Craving

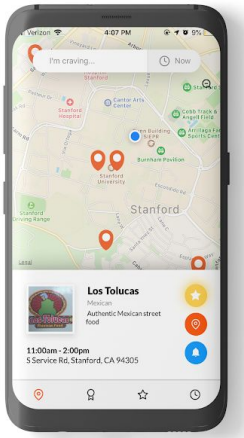
Find a food truck. Calm your craving.

Jessica Chen
Thomas Hsieh
Paula Sáyago
Jennifer Tao

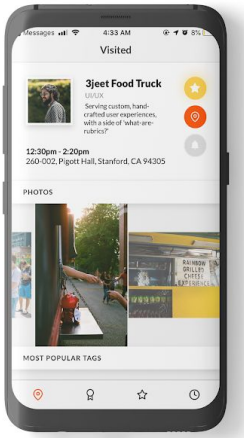
Problem and Solution Overview

Young working adults are always looking for cheap, convenient, and delicious food. However, they often do not consider food trucks during their search. We found that food trucks are considered unreliable because of their changing schedules and locations. Our mission is to connect consumers to food trucks by simplifying their discovery, tracking, and evaluation.

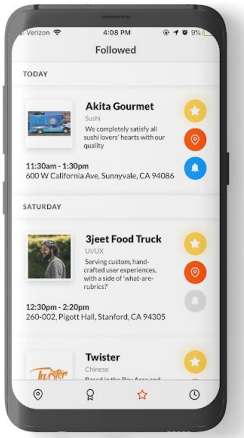
Our basic approach is a service that locates food trucks on a map, displays detailed truck profiles including a tag-based review system, allows users to follow food trucks, and gives rewards for continued engagement with the app.



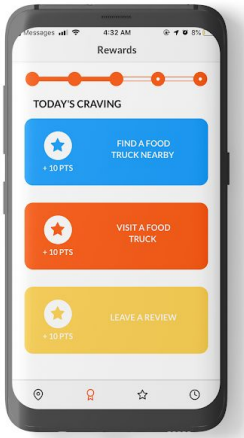
Search trucks by location and time



View detailed truck information



Follow favorite trucks' schedules



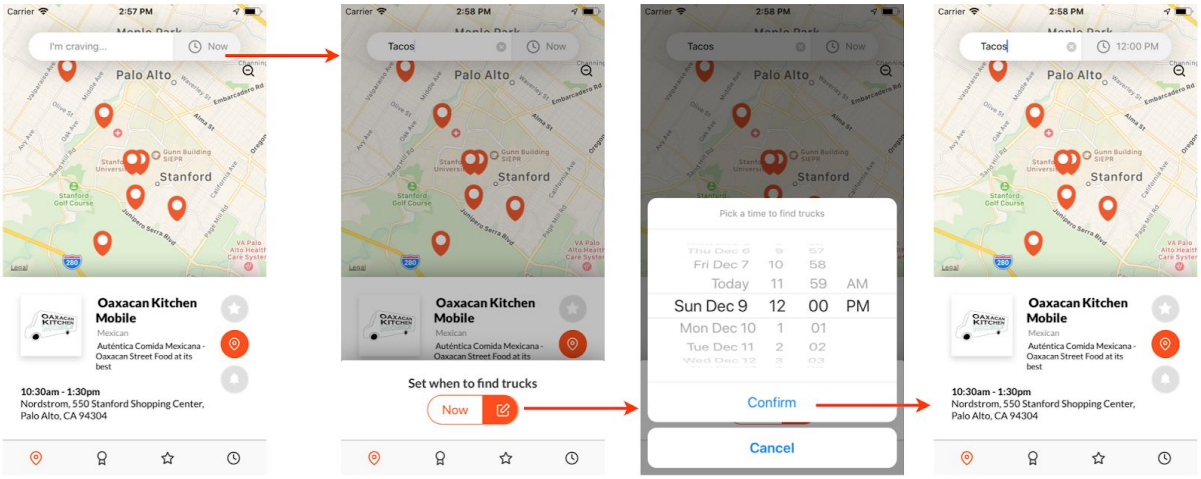
Get rewards for visiting trucks

Tasks and Final Interface Scenarios

Our three main tasks are as follows:

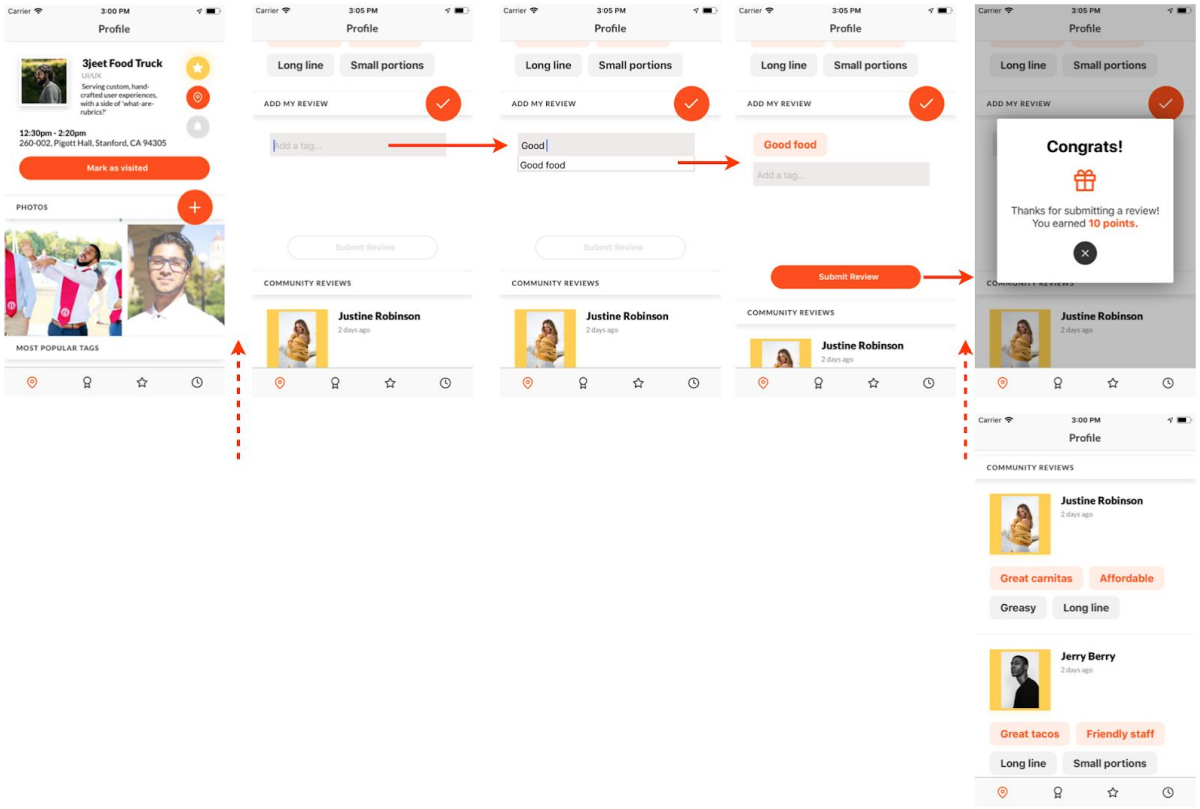
1. Find nearby food trucks – simple

   It is difficult to find food trucks because their locations change. Being able to search for food trucks by both time and location makes it easier for our user to enjoy food trucks.



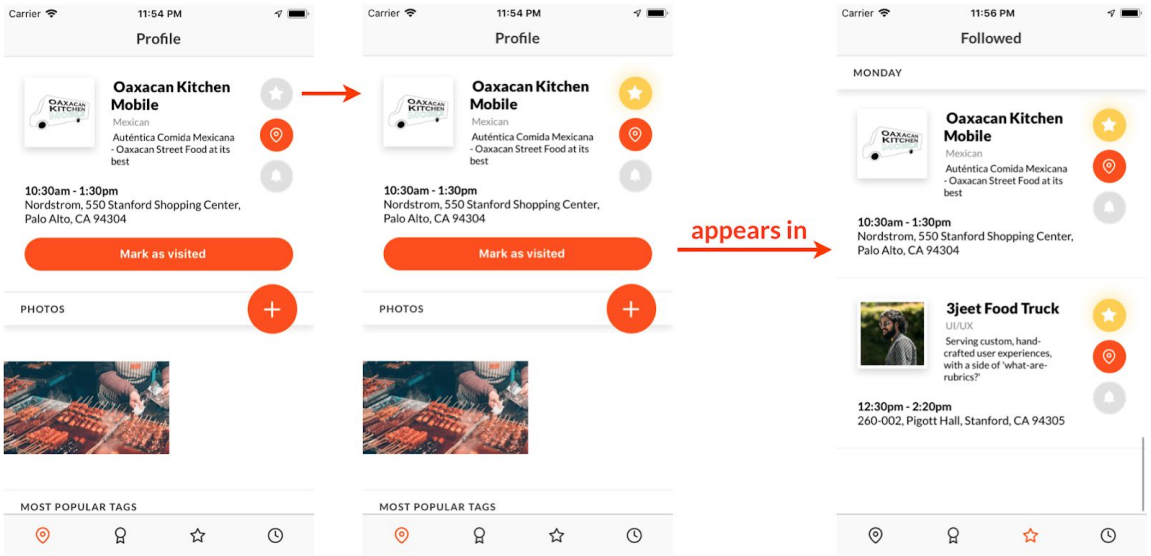
2. Read and write food truck reviews – medium

   Our user's preferred way to find new places to eat is by word-of-mouth. However, again because of the mobility of food trucks, it is difficult for food trucks to establish a local word-of-mouth community. This is why our review system prioritizes reviews from our user's friends, but still makes reviews from anyone available.



3. Keep track of visited and liked food trucks – complex

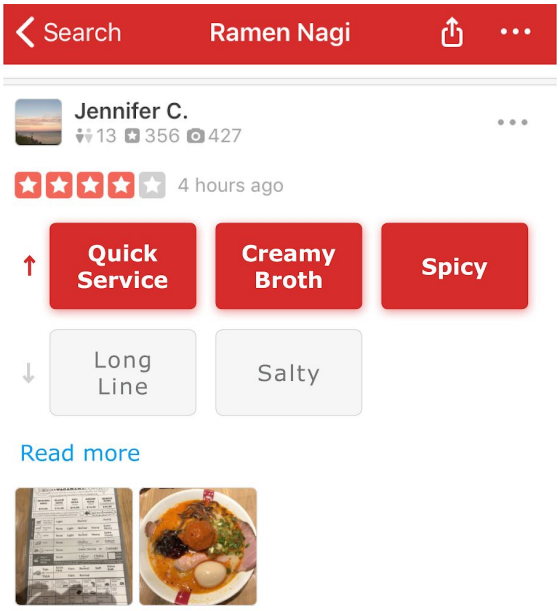
Although food trucks are delicious and our user would like to return to their favorites, without Craving, finding the same food truck requires remembering both its name and appearance, searching for a truck website that may or may not exist, and potentially needing to make phone calls. This complicated process prevents our user from returning to food trucks. Craving provides list views of visited and followed food trucks and reminds the user of when the food truck will come back to their area.



