pedometer based on Android

# Abstract

With the development of sensors, smartphones are an ideal tool for monitoring and promoting physical exercise (Harries Tim, 2016). Smartphones are becoming the sensor hubs in a way, opening a rich experience for users. There are multiple different types of sensors which are built in android, such as motion sensors, position sensors and environment sensors (Google, 2018).

# Introduction

There are different ways to calculate the number of steps taken by the user. The first method is based on accelerometer using the Pythagorean theorem to calculate the magnitude of the acceleration vector of each sample. In this way, a service has to keep the device awake for calculating the steps (Martin Mladenov, 2009 ). In other words, the program can prevent the device from going into sleep during a lifetime of your service. But this approach will drain the battery fast. Another different way is built-in step counter and step detector sensor, which internally uses an accelerometer, but Android still treats them as logically separate sensors. Meanwhile, sensors are high battery optimized and consume very low power (Nagpa, 2016). Its high accuracy has been verified.

The report will discuss how to change the existing code from accelerometer to built-in step. Now, there are two different choices. Firstly, the outdated code can be refactored from accelerometer to built-in step counter. Another choice, the existing code can be abolished, the new app will be rewritten.

In the report, the existing code has been accessed by three aspects. Firstly, the function of the application need be accessed properly. Secondly, the structure of application should be reasonable. Lastly, readable code is important as well such as enough comments. The author read the existing code and document, meanwhile, the writer will learn and consider the prospective method and structure to decide whether the outdated is useful.

# Environment

This chapter is mainly to study and introduce the needed platform for Android pedometer, and introduction of the needed configuration environment.

## developing environment of Android

The application of Android need to run based on Android environment. The following is the configuration requirement and installation of Android development environment.

The software of developing environment:

Operation system: Windows 10

Software: Android SDK Version 19+

IDE environment: Android Studio 3.0

JDK: Java Development Kit(JDK) 1.8

Testing phone: Android version 6.0 (MIUI 9.2)

# Existing system

## Analysis of existing system and technologies

## [Thunkable for Android](https://docs.thunkable.com/android/components/sensors/pedometer.html)

Thunkable for Android based on the step counter sensor, which can implement the counting steps when the users open their app all the time (Figure 1.1).

There are some different platforms, such as [MIT App Inventor](http://ai2.appinventor.mit.edu/?locale=en#6558156633276416). The app has been built (figure 1.2).it also has similar problems with another one (Figure 1.1). So, we will discuss them together.

The drawbacks of thinkable:

* The website do not provide the source code, I tried to use some tools decompiling the .apk file. However, the source code cannot load in Android studio. I also do some research for converting ours App to a Java Android app, but some existing tutorials are not convert the app, they are recreating their app in android studio.
* The accuracy has big problem in real life and will discuss below.
* Currently, thinkable apps generally don’t work in the background when the apps are not open.

The advantages of Thunkable:

* It saves the time to develop, and the developer will spend more time on designing interface.

In conclusion, according to the accuracy and limitation, I don’t think the Thunkable is reliable tools for our development.

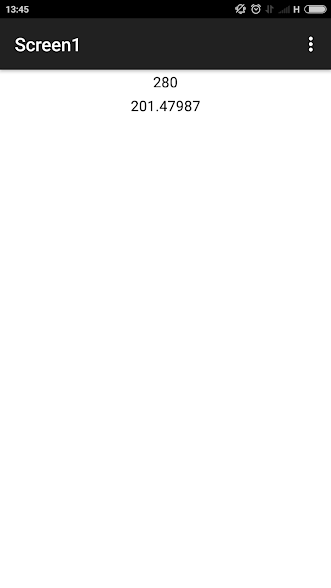
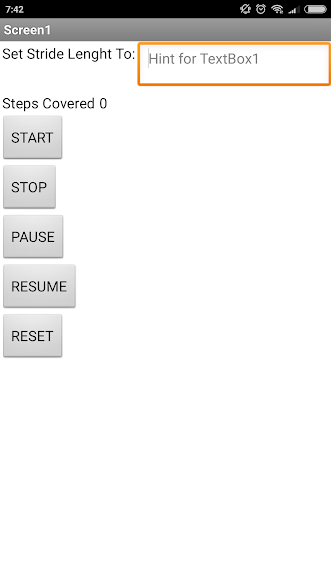
 

Figure 1.1 Figure 1.2

## [Pedometer (j4velin)](https://github.com/j4velin/Pedometer)

The project of pedometer has been published on google store. It includes a number of functions, such as the setting of step size, signing in with google account. It can allow users to use the achievements and leaderboards features(Figure 1.3).

