Mobile Location Based Indexing of Notification System

Thu Thu Zan, Sabai Phyu

Abstract—**With mobile technology, the use of mobile phones become easier and faster around the world. At the same time, the mobile users offer valuable information according to their current locations. Then, location based services are important within the emerging new technologies. The users usually desire not only search or query by themselves for information but also get information automatically without searching anything. In this system, the main service task is sending notification to the mobile users according to their current locations. To store the enormous increment of mobile user locations, the system proposed presort range tree indexing and it will support two dimensional range queries. As an experiment, the system will show the performance evaluation with comparison of range searching based on presort range tree and distance based range searching.**

Index Terms—**Location Based Service (LBS), FCM, Google API, Presort Range Tree, 2D Range Query**

I. INTRODUCTION

Everyone who is in IT field says "Today is the age of three things: Cloud Computing, Internet of Things, and Mobile." This word is true because there is no doubt that everything can get huge benefits from them [5].

In mobile age, tracking moving objects are one of the most common requirements for many location based services. In fact, the mobile locations are regularly changed according to moving positions therefore an appropriate strategy for storing and processing them is required.

In summary, in this paper we introduce the presort range tree for dynamic attributes indexing whose main contributions are as follows.

i. Presort Range tree structure is proposed for moving mobile locations with the availability of dynamic range query.
ii. A complete notification system is proposed according to the mobile user locations.

We explain how to incorporate dynamic attributes in presort range tree and a model is added to deal with overall system. Finally, we made a comparison that will show the experimental result based on presort range tree and distance-based range search.

Manuscript received January 15, 2018; revised January 22, 2018. This work was supported in part by University of Computer Studies, Yangon.

Thu Thu Zan was with Software Department, Computer University, Sittwe. She is now with Cloud Computing Lab, University of Computer Studies, Yangon, Myanmar (phone:+95421711504;e-mail:thuthuzan@ucsy .edu. mm).

Sabai Phyu is with the Cloud Computing Lab, University of Computer Studies, Yangon, Myanmar (e-mail: sabaiphyu@ucsy.edu.mm).

II. LOCATION BASED SERVICES

Location based services (LBS) are services based on the device's geographical location given to the mobile phones.

LBS typically provide information or entertainment. LBS largely depend on the mobile user's location. These services can be classified into two types: Pull and Push. In a Pull type, the user has to actively request for information. In a Push type of service, the user receives information from the service provider without requesting it at that instant [2].

III. RELATED WORKS

There are a number of papers that describe about timely disaster notification or alert. Disaster Management Center of The University of Wisconsin said the term Disaster management can be described as "The range of activities designed to maintain control over disaster and emergency situations and to provide a framework for helping at-risk persons to avoid or recover from the impact of the disaster"[6]. Most papers are focus on their nations. Some discuss not only prevention but also evacuation for mobile users.

Prof. Harish Barapatre, Ms shweta rane, Ms salma attar, Ms naina chaudhari made an application to help in the efficient provision of rescue and relief to disaster-affected areas [3]. This application is used for sending the location wherever disaster has taken place. It is also used by user who can provide help in affected areas. It works with two buttons, I NEED HELP and I WANT TO HELP. So this makes an interaction between the victim who is facing disaster and the volunteer who desire to help the victim. Lack of details on Google Map will be the main problem.

Saravana Kumar and Veeramani proposed their own algorithm "Extended Polygon Match Algorithm using Quadtree" to find whether a mobile is within a defined polygon shaped area using their GPS coordinates [7].

Their objectives are to build a prototype system using Android software to send alerts to mobile devices within the defined geographical area and to fix the search area and accurate location is easy using GPS. The advantage of their system is that the disaster target region is perfectly contained. But the other regions also contain when taking the corner points of the polygon in map. They have planned to implement this concept in all the mobiles which is equipped with GPS.

Amit Gosavil, Vishnu proposed to notify the user located in possible disaster zone with visual and audio disaster warning and evacuation guideline combine with nearest location of shelter or safe zone on the map of the application [1]. This system helps out to both normal and blind people to reach to the nearest safe place prior to disaster. However, Lack of details on Google Map of developing countries is the main challenge of their work. They have to implement an application for rescue and relief operation with better server side application to totally automate the system of detecting disaster prone area as future work.

