Candidate: **Avdhoot Saple**

Exercise assigned: **Friday 04/24/2015**

Exercise completed: **Sunday 04/26/205**

This is the first time, I've developed a mobile application end to end. I've given my best efforts to peak over my learning curve ASAP and make the application as user friendly as possible, I apologize for the look and feel being not so great. All the functionalities have been addressed including the bonus items.

Please feel free to contact be at **518-301-1181** for any clarifications. I can also be reached at **asaple1983@gmail.com**

Overview:

This document describes the various aspects of the solution designed as per the Enterprise Engineer Exercise

Features and Assumptions:

1. The application is developed with Android SDK 16 as the target version. Additionally I have performed testing on emulator and also on my Samsung Galaxy Note 4 (Lollipop). The screenshots below are from the deployed app on my Note 4
2. Google Places API is utilized for retrieving information of places relating to "food" and also to retrieve real time photos and icons (library used: google play services)
3. The default range is set to 500 meters (the exercise mentions 500 feet but a safe assumption is made about it being a typo since 500 feet is too short of a distance; nevertheless the app can be configured with a default of 500 ft with minimal effort)
4. The application lets user override the default distance and add a custom distance to scan for food outlets
5. Location updates are enabled so that the list will auto refresh based on the geo coordinates of the device. The update interval is set to 10000 milliseconds. I tested this by keeping the app active and driving few miles away from my location.
6. It is safely assumed that location services will be enabled on the device on which this application will be tested. If location services are disabled, the application will use a set of hard coded coordinates (40.7463956,-73.9852992) so that functionality can be restored for testing.
7. I've used my gmail acccount (asaple1983@gmail.com) to get API keys from google API's. Those keys are being used in the application.
8. Google maps API is utilized to address one of the bonus question (different geo location app with coordinates should open up) (library used: google play services)
9. Email clients available on the mobile device are used to address one of the bonus question (share information about restaurant)
10. No other third party components (apart from the above) are being used.
11. The Eclipse IDE was used for development. (Eclipse Juno)

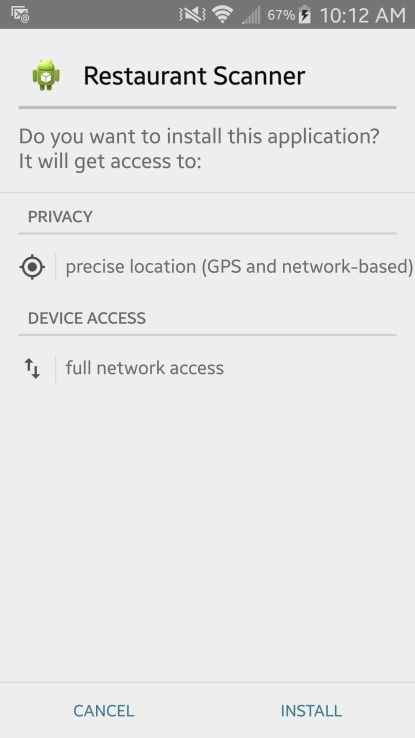
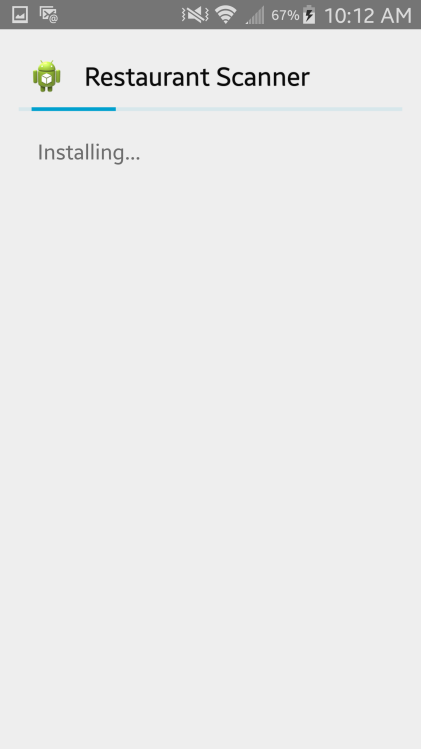
Following artifacts are attached as part of the solution

