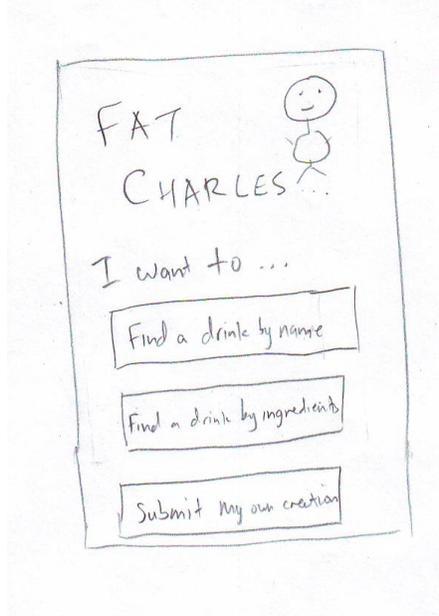


Fat Charles: GR2

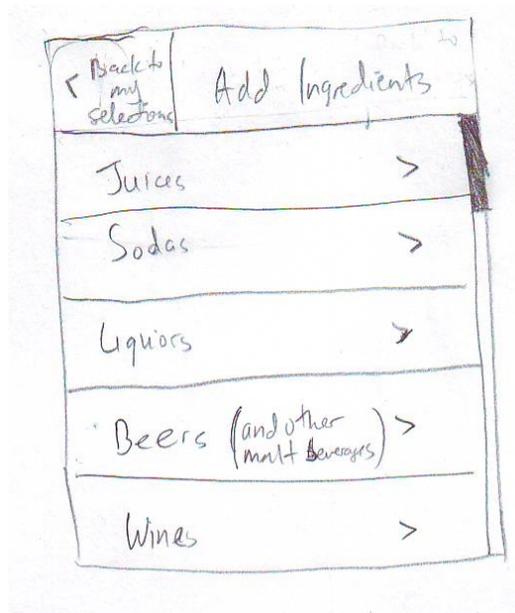
Design 1

I. Design

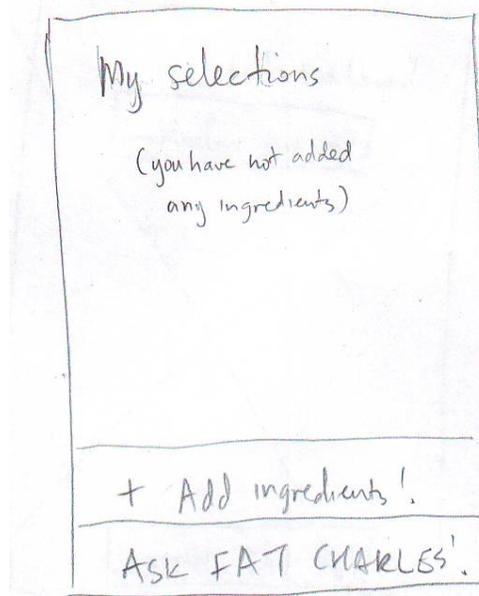
When the user starts the program, he is greeted by Fat Charles and presented with 3 options: to find a drink by name, to find a drink by ingredients, and to submit an original creation.



Referencing our scenario, our imaginary user John would like to find a drink by ingredients. He taps that button, and is presented with a list of options to add ingredients. The options are broad categories of ingredients (juices, sodas, etc.) as shown below:



If John hits the “back to my selections” button without adding anything, the “My Selections” screen will contain no ingredients and an obvious button with a + sign to add ingredients.



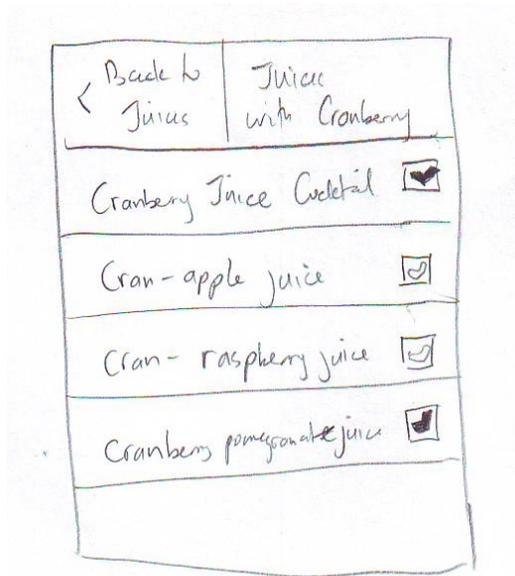
Suppose that John wants to add a juice. He taps the juices button, and is presented with a list of kinds of juices:

< Back to Ingredients	Juices
Orange	>
Apple	>
Cranberry	>
Pineapple	>
Pomegranate	>

