ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 13

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. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Submission details:
   a. All students are required to submit this assignment individually.
   b. The file name is very important and should follow the following format: **800#_InClass13.zip**
   c. You should submit the assignment through Moodle: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 13 (100 Points)

In this assignment you will develop a google maps and places based application. The App search nearby places according to user query by place type. The app will show the map with markers of the places that are returned from the API call.

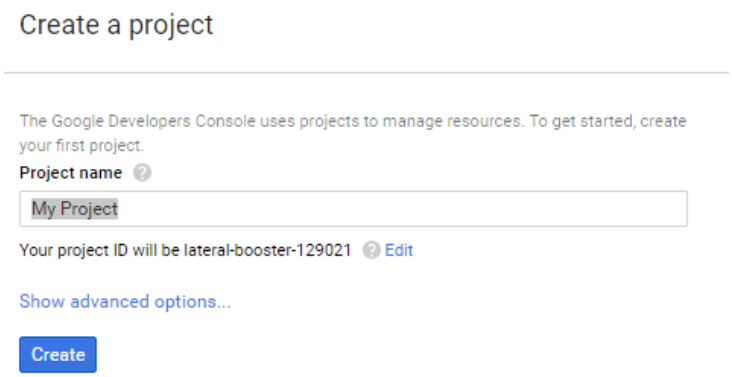
Google Maps and Places Android API

In this assignment you need to create map objects and also get google places for a particular place type for the current location. In order to do that you need to first get the google maps android API and places android API key.

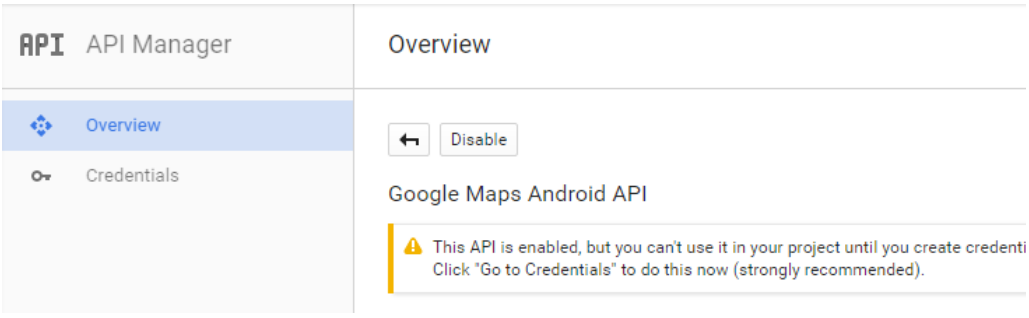
Get API Key:

Please follow the below steps to get the API key for both maps and places. Note that both maps and places share the same API key.

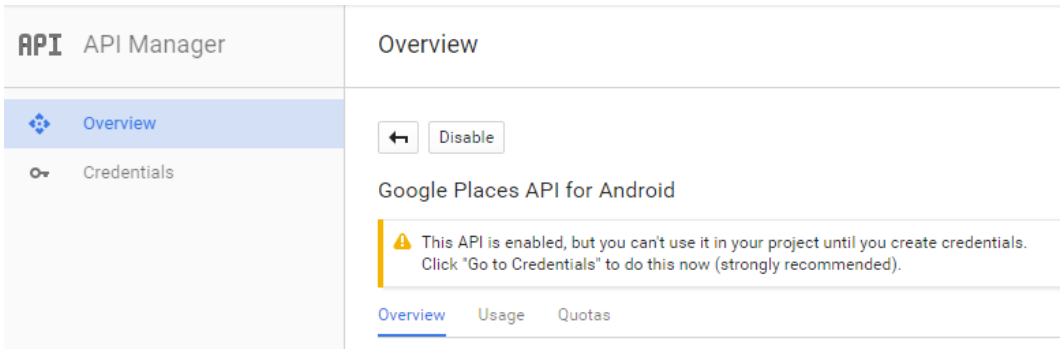
1. Create New Project -> Google Maps Activity
2. To get Google Maps API Key follow instructions in **`google_maps_api.xml`**
3. You will be directed to your google account developer's console.
4. Create a project with name.