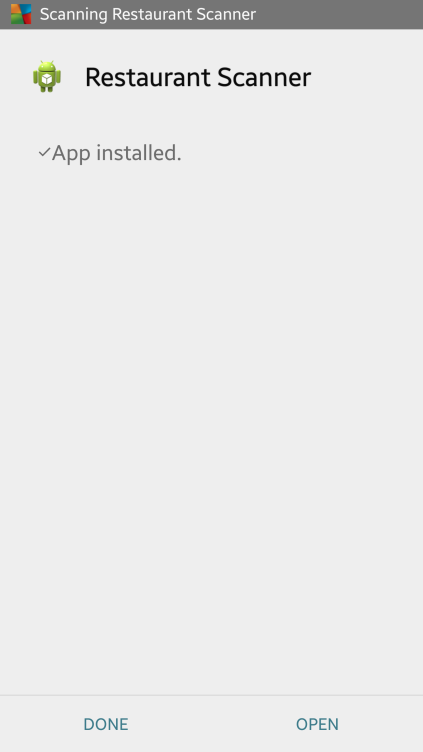
1. The complete source project folder from Eclipse (RestaurantScan)
2. the google-play-services library that was used as a dependency (google-play-services\_lib)
3. The RestaurantScan.apk file which was tested on a mobile device.
4. All the artifacts can be found at the following Github location

<https://github.com/asaple/SamsungEntAvdhootsaple>

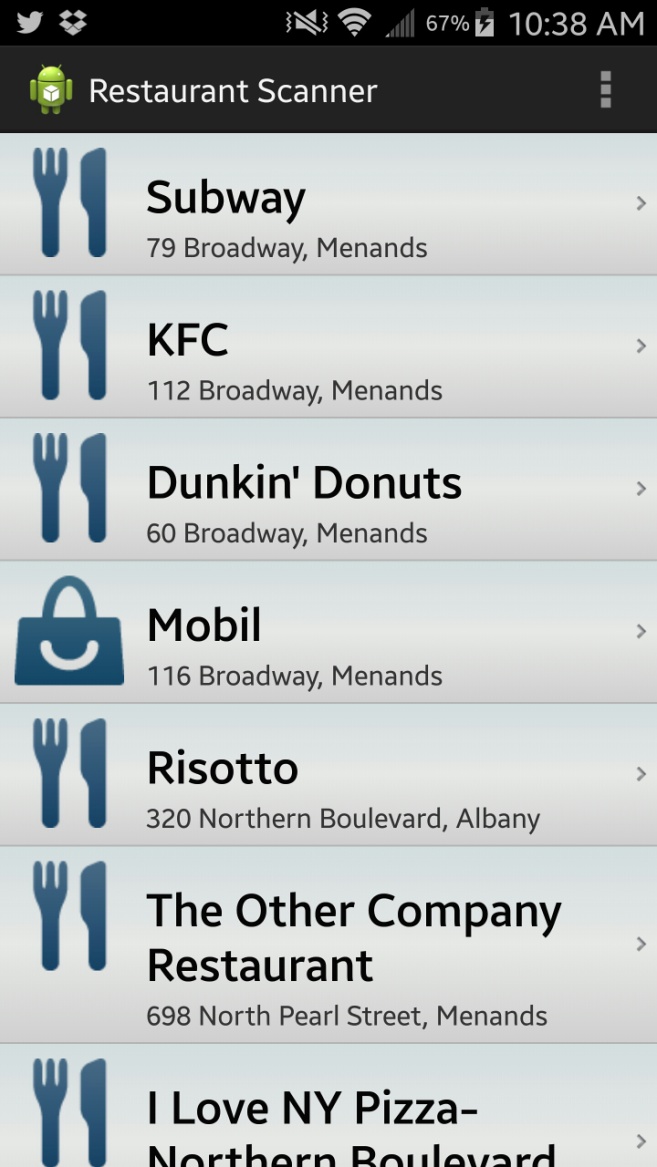
**Screenshots**:

1. The apk file was copied on mobile device to start the installation



2. The application is started. At this point it will display the nearest food outlets in the vicinity of 500 meters. The icons are retrieved as part of google places response depending on the type of food outlet

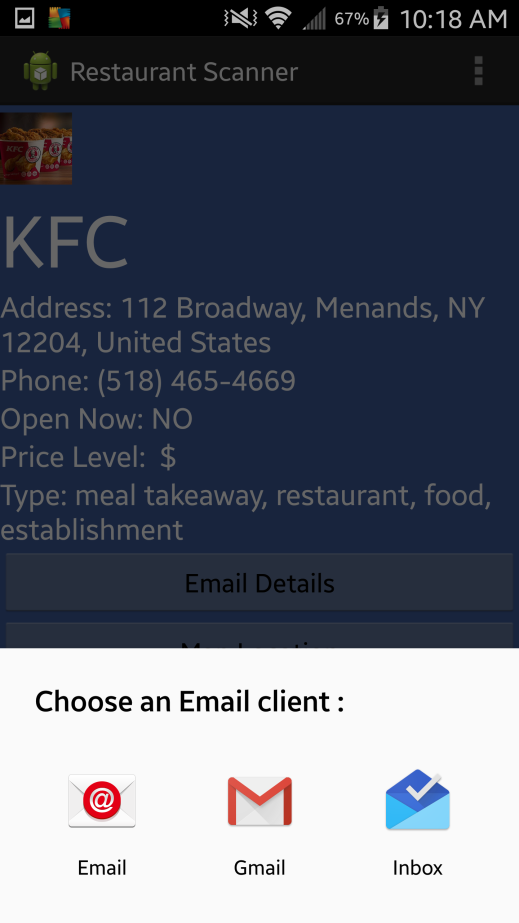
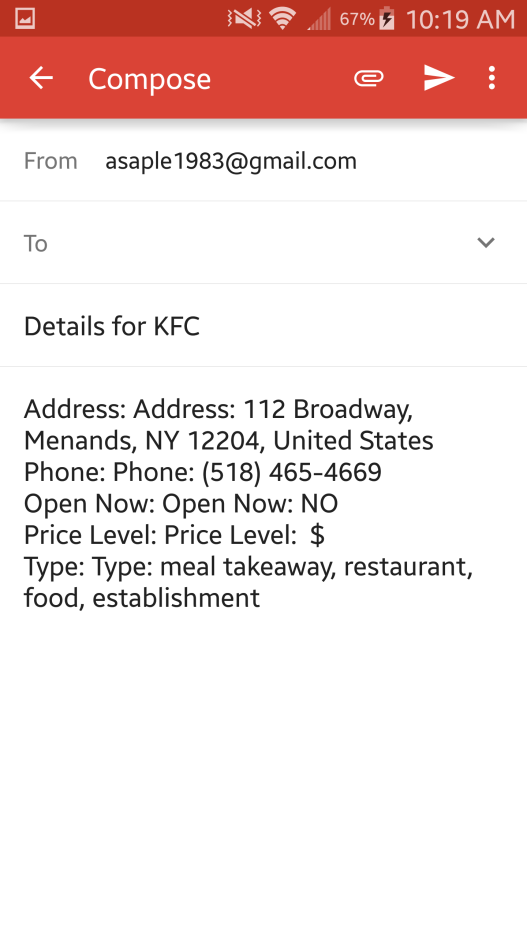


