**Project Report**

**Garbage Monitoring System**

**by 21400772 최인회, 21100242 박경배, 20900527 이원희**

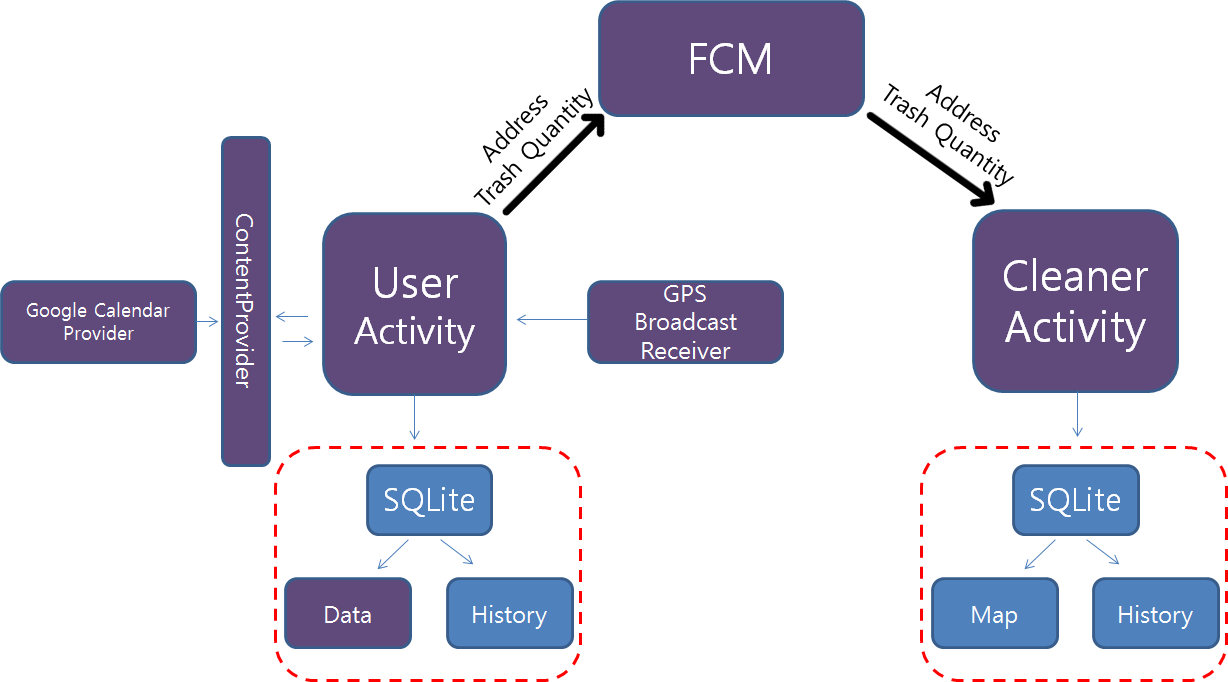
**Goal and purpose:**

To develop a system which can reduce cost, time and effort needed with those, associated with the garbage management system. Garbage trucks are required to go through all the streets every day without considering how much trash is actually out on the streets. If we were to create a path for the garbage truck, according to how much garbage was put out, we could use this to create efficient path, ignoring areas where the accumulated amount of garbage is not up to a standard. Areas with little garbage would be skipped and monitored until it would have enough trash to be picked up. This can ultimately, reduce fuel, time and effort by creating a more efficient, short and non-overlapping path.

**Simple Usage:**

Users are separated into two groups. The users (those who throw away trash) and the cleaner/collectors (those who will see the accumulated data). Users will send the amount they are throwing away to the garbage collector. By doing so, the collector would receive a sum of all the amounts and give clear presentation and guide to create their efficient pathway.

**Design Overview**

****

**MainActivity :** This activity has a log-in function. When users log in this app, they should select user-mode or cleaner-mode.

**UserActivity :** The major purpose of this activity is sending message to Firebase Cloud server to notify cleaner-mode users. After sending a message, contents of a message will be saved in SQLite Database and Google Calendar. Also, it will be showed up on list-view.

**CleanerAcitivity :** The major purpose of this activity is receiving message from Firebase Cloud server. When it receives message, message will be saved into SQLite Database and showed up through list-view.

**FCM :** FCM(FirebaseCloudMessaging) is a connector between User and Cleaner. As a kind of server, It receive a message from user and send a message to cleaner.

**GPS BroadCastReiver :** When users join our system, they have to write their address. So, GPS will be needed. If user didn’t turn on GPS, It would alarmed users to turn on GPS through Toast message..

