MetaGrocer

Crowdsourcing Grocery Shopping App

Team

Tim Adamson
Kevin Birrell
Ryan Milem
Lulu Sun

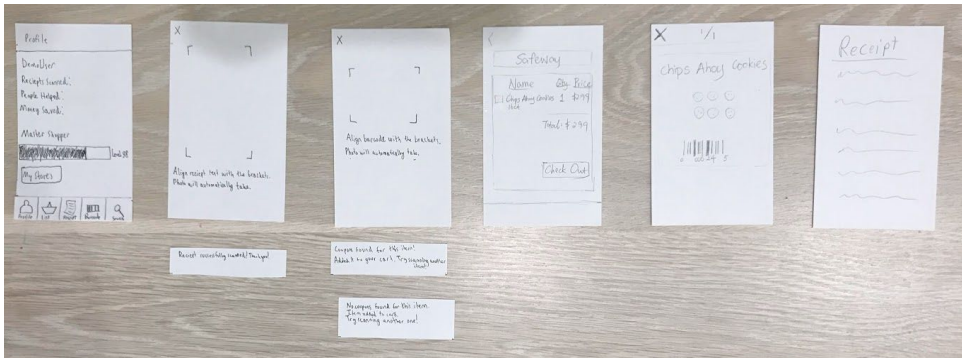
Problem and Solution Overview

Grocery shopping is something that almost everyone has experience with, often of the negative kind. When shopping, it can be infuriating to come into a store and see the prices of the items you want to buy be either more expensive than they were the last time you came in or cheaper (meaning you wasted money). Almost everyone knows that coupons and other ways to save exist, but few people actually go out of their way to use them due to the incredible inconvenience of clipping out physical coupons or printing out PDFs from poorly designed coupon websites. As we found during our research, college students and young people tend in general to not bother with these coupons, even though paradoxically they are the people who could benefit from them the most. While some grocery stores do now have apps that allow customers to save more conveniently, it's not in the interest of the grocery store to make these apps that useful so frequently they are lacking and (of course) only deal with one store at a time.

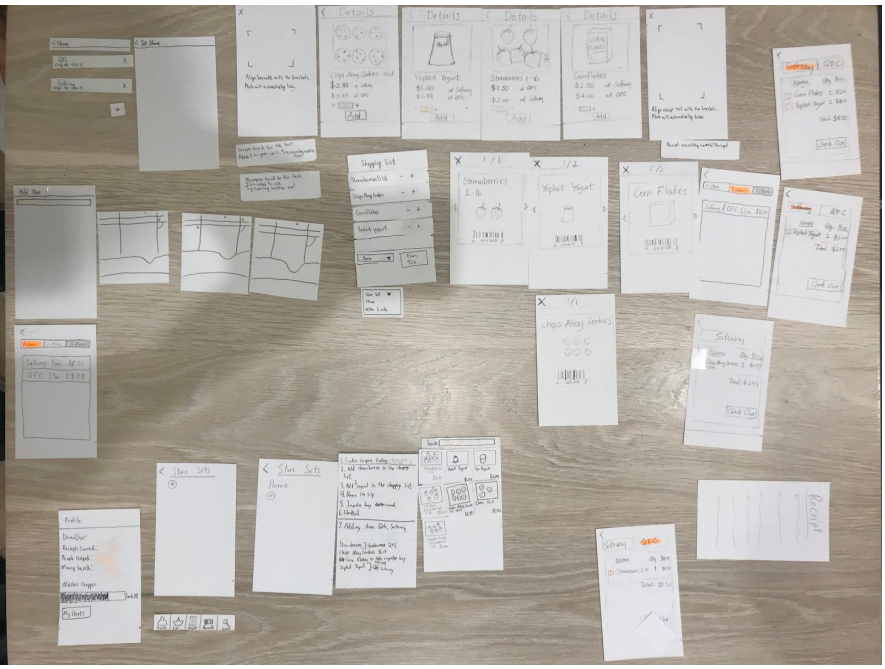
Our solution to this problem, MetaGrocer, is an app that uses crowdsourcing to maintain a database of current product prices and coupons across all major and local grocery stores. By telling our app what items they want to buy, the customer can then receive trip suggestions that allow them to split their purchases across multiple stores in order to save money. Our app's database also allows for quick price and coupon retrieval for any item that a customer might want to buy. This allows the customer to scan items and automatically receive any applicable coupons, while also enabling them to contribute to the database with just a couple clicks.

Initial Paper Prototype

Design / Tasks

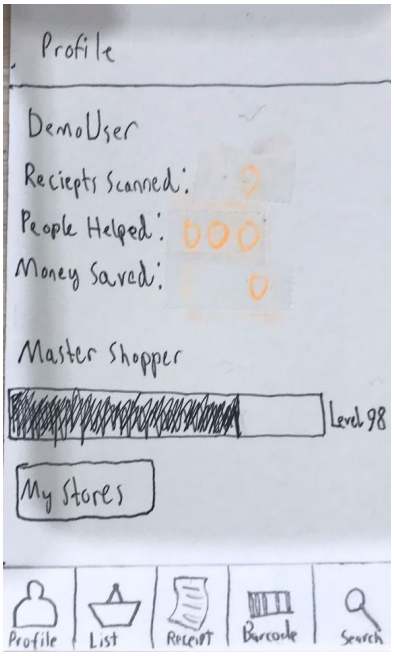


Our initial prototype design was primarily focused on allowing users to save on groceries in a variety of ways (planned / unplanned) without having to spend a significant amount of time dealing with coupons and researching prices at stores. To this end, we implemented a shopping list “trip plan” feature that would convert a user’s wanted items into an easy to read trip plan for multiple stores telling them where to buy which items. Coupons, where applicable, are automatically retrieved and displayed in a manner that the user could scan at checkout from their phone without hassle (and without holding up the line).



While this feature enabled users to perform the task of planning their trip in a way that saves money without wasting much time, we also implemented a scanning feature so that users could make impulse buys in the most cost-effective and time-efficient way possible. By scanning a barcode, a user can see a product’s price at nearby stores and have any available coupons for the item automatically added to the app’s coupon list.

Task 1: Price Comparison

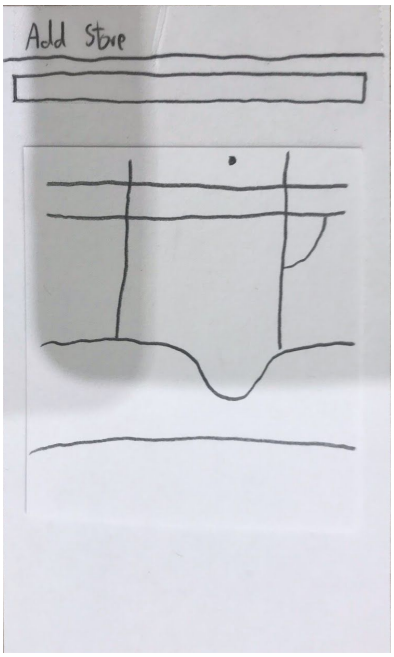


After downloading the app, the user opens it for the first time, and is taken to the profile page. The user clicks on “My Stores”.

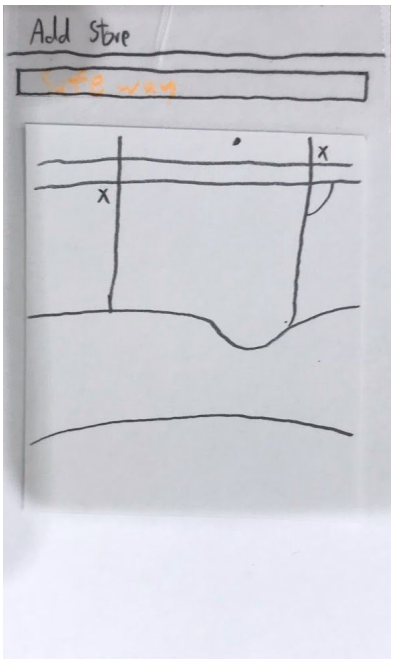
The user clicks on the + to add a store set.

The user types in the set name as “Home”.

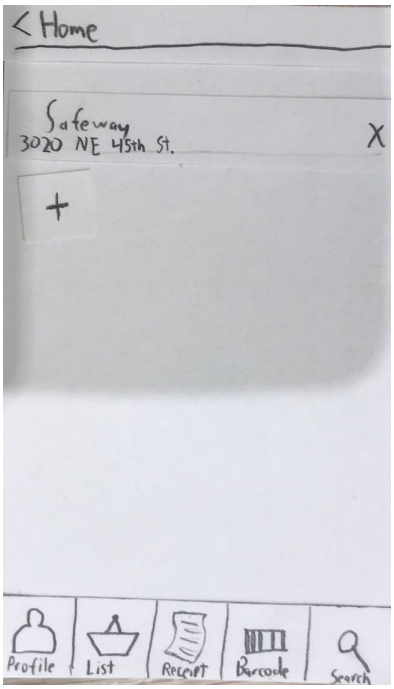
The user then clicks on the + to add a store to the “Home” set.



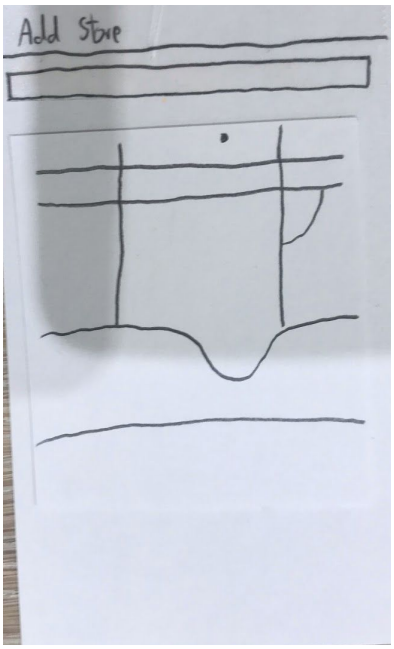
The user clicks in the search bar and types “Safeway”.



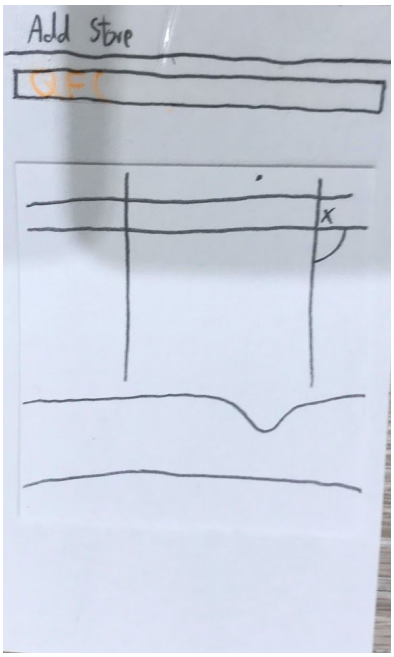
The user selects the Safeway that she wants to shop at.



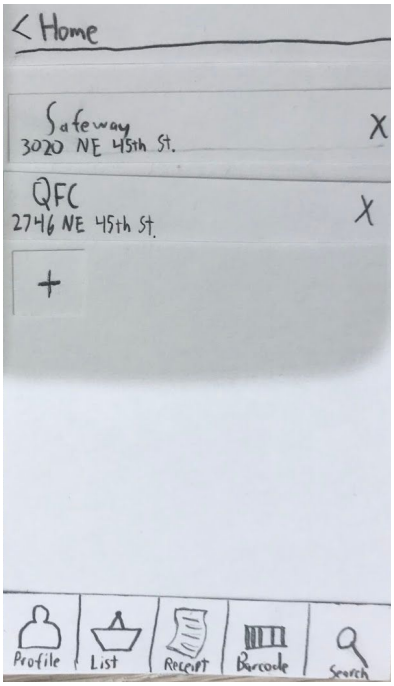
That safeway is now added as one of the stores in their “Home” set. She then selects the + again to add another store to the “Home” set.



Again she sees a map of her current location. This time she types in “QFC”.

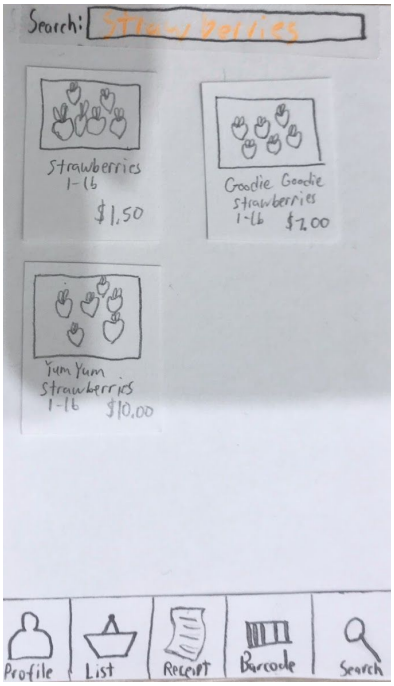


The map shows the one QFC nearby which the user selects.

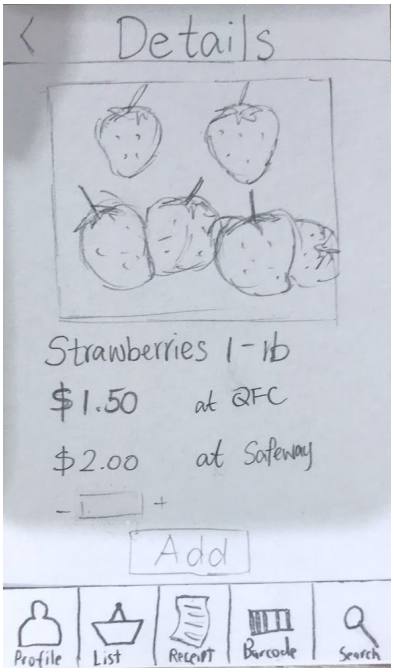
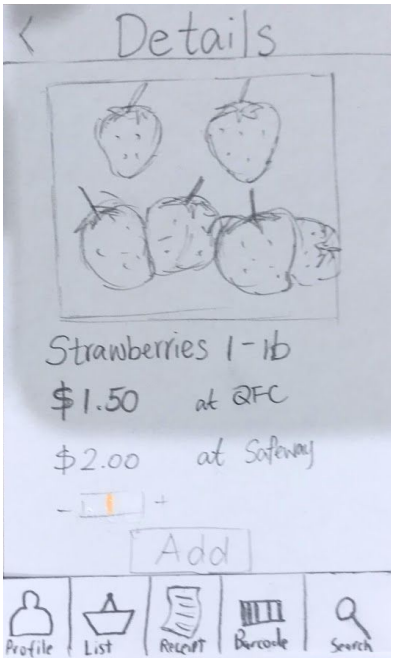
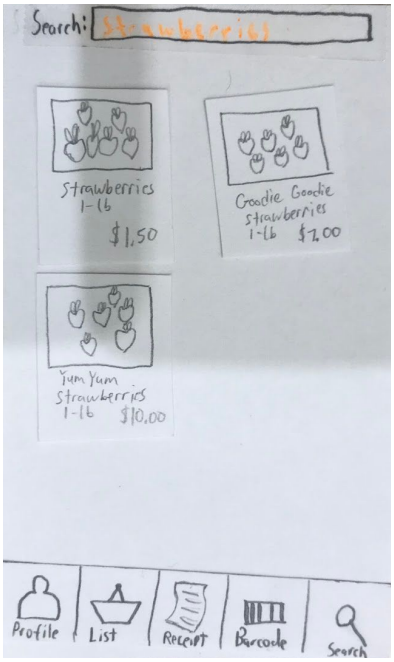
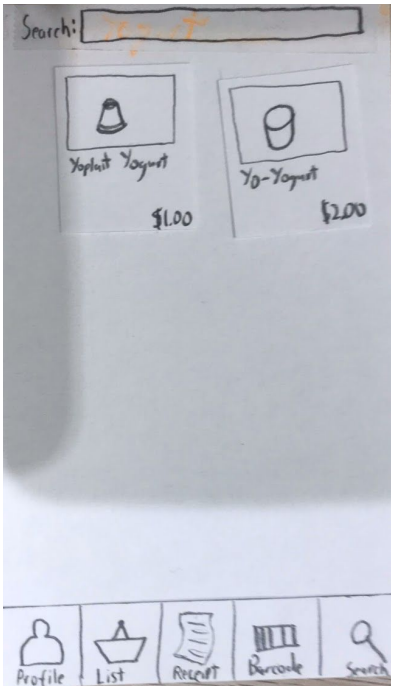
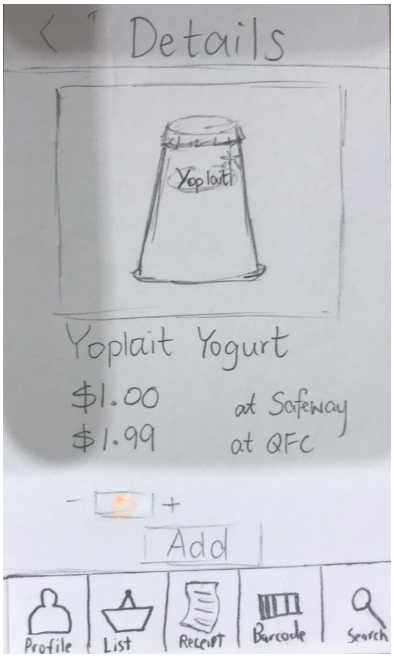
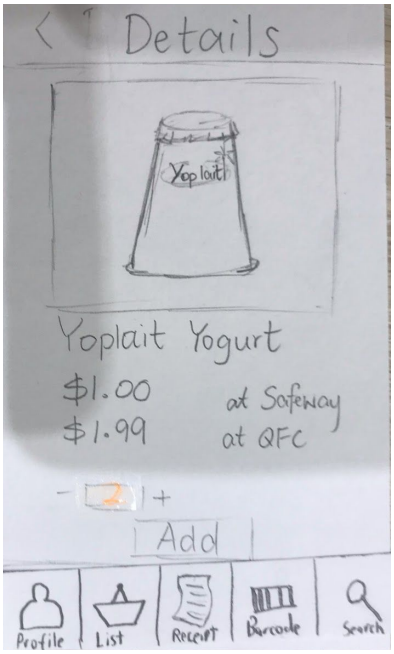


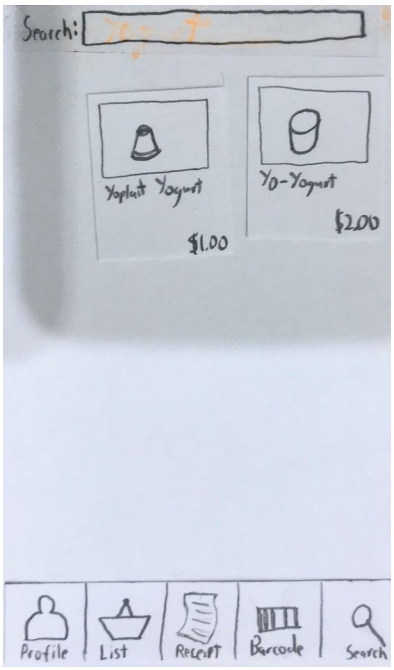
That QFC is now added to the “Home” set as well. The user is now satisfied and clicks the

The user types in “Strawberries” into the item search bar.

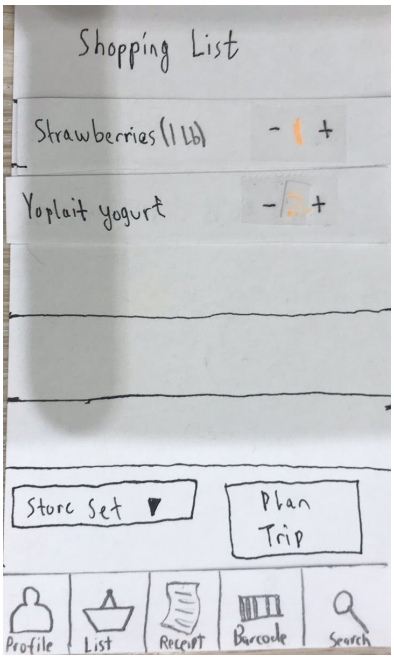


The user selects the cheapest strawberries found.

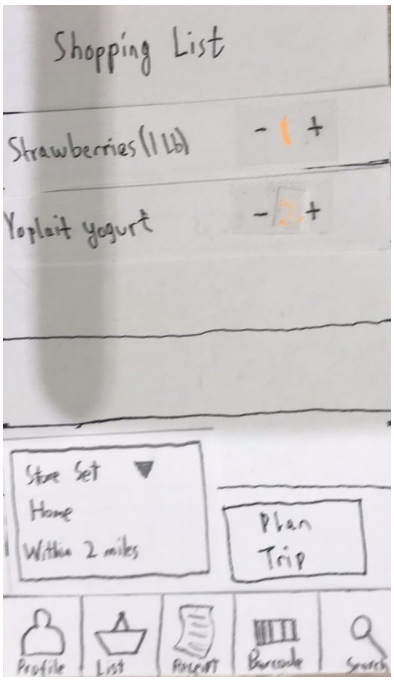
search button to look for an item.		
 The user selects the + to change the add value to 1.	 The user then clicks the add button to add one box of strawberries to the list.	 This then takes the user back to the strawberries page. From here she searches “yogurt”.
 The user sees two types of yogurt for sale and chooses the cheapest one.	 The user changes the add value to 2.	 The user clicks the add button to add two yogurt to the shopping cart.



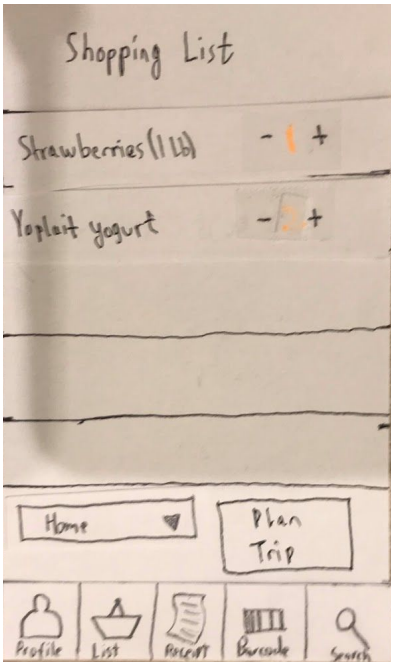
The user is then taken back to the search screen. From here she selects the shopping list on the bottom bar.



