Quick Cart App
Android Mobile Application

<Version 1.0.3>

Prepared by

Dejeon Battick	180914	dejeon.battick11@gmail.com
Shemar Lundy	180916	lundyshemar@yahoo.com
Shemar Henry	180915	shemarhenry24@yahoo.com
Mark-Anthony Jones	180920	jmarkanthony.062@gmail.com

Course Instructor:	Thomas Xu
Course:	Web and Mobile Application development II
Date:	June 23, 2019

Overall Description

Product Context

The Quick Cart application was designed for everyday shoppers who visit a supermarket. The Quick Cart mobile application for android also has an iOS counterpart and allows the user to shop easily in the comfort of their homes, from work or wherever. The application allows users to search for items in the supermarket and add it to their cart. They would then check out such items from the cart and a QR code would be generated for those items, after which the order would be sent to the supermarket and prepared and packaged by the workers in the supermarket. The customer would then go to the supermarket and present the QR code and collect their items which have been prepared. The customer could collect their items after payment has been made. The Application is also very useful for customers who visit the supermarket and cannot find the items they are looking for. The locate page allows users to search for items and find where the item is located, rather than manually searching for the items while in the supermarket. The application also has an online community blog. This community blog will allows users to share recipes with each other and be kept abreast on posts made by the Quick Cart Community.

Product Functionality

Major functions of the system will be to:

1. Allow users to search for items.
2. Allow users to add items to a cart.
3. Allow users to locate items as in the supermarket.
4. Allow users to order and check out items.
5. Allow users to visit the community blog.

Non-functional requirements of the system will be to:

1. users information should be stored on a secured database.
2. User password should be hashed to ensure security.
3. System must check credentials against the user database to ensure that there isn't duplication in the email field.
4. System should ensure search results are present in a timely manner.
5. System should ensure that a user login session is kept until the user manually logout .
6. System should ensure that the main flow of the application is smooth.

Stakeholders and Users Characteristics

The key stakeholders of this system are:

Shoppers:

- Persons who use the Quick Cart application to shop from their homes.
- Persons who visit the supermarket to shop in store.

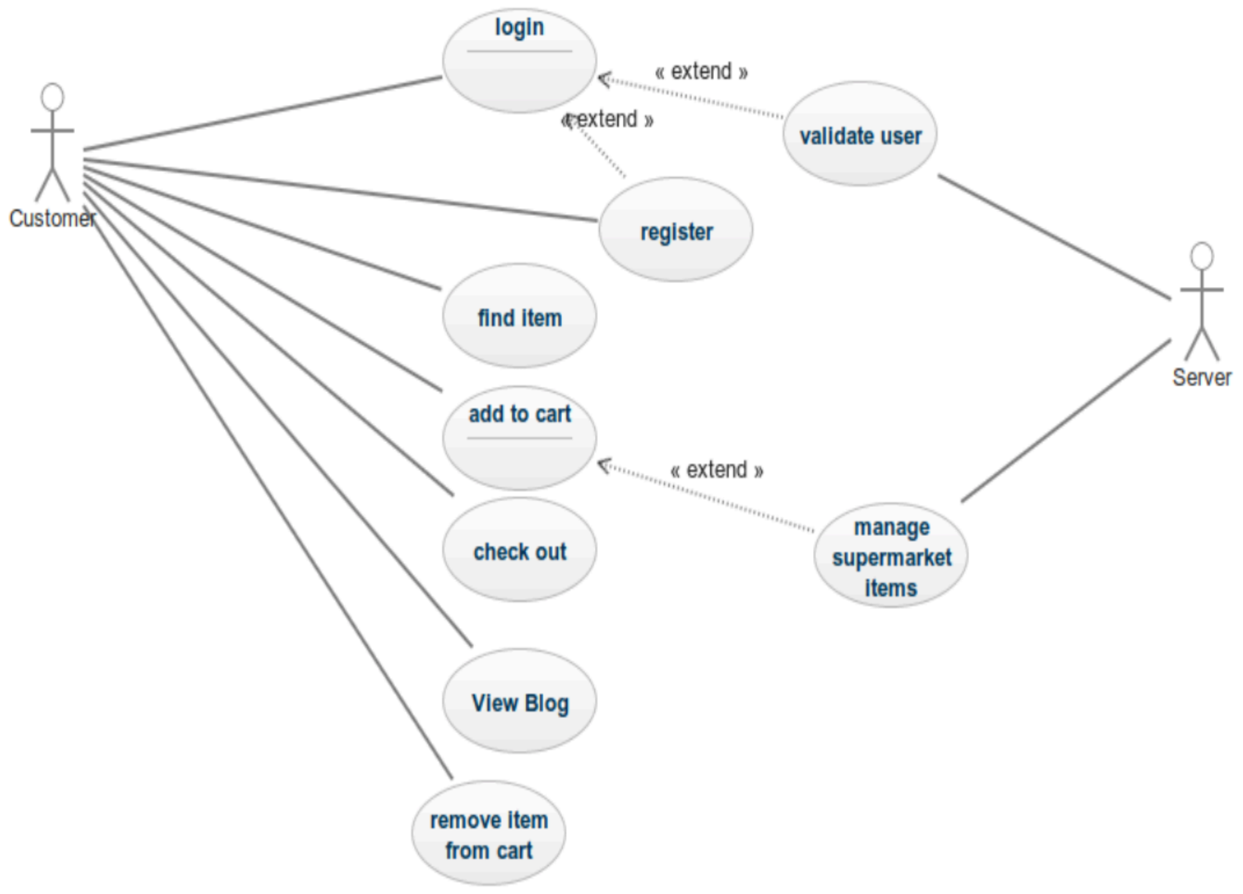
Store Staff:

- Persons who view shoppers receipt to prepare their packages.

Operating Environment

The application is suited for android Operating Systems Oreo 8.1 and all newer versions. This android version was chosen because of the requirements and to be able to suit most android devices. Internet connection is also required to query for supermarket items, to get a list of things to be prepared for that user. Internet connection is also required to checkout order and visit the blog.

Use Case Diagram



Use Case Description

Use case name: Login

Summary: the actor should be able to log into the application.

Primary Actor: Customer

Secondary Actor: System

Precondition: the user must have an account registered.

Main Sequence:

1. The user enter their email
2. The user enter their password.

3. The user click the submit button.
4. The user credentials are validated by system server.
5. the user logs into the system.

Alternative sequence:

Step 5b: if credentials are invalid appropriate error reported to user

Step 5c: if the user does not have a registered account then they can go to the registration page.

Post-condition: Login success confirmation

Use case name: Registration

Summary: the actor should be able to create a new account.

Primary Actor: Customer

Secondary Actor: n/a

Precondition: the user must have the application installed.

Main Sequence:

1. The user clicks the registration text field
2. The user enter their user name.
3. The user enter their email.
4. The user enter their password.
5. The user enter their password again for confirmation.
6. The user click the submit button.
7. The user logs into the system.

Alternative sequence:

Step 6b: if the user clicks the login button then the can return to login page.

Step 6c: if form is erroneous appropriate error reported to user

Post-condition: A new user is created and logged in

Use case name: add to cart

Summary: the actor should be able to add a searched them to the cart.

Primary Actor: Customer

Precondition: User logged in, searched for and found item

Main Sequence:

1. The user adjusts the amount for desired item
2. The user chooses to add the item to the cart

Post-condition: Item is added to user cart.

Use case name: find item

Summary: the actor should be able to search for item to know where they are located in the supermarket.

Primary Actor: Customer

Secondary Actor: System

Precondition: The user must be logged in.

Main Sequence:

1. The user goes to locate
2. The user accesses the search bar
3. The user search for the item they want to find

Post-condition: The system server responds with matching items displayed with their in-store location.

Use case name: search for item

Summary: the actor should be able to search for item to know where they are located in the supermarket.

Primary Actor: Customer

Secondary Actor: Server

Precondition: The user must be logged in.

Main Sequence:

1. The user goes to search
2. The user click on the search bar
3. The user search for the item they want to find

Post-condition: The system server responds with matching items.

Use case name: check out

Summary: the actor should be able to check out items added to the cart.

Primary Actor: Customer

Secondary Actor: System

Precondition: The user must have items added to cart.

