

# Geofencing-based Auto-Silent Mode Application

M. S. I. M. Zin, M. F. M. Nurji, A. A. M. Isa, M. S. M. Isa  
Centre of Telecommunication Research and Innovation (CeTRI),  
Faculty of Electronic and Computer Engineering (FKEKK),  
Universiti Teknikal Malaysia Melaka (UTeM), Malaysia.  
shahril@utem.edu.my

**Abstract**—The usage of mobile devices such as smartphones and tablets has increased dramatically over the past years. Most of people carry at least one mobile device wherever they go. Mobile devices are becoming really important nowadays because they are usually the main tool for communications. However, sometimes the ringing sound of mobile devices can be a nuisance in certain circumstances such as during an important meeting or inside places that require silence such as library, cinema and prayer area. This problem occurs because most users forgot to switch their mobile device into silent mode. To address the problem, this paper presents a novel concept of automatically switching mobile devices into silent mode. This concept is developed based on the geofencing technique where a virtual fence will be created around a specific area. Whenever a mobile device crosses the virtual fence into the area, the device will be automatically switched into silent or vibrate mode. The device will be switched back to normal mode once it crosses the virtual fence to exit the area. This is done by utilizing the current location of the user based on the Global Positioning System (GPS) data provided through the device. The advantage of this application over other geofencing applications is that the geofence locations will be preloaded in the application, allowing applications with specific purpose and pre-determined locations to be developed.

**Index Terms**—Auto silent; Geofencing; GPS; Location Based Services (LBS).

## I. INTRODUCTION

Mobile device is usually defined as a small handheld computing device that has a main purpose of providing communication function over wireless networks. Two of the most popular mobile devices are smartphones and tablets. Mobile devices usage has increased substantially over the past years. The statistics shows that at the end of 2013, global smartphone penetration has reached 22% of world population where there are estimated 1.4 billion smartphones in use [1]. On average, there are 2 smartphones for every 9 people. These statistics are staggering, considering there are only 5% of global population using smartphone in 2009, an increase of 1.3 billion smartphones in four years' time. The usage of tablets is also on the rise, reaching 6% global penetration from when the devices first started to be widely available commercially back in 2011. Mobile devices are becoming a necessity and everyone is expected to carry at least one mobile device wherever they go. People mainly used mobile devices to communicate with each other, either through cellular voice telephony service or through usage of mobile data consumed by social media applications such as Facebook, Twitter, Instagram and WhatsApp. These services and applications needed notifications to attract user

attention. For telephone service, the devices will ring so that the user knows someone is trying to call them, whereas for social media applications, notifications are usually done through some short audible sound. However these ringing and notification sound can be a nuisance in some situations such as during important meetings, watching movies in cinema, reading books in library or during religious prayer time. People are usually advised to either turn off their mobile devices or turn the silent or vibrate mode on whenever they are about to enter any places that require silence such as meeting room, cinema, library and prayer area. Nevertheless, there are still a lot of people who forgot to mute their devices, resulting in awkward situations when the devices make an audible sound whenever it is supposed to be silenced. To address this problem, this paper is proposing a novel concept of 'auto-silent' mode based on geofencing technique as shown in Figure 1. Pre-determined locations can be fixed and stored in a database. A virtual fence or boundary can then be created around those areas. Whenever a mobile device which is equipped with an 'auto-silent' mode operation crosses the virtual fence into any area that have been stored in the database, the device will be automatically switched into silent or vibrate mode. This is done by utilizing the real-time location of the device, provided by the location data from the device's GPS chip. Although the proposed concept is applied on Android mobile operating system (OS) platform, it can be applied to any major OS platform for mobile devices such as iOS, Windows Phone and Blackberry [2].

