



COMP 4905 – Computer Science Honours Project
Final Report
Carleton University

Author: Malik Ehsan
Student ID: 100833433
Supervisor: Dr. Dwight Deugo

Table of Contents

0.0 - Abstract:	4
1.0 - Introduction:	5
1.1 - Problem and motivation:	5
1.1.1 Knowledge Sharing:	5
1.1.2 - Local geographic Knowledge:	6
1.1.3 - Weather to fish ! :	6
1.1.4 - Equipment to use:	6
1.2 - Goal:	7
1.3 - Objectives:	7
1.4 Outline	8
2.0 - Background:	9
2.1 - Fishing maps and books:	9
2.2 Fishing Forums:	10
2.3 - Mapping on Mobile:	10
2.4 - Geocoding:	10
2.5 - Reverse Geocoding:	10
2.6- Android:	11
2.7 Location Services:	12
2.8 - Similar work:	12
3.0 Approach:	14
3.1 Project Design:	14
3.2 UI	15
3.3 Backend:	15
3.3.1 Backend as a Service:	15
3.3.2 - A little bit about Parse:	18
3.4 - Obstacles:	19
3.4.1 REST:	19
3.4.2 JSON:	20
3.4.3 Problem of finding water body:	20
3.4.4 - Google Elevation library:	22
3.4.5 - Google places API:	23
3.4.6 - ReverseGeocoder.com:	24
3.4.7 - Geonames:	24
3.4.8 - How to process JSON objects in Android.	27
3.4.9- Nested scrolling:	29
3.5 - Design Pattern:	30
3.6 - Development Model:	31
4 - Results and Validation:	32
4.1 Self-Evaluation:	35
4.1.1- Objective 1:	35
4.1.2- Objective 2:	35
4.1.3 - Objective 3:	36
4.1.4- Objective 4:	36
4.1.5- Objective 5:	37
4.1.6- Objective 6:	38

6.0 - Conclusion:	40
5.0 - Future work:	40
7.0 - Work cited:	42

Table of Figures:

FIGURE 1- EXAMPLE OF A CONTOUR MAP DISPLAYING WATER DEPTH.	9
FIGURE 2 – EXAMPLE OF A MAP SHOWING GOOD FISHING SPOTS	9
FIGURE 4 – SPLASH SCREEN	15
FIGURE 5 – GARTNER’S MAGIC QUADRANT	18
FIGURE 3 - HIGH LEVEL ARCHITECTURE DESIGN	14
FIGURE 6 - EXAMPLE OF A JSON OBJECT RETURN BY GOOGLE ELEVATION API	23
FIGURE 7 - REVERSEGEOCODER.COM EXAMPLE	24
FIGURE 8- ASYNCTASK’S STEPS	28
FIGURE 10- CLIENT MAKING CALLS THROUGH API	31
FIGURE 11 - APP SCREEN SHOT 1	37
FIGURE 12 - APP SCREEN SHOT 2	38

0.0 - Abstract:

The report describes design and implementation of AnglersMate, an android-based app that makes use of various APIs and cloud computing, enabling users to save their favorite fishing spots and share valuable information about their fishing trip with other fellow anglers around the world. The app enables other anglers to figure out best weather, suitable rigs and direction to a particular fishing spot.

1.0 - Introduction:

1.1 - Problem and motivation:

Fishing or sometimes known as angling in its many forms. It is one of the most popular sports in the world and especially in North America. The sport is liked by many people of varying ages taking to rivers, lakes, canals and shores. Both boat fishing and bank fishing are popular way of catching. Bank fishing is fishing from places where the land meets the water edge. Bank fishing is performed with a reel, nets, traps, and spears, fishing bait and different types of lures, fishing lines are also used with or without a rod. Many other factors contribute to a successful fishing trip, such as local knowledge, water depth, bank structure, location, weather and types of bait and lures. Fishing from boat is another form of recreational angling, which uses the same knowledge and techniques and at the same time allows access to different fishing ground and many different species of fish that reside in the deeper water. Mostly Anglers rely on a paper map sold by local tackle shop. These paper maps are usually outdated and don't contain information about the types of baits and lures that are suitable for a particular lake, as different fish are attracted to different kind of baits. Lakes with high traffic usually suffer from extinction and overfishing. Taking a trip to lakes, reservoirs or oceans close to the city proximity usually results in an unsuccessful trip with no decent catch.

1.1.1 Knowledge Sharing:

Local knowledge is crucial for a good fishing trip. Local knowledge ranges from knowing where good fishing points are, both on banks and on water, to where the lake gradients

into deeper depth. Also, sunken logs and rocks are the points where game fish usually hide and only the local dwellers or local fishing guides know the spots that yield high fish presence. Knowing where these locations are is crucial for an effective trip. Going to a lake without being equipped with this knowledge will result in angler trialing various spots till they find a decent one. The paper maps provided at the local tackle shops do not have specifics on where to find suitable structures to fish.

1.1.2 - Local geographic Knowledge:

Local geographic knowledge is crucial for a good fishing trip. Local knowledge ranges from knowing where good fishing points are, both on banks and on water, to where the lake gradients into deeper depth. Also, sunken logs and rocks are the points where game fish usually hide and only the local dwellers or local fishing guides know the spots that yield high fish presence. Knowing where these locations are is crucial for an effective trip. Going to a lake without being equipped with this knowledge will result in angler trialing various spots till they find a decent one. The paper maps provided at the local tackle shops don't have this local sort of specific knowledge about the water body.

1.1.3 - Weather to fish || ! :

Any time is good time for fishing but if you want to increase your chances of success, it is recommended to stay away from the cold fronts, in other words, it recommended to not fish on days when temperature drops sharply [1]. It is recommended not to fish on days when the wind is blowing from the north, as it usually brings colder air. Warmer days are usually better for fishing, especially if it's overcast

1.1.4 - Equipment to use:

Choosing the right gear when going fishing is crucial. Basic equipment used for fishing are, a rod, reel, line and a lure or a bait. Different types of reels include spin casting, spinning; bait casting, fly-fishing, down rigging and trolling. Once you have your hook and bait set up, last thing you want to do is have your line broken. It's important to match the strength of your fishing line to the size of fish biting on your hook. Fishing line strength is measured in pounds. Lastly lures play a vital part in catching the right fish. Certain fish don't bite on certain lures. Three main kinds of lures are surface lures (fishing from surface of water, these lures don't sink and stay afloat, shallow water lures (to be used in 5 to 6 feet of water) and deep water lures.

1.2 - Goal:

Goal of this project is to design develop a mobile platform for anglers to share geographical, weather and equipment information with other anglers any where in the world. Anglers will be able to search a specific lake, reservoir or sea and check out fishing spots around it. Anglers will be able to rate their angling experience at a particular spot for everyone else to see and plan their next fishing trip accordingly.

1.3 - Objectives:

The objective of this project is a design an android mobile app with the following features:

- 1) A user login / sign up feature.
- 2) A user dashboard, which gives user choice of browsing the map or searching a specific water body in a search bar.

- 3) Map generated with the GOOGLE maps API.
- 4) Giving users capability to hold a point on the map and save it.
 - a. Users inputting information about the
 - i. Place
 - ii. Kind of fish they caught.
 - iii. Type of equipment they used
 - iv. Weather conditions
 - v. Photos of their catches.
- 5) Other uses adding photos, rating their trips and adding their own experience at the spot.
- 6) Viewing various spots added by other users, color-coded by the average trip rating.