**SQLite :** SQLite saves usage list of users or received message list from user.

**History :** Usage list of users or Received message list from user.

**Calendar :** Contains usage list of users on Google Calendar.

**MAP :** Show map from area which received message contains

**Application Components**

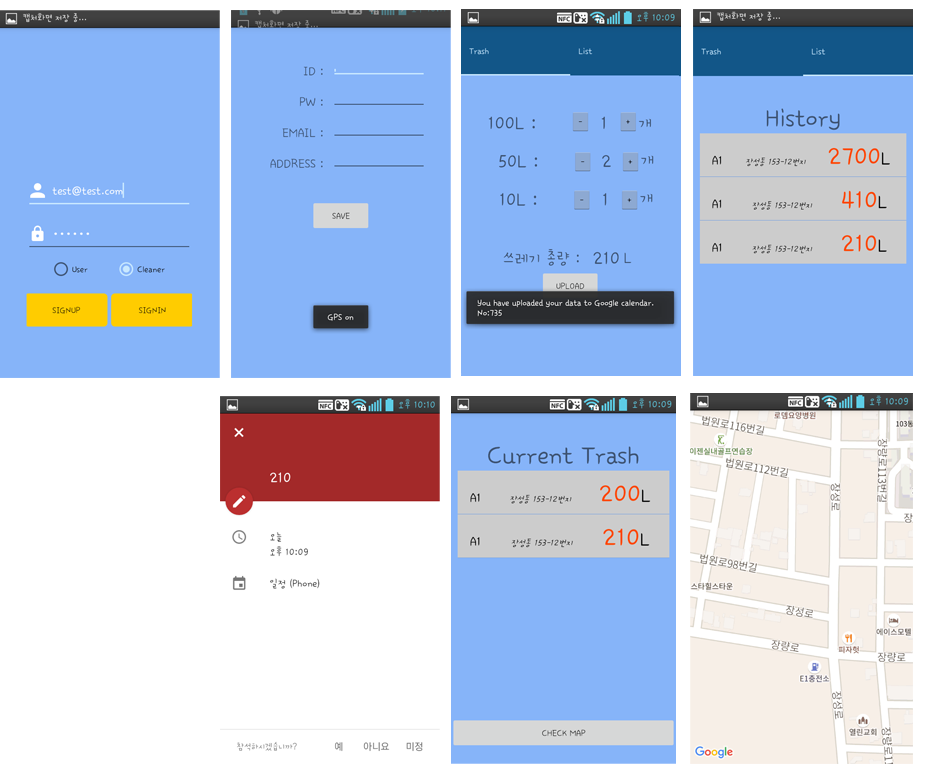
**MainActivity –** Composed of EditText, RadioButton, Button. StartActivity(intent) to open *SignupActivity* or *UserActivity*/*CleanerActivity*.

**SignupActivity** – Composed of simple EditText and LocationManager to implement BroadCast Receiver. We have chosen the GPS location for our Broadcast. Receiving through *GPSModeReceiver* class. Future implementation can include, auto-addressing with the use of GPS.

**UserActivity** – Composed of fragments, buttons and ContentProvider. The two fragments included are *UserHistoryFragment*, *UserTrashFragment*. In UserTrashFragment, we used the data from Google Calendar ContentProvider. The data gathered from the user (100lt,50lt,10lt buttons) get sent through the ContentProvider to create an event in Google Calendar. Another important part of UserTrashFragment that it sends notifications with the Trash amount data. Which will be later used by FCM. In order to use the Content Provider, we included a class *AddusingContentProvider* to use the appropriate Calendar provider. Using interface *EventAdder* and *ParseUtils*, we have successfully implemented Contentprovider in a fragment. In UserHistoryFragment, the data that is gathered, gets sent to a ListView.

**CleanerActivity** – Acquires data from UserActivity through FCM. Having a similar UI with the UserHistoryFragment, CleanerActivity provides listview of what people have uploaded. Clicking on the button will lead to *CleanerMapActivity* which uses Google Maps.

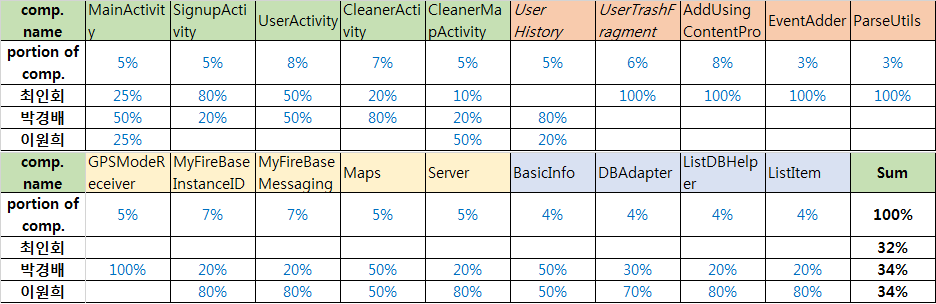
**FireBaseCloudMessaging (FCM) –** In order to use the Firebase Messaging system, it needs two classes. One to connect to the server, and one to receive and notify. In our case *MyFirebaseInstanceIDService* connects to the server, while *MyFirebaseMessagingService* receives and notifies. FCM is used as service. Also, after app was killed, App can receive message from user by acting as a Service.



**Name of App. Components and Advanced Features**



**Role and Contribution**

****

**Feeling of Each Member after finishing the project.**

**최인회 21400772**

It was my first time developing a project of this size. What started as a simple idea grew and grew. By the end of the project we were overwhelmed but the fact that we finished gave me more motivation to work harder. It was an amazing experience working with each other and building teamwork.

**이원희 20900527**

After finished our project, I felt that building a large application is very hard. There were a lot of exceptions or errors. Also, I wished that more of these kind of Application that are aimed to make hardworking laborers’ work more convenient would become a trend.

**박경배 21100242**

First of all, our team work and role allocation was done well. Understanding about android application and confidence about implementation of application was increased. In design and implementation process, I learned that my plan was not perfectly achieved and 'good output' comes from 'good design'.