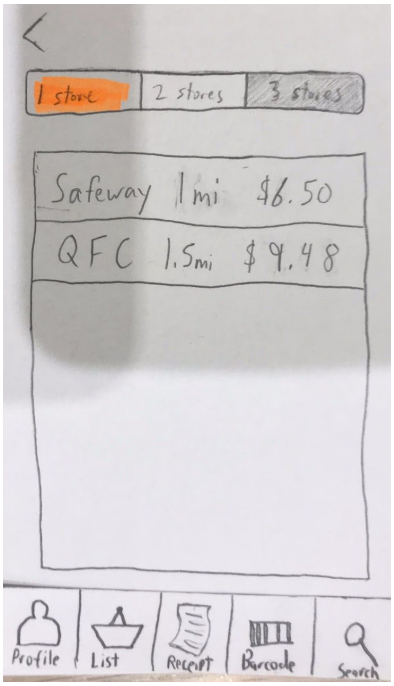
The user looks at the shopping list and selects a store set.



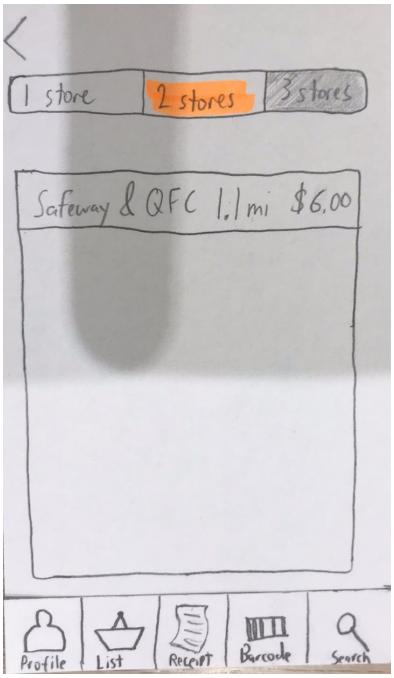
The user selects the “Home” store set made previously.



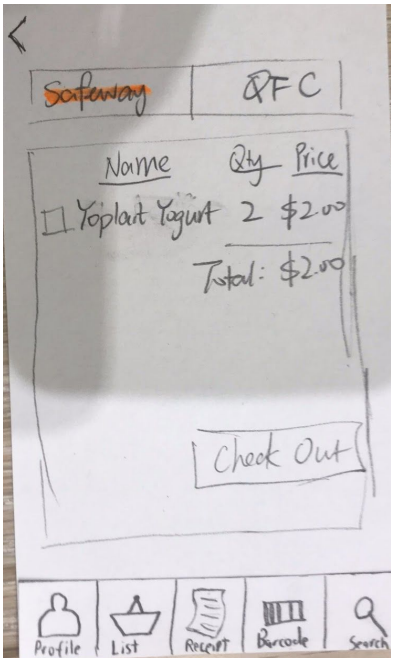
The user then chooses to plan the trip.



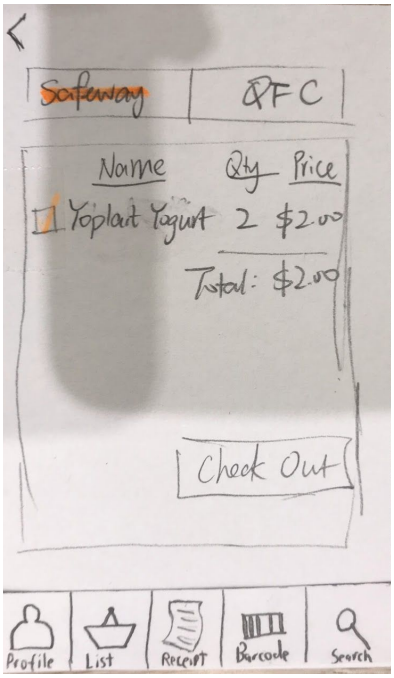
The user is first shown the 1 store shopping options, but also checks out the 2 store option as well.



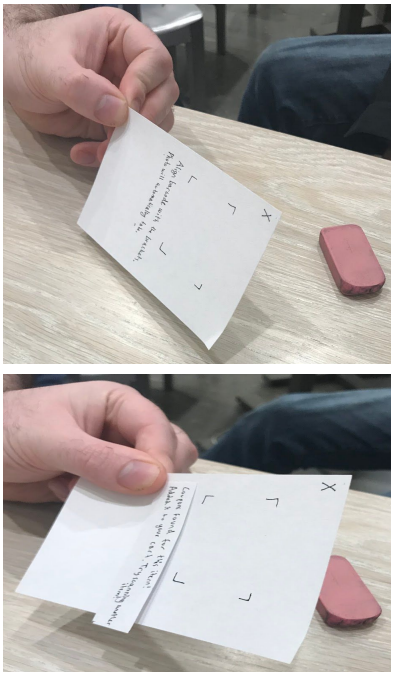
The user sees that the two store option is cheaper, and chooses to shop at the two stores.



The app shows what items need to be purchased at the first store. The user find the yogurt and checks it off the list.

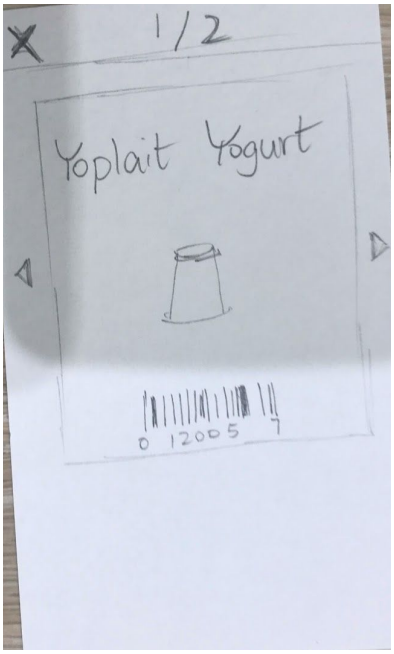


After checking the item off the list, the user find some cereal in the store that she wishes to buy, so she selects the “barcode” button at the bottom.

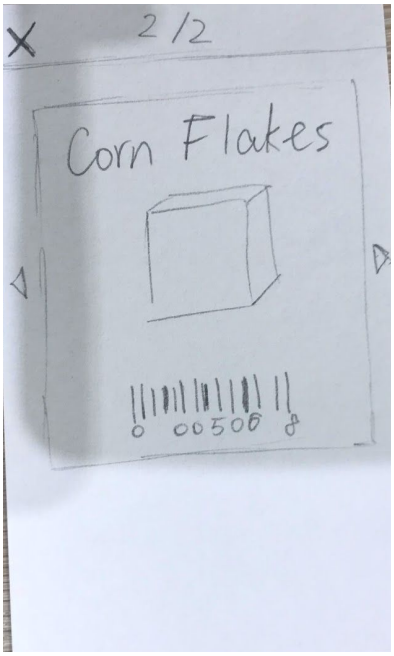


She scans the barcode of the cereal, and a popup tells her that a coupon was added for that item.

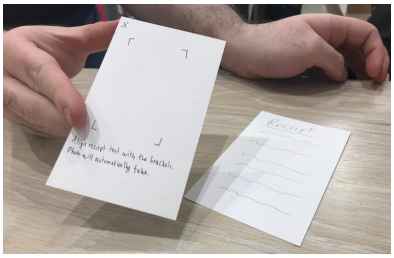
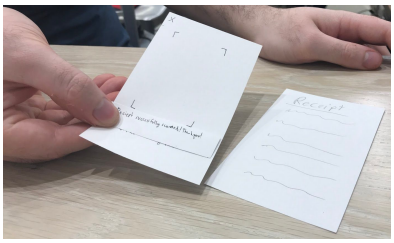
The user is sent to the in-store grocery list from where she can check out.



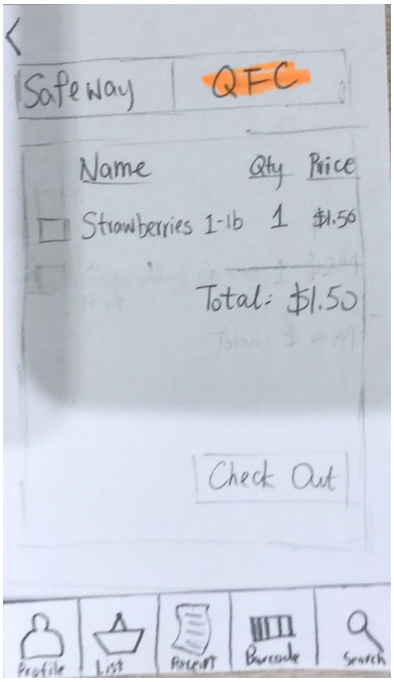
The user then scans the coupons to get the discounts.



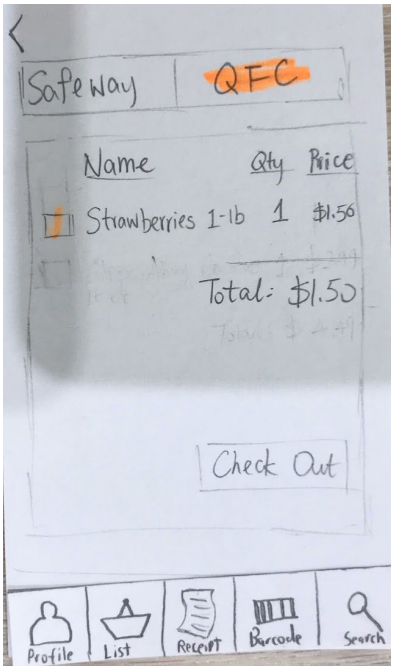
The user swipes after scanning the first item in order to scan the second one.

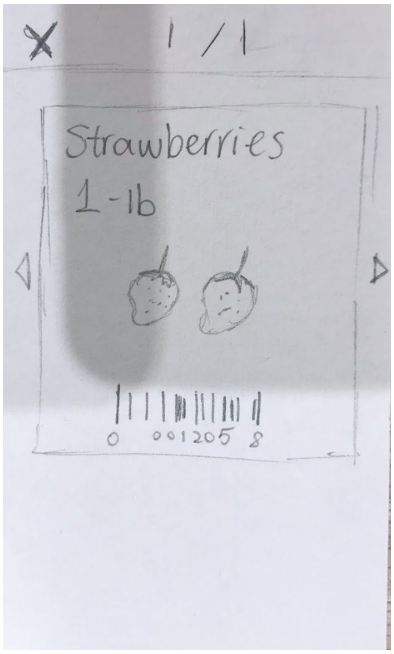
The user then scans her receipt to help future shoppers.



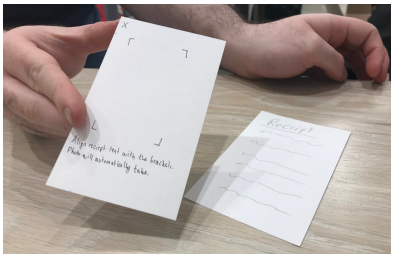
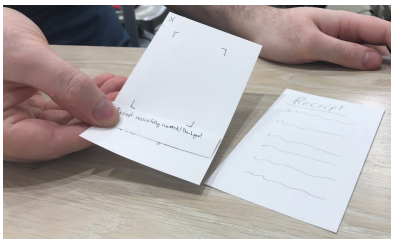
After scanning her receipt, the user is taken back to the in store shopping list, except for QFC now. The user finds strawberries and checks them off the list.



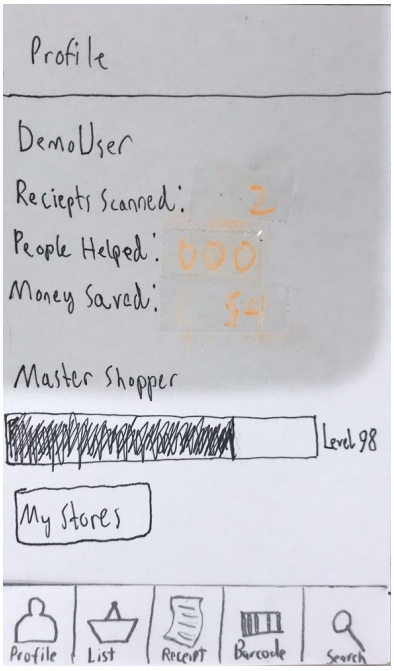
The user is now ready to check out.



The user scans her receipt for the strawberries to get the discount.

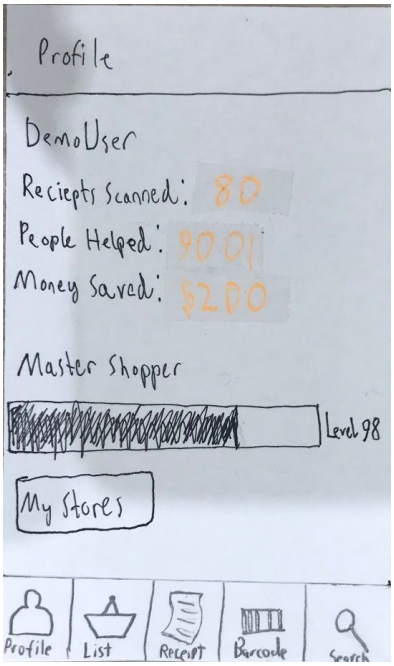
She then scans her receipt to help future shoppers, and to earn in-app points.



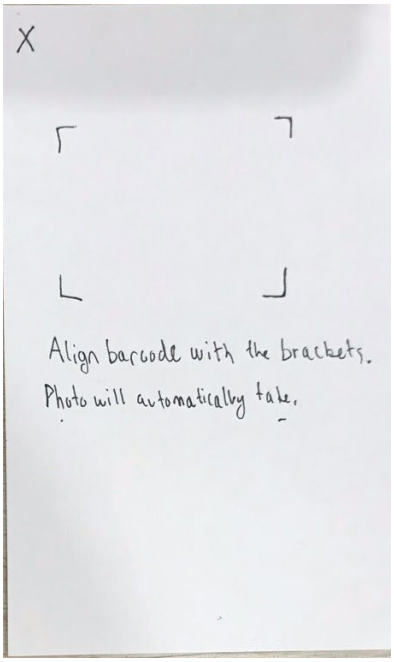
The user is taken back to her profile, where she can see the number of receipts she has

		scanned, and the money that she has saved.

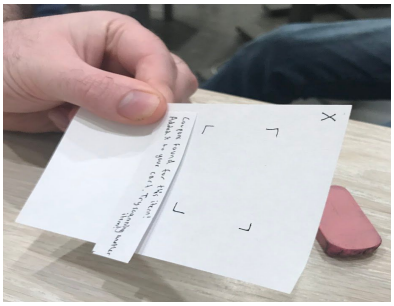
Task 2: Finding Coupons



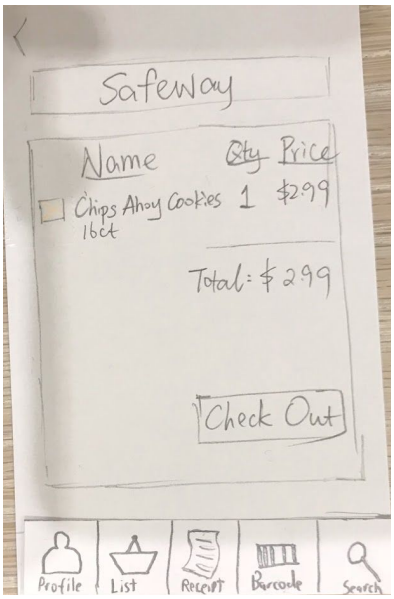
The user opens the app and is taken to their profile, which displays various stats about their lifetime use of the app. They then tap the “Barcode” icon to scan an item that they wish to purchase.



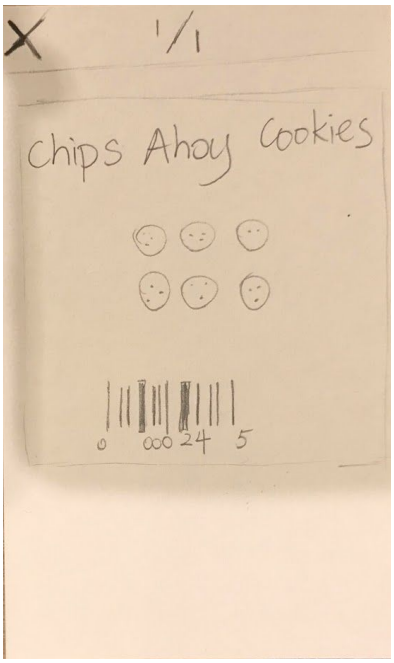
On this screen, the user aligns the barcode of their item with the prompt and a photo is automatically taken and processed.



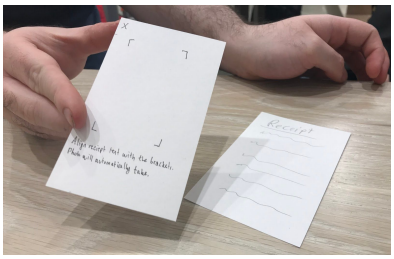
The text is replaced with a message confirming that the barcode was successfully scanned.

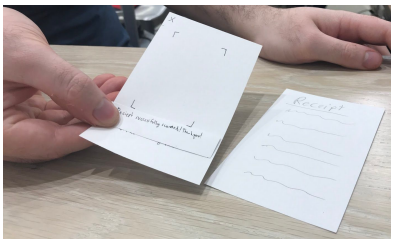


Once the item is processed, the user is taken to a shopping “list” that contains the item they’ve scanned (as well as any others already in their pre-planned list, although there are none in this case). They then press the “Check Out” button when they are ready to check out at the grocery store.



When the user presses the “Check Out” button, they are brought to a screen which shows all applicable coupons that they can scan at the cashier. After they exit this screen, they are prompted to scan their receipts for use in the crowd-sourced price database.





The user scans their receipts in a similar manner to the item barcodes.

Testing Process

Heuristic Evaluations

To gain some initial feedback on whether or not our design violated basic design heuristics principles, we conducted two heuristic evaluations with other groups (the “Social Parking” and “Food Waste” projects) from CSE 440. This essentially consisted of Lulu and Kevin explaining the concept behind MetaGrocer to other project teams and then having them explore the interface in a semi-guided (but mostly free) manner while Ryan “executed” the app and Tim made notes of their heuristic evaluation feedback. After the heuristic evaluation was completed, we then went through the notes and ranked them in order of importance, and noted which part of the interface they pertained to. We used these rankings and prioritizations to make revisions to our interface.

Usability Testing Sessions

After making some changes based on our heuristic evaluations (detailed in the “Testing Results” section), we then conducted three usability tests featuring random UW students as our participants. Our testing methodology did not change much throughout the three tests since the first one went well and ran quite smoothly; however, one significant change we did make is the location of our subsequent tests. To avoid accidentally recruiting students with engineering backgrounds, we decided to move our test location from the CSE building to the Odegaard Undergraduate Library. This worked well, as our second and third test candidates came from a much broader pool of students than primarily CS/EE majors. Throughout the tests, Tim played the role of the “computer” while Lulu took notes and Ryan/Kevin performed a combined greeter/facilitator role. We gained *tons* of useful insights from this part of the design process, and have documented them extensively in the “Testing Results” section of this report.

Refinement

We refined our testing process over the course of the heuristic and user tests by solidifying the roles that we were originally assigned and preparing as much of the test beforehand as possible. During our first user test, we each had different roles than we did during the heuristic evaluation. After conducting the test, we decided that these roles fit us well, and that it would be best if we each continued in them for the rest of the tests. Our first user test also took longer than the following tests because we had not laid out the elements of the interface very well. After the first test, we improved our element layout, so that the second test went faster, and we improved it even more after that, so that the third test was the fastest, all other factors being held constant.

Testing Results

Heuristic Evaluation

From our heuristic evaluations, we primarily received feedback about displaying information to the user. This came in two forms. The first was displaying the system's state. It was suggested that coupon counts per item, how much money the coupons would save in total, and the quantity of an item should be shown to the user. The second was giving a first-time user more instruction about the app's use. Through the evaluation, we learned that some of the parts that we thought were evident to the user actually needed some prompting. We added clarification in parts of the app, such as what the levels on the profile page mean and what exactly the "Search" tab searches for. With these changes, we reduced the learning curve for a first-time user.

Usability Test 1

One type of issue that came up in this usability test was making some of the app's functionality apparent to the user. This cropped up in a couple of places, the first being the trip planning page. When the participant entered the page, they immediately selected one of the routes from the "1 Store" tab, not even noticing that there were routes on other tabs that would have been cheaper. Another part of the app that the participant bypassed was the barcode scanning. Knowing that they needed to add an item, they simply used the item searching as they had before. To assist the user in discovering the barcode scanner, we decided to move it directly into the search bar. Apps like MyFitnessPal already do this.

The most important change that came from this test was with the Store Set functionality. Originally, we allowed the user to maintain separate lists of stores to shop at. This way, a user could have one list for when they are at home and another list for when they are visiting their parents. While this would have been useful, it was not worth the confusion that it caused the testing participant. They immediately tried searching for specific stores when entering the screen, rather than making a new set and adding the store. To solve this problem, users now have only one group of stores to edit. We did decide to keep the "Within 2 Miles" option on the list screen, though, to make one-time shopping trips quicker.

Usability Test 2

This was the quickest and smoothest of the usability tests. A lot of the intuitive leaps that the participant made matched up with the design of the app. There were still a couple of minor

issues that cropped up during the testing, though. The first was simply due to the “low-fi” nature of the prototype. The participant had issues recognizing what the symbols on the map in the “Add Store” screen meant. To fix this in future tests, we decided to do this screen in color matching Google Map’s symbols. Another issue occurred when the participant tried to start planning a trip without selecting a store group. We quickly made an error pop-up to continue the testing, but later made the “Plan Trip” button greyed out until a group is selected.

Usability Test 3

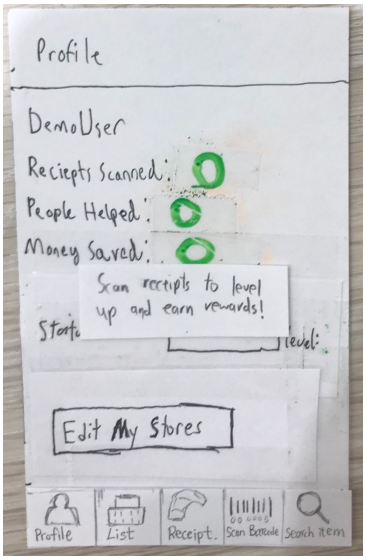
This test resulted in large changes to our “Shopping Plan” screen. Originally, tabs were used for each store, similar to the “Trip Planning” screen. However, two issues arose with that design. After shopping at the first store, the participant wanted to click the tab and move on to the next store. By missing the “Check Out” button, the user would have lost out on their coupons and we would have lost the opportunity to gather data from their receipt. The fact that the participant couldn’t see their entire shopping list on the first tab made it seem like some of their items had been lost. We fixed these issues by changing the shopping screen from tabs to a list of stores and their item counts. These stores can be expanded by tapping to show the grocery lists for that store as well as the checkout button. When the user first enters the screen, they can see by looking at the item counts that their entire shopping list is represented. Also, the vertical layout of the new screen implies an order between the stores and their checkout buttons. The redesigned screen has more similarities to a traditional grocery list than the original did.

