

# ***FLAVR***

Andrew Luo: design/development  
Ben Ulmer: documentation/development  
Alex Engel: user testing/development  
Onkur Sen: manager/development

## **Problem and Solution Overview**

The words "free food" have been the bait for students attending countless events and gatherings across college campuses for good reason. Food is often high on a college student's agenda. However, events advertising free food are often met with an unenthusiastic turnout, spawning a host of problems ranging from food waste to unnecessarily hungry students. Flavr remedies these problems by connecting students with food in real time through a more extensive food distribution network.

## **Representative Tasks**

Flavr helps users accomplish 3 food-based tasks. Our primary task is for students to finding free food on campus. The user must specify which type of food he or she wants, understand the directions to the food, and verify that they're coming. This task is of medium complexity as it requires the student to both find and respond to an event.

However, for the first task to be successful, providers must first complete a simple task by posting an event with information about what, when, and where food will be available. Once the event is posted, providers can get real-time information about the number of people who will go to the event.

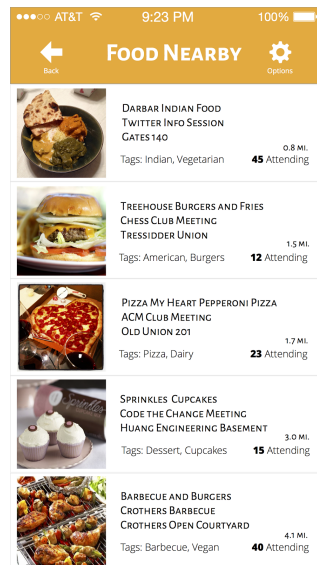
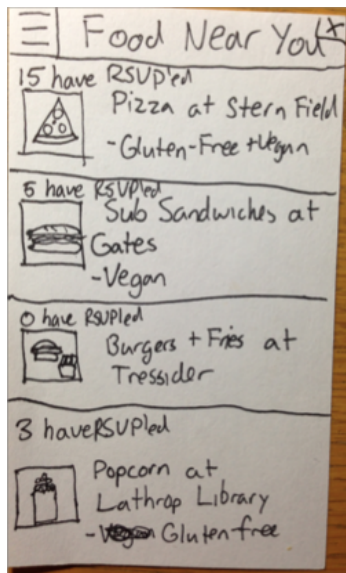
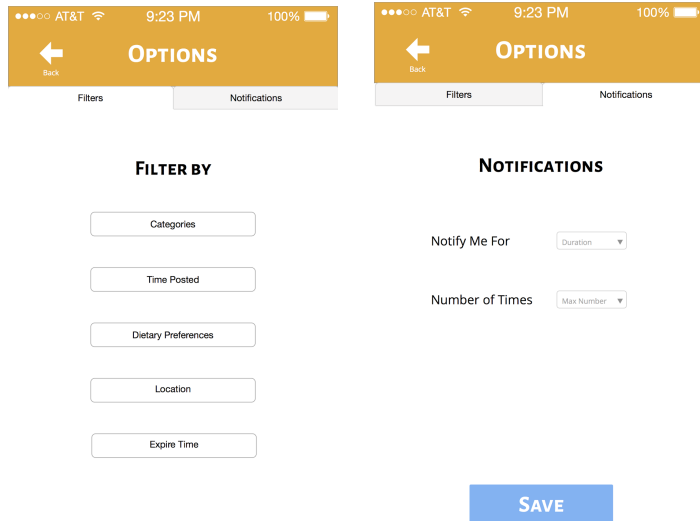
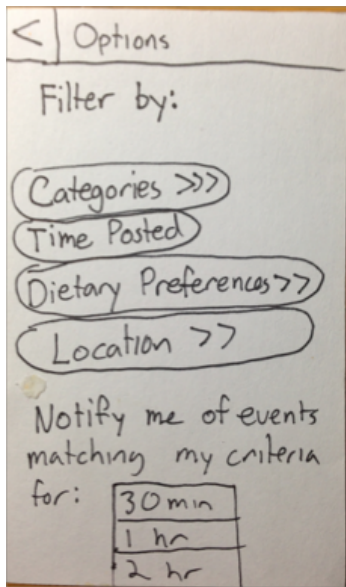
Lastly, consumers can customize their experience, ranging from dietary preferences to restricting the radius within which results should be displayed. Furthermore, we allow users to set up a notification system, which will let them know if food becomes available in the near future. Due to the many options, we rated this a complex task.