1.4 Outline

First section describes the problem, my motivation behind the idea for this app and also the goals and objectives I had set for myself. Section 2 gives the readers brief background information on various concepts necessary for the development of this app. Section 3 I have shed light on various implementation goals I set for myself and also obstacles I faced in the process. Last section is has my personal evaluation, results and conclusion.

2.0 - Background:

2.1 - Fishing maps and books:

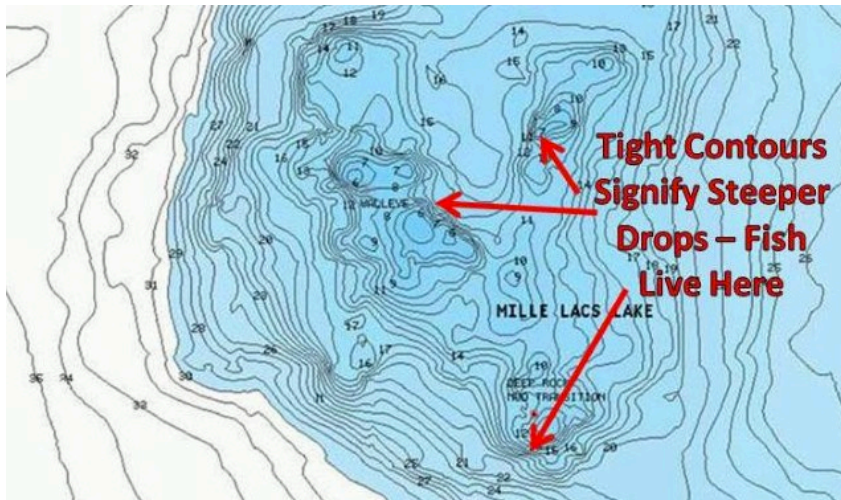


Figure 1 Example of a contour map displaying water depth.

a particular lake. These fishing maps contain detailed fishing maps and lake and river charts outlining types of fish species found in these water bodies. These charts also contain information about local tips and techniques and sometimes outlining **provincial and federal** fishing regulations at a particular water body or a region. Some more expensive maps contain information on water contours, gradients of a water body.

Problem with these maps is that they are not every specific and usually give the anglers a general idea as to where to try their luck.

The app can be contrasted with hard copy fish maps or maps books. These fishing maps and map books are usually jam-packed with information one needs to know to fish at any region or

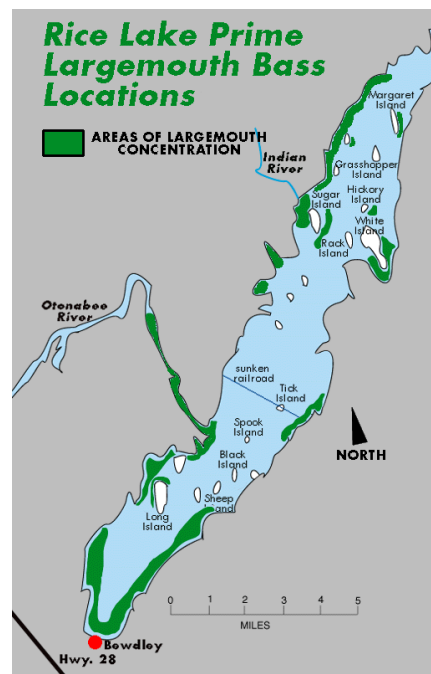


Figure 2 - Example of a map showing good fishing spots

2.2 Fishing Forums:

Fishing forums are widely used as reference by anglers when visiting a new lake. It doesn't matter if you are a seasoned angler or a beginner, one of the first things one does before going to a fishing trip is hopping on the internet to see what other anglers might have to say about fishing in that area, If you're lucky you might get a few pointers on where to start your trip. The forums are usually a hit or miss, sometimes people on the forums are helpful and friendly and give out tips and sometimes people don't want to share information.

2.3 - Mapping on Mobile:

The most important component of any app that uses mapping is geocoding and reverse geo-coding:

2.4 - Geocoding:

In layman's terms, geocoding is a process of converting a human-readable address or postal code into geographical coordinates (in terms of longitude and latitude); these coordinates can then be placed on a map as a position or a marker for representation.

2.5 - Reverse Geocoding:

It's the opposite of geocoding; here input is a set of latitude and longitude, which in turns returns a human readable street address or postal code or even a place name. Both reverse geocoding and geocoding are going to be crucial part of my app. There are several map APIs available on android for this purpose. For this project I will stick to Google Mapping API. Google provides bunch of Google Maps APIs, its APIs range from

static image API to web services API to Google places API. Google provides API picker, which puts all the available APIs in one place for developer to choose from.

Google maps' android API automatically handles calls to Google Maps servers, data downloading, and displaying maps and also takes cares of the touch gestures on the map. It gives powerful functionalities to the developers, like adding markers, polygons and overlays directly on the map and also gives developers ability to change the users' view of the map i.e. satellite view. Google is also great when it comes to providing documentation, code samples, tutorials and SDKs. Google is all about innovation and Google Maps API is no different, Google is constantly adding more features to it Mapping tools and staying way ahead of the competitors. Before we look at the Google Maps' API, lets look at the Operating system itself.

2.6- Android:

Android is an Operating system for mobile device, which provides full software stack. Android provides both NDK (native development kit) and SDK (software development kit) for development purposes. Android SDK provides libraries to developers that make it easier for developers to access and make use of sensory tools like accelerometer, gyroscope and GPS on an android device. NDK uses native code languages like C and C++, even though you can still develop using NDK, using native code in android doesn't always optimize the performance but it does make the coding process more complex. The android SDK is written in Java and uses Dalvik virtual machine and consists of various libraries and development tools^[6]. NDK on the other hand is chiefly used for getting access to resources at the lower level and also to port c and C++ codes from

other projects. NDK also comes in handy in developing multiplatform applications. A code written in C++ for android can be easily ported in and run on an iOS or a windows device. For this project I will be using an android SDK in java. ^[6]

2.7 Location Services:

Android serves as a middleware between applications and location services provided by the android hardware. The main component of the location services on any android hardware is the LocationManager class. LocationManager provides APIs access to the devices' GPS unit, which helps mapping APIs being used in an app to determine the location of the device. Location manager provides applications cyclic updates on the device's geographical whereabouts. Location manager will be a crucial part of this project, as I want the users' devices to be geographically aware ^[16]

2.8 - Similar work:

There are few apps that compare to AnglersMate. Fishbrain for example is the worlds biggest, community based fishing app. It provides anglers to upload picture of their catches, fishing reports and observations viewable for their own personal use, their friends' circle or for everyone to view publicly. It's a social network for anglers. It gives anglers a platform to share their experience at a particular geographical spot. It also contains information about baits and techniques for less seasoned anglers to benefit from when visiting a fishing spot.

GoFree Hooked (Lowrance) is another app that uses GPS to let users record catches. It has the same functionalities as FishBrain except it allows anglers to compete in tournaments with other users, other anglers can view tournaments via a feature called live leaderboard. [15]

The Fishidy app allows members to utilize online map to record catches and log information pertaining to those catches, on something the app calls “bragging board”. Other features (like picture sharing, message boards, details about catches) in Fishidy are similar to “GoFree Hooked” and “Fishbrain”. [7]

3.0 Approach:

3.1 Project Design:

Having no prior experience in android programming, I was pretty much researching everything online to get tips on where to start my project. I had a basic idea of what activities I wanted my app to have. With the sole purpose of making my app easily comprehensible to average users, I started my journey. My app will have a simple login, signup and logout feature, a map view and lastly a forum style discussion page where users will be able to interact and share their experiences, along with photo sharing option. I came up with a basic, high-level design as show in figure 3 after doing a brief research on the options available to implement the app.