Yavuz Selim, Yilmaz Bahadir, Ismail Aydin Murat Demirbas evaluate arrival times to elaborate how GCM performs (timing performance of GCM), Poisson distribution to the number of devices per time, and conducted chi-squared goodness-of fit test on their models [8]. They point out GCM servers on client device does not by itself guarantee a timely message arrival. GCM is not a good fit for the applications where the broadcasting is mission critical, i.e. the message arrival to all client devices is vital, such as emergency alert services, fire alert systems, instant messaging apps, disaster alert services etc.

Harminder Singh, Dr Sudesh Kumar, Harpreet Kaur explain how GCM service and location service can be combined to develop a new kind of service [4]. They point out the message delivery using GCM is unpredictable, the devices may respond to user commands with different delay. The device must have stable internet connection.

IV. PROPOSED APPROACH

This system is integrated by six major modules: These are: (a) getting locations of registered mobile phones (b) building a system model that will trade off between client and server (c) building presort range tree for dynamic range searches (d) notification message module (e) scheduler process module for not re-compiling and re-deploying the entire system with process flowchart (f) simulation results about the range search, tree construction and preprocessing time for presort range tree. Besides, the comparison between presort range tree with range search and distance-based range search.

A.Getting Mobile Location Framework

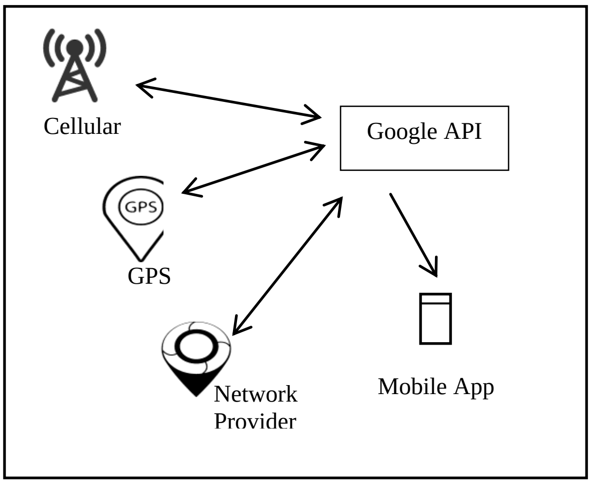

Fig.1. Getting location for Mobile Application

The foundation of location provider is Global Positioning System (GPS). Most of the mobile phones are usually based on GPS to get their locations. In fact, different mobile phones have their own different features and functions to getting location. Some mobile phones have high power of cellular network features and they support accurate locations. Likewise, some mobiles are usually generating their locations based on network provider. Therefore, there is a tradeoff between different mobile phone features and functions; Google API is an optimal solution.

This system has a framework that aids to get current location as fast as it can. This is shown in figure1. It helps to provide a more powerful location framework than usual. This framework is intended to automatically handle location provider's support, accurate location, and update scheduling. It includes the following features.

(a) GPS features –> (GPS, AGPS):**.**

i. determines location using satellites.
ii. does not need any kind of internet or wireless connection.
iii. depending on conditions, this provider may take a while to return a location fix.

(b) Network provider –> (AGPS, CellID, WiFi MACID):

i. determines location based on availability of cell tower and WiFi access points.
ii. results are retrieved by means of a network lookup.

(c) Cellular Network –> (CellID, WiFi MACID):

i. supports for receiving locations without fixing initial location.
ii. returns coarse fixes when the GPS is not enabled.
iii. provides capabilities by connecting to the specific set of hardware and telecom.

In this framework, the require provider should be used by switching between providers depending on the situations.

B. System Model

A model, dynamic object model is built to incorporate dynamic attributes in presort range tree and query processing. This includes a server and a collection of registered mobile objects.

Through a wireless communication network the registered mobile objects connected to database server. Where the current position of mobile object is stored and updated by a Google API at a fixed rate (e.g. 2 sec.). The database is managed by a DBMS which supports schedulers.

In order to keep the location information up to date, these objects regularly send their updated positions to the server. Unnecessary updates wouldn't be performed at the server because Hybrid Update Algorithm is applied to the client side. A scheduler fires and updates the database when the update policies bound are reached. The server is used to range queries like "which mobiles are currently located within a disaster area?" To process such queries

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

efficiently, the server maintains an index tree that, in addition to speeding up the query processing, is also able to absorb all of the incoming updates.

