ITIS/ITCS 4180/5180 Mobile Application Development
Homework 8
Date Posted: 11/19/2014 at 21:30
Due Date: 11/25/2014 at 23:55

Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your project as follows:
   a. From eclipse, choose *"Export…"* from the File menu.
   b. From the Export window, choose *General* then *File System*. Click *Next*.
   c. Make sure that your project for this assignment is selected. Make sure that all of its subfolders are also selected.
   d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
   e. When exporting make sure you select *Create directory structure for files*.
   f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission details:
   a. When you submit the assignment, compress your exported project into a single zip file. The format of compressed file name is HW#.zip
   b. You should submit the assignment through Moodle: Submit the zip file.
   c. You should also include the APK file for your application.
6. **Failure to follow the above instructions will result in point deductions.**

HW8 (100 Points)

In this assignment you will develop “Locate NearBy!”. The App search near by places according to user query by place type. You will learn how to use Google Places API (https://developers.google.com/places/documentation/search). A Nearby Search lets you search for places within a specified area. You can refine your search request by supplying keywords or specifying the type of place you are searching for.

Important App Requirements:

Points will be deducted if your application does not follow the below requirements:

1. The required Android Virtual Device (AVD) should have **minimum SDK version set to 14 and target SDK at least 17**. The app should display correctly on 3.2” QVGA (ADP2) (320x480: mdpi). Your assignment will not be graded if it does not meet these requirements, and you will not be granted any points on your submission.
2. All image downloading should be performed using Threads (or AsyncTask) and your code should not block the main thread.

Setup Your Project

To be able to use Google Places API, you need to follow the steps you learned to use to set up the project to use Google Maps.

- Setup your project and required library dependencies. Import google play services into your project and then add it as a library to your project.
  <android-sdk-folder>/extras/google/google_play_services/
- Get an API Key from Google maps. Using the same steps you used before to get API key for google maps, **BUT** make sure to do the following:
  - Click Services
  - Click on ***Places API*** in addition to the services you used before for Google Maps like Google Maps Android API v2 and Google Maps Geolocation API.
- Setup the required permissions and settings in AndroidManifest.

Search nearby API

As mentioned earlier, the app will use the search nearby API , example URL call:

```
https://maps.googleapis.com/maps/api/place/nearbysearch/json?
location=lat,long&radius=value&types=place_type&key=AddYourOwnKeyHere
```

This API takes as input parameters:

- output format: either json or xml, we can use what ever format.
- location: The latitude/longitude around which to retrieve place information. In this project we will use current location, which should be retrieved by using the Android LocationManager.
- radius:Defines the distance (in meters) within which to return place results. The maximum allowed radius is 50 000 meters.
- types: Restricts the results to places matching at least one of the specified types. See the list of supported types. In this project you will use a specified set of types.
- key: Your application’s API key.

The API will return an array of places, a sample element in JSON should be like the following:


```
"geometry" : {
            "location" : {
               "lat" : 35.313468,
               "lng" : -80.74304600000001
            }
         },
         "icon" : “http://maps.gstatic.com/mapfiles/place_api/icons/cafe-71.png”,
         "id" : "86b740f36c412ea5fac3994f4d77b07c93e75616",
         "name" : "Starbucks",
         "opening_hours" : {
            "open_now" : false,
            "weekday_text" : []
         },
         "place_id" : "ChIJbfPqPD0cVIgRgZm-jxDD3A8",
         "price_level" : 2,
         "rating" : 4,
         "reference" :
"CnRqAAAALTyjkoW4kCziMVnXh1DgWJM0S1X5KmKMM5ildPbWkhP8VUH1FGCIe_rynT3WoYiPLI7YRBDs4tYlOPviosqCqZpSHZic3w3iJORocR
zlt9liTyMqXNcC_DpHUnS1aGROshiKNSsWLEeKFCsZ1tYW5RIQXOsrwOapiif4zXPcaSaEGRoU9CK4Fo2Bb7_l9vQkIVmfs4TTf2Q",
         "scope" : "GOOGLE",
         "types" : [ "cafe", "food", "establishment" ],
         "vicinity" : "9335 N Tryon St, Charlotte"
      },
```


