



Team R Project: Stage 5

12.01.2020

— CPSC 481

Sebastien Wong - 30040657

Danny Tran - 30027536

Noah Bensler - 30020684

Shamim Khalili - 30022743

Term: Fall 2020

TA: Philmo Gu

Team R Portfolio

<https://sebastienwong.github.io/cpsc481teamr/portfolio/>

Team R Repository

<https://github.com/sebastienwong/cpsc481teamr>

Executive Summary

Grocery shopping on a budget is an arduous task. With the rise of app culture, there is an increased expectation for tasks to be simple, efficient, and accessible, and shopping is no exception. Our project aims to create an application that immensely simplifies the act of grocery shopping on a budget. Specifically, it utilizes barcode scanning for efficient shopping list use and creation, as well as budget tracking and maintenance features.

With user surveys, error analysis, and all manners of prototyping, our application was designed with the user as top priority. This rigorous testing and constant iterating allowed us to create an app that is simple, easy to use, and accessible to many different types of shoppers.

Introduction

Let's face it, it's hard to save money. With insurance, rent, food, and Amazon deals, it seems like every month we're just scraping by. But with our app, we hope to change that (at least the food part). Our budget grocery list app is an easy and simple way to keep up with your groceries, *and* your wallet. Make and save lists, scan barcodes for easy shopping, and track your budget along the way! Our app will save you time, sanity, and most importantly, money!

Description of your design problem

When shopping, it can be difficult to remember everything you need and a regular shopping list can be slow to use. Not only that, but if you are shopping on a budget it can also be difficult to keep track of the cost of all the items you have chosen. The service is trying to solve these issues in a way that is easier, quicker, and more intuitive to use than traditional shopping lists.

Description of your design solution

When you run out of a product at your house, simply scan the barcode and it adds it to your grocery list. Later when you're at the store, you can just scan the item that you are buying and it will recognize the same item was on your grocery list and check it off. It would be introduced as a much simpler yet fuller grocery list app so that you never forget the products you need. It can also be used to find coupons or other savings on products that you have on your list.

End-user and stakeholders

The stakeholders for this project are people who buy lots of groceries at once, people shopping on a tight budget, people with poor memory, and grocery stores. Traditional shopping lists allow shoppers to record what they need to buy. However, they can be slow and cumbersome to use, especially when buying large amounts of items. Because of this, an app which removes items from the list as the user shops, makes it much easier to keep track of what they have and have not yet added to their cart. Shopping on a budget can be difficult. Knowing the cost of the groceries before shopping allows the user to change the list before they get to the store and so, stick to their budget. Shopping can be difficult for people with poor memory. An app which provides a simple and easy-to-use interface can greatly improve the shopping experience for those people. Finally, grocery stores are interested in keeping as many customers as possible. Because of this, an app that improves the shopping experience for their customers would be of interest to them.

User research methods and process

We picked "Ask: Surveys & Questionnaires" in order to quickly and efficiently gather a large amount of answers from potential users about grocery shopping habits, preferences for grocery lists, budgeting practices, and other relevant information. We performed this card with a Google Forms survey distributed to our classmates. We picked "Learn: Error Analysis" because it allowed us to find potential errors and misuses of the app which would be significant to consider when designing the interface in order to avoid such errors from occurring. We picked "Try: Paper Prototyping" because this method was a simple way to practically consider the interface for the user and begin articulating how the features and elements of the app would be visually represented. In our appendix, we have attached the questions/results of each method.

User research findings

We picked these three method cards because they allowed us to conduct research while following safety measures for the pandemic. In terms of the complementarity of these method cards, it was clear that the insights from one method card would help the team in the next method card we were doing. For example, when the team was conducting error analysis, the different insights gained from the survey helped us in generating a more diverse variety of errors and mistakes that could occur. In another instance, the answers from the survey and the identification of some errors both helped us in considering how to design certain parts of the app in paper prototyping so that they would be clear for the user. One of the things we think could have gone better is our sample size for the survey, since it would have been valuable to get a few more responses. Overall, we have no complaints with the IDEO Cards that we chose and how they performed. Each of the IDEO Cards taught us something that we would otherwise have missed or not thought about.

