Mobile Cloud Computing Project Plan

Group 4

Anas Katib

Jeff Lanning

Zheng (John) Li

Dr. Lee

Application Name: “Look Around”

Revised Proposal:

In purpose of making our project useful to a common user and fun for us at same time, we came to the idea of an application that may utilize the GPS on the mobile devices to retrieve user’s location and then feed the device with location information.

At today’s society, GPS is not only used for navigation but also for many location related informational searches. There are a lot of applications already take advantage of the GPS,  such as TripAdviser when looking for a restaurant or Mobile Banking when finding a ATM. Because of the combination of mobile device and GPS, our current surrounding became more transparent to us than ever.  We think this is only the beginning for GPS; in the future we will be able to locate the closest taxi or an specific grocery item. We believe this is the trend, and many companies are picking up the trend. Many companies such as Google have already begun develop advertisement base on phone locations using mobile applications.

We feel current there is no application out in the market which can give a good overall interests of the location. For example if someone came to Kansas City for business or any other reasons for a short time, he/she may want to find out what to do after or between work such as bars, churches, activities etc… What we are looking for is at a glance, people are able to have a general idea of the interesting spots or information about the current area.

One of our objective is to learn the architectures and technologies that are used in mobile cloud computing. We believe through this project we will know how information is pulled and push from the cloud to the devices and vise versa. We will learn the  technologies to pull and organize information from the Big Data. In our case, how would Hadoop Distributed file system pull information from the internet source such as Google and other informational sites, then push down to our devices and what kind of calls does the device needs to make to display the location information.

Our Application would feature current location’s weather, top rated restaurant, bar or church at user defined range. We will use Hadoop Distributed file system in the IBM cloud instance we created in class as our data point with RESTful web services. The RESTful web services will be written in Java and deploy to the IBM HDFS using WAR file format. Our code will be combining information from a few REST APIs. The APIs we currently planning to use are google map APIs, Google Location APIs, Open Weather APIs and Yelp API. Our code in the HDSF will  store and process any information in its local database system. Our mobile application will also be written in Java and will run on Android devices. When user opens the application, the first page will provide information as the beginning of this paragraph stated, according to the current location of the device and the last user setting. When user clicks on the link for restaurants and such, it will redirect the user to the Google search of location name. Mobile devices would not save weather or any location information

There are two similar applications we found in the current market, AroundMe [1] and What is Around Me [1] on iOS or Android. AroundMe list the possible categories on the open page, and user must drill down to each category to search in the category. What is Around Me uses the Google Map to show locations, user also can go into list view as preference, user may touch/tab to the locations to show relevant location information. Our application would mainly be different in the opening page, where it will show informations from different categories based on user search radius. It will contain 5 days of weather, top three restaurants, bars and churches.

Our backup project is a friend’s activity calendar. It is like Google Circles [2] but with calendar and updates. Activities are mainly user defined, and one circle can have one activity. One person can in more than one circle. The calendar would update user’s friends who are only in the same interest/activity circle to know what the user is doing or going to do. Like Google Circle or Facebook, all calendar events for all the user will be held in the clouds. We will use HDFS as our cloud system to serve as our cloud. Information (or calendar events) updates would be would be pulled to the mobile device depends on user request or user defined frequency using RESTful web services. Mobile devices would held a copy of user’s calendar and events locally from the last updates and update references.

Bibliography:

[1] AroundMe: <http://www.aroundmeapp.com/>

[2] Google Circles: <http://www.google.com/+/learnmore/circles/>

Project Plan:

I. Introduction

In the purpose of giving a good overview of the current location to the user, we came to the idea of an application that may utilize the GPS on the mobile devices to retrieve user’s location and then feed the device with location information such as weather and restaurant.

II. Goal and Objectives

* Overall goal : Our Goal is create an mobile web application gives user general location information of user’s current location such as weather and restaurant.
* Specific objectives (problem statement) : We want the user has the ability to have an overview of the current location with one click. We feel current there is no application out in the market which can give a good overall interests of the location such as weather and restaurant under one page. All application we have found, needs to drill down into different categories, or use different application to get different information. We will use OpenWeather, Google Map & Location and Yelp API service to gather the information and save to the IBM Hadoop system. Mobile devices will be able to make one call the RESTful services we set up in the Hadoop we setup to retrieve and display current location information.
* Significance: There are websites such as iGoogle or MyYahoo, where people are able to custom some of the information feeds according to individual preference. Our application pave the way for customized location information look up under one page on mobile devices.