Main Sequence:

1. The user go to the cart page
2. The user chooses check out option
3. Application generates a QR code for order
4. Application sends order of items in cart and matching code to system server
5. The user's cart is emptied

Post-condition: QR code is saved for future reference.

Use case name: remove item from cart

Summary: the actor should be able to remove items from the cart.

Primary Actor: Customer

Precondition: The user must have items added to the cart and on the cart page.

Main Sequence:

1. The user prompts item options
2. The user chooses delete
3. The item is removed from the users cart

Post-condition: Items are permanently and persistently removed from the users cart.

Use case name: View Blog

Summary: The actor should be able to visit the community blog which is the web service of the application.

Primary Actor: Customer

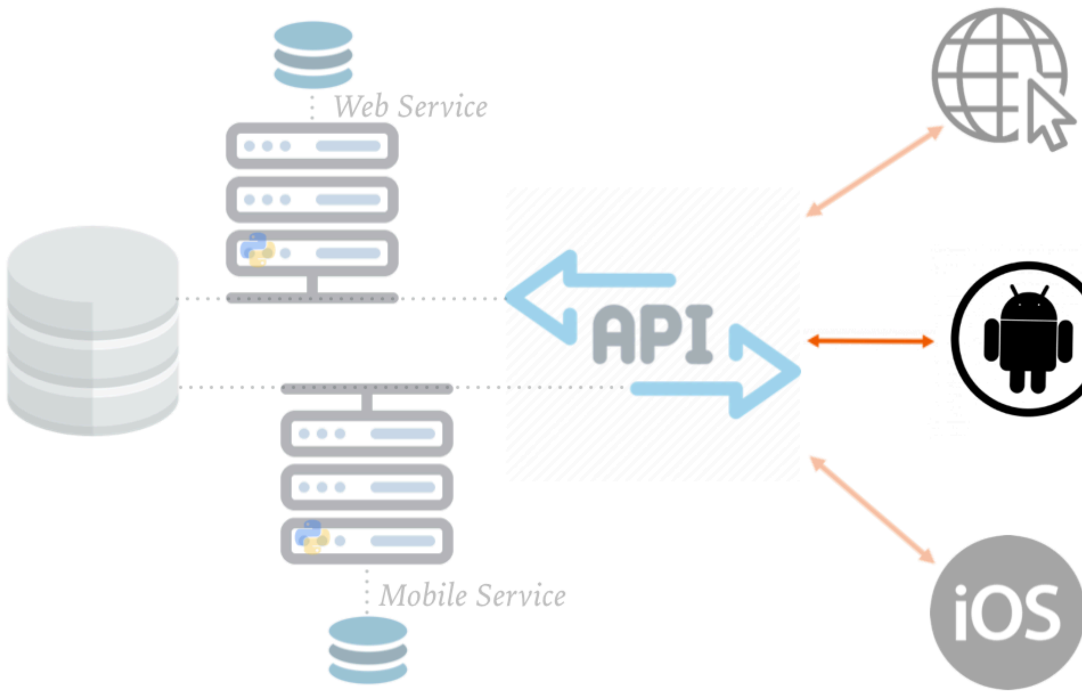
Precondition: The user should have internet connection.

Main Sequence:

1. The user click on the navigation button.
2. The navigation bar is open.
3. The user clicks on the community text.
4. The online blog is launched.

Post-condition: The blog web page is open online.

