Development of Auto-Notification Application for Mobile Device using Geofencing Technique

M. S. I. M. Zin, M. A. Sazali, 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)
shahril@utem.edu.my

***Abstract*—Mobile devices have become an integral part of human lifestyle. The convenience offered by mobile devices usage can't be denied. However, there are growing concerns about the disturbances caused by usage of mobile devices. One of the concerns is the disturbances caused by ringing sound of mobile devices at places that require silence such as mosque, library, cinema, and meeting room. Mobile device owners often forget to turn on silence mode on their devices in these places. To overcome such problem, this paper proposes an auto-notification application for mobile device using geofencing technique. Geofencing technique is used to detect virtual boundary that have been preset around required places. When a user enters the particular place, the application will detect that the user's mobile device has crossed the virtual boundary and will automatically notify the user to turn the silence mode on the device. The opposite process will happen when the user crosses the boundary to exit the place. The main advantage of the application is that it will remind mobile device users to mute their devices when they are entering places that require silence**

***Index Terms*—auto-notification, geofencing, mobile device, windows phone.**

I. Introduction

Mobile devices such as smartphones and tablets are rapidly becoming the most prevalent and widespread form of communication in our society. The usage of mobile devices 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. According to the International Telecommunications Union (ITU), a United Nations agency that conducts statistical data on Information Communications Technology (ICT), mobile device usage has grown at a remarkably fast rate in the past 10 years [1]. This change in the way human communicate comes as a direct result from the lower prices and conveniences that come with using mobile devices as opposed to landline options. However, with the dramatic rise of mobile devices usage, there are growing concerns about the disturbances caused by usage of mobile devices. One of the concerns is the disturbances caused by ringing of mobile devices at places that require silence such as mosque, library, cinema, and meeting room. This problem arises mainly because mobile device owners often forget to turn on silence mode on their devices in these places.

To overcome the problem, an auto-notification application for mobile device using geofencing technique has been developed on Windows Phone platform. This concept is developed based on the geofencing technique where a virtual boundary or a geofence will be created around a specific area. When a user enters that particular place, the application will detect that the user's mobile device has crossed the geofence and will notify the user to turn the silence mode on the device. This is done by utilizing the current location of the user based on the Global Positioning System (GPS) data provided through the device. The opposite process will happen when the user crosses the boundary to exit the place. Figure 1 summarizes the concept of the application. The main advantage of the application is that it will remind mobile device users to mute their devices when they are entering places that require silence.

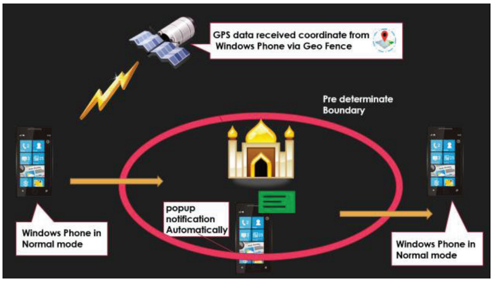

GPS data received coordinate from Windows Phone via Geo Fence
Pre determinate Boundary
Windows Phone in Normal mode
popup notification Automatically
Windows Phone in Normal mode

Figure 1: Concept of auto-notification application on Windows Phone device using geofencing technique

II. Background Study

A. *Global Positioning System (GPS)*

The Global Positioning System (GPS) is a satellite-based navigation system that was originally designed in the early 1970s for the United States (US) military by the US Department of Defense (DoD) [2]. 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 provides location and time information anywhere on

Earth where there is an unobstructed line of sight to at least four GPS satellites among a constellation of 31 satellites currently orbiting the Earth. 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. Geofencing

Geofencing is a feature in a software program that utilizes GPS or radio frequency identification (RFID) to define geographical boundaries [3]. A geofence is a virtual barrier.

Applications or programs that feature geofencing allow notifications or triggers to be set up so when a device enters or exits the boundaries defined by the administrator, a series of event can be executed such as a notification being delivered to user, a text message or email alert is sent, or a trigger to perform specific functions on the mobile device. Many geofencing applications incorporate map services such as Google Maps and Apple Maps, allowing administrators to create boundaries on top of a satellite view of a specific geographical area. Boundaries can also be defined by using longitude and latitude.