As in previous tests, we continued to run into issues with the participant not realizing that barcode scanning was an option. We also had second thoughts about having the scanner only be in the search bar, as it would require more actions to get to that screen. To fix both these issues, we re-added the barcode scanner to the bottom bar, while also keeping it in the search bar. Having the scanner in the navigation bar makes it easier for the non-planned shoppers to use, while also keeping it in a place that the planned shoppers will notice.

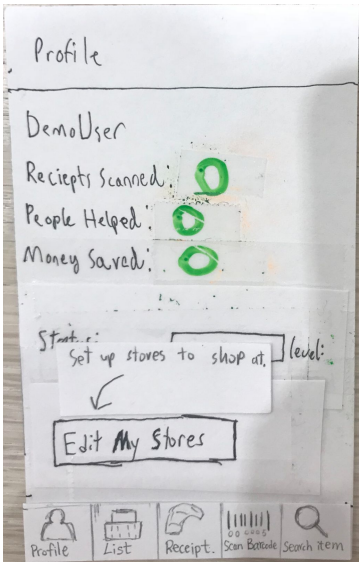
Final Paper Prototype

Overview

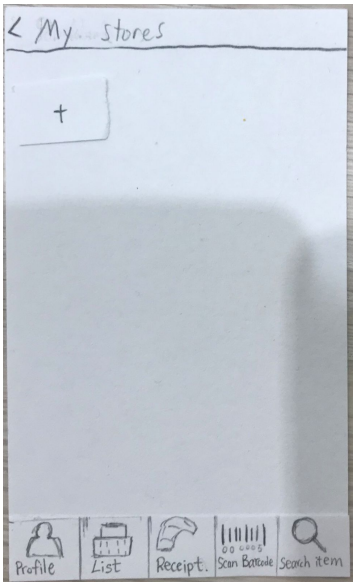
Task 1: Price Comparison



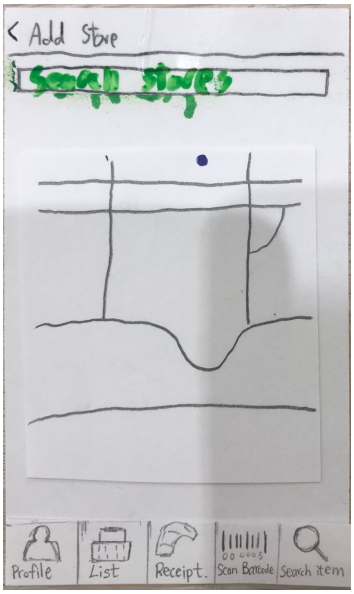
When the user opens the app for the first time, a series of pop up windows will appear. The first one tells the user to scan their recipes.



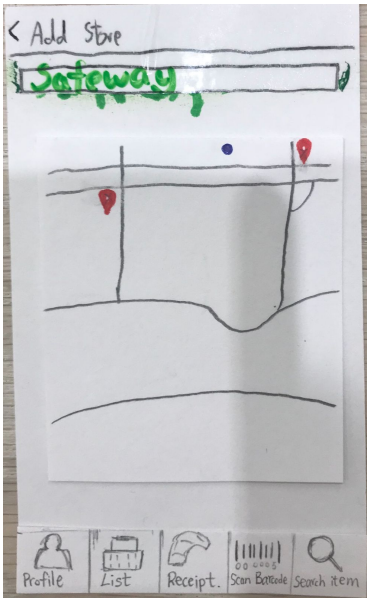
A second pop up prompts the user to add grocery stores.



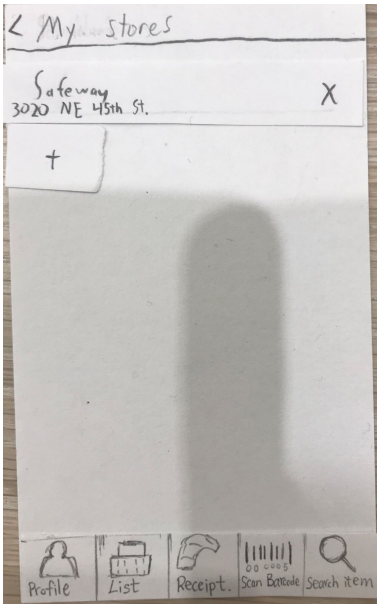
The user taps the + button to add grocery stores.



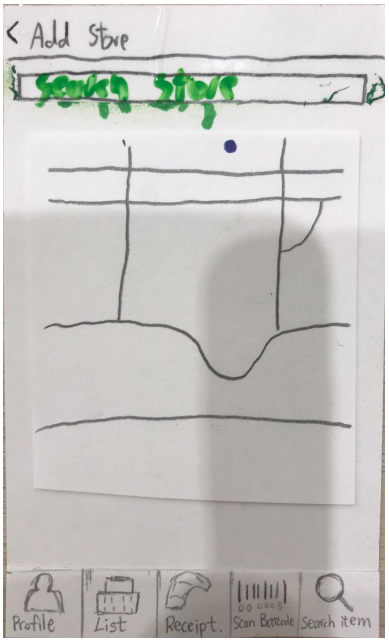
The user can then see their current location on the map, being shown as the blue dot.



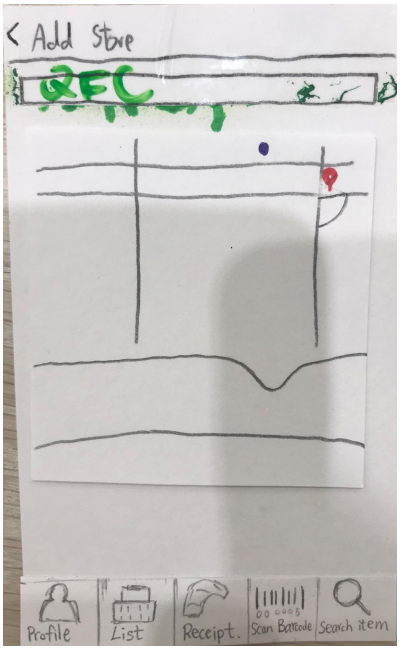
When the user searches for Safeway, the map shows the close by stores, represented by the red pin icons.



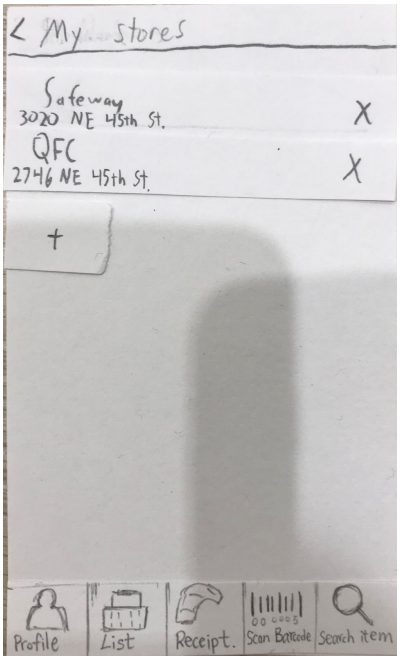
Safeway is added to My Stores list.
The user taps the + button to add more grocery stores.



The user can then see their current location on the map, being shown as the blue dot.

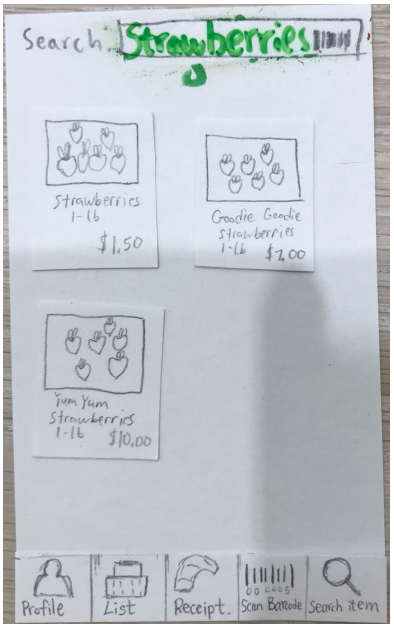


When the user searches for QFC, the map shows the close by stores, represented by the red pin icons.

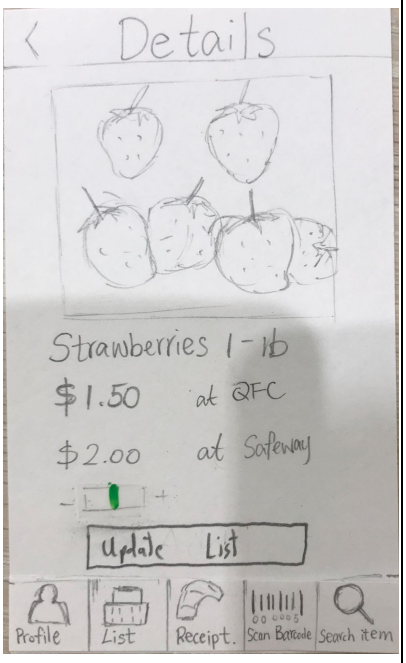


QFC is added to My Stores list.

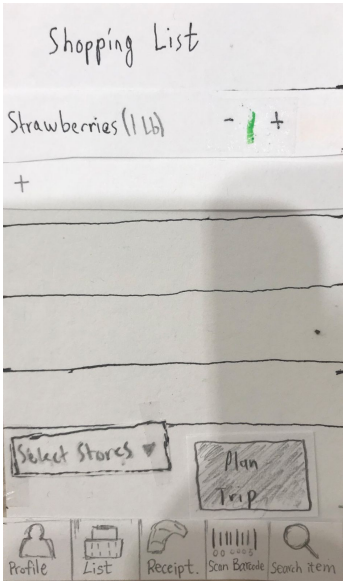
The user taps the Search item button in the tab bar.



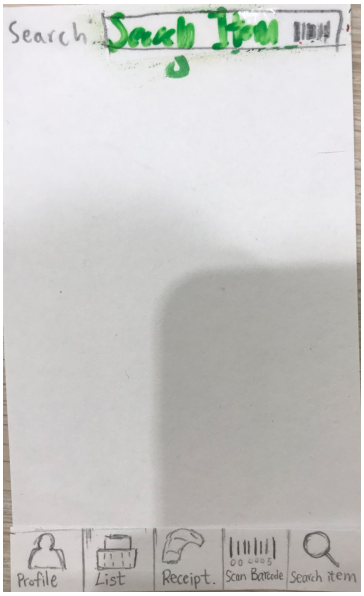
The user searches for strawberries.



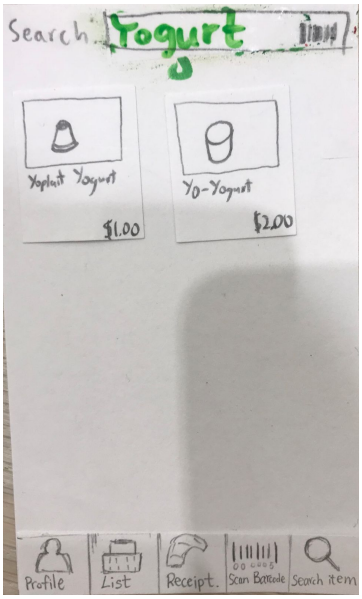
The user adds 1 1-lb of strawberries to their shopping list.



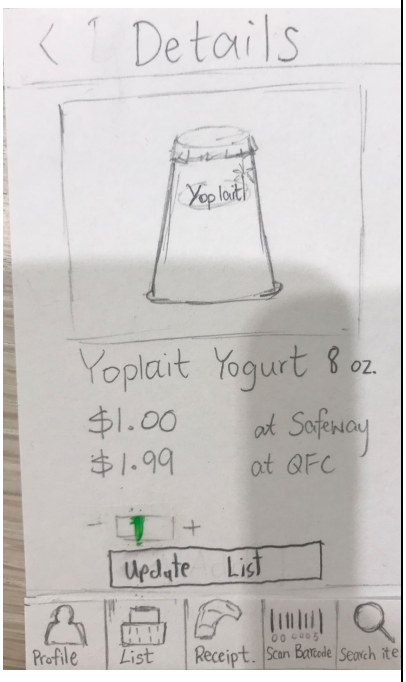
Strawberries are now in the user's shopping list.
Plan Trip is greyed out because stores haven't been selected yet.



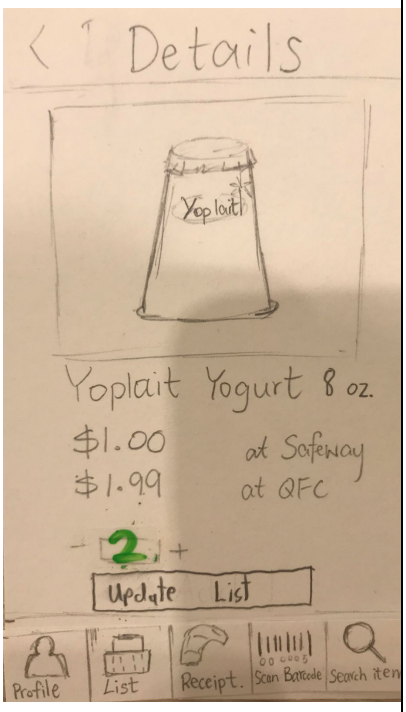
The user taps the Search item button in the tab bar.



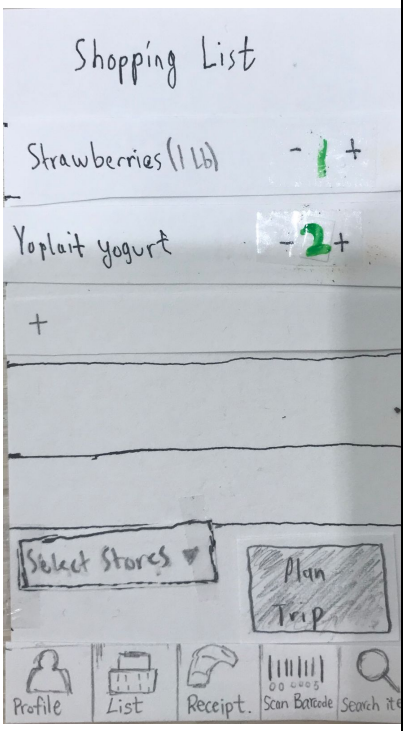
The user selects Yoplait Yogurt.



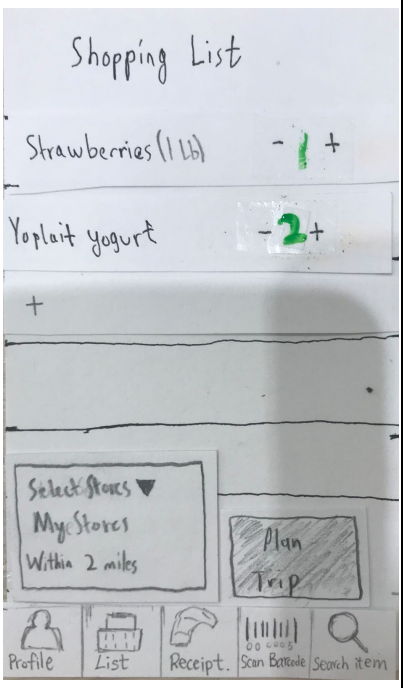
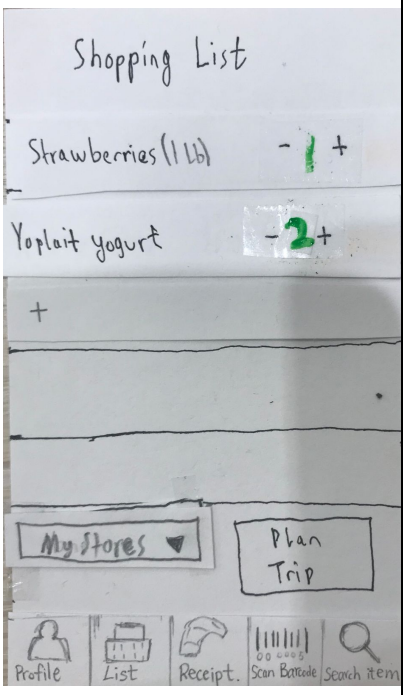
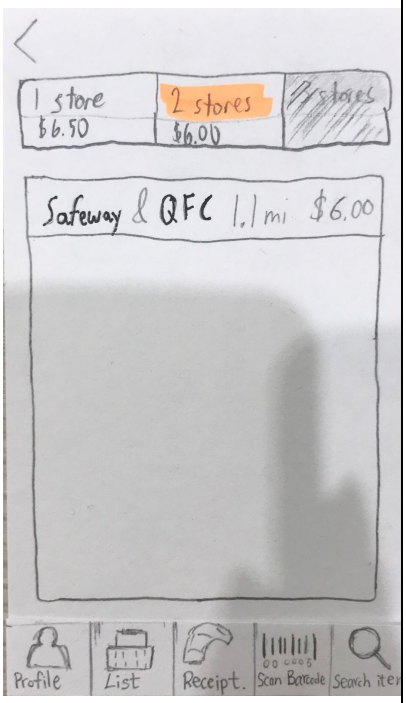
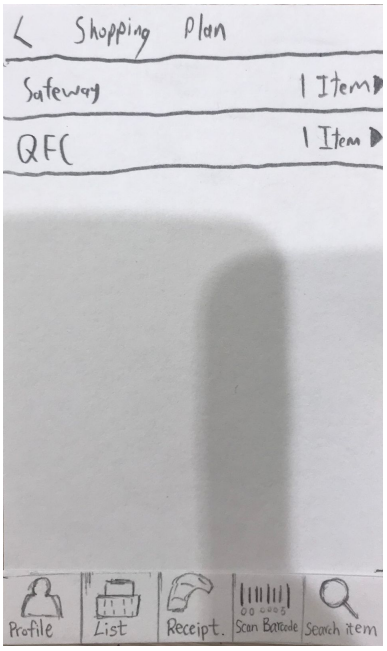
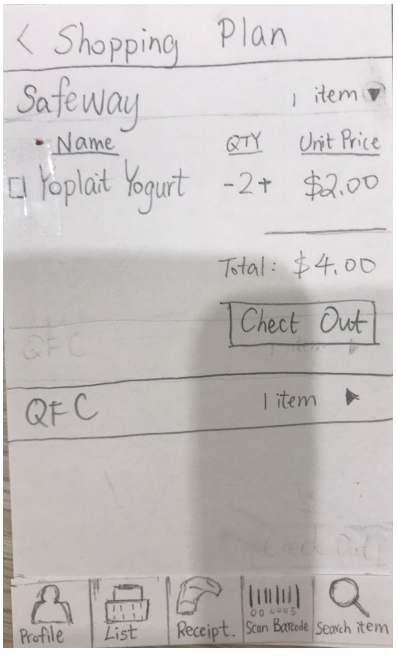
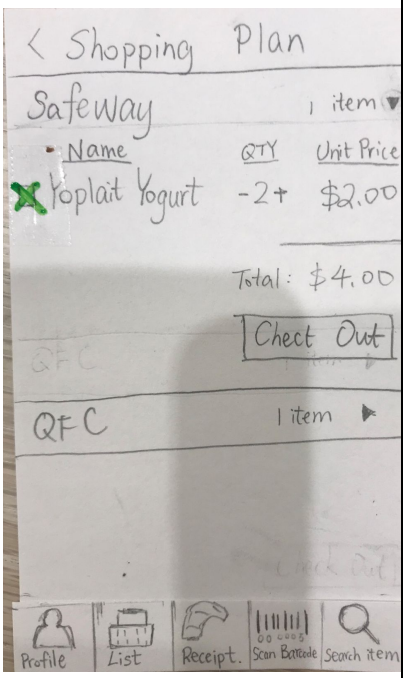
The user increases the reases the number of the yogurt they want to get.