The client-server system model is shown in figure 2.

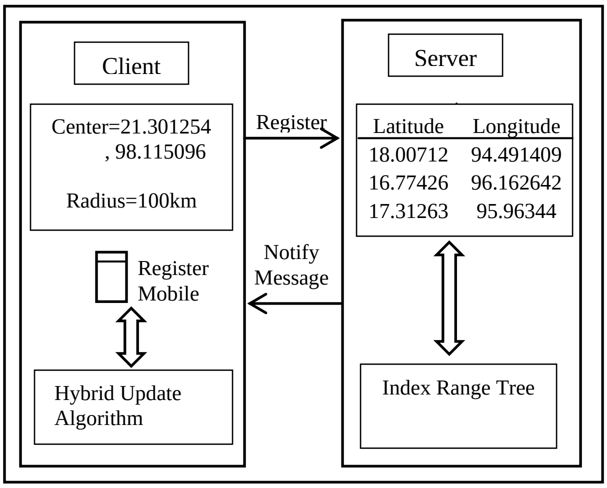

Fig. 2. Client-Server System Model

C. Proposed Presort Range Tree

The procedure of proposed presort range tree is the following;

Input: Lats=Array of two dimensional points sort on latitudes
Longs=Array of two dimensional points sort on longitudes

Procedure PRTree (Lats, Longs)

1. If Lats.length==1 then return new LeafNode(Lats[1]);
2. medium= [Lats.length/2];
3. Copy Lats[1….medium] to Lats L and Lats[medium+1….. Lats.length] to LatsR ;
4. for i=1 to Longs.length do
5. if Longs[i].x <= Lats[medium].x then append Longs[i] to LongsL ;
6. else append Longs[i] to LongsR ;
7. root= new Node((Lats[medium].x), OneDRange (Y));
8. root.left= PRTree(LatsL , LongsL);
9. root.right= PRTree(LatsR , LongsR);
10. return root;

D. Circular Range Search

After preprocessing of tree construction is done, the structure allows searching circular range query for mobile objects. To determines whether registered mobiles are in service area or not so that this system has to **get bounding coordinates** with center and service distance: (centerLat, centerLong, bearing, distance).

bearingRadians = Radians(bearing);
lonRads = Radians(centerLong);
latRads = Radians(centerLat);
maxLatRads = asin((sin(latRads) * cos(distance / 6371) + cos(latRads) sin(distance / 6371) * cos(bearingRadians)));
maxLonRads = lonRads + atan2((sin(bearingRadians) * sin(distance / 6371) cos(latRads)),(cos(distance / 6371) - sin(latRads) * sin(maxLatRads)));

E. Example: Calculating Presort Range Tree with center and service distance

Firstly, sort the mobile locations by latitudes and longitudes.

TABLE I
PRESORTING BY EACH DIMENSION

Sort by X	
16.35099	96.44281
16.77923	96.03917
16.80958	96.12909
24.77906	96.3732
24.99183	96.53019
25.38048	97.87883
25.40319	98.11739
25.59866	98.37863
25.82991	97.72671
25.88635	98.12976
26.15312	98.27074
26.35797	96.71655
26.69478	96.2094

Then the Presort Range Tree is built and shows as the following;

25.40319 98.11739
LEFT: 16.80958 96.12909
LEFT: 16.35099 96.44281
RIGHT: 16.77923 96.03917
RIGHT: 24.99183 96.53019
LEFT: 24.77906 96.3732
RIGHT: 25.38048 97.87883
RIGHT: 25.88635 98.12976
LEFT: 25.59866 98.37863
RIGHT: 25.82991 97.72671
RIGHT: 26.35797 96.71655
LEFT: 26.15312 98.27074
RIGHT: 26.69478 96.2094

The results of sample range search in centerLat, centerLng, distance: 26.693, 96.208, 1000km that are registered mobile locations to send notification as follows:

node (25.40319, 98.11739)
RIGHT: node (24.99183, 96.53019)
LEFT: node (24.77906, 96.3732)
RIGHT: node (25.38048, 97.87883)
RIGHT: node (25.88635, 98.12976)
LEFT: node (25.59866, 98.37863)
RIGHT: node (25.82991, 97.72671)
RIGHT: node (26.35797, 96.71655)
LEFT: node (26.15312, 98.27074)
RIGHT: node (26.69478, 96.2094)

