

PiggyBag

Pool Your Grocery Runs

Final Report



Fabian B.



James B.



Srinivas M.



Jin Woo Y.



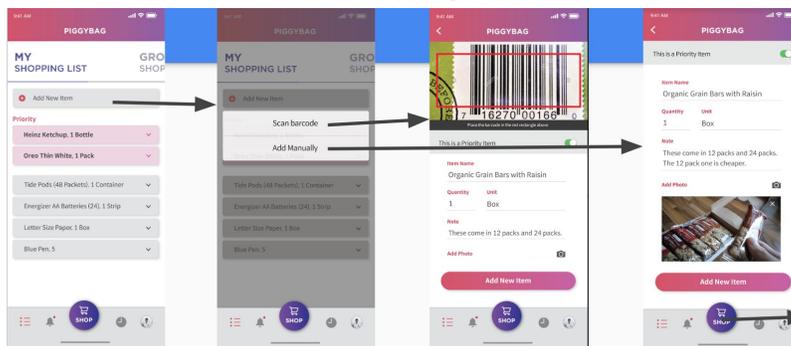
- **Problem and Solution Overview**



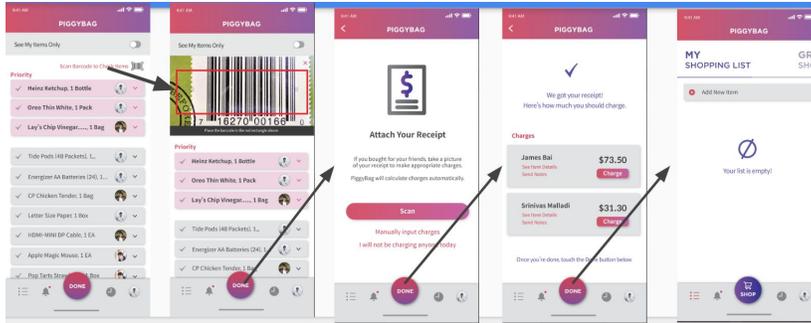
- From our user experience research, we heard many stories on the inefficiencies of planning and making shopping runs among housemates. We observed that it can be difficult for roommates to delegate responsibilities, more specifically, figure out who buys what, and when groceries will be bought. Therefore, we aim to create a platform that provides users with an efficient way of tracking what they and their housemates need. Furthermore, our solution will allow users to reduce the number of grocery runs made by pooling shopping lists among housemates.

- **Tasks and Final Interface Scenarios**

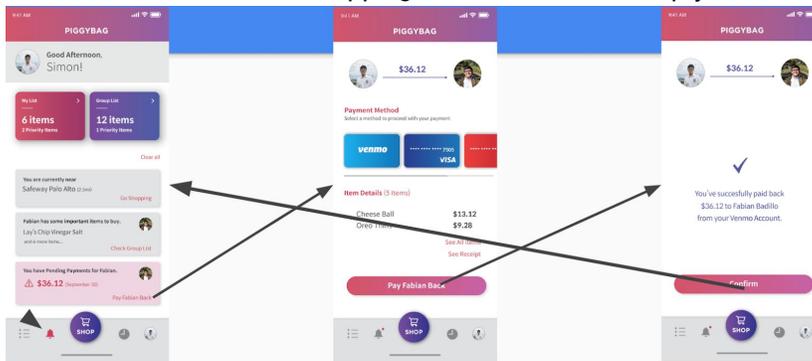
- Simple Task: Manage and share grocery lists among housemates.
 - Rationale: We believe this capability will set a great foundation for efficient communication among housemates in terms of groceries. The user will be able to view and edit his or her list while also being able to check on others' lists while shopping.



- Description: When the user is looking at her shopping list, she can click the “add new item” button. It will bring up a popup option that has two choices. The first choice will bring you to the 3rd interface while the second to the 4th. The barcode will automatically fill in the slots for you while the other option lets you type it on your own. Both choices have the option to be labeled as a priority.
 - Moderate Task: Shop for each other, and get rewarded in return.
 - Rationale: We believe the implementation of the reward system will serve as an impetus to push users to go on shopping runs for their housemates. The reward system will include tipping features and point systems to start.



