TOUR LOGGER

ASE Project Document Phase 3

04/13/2015

Team Members:

1. Rajesh Kapa
2. Raghu Vital Gummadi

1. Objectives

In this Increment, there are several objectives which are to be achieved are described below. They are,

a. Making a WCF service for including trips and expend it in our android application TOUR LOGGER.
b. Making a WCF service for the displaying of trips and expend it in our android application TOUR LOGGER.
c. Using adjusted Google Places API to recover the list of places.
d. Retrieving the list of Attractions present in the particular place or trip added by end user.
e. Retrieving the list of hotels and restaurants present in the particular place or trip added by the end user.
f. Using DatePicker device for android with a specific end goal to have a Start date and end date for a trip and making end dates available only after the start dates.
g. Creating a few activities with altered activity bar and background pictures.

2. Import Existing Services/API

The services that are used for this increment are

Service1: WCF Service using Microsoft visual studio for the addition of trips.

Service2: WCF Service using Microsoft visual studio for displaying the trips.

Service3: Modified Google places API is utilized for showing the list items of places when one string of a place or a part of the place is entered in search box provided.

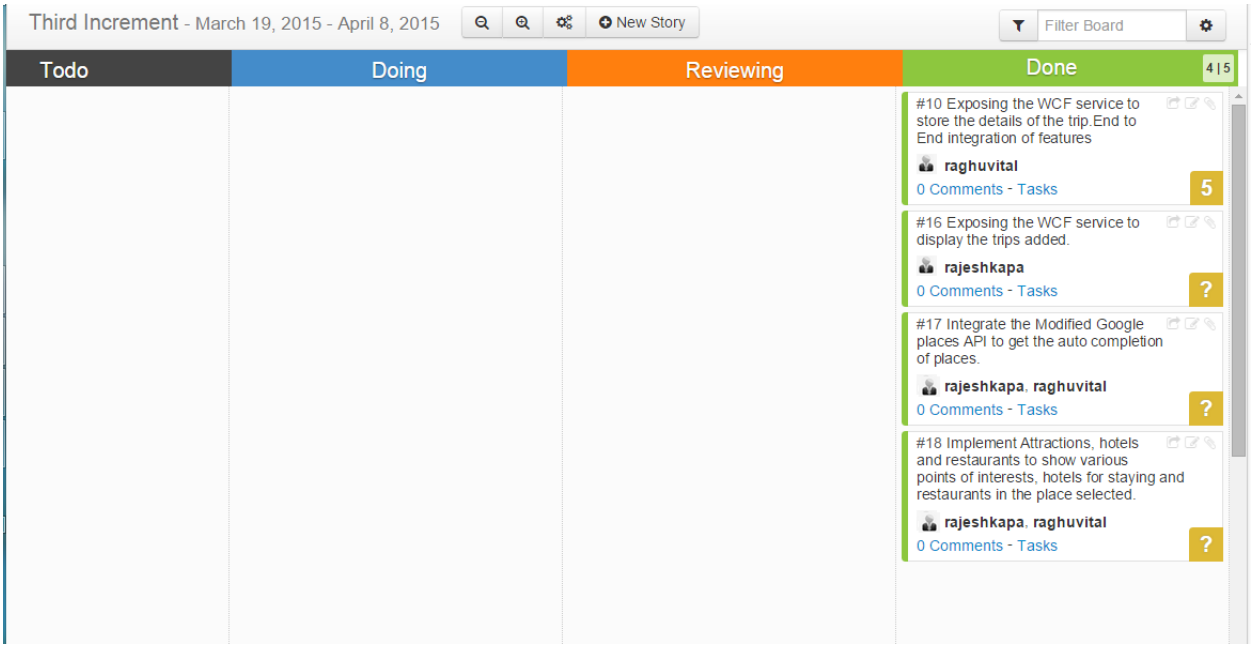
3. Detail Design of Services

3.1 Write User Stories /Use Case (Using ScrumDo – Include the screenshots)

Below are the four stories that are implemented in iteration 3

1. Exposing the WCF service to store the details of the trip.
2. Exposing the WCF service to display the trips added.
3. Integrate the Modified Google places API to get the auto completion of places.
4. Implement Attractions, hotels and restaurants to show various points of interests, hotels for staying and restaurants in the place selected.

Screenshots of the ScrumDo.



3.2 Service description

Tripadd service: This a WCF service, when the end user searches for a place and give the start dates and the end dates and save the details, this service is evoked. The values such as the place id, trip name, latitude and longitude are saved for a particular user with a user id.

