

# Report

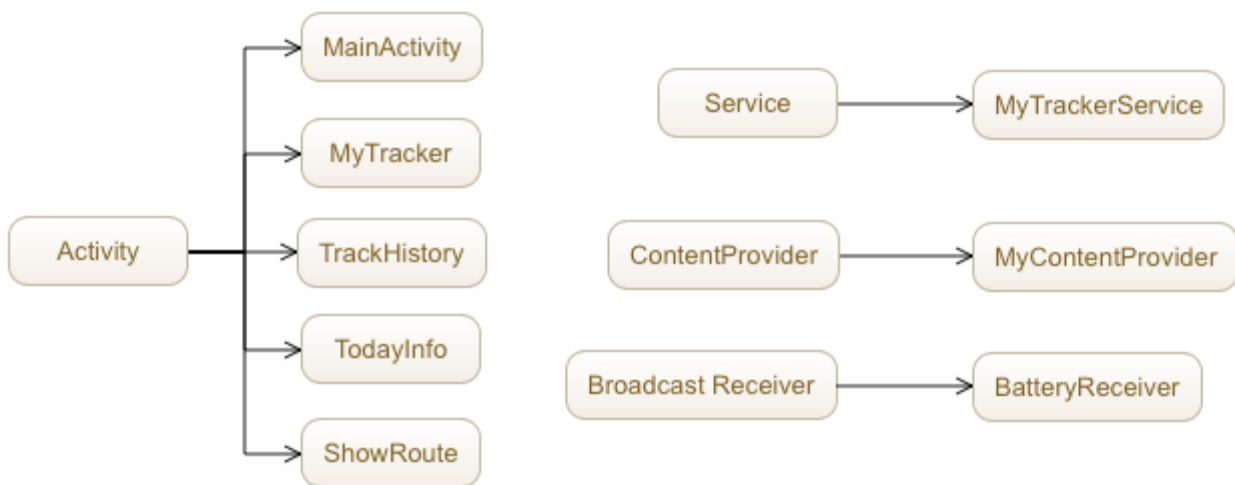
Name: Yusheng Wu  
Student ID: 4256684

## Introduction

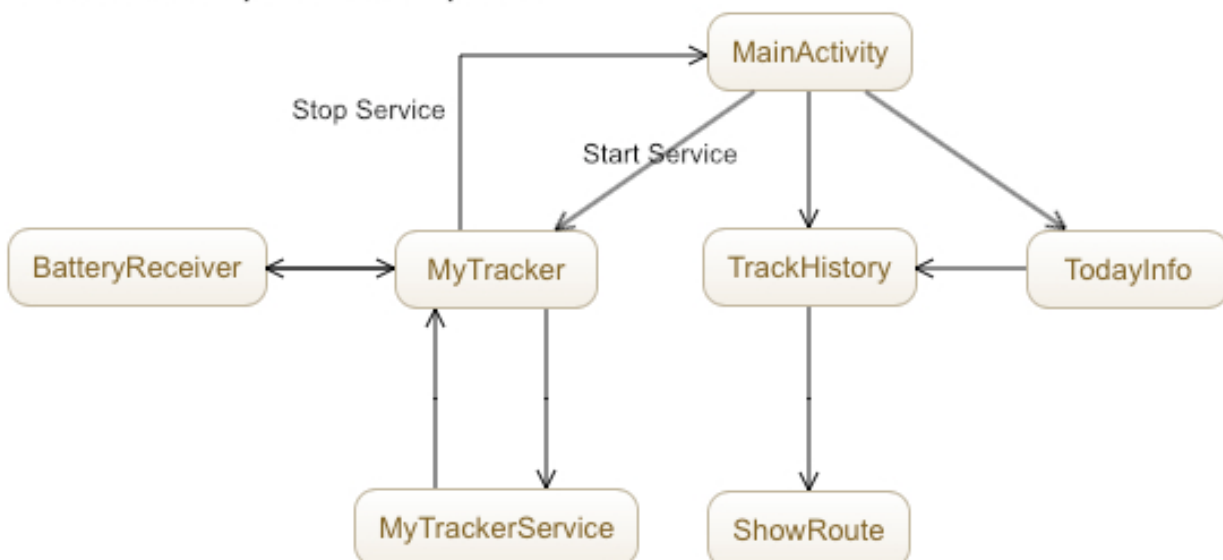
The activity tracker app I designed is expected to be used for people who want to track his movement lively or record his footprints. In our daily life, we often have questions such that how many distances have I moved today, which places did I go yesterday and even more specific question like where I was at three o'clock on last sunday afternoon. Fortunately, my activity tracker could help people to answer these questions.