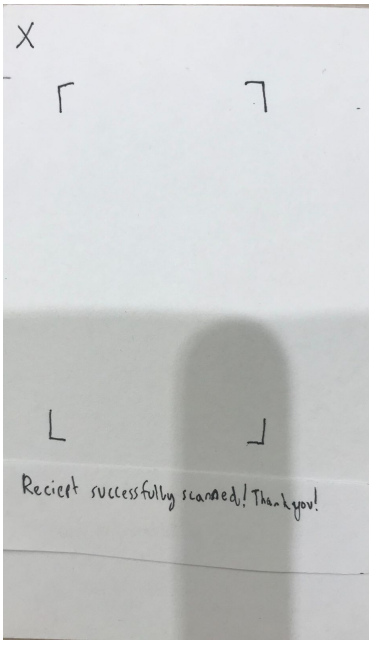
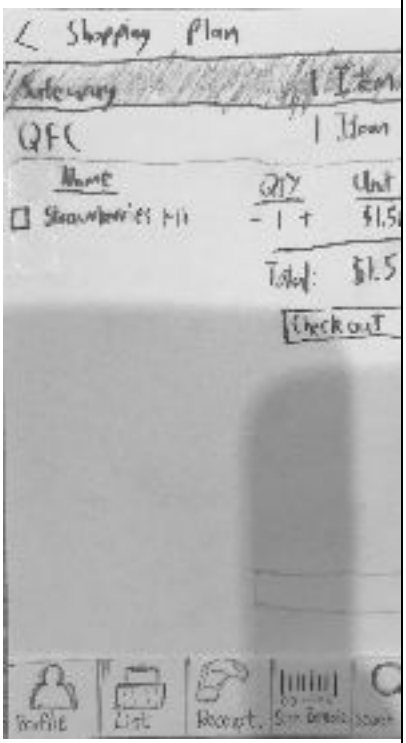
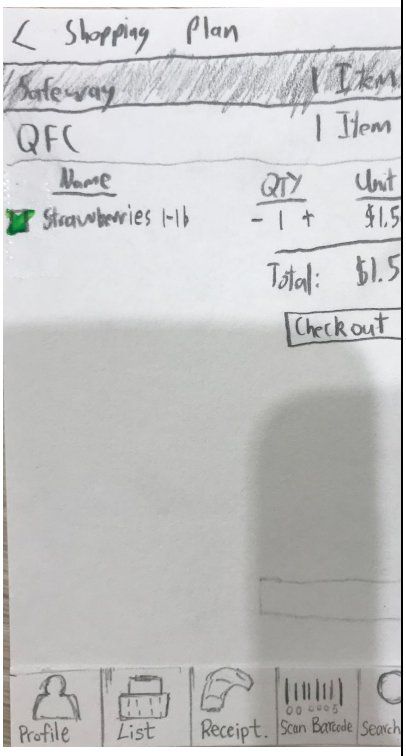
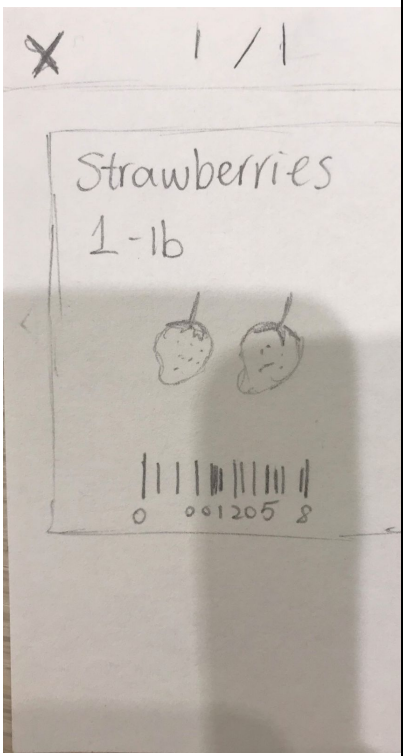


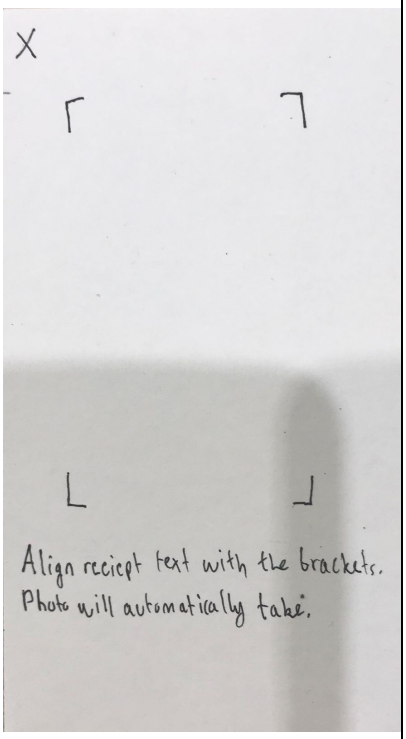
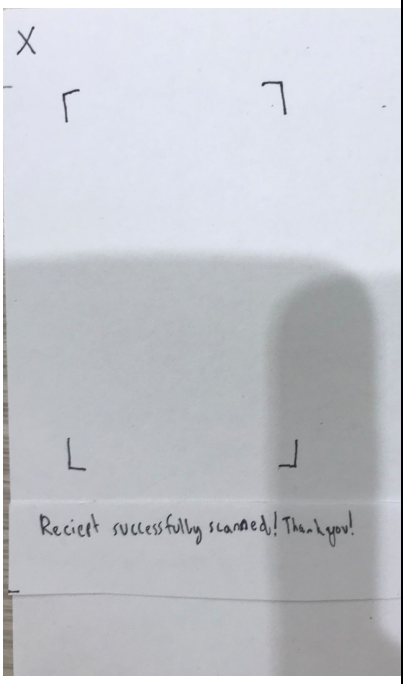
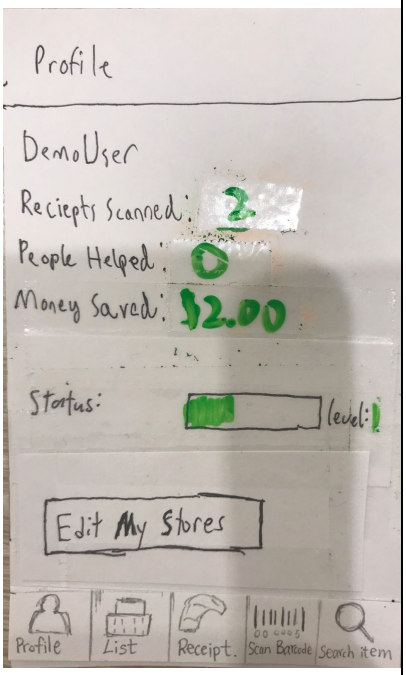
The user adds 2 Yoplait Yogurt to their shopping list.



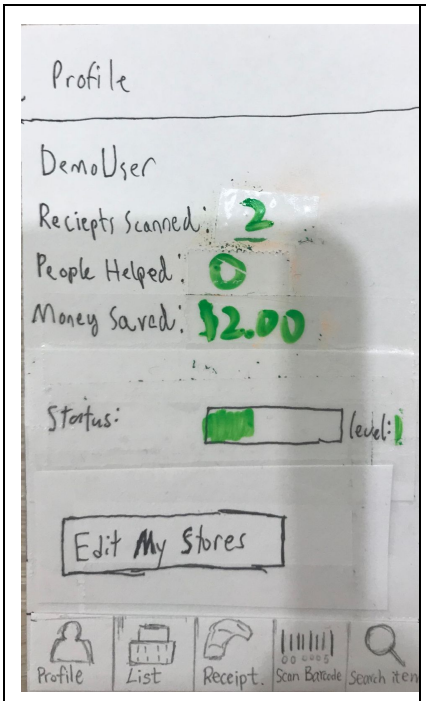
Yoplait yogurt is now in the user's shopping list along with the strawberries they have previously added.

		The Plan Trip button is still greyed out.
 The user taps the Select Stores dropdown menu.	 The user selects My Stores, which includes Safeway and QFC. The user taps Plan Trip.	 The user selects to go to 2 stores because it's a cheaper option.
 The shopping plan shows that the user needs to get one item	 The app displays a list of items to be purchased at Safeway.	 The user checks the Yoplait Yogurt off the list to indicate that

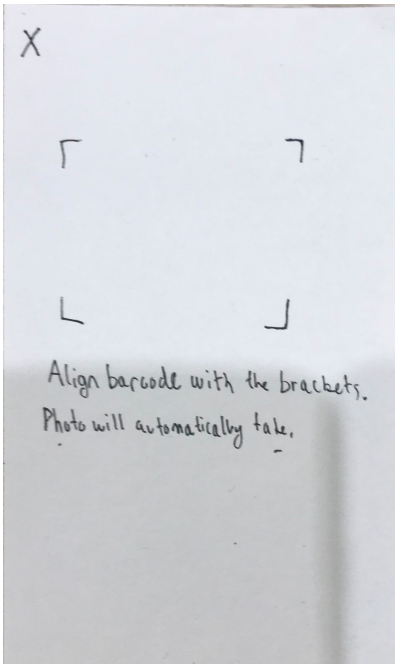
from Safeway and one item from QFC.		they have been added to the physical basket/cart.
 The app displays the yogurt coupon for the user to scan.	 Camera turns on for user to scan their receipt.	 The user receives a thank you message for scanning the receipt.
 The app displays a list of items to be purchased at QFC.	 The user checks the Strawberries off the list to	

	indicate that they have been added to the physical basket/cart.	The app displays the strawberries coupon for the user to scan.
 Camera turns on for user to scan their receipt.	 The user receives a thank you message for scanning the receipt.	 User is sent back to the profile page and the stats are updated.

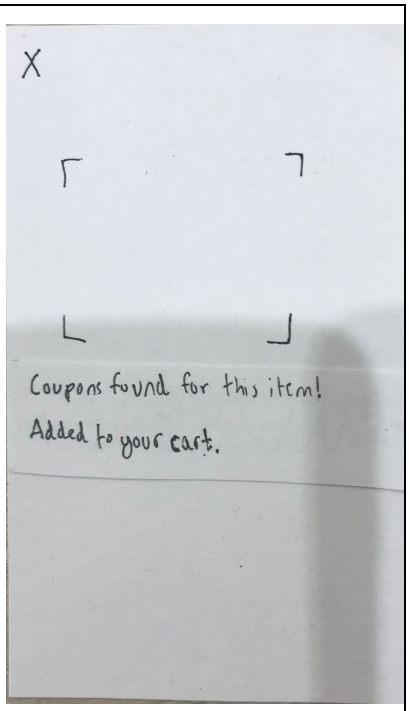
Task 2: Finding Coupons



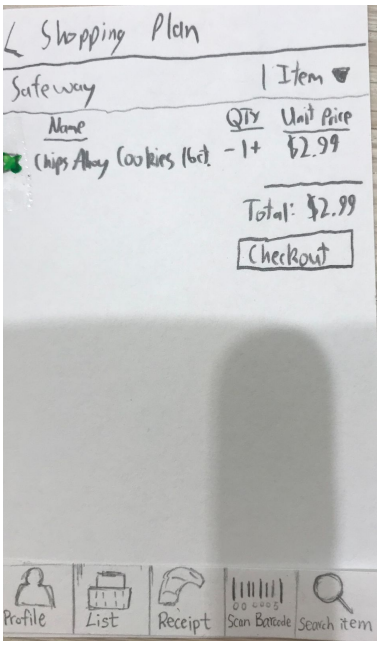
User taps the Scan Barcode button in the tab bar.



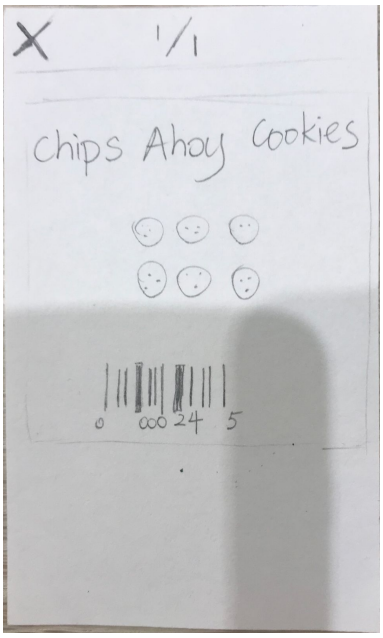
Camera turns on for user to scan item barcode.



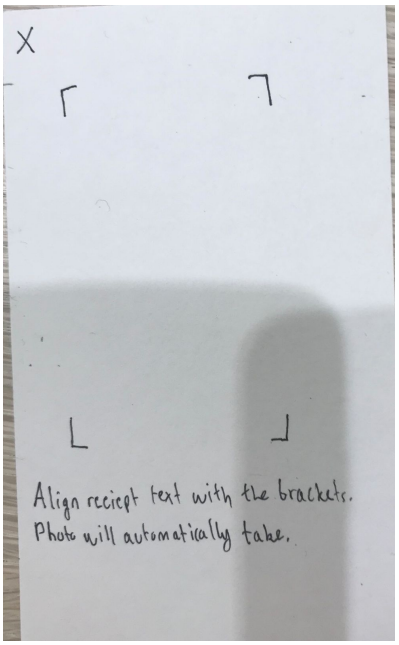
After the scanning, if coupon is found, it is automatically added to user's cart.



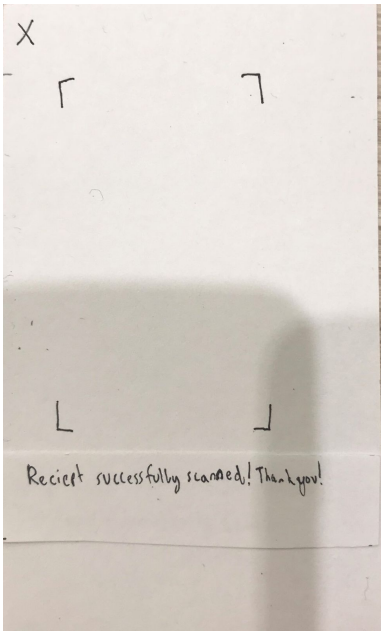
Chips Ahoy Cookies is checked off in the shopping plan because the item is added through barcode scanning.



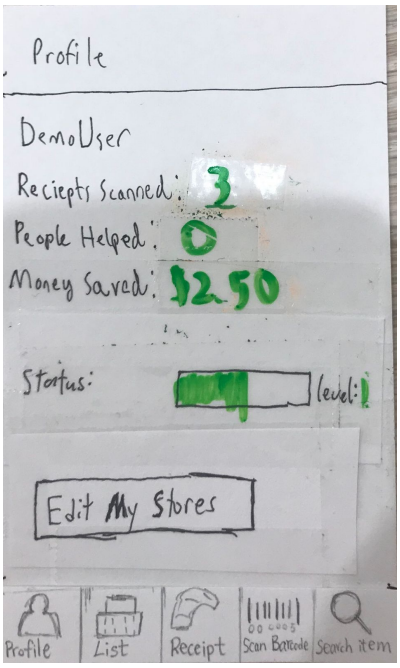
User scans the Chips Ahoy Cookies coupon as they check out.



Camera turns on for user to scan their receipt.



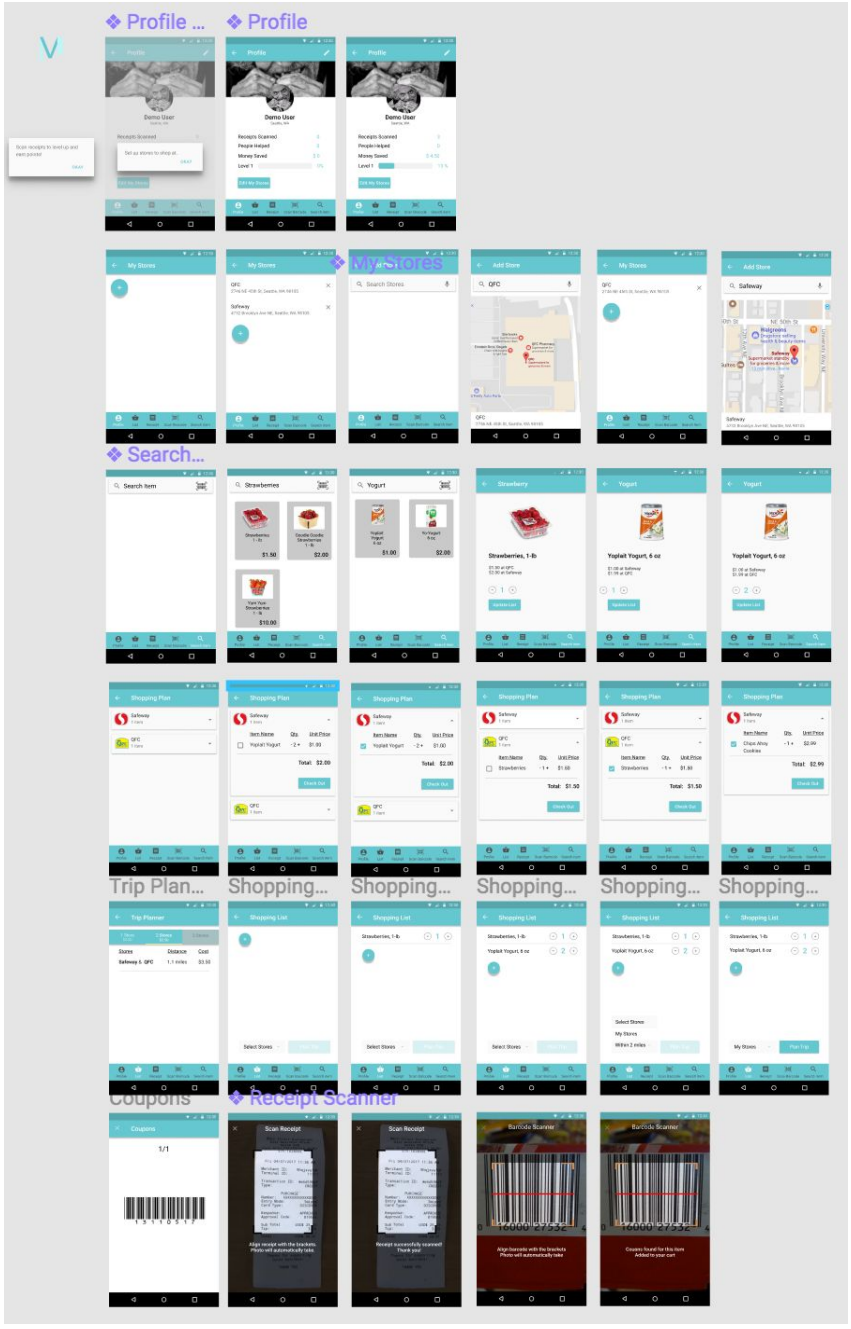
The user receives a thank you message for scanning the receipt.

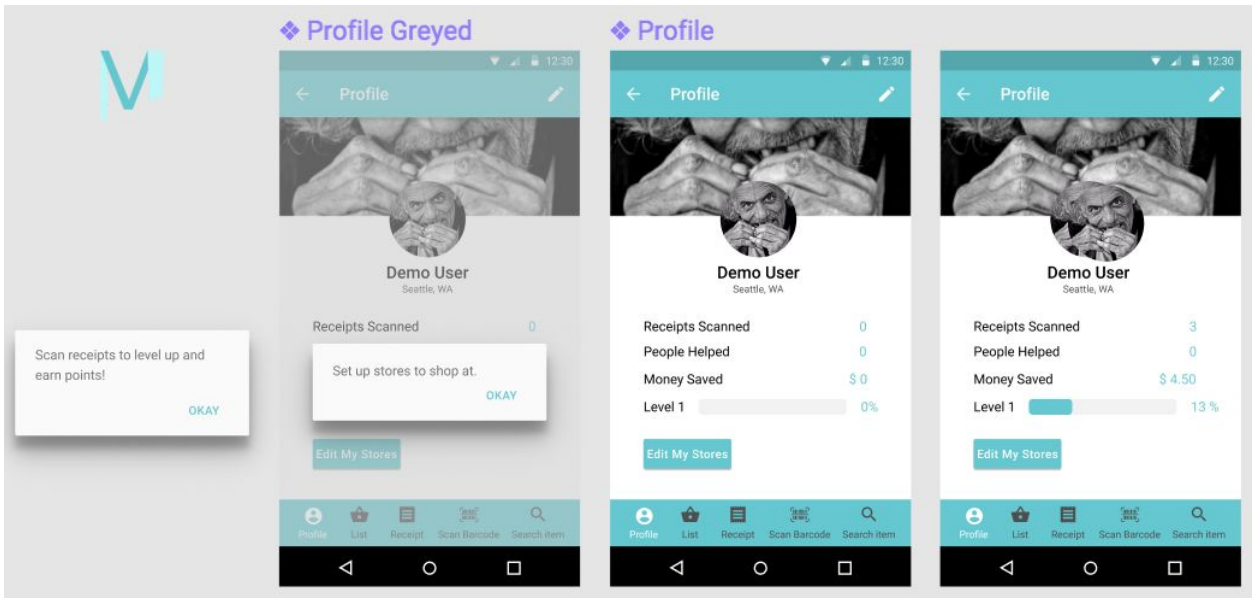


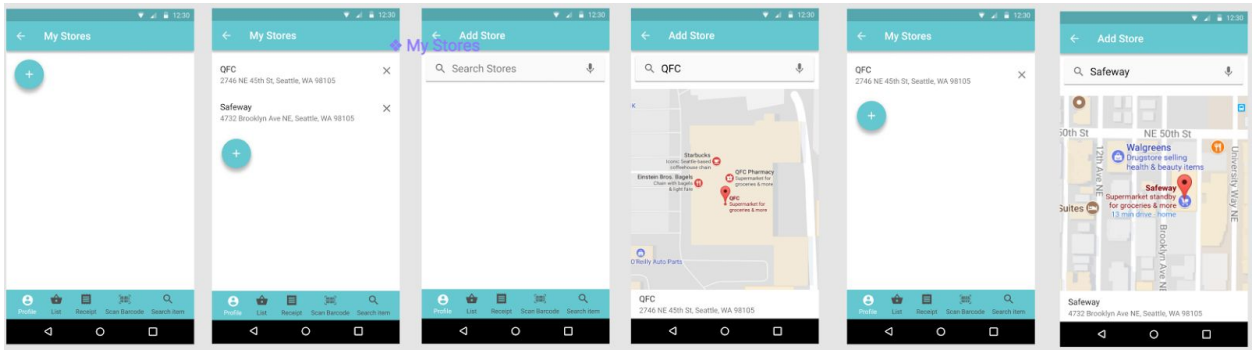
User's profile updates.