The technology has many practical uses. For example, a network administrator can set up alerts so when a organization-owned mobile device such as tablet or laptop leaves the organization grounds, the administrator can disable and wipe out confidential materials from the device. Geofencing can also be used in advertising where a marketer can geofence a retail store in a shopping mall and automatically send a mobile coupon to customers who has downloaded a particular mobile application when the customer device crosses the boundary. Table 1 lists other potential uses of geofencing technique.

Table 1
Potential uses of geofencing technique [3]

Geofencing Application	Example
Law enforcement	An ankle bracelet can alert authorities if an individual under house arrest leaves the premises.
Asset management	An RFID tag on a pallet can send an alert if the pallet is removed from the warehouse without authorization.
Marketing	A restaurant can trigger a text message with the day's specials to an opt-in customer when the customer enters a defined geographical area.
Human resource management	An employee smart card will trigger an alert to security if an employee attempts to enter area without authorization.
Fleet management	An alert will be sent to the dispatcher if a truck driver deviates from the planned route

C. Windows Phone Operating System

The mobile device industry is highly competitive and after stiff competition from other mobile operating system (OS) such as Android from Google and iOS from Apple, Microsoft has reenter the mobile OS platform market with a strong emphasis on software, hardware and services [4]. WP is a family of mobile OS developed by Microsoft for mobile devices such as smartphones and tablets as the replacement successor to Windows Mobile (WM) [5]. Unlike WM, WP is primarily aimed at the consumer market rather than the enterprise market [6] and it was first launched in October 2010 as Windows Phone 7 (WP 7) [7]. Windows Phone 8.1 (WP 8.1) was the last public release of WP, released to manufacturing on April 14, 2014 [8]. Windows 10 Mobile succeeded WP 8.1 in November 2015 where it accentuates a larger amount of integration and unification with its personal computer (PC) counterpart, including a new, unified application ecosystem, along with an expansion of its scope to include small-screened tablets [9].

WP 8.1 has been chosen as the mobile OS platform to develop the automatic notification application using geofencing technique. WP 8.1 provides developers with three powerful user interface (UI) technologies for developing applications. The UI technologies are XAML, XNA and Direct3D, and these technologies allows the creation of complex and compelling application design.

The geofencing application program interfaces (APIs) in WP allow applications to offer geographically contextual experiences in a timely manner without the need for the application to be continuously running and consuming device resources [10]. Geofencing in WP enables scenarios like popping up a reminder for a user when they are leaving a location or displaying advertisements and coupons when the user gets within range of a store.

Geofencing applications in WP define areas of interest or geofences and corresponding trigger conditions, such as triggering when the user enters or exits the geofence. The application registers to be notified when the trigger conditions are met. WP supports hundreds of geofences per application. The OS efficiently manages geofence tracking for all applications. When the trigger conditions for one or more geofences are met, the associated application is alerted even when it is not actively running. This high capacity, power-aware, adaptive geofence tracking system allows application developer to quickly and easily add location-triggered functionality to the application.

The geofencing APIs in WP enable applications to do the following [10]:

- Create one or more geofences, or areas of interest.
- Request to be notified when the device enters a geofence.
- Request to be notified when the device leaves a geofence.
- Specify a time window during which the geofence is active.
- Specify a dwell time for each geofence. Dwell time is the total time the device should be in or out of the geofence before the notification is triggered.
- Dynamically add and remove a geofences from applications collection of monitored geofences.
- Receive geofence events while the app is active.
- Register with the system to have a background task launch when the state of one of application's geofences changes.
- Use simulator to simulate device movement and test your app's geofencing features.

III. Application Development

The development of the auto-notification application on WP platform is done in several phases. Figure 2 summarizes the application development process.