## **Revised Interface Design**

One piece of feedback we incorporated was that our "back" button looks more like an arrow instead of an angle bracket, and our "settings/options" button is a gear instead of a drawer. We also labeled these buttons so that their purpose would be clear.

Furthermore, we separated the notifications and filtering sections in our options menu for a more streamlined approach. In addition, we added information about food events

to the “Food Nearby” screen (more explicitly, distance to the event and number of people attending).



Our revised prototype is available at <http://invis.io/G41LFBQQJ>.

Click the comments button on the bottom right to see our README on the home screen.

For our first task of finding and attending a food event, a consumer begins at the home page (Figure 1a), clicks “Get Food” to get to the “Food Nearby” page (Figure 1b), clicks on a food event in the list to get to a page with detailed information (Figure 1c), and

finally clicks “I’m Going” to confirm attendance and get directions to the food event (Figure 1d).

For the second task of posting a food event, a provider begins at the home page (Figure 2a), clicks “Give Food” to get to the “Create a Post” page (Figure 2b), fills out information regarding the event, and finally clicks “Post” to confirm event creation and wait for people to respond and arrive (Figure 2c).

For the final task of setting up notifications for an event, a consumer begins at the home page (Figure 3a), clicks “Get Food” to get to the “Food Nearby” page (Figure 3b), clicks the “Options” gear in the upper right-hand corner to get to the “Options” page (Figure 3c), clicks on the “Notifications” submenu to get to the “Notifications” tab (Figure 3d), sets the duration and number of notifications desired, and finally clicks “Save” to get confirmation that their preferences have been saved (Figure 3d).

## Prototype Overview

To create our medium-fi prototype, we used a combination of Sketch 3 and InVision. In particular, we used Sketch to mock up the details of our prototype. We then used InVision to import the resultant screens and link them together using hotspotting. We found Sketch to be extremely robust in creating screens that had a lot of textual detail and various colors. We also enjoyed InVision’s ease of use in linking static screens together and creating a more interactive experience. InVision was limited, however, by the fact that it didn’t provide an iPhone frame for our images and also did not allow us to reuse parts of our sketches in many places. A new image had to be made for each screen.

Our medium-fi prototype’s limitations mainly lie in its interactivity with the user since the prototype is static (we’re currently operating with hard-coded data points). For instance, a provider cannot actually enter information about their event. Another example is that a consumer’s dietary preferences are not reflected on the list of results of nearby food. Furthermore, because there were many similar screens with just varying options, not every option in filters for example had an accompanying screen made, because the different screens for various filters look almost the same with only different text and no new designs. These problems will be solved in our hi-fi prototype because our backend will be instantly responsive to these choices.

Furthermore, we would like to redesign our home page by replacing the text for “FLAVR” with a logo, changing the background color, and simplifying the buttons.

