Grocery Grabber

Carly Moss
Northeastern University
Boston, MA
Moss.ca@husky.neu.edu

Arash Yamin
Northeastern University
Boston, MA
Yamin.a@husky.neu.edu

Matt Clamp
Northeastern University
Boston, MA
Clamp.m@husky.neu.edu

Jonathan Millman
Northeastern
University
Boston, MA
Millman.j@husky.neu
.edu

Liam Fratturo
Northeastern University
Boston, MA
fratturo.l@husky.neu.edu

ABSTRACT

In this paper, we describe the process of designing a mobile application to assist with going grocery shopping. This is an overview of how we made design choices, implemented those choices in various stages of prototyping, and evaluated the usability of the application at each step in the process.

Author Keywords

HCI; Human-Computer interaction; grocery shopping; mobile applications

ACM Classification Keywords

C.5.3:iPhone

INTRODUCTION

The Grocery Grabber application was designed to make the task of going to the grocery store an easier one. Most adults choose to buy their own groceries, so grocery shopping ends up taking up a large portion of their time, especially when done inefficiently. We wanted to tackle the most important challenges that arise when going grocery shopping, such as finding items in a store, sharing a list, and comparing prices. This process was concerned with the user interface design of the mobile application as opposed to the back-end programming aspect of the application.

PROBLEM

We found that a common issue that we all shared was that going to the grocery store is often an inconvenient and frustrating task. Many people find it hard to keep track of the different prices of groceries at stores, and never seem to know if they are getting a good value. It can also be frustrating to attempt locating products in stores as they comprised of so many aisles. People end up wandering around and backtracking to get different items, as well as constantly asking different employees for item locations. Traditional aisle signs only help vaguely when looking for items, but some categories are misleading and specialized items such as foreign foods are much more difficult to find as they could be in various aisles. For families, it can be difficult to ensure that your loved ones get the right products at the store. People often forget which milk is the correct milk to buy, or they will leave an item off of the list entirely. It is also easy to buy the wrong thing at the supermarket and only find out when it is too late that it is an inferior product. We wanted to design an application that addressed these various problems and made the process of going to the grocery store a more convenient and efficient one.

USERS

Our target audience for this application was anyone who shops at grocery stores and wants to better manage their money or their time. Our primary focus was targeting this application towards college students who are on a budget and looking for the best prices on their groceries, as well as parents who are trying to plan the quickest and most cost efficient shopping trip to get food for their families. Our personas ranged from age 20 to age 50, but our most accessible audience was college students, since we have plenty of those on our Northeastern University campus available for user testing at any time.

TASKS

There were many tasks that we thought would be useful for this application, but there were a handful of essential tasks that were absolutely necessary. First was adding and deleting items from a list, because it is important to keep track of a shopping list before going to and while being at the grocery store. This required a browsing feature as well, that really relied on having a database of food items available at all stores. Another element to the list task was sharing the list, so having the ability to send a list to a family member or a friend in case the user wants to split up the shopping process. We wanted to have the ability to share a list through SMS, Facebook, Twitter, or e-mail. This added a somewhat social aspect to the application. The next task was a rating system for items so that users could compare one food item to another, see what other users liked, and add their own ratings if they felt that a particular food was either very good or very bad. Another important task was finding local stores and their information, to compare the average prices of stores and to see which stores are closest to the user. Once in the store, we wanted to implement an in-store item locater that would help users find items and help them navigate through the aisles of the store. This relied on technology that does not necessarily exist yet, because GPS is not precise enough to accomplish this task. However, we were aiming to create our ideal prototype, so we were not concerned about these practical constraints just yet. Had we actually implemented the back-end of this application, we would have had to deal with that feature. Instead, this was very much about the design of the overall user-interface, so we focused on that.

DESIGN

Our design went through several stages but the following discussion will emphasize our original design sketches and how they evolved into the final design. Initially our design did not include a home screen since we anticipated that user would most frequently want access to their shopping list. Thus the shopping list functioned as a home screen and all other functions were accessed as buttons in a row on the bottom.

