Assignment #8: Final Report

Problem & Solution Overview

The current problem is that people are having trouble finding suitable parking spaces in the Bay Area. And we aim to solve this problem by creating new parking spaces, maximizing the usage of currently available parking spots and conveying these info to our users (Fig.1).

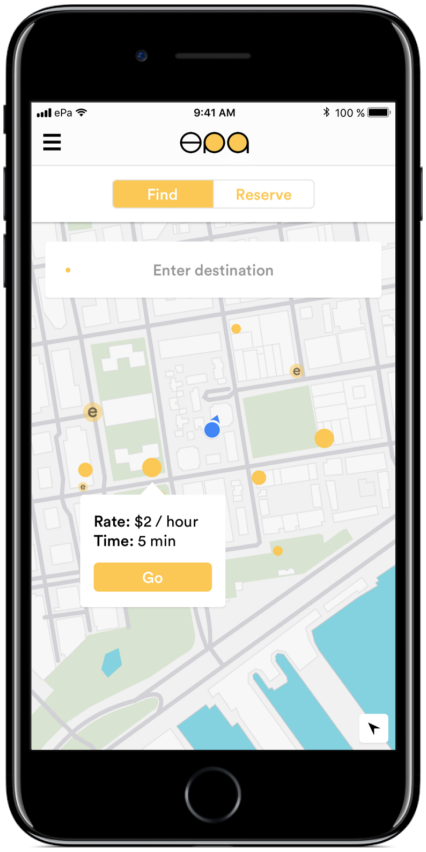


Fig.1 Homepage

Tasks & Final Interface Scenarios

Simple Task: Find parking spots in real time (Fig.2)
Medium Task: Reserve spots in advance (Fig.3)
Complex Task: Rent out private parking spots (Fig.4)

The three tasks are all chosen, first of all, for their functionalities and potential in solving our identified user problems as stated in the “Problem & Solution Overview” section above. More specifically, the simple task provides the basic and probably most-frequently used functionality of allowing user to find parking spots in real time while the medium and complex tasks fulfill the

need to maximize the usage of currently available parking spots and create new parking spaces and fundamentally differentiate us from our competitors.

Fig.2 Simple task

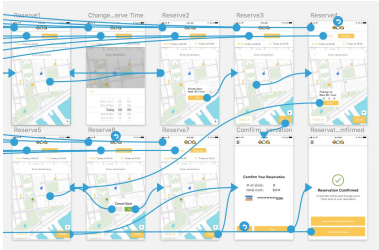


Fig.3 Medium task

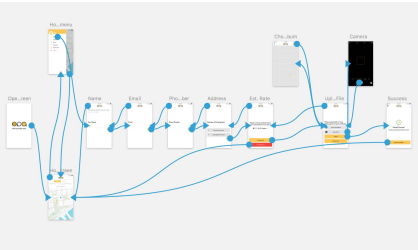


Fig.4 Complex task

Design Evolution

1. First of all, after deciding to focus on improving the transportation experience in Bay Area, we conducted needfinding through in-person interviews. We developed several POVs and from there, derived our 3 HMWs, namely HMW enable passengers who may be affected by the malfunctioning accommodation facilities to learn about the situation in advance, HMW encourage more people to share ride while going to work and last but not least, and HMW make a car like an office.

2. Based on our 3 HMW statements, we built our experience prototypes (Fig.5,6) around the themes of carpooling and parking. We then conducted user testing with our experience prototypes. From users' feedback, we eventually decided to focus on building an app that addresses the parking issue in the Bay Area.

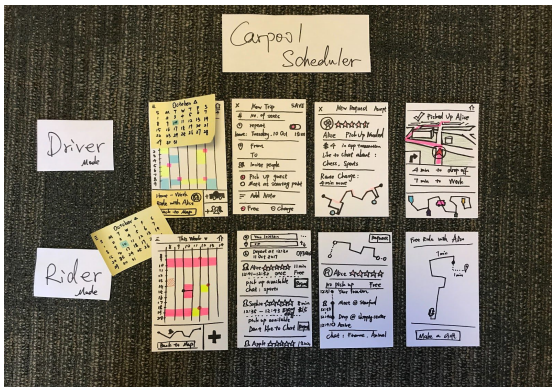


Fig.5 Carpool experience prototype

Fig.6 Parking experience prototype

3. Then, through the use of concept sketches (Fig.7), UI sketches (Fig.8) and storyboarding (Fig.9), we explored different mediums and eventually decided to build a mobile app. The mobile platform was chosen because, first of all, during the needfinding process, all the interviewees who drive to work acknowledge that they rely on Google Maps, hence their mobile phones to navigate while driving. And none of them mentioned using their laptops or

any web apps in the car. In addition, from the design perspective, web apps usually involve complicated UI components, which cooperate with one another to enable more complicated tasks. However, in the case of our tasks which are rather simple and straightforward, we believe mobile app is a better fit. Last but not least, based on our intuition, we feel that the mobile platform has become the mainstream and is used by the majority of the population. Thus, a mobile app would be more suitable than a web app as it could best take advantage of people's current interaction habits and thus lead to better user experiences.

