CS 147 Assignment 8
Local Community Studio

Wanderlust

Kye Kim - Visual Designer, Developer
KiJung Park - UX Designer, Developer
Julia Truitt - Developer, Designer

Value Proposition:

Explore More, Worry Less

Problem and Solution Overview

People need a way to find interesting places, avoid unsafe areas, and be immersed in their current environment while navigating. Because traditional maps only suggest the fastest routes, people with the aforementioned desires cannot fulfill their needs. By providing users with high-level information about their surroundings and letting them set their navigation boundary, our app aims to facilitate immersion in the environment and worry-free exploration.

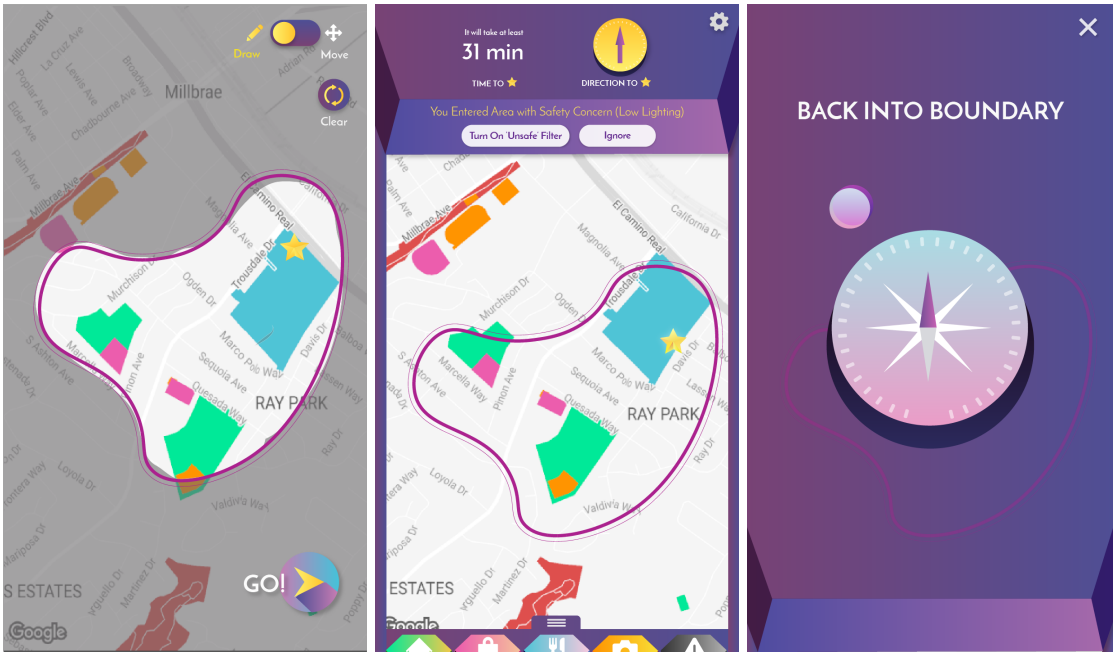

Image 1: Screens of the main three solutions: Be Safe, Be There, Be In the Moment

Tasks & Final Interface Scenarios

Simple: Set a Destination

The user presses the "Set Destination" star on the app's landing page, which then transports the user to the search page. Pressing inside the search box triggers a keyboard to pop up, allowing the user to type in a destination. The user can then select the correct destination from the list of filtered locations resulting from their search. This task addresses the first of Wanderlust's three mission statements: "Be There."

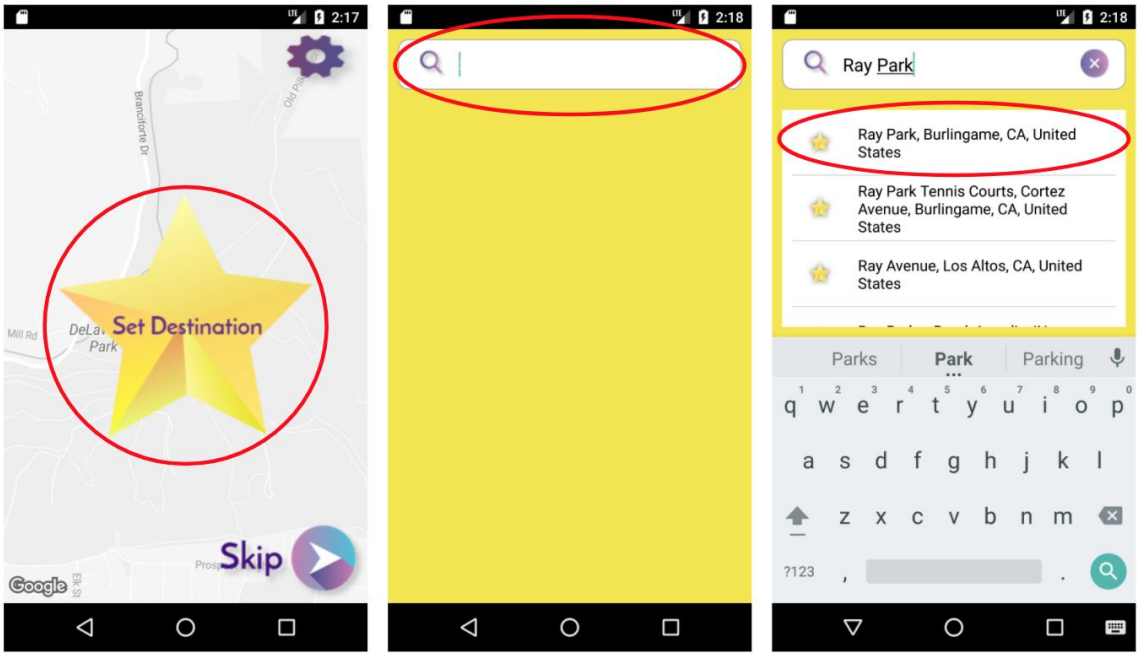

Image 2: Set Destination task

Moderate: Draw a Wander Boundary

After selecting their desired location in the "Set a Destination" task, the app transitions to a page displaying a map with a star marking the user's destination and five filters (Residences, Shops, Foods, Attractions, and Unsafe). The user can toggle any of these filters on or off, depending on what information they are interested in seeing around their intended area of exploration. After filtering their map, the user presses circular arrow button (the next button) in the lower-right corner of the page which takes them to the Draw Page. In the upper-right corner of this page, there is a switch that the user can toggle on to enter "Draw Mode" or off to enter "Move Mode," where the user can drag their finger on the screen to change the focus point of the map. In "Draw Mode," the user can use their finger to draw a boundary that includes all the areas

they want to explore and excludes areas they want to avoid (e.g. unsafe areas). The filters set on the previous page inform the user's decision of how to draw their boundary. If the user is unsatisfied with the boundary they drew, they can press "Clear" to start over. When they are finished drawing their boundary, pressing "Go" will take them to the navigation page (once the user's current location is inside the wander boundary, which we simulate for usability testing). This task addresses two of Wanderlust's mission statements ("Be Safe" and "Be Immersed"). The wander boundary allows the user to explore their area within a predefined safe zone, giving them the comfort and confidence to be engaged in their environment rather than looking at their phone screen the whole time.

