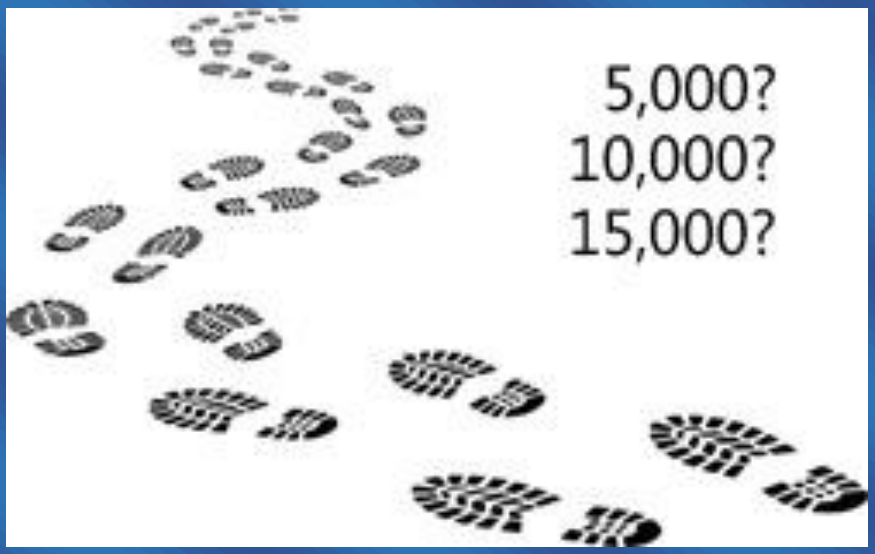
A

Project Report on

# "Designing Step Counter using Smart Phone Accelerometer"




Project submitted
In partial fulfilment of the requirements for the degree of

**MASTER OF COMPUTER APPLICATIONS**

By
**SAMRAT LAHIRI**
**Class Roll No: 001510503026**
**Exam Roll No: MCA186024**
**Registration No: 133688 of 2015 – 2016**

Under the supervision of
**DR. CHANDREYEE CHOWDHURY**

**Department of Computer Science and Engineering**
**Faculty of Engineering and Technology**
**Jadavpur University**
**Kolkata – 700 032**
**India**

# A
# Project Report on

## “Designing Step Counter using Smart Phone Accelerometer”

Project submitted
In partial fulfilment of the requirements for the degree of

**MASTER OF COMPUTER APPLICATIONS**

By
**SAMRAT LAHIRI**

**Class Roll No:** 001510503026
**Exam Roll No:** MCA186024
**Registration No:** 133688 of 2015 – 2016

Under the supervision of
**DR. CHANDREYEE CHOWDHURY**

**Department of Computer Science and Engineering**
**Faculty of Engineering and Technology**
**Jadavpur University**
**Kolkata – 700 032**
**India**

# Jadavpur University
## Faculty of Engineering and Technology
### Department of Computer Science & Engineering

## TO WHOM IT MAY CONCERN

*This is to clarify that the project entitled* ***"Designing Step Counter using Smart Phone Accelerometer"*** *has been completed by* ***Samrat Lahiri.*** *This work is carried out under the supervision of* ***Dr. Chandreyee Chowdhury*** *in partial fulfilment for the award of the degree of* ***Master of Computer Application*** *of the department of* ***Computer Science and Engineering, Jadavpur University,*** *during the session 2017-2018. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.*

(Signature of the Head of the Department)
Dr. Ujjwal Maulik, Head of the Department of Computer Science and Engineering

(Signature of the project supervisor)
Dr. Chandreyee Chowdhury

(Signature of the Dean of the Faculty)
Prof. Chiranjib Bhattacharjee, Dean, Faculty Council of Engineering and Technology

# Jadavpur University

## Faculty of Engineering and Technology

### Department of Computer Science & Engineering

## CERTIFICATE OF APPROVAL

*This is to clarify that the project entitled* ***"Designing Step Counter using Smart Phone Accelerometer"*** *has been completed by* ***Samrat Lahiri.*** *This work is carried out under the supervision of* ***Dr. Chandreyee Chowdhury*** *in partial fulfilment for the award of the degree of* ***Master of Computer Application*** *of the department of* ***Computer Science and Engineering, Jadavpur University,*** *during the session 2017-2018. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.*

(Signature of the external examiner)

(Signature of the internal examiner)

# DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

*I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.*

*I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices.*

*I declare that this is a true copy of my thesis, including any final revisions, and that this thesis has not been submitted for a higher degree to any other University or Institution.*

**Candidate's Name:** Samrat Lahiri
**Class Roll No:** 001510503026
**Exam Roll No:** MCA186024
**Project Title:** Designing Step Counter using Smart Phone Accelerometer

**Date:** _____ / _____ / _____

**Place:** _____

_____
(Signature of the candidate)

# ACKNOWLEDGEMENTS

*I would first like to thank my thesis advisor and also the project supervisor,* ***Dr. Chandreyee Chowdhury*** *of the department of* ***Computer Science and Engineering*** *of* ***Jadavpur University****. The door to his office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work and steered me in the right direction whenever he thought I needed it.*

*I would also like to thank the experts who were involved in the validation survey for this research project:* ***Dr. Ujjwal Maulik,*** *the* ***Head of the Department*** *of* ***Computer Science and Engineering.*** *Without his passionate participation and input, the validation survey could not be successfully conducted.*

*Finally, I must express my very profound gratitude to my parents and also to my friends cum colleagues for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not be possible without them. Thank you.*

**Date:** _____ / _____ / _____

**Place:** _____

_____
(Signature of the candidate)

# TABLE OF CONTENTS


| CONTENTS | PAGE NO. |
|---|---|
| • ABSTRACT | 1 |
| • INTRODUCTION | 2 |
| • CHAPTER 1 : PEDOMETER | 3 |
| ○ What is a PEDOMETER? | 3 |
| ○ Mechanical Pedometer | 3 |
| ○ Working of a Mechanical Pedometer | 4 |
| ○ Electronic Pedometers | 5 |
| ○ Technology | 5 |
| ○ Study : Pedometers on Smartphones or Wearables | 6 |
| ○ Wearables vs Smartphones Applications | 7 |
| ○ Step Tracking works best on newer phones | 7 |
| ○ Google Fit on Android Phones | 8 |
| ○ Accuracy | 8 |
| • CHAPTER 2 : ACTIVITY RECOGNITION | 9 |
| ○ What is ACTIVITY? | 9 |
| ○ Receive information about activities using minimal resources | 9 |
| ○ Types of Activities | 10 |
| ○ Process of Activity Recognition | 11 |
| ○ Android Sensors | 12 |

- CHAPTER 3 : STEP DETECTION 13
  - How does a Pedometer detect/count steps? 13
  - The Nature of Walking 14
- CHAPTER 4 : ANDROID STUDIO AND MAKING THE APPLICATION 15
  - Intro to Android Studio 15
  - Making an Application 15
  - Basic Details of the Application 16
  - Purpose of the Application 16
  - Description of the files used 17
  - Understanding the Code 17
  - Features of the Application 19
- CHAPTER 5: TESTING THE APPLICATION 21
  - Results 21
  - Testing 22
  - Other Tests 25
- CONCLUSION 29
- REFERENCES 30

# ABSTRACT

Step Counters are used to count the number of steps of a person and display the results as counting is not possible for anyone. A step is counted whether a person is running, walking or jogging. Whenever a step is taken by a person, the counter increases.