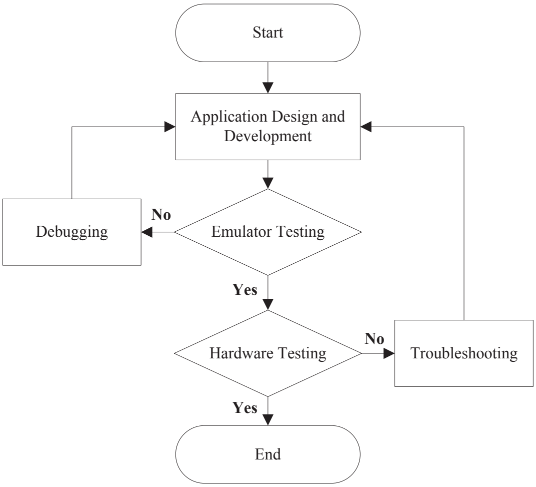


Figure 2: The process in the application development

The WP software development kit (SDK) 8.0 is used in the software design and development phase. It includes all the tools required to develop WP applications and can be downloaded from WP Dev Center site. WP SDK 8.0 requires Windows 8 64-bit (x64) and contains three main components; Visual Studio, WP Emulator and Microsoft Expression Blend for WP. Figure 3 shows the integrated development environment (IDE) interface of Visual Studio, where the main programming activities takes place. The IDE contains several panes such as toolboxes, design view, XAML view, properties window and solution explorer. The coding for the application is done in the XAML view and displayed visually in the design view.

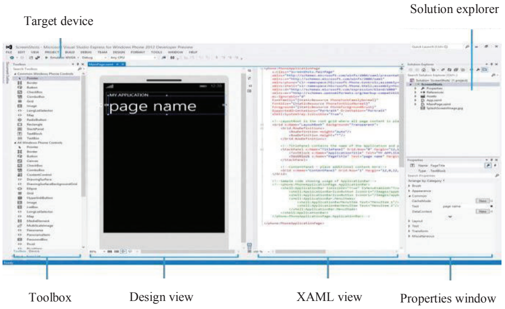
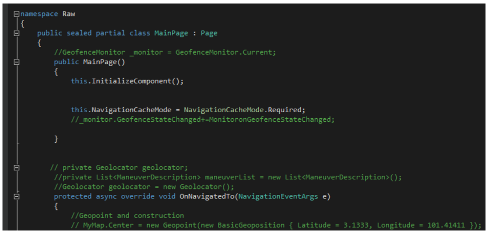


Figure 3: The IDE interface of Microsoft Visual Studio

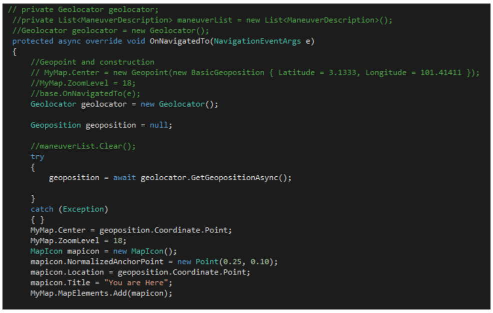
Figure 4 shows the initial part of the coding. This part initializes the geofencing component of the application and set up the necessary tool for the application to detect the geofences.



```
namespace Raw
{
    public sealed partial class MainPage : Page
    {
        //GeofenceMonitor _monitor = GeofenceMonitor.Current;
        public MainPage()
        {
            this.InitializeComponent();


            this.NavigationCacheMode = NavigationCacheMode.Required;
            //_monitor.GeofenceStateChanged+=MonitoronGeofenceStateChanged;

        }


       // private Geolocator geolocator;
        //private List<ManeuverDescription> maneuverList = new List<ManeuverDescription>();
        //Geolocator geolocator = new Geolocator();
        protected async override void OnNavigatedTo(NavigationEventArgs e)
        {
            //Geopoint and construction
            // MyMap.Center = new Geopoint(new BasicGeoposition { Latitude = 3.1333, Longitude = 101.41411 });
```

Figure 4: Application initialization

Figure 5 shows part of the coding to detect current location of the device. The current location will be displayed on the map and labelled as "You are here", as shown in Figure 6.



```
// private Geolocator geolocator;
 //private List<ManeuverDescription> maneuverList = new List<ManeuverDescription>();
 //Geolocator geolocator = new Geolocator();
 protected async override void OnNavigatedTo(NavigationEventArgs e)
 {
     //Geopoint and construction
     // MyMap.Center = new Geopoint(new BasicGeoposition { Latitude = 3.1333, Longitude = 101.41411 });
     //MyMap.ZoomLevel = 18;
     //base.OnNavigatedTo(e);
     Geolocator geolocator = new Geolocator();

     Geoposition geoposition = null;

     //maneuverList.Clear();
     try
     {
         geoposition = await geolocator.GetGeopositionAsync();

     }
     catch (Exception)
     { }
     MyMap.Center = geoposition.Coordinate.Point;
     MyMap.ZoomLevel = 18;
     MapIcon mapicon = new MapIcon();
     mapicon.NormalizedAnchorPoint = new Point(0.25, 0.10);
     mapicon.Location = geoposition.Coordinate.Point;
     mapicon.Title = "You are Here";
     MyMap.MapElements.Add(mapicon);
```

