

GroupShop



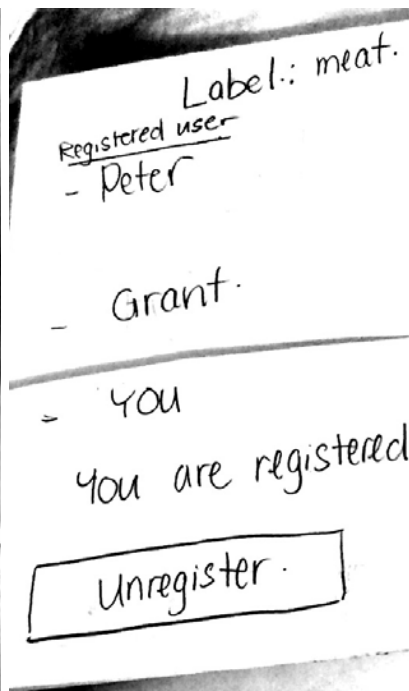
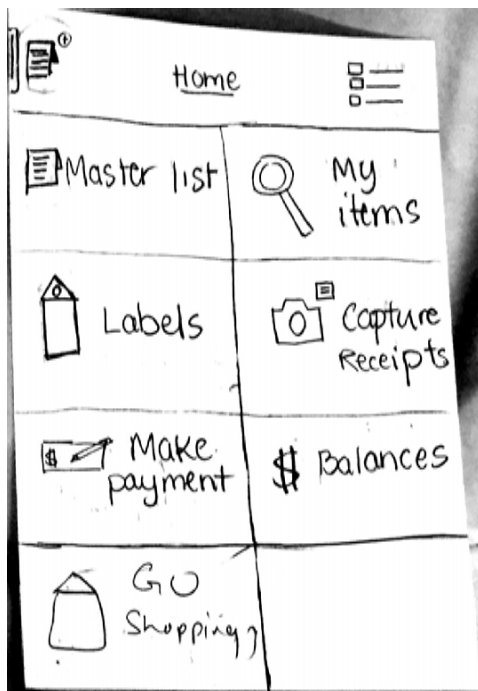
“The future of group shopping is now just a click away”

Peter Hu: Team Leader
Antonio Diaz: User Testing Lead
Grant Azure: Developer Lead
Hieu Trung Le: Design Lead

Problem and Solution Overview

This quarter, our group is tackling the issue of grocery shopping in shared living situations, such as college students living in an apartment. Specifically, we are introducing ways to make the communal grocery shopping process as easy and efficient as possible without any conflict happening between household roommates. A problem situation could look like this: You check your apartment's refrigerator and see that you are out of eggs. Seeing this, you buy eggs later that day, but return and see that one of your roommates did the exact same thing. In addition to this, you and your roommates have agreed to pay for eggs as a group and you now need to determine how much money is owed to you and get money from your roommates for buying the eggs. This situation gets very complicated as you add in more groceries. Our solution is a mobile app where every roommate has access to and can add things to the apartment list. When a roommate goes to the grocery store to buy groceries, he/she will then access the communal list on their phone and see what is left to be purchased. This will help avoid a duplicate purchase. As one trip to grocery store involves 1 person purchasing the groceries at the store, that person needs to be reimbursed by his roommates for their groceries. Once the groceries have been purchased, the app facilitates the reimbursement process by notifying each roommate with the appropriate amount owed and providing options to pay that amount.

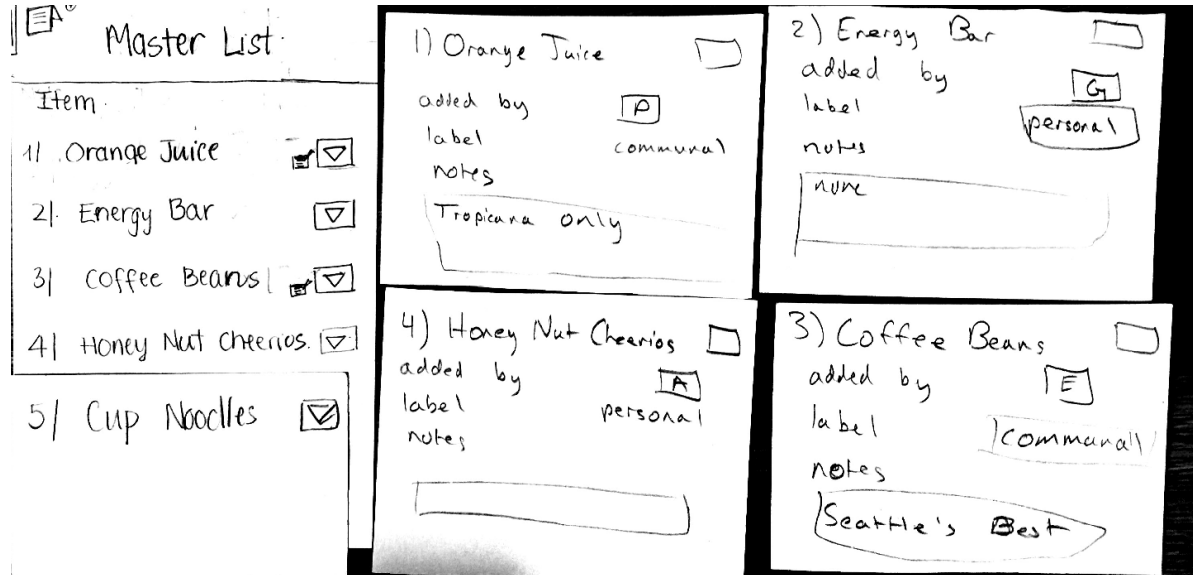
First Paper Prototype



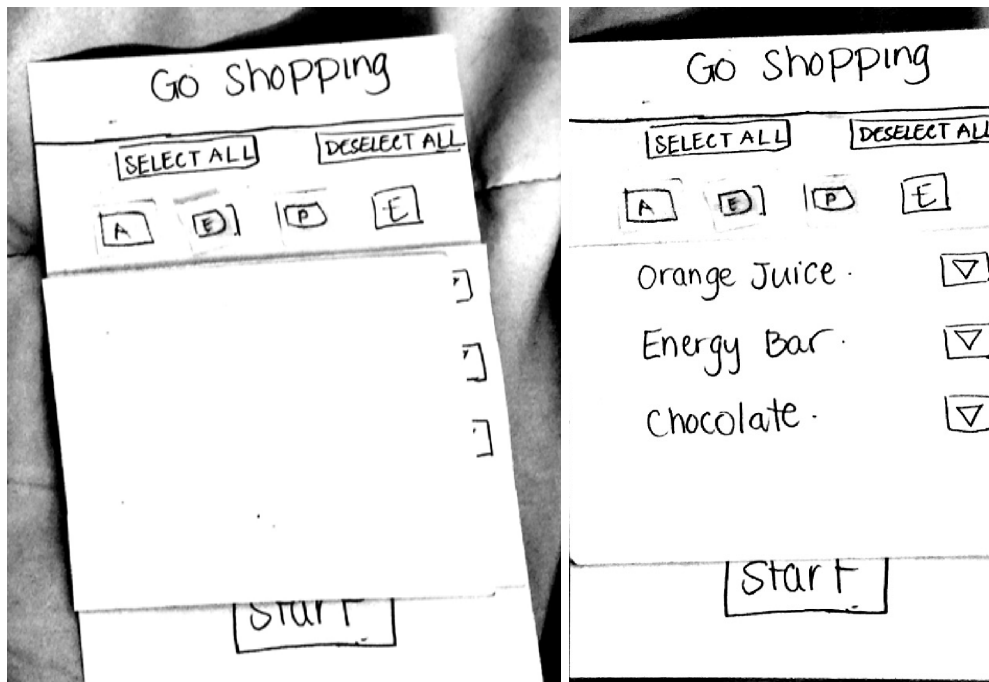
Our first paper prototype is defined by labels on groceries to determine the reimbursement logic. For example, if an item was labelled as "meat", then the roommates who are registered to the "meat" label, which would be anyone who eats the meat that is purchased, is responsible for reimbursing the roommate who purchased the meat at the grocery store. For various reasons, we had to simplify this feature in