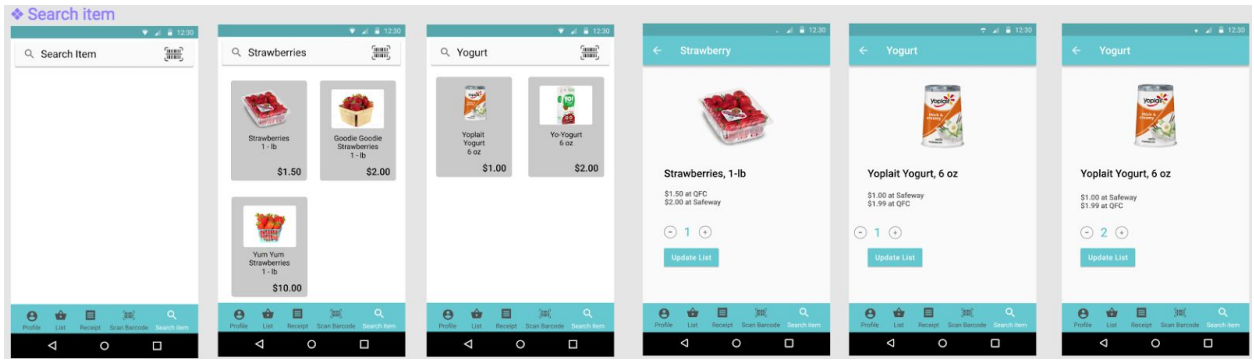
Digital Mockup

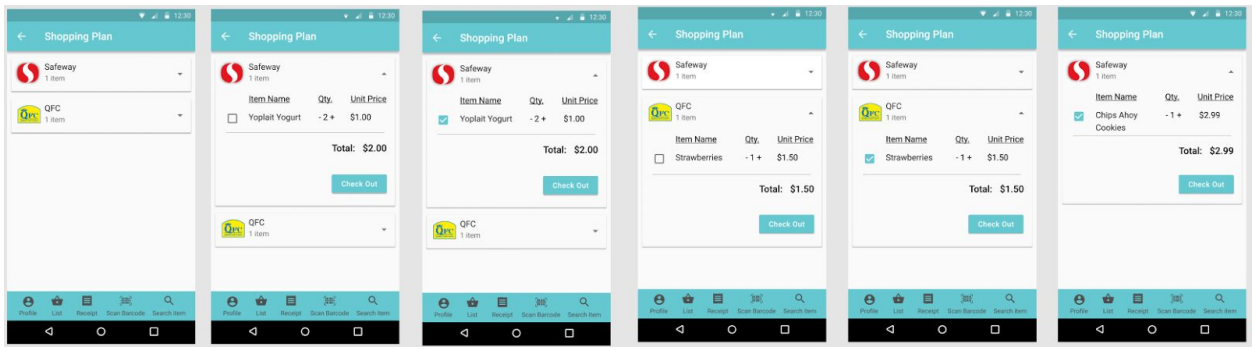
Overview

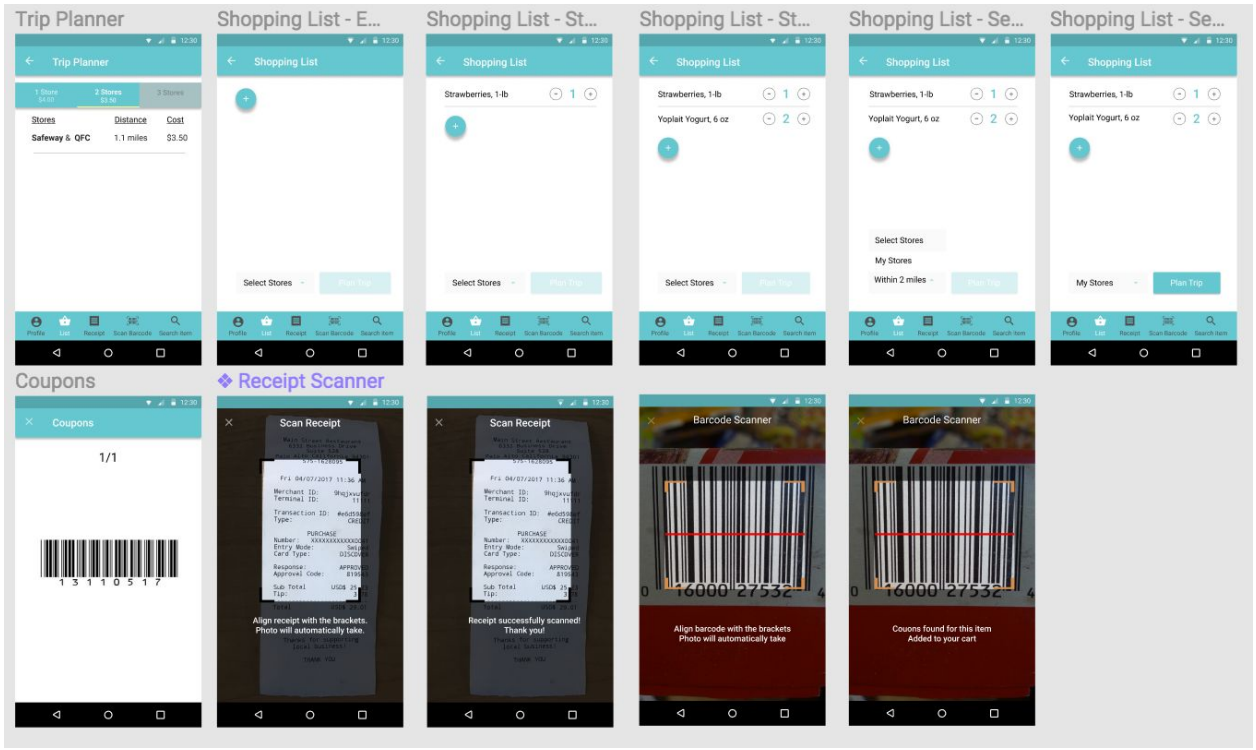



Profile Greyed
Profile
Profile
Demo User
Seattle, WA
Receipts Scanned
People Helped
Money Saved
Level 1
Edit My Stores
Scan receipts to level up and earn points!
OKAY
Set up stores to shop at.
OKAY
Profile
List
Receipt
Scan Barcode
Search item


My Stores
Add Store
QFC
Safeway
Search Stores


Search item
Search Item
Strawberries
Yogurt
Strawberry
Strawberries, 1-lb
Yoplait Yogurt, 6 oz
Update List


Shopping Plan
Safeway
QFC
Item Name
Qty.
Unit Price
Yoplait Yogurt
Strawberries
Chips Ahoy Cookies
Total:
Check Out

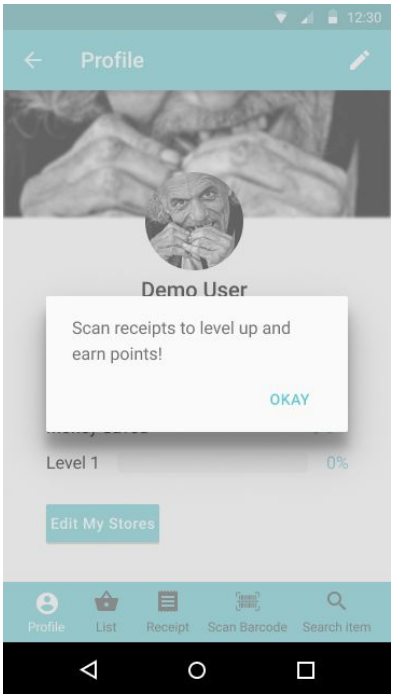

Trip Planner
Trip Planner
Stores
Distance
Cost
Safeway & QFC
1.1 miles
$3.50
Shopping List - E...
Shopping List
Select Stores
Shopping List - St...
Shopping List
Strawberries, 1-lb
Select Stores
Shopping List - St...
Shopping List
Strawberries, 1-lb
Yoplait Yogurt, 6 oz
Select Stores
Shopping List - Se...
Shopping List
Strawberries, 1-lb
Yoplait Yogurt, 6 oz
Select Stores
My Stores
Within 2 miles
Shopping List - Se...
Shopping List
Strawberries, 1-lb
Yoplait Yogurt, 6 oz
My Stores
Plan Trip
Coupons
Coupons
1/1
Receipt Scanner
Scan Receipt
Align receipt with the brackets.
Photo will automatically take.
Scan Receipt
Receipt successfully scanned!
Thank you!
Barcode Scanner
Align barcode with the brackets
Photo will automatically take
Barcode Scanner
Couons found for this item
Added to your cart

Task 1: Price Comparison

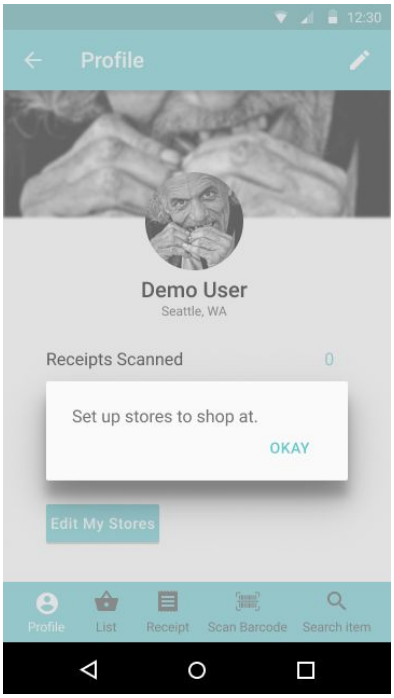
Our goal was to assist the user in saving money. Doing this involves taking information from the user such as what stores they are willing to shop at and what items they want. We did our best to make gathering this as low-effort as possible for the user. The store selection is a one-time setup that can be reused for future trips. Meanwhile, after the initial setup, the app defaults to the shopping list page. As adding an item to their shopping list will be an action repeatedly taken over the course of a week, it was important to make that as efficiently as possible.

Another aspect of the design is that it allows users to make their own decisions of how much time they are willing to spend in order to save money. This is important to do, as our usability testing showed that people had different values of these two resources. The first way this is accomplished is through letting the user select their own stores. They can add further stores if they are willing to make the trip. Letting the user see multiple routes and compare their costs relative to their travel distances also lets them make an informed decision.

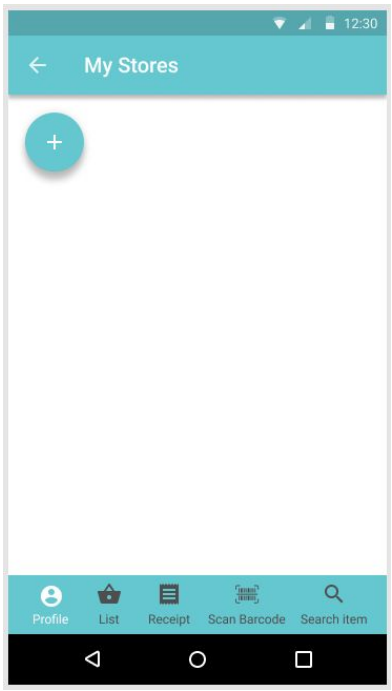
Through the use of our design, the user is able to quickly find ways to save money at stores they shop at.



When the app is opened for the first time, a series of popups will appear. The first one tells the user to scan their receipts.

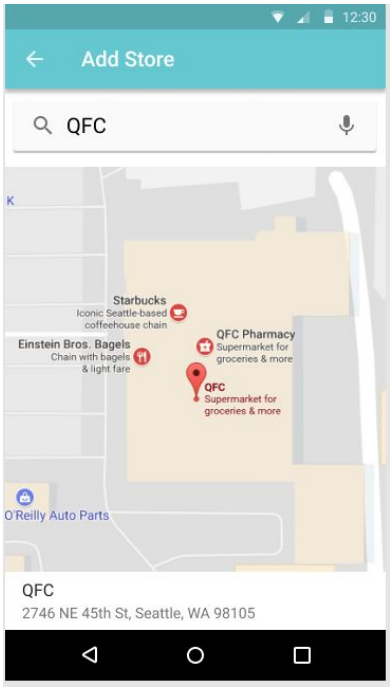


A second pop up prompts the user to add grocery stores.

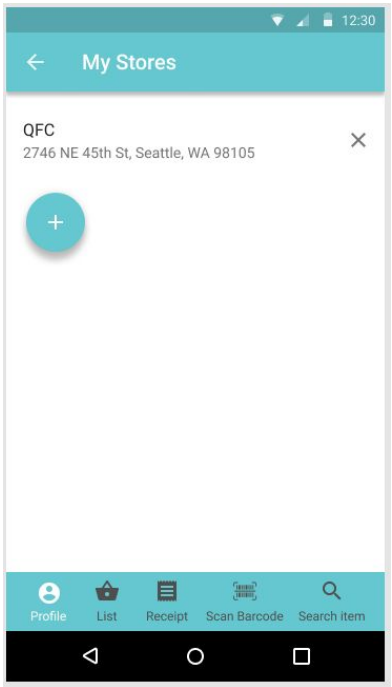


The user taps the + button to add grocery stores.

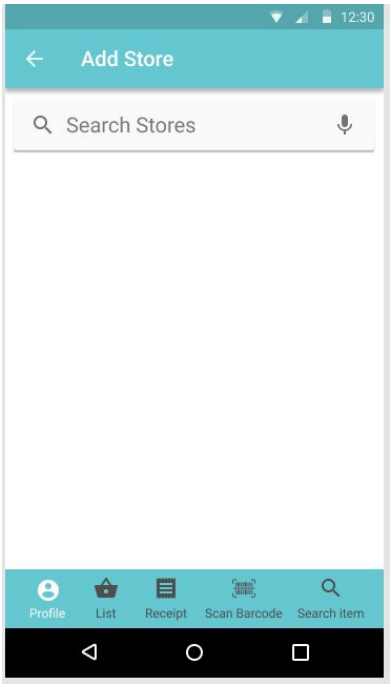
The user is prompted to search for a store.



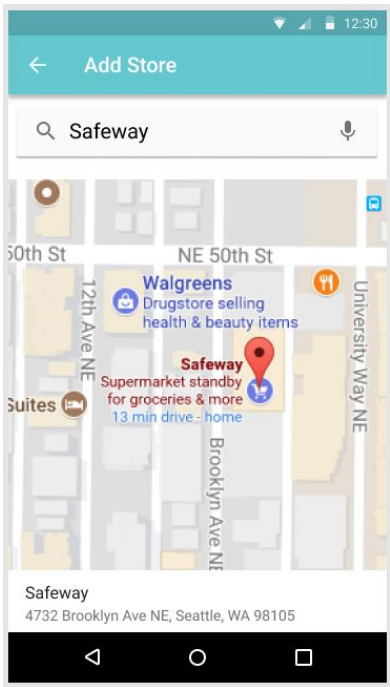
When the user searches for QFC, the map shows the close by stores, represented by the red pin icons.



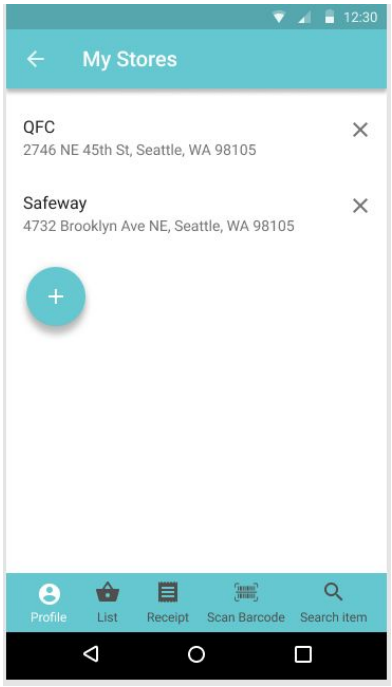
QFC is added to My Stores list. The user taps the + button to add more grocery stores.



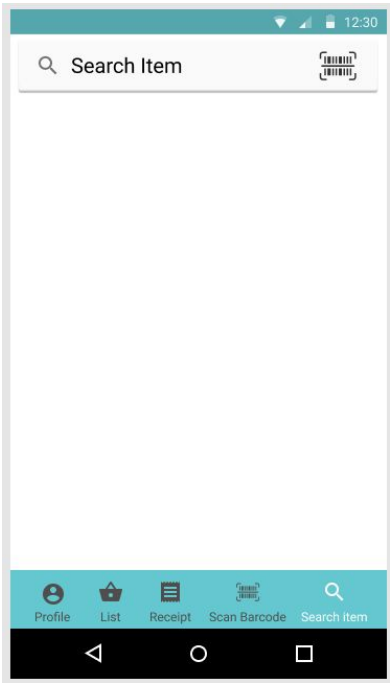
The user is prompted again to search for another store.



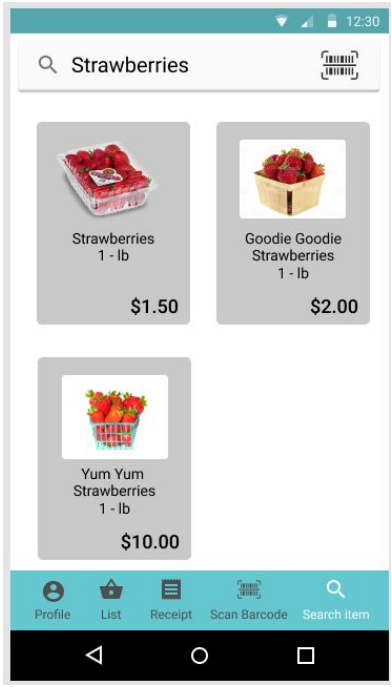
When the user searches for Safeway, the map shows the close by stores, represented by the red pin icons.



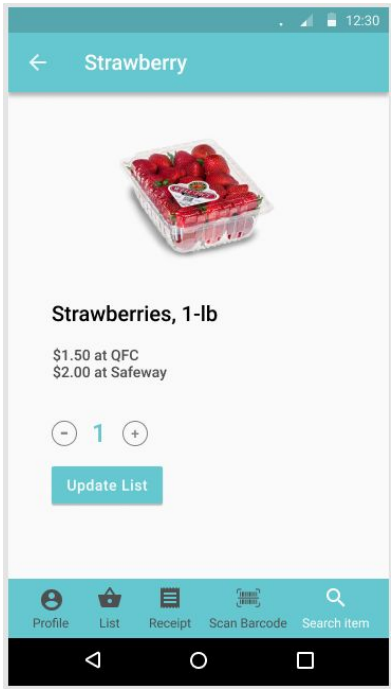
Safeway is added to My Stores list.



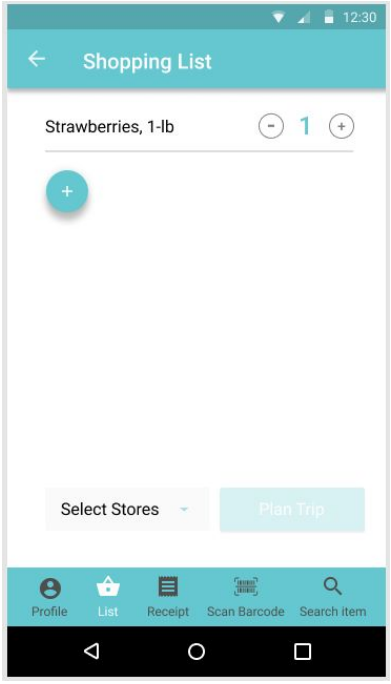
The user taps the Search item button in the tab bar.



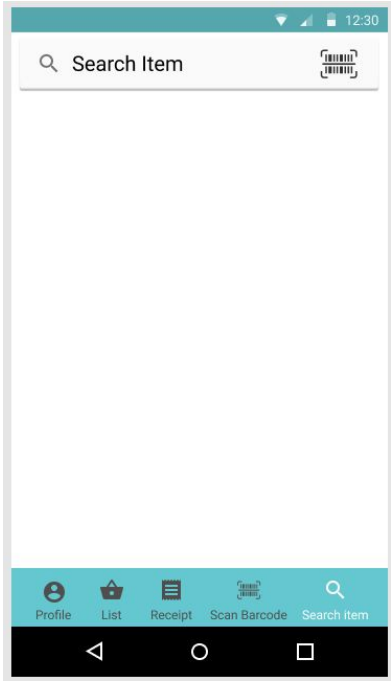
The user searches for strawberries.



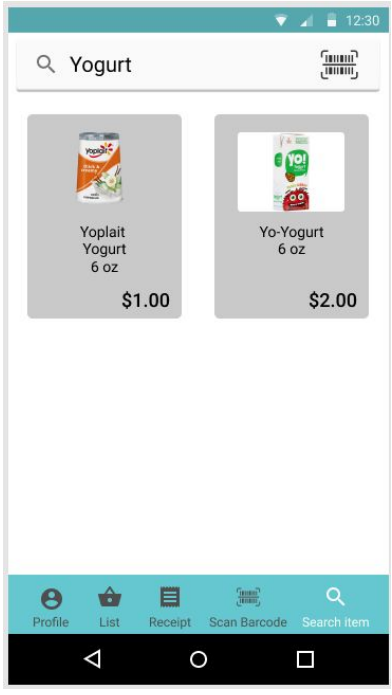
The user adds 1 1-lb of strawberries to their shopping list.



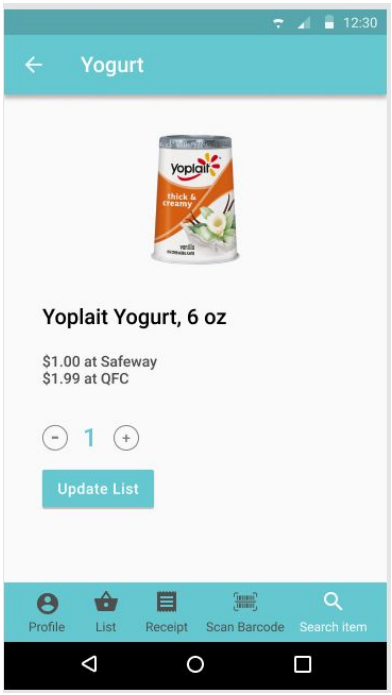
Strawberries are now in the user's shopping list.
Plan Trip is greyed out because stores haven't been selected yet.



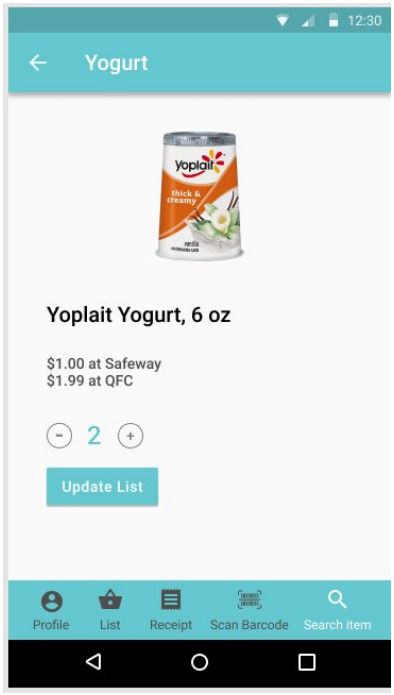
The user taps the Search item button in the tab bar.



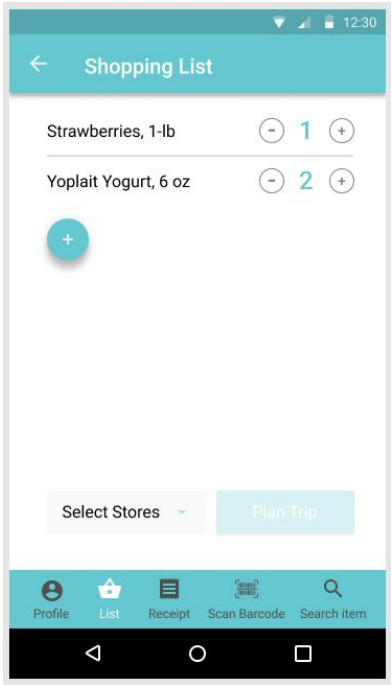
The user selects Yoplait Yogurt.



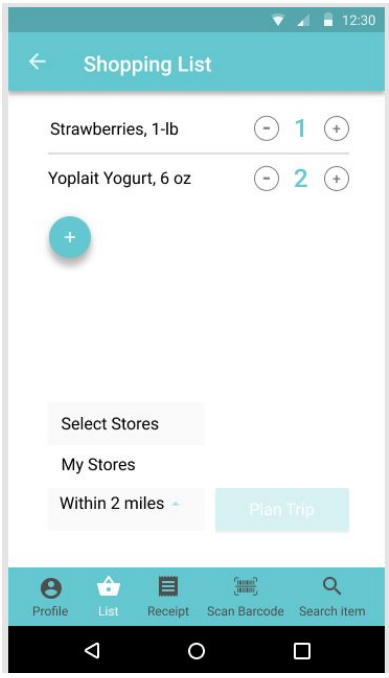
The user increases the quantity of the yogurt they want to get.