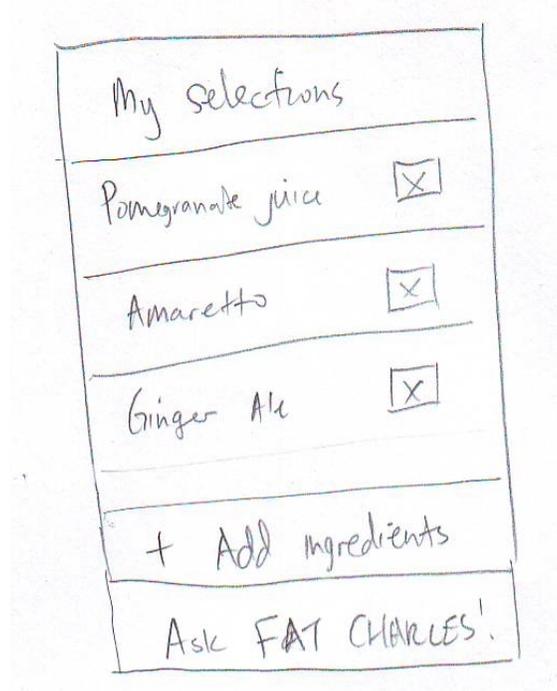
He then taps the “Cranberry” option to see juices with cranberry in them. Notice that some juices fall into more than one category (for example, cran-apple juice) and will show up in both the “cranberry” and “apple” categories. Also, at each stage, there is the option to go back to the previous screen. John only needs to tap the top left corner if he meant to add orange juice instead of cranberry juice.

Once John taps “Cranberry”, he is presented with list of actual juices to select from. Each entry in the list has a check mark that start off greyed out. Tapping anywhere in that row (not necessarily only in the checkbox region) selects the ingredient and darkens the check mark. Tapping the ingredient again will

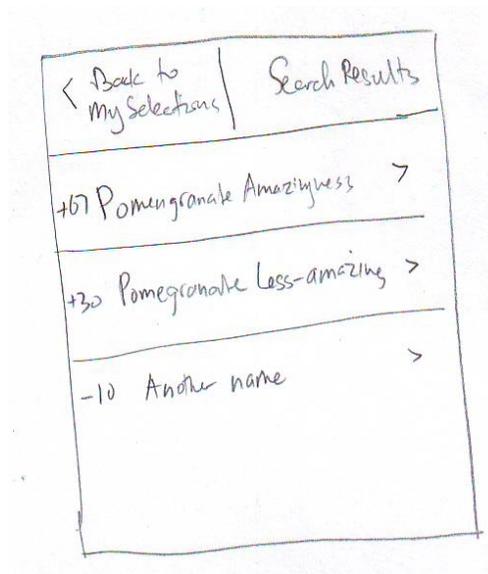
de-select the ingredient



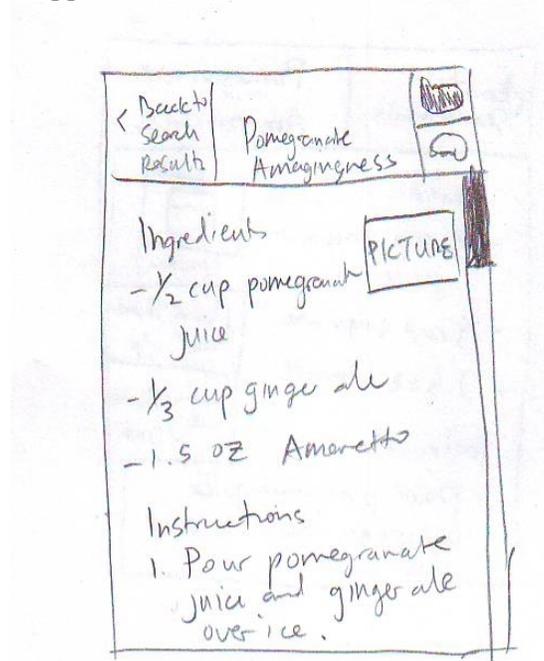
At this point, suppose that John has selected a few ingredients as indicated in the scenario. His “My selections” page will look like the sketch shown below. He has the option to add more ingredients, as well as to remove ingredients by tapping on the X. Note that since it takes relatively few clicks to get to an ingredient and the area of the X box is relatively small relative to the screen, we decided not to have an extra confirmation window before removing an ingredient.