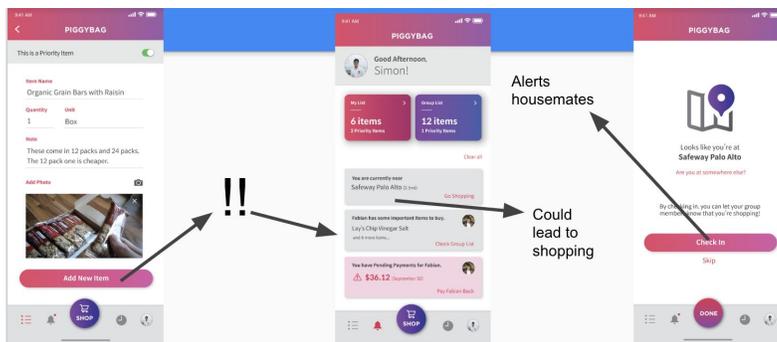
- Description: The user touches the “shop” button in the middle. It brings user to shop mode in the first interface where she can check all the list. She can scan barcode of items she is buying to check off the items as shown in the 2nd picture. After she gets all the items, she clicks done and is presented with an option to scan the receipt in the 3rd picture. After scanning, she clicks done and she is brought to the 4th picture where she can charge the stakeholders of the items. She clicks done and is returned to her shopping list where it reads empty.



- Description: After the shopper charges, the charged person will get a notification in his notification icon in the first picture. He clicks “Pay Fabian Back” and gets to the second picture, where he can use a variety of payment options to pay Fabian back. He clicks the button and then clicks Confirm on the third interface in the third picture to return to the first interface without the pending payment notification.

○ Complex Task: Discover On-The-Go Shopping Opportunities

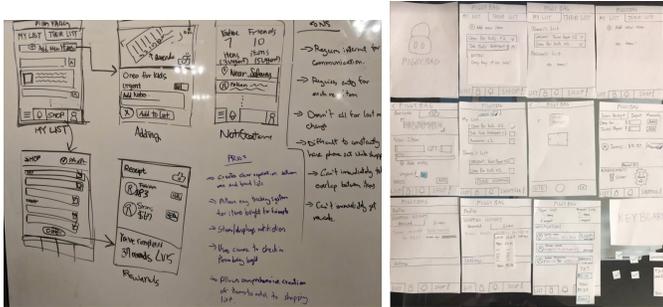
- Rationale: Users will be notified of available shopping opportunities. This includes, but is not limited to when other users urgently need items, when a user is nearby a shopping center, or the item that the user has in his/her list is on sale.



- Description: Adding a priority item sends a notification to the potential shopper. Upon looking at the priority item notification and the fact that she is near Safeway, the potential shopper could be inclined to get the item for them. And when the shopper is at the store, she can notify the housemates by checking in, so that the other housemates can potentially add more priority items. The user also gets notified of his/her proximity to shopping centers, and retail events including sales.

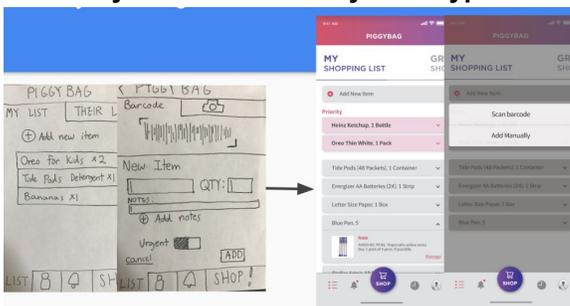
● Design Evolution

○ From Concept Sketch to Low Fidelity Prototype

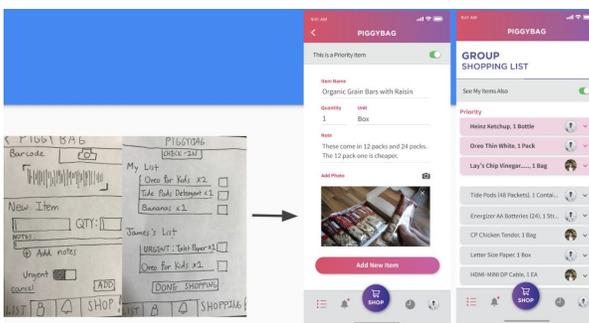


- We added a drop-down notes section for each of the items in the grocery list to keep the interface simple and less cluttered.
- We also added a “Shopping History” section in the low-fi prototype to keep track of past transactions and grocery runs done for and by housemates. Keeping track of the history seemed to solve a core functionality problem for transactions.

○ From Low Fidelity to Medium Fidelity Prototype

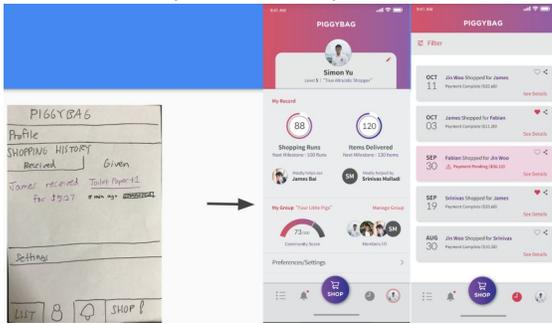


- In the low-fi prototype, when the user clicks “add new item” they are automatically taken to the next interface with the option to scan the barcode or enter manually.
- In the med-fi prototype, when the user clicks “ add new item” the option to scan or enter manually pops-up.
- We made these changes because from the user testing, the users had trouble understanding what to do with the second interface. They were presented with many options to click and immediately paused for a long time. We believe the pop-up option will direct the user to perform the right task and make the second interface look less cluttered.



