Platelist: med-fi prototype report

Wen Sun, Gene Oetomo, Kyle Qian, Omar R.

October 31, 2014

1 Problem and solution overview

When trying to cook, people encounter many small obstacles: they have to buy groceries, they have to plan their meals in advance, they have to find recipes.

Platelist lets you:

- keep a ``platelist'' of recipes you like, like a music playlist of songs
- plan your meals for the week from that list
- order the ingredients you need for the week

making it easy and quick to cook for yourself. We remove the barriers that stop many people from cooking.

2 Tasks

2.1 Simple: Creating a Platelist

A Platelist (Figure 1) is a list of meal recipes; each meal recipe is like a song on a music playlist. That way, you don't have to bookmark recipe pages on your web browser, or have sheets of paper lying around with your favorite recipes. Once you've accumulated a collection of recipes, you can plan a week's worth of meals out by deciding what you will eat for each meal (Task 3).

2.2 Moderate: Searching for recipes

We offer the user two types of search (Figure 2): a simple search by keyword (eggs, burger, or some other simple query), and an advanced search with a more complicated querying interface. You search for a meal, find a recipe you want, then add it to a Platelist to use when planning your week later.

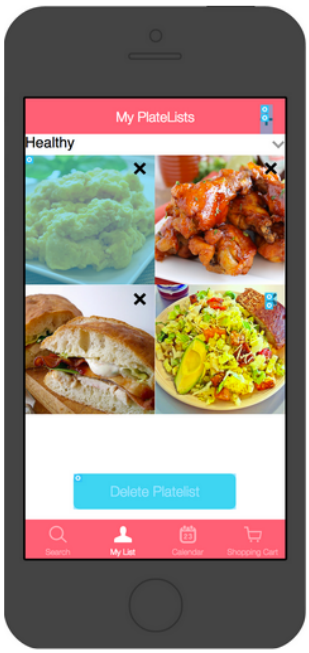

Figure 1: Simple task: creating a Platelist

2.3 Complex: Weekly meal plan

Once you have Platelists containing meals you'd want to eat, you can plan an upcoming week out on our Calendar interface (Figure 3). On Sunday, for example, you might sit down and plan your meals until the next Sunday. Your Platelist might have a salad; then you'd select that and assign it to lunch on Thursday. We also give you the option to choose how many meals you want to eat on each day.

With your week's meals planned out, you can order the groceries you need all at once. A button adds everything to your Shopping Cart.

3 Revised interface design

In general, we made a number of small aesthetic and behavior changes to the UI to optimize for the mobile multitouch prototype. For example, we added the swipe gesture to open the app.

We believe the leap from the medium-fi prototype to the hi-fi prototype will be smaller than the one we've made from the lo-fi prototype to here.

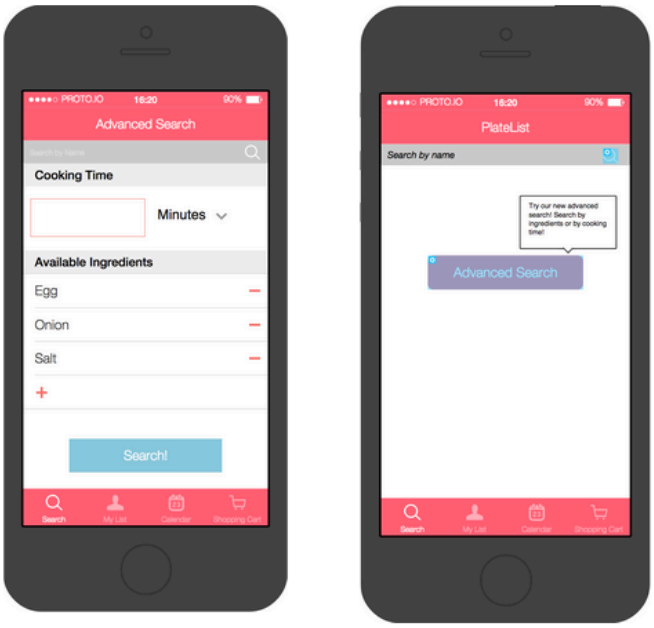

Figure 2: Moderate task: searching for recipes

3.1 Multiple platelists

In the lo-fi prototype, we had a screen with an empty Platelist, but we had no functionality to manage multiple Platelists. You were confined to the one built-in list.