However, several of our test users found this confusing, so the final design of our interface incorporates a home screen with links to each of the major functions. The home page also incorporates a header with a shopping cart logo and the name of our app. These elements help to orient new users as to the purpose of the app. Below the header is a list of functions presented in a common iOS style. Activating each function brings up a screen dedicated to the given function.

Throughout the interface we keep a consistent navigation bar on top with back and home buttons. We use black and white for most text content and use red and yellow for buttons, and other graphical content. Our initial design used red more heavily but after the heuristic evaluation it was clear that it was distracting and hard to read in places. We use black text on a white box with a caret to the right to indicate a link or transition to more detailed information.

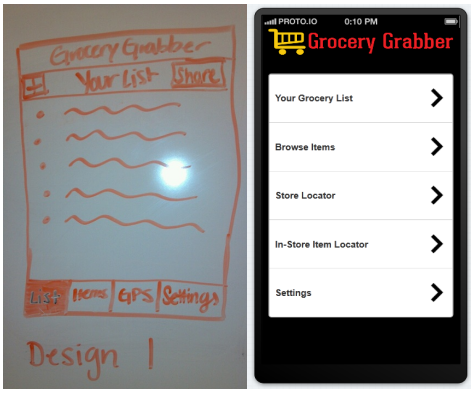


Figure 1: Initial and final designs of "home" screen

The shopping list screen remained substantially similar to our initial design sketches. The biggest change was putting all the "action" buttons on the bottom, and reserving the header for general navigation. After user complaints, we decided to consistently include a back and home button in this top navigation bar. We adjusted the order of the buttons on the bottom to line up the check boxes with the select all and put the most common tasks first.

From the shopping list page, users can share their list via multiple services displayed in popup menu. Selecting a service would transfer data and control to its sharing functionality.

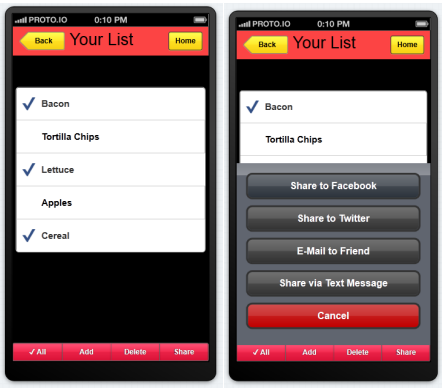
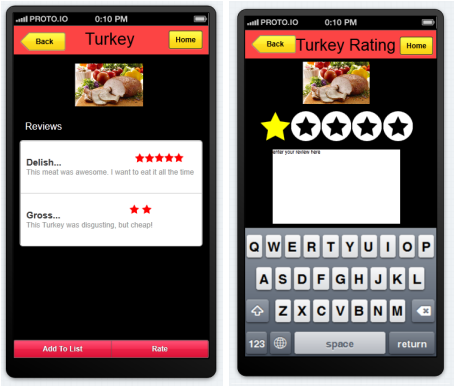


Figure 2:shopping list and sharing

The browsing function also stayed close to our original design sketches. The main differences stemming from the decision use a back button and a home screen for navigation rather than ever present links on the bottom. The browse works via a sequence of progressively narrow categories that the user selects until the level of individual items in reached. At this point selecting an item brings up a info screen about the item with available actions such as add a rating or add to shopping list. This screen went though minor revision: star and plus icons were replaced with less ambiguous text labels, and the action buttons were moved to the bottom of the screen for consistency.



Figures 3 and 4: item details/adding a rating/review

If the user chooses to add a rating they are presented with a five star scale and a text area and on screen keyboard to enter a review.

The store locator presents a Google Maps-style map with icons indicating nearby stores. Selecting a store brings up a screen with details including store hours, and aggregate price level. This function was unproblematic and no significant changes were made from the sketches.

The in-store item locator was in many ways the most challenging section to design because it was the most novel. Our initial design and paper prototype was similar in some ways to our final implementation but different other important ways. Firstly the decision was made after our paper prototyping session to throw out our attempt at a cure metaphor, the Grocery Positioning System, as it was confusing, and users didn't understand what it was for.

