SP-5 Orange Dynamic Grocery List

CS 4850 W02 Senior Project Spring 2023

Chris Nguyen, Jade Timothy, Justin Wade

Project Owner: Sharon Perry

April 29, 2023

Source Code:

https://chrisngucs.github.io/sp2023-Grocery-App-Website

Presentation Video:

https://drive.google.com/file/d/13LvJUo5TkaH7TANNYEcys-05JRE9Vddf/view?usp=sharing

Table of Contents

Introduction & Project Plan

Abstract A grocery list app built with React Native that allows users to easily manage their shopping lists on their mobile devices (iOS & Android). The app features a clean and intuitive user interface that allows users to customize and categorize items for efficient shopping. The app also includes a variety of useful features such as the ability to share lists with family and friends. Additionally, the app utilizes the power of Firebase to provide a smooth and responsive user experience.

Project Goal To effectively design an application that allows users to access a customizable grocery list that enables users to edit, organize and share lists with others.

Gantt Chart & Version Control

Project Name:	Dynamic Grocery List (SP-5-Orange)																		
Report Date:	4/29/2023																		
					Milestone #1				*Milestone #2*				*Milestone #3*				Final	
Deliverable	**Tasks**	**Complete%**	**Current Status Memo**	**Assigned To**	**01/27**	**02/03**	**02/10**	**02/17**	**02/24**	**03/03**	**03/10**	**03/17**	**03/24**	**03/31**	**04/07**	**04/14**	**04/21**	**04/28**
Requirements/Design	Create project plan	100%	Complete	Justin, Jade, Chris	5	10												
	Define requirements	100%	Complete	Justin, Jade, Chris	5	10	10											
	Review requirements	100%	Complete	Justin, Jade, Chris			5	5										
Project development	Define tech required	100%	Complete	Justin, Jade, Chris			5	10	5									
	Website design	100%	Complete	Chris				10										
	App design	100%	Complete	Justin, Jade				5	15	10	5	10						
	Database design	100%	Complete	Chris				5	10	10	5	5	5					
	Network design	100%	Complete	Justin					5	10	5	5						
	Develop working prototype	100%	Complete	Justin, Chris						10	15	5	10					
	Test prototype	100%	Complete	Jade							5	15	5					
Final Product and Report	Review prototype design	100%	Complete	Justin, Jade, Chris									10					
	Rework requirements	100%	Complete	Justin, Jade, Chris									5	10				
	C-Day submission	0%	Cancelled	Justin										~~5~~	~~10~~			
	Test product	100%	Complete	Jade										5	20	5		
	Complete report draft	100%	Complete	Justin, Jade											5	20		
Final Submission	Presentation preparation	100%	Complete	Justin, Jade, Chris													10	10
	Website preparation	100%	Complete	Chris												5	10	
	Poster preparation	0%	Cancelled	Jade												~~5~~	~~5~~	
	Final report submission	100%	Complete	Justin, Jade														10
			Total work hours	**405**	**10**	**20**	**20**	**35**	**35**	**40**	**35**	**40**	**35**	**20**	**35**	**35**	**25**	**20**

Legend
Planned
Delayed
Number Work: man hours

Version Control GitHub was utilized to develop this application. Group members communicated through Groupme, iMessage, and Discord.