Structure



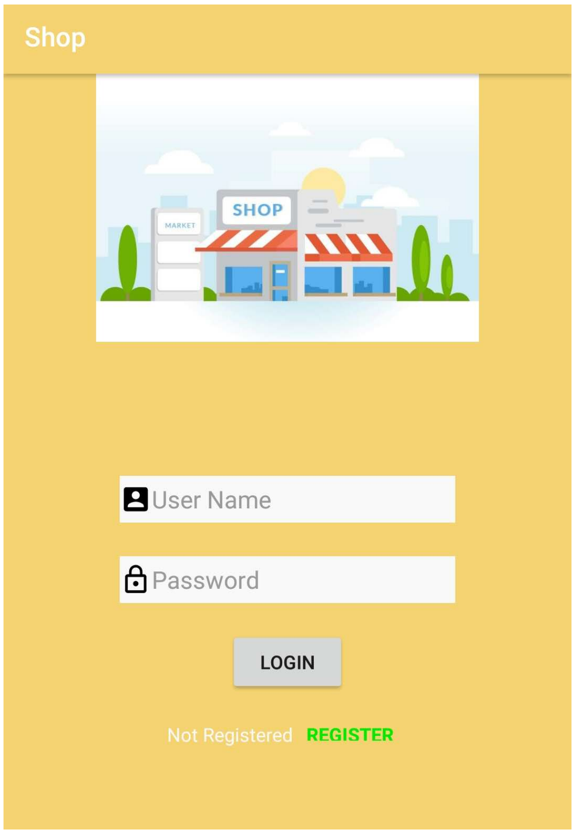
Above is an image of the structure of the quick cart application. As mentioned before there is an iOS version that is discussed in more details in the iOS submission. There is also a Web service community that is also discussed in the web submission. There is a middle layer API that handles all the heavy lifting and communication between layers.

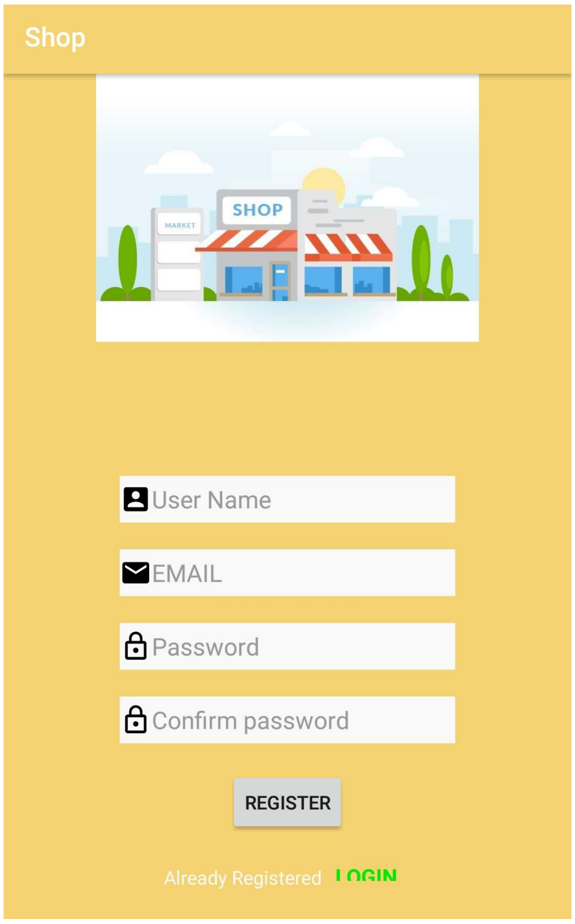
Supermercado was developed using the following set of android studio frameworks and structures:

- HTTPRequests - To allow communication with Supermercado Server
- JSONArray and JSONObject - To parse and manage server information
- Asynchronous Tasks - To manage threads, allowing HTTPRequest results to be shown to user in real time
- Volley - To assist with parsing server information
- Session - To utilise persistent operation for login and general data management
- SQLite - To store searched items and QR-codes locally
- RecyclerViews and Adapters - To display listed information to user
- Object Classes - To manage specific data objects like QR-code and shopping items
- Activities and Fragments - To control the processes, views and manage the data that the user has access to within this solution.