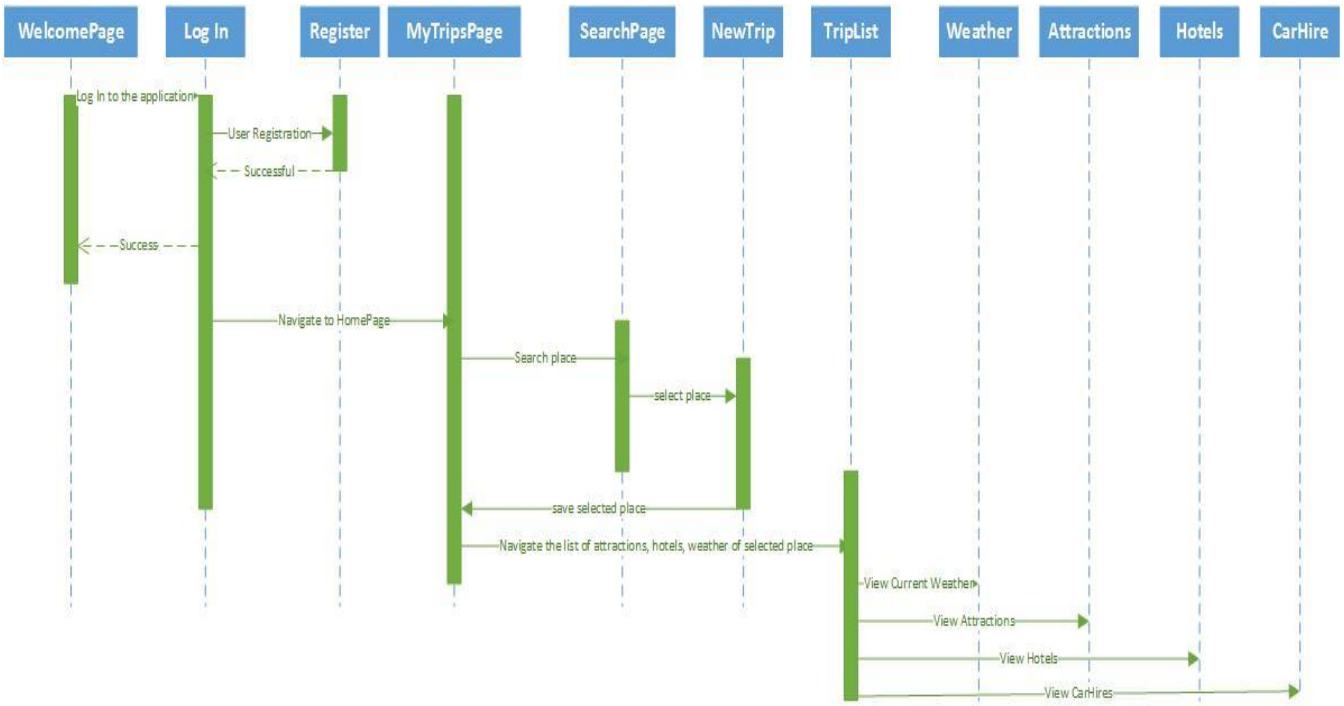
Tripdisplay service: This is also a WCF service, when the end user navigates to his mytrips page, this service is called and the trips or places that are added by the end user will be displayed.

Google Places API: Through this API, end users can utilize a few highlights like details of a place which demonstrates more definite information about the chose place, Place add permits to include the information show in the google database with our application, place hunts gives back a list of places relying on the client info or the current location, place autocomplete naturally finishes the name of the place as the client sorts in the search box.

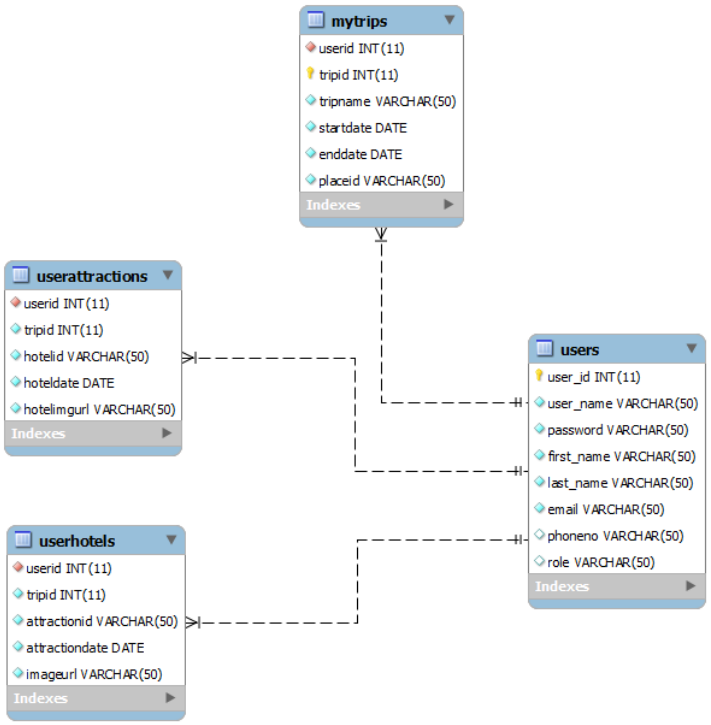
In this increment, we are utilizing the place autocomplete highlight of the google places API. At the point when the client enters a string with least of one letter, this feature demonstrates the accessible list of cities present beginning with that entered string. In this way, we can choose one place in that rundown.

Also using this google places API, several elements in the returned JSON object are stored in the database. With the place id, several attractions that is points of interests in that particular place are retrieved and shown in a list view. Also the hotels present in that particular place are shown. So, this google places API is modified and used to fetch and retrieve the information needed.

