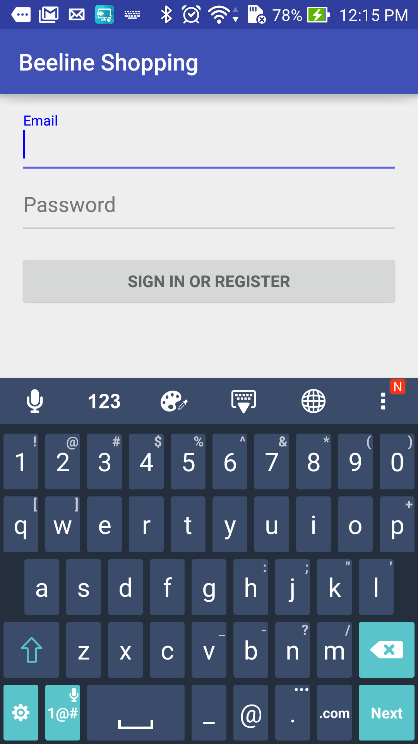
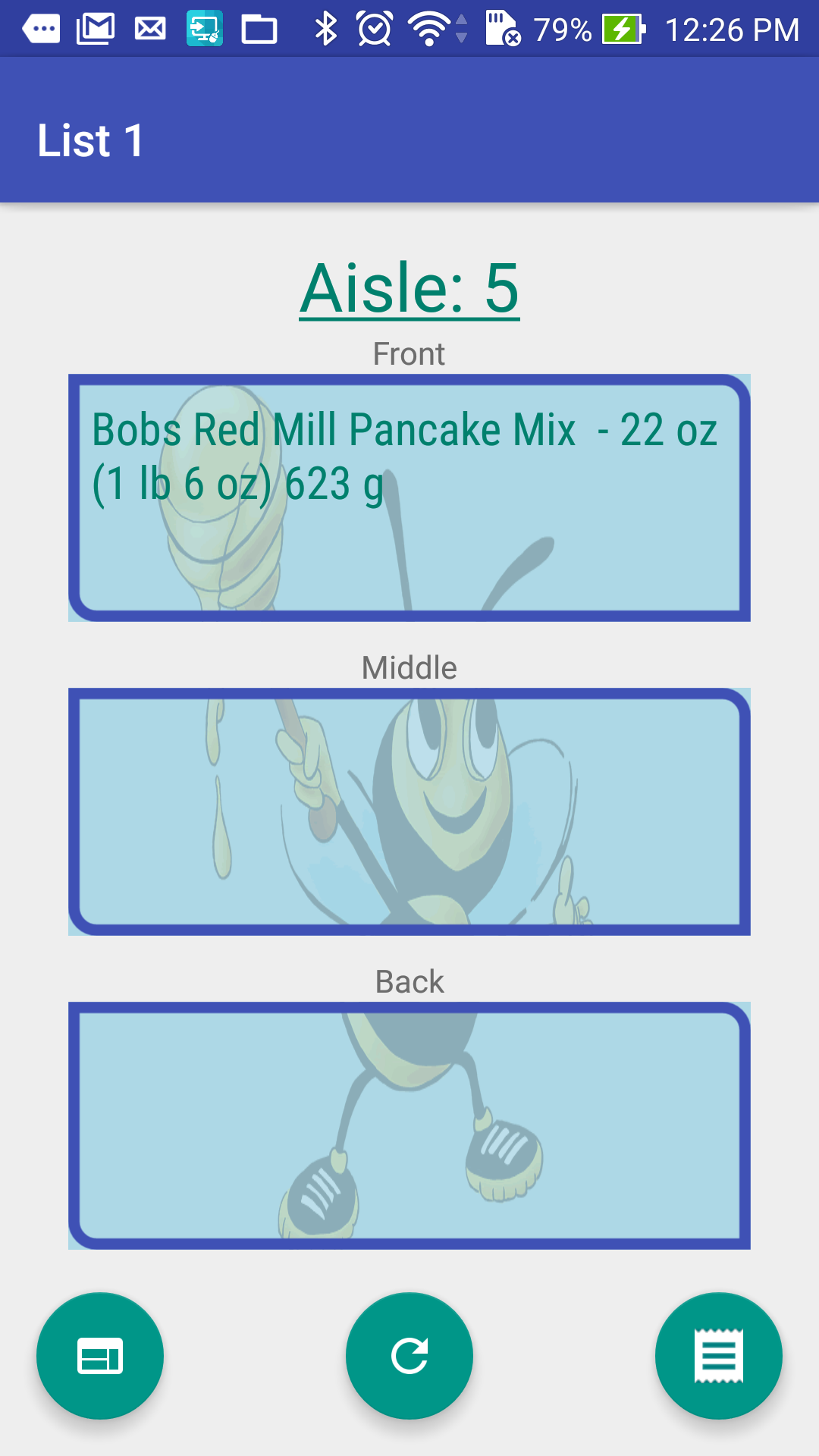
Beeline Shopping