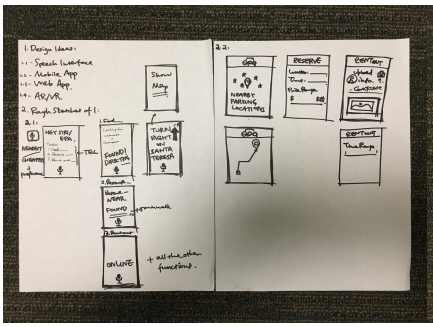


Fig.7 Concept sketches

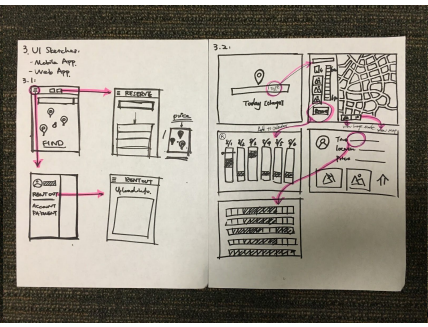


Fig.8 UI sketches

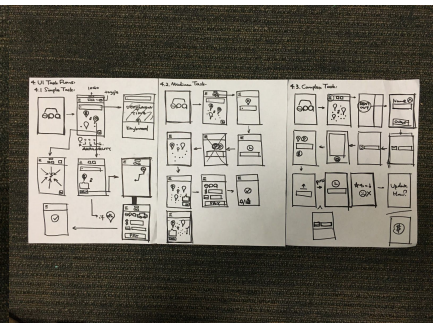


Fig.9 Storyboarding

4. We then started to build our low-fi prototype (Fig.10) and tested it with multiple users. Based on users' feedback, we realized that the first two tasks could be largely combined to make it more intuitive for the users while the third task should support more detailed features. The missing relevant guiding information, conflict between intuitions of "find" and "reserve" features and the complicated reservation process were also taken into consideration in the next phase.

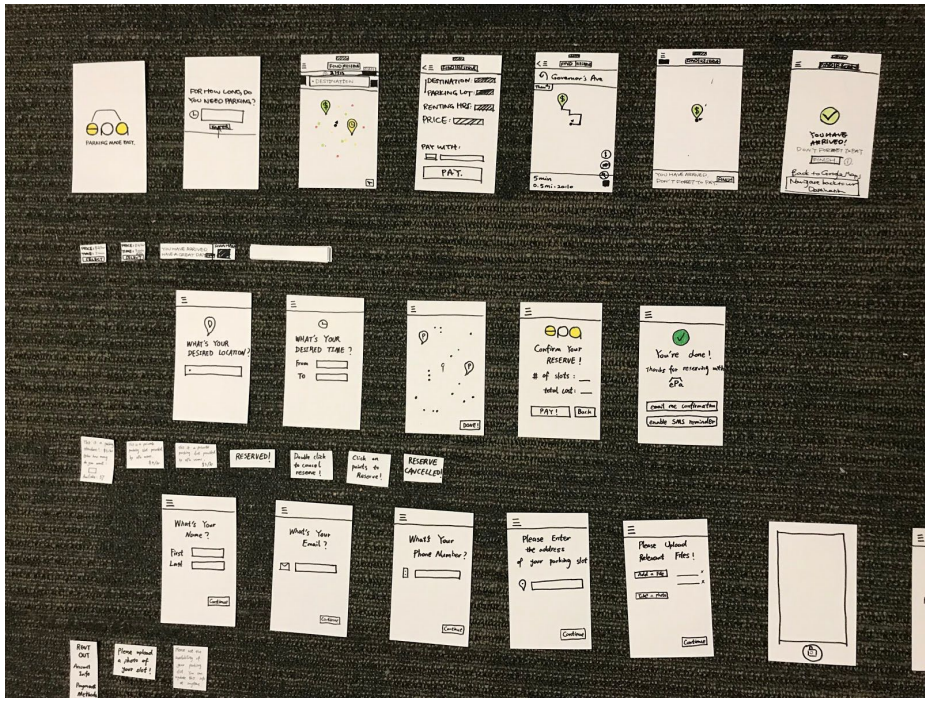


Fig.10 Low-fi prototype

5. After low-fi prototype, we started building interactive medium-fi prototype using Sketch and InVision (Fig.11). From further feedback gathered during low-fi prototype testing, we implemented three design changes. For one, on our home screen, all spots were of same color but varied in size to indicate system recommendation. User could also customize whether they prioritize cost, distance or availability. This is because multiple colors / sizes / symbols could be confusing. During our testing, no tester understood the meaning of these designs. One of them even thought that dollar symbol indicates an expensive spot. In addition, aesthetically, having different colors, symbols and sizes on a single screen was very messy. Furthermore, we decided to separate public spots from ePa spots. ePa spots would appear semi-transparent on the screen. If users failed to click on any of them for a period of time, or click on semi-transparent ones, we would direct them to the reserve screen (medium task) to make a reservation on ePa spots. This helps make the simple task extremely simple and further clarify the task flows. Last but not least, for the complex task, we decided to estimate the rent price for this parking spot based on the location, in terms of dollars per hour before the request is sent. This is because, based on our low-fi prototype testing, users are curious about how much they can earn and their decisions may be affected by this information.