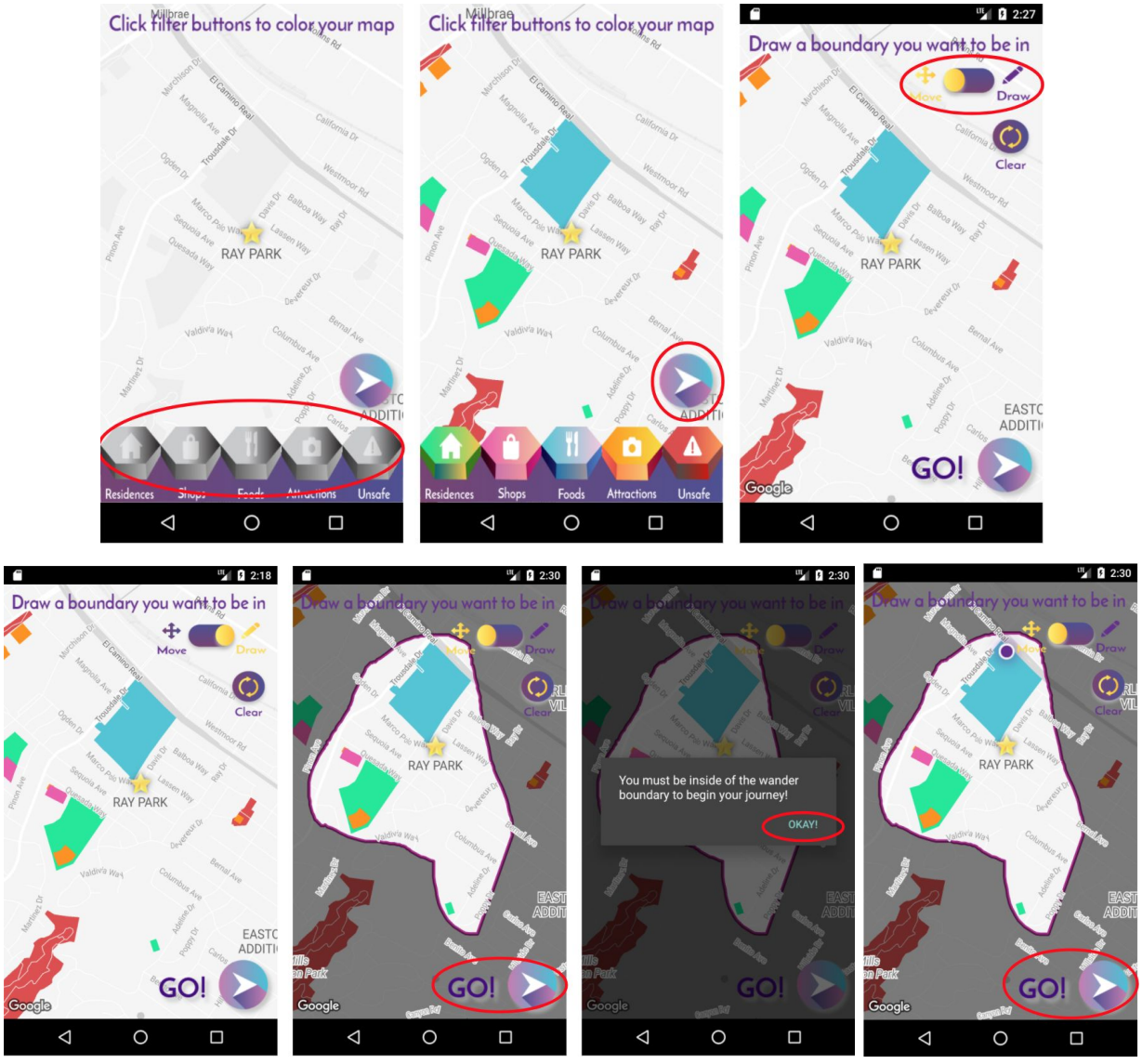

Image 3: Draw a Wander Boundary task

Complex: Navigate to a Desired Area

The majority of the Navigation Page page shows a map with a star representing the user's destination and a purple dot showing the user's current location. The navigation deck at the top of the page includes the minimum time to their destination from their current location, as well as a rough direction they should travel in to reach their destination. If the user travels outside their wander boundary, the app will alarm, and upon pressing the "Guide Me Back" button that appears, the user is taken to a page that will guide them back inside their wander boundary. The navigation page will also notify the user if they have wandered into an unsafe area or if they have reached their destination. Advanced users can also toggle on or off the filters they have selected by swiping up at the bottom of their screens to reveal the filter menu. This task also addresses two of Wanderlust's mission statements ("Be Safe" and "Be Immersed") by providing minimalistic guidance on the user's journey.