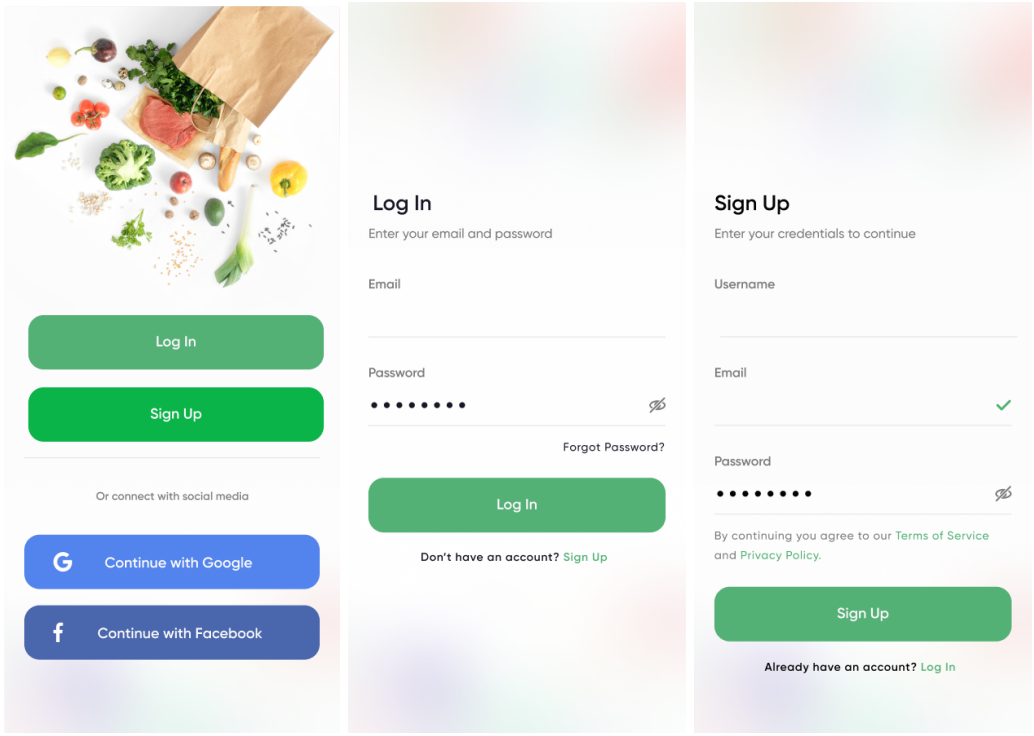
Software Requirements

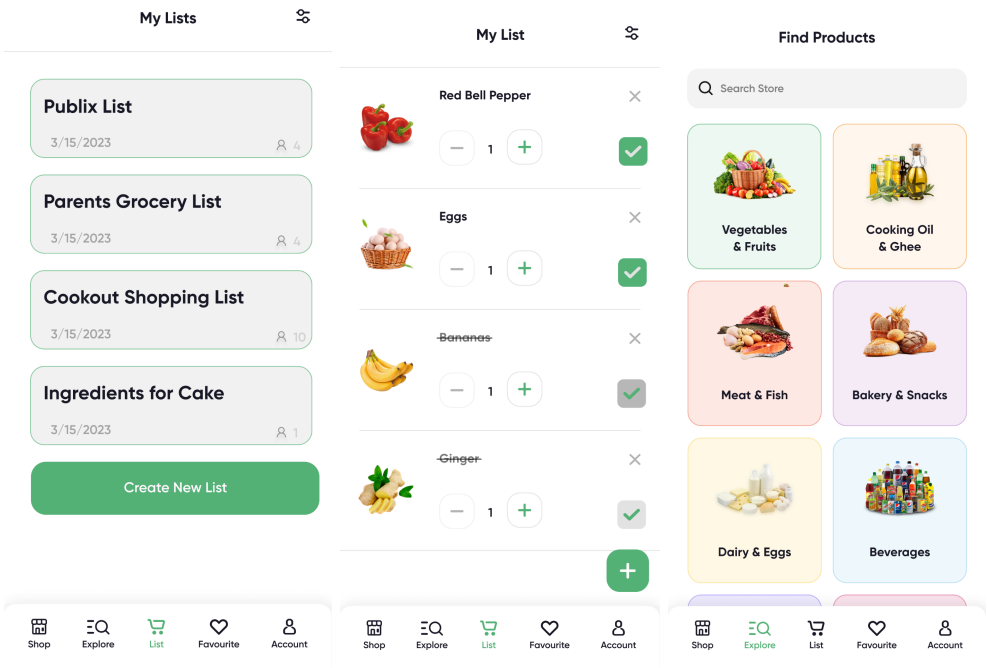
- Startup screen while launching
- Login screen
- Account creation screen
- Authentication
- Forgot password screen
- Verification email (once user signs up)
- Home screen shows an organized view of grocery lists with option to create new lists
- Users can create and add other users in group
- All users that are in group can access and edit grocery list
- In-app text editor for lists
- Sharing screen with information about who the document is shared with and functionality to add or remove users
- Create, delete, and edit list entries
- Sync list entries with all users
- Add and remove users to a list
- Update list live when list collaborators edit lists
- User authentication and registration
- Store user account info