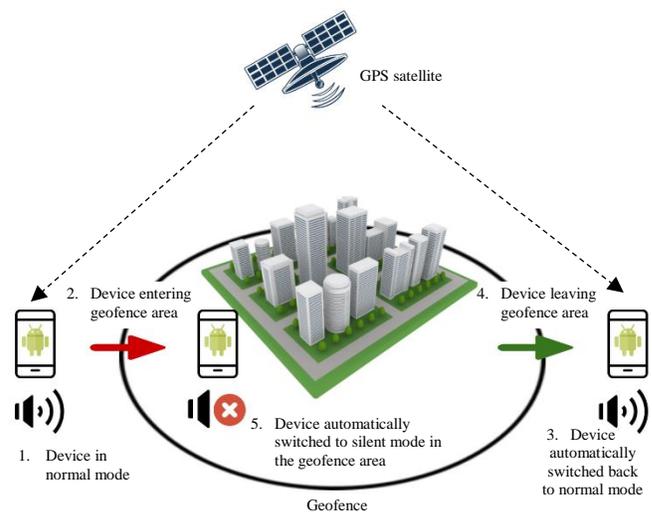


Figure 1: Overview of the concept

## II. BACKGROUND STUDY

### A. Global Positioning System (GPS)

The Global Positioning System (GPS) is a space-based navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth, where there is an unobstructed line of sight to four or more GPS satellites [3, 4, 5]. GPS was originally designed in the early 1970s for the United States (US) military by the US Department of Defense (DoD). Current GPS system offers critical capabilities to military, public and commercial users around the world. The system is maintained by the United States government and is freely accessible to anyone with a GPS receiver.

GPS consist of approximately 30 satellites orbiting in Medium Earth Orbit around the Earth. However it only takes 24 satellites to provide global coverage [6]. This is made possible through the precise radio signals that the satellites are transmitting which consist of the satellite's position and the time they transmitted the signals. The signals can be easily received by GPS receivers, allowing the calculation and determination of the accurate location, speed and time information. Using the information, GPS receivers are able to triangulate data and pinpoint user location.

### B. Geofence

Geofence is a tool or software program that uses the GPS to create a virtual perimeter or barrier around a real-world geographical area [7, 8, 9]. The procedure of using a geofence is called geofencing and it use the same concept as location-based service (LBS). LBS is a common class of computer program-level services that use location data to control features [10, 11, 12]. A geofence can be used to trigger an activity in any geo-enabled mobile devices.

Geofencing can be used to trigger automatic notifications when a device enters or exits the boundaries defined by the application. Many geofencing applications incorporate map services such as Google Maps and Apple Maps, allowing the applications to create boundaries on top of a satellite view of a specific geographical area. Boundaries can also be defined by using longitude and latitude.

Geofencing has a lot of potential usage. For example, parents can receive an email or text message that is automatically generated when their child leave from school and arrive at home. The geofence would be a virtual geographic barrier placed around the school and house. When the child's smartphone or GPS tracker exits or enters the geofences, a notification will be automatically sent to the parent by the geofence application on the device. Table 1 list out some potential geofence applications [13].

There are many advantages on using geofence because it can be very useful in many situations. With the ability to send an alert to the user, it can act as a reminder, advertising and many more. Unfortunately, no advance technology ever comes without limitations, so is the geofence technology. Table 2 shows several advantages and disadvantages of geofencing applications [14, 15].

Table 1  
Uses of geofence in various applications

Applications	Example
Fleet management	The fleet company receives a notification when a truck driver gone off from his route.
Human resource management	If an employee attempts to enter a prohibited area, the employee ID card trigger an alert to security.
Advertising	When the customer enters a geofence area, a notification with today's special will be triggered to opt-in customer by the shop.
Asset management	If the pallet is removed from the warehouse without permission, an RFID tag on the pallet can send an alert.
Law enforcement	If a person under house arrest leaves the premises, the ankle bracelet can warn the authorities.

Table 2  
Advantages and Disadvantages of Geo-fence

Advantages of Geofencing	Disadvantages of Geofencing
Engaging customers and potential buyers with the business.	Requires approval or participation from the user.
Better control and tracking on-road vehicle.	Battery usage is significantly higher because the use of GPS.
Adding value to sales process by contextually making offers and promotional freebies and gifts.	Only limited to fixed radius.
Great event management through real time location alert and data.	
Enhance security measures in any event or situation.	