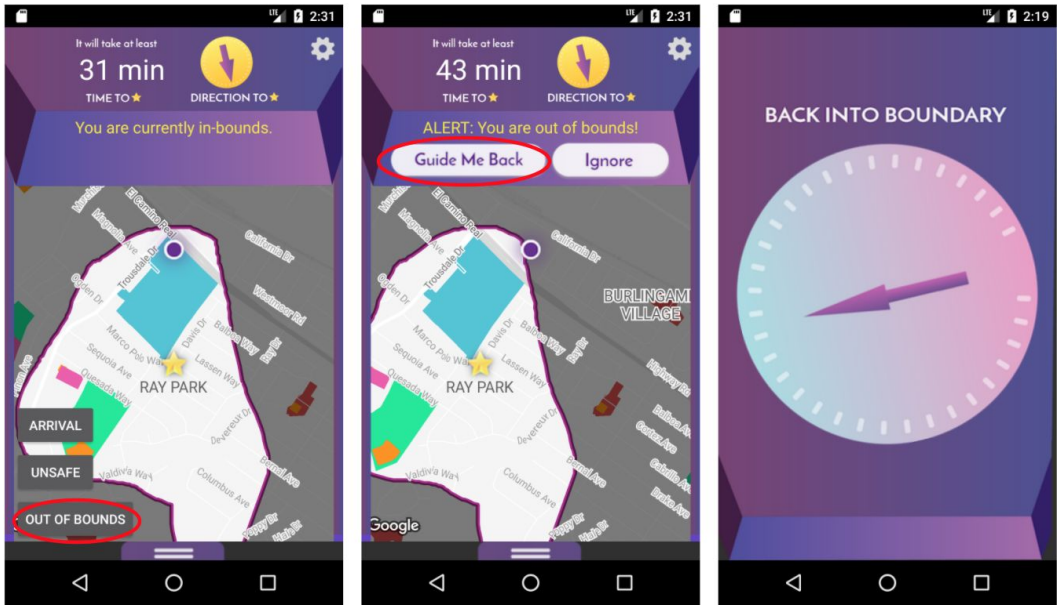

Image 4: Navigate to a Desired Area

Design Evolution

1. Initial Sketch

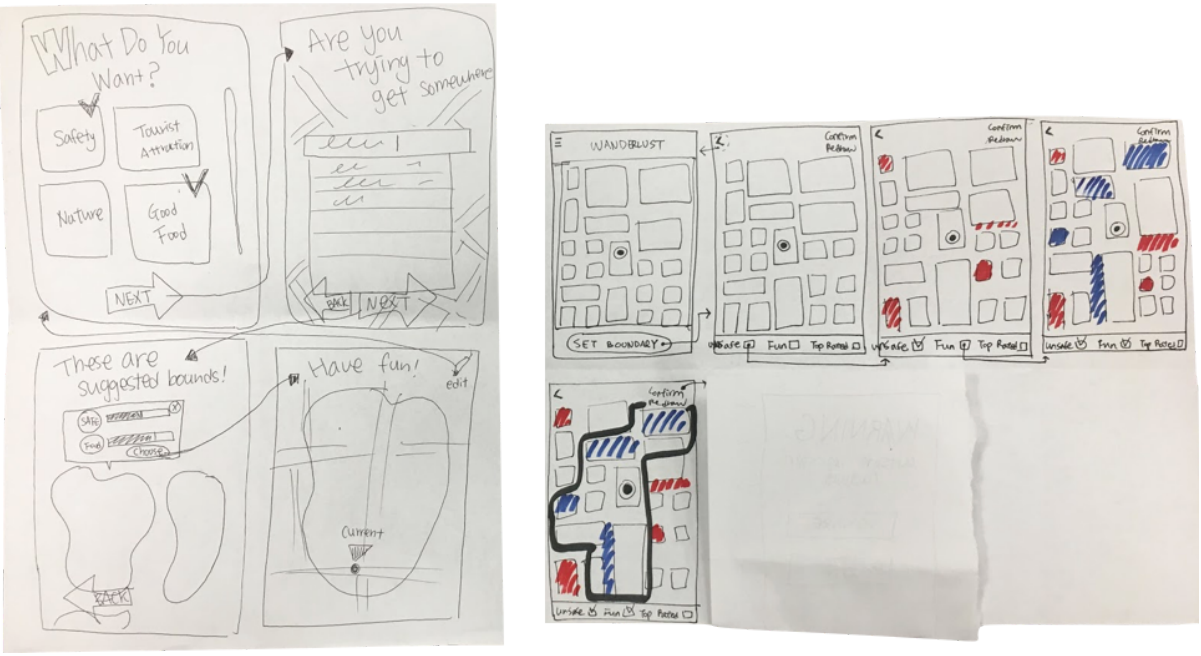

Image 5: Initial Sketches of two interface design approaches

We explored two possible interface designs through initial sketches. One [left] was where user answers a series of questions that ask preferences for the trip and selects one of predefined boundaries. The other [right] was where user references color coded filters containing information about the neighborhood, and hand-draws boundary by themselves. We decided the latter one since it is more flexible in accommodating unique navigation needs and desires of each user that may not be captured by predefined choices.

2. Low-Fi Prototype

Image 6: Low-Fi Prototype

We tested our low-fi prototype with participants, including students and visitors to the university, who performed three main tasks on the paper screens. One major finding was that participants did not see the point in having to click extra buttons to turn on color filters when exploring. In response, we made the filters turn on as a default when user starts exploring. Another major finding was that some users were misinterpreting the term “wander boundary,” drawing boundaries around areas they did *not* want to go to, rather than areas they *did* want to go to. As a result, we decided to put a dark overlay over the areas outside the user’s wander boundary to make it clear that they will be exploring *inside* the boundary. We also made it impossible for user’s to draw more than one boundary.

3. Medium-Fi Prototype

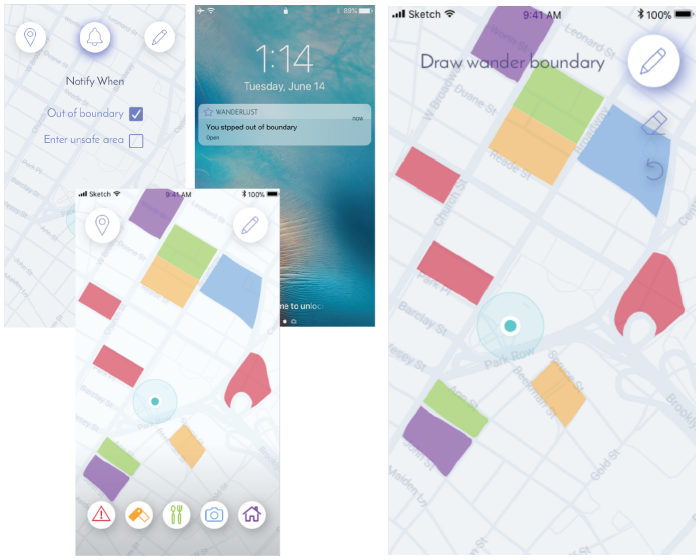

Image 7: Medium-Fi Prototype

We received a heuristic evaluation report from other students in our studio (this is discussed in more detail in HE section below).

4. High-Fi Prototype

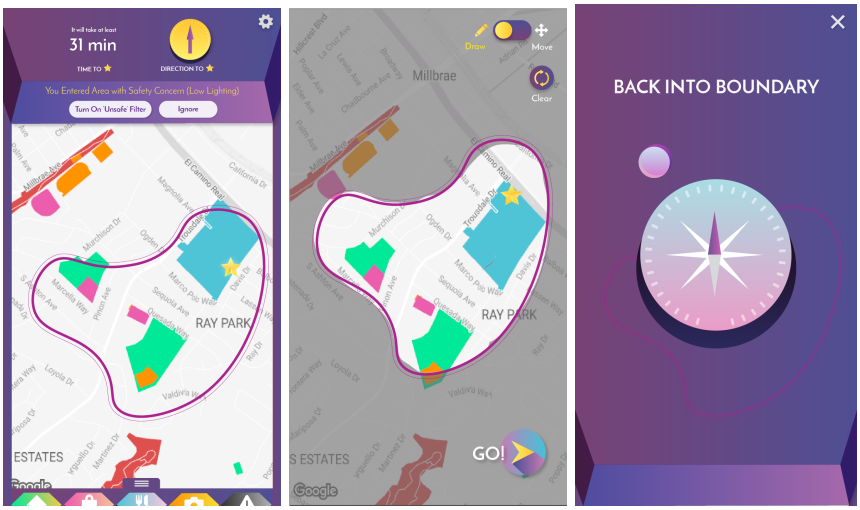

Image 8: High-Fi Prototype

Major Usability Problems Addressed

1. **[H1: Visibility of System Status]** *Not sure how the user knows they got to their destination.*

In response, we decided to notify the user when they reach their destination, with the action option of ending the trip. In this way, the user does not have to keep checking the app to see whether they have arrived at their destination.