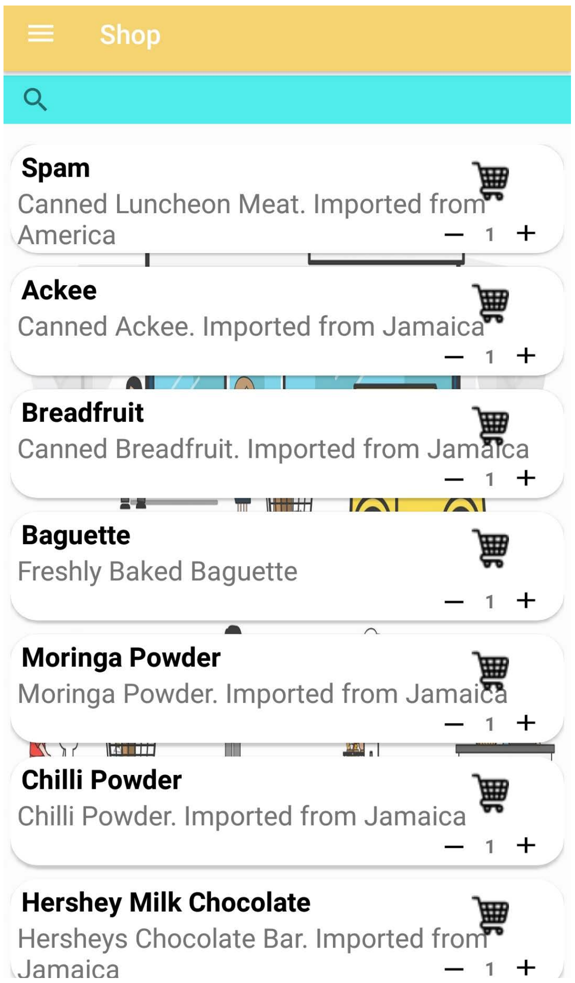
The Solution

The right image is the login page where existing users can login with their valid information. The users credentials are stored in the My SQL database. The validation is done on the back-end of the application to ensure that passwords and usernames match. If the user is a new user they can click on the register text field to go there and register.





The left image is the registration page in which a new user can sign up. The page takes the user's, name, email and password. It also requires the password to be entered twice for verification. The new user's information is stored in the database as a new field is created with their information.



Shop

Spam
Canned Luncheon Meat. Imported from America
— 1 +

Ackee
Canned Ackee. Imported from Jamaica
— 1 +

Breadfruit
Canned Breadfruit. Imported from Jamaica
— 1 +

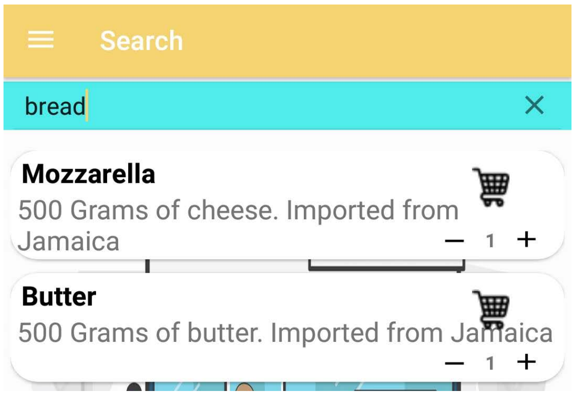
Baguette
Freshly Baked Baguette
— 1 +

Moringa Powder
Moringa Powder. Imported from Jamaica
— 1 +

Chilli Powder
Chilli Powder. Imported from Jamaica
— 1 +

Hershey Milk Chocolate
Hersheys Chocolate Bar. Imported from Jamaica
— 1 +

The image to the left shows the main page when logged in, the shopping page where persons can search, view items and add them to cart. This page features a search bar for persons to search for a specific item they are looking for. The search bar accepts the user's request and make a query to the server, resulting in the gathering the search related items from the database which are then displayed to the users using a recycler view.