### C. Geofence Algorithm

Geofencing uses the Haversine Formula to calculate the distance between two points of the earth's curvature [16]. The Haversine Formula is an equation in navigation which provides great-circle distance between two points on a sphere from its latitudes and longitudes.

The haversine of the central angle between any two points on a sphere is given by:

$$\text{hav}\left(\frac{d}{r}\right) = \text{hav}(\phi_2 - \phi_1) + \cos(\phi_1)\cos(\phi_2)\text{hav}(\lambda_2 - \lambda_1) \quad (1)$$

where *hav* is the haversine function:

$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2} \quad (2)$$

*d* is the distance between two points along the sphere

*r* is the radius of the sphere

$\phi_1, \phi_2$ : latitude of point 1 and point 2

$\lambda_1, \lambda_2$ : longitude of point 1 and point 2

## III. APPLICATION DEVELOPMENT

The geofencing-based auto-silent mode application for Android is developed using Android Studio Software Development Kit (SDK). Android Studio SDK is used to design the user interface (UI) and to code the main activity java codes. It also acts as an emulator to test the running project or application without installing the application on an actual device. Android Studio SDK is the official Integrated

Development Environment (IDE) launched by Google, for Android application development, based on IntelliJ IDEA’s powerful code editor and developers’ tools.

The overall flow chart of how the application works is shown in Figure 1. The application will be developed for Android 6.0 Marshmallow, which is the latest Android version at the moment of development.

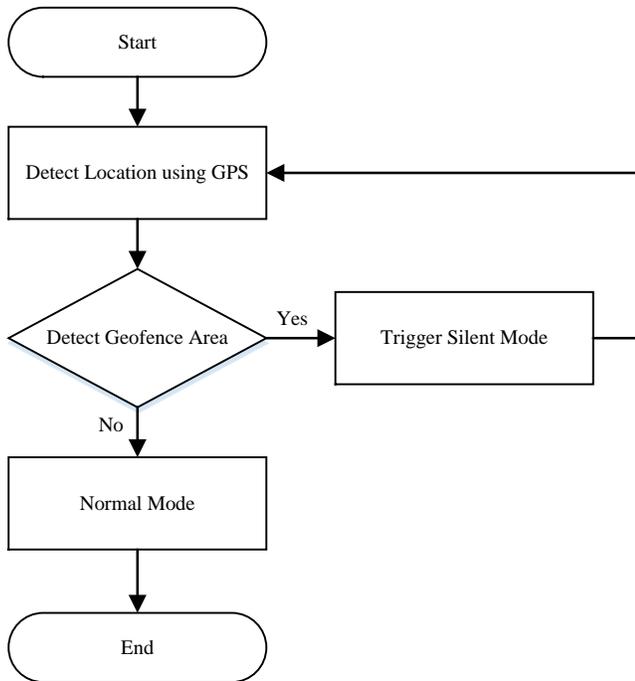


Figure 1: Flow chart of the application

The application uses *ACCESS\_FINE\_LOCATION* permission, as shown in Figure 2, to acquire the user’s device address level location, allowing the application to acquire precise location information of the device. It also uses *INTERNET* and *ACCESS\_NETWORK\_STATE* permission for Assisted GPS (A-GPS) ability. A-GPS significantly improves startup performance and time-to-first-fix (TTFF), of a GPS, thus allowing faster and more accurate location determination.

```

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />

<!--
  Requests address-level location access, which is usually
  necessary for Geofencing
-->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
  
```

Figure 2: Coding for GPS and A-GPS permissions

Figure 3 shows the Google Maps application program interface (API) key for the application to access and uses the Google Maps service. The API key is provided by Google to allow communication with Google Services and their integration to other services. This is essential for the application because it uses the Google Map’s GPS and services.

```

<!-- Required for Google Maps -->
<meta-data
  android:name="com.google.android.gms.version"
  android:value="@integer/google_play_services_version" />
<meta-data
  android:name="com.google.android.maps.v2.API_KEY"
  android:value="AIzaSyD4JICOMszc9JcEbgW3aIx1LBGb91KyHp6Q" />
</application>
</manifest>
  
```

Figure 3: Google Maps API key

The UI layout is coded in the *activity\_main.xml* file, as shown in Figure 4. It will represent the main page of the application. The UI layout consists of screen fragment that will use Google Map to display the device’s current location as shown in Figure 5.