Part A: Search Activity (20 Points)

First activity should implement a simple search UI. (Figure 1)

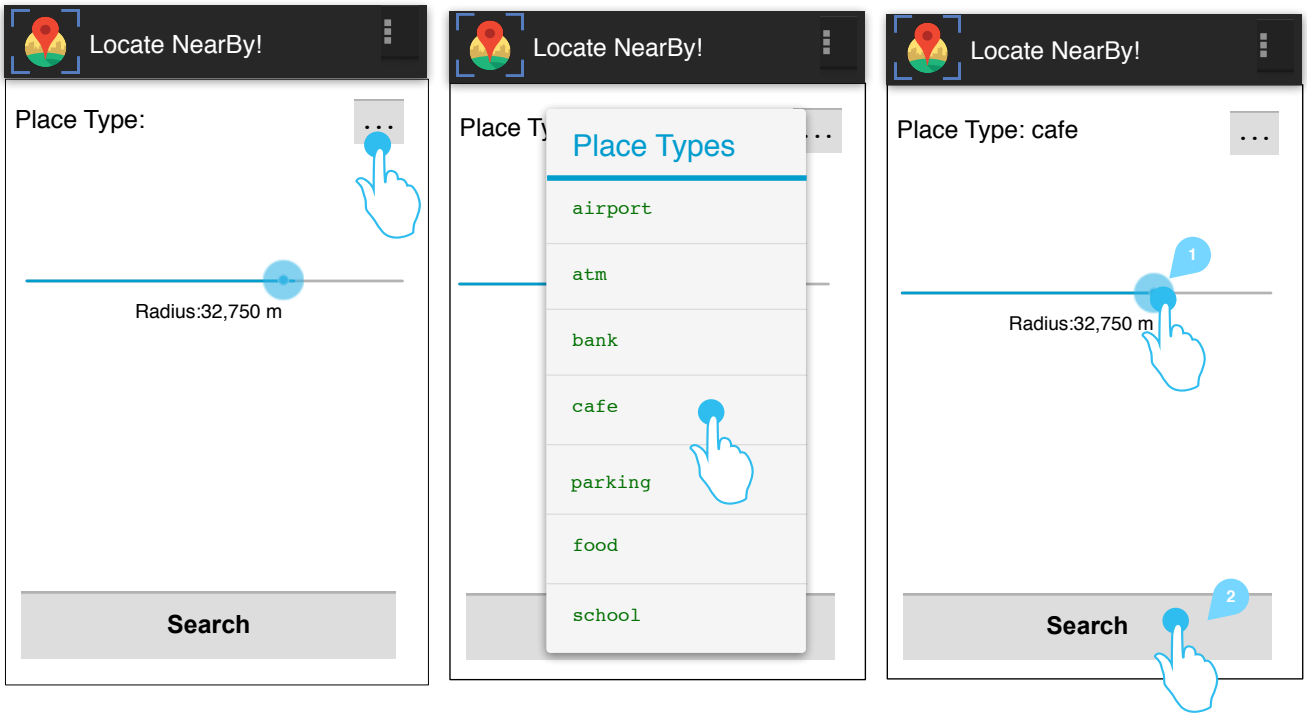


Figure 1

The app should implement the following functionalities.

