

CleanPlate: Hi-fi Report

Brandon Liu
Bryan McLellan
Michael Zhu

Value proposition:

Give more, get more, waste less

Problem and Solution Overview

Millions in America are going hungry. Meanwhile, restaurants are wasting perfectly edible food at the end of the day. CleanPlate seeks to bridge the gap between hunger and waste by delivering excess food from restaurants to food banks and rewards you with discounts from those restaurants.



Tasks and Final Interface Scenarios

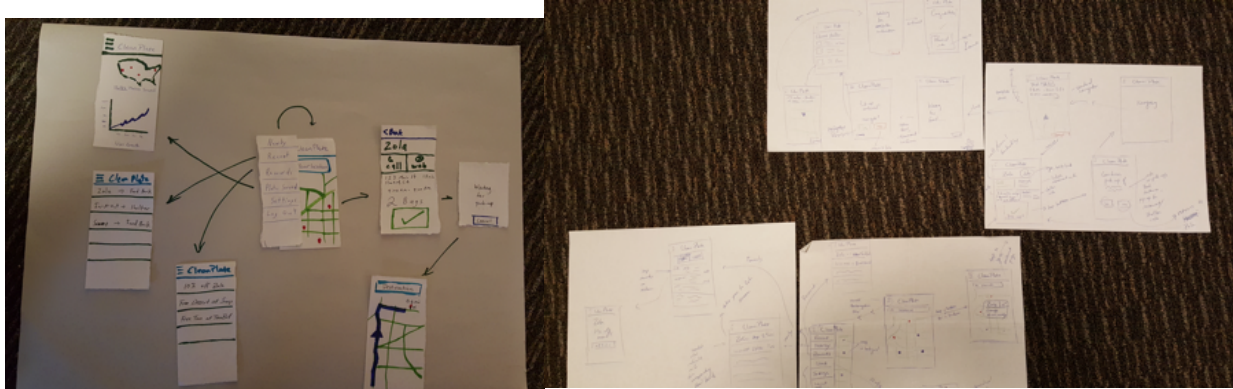
Simple: Redeem a reward

Medium: Add your favorite restaurant so that you can receive notifications for your “favorite” restaurants that currently have food

Complex: Pick up leftover food from a restaurant and deliver to a food shelter

Design Evolution

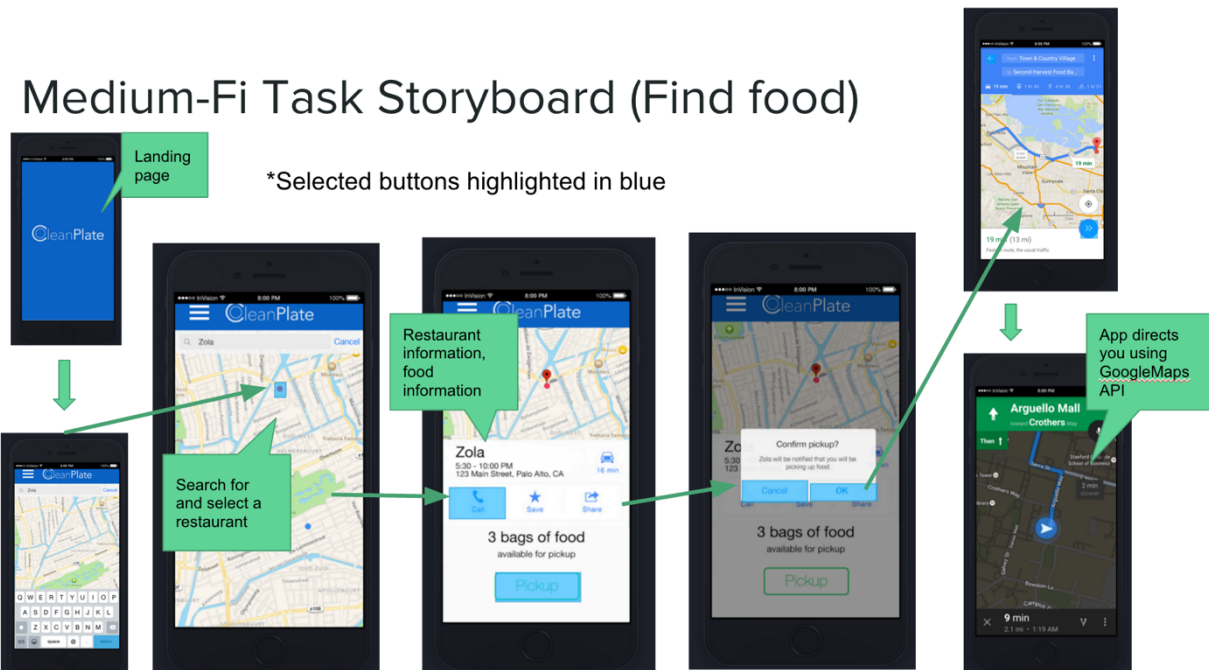
Early-stage storyboarding featured pen-and-paper sketches. Our simple UI included a main page and side navbar to go between different pages, including “nearby,” “recent pages,” “global stats,” and “rewards.” After user testing, we maintained the core functions but revised many of the button layouts and made changes to increase the visibility of the system status by more clearly labeling the buttons.