# Prototype Screenshots

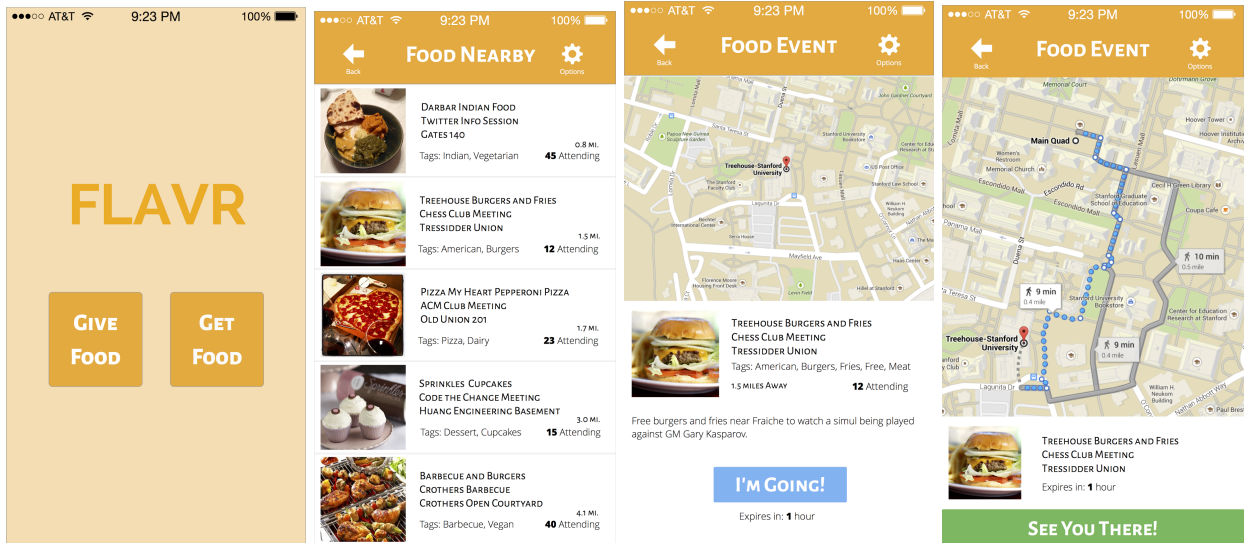


Figure 1a-d: The Get Food flow from the home screen in a) through the confirmation in d)

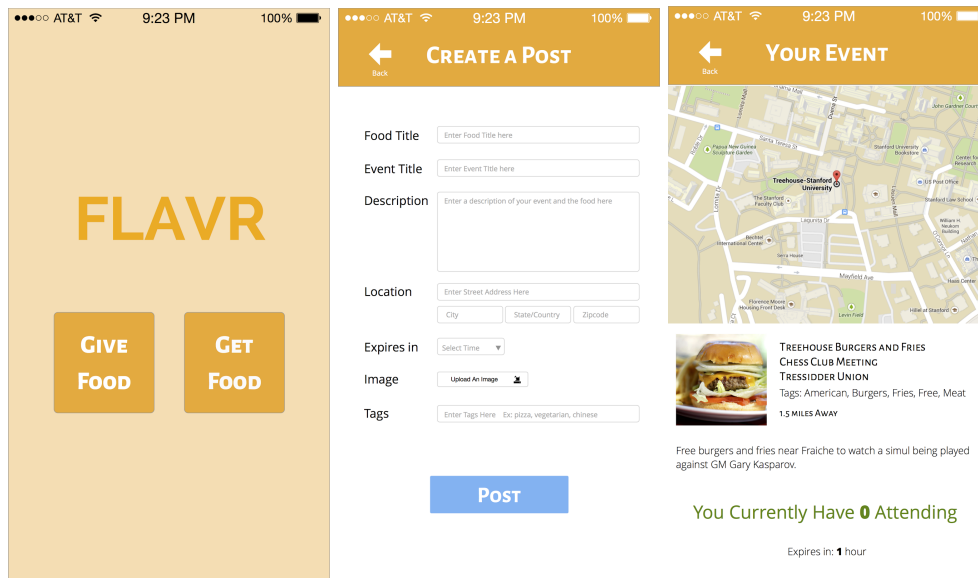


Figure 2a-c: The Give Food flow from the home screen in a) through the confirmation in c)