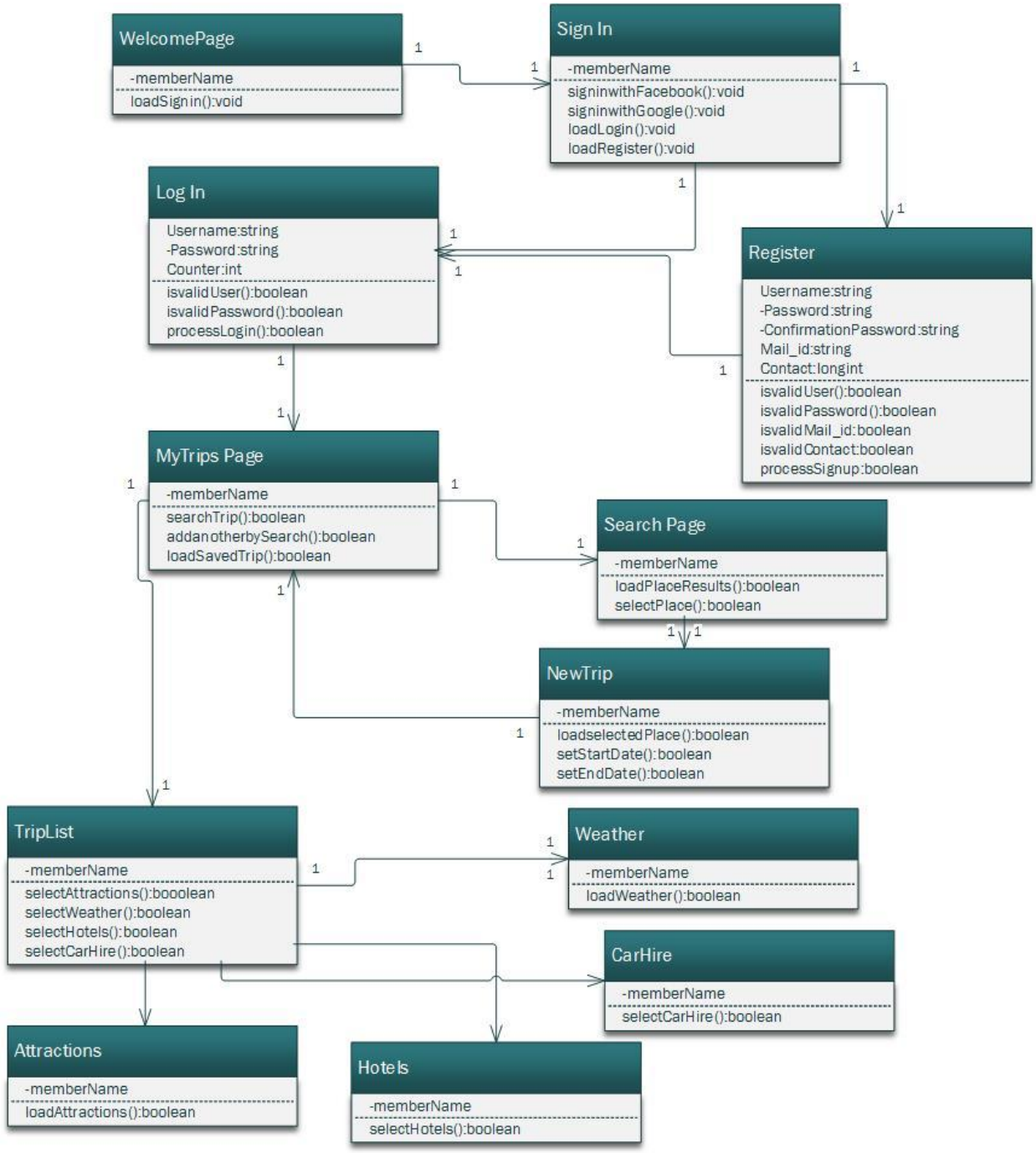
3.3 Sequence diagram (using Visio)



DATABASE SCHEMA:



3.4 Class diagram (using Visio)



3.5 Design of Mobile Client Interface

Our TOUR LOGGER application's customer Interface has a few screens which are outlined easy to use. These are SearchLocation Page, NewTrip Page, MyTrips Page, TripList Page, Attractions Page, Hotels Page and Weather Page. In the wake of Logging in the client cooperates with the MyTrips Page that has choice to include a place and demonstrate a place that is chosen. The SearchLocation page

helps client to look for places and in the wake of selecting a place he will be explored to another page NewTrip where he inputs the begin date and end date and spares his outing. At that point the insights about the spot and dates are spared in the database. The spared trip will be shown in the MyTrips page and for this excursion there will be a TripList Page which has a list see that shows a few substance like attractions, hotels and weather and so forth. The weather page shows the climate of the spared city by utilizing the OpenWeatherMap API.

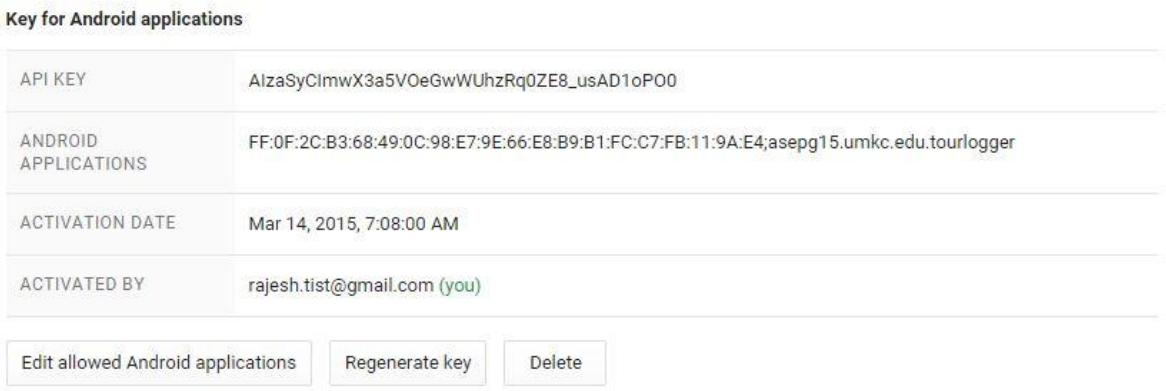
3.6 Design of Unit test cases (using NUnit tool)

Unit Test cases are planned in NUnit tool. Experiments are composed for login, enrollment, Tripadd and Tripdisplay services. They are designed and tried in positive situations.

