**Stop Spots**



Project Documentation

CSC 371/471 - Mobile Application Development (iOS)

DePaul University - Winter 2020

**Table of Contents**

[**0. Overview**](#_d4vzku4itg9j)3

[**1. Description of all the features that have been completed**](#_41mbwpefxi2q)3

[**2. Screenshots of everyscreen with explanations**](#_dgzd7x3j0pw8) **4**

[**3. Final Project Discussion**](#_mxv8ajr40f12)  **7**

[**3. Team Contributions**](#_kxoh66r93ey) **9**

# 0. Overview

## A. Team Members & Roles

1. Jess Bender - Front End Developer
2. Jackson Hurst - Back End Developer
3. Jennifer Nguyen - Full-Stack Developer

## B. Source Code - GitHub

<https://github.com/jacksonhurst/Stop-Spots>

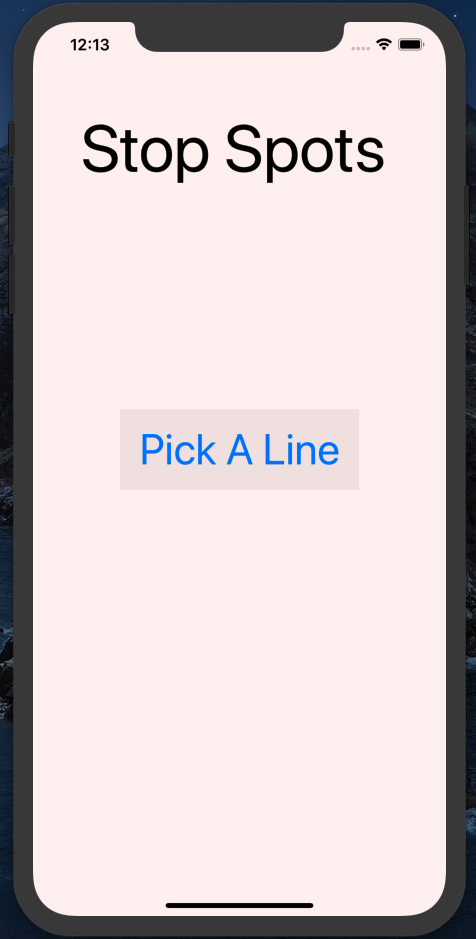
# 1. Description of all the features that have been completed

* List of all CTA Lines (only red line currently functional)
* Alert notification for all the CTA Lines that are still under development
* List of Red Line Stops (only Fullerton -> Jackson currently functional)
* Pull in a list of 20 nearby spots (shops, restaurants, hotels, etc.) for each stop using the Google Places API
* Pull in ratings based on Google Reviews for each spot using the Google Places API
* Generate Google Maps with a marker on the spot for each individual spot using the Google Maps API

# 

# 

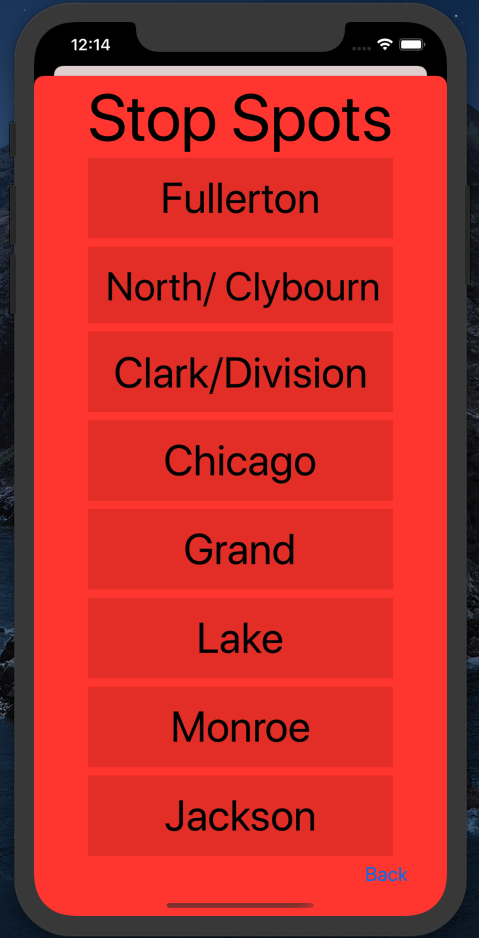
# 2. Screenshots of every screen with explanations