Instead we went with the more straight descriptive "in-store item locator." Secondly, the initial vision was that users would click on items directly on the map to indicate that item was found physically, and to have the map update with the next section of the in-store route. Users did not understand this in the paper-prototyping, although maybe they would have in a computer based prototype, so we decided to move the interaction into a separate list of items below the map. This was the last section of the interface implemented, so we don't have detailed user testing results of this revised design.

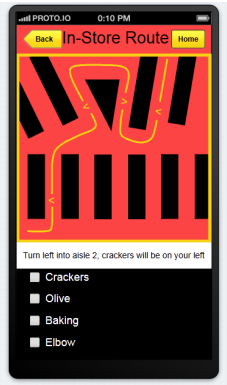


Figure 3: in-store item locator

Finally, there is settings screen, incorporating an about link and multiple account/login sections.

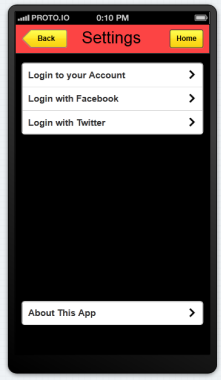


Figure 4: settings

IMPLEMENTATION

We used a mobile app prototyping tool called Proto.io to implement our application prototype. Proto.io is a browser-based, drag and drop development environment, with support for multiple mobile operating systems. It allows for multiple collaborators on a single project, provided they don't all use the editor at the same time. It supports continuous live testing of each saved version of your app, and also creating a published version for external testing at any time. While it offers a great deal of interactivity for its various built-in forms, it isn't particularly good at dynamically displaying content; many of the 'dynamic' aspects of our prototype had to be implemented by creating several slightly modified copies of static pages, and seamlessly switching between them to maintain the illusion of a single page.

Given that dynamic interactions were difficult to render, our prototype ended up dropping several of them between the paper and computer stages. Several pages had pop-up windows which would link to further pages in the paper stage, but many of these were replaced with direct page to page transitions in the computer version.

Our initial design for the app of course called for a completely user-created shopping list, letting users add or remove any items they wished. Users could then use the list to generate an in-store route, plotting out the most efficient way of navigating the store when gathering their desired items. We were able to convey this idea in the app with a single fixed in-store map and item list, but again dynamic generation of content eluded us. We had also planned to have a fully interactive map of nearby stores, with scrolling and zooming on par with most modern map sites, but that too ended up being not a behavior supported by this tool; our map ended up being a static image with location markers placed on it.

Another major drawback of Proto.io was its lack of support for simultaneous collaboration on a project; while it allowed multiple accounts access and editing privileges, more than one user working at a time would cause pesky notifications about overwriting another user's work, and several times even resulted in corrupted saves that we were unable to edit or preview (fortunately we could revert to an earlier copy whenever this happened). We eventually settled upon a system where everyone would check with the group before they opened the editor, so that nobody else's work would be overwritten.

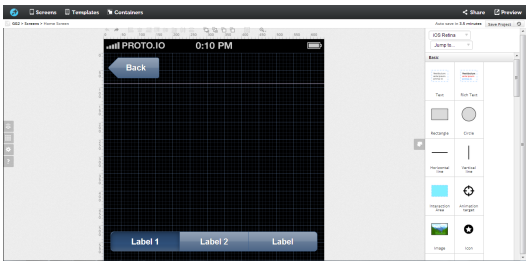


Figure 7. The Proto.io editor interface, with work area on the center left and item toolbox on the right.

On the upside, Proto.io provides a great number of standard forms and buttons for various mobile platforms, including Android and multiple

iPhone iterations. To add an item to the prototype, users drag it from the item toolbox on the right of the interface to the working area. Users can then modify various attributes, such as its width, height, color, displayed image, number of options, and screen location. They can then add interactions to item types that allow for them, such as buttons or lists. Interactions allow the app to change state when a user clicks, taps, or swipes the area with the interaction attached. The interaction can cause items to change their visibility, animate, be highlighted, it may play video or audio, or it can cause the app to navigate to a different screen. Multiple results from a single user input can be implemented by adding one or more callback actions after the initial one; for example, a user tapping a button might have an initial action of an object being highlighted, with a callback to a play audio action, giving the overall result of a single 'highlight with sound' interaction. One of the few built-in functions that was even remotely dynamic came to light here, as you can set a screen navigation interaction to return the user to the previously viewed screen, no matter where in the app's organizational hierarchy it's located.

