Ion Driving App Development for Android Based Smartphones

Pooja Vishwanath[1] Soumen Roy[2]
[2]Technical Lead
[1]Department of Digital Communication Engineering
[1]RIT, Bengaluru [2]Effone Technologies Pvt Ltd, Bengaluru

Abstract— iON Driving is a standalone application that accurately collects and formats the information. This information can be used for analytics and predictive model generation of trending of risk for any insurance company. The application can be connected to a central administration web application to configure various critical parameters of the iON Driving App. Few critical parameters are Minimum speed, threshold for acceleration and breaking. The central data collection server is a repository of performance data transmitted by the iON Driving application. This app can be used as a "Pay as You Drive (PAYD) insurance product" and can be distributed through a web site branded with the insurers name.

Key words: Latitude, Longitude, GPS Accuracy, Speed, Accelerometer XYZ, Breaking and Acceleration flags, Timestamp, Voice Alert during over speed and hard breaking

I. INTRODUCTION

The "iON Driving" smart phone application is an Android based application that automatically collects driver performance information from an identified driver. The application monitors the driving capabilities and generates a trip file. The application monitors the trip and stores it locally on the device. The driver can inspect the report and see what errors/faults he has done while driving. The app uses GPS service to detect the motion. The motion detection routine detects if the vehicle is stationary and it stop collecting the trip data. iON Driving can be connected to a web based application used for reporting and trending of risk.

II. SCOPE OF THE PROJECT

The scope of this project is to develop a real-time application which will get connected to the server and they are:

- Application graphics (Speedometer).
- A loader program monitors the vehicle motion.
- Monitor speed.
- User can select unit of speed from the provided list (kph or mph).
- Provide option to Start and Stop of the trip
- Detects acceleration (over speed) and deceleration (breaking).
- Measures the distance travelled per trip.
- Reads time and date from phone and uses as time stamp in log file.
- Log trip details.
- Log lost satellite events.
- Offline speed graph.
- Detects paths between source and destination of the travelled path in the map.
- Display acceleration and breaking points in the map.
- Configurable audible feedback, On or Off in settings screen.
- Over Speed audible feedback – greater than over speed limit.
- Hard breaking audible feedback – greater than a reduction in speed per second.
- Works with the two most recent versions of Android OS.
- Log file header details Date, Time, Speed, "3D Accelerometer readings", Latitude, Longitude, GPS accuracy, Signal Strength, Acceleration and Breaking flags.

A. *Advanced Features*

- Calibration routine- Automatically calibrates sensor output when application is launched
- Accelerometer collector- Real-time calibrated data with a capability to measure up to 1000 readings/seconds.
- Gyroscope collector- If present, collects calibrated gyroscope readings at one second intervals
- ABC Calculator, calculates Acceleration, Braking and cornering
- Score Calculator Program- continuously calculates driver performance score on a scale of 1 to 100 with 100 being best
- Compression program- Compress the trip file before transmitting to the central collection server
- Upload program- Uploads the trip file to the central sever.
- Audible feedback for cornering and pit.
- Configuration management routine- Accepts application and sensor configuration/setting changes from the central sever and applies them to the iON app.

III. BLOCK DIAGRAM

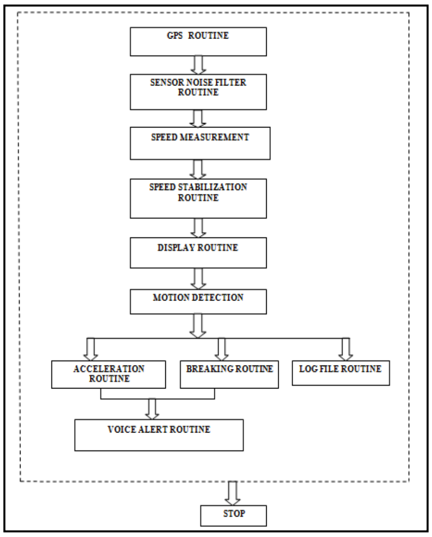


Fig. 1: Block diagram of the application

IV. BLOCK DESCRIPTION

A. *GPS Routine*

The Global Positioning System is a network of satellites that continuously transmits coded information, which makes it possible to precisely identify locations on earth by measuring distance from the satellites. A GPS tracking is a device that uses the Global Positioning System to determine the precise location of an object. The recorded location data can be stored within the tracking unit, or it may be transmitted to a central location data base. The major applications used by the GPS are navigation, location, mapping, search and rescue, tracking and timing synchronization.