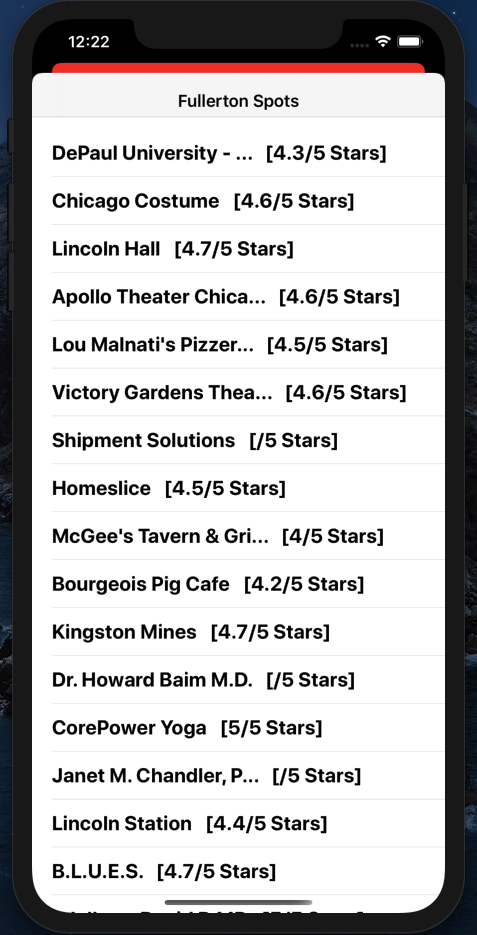
**Home Screen:** This is the home screen. Super simple entry page to start the user off. Simply click Pick a Line to get started



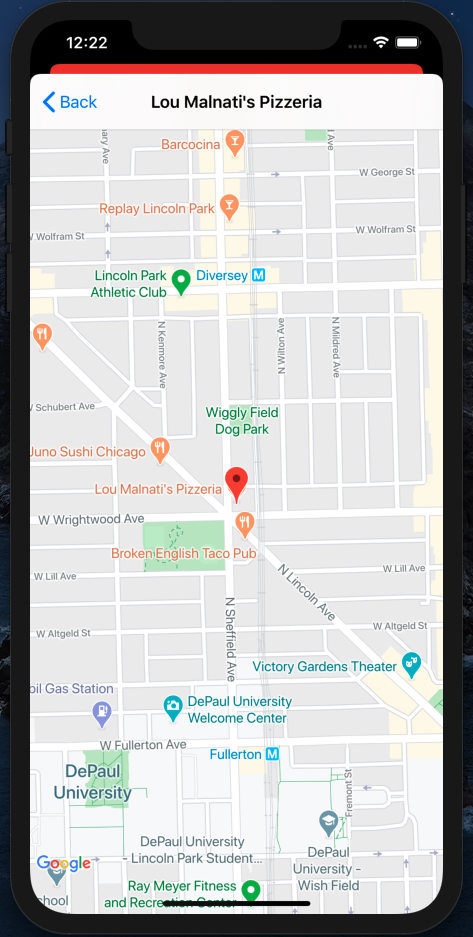
**Pick a Line Screen:** Another simple screen. This screen serves as a simple way to pick the line you want to pick a stop for. We chose to just implement the red line at the moment, as we believe implementing multiple lines would be overly ambitious for the scope of this project. When you tap on the button for any line besides the Red Line, you will receive a alert message saying that we are still working on other lines, and red is the only functioning line at the moment.



**Pick a Stop Screen:** Once again a simple screen with a limited number of the stops for the red line. Once again, due to the amount of work that goes into building the API requests and other functionality for each Line/Stop, we chose to limit the stops we had for the sake of this project. We chose to limit it to the stops between the DePaul Lincoln Park Campus and the DePaul Loop Campus.



**Pick a Spot Screen:** This was the first technically challenging screen to build out. We chose to use the detail/master view that we had learned for assignment 8. This screen requires an API call to the Google Places API to fetch all of the shops/restaurants/hotels/etc. within a given range of the stop. This is completed by getting the latitude and longitude of each of the stops, and the doing the search. For the sake of this project we store the lat/lon of each stop statically in the code.



**Google Maps Screen:** Finally once a spot is picked, we use our other API which is Google Maps. The request for Google Maps requires the latitude and longitude which we store in values in a “spot” struct. All of the spots for each stop are built into an array of these spot structs, and then we simply access the values once a specific one is selected. It also requires that we build a “marker” which we once again just use the same latitude and longitude.

