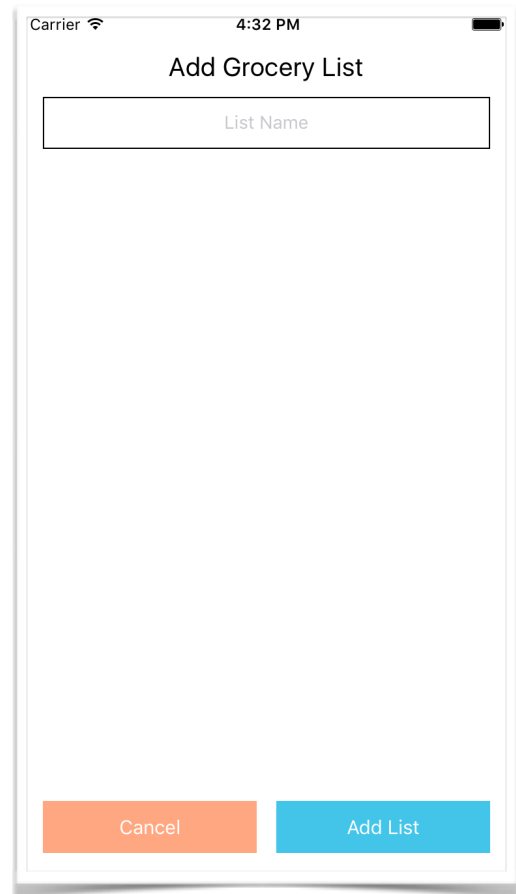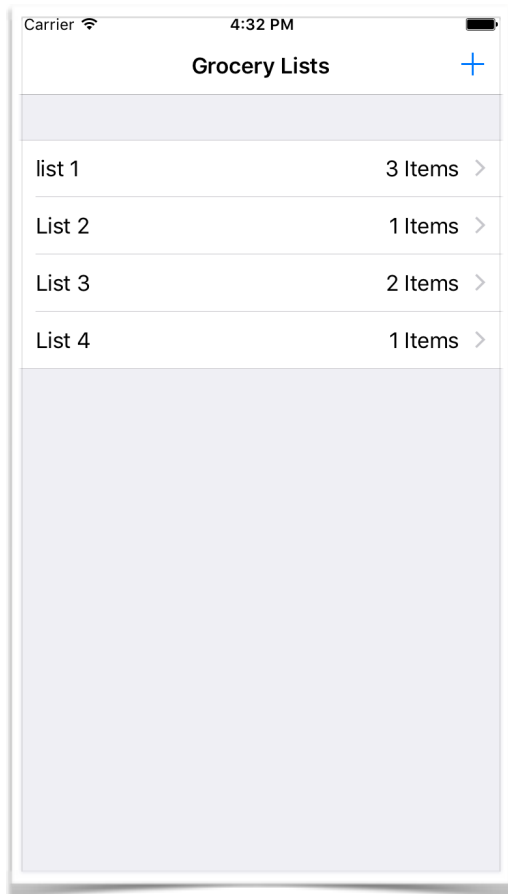iOS Class