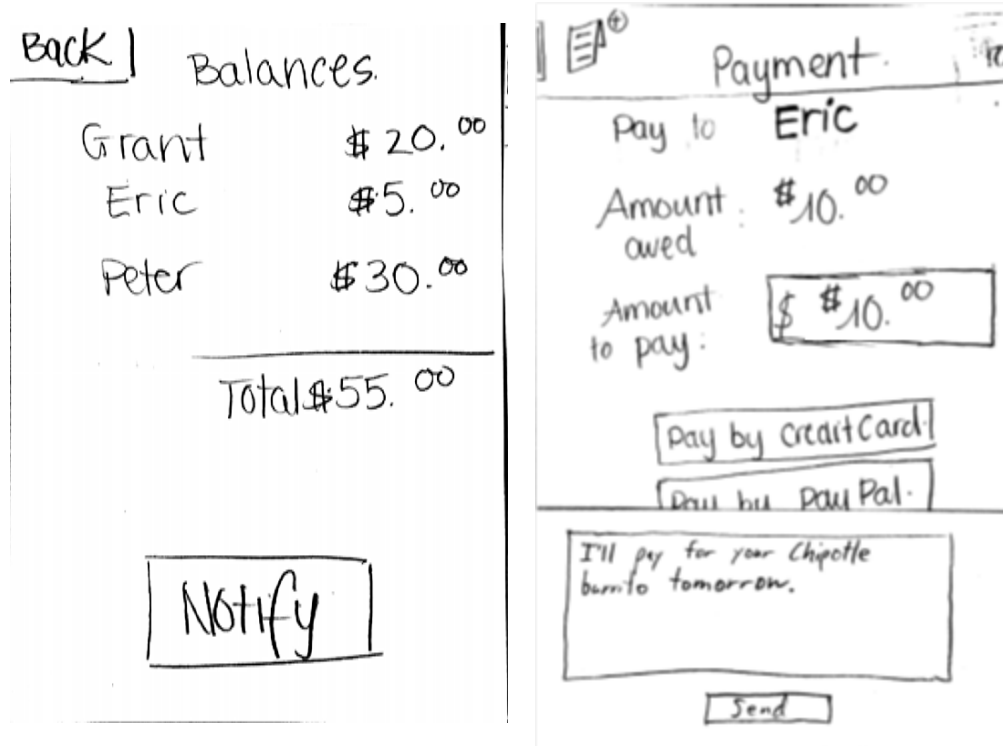
later prototypes.

The part of this prototype that ended up sticking around in more-or-less the same for was the list viewing system. The first task would be to view the list of item that the shopper had input in. Within the master list, each grocery would have a handle next to it that has information such as label, notes, added by, etc. The shopper would be able to select and input in further details about each item if he/she's not happy with that it is currently displaying



The second task is to begin a shopping trip. The shopping coordination features of the app involved switching between an in-shopping and out-of-shopping mode. Whereby going into the in-shopping mode involved declaring and notifying your roommates of the groceries you intend to purchase. Switching back to out-of-shopping mode required a shopper to input the outcome of his shopping trip, either by inputting his/her receipt or by stating that he or she never got around to buying the groceries and needs to cancel the trip in the app.





Finally, the third task is to reimburse the one who went shopping. Reimbursement system was split between two top-level functions, one for paying roommates, and one for viewing balances between roommates.

Testing Process

For testing, we targeted student participants who are living in or have lived in an apartment with other students. Due to time and schedule restrictions, we would meet in locations convenient to the testers like the HUB, Engineering Library, or their apartment. At least two members of our team were present at each test. All of our participants showed interest in exploring coordinated grocery trips and simplified reimbursements. Our participants included an engineering student who lives with 2 other people, a college senior who currently lives alone, but lived with 3 roommates last year, and another engineering student who lives with 3 other students on an off-campus apartment.

The test process went as follows:

1. Introduce the product broadly
2. Tell participant about their "test living situation"
3. Start with the coordination task.
4. Move to the announcement/receipt capture test
5. Introduce the second "test living situation" for the reimbursement task.
6. Continue user testing with the reimbursement test.
7. Debrief, discuss the prototype, ask where they thought it failed.

8. Part ways with the test subject.

As we continued our testing, we found that clear and explicit situations and tasks produced more thorough data for us work with. When our tasks and situations were too simple, we got too many clarifying questions that weren't relevant to the product and how the participant used it. More specific scenarios allowed the participant to focus on the product and saved us time and stress during tests because we weren't answering so many questions about the specifics of the living situation (Who am I living with? What am I going to add to the list?). However, these situations did require more reminders to talk aloud.

Testing Results

From Heuristic Evaluation

Our in-class heuristic evaluations exposed an issue with the in-shopping and out-of-shopping mode. It was confusing as to when one needed to go into in-shopping mode. As the capture receipt function was visible regardless of what mode the app is in, moving into in-shopping mode seemed irrelevant from the user perspective. However, as the designers we knew that the in-shopping mode was important as it facilitated coordination between roommates as they declared their groceries. To fix this, we rebranded the in- and out-of-shopping modes as a simple "announcing" feature. Where the underlying functionality was still the same (i.e. roommates declared their groceries when they were going shopping), but there was no mode switch and shoppers would feel more compelled to use the "announcements" to coordinate with their roommates instead of an odd button labelled "Go Shopping".

From User Testing

Issue 1: Single master list does not offer enough flexibility

One Candidate was disappointed with the flexibility of the master list. They identified this as a problem when they wanted to add items to lists specific to the store where they would purchase such items. Currently, the inability to narrow the list overloads the shopper with information. We classified this issue with a severity of 3 (major usability problem), since the candidate was inhibited from accomplishing the task of building a list. As a result, we introduced an intermediate screen with ability to organize multiple lists (List Manager). Introducing lists organized by people, store, personal/communal solves the flexibility issue and helps shoppers accomplish the task of building a list. We also added 'My List' button, where the shopper can quickly navigate to their own personal shopping list.

Issue 2: Takes too long to get to "Add Item."

Adding an item to the shopping list is a common task. There is too much friction to get to the add item step. The shopper has the option to add to a variety of different lists.