Haversine formula is used to calculate distance between two GPS points coordinates:

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2) \quad \text{............(1.1)}$$

$$c = 2 \cdot \mathrm{atan2}(\sqrt{a}, \sqrt{(1-a)}) \quad \text{.........(1.2)}$$

$$d = R\,c \quad \text{.................................(1.3)}$$

Where, φ *is* latitude, λ *is* longitude, R is earth's radius (mean radius = 6,371km)

Alternatively, distance between 2 GPS coordinates can be expressed in degrees as:

d=ACOS(SIN(lat1*PI()/180)*SIN(lat2*PI()/180)+COS(lat1*PI()/180)*COS(lat2*PI()/180)*COS(lon2*PI()/180-lon1*PI()/180)) * 6371000 ....(1.4)

Thus these equations gives an estimate of average speed between 2 points and accuracy will depends on the distance and time elapsed between 2 GPS measurements.

B. *Sensor Noise Filter Routine*

A low-pass filter is a filter that passes low-frequency signals/values and attenuates (reduces the amplitude of) signals/values with frequencies higher than the cutoff frequency.

C. *Speed Measurement And Speed Stabilization Routine*

Using time and location data, a GPS unit is used to calculate the relative speed of the object, based on how much distance it covered in each time. Speeds are updated at short intervals to maintain accuracy always. It uses frequent calculations to determine the vehicle's speed.

D. *Display Routine*

The application consists of two buttons for start and stop monitoring. The distance measurements for the raw data, is displayed on the map where the different position estimates are displayed.

E. *Motion Detection*

GPS service is used to detect the motion. The motion detection routine detects if the vehicles are in stationary or not. If the vehicles are stationary, it stops collecting the trip data. If the vehicles are in motion, calculate acceleration routine and breaking routine by using voice alert output.

F. *Log File Routine*

Log file is a file that records all the events that occur in an operating system such as Date, Time, Speed, "3D Accelerometer readings", Latitude, Longitude, GPS accuracy, Signal Strength, Acceleration and Breaking flags are logged into the file.

V. HARDWARE COMPONENTS USED DETALS

A. *Accelerometer*

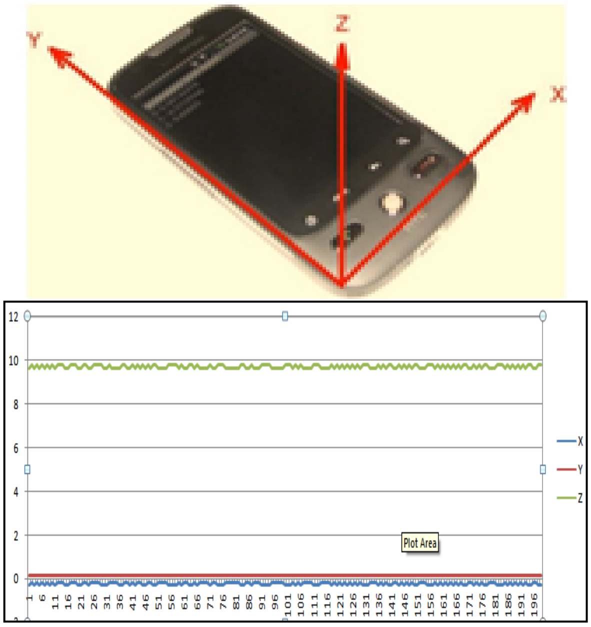


Fig. 2: The accelerometer coordinate system with readings in the graph

Many smart phones today use tiny sensors called accelerometers to provide enhanced user interface controls. Acceleration measure the linear acceleration in the x, y, z direction based on the movement of the phone.

- Acceleration X (horizontal)- The x-axis runs from the left to the right of the device when it is in the upright position. The acceleration is positive if the device moves towards the right.
- Acceleration Y (vertical)- The y-axis runs from the bottom to the top of the device when it is in the upright position. The acceleration is positive if the device moves up relative to the axis.
- Acceleration Z- The z-axis runs perpendicular to face of the device. The acceleration is positive if you move the device so that the face of the device points up. The acceleration is negative if the face of the device points towards the ground.
- Timestamp-The number of seconds at the time of the event since the runtime is initialized.