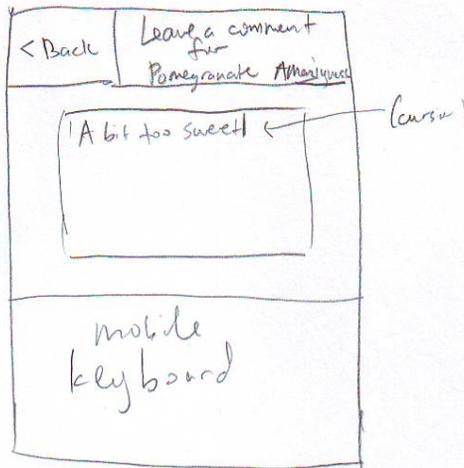
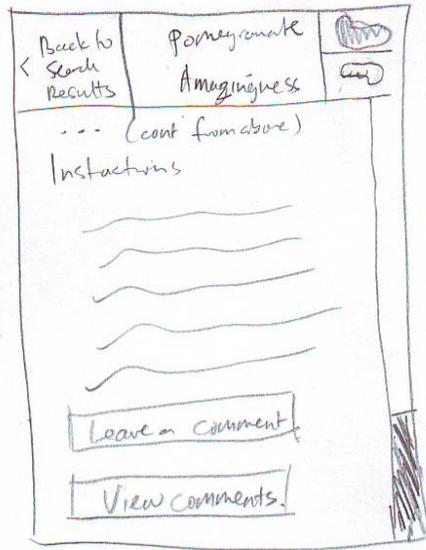
After asking Fat Charles to give recommendations, Fat Charles responds with a list of drinks containing those ingredients (or a subset of those ingredients), ranked by upvote-downvote score (more on this later). Tapping the row of a drink name brings up the drink's recipe.



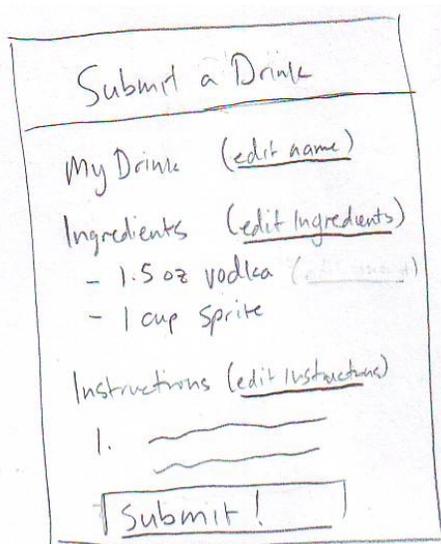
The drink's recipe contains a listing of all ingredients followed by instructions. At the top, in addition to the usual title and back button, there is also a thumbs up or thumbs down option in the top right corner. This allows the user to say whether he or she enjoyed the drink and factors into the overall score of the drink. When one of the buttons is tapped, the corresponding “hand” is shaded in. Tapping the same button again removes the vote and shading. Tapping the button of the opposite option removes the current selection and selects the opposite.



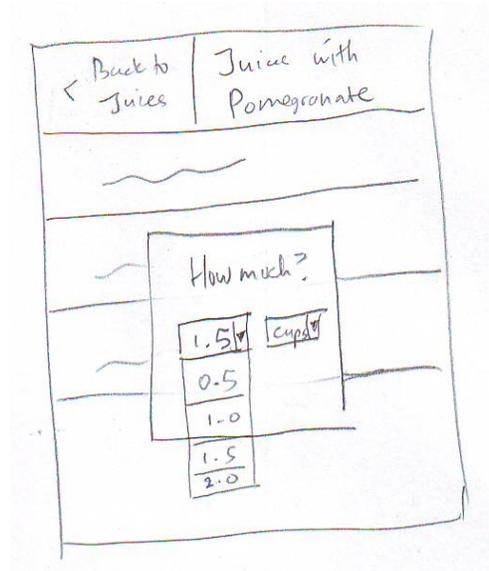
Scrolling down to the bottom produces options for leaving comments or viewing comments. In the scenario, Amy wanted to leave a comment, so tapping “Leave a comment” creates a text field with a mobile keyboard. Closing the keyboard, you can see the entire page, including the button to submit the comment.



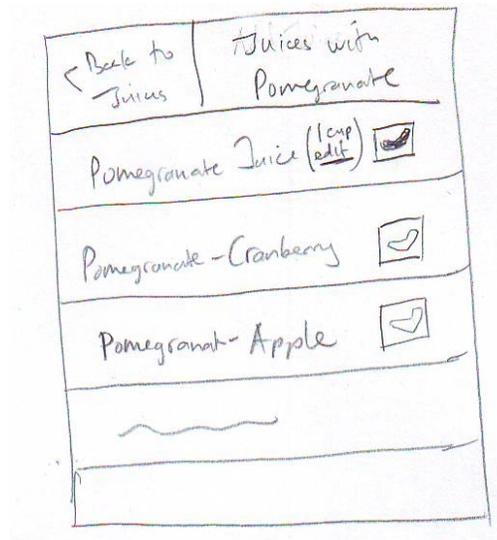
When Amy wants to submit a drink, the screen she is presented with is extremely similar to what a recipe looks like, except each section is editable, as indicated by an underlined “link” to edit. Editing name and instructions make editable text fields containing the current contents of the drink's name or instructions.