3. On selecting any item, the detail page is shown. Eg: KFC is selected above

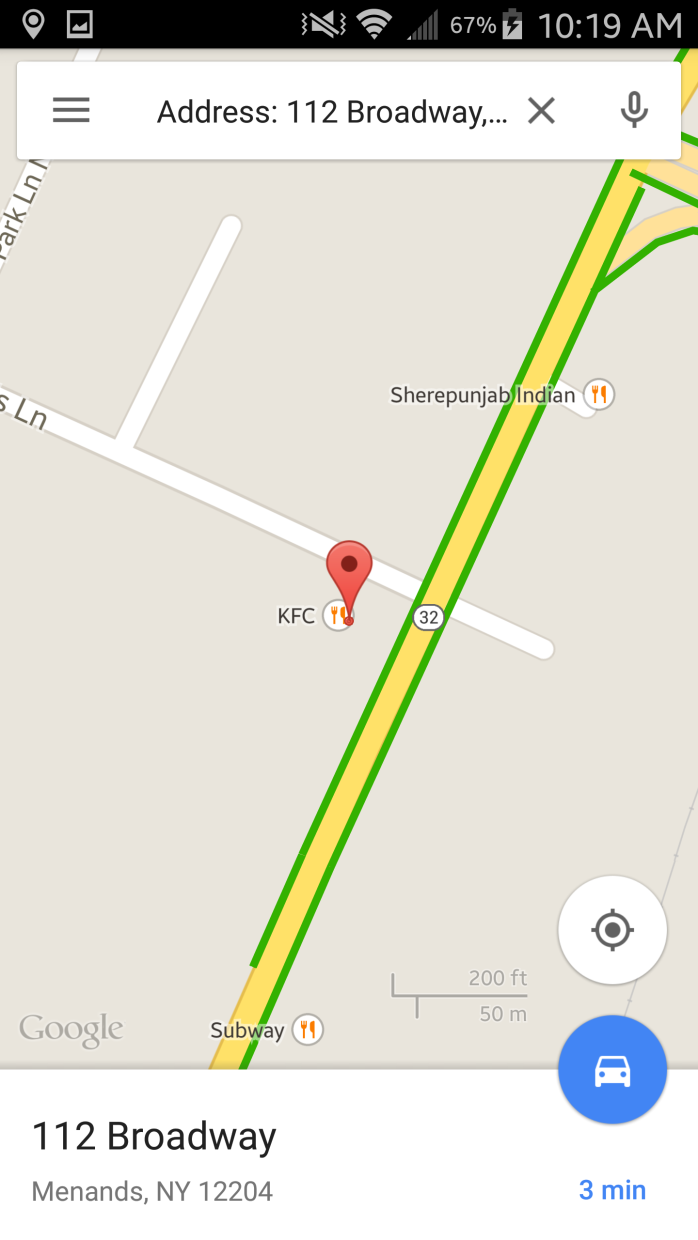


The image for the outlet are retrieved as part of google places photo api

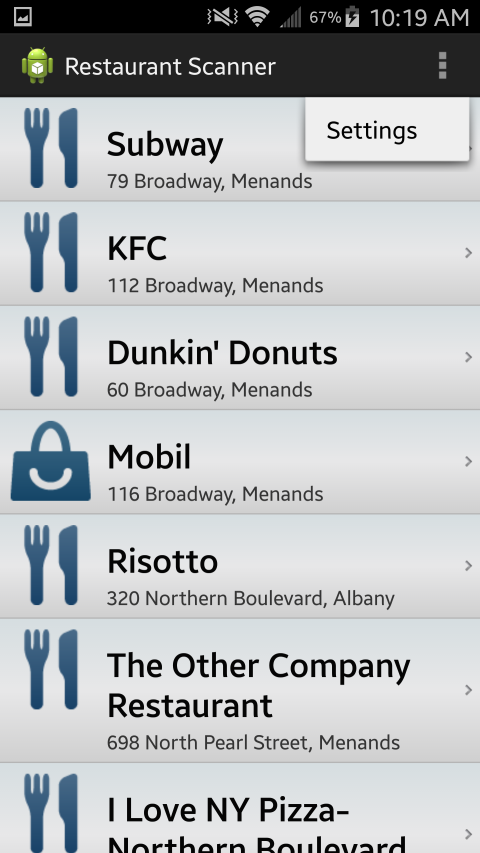
4. Email details delegates the detail to an email client selected by the user

5. The Map Location opens up google maps with pin pointing the location of the restaurant and also lets user to navigate directions from current location of device



6. The Action bar - settings menu item lets user modify the range at anytime. A validation is also provided for user to enter only numeric values



7. As mentioned in the features section, location update is configured for every 10000 milliseconds, the list auto refreshes itself.