International Journal of Computer Science and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 3, Issue 6, Nov 2014, 11-16

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

ALERT THE EMERGENCY SERVICES THROUGH A MOBILE APPLICATION

DEV MEHTA & HIRAK MODI

Department of Computer Engineering, Thadomal Shahani Engineering College, Mumbai, India

ABSTRACT

Our paper represents a mobile application which will send the location of the user to the main server hosted live, which in turn forwards the co-ordinates to an interface installed at the end of the emergency services such as police, ambulances & fire brigades etc. In just a click!

This mobile application is implemented using the open source android platform. It provides the location using Global Positioning System (GPS) and integration of Google maps by implementing a client server system. The average location accuracy is about 10-20m.

KEYWORDS: Android SDK, Google Maps, GPS, Haversine Formula, Mobile Application, My SQL, PHP

1. INTRODUCTION

Mobile Technology is exactly what the name implies – technology that is portable. Examples of mobile IT devices include: laptops, tablets, mobile phones, smart phone, GPS devices, E-commerce terminals etc. Mobile devices can be enabled to use a variety of communications technologies such as wireless fidelity (Wi-Fi), Bluetooth, 3G, GPS and general packet radio service (GPRS) data services. Smart phones are the most developed of the mobile phones these days [1].

Our paper is about an android based application that is based on the concepts of GPS and integration of Google maps. GPS is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth. There are many commercial users of GPS around the world. The application caters to the people who are in distress and can be helped by the emergencies services. This android based GPS application aims to provide its user with the feature to send an alert message to any of the emergency services (depending on the distress condition) by just clicking the alert button.

2. REQUIREMENTS OF THE SYSTEM

To implement this idea into real world mobile phones, we had to ensure certain requirements that will make sure that the system functionality is ideal and the results are accurate.

Accessibility: The application can be downloaded by anybody with an Android phone. The. apk file can be installed on the device manually or can be downloaded from the Google Play Store.

Usability: The application must be easy to learn and understand. The application interface must be user friendly and simple to use.

Reliability: The latitude and longitude detected by the GPS should be accurate up to 30m range from the user's location.

Optional Requirement: Active internet connection for accurate results.

3. SYSTEM WORKING

Depending upon the situation, there are two cases that can occur. First when the nearest emergency service is available and second when the nearest service is unavailable.

CASE 1: When the Nearest Service is Available

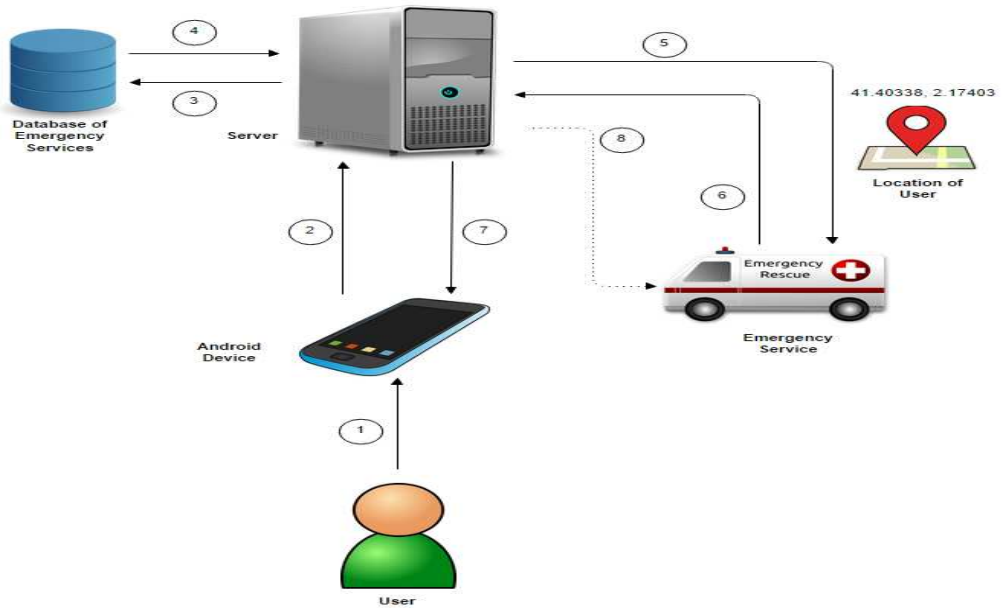

Figure 3-1: Flow Diagram of Case-1 (When the Nearest Service is Available)

Figure 3-1 represents the case when the nearest service is available and the steps followed by the system are as follows:

Step 1: The user starts the application and activates the key to the emergency services.

Step 2: His/her coordinates (Latitude, Longitude) are detected and sent to the server.