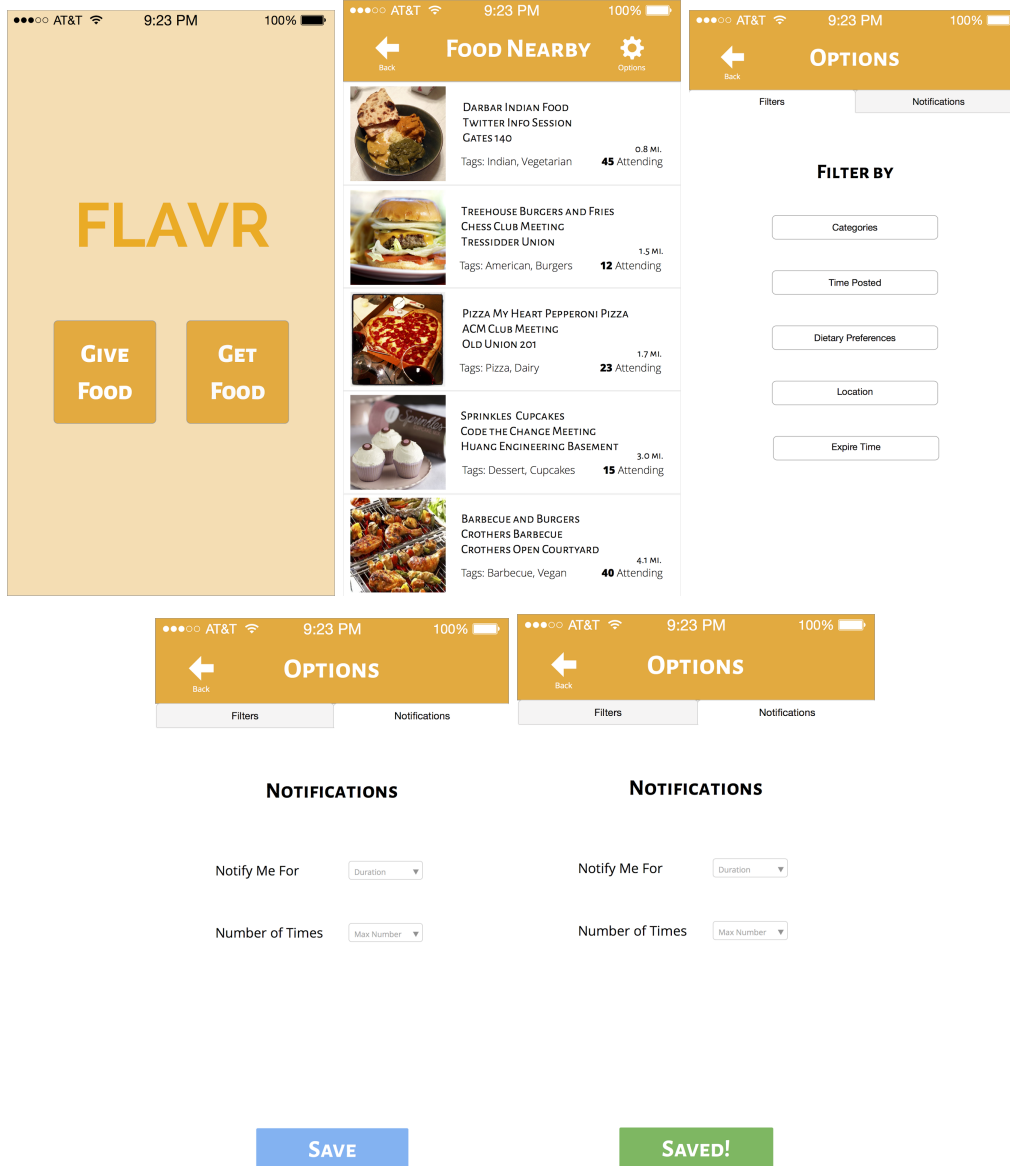


Figure 3a-e: The Notifications flow from the home screen in a) through the confirmation in e)