These sensors basically reduce the need of dedicated navigation and function keys on the mobile device. Accelerometer based mobile device use the principle for creating applications like games, controlling the orientation of the display screen, etc.

B. *GPS*

Global Positioning System (GPS) is a satellite based navigation system developed and operated by the United States Department of Defense for military purposes. GPS Satellites continually transmit messages that include a timestamp and its location, allowing the GPS receiver to compute its position. To level up for any mistakes, the receiver uses always four or more satellites to estimate the position using geometric trilateration, and line of sight is required at all time. This means that for outdoors navigation GPS systems can reach high levels of accuracy, however when line of sight (LOS) is not provided, for instance in indoors navigation, the system can't be used because LOS is blocked by walls and roofs. However, in outdoors navigation, the system provides precise information about latitude, longitude and altitude. At this moment, the satellite constellation includes 24 satellites, which are all available 95% of the time.

GPS uses technique of triangulation to find location. To triangulate a GPS receiver measures distance from the satellite using the travel time of radio signals. To measure the travel time, GPS needs very accurate timing, which is provided by atomic clocks used in the satellites. To compute positions in three dimensions we need to have four satellite measurements. The GPS uses trigonometric function to calculate the position. The position calculated by GPS receiver relies on three accurate measurements i.e. current time, position of the satellite and time delay for the signal.

VI. SOFTWARE TECHNOLOGIES AND TOOLS

Operating System	Android
Platform	Windows
Development Editor	Android Studio. 2.3.1
Storage	File System
Programming Language	Java
SDK Version	25

Table 2:

VII. IMPLEMENTATION AND TESTING OF THE PROJECT

A. *Software Modules Used Detailed Explaination Of The Code*

Software modules being used in this Android application can be described in the form of different activities. The activity is defined through a class file and an associated layout. Multiple activities have been used in this application to handle different functional requirements and they are as follows:

1) *LocationManager.java*

Location Manager is a class that provides access to location services. LM allows an app to tell Android when it is necessary to update the location information and when it is not needed. LM also provides information about the current state, such as GPS status information and enabled location providers. LM also gives details about the last known location in which the device has been.

Location Manager is usually the front door to the location services on an Android device, and the app needs to get reference to it at some point. This is done by using the Activity.getSystemService (LOCATION_SERVICE). Most commonly this is done in the onCreate() method because of the multiple calls that are going to be made to Location Manager throughout the Activity lifetime. Because the onCreate() method is the very first method that is going to be called in any activity's lifetime, it is required to acquire the location manager at the start. Location of the Android device can also be used to track the device everywhere it moves, it requires the use of GPS providers continuously. This also requires the collection and persistent location data to be sent to the app. It should continue collecting the data until the app is either turned off or the phone is turned off.

2) *Motion Manager*

Sensor Manager is part of the API that allows you to get sensor instance and access your device's sensors. The sensor data can be collected after the sensor has been registered. Sensor Event Listener is a listener that is used for receiving notifications from Sensor Manager when any sensor value has changed. Sensor Event object is related to Sensor Event Listener in such a way that Sensor Event object is going to be passed to callback methods in Sensor Event Listener. The listener will then process the object in an application-specific manner.

3) *Map Activity*

Map Activity uses the location services supported by the device through the classes in android. location package. The Maps package com.google.android.maps.MapView, a subclass of ViewGroup is used to display a map with data obtained from the Google Maps service. To use the Google maps data, Maps service had to be registered and MD5 fingerprint of the certificate which is required to sign the application had to be provided to obtain a Maps API key. This key had to be inserted in the .xml file of the Map View for the value of android: apiKey to enable the Google maps. To use the Maps external library, <uses-library> element in the Android Manifest file is defined to "com.google.android.maps". This thus enables the build tools to link the application against the Maps external library. In AndroidManifest.xml the key will be displayed in meta tags and we need to add permissions for accessing location of device. The required permission should be as follows:

- INTERNET – If we are connected to Internet or not.
- ACCESS_COARSE_LOCATION – To determine user's location using Wi-Fi and mobile. It will give us an approximate location.
- ACCESS_FINE_LOCATION – To determine user's location using GPS. It will give us precise location. This thus ensures that the Android system checks the required library to install the application on the device.