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:orientation="vertical"
  tools:context="com.faaiznurji.silentfence.MainActivity" >
  <fragment
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="Odp"
    android:layout_weight="45"
    class="com.google.android.gms.maps.SupportMapFragment" />
</LinearLayout>
  
```

Figure 4: User Interface Layout code



Figure 5: Google Map displayed with the geofences area

*MainActivity.java* is one of the crucial files inside the project’s java section because it’s where the geofence’s coordinates (latitude and longitude) and geofence’s radius are inserted in the source code. Figure 6 shows the geofence’s coordinates and radius in geofence array list. For the application prototype, several locations around Universiti Teknikal Malaysia Melaka (UTeM) main campus have been selected such as Faculty of Electronic and Computer Engineering (FKEKK) main building, mosque, main library and main hall. The coordinates of these locations will be added to the geofence coordinates array. The geofence’s radius for each of the

location will be determined based on the size of the locations. These radius information, in meters, will be added to the geofence radius array as shown in Figure 6. Based on the coordinates and radius information, the geofences will be created and shown on the map as a red circle, covering the area enclosed by the geofences, as can be seen in Figure 5.

```
// Adding geofence coordinates to array.
mGeofenceCoordinates.add(new LatLng(2.236270, 102.303920));
mGeofenceCoordinates.add(new LatLng(2.313292, 102.318190));
mGeofenceCoordinates.add(new LatLng(2.311952, 102.318614));
mGeofenceCoordinates.add(new LatLng(2.309144, 102.320456));
mGeofenceCoordinates.add(new LatLng(2.314260, 102.317890));
mGeofenceCoordinates.add(new LatLng(2.311328, 102.322253));
mGeofenceCoordinates.add(new LatLng(2.316785, 102.320630));
mGeofenceCoordinates.add(new LatLng(2.314653, 102.318678));

// Adding associated geofence radius' to array.
mGeofenceRadius.add(25);
mGeofenceRadius.add(50);
mGeofenceRadius.add(50);
mGeofenceRadius.add(50);
mGeofenceRadius.add(30);
mGeofenceRadius.add(70);
mGeofenceRadius.add(40);
mGeofenceRadius.add(40);
```

Figure 6: Geofence Coordinates and Radius source code

Figure 7 shows the coding to construct part of a geofence on one location, which is the FKEKK main building. Geofence Builder sets an ID for the geofence, and obtains the coordinates of the building and geofence radius from the geofence array list to build FKEKK's geofence on Google Map. In order to execute the geofence function, *setTransitionTypes* code was written for specific functions to link with *GeofenceIntentService.java* file. It will set the type of transitions when entering and exiting the geofence.

The application ability to automatically switch the device into silent or vibrate mode when entering a geofence and switch back to normal mode when exiting a geofence is coded in *GeofenceIntentService.java* file, as shown in Figure 8. It will also send notification to smartphone's notification bar when a user's location interacts with the geofence transitions.

Ringer mode in Android devices can be controlled using *AudioManager* class, which allow application to control ringer volume and ringer profile. The application uses only two types of ringer mode:

- *RINGER\_MODE\_NORMAL* for exiting geofence
- *RINGER\_MODE\_VIBRATE* for entering geofence

```
// Building the geofences and adding them to the geofence array.