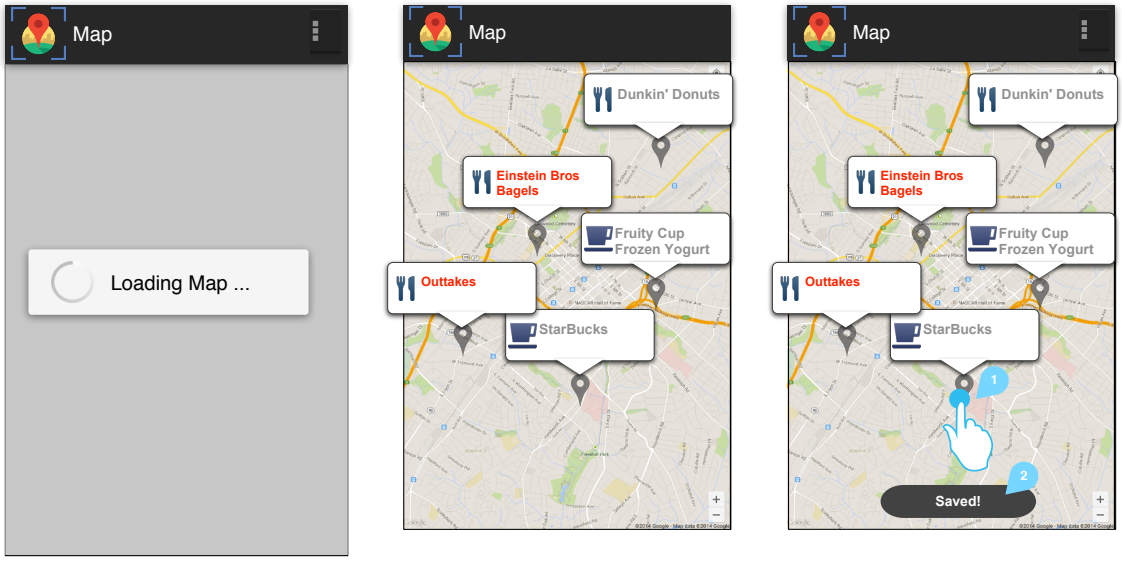
1. Place Type can be selected by choosing from alert dialogue. You are required to use only this set of places types (airport, atm, bank, cafe, parking, food, school)
2. Radius can be set through a seek-bar, the value ranges from 0-50000 meters.

3. Search Button should build the API request call after constructing the parameters properly. This should start the next activity that will display the search result. You should use an AsyncTask to retrieve and parse the data from the Google Places API.

Part B: Map Activity (50 Points)

The Map activity displays a map containing markers of the places returned by the API call (Figure 2).

1. Google Map: Display a map containing markers of all places returned from API call. Map should be properly zoomed so that all markers are displayed properly.
2. Markers: Add marker on each location returned.
3. Info Window: Each marker should display a customized info window that displays:
    - icon
    - place name colored red if the place is currently opened, grey otherwise.
    - You can read more about Info Window/ Customized on this link: https://developers.google.com/maps/documentation/android/infowindows#showhide_an_info_window
4. Clicking on a marker, should save the place into SQLite DB that has one table of FavoritePlaces.



(a) Map Activity Loading (b) Display Places Markers/ InfoWindow (c) Tapping on place will save it in DB