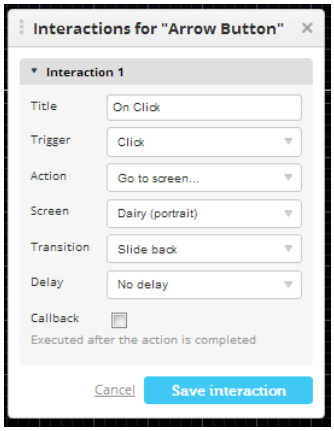


Figure 8. The Proto.io interaction editor, here set to direct the user to a different screen upon clicking the arrow button.

We used these dynamic back buttons in most of the pages in our design, but had to replace them with static ones in a few places where users tended to reach the page via a dialogue it wouldn't make sense to repeat. Beyond that, we used page switching interactions on buttons and list items to enable users to navigate the app, as well as to implement some otherwise tricky features; the 'select all' button on our list screen simply takes the user to a copy of the list screen with every item selected. We used interactions that modified object visibility in order to implement the in-store item map; as the user makes his or her way through the store and checks off the items so far, more updated versions of the map become visible.

A usability problem that were noticed as a result of our use of this platform was that often the built-in check marks on lists were too small, or were positioned in an unusual way, leading to users not noticing them or interpreting them correctly. Where possible we tried to rectify this by selecting new graphics from the tool's extensive database, but in a few we ended up dropping the built-in list entirely and constructing our own series of text boxes and check marks to get around the problem.

In addition to this, the way the editor is set up by default makes it difficult to judge the scaling of your app; you can zoom in and out from the virtual phone screen, but by default it's fairly zoomed in, so adding text that appears to be clearly legible can result in problems with readability in the actual prototype preview. We received a great deal of feedback about this during our second round of user testing, and ended up increasing font sizes almost across the board in order to rectify it.

Overall, while having a more complete toolset when it came to dynamic content and handling user input would have been useful, we are fairly satisfied with our choice of Proto.io as a development tool. It provides a simple, easy to pick up interface for constructing and testing mobile apps, and doesn't require you to reinvent the wheel for every common item you might want to place. All that was lacking was the ability to implement certain functionality.

EVALUATION

We as a team conducted user testing and evaluations separately on students who go to Northeastern University. Because of the small amount of resources, we felt this was the only way that we could fully test users on the product.

We designed sample steps for these users to complete, having them working through it on their own. Once they did complete these steps, we had them take a survey, and interviewed them

on what they thought of the design. Both forms of feedback were used for us to determine what to fix in the final product that would make Grocery Grabber that much more reliable and easy-to-use.

USERS

As mentioned before, our users were Northeastern University students. Because we have such varying age groups, we asked these users survey questions that could help us identify issues for all age groups, which we go into later.

It is important that we mention that we want this application to be used by different age groups for different reasons; therefore, we tested different aspects that would be used by different people during user testing. A college student may use this application because they are lazy and want to have it all mapped out for them and be able to run to the closest grocery store. A parent might be looking for the best bargain on the items that they might need. We designed prototype users based on all of these different needs.

EARLY STAGE TESTING

Just to recap on what has already occurred before user testing, we wanted to show you some of the results of paper and computer prototyping so you have an idea of where this left us.

The paper prototype was a successful way for us to hone in on the true issues that our application had. This was the first time anyone outside of the team was seeing as piece of it, so it was important for us to show them the full extent of what we were thinking. From this prototype, we developed a new home screen, improved design layout, and made cosmetic changes that affected the flow of the project.

Computer prototyping came with Heuristic Evaluations from our peers. These evaluations both complimented and suggested new ideas for the developing application. We had suggestions to add bread crumbs, which resulted in a Home button on every page, suggestions to change the color schemes so the application was much more readable, and various suggestions that in the end led to what you see.

