Week 3 Google Map API

Larissa Thompson

MBL/402 Android™ Mobile Development

May 22, 2017

Chen Li

Week 3 Google Map API

The weekly assignment was to research and implement the Google Map API and learn about permissions. The introduction of a standard API like Google Maps was a challenge because there are many different ways to using the API. Also learning about permissions and the way they can differ based on Android versions was essential for developing an application for a broad audience. I was familiar with the various permission and requesting them for both the Android permission request types.

The user interface components were very basic. The central map area was generated by the Android Studio Mapp Activity template. I did modify the layout to utilize a relative layout to hold the map fragment. The relative layout allowed for the Menu app bar to be used and buttons at the bottom of the UI. The UI changes were minor but did require some troubleshooting to have everything aligned as desired.

Once the user interface was completed and the necessary elements in place, it came time to add in on click listeners, add permissions, and implement the Google Map location handling. I began with the most straightforward task of setting up the zoom in and out buttons. I did have to rely on the Google Map API documentation for the proper syntax and calls for the buttons to function properly (Google, 2017). After the buttons, I added the menu options and created a switch case statement to handle the various options.

After the user interface had been functioning properly, I added in the permissions. I did research the permission that should be given for the Google Map API and found that the documentation recommended not only the location service but access to the internet and data network to give more options for locating the user. I followed the Google samples for permissions requests and made the allowance for the application to run on both old and newer devices.

The challenge of the assignment was getting the accurate location of the device and the address. Using the built-in methods of the API, I was able to set up an onConnect method to ensure permissions were given and a location watcher. The location watcher allows the marker on the map to move as new coordinates are provided. Within the location watch, I included the processes to get the address of the marker. I used the Google Geocoder to pull the address of the coordinates. While this method can provide all of the requested information in populated areas, otherwise the map marker will show unknown.

The application requirements gave an excellent introduction to all of what the Google Map API can perform. As more users want location-specific results, being able to access location per various permissions is an important lesson. While this app itself is not more than a pinpoint location application, it does introduce how to use the map properly.

References

Google. (2017). *Map*. Retrieved from Android Developer: https://developer.android.com/reference/java/util/Map.html