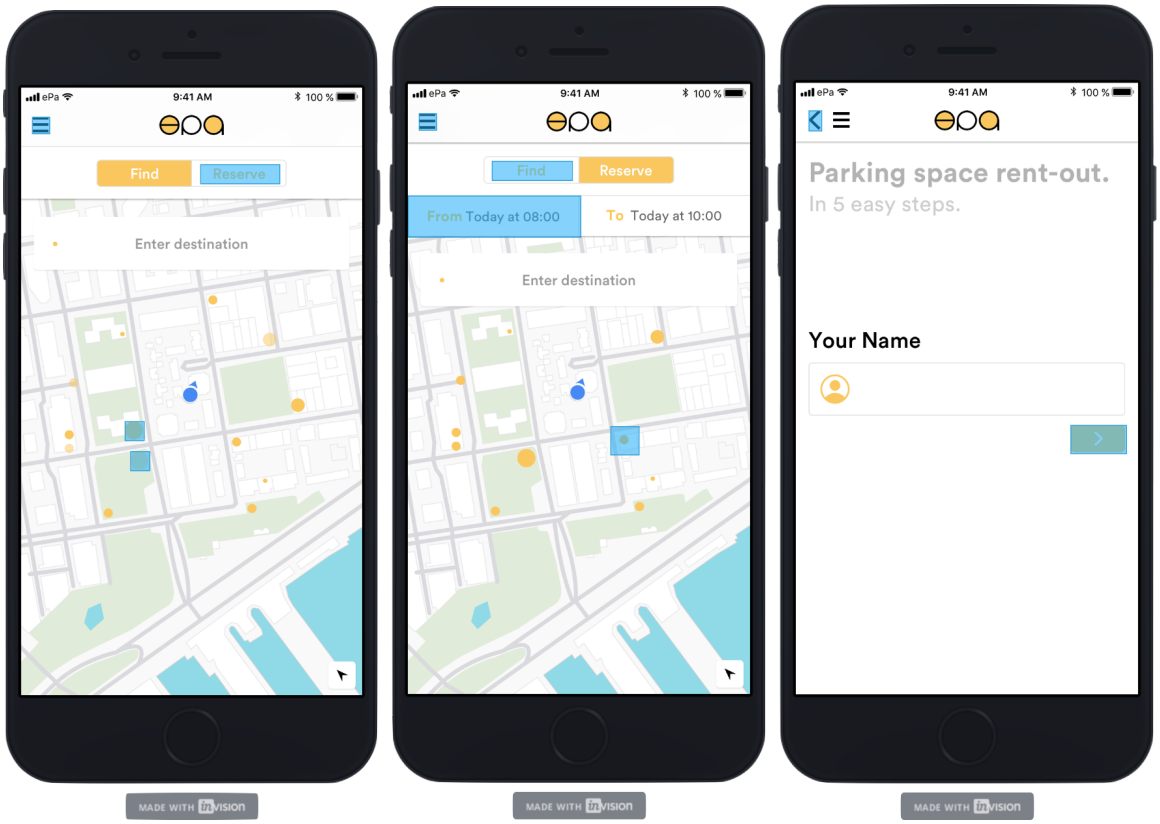


Fig.11 Medium-fi prototype

6. We then moved onto our high-fi prototype where further changes due to Heuristic Evaluation are discussed in the section below.

Major Usability Problems Addressed

- **Usability Problem 1**

  H5. Error Prevention, H8. Aesthetic & Minimalist design
  Severity 4, Found by three people
  [Problem]
  While using opacity alone to differentiate between regular parking spots and ePa spots is minimalist, the opacity is difficult to detect by users who are not as color-sensitive. The user more prone to making an error on the app because they will most likely not notice the difference between the two.
  [Solution]
  We differentiated regular parking spot and ePa spot using different logo styles (with or without the letter “e”) rather than using opacity. This helps user better distinguish between the two (Fig.12,13).

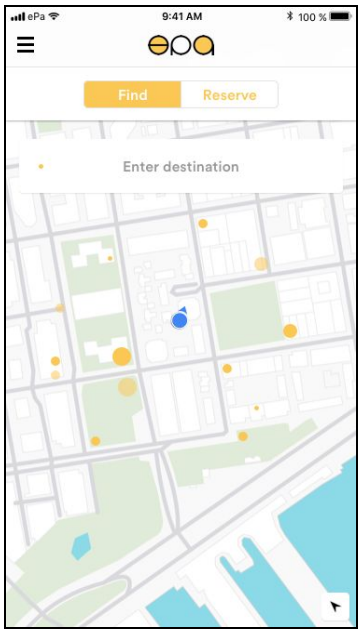


Fig.12 Before

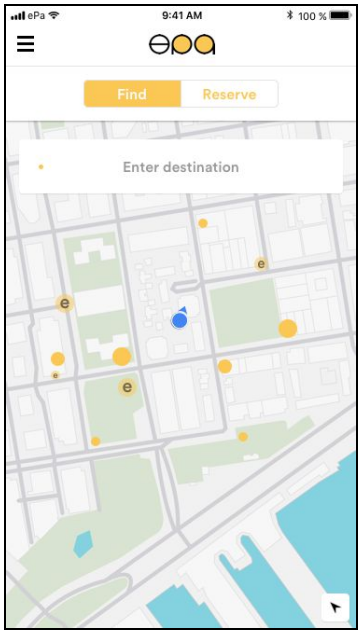


Fig.13 After

- **Usability Problem 2**

  H7. Flexibility and efficiency of use
  Severity 3, Found by two people
  [Problem]
  When a user is renting out a spot, there are a lot of pages with just one input per page. As a result, they have to click on next after the addition of a single piece of info. So when a user decides to change username after they’ve entered their location, they are forced to click the back button three times -- definitely not fun!
  [Solution]

We decided to keep the design of one piece of info per screen but made information entered on previous screens available for editing on subsequent screens. This helps user stay in the process tunnel while provides the flexibility required (Fig.14,15).