4. Implementation

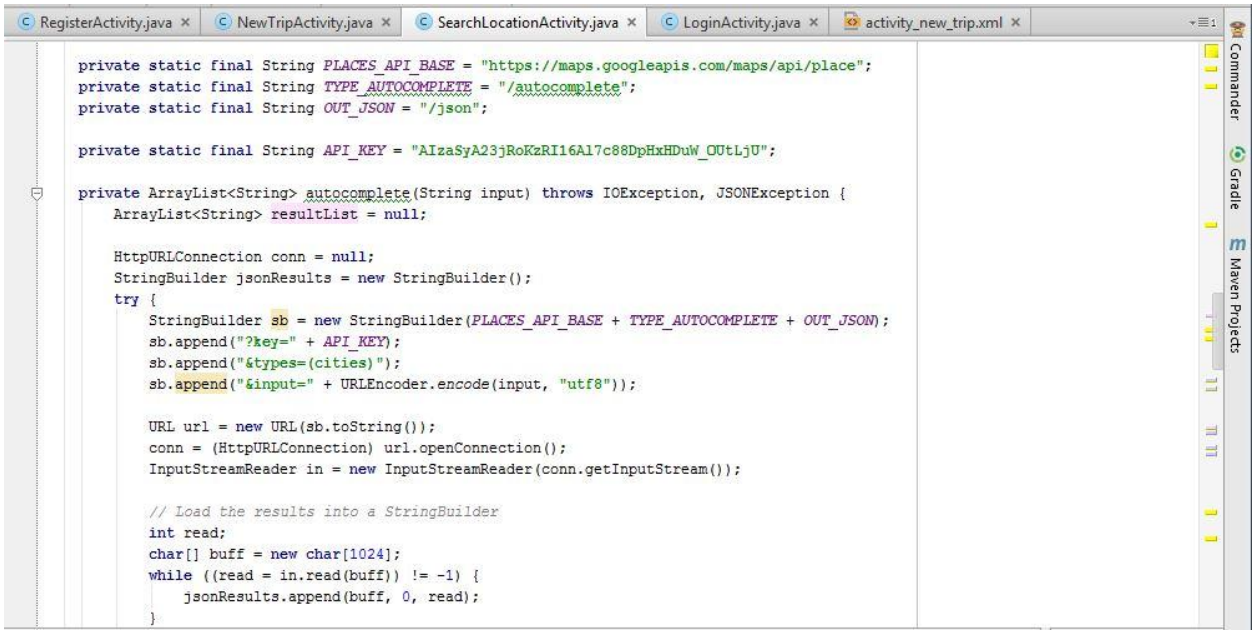
4.1 Implementation of REST services

Underneath REST Services are executed in Tour Logger project are actualized as beneath. Created google API key and utilized as a part of service call.



a. Google Places REST Service:

By using this API key auto generation of cities was implemented and tested manually. Below screenshot shows the code.




```
private static final String PLACES_API_BASE = "https://maps.googleapis.com/maps/api/place";
private static final String TYPE_AUTOCOMPLETE = "/autocomplete";
private static final String OUT_JSON = "/json";

private static final String API_KEY = "AIzaSyA23jRoKzRI16Al7c88DpHxHDuW_OUtLjU";

private ArrayList<String> autocomplete(String input) throws IOException, JSONException {
    ArrayList<String> resultList = null;

    HttpURLConnection conn = null;
    StringBuilder jsonResults = new StringBuilder();
    try {
        StringBuilder sb = new StringBuilder(PLACES_API_BASE + TYPE_AUTOCOMPLETE + OUT_JSON);
        sb.append("?key=" + API_KEY);
        sb.append("&types=(cities)");
        sb.append("&input=" + URLEncoder.encode(input, "utf8"));

        URL url = new URL(sb.toString());
        conn = (HttpURLConnection) url.openConnection();
        InputStreamReader in = new InputStreamReader(conn.getInputStream());

        // Load the results into a StringBuilder
        int read;
        char[] buff = new char[1024];
        while ((read = in.read(buff)) != -1) {
            jsonResults.append(buff, 0, read);
        }
```

b. <u>Using exposed WCF services:</u>

Login service to check whether details are present in Database or not.

```
}
else {
    UserLoginCheck logincheck = new UserLoginCheck();
    logincheck.execute(new String[]{"http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client/MPG15/TourLoggerService/Service1.svc/login/user/"+ un + "/" + pw});

}
```

Registration of user using REST WCF service

```
}
else
{
    UserRegistration regcheck = new UserRegistration();
    regcheck.execute(new String[]{"http://kc-sce-cs551-2.kc.umkc.edu/aspnet_client/MPG15/TourLoggerService/Service1.svc/register/user/"+ fn +

}
```