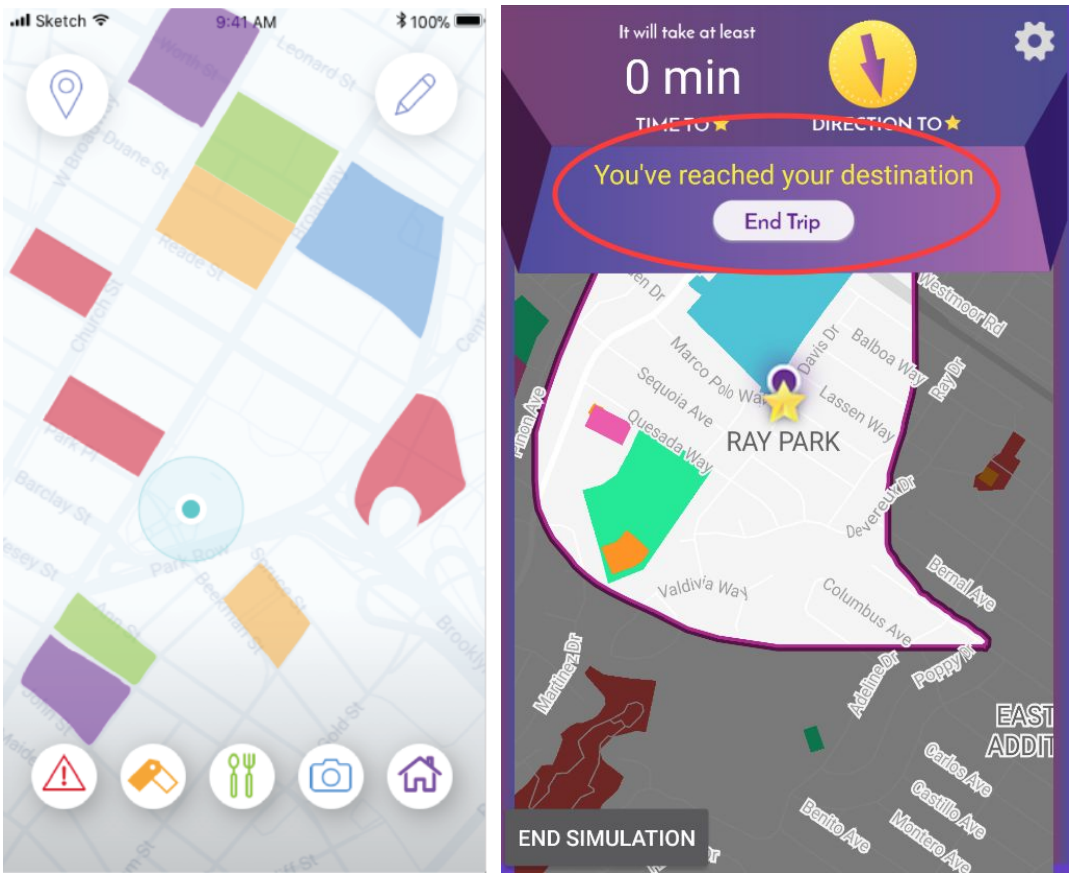

Image 9: Before (left) and after (right) screens of the navigation page. You can see the arrival notification in the red circle of the right page.

2. **[H2 Match Between System & Real World]** *Icons used for selecting area filters are not super clear.*
Having no clear description of what the icon for each area filter represented confused users. In response, we put the name of each filter under the filter icons.

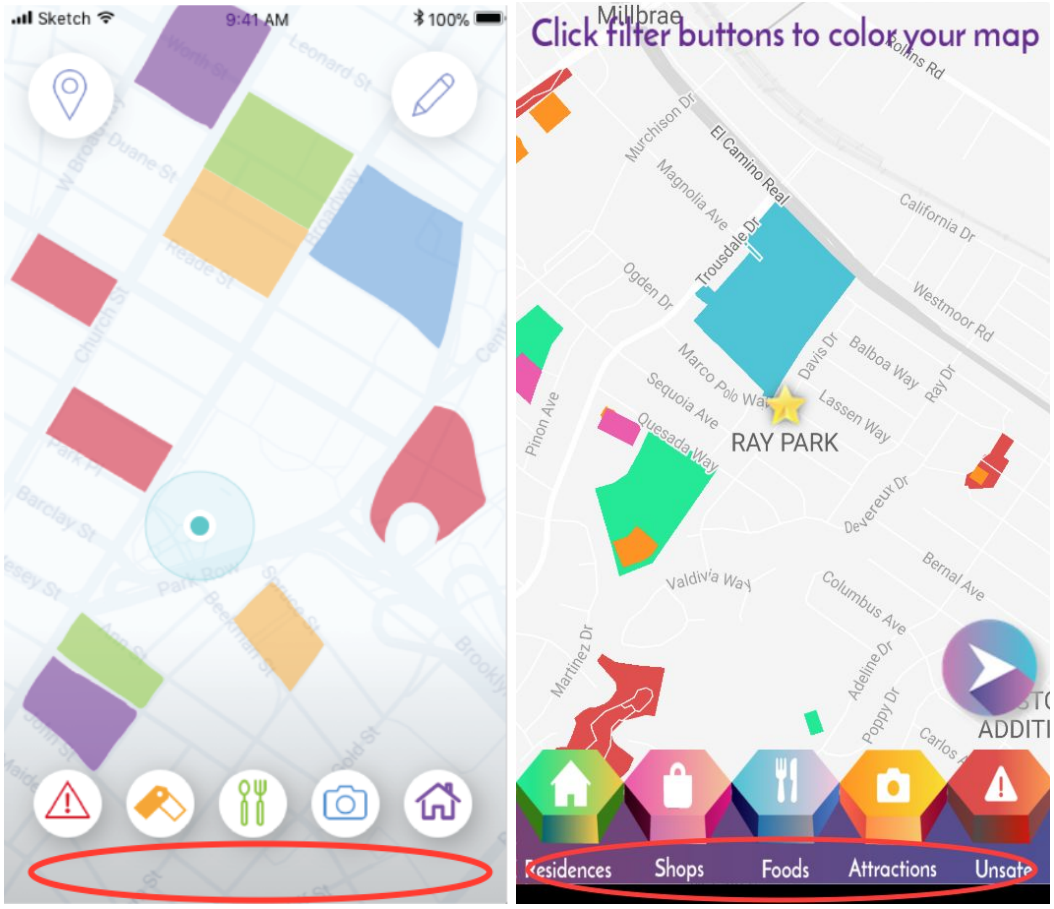

Image 10: Before (left) and after (right) screens of the filter selection page. You can see the names below each icon in the red circle of the right image.

3. **[H3 User control and freedom]** *Once I select a destination, there is no way to deselect it.*

We put an (x) icon in the location search bar to allow the user to re-start their search and enter a new destination. Moreover, after selecting the destination and transitioning to the filter selection page, the user can always use the native Android back button to return to the destination selection page and change their destination, or to go back to the main page and skip destination selection altogether.