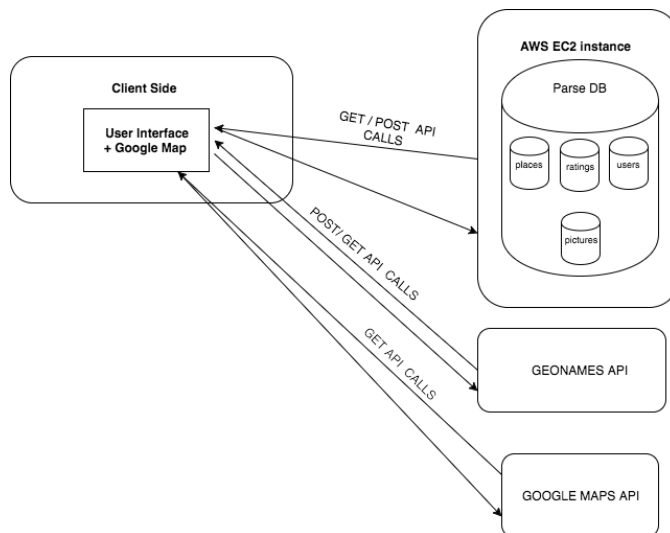


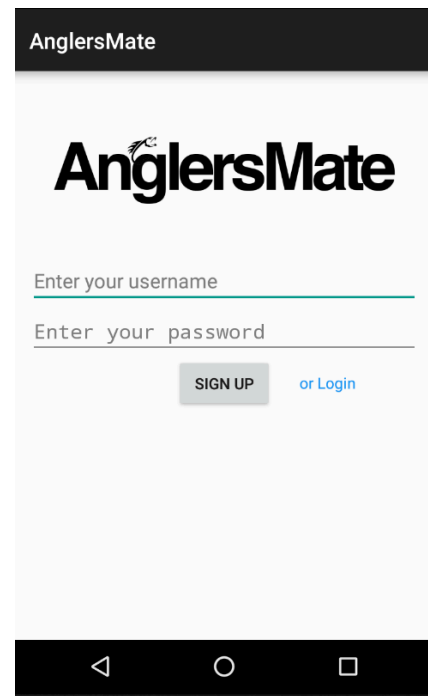
Figure 3 - High Level Architecture Design

3.2 UI

Having no experience in android front end, my app's interface isn't the greatest looking. I was aiming to make it look as simple as I could for people to easily get a hang of the app. I started with watching tutorials on learning android programming on YouTube and watched couple videos to figure out the basics of Android software development life cycle.

I dived right into it and made my first activity by creating the first layout file (MainActivity.xml), which will be the app's login/signup page. I made a basic logo to make the splash screen a little bit appealing.

Figure 4 - Splash Screen



3.3 Backend:

For backend I wanted to experiment with cloud storage. I decided to use parse open source server, which I will be hosting on free amazon web services' instance. This practice of having 3rd party deal with the backend of your application is becoming common practice and BaaS (backend as a service) providers are increasing in numbers rapidly.

3.3.1 Backend as a Service:

The increase in demand for BaaS services among the starts up and big enterprises is driven by enormous adoption of mobile and smart devices and also influx in the number

of cloud-based applications. Developers are also attracted by the rapid development and deployment rate catalyzed by BaaS.

3.3.1.1 - Advantages of using BaaS:

- 1) BaaS service providers provide services like monitoring active users, devices and events which assists the start-ups or even big enterprises to drive their usage analytics in this competitive market. It also provides tools to the developers to interact with their users and also enables users to communicate with each other using push notification services that usually come standard with most of the BaaS systems. The analytics provided by the BaaS systems help the developers in tracking the usage and also give them a better understanding of how to deliver an exceptional user experience across all the platforms and devices. ^[8]
- 2) Before BaaS developers had to incorporate individual APIs into their application for providing services like email notification, social network integration, push notifications and cloud storage etc. BaaS providers have automated these services under one platform. ^[10]
- 3) BaaS significantly reduces time when it comes to backend development and also when updates are required in the backend. Large companies don't have to wait months to implement a change request and/or deployment.
- 4) BaaS has a very low learning curve and demands very little knowledge and effort from the developer to use a BaaS system. This gives developers time to work on the front-end without having to worry too much about the back-end development.

This reduces cost as well, as the time required to development a full functioning app is reduced.

- 5) BaaS service providers are very efficient in accommodating high traffic. BaaS providers are keen about scalability to reduce the downtime for their clients. ^[8]
- 6) Lastly most of the BaaS providers have decent security protocols in place to keep their clients' data safe. ^[5]
- 7) Due to a high demand for BaaS services, there are many vendors popping up all across the world. BaaS is increasingly in demand, about 35% of the developers in North America have adopted BaaS for their application. ^[9]

3.3.1.2 Disadvantages of using BaaS:

1. The backend source code is hard to access when using BaaS systems.
Developers are given bunch of function and API calls that they are restricted to.
2. If as a developer you need to devise your own security policies, you are better off implementing your own back-end, as BaaS do not allow client's to modify security policies. ^[13]

The BaaS service I'm interested in is called Parse. Parse server is an offshoot of parse.com and it allows you to have a ready to go backend without having to code your database from scratch. Parse's Facebook owned and operated cloud application platform shutdown on Jan 28th 2017, since then Parse had teamed up with Heroku, Azure, AWS, Google cloud and MongoLab to provide its users to host their parse server on these infrastructure ^[2]. For this project I decided to go with amazon web services. AWS is a secure cloud service and provides a large range of infrastructure tools for the developers for example:

- 1) It allows you to send updates, notifications and promos to individual users, a segment of users or all of your users with a single push notification.
- 2) AWS has scalable, built-in data security features.

AWS being industry leader at the moment is highly dependable and always adding new and beneficial tools.

It is also worth mentioning that in

2016 magic quadrant for cloud infrastructure as a service, Gartner (world's leading information technology research and advisory firm) placed AWS in the “Leaders” quadrant and acknowledged AWS as the service provider with the furthest completeness of vision and highest ability to execute. [3]

Figure 5 – Gartner’s Magic Quadrant



3.3.2 - A little bit about Parse:

I have never worked with a cloud database so I wanted to take this opportunity to experiment with it. Parse is increasingly becoming famous amongst regular developers and enterprises. When Facebook acquired it in 2013 and in 2014 it was powering more than 500,000 apps. [3]

- 1) Since parse is based off Mongo DB it has a strong object-to-object mapping. You do not need to create tables for data storage. You just create objects that hold the data.
- 2) Parse has multi data type support.
- 3) User management is another aspect of parse database that made it a good choice for me. Parse has built in User control, it automatically provides ParseUser class that store information about the users. The User class comes equipped with methods like signUpInBackground and loginInBackground, which takes the grime work out of providing, sign up and login features for your application.
- 4) ParseFile class has built-in methods for storing multimedia objects like pictures, videos and audios.
- 5) Parse is free till your application starts to get more than 30 hits in 1 second. It gives room to the developers and startups to use parse without worrying about server side coding and deployment charges.