- In the low-fi prototype, we labeled the important shopping items as “urgent” to nudge the housemates nearby a store to get the items for the one in need.
- In the med-fi prototype, we labeled the important items as a “priority” and intentionally put them on the top of MY list and GROUP list.
- We made this subtle change because in our user testing, the term urgent didn’t have the appropriate connotation. If someone made an urgent request--might as well get it themselves, right? We believe putting items in the priority list communicates more accurately to the shoppers. As before though, when items are labeled as a priority item, the nearby shoppers will receive a notification to

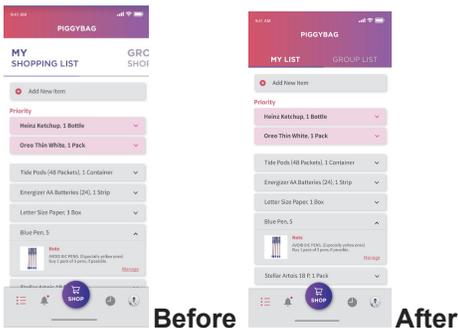
accomplish our complex task.



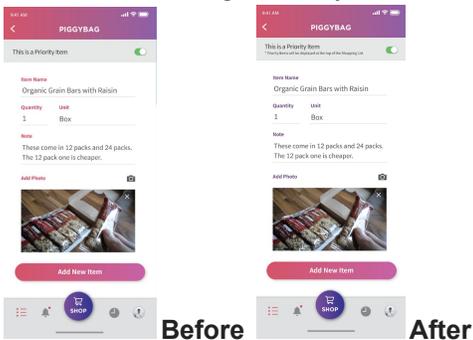
- In the low-fi prototype, we grouped the shopping history and the profile page in the same icon.
- In the med-fi prototype, we separated the two and gave rise to two more icons: the clock and the profile pic. The profile icon holds the shopping records and the point-system that we implemented, while the clock icon shows all the past transactions and shopping you did or others did for you.
- This was an obvious choice because having shopping history and profile together didn't make sense in a design and ease of use point of view. We decided viewing history deserved its own icon because it would be important to keep track of all your records while also being able to request and complete pending transactions.

● Major Usability Problems Addressed

- *H4 - Consistency and Standards:* Alternating between "My Shopping List" and "Group Shopping List" is confusing and does not make sense.
 - Fix: Instead of a horizontally laid out list that users can swipe between, we divided the screen into two tabs., This would prevent confusion on how to alternate between lists.
 - Rationale for Change: There was a slight error in switching from group list to my list due to little technical oversight. Using "swipe" to switch between two things would be unnecessary as there are multiple tabs on the bottom that would confuse the user.

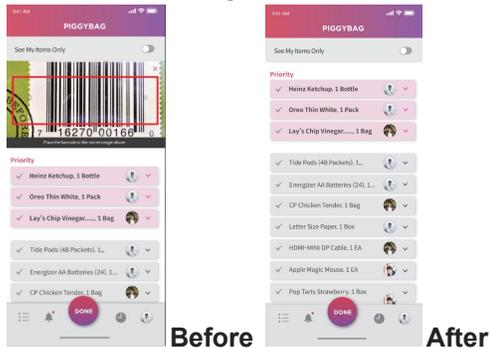


- *H5 - Error Prevention:* "Priority" is selected as default.
 - Fix: We made sure that priority items are not set by default, and also include a further description on the screen about what making an item a 'priority item' would mean.
 - Rationale for Change: Priority items are only meant for small select items that are urgently in need.

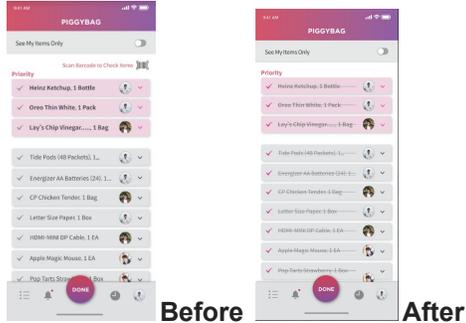


- *H7 - Flexibility and Efficiency of Use:* In shopping mode, the instructions say to "Scan Barcode to Check Items" but it is hard to do that.3.