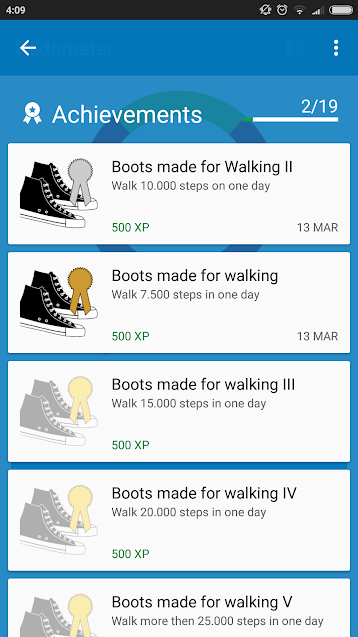
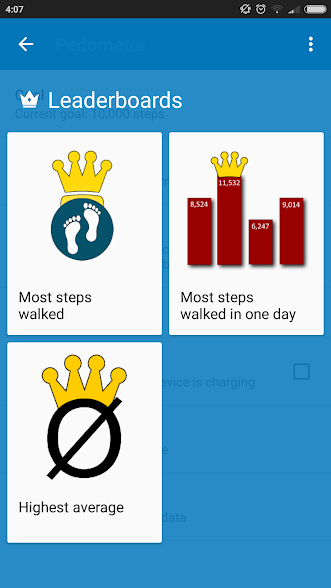
 

Figure 1.3

The advantage of App:

* The source code has been provided on GitHub.
* The author uses the EazeGraph for making the data clearly and understandable (Figure 1.4). Meanwhile, the chart is beautiful and fancy.
* The accuracy is acceptable. The different is that it doesn’t need open app everyday.

The disadvantage of App:

* the app has redundant function, such as achievements and achievements, which need log in your google account.

The main parts in this application is:

1. Service and Broadcast

|  |
| --- |
| <service  android:name=".SensorListener"/>  <receiver  android:name=".ShutdownRecevier">  <intent-filter>  <action android:name="android.intent.action.ACTION\_SHUTDOWN"/>  <action android:name="de.j4velin.ACTION\_SHUTDOWN"/>  </intent-filter> </receiver> |

The sensor listener service is major service for the application. There is another widget service in application, but I omit this redundant service. Actually, the most important things for pedometer is that the service prevents the data loss at shutdown, because the step counter sensor provides the number of steps since the last reboot while the sensor was activated, this is to say that the steps are going to become zero after rebooting the phone.

1. Database

|  |
| --- |
| @Override public void onCreate(final SQLiteDatabase db) {  db.execSQL("CREATE TABLE " + *DB\_NAME* + " (date INTEGER, steps INTEGER)"); } |



Figure 1.4

## [TodayStepCounter (jiahongfei)](https://github.com/jiahongfei/TodayStepCounter)

The step counter has been built in Chinese (Figure 1.5). It is integrated project in GitHub, and the author claims his project can run in many different versions in Android, he also built the steps counter service as a standalone module (Figure 1.4). But in his service, it includes two different methods to count steps in order to applying for low version android. Jiahong’s app implemented the function of pedometer based on accelerometer and step counter. In other words, his project can decide on which sensors will be used.

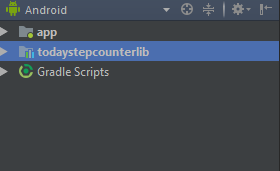


Figure 1.4

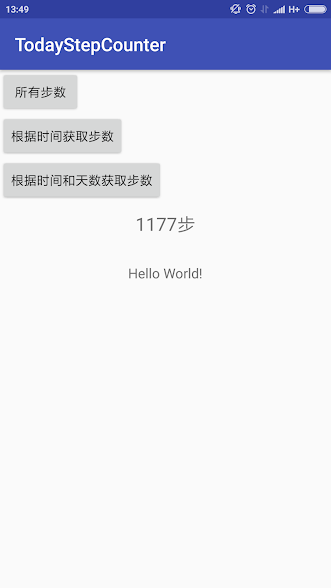


Figure 1.5

## [Pedometer(appypie)](https://www.appypie.com/)

The website did not provide the source code, and its accuracy is worse than thinkable. Meanwhile, the application (Figure 1.6) are running slowly with annoying Pop-up ads. We won’t discuss it anymore.



Figure 1.6

## [Mytestdemo(Steps)](https://github.com/kangqiwang/Steps)

It automatically starts counting as soon as the phone is powered on. But we need deal with the total number of steps taken by the user since the last reboot (power on) of the phone. I create Steps and implemented it with the SensorEventListener interface, so that it can receive the sensor events. I used a TextView to display the total number of steps taken and update its latest value.

At next stage, The new app will consist of three major components of namely service, SQLite database and activity. The service will be used to listening to all the individual step details using the step counter sensor. When the app is in the background, All the individual step details will be store in the SQLite database and the activity will be used to display the list of total number of steps along with dates.

# Access the accuracy



The table provides an information about the accuracy of some software. The google fit have been list in order to becoming a standard based on step counter sensor. According to these data, the step counter sensor has the limited speed. In other words, the fast speed causes the unacceptable accuracy. The number of counted step in Pedometer-j4velin is same with google fit and my demo, however my demo only read the total number of step counter, I have to calculate the steps manually. The good news is that I am going to finish the new version of my demo.

As for thinkable app for Android, the result is same with other applications only one time, it is because that the mobile phone keeps lighting up during the time I walked.