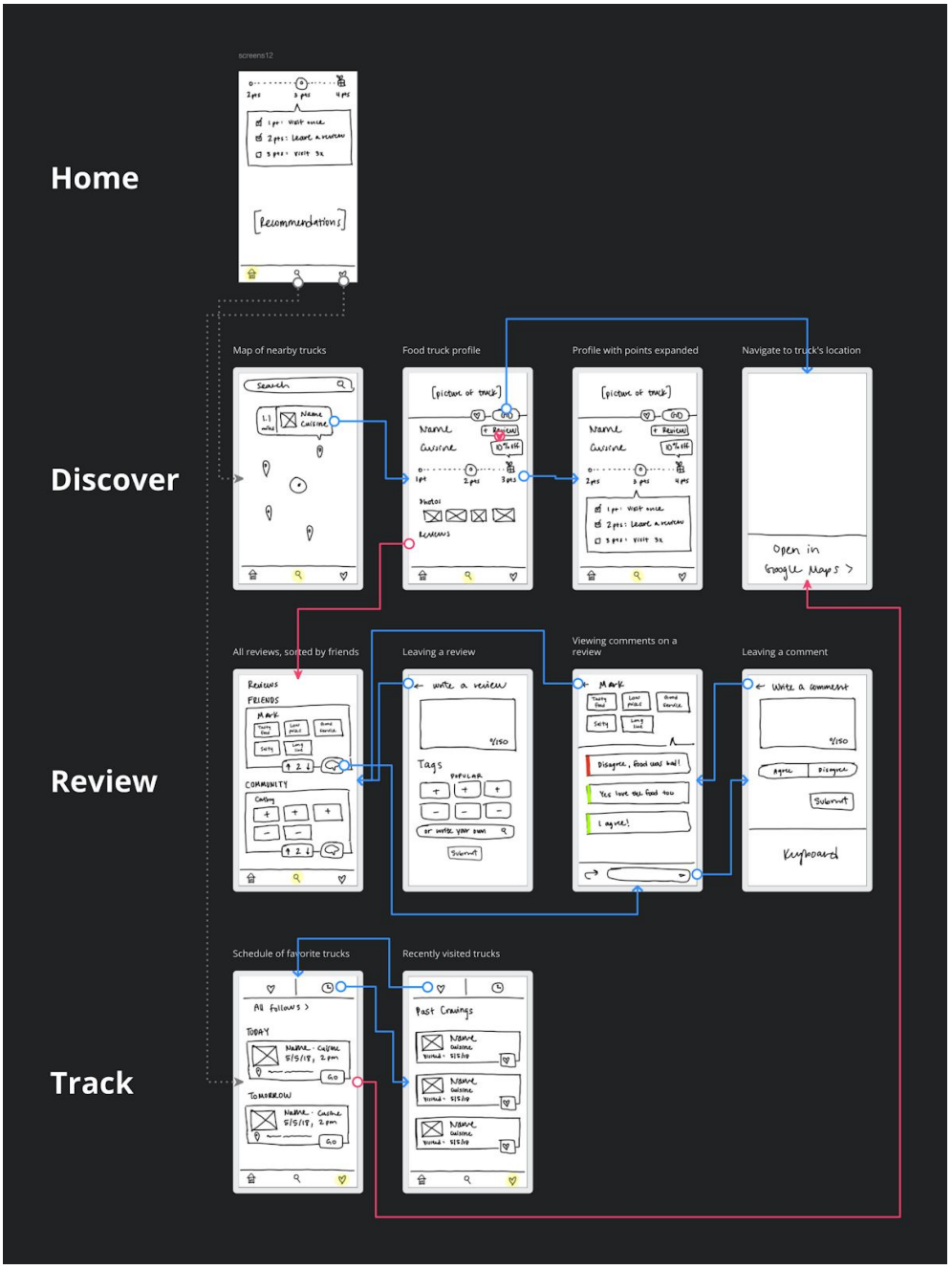
Design Evolution

We started our design with three experience prototypes. The one most related to our final product is a tag-based review system. During our needfinding, we learned that people don't like to read long Yelp reviews because they contain a lot of detail and stories that may be irrelevant, especially when they are trying to find a place to eat quickly. At the same time, short reviews such as "Great place. Would recommend!" or "Terrible, don't eat here" do not provide any useful information.

We solved this issue by using tags. Tags force reviewers to give concrete detail, but also limits the content to information that is universally relevant. We used an edited version of Yelp, a familiar interface, to test this idea. Users reacted positively to this prototype.