The App

The app was designed such that it was intuitive and easily able to access every feature. It needed to be visually aesthetic while not being distracted by unnecessary features. After logging in with your Beeline Shopping account, you are taken to a List View which displays one of your lists. These two views can be seen in figures 1a and 1b.

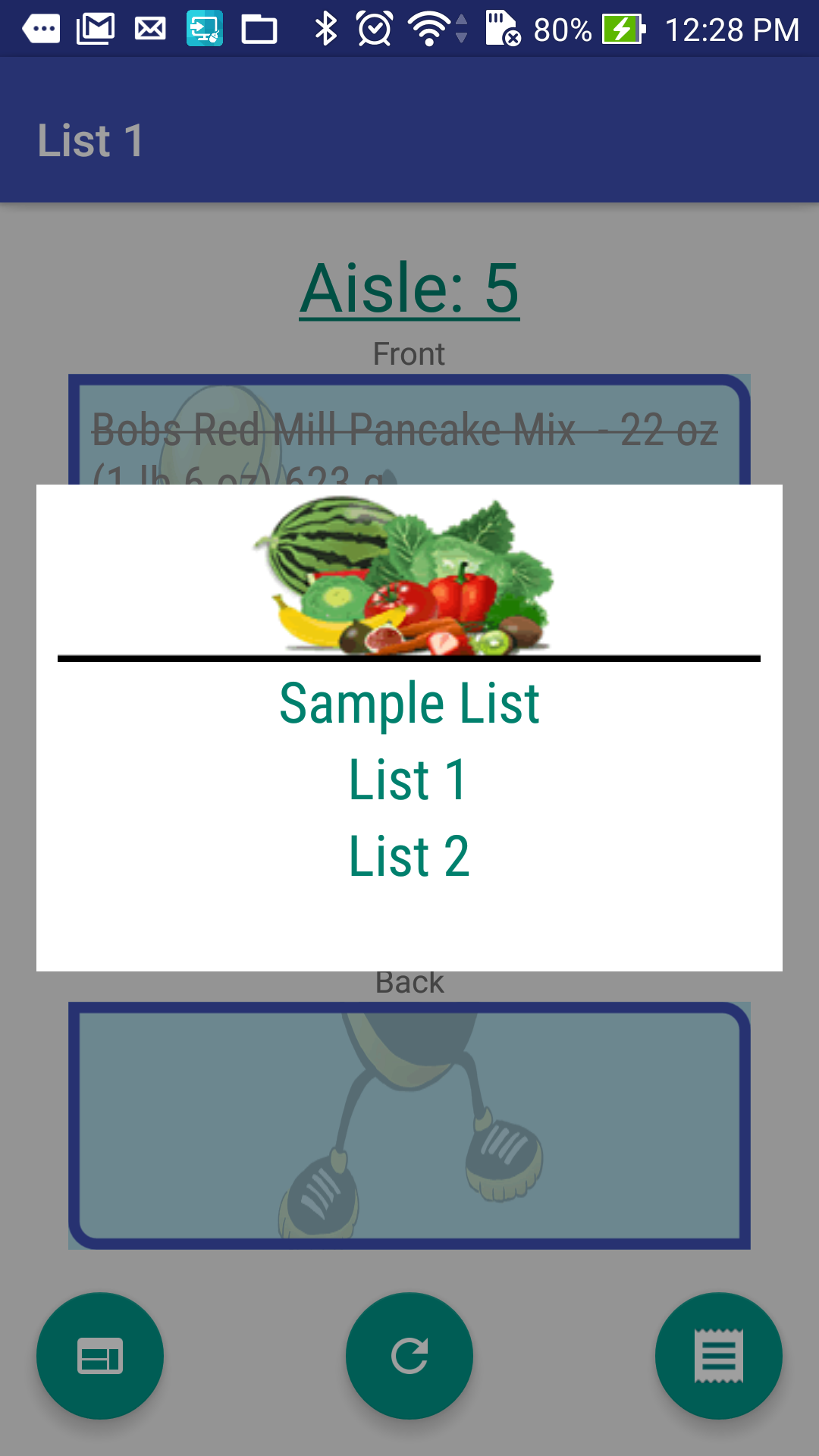
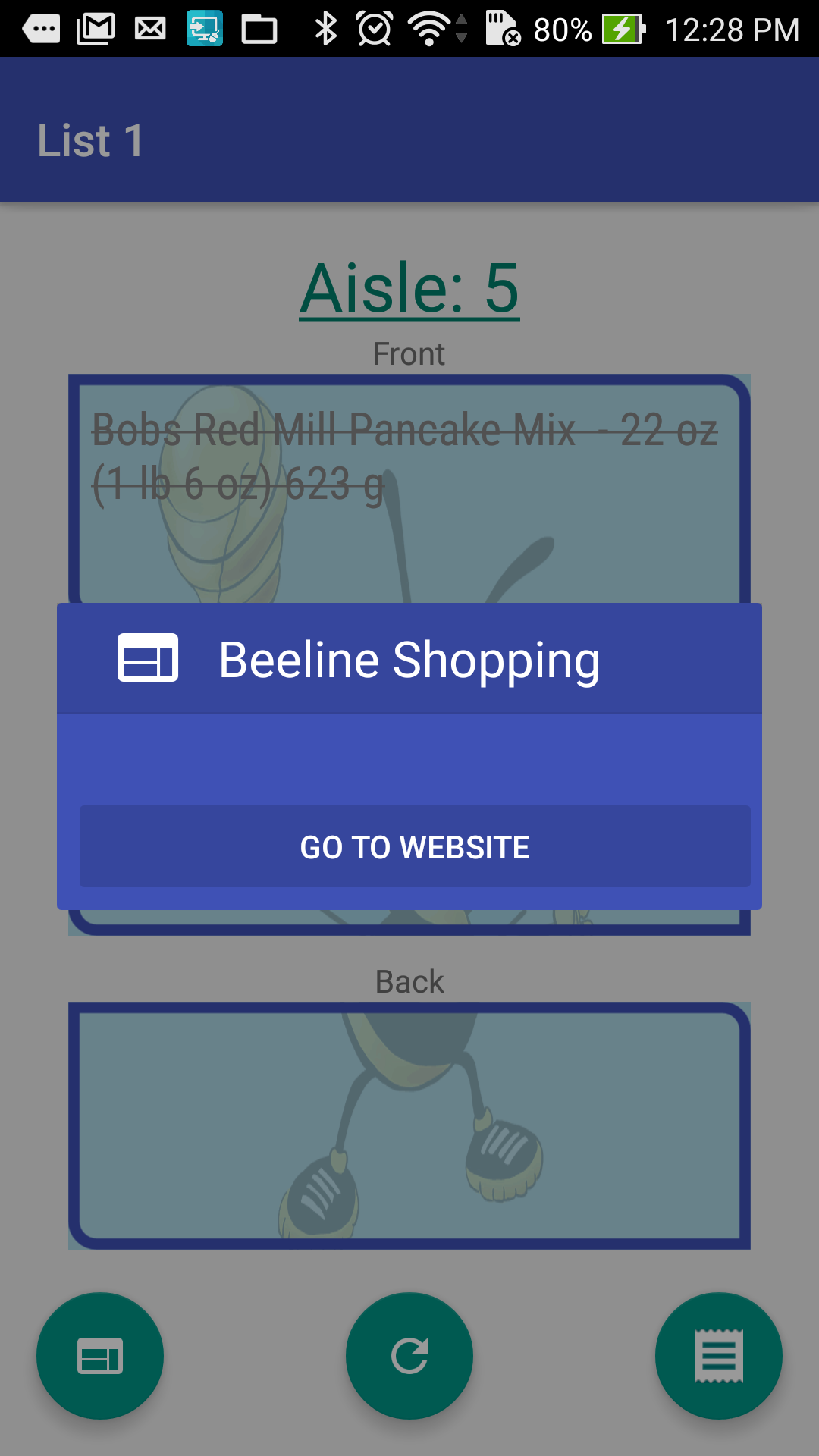