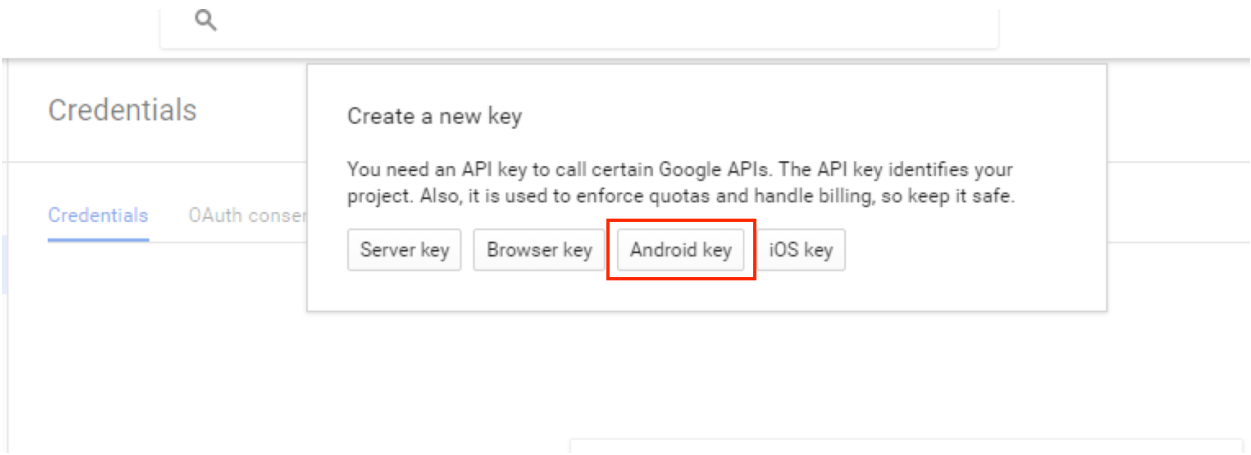


5. Now select Google **Maps Android API** from the products list and enable the API by clicking on Enable button in the Overview section.