It was generally done by using a wearable device, for example, a device in our wrist or wrapped around the knee. This helps us in counting steps. Today such applications are available in our smart phones which counts steps using Accelerometer sensor available in the smart phones. So, a person does not need to have a wearable with him/her in order to count the number of steps. He/She can use the smart phone, which is almost owned by everyone today, to count the steps taken.

We have designed an application which is found to count 301 steps out of 336 steps actually taken.

# INTRODUCTION

## Definition of PEDOMETER

An Application that is installed, usually in a wearable device, and records the distance a person covers on foot by responding to the body motion at each step.

In this kind of Applications, steps are counted by using the Accelerometer sensor data collected using the Android devices. These Applications increase the step counter each time they receive data from the Accelerometer sensors. But the problem which occurs is that it even counts steps when the user is just moving his/her hand with the device in his/her hand and data is being read from the Accelerometer sensor. So, to improve the working of these Applications, Google's Activity Recognition API has been used, which at first reads whether the user is Walking, Running or Jogging and if it is true, then the Step Counter will count steps.

The Google Activity Recognition API will enable us to detect users' activity using low power signals from multiple sensors in the Android devices. It automatically detects activities by periodically reading short bursts of sensor data and processes them by using machine learning models. Google developers provides APIs, software development tools, and technical resources that allows communication with Google Services and their integration to other services.

# CHAPTER 1

# PEDOMETER

## What is a PEDOMETER? [1]

The pedometer is a portable device that is used to measure a person's physical activity level throughout the day. Pedometers are budget-friendly and reliable. It is a great motivational tool. These were generally designed to be worn at the waist, but could also be worn in other body parts, like the wrist and the ankle. Now-a-days, they are generally worn in the wrist. The key element of using such a device regularly is that it should be worn in the same location regularly to keep track of the activities of the daily life of the device's user like transportation, recreational activity, but mainly fitness activities. Family members can motivate each other much easily if a device is present to keep track of the daily activities.

Made originally for sports and fitness enthusiasts, Pedometers are becoming very popular these days. It is being used in a significant scale today. Generally, they are used to track the number of steps taken by a person on a particular day. It helps a person because it allows him/her to keep track of the estimated amount of physical work done in a day and also at the same time keep track of the distance covered in a day.

## Mechanical pedometers [2]

Early pedometers were entirely mechanical and they worked kind of like a pendulum clocks (A clock with a swinging bar which is powered by a slowly falling weight). As the pendulum rocks back and forth, a kind of see-saw lever, called an escapement, flicks up and down. Then a gear wheel inside the clock (which was used to count seconds) advanced by one position. Now, a pendulum clock is a mechanism which is used to count seconds. In the original pedometers, a swinging pendulum was used to count steps and the count was displayed with a pointer that moves round a dial (Just like an analog watch). It was worn on the waist. Every time a step was taken, the pendulum used to swing to one side then back again. It caused a gear in the device to advance one position and move the hand around the dial.

## Working of Mechanical Pedometer [2]

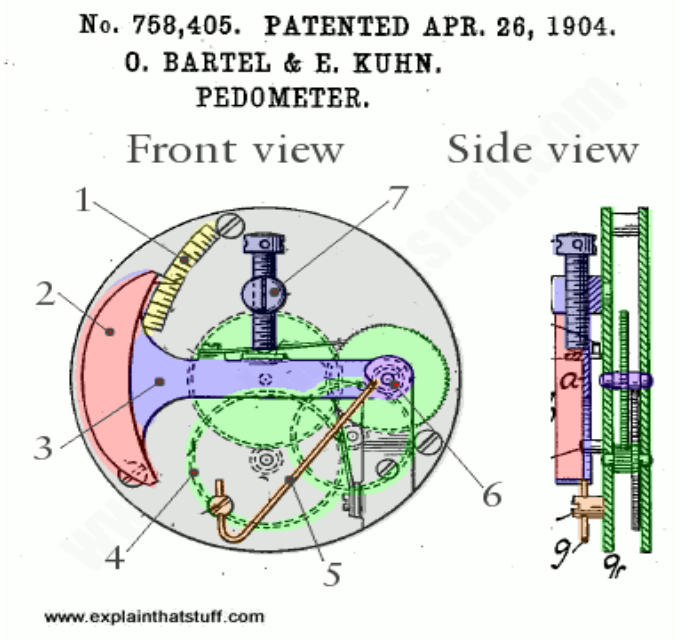



*Figure 1:Design of a mechanical pedometer (Source: https://cdn4.explainthatstuff.com/pedometer-patent-bartel-kuhn.png)*

Inexpensive modern pedometers are partly mechanical as they register steps using a swinging pendulum, and partly electronic as they use microchips to count the steps that the pendulum detects. Early pedometers used to be totally mechanical.

The working:

1. A pointer and scale at the top helps you adjust the pendulum accurately.
2. A weight (red) at the end of the pendulum (blue) ensures it swings back and forth by a measurable amount as shown in Fig. 1.
3. The pendulum swings as you walk. On this diagram, the pendulum would move up and down but, in reality, it would hang downward way and swing from side to side.
4. The swinging pendulum turns various gears (green), advancing a pointer that indicates your step count (not shown).
5. A spring keeps the pendulum tight. The spring is part of the adjustment mechanism for different step sizes.
6. The pendulum swings about this pivot (purple).

The large screw was used to adjust/regulate the device for different step sizes.

## Electronic pedometers [2]

The pedometers used today works in a way that is similar to the Mechanical Pedometers but are partly electronic. A metal pendulum (a hammer with a weight on one end) is wired into an electronic counting circuit by a thin spring. Normally, the circuit is open and no current flows through it. But as soon as a step is taken, the hammer swings and touches a metal in the centre. This allows current to flow and adds one to the step count. As the step is completed, the hammer swings back to the original position by the help of the spring and the device is ready for the next step.