## Architecture

I use all four major components as expected in my app. In general, the file is distributed like this



### Basic Relationship Between Components



## Activity

Each five activities have their unique function:

- MainActivity

This activity is acted as the entrance of the app. User could click "start track" to let the app starting track himself or click "history" to get the information of previous track records.

- MyTracker

This activity is used to control the track. User is able to pause or resume the tracking status here. Additionally, after the track is paused, user could choose to save the track record by clicking "save" button. Certainly, if user doesn't want to save those recorded locations, just simply back to "MainActivity" and the track records will not be inserted in the database.

- TrackHistory

"Track History" is used to show all the information stored in the database. Recorded information includes the latitude, longitude and record time. User could also use this activity to export the data by Email or delete all the data. The requirement of sending an email including the exported track records is that there exists a mail client app like "Gmail" in user's device.

- ShowRoute

This activity is able to visually display the track record for a specified day. The route shown in google map could help user to understand the data more intuitively.

- TodayInfo

This activity is used to show the statistical information of the current day. User could click the button "Today" to enter this activity to get the moving achievement of himself.

## Service

Although I understand that the LocationManager is perfectly adequate for this coursework, I still tried to use Google Play service location APIs because it offers higher accuracy and low-power geofencing. Basically, I use location APIs to receive the location updates and get the information like latitude, longitude of a specific location. The service will be started once the user click "Start Track" button in "MainActivity", then keep running until the user leaves the "MyTracker" activity. When the service is running, users could use the device to do other things and they can go back to the "MyTracker" activity by clicking the notification.

## ContentProvider

A typical content provider is made to communicate with SQLiteDatabase.

## Broadcast Receiver

As we know, the app related to location tracking is usually a power hog like the Google Map App. Thus, to remind the user of battery status, I use the BatteryManager to get the status of battery and display it in the "MyTracker" activity, then user is capable of noticing the battery condition and stop track service in time if the device is out of power.

## Design

There are four design issues I want to explain here:

- "MyTracker" activity

When I designed my activity tracker app, I believe that the user's control right is the most significant part. The system should allow the user to control the start and end of the track service conveniently. Therefore, I deliberately built activity ("MyTracker") to let user control the service status, they could also stop the service by going back to the MainActivity or save the tracked locations.

- Data Export

In my opinion, the user should be allowed to get track record from app easily if they want to extract deeper information from the data. Compared to ask user search the exported file in his device, automatically calling the email client is considered as a more practical way to send user track records.

- Distance Calculation

When I calculated the moving distance, how to determine whether two locations belong to the same track record becomes a real problem. To solve this, I decided that if the difference of record time of two locations is less than 10 seconds, then regarded them as the same track.

I will use a simple example to clarify this clearer: Assume A,B,C,D are 4 locations and the number above the arrow indicate the difference of recorded time between 2 locations, which means user spend 6 seconds move from A to B in both images. According to my design, the second image is regarded as a simple track A->B->C->D so the calculated distance is the same as the moving distance from A to D. However, the first image will be regarded as two tracks :the track A-> B and the track C->D because the time difference between B and C is larger than 10 seconds. Then the actual tracked distance between A to D equals the distance between A to B plus the distance between C to D.



- Show the past route

In the "TrackHistory" activity, I listed all the location information from the database to help the user remember where he had been to. However, I think the latitude and longitude information is not intuitive enough, that is the reason why I show the past route on google map to give the user a clearer past footprints. Each track is marked by different color and user could pick the date to check the places he had been to before.

## Libraries Used

A library called Opencsv helps me export the track records from SQLitedatabase. Opencsv is a simple CSV parser for Java under a commercial-friendly Apache 2.0 license. The official website of this library is <http://opencsv.sourceforge.net> .