Figure 5: Getting current location

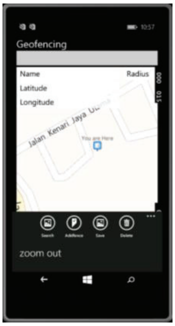
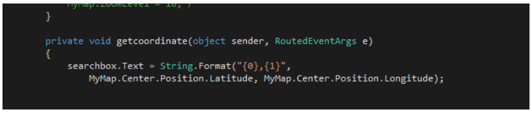


Figure 6: Current location on map

Figure 7 shows part of the coding to acquire the current location's latitude and longitude. These data will be updated in real-time to reflect the movement of the device, as shown in Figure 8.



```
        MyMap.ZoomLevel = 18; */
    }

    private void getcoordinate(object sender, RoutedEventArgs e)
    {
        searchbox.Text = String.Format("{0},{1}",
            MyMap.Center.Position.Latitude, MyMap.Center.Position.Longitude);
```

Figure 7: Acquire current location coordinate

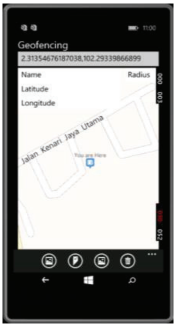
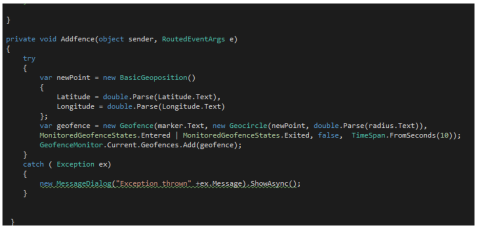


Figure 8: Displaying latitude and longitude of current location

Figure 9 shows part of the coding to manually define and register location. Specific name or GeoID can be assigned to the location, together with its latitude and longitude as shown in Figure 10. User can also set the required radius around the location, to create the virtual boundary or geofence. These data will be stored in the database.



```
}

private void Addfence(object sender, RoutedEventArgs e)
{
    try
    {
        var newPoint = new BasicGeoposition()
        {
            Latitude = double.Parse(Latitude.Text),
            Longitude = double.Parse(Longitude.Text)
        };
        var geofence = new Geofence(marker.Text, new Geocircle(newPoint, double.Parse(radius.Text)),
        MonitoredGeofenceStates.Entered | MonitoredGeofenceStates.Exited, false,  TimeSpan.FromSeconds(10));
        GeofenceMonitor.Current.Geofences.Add(geofence);
    }
    catch ( Exception ex)
    {
        new MessageDialog("Exception thrown" +ex.Message).ShowAsync();
    }


 }
```

Figure 9: Register and define location with radius

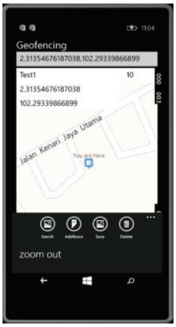
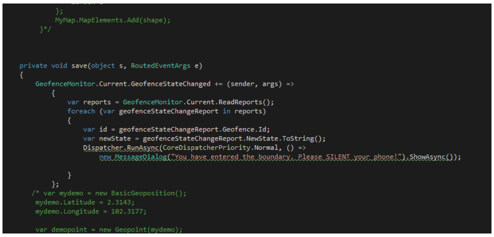


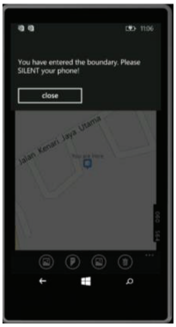
Figure 10: Saving location name, latitude, longitude and radius