More sophisticated pedometers are being made today which work entirely electronically. They are much more reliable, accurate and long-lasting. They use two or three accelerometers to count step rather than a pendulum. There are microchips, arranged at right angles, which can detect very minute changes too. Since accelerometers are found in smart phones, there are numerous applications found that can count the number of steps. GPS satellite navigation devices also has the capability to count steps by figuring out whether the user is running or walking. This is done by calculating satellite signals rather than counting the number of steps.

## Technology [2]

The technology for a pedometer includes a mechanical sensor and software to count steps. Early forms just had a mechanical sensor. When one shook those devices when it was in hand, one could hear a lead ball. Now, the advanced step counters rely on MEMS inertial sensors and advanced software to detect steps. MEMS sensors are more accurate and gives lesser false positive. The software used varies widely and each has their own way of calculating "sensible steps". Modern step counters are expected to count steps more accurately in places where they are usually kept, like in shirt or pant pockets.

## Study: Pedometers on Smartphones or Wearables? [3]

A small study conducted by researchers at the University of Pennsylvania shows that Activity tracking applications in the smart phones gives more accurate results than the wearables used. Interestingly, the study found that clip-on pedometers were the most accurate.

"Since step counts are such an important part of how these devices and applications measure physical activity, including calculating distance or calories burned, their accuracy is key," senior author and assistant professor at the University of Pennsylvania, Mitesh Patel, said in a statement. "Compared to the one to two percent of adults in the US that own a wearable device, more than 65 percent of adults carry a smartphone. Our findings suggest that smartphone applications could prove to be a more widely accessible and affordable way of tracking health behaviours."

The researcher included 14 users who wore one pedometer and two accelerometers on their waistband, three wearable devices on their wrist, and two different smartphones in their pockets -- one running three applications and the other running one -- during the study.

One of the smartphones used was the iPhone 5S. It ran the applications from Fitbit, Withings, and Moves. The other phone, Galaxy S4 in which the Moves application was running. The three wearable devices were Nike FuelBand, Jawbone UP24, and Fitbit Flex and the pedometers used were Fitbit One, Fitbit Zip, and Digi-Walker SW-200.

Each participant walked on a treadmill set at three miles per hour for 500 steps and 1,500 steps two times. An observer counted the steps using a tally counter.

Researchers found that the relative difference in mean step count ranged from -0.3 percent to 1 percent for pedometers, -6.7 percent to 6.2 percent for the applications, and -22.7 percent to -1.5 percent for wearable devices. The Nike FuelBand was the least accurate wearable tracker as the device reported step counts more than 20 percent lower than observed.

This study was funded, in part, through a grant from the National Institute on Aging.

## Wearables vs. Smartphone Applications: Which Are Better to Count Steps? [4]

So, the above study raises the question that if smart phones are better for counting steps, why use an extra wearable device? Today there is a constant debate on step counting in wearables vs in smart phones. Study suggests that smart phone applications uses sensors which can detect steps more accurately than any wearables. The best applications should have the accuracy within 5% error. Gradually the error is decreasing, but then why use wearables. The problems with smart phones is that it uses the concept of seamless connectivity to count the actual number of steps. But, most countries lack this feature of proper network connectivity. So, as far as seamless internet connectivity is concerned, the smartphone applications count steps much more efficiently than the wearables. But else, wearables can be used to count steps in a better way.

## Step Tracking Works Best on Newer Phones [5]

Newer phones have better sensors and can be used to track activities much more efficiently. As the technology improves, it is installed in newer phones. The step counting algorithm is also improving by each passing day allowing the users to get more accurate and reliable results.

In the Android phones, it's a bit more complicated. Google Fit will attempt to work even on older Android phones, but it will work most accurately — and with the least battery drain — on newer phones that include these low-power sensors. As a Google Fit engineer explained on StackOverFlow:

> We periodically poll accelerometer and use Machine Learning and heuristics to correctly identify the activity and duration. For devices with hardware step counters, we use these step counters to monitor step counts. For older devices, we use the activity detected to predict the right number of steps.

Information can be accessed by tapping the "Health" application icon on your home screen.

## Google Fit on Android Phones [6]

Google Fit is Google's competitor to Apple Health. It is included in some newer version of Android phones. It can also be downloaded from Play Store. It has multiple options which can be set by the user according to their preferences. Also, the data can be viewed from the web to get a better result and keep track of daily activities.

This information is tied to your Google account, so you can also access it at Google Fit on the web.

Wearables track the major step taken in a day. But the phone needs to be connected to the wearables to also count the steps that we casually take in the house or in our office like climbing up and down stairs, walking from one room to the other.

## Accuracy [1]

Counting steps with a pedometer sounds very scientific but one should remember that it is generally never the actual number of steps that is counted. Rather it is an approximation. One needs to assume that all steps will never be counted. The applications that are used to count steps may be good, but they never give 100 % accuracy. The best pedometers claims that the error will not be more than 5 percent.

For a pedometer to work correctly, it needs to be fixed to the waist of a subject—and not put it in the pocket—because a pedometer needs to detect the side-to-side tilting motion of the body to register each step correctly. Most devices come with a belt clip, making it easy to attach them properly.

# CHAPTER 2

# ACTIVITY RECOGNITION

## What is ACTIVITY? [7]

Activity is any type of work done by a person. Activity Recognition is a process of detecting any kind of activity of a person. The main challenge is recognizing the activities correctly. There are many similar kinds of activities. Those activities should be displayed properly. Also, the activity should be actual. The result should not be shown if the activities are imitated.

## Receive information about activities using minimal resources [8]

### *Users get notified when they start or end a particular activity*

Some applications are interested in knowing when a user starts or stops a specific activity. For example, a mileage tracking application could start tracking miles when a user starts driving, or a messaging application could mute all conversations until the user stops driving.

The Activity Recognition Transition API enables these scenarios by performing the heavy lifting associated with detecting changes in the user's activity. The applications subscribe to a transition in activities of interest and the API notifies the application only when needed. One does not need to implement complex heuristics to detect when an activity starts or ends.

### *Perform an action when an application receives activity information*

The Activity Recognition API delivers its results to a callback, which is usually implemented as an IntentService in an application. The results are delivered at intervals that is specified, or the application can use the results requested by other clients without consuming additional power itself.

One can tell the API how to deliver results by using a PendingIntent, which removes the need to have a service constantly running in the background for activity detection purposes. The application receives the corresponding Intentsfrom the API, extracts the detected activities, and decides if it should take an action. Invoking the service only when an activity is received preserves resources, such as memory.

