



College of Engineering

CS CAPSTONE TECHNOLOGY REVIEW

JUNE 4, 2018

I HEART CORVALLIS

PREPARED FOR

LYNDI-RAE ELISABETH PETTY

Signature

Date

PREPARED BY

GROUP 14

I HEART CORVALLIS

BRADLEY IMAI

Signature

Date

Abstract

This document takes a closer look at the technologies that will be carried out into the I Heart Corvallis Mobile application. Social media integration, user action verification (geolocation) and event display will be the three topics discussed. Each topic will be broken down into three potential options which are all compared and contrasted. Lastly, we will determine which implementation will be used in our application.

CONTENTS

1	Social Media Integration	2
1.1	Overview	2
1.2	Criteria	2
1.3	Potential Choices	2
1.3.1	Instagram API	2
1.3.2	Twitter API	2
1.3.3	Facebook API	2
1.4	Discussion	3
1.5	Conclusion	3
2	User action - verification (Geolocation)	3
2.1	Overview	3
2.2	Criteria	3
2.3	Potential Choices	3
2.3.1	Google Map API	3
2.3.2	Cordova - Plugin - Geolocation API	4
2.3.3	Android Location API	4
2.4	Discussion	4
2.5	Conclusion	4
3	Event Data Display	4
3.1	Overview	4
3.2	Criteria	4
3.3	Potential Choices	5
3.3.1	Google Map API	5
3.3.2	MapBox	5
3.3.3	Microsoft Bing Map	5
3.4	Discussion	5
3.5	Conclusion	5
	References	6

1 SOCIAL MEDIA INTEGRATION

1.1 Overview

Integrating a social media aspect such as twitter and Instagram photos from an event into our application will allow the user to view what others have done at those particular events and also entice them into exploring them. Having tweets and photos posted on that specific event will be crucial to keeping that events page organized.

1.2 Criteria

One of the main features to this application is to allow the users to post and also view others experiences from that event. Each events information page will list photos and tweets from the most recent posted. By giving each event a specific hashtag, users will be able to upload their photos and tweets to the correct events. On the back end, we will implement a filter that searches for specific locations and event hashtags so that the user can easily navigate and view others posts.

1.3 Potential Choices

1.3.1 Instagram API

Instagram's API will allow us to post Instagram photos or a photostream on our mobile application from any personal account. This feature will also create a gallery of images that will automatically update as new pictures are added. The process of implementing these features into our application will first start off by setting up the library, followed by obtaining the web API keys, configuring the login page, building the API output, and lastly dumping the data out. [1]

1.3.2 Twitter API

Twitter Kit SDK will allow us to display Twitter content onto our mobile application. Below will be a list of kits we can used to implement into our application.

- TwitterCore - Let users log in with Twitter via Single Sign-On. Make authenticated requests to the Twitter API to load Tweets, users, search results, and other Twitter content. [2]
- TweetUI - Embed media-forward or compact Tweet views into your views and list views to show users information that is relevant to your app. Let users share Tweets discovered via your app. [2]
- TweetComposer - Let users compose new Tweets as part of your app or share content discovered via your app. [2]

For our feature, we will most likely be incorporating the TweetUI kit in order to display content into our application. Within TweetUI we will be using the tweetView and CompactTweetView to render our tweets. The Tweets will be requested through the Tweet API or TweetUtils which will cache recent requests. From there we will be able to display the correct content to our application. [3]

1.3.3 Facebook API

Facebook provides its own SDK to integrate with our Android application. There are three different APIs that perform different functions. We will need to first get an Access token to access those APIs. Reviewing the developer's page on access tokens, we will be using the user access token and page access token. These functionalities will be provided below.

- User access token - this kind of access token is needed any time the app calls an API to read, modify or write a specific person's Facebook data on their behalf. [4]
- Page Access Token - This token is very similar to the User access token however, it provides permission to APIs that read, write or modify the data belonging to a Facebook Page. [4]

To obtain a page access token, we will first obtain the user access token and get the manage pages permissions. Once that is achieved, we will get the page access token or the Graph API which will allow us to make API calls on those pages.

1.4 Discussion

All three of these APIs can be implemented into our application. The features that these APIs provide are all very similar, however the information that they display is a little different. Facebook's API seems to be the middleman of Instagram and Twitter. It will be able to display photos and status of users, where Instagram only posts photos, and Twitter only postes statuses.

1.5 Conclusion

While all three APIs provide great features to our application, we will first be implementing Instagram's API. By providing a feed of photos in each event, other users could be encouraged to attend that event.

2 USER ACTION - VERIFICATION (GEOLOCATION)

2.1 Overview

Retrieving the geolocation (current location) for our application is a critical tool to our mobile application. By obtaining a user's current location, we will allow the user to see various events that are happening around them. After reviewing a few potential choices, I came up with the Google Maps API, Cordova-Plugin-Geolocation API, and Android Location API.

2.2 Criteria

The user will be able to turn on their location services which will allow them to view the events that are happening around them. By clicking on an event, the user will be prompted if they would like directions to that event. If so, this feature will direct the user to their event in the fastest way.