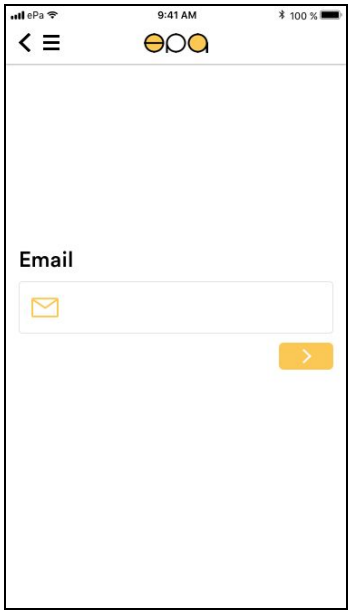


Fig.14 Before

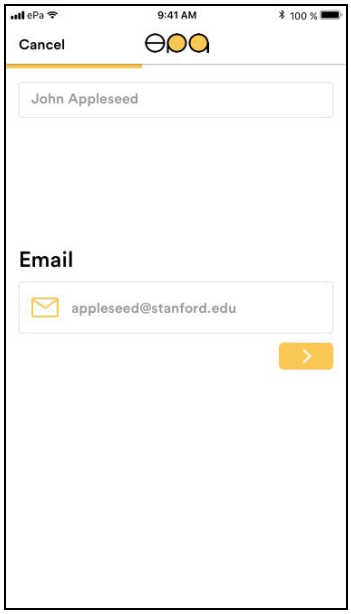


Fig.15 After

- **Usability Problem 3**

  H7. Flexibility and efficiency of use

  Severity 3, Found by two people

  [Problem]

  The use of the color green to highlight that a spot is reserved can be problematic for people with color vision deficiency. Also, some of the dots are too small and are very hard to spot in both the “Find” and “Reserve” screens.

  [Solution]

  In addition to the green dot to highlight that a spot is reserved, we used a simple, white-in-color check mark to help people with color vision deficiency to see better. We also abandoned using 5*5 pt circles to make it easier for our users to more accurately tap on the desired spot (Fig.16,17)(Images on the next page).

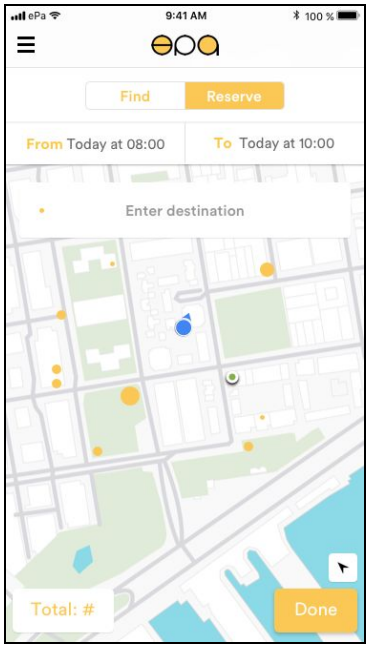


Fig.16 Before

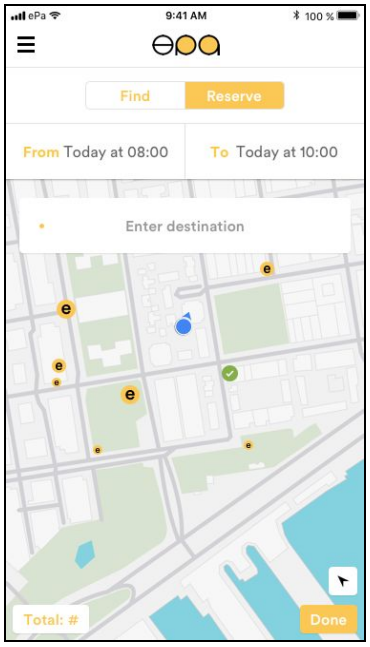


Fig.17 After

- **Usability Problem 4**

  H10. Help and Documentation

  Severity 3, Found by one person

  [Problem]

  When renting out, I had no idea what a “spot certificate” is. As a new user, this is the point I’m most likely to give up on the registration.

  [Solution]

  We added an Info icon, which you can click on and see the detailed explanation on “spot certificate” (Fig.18,19).