You might want one Platelist for simple breakfasts, and one for fancy meals to cook for parties, and others for other kinds of meals, and the lo-fi prototype did not support those cases. In the med-fi prototype, we added the ability to manage (add and delete) more than one Platelist in the app, making our simple task (creating a Platelist) more meaningful. We added a drop-down menu so you could scroll through the different Platelists.

3.2 Advanced search function

Our lo-fi prototype had a complex query UI where you could search by specific ingredients, cooking time, and other parameters. But people we talked to said that they often just wanted to search for a specific meal (a hamburger, or a salad, for instance), and the search UI had an intimidating number of parameters. So we added a simple search query box where you can type keywords, much like a Google search box. We preserved the advanced search in a separate screen for users with

Figure 3: Complex task: weekly meal plan

more specific needs -- you click the Advanced Search button to reach it.

We also added a new goal in designing advanced search: we wanted to give the user the ability to use premade food options, since not everyone wants to cook all the time. Then the user can manage all their meals from within Platelist, even ones where they aren't cooking a recipe.

3.3 Scheduling

In our lo-fi prototype, we only had one shuffle button that created an entire week of items for the user -- the user had no control. In our new UI, we created slots for each day that the user can check if they want a meal. For example, each day automatically has 3 empty slots for food items that they want for that day. If they want breakfast and lunch, they just tap on two different slots to put those into the week. When they are finished selecting their choices, they just press the shuffle button, and it places the corresponding types of food into their selected slots.

3.4 Tab bar

We added a tab bar to the bottom of the screen. Participants in the lo-fi prototype complained that modes of the app were hard to see, and the overall structure wasn't obvious. The tab bar shows that you can Search for recipes, view a Platelist, view your weekly Calendar, or order groceries in the Shopping Cart. You no longer need to remember the app's structure and where you are in the flow.

3.5 Shopping cart

The Shopping Cart screen was previously only accessible from the Calendar screen. Now it's a top-level screen, visible from the tab bar. This change makes ordering a more prominent and easier-to-manage function.

3.6 Text bubbles for instructions

We added little text bubbles in places like the Calendar to alert first-time users about how the UI works. Blank meal slots inform the user that they can add a meal to the calendar.

4 Prototype overview

4.1 Tools

We used Proto.io to construct the prototype. It was easy to pick up, and the basic functionality was straightforward. The resulting prototype was clean and simple, considering it was a quick mock-up. Storyboarding was also easy: we simply linked specific buttons to specific pages. Overall, it was easy because all the design aspects were visual, and we just had to drag and drop.

4.2 Limitations and tradeoffs

Although it was easy to pick up, there were many advanced functions of Proto.io that were not intuitive. There were some tutorials, but we didn't want to have to watch all of them to understand how to use the software. Some features we wanted to add to the prototype were really hard to figure out and frustrated us (such as editing the navigation bar).

Also, since everything was hard-coded, we couldn't make a simple wrapper function to do the same thing for multiple items. For example, to delete an item from the Platelist, you would press the X on the corner. However, to do this with every item, we would have to implement each one individually, which would have

been extremely mundane, so instead we just implemented it on a few and used those select few to demonstrate.

Working in a group was also difficult; Proto.io didn't have a way to synchronize multiple edits going on simultaneously, unlike Google Docs, for example.

Overall, Proto.io was easy to use at first, but we had trouble with more advanced functionality later. It's easy to make a quick sketch, but not reasonable to implement full functionality because that would require repetitive hard-coding, such as the deleting function mentioned above.

4.3 Wizard of Oz techniques

The only recipe we implemented was Scrambled Eggs, so we were careful when giving the demo to just click that specific item.

We also mentioned earlier that because Proto.io does not have an easy-to-use way to repeat behavior, we only implemented deleting for a few items on the Platelist.

4.4 Hand-coded features

TODO

5 Prototype screenshots