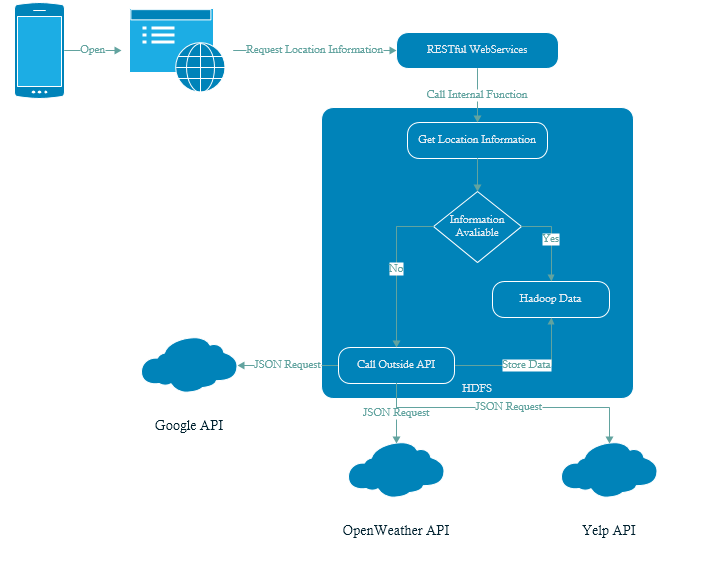
III. Project Background and Related Work:

As we descriptive in the Goals and Objectives, we could not find any application that are truly similar. There are two applications runs under Android and iOS named “Around Me” and “What is Around Me”, however they different from what we are looking for. “Around Me” search only under the individual categories such as hotels or restaurants, where our application emphasis obtain information at one glance. In the current state of the planning we are going to include only weather and restaurant information. The other application “What is Around Me” display hotel and photo informations on the map and in list mode. Information on a map be chaotic and uncategorized if the user is looking for specific categories. And again in list mode, user must view each categories on different page, therefore different from out one page, one glance concept.

IV. Proposed System

1) Requirement Specification

* Business Requirement
* Functional
  + Mobile Web Application should obtain GPS location of the device.
  + Mobile Web Application should display current location’s name.
  + Mobile Web Application should display the weather of device’s current location on the opening page.
  + When user tap the weather display, the application to direct user to a five day forecast of the weather.
  + Mobile Web Application should display three restaurants that is closest to the device and order them by closet on top.
  + Mobile Web Application should display the distance of the restaurant that is closest to the device.
  + When user tap the any restaurant on the display, the application will direct the user to a different display information restaurant. .
  + There should be a refresh button on the first page.
  + when refresh button is tapped, the page should refresh the page
  + Mobile Web Application should display a error message when it cannot talk to the web services
* Non-functional
  + The Page refresh should be very responsive (under 30 sec)
* Technical & Architectural Requirements
  + Mobile Web Application would be run on mobile web browser
  + Mobile Web Application will be talking to RESTful Web Services using JSON.
  + RESTful Web Services would be hosted on the Hadoop Master Node
  + Our Hadoop will request weather info from Open Weather API.
  + Hadoop will request restaurant info from Google Location and Yelp API.
  + Hadoop will request distant info from Google Map API
  + Weather info will be saved in Hadoop.
  + Web Service in Hadoop will only send the information mobile information need in one response
* Business Process/Workflow analysis



2) Framework Specification: Build an overall system model

* The overall system will be consist of three layers: a data layer consisting of a Hadoop master node and data node, a service layer interfacing with Hadoop, and a client layer for the user interface. Hadoop is a framework supporting MapReduce dealing with large amounts of data and supporting a distributed architecture. Our Hadoop layer will consist of a master node and data node. The data node will contain the data we will import into Hadoop and the master node will serve as the interface/controller to manipulate the data. Our server will utilize the RESTful architecture pattern and reside on the master Hadoop node. It will serve as the bridge between the client and our Hadoop database by providing transactions for the client to call. Our service will also support transactions to call into existing services, such as for weather, Google Maps, and restaurant information. Some of the data from these existing services will be inserted into our Hadoop database on the fly so it can be retrieved more quickly later. The client will be web-based using JQuery Mobile, HTML5, CSS3, and AJAX. The purpose of a web-based client is so that it is platform independent - such that it can be accessible by multiple mobile devices, tablets, and the desktop computer. If time permits, we may consider adding a light-weight android client to serve as a wrapper to our web site.

· Assumptions and Principles

* Our architecture will rely on data being pre-imported into Hadoop for performance reasons. Our application will be focused on weather and restaurants/bars in the local Kansas City area, so we will gather existing data from the public domain and import it into our Hadoop database. This is done because importing data into Hadoop is slow (~10 seconds) and will reduce the burden when the user initially runs our application.