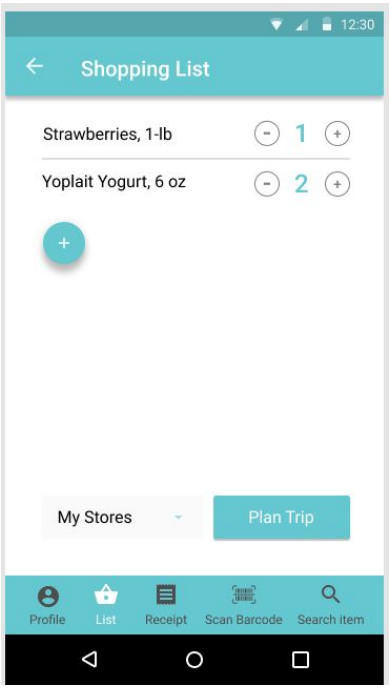
The user adds 2 Yoplait Yogurt to their shopping list.



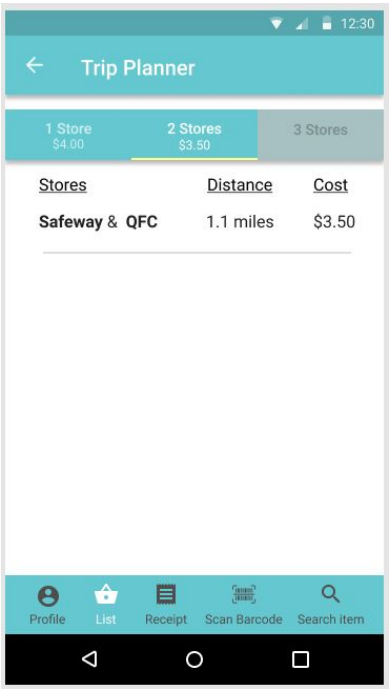
Yoplait yogurt is now in the user's shopping list along with the strawberries they have previously added.
The Plan Trip button is still greyed out.



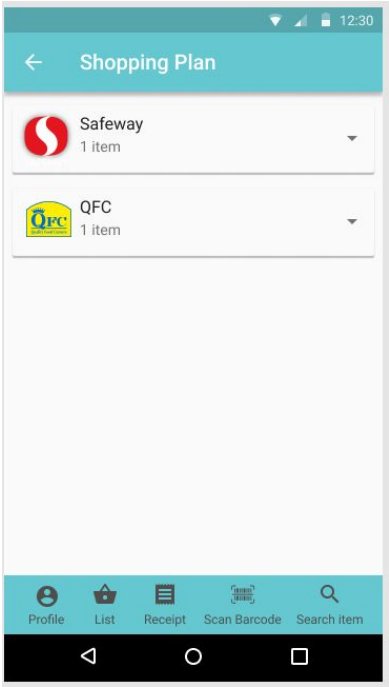
The user taps the Select Stores dropdown menu.



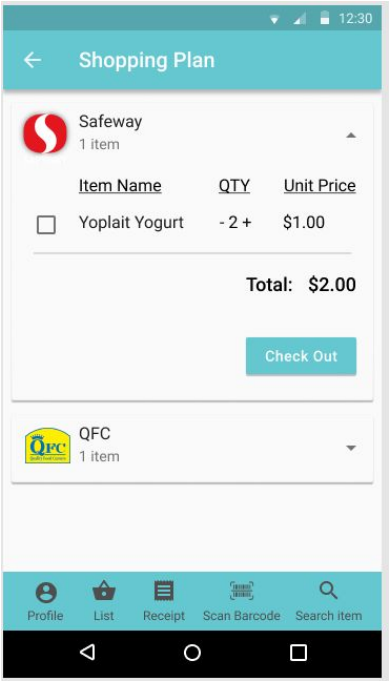
The user selects My Stores, which includes Safeway and QFC. The user taps Plan Trip.



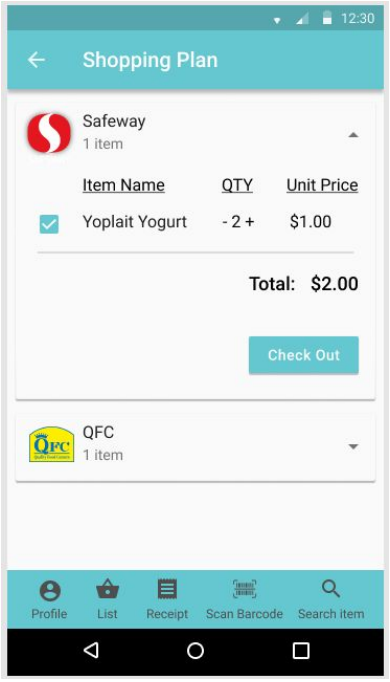
The user selects to go to 2 stores because it's a cheaper option.



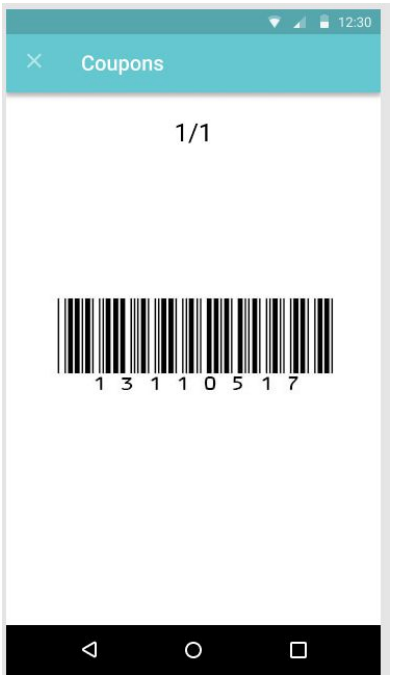
The shopping plan shows that the user needs to get one item from Safeway and one item from QFC.



The app displays a list of items to be purchased at Safeway.



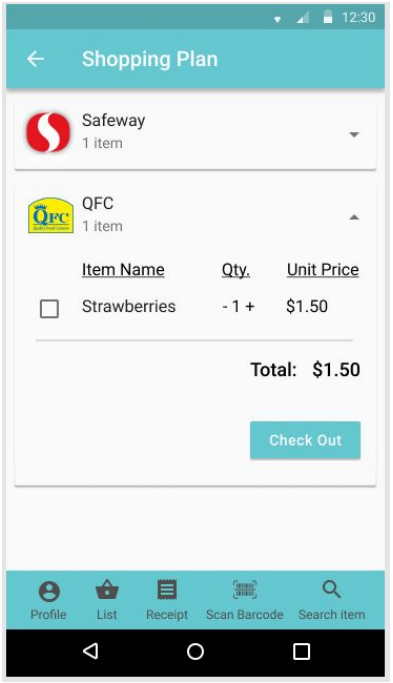
The user checks the Yoplait Yogurt off the list to indicate that they have been added to the physical basket/cart.



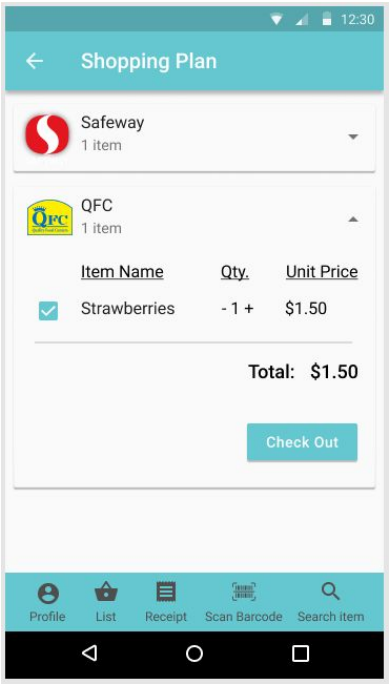
The app displays the yogurt coupon for the user to scan.

Camera turns on for user to scan their receipt.

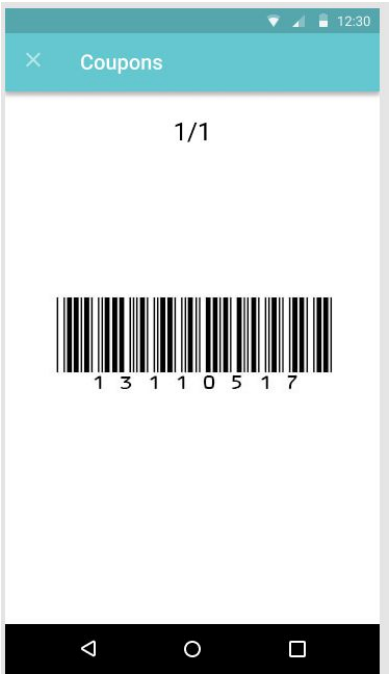
The user receives a thank you message for scanning the receipt.



The app displays a list of items to be purchased at QFC.



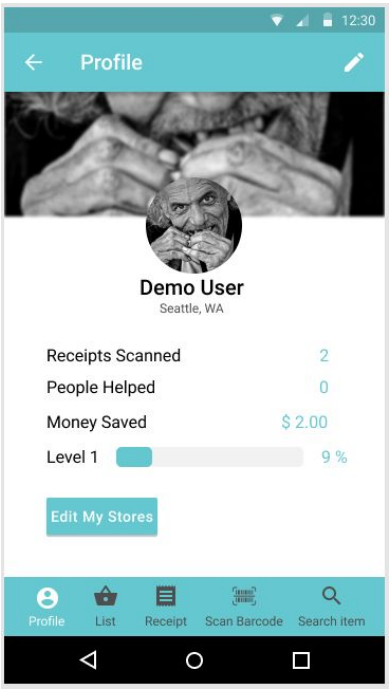
The user checks the Strawberries off the list to indicate that they have been added to the physical basket/cart.



The app displays the strawberries coupon for the user to scan.

Camera turns on for user to scan their receipt.

The user receives a thank you message for scanning the receipt.

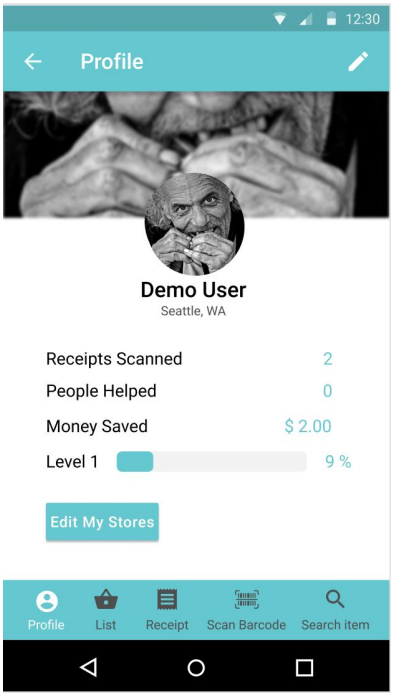


User is sent back to the profile page and the stats are updated.

Task 2: Finding Coupons

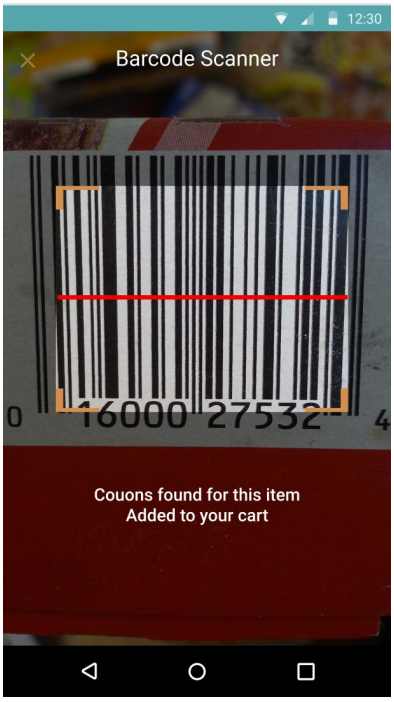
While we tried to make the process of making a shopping list as easy as possible, there were some design research participants who didn't us shopping lists at all. We still wanted to help those potential users save money without the hassle of planning trips. One way we found to do this was through couponing. The app allows the user to scan the barcode of an items they pick up and get any applicable coupons for them to check out.

Although coupons are a way for us to help these users, we wanted to keep the coupons themselves as hidden as possible. Our research showed that, apart from the relatively low-effort Safeway app coupons, coupons were largely ignored. To keep the process as simple as possible, all prices in the app already have coupons applied. This way, the only time the user needs to worry about coupons is when they scan them at checkout.

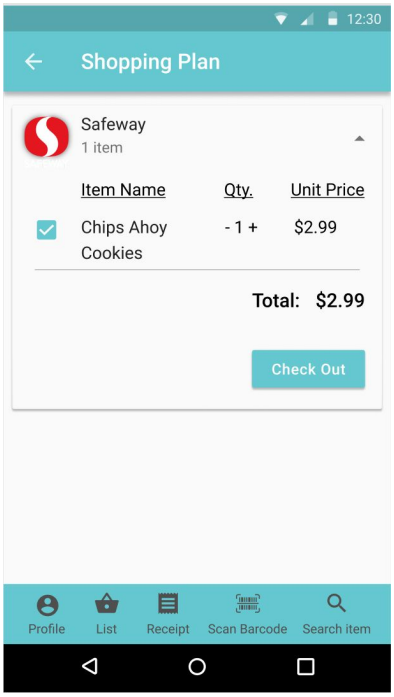


User taps the Scan Barcode button in the tab bar.

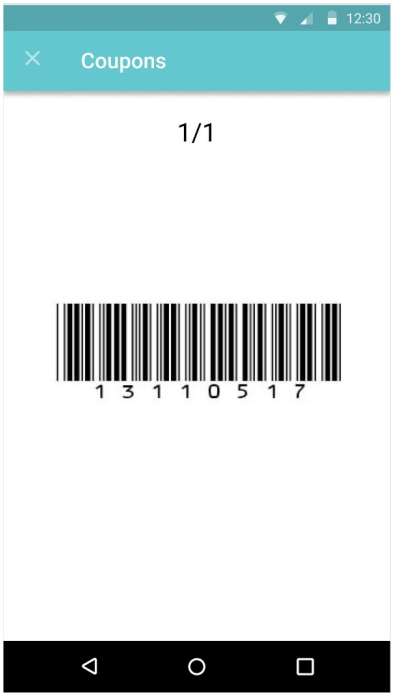
Camera turns on for user to scan item barcode.



After the scanning, if coupon is found, it is automatically added to user's cart.



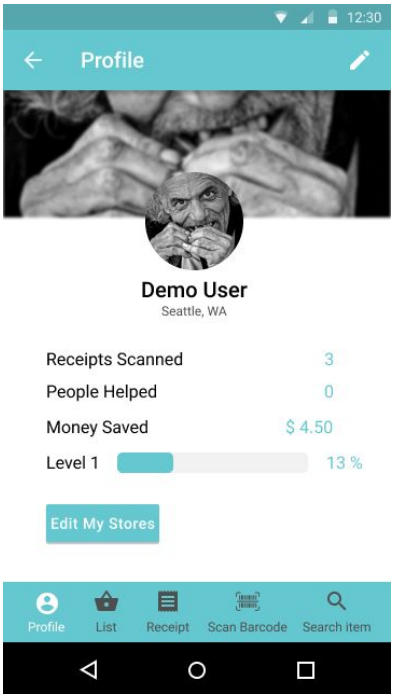
Chips Ahoy Cookies is checked off in the shopping plan because the item is added through barcode scanning.



User scans the Chips Ahoy Cookies coupon as they check out.

Camera turns on for user to scan their receipt.

The user receives a thank you message for scanning the receipt.



User's profile updates.

Changes

Though we did not make many changes when going from our paper prototype to our digital mock up, we did decide that a few changes were necessary. We chose to allow the user to add an image to their profile screen, which also meant that we needed to add a button which would allow the user to edit that image. Even though this change is not considered critical, we do think that it makes the profile page look better than before, and is therefore worthwhile. We also added grocery store logos to the shopping plan because we realized that this will make it much easier for the user to quickly identify which store is which, especially since most stores use their own logo in many places, so that people become very accustomed to them. These logos also make the shopping plan screen more aesthetically appealing. In the “Add Store” section we added the address of each store below its name so that the user can make sure they are selecting the correct store to add to their list. Instead of using indented buttons on the “Trip Planner” screen, we decided to add a yellow bar to the bottom of the selected tab. We did this because the UI element that we chose was better suited for the device we were using, and went along better with the rest of the design than using indented buttons. Which tab is selected is still easy to see. At first we just used a + and - sign to have the user change the quantity of the items in the shopping list. The only problem with this is that the user would have to tap a very small area to select the buttons. To solve this problem, we put the + and - inside of circles that are larger, and much easier to tap. When we made our paper prototype, we transferred the bottom navigation from one screen to the next. This meant that the bottom bar looked the same for all the screens. When making our digital mockup we decided to highlight the tab that the user was on to provide visual feedback, and make the app easier to use. Because the “Receipt” and “Scan Barcode” tabs take the user to the camera right away and remove the bottom navigation when they do so, the user will never see either of those two tabs highlighted.

Discussion

The iterative design process that we went through in the development of MetaGrocer was quite interesting and incredibly useful. Although our first selected design from our early sketches was “solid” and fundamentally “on the right track”, it had *many* glaring issues that we were blind to by virtue of being the designers and not uninvested users. Many interface elements broke basic design heuristics, but in ways that were not obvious to experienced users (in this case, the designers) of the interface. If we had more or less settled with our initial paper prototype design, we would have essentially ended up with a practically unusable app for everyone but us!

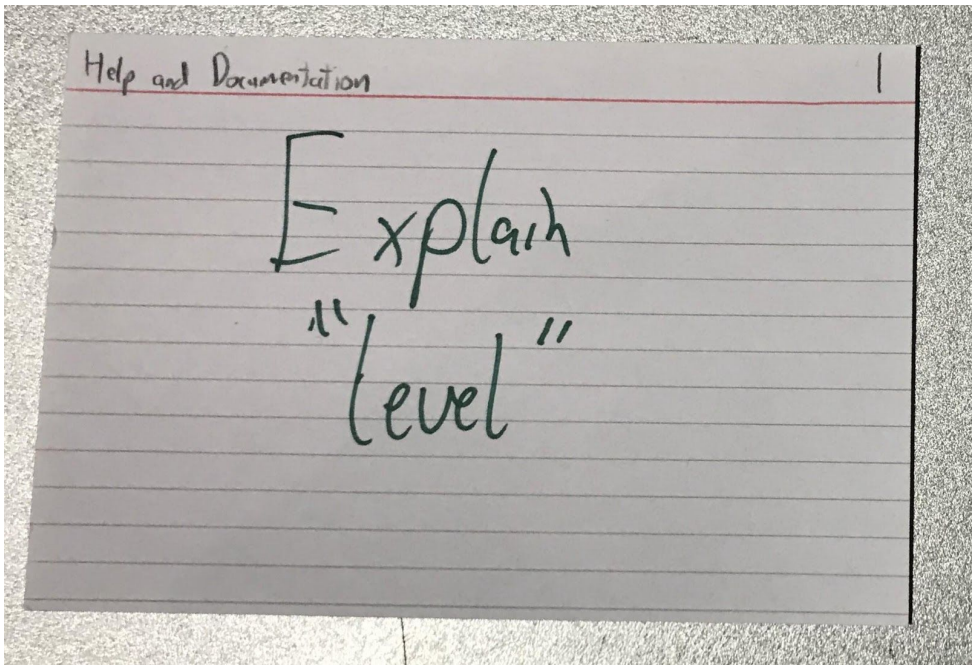
Our earliest designs centered around the two greater goals of saving money *and* time while shopping through use of an app. After our later contextual inquiries, we realized that the scope of the design was possibly too broad to really serve any of its users well. We narrowed down our focus to the specific tasks of saving money in both planned and spontaneous ways while shopping, but initially did not change our design in some ways that would be conducive to that. For instance, knowing that we needed to significantly overhaul the “flow” of the shopping trip functionality of our app truly required usability testing, as we were unable to notice the flaws in our initial design that could lead to users getting stuck or potentially “breaking” the state of the interface.

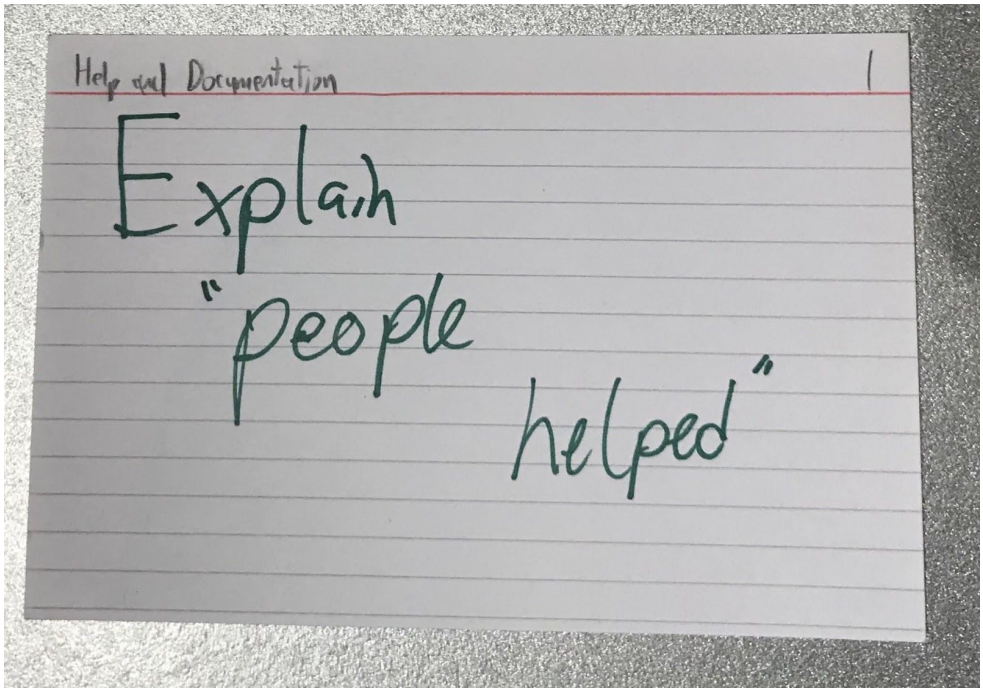
One of the other issues that the testing process helped with was finding a way to introduce new users to the interface without having to compromise the speed of the interface for experienced users too much. We had floated a variety of ideas to help with this, but ultimately we wouldn’t have known if the one we settled on (brief hideable pop-ups that describe key features of the app on the first launch) was *actually* useful without testing it on a variety of users (which we did).