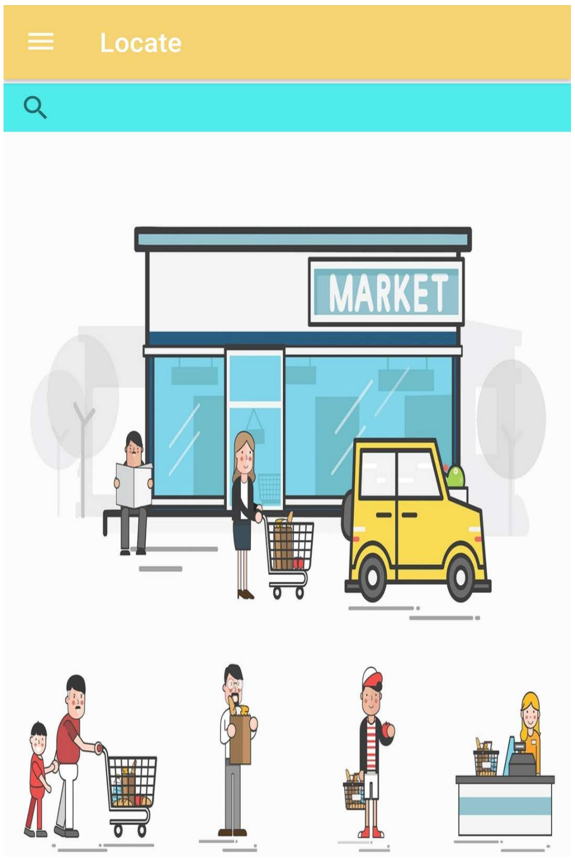


Search

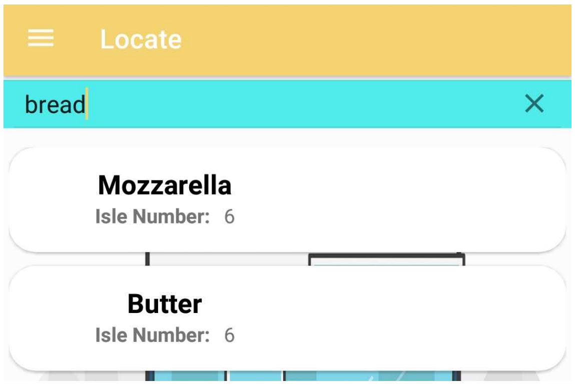
bread

Mozzarella
500 Grams of cheese. Imported from Jamaica
— 1 +

Butter
500 Grams of butter. Imported from Jamaica
— 1 +



To the left is an image of the find item page which is very useful for in store shoppers to locate items they are searching for. The view for this page only shows the name of the item and where to find it. This page also has a search view which works similar to the search page.



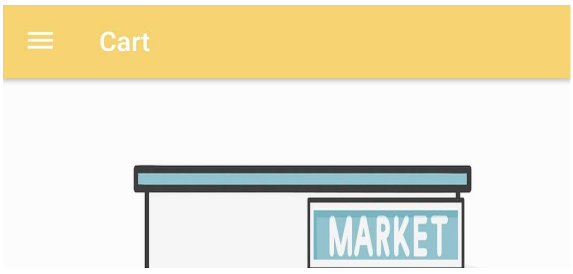
Locate

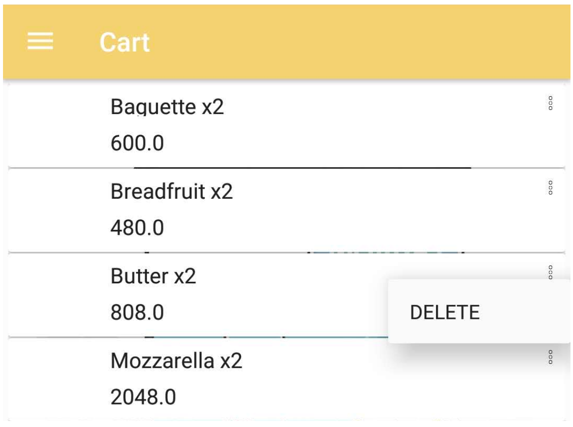
bread

Mozzarella
Isle Number: 6

Butter
Isle Number: 6

The image to the right is of the cart that users can view the items they added. They are also able to delete items before they check out. After they check out they see a QR code which is their receipt to be scanned to collect their items.