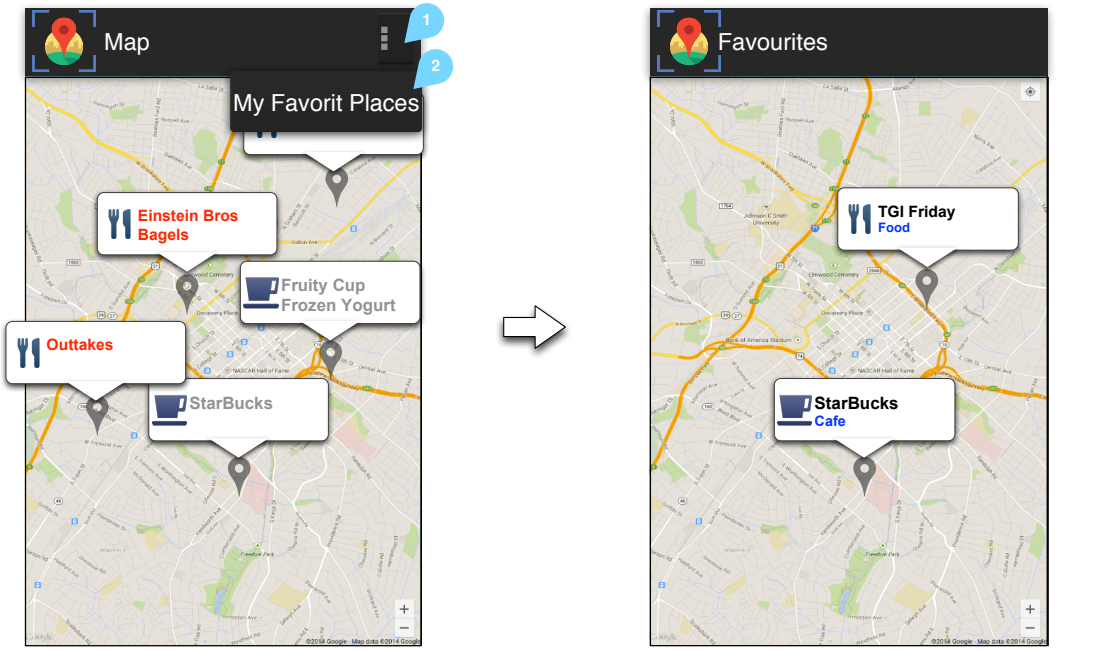
Figure 2

Part C: Favorite Places -SQLite DB- (30 Points)

As mentioned earlier, tapping any location should save it to internal SQLite DB. This database will contain data about you favorite places of different types.

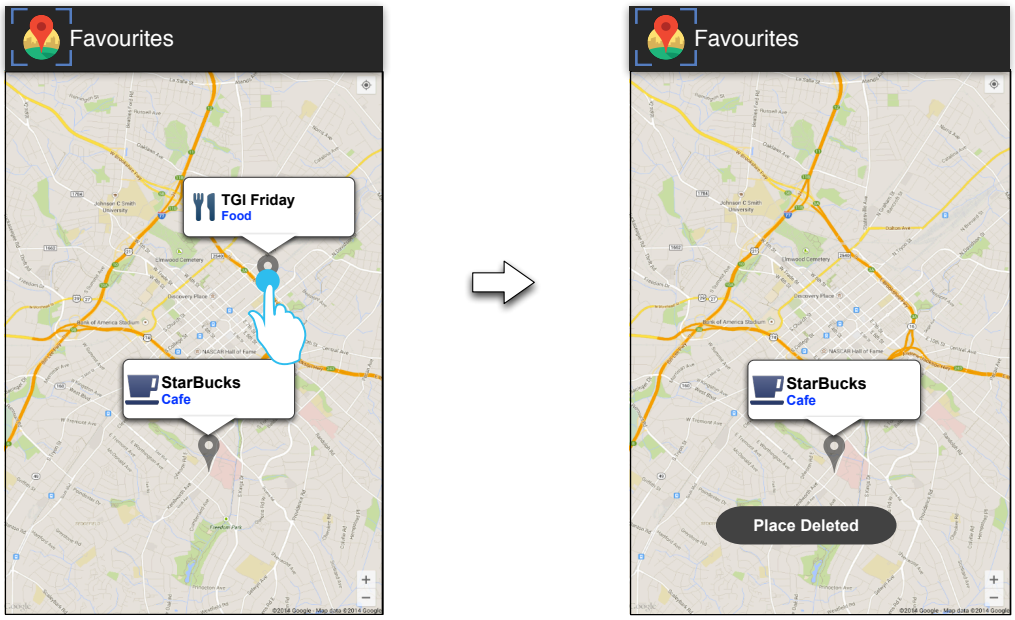
1. To view favorite places, you should implement an action bar in the second activity with menu item “My Favorite Places”. Tapping this menu item should display a map showing markers on all saved locations.(Figure 3-a)

2. Each marker should show the following (Figure 3-(b)):
    - icon
    - location name
    - location type
3. Tapping a marker should delete it from DB (Figure 3-(c)/(d))



(a) Action menu item "My Favorite Places" to display a map with markers on favorite places

(b) Map with markers on favorite locations, info window show place icon, name and type



(c) Tapping a location should delete it from DB

(d) Location deleted

Figure 3