## *Receive detected activities that include a confidence grade*

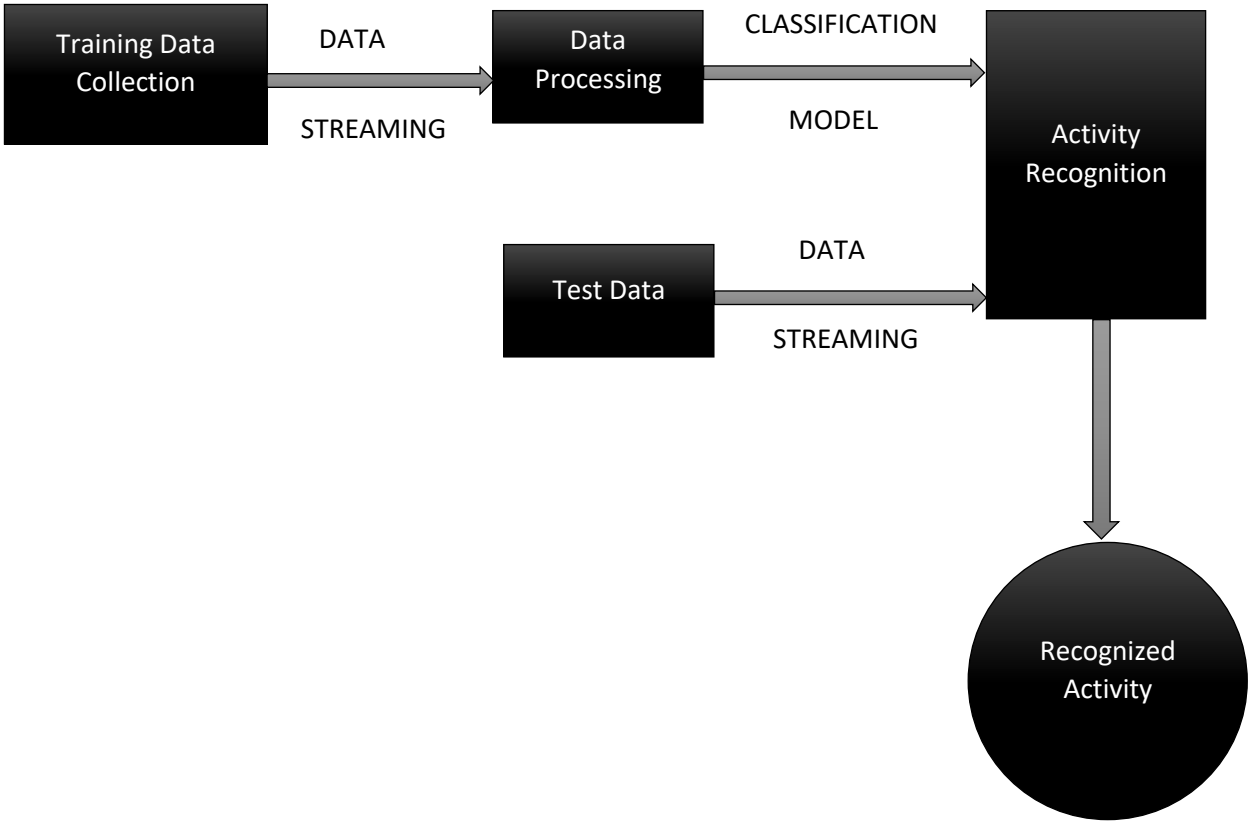
The Activity Recognition API does the heavy lifting by processing the signals from the device to identify the current activities. The application receives a list of detected activities, each of which includes confidence and type properties.

The confidence property indicates the likelihood that the user is performing the activity represented in the result. The type property represents the detected activity of the device relative to entities in the physical world, for example, the device is on a bicycle or the device is on a user who is running.

## Types of Activities in Real Life [9]

| | CATEGORY | ACTIVITY TYPE |
|---|---|---|
| Complexity | Simple Activity | Walking, Jogging, Sitting, Standing, Lying, Walking Upstairs, Walking Downstairs, Jumping, Taking escalator up, Taking escalator down, Taking elevator up, Taking elevator down. |
| | Complex Activity | Shopping, Taking buses, Driving a car. |
| Scenario | Living Activity | Brushing Teeth, Vacuuming, Eating, Cooking, Washing hands, Meditation, Clapping hands, Watering plants, Sweeping, Shaving, Blow drying hair, Washing dishes, Ironing, Flushing the toilet, Cleaning. |
| | Working Activity | Working on PC, Talking on the phone, On a break, Meeting, Typing, Writing, Doing a presentation. |
| | Health Activity | Exercising, Fall, Rehabilitation activities, Following routines. |

## Process of Activity Recognition [9]




The above diagram is a representation of how Activity is recognised in an electronic device like a phone or in a watch.

## Android Sensors [9]

| SENSOR | DESCRIPTION |
|---|---|
| Accelerometer | Measure the acceleration force that applied to the device, including force of gravity. |
| Ambient temperature Sensor | Measure the ambient room temperature. |
| Gravity Sensor | Measure the force of the gravity that applied to the device, in three axes (x,y,z). |
| Gyroscope | Measure the device's rotation in three axes (x,y,z). |
| Light Sensor | Measure the ambient light level (illumination) |
| Linear Acceleration | Measure the acceleration force that applied to the device, force of gravity is excluded. |
| Magnetometer | Measure the ambient geomagnetic field in three axes (x,y,z). |
| Barometer | Measure the ambient air pressure. |
| Proximity Sensor | Measure the proximity of an object relative to the view screen of a device. |
| Humidity Sensor | Measure the humidity of ambient environment. |

# CHAPTER 3

# STEP DETECTION

## How does a Pedometer detect/count steps? [10]

- As example of processing Accelerometer data
- Walking or running results in motion along the 3 body axes (forward, vertical, side) as shown in Fig. 2 and Fig. 3
- Smartphone has similar axes

  Alignment depends on phone orientation

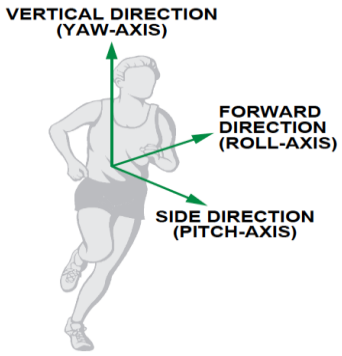



*Figure 2: Motion along 3 axes of the accelerometer (source:http://web.cs.wpi.edu/~emmanuel/courses/cs528/F17/slides/lecture06b.pdf)*