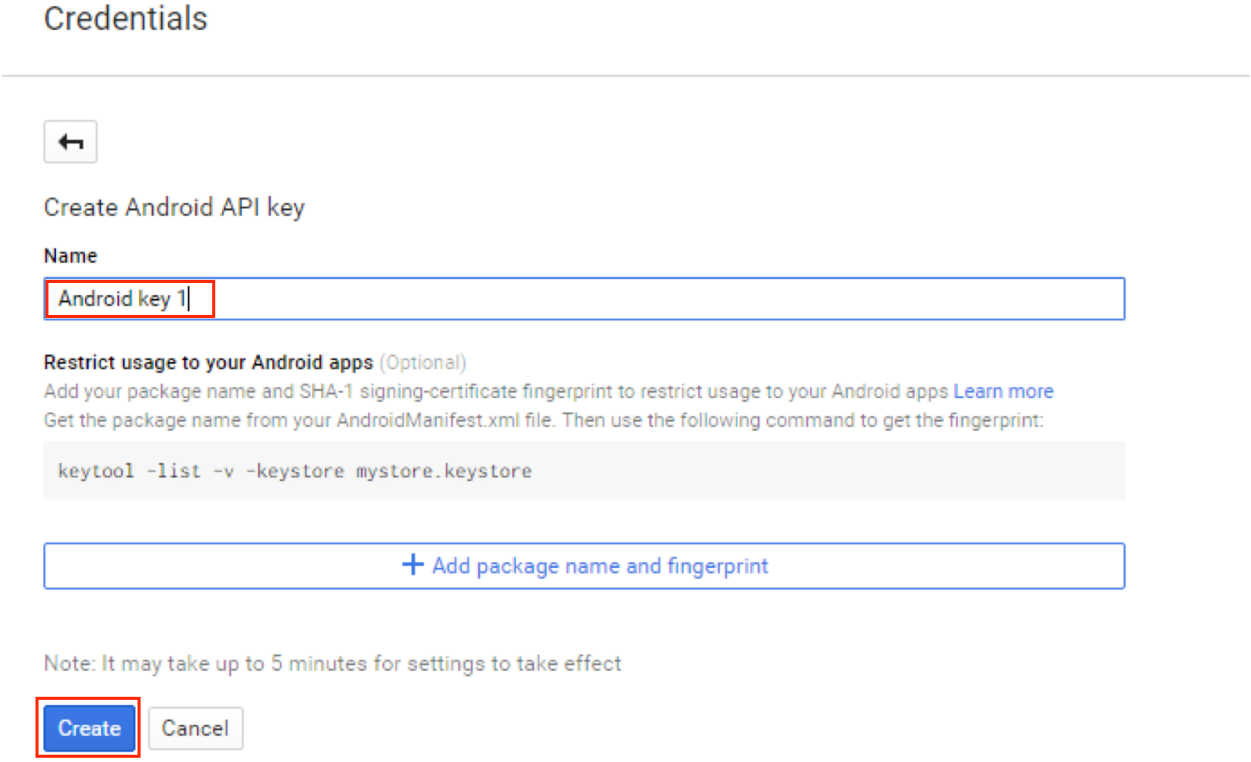


6. Likewise, Select the **Google Places API** for android from the products list and enable the API .





7. Create the Android API key and click on the create button.

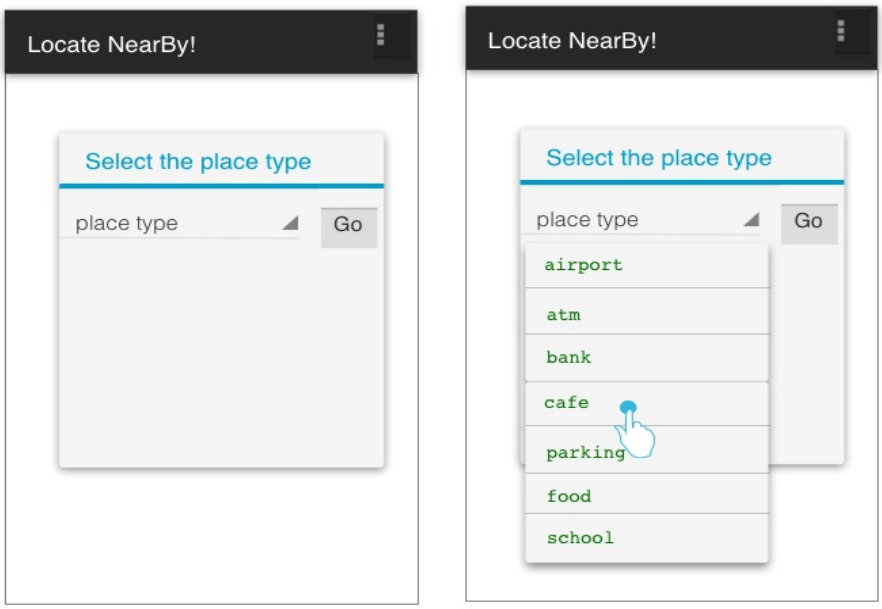


8. Please copy the API key into your Manifest file.

```
<meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="YOUR_API_KEY"/>
```

9. Please find the documentation for places API https://developers.google.com/places/android-api/start here.

10. Also Please find the documentation for maps API https://developers.google.com/maps/documentation/android-api/start here



(a) custom alert dialog **(b) select place type**

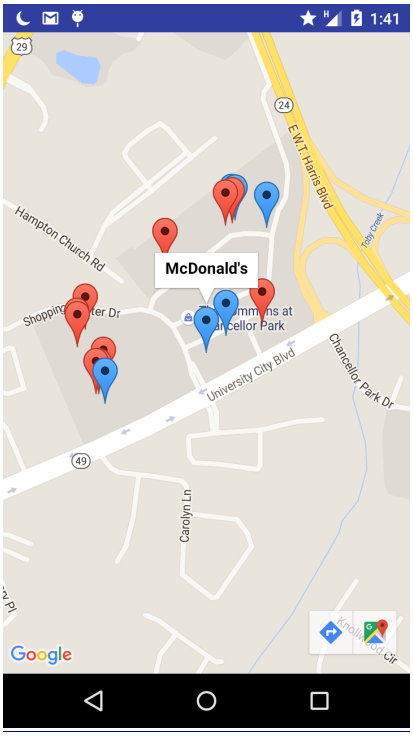
Figure 1: First Selecting a Place Type

Map Activity (100 points)

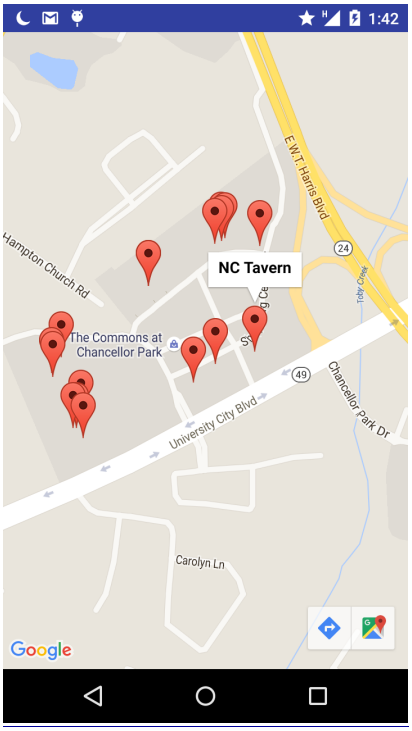
The Map activity should display a map with markers of nearby places returned by the API call. The app should implement the following functionalities:

1. Initially, the activity will show a customized alert dialog with a list of places types (choose at least 5 place types from this list: https://developers.google.com/places/supported_types#table1 (See Figure 1)
2. Google Map: After selecting a place type, the Google map should appear navigating by default to the current location. To get the current location use Google Places API: https://developers.google.com/places/android-api/current-place

3. Markers: You are asked to add markers on all returned places.
4. `Places.PlaceDetectionApi.getCurrentPlace` returns a list of near by places. Map should be properly zoomed so that all markers are included within view. Please check the documentation here: https://developers.google.com/maps/documentation/android-api/views#updating_the_camera_view
5. Info Window: A simple Info Window should appear when clicking on the marker displaying the place name.
6. For places that matches the place type selected by the user, the color of marker should be changed (See Figure 2)



(a) **Search: TYPE_FOOD**



(b) **Search: TYPE_MEAL_DELIVERY**

Figure 2: Different Search Types