In the Map Activity class markers are placed to find the source and destination of the location and routes are drawn using Google Map Polyline [Polyline Options.add AllLat Lng (LatLngpoints)]. Coordinates of tapped points will be accessed and stored using MarkerPoints.get () and LatLng origin respectively. In this class, paths are displayed with acceleration and breaking points in the map. The flags are given for acceleration as '1',breaking as '2' and normal as '0'.

4) *Directions Json Parser*

This class is used to define the behavior of the entity which would handle the transaction with remote server using network connection. Data is formatted as JavaScript Object Notation (JSON) objects. In this application, it receives a JSON Object and returns a list of lists containing latitude and longitude.

5) *FileHandler.java*

File handler consists of Comma Separated value(CSV) file format which is used to exchange data between disparate applications. CSV files are mostly used for creating data files either for exporting the data or importing the data. It is mainly used to transport large amount of tabular data between applications that are not directly connected. The files are easily editable using common spreadsheet applications like Microsoft Excel.

B. User Interface

The User Interface is build using View and ViewGroup objects. Android UI is responsible for the Layout manager and the widget organizer. View objects are the basic unit of user interface on the Android platform. The View class serves as the base for subclasses called widgets, which offers fully implemented User Interface objects, like text fields and buttons. The ViewGroup class serves as the base for subclasses called layouts, which offers different kinds of layout architecture, like linear, tabular and relative. A View object is a data structure whose properties store the layout parameters and content for a specific rectangular area of the screen. The Android UI offers several layout managers that are used to implement an user interface. This application employs Linear Layout and Relative Layout. Linear Layout wraps the Text View, Table Row, Edit Text, List View and Buttons.

Relative Layout displays child views in relative positions and by default all child views are drawn at the top-left of the layout, so we must define position of each view by using RelativeLayout.LayoutParams. The value of each layout is either a Boolean to enable layout position relative to the parents or an ID that references another view in layout against the view positioned. It is a powerful utility for designing UI because it can eliminate nested view groups and keep layout hierarchy flat, which improves the performance.

C. Unified Modelling Language (Uml)

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was used for diagrams. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design.

There are many goals for developing UML but the most important is to define some general-purpose modeling language, which all modelers can use and it also needs to be made simple to understand and use.

D. Class Diagram

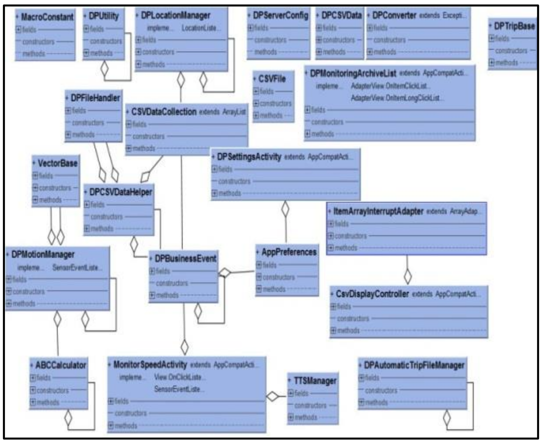


Fig. 3: Class Diagram

1) Use Case Diagram

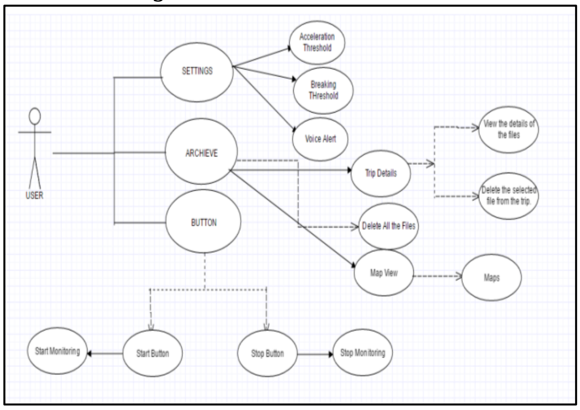


Fig. 3: Use Case Diagram

VIII. RESULTS AND DISCUSSIONS

Fig. 4: The main screen for the developed app which monitors the speed

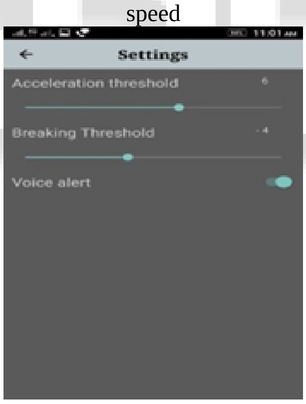


Fig. 5: Setting screen to adjust the threshold values from 1 to 10 for acceleration and breaking and by default, voice alert should be on