3.4 - Obstacles:

Before we dive in to this section it is important to understand what JSON object is and how it is attained through RESTful API. Processing of the JSON object and making use of it on an android device is explained later in this section.

3.4.1 REST:

REST stands for (representational state transfer). REST revolves around the resources, which is accessed through HTTP methods like GET, PUT, POST and DELETE. In the

REST architect, server gives REST client access to resources to use and present.

REST returns the information in various formats, like TXT, JSON and XML. JSON is now the most popular format used by the web services. ^[11]

3.4.2 JSON:

JSON stands for JavaScript object notation, it's not specific to JavaScript in any way.

It was developed to be able to parse information between websites or pages on websites but it has grown to become one of the standard ways of parsing information on the web. ^[14]

JSON object is bit like an array. It is a way of structuring data so that programming languages can process it to get the required information. JSON is important because APIs (Application programming interfaces) make use of it and it is great way to add interactive features to web or a mobile applications. ^[14]

JSON Object example:

```
var myObj = { "name":"John", "age":31, "city":"New York" };
```

3.4.3 Problem of finding water body:

One of the issues I faced while trying to decide whether or not I should deviate away from using Google maps and try a different API was finding a way to constrain users to put markers only on water bodies. It's obvious that Google has data on water, they are clearly differentiating water bodies in blue and land in green or brown in Google maps but I Google never thought about providing API that would differentiate between water and land for a selected Latitude and longitude.

For me this would have been useful, as I do not want users to place markers on places other than on a water body. One of the harder ways to accomplish this would be to check by pixel if a marker selected is really on water.

1) Receive the input in terms of Latitude and longitude and also current zoom.

2) Send this request to

**`http://maps.googleapis.com/maps/api/staticmap?center={latitude,
longitude}&zoom={current zoom}&size=1x1&motype=roadmap&sensor=false`**

The Google static maps API lets you convert Google map tile specified by location and zoom level into a static image. The service is a simple HTTP request, which returns the map as an image.

3) Detect the pixel color of the 1x1 static image returned by this HTTP request.

4) There are several APIs available for color detection. Upon quick research I found color tag. <http://apicloud.me/apis/colortag/docs/>

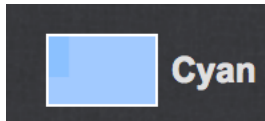
5) The idea is simple, you send them the image you want to find unique colors in through a HTTP GET request and they send you a JSON object with dominant color list.

On their demo page you can actually trial run the API. So I did. I took a random coordinate in Lake Ontario, plugged in the coordinate and zoom level in the Google static map image API and then inputted the image into the “colortag” GET request.

Google static image API GET request for a random point in Lake Ontario:

[http://maps.googleapis.com/maps/api/staticmap?center=43.595857,-
79.448979&zoom=10&size=1x1&motype=roadmap&sensor=false](http://maps.googleapis.com/maps/api/staticmap?center=43.595857,-79.448979&zoom=10&size=1x1&motype=roadmap&sensor=false)

This returned a 1x1 pixel image:



The ColorTag API returned the following JSON object:

```
{
  "tags" : [
    {
      "label" : "Cyan",
      "color" : "#A3CBFF"
    }
  ]
}
```

All I have to do is check what value tags[0].label has, if it is not Cyan then app displays a toast message stating something along the lines of “Marker cannot be placed, Not a water body”, if it is, app lets users place the marker and proceed with filling their information about the trip.

3.4.4 - Google Elevation library:

Another option was to use the Google’s elevation API. This API provides elevation data of a given location in terms of coordinates all over the world in JSON format. It also gives depth of ocean floor (in negative values). This approach is perfect for oceanic water bodies but when it comes to landlocked water bodies like lakes, dams, reservoirs and ponds that are above the sea level the elevation API fails.

```
{
  "results" : [
    {
      "elevation" : 1608.637939453125,
      "location" : {
        "lat" : 39.73915360,
        "lng" : -104.98470340
      },
      "resolution" : 4.771975994110107
    }
  ],
  "status" : "OK"
}
```

Figure 6 - Example of a JSON object return by Google Elevation API

3.4.5 - Google places API:

A Google places is a power API that gives mobile users contextual information about their where about. It provides users with rich, information about the local businesses and points of interest close to them.

Although Google places is a useful API for local businesses and street level searches it's not very effective for locating water bodies.

Before incorporating the API into my app I decided to trial the Google places API using their simple URL request, which returns a JSON object with nearby places names and information.

https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=44.458077,-79.315329&radius=2000&key=AlzaSyCbbmF4HW5v4S3dGe0CrJXuxdck_Tihr0c

In the above link, the characters after “key” are part of my unique API key provided by Google to use their maps services.

Unfortunately, Google places do not return effectively names of lakes, ponds, reservoirs, oceans and ponds as the API is more centered towards the local businesses.

3.4.6 - ReverseGeocoder.com:

Another service I tried was reversegeocoder.com. Their API works in a similar fashion (i.e. Return a JSON object) but unfortunately this API failed to retrieve water body information as well. As evident in the picture below, the pointer on top of the Atlantic Ocean just returns the coordinates and not the name of the place.

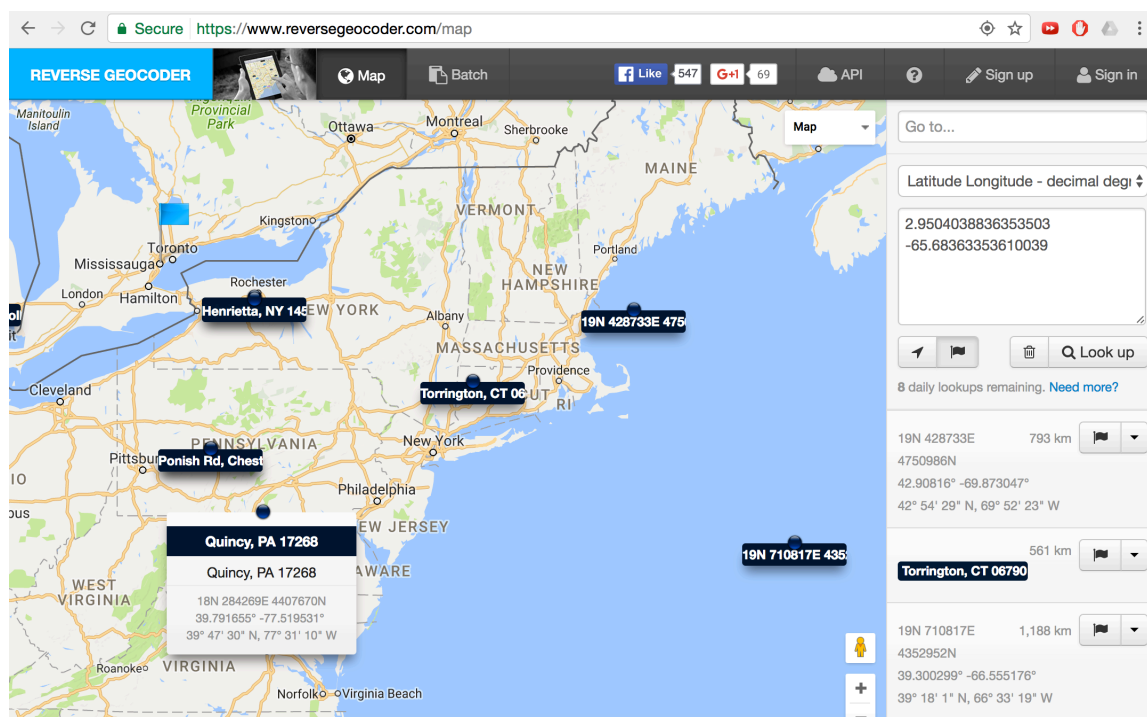


Figure 7 - ReverseGeocoder.com Example

3.4.7 - Geonames:

Another viable option was Geonames. It is a project that compiles free global geographic database. Geonames has made this database available for people to use through range of web services [17]. The information for places covers coordinates,

administrative divisions, postal codes, population, elevation and also time zones.