(Figure 1a. Login View) (Figure 1b. List View)

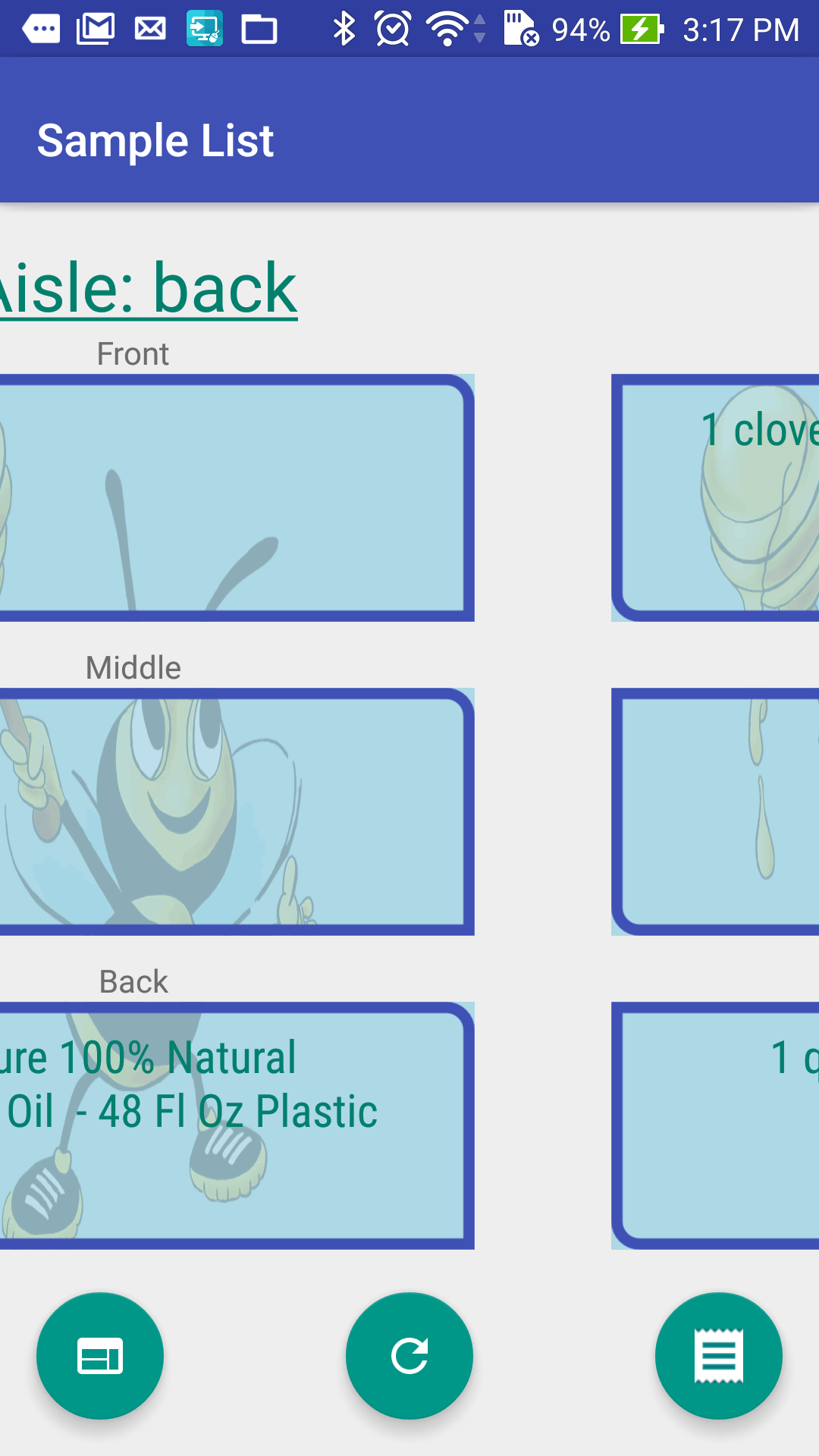
In List View, you can easily see which list is currently displayed and what aisle the current items are located in. The three boxes which hold the items are titled front, middle and back. This is to indicate to the user where in the aisle that item can be found. The bee in that background gives a visual representation by splitting up the parts of the bee; the front, middle and the back.