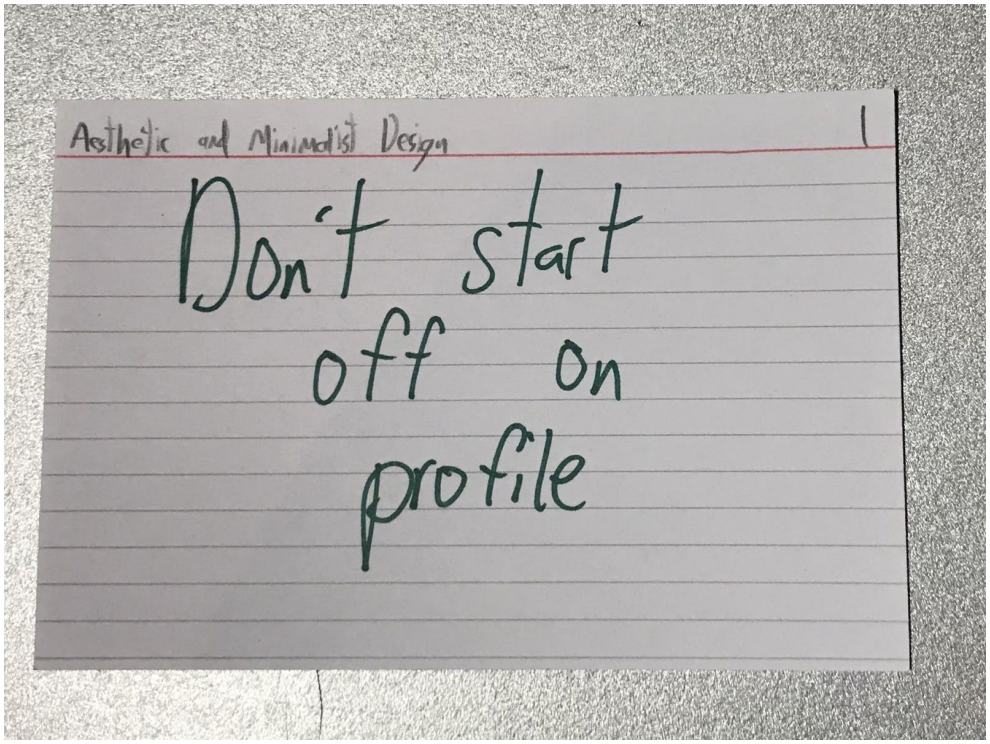
In the move to a digital mockup, some minor design ambiguities that were difficult to elucidate on paper (such as embossed/selected/grayed out buttons, for instance) became much clearer, and it would in fact be nice to see if users responded better to these final changes than they did to the same screens in the final paper prototype; unfortunately, due to the structure of the project it is unlikely we will have a chance to test this. Essentially, we still feel there’s some significant possibility for improvement from further iterations, but the work we’ve done so far probably brings us 80-90% of the way to the maximum design improvement possible from testing.

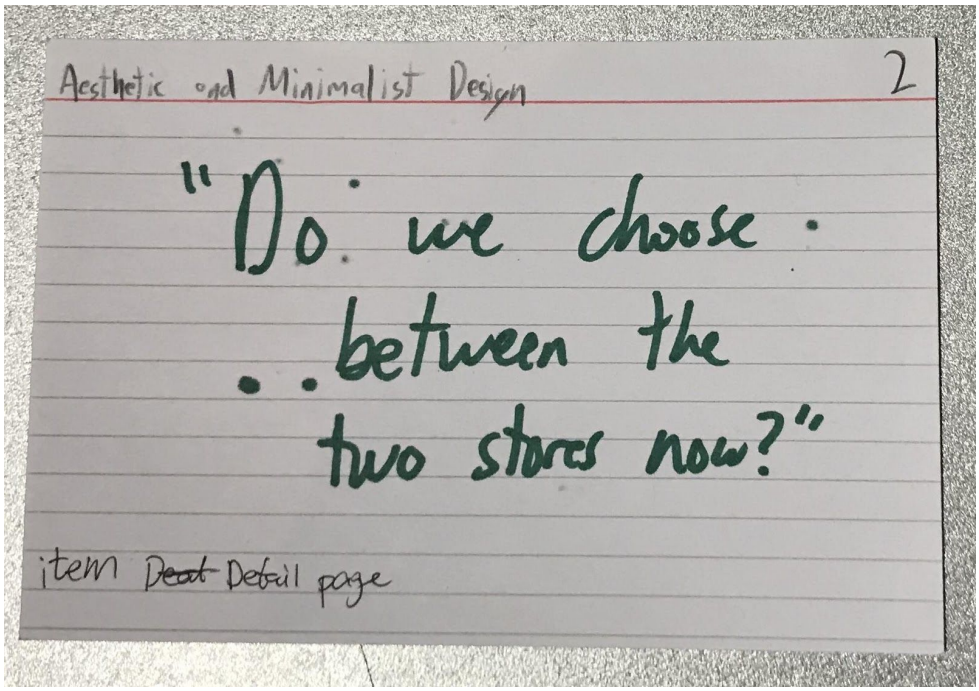
Appendix

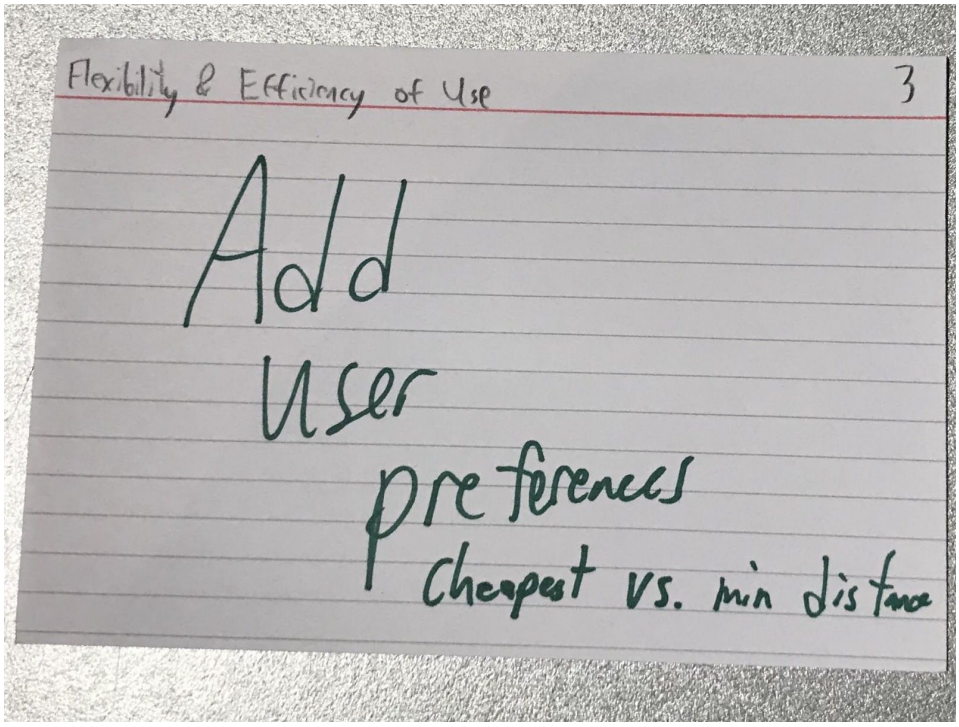
Heuristic Evaluation Cards

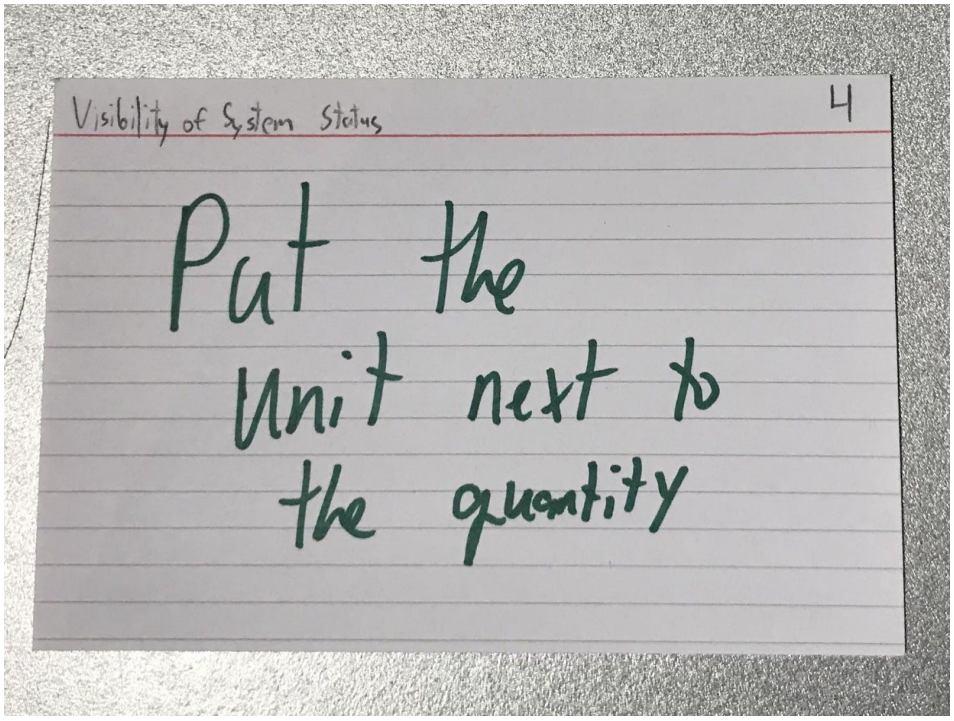


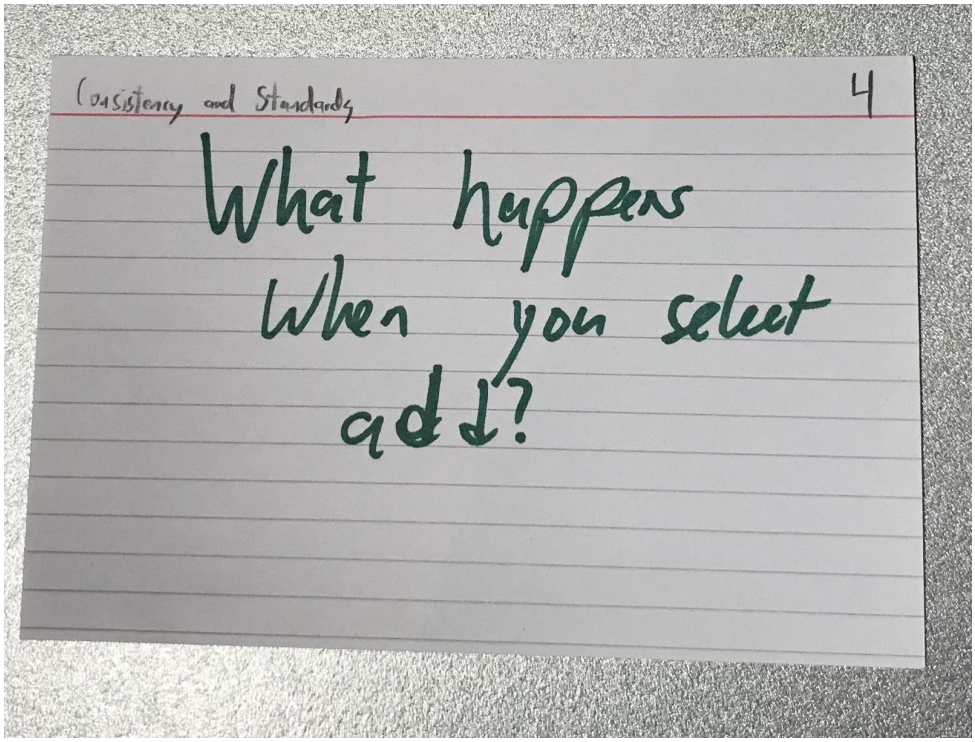


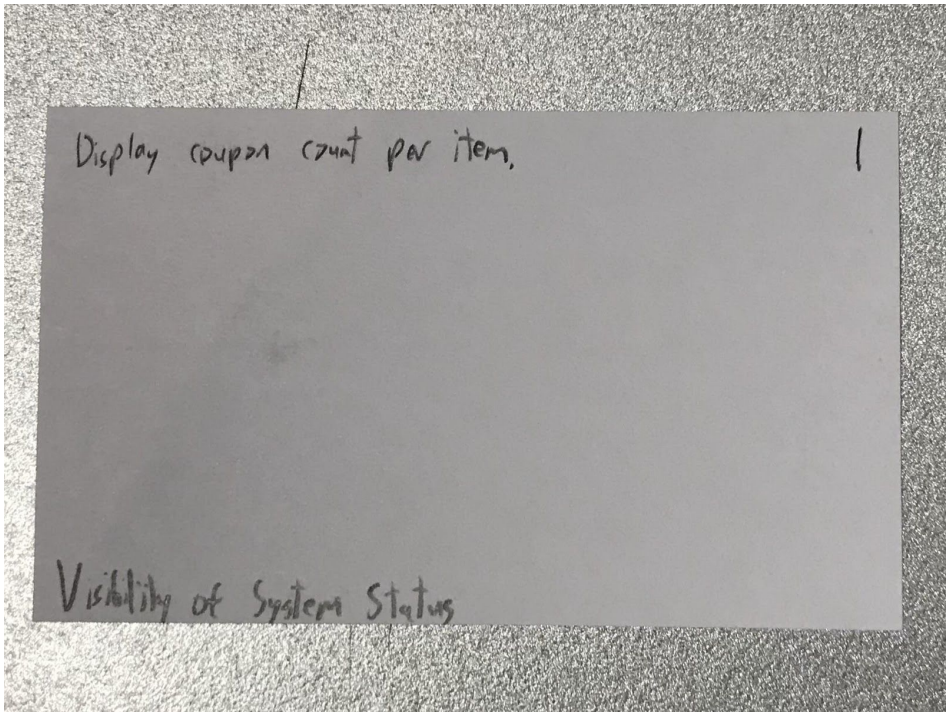

Aesthetic and Minimalist Design
1
Don't start off on profile

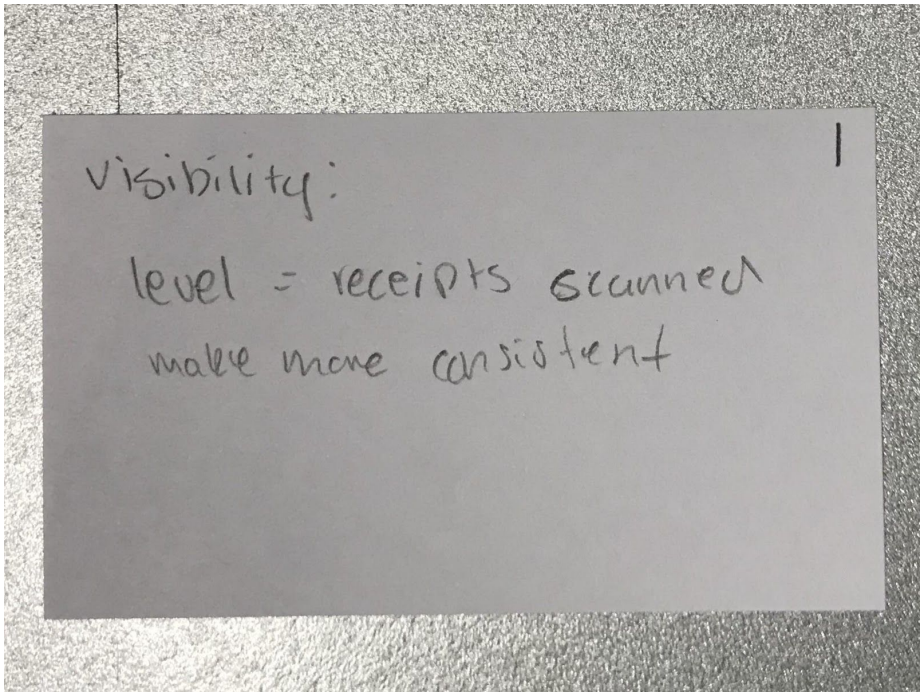

Aesthetic and Minimalist Design
2
"Do we choose ... between the two stores now?"
item ~~Dead~~ Detail page

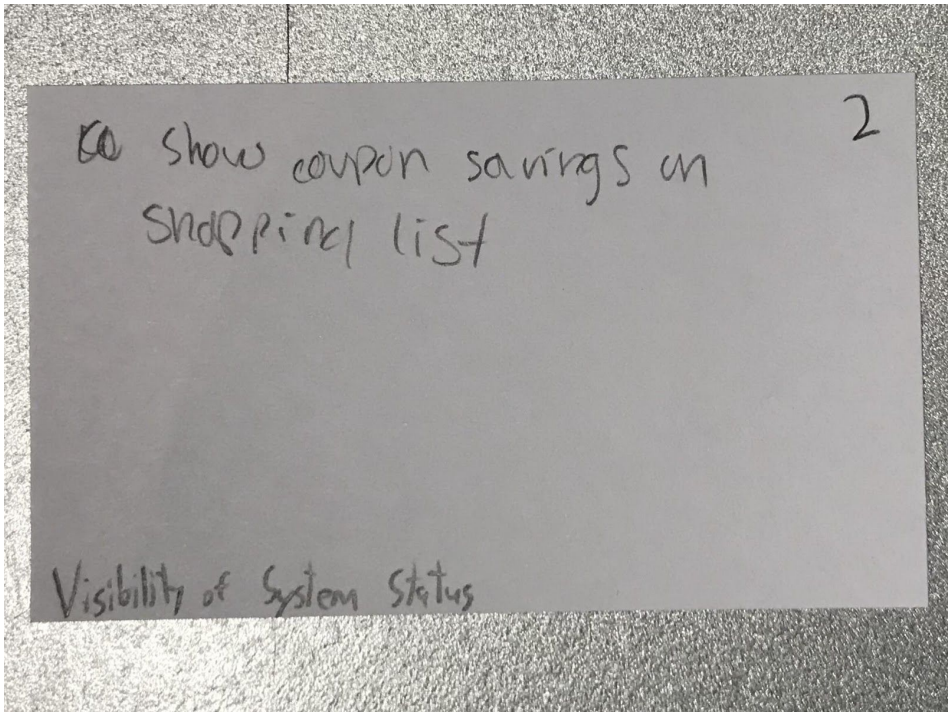

Flexibility & Efficiency of Use
3
Add
user
preferences
cheapest vs. min distance

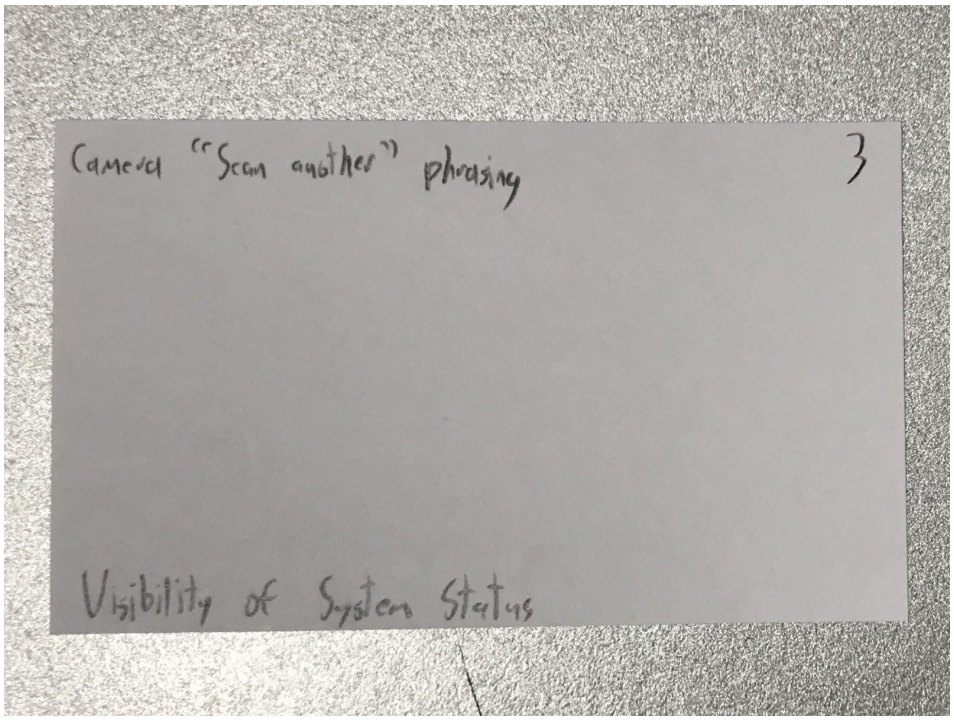

Visibility of System Status
4
Put the
unit next to
the quantity

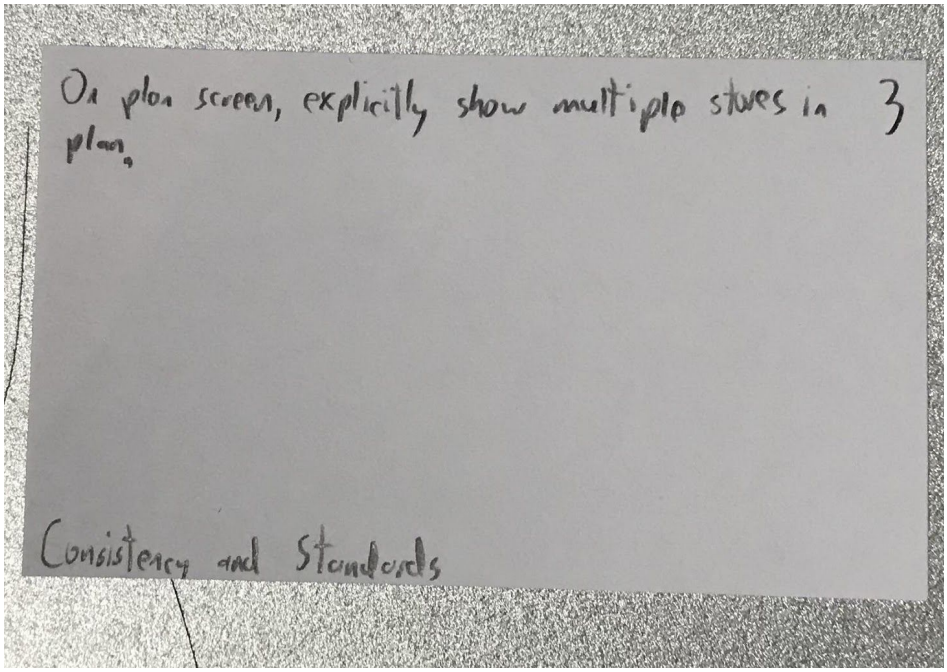

Consistency and Standards
4
What happens when you select add?