Geonames contain over 6 million unique places all over the world. Geonames gathers its data from national mapping agencies, national statistical offices, national postal services as well as the US Army. Geonames provides range of web services and APIs. Geonames APIs use RESTful protocol and responses and the APIs format range from XML, JSON, CSV and TXT. ^[12]

Getting the name of the places by longitude and latitude with geonames is simple, we start with the following URL and incorporate it in our way so that our app can process the returned JSON object. I tried the coordinates close to Algonquin Park, ON, as the areas has a large amount of lakes and camping spots and is possibly one of the best camping and fishing attractions close to Toronto and Ottawa.

<http://api.geonames.org/findNearbyWikipediaJSON?formatted=true&lat=45.9274622619106&lng=-78.2928778231144&radius=20&maxRows=6&username=malike&style=full>

Result:

```
{
  "geonames": [
    {
      "summary": "Radiant is an unincorporated place and former railway point in geographic Deacon Township in the Unorganized South Part of Nipissing District in northeastern Ontario, Canada. Radiant is located within Algonquin Provincial Park on Radiant Lake on the Petawawa River (...)",
      "elevation": 282,
      "feature": "city",
      "lng": -78.27416666666666,
      "distance": "6.6193",
      "countryCode": "CA",
      "rank": 65,
      "lang": "en",
      "title": "Radiant, Ontario",
      "lat": 45.98555555555556,
      "wikipediaUrl": "en.wikipedia.org/wiki/Radiant%2C_Ontario"
    },
    {
      "summary": "Odenback is an unincorporated place and former railway point in geographic Deacon Township in the Unorganized South Part of Nipissing District in northeastern Ontario, Canada. Odenback is located within Algonquin Provincial Park on Radiant Lake at the confluence of the tributary Little Madawaska (...)",
      "elevation": 281,
      "feature": "city",
      "lng": -78.31833333333333,
      "distance": "6.6638",
      "countryCode": "CA",

```

```

"rank": 65,
"lang": "en",
"title": "Odenback, Ontario",
"lat": 45.984722222222224,
"wikipediaUrl": "en.wikipedia.org/wiki/Odenback%2C_Ontario"
},
{
  "summary": "Camp Northway, formerly Northway Lodge, is the oldest summer camp for girls in Canada, and overall Canada's fourth oldest summer camp. Founded in 1906 and relocated to Algonquin Park, Ontario, in 1908. Camp Northway has maintained a distinctive ethos of simple camping, crafts, and drama (...)",
  "elevation": 451,
  "lng": -78.379124,
  "distance": "12.0581",
  "rank": 27,
  "lang": "en",
  "title": "Camp Northway",
  "lat": 45.837152,
  "wikipediaUrl": "en.wikipedia.org/wiki/Camp_Northway"
},
{
  "summary": "Unorganized South Nipissing District is an unorganized area in north-central Ontario, in the District of Nipissing. It is almost entirely within and includes most of Algonquin Provincial Park. (...)",
  "elevation": 416,
  "feature": "city",
  "lng": -78.41666666666667,
  "distance": "14.1899",
  "countryCode": "CA",
  "rank": 95,
  "lang": "en",
  "title": "Unorganized South Nipissing District",
  "lat": 45.833333333333336,
  "wikipediaUrl": "en.wikipedia.org/wiki/Unorganized_South_Nipissing_District"
},
{
  "summary": "Acanthus is an unincorporated place and former railway point in geographic Deacon Township in the Unorganized South Part of Nipissing District in northeastern Ontario, Canada. Acanthus is located within Algonquin Provincial Park on Cedar Lake on the Petawawa River (...)",
  "elevation": 323,
  "feature": "city",
  "lng": -78.40444444444445,
  "distance": "14.3673",
  "countryCode": "CA",
  "rank": 73,
  "lang": "en",
  "title": "Acanthus, Ontario",
  "lat": 46.030833333333334,
  "wikipediaUrl": "en.wikipedia.org/wiki/Acanthus%2C_Ontario"
},
{
  "summary": "Lake Traverse is the name of two adjacent unincorporated places, one a former railway point Source of coordinates in infobox. and the other a compact rural community, in geographic White Township in the Unorganized South Part of Nipissing District in northeastern Ontario, Canada (...)",
  "elevation": 250,
  "feature": "city",
  "lng": -78.07611111111111,
  "distance": "16.8648",
  "countryCode": "CA",
  "rank": 70,
  "lang": "en",
  "title": "Lake Traverse, Ontario",
  "lat": 45.944166666666666,
  "wikipediaUrl": "en.wikipedia.org/wiki/Lake_Traverse%2C_Ontario"
}

```

```
}  
}}
```

For the same coordinate values I tried Google places with the following URL:

https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=45.9274622619106,-78.2928778231144&radius=2000&key=AlzaSyCbbmF4HW5v4S3dGe0CrJXuxdck_Tihr0c

Result:

```
{  "html_attributions" : [],  "results" : [],  "status" : "ZERO_RESULTS" }
```

Google places couldn't find any close by places, as the area is mostly forests, lakes and campsites. It was obvious that Geonames is a better choice for suggesting users place names.

Now that I had a JSON object with sufficient data next step was to process it for my application:

3.4.8 - How to process JSON objects in Android.

AsyncTask class in android allows application to execute a task on the background thread; anything that requires a long time to finish execution (for example, downloading web content) can be performed with AsyncTask. It is a bad practice to engage Main UI thread in getting tasks done other than tasks related to the UI. AsyncTask has 3 main generic types namely params, progress and result. Asynchronous task in android have 4 steps, namely, onPreExecute, doInBackground, onProgressUpdate and onPostExecute.

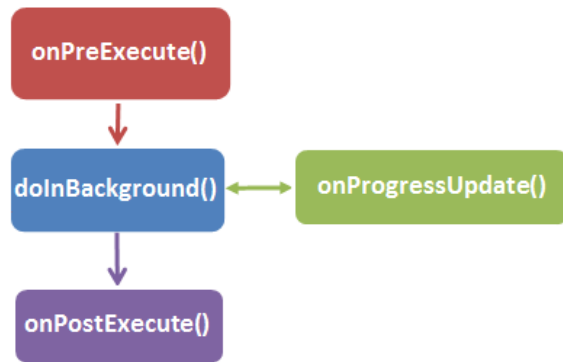


Figure 8- AsyncTask's Steps

3.4.8.1 - AsyncTask's generic types:

The three types used by an asynchronous task are the following:

Params: the type of the parameters sent to the task execution

Progress: types of progress units published during the background computation.

Result: the type of result of the background computation.

These generic data types are used by the `doInBackground`, `onProgressUpdate` and `onPostExecute` methods respectively. According to android reference `doInBackground` handles the first generic data type, which is params (String in our case, as we are looking for JSON string). Progress is usually set to Void (capital V as it is an object data type) and is handled by `onProgressUpdate`. Result is set to String (in case of getting of getting JSON string) and is handled by `onPostExecute`.