· Methodologies and Algorithms (MapReduce)

* The principle algorithm we will use will be MapReduce since Hadoop is the framework built on top of it. Hadoop will manage much of the overhead for us, but we will be required to use MapReduce algorithms. We will using MapReduce algorithms in Hadoop for the following use cases:

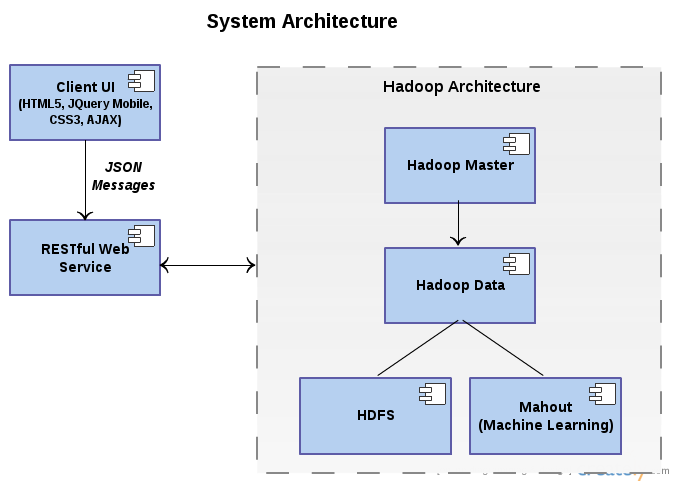
1. Aggregating weather data including temperature, sky conditions (cloudy, sunny, raining, etc.), precipitation, and possibly forecasts into the future.
2. Aggregating restaurant/bar data including restaurant names, style of food, locations, ratings, and hours.

These main use cases may be divided into smaller MapReduce algorithms each handling one aspect of the data.

· Design Pattern (if any)

* We will use the Bridge pattern for our service by separating the interface for our API with the underlying implementation and for our business/data objects. This will be useful if in the future we decide to have a different implementation for different devices or use multiple sources of data. It will also help us write unit tests allowing us to provide a mock implementation of business objects.

· System Architecture Diagram



3) System Specification: Identify Primary Services

· Existing Services: Name, Description, URL

**Name:** Open Weather

**Description:** Open Weather is a free weather service that provides an API for obtaining the current weather or forecasted weather. For the current weather, it provides the ability to retrieve weather by city name, geographic coordinates, or city ID.

**URL:**<http://bugs.openweathermap.org/projects/api/wiki/Api_2_5_weather>

**Name:** Yelp

**Description:** Yelp provides a search service to obtain data about restaurants, bars and other food and entertainment establishments. The service contains a detailed API that can retrieve restaurants, their food styles, ratings, and distance from a specified location.

**URL:**<http://www.yelp.com/developers/documentation/v2/search_api>

**Name:** Google Maps

**Description:** Google Maps provides the ability to display a map based off location (latitude/longitude), and directions for locations such as directions from the user’s current location to a restaurant/bar.

**URL:**<https://developers.google.com/maps/documentation/javascript/reference>

**Name:** Google Places

**Description:** The Google Places API is a service that returns information about Places — defined within this API as establishments, geographic locations, or prominent points of interest — using HTTP requests. Place requests specify locations as latitude/longitude coordinates.

**URL:**[https://developers.google.com/places/documentation/](https://developers.google.com/maps/documentation/javascript/reference)

· BigData source:

**Name: Factual**

**Description:** Factual’s location platform enriches mobile location signals with definitive global Data, enabling personalized and contextually relevant mobile experiences. Built from billions of inputs, the data is constantly updated by Factual’s real-time data stack.

**URL:**[**http://factual.com/**](http://factual.com/)

· New Services to be built:

Service Specification:

**Name:** GetLocationInformation

**Operational Description:** This service will take a GPS location and in turn call the API services above, or grab the data from HDFS.

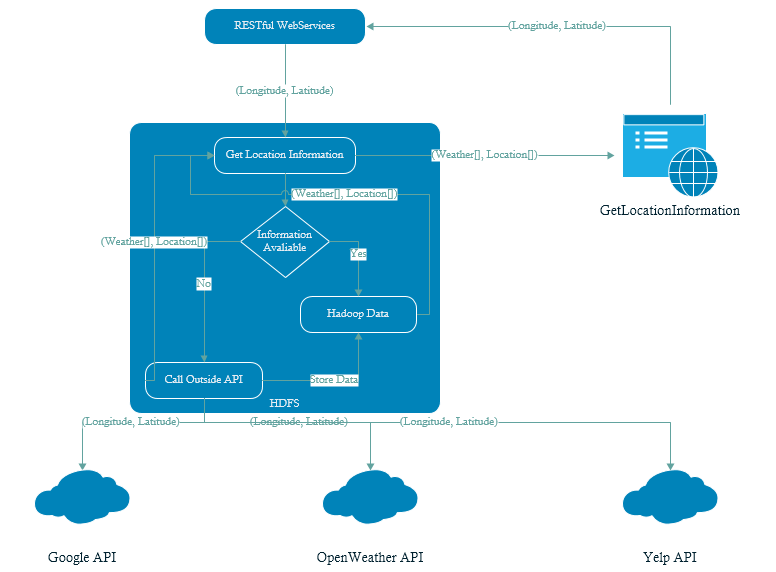
**Input/output for services:**

* Input**:** Latitude(double), Longitude(double)
* Output: Array of Weather Object, Array of Restaurant Object.