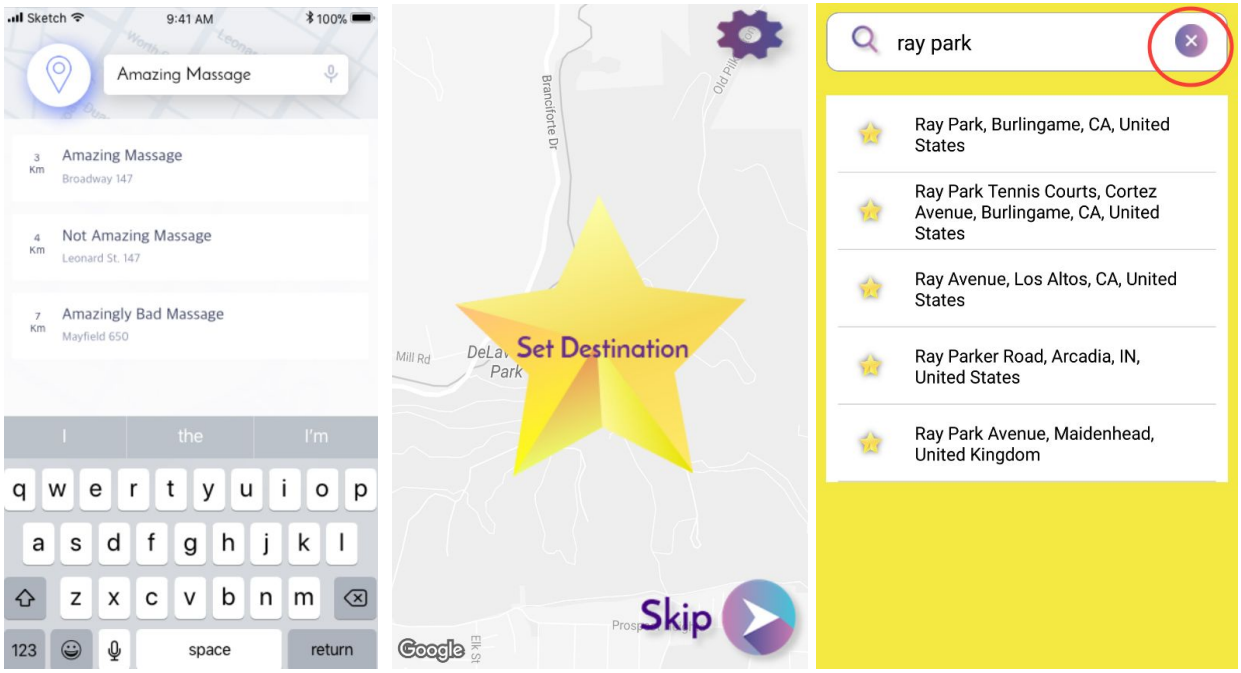

Image 11: Before (left) and after (middle and right) screens of destination selection page.

4. **[H4 Consistency and Standards]** *I wonder in what setting would one need an eraser button and an undo button (when drawing a boundary).*

Since the boundary drawing is relatively simple drawing made in one stroke, we decided that an eraser button would be unnecessary, and thus removed it. Instead, we kept the undo button, but changed its name to "clear" to be more direct.

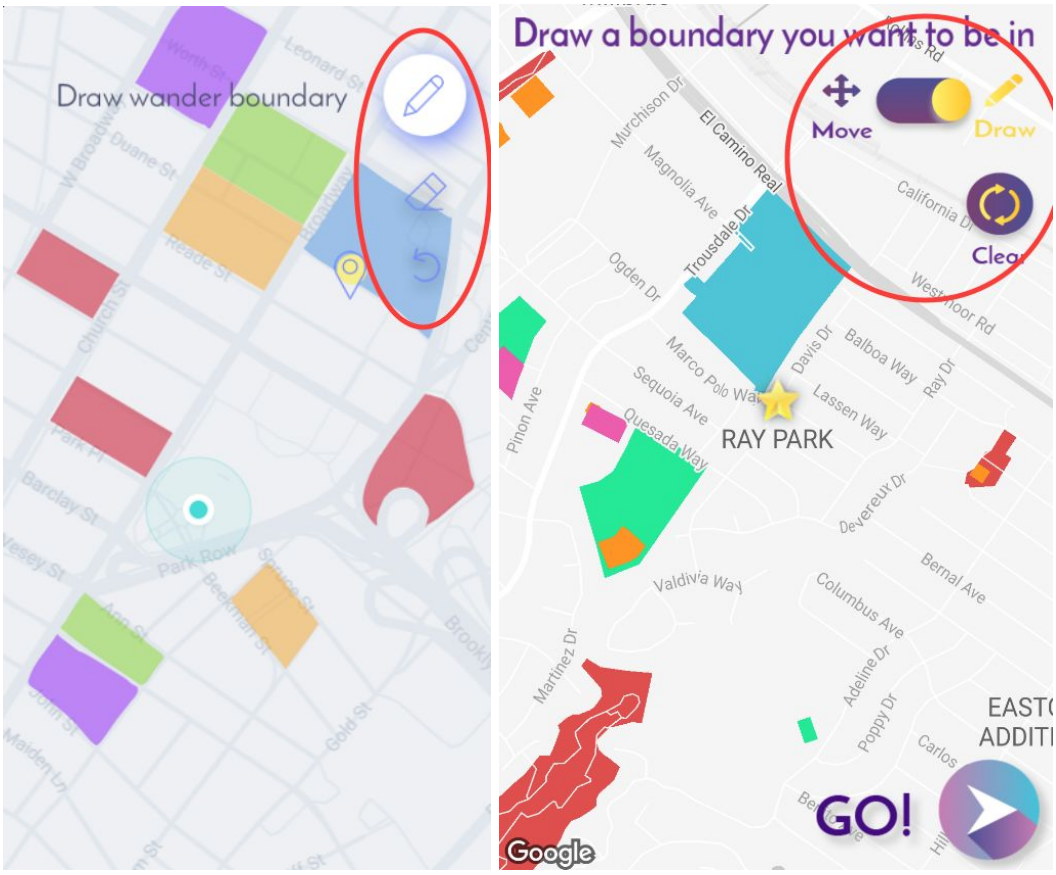

Image 12: Before (left) and after (right) screens of the draw wander boundary page.

5. **[H5 Error Prevention]** *User may attempt to draw a wander boundary that is not a closed loop.*
This was a major issue, as the wander boundary only functions if the boundary is a closed loop. To address this problem, we implemented automatic boundary closing.

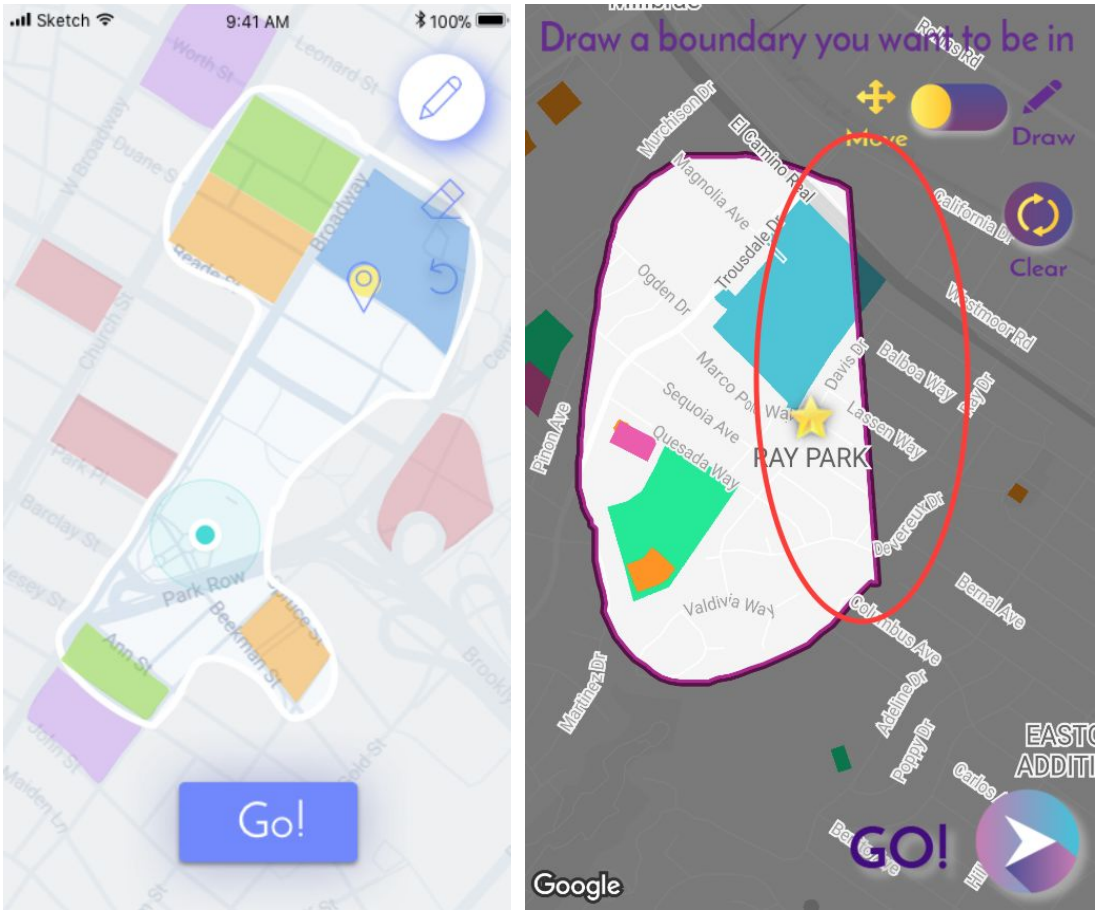

Image 13: Before (left) and after (right) screens of the draw boundary page. Shows how the drawing is automatically closed if the user does not close it themselves (in red circle).