Adding of Trip details using REST WCF service

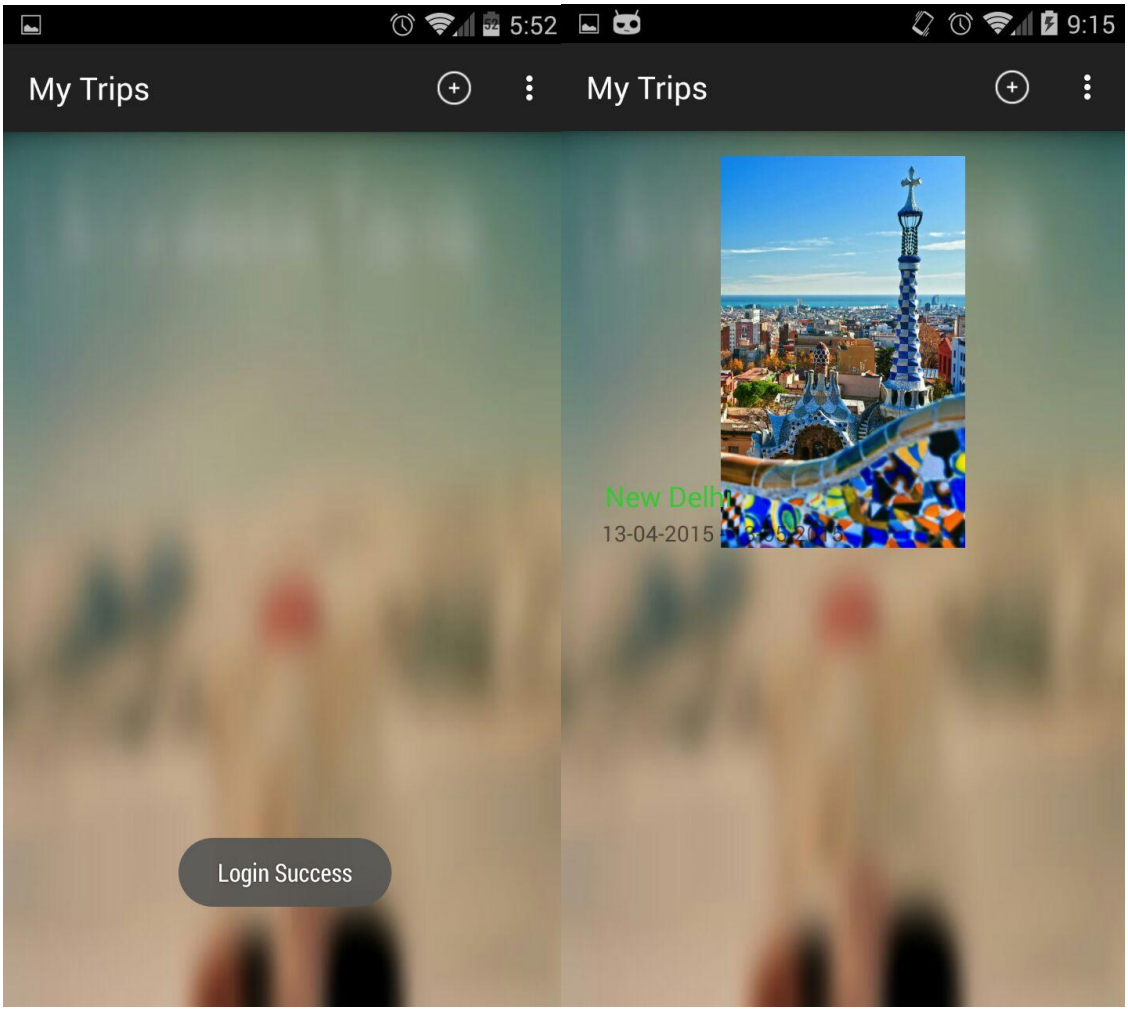
```
[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json, UriTemplate = "TripDetails/{userid}")]

public TripObject GetTripDetails(string userid)
```

c. openweathermap api is used to get weather of particular place. Below is the code used for consuming the existing service:

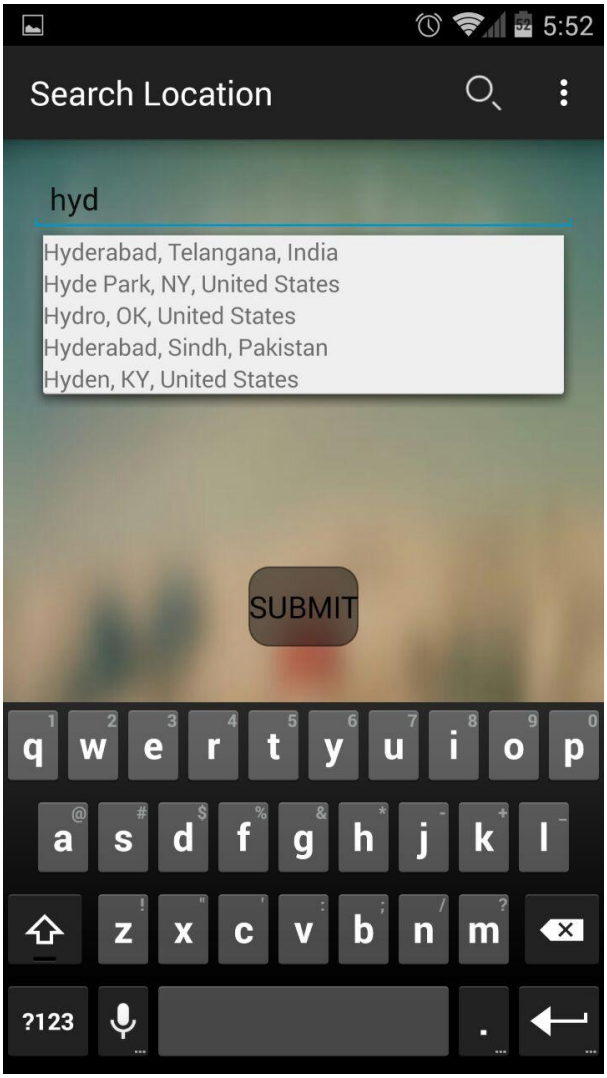
```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

import org.json.JSONObject;

import android.content.Context;
import android.util.Log;

public class RemoteFetch {

    private static final String OPEN_WEATHER_MAP_API =
            "http://api.openweathermap.org/data/2.5/weather?q=%s&units=metric";

    public static JSONObject getJSON(Context context, String city){
        try {
            URL url = new URL(String.format(OPEN_WEATHER_MAP_API, city));
            HttpURLConnection connection =
                    (HttpURLConnection)url.openConnection();

            connection.addRequestProperty("x-api-key",
                    context.getString(R.string.open_weather_maps_app_id));

            BufferedReader reader = new BufferedReader(
                    new InputStreamReader(connection.getInputStream()));

            StringBuffer json = new StringBuffer(1024);
            String tmp="";
            while((tmp=reader.readLine())!=null)
                json.append(tmp).append("\n");
            reader.close();
```