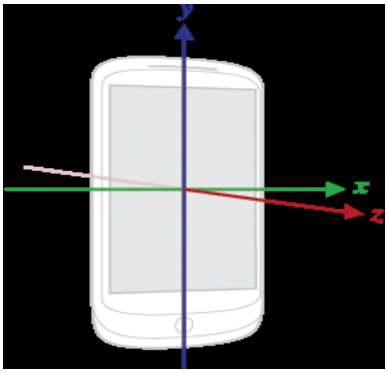



*Figure 3: Accelerometer axes with respect to a smartphone (source: http://web.cs.wpi.edu/~emmanuel/courses/cs528/F17/slides/lecture06b.pdf)*

-

# The Nature of Walking [10]

- Vertical and forward acceleration increases/decreases during different phases of walking
- Walking causes a large periodic spike in one of the accelerometer axes

- The magnitude depends on phone orientation

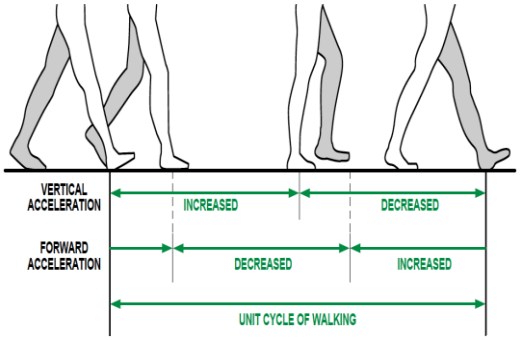



*Figure 4: Vertical and Forward acceleration (source:http://web.cs.wpi.edu/~emmanuel/courses/cs528/F17/slides/lecture06b.pdf)*

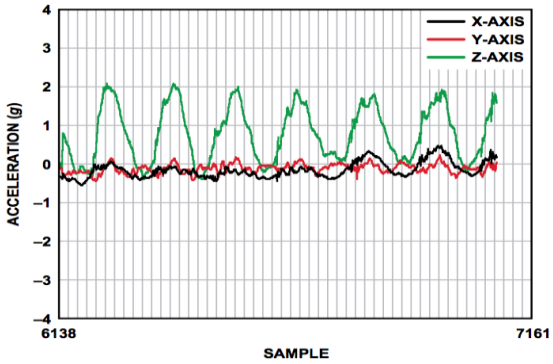



*Figure 5: Raw data collected from accelerometer (Source: http://web.cs.wpi.edu/~emmanuel/courses/cs528/F17/slides/lecture06b.pdf)*

# CHAPTER 4

# ANDROID STUDIO AND MAKING THE APPLICATION

## Intro to Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.1 released in March 2018.

Android Studio supports all the same programming languages of IntelliJ, and PyCharm e.g. Python, and Kotlin; and Android Studio 3.0 supports "Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features.

## Making an Application with Android Studio

- To make an application we need to first install the software '**Android Studio**'
- Then we need to click on the option '***Create a New Project***'
- A dialog box appears. We give a suitable **Application Name** and choose location for storing the project and Click **Next**.
- Now we select the minimum API version in which we want the application to work. Currently selecting **API 14 (Android 4.0 i.e. Ice-Cream Sandwich)** gives access to almost **100%** of the Android devices used around the globe. We can also select a higher minimum version and we click **Next**.
- Then we select an **Activity Layout**. Generally, we select '***Empty Activity***' but we can also choose others if needed.
- At the end we give the main activity and layout files a name. It is a convention to name the **Main Activity** class '***MainActivity***' and the main Layout file '***activity_main***'
- We click next and the gradle build takes some times. Varies from machine to machine.

To build this Application, we have first follow the basic steps which has already been mentioned in the previous chapter.

## Basic Details about the Application:

***Name:*** Pedometer Activity

***Project Name:*** ActivityRecognitionPedometer

***Minimum API Version:*** API 14: Android 4.0 (Ice-Cream Sandwich)

***Activity Layout:*** Empty Activity

***Activity Class:*** MainActivity.java

***Layout File:*** activity_main.xml

## Purpose of the Application:

The sole purpose of the Application is to count the number of steps taken. But for better performance, we are using Activity Recognition API. Whenever an activity is detected by Activity Recognition API, the user can start tracking the number of steps taken, but the steps are counted only if Activity Recognition API detects Walking or Running. We have taken a particular threshold and a time interval but that can be changed as per the requirements of a phone's sensors or a person's activity in that case. The threshold and time interval is different for the Pedometer and the Activity Recognition API as Activity needs to be read much faster than a step.

## *Description of the files used*

- **StepListener.java:** This interface will listen to alerts about steps being detected.
- **SensorFilter.java:** This class will filter out values that has close approximation to steps.
- **StepDetector.java:** This class which will accept updates from **Accelerometer** sensor and deploys the filter to detect if a step has been covered by the user.
- **Constants.java**: To keep few global variables like activity frequency interval and confidence threshold.
- **DetectedActivitiesIntentService.java:** Extend it from **IntentService**. In ***onHandleIntent()*** method, list of probable activities will be retried and broadcasted using **LocalBroadcastManager**.
- **BackgroundDetectedActivitiesService.java:** We need this background service class which runs in background and triggers the activities in an interval basis.

## *Understanding the Code*

In **AndroidManifest.xml**,

```
<uses-
permissionandroid:name="com.google.android.gms.permission.ACTIVITY_RECOG
NITION"/>

<application

…

…

/application>
```

It allows the application to access the Activity Recognition API and display data accordingly.

In **AndroidManifest.xml**,

```
<application
<service
        android:name=".DetectedActivitiesIntentService"
        android:exported="false" />

<service android:name=".BackgroundDetectedActivitiesService"></service>
</application>
```

Service that provides activity recognition data. Setting the android:exported attribute to "false" stops other applications from starting this service, even when using an explicit intent.

In **MainActivity.java**,

```
ActivityRecognitionClient mActivityRecognitionClient =
newActivityRecognitionClient(this);

Task<Void> task = mActivityRecognitionClient.requestActivityUpdates(
            Constants.DETECTION_INTERVAL_IN_MILLISECONDS,
            mPendingIntent);

ActivityRecognitionResult result =
ActivityRecognitionResult.extractResult(intent);
ArrayList<DetectedActivity> detectedActivities = (ArrayList)
result.getProbableActivities();

for(DetectedActivity activity : detectedActivities) {
     Log.e(TAG, "Detected activity: "+ activity.getType() + ", "+
activity.getConfidence());
}
```

Used to detect the Activity via the **ActivityRecognitionClient** class.