First step was to create an AsyncTask sub-class. Then I Overridden the `doInBackground` function call in that class to make it do what I wanted, which was to create an `HttpURLConnection` and get the JSON String. In the `onPostExecute` I extracted the required string and placed it in a `JSONObject` from `JSONObject` I

converted it into JSONArray and in the end I iterated through the JSONArray to get the required String, I then placed these String Array List. Using the following line of code: (variable 'places' is an Array List and variable s is the returned JSON String from the API)

```
JSONObject jObject = new JSONObject(s);
JSONArray geo = jObject.getJSONArray("geonames");
for (int i = 0; i < geo.length(); i++) {
    places.add(i, geo.getJSONObject(i).getString("title"));
}
```

Now that I have all the nearby places in an array I can use this list to populate my ListView in placeSuggestion class.

3.4.9- Nested scrolling:

My next problem arose very soon after. My suggestedPlace activity needed scrolling therefore its children were enclosed in a Scroll View tag in the xml file.

How do I scroll ListView inside a Scroll View tag?

NestedScrollView is used in such instances, where you have a widget that needs scrolling within ScrollView.

NestedScrollView is just like ScrollView but it allows scrolling on both parent and child. It is supported on both old and new versions of android.

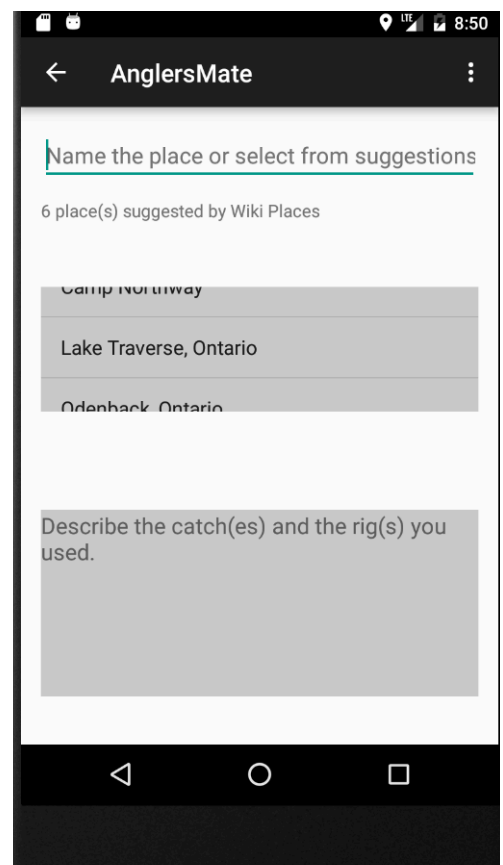


Figure 9 - Nested Scrolling for Relative Layout and for List View

The problem wasn't done yet. The next problem that arose was NestedScrollView interrupting with the ListView scroll. I had to set it up in a way that two scrolled independently and ListView only scrolls when it is touched.

```
placesList.setOnTouchListener(new ListView.OnTouchListener() {  
    @Override  
    public boolean onTouch(View v, MotionEvent event) {  
        int action = event.getAction();  
        switch (action) {  
            case MotionEvent.ACTION_DOWN:  
                // Disallow ScrollView to intercept touch events.  
                v.getParent().requestDisallowInterceptTouchEvent(true);  
                break;  
            case MotionEvent.ACTION_UP:  
                // Allow ScrollView to intercept touch events.  
                v.getParent().requestDisallowInterceptTouchEvent(false);  
                break;  
        }  
        // Handle ListView touch events.  
        v.onTouchEvent(event);  
        return true;  
    }  
});
```

This piece of code was all that was needed to fix this issue. Setting up touch listener on the ListView and calling requestDisallowInterceptTouchEvent(true) when ListView is pressed down and requestDisallowInterceptTouchEvent(false) when it is not being touched.

3.5 - Design Pattern:

Since Parse has provided client side with higher-level interface to interact with the database through API calls, it is safe to say that my application is following façade design pattern.

The user for my app can retrieve the information about a lake without knowing the inner workings of my database, cache and API client. The client side is simply making function calls like `getMarkers()` to retrieve places names and coordinates to be displayed on the map.

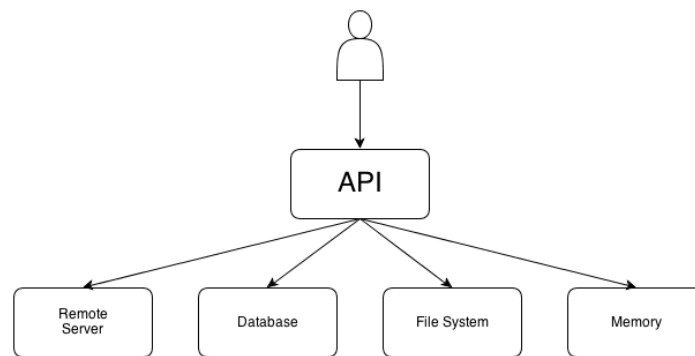


Figure 10- Client Making calls through API

3.6 - Development Model:

I followed the waterfall model for my design process. Despite having limited time to learn, implement and finish the report I stream lined the process of design and implementation, I didn't leave the testing and implementation for the very end, after developing each feature I constantly went back and tested them.

4 - Results and Validation:

I will evaluate the app by comparing how I did on each object initially mentioned in the project proposal.

Objectives	Evaluation
1) Develop easy to navigate, explore and browse-able user-interface/GUI.	<u>Successfully Completed:</u> I learnt the basics of android development to implement a decent UI design.
2) Develop login system for users.	<u>Successfully Completed:</u> Parse made this object really easy for me to complete. Parse provided me with user class called ParseUser that took care most of the functionalities pertaining to the user account management. It provided me with function calls for login, logout, signUp currentUser / user session.

<p>3) Develop user's dashboard.</p>	<p><u>Successfully Completed:</u></p> <p>My user dashboard is quite simple. It consists of a search EditText search bar for searching a particular water body on the map.</p> <p>And a Button to browse the map directly and browse all the markers.</p>
<p>4) Develop mechanism to populate the app with lakes names by kilometers range and location.</p>	<p><u>Successfully Completed:</u></p> <p>Reverse-geocoding to find name of a water-body doesn't work very effectively on Google Maps API. So I chose geonames API to reverse geocode by latitude and longitude.</p>
<p>5) Enable users to interact on a discussion board and share pictures.</p>	<p><u>Successfully Completed:</u></p> <p>Each fishing spot has its own discussion board where users can view other users' reviews and also add their</p>

	own.
6) Use Google maps API to display maps on the app.	<p><u>Successfully Completed:</u></p> <p>I used the Google Maps android API, which is a part of Google play services SDK to accomplish this task.</p>
7) Develop Pop-up window for displaying rating with comments from other users.	<p><u>Successfully Completed:</u></p> <p>I used the android RatingBar widget to get and store ratings. Comments are incorporated in the discussion board.</p>
8) Develop Pop-up window for recommended rigs and baits to use.	<p><u>Successfully Completed:</u></p> <p>This is also incorporated in the discussion board feature of the app.</p>
9) Develop a message board for each lake for users to discuss different strategies they used.	<p><u>Successfully Completed:</u></p> <p>Users' strategies are also part of the discussion board.</p>

4.1 Self-Evaluation:

4.1.1- Objective 1:

1) Develop easy to navigate, explore and browse-able user-interface/GUI.