USER TESTING TASKS

We started every user off with a brief statement on what they would be doing and that their participation was voluntary. Once we briefed each user on what they would be looking at, and they agreed to participate, we set them in front of a computer and asked them to do four certain tasks for us:

1. Add Milk to Your Grocery List.
2. Find the Stop & Shop at Brigham Circle.
3. Find and rate Turkey when you are browsing items.
4. Share items from your grocery list with Facebook.

These four tasks were given to each user in order. As team members, we would let them explore on their own, not giving them any helpful hints. We would see how long it would take to complete each task, and after each task ask what they thought would make that task simpler.

Once they had completed all four tasks, they were asked to take a brief survey. This survey was set up on a computer where we could not hover over their shoulders while they took it. This was to prevent any skewing of the data that could possibly have occurred. Once this survey was completed, the user thanked and sent on their way. Nothing else of them was asked.

FIXING

With our user testing, many fixes and ideas came forward. Some were small things like the fact that when you go to rate turkey, it accidentally said Milk. Others were larger things that prevented some users from sharing the list because of the certain way they had products checked off or even the fact that the back button would account for checking and unchecking.

Most of the cosmetic issues were fixed, but some issues were things that were not capable of implementing fully in the proto.io environment. We wish we could have completed them so that the fit the scenario.

The user testing yielded many results that we were proud of. Most of these will be highlighted in the surveys below, but can also be seen in the countless ideas poured into the application.

SURVEYING

Because of some different usages, the survey was only taken by five users. Some users that

were tested gave us feedback, but opted not to take the survey. Therefore, the survey results that you will see may be completely different than what actually might happen.

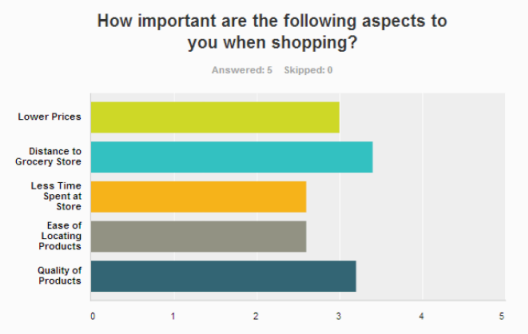


Figure 9

As you can see by Figure 9, we had users rate how important they felt certain aspects of their shopping experience was. Since these were college students, we expected distance to the grocery store to be the number one issue, which is was. But, as you can see, the users we tested felt that almost every aspect we listed was important when shopping. This gave us a good idea that the application should encompass everything, which drove us to fixing and adding some features – like completing the in-store navigation. The results of the data may not be fully reflective because of the low number of users that were surveyed.

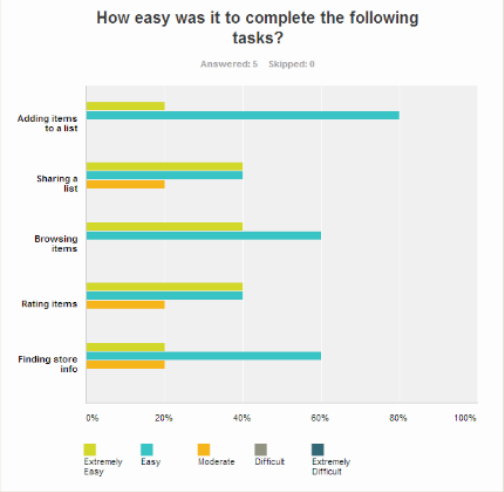


Figure 10

The second piece of the survey found in Figure 10 evaluated users on how difficult they might have found a certain task. An overwhelming number of the tasks were ranked as "easy" by our users. We were glad to see that most users found the tasks easy, and we can understand that very easy might not be achievable, as the application does have a bit of a learning curve for most of its tasks.

It seems that users had the most difficulty sharing a list and rating items. At the time of the user testing, these were not fully implemented. Because of the responses to it, we swiftly got to working on it, and sharing and rating now have a much simpler, complete interface. However, some of the design needs were impossible with the prototyping tool, which will be talked about.