In **StepDetector.java**,

```
private static final float STEP_THRESHOLD = 50f;

private static final int STEP_DELAY_NS = 250000000;
```

Used in to keep track of the Threshold and Time Delay between two successful steps. Can be changed according to the sensor activity data.

In **Constants.java**,

```
public class Constants {
public static final String BROADCAST_DETECTED_ACTIVITY =
"activity_intent";

static final long DETECTION_INTERVAL_IN_MILLISECONDS = 30 * 1000;

public static final int CONFIDENCE = 70;
}
```

Used in to keep track of the Threshold and Time Delay between two successful steps by the Activity Recognition API. Also, a Confidence is set. This means the base confidence for the display of an activity is 70. Can be changed according to the sensor activity data.

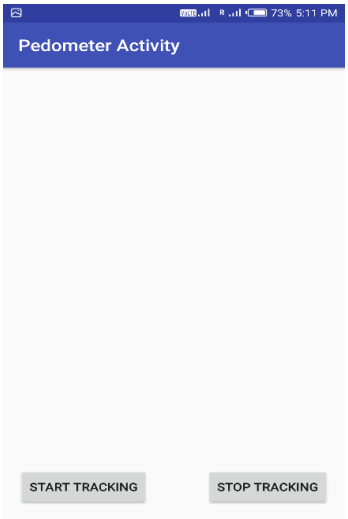
In application's **build.gradle**,

```
Dependencies {
….
implementation 'com.google.android.gms:play-services-location:12.0.0'
….
}
```

phone and display data according to the code.

## ***Features of the Application***

### **Default view of the Application:**



**Once Activity Recognition API recognises any Activity:**

**Buttons and their Purpose:**

- START TRACKING

  This Button is solely for the purpose of starting to track the Activity of the user using the Phone's sensors. However, when the application starts, the user physically does not need to start tracking as that will be done automatically. If the user wants to restart the recognition, then he/she needs to Stop Tracking and then Start Tracking again.

- STOP TRACKING

  This Button is used in case the user desires or requires to stop recognizing the activities. If the user quits the application, Tracking will be stopped in 15 seconds.

Whenever an Activity is detected, the following 2 buttons can be seen.

- START PEDOMETER

  The user needs to click on this Button in order to Start the Pedometer. It starts counting the steps

- STOP PEDOMETER

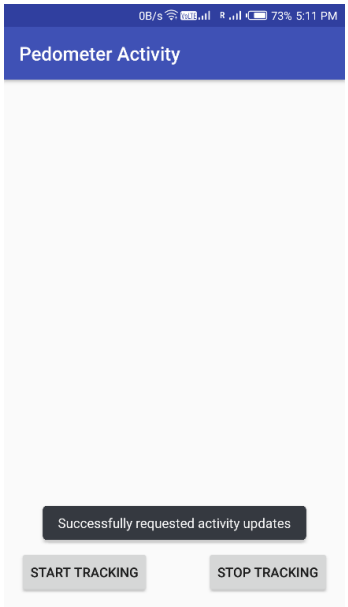
  The user can also Stop the Pedometer by clicking on this Button. It pauses counting and starts counting from that position itself unless the Activity Recognition is stopped.

# CHAPTER 5

# TESTING THE APPLICATION

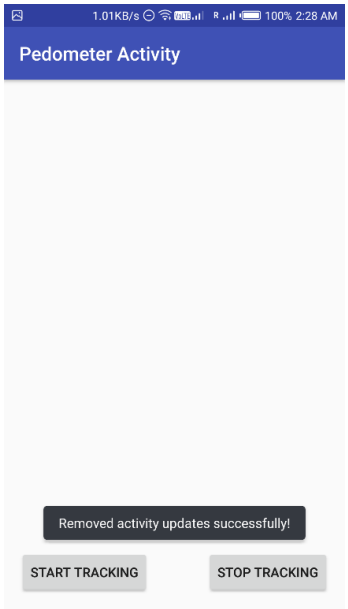
## RESULTS

### *When Application Starts:*




A pop-up notification "Successfully requested activity updates" appears indicating that the Activity Recognition API has started to recognize user's activities.

### *When the Tracking is stopped:*




A pop-up notification "Removed activity updates successfully!" appears indicating that the Activity Recognition API has stopped to recognize user's activities. It will also execute automatically after 15 seconds of exiting the Application.

The above results are found in each and every Android device which is API 14 or higher.

# TESTING

These are the test results from different devices.

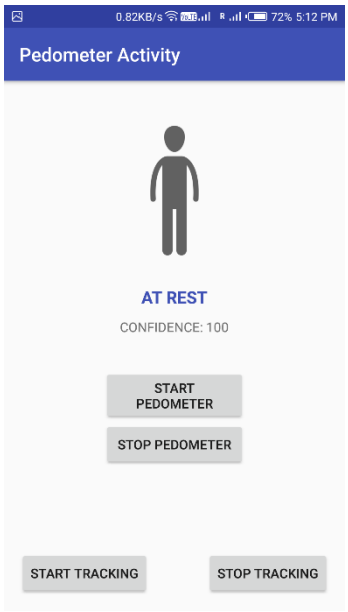
## ***Test 1:***

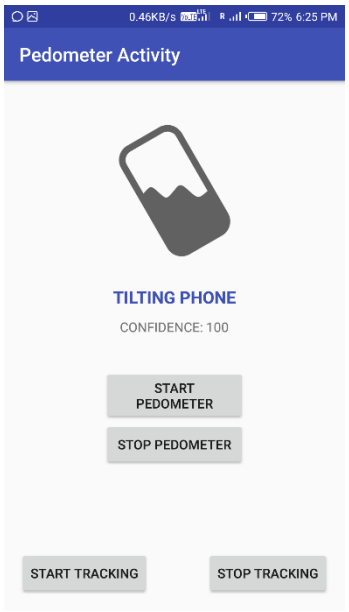
***Phone Model:*** Infinix Note 4

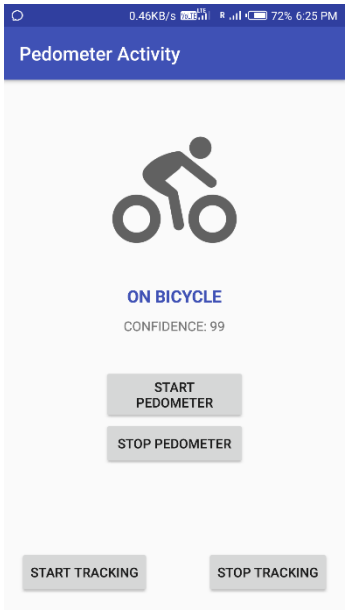
***Phone OS:*** XOS

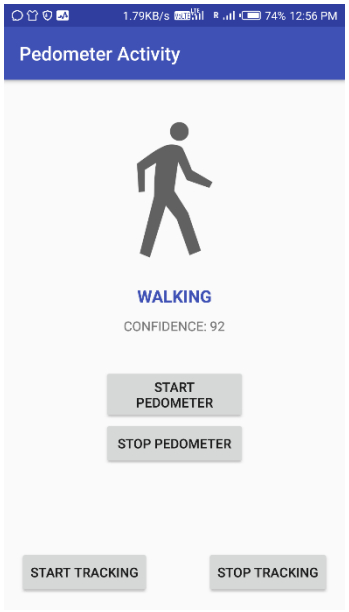
***Android Version:*** Android 8.1.0(API 27) / Android Oreo