ISBN: 978-988-14047-8-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

F. Notification Message System

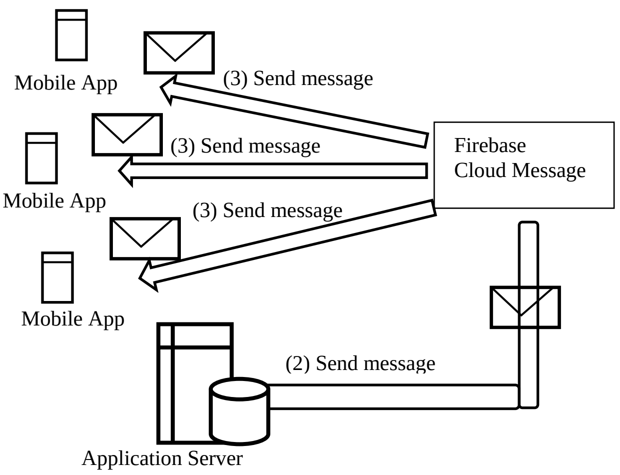

Fig. 3. Notification Message System

The general notification message system is shown in figure**. In this system, mobile Application registers to firebase cloud messaging (FCM) that generates token ID. Server authenticates with FCM by server key. It communicates with FCM by token ID for sending notification.

To send push notification using Firebase Cloud Messaging, the following things are needed for each of the client app and web server. At client app, json configuration file (google-services-json) and libraries, dependencies and plugin which include firebase features are required. Then, the required IDs are the following.

(1) **Sender ID** : unique numerical value API project
(2) **API Key** : save on app server (header post) <<not include in client code>>
(3) **Application ID** : client app register to receive message
(4) **Token** : ID issued by FCM connection servers to the client app allows to receive the message (kept secret)

G. Scheduler Process Module

To schedule tasks without re-compiling and re-deploying the entire system, Spring scheduler is used that supports as the abstraction layer with flexibility and loose coupling. In this system, each execution of the range search is independent so that the beginning of the range search execution doesn't wait for the completion of the previous range search. As a result, multiple range searches are allowed in this system.

To send notification message, the system does the following steps. Firstly, message lists query from the database and check lists that have to send. Based on the message lists, mobile users lists those are not yet received by notification message query from the database and check user lists that have to receive. Presort Range Tree is built and 2D range query are executed according to the require information. At the same time, the mobile user lists are checked whether they are in range or not. If they exist, notification message is sent by Firebase Cloud Messaging and save their lists in the database. The system does not have to send any message or the mobile users are not in range, the system is finished.

The process flow of notification message system is the following figure 3.