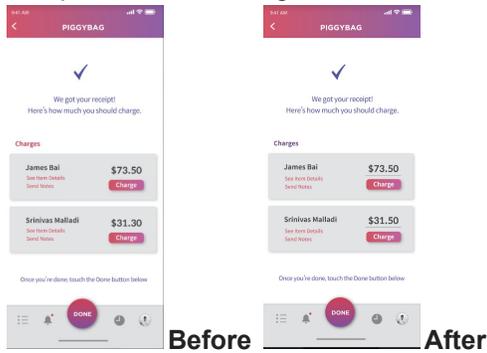
- Fix: We got rid of the barcode feature.
- Rationale for Change: After internal discussions, crossing off with a simple tap is much faster and easier.



- H1 - Visibility of System Status: In shopping mode, it is hard to know whether the items have been checked or not.
 - Fix: When items are crossed off the list, instead of just putting a check in the checkbox, we crossed the item out and decrease the opacity, so that it is more obvious that the item has been picked up.
 - Rationale for Change: It is hard to distinguish between highlighted and unhighlighted check marks.



- H9 - Help users recognize, diagnose, recover from errors: After the receipt scan at the end of a shopping run, there is no option to edit the amount to be charged to each housemates.
 - Fix: We added that functionality.
 - Rationale for Change: The user should be able to change the amount to be charged to housemates, as the receipt scan could charge erroneous amounts.



● Prototype Implementation

- Tools
 - The final prototype was created using React Native to code the application and Expo to run and test the application. We coded primarily in iOS with support for Android as well. The tools helped in iterating rapidly on the screens we were making. We were able to code something up and see it directly on our phones. Except this wasn't always the case.
 - Expo is a bit buggy, so we spent a good chunk of our time debugging Expo and not our app itself. Coding on two platforms also brought up interesting bugs within React Native. We would code something up for our iOS simulator, but then see that it didn't work fully on our physical Android phones. We do appreciate how we didn't have to change our code too much to make it function on either platform.
- Wizard of Oz

- Within our app we have a few features that make the shopping process easier. These include push notifications, scanning with the camera of a phone, and allocating proper cost values to items bought. One part of our task flow is being notified of shopping opportunities. In the full implementation, the user would receive a push notification alerting them that they're near a grocery store. In the actual implementation we have this as a notification bubble within the notification tab. The next Wizard of Oz technique used is in the use of scanning via the camera of the phone. Currently, we have it as part of the task flow in the shopping tab. The app does the scanning, and assigning of values for items "magically", which is displayed in the charging page of the shopping tab. We envision that the actual implementation would use the camera to scan the price values of items, then account for tax. The user is also given the chance to change the amount that is charged for each item, but for now this isn't something that is available on the final implementation.
 - **Hard Coded Data**
 - The app contains a pre-existing list of items added by the user. This allows for the user to see how the list feature is used in the app. Going through the task flow of adding items will contribute to the existing list of items. The profiles within the app have also been hard coded to show how a group of users would interact with the app. Currently, the group of users contains our group member names. The Wizard of Oz technique for the charging of items bought for another person has hard coded values for the items bought. The values would be changeable in the full implementation of the product. One smaller hard coded feature is that the app automatically signs you in with an existing account.
 - **Missing Implementations**
 - To make the date of the project expo a few not-so-critical features were left out. The sign in screen has it so that it automatically signs in, so the option to make accounts is not there.
 - On the home page, the labels for items are not fully implemented in showing the image of the user who requested the item. In the notification page, the top part of the screen is not made to be interacted with. We found this challenging to implement, so we would want to make it clickable in a full implementation. We also don't have push notifications available for the app that would be useful in taking advantage of opportunities to go shopping. The option to pay others is shown in our med-fi prototype, but not in our full implementation.
 - In the shopping tab, we chose to leave out the option to scan items. We saw in our evaluations that it may not be so efficient as just tapping on the items being crossed off from the list. Being able to change how much is charged is something we hope to include in the full implementation.
 - In the history tab, we have a filter feature available, but not currently implemented. Being able to find specific items the user bought is something we would want to include in the full implementation.
 - The profile tab is purely cosmetic in the prototype. In a full implementation, we would track the progress of the user and make it so that the profile page was something that the user could interact with.

● Summary

- As our team conducted an extensive needfinding research with the theme, "Transforming Living Spaces", in mind, we were able to arrive at a very specific problem among housemates: Inefficient grocery shopping. We observed that housemates needed a way to do grocery shopping in an efficient manner, so we created an interactive mobile app to solve the problem. We came up with three specific tasks for the app to perform: managing and sharing grocery lists, shopping for a reward, and communicating shopping requests in real-time. After categorizing them into simple, moderate, and complex tasks, respectively, we began to create our concept sketches. The concept sketches gave us insight into the skeleton of our app, which led to the creation of our low-fi prototype. Upon receiving feedback on our low-fi prototype from potential users, we created our medium-fi prototype. The med-fi prototype was further scrutinized using heuristic evaluation to identify potentially fatal violations in our design. Finally, upon making fixes with the recent feedback, we came up with a functional hi-fi prototype.