Oftentimes, they only need to add to a their list with high frequency. We classified this issue with a severity of 2 (minor usability problem), since the candidate could accomplish the task of building a shopping list (just with less efficiency). As a result, we introduced a “Quick Add Item” button from the home screen to expedite the process.

Issue 3: Labels are confusing

The term “label” immediately spawned confusion with test participants, one in particular decided to disregard it during the test and we went ahead with the test. The labelling status of groceries was not part of her listing process. During the debrief, we talked with her about maybe changing the language to “billing category” or something similar and that didn’t resonate with her either. We classified this issue with a severity of 2 (minor usability problem), since the candidate could accomplish the task of building a shopping list (just with less efficiency). Our revision to this was to dumb down the billing labels simply to personal or communal. For all billing purposes, this still gives enough information to divide billing costs and is generally easier to understand.

Issue 4: Messaging/Communication

The participant had concerns with the notifications system. He asked, “Will the app notify me if the roommates have received the message and viewed it (of me going shopping)? Similar to WhatsApp?” He also had concerns about the “Question” functionality to dispute charges with people. He felt that the system was open-ended and did not make it seem urgent enough to handle a conflict. We classified this issue with a severity of 3 (major usability problem), since the candidate could not effectively accomplish the task of reimbursement. Our revision to this problem is to add read receipts to notifications. This provides valuable information to the person who needs reimbursement. We also added an Active Dispute Center. To create urgency, active disputes are visible on the home screen and no new disputes can be created between two people until the previous one has been resolved. We will provide a list of list of suggested prompts such as “Can I pay you with Chipotle instead?” and “I think my charge is too much” to keep messages somewhat directed.

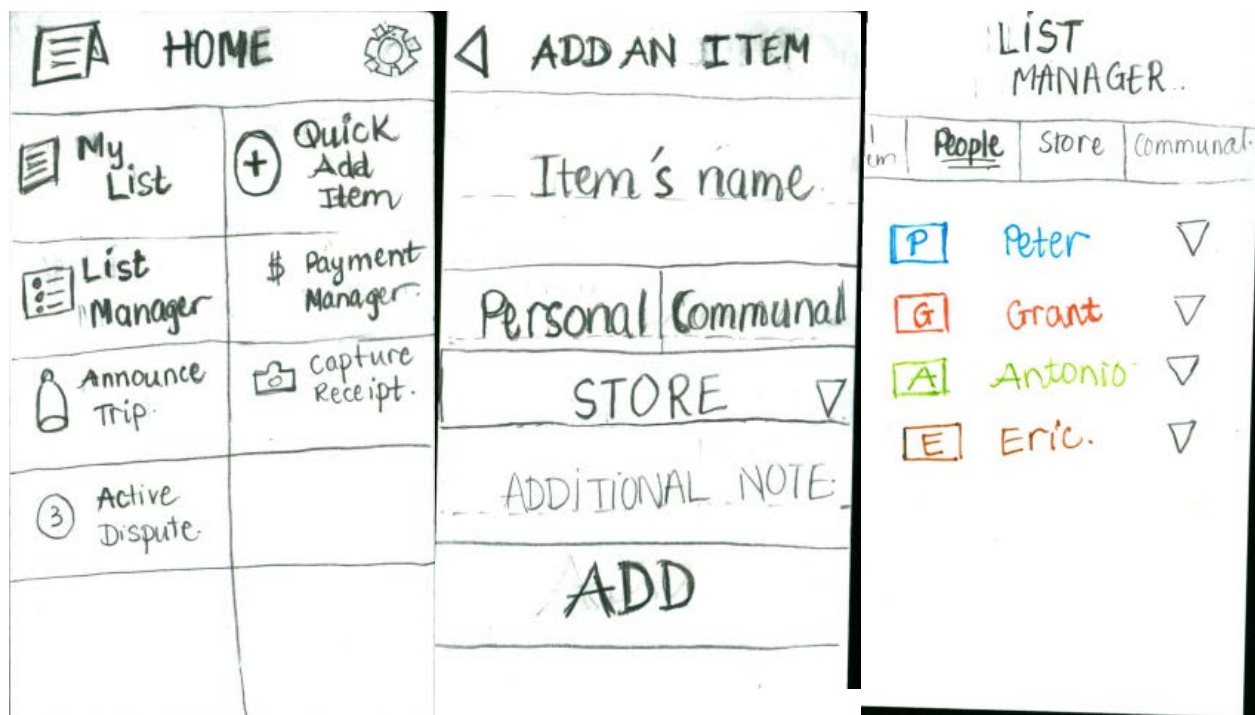
From Design Mockup Critique

The feedback from our design critique mostly involved transferring the affordances given by a paper prototype to the mobile UI space. Things that worked for a paper version of the application would have to be cleaner and more consistent for a digital mockup. Under this part of the critique, we identified that our home screen would have to be re-laid out so it made sense in the context of a mobile app. For example, we had

to clean up button gaps, use proper design practices for cancel buttons (locate below the positive action button), and pick a consistent color scheme. (see Digital Mockup)

The home screen got most of the major overhauls, with a completely new layout. Four of the eight buttons were moved to a new sidebar. The sidebar can be accessed via a hamburger menu in the top left corner. We decided to use a hamburger instead of our logo to remove ambiguity for pressing on the logo did. On the home screen, one row of buttons was kept for the key features that are accessed the most. The clutter on the home page was drastically reduced and gave us an opportunity to preview the “my list” up front.

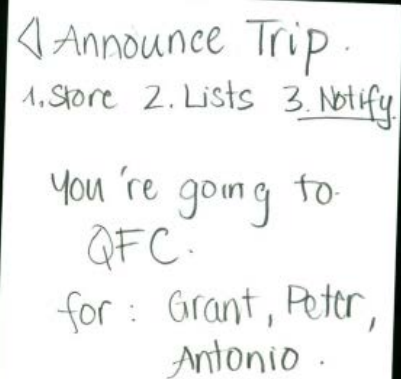
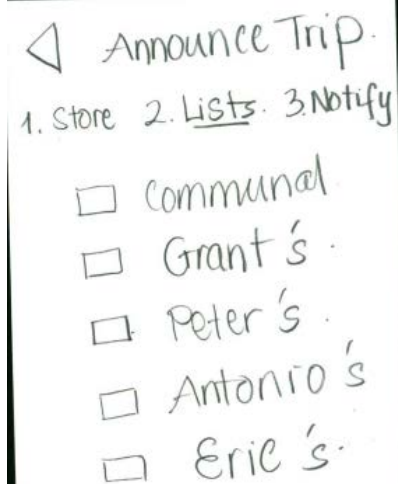
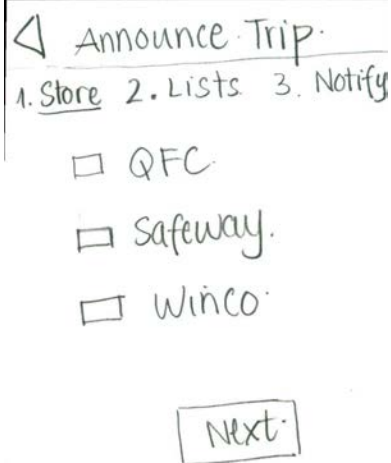
Final Paper Prototype