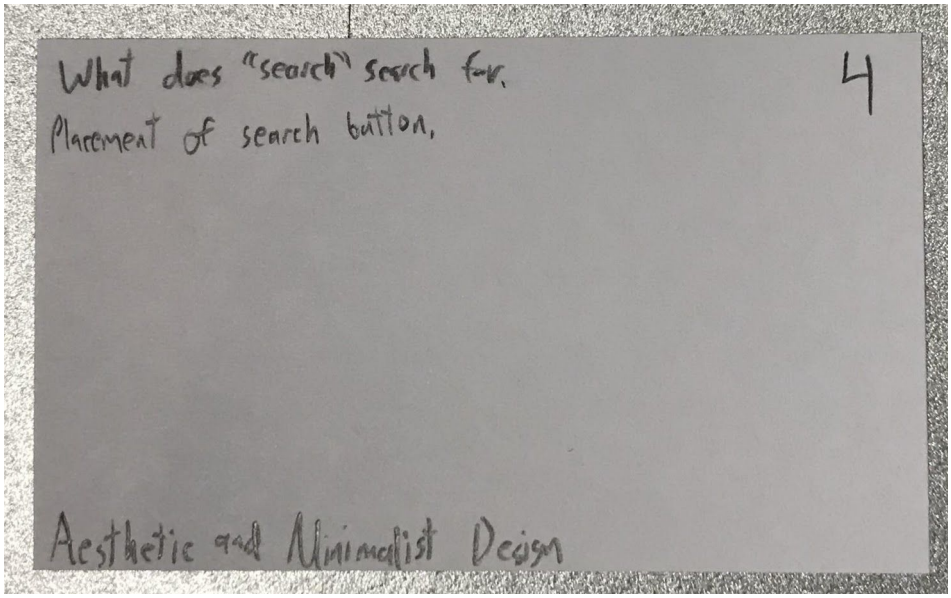

Display coupon count per item.
1
Visibility of System Status


1
visibility:
level = receipts scanned
make more consistent


2
show coupon savings on shopping list
Visibility of System Status


Camera "Scan another" phrasing
3
Visibility of System Status


On plan screen, explicitly show multiple stores in plan.
3
Consistency and Standards


What does "search" search for.
Placement of search button.
4
Aesthetic and Minimalist Design

Usability Testing Tasks

MetaGrocer

This is an app that helps users plan their grocery shopping trips. You search the items you want in MetaGrocer and the app will give you the best route.

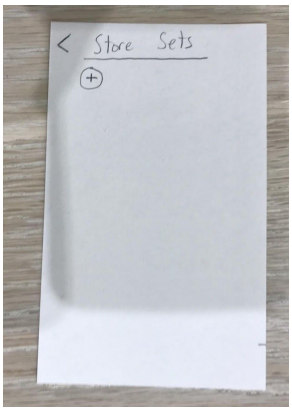
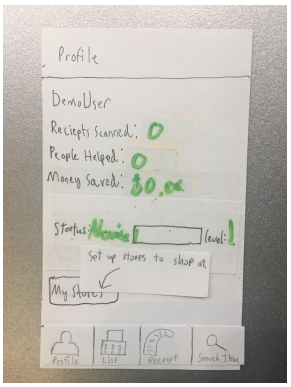
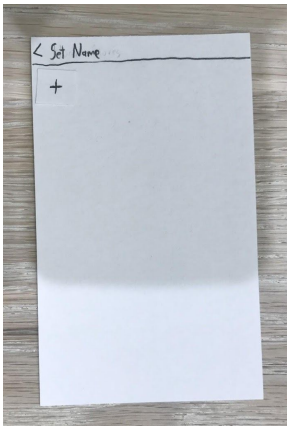
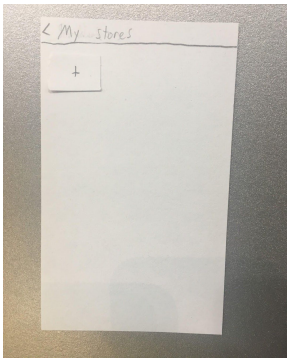
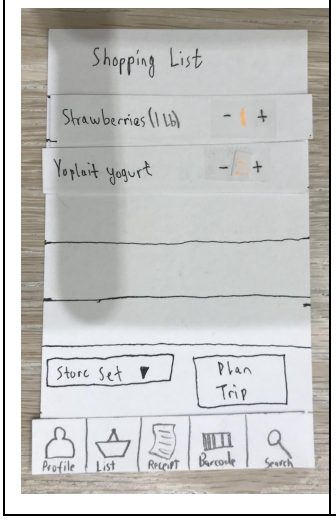
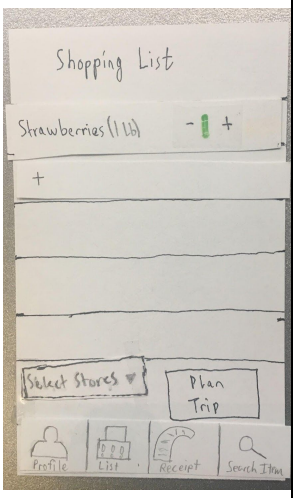
Task 1

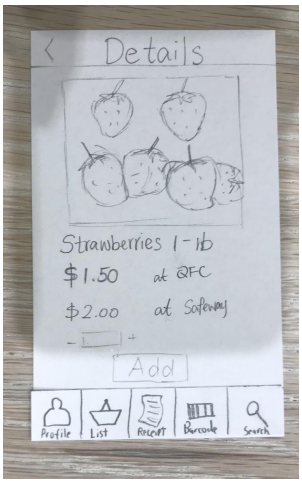
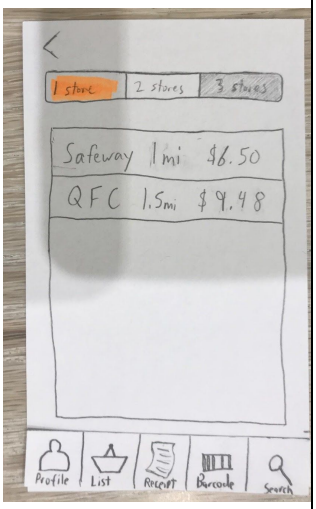
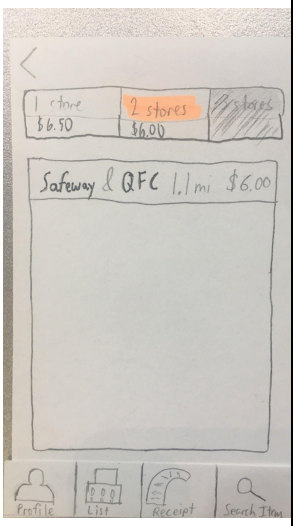
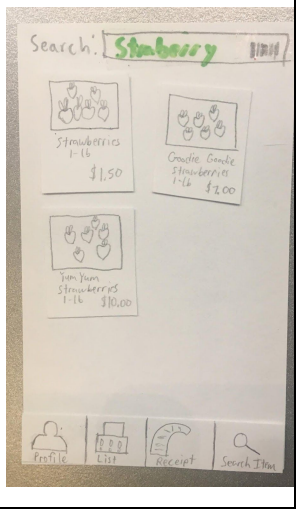
1. Add **QFC** and **Safeway** to the store list you want to go to
2. Add the cheapest **strawberries** to your shopping list
3. Add **two** of the cheapest **yogurt** to your shopping list
4. Begin the shopping trip
5. Find the **cheapest** route
6. Complete your shopping trip

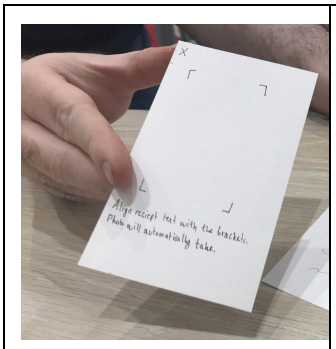
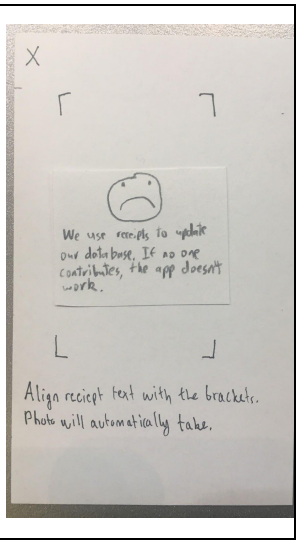
Task 2

Pretend you are already at Safeway, looking for Chip Ahoy cookies. You open MetaGrocer to look for coupons for the cookies. Add the cookies to your shopping list and use the coupon when you check out.

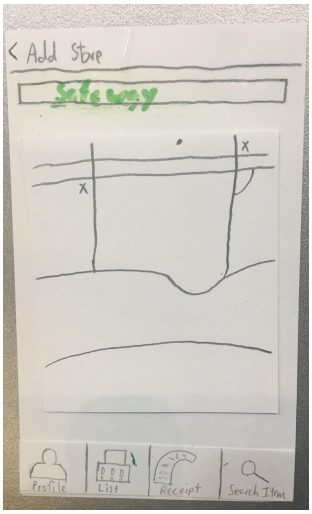
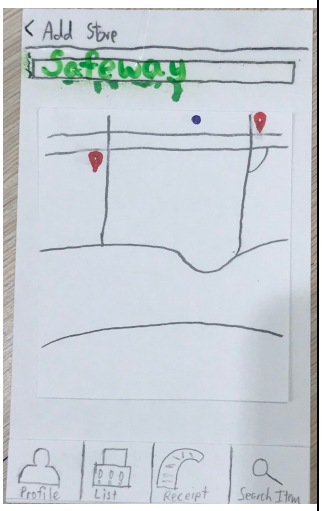
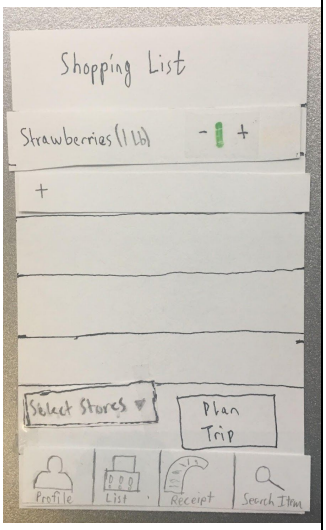
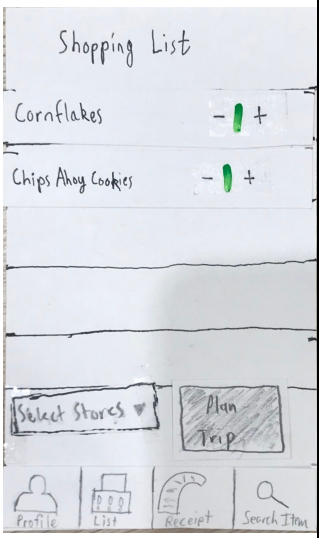
Usability Test 1

Before Image	Severity	Issue	Design Changes	After Image
	4	The participant expressed confusion about what a "Store Set" was and what the process of adding stores was accomplishing.	We decided that the usefulness of adding multiple sets of stores was not worth the complexity setting them up required. Users now add stores to one group, labeled "My Stores".	
	3	The participant tried to search for a store in the text field meant to name a set of stores. They were confused when their "search" didn't show any results.	Similar to the previous issue, we have decided to give the users one store group to edit. Users are immediately brought to a screen listing their stores when they choose to edit "My Stores", rather than a list of sets of stores.	
	4	The participant wasn't sure how to add items to their shopping list. Deciding to press a button and see what happened, they attempted to start their shopping trip with no items in their shopping list, generating an error message.	The shopping list screen now has a "+" symbol as the bottom entry in the list. Pressing this button brings the user to the search screen, just like pressing "Search Items" would.	

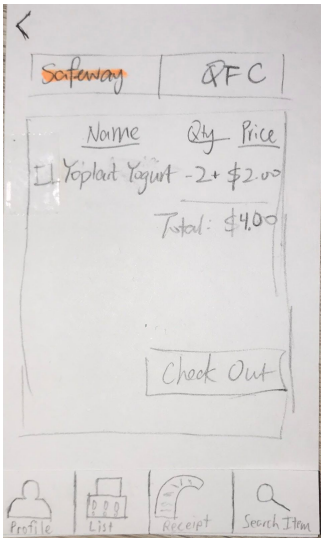
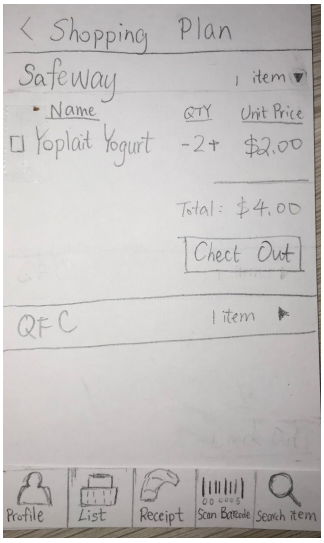
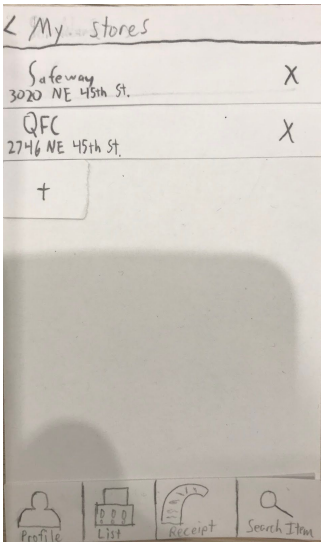
	1	When on the item details page, the participant would adjust the quantity, press a store name, and then hit "Update List". At this point in the process stores don't need to be selected yet, so pressing them was unnecessary.	Since the store names are not interactable, the participant pressing them has no bad effects. However, as we move into the digital mockup phase, we will keep in mind to clearly show that the names are not buttons.	N/A
	3	When choosing a route for the shopping trip, the user immediately chose one of the "1 Store" routes, rather than exploring the cheaper routes on the "2 Store" page.	To make it more apparent that other pages may have cheaper options, the cost of the cheapest route in that category is now displayed on the buttons.	
N/A	2	During the impulse buy portion of the task, the user used the text search menu, rather than using the quicker barcode scanner.	The barcode scanner is now part of the search bar. Users can find an item either by searching for it or scanning the barcode.	

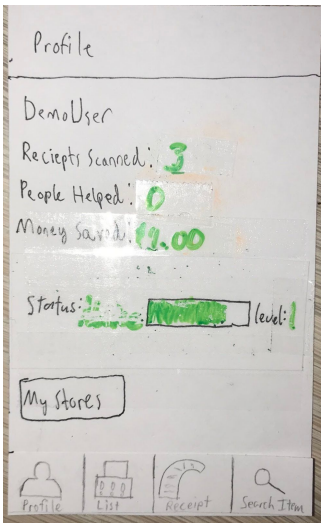
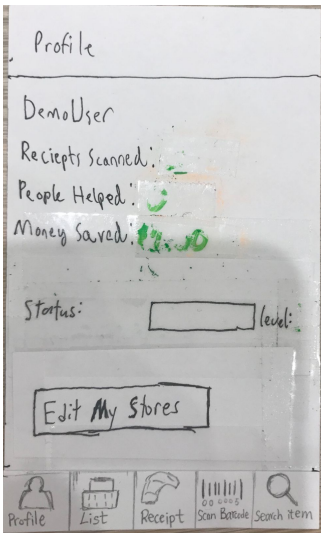
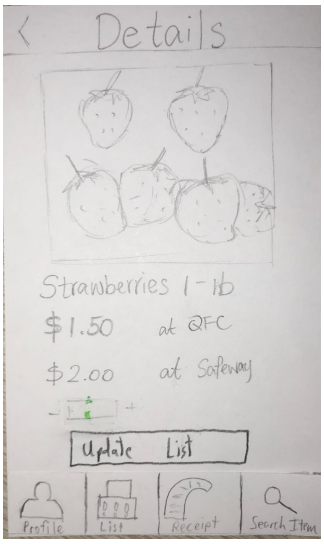
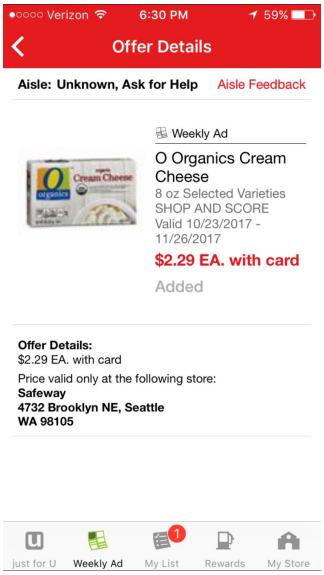
	4	The participant didn't want to scan their receipt after checking out. Since we rely on the users scanning their receipts, this is a huge problem.	If a user closes the scanning screen without scanning a receipt, we now display a textbox talking about the importance of contributing to the database.	

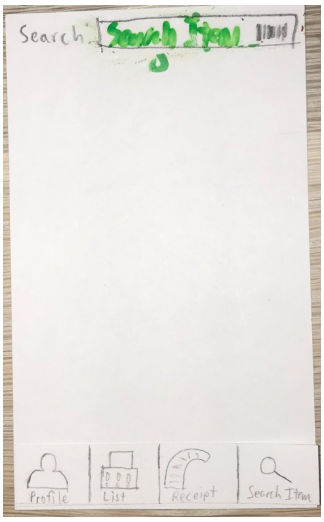
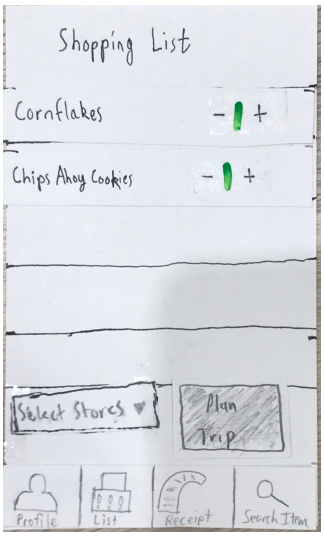
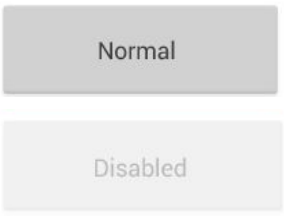
Usability Test 2

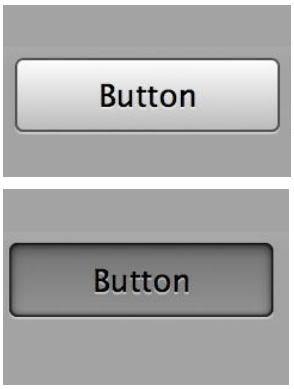
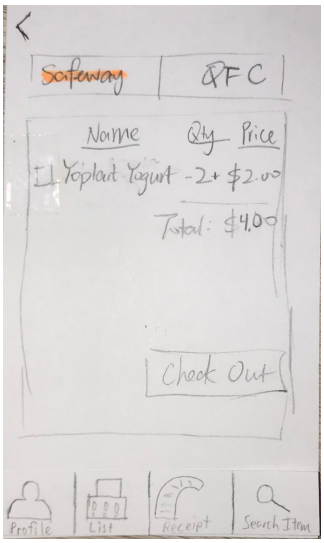
Before Image	Severity	Issue	Design Changes	After Image
	1	The participant asked us if the dot was them on the map. They were unsure what the dot was meant to indicate.	While it would be clearer with a real map, as the user would be able to recognize the blue dot as being in the same place they are currently at, we still decided to make changes to our icons. We now use the same icons as Google Maps. These include a blue dot for the user's location and red pins for the searched stores.	
	2	The participant attempted to start planning a trip without selecting what group of stores to use. We had to quickly make an error message pop-up, as continuing without selecting stores is not possible.	To avoid error messages in the future, the "Plan Trip" button is now greyed out until a store group is selected. This will pre-emptively communicate to the user that a trip cannot be started yet, avoiding the user running into an error message.	

Usability Test 3

Before Image	Severity	Issue	Design Changes	After Image
	4	After picking up all their items at the first store on the trip, the participant attempted to select the other store, rather than pressing the checkout button. This would cause them to miss out on the coupons that would have been displayed.	In response to a couple of issues on our shopping page, we decided to greatly change it. The new page now shows multiple stores on the same page that can be selected to show an expanded list of the items, as well as the checkout button. We believe that this indicates that the stores are in an order and should be "checked out" individually, rather than as a couple.	
		After adding QFC to the list of stores, the participant tapped the QFC entry again. This led them back to the map screen where the stores are selected.	This wasn't an action we had previously considered, so we were uncertain how the app should behave in this situation. We alter decided that store list entries cannot be edited. Stores can be added and removed from the list, but not changed. As a result, the participants action would have resulted in no change to the app's state.	N/A

	3	The participant went to the "Search Items" screen to find stores. They later said that they were confused by the "My Stores" button, thinking it would show them only stores that they had already selected rather than letting them find more stores.	We changed the button's label to say "Edit My Stores" to better communicate its use.	
	3	The participant would attempt to select a store at this page, which isn't something that is decided at this point in the process.	We believe that this issue comes from a lack of fidelity in the paper prototype. It can sometimes be difficult to tell what is interactable and what is not. As we move on to the digital mock-up, we plan to ensure that the text does not look clickable in any way, just like the "Offer Details" section on the screen to the right.	

	3	When adding an item at the store, the participant would use the text search to find it rather than the quicker barcode scanner.	This is an issue that has cropped up a couple of times. We've attempted moving the location of the scanner and even adding pop-ups with arrows pointing at the button, but nothing has worked. We have decided that the best course of action is to keep the scanner both in the search bar and in the bottom navigation bar., giving the user multiple ways to find it.	
	3	The participant tried to plan a trip without selecting any store group.	As with some other issues, we believe this is an issue with fidelity. When we create this screen in higher fidelity, the "Plan Trip" button will be faded until a store group is selected.	

	3	The participant pressed the already selected "2 Stores" button. This resulted in no change to the state of the app.	Again, this is a fidelity issue. In the digital mock-up phase, the buttons along the top like the ones in the image to the right, showing whether they are pressed or not.	
	3	The participant expressed confusion about why not all the items on the grocery list showed up on this screen.	To fix this we completely changed our Shopping Plan screen. Now, when the user enters this screen they can see all the stores in the plan, as well as how many items are to be purchased at each store. This will show the user that all the items in their shopping list have been incorporated into the plan. When a store is selected, the entry will expand to show the individual items to be bought as well as the checkout button for that store.