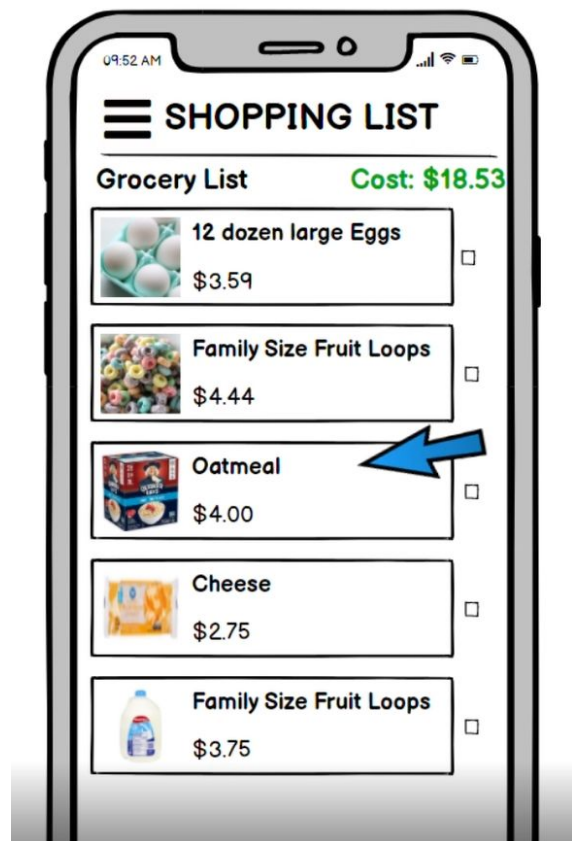
Important design choices and justification

Since we learned during our research that there were a few particular features that would most benefit and were most desired by a potential user, during the ideation process we narrowed down what these ideas were and started designing the app by separately prototyping the 3 major user tasks vertically, and adding others later. Each of these tasks included design elements such as buttons to add and remove items and lists, scanning an item's barcode with the device's camera, searching, and others. These elements we designed in a very simple and familiar way, much like one would find in another app so that users could quickly and efficiently navigate through the app, without having to learn new ways of doing things they are already familiar with. This was because during our research we found that potential users were familiar with other shopping apps like Walmart's, Amazon's, and Superstore's apps. Another significant design choice was in regard to the user's budget when adding items to the shopping list. Since this was a very significant aspect of our app, we decided to display it at all times on the top of the screen when the user was viewing the shopping list, and for it to be updated whenever a change was made to the list to help the user keep track of the budget easily. Furthermore, another design choice included placing a bar graph of the shopping history of the user, allowing another way to track grocery shopping habits in context of the budget limit with this same purpose of more accessibly allowing the user to track their grocery finances.

Low-Fi design, lessons learned

The lessons that we learned while making our low-fi prototype mainly revolved around the principle of balancing functionality and usability. Much thought went into just the creation of elements, such as buttons and titles. Questions such as, "Do we make it an icon or a box?" and "Do we add text?" and most importantly, "Will the user know how to use it?" came up while we designed each of the screens of the low-fidelity prototype. We had not previously considered that there would be so many small details to think about to make sure the design is functional as well as usable. Furthermore, we learned that there was an efficiency in space we needed to consider when prototyping. Especially with paper-type low-fi prototypes, we needed to be able to show everything we wanted the user to see and interact with on one screen. This meant sacrificing, compacting, shrinking, and cutting elements, to include really just the essentials for the user to understand and act on whatever task

needed to be done. It came back to the idea of making sure to balance functionality with usability. Finally, another lesson learned was the importance of flow. Being able to see what we had been ideating in an interactable form was quite helpful, and to be able to interact with and use a somewhat functional version of the app gave us a lot of information about the sequence of steps the user had to go through for a specific task. It helped us understand the importance of the first screen a user sees, for instance, as well as how important it is to choose the right screen to follow. This was a matter of keeping things efficient to ensure the app was not wasting the user's time, and again, came back to the idea of balancing functionality and usability.