Grocery App Design Documentation

Explanation

We will be building a grocery application. The Grocery App holds a user's grocery list data. Each grocery list has a name and a list of items. Each item has a name and a quantity. Users can add grocery lists and associate items with a selected grocery list. This data can either come from web micro services, or be stored locally by using Apple's Core Data. The screenshots below show the basics of the grocery app. There are many other features that can be utilized in this app. These will be a part of the stretch goals.
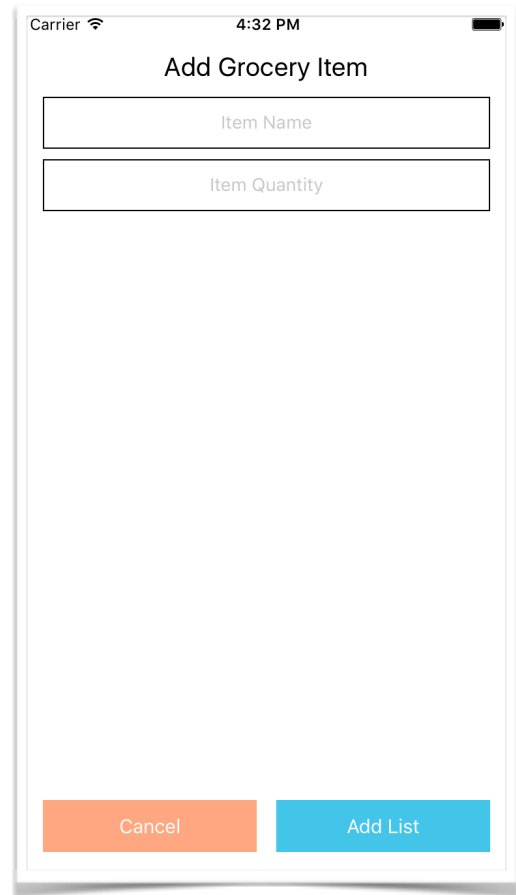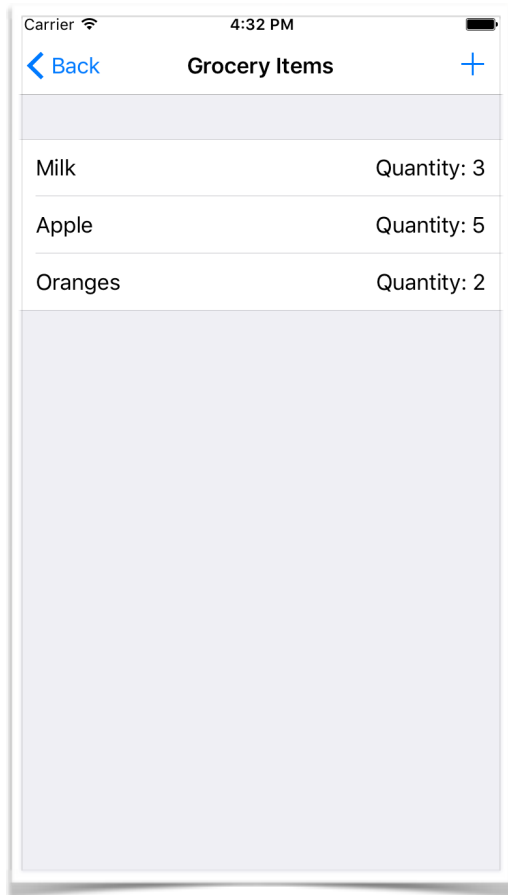
Grocery Lists

Carrier 🗢	4:32 PM	🔋
Grocery Lists		**+**

list 1	3 Items ›
List 2	1 Items ›
List 3	2 Items ›
List 4	1 Items ›

Carrier 🗢	4:32 PM	🔋
Add Grocery List		

List Name

Cancel | Add List

Explanation

The first two screens are centered around grocery lists. The lists are displayed in a table view with the name of the list and the quantity of items within each list. There is a generic title and a plus button the right hand side. This plus button will display the next screen on the right above. This screen allows the user to enter the name of the new grocery list. There is a "Cancel" button which will dismiss the view and an "Add List" button which will create the new list and dismiss the view if successful.

Grocery Items

Carrier 📶	4:32 PM	🔋
‹ Back	**Grocery Items**	+

Milk	Quantity: 3
Apple	Quantity: 5
Oranges	Quantity: 2

Carrier 📶	4:32 PM	🔋

Add Grocery Item

Item Name

Item Quantity

Cancel   Add List

Explanation

The above images are the second two screens within the app. These screens are centered around the grocery items. Just like the lists, the items are displayed within a table view that displays the name and quantity. The second image above shows the screen that allows the user to add grocery items to the selected grocery list. The screen is almost identical, except for the extra text field for the item quantity.

Stretch Goals

- Display a checkmark next to every grocery list and items. Once every item in a list is marked with a check, the list itself will be complete. This data will then persist on the backend (Web Services / Core Data)
- Allow Grocery Lists and Grocery Items to be editable. This will require the user to be able to select a grocery list for editing. Once edited the data will then persist on the backend.
- Allow the user to use a quantity like weight or gallons instead of generic quantity.
- Hold onto the list of items and then calculate the rough price of an entire grocery list.