CS 1998 Project 2

Due Sunday 3/8,11:59pm

Congratulations on finishing Project 1! You have successfully learned basic Swift syntax and learned about debugging! For Project 2, we will create a more complicated app using **programmatic AutoLayout** with **NSLayoutAnchor** to dynamically layout subviews. Remember the power of breakpoints and debugging tools in Xcode when doing this project. If you have any questions, ask away on Piazza. The challenge problems are optional and are there if you wish to impress us or refine your skills!

The Problem

Your task is to complete a grocery list app! A user with this app will be able to input a grocery item and quantity then add it to their grocery list and see their grocery list displayed. Feel free to get creative as you want (implementing some of the extras or any other way you want to improve your grocery app!). The app should have the basic functionality described below and shown in **Figure 1**.

Feature #1: There should be an input field for a grocery item and quantity of item.

Feature #2: There should be a button to add the (item / quantity) pairing to the grocery list.

Feature #3: There should be some display of the current grocery list.

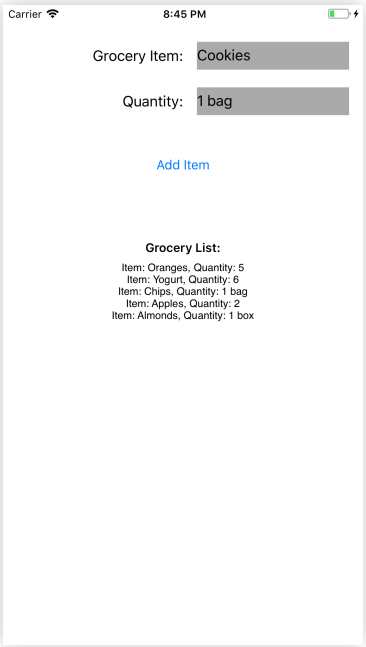

Figure 1: Example Screenshot of Basic Functionality

Challenge Problems

Below are a set of extra features you could add to your app. They are given in order of ascending difficulty.

Challenge #1: Create data validation! Notify users with a message when the inputted data (item and quantity) is invalid (if the name and/or quantity of the added item is empty when inputted, a user tried to add an item with the same name, etc.). Do not let the user add the item if the data is invalid.

Challenge #2: Create a switch that only allows a user to add an item when the switch is active (like an app version of a child lock)!

Challenge #3: Display the grocery list sorted by item name alphabetically (both ascending and descending). Add a segmented control in order to switch between these options!

Challenge #4: Enable a way to remove an item (a specific item, not just the first or last) after it is added!

Reference

For this project, you'll need to figure out a few things on your own. One of the best ways to do this is to be able to lookup the *class references* of any class you need to learn more about. Pay attention to the difference between UITextView and UITextField. Some that may be particularly useful:

For base functionality:

- NSLayoutAnchor Class Reference
- UITextView Class Reference
- UITextField Class Reference
- UILabel Class Reference
- UIButton Class Reference
- The Swift Programming Language (available on the iBooks Store or Google)

For challenge problems:

- UISwitch Class Reference
- UISegmentedControl Class Reference

Academic Integrity

All University-standard Academic Integrity guidelines should be followed. This includes proper attribution of any resources found online, including anything that may be open-sourced by AppDev. The University guidelines for Academic Integrity can be found here.