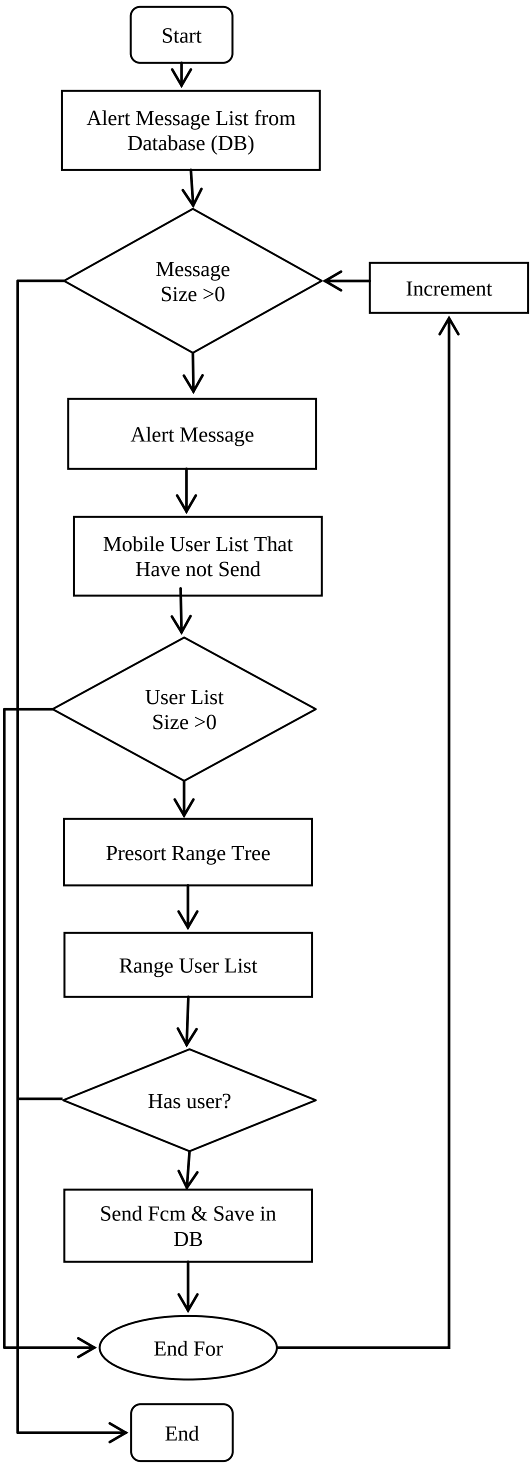

Fig.4. Process flowchart

V. DISTANCE-BASED RANGE SEARCH

This distance-based search determines whether registered mobiles are in service area or not so that this system compares distance between mobile device and center with circle's radius. If the radius of circle is greater than or equal to the distance, the mobile is inside the service area. The following Mobile Region Check (MRC) procedure can be used to decide registered mobile region.

Procedure: MRC (m_latitude, m_longitude, c_latitude, c_longitude, radius)

1. Initialization: mlatitude=mobile's latitude; m_longitude=mobile's longitude; c_latitude= epicenter's latitude; c_longitude =center's longitude; radius=circle's radius, d= distance between mobile device and center;
2. $$d= \sqrt{(clatitude - mlatitude) * (clatitude - mlatitude) + (clongitude - mlongitude) * (clongitude - mlongitude)}$$
3. if (d <= radius) then
4. print: Given point is inside the service area;
5. else
6. print: Given point is outside the service area;

VI. SIMULATION RESULTS

The experiment has been performed on a 2.60 GHz ASUS PC, with Intel (R) Core (TM) i7 CPU and 4 GB memory. For this experiment, we use the most popular testing framework in Java, JUnit. It is an open source testing framework that can be used to write and run repeatable automated tests. The experiment was performed for computing query and execution time for range search, with number of data set points with two dimensions.

To construct presort range tree structure; the first thing is preprocessing the data into the data structure. In this process, presorting is done by each dimension and then it takes into the input parameter for building range tree. The presorting time by different number of dataset is tested in figure 5.

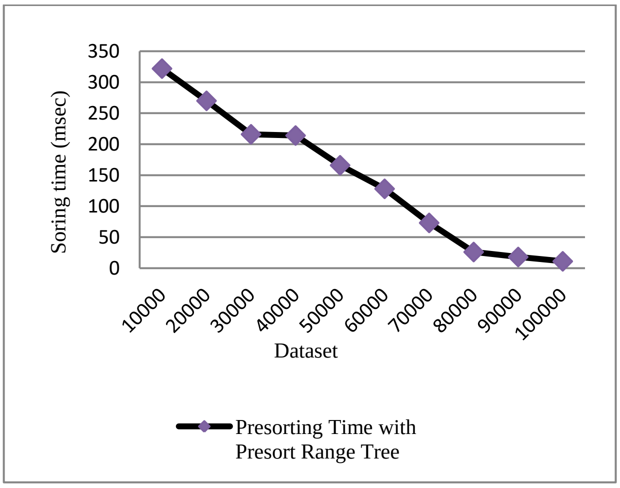

Fig. 5. Preprocessing Time of Presort Range Tree

After taking preprocessing, queries and updates on the data structure are performed. It has been used to compare the performance of both, presort range tree and distance based range search, for data set points in a 2-dimentional space. The execution time required by both approaches was quite different. Generally, query results of both approaches are the same points. The range search time by itself for presort range tree is shown in figure 6.. In this result, range search time takes barely for all different number of dataset.