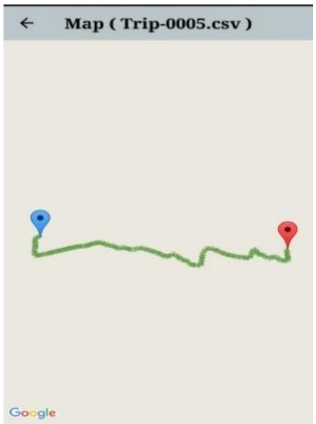


Fig. 6: Detects paths between Source(blue) destination(red) of

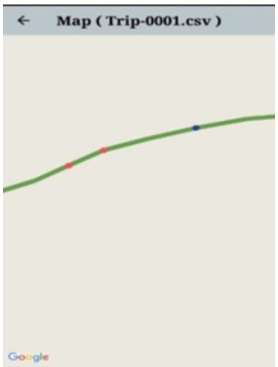
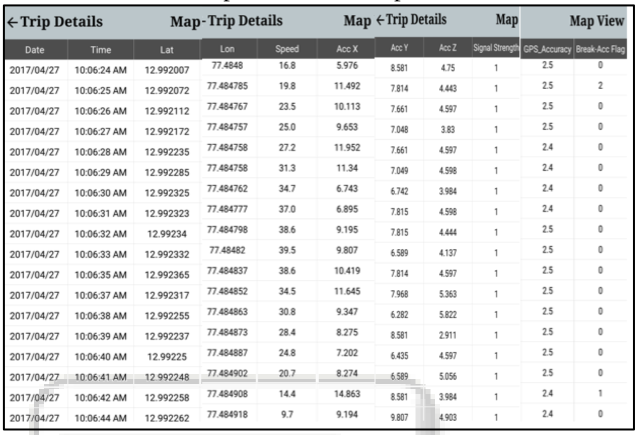


Fig. 7: Displays the acceleration (Red)and breaking(blue) points in the map.


←Trip Details Map-Trip Details Map ←Trip Details Map Map View

Date	Time	Lat	Lon	Speed	Acc X	Acc Y	Acc Z	Signal Strength	GPS_Accuracy	Break-Acc Flag
2017/04/27	10:06:24 AM	12.992007	77.4848	16.8	5.976	8.581	4.75	1	2.5	0
2017/04/27	10:06:25 AM	12.992072	77.484785	19.8	11.492	7.814	4.443	1	2.5	2
2017/04/27	10:06:26 AM	12.992112	77.484767	23.5	10.113	7.661	4.597	1	2.5	0
2017/04/27	10:06:27 AM	12.992172	77.484757	25.0	9.653	7.048	3.83	1	2.5	0
2017/04/27	10:06:28 AM	12.992235	77.484758	27.2	11.952	7.661	4.597	1	2.4	0
2017/04/27	10:06:29 AM	12.992285	77.484758	31.3	11.34	7.049	4.598	1	2.4	0
2017/04/27	10:06:30 AM	12.992325	77.484762	34.7	6.743	6.742	3.984	1	2.4	0
2017/04/27	10:06:31 AM	12.992323	77.484777	37.0	6.895	7.815	4.598	1	2.4	0
2017/04/27	10:06:32 AM	12.99234	77.484798	38.6	9.195	7.815	4.444	1	2.5	0
2017/04/27	10:06:33 AM	12.992332	77.48482	39.5	9.807	6.589	4.137	1	2.5	0
2017/04/27	10:06:35 AM	12.992365	77.484837	38.6	10.419	7.814	4.597	1	2.5	0
2017/04/27	10:06:37 AM	12.992317	77.484852	34.5	11.645	7.968	5.363	1	2.5	0
2017/04/27	10:06:38 AM	12.992255	77.484863	30.8	9.347	6.282	5.822	1	2.5	0
2017/04/27	10:06:39 AM	12.992237	77.484873	28.4	8.275	8.581	2.911	1	2.5	0
2017/04/27	10:06:40 AM	12.99225	77.484887	24.8	7.202	6.435	4.597	1	2.5	0
2017/04/27	10:06:41 AM	12.992248	77.484902	20.7	8.274	6.589	5.056	1	2.5	0
2017/04/27	10:06:42 AM	12.992258	77.484908	14.4	14.863	8.581	3.984	1	2.4	1
2017/04/27	10:06:44 AM	12.992262	77.484918	9.7	9.194	9.807	4.903	1	2.4	0

Fig. 8: When vehicle is in motion the trip file collects the details such as Date, Time, Speed, "3D Accelerometer readings", Latitude, Longitude, GPS accuracy, Signal Strength, Normal [0], Acceleration [1] and Breaking [2] flags.