***Other Details:*** 3 GB RAM, 32 GB ROM, 1.3 GHz Quad-Core Mediatek MT6753 Processor




The above test was conducted with the phone in the right pocket of the trouser.

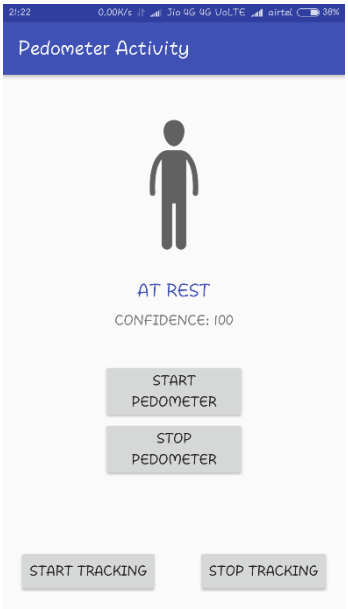
## ***Test 2:***

***Phone Model:*** Redmi Note 4

***Phone OS:*** MIUI

***Android Version:*** Android 7.0.0 (API 24) / Android Nougat

***Other Details:*** 4 GB RAM, 64 GB ROM, 2 GHz Octa-Core Qualcomm Snapdragon 625 Processor



The above test was conducted with the phone in the right pocket of the trouser.

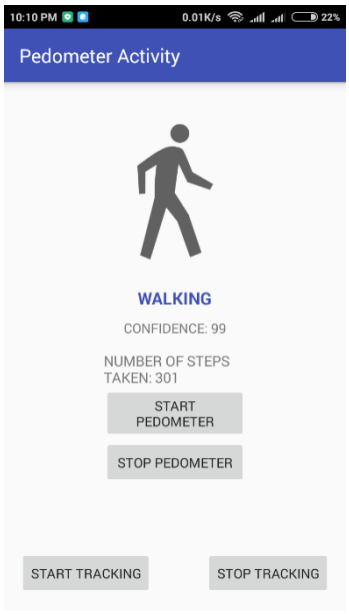
## ***Test 3:***

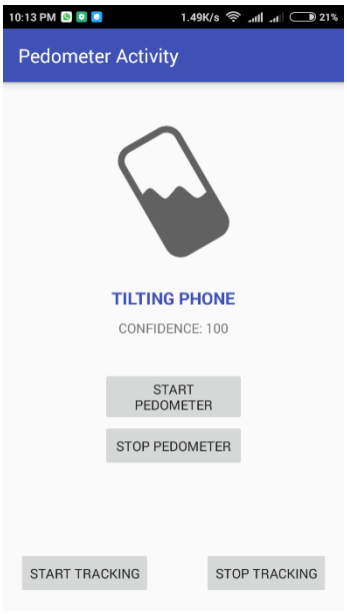
***Phone Model:*** Redmi 1S

***Phone OS:*** MIUI

***Android Version:*** Android 4.4.4 (API 27) / Android KitKat

***Other Details:*** 1 GB RAM, 8 GB ROM, 1.6 GHz Quad-Core Qualcomm Snapdragon 400 Processor




The above test was conducted with the phone in the left pocket of the trouser.

# Other tests

## *Test 1:*

***Left Pocket of Trouser***

***Phone Model:*** Infinix Note 4

***Phone OS:*** XOS

***Android Version:*** Android 8.1.0 (API 27) / Android Oreo

***Other Details:*** 3 GB RAM, 32 GB ROM, 1.3 GHz Quad-Core Mediatek MT6753 Processor

Used to create the csv file to track the steps at definite time intervals.

***Right Pocket of Trouser***

***Phone Model:*** Redmi Note 4

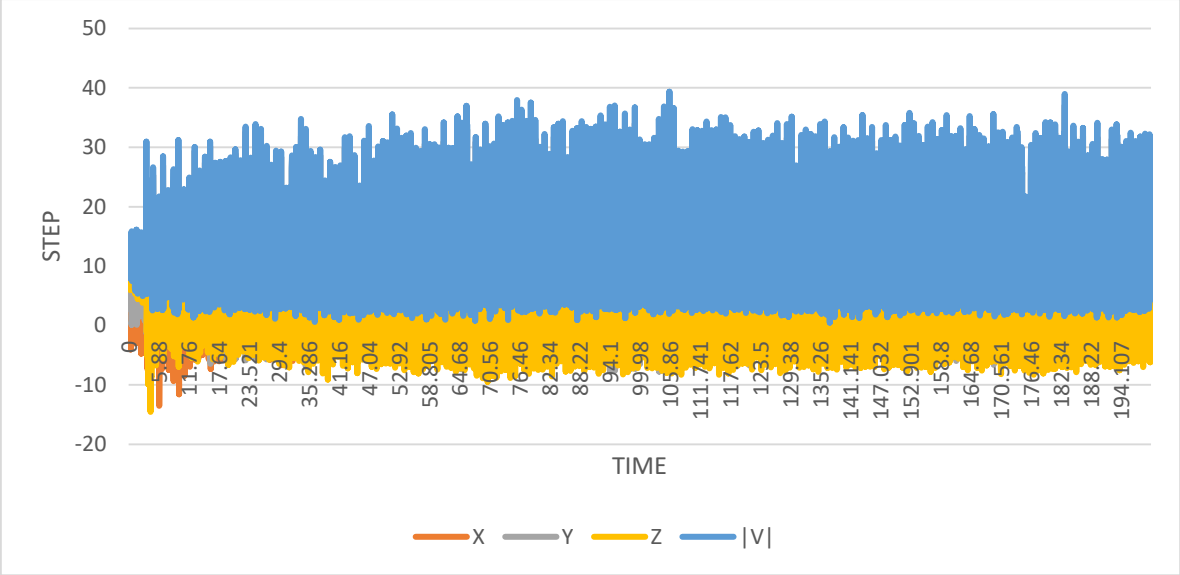
***Phone OS:*** MIUI

***Android Version:*** Android 7.0.0 (API 24) / Android Nougat

***Other Details:*** 4 GB RAM, 64 GB ROM, 2 GHz Octa-Core Qualcomm Snapdragon 625 Processor

Used to track the number of steps taken simultaneously.

