

STUDY ON LOCATION BASED SERVICES

JAKIA MUN
ID: 05301007

Department of Computer Science and Engineering
Summer 2010



BRAC University, Dhaka, Bangladesh

DECLARATION

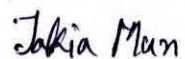
I hereby declare that this thesis is based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.



Signature of Supervisor

Signature of Co-Supervisor

Signature of Author



Acknowledgements

By working on this project I acquired fulfilling experience and passed extremely exciting time. Without the help of some great people and Grameen Phone APIs I could not have finished my thesis.

First, I want to thank my thesis supervisor Dr. Mumit Khan. He was very kind and supportive. Without his supervision it was completely impossible to work on this project. He was always there to correct me and to show me the right and easiest path. I will always be grateful to him for the generous behavior he showed me.

I also want to thank my thesis co-supervisor A.S.M Zillur Rahman. Without his co-operation I could not get the courage to finish my project. Having such busy work schedule he managed to give me some time and allow me to work with his employees. It was been a great help for me to finish my project.

I would also want to thank Grameen phone as they provide us their APIs and that was really such a relief.

Finally, I would like to thank Shiblee and Adnan sir with whom I talked about my project and saved my time to find some solutions.

Special thanks to Rumi vai and Sakib vai from Softalk Pvt Ltd.

Abstract

The structure of my thesis consists of two parts. In Part I, I will explain what location based service is and a short note about GSM network architecture. The thesis will then present some ideas about how SMS service work. Finally, I will finish my Part I by providing a brief idea of how we can find present location of a user in GSM network architecture.

Part II of the thesis will focus on presenting some mobile applications which give location based services. Then my thesis will explain the requirements and implementations of those applications. The thesis will also examine testing phases and results of my applications.

The conclusion of this thesis includes the challenges of my applications and its corresponding solutions along with my future works in this field.

LIST OF FIGURES

Fig 1: the basic GSM network elements.....	[10]
Fig 2: How does SMS service work.....	[13]
Fig 3: A SMS gateway acts as a relay between two SMS centers.....	[14]
Fig 4: A system connects to SMSCs through a SMS gateway.....	[15]
Fig 5: Diagram of Location based SMS Search Application.....	[18]
Fig 6: System diagram for the SMS Search Application.....	[20]
Sequence Diagram for SMS Search Application