Hi-Fi design, lessons learned

For our Hi-Fi prototype we used Adobe XD. While creating the prototype we learned many lessons, mainly revolving around creating a consistent experience and look across the entire prototype. We took a lot of what we learned from the Lo-Fi prototype and implemented it in our Hi-Fi prototype. We implemented all the small details, including

buttons, scrolling and easy to read text. What we did not consider at first was how the user could recover from errors. While we were creating and testing the prototype, many of the small details that are important for first-time or newer users were not considered during the initial stages of the prototype. For example, questions such as “How would I undo checking off an item on the list if I misclicked?” or “Can I reach the navigation menu from anywhere I want?” was not really considered at first and that is what helps to create a cohesive experience for the user. After the Heuristic evaluation we were able to identify these issues and address them effectively so as to not create a jarring experience for any new users. Creating a complete and consistent user experience during the Hi-Fi prototype was a most important lesson we learned.

Heuristic evaluation and findings

The Heuristic Evaluation process was completed using the heuristic evaluation form. This process involved completing ten steps which each evaluated a different aspect of the prototype. The prototype was evaluated one step at a time, finding all the instances where each rule was either applied or violated. Each time this occurred, a brief description was written, detailing the application or violation. The evaluation was done by Noah and Danny (See Fig. B and C).

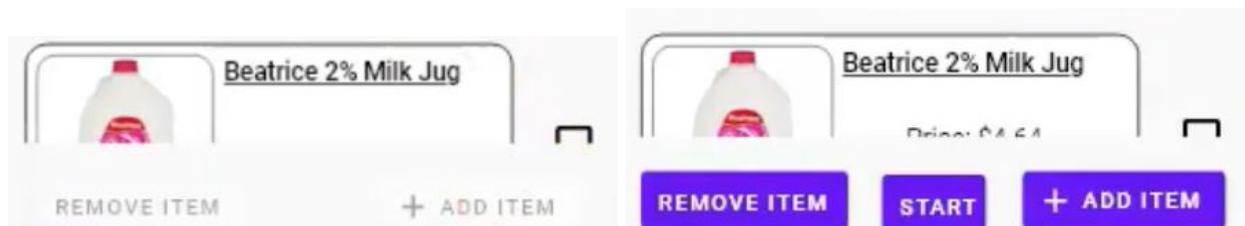
The results that were obtained from this process were consistent between the two independent evaluations. Both found that the prototype lacked functions to prevent errors from occurring or to easily recover from errors once they had occurred. They also found that many of the buttons lacked functionality. Overall, the evaluations were positive and found that nearly all the rules were applied consistently throughout the prototype.

The process of reviewing was rather straightforward, and Sebastien and Shamim acted as the reviewers for heuristic evaluation (see Fig. D and E). After reviewing and sorting the issues separately, we found that we shared many of the same observations and many of the issues were graded with similar severity.

Our findings revealed that the most severe problems were issues where functionality was expected but not delivered in relation to the usability and desirability. For instance, check boxes not being able to be unchecked, or non-functional back buttons. We agreed these would be the most severe, as they are needed to keep the flow of the prototype functional and understandable. We agreed that the least severe issues were issues regarding documentation or error messaging. While these issues are important, for a user this information is not overly important, since they just want an app that is intuitive to use and will work. And in the middle ground of severity, we agreed that issues regarding convenience and utility belonged here. These issues do not break the functionality of the app, but they can help improve usability and flow, and should not be unaccounted for.

Changes based on heuristic evaluation

After our Heuristic evaluation we decided to change the problems that were most severe. This included adding confirmation screens when the user is removing items from the list, making all active buttons clickable, making all inactive buttons look different from active buttons, making the navigation bar accessible when it makes sense, and flushing out the adding-item-by-searching feature.



(Greyed out Inactive vs purple active buttons)

These features were the ones that impacted the user experience the most and allowed for a more consistent and complete experience overall. We would have liked to clean up the look of the prototype, but due to time constraints, we were not able to do so.