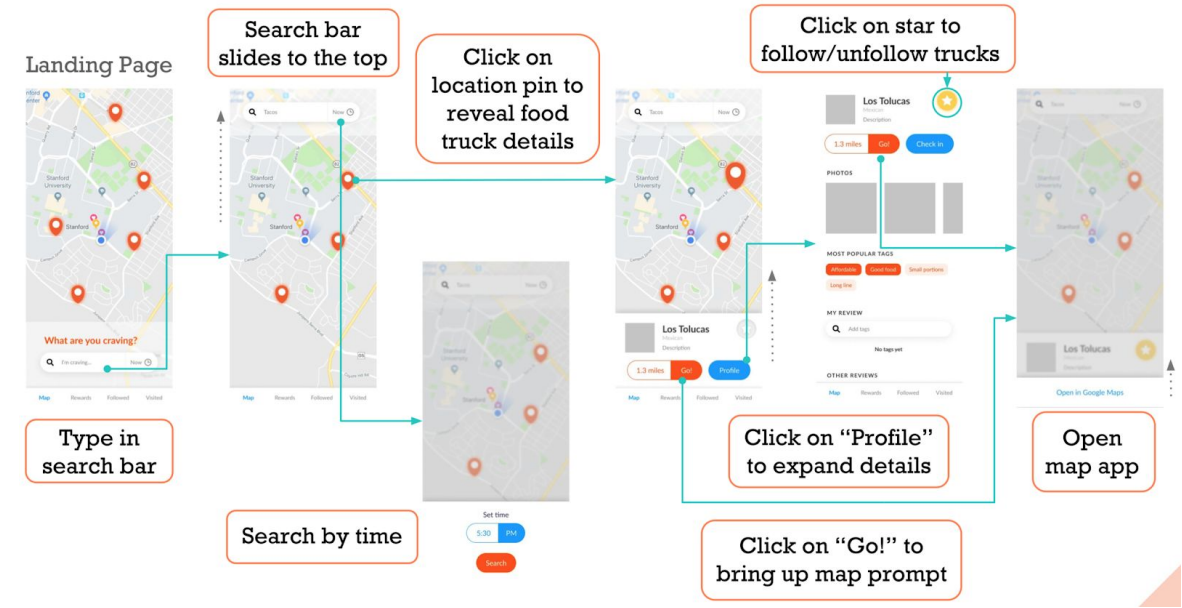
Our next step was to create a hand-drawn prototype (accessible here). We incorporated the tag-based reviews. We also added a map view, truck profile pages, a visited/following trucks page, and a rewards page.

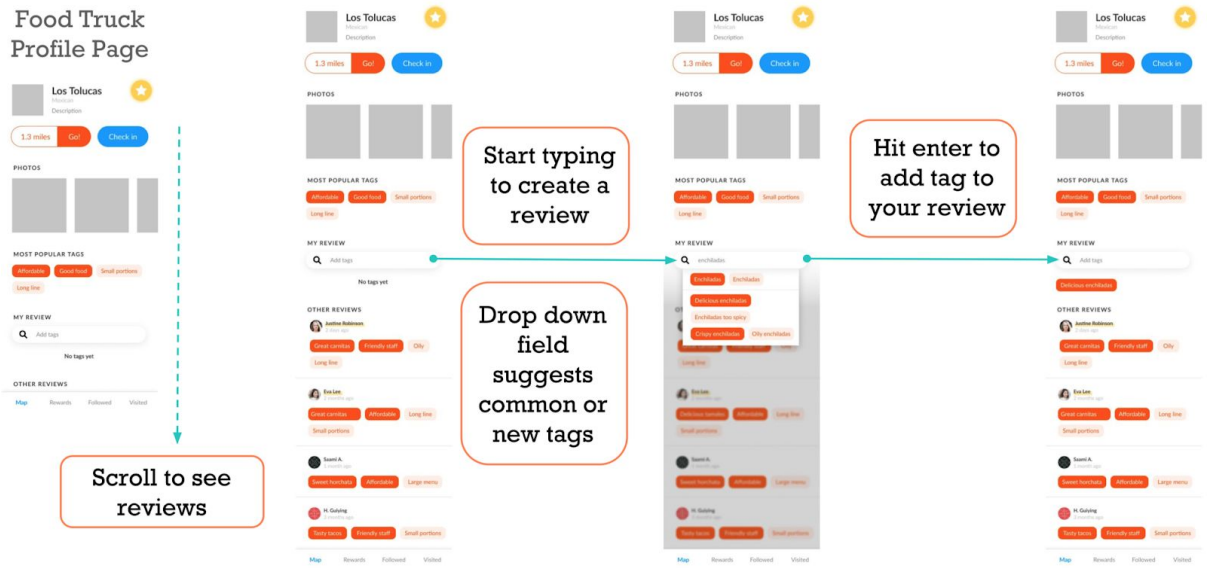


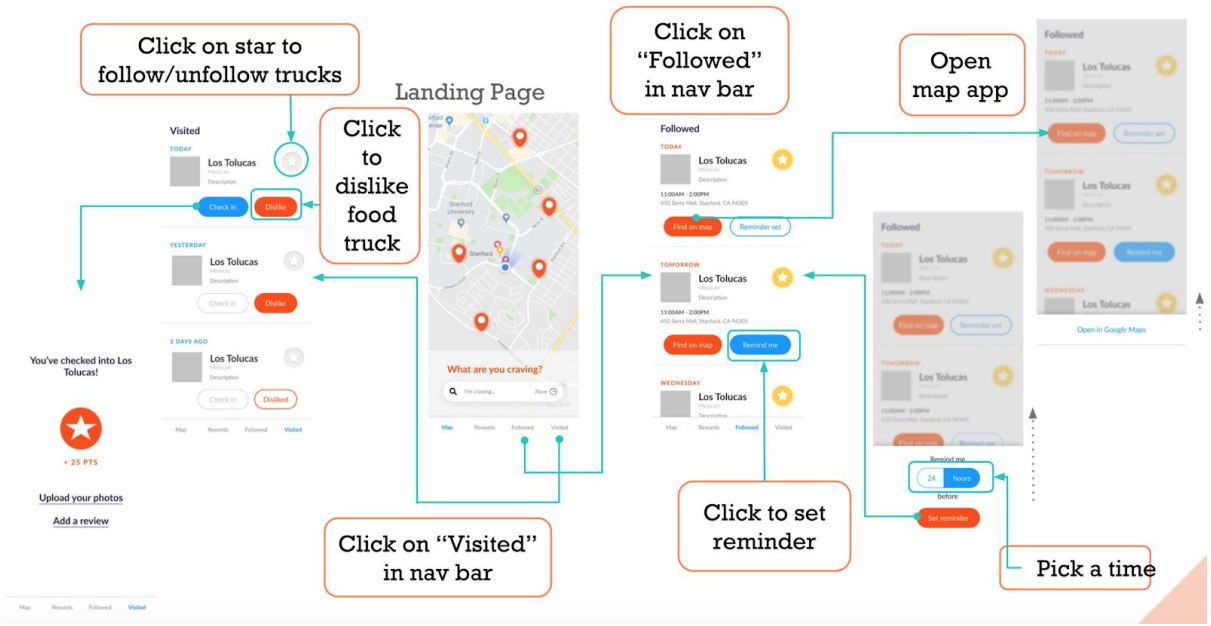
We followed a script while doing user testing, asked participants to talk us through their thoughts and actions, and recorded video of the participants interacting with the prototype. Participants easily understood our navigation structure and icons.