6. **[H6 Recognition rather than recall]** *After selecting destination, all the areas outside wander boundary have lost their* filters.

We realized that users might want to know what area they ended up being in if they happen to go outside their wander boundary. To address this problem, we decided to keep the color coded filters outside the boundary, but have them darken out. We also let the user select or deselect the filters in the navigation page.

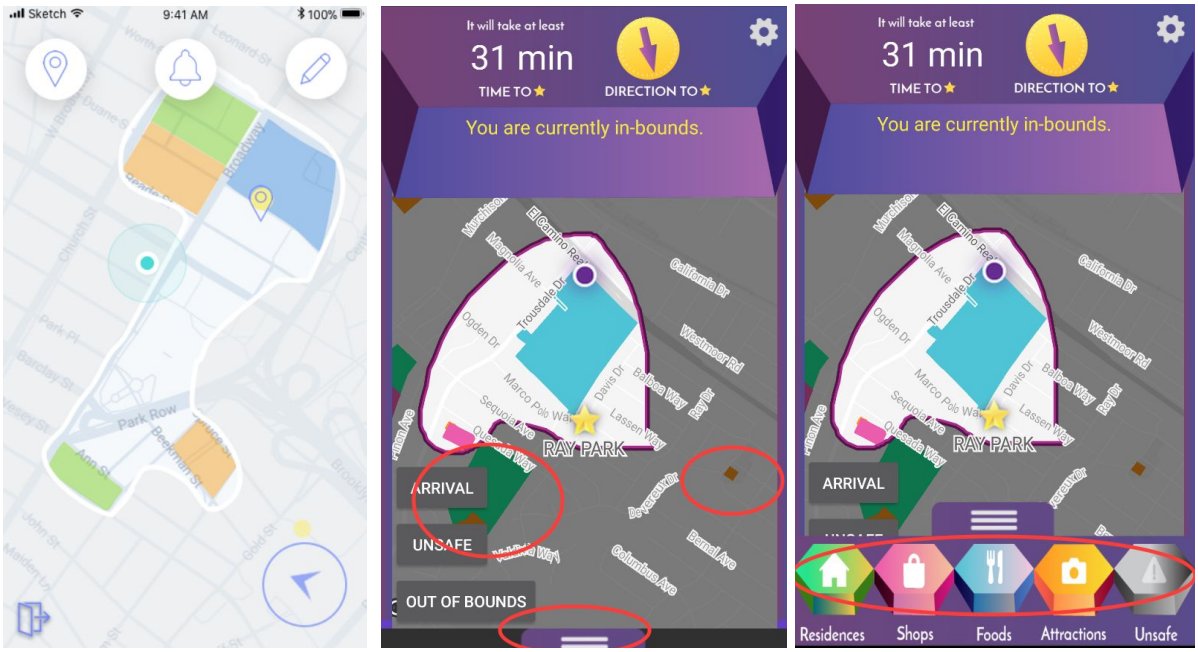

Image 14: Before (left) and after (middle and right) screens of the navigation page. You can see the filters outside the boundary in red circles on the middle page. The filter selection area is shown in the red circle on the right page.

7. **[H7 Flexibility and efficiency of use]** *Have to manually go to the page to set a wander boundary after selecting destination.*

Removing extra, unnecessary actions in the user flow is crucial. In our medium-fi prototype, some users were confused where to start the process, and we also wanted to address the case where the user does not wish to set a destination. To make the flow simpler and more focused for the user, we implemented the set destination and drawing steps on two separate pages in order, rather than having two activity options in one screen.

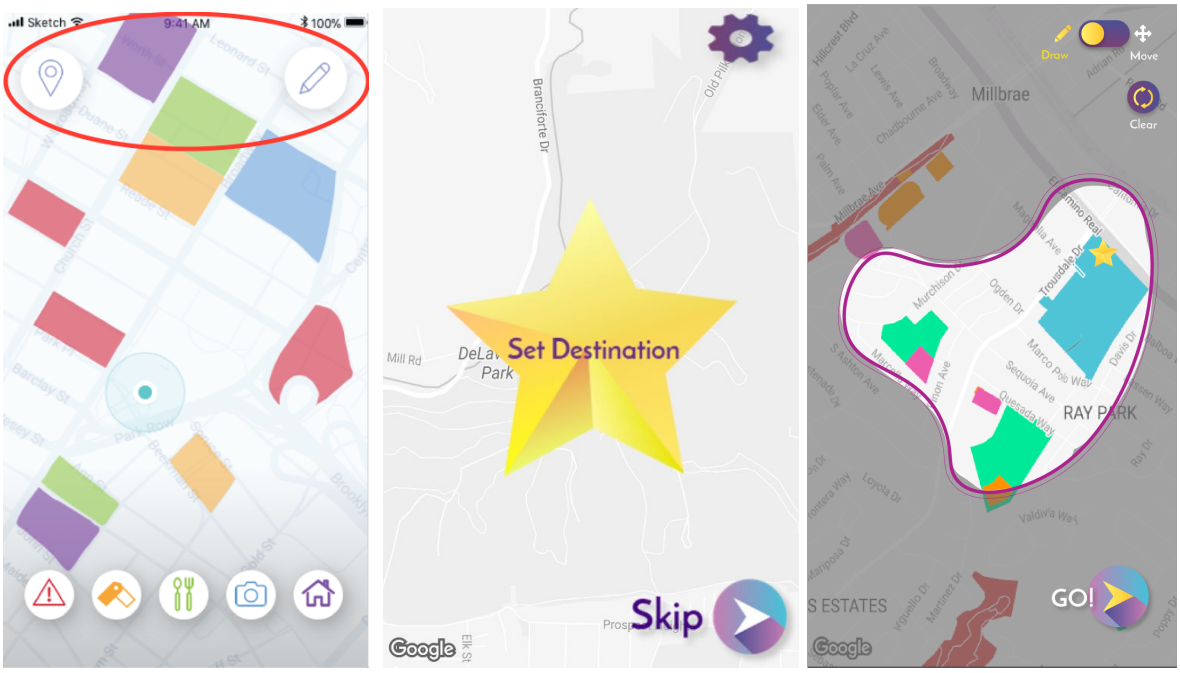

Image 15: Before (left) and after (middle and right) screens of the overall flow of the app. In the left image, you can see the "set destination" button and "draw boundary" button being both on one page in the red circle. In the middle and right pages, you can see these two actions are separated in different pages, in order.