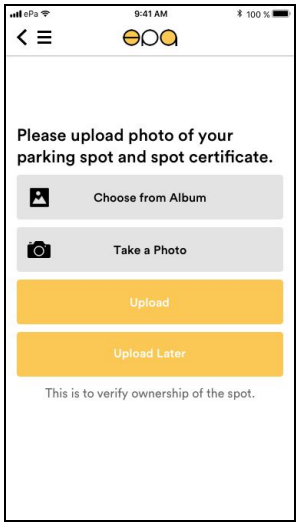


Fig.18 Before

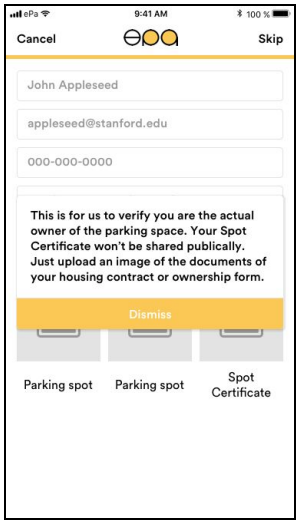


Fig.19 After

- **Usability Problem 5**
  H3. User control and freedom, H7. Flexibility and efficiency of use
  Severity 3, Found by two people
  [Problem]
  When users have found a spot and chose to stop navigation, a screen “You’ve arrived” pops up. In the screen, a button at the bottom reads “Navigate to destination”. This is both unnecessary and confusing because the user might have just selected something by accident and wants to pick another spot or might have already arrived.
  [Solution]
  This “Navigate to destination” button actually means “Navigate to the destination you entered, from the current location which is the parking spot”. We rephrase the words on this button. Also, choosing to stop navigation now means “cancel navigation”, rather than arrived; only if you stop navigation after we detected that your arrived, will the “You arrived” screen appear. See Usability Problem 11 for more detail on this issue (Fig.20,21).

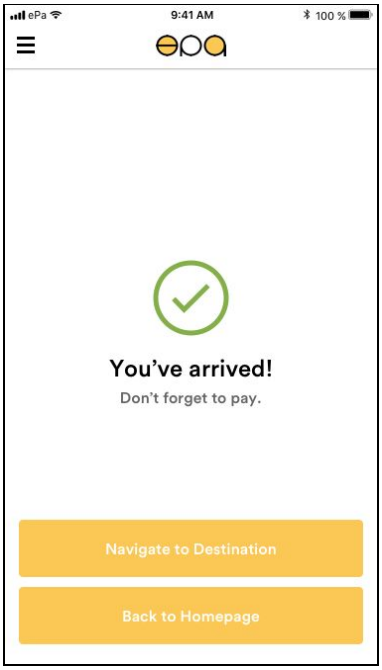


Fig.20 Before

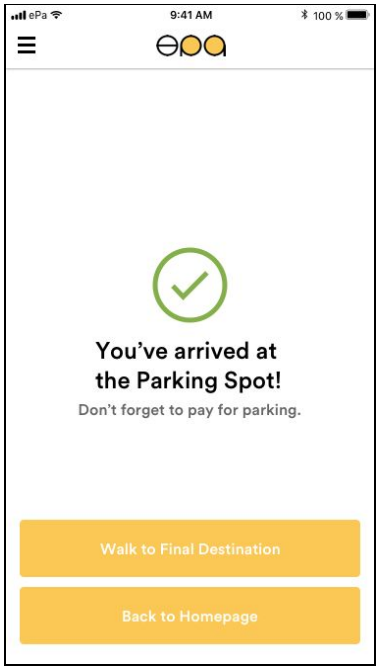


Fig.21 After

- **Usability Problem 6**
  H4. Consistency and Standards
  Severity 3, Found by one person
  [Problem]
  Some pages have a back button on the top left while others don’t. It shifts where the menu button is. It’s unclear whether you are developing for Android or iPhone, but typically menu icons are are the left for Android because those phones already have a back button in place, while iPhone doesn’t.

Also, some screens have Back button on the top left while payment screen has a back button within the page. Pick one method and be consistent about it. (I personally prefer the back button on the page, not the title bar.)
[Solution]
We re-design some of the navigation logic so that exactly one button will appear on the top left of each screen (Fig.22,23). For example, in Rent-out, since all the previously entered information will be displayed and can be edited, there's no use to put a back button, so we put a cancel button (menu button also implies cancel, so we decided to do this explicit cancel and ask user for confirmation); for the confirm payment screen, we do need a back button, and we move it to the top left (Fig.24,25).