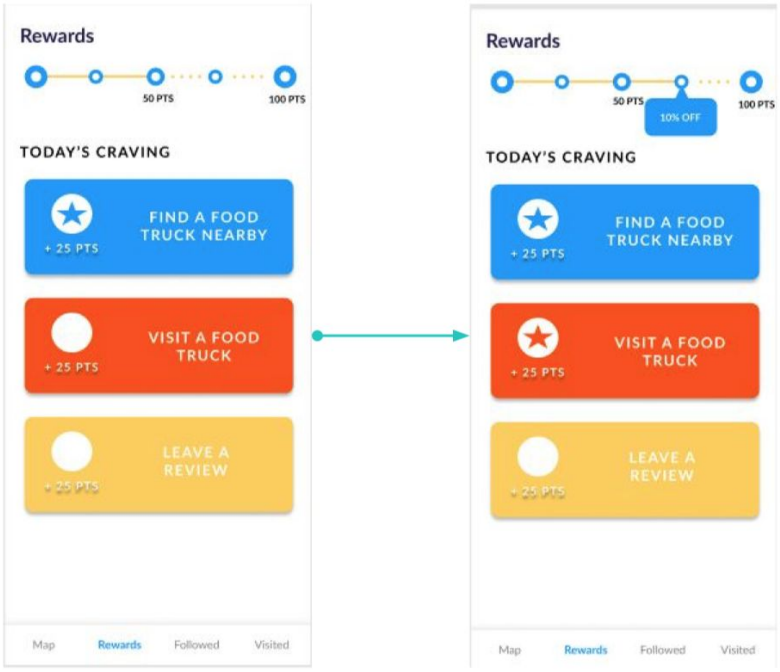
The rewards system caused some confusion because we had two different incentive structures: one for interacting with the app overall, and one rewarding repeated interaction with the same truck. Participants also expressed confusion over the up/downvote and commenting systems within the reviews screen. Additionally, participants expressed a strong preferences for the map screen to be the first screen that loads upon opening the app.

We continued design iteration through a mid-fidelity prototype created on Figma. We added a pop-up card on the Map view to help users get quick information on food trucks without switching views, as well as an option to search by time so that users could plan ahead. The Following and Visited Trucks pages were separated because we decided that they actually served different purposes – for example, a user may be following a truck they have never visited. We eliminated commenting and voting from the review system, and centralized rewards onto one screen.