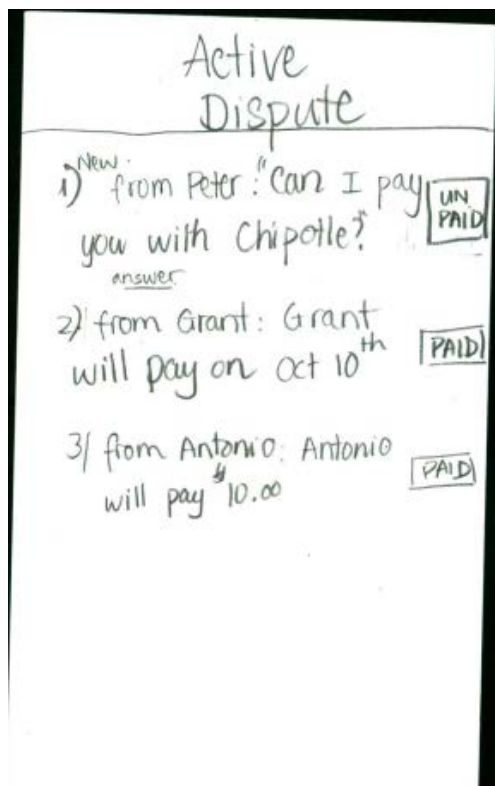
Our final paper prototype addresses the issues we encountered in user testing. We consolidated functionality and added new functionality to the home screen for higher ease-of-use. We added a “store” label that was needed after heuristic evaluations and user testing. The store label exists to consolidate a current grocery list based on the store a shopper goes to that day. This way, a shopper never sees items he or she wants to buy at Safeway when shopping at Costco. The billing label was diluted having each item marked “personal” or “communal”, with personal being the default. This proved to be easier to use and more popular than managing their own custom labels.

The Master List of the previous iteration was overhauled into a List Manager. Instead of using some vague set of filters to distill information everything in the households lists, roommates could select their filters first, giving them more concrete choice about what section of the their

total list they will view.



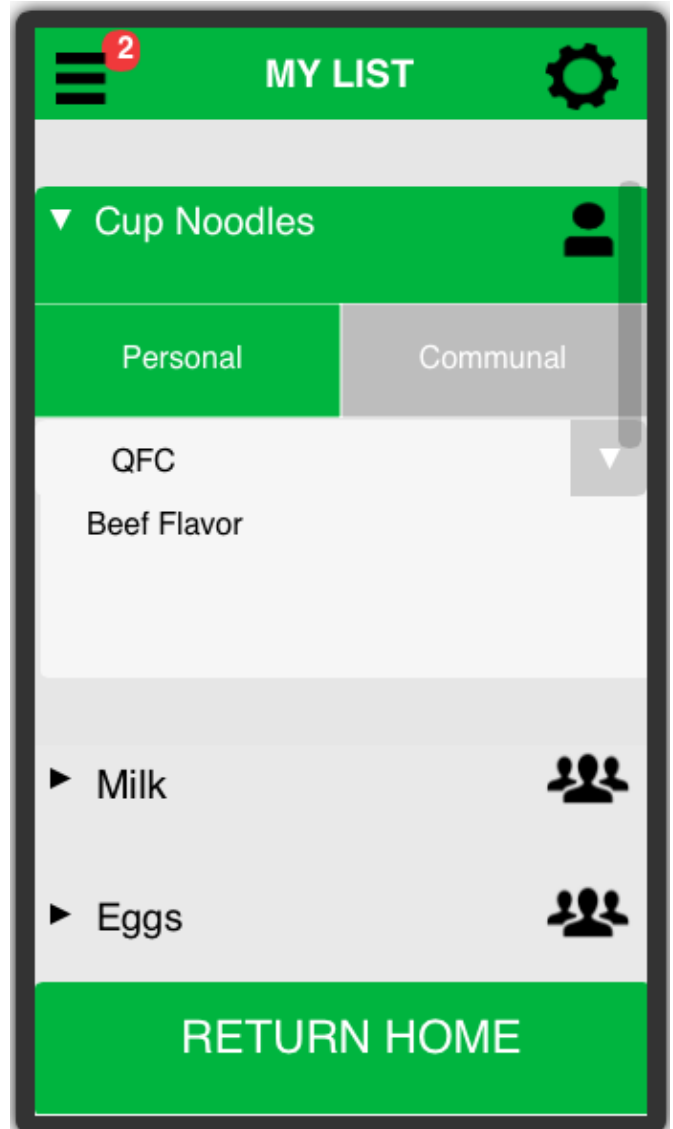
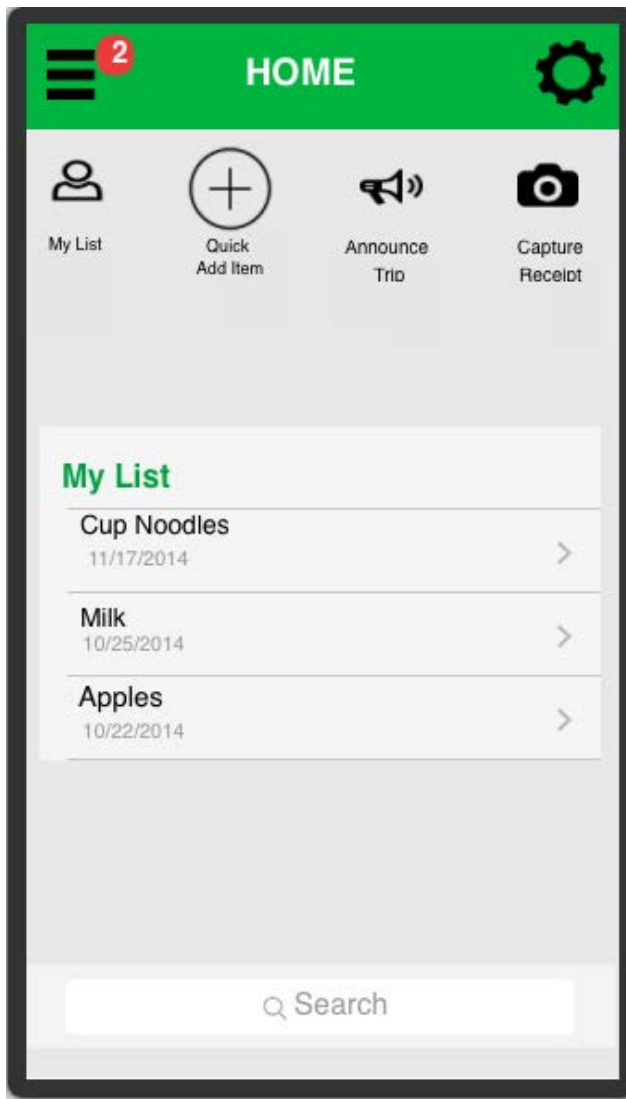
The "Announce" functionality changed very slightly. Shoppers are now able to select multiple stores instead of just one. Instead of filtering on labels and people in the second stage, they can select the communal items list or specific roommates lists.