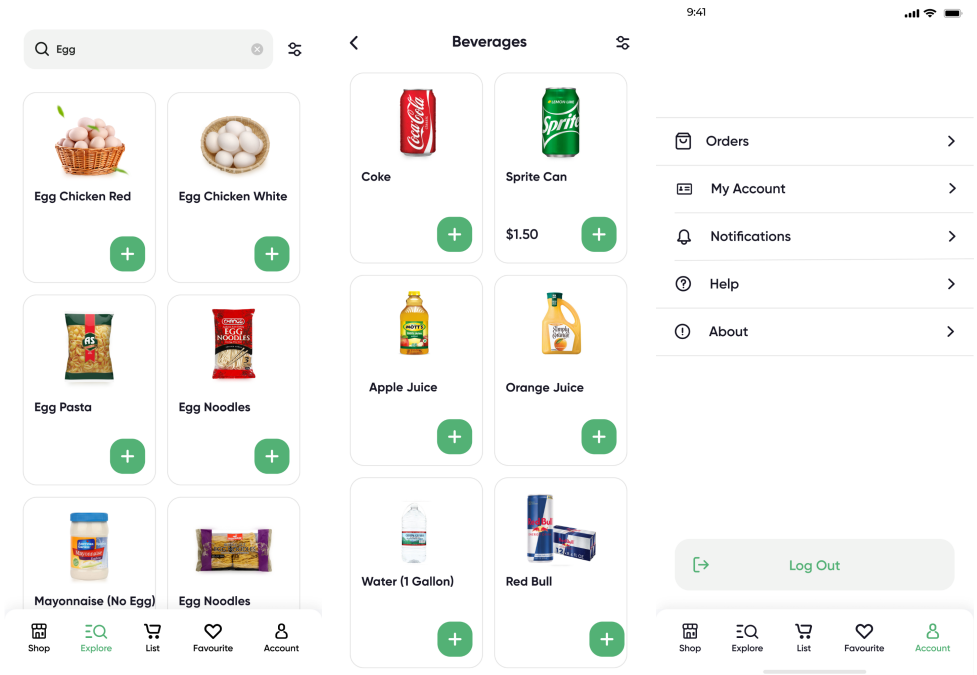
Test Report

Functions	Description	Test Outcome
Create account	Check if new account is created	*PASSED*
Create list	Check if list is created	*PASSED*
Add members	Check if users can be added to list	*PASSED*
Customize list	Verify that items can be added and removed on list	*PASSED*
Verify password	Authenticate account	
Third Party Application	Ability to login through third party application	
List update	Verify list updates in a timely manner	*PASSED*

Mockups


Log In
Sign Up
Or connect with social media
Continue with Google
Continue with Facebook
Log In
Enter your email and password
Email
Password
Forgot Password?
Log In
Don't have an account? Sign Up
Sign Up
Enter your credentials to continue
Username
Email
Password
By continuing you agree to our Terms of Service and Privacy Policy.
Sign Up
Already have an account? Log In


My Lists
Publix List
3/15/2023
Parents Grocery List
3/15/2023
Cookout Shopping List
3/15/2023
Ingredients for Cake
3/15/2023
Create New List
Shop
Explore
List
Favourite
Account
My List
Red Bell Pepper
Eggs
Bananas
Ginger
Find Products
Search Store
Vegetables & Fruits
Cooking Oil & Ghee
Meat & Fish
Bakery & Snacks
Dairy & Eggs
Beverages


Egg
Egg Chicken Red
Egg Chicken White
Egg Pasta
Egg Noodles
Mayonnaise (No Egg)
Egg Noodles
Shop
Explore
List
Favourite
Account
Beverages
Coke
Sprite Can
$1.50
Apple Juice
Orange Juice
Water (1 Gallon)
Red Bull
9:41
Orders
My Account
Notifications
Help
About
Log Out
Shop
Explore
List
Favourite
Account

Development Discussion

The first stage of our primary development involved selecting and familiarizing ourselves with the technologies to be used. We chose to use React Native for several reasons. First, its cross-platform nature simplified development and made it extremely simple to run on multiple operating systems. Second, it's widespread popularity in app development meant that there were numerous resources for learning it and implementing what we needed to implement. Thirdly, it being JavaScript-based was an advantage as we were already familiar with JavaScript, reducing the learning curve and making development easier.