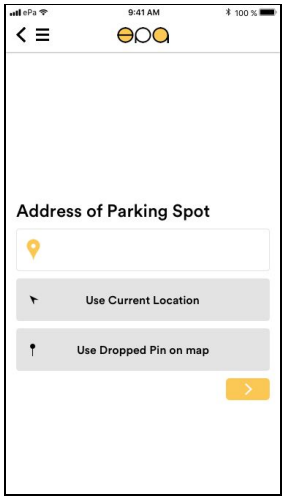


Fig.22 Before

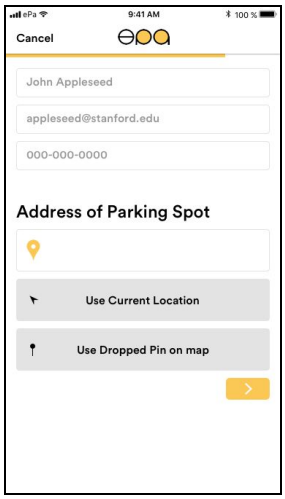


Fig.23 After

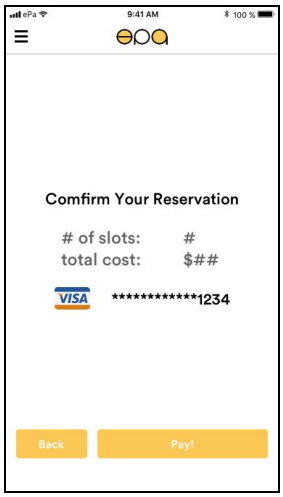


Fig.24 Before

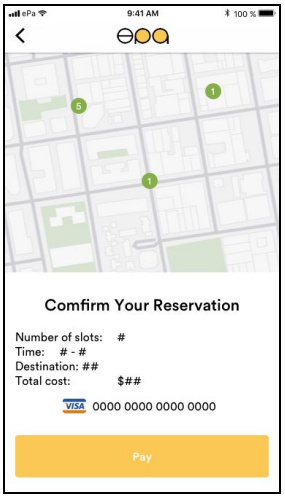


Fig.25 After

- **Usability Problem 7**
  H10. Help and Documentation
  Severity 3, Found by three people
  [Problem]
  The prototype doesn't have any in-app onboarding at the moment or any place to get help or how-to instructions. As a first time user I'm confused about what I should do when I open the app or what the different colors and spot sizes mean. While this might not be a problem for returning users, it might make it harder for new users to get started.
  [Solution]
  We decided to add a short instruction (as a pop-up window) on the first time you see the map view, after the onboarding process. This instruction will explain the symbols used in finding and reserving parking spots, and it also makes it clear that you should click on the dots if you're interested in that spot (Fig.26,27).

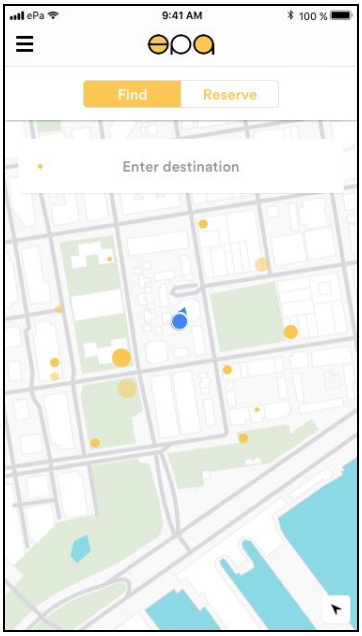


Fig.26 Before

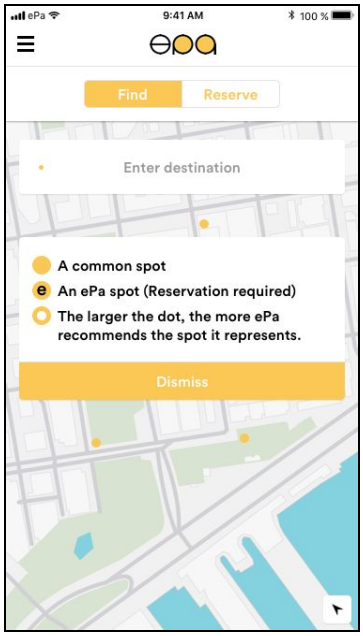


Fig.27 After

- **Usability Problem 8**

  H1. Visibility of system status, H5. error prevention

  Severity 3, Found by one person

  [Problem]

  One problem with the rent-out task is that it does not give the user a summary of what the user have just inputted/submitted in all the previous input fields. This poses a problem as if users want to double check what their input is, they have to go back and check through everything again.

  [Solution]

  The solution implemented to solve problem 2 nicely addressed this problem as well. Therefore, the same before and after screens are used to showcase our solution (Fig.28,29)(Images on the next page).

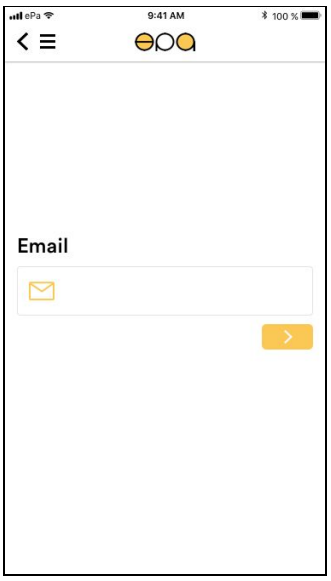


Fig.28 Before

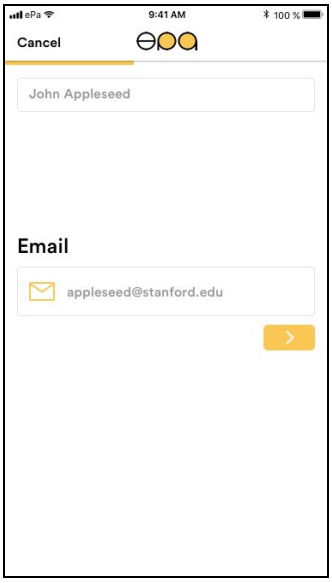


Fig.29 After

- **Usability Problem 9**
  H1. Visibility of system status
  Severity 3, Found by one person
  [Problem]
  The user should be able to see all possible parking spots, including the ones which have been taken so that they know where they could park in the future, or where to reserve in advance, and the colors for such spots should be greyed out.
  [Solution]
  After some discussion, we decided not to implement changes suggested by this feedback. The main reason is, this will make our UI very messy - consider somewhere where you find it hard to park, say at Downtown SF or Palo Alto, there are actually lots of parking spots which can fill up the map view, but most of them are not available. It will just bring trouble to users.

