

The SmartShoppingList application is an application meant to showcase all the various things I've learned about coding such as polymorphism, UI development, and collections; While allowing me to hone my skills in mobile development. It was inspired by Walmart's catalog system, and is a shopping list app capable of finding an items features through web scraping, automatically sorting them in to categories, checking availability it stores, and allowing the user to make custom lists that can be fitted with recurring alerts.

Day1 (6hrs): The first day was spent writing out a plan of the app (included bellow), including all the features, Main app classes, and the basic layout of the apps UI. Along with researching the best ways to implement certain features such as the barcode scanner; Settling on using the zbar and opencv libraries due to them being open sourced and well documented for my needs; And brickseek.com and barcodespider.com for sourcing the data.

Day2&3 (13hrs): The second and third day were spent learning and researching aspects of Android studio and getting well acquainted to its features. The Theme and UX of the app was also researched, with the final decision being a dark theme with a simple bottom navigation bar with 4 page fragments.

Day 4 (7hrs): On the fourth day I was able to finish a good base for the application fragments and bottom navigation bar; With the listed items, all items, and user stores fragments all using recycler view to display a list of their respective elements, and the add item fragments using a full screen fragment to latter display the camera. However I wanted the application to have a more fitting top action bar than the default one, so I began researching how to accomplish this.

Day 5 (7hrs): During day seven I continued my research in to customizing the action bar and found a few methods. The first was to switch to a noActionBar theme in the styles and set your own, but doing so would cause the program to crash since the bottom navigation bar relied on functions relating to the action bar; So I decided the best way to accomplish this would be to hide the action bar in main activity and build my own to display. My first attempt was to add the bar in each fragment so that each would be able to simply load it in when they populate the main fragment display. However attempting to do this would cause an exception where the recycler view would not be found since it needs to be the only element in the fragment; So I simply ended up adding a second frame viewer in main activity that would display the custom action bar

Day 6 (7hrs): The sixed day I styled the action bar, recycler view rows, and font; Adding a settings button and an option button to the action bar, and moving around text boxes to better fit item names in the recycler view rows. Along with changing the font to serif, and adding the ability to check of unneeded items from active lists.

(REST OF PROJECT IN PROGRESS)

## Shopping Organization App:

### Features:

- Store all your usual recurring items via search or bar scan;
- Pick you favorite stores/locations;
- Organize your items in different categories like kitchen, wish, food, bathroom etc..
- Have an optional timer to remind you to get more of an item on your next shopping trip.
- Use a web Scraper to find out in store location, if they're in stock, or if there is an equivalent at the current store you are looking at. Along with any details such as nutrition, and store price in comparison to other stores.
- Save recipes with ingredients that you may need to quickly find at nearest store; Along with optionally recommending recipients for foods you like.
- Along with a select diet feature for seeing if the food fits in to said diet.

+++++

### Basic Design:

- A base item class with a price map, SKU vector, UPC, brand, and description; Along with web scraped online data such as in store location, and #of stock.
- Different Inherited classes such as Miscellaneous, Electronics, Clothing, Home, Garden, Baby, Toys, Fitness& Outdoors, Video Games, Movies, Home Improvement, Pets, Beauty, Health, Auto& Industrial, and Arts& Craft; With extra variables such as nutrition info, model number, release date, size, age range, etc..
- An account class containing the username ID, customer name, and password.
- A store classes containing zip code, store address, store number, hours open, and website; For stores such as, Walmart, target, Lowes, home depot, office depot, Macy's, CVS, staples, bjs, -- to website HEB, Sam's

+++++

### GUI Design:

- A dock with 4 buttons for recurring items, add item, all items, and stores.
- The recurring items fragment has a dynamic action bar that can switch between category view, time interval view, store view, and custom list view.
- And the general action bar having a settings and account button.