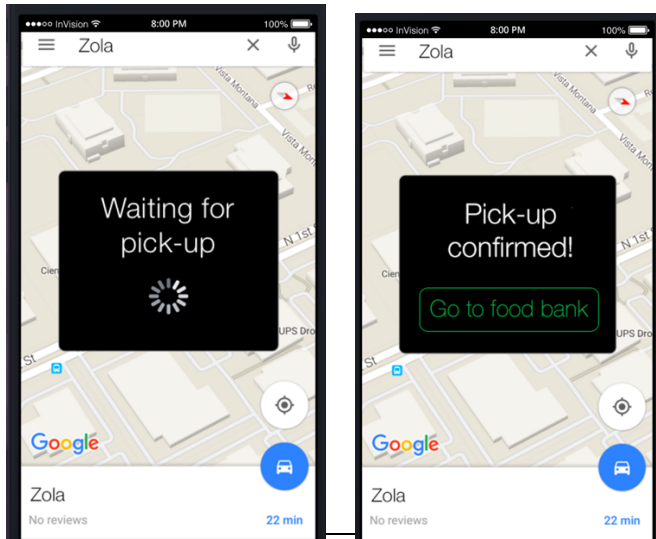
Annotated sketch revisions included modifications to the restaurant information page. User-testing feedback showed that testers were confused by this screen and the lack of affordances. The information page now features labeled buttons and more information for the user (e.g. phone number, quantity and type of food, location, website, etc.)

The updated medium-fi prototype featured modifications to the store information page, the rewards page, and added additional confirmation buttons at various stages during the delivery process. The medium-fi prototype included additional affordances to assist navigation through the pick-up and delivery process.

Medium-Fi Task Storyboard (Find food)



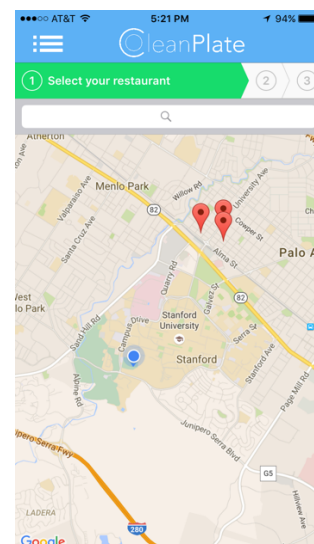
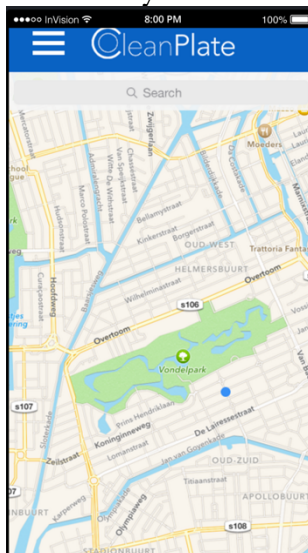
We also modified pending and approval screens to have a pop-up instead of generating a new screen in order to keep the information in the same location.



Major Usability Problems Addressed

1) Visibility of System Status – 3

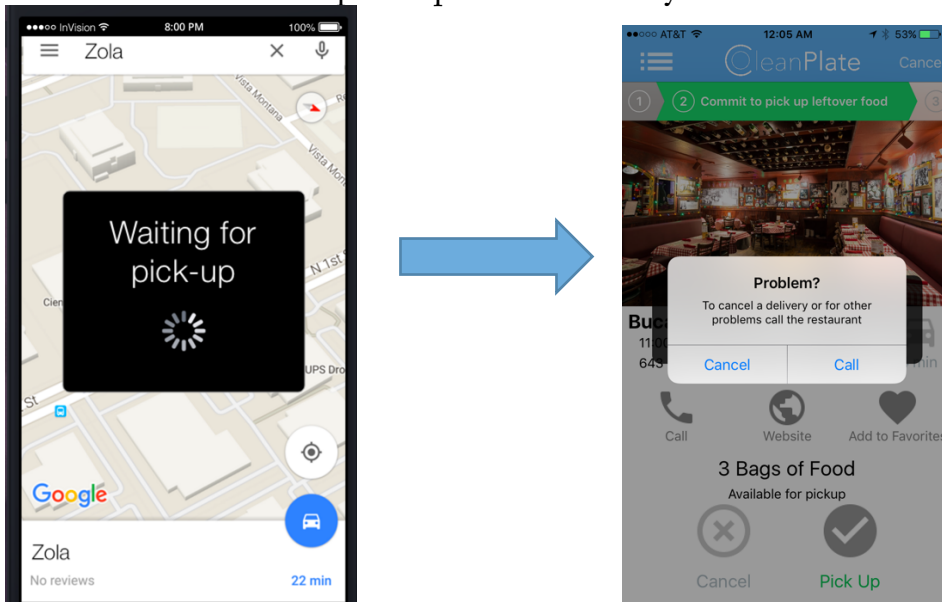
Evaluators found the home screen map confusing to use. We opted to maintain this home screen but added a progress bar to include the current system status as users complete steps to pick up and deliver food. The main home screen status is “select your restaurant.”



2) User control and freedom – 4

Once the user clicks “pickup” to pickup the food, they are stuck there. If something messes up or the restaurant forgets to confirm the pick-up, the

application could spin indefinitely. We added a cancel button in the top right corner to back out of the pick-up which allows you to cancel or call the restaurant.

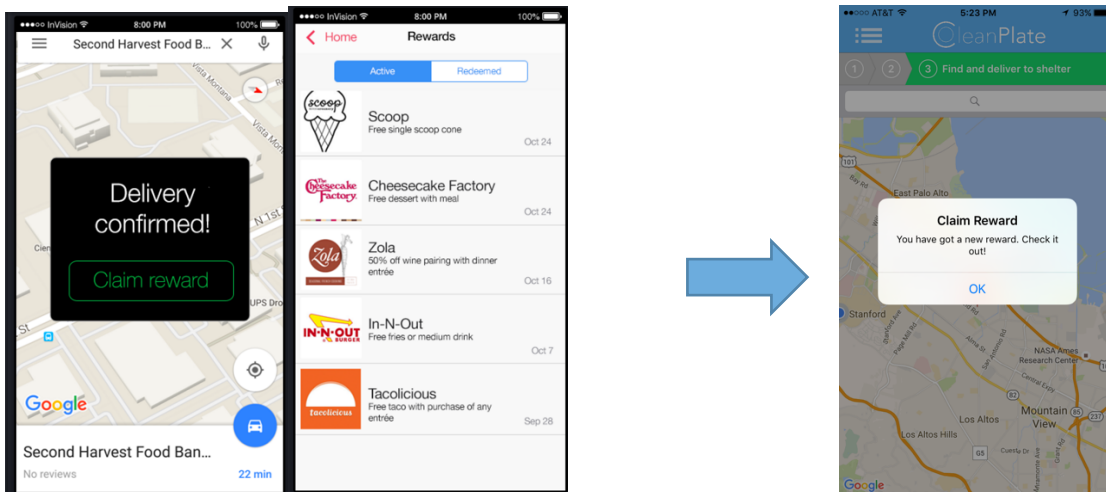


3) *User control and freedom – 4*

Users complained that they got stuck in the phone interface once they click call—this was an issue with the InVision prototype. Leaving the call now takes you directly back to the application.

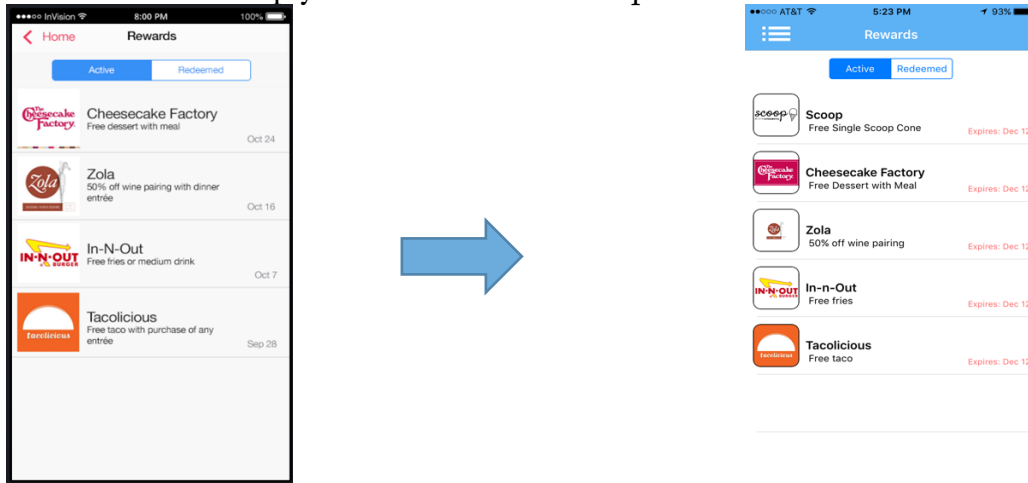
4) *Consistency and standards – 3*

After completing a task, it was unclear if the user was adding a new reward or redeeming an existing one—the application simply navigated the user to the rewards page. We updated the description to display a message stating that you have a new reward.



5) *Recognition rather than recall – 3*

On the rewards menu, dates are listed to each reward, but it is unclear what the date means. We simply listed the dates as expiration dates.



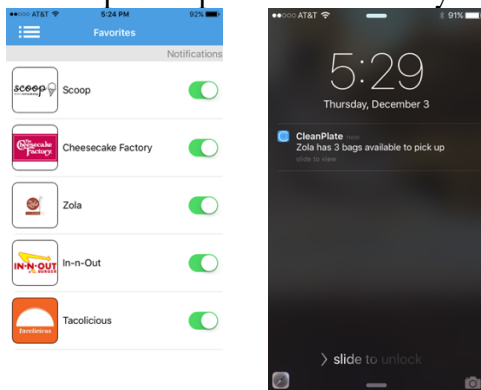
6) *Error prevention – 4*

After delivering food to the food bank, a user could potentially be stuck indefinitely on the “waiting for delivery confirmation” screen indefinitely if the food bank forgets to check the delivery completion.

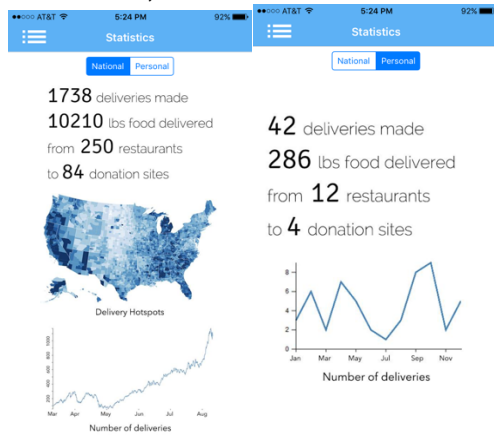
We chose not to address this because in a polished production quality prototype, there would ideally be a timeout after some period of time has passed. However, our Final prototype does not feature a separate client for restaurants and shelters. Since we use a “Wizard of Oz” technique to simulate backend, it does not make sense to address this.

New features

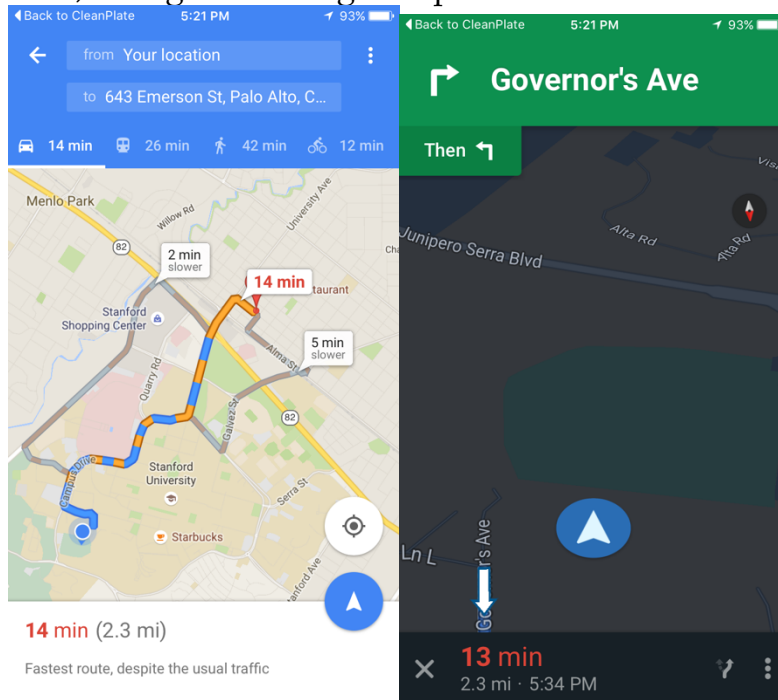
- 1) We added favorites so that a user can receive push notifications whenever their favorite restaurants have food. We revised our tasks to allow users to participate more fluidly without having to open the app every time.



2) We implemented global statistics (hardcoded data but wasn't functioning for med-fi).



3) Integrated GoogleMaps API



Prototype Implementation (Bryan)

Bryan was familiar with Xcode from prior iOS projects, so he knew how to use the development environment and Swift. He laid out all of the screens using the Main.storyboard file and wired segues between them to dictate the navigation of the app. He wrote custom swift files to handle each view controller and used Parse as a service to send push notifications.

Xcode was a bit tough to get to work with in implementing the push notifications. Additionally, the storyboard file, though it helped Bryan lay out the views, does not help that much in terms of designing specifically for an iPhone 6, which Bryan was developing on. Also, some of the updates to iOS 9 from iOS 8 changed the implementation of some of the app.

The Wizard of Oz functionality is in the backend of the application. We did not develop the restaurant/shelter sides of the application. Therefore, all waiting screens and receipt of rewards are simulated.

Most information on the screens, apart from the GoogleMaps navigation portions are hard-coded data. Restaurant data and food data were hard-coded. We also hard-coded shelter data, favorites, and preset rewards. The global statistics data is also hard-coded.

Of course, this hi-fi prototype information only features the user-end of the application. The future implementation will need to include a separate restaurant/shelter application to interface with our user's client. We'll need to develop the infrastructure and data storage tools to track user, restaurant, and food bank information. This means additional quality assurance metrics (i.e. ensuring that restaurants are providing quality produce to food banks), potential tax break agreements with local and state governments for participating restaurants, and fully formed media and publicity packages for participating restaurants.

On the user side, we'd also like to be able to dynamically calculate trip length for picking up and delivering food. Settings, account info, log in/out would also be additional features in a production quality application. We could also apply more detailed data analysis to our statistics page.

Summary

We began with the simple idea of empowering the average person to bridge the gap between hunger issues and food waste. Through iterative user testing and additional needfinding, we arrived at the current solution: to allow users to save excess, unopened food and produce from restaurants, and deliver that to food banks and food shelters in exchange for rewards.

The UI design process began with pen-and-paper sketches that included a side navbar to shift between different sections of the app and a map on the home page for finding nearby restaurants. Medium-fi and low-fi prototypes featured separate screens for nearly every step of the process. However, in the final prototype, we opted to minimize the number of screens if possible to eliminate recognition and recall issues throughout the pick-up and delivery process.

We also improved the consistency and system visibility throughout the app by providing users with a status bar at the top, updating colors and the logo, and adding affordances to the pick-up and delivery screens. The med-fi prototype featured Wizard of Oz techniques for navigation and pick-up, but we added hard-coded locations and interfacing with the GoogleMaps API to provide users with realtime navigation capability.

The final product features an efficient user experiences but still lacks the restaurant/food bank interface and a true back-end. We will look to further develop this and gather more need finding and user testing information to determine usage habits and needs on the restaurant/food bank sides of the application as well.