The final paper prototype introduced functionality that allowed roommates to 'dispute' their grocery bills. Going back to the contextual inquiry, the biggest problem with reimbursement was the uncertainty about when a roommate would reimburse a shopper. With the dispute function of the Balance Manager, a roommate who owes money could declare when he or she can make the payment. Additionally, they can make a partial payment or negotiate with their roommate about an alternative payment.

Digital Mockup

As we first made transition from paper prototype to digital mockup, we decided to make digital mockup exactly as it appears on paper. We thought that our prototype is well done and our user are well versed with what we are presenting. During our Design Mock-up Critique, we received feedback to revise the home interface, remove the cluster of button, fix the font color, change the background, fix dead button, etc. After a second iteration, we came up with a final design of our digital mockup (see below)



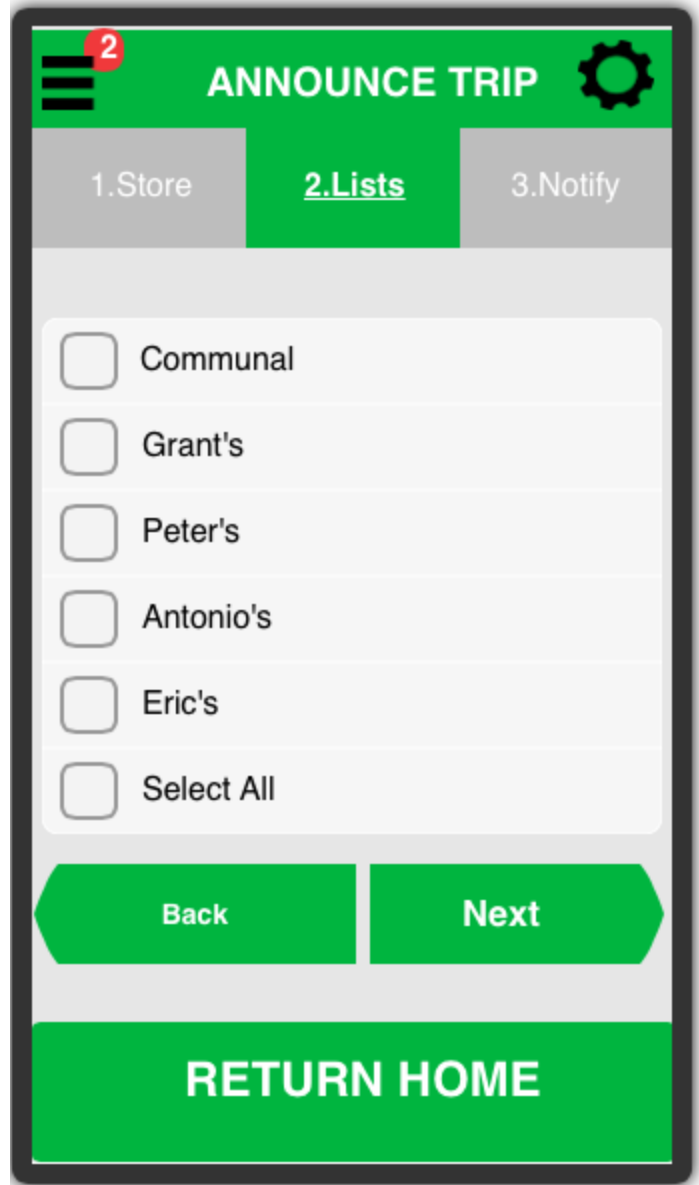
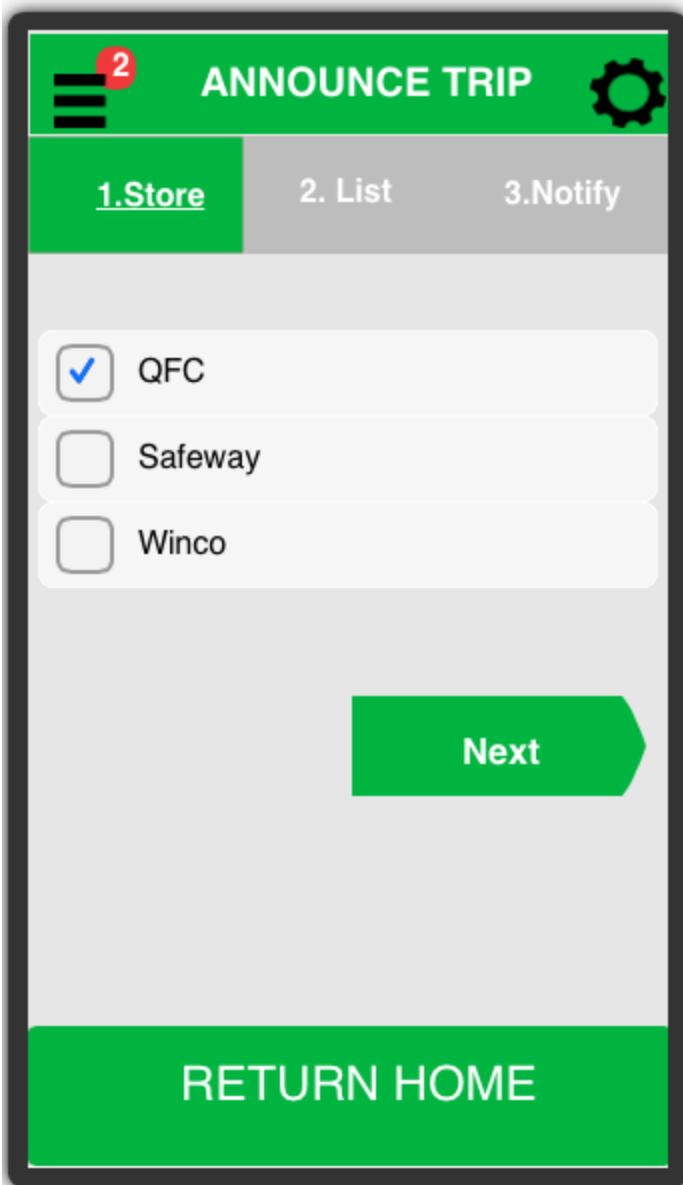
The critical aspect of our design is an intuitive interface that allows user have access to most common task: view “My List”, Quick Add an Item, Announcing a Trip and Capture a Receipt. The users will now be able to view the brief summary of their list right on the home screen. If they wants to view this in more detail, they can select the My List view, which disclose information such as Personal/Communal, Store, and Note.