This was a fun experience, as I had no prior exposure to android. It took me 2-3 days of doing extensive research to figure out essentials of android development. I started with developer.android.com and followed along. The tutorial on developer.android.com starts with an android studio project and as I was following the tutorial I found android studios straightforward and easy to grasp. One of the very first problems I initially had was not being able to start the app on the main login page. Upon research I found that intent-filter in the “AndroidManifest.xml” specifies the starting activity. It didn’t take long to learn the basics of developing a simple user-interface after that. I started off with a pilot app with a simple RelativeLayout that included a TextBox and a Button, the goal of the tutorial was to expose the students to the concepts of ViewGroups, activities, intents and `intent.putExtra()` to pass messages between the activities.

4.1.2- Objective 2:

2) Develop login system for users.

I wanted to experiment with NOSQL database and also wanted to experiment with BaaS, from the price point of view and also in terms of documentation, Parse was the choice of my BaaS for this app. I was even more delighted to find out that Parse comes with integrated User Management. Parse provides a specialized user class called `ParseUser` that handles much of user account management. `ParseUser` is a subclass of `ParseObject` and shares similar properties, except for additional attributes like

username, password and email. The signup functionality is also provided by Parse. Sign up call is an asynchronous call, it checks whether the entered username is unique or not, if the check passes Parse stores the new user. Parse also securely hashes user password in cloud using bcrypt.

4.1.3 - Objective 3: **Develop user's dashboard.**

The Dashboard contains a simple button and an EditText widget for searching a particular location. The button triggers a simple startActivity(Intent) call to the map while the search bar is implemented with android Geocoder class to get the Latitude and longitude of the desired place.

4.1.4- Objective 4: **4) Develop mechanism to populate the app with lakes names by kilometers range and location.**

The task is dependent on the location services being enabled and permitted by the user. The location manager provides the application with the user's location i.e. Longitude and Latitude. I already have the markers' location in the database. Whenever a user presses on a marker I calculate the distance between the two locations in KM and display it on the marker snippet. This is a very poor way of doing this as the distance between the two points is calculated linearly and doesn't account for actual street distance. Unfortunately, I found later on that Google Maps Distance Matrix API offers this service and returns distance between two points taking in consideration factors like tolls, traffic and speed limits.

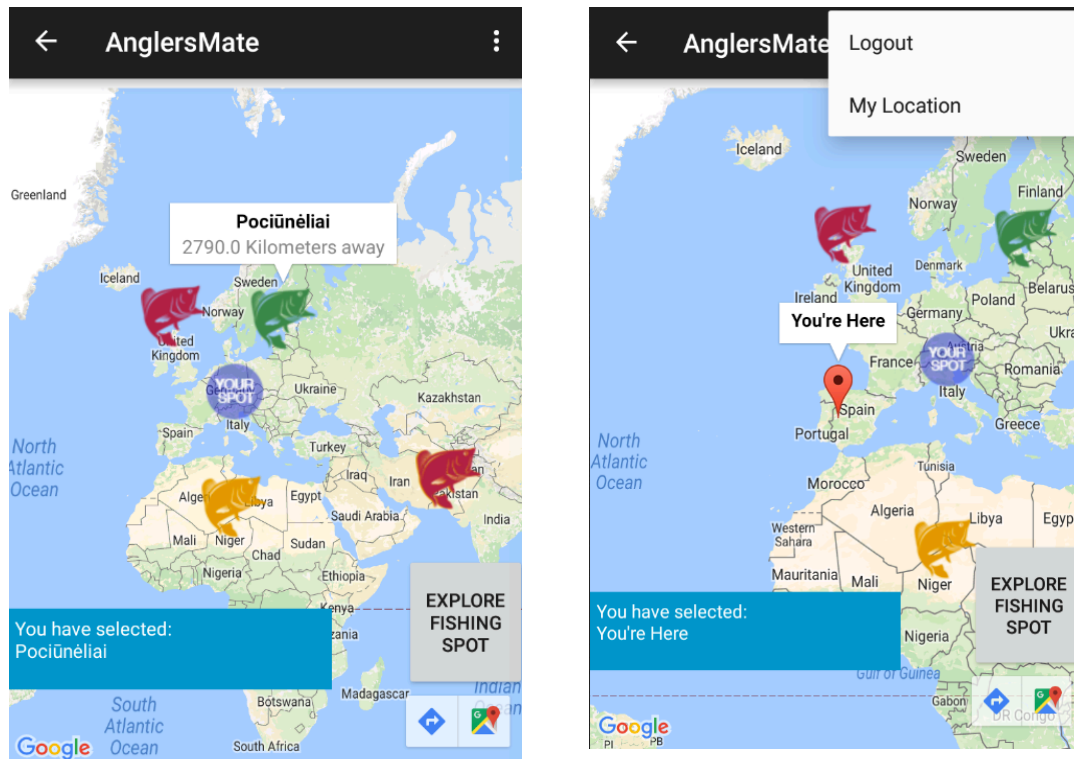


Figure 11 - App Screen Shot 1

4.1.5- Objective 5:

5) Enable users to interact on a discussion board and share pictures.

Discussion board was a fairly daunting task, mostly because I had no idea how to programmatically add content in my layout without defining the xml beforehand. I found some tutorial on YouTube to help me through the process. Some esthetical changes can be made to the discussion board but due to time constraints I couldn't add much to it visually. The figure below shows the discussion board. (The picture is a test picture).