Figure 11 shows part of the coding where the geofencing technique is used to detect if the device has crossed the geofence, based on the information that have been stored in the database. Once the device crossed the geofence, a notification will be displayed to remind the user to turn on the silent mode in the device, as shown in Figure 12.



```
            };
            MyMap.MapElements.Add(shape);
        }*/


private void save(object s, RoutedEventArgs e)
{
    GeofenceMonitor.Current.GeofenceStateChanged += (sender, args) =>
        {
            var reports = GeofenceMonitor.Current.ReadReports();
            foreach (var geofenceStateChangeReport in reports)
            {
                var id = geofenceStateChangeReport.Geofence.Id;
                var newState = geofenceStateChangeReport.NewState.ToString();
                Dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>
                    new MessageDialog("You have entered the boundary. Please SILENT your phone!").ShowAsync());

            }
        };
   /* var mydemo = new BasicGeoposition();
    mydemo.Latitude = 2.3143;
    mydemo.Longitude = 102.3177;

    var demopoint = new Geopoint(mydemo);
```

Figure 11: Detecting device entering geofence



You have entered the boundary. Please SILENT your phone!

Figure 12: Notification to remind user to turn on silent mode

The notification to silence the device will be triggered whenever the user crosses the geofence to enter the area of interest. This will prompt the user to set the device to silent mode. When the user crosses the geofence to exit the area of interest, a notification to return the device to normal mode will be triggered, so that the user will be reminded to set the device back to normal mode. The geofencing technique relies on the use of GPS technology. The accuracy of the technique can be further improved by combining GPS with other positioning technique based on WiMAX or UWB [11].

IV. Conclusion

In this paper, the development of auto-notification application for mobile device using geofencing technique has been presented. The application is developed on Windows Phone platform but the technique can easily be applied to other mobile platform. The application helps to remind the user to make sure their mobile device is in silent mode operation whenever they are entering places that require silence such as mosque, library, cinema, and meeting room. The location information of the places such as latitude, longitude and radius will be stored in the database. This information will be used to determine whether the device has crossed the virtual boundary or geofence. If the device has crossed the geofence, a notification will be displayed automatically on the device to remind the owner to turn on the silent mode.

ACKNOWLEDGMENT

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) for their research grant funding under PJP/2012/FKEKK(5B)/S01095.

REFERENCES

[1] R. Penwarden, The Rise of the Smartphone, FluidSurveys, 2014.

[2] National Research Council, The Global Positioning System: A Shared National Asset. Washington, DC: The National Academies Press, 1995.

[3] M. Rouse, "What is geo-fencing (geofencing)? - Definition from WhatIs.com", WhatIs.com, 2015. [Online]. Available: http://whatis.techtarget.com/definition/geofencing.

[4] D. Vaughan, Windows Phone 8 unleashed. Indianapolis, Ind.: Sams Pub., 2013.

[5] S. Wildermuth, Essential Windows Phone 8. Upper Saddle River, NJ: Addison-Wesley, 2013.

[6] P. Bright, "Windows Phone 7 Series in the Enterprise: not all good news", Ars Technica, 2010. [Online]. Available: http://arstechnica.com/information-technology/2010/03/windows-phone-7-series-in-the-enterprise-not-all-good-news/.

[7] S. Hollister, "Microsoft prepping Windows Phone 7 for an October 21st launch? (update: US on Nov. 8?)", Engadget, 2015. [Online]. Available: http://www.engadget.com/2010/09/26/microsoft-prepping-windows-phone-7-for-an-october-21st-launch/.

[8] D. Bass and I. King, "Microsoft Unveils Phone Software With Rival to Apple’s Siri", Bloomberg.com, 2014. [Online]. Available: http://www.bloomberg.com/news/articles/2014-04-02/microsoft-unveils-new-phone-software-with-rival-to-apple-s-siri.

[9] J. Sherman, "Goodbye Windows Phone! Windows 10 will take it from here", Digital Trends, 2014. [Online]. Available: http://www.digitaltrends.com/mobile/windows-phone-dead-windows-10-instead/.

[10] Technet.microsoft.com, "Windows.Devices.Geolocation.Geofencing namespace - Windows App Development", 2015. [Online]. Available: https://technet.microsoft.com/en-us/windows.devices.geolocation.geo-fencing.

[11] M. S. I. M. Zin, M. Hope, A Review of UWB MAC Protocols, 2010 Sixth Advanced International Conference on Telecommunications (AICT), pp.526-534, 9-15 May 2010.