4.2 Implementation of user interface (Mobile Apps)

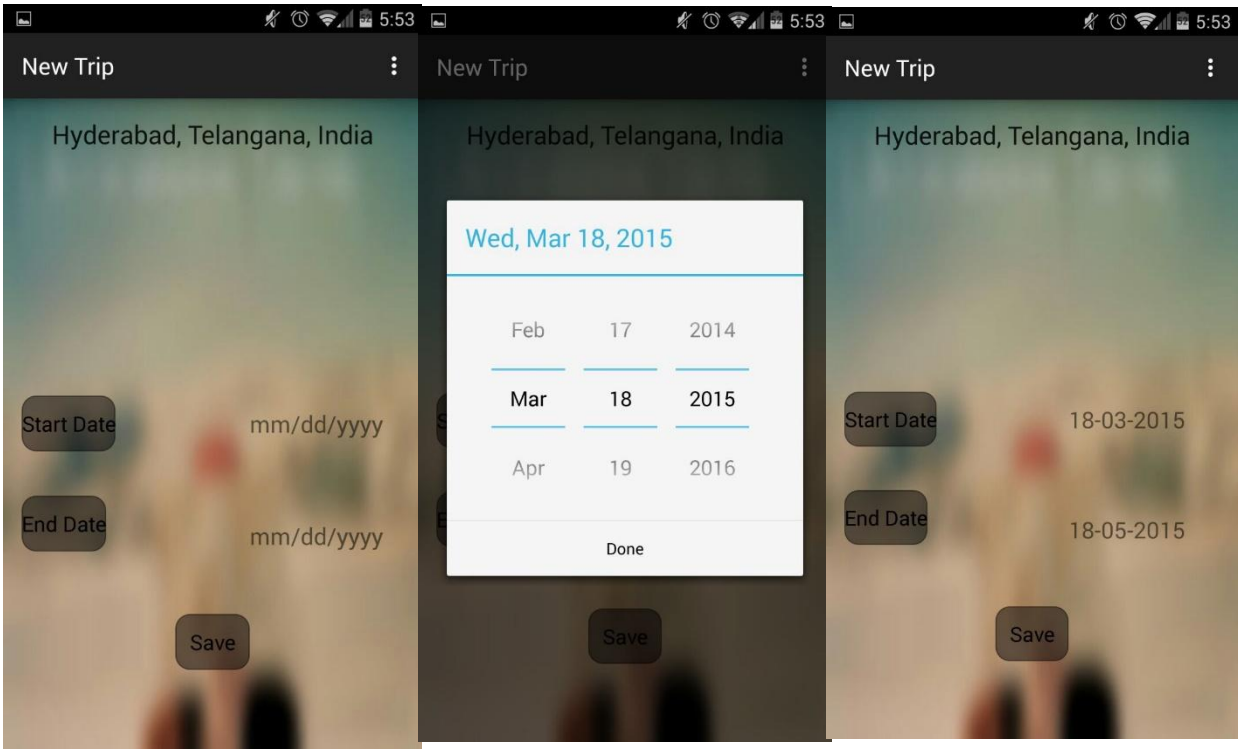
a. MyTrips Page: After effective login, a toast will be shown as login achievement. Furthermore, this action has a choice in the activity bar with an in addition to image on it. When it is clicked it goes to another activity SearchLocationActivity. Also, in the wake of including and saving the trip, the trips are added in this page. These subtle elements are put away in the database alongside the client id since these are referenced to a unique user id. The following is the screen of this MyTrips Page.



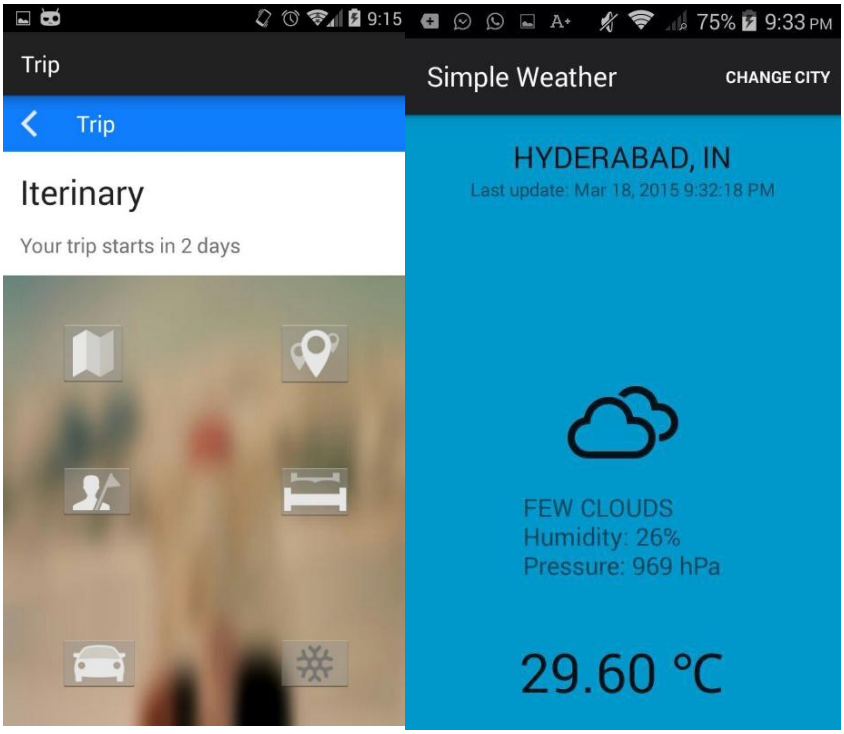
b. Search Location Page: This page has an action bar and on click on it empowers the client to give an info. At the point when client enters at least one string, the inquiry shows a list of better places identified with the search string entered by the client. This methodology includes calling the google places API feature and having the JSON item values and setting them in a listview. The following is the screenshot of this page demonstrating a few outcomes when a string is entered.



