Hi Matt,

This report documents the basic functionality of the app we have designed for you. In it, we describe the various technologies which it involves, the pros and cons of each, and provide you with a guide for interacting with them. If you have any further questions, we would be more than happy to answer them.

**Technologies Examined**

* Yelp
* Google Places
* WalkScore
* Dark Skies
* Geolocation

**Review of Reviews**

Of the technologies available for getting user reviews of establishments, we found Yelp and Google Places to be the most well-supported and user friendly. Compared to other interfaces, these provide us with the most relevant and complete information. If you are not familiar, both Yelp and Google Places are tools by which users may indicate their preferences toward certain establishments. Both provide an application programming interface (or API) by which other programs (such as our app) may download and process information from these services.

As far as differences between these API’s, we found that Google Places provides more general information about establishments than Yelp. This includes things such as hours of operation, etc. While this information is typically freely available online, we liked that Google Places made it easier for us to obtain it programmatically. A downside of Google Places, though, is that it requires you to specify latitude and longitude coordinates whereas Yelp and most other API’s allow you to specify an address. This isn’t a big deal--converting an address to coordinates is easy enough--but it is nice that Yelp does it for you.

**Mapping and Directions**

We examined a few different options for finding directions, but the clear winner is Google Maps. It’s an interface with which everyone is familiar, it has an easy-to-use API, and it, perhaps most importantly, gives correct directions. Alternatives include Mapquest, Bing, and Apple (for devices of that manufacturer), but our recommendation is resoundingly Google Maps.

**API Calls**

Most API’s have a limit on the number of “calls” or requests you can make of them per day. Typically this number is large enough to enable testing and some small-scale user activity, but some API’s (like WalkScore) only give you a very limited number of calls. If you get to the point where you are running out of calls, you’ll have to pay the API for more. Here’s a list of API’s we used and how many free calls you get per day. Exceeding that, most API’s require that you contact them to discuss pricing going forward.

|  |  |
| --- | --- |
| API | Free Calls/Day |
| Google Places | 100,000 |
| Google Distance Matrix | 2,500 |
| Google Geocoding | 2,500 |
| Yelp | 10,000 |
| WalkScore | 100 |
| Dark Skies | 1,000 |

**Geolocation: A Note**

Something we also considered was having the app search for places based on your phone’s current location. As we can expect many users to be interacting with the program while walking, it may make sense to just grab their GPS location and feed that directly to Google Places or Yelp without worrying about addresses and such. If that’s a route you’d like to take, there would be a tiny bit of overhead using WalkScore as that API requires both GPS coordinates and an address. That is the purpose of Google’s Geocoding API. It converts between GPS coordinates and addresses. It’s important to note that we need to use this API whether the user is on a computer or using her phone; we’re not using it to get GPS coordinates, but to convert those provided by a phone or web browser to an address.

**Display of Information**

Because you’re designing an app, there are many aspects of smartphones of which you can take advantage. One of the downsides of smartphone devices, however, is that they make it somewhat difficult to type. One way of getting around that, especially for your app, is to simply display the information you think users want and let them choose whether they want to use it or, if it is insufficient, manually search for something else.

Because you already have categories set up for your signs (Commercial District, Public Space, etc), it may be helpful to grab the phone’s current location and use it to display the top 3-5 results for each category based on what’s nearby. If this information isn’t helpful, a traditional search box would be available, but this way users would be able to see the most relevant results in the area.

**Lots of API’s**

While there are many advantages to utilizing multiple API’s to achieve your goals, it’s important to note that making all the calls required to do so takes time. In our tests, we found that the time before results are displayed is increased substantially for each API used. We chose to leave this alone in our product so you can both get a feel for what each API provides as well as for what sort of delay we’re talking about here. In a commercial product, though, we recommend utilizing only what API’s serve your needs most. It’s also possible to provide baseline functionality with minimal API calls and then provide more information as the user requests it, but this might impact the usability of your eventual app.

**Informational Differences**

We should also note that there are quite substantial differences in how various API’s return information to us. Most of the information is the same (names of locations, phone numbers, etc), but how it is organized differs based on the penchants and proclivities of the people who wrote the API. This means that anyone wanting to get the same information from, say, Google Places and Yelp would have to go about it in different ways. This isn’t really a big deal to do (it basically involves asking for a different bit of information, say “location” versus “place”), but it means there is some overhead involved in asking different API’s for the same information. It’s definitely doable, but it would require additional development time to accomplish and maintain connections with multiple API’s. For this reason, we recommend sticking to just a couple services which you like best.

Thanks for reading this report. We hope you’ve found it helpful as you consider which API’s you would like to use in the future. If you have any further questions, feel free to contact either us or Dr. Watson--we’d both be happy to help.

Thanks again,

Corey Glasson

Erik Minges

Gavin Nicol

Travis Pomeroy

Justin Toniazzo

Haizhou Wu