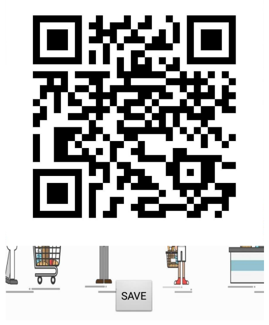


Baguette x2
600.0

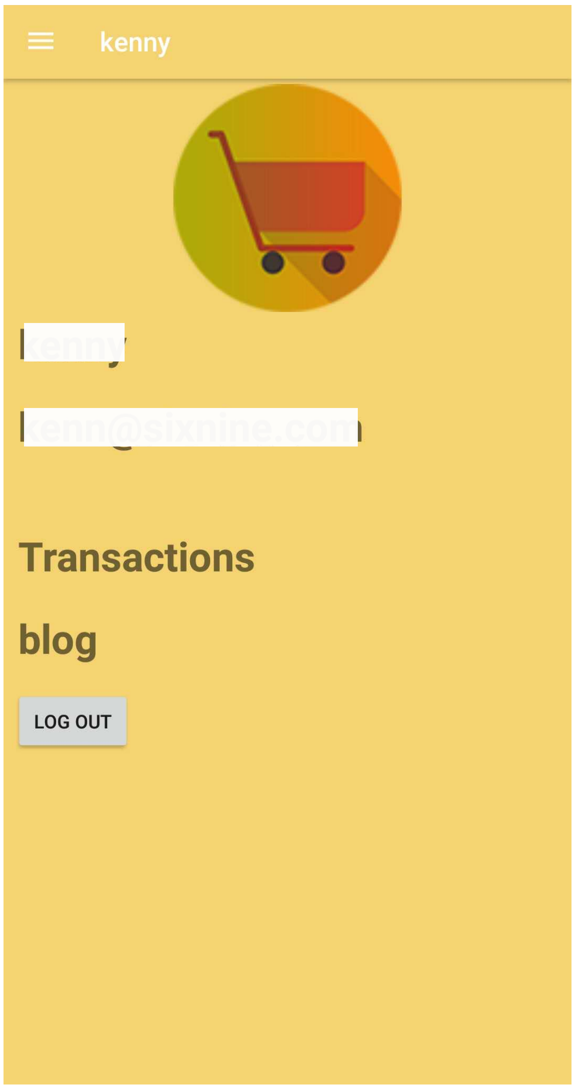
Breadfruit x2
480.0

Butter x2
808.0

Mozzarella x2
2048.0



To the right is an image of the profile page which has the information of the user, a link to the online blog and a link to their transactions. A user can click on blog to take them to the online website which also requires internet connection. Persons can click on transaction to view their stored QR codes for their packages to be collected

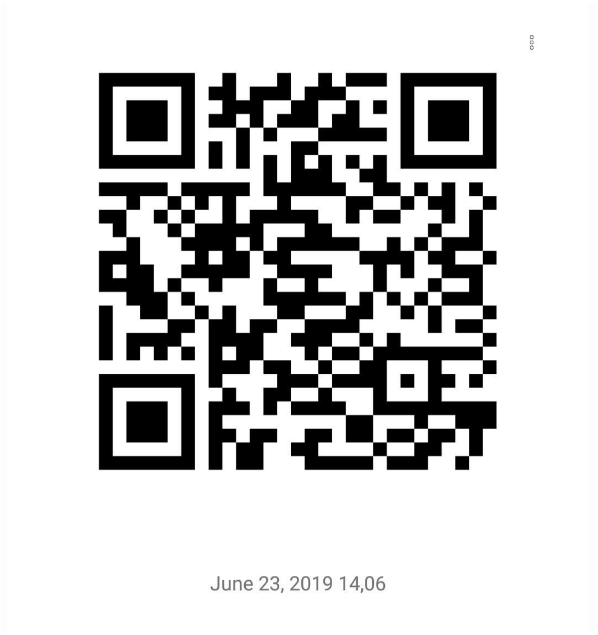


[redacted]

[redacted]

Transactions

blog



June 23, 2019 14,06

Future Implementations:

- Implementation of in app payment
- Implementation for users in app information management
- Implementation for a delivery and delivery tracking service
- Implementation of better user purchase management, including history
- Implementation of a “favourites” feature for users to add items to a favourite list

Group members contributions

Our group members, namely Dejeon Battick, Shemar Lundy, Shemar Henry and Mark-Anthony Jones worked to achieve our goals for this project. We are proud of what we have done and with each other’s contributions to the project.

Dejeon Battick: worked on the login and registration features and user interface and middle layer(API).
Shemar Lundy: worked on the user interface for the web server
Shemar Henry: worked on the middle layer(API) with interaction between the app and the database, and the cart management
Mark-Anthony Jones: worked on back-end including the database and servers