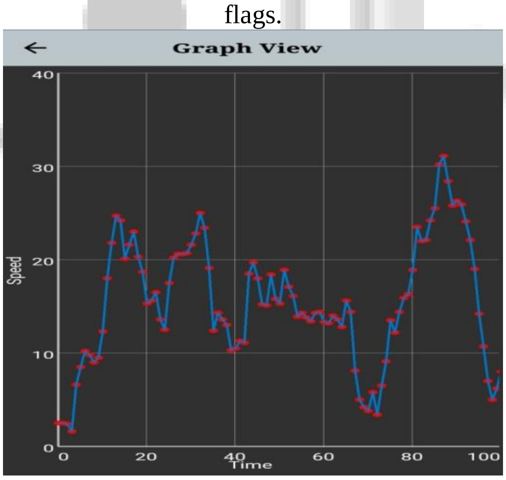


Fig. 8: Graph View of the speed points in the trip.

Classification of driving performance of driver A , B and C using GPS and accelerometer sensor.

PARAMETERS	DESCRIPTION	DRIVER A	DRIVER B	DRIVER C
ACCELERATION	Number of rapid acceleration events and harshness.	MEDIUM	HIGH	LOW
BREAKING	Number of hard breaking events in the trip.	HIGH	MEDIUM	MEDIUM
SPEED	Amount of speed relative to location dependent.	HIGH	HIGH	MEDIUM
ELAPSED DISTANCE	Distance of the trip	MEDIUM	MEDIUM	MEDIUM
ELAPSED TIME	Time duration of the trip.	MEDIUM	MEDIUM	MEDIUM

Table 3:

IX. CONCLUSIONS AND FUTURE WORK

From this paper, we can conclude that by using GPS and accelerometer sensor the android Smartphone detects motion, monitors the speed and distance travelled, provides audible alert for over speeding and hard breaking, detects paths between source and destination of the travelled path in the map and displays breaking and acceleration points in the map and monitors the driving performance of the identified driver. The development/extension of this project can be done by using calibration routine and uploading the trip file to the central sever can be done in further areas.

ACKNOWLEDGMENT

I wish to express my deep sense of gratitude to Mr. SOUMEN ROY, Technical Lead, EffOne Technologies Private Limited, Bangalore and express my humble gratitude for the valuable guidance, inspiration, encouragement and immense help which made this work a success.

I also take this opportunity to thank Mr. KRISHNA PRASAD S.J, Associate Professor, Department of TCE, MSRIT, Bangalore for his constant support, encouragement and have given his precious time in discussing the progress of the work.

REFERENCES

[1] Terra, Marco H.; Cerri, Joao P.; Ishihara, Joao Y. (2014). "Optimal Robust Linear Quadratic Regulator for Systems Subject to Uncertainties". IEEE Transactions Automatic Control. 59 (9): 2586–2591.

[2] Einicke, G.A. (December 2014). "Iterative Frequency-Weighted Filtering and Smoothing Procedures". IEEE Signal Processing. 21 (12): 1467–1470.

[3] Engelbrecht, J., Booysen, M.J. and van Rooyen, G.J.(2014). Recognition of Driving Manoeuvres using Smartphone-Based Inertial and GPS Measurement. Paper presented at Proceedings of the first International Conference on the use of Mobile Informations and Communication Technology (ICT) in Africa UMICTA. STIAS Conference Centre, Stellenbosch: Stellenbosch University, Department of Electrical & Electronic Engineering, South Africa. 9-10 December 2014.

[4] M. Mu~noz-Organero and V.C. Magana. Validating the impact on reducing fuel consumption by using an ecodriving assistant based on tra_c sign detection andoptimal deceleration patterns. Intelligent Transportation Systems, IEEE Transactions on,

14(2):1023{1028, 2013. ISSN1524-9050. doi: 10:1109/ TITS:2013:2247400
[5] Information is referred from the link: https://developer.android.com/reference/android/location/LocationManager.html.