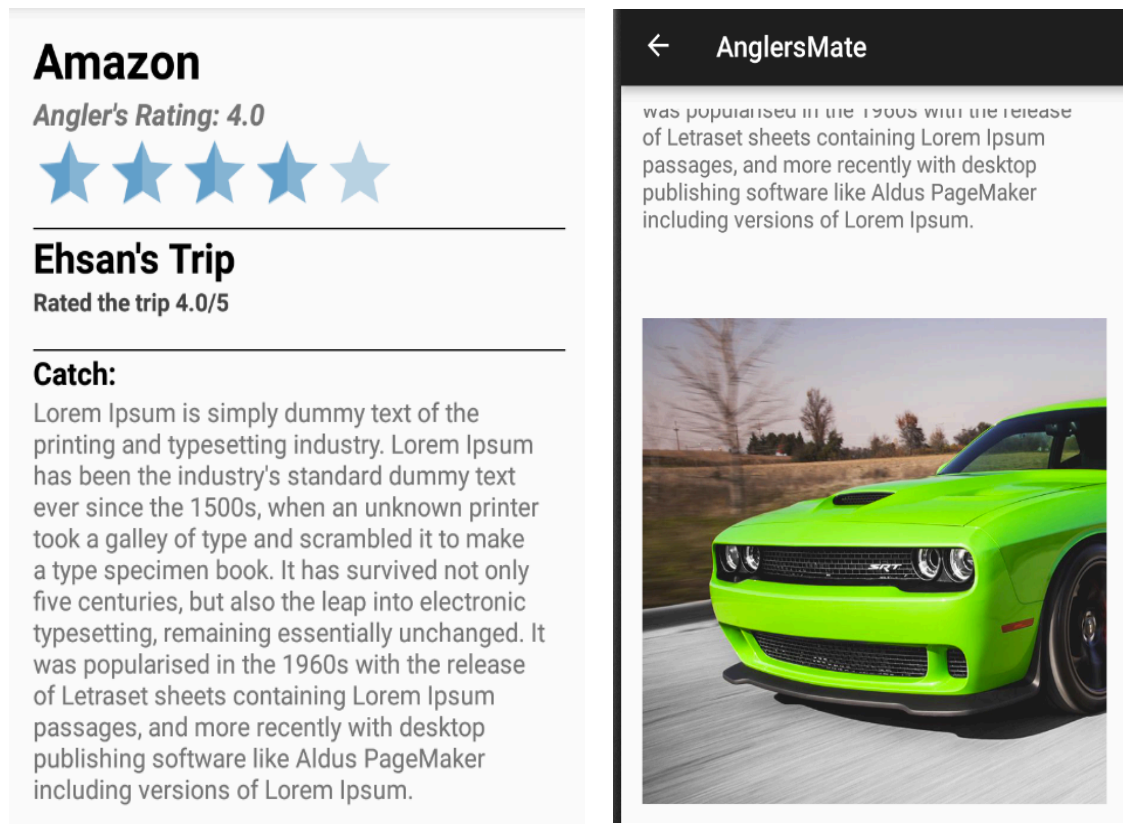


Figure 12 - App Screen Shot 2

4.1.6- Objective 6:

5) Use Google maps API to display maps on the app.

Displaying the map was a pretty straightforward process. To save a spot user has to long press and hold a location on the map and within 2-3 seconds they will be asked whether or not they want to save the spot or not.

Any spot that user save themselves will be shown as “YOUR SPOT” on the map.

Any spot that has low rating (<3) will be shown as a RED Bass.

Any spot that is average will be shown as a YELLOW Bass. (< 4 & >= 3.)

Any spot that has high trip average will be show as a GREEN Bass (<= 5 & >= 4) (figure 11)

6.0 - Conclusion:

Working on this project was a great learning experience. Given limited amount of time and juggling with other courses I don't think I was able to do my best but in the process of developing this app I explored areas I have never explored before namely BaaS, working with APIs and android development itself. All these were new territories for me. The idea for this project was unique but I think implementation could have been improved.

6.1 - Future work:

1. The app can definitely use a face-lift in terms of the UI. I'm going to continue experimenting with the front-end tools available for android, particularly JavaScript, as it seems to be everywhere.
2. There are couple other features, I would like to add to my app. First of all, I would like to add offline map feature. Most of the times, good fishing spots are far from the proximity of cities, hence chances of getting phone reception is measly. I'm not sure how to implement this feature but it is a needed feature especially for fishing in distant lake.
3. I would like to turn this app into a social media platform for anglers, just like Facebook anglers would be able to follow other anglers and will be notified of their trips and catches. This goes hand in hand with the next feature I'd like to add to this app "online chat", for anglers to directly message each other to invite them for a fishing trip or just to make new friends. I would like to add feature that

would help the anglers with organizing tournaments, guide tours or just a day out on the lake by coordinating with other anglers on the app.

7.0 - Work cited:

- [1] AccuWeather. "How Does Weather Affect Fish?" How Does Weather Affect Fish? N.p., n.d. Web. 04 Apr. 2017. <http://www.accuweather.com/en/features/trend/a-change-in-atmospheric-pressure/63489>
- [2] Bender, Jason. "Parse is Set to Shut Down. Now What?" CODE Online. N.p., n.d. Web. 04 Apr. 2017. <http://www.codemag.com/Article/1605021>
- [3] Crum, Chris . "Facebook's Parse Hits 500,000 Apps." WebProNews. N.p., 02 Dec. 2014. Web. 04 Apr. 2017. <http://archive.devwebprobr.com/2014/1204.html>
- [4] Dignan, Larry. "AWS, Microsoft seen rated top dogs in IaaS in Gartner's Magic Quadrant." ZDNet. ZDNet, 05 Aug. 2016. Web. 04 Apr. 2017. <http://www.zdnet.com/article/aws-microsoft-seen-rated-top-dogs-in-iaas-in-gartners-magic-quadrant/>
- [5] George Batschinski. "BaaS - Backend as a Service - Pros and Cons." Hashnode. N.p., 26 June 2016. Web. 04 Apr. 2017. <https://hashnode.com/post/baas-backend-as-a-service-pros-and-cons-cipwsw0ua07wa1753pbpbm1b6>
- [6] Mey, Dieter a., and SpringerLink (Online service). Euro-Par 2013: Parallel Processing Workshops : BigDataCloud, DIHC, FedICI, HeteroPar, HiBB, LSDVE, MHPC, OMHI, PADABS, PROPER, Resilience, ROME, and UCHPC 2013, Aachen, Germany, August 26-27, 2013. Revised Selected Papers. vol. 8374;8374., Springer, Berlin, 2014, doi:10.1007/978-3-642-54420-0.
- [7] MysteryTackleBox. "The 6 Best Fishing Apps For The Modern Angler." Mystery Tackle Box. N.p., 12 Feb. 2016. Web. 04 Apr. 2017. <https://mysterytacklebox.com/blog/best-fishing-apps/>
- [8] Mann, Stephanie. "Backend as a Service players, users and trends." SearchMicroservices. N.p., n.d. Web. 04 Apr. 2017. <http://searchmicroservices.techtarget.com/feature/Backend-as-a-Service-players-users-and-trends>
- [9] Market Research Firm. "Cloud/Mobile Backend as a Service (BaaS) Market by Service Type (Data and Application Integration, Identity and Access Management, Usage Analytics, Professional Service , and Support and Maintenance Service) - Global Forecast to 2020." Market Research Firm. N.p., n.d. Web. 04 Apr. 2017. <http://www.marketsandmarkets.com/PressReleases/baas.asp>
- [10] Riggins, Jennifer. "Why You Should Build Apps With An API Backend - BaaS | Nordic APIs |." Nordic APIs. N.p., 04 Nov. 2015. Web. 04 Apr. 2017.

<http://nordicapis.com/why-you-should-build-apps-with-an-api-backend-baas/>

[11] Rouse, Margaret. "What is RESTful API? - Definition from WhatIs.com." SearchCloudStorage. N.p., n.d. Web. 04 Apr. 2017.
<http://searchmicroservices.techtarget.com/definition/REST-representational-state-transfer>

[12] Riley, Duncan. "GeoNames: Wikipedia For Geographical Data." TechCrunch. TechCrunch, 26 May 2007. Web. 04 Apr. 2017.
<https://techcrunch.com/2007/05/26/geonames-wikipedia-for-geographical-data/>

[13] Yesweb. Advantages and Disadvantages of Backend-as-a-service - What is Baas? And Benefits of BaaS Solution - Wattpad. N.p., n.d. Web. 04 Apr. 2017.
<https://www.wattpad.com/181041496-advantages-and-disadvantages-of-backend-as-a>

[14] Smith, Ben, and SpringerLink (Online service). beginning Json. Apress, Berkeley, CA, 2015.

[15] "GoFree Hooked Fishing App." GoFree. N.p., n.d. Web. 21 Apr. 2017.

[16] Singhal, Manav, and Anupam Shukla. "Implementation of Location Based Services in Android using GPS and Web Services." International Journal of Computer Science Issues, vol. 9, no. 1, 2012, pp. 237-242.

[17] Zhang L., Yi J. (2012) A Brief Analysis of Geocoding. In: Wu Y. (eds) Software Engineering and Knowledge Engineering: Theory and Practice. Advances in Intelligent and Soft Computing, vol 115. Springer, Berlin, Heidelberg