8. **[H9 Help Users Recognize, Diagnose, and Recover from Errors]** *The notification only says what the issue is, does not suggest for how to get back on track or reset the original boundary.*

Our previous notification did not help the user recover from their errors. To address this problem, we added "Guide Me Back" to boundary feature when the user is notified that they are out of bounds. Additionally, the user can edit the boundary by going back to the Draw Page (pressing the native back button in Android).

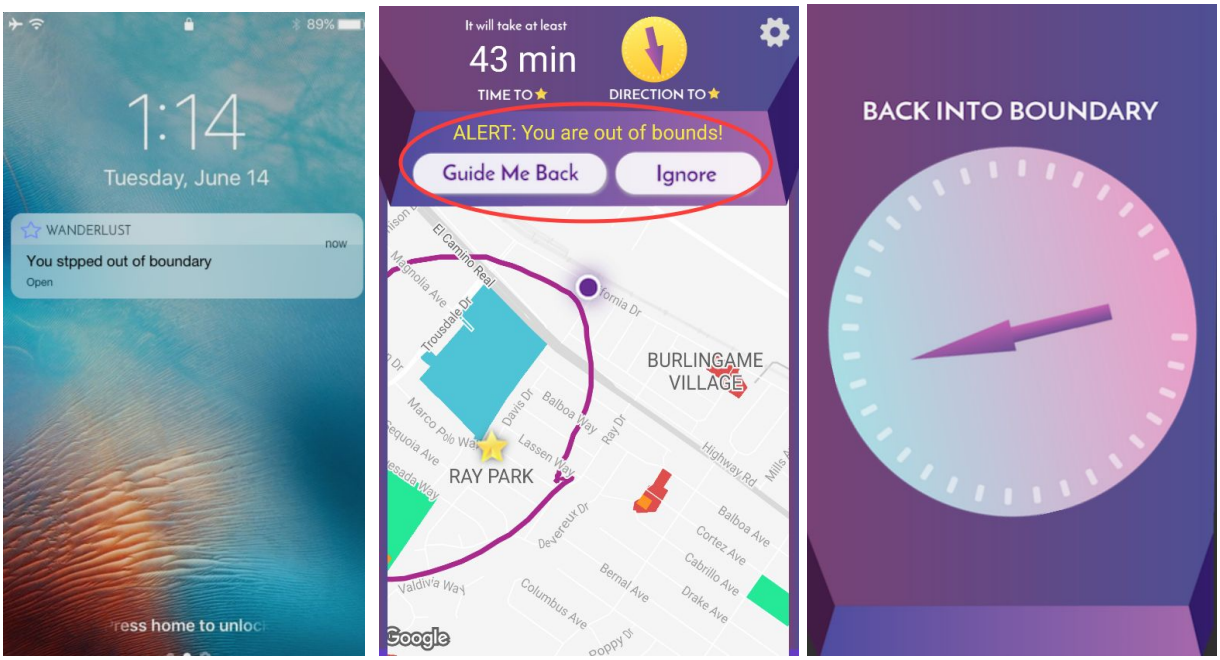

Image 16: Before (left) and after (middle and right) screens of the notification. In the middle image, you can see the guide me back button in the red circle, which if pressed, takes the user to the right screen.

Other Changes:

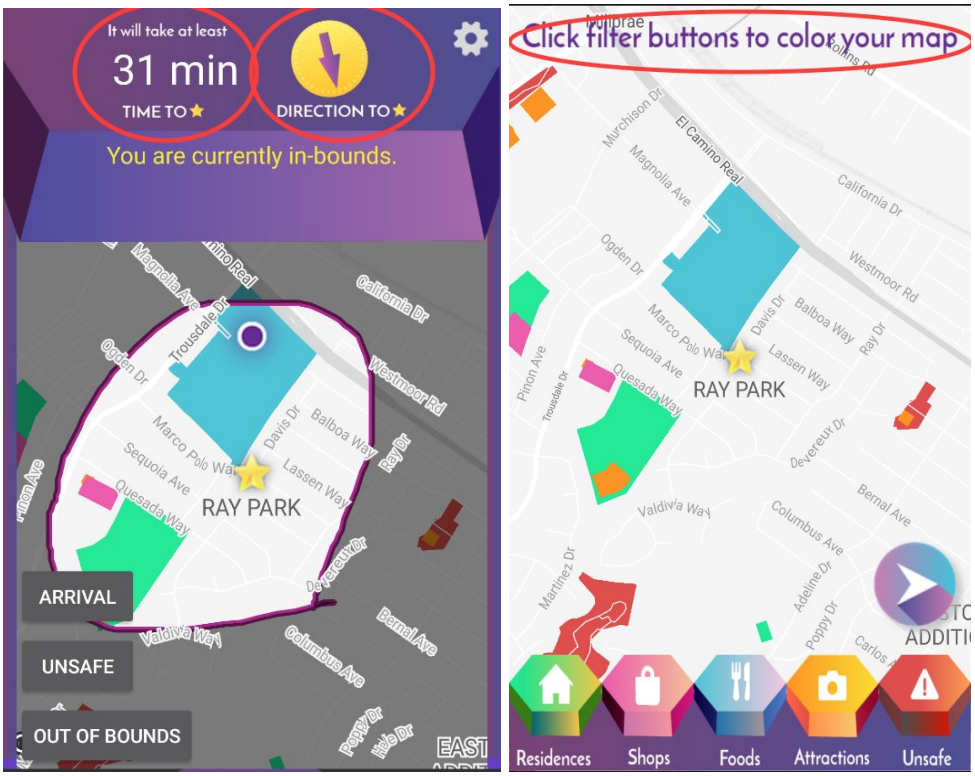

Image 17: Other notable changes in our hi-fi prototype are circled in red.

1. Included minimum time left to get to destination:

We did not want users to worry about making it to their destination on time, so we provided the minimum time to get to their destination from their current location.

2. Included rough direction pointer to destination:

We provided a rough direction that the user can follow to help them reach their destination without confining the experience to an absolute route.

3. Included guide text on top of screen when setting filter and drawing boundary:

For a better help/documentation experience, we made it more clear what the user is supposed to do on each page.

Prototype Implementation

How Tools Helped

We used Android Studio to develop the application and Adobe Illustrator to design the application. Using Android Studio, we were able to overcome many limits of the medium-fidelity prototype. By using Google Maps API, we were able to let users set real destinations and color code the map based on the different combinations of filter selection. Moreover, Google Polygon API let us implement the draw boundary function. Android Studio helped us communicate easily with users by sending them notifications whenever they made an error. Lastly, the native back button on Android devices made giving users flexibility (e.g. changing the destination, modifying filter selection, and redrawing the boundary) easier.

How Tools Did Not Help

One of the biggest challenges of using Android Studio was the difficulty in visually designing the application. Because Android Studio did not let us move the design elements freely in the screen but instead made us define the properties of each element, matching the screen of the prototype to the illustrator mocked-up screen was hard. Another challenge came from drawing the boundary. Although we could recognize some corrupted drawings, such as determining whether the destination and current location is within the boundary, we could not identify illegal boundaries (e.g. Image 18). Lastly, because we were using Google Maps API to set a destination, Internet connection was necessary for the app to function.