# 3. Final Project Discussion - Done individually (Jess)

For this project we used two API features. The first API feature we used was Google Places. This API was a big part of our project. Inorder to access the data we needed with this API we had to create a google cloud project. Then, we used the API key to retrieve 20 spots nearby the train stops we selected. This gave us a total of 160 spots between the 8 train stops. For each of the 160 spots we collected were given latitude and longitude. This way we were able to store the latitude and longitude and run a HTTP GET request to the Google Places API. The API then returned an Array of the 20 spots in a JSON file. Google Places is a very cool API that gives a ton of information including but not limited to the name of the place, the phone number, the latitude and longitude, and the average reviews. However, the purpose of our project, we only needed the name of the place, the latitude and longitude, and the average reviews. All the results gathered within this API are in the detail/master view in our Xcode project. In the master view the user can see the spot name, for example, DePaul University, the rating out of 5 stars.

The second API feature we used was Google Maps API. For this API we had to install the Google Maps SDK package as a dependency in our project since a simple HTTP request was not available like the first API we used. Inorder to use this we had to install cocoapods and created a podfile. This allowed us to put a map on the detail view page in our app. We took the latitude and longitude of the stop and put that into the center of the app and made it the maker showing the exact location of the spot. This map also shows nearby locations and the street names. It even shows the train stop near it. The user is also able to move the map around to see anything they want to see in the area. The user can also zoom in and out of the map allowing them to read the street names and other locations better.

One of the biggest challenges I faced was getting the app we envisioned in our heads on the screen. We wanted to use a lot of things that we did not know how to use. For example, the train stops page was a big challenge. We did not know how to fit all the stops onto one screen. We looked into a scroll option but did not know how to use that. Then we looked into what I believe is called a picker. This would have been a good option for us, but we were not sure how this would work with our backend code. Thus, we decided to limit the number of stops the user could choose from to just eight stops. This ended up working in our favor since the API took up a lot of time and we would have never been able to build out all the stops for the red line anyways.

Another issue we had was with the master and detail view page. We wanted to use this page to show the spots the user can choose from and once they picked a spot the next page would show them the location of the spot on google maps. We had a hard time connecting the master view set up to the rest of the screens we already set up. When we simply Ctrl click and drag to the master view set up for some reason it would show us the detail view page before the master view page. We had no idea how to fix this. We first tried to connect it to different pages and that all made the app crash. Then we tried playing around with the connections on the master view setup. We were able to get the app to work by deleting the bottom connection to the master view page. I have no idea why that worked but it solved our problem and were able to get the app completely functional.

Our app has a lot of limitations and definitely could have been improved with more time. Our first limitation is that we only have the redline working. If we had more time, we would have liked to have all lines working. Right now, our app only has the red line functioning. When the user clicks on a line other than the red line it brings up a warning box that states for example, blue line coming soon, only red is working. Thus, ideally the app would eventually advance to all the lines stops.

Another limitation we have is for the one line we do have working, red, we only have 8 stops that show the local spots. Currently we only have Fullerton to Jackson working. If we had more time, we would have liked to have all the redline stops available. We also had trouble finding a way to list all the stops on the screen. We did not learn about any scroll options and since there are a lot of stops for the redline it would have been nice to use a scroll to see all the stops in a reasonable size. Since we did not know how to do that, we choose just to do eight stops since that would fit on the screen.

Another limitation is the pick a spot screen we only show twenty spots and we would have liked to include more. Another thing we wanted to do was have it tabbed so that you can choose how to filter the spots. For example, one tab would be for restaurants and another could be for shops in the area. However, again due to time constraints we could not separate them out.

Finally, our last limitation is that the screens use segues. This just opens the screen on top of the other screen meaning if you went through the screens multiple times it would eventually start lagging. This is also not very user friendly because you can accidentally scroll down and go back a page you did not want to go back to.

Overall, I like developing in iOS. It has been a challenging learning curve as for I have never used an Iphone or a mac before in my life but I found this class very interesting. I think by the end of this project I finally got a handle of coding in swift and using Xcode. There is a lot more I would like to learn with it however. I want to become an app developer in the future when I graduate, thus this class was very helpful in showing me what I might be doing after graduation. I've learned that building an app is very time consuming. There are a lot of individual parts that go into building an app. For example, you have to put all the items you want on the screen and build constraints for the app and test the constraints on different versions of the Iphone to ensure that the view is similar for each version of the Iphone. Another thing that you have to pay attention to is the app icons. For each version of the Iphone the icon is a different size. Thus, you have to resize the image for the icon to fit all the different versions of Iphone. Paying attention to all these small details is not something I would have expected with app development however it has proven to be very important to it.

# 4. Team Contributions

Jess worked on the front-end of the app specifically creating all the screen views, design, and connecting the buttons to go to each screen. Jennifer created the detail/master view screens and created alert boxes for the other lines that were not in function yet. Lastly, Jackson worked on the back-end of the application using the APIs to get the spots and maps working. We came together on Sunday and worked on the project as a team as well. We all equally put in work into the final documentation.