Step 3: The server then calculates the nearest emergency service around the victim via the Haversine Formula.

Step 4: The server will have a database of all the emergency services along with their GPS locations, so the server can spontaneously formulate a list of the emergency services in an ascending order of the distances from the user.

Step 5: The server will then forward a request to the nearest emergency service along with the user's coordinates, for which the emergency service is required to respond to.

Step 6: On receiving the request along with the user co-ordinates, here, since the emergency service is available and can accept the request, the service provider sends an 'OK' message to the server and the server establishes a connection.

Step 7: The server periodically keeps track of the user's GPS location (in case there is mobility).

Step 8: The server then gives an update to the emergency service about the location of the user, periodically.

CASE 2: When the Nearest Service is Unavailable

This case arises when the nearest emergency service is unable to accept the request sent by the server. In this case the server chooses the next closest service from the database and forwards the user details to it.

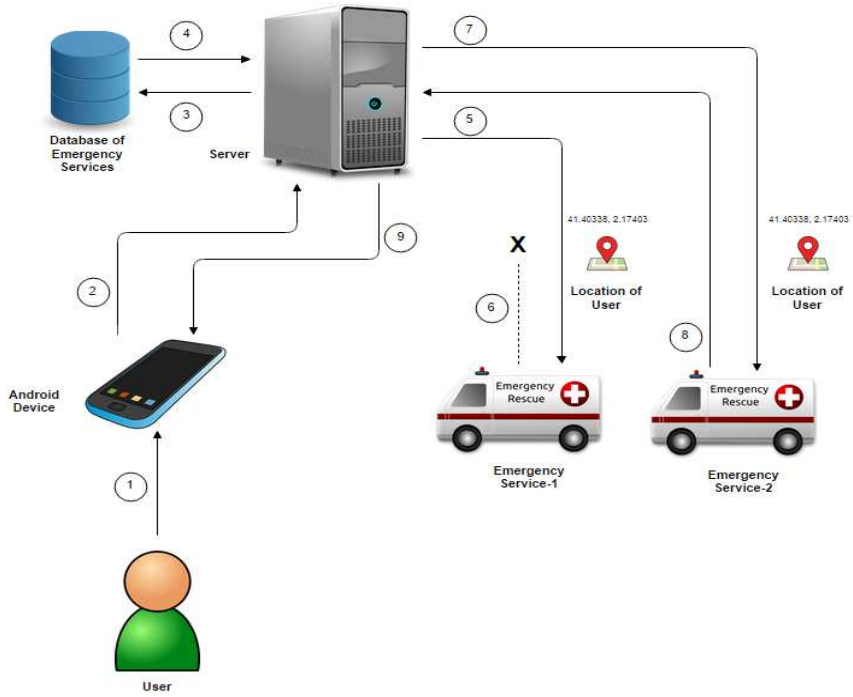

Figure 3-2: Flow Diagram of Case-2 (When Nearest Service is Unavailable)

Figure 3-2 represents the case when the nearest service is unavailable and the steps followed by the system are as follows. Here the first 5 steps are same as Case 1

Step 1 – 5: Similar to Case 1

Step 6 & 7: In case there is no response from the emergency service in a fixed duration or the service denies accepting the request, the server then selects the second (i.e. next nearest) emergency service from the list populated by it and repeats the above procedure again until a connection is established (i.e. when Case 1 holds true).

Step 8: On receiving the request along with the user coordinates, if the emergency service is available and can accept the request, the service then sends an 'OK' message to the server and the server establishes a connection.

Step 9: The server periodically keeps track of the user's GPS location (in case there is mobility) so that it can continuously update the emergency service about the user's location.

4. HAVERSINE FORMULA

As proposed, the server will calculate all the distances on the basis of the GPS locations consisting of the latitude and the longitude only. The server makes use of the Haversine Formula to calculate the great-circle distance between two points – that is, the shortest distance over the earth's surface – giving an 'as-the-crow-flies' distance between the points [2].

The Haversine formula can be seen as follows:

$a = \sin^2(\Delta\varphi/2) + \cos(\varphi 1).\cos(\varphi 2).\sin^2(\Delta\lambda/2)$

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

$d = R.\ c$

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

The Haversine formula 'remains particularly well-conditioned for numerical computa-tion even at small distances' – unlike calculations based on the spherical law of cosines. The 'versed sine' is $1 - cos\theta$, and the 'half-versed-sine' is $(1 - cos\theta)/2 = sin^2(\theta/2)$ as used above. Once widely used by navigators, it was described by Roger Sinnott in Sky & Telescope magazine in 1984 ("Virtues of the Haversine"): Sinnott explained that the angular separation between Mizar and Alcor in Ursa Major – 0°11′49.69″ – could be accurately calculated on a TRS-80 using the haversine [3].