- **Usability Problem 10**
  H9. Help users recognize, diagnose, and recover from errors
  Severity 3, Found by one person
  [Problem]
  Whenever I add multiple spots for the sake of reservation, there is no easy and intuitive way to change the number of spots from something like 3 to 2.
  [Solution]
  We agree that there's no one-step method to make such change. However, to cancel this spot and re-click on it won't take too long. We didn't design a one-step method because we don't expect this to be a common situation - according to our interviewees, most users will only reserve for his/her own car. However, inspired by this advice, we did add the support of editing the reservation even after you've already made a payment

(you may need to pay more or may get refund after you edit your reservation). Here's the med-fi design of this feature (Fig.30).

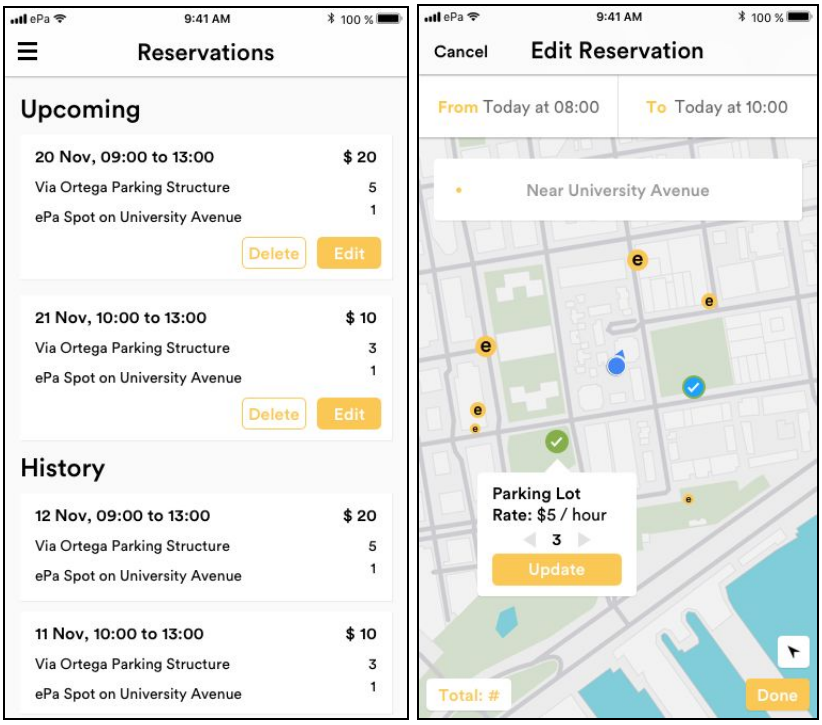


Fig.30 Edit Reservation

Since this is not part of our original three tasks, we didn't implement this feature in high-fi prototype.

- **Usability Problem 11**

  H4. Consistency and standards

  Severity 3, Found by two people

  [Problem]

  After selecting a parking spot in find mode and getting directions, Clicking the "X" shows a new screen telling the user they've arrived. This is confusing since most user's expectations are that the "X" means cancel the current route.

  [Solution]

  We agree that this is confusing. To make it more intuitive, in our revised design, clicking the "X" will actually cancel the navigation (we'll pop-up a confirm window before actually canceling the navigation) (Fig.31,32). On the other hand, we will use GPS to auto-detect whether user arrives at their destination; if arrived, we will display "You're arrived" on the bottom bar, and if user click on the "X" for now, it means confirm the arrival and the arrival screen will appear (Fig.33).