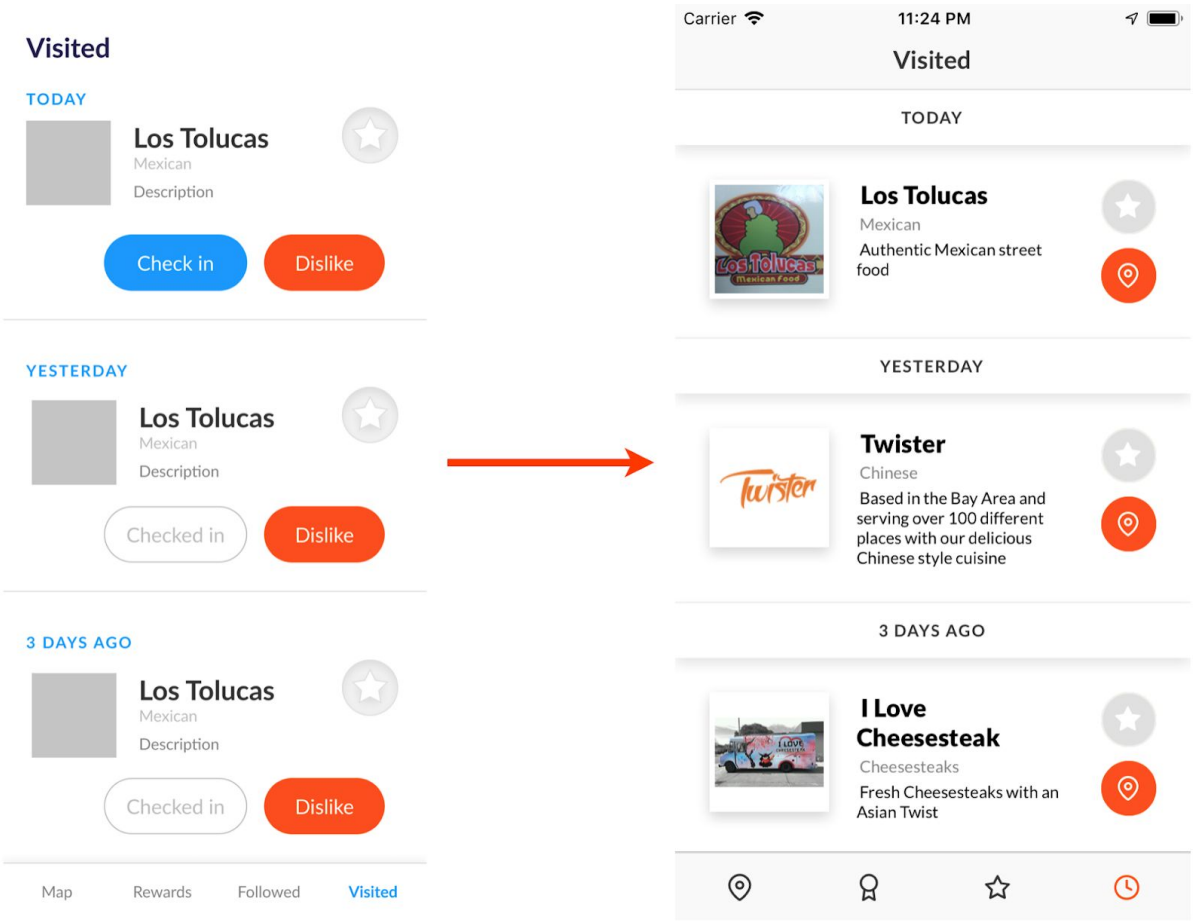


Major Usability Problems Addressed

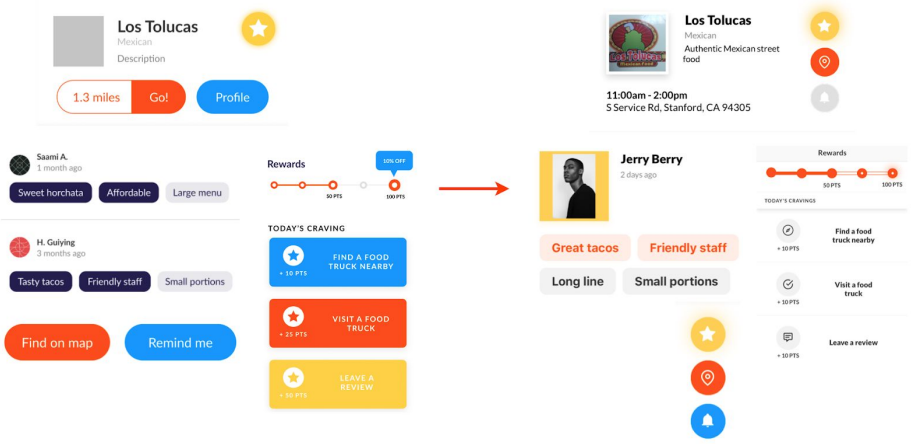
Following from our heuristic evaluation, we implemented several changes to our app. Below is a summary of changes. Violation numbers refer to the linked heuristic evaluation document.

Eliminating dislike feature (Violations 4, 29)
Our evaluators were not sure what disliking a truck would do, or how to undo an accidental dislike. We decided that the dislike feature did not provide enough benefit to justify the added complexity that would be necessary to fix these issues. Theoretically, this would make users have to remember which food trucks they

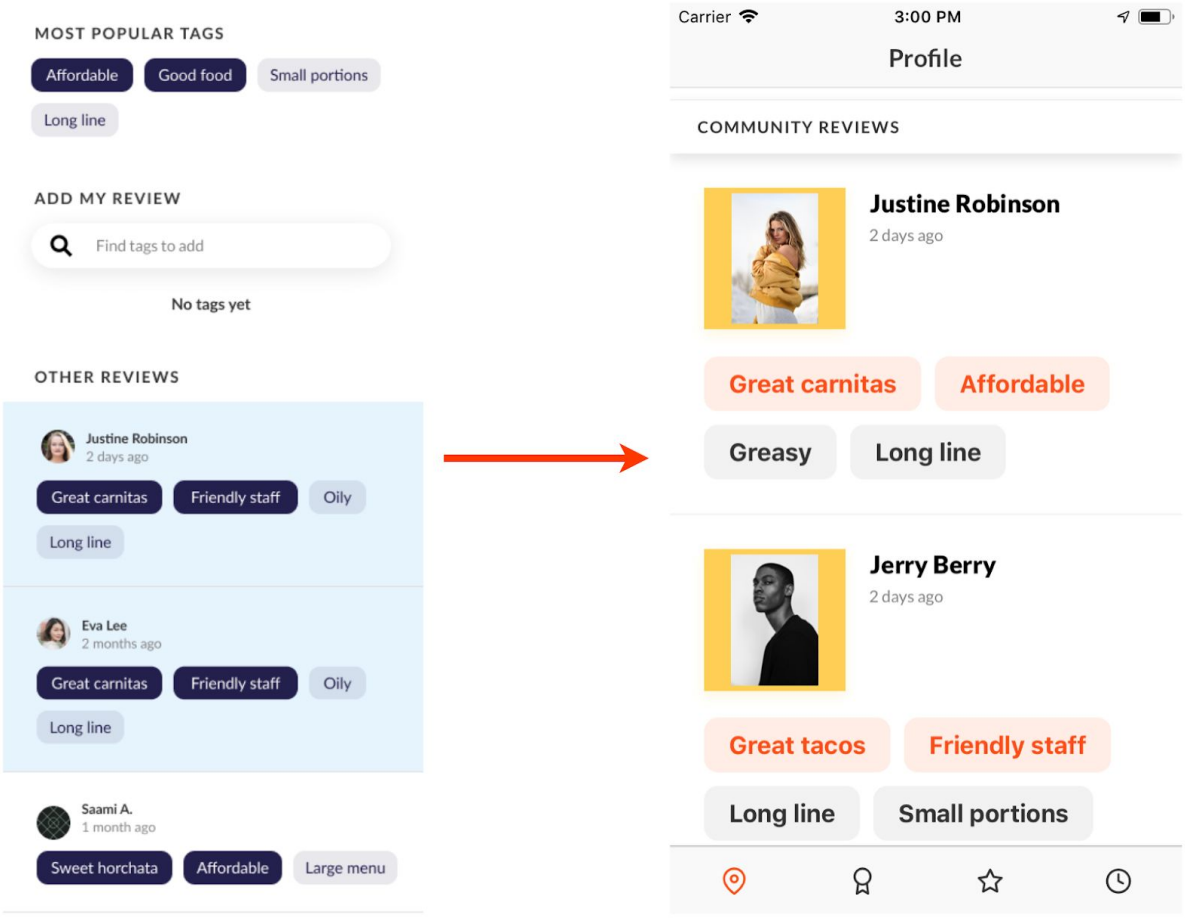
disliked in real life, but this issue is prevented because each truck profile page prominently displays the user's own review.