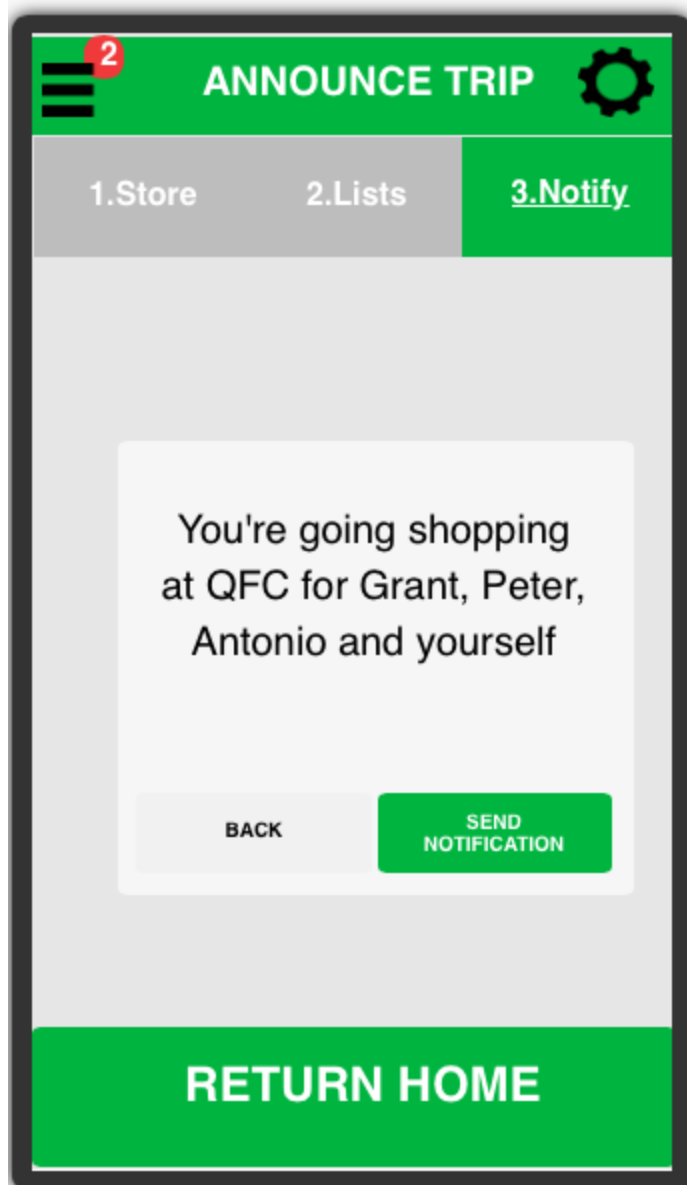
The user now can be able to quickly add an item using “Quick Add Item” on homescreen. They will be asked to fill out item’s name, store where the item is purchased and additional note. They can then decide to go ahead and add item or Cancel and return

The form is titled "ADD AN ITEM" in a green header bar. It features a hamburger menu icon with a red notification bubble containing the number "2" and a gear icon for settings. The main form area is white and contains a text input field for the item name, currently showing "(Item Name)". Below this is a segmented control with two options: "Personal" (highlighted in green) and "Communal" (grey). Underneath is a dropdown menu for the store, currently showing "(Store)". A section labeled "Additional Notes:" contains a text area with the placeholder "Text Area". At the bottom, there are two large buttons: a green "ADD ITEM" button and a red "CANCEL" button.

This form is identical in layout to the one on the left, but with the following data entered: the item name is "Cup Noodles", the segmented control is set to "Personal", the store dropdown is set to "QFC", and the additional notes text area contains "Beef Flavor". The "ADD ITEM" and "CANCEL" buttons remain at the bottom.

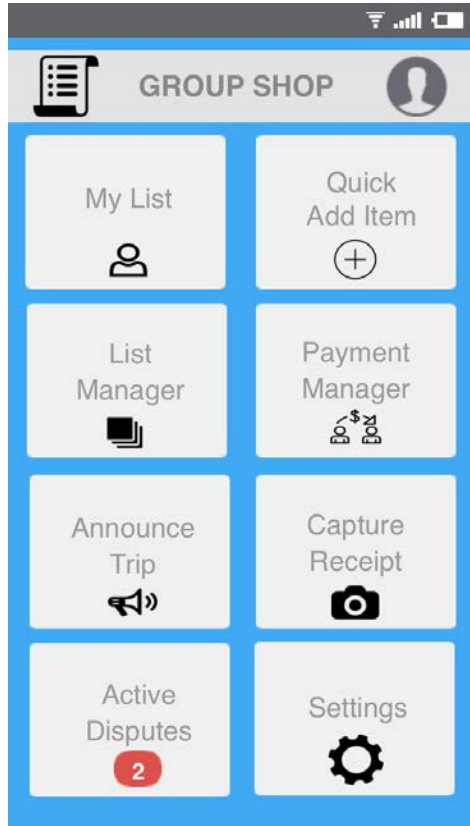
Once the user is ready to go shop, they can click on “Announce trip” right on the home screen. The app will prompt them with a few simple question: Store Name, Whom are they going shopping for, and whether to notify their roommate about the trip.



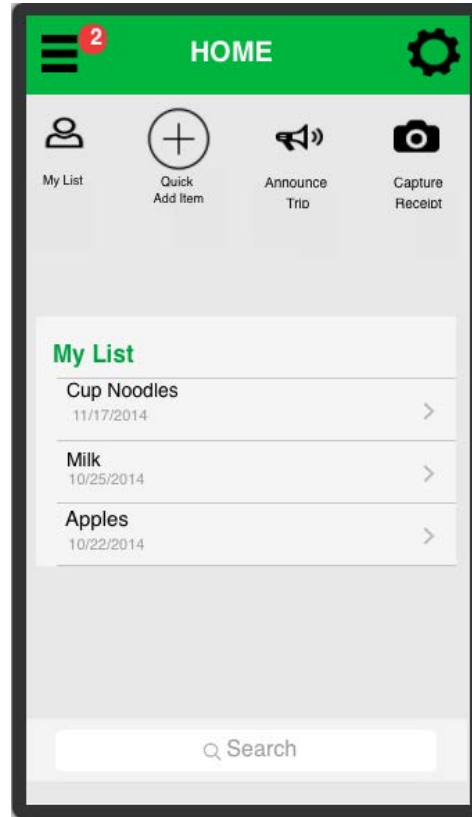


There are very few changes we made in transition between paper prototype and digital mockup. Between the first and second digital mockup, we have received a lot of feedback from critique that our homepage looked cluttered with buttons, misleading grey font, dead button, and weird lines. We decided to revamp our design by making a few necessary changes.

1. Uncluster the home screen to only display the most necessary function of the app. List manager, Payment Manager and Active Dispute is moved into the sidebar, which can be accessible by the hamburger button