A trickier situation arises when selecting ingredients. This design tried to copy the interface for selecting ingredients to find a drink (the task described earlier) as much as possible. The one complication is that we need to take into account how much of each ingredient is needed. When the user first chooses an ingredient, there will be a popup that selects the amount and units of measurement with dropdown menus.



After this has been selected, it will show up next to the ingredient's name. Once again, there will be an underlined "link" to edit this amount via the same popup.



II. Analysis

If this design could be summarized in a single sentence, it would be “*don't force the user to type unless there is absolutely no way around it*”. With this idea, the only times that typing is required is when leaving a comment, describing the instructions, or giving a drink a name. In these situations, there is almost no other way to accomplish the task. Perhaps the most popular task, the simple searching for a drink given the ingredients present at a party and even upvoting and downvoting a drink require no typing at all, and on a mobile phone can be done efficiently using one hand.

One general point that we ran into while making this design: when designing for a mobile platform, sometimes the platform creates interesting constraints. For example, on Android, we can hide a menu and show it when the user presses the menu button, which opens up more parts of the screen for displaying other things. On the other hand, the iPhone has no such menu button, so we may need to look at workarounds. In this design, the iPhone's lack of a menu button has a slight impact on efficiency (more on this in the efficiency analysis).

Learnability

Overall, this user interface is very learnable. Generally, the user interface self-explanatory, cleanly presenting the user with options starting from the very beginning in the home screen. When adding ingredients, the various pages progressively get more specific (for example: juices → cranberry juices → cranberry-apple juice selection) so that the user is not overwhelmed with too many options. Additionally, if one ingredient falls under multiple categories, it is listed in all the categories to aid the user in finding the ingredient.

Another feature that helps learnability is the consistency across pages and the use of symbols or arrows. For example, the buttons that go from general category to specific category all have an arrow pointing to the right. So, the user's conceptual model has click “moving” things to the right. The movement from big-picture general to specific ingredient corresponds to left-to-right movement. Then, if the user wants to go back, the arrow points to the left, which is the opposite direction of how the user got there. The positioning of these arrows and back buttons is also consistent across screens to enhance learnability.

The other aspects of the user interface, such as checks for selection and X's for delete should be familiar to many users. In the user's drink creation page, the resemblance to an actual recipe page ensures that the user knows exactly what the creation will look like when displayed on someone else's phone. The user does not have to get used to two different interfaces for searching and creating drinks.

Efficiency

Every selection can be made at most a few clicks from the main page. In this design, some efficiency is sacrificed for the sake of simplicity and not requiring the user to type. Although this requires more pages (contrast with design #2), each page involves quick taps on the screen and thus the efficiency can still remain high. The use of increasingly specific categories means that each page does not have too many options and thus not too much scrolling is required.

One issue with efficiency is when the user has finished selecting an ingredient and wants to get back to see all selections before searching. In the current design, the user has to click on back two or three times. When coming up with this design, we thought that the pages' designs were consistent, and including a “shortcut button” from the selecting the individual ingredients to the page containing all ingredients would break this consistency. Eliminating this shortcut button would also give more screen space for other things, like displaying an extra ingredient. Eventually, we decided on a compromise. On

Android, the option to go straight back to the page containing all ingredients can be shown when the menu button is pressed. This effectively hides it during normal operation, and if the user decides to take this shortcut he or she has the option to do so. Unfortunately, the menu button does not exist on the iPhone, so we could not do this. As of right now, we have not found a better way to create this “shortcut” functionality on the iPhone without sacrificing a bit of consistency across all the pages.

Safety

Overall, the errors are easy to correct. An erroneous add can be corrected immediately by selecting the entry again to clear the checkbox. Later, it can be removed by tapping the X. Back buttons at the top of every page allow for easy correction of choosing the wrong category. Before a search is done, all the ingredients are shown to the user so he or she knows if any are missing or extraneous. A bit of safety was sacrificed when removal of an ingredient from the list was not protected by a confirmation dialog, but we felt that the user needs to be quite deliberate in tapping the X and it was not worth the decrease in efficiency. However, if in our user studies this becomes an issue then we will implement a confirmation dialog.

When submitting a drink, the user interface looks almost identical to the recipe page, so the user can read the entire recipe with ingredients and instructions and knows if anything looks wrong.