// FKEKK
mGeofences.add(new Geofence.Builder()
    .setRequestId("FKEKK")
    // The coordinates of the center of the geofence and the radius in meters.
    .setCircularRegion(mGeofenceCoordinates.get(1).latitude, mGeofenceCoordinates.get(1).longitude, mGeofenceRadius.get(1).intValue())
    .setExpirationDuration(Geofence.NEVER_EXPIRE)
    // Required when we use the transition type of GEOFENCE_TRANSITION_DWELL
    .setLoiteringDelay(30000)
    .setTransitionTypes(
        Geofence.GEOFENCE_TRANSITION_ENTER
        | Geofence.GEOFENCE_TRANSITION_EXIT).build());
```

Figure 7: Building geofence source code

```
@Override
protected void onHandleIntent(Intent intent) {
    myAudioManager = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
    GeofencingEvent geofencingEvent = GeofencingEvent.fromIntent(intent);
    Log.v(TAG, "onHandleIntent");
    if (!geofencingEvent.hasError()) {
        int transition = geofencingEvent.getGeofenceTransition();
        String notificationTitle;

        switch(transition) {
            case Geofence.GEOFENCE_TRANSITION_ENTER:
                notificationTitle = "Silent Mode Activate";
                Log.v(TAG, "Geofence Entered:Your phone is now in Silent Mode");
                myAudioManager.setRingerMode(AudioManager.RINGER_MODE_VIBRATE);
                Toast.makeText(getApplicationContext(), "You're entering a Silentfence",
                    Toast.LENGTH_LONG).show();
                break;
            case Geofence.GEOFENCE_TRANSITION_DWELL:
                notificationTitle = "Silent Mode Activate";
                Log.v(TAG, "Dwelling in Geofence");
                myAudioManager.setRingerMode(AudioManager.RINGER_MODE_VIBRATE);
                Toast.makeText(getApplicationContext(), "You're in a Silentfence.",
                    Toast.LENGTH_LONG).show();
                break;
            case Geofence.GEOFENCE_TRANSITION_EXIT:
                notificationTitle = "Normal Mode Activate";
                Log.v(TAG, "Geofence Exited");
                myAudioManager.setRingerMode(AudioManager.RINGER_MODE_NORMAL);
                Toast.makeText(getApplicationContext(), "You're exiting a Silentfence.",
                    Toast.LENGTH_LONG).show();
                break;
            default:
                break;
        }
    }
}
```

Figure 8: Geofence Transitions source code

There are 3 types of geofence transitions which are:

- *GEOFENCE\_TRANSITION\_ENTER*
- *GEOFENCE\_TRANSITION\_DWELL*
- *GEOFENCE\_TRANSITION\_EXIT*

Each transitions has its own function. First, *GEOFENCE\_TRANSITION\_ENTER* will turn the smartphone into silent/vibrate mode when user enters the geofence and sends a notification to alert the user. The second transition, *GEOFENCE\_TRANSITION\_DWELL*, will keep the smartphone in silent/vibrate mode. This transition will prevent the smartphone from switching back to normal mode as long as the device remains in the geofence area. The last transition, *GEOFENCE\_TRANSITION\_EXIT*, will switch back from silent/vibrate mode to normal mode when the device is exiting the geofence area and sends a notification to alert the user.

After development of the application has been complete, it needs to be tested on real hardware. The application will be sideloaded to an Android device by using Android Studio. The application will then be evaluated on an actual device to test the stability of the application and the geofence transitions. The virtual device emulator in Android Studio cannot evaluate the geofence transition because it does not have a GPS transceiver.