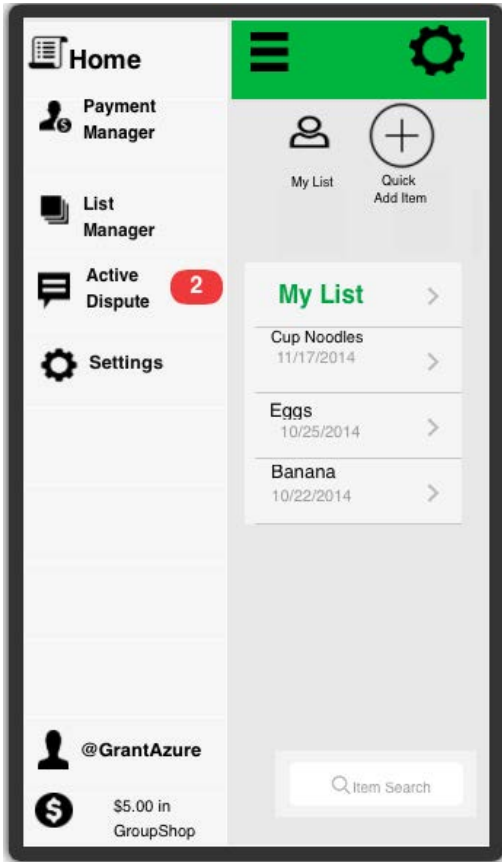


Old digital mockup



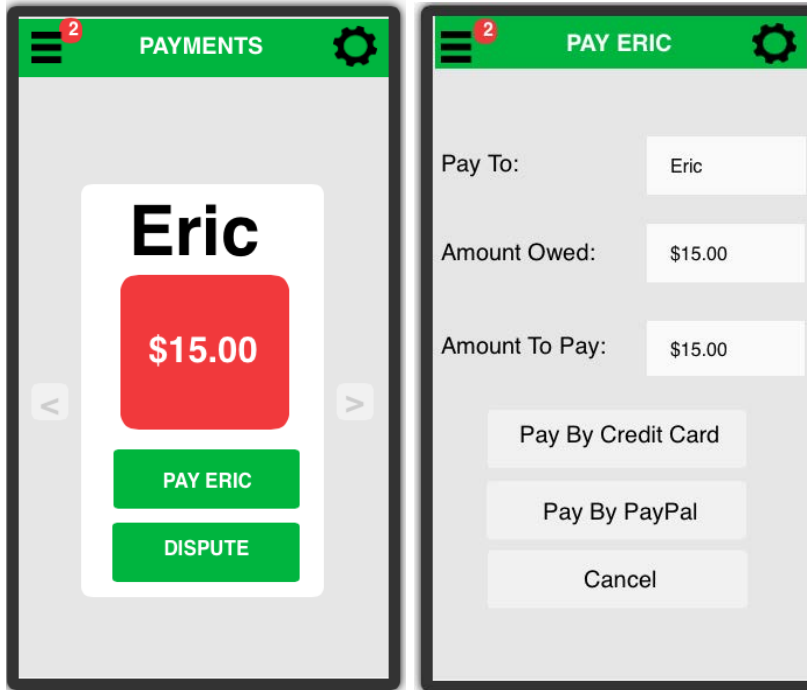
New digital mockup

Sidebar in New Digital mockup



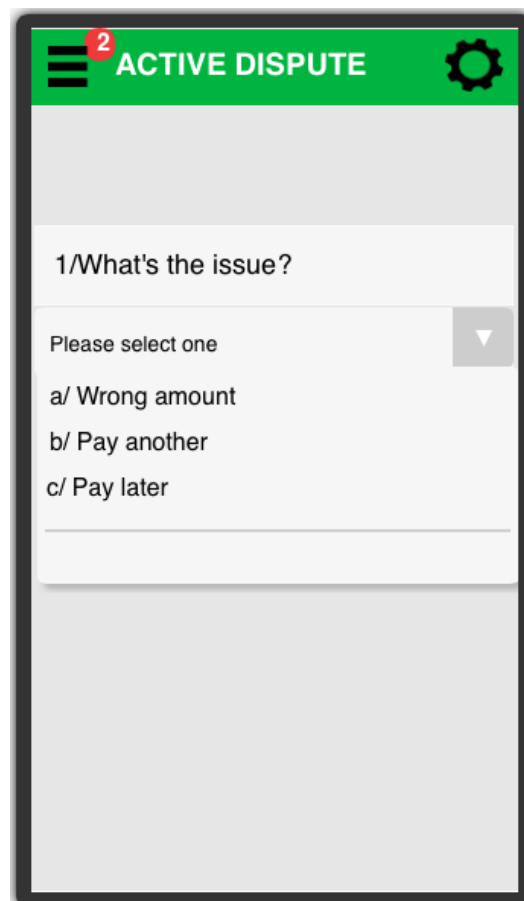
2. We also made some changes on the font color and background color(blue to green). These are minor and does not affect the main functionality that was previously there.

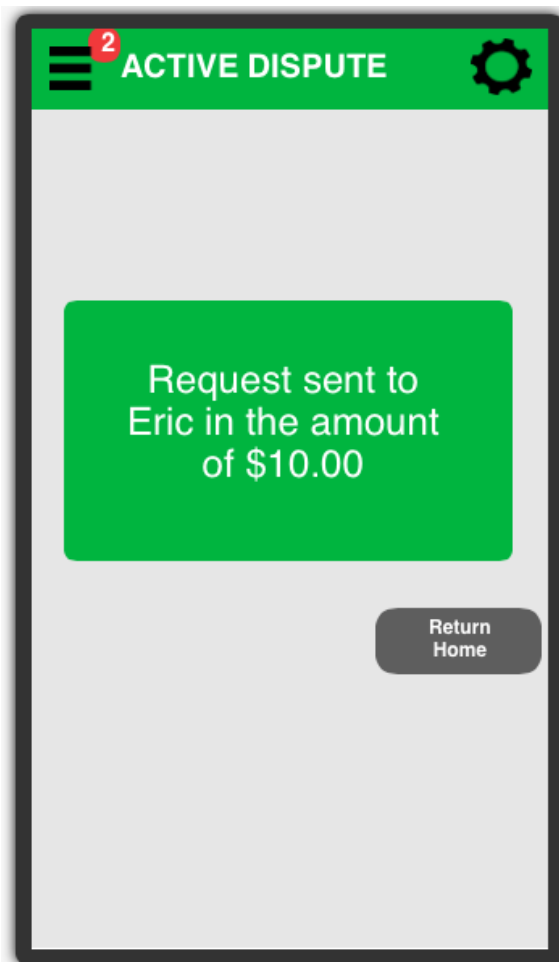
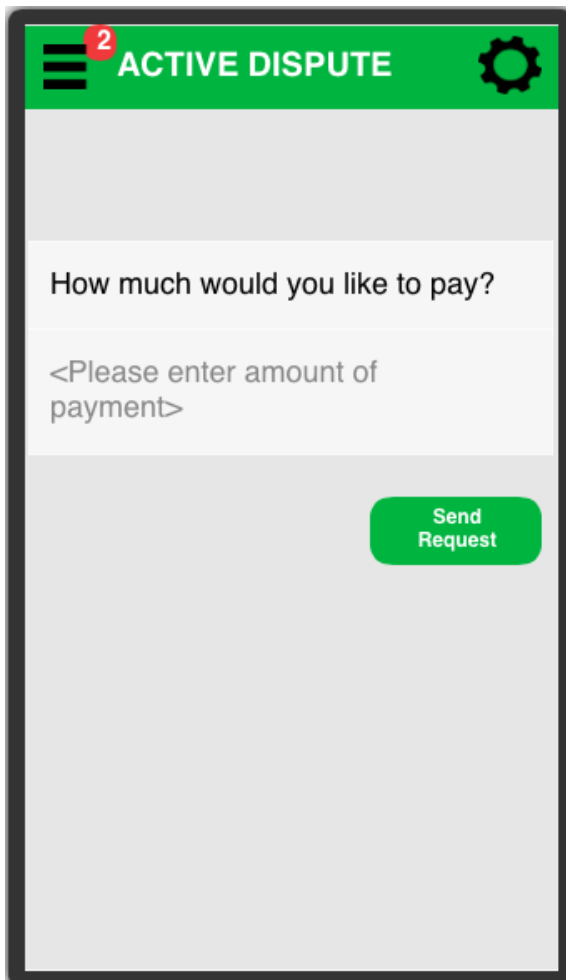
First task: Make Pending Payment. In this scenario, Grant owes Eric \$15.00. Grant went to payment manager and sees the \$15 in red. He selects Pay Eric, specifies how much he wants to pay, and selects the method of payment. Once done, his successful payment to Eric is confirmed.