Todaystepcounter should be opened before counting steps. Otherwise, the step counter did not collect data form sensors in our test. I think the main problems are that the service may be killed by system. Because there are not receivers for the application. For example, in Pedometer (j4velin):

|  |
| --- |
| <receiver  android:name=".ShutdownRecevier">  <intent-filter>  <action android:name="android.intent.action.ACTION\_SHUTDOWN"/>  <action android:name="de.j4velin.ACTION\_SHUTDOWN"/>  </intent-filter> </receiver> |

But in todaystepcounter

|  |
| --- |
| <receiver android:name=".TodayStepShutdownReceiver" >  <intent-filter>  <action android:name="android.intent.action.ACTION\_SHUTDOWN" />  </intent-filter> </receiver> |

In todaystepcounter, the author did not provide a broadcast for listening to kill its own process. In other words, the process may be killed by the system without effective solutions.

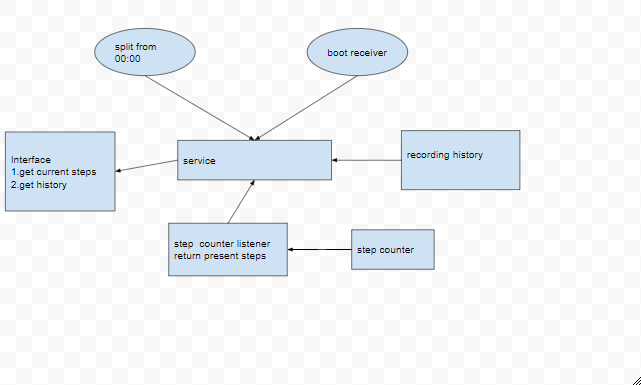
The most accurate step counter is the pedometer- j4velin and my demo and jiahong’s project also has great performance. But the perform of thinkable project is unacceptable.

# Feasibility analysis

1. The development of sensor: the availability of the software-based sensors is more variable because we often rely on one or more hardware sensors to derive their data. There are two step sensor (step counter sensor and step detector sensor). The step counter sensor provides the number of steps, and the step detector sensor triggers an event each time the user takes a step.
2. The latest version of bActive provides the structure and function of program, however a number of functions are redundant at the existing code. Starting form Android 3.1, all components in an app are prevented from running (receiving broadcasts) until the app is explicitly started by the user. In other words, there are four services stopped by system.
3. There are many open source projects to learn for me, such as todaystepcounter by jiahong and pedometer by j4velin. the author(jiahong) has been built a standalone module for his project and j4velin claims that his project is lightweight pedometer app using the hardware step-sensor for minimal battery consumption.
4. The client – server framework called Retrofit is for exchanging data between an app and a backend server, which is our backend interaction framework. Realm is powerful framework for mobile databases. It replaces the heavy and cumbersome SQLite and is much simpler and faster. EazeGraph is an Android library for creating beautiful and fancy charts. It can help us build lightweight and beautiful interface. We also learnt to use ACRA, which is a library enabling android application to automatically post their crash reports to report server.

# System design

The flowchart provides an information about the idea of new structure based on the method of step counter. But some functions are omitted such as debug logger and traffic logger. In comparison with the existing application, the main idea is similar. Both of them separate the views, the main algorithms, and services. But I would like to build a module for the Pedometer module and provide an interface, which is easier to test and maintain



## System function

The step counter function:

* some users can check their own history based on his authority.
* some users can check others’ history based on his authority.
* some users are not allowed to check others’ history and theirs.
* All user’s data and history will update on our server for researching.

## system structure

Receiver:

There are two important receivers:

1. Phone shutdown

Application should listener the action of phone shutdown, which prevents the data loss

1. Application shutdown

Application should prevent its service has been killed.

Database:

1. user

the user can create, read, update and delete the all the data

1. steps

the number of steps.

1. Date

The date will save for inquire.

1. Friends

It is for inquiring other users’ data.

1. management authority

defined by int

Activity:

There are three main activities in our project:

1. Personalsteps activity

It displays the number of steps today and at this week, which can easily make user know their steps and latest history.

1. Setting activity

It displays some basic settings such as step size.

1. Rankactivity

It displays the users’ ranking / other’s average steps, some of users cannot access the RankActivity.

# Conclusion

In my point of view, the rewriting code is a more suitable choice in this situation due to redundant functions. The new project will consist of the existing function and the chart, graph and accuracy of j4velin’s project and the standalone module of jiahong’s project. I am confident in developing application on Android after two weeks of effort.

# References

Google, 2018. *Android API.* [Online]   
Available at: https://developer.android.com/guide/topics/manifest/uses-sdk-element.html  
[Accessed 9 2 2018].

Harries Tim, E. P. R. R., 2016. Effectiveness of a smartphone app in increasing physical activity amongst male adults: a randomised controlled trial. *BMC Public Health,* Volume 1, p. 925.

Martin Mladenov, M. M., 2009 . *A Step Counter Service for Java-Enabled Devices Using a Built-In Accelerometer.* Dublin, Proceedings of the 1st International Workshop on Context-Aware Middleware and Services: affiliated with the 4th International Conference on Communication System Software and Middleware (COMSWARE 2009).

Nagpa, V., 2016. *Android Sensor Programming By Example.* 1st ed. Birmingham: packt publishing.