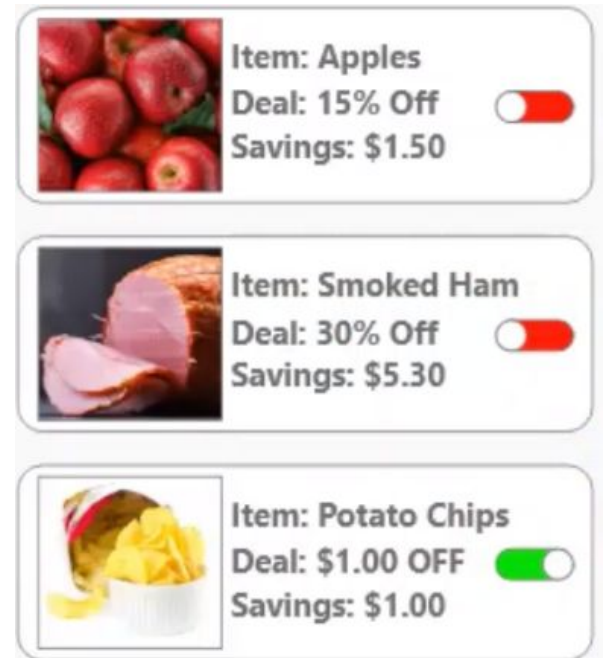
Be critical

Changes Made

The changes made to our final Hi-Fi prototype in Stage Five can be categorized into main feature changes and minor detail changes. In terms of main feature changes, there were two more tasks that were implemented. First, the Sorting feature for the shopping list was fully implemented, including alphabetical sorting and price sorting (from lowest price to highest and from highest price to lowest). The second task that was implemented was the Shopping Mode feature. Once users are finished making the shopping list, they can tap "Start" and enter the Shopping Mode. Under this mode, users can scan items to remove, making for a better and simpler experience while they are at the grocery store actually buying the products on the list. Once they are finished with the Shopping Mode, the app displays the final cost for this shopping trip and how much the user went over-budget or under-budget.

In terms of changes of more minor details, there were several changes in order to make for a more visually appealing and clearer interface. For example, in the Coupons section of the app the toggle button changes colour and the button now also flips position, which was a

visual detail not present in the last stage's prototype. Another change was the formatting of the text in the shopping list, which now includes bolded and underlined product names, prices in centered position, all in order to make the text in the shopping list look more visually polished in both size and layout so that users have a cue as to where their eyes should be drawn. The app's logo is now featured more prominently in size on the menu screen, which also helps lessen the large block of grey area that made the menu look sparse. In terms of color scheme, the harsh white background of the previous stage was changed in Stage Five to a very light grey-white color for a better visual experience on the users' eyes. This also helped the item boxes (which are white) to pop out more from the background, thus giving a very subtle cue as to where the eyes of the users should be looking.



Changes That Should Be Done in the Future


The changes that should be made in the future include a more complete integration of the shopping templates feature with the shopping list, so that a certain shopping template can be added to the user's current shopping list to make their grocery shopping more efficient. Another change should be the inclusion of a small Undo button for when a user removes an item from their shopping list and would like to get it back quickly.

Changes That Could Be Done

Some of the possible changes that could be made to better the app in subtle ways include adding help options in the menu or some sort of tutorial when a user first uses the app in order to introduce and explain how to take advantage of some of the unique features of the app. Another change that could be made would be to implement a distinct health section for the app.

Conclusion

It is interesting to note that although our idea for a grocery shopping list mobile application appeared quite simple at the very beginning of the project, it provided this team with ample experience within only a few months to learn about researching and designing a prototype with strong user experience. Even for ideas and tasks that seemed basic and simple, we soon learned that this work required a great deal of attention to subtle but important details. Learning how to conduct research relevant to our project, to use



Balsamiq and Adobe XD, to communicate and coordinate our efforts as a team, to refine our presentation skills, and to create high quality reports, were among the most valuable learnings and experiences of this project.

Appendix

Questions Used for the Questionnaire

1. How often do you go shopping?
2. Do you use a shopping list or any other type of shopping tracker?
3. Do you have a budget when shopping?
4. What is your average budget per shopping trip?
5. If you went over budget, which items would keep? Which ones would you let go of?
6. When do you use coupons? Would you use coupons more often if you could?
7. What functionality would you find useful in a shopping list tracker?
8. Do you buy the same brand items every time you go shopping?
9. Who is the one that goes shopping in your family? How old are they?