**Constraints/exceptions:**

* Constraints**:** Valid Latitude(double) and Longitude(double)
* Exceptions: Invalid Input Exception

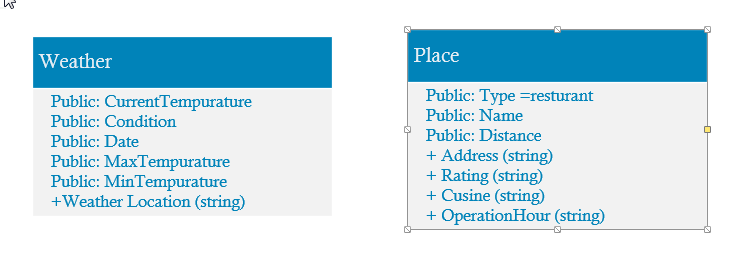
**Service flow/alternative flow:**



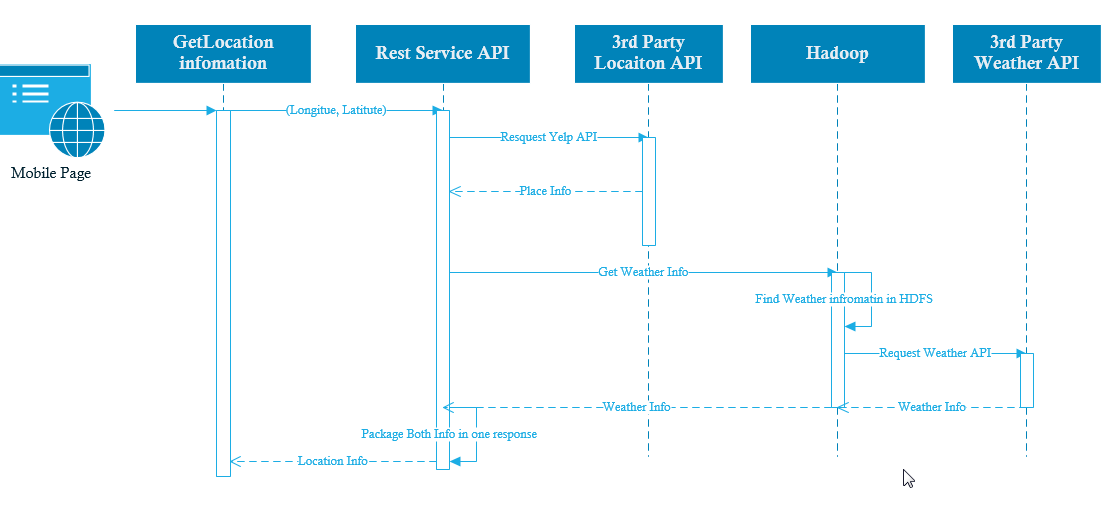
**Priorities**: Critical, and difficulty is medium.

For each service, specify the followings:

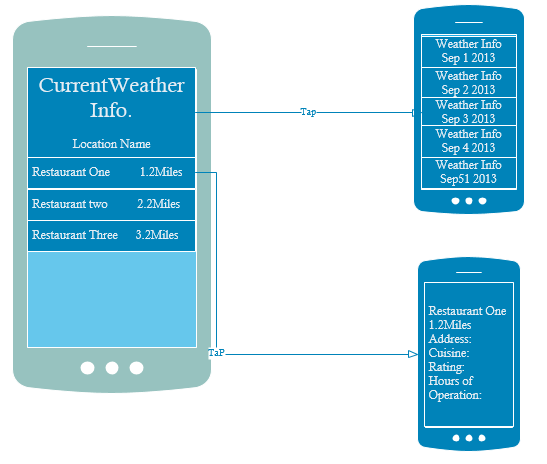
· Class diagram



· Sequence diagram



Design of Mobile Client



V. Plan by Services (using ScrumDo)

<https://www.scrumdo.com/projects/project/mobile-cloud-comp-project/>

VI. Risk management

* Internet Connectivity
* Web Service are not available or API is changed
* Location is not found.
* Hadoop is failing to respond in time.

VII. Bibliography

[**http://www.youtube.com/watch?v=yjPBkvYh-ss**](http://www.youtube.com/watch?v=yjPBkvYh-ss)

<http://www.youtube.com/watch?v=yjPBkvYh-ss>