While we were familiar with JavaScript, none of us had prior experience with React Native. In the initial stages of this project, we each independently began learning React Native using a combination of online guides, YouTube tutorials, and small practice projects. We also looked at other publicly accessible software applications created with React Native to see how they designed the of architecture their programs. Because of this, we were all able to gain passable familiarity with React Native before beginning serious development.

Once we settled on React Native, our next obstacle was how to emulate our application throughout the development process. There are several ways to do this, but one of the most popular is the Expo platform. Expo allows you to take a React Native project and run it on a device with the Expo client installed or an emulator. For instance, one way we tested our app was by using Expo to run it in an emulated Google Pixel phone through Android Studio. We also ran the application on a personal phone with the Expo app when demonstrating it. Expo turned out to be an incredibly helpful and versatile tool throughout our development. One useful feature that Expo offered was Expo Router. We were able to use Expo Router to manage navigation throughout the app. This gave us an easy way to map our architecture, as it used a different JavaScript file for each screen within the app. For instance, in intro screen the "Log In" button is mapped to the JavaScript file corresponding with the "Log In" screen. This made organization very simple.

The last major technology that we needed was a framework to manage the cloud aspects of the application, including storing lists in a database and managing user data. For this, we chose to use Cloud Firestore, a cloud database that is a subset of the more commonly known Firebase. We chose to use Firestore for a few reasons. As with React Native, its prevalence meant that there were numerous resources for learning and using it. Second, its versatility and scalability were both important selling points. We didn't

have to worry about using multiple services for different parts of our app, it could do all of our backend by itself.

Once all of our major building block technologies were defined and in place, our main concern was in the design of the application and its features. We chose to have each list be a collection of editable items of tech. One of our original ideas was to have the user be able to add items from a list of predefined grocery items. Our plan was to have a database of possible items, with each item having an associated image and possible price. However, relatively early in the development process we decided that this was not an effective route to take. The added complexity of having to have a database of possible items seemed excessive and not worth it. It limited the users choice for relatively little payoff. We felt the simplest design was to have the user be able to add and name each item, allowing them to enter in whatever they wanted with no limitations. This gave them enough freedom while not being to freeform. We didn't want to simply have an empty shared document, as there are other services for that and it doesn’t fulfill the specific needs of a grocery app. Therefore, we chose the option of an editable list as the best choice with the most reasonable tradeoffs.

Challenges and Assumptions

A lot of the challenges with creating the app was due to firebase and the lack of documentation for Firebase v9. Recently, Firebase had a large update to it and a lot of the documentation that was created became obsolete so a lot of the way the storage worked wasn't ideal. Originally it was planned when a list was made it would have a couple of strings attached to it containing the id and name as well as two collections inside the list containing the items and members. In Firebase v8 that was easy to implement, but with the new syntax for Firebase v9 it was much more difficult and there wasn't much documentation about this specific use case.

Another challenge we had was debugging the code. React Native doesn't really have a good debugger tool so it makes it difficult to find bugs within the code. Other languages or frameworks would tell you what line or what function causes the code to not run properly, but with React Native it gives a lot of errors inside of the libraries used not the code itself.

User Guide

When users are first opening the app there will be three options that appear the first being log in and the second and third options being to sign up with either email and password or google. After the users sign up they can now log in and view or add lists that they want to create. At the bottom will be a text box and a plus icon that will add the list to the database. Once a list has been created it is viewable from the lists screen and there will be a plus icon on the right side of it. Once pressed it will change to the individual list screen that will show all list items. This screen has the same functionality as the last screen with the same text input box and a button to add list items. This time however there will be a checkmark next to each item that will check the items off and remove them from the list. Finally in each list the top right icon will copy a link to the user's clipboard containing a shareable link that once another user uses they will also have access to the list and are able to make changes to that list.