Error Analysis

- Using the wrong template
- Accidentally saying you bought something when you haven't
- Getting the wrong item or too many of the same item
- Multiple people in one household buying duplicates of items
- Going over budget without knowing it
- Finishing the shopping trip without everything you need
- Not applying or showing a coupon when one was available
- Not being able to add items to a shopping list

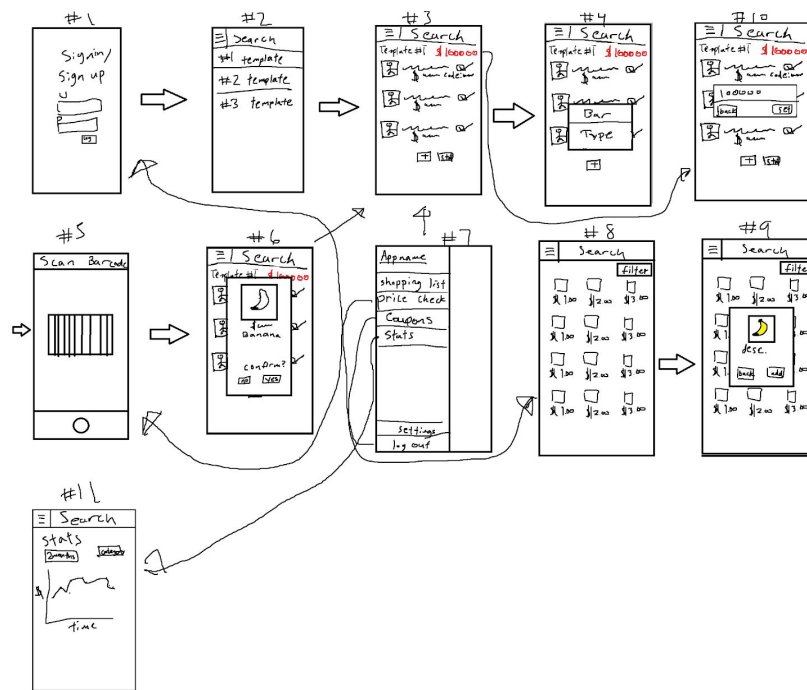


Figure A: Digital paper prototype for grocery list app.

Rule of Thumb	Is this rule being applied? How so?	Is this rule violated? How so?	How can this rule further <u>improve</u> usability, utility and desirability?
1. Visibility of system status	Yes, it is applied. The screen changes quickly when a button is pressed.	NA	It improves Usability by providing immediate feedback to the user.
2. Match between system and the real world	Yes, it is applied. The pictures of grocery items match real world items.	NA	It improves usability by providing the user with recognizable items and symbols.
3. User control and freedom	There is a back button on several screens and the menu is always accessible.	There are functions to remove items from the list but there is no undo button in the program.	It improves utility by allowing the user to undo mistakes made while using the app.
4. Consistency and standards	The button size, shape, font and color <u>is</u> consistent.	The buttons are not always functional.	It improves usability because its consistency makes it easier to learn how to use.
5. Error prevention	NA	No it is not applied. Users must confirm adding an item, but not removing one.	It improves usability and utility by preventing errors from occurring.
6. Recognition rather than recall	The rule is applied. Options are visible and easy to find	NA	It improves usability by making the various functions easier to access.
7. Flexibility and efficiency of use	Templates allow users to quickly make a list with frequently used items.	There are no accelerators in the rest of the app.	It improves utility by increasing the number of available functions for experienced users.
8. Aesthetic and minimalist design	The rule is applied. There is no unnecessary information displayed on the screen.	NA	It improves desirability by being aesthetically pleasing.
9. Help users recognize, diagnose and recover from errors	NA	The rule is violated throughout the app. There are no error messages.	It improves utility by helping the user recover from errors.
10. Help and documentation	NA	There are no help options or documentation.	It improves utility by helping users who encounter errors or don't understand how to use the app.