In this view, you also have access to three buttons. Figures 2a and 2b show the website and list buttons in action. On the bottom left, you have the website button. This button allows you to go to the website and create or edit lists. To the right of this button, you have a refresh button which, when tapped, will uncheck all items; and if any changes have been made to your list, they will be shown. Finally, on the far right, you have the list button. This button, when tapped, will allow you to see all your lists. From the list of lists, you choose the one you want to view and it will be loaded to view on the screen. This button was purposely put on the right side of the screen for easy access, considering it will probably be the most used button.



(Figure 2a. Website Button) (Figure 2b. List Button)

When viewing your lists you can swipe back and forth between aisles as shown in figure 3a.



(Figure 3a. Swiping Action)

This app had a lot of changes from the last version to this particular version. It was concluded that there wasn’t a need for a dedicated List View. As a result, the List View and Items View were integrated. This was done in a way so that the app was still intuitive and easy to use. Another change made was in providing the user the ability to swipe instead of having to use a “next” button. This is a technique that is very widely used and accepted. A website refresh and list button were also added. Also, a logo was used in the background while items, still easily viewable can now be checked and unchecked at user will.  
  
  
  
  
**The Back-end**

In order to add the ability to add entire recipes to a list, we utilized the Food2Fork API which returns a Food2Fork URL for each recipe. The nokogiri gem was then used to grab the ingredients from the HTML document and processed with the Conditional Random Field (CRF) natural language processing library. CRF is a statistically-based algorithm which separates fields into different categories and calculates the probability that a certain portion of a string belongs to a field. In this instance it was trained off of a dataset produced by the New York Times which separates recipe ingredients into quantity, measurement, and name. This allowed for us to match the ingredients from a recipe with items in our database. Given our limited manpower, we were only able to tag the ingredients for the first result when searching for ‘cheese’ under the recipes, as well as the first result when searching for ‘chicken’.

Originally we had hosted our server on Heroku. However, once we added the CRF library, we were unable to continue that service and had to migrate to a DigitalOcean droplet. Although it was much more difficult to setup than a Heroku server it allowed for us to customize our server even more to add necessary features. The new server with the CRF library is located at <http://45.55.5.84/> and is functional in it’s current state.