#### IV. APPLICATION TESTING

To test the application, a Samsung Galaxy S4 smartphone, running on the latest Android 6.0 Marshmallow is used. All the results are from the device screenshots during the application's testing process.

##### A. Home Screen

The application's icon will appear in the home screen after it has been successfully installed on the device, as shown in Figure 9. Before the application is launched, the Location or GPS function must be switched on to obtain the user's current location. The device also need to be in normal ringing mode. The WiFi or mobile data can be used to get a more accurate GPS result.

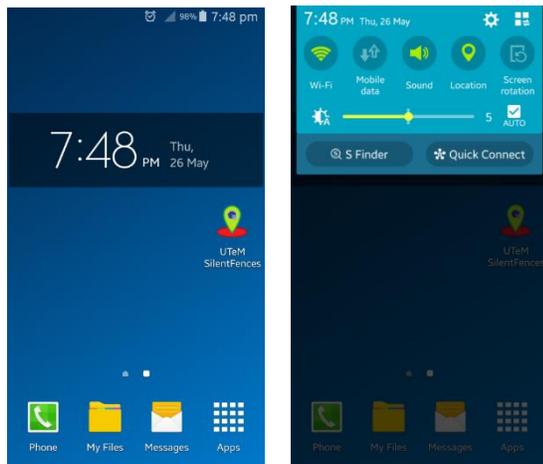


Figure 9: Home Screen of Samsung Galaxy S4 with the application icon

##### B. Main Page

After the application is launched, a Google Maps screen will display the device's current location, as shown in Figure 10. It will also display the geofences areas based on the geolocation information in the *MainActivity.java*. The geofence areas are marked with red circles in the map.



Figure 10: The application main page showing the geofence areas

##### C. Geofence Enter Result

Figure 11 shows the result when the device enters the

geofence area. The blue dots represents the user's current location in Google Map, based on GPS. The device will automatically switched to silent/vibrate mode because the application has detected the device entering a geofence area, depicted by the red circle. A notification will be sent to the device's notification bar, informing that the silent/vibrate mode has been activated. The device will vibrate if there are any incoming call or text.

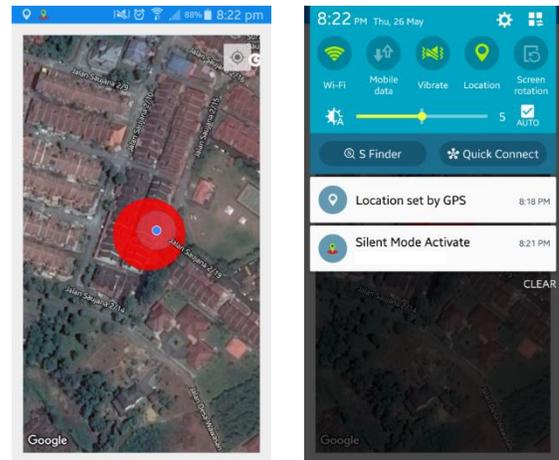


Figure 11: Device enters geofence and silent/vibrate mode is activated

##### D. Geofence Exit Result

Figure 12 shows that when the device exits the geofence area, the application will automatically switched the device back into normal mode. The device's notification bar will display a notification showing that the device's normal mode has been activated.

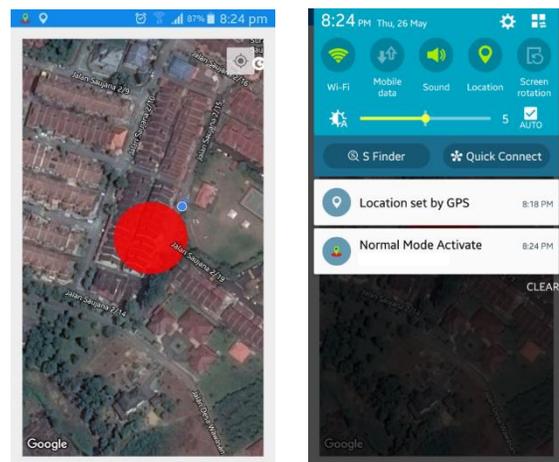


Figure 12: Device exits geofence and normal mode is activated

#### V. CONCLUSION

In this paper, the development of geofencing-based auto-silent mode application has been presented. The application is developed for Android mobile platform using Android Studio SDK, but the concept can be easily applied to other mobile platform such as iOS, Windows Phone or Blackberry. This

application is developed based on the geofencing technique where virtual fences will be created around specific areas. Whenever the mobile device crosses the virtual fence into the area, the application will automatically mute or switch the device into silent/vibrate mode while also sending a notification to the device's notification bar to inform the user. When the device leaves the area and crosses the geofence, the application will switch the device back to normal mode. The accuracy of the location information can be improved by using Assisted GPS (A-GPS) or other positioning techniques [17, 18]. The main advantage of this application over other geofencing applications is the geofence locations will be preloaded in the application. This concept allows applications with specific purpose to be developed. For example, an application with the location of all the mosque in the world can be developed, and whenever users enter any mosque, their devices will be automatically turned into silent/vibrate mode. This will solve the problem of ringing phone disturbing the concentration of congregation performing prayer in the mosque. Separate application for other places that require silence such as cinema, library and lecture halls can also be easily developed.