2.3 Potential Choices

2.3.1 Google Map API

Google Maps API is a useful tool for retrieving the current location of an individual. Provided on the Google Maps API website is a tutorial on how to display the geographic location of a user or device on Google Maps using the browser's HTML5 Geolocation feature along with the Google Maps JavaScript API. This website also provides example code and comments on how it works, making this a very useful tool. Both Android and iOS Google Maps APIs require the app to prompt the user for consent to use their location services. [5]

2.3.2 Cordova - Plugin - Geolocation API

This API provides the location information of any device in the form of its latitude and longitude. According to Cordova's website, Common sources of location information include Global Position System (GPS) and location inferred from network signals such as IP address, WiFi, and Bluetooth MAC addresses. A quick note is that the API is based on the W3C geolocation API specification and only executes on devices that don't already provide an implementation. [6]

2.3.3 Android Location API

Within the Android API, you can call `android.location.LocationListener`. There are three main components of the API. The `Location` class gives geographic location, including the latitude and longitude. `LocationManager` provides access to the system location services. And lastly, `LocationListener` is used for receiving notifications from the `LocationManager` when the location has changed. [7]

2.4 Discussion

All three APIs can be incorporated into our Android application. They all can retrieve the location of a mobile device. Cordova, on the other hand, can retrieve a few more features such as the IP address, WiFi, and Bluetooth MAC addresses. These features are interesting but not what we are looking for. Android Locations API has a built function within Android Studio making it very simple to implement a function to retrieve the location. However, according to the Android developer's page, it recommends to use the Google Location Service API.

2.5 Conclusion

Even though all three location service APIs work for our application, we will be using the Google Location Services API. The Google Play Services provides a more powerful, high-level framework that automatically handles location providers, user movement, and location accuracy. It also handles location update schedule based on power consumption parameters we provide. In most cases, you'll get better battery performance, as well as more appropriate accuracy, by using the Location Services API.

3 EVENT DATA DISPLAY

3.1 Overview

This application will also serve as a platform to highlight useful resources around Corvallis. By implementing two maps into our application we will be able to fulfill this requirement. The first map will showcase notable establishments which consist of activities and entertainment, grocery stores, restaurants, shopping centers and city offices. The second map will display events approved by the Corvallis Community Relations office (CCR) which students will be able to accumulate stamps from. Having separate maps will allow the user to easily navigate to local events and useful establishments.

3.2 Criteria

Students will be able to access two maps which display useful resources around Corvallis and events that are approved by the CCR. We will incorporate a sidebar on the first map (useful resources) that filters out establishments by type which will help students narrow down their searches.

3.3 Potential Choices

3.3.1 Google Map API

Google Maps API provides a feature called markers. A marker identifies a location on a map and can display a custom image of that location. According to the Google Maps marker developers page, markers are designed to be interactive. By default they receive click events, so you can add an event listener to bring up an information window displaying custom information. Another feature of this API allows the user to remove a marker from the map. Under the title, adding a Map with a marker on Google's website explains the process on how to implement markers or events on the map. The website provides sample code for us to implement into our application. [8]

3.3.2 MapBox

MapBox is very similar to Google Maps on displaying various icons on a map. However, you can customize every aspect of the map from the colors, hiding or showing specific layers, to choosing which information to present on the map, all while the users are interacting with the map. The only downside to MapBox is that there is a fee after so many views and requests to the API. [9]

3.3.3 Microsoft Bing Map

The Microsoft Bing Maps platform provides many controls and service APIs for our application. A list of them will be provided below from their website.

- Bing Maps V8 Web Control - the latest Bing Maps JavaScript API. Combine the AJAX map control with the Bing Maps REST Services and the Bing Spatial Data Services to create powerful Web sites and mobile applications with the latest imagery and location functionality. [10]
- Bing Maps REST Services - perform tasks such as creating a map with pushpins, geocoding an address, retrieving imagery metadata or calculating a route. [10]
- Bing Maps WPF Control - The Bing Maps WPF Control SDK lets developers integrate Bing Maps into rich Windows Presentation Foundation (WPF) applications. [10]

In a nutshell, Bing Maps provides many API features and platforms that can be incorporated into our application. All of the API keys are also provided on their website for easy access.

3.4 Discussion

Reviewing the three APIs, they all have their ups and downs. All three of the APIs provide a map in which we can modify to our own specifications. However, MapBox has a fee after so many requests to its API of which is a huge downfall. On the other hand, Google Maps and Bing's API are very promising for the features they provide. Implementing Bing's API may be a challenge, on the other hand, Google Maps API seems very reasonable as it provides example code and good explanations on how to apply it to our application.

3.5 Conclusion

Although all three APIs provide great features, we will be using Google Maps API as our source to display events to our application. The Google Developers manual for this API provides detailed information on how to implement their code into our application.

REFERENCES

- [1] Connecting with api data from instagr.am using php. [Online]. Available: <http://spyrestudios.com/connecting-with-api-data-from-instagr-am-using-php/>
- [2] Twitter kit for android. [Online]. Available: <https://dev.twitter.com/twitterkit/android/overview>
- [3] Show tweets. [Online]. Available: <https://dev.twitter.com/twitterkit/android/show-tweets>
- [4] Facebook access tokens. [Online]. Available: <https://developers.facebook.com/docs/facebook-login/access-tokens/#pagetokens>
- [5] The google maps geolocation api. [Online]. Available: <https://developers.facebook.com/docs/facebook-login/access-tokens/#pagetokens>
- [6] cordova-plugin-geolocation. [Online]. Available: <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-geolocation/>
- [7] Android location and maps. [Online]. Available: <https://developer.android.com/guide/topics/location/index.html>
- [8] Displaying marker and evnts google map. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/markers>
- [9] First steps with the mapbox android sdk. [Online]. Available: <https://www.mapbox.com/help/first-steps-android-sdk/>
- [10] Mocrosoft bing maps. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ff428643.aspx>