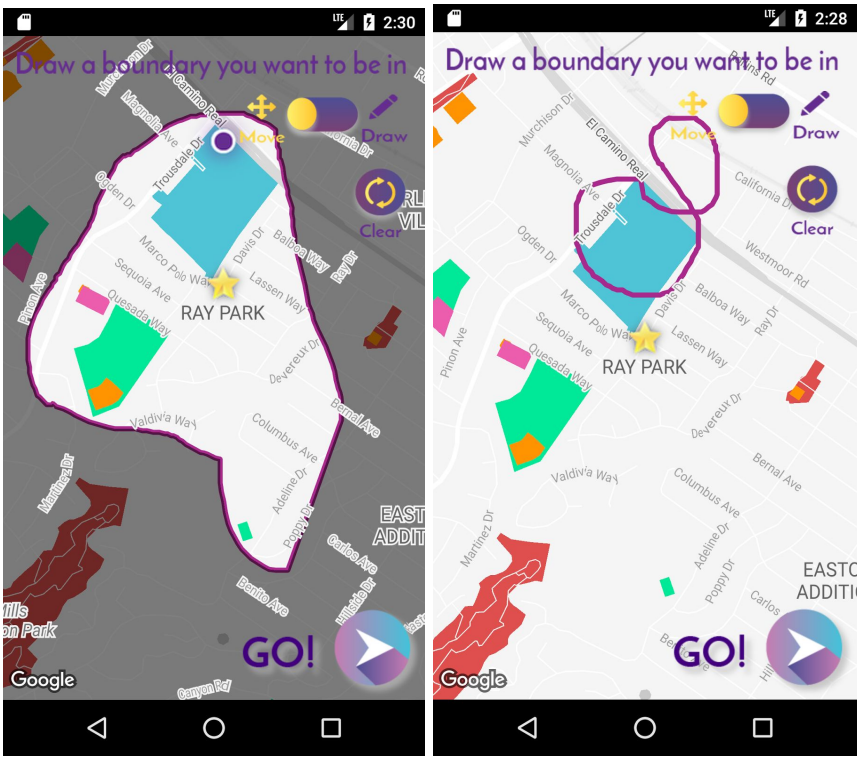

Image 18: Legal Boundary (left) vs. Illegal Boundary (right)

Wizard of OZ

Because our application is a navigation application, many major functions from the navigating page, such as notifying the user when they are out of boundary, in an unsafe area, or arrived at their destination, could not be realized if the user is not actually moving around. To address this problem, we made three simulations to show each scenario. For example, pressing the "OUT OF BOUNDS" button would simulate the user's current location moving out of boundary and show the relevant notification message and prompt the user to take action (Image 19).

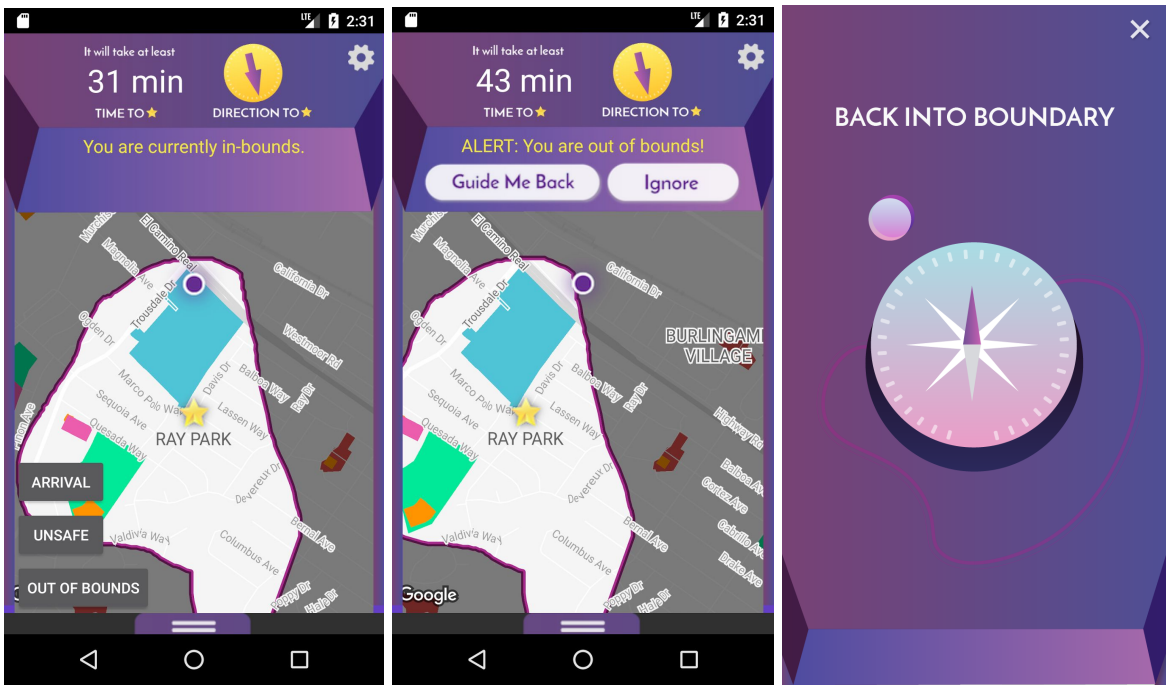

Image 19: Simulation buttons for each scenario (left) and helping user recover from stepping out of bounds (middle and right).

Hard-Coded Data

We used Google Maps API to color code the different areas of the map. The Google Maps Style Tool (https://mapstyle.withgoogle.com/) allowed us to select different types of regions we wanted to color on our map. However, some types of areas like restaurants and shops were not listed as features we could style using this tool, and other types of areas were listed, but were not dense enough (e.g. attractions were listed, but we were not able to locate any attractions on a map because they were located so sparsely). To compensate, we selected different types of areas to color in order to demonstrate our concept (e.g. we colored sports stadiums to represent attractions and businesses to represent shops).

Additionally, we were able to use the styles we designed using the Google Maps Style Tool by exporting the styles as JSON files. However, once a style is exported as a JSON file, we were unable to turn on any colors that were not on at the time of exporting, and we were unable to turn

off any colors that were on at the time of exporting. Because of this, we had to create 32 different JSON files, one for each possible combination of our five filters.

The user's current location is also currently hard-coded as Stanford University, which made creating the simulations and testing our app using the Android Studio emulator simpler.

What is missing and what might you add in the future?

In the Draw Page, there is no key provided for each color coded area of the map. We were planning on putting the relevant icon of the filter on top of the color coded area (e.g. house icon on top of the green residences color coded area), but we were not able to do it because Google Maps API does not provide that feature. Additionally, in the future, we would like to provide real data for each filter and crowdsource users' experiences to determine "unsafe" areas.

Summary

We began this quarter with a broad theme: local community. Within the first week, we had narrowed our focus down to navigation within the local community, and through numerous need-finding interviews on this topic, we discovered that for many people, the fastest route offered by many popular navigation apps is not always preferable. Through POVs and HMWs we brainstormed various ways to address this problem, eventually deciding upon an application that would encourage serendipity and exploration, while also providing users with comfort and safety. We went through several iterations of our idea, beginning with dozens of sketches, moving on to low and medium-fi prototypes, and finally ending up with a high-fidelity prototype that embodied our three core mission statements: "Be There," "Be Safe," and "Be Immersed."