Second Task: Disputing a charge

Grant is unhappy about a charge that Eric had sent him. He goes to Active Dispute Center in the side bar, selects Create New Dispute, chooses Wrong Amount as the issue, specifies the amount he wants to pay and hits Send Request. The app will then send the dispute to Eric who can then decide whether to accept or reject this request.





Discussion

During the design process for GroupShop, we figured that changes to design should be rapidly included into the new design and then tested to see if the new design was better. Going off in endless discussions about whether Layout A or Layout B was better didn't help at all compared to drawing up the layout and seeing how the user interacted with it. The iterative design process shaped our product by helping us identify which areas needed improvement for the next iteration in the design. Eventually, certain areas that were fine in previous iterations needed to be reworked in later iterations, for example, the complicated label system survived for the first 2 iterations, but was simplified as we conducted more user tests. Without the design process, excess subprime functionality may have leaked into the final product.

Our tasks did evolve somewhat throughout our usability tests by being more detailed. In our initial usability test we did not ask the participant to notify roommates that you were going shopping, but this task became more detailed in subsequent tests. By our third usability test we gave clear directions on what to do in this task, including selecting who you were shopping

for and which store you were going to go to. We felt like giving specific instructions narrowed down where we wanted to find problems, if any, and also gave the user a clear cut way of doing things. In our first usability test, we were general with our tasks because we wanted some general feedback on our design. We then changed our design based on this and with each usability test we became more focused on what we wanted to test.

However, we feel like we could have used more iterations for our design. After completing our 3 usability tests, we had our two main tasks focused down pretty clearly, but the in between stages still needed some work. In particular, we were still figuring out the dispute and reimbursement methods in our design and how to best convey this to the user. We changed our design for this based on feedback received from our third test, but we didn't have another iteration to go through to see if we got it right. Having participants test out this in between process would have given us better insight on how to best design for this situation. Overall, the main design areas were narrowed down well enough with our three iterations, but more iterations definitely would have helped us narrow down more on finer details.

Appendix

Usability Test #1; 11/6/14

Facilitator: Eric

Computer: Grant

Note Taker/Observer: Antonio

Usability Error Tracker: Peter

Participant: Amy

Test Protocol:

1. Introduced team to Amy
2. Explained the test was about our design and not about the participant.
3. Gave the participant two tasks:
 - a. Add Ramen Noodle Soup to my items
 - i. Master list and my items seemed redundant. Master list seemed more intuitive to use than my items to the user. Items could be added from both the master list and my items, so it was confusing to decide which one to use.
 - ii. Master list didn't have option to filter items based on store, which overloaded the user with information.
 - iii. User suggested there should be a faster way to add items to the list, since this seems to be a common task.
 - b. Enter receipts and manage reimbursement after grocery trip
 - i. User was confused what to do in this step. She selects capture receipts to end her trip, but in reality the app was still in shopping mode. This was confusing because these options were presented as two separate buttons on the home screen.
4. Thanked participant for taking our usability test
5. Reviewed usability errors

Usability Test #2; 11/7/14

Facilitator/Computer: Grant

Note Taker/Observer: Antonio

Participant: Ivana

Test Protocol:

1. Found a quiet spot in the HUB to conduct usability test
2. Introduce product, talk about the purpose it's trying to fulfill
3. Introduce "test living situation" (i.e. You are Peter, who lives with Antonio, Eric and Grant); ensure participant that yes this is weird, but go with it.
4. Give first task: You are out of ramen noodles and need to put them on your grocery list. You're also going to announce to your roommates that you're going shopping for all of them today
 - a. Participant figured out adding very fast.
 - i. went to "my list" first to see that status of her list
 - ii. then to "+" sign to add ramen noodles
 - iii. choice of store was intuitive
 - iv. labels were an issue, not clear what they were for
 - b. Announcements went well too
 - i. use of "announce trip" button from the top menu
 - ii. took a few seconds to find, but participant was sure she found the right menu
 - iii. again, confusion on the labels part when completing the announcement
5. Second task: Your roommate has bought you groceries and you need to pay him back.
 - a. very simple
 - b. "payment manager"
 - c. swipe to desired roommate
 - d. pay off debt
6. Second task (part 2): Later on, you buy groceries for a different roommate, but wish to reduce his cash burden to you for whatever reason
 - a. for test purposes, participant decided that her roommate had been mugged and wished to forgive him of his grocery debt as he lost all his money
 - b. swipe to desired roommate
 - c. adjust debt down to \$0
7. debrief
 - a. talk about the overall layout
 - i. mostly positive feedback about menu flow
 - b. show the list manager
 - i. the new list manager never came up in testing, wanted to make sure test knew about it and could provide any feedback.
 - c. In general: design is easy to figure out, except for labels.

Usability Test #3; 11/10/14

Facilitator: Antonio

Computer: Eric

Note Taker/Observer: Grant

Usability Error Tracker: Peter

Participant: Luis

Test Protocol:

1. Introduced team to Luis
2. Introduced app and what it tries to do
3. Stated intro script explaining the test was about our design and not about the participant.
4. Gave the participant three tasks:
 - a. "You realize you are out of Ramen Noodles and need to add this to your list."
 - b. "You are about to go shopping for yourself, Grant, and the community and you want to notify everyone of this."
 - c. "Your roommate Antonio has bought you groceries and need to pay him back, but you would rather pay him with Chipotle."
5. Thanked participant for taking our usability
6. Reviewed usability errors

Notes:

Task 1: "You realize you are out of Ramen Noodles and need to add this to your list."

- Selected "Quick Add Item"
 - Entered Item Name
 - When choosing store, there was confusion of the necessity of choosing a store or not having a preference.
 - When selecting personal, will others be able to see what's on my list?
"Personal" list could elude to it being private compared to my list or my items.
Maybe having a notification saying your list can be viewed by others or having an option for private and public?
 - Overall, pretty intuitive and easy to do

Task 2: "You are about to go shopping for yourself, Grant, and the community and you want to notify everyone of this."

- Announce trip was selected
 - Chose store, the appropriate lists, and notified roommates.
 - Will the app notify me if the roommates have received the message and viewed it? Similar to WhatsApp?
 - I didn't have this functionality readily available.

Task 3: “Your roommate Antonio has bought you groceries and need to pay him back, but you would rather pay him with Chipotle.”

- Went to Payment manager and “Swiped” to Antonio’s budget details.
 - “Requested Adjustment” and sent message.
 - Same question of has the message been received? Can I view this message in the future? Will the app keep track of conversations within the app? We might need a messaging section to the app or have it designed right into the request adjustment section.
 - How will this request be accepted? And how will I be notified?