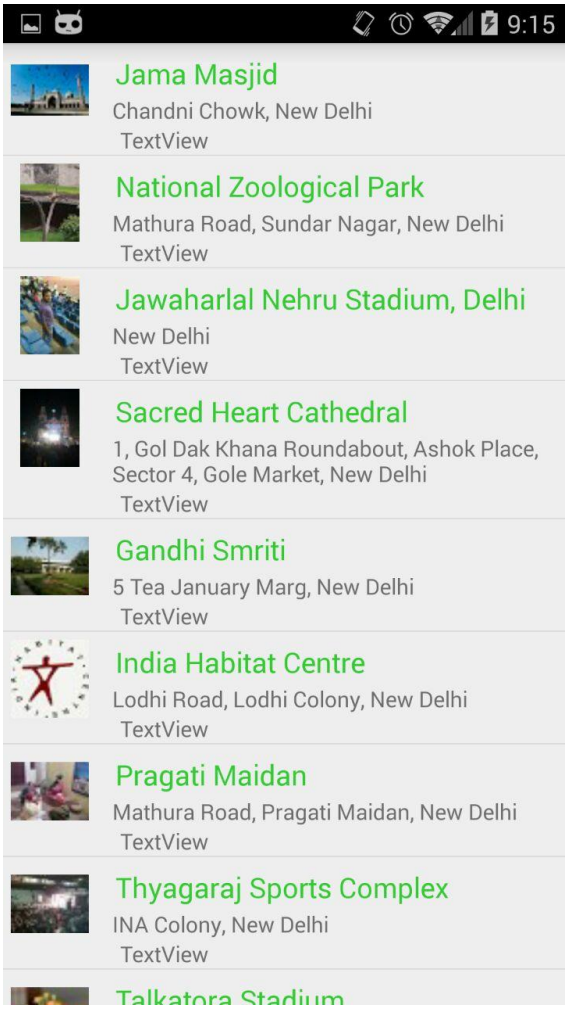
c. NewTrip Page: This Page takes the place chosen in the past page and has choices to choose begin date and end date. At the point when the client clicks on the begin date and end date, a dialog shows up which is a date picker dialog. It is an instrument in android. It permits client to choose date with a decent interface. At the point when a date is chosen, the dates are spared in the database and are demonstrated the content view in that page. The following are the screens of the new trip page.

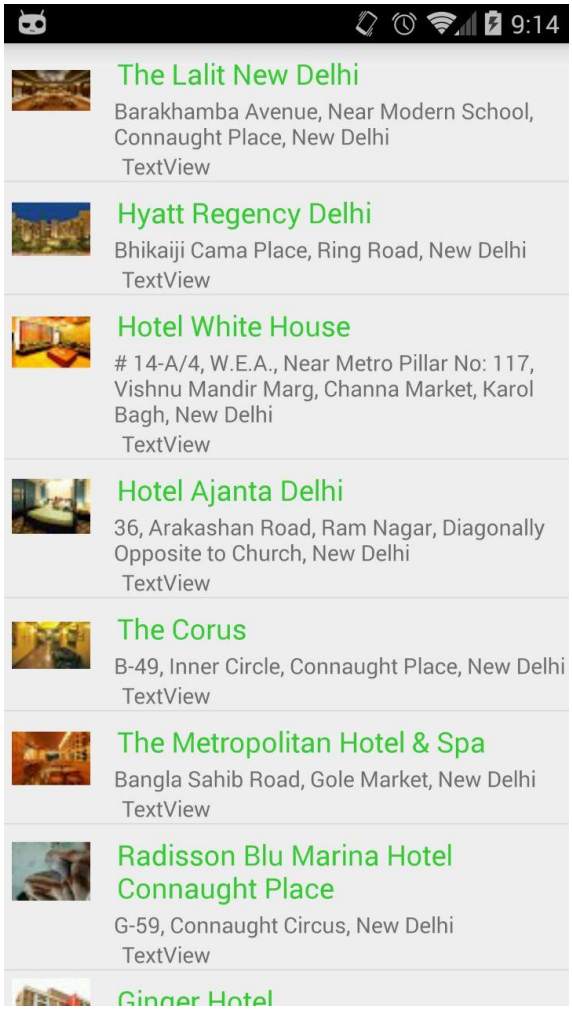


d. <u>TripList Page and Weather Page:</u> In the wake of selecting the spot, the attractions, hotels, weather all in listview are shown. Also, on click of weather shows the current climate of the chosen place. The following are the screens of the TripList Page and Weather Page.



e. Attractions Page and Hotels Page: These pages consists of attractions that is points of interests and hotels page consists of available hotels and restaurants in the selected place by the end user. Below are the screen shots of both the pages for New Delhi city.





4.3 Implementation of test cases

The Following Test cases are executed in Nunit Tool.

