**Chapter 2: Design**

**Section 1: Design of LTE class**

Each LTE channel is an instance of the LTE class. As for the identify of each class, Android provided the following options, Mobile Country Code (MCC), Mobile Network Code (MNC), Cell Identity (CID), Physical Cell Identity (PID), Tracking Area Code (TAC).

MCC and MNC are always combined to use for identification about network providers across countries but not for identifying different signal tower in a local region. TAC is always used to identify a group of cells but not a specific one which not applicable for the app in the project. PCI are not unique in a network, however, two nearby towers do not share the same PCI. In our application, we mainly focus on the local region detection, so PCI is still one of the potential candidates. Comparing with CI, PCI is more easy to computed. Also, as for users, CI is a 28-bit code and PCI is just in range from 0 to 503, PCI is more easy to be interpreted by the users. For these reasons, we choose PCI as our identity for each LTE channel.

As for signal strength, we both logging the RSRP and ASU level on the LTE List Fragment, but we use the RSRP to distinguish the LTE channel’s condition in the Localization activity, because RSRP has wider range than the ASU level which allows us to have more detailed signal levels.

**Section 2: Design of Wi-Fi class**

Each Wi-Fi channel is an instance of the Wi-Fi class. The BSSID and SSID are two of the private members of the class. Specially, as there may be different APs with same SSID in a region, the BSSID is unique for every AP. Therefore, it is also used later to be passed to the Location Activity to search the AP users are interested in.

The signal level is another important private member of the class. The signal level is the RSSI of the Wi-Fi channel. RSSI is the most common indicator used to assess the signal strength of the Wi-Fi signal.

**Section 3: Usage of Fragment on UI handling**

In android applications, activity class and fragment class are both able to controller classes to handle UI. Activities are designed to represent a single screen and fragments are designed to be reusable UI layouts that can be plug-and-play by different activities and components. In our application, we heavily rely on the fragments control.

First of all, fragments separate the View components and the controller components. In android, a rotation is considered as a re-configuration. When users rotate the screen, all the activities and fragments are destroyed and have to be reloaded again. In activities, the instant variables are destroyed as well. As for fragments, even though the View components are destroyed, all the states are preserved. If the number of states are huge, this configuration saves many resources.

Also, fragments can be hosted in many different ways. Not only can be hosted by the activities, it also can be hosted by the components such as ViewPager. ViewPager is a component in Android allows users to swipe across different fragments. Fragments provide more flexibility than activities.

In conclude, for the above reasons, we use fragment as our main controller class in this application.

**Section 4: Usage of Singleton Pattern on Channel List Activity**

A singleton pattern is a class that only have one instance of itself. In android, a singleton exists across rotation, different activities and fragments. In our application, we have two singletons, LTE list and Wi-Fi list.

In our application, there are two activities updating the channel list class – channel list activity and location activity. As we pass the control from channel list to location activity, the singleton pattern makes sure that we can access the same channel list in the location activity. The original list is not destroyed. This feature is extremely helpful if we want to keep the whole history of the list, and this is what we need.

**Section 6: Usage of Google Play Services on Localization**

Android provides a basic Location API. The API allows us obtain the location data from fine location points such as a GPS radio and also some coarse location points such as cell towers and Wi-Fi connections. However, they are not integrated together. Occasionally, GPS is not available, developers have to manually transform between these location data resources. This tasks are tedious. However, the location API from Google Play service handles all these transactions for us. It is a better API we should use.