Figure B. Team R Heuristic Evaluation done by Noah

Rule of Thumb	Is this rule being applied? How so?	Is this rule violated? How so?	How can this rule further <u>improve</u> usability, utility and desirability?
1. Visibility of system status	Yes, this rule is being applied because all feedback is provided quickly	<u>Yes</u> because some of the buttons don't work and that may cause confusion with the user, but this is a limitation of the prototype.	It improves the usability because the user is not second guessing themselves or waiting around for something to happen.
2. Match between system and the real world	Yes, because all forms of interaction used are ones that users should already be familiar with, like back buttons, scrolling and distinct clickable buttons	no	It improves the desirability because it allows the user to carry over what they have learned from other systems and use it in others.
3. User control and freedom	Yes, the coupons page allows you to select which coupons you want applied and you can switch them on and off	Yes, in the shopping list page, after you check a box you cannot uncheck it.	It improves both the usability and utility because the user <u>is able to</u> correct their mistakes easily.
4. Consistency and standards	Yes, <u>all</u> of the back buttons used are the same and all buttons that have different functions are labeled differently. The system is metaphorically consistent because it is <u>similar to</u> real world objects.	no	It improves both usability and desirability because the user isn't guessing if a button does the same thing as another one and it allows that to carry on skill learned from other places.
5. Error prevention	Yes, in the adding items section, there is always a confirmation popup, screen, or button that allows the user to make sure they want to do that action.	Yes, in the removing item screen there is no way to go back after selection an item to remove.	It improves the usability and utility because the user <u>is able to</u> make changes if they have forgotten something or done something they didn't mean to.
6. Recognition rather than recall	Yes, because all clickable buttons are clearly visible for the user and labeled such that they tell the user what they are intended for.	no	It improves the usability because the user can see what they can and cannot do in certain situations, which guides the user through the system.
7. Flexibility and efficiency of use		Yes, because there is only one way to do things and there isn't any way to complete tasks faster	It improves utility because it lets advanced users' complete tasks that would otherwise be slower.
8. Aesthetic and minimalist design	Yes, all screens have all the information needed to complete a task on the screen and nothing more. Ex. You can't select templates when adding an item to the list.		It improves usability because it forces the user to only be able to do a few things at a given time instead of overloading them
9. Help users recognize, diagnose and recover from errors		Yes, there are no error messages telling the user that something cannot happen without something else being done. You cannot remove an item without selecting one first and there is no error message that states that to be the case.	It improves usability because the system is telling the user what the system is expecting and helps the user make less errors in the future.
10. Help and documentation	Yes, in the adding items pop-up, there is a bar that has a grey text within the bar to tell the user that it is a search bar, and in the edit budget screen there is a \$ sign next to the bar to tell the user to input numbers		It improves the usability and desirability because the system is hinting at what the system wants and how to use the s

Figure C. Team R Heuristic Evaluation done by Danny

Severity	Problems
Critical	<ul style="list-style-type: none">• In the shopping list, you cannot uncheck a box• There is no way to go back after selecting an item to remove• Buttons are not always functional
Serious	<ul style="list-style-type: none">• Users must confirm adding an item, but not removing one• No undo button for removal of items
Minor	<ul style="list-style-type: none">• There are no error messages• There are no help options or documentation.• There are no accelerators in the app• There is only one way to do things and there isn't any way to complete tasks faster

Figure D. Team R Heuristic Review done by Sebastien

Shamim's Review of Evaluations

Ratings	Severity
0	Doesn't seem to be a usability problem
1	Cosmetic problem
2	Minor usability problem
3	Major usability problem; important to fix
4	Usability catastrophe; must fix

(Severity Ratings as seen in slide 23 of Heuristic Evaluation lecture slides)

After reviewing the evaluations, all the problems were identified and then, if repeated or slight variations of the same problem, were combined into one problem and rated according to severity.

Problem	Severity Rating
User cannot uncheck a box on shopping list page once checked.	3
No way to go back after selecting item to remove. There is no undo button for when items are removed.	3
No confirmation button for removing items.	3
Some buttons don't work because of prototype limitations. The buttons are not always functional.	3
There is only one way to do things and no way to complete tasks faster. There are no accelerators in the app other than templates feature.	2
There are no help options or documentation.	2
There are not error messages communicating something cannot happen without something else being done.	1

Figure E. Team R Heuristic Review done by Shamim