In the formula, c is the angular distance in radians, and a is the square of half the chord length between the points. A (remarkably marginal) performance improvement can be obtained by factoring out the terms which get squared.

5. IMPLEMENTATION AND WORKING

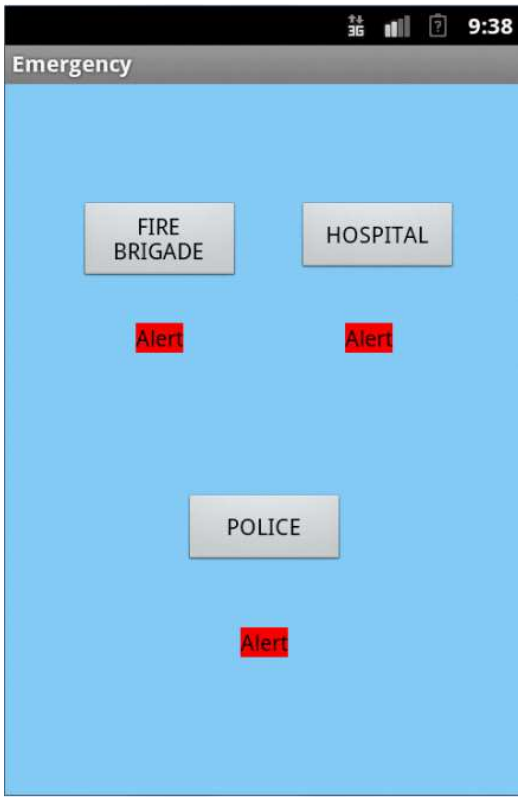

Figure 5-1: Home Screen of Prototype Application

Figure 5-1 shows the home screen of the prototype application. Here we are considering 3 emergency services viz. Fire Brigade, Hospital and Police. The home screen contains three buttons, one for each of the emergency services. As soon as the user clicks on any of the alert button, the location of the user is detected and forwarded to the live server. The server filters the database corresponding to the service selected by the user and then calculates the distances.

The Figure 5-2 shows the filtered results generated by the server when the user clicks on the Fire Brigade Alert button. In this case the user is activating the feature from Andheri, Mumbai, India. So the nearest fire station is displayed first.

Figure 5-2: Screen Displaying the Closest Results for the Fire Brigade Service

Figure 5-3 shows the connection request sent by the server on the android device of the selected emergency service. The emergency service will be provided with the choice of accepting or rejecting the request by the server. Depending upon the response from the service provider, either the connection is established or a new emergency service provider is contacted as explained in section 3.

Figure 5-3: Message Pop-Up at Emergency Service Provider's End

The technologies used to implement this mobile application are Android SDK [4], Eclipse, Google Maps, GPS, MySQL, PHP, and Wamp Server. A live server was hosted and the database of all the emergency services along with their locations updating continuously was created on Wamp Server using MySQL. The Android SDK and Eclipse were used to design the mobile application for Android devices. The application had an integration of Google Maps and GPS in order to detect the locations. PHP was used to create the server.

6. ADVANTAGES

The advantages of this application are as follows:

- A simple hassle free method to instantly alter the emergency services.
- Easily accessible.
- Useful for people in distress conditions.

7. FUTURE SCOPE

While developing the application, we realised the number of features that can be implemented in the application.

They are as follows:

- A novel feature called ‘Get me there safely’ could be implemented, which shall have the ability to point out a crime free road for the user from the source to the destination for late night travels etc.
- A static database addition of all the emergency services. For e.g. a list of all doctors in the area, blood banks and donors.
- An access to information about the general laws, consumer rights, First-aid and health tips.

8. CONCLUSIONS

In this paper, we have presented a way in which a mobile application can be developed in order to help the people in distress conditions. It has great potential for the future. We have illustrated in detail the principle, the functionality of the application and as to how the application can be developed and implemented.

REFERENCES

1. Muthu Natarajanl, S. Muthu Kumar, “Mobile Market Dynamics in India: A New Marketing Paradigm”, International Journal of Applied Services Marketing Perspectives, Volume 2, Number 2, April-June 2013, pp. 354-358.
2. R. W. Sinnott, "Virtues of the Haversine", Sky and Telescope, Vol. 68, No. 2. (1984), pp. 159.
3. Chris Veness, “Calculate distance, bearing and more between Latitude/Longitude points” [online], Available: http://www.movable-type.co.uk/scripts/latlong.html. [Accessed: Sept. 1, 2014].
4. “Location Strategies” [online], Available: http://developer.android.com/guide/topics/location/strategies.html. [Accessed: Sept 2, 2014].