The above is the number of steps taken




The Time vs Step chart created based on readings taken by G-Sensor Logger

***No. of Steps taken in the above experiment:*** 637

***No. of Steps shown by the Application:*** 548

***Percentage success:*** 86.02%

## ***Test 2:***

***Left Pocket of Trouser***

***Phone Model:*** Infinix Note 4

***Phone OS:*** XOS

***Android Version:*** Android 8.1.0 (API 27) / Android Oreo

***Other Details:*** 3 GB RAM, 32 GB ROM, 1.3 GHz Quad-Core Mediatek MT6753 Processor

Used to create the csv file to track the steps at definite time intervals.

***Right Pocket of Trouser***

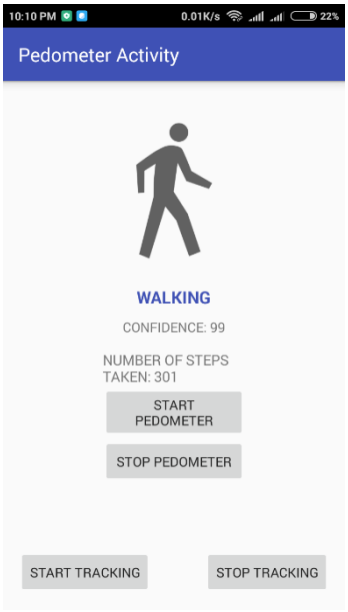
***Phone Model:*** Redmi 1S

***Phone OS:*** MIUI

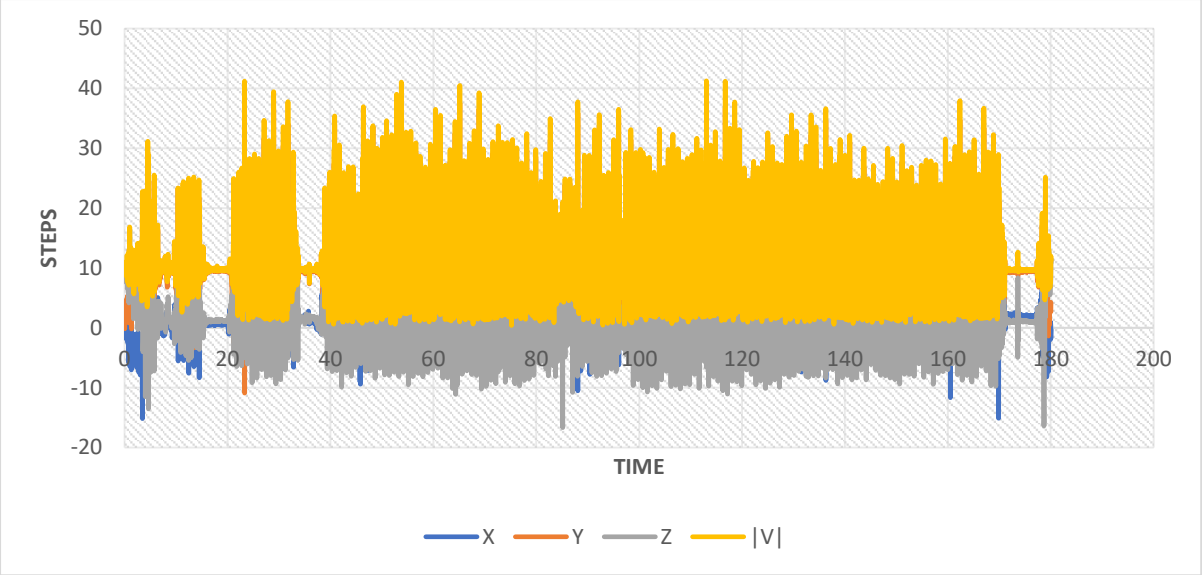
***Android Version:*** Android 4.4.4 (API 27) / Android KitKat

***Other Details:*** 1 GB RAM, 8 GB ROM, 1.6 GHz Quad-Core Qualcomm Snapdragon 400 Processor

Used to track the number of steps taken simultaneously.




The above is the number of steps taken




The Time vs Step chart created based on readings taken by G-Sensor Logger

***No. of Steps taken in the above experiment:*** 336

***No. of Steps shown by the Application:*** 301

***Percentage success:*** 89.58%

# CONCLUSION

Pedometers have been developed for quite some time. It has evolved for many years and has been improving by each passing day. There are still several problems in Pedometers. The main problem is that either the Activity is recognised by keeping the device connected to the network or the device is not able to give sufficient power backup if the code is run within the device itself. The problem is that in developing or under-developed countries, there is no presence of seamless Internet connectivity, which creates a problem of recognition of the activities.

## Future of Pedometers

A huge proportion of the world population is suffering from lifestyle related and preventable welfare diseases. Physical activity is probably the only means to achieve benefits from health impacts and reduce the risk of chronic disease. Pedometers are often used to measure physical activity. Nowadays, cell phones are very common, and pedometer applications enable measurements to be made by a device. So it is safe to say that Pedometers are the future of health fitness as it allows tracking of our everyday work very easily. Also, by each passing day, they are becoming much more accurate and reliable. So it might be not far that these devices are being used on a regular basis in every step of our daily life – to motivate, to increase fitness and to reduce chronic diseases.

# REFERENCES


- **[1]** https://en.wikipedia.org/wiki/Pedometer
- **[2]** https://www.explainthatstuff.com/how-pedometers-work.html
- **[3]** http://www.mobihealthnews.com/40514/study-smartphone-apps-count-steps-more-accurately-than-wearables
- **[4]** https://www.livescience.com/49756-smartphone-apps-wearables-step-counts.html
- **[5]** https://www.howtogeek.com/238904/how-to-track-your-steps-with-just-an-iphone-or-android-phone/
- **[6]** https://www.wareable.com/sport/how-to-use-google-fit-get-set-with-the-android-fitness-platform
- **[7]** https://en.wikipedia.org/wiki/Activity_recognition
- **[8]** https://developers.google.com/location-context/activity-recognition/
- **[9]** https://www.gc.cuny.edu/CUNY_GC/media/Computer-Science
- **[10]** http://web.cs.wpi.edu/~emmanuel/courses/cs528/F17/