Changed colors, text, and navigation bar (Violations 13, 14, 22)
Our evaluators and TA noticed some inconsistencies with the way we were using color. Our mid-fi prototype uses a primary-color scheme, but we designed it with an eye for aesthetics rather than for informational consistency. Therefore, we standardized to red for reviews, as well as rewards and anything that contributes to them; yellow for following; and blue for reminders.

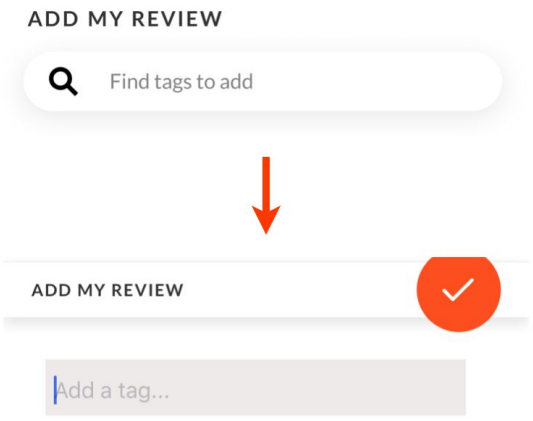


Evaluators were also confused by the dark and light blue of positive and negative review tags, respectively. We changed the colors to red for positive and gray for negative, as consistent with our revised color scheme.

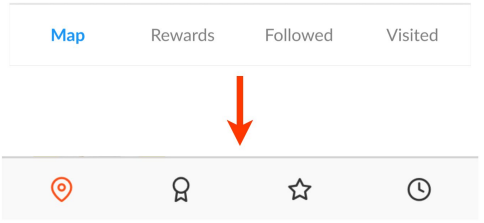


Our tag-adding field was originally labelled “find tags to add.” This suggested that the app already had a database of all possible tags, while in reality, users can add either custom tags or tags that other users have added. We changed the field label to “add a

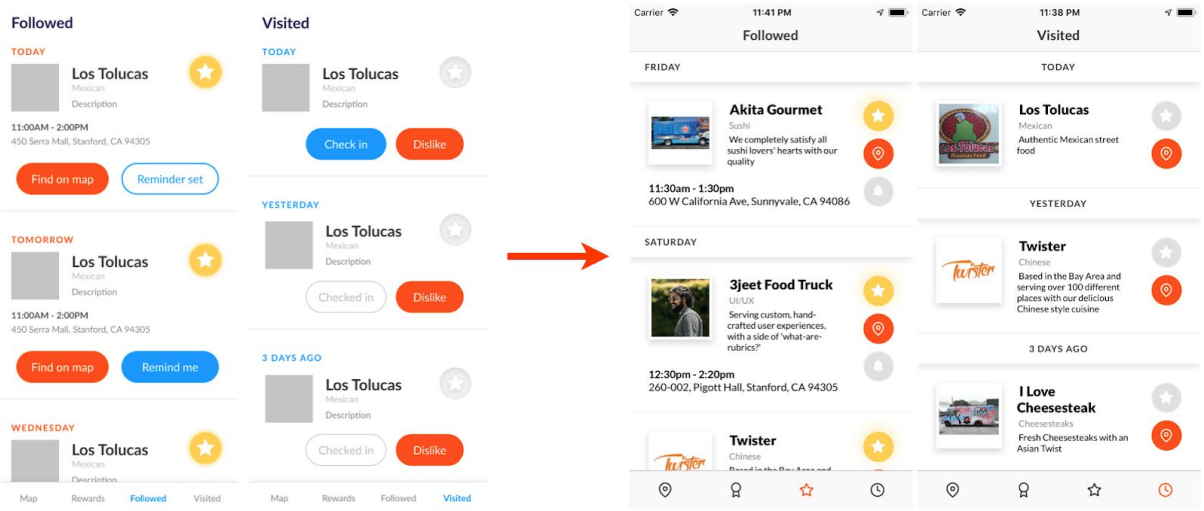
tag," which does not imply a method for how the tag is added; searching existing tags happens via a pop-up suggestion as the user is typing.



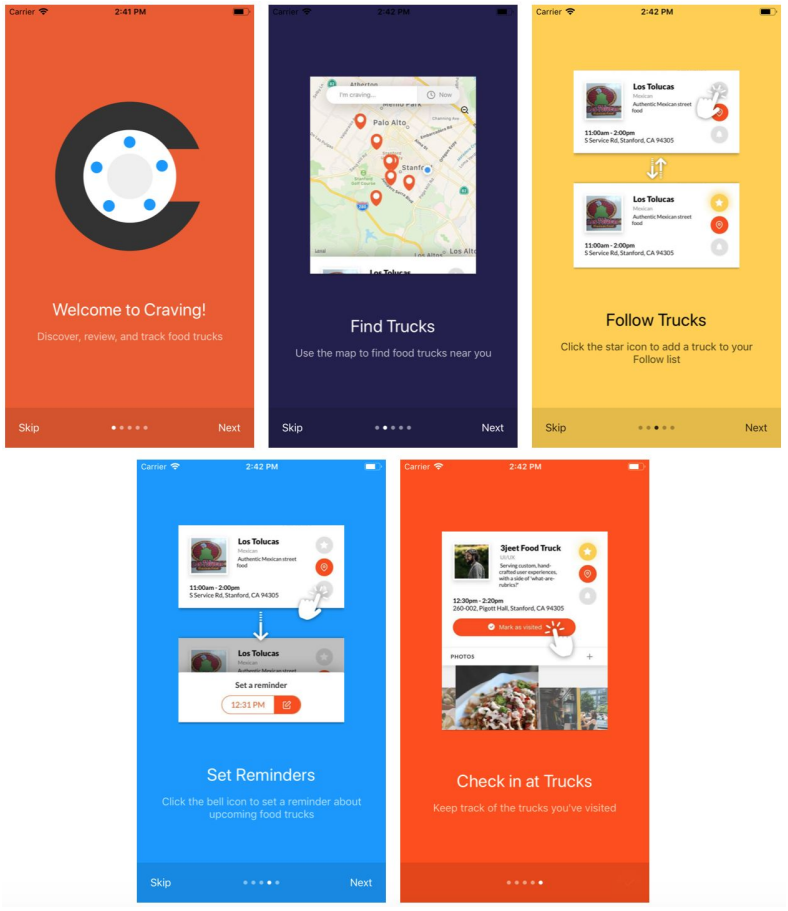
Finally, we changed the navigation from text-based to icon-based to reduce visual clutter.



Differentiation of Following and Visited screens (Violation 26)
Our med-fi prototype has two separate Following and Visited screens, but their functions are similar. One especially confusing aspect of this prototype is that Visited trucks still have a button to check in. After rethinking the process of checking into a food truck, we decided that checking in is the process that adds trucks to the Visited list. Therefore, it made sense to take the check-in button out from the Visited page.



Added onboarding on first launch (Violations 3, 13)
The check-in and following features were not sufficiently explained by the use of icons. Therefore, we added an onboarding tutorial that explains the main features of the app upon first launch.



Completing prototype implementation (Violations 1, 2, 6)
Several of our high-severity heuristic violations were due to prototyping limitations in Figma. Many of these were fixed by using React Native to build our high-fidelity prototype. These fixes include:

- Allowing follow/unfollow using the same icon.
- Allowing selection of food trucks from the markers seen on the main screen without first searching with a query.
- Linking truck listings in the Following and Visited pages to their respective profiles.

Unaddressed heuristic violations (Violations 27, 28)
Our evaluators wanted the option to add longer reviews in addition to tag-based. Although we would be open to exploring this option in future versions of the app, we did not prioritize this implementation because we felt that it added too much complexity to deal with two types of reviews, considering the scope of this prototype.

Finally, despite our evaluators' recommendation, our app currently does not provide food truck menus. This is another feature that we would be interested in recommending, but did not feel was critical to implement in these early stages of the product.

Prototype implementation

We used React Native and Expo to build our high-fidelity prototype. We chose React Native and Expo due to our familiarity with these technologies. We did not want our prototype to run exclusively on one mobile platform, and React Native + Expo allowed us to build our prototype to run on both iOS and Android.

One perk of working with React Native was that there were many third party components that provided functionality we desired. Using third party solutions drastically cut down the amount of programming from scratch that we needed to do ourselves. At the same time, many of these components did not include good documentation, so figuring out which component best fit our needs and how to use it in our own app was at times incredibly time-consuming. A complete list of third party components that we used can be found in the README of our project GitHub repository.

Several points of our app use Wizard of Oz techniques. The search bar on the initial map screen is programmed to display one search result when a user types in "tacos." We wanted to demonstrate what the search experience felt like, but did not have the data necessary to fully and correctly implement location-based search. The tags-based review system is also powered by Wizard of Oz techniques. We programmed in an

initial list of tags which populate the drop-down recommendations list. Any new tags a user adds to the system do not persist and will not show up as a recommended tag.

Additionally, all food truck data was hardcoded. This was not just for ease of implementation, but because a database that includes information necessary to our app does not exist, to the best of our knowledge. The location pins on the map are generated based on hard-coded latitude and longitude coordinates, rather than real-world locations. The lists of Followed and Visited trucks are hardcoded, as well as the reviews and popular tags displayed on food truck profiles are also hardcoded.

In terms of features, one that was suggested to us during needfinding and heuristic evaluation was adding food truck menus to their profiles. This feature is currently missing from our high fidelity prototype, but is one we would want to include in the future. As a next step, we would also really like to add a database. Having a proper database powering our prototype would enable us to dynamically load views such as the Followed and Visited food truck lists. We would also love to correctly implement searching and loading location pins on the map. One last feature that we might add in the future is a proper review system that actually adds user reviews to trucks' profile pages and adds tags to a database of tag suggestions.

Summary

Craving is an innovative new app that solves the problems inherent to the mobile nature of food trucks. It allows food enthusiasts to connect to food trucks in the community, giving food trucks the flexibility to move around without sacrificing the strong ties to their customers that more sedentary businesses enjoy. We interviewed young working adults around the Bay Area about their attitudes towards food trucks and kept the focus on their needs as we refined our app. The result is a strong foundational product that improves the visibility of food trucks and satisfies the cravings of young workers.