#### ACKNOWLEDGMENT

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) and Ministry of Higher Education (MOHE), Malaysia for funding this work under the Research Acculturation Collaborative Effort (RACE) research grant RACE/F3/TK8/ FKEKK/F00251.

#### REFERENCES

- [1] Smartphone and Tablet Penetration: <http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10?IR=T&>.
- [2] Zin, M. S. I. M., M. A. Sazali, A. A. M. Isa, and M. S. M. Isa. "Development of Auto-Notification Application for Mobile Device using Geofencing Technique." *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, Vol. 7, No. 2 (2015): 169-173.
- [3] National Research Council (U.S.). Committee on the Future of the Global Positioning System; National Academy of Public Administration (1995).

- The global positioning system: a shared national asset: recommendations for technical improvements and enhancements. National Academies Press. p. 16. ISBN 0-309-05283-1.
- [4] Hofmann-Wellenhof, Bernhard, Herbert Lichtenegger, and James Collins. *Global positioning system: theory and practice*. Springer Science & Business Media, 2012.
  - [5] Enge, Per, and Pratap Misra. "Special issue on global positioning system." *Proceedings of the IEEE* 87, no. 1 (1999): 3-15.
  - [6] El-Rabbany, Ahmed. *Introduction to GPS: the global positioning system*. Artech House, 2002.
  - [7] Namiot, Dmitry, and Manfred Sneys-Snepp. "Geofence and network proximity." In *Internet of Things, Smart Spaces, and Next Generation Networking*, pp. 117-127. Springer Berlin Heidelberg, 2013.
  - [8] Munson, Jonathan P., and Vineet K. Gupta. "Location-based notification as a general-purpose service." In *Proceedings of the 2nd international workshop on Mobile commerce*, pp. 40-44. ACM, 2002.
  - [9] Namiot, Dmitry. "GeoFence services." *International Journal of Open Information Technologies* 1, no. 9 (2013).
  - [10] Hahn, James F. "Indoor Positioning Services & Location Based Recommendations." (2016).
  - [11] Yin, Hongying. "Location Based Service." In *T-109.551 Research Seminar on Location Business II*. 2002.
  - [12] Tilson, David, Kalle Lyytinen, and Ryan Baxter. "A framework for selecting a location based service (LBS) strategy and service portfolio." In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pp. 10-pp. IEEE, 2004.
  - [13] M. Rouse, "What is geo-fencing (geofencing)? - Definition from WhatIs.com", *WhatIs.com*, 2015. [Online]. Available: <http://whatis.techtarget.com/definition/geofencing>.
  - [14] "Advantages and Disadvantages of Geofencing Applications." Medium. 2015. <https://medium.com/@indianapp/advantages-and-disadvantages-of-geofencing-applications-a20e47bd2cc4#ybs4pvpx>.
  - [15] Haddock, William Jason, and Phil Lunsford. "Geo-fencing Technologies and Security."
  - [16] Statler, Stephen. "Geofencing: Everything You Need to Know." In *Beacon Technologies*, pp. 307-316. Apress, 2016.
  - [17] A. A. M. Isa, M. H. Othman, M. S. M. Isa, N. Z. Haron, Z. Zakaria and M. S. I. M. Zin, "Utilising MIMO for positioning in mobile WiMAX systems," *2013 IEEE Symposium on Wireless Technology & Applications (ISWTA)*, Kuching, 2013, pp. 7-10.
  - [18] Awang Md Isa, Azmi and Markarian, Garik and Isa, Mohamad and Haron, N. Z. and M. S. I. M. Zin, "Hybrid TOA-Based MIMO and DOA-Based Beamforming for Location and Positioning in WiMAX Networks." *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, Vol. 4, No 2 (2012): 11-20.