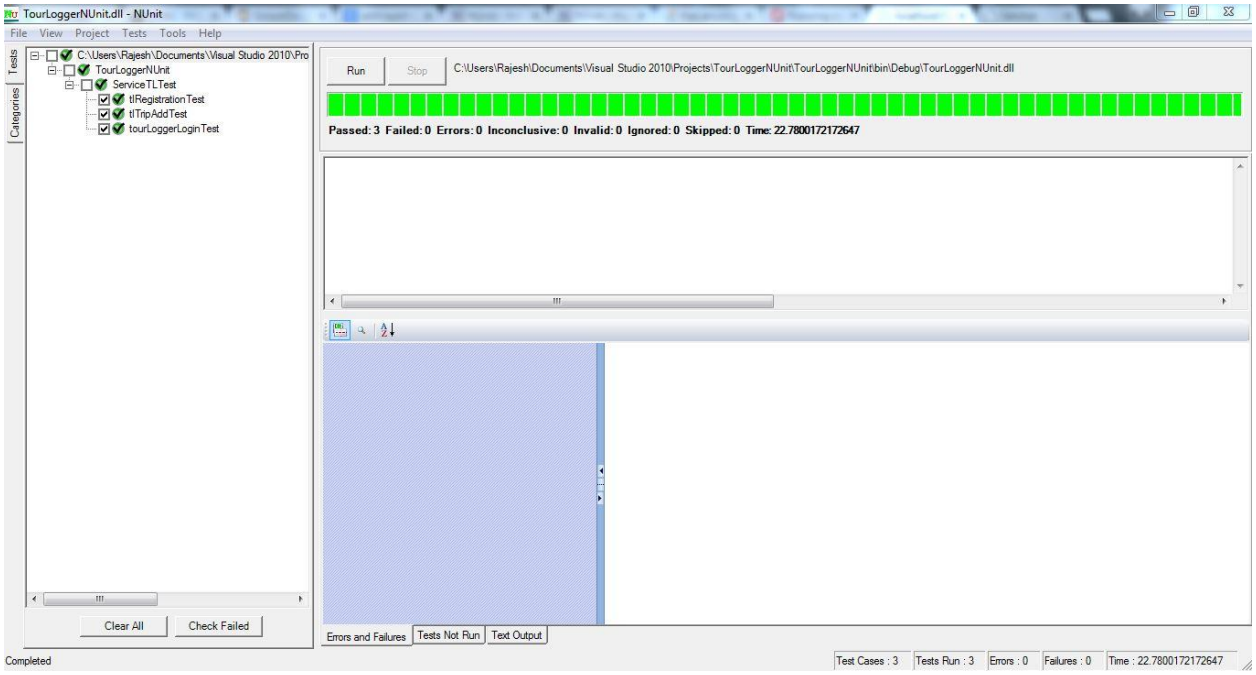
tourLoggerLoginTest() – To test whether the login details are present in database.

tlRegistrationTest() – To test inclusion of end user registration details into database.

tlTripAddTest() – To test inclusion in of the trip details into database.

5. Testing:

Below screenshot shows the test cases run in Nunit and the results.



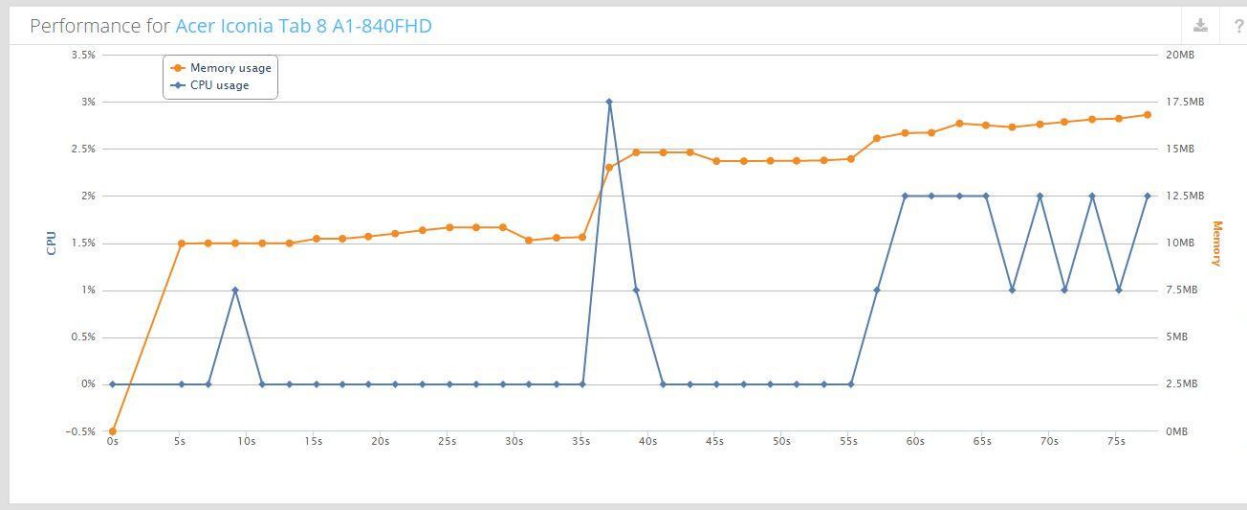
Systrace Performance Testing:

Systrace logs for performance was captured in android studio. Attached in github contains the trace file.

GitHub link: https://github.com/rajeshkapa/ASETourLogger

Testdroid:

Below performance graph was taken from testdroid application for the tourlogger application.



6. Deployment:

6.1 Project ScrumDo link

https://www.scrumdo.com/projects/project/asetourlogger/iteration/121729/board

6.2 GitHub site URL

GitHub link: https://github.com/rajeshkapa/ASETourLogger

7. Project Management:

ScrumDo link: https://www.scrumdo.com/projects/project/asetourlogger/iteration/121729/board

Implementation status report:

Work Completed:

Description:

1. Exposing the WCF service to store the details of the trip.
Responsibility: Raghu Vital

Time Taken: 20 hours
Contribution: 100%

2. Exposing the WCF service to display the trips added.

Responsibility: Rajesh
Time Taken: 20 hours
Contribution: 100%

3. Integrate the Modified Google places API to get the auto completion of places.

Responsibility: Rajesh, Raghu Vital
Time Taken: 20 hours
Contribution: 100%

4. Implement Attractions, hotels and restaurants to show various points of interests, hotels for staying and restaurants in the place selected.
   Responsibility: Raghu Vital, Rajesh
   Time Taken: 20 hours
   Contribution: 100%

Work To be completed: None

Issues/Concerns:

1. Faced issues with the google API key quota for searching Location, Attractions and Hotels.
2. Faced problems with performance testing with systrace.

8. RESOURCES:

https://developers.google.com/places/documentation/