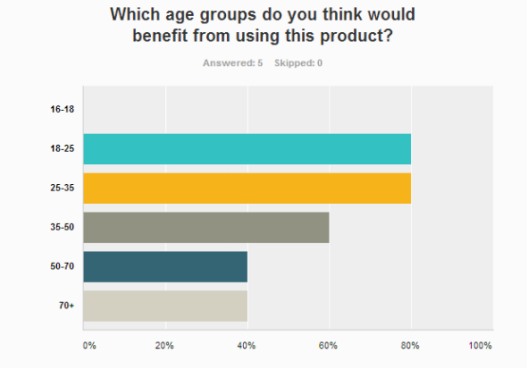


Figure 11

The last piece of aggregated data in Figure 11 is this survey question where we ask users what age groups they believe would benefit the most from this product. None of our users felt that users under the age of 18 had any benefit, probably because they live at home with their parents who do all of the shopping. The highest ranked were both the 18-25 and 25-35 age group. We felt that this would be the highest rated because it includes most people who use technology on a daily basis to get by. We were surprised that some found that 70+ users would use the product, mainly because technology can be lost on the elderly.

REFLECTION

Through our entire project, we ran into various issues. Upon reflection, it became evident that some of these issues could have been foreseen. while others were as a result of limitations of our prototyping tools. These are the following issues we ran into, and what we would do differently in

future projects in order to make the design process easier and more successful.

ISSUES

Our first problem we were tasked with was deciding upon a prototyping tool to utilize to best represent our application. We tested many and they had various limitations, and we ran into a prototyping tool called Proto.io. It had many customizable features, and looked visually pleasing for what could be a potential IOS application. We were pleased to find that we could all log in at the same time and work on it, and after spending countless hours working we all saved. To our horror, every page except for one was erased. After so many hours of work we had to start all over. Why would a prototyping tool that supported multi user login on a single project not allow working on the prototype simultaneously?

The next issue we ran into which we didn't realize until after our first prototype was that we originally didn't have a home page implementation. This was an issue because it did not allow us to go home without hitting the back button multiple times, and our user testers were frustrated.

After identifying this issue, we also found that as users hit the back button multiple time, we ran into verification screens again as well as loading screens which should not have occurred.

The next issues we had were related to font and color scheme. The fonts were deemed to be too small and illegible, and the color scheme we used utilized green and red. This was an issue because people who are colorblind would not be able to differentiate the colors.

Finally, the last issue was that we found our process to be incredibly tedious because we thought by implementing the pages statically it would be easier, however in reality it ended up being very time consuming, and we had an incredibly large amount of pages.

WHAT WE WOULD DO DIFFERENTLY

For our next revolutionary prototype, we would utilize a more common, or even paid prototyping tool. This would ensure a more problem free experience, and we could even be offered customer service in the case that we run into any issues. It could also be a prototyping tool that has more options in creating other interfaces such as android, which wasn't really an option in this prototyping tool.

Next, we realized how important back and home page buttons are. We would be sure to include a back option from the beginning, as well as a home button to prevent frustration of users, and lack of navigability. In addition, back and home buttons are essential to any mobile application.

In regards to font size and color, we would be sure to a font that is legible based upon how far the prototype should be held away from the eyes, and we would avoid colors such as green and red, and also provide a contrast between the background and text colors.

Finally, we would actually code the prototype and make it dynamic rather than static. This would let us avoid many issues we encountered, and it would actually be functional rather than literally just a model.

CONCLUSION

The design process was incredibly informative and the experience was a excellent one. We learned a lot, not only technically related, but also in regards to time management and teamwork. We learned about Usability and how important it is in the design process, as well as imagining ourselves in the place of the users. Through our project we found that what we necessarily thought was important and needed in an application wasn't exactly what the user wanted and needed. This showed us how important prototyping is because a company that dives into a project without actually taking the time to make a prototype may end up with catastrophic sales if there is a certain feature that either is inconvenient to use, doesn't do what it is intended to, or isn't included at all.

ACKNOWLEDGEMENTS

All in all, it was a pleasure to be in this class, and we learned a great deal of information regarding HCI. It is extremely valuable because many of us will be applying these concepts to our future lives and our technical fields. Thank you to Elisabeth Sylvan for this valuable experience, and to our classmates who helped us along in this iterative process by providing feedback and input along the way. We hope you have an excellent summer!