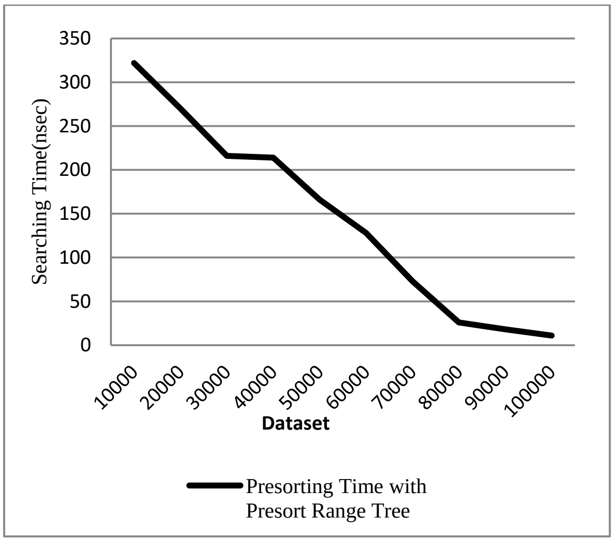

Fig. 6. Range Search Time of Presort Range Tree

Better performance was achieved when the presort range tree was used for larger number of data sets for range search. The more volumes of data tests, the less number of seconds needs in presort range tree. The total execution time is compared in figure 7.

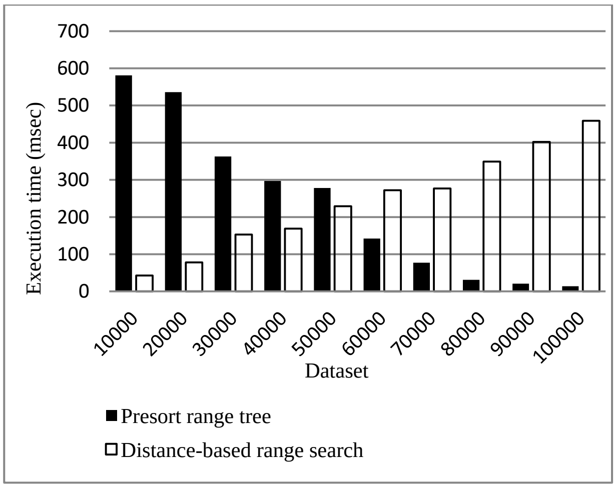

Fig. 7. Execution Time (Preprocessing time +Query time) of Tree and Distance-based range search

VII. CONCLUSION AND FUTURE WORKS

In this paper, the main service task is sending notification to registered mobile phones. The system maintains the moving mobile locations and circular range query is available from the server. Therefore, the system is done for monitoring of mobile objects, to be able to efficiently locate and answer queries related to the position of these objects in desire time. The system will helps to be tradeoff frequency of update due to the locations of mobile objects and reduce server update cost. It also support range query with dynamic object locations. Besides, this system compare presort range tree with range search and distance-based range search by execution time, range time and query time. For future works, the proposed Hybrid Update approach will be applied to other index structures (e.g. the quad tree, the K-D-B tree). Moreover, this proposed system can be used to storing other moving objects such as temperature, vehicle location and so on. The results obtained from the other index tree structure can be compared to this paper's results.

REFERENCES

[1] A.Gosavi1, S. S. Vishnu, “Disaster Alert and Notification System Via Android Mobile Phone by Using Google Map”, India International Journal of Emerging Technology and Advanced Engineering, Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 11, November 2014.

[2] Christian S. Jensen,_ Dan Lin, Beng Chin Ooi, "Query and Update Efficient B+-Tree Based Indexing of Moving Objects", VLDB 04 Proceedings of the Thirtieth international conference on Very large databases, Volume 30 pages 768-779.

[3] Prof. H. Barapatre, “Disaster Management Using Android Technology”, IJRIT International Journal of Research in Information Technology, Volume 2, Issue 4, April 2014, Pg: 171- 183.

[4] H.Singh, Dr S.Kumar, H. Kaur, “Location Based System Using Google Cloud Messaging”, Proceedings f National Conference on Innovative Trends in Computer Science Engineering (ITCSE-2015), 4th April 2015.

[5] Rundle, M.Huffington, “future of technology whitepaper”, UK, 2015.

[6] Sabine Friederike Schulz, “Disaster Relief Logistics: Benefits of and Impediments to Cooperation between Humanitarian Organizations”, January, 2009.

[7] S. Kumar, Veeramani, “GPS Location Alert System”, IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. IV (Mar-Apr. 2014), PP 36-42, www.iosrjournals.org.

[8] Y.Selim, Y. Bahadir, I. A. M. Demirbas, “Google Cloud Messaging (GCM): An Evaluation”, Symposium on Selected Areas in Communications: GC14 SAC Internet of Things, 2014.

,