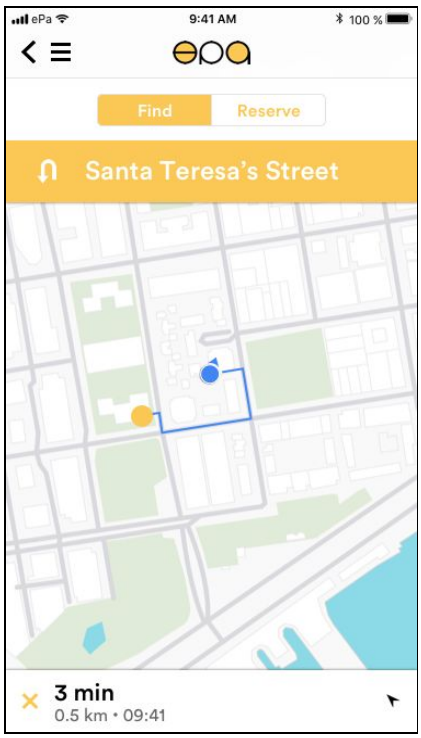


Fig.31 Before

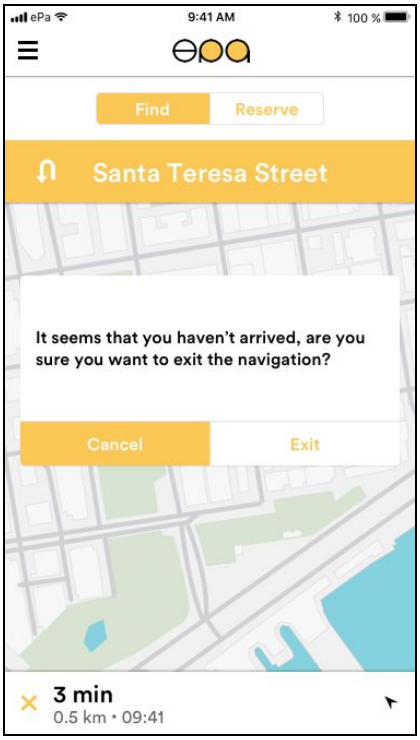
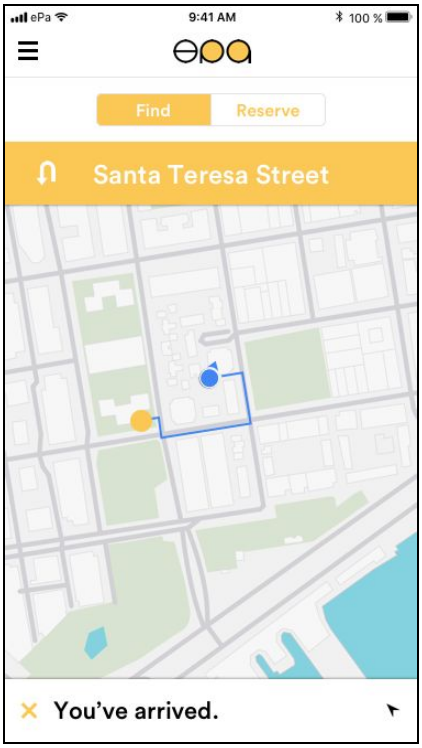


Fig.32 After



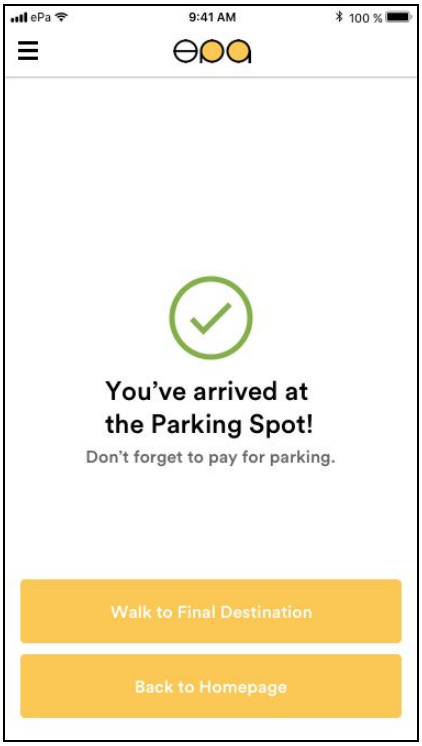


Fig.33 Arrived screen

Prototype Implementation

- **Tools used for the hi-fi prototype**
  - **Sketch**
    We built our medium-fi prototype on sketch. And based on the feedback from heuristic evaluation, we made several design changes before our implementation. Sketch is very helpful to high-fi prototype implementation from three perspectives: first, it clearly states the logic of screen navigation, which is very important to our thinking about how to design the code structure; second, it provides a detailed layout for each screen, and most of the parameters (e.g. location of a specific component) can be directly used in the code; finally, all the components can be exported and inserted into our high-fi prototype code easily, so we don't need to use code to generate many views - we can export those view as images and import them, as long as there are not too much interaction logic behind. One such example is the 'Successfully Submitted' screen, where the content is always the same regardless of the input.
    The main limitation of sketch is that, it cannot define complicated logic within different screens. These logics can only be represented with multiple screens which required some re-thinking during coding.
  - **React-native on Expo, Snack.expo.io**
    We wrote all the code using Expo, which is basically a wrapper of react-native. We use snack.expo.io for development, which is an online version of Expo. The main advantage for Expo, and its online version, is its fast development period. Whenever we make any small code changes, it can be reflected on our mobile phone immediately, which is important given that we focus on design details. It is also very flexible; there are no environment requirements, no external package dependencies, and no build process.
    The main limitation of Expo is that, it doesn't support exporting as a standalone app very well, so our final submission still relies on Expo app itself. It's also a little buggy in terms of caching/refreshing - sometimes we do need to force quit and force refresh in order to get things work.
- **Wizard of Oz Techniques**
  - We assume we have access to real-time parking spots information.
  - We give spots recommendations to the user without actually calculating what is the best spots.
  - Estimation for the price during rent-out is not actually calculated.
  - We don't actually conduct navigation; rather, we display the route as pictures.
- **Hard-coded Data**
  - The Map view is hard-coded, as well as our current location.
  - Time for reservation is hard-coded, as 8am - 10am today.
  - During rent-out, the current location is hard-coded.
- **Unimplemented features**
  - Dropped-pin is not supported when picking rent-out location.

- “Walk to final destination” after navigation to the spot is not supported.
- All features in pull-out menu, except rent-out, is not supported.

Summary

We started this quarter brainstorming potential topics for our project. After narrowing down the wide range of ideas we came up with and eventually deciding to focus on the transportation issue, we began to conduct user interviews and found that parking had long been a main pain point for many users we interviewed. That was the beginning of ePa. ePa is where parking is made easy. We aim to do so by creating new parking spaces, maximizing the usage of currently available parking spots and conveying these info to our users. Throughout the quarter, we continuously iterated on and improved our app based on user feedback and novel insights. We learned conducting in-person user interviews, brainstorming, prototyping in both low and high fidelities and collaborating as a team. Moving forward, our goal is to continue apply the process and principles we learned in this course to future ideas and projects.