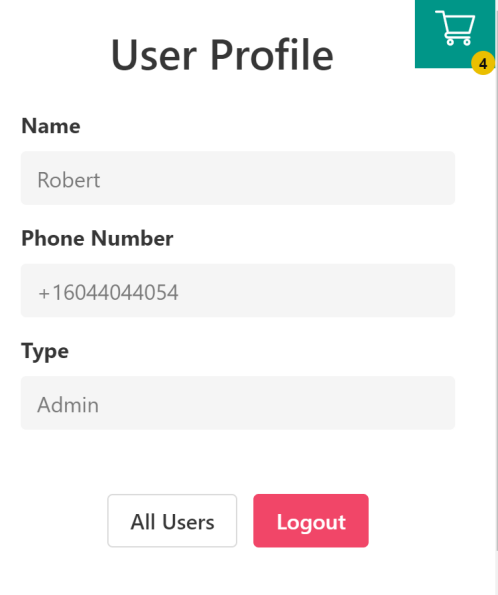
1. Post request Test

Test Parameters: the user’s post request

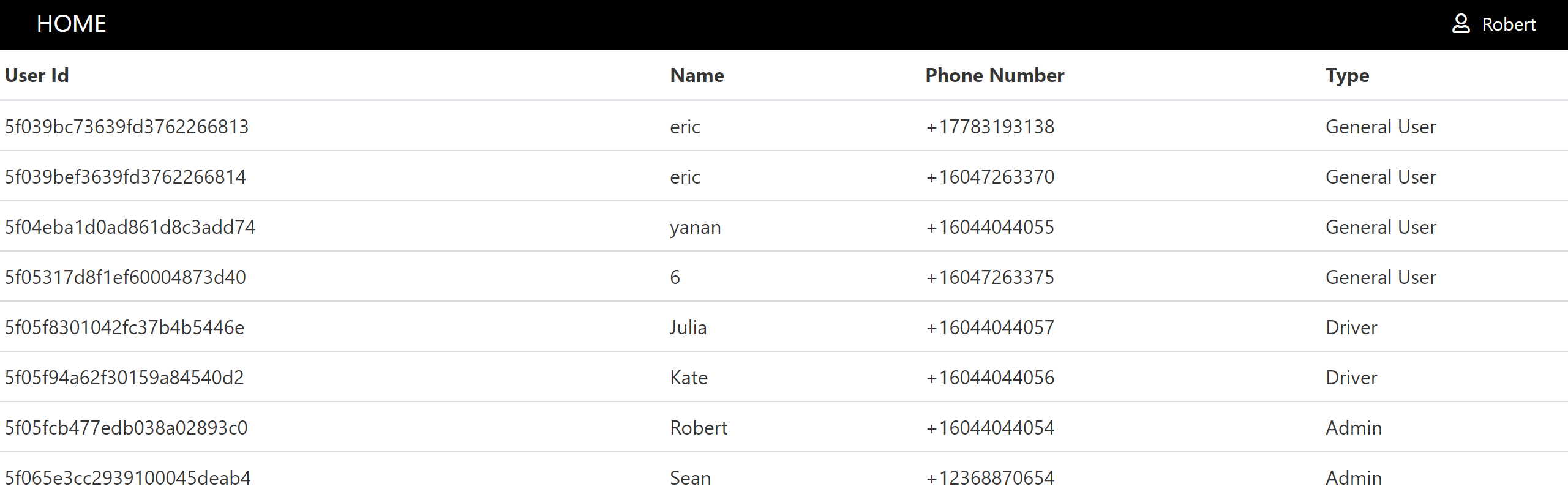
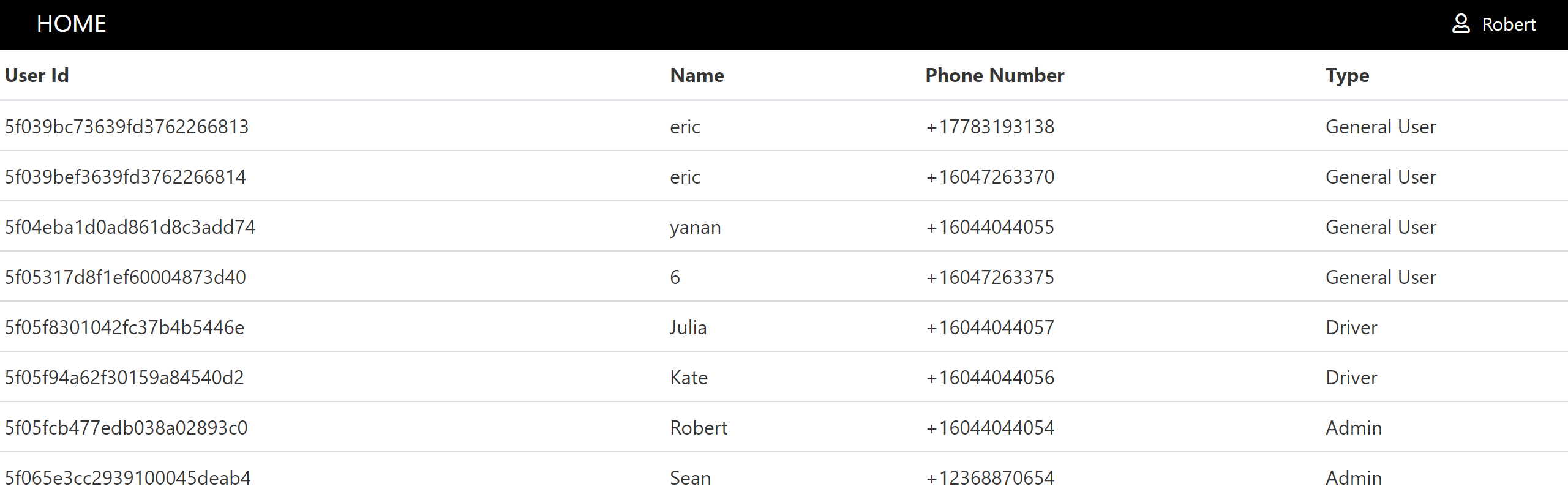
Assertion: The post request containing user’s phone number correctly

**User Story: Administrator can check all Users in the system (Iteration One)**

**Prototype:**



(User Profile for Admin)



(All users View for administrator)

**Triggers:**

As an administrator, I need to know how many customers and drivers are using the system during

**Actions:**

After I log into my account, I can click the all users button at the user panel, all different types of users and their contact information will be shown to me.

**Unit Test(Iteration 2):**

Test Parameters:All user’s information in the post request

Test Assertions:server gets post request and sends response correctly

**Functional Test(Iteration 3):**

Test Parameters: the all users component and its table’s information

Assertion: This component is rendered and table’s component containing user’s registered information

**To Do List (Iteration One)**

**Regular user story ToDoList:**

* Users can view all items and click on specific items to view more details.
* Users can save items to the shopping cart.
* User can view order summary and proceed to checkout
* User can type in a valid delivery address, and pay for the purchase
* User can view their orders’ status and track order’s location
* User can review their orders and grade their drivers’ service

**Administrator story ToDoList:**

* Administer can add, delete and edit drivers
* Administer can match drivers with customers according to driver’s availability and distance
* Administer can check the working status of all drivers

**Driver story ToDoList:**

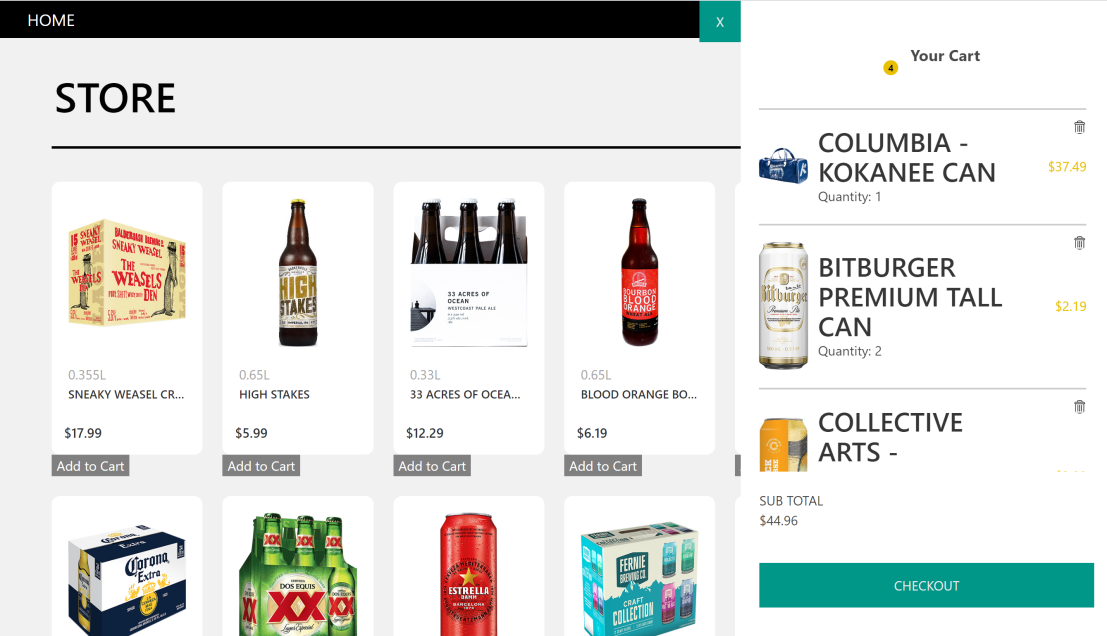
* Drivers can view all the service reviews from the customer.
* The driver can set their working status to “available” for delivery, after being assigned for a delivery task, the status is automatically set to “not available”.

**Discussion about the velocity of the Iteration Two:**

1. **There are five members in our team. Each member should contribute a least one complete story, each story consist of front end code , database implementation and back end code, unit test.**
2. **For Iteration One, we completed designing our project, learning React, MongoDB, Node and a Testing Framework called ’JEST’. We have finished Sign Up, Login and Administrator’s View . We didn’t complete test so we need to complete all unit tests for both iteration one and two. We will leave the functional test to complete for Iteration Three.**
3. **After group discussion, due to the tight deadline, we decide to make our delivery system simple. So instead of making administrator assigning orders, we decide to allow drivers view all available orders’ status, the delivery location showing in the map and drivers can decide to accept their favorite order .**

**User Story**: Add products to the shopping cart (Iteration 2)

**Prototype:**



(Shopping Cart)

**Name and Actor** : All Users

**Triggers:** As a user, I need to add products to my shopping cart

**Actions:** After deciding to buy a specific product, I click the “Add to Cart” button below the product’s picture and add it to my shopping cart. The number of items I selected should show up on the green shopping cart icon at the top corner of the web page. I can click on the cart button to view all the product in my shopping cart. The number, unit price, and picture of each product should show up. I can click the “recycle bin” button to delete any product and click on the “X” button to close the shopping cart view

**Unit Test(Iteration 2):**

Products Page post request

Test Parameters: The user’s post request for product information

Test Assertions:server gets post request and sends response correctly

**Functional Test(Iteration 3):**

1. Test Parameters: Shopping cart’s state

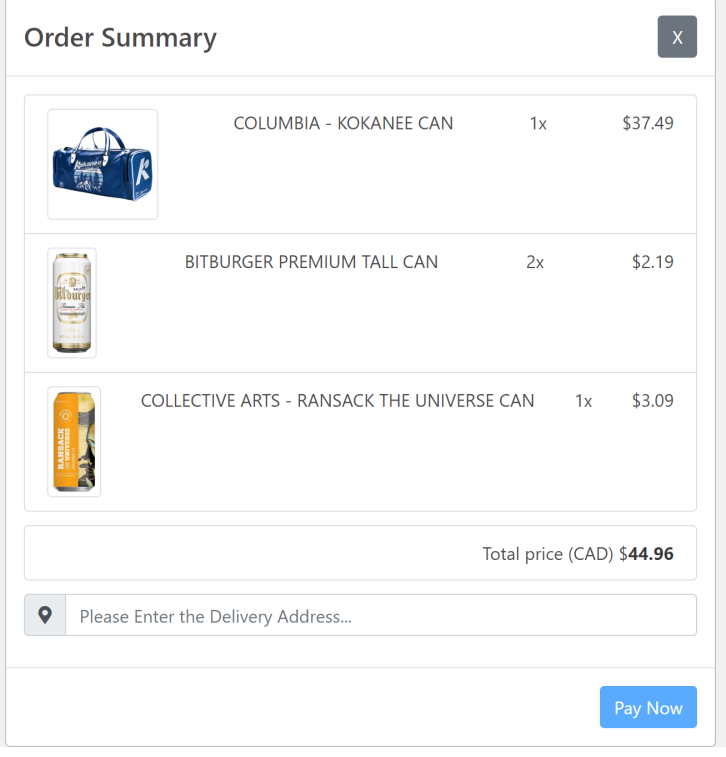
Test Assersions: The shopping cart’s state is updated once user clicked “Add to Cart” button

1. Test Parameters: Product component

Test Asserstions: Product component is rendered according to passing props

**User Story**: CheckOut and Input a Valid delivery Address (Iteration 2)

**Prototype:**



(Order Summary Dialog Box)

**Name and Actor** : General User

**Triggers:** As a general user, I need to confirm my order and type in a valid delivery address to place an order

Actions: I decide to place an order so I click on the “Check Out Button” on my shopping cart. If I am not logged in at that time, I will be redirected to the login page in order to complete my order placement. If I am logged in, a Order summary dialog box will show up. Then I can type in my delivery address into the input box at the bottom of the order summary page. A list of address suggestions will show up and I need to select my address from list. If my input is not selected from the address suggestions list, I can’t click the “Pay Now” Button. This helps ensure delivery address is valid.

**Unit Test(Iteration 2):**

Shopping Cart Check out function

Test Parameters: The user’s passing passing product information from shopping cart such as each product’s “id, name , price ” to “Order Summary” component

Test Assertions:The checkout component receive all information correctly

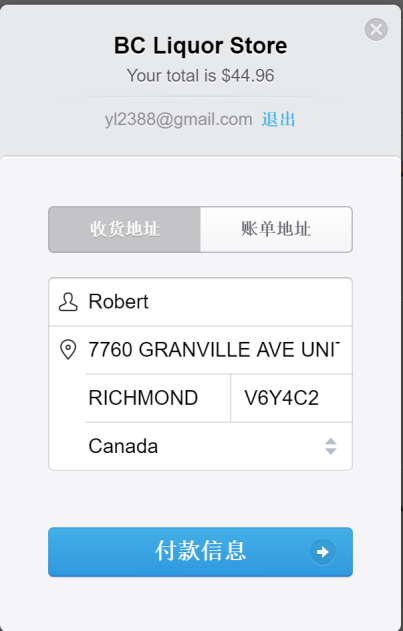
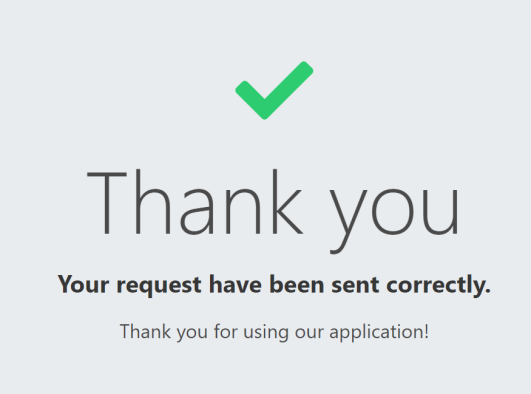
**Functional Test(Iteration 3):**

Test Parameters: the address input box component

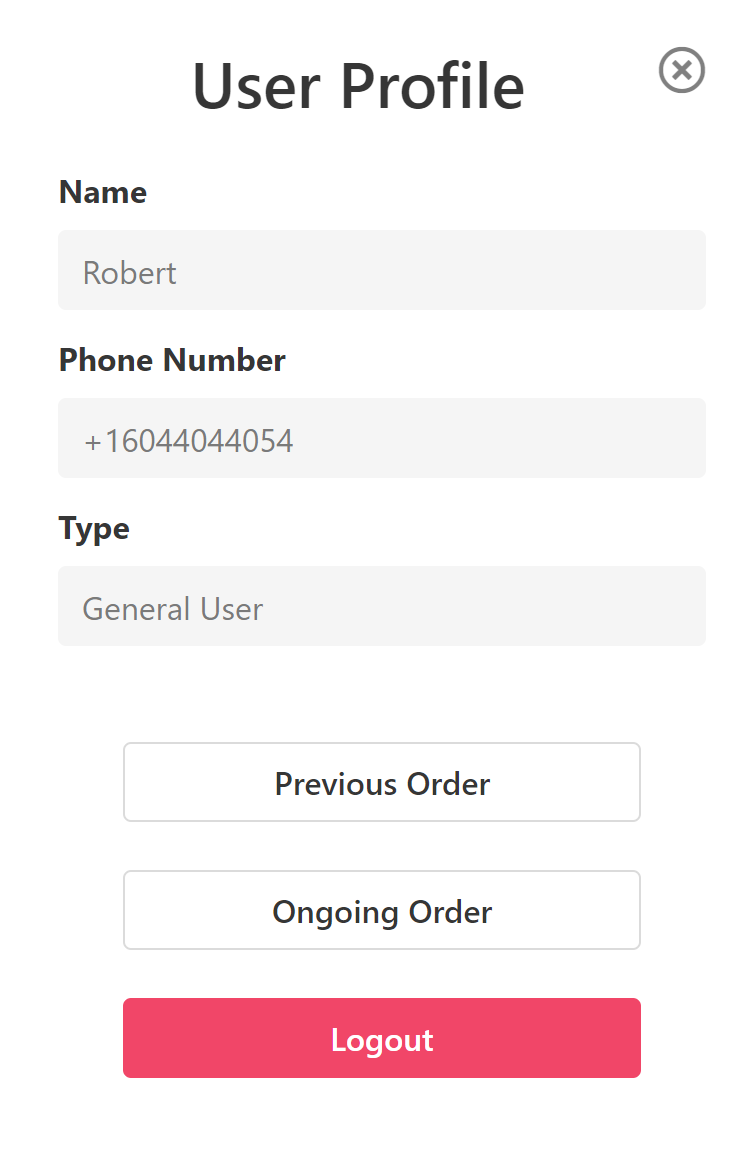
Test Assersions: The address suggestion is updated according to users’ input and the user’s selections status is also updated.

**User Story**: Pay and View Orders (Iteration 2)

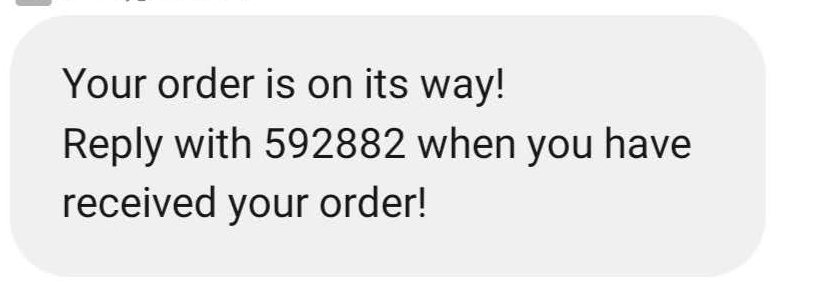
**Prototype:**

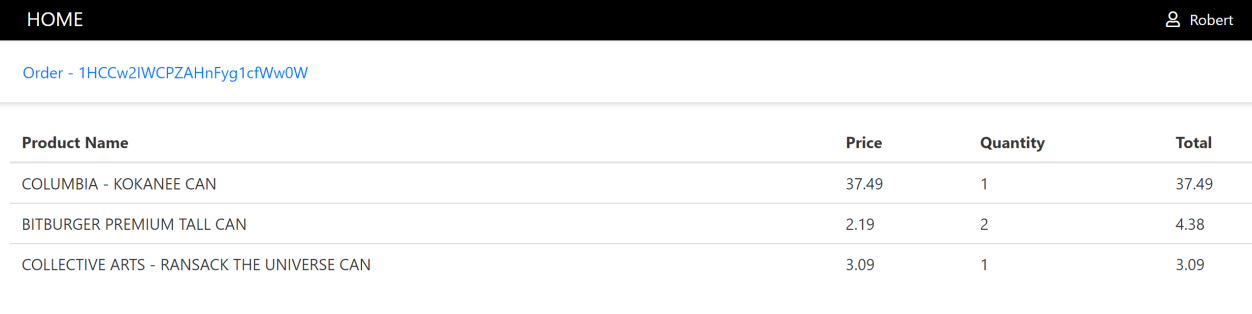
(Payment Form and Thank you Page)



(General User’s User Profile)



（Cell Phone Text Confirmton ）



(General User’s Order Display Page)

**Name and Actor** : General User

**Triggers:** As a general user, I need to pay for my purchase, get a text message for confirmation of my order and view my orders from my user profile.

**Actions:**

After I click “Pay Now ” button on the order Summary page, the payment dialog box shows up and asks for my credit card and other information to finish the payment. After typing in my Credit Card number, expiry Date and click “Pay Now”, if the payment go through successfully, I will be redirected to the thank you page. Otherwise, a warning message will show up “Sorry, the payment is not completed, please give another try”.If the order is placed. I will receive a text message to confirm my order and I may click on my user profile to view all my orders.

**Unit Test(Iteration 2):**

Test Parameters: The checkout component props passing after clicking “Pay Now” Button

Test Assertions：User’s name, phone number,address and user ‘s selected product’s information is passed as props correctly.

**Functional Test(Iteration 3):**

1. Test Parameters: The Stripe Check Out Button

Test Assertions: The Button passing price props to check out form

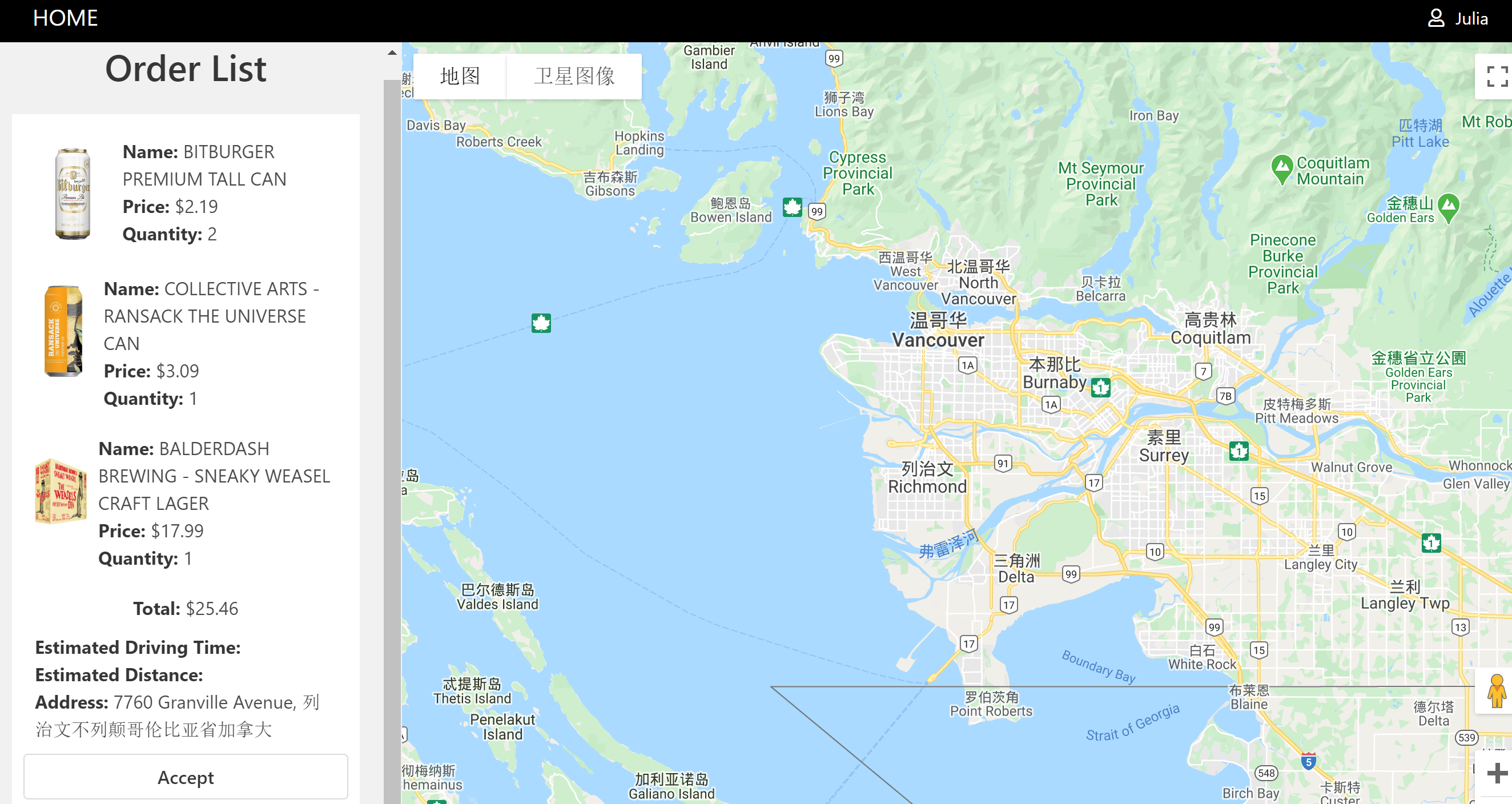
1. Test Parameters: The Previous Order and Ongoing Order components

Test Assertions: After clicking on “Previous Order” or “Ongoing Order” in user profile,

The axios sends a post request and receive server’s response with correct length and data saved in our database

**User Story**: Driver can view and accept ongoing order (Iteration 2)

**Prototype:**



(General User’s Order Display Page)

**Name and Actor** : Driver

**Triggers:** As a driver, I am able to view all ongoing order and accept order

**Actions:**

When I start working, I log into my account, the system recognizes my account as a driver, so the system shows me a map and the list of order which are already paid by customers and waiting for drivers to delivery. I select one from the order list and click “Accept” button to accept the order, the system issue the job to me to deliver the alcohol to the customer. When the customer receives the alcohol that he ordered, he sends a code that provides by the system so the liquor store knows that I finish the order and I get paid for that.

**Unit Test(Iteration 2):**

Test Parameters: Driver’s post request to get all ongoing order

Test Assertions：Server receive user’s post request and give response of all ongoing orders’ information

**Functional Test(Iteration 3):**

1. Test Parameters: Driver Component

Test Assertions: This component is rendered when user is logged in as a role of “Driver”

1. Test Parameters: The Google Map components

Test Assertions: Map is rendered with given points, longitude and latitude, and zoom degree prop

**To Do List (Iteration Two)**

**Regular user story ToDoList:**

* Users can view all items and click on specific items to view more details.
* User can search products in the search box without using search button
* Users can view all products and click the page number to view more products on another page
* After placing order, if user’s order is accepted by a driver, user will receive a system real time notification

**Driver story ToDoList:**

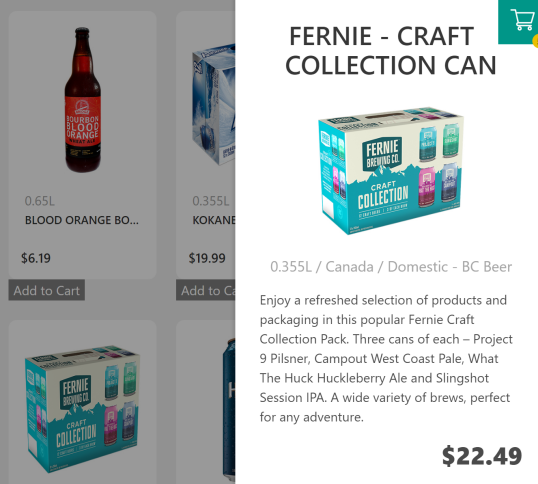
* On order card, Driver can view the delivery location shown in the map and distance from the client
* Drivers can accept the order and once the order is accepted, other drivers can’t view accepted orders anymore

**Discussion about the velocity of the Iteration Three:**

1. **After group discussion, we found our idea is pretty ambitious and hard to complete all desired detailed functionality in two weeks. So we decide to keep our project simple and make our efforts to complete all planned essential functions. Each member should contribute at least one complete story. After Iteration Three deadline, we will still try to implement more functionality to improve our project.**
2. **During our course lecture, our instructor emphasis the importance of project test for the last iteration, so instead of adding more features, we decide to complete thorough functional tests for all features for our project.**

**User Story:**View Product Detail (Iteration 3)

**Prototype:**



**Name and Actor** : All Users

**Triggers:** As a regular user, I need to view a product description and zoom in picture

**Actions:** I click on the picture I am interested and I can see a zoom in product picture, the full name of this product and detailed product description.

**Unit Test(Iteration 2):**

Test Parameters: After opening the home page, user sends a post request to get all product displaying on the web page

Test Assertions：Server receive user’s post request and give response of all product information

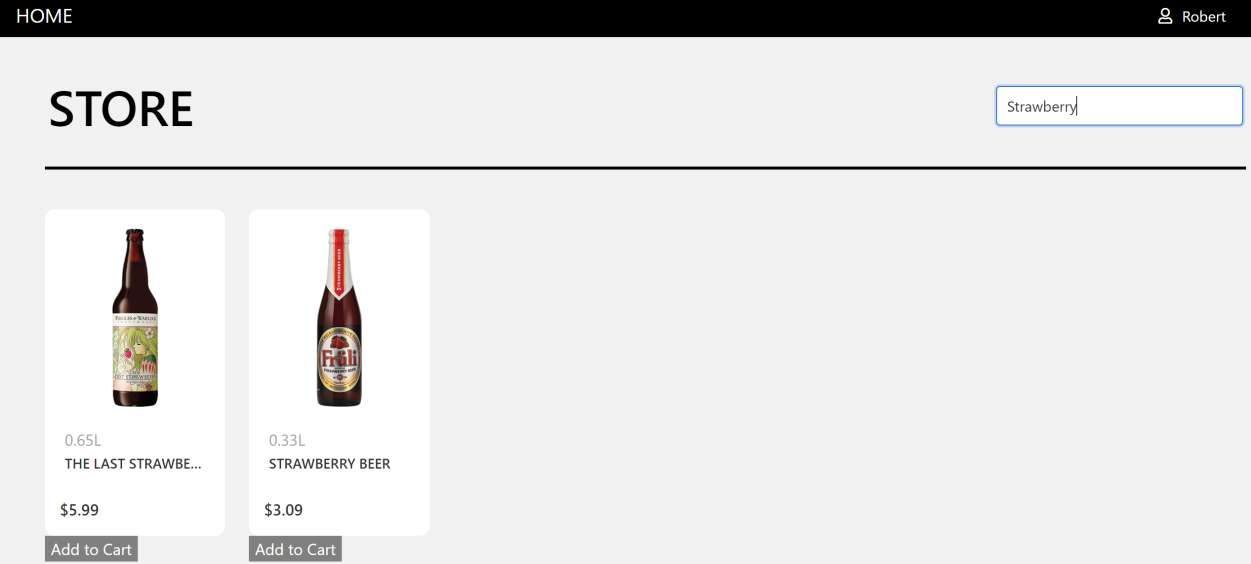
**Functional Test(Iteration 3):**

Test Parameters: ProductDetail Component

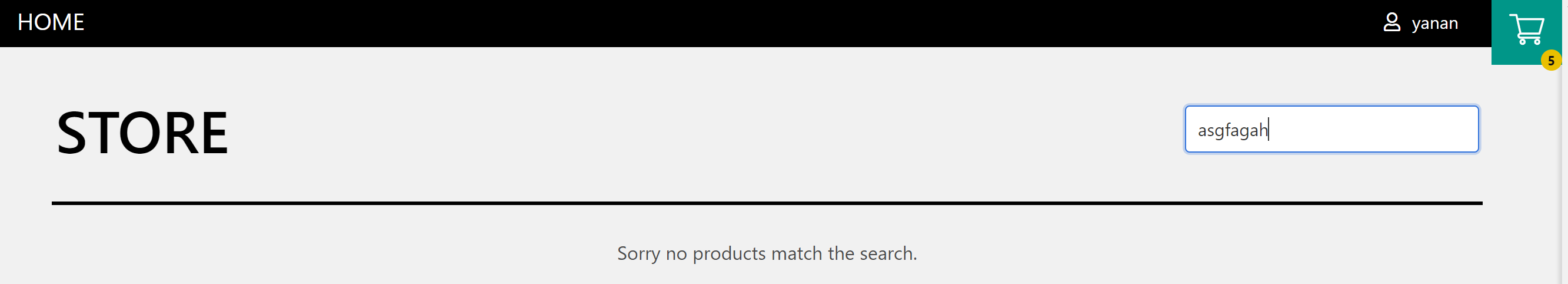
Test Assertions:ProductDetail Component is rendered with given image, title and price props

**User Story:** Search Products(Iteration 3)

**Prototype**:



(Search Product)



(No Match Result)

**Name and Actor:** All User s

**Triggers:**

As a regular user, I need to search interested products.

**Actions:**

After opening the homepage, I type the name of the alcohol in the tool box. As my input changes, I can see the a list of products with search keyword showing up.

If there is no matched result, I will see “Sorry, no products match the search”

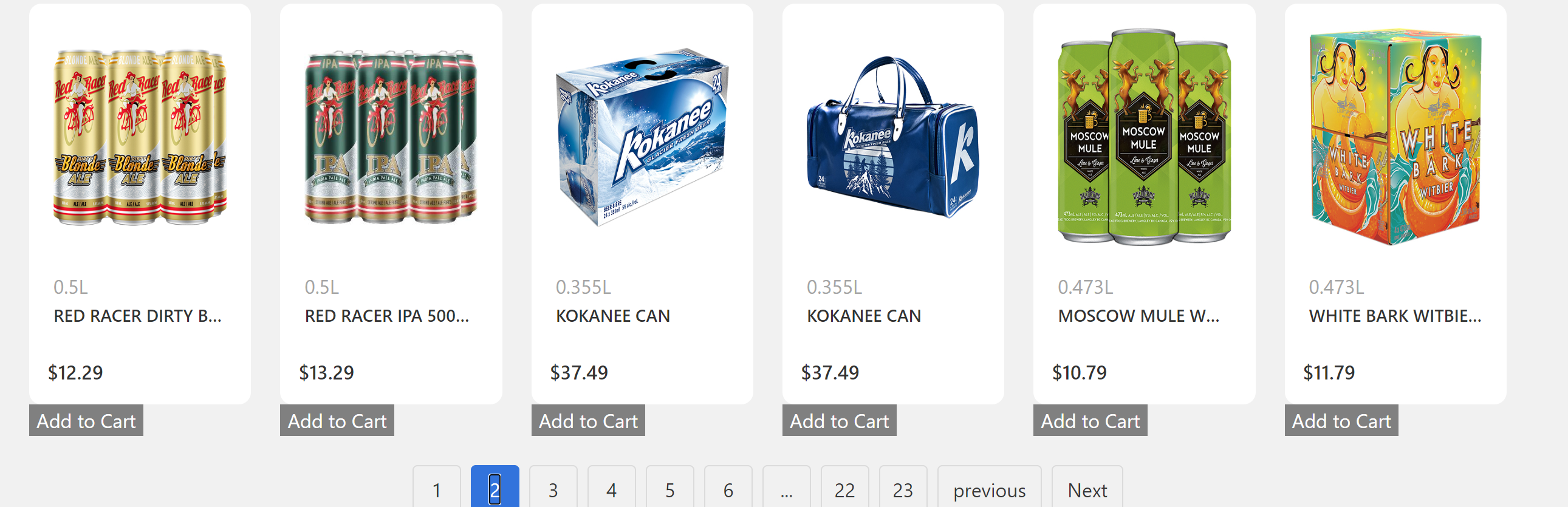
**Functional Test(Iteration 3):**

Test Parameters: ToolBox Component with search string

Test Assertions:ToolBox Component is rendered with results with desired search keyword

**User Story:**View All Products (Iteration 3)

**Prototype:**



**Name and Actor** : All Users

**Triggers:** As a regular user, I need to view all products

**Actions:** I can see the page numbers on the bottom part of the products page, after clicking on the page number, the products page should be reloaded with different product. The page number I am currently viewing should be highlighted and am able to click on another page number to view different products

**Unit Test(Iteration 3):**

Test Parameters: After opening the home page, user sends a post request to get all products displaying on the web page

Test Assertions: Server receive user’s post request and give response of all products information

**Functional Test(Iteration 3):**

1. Test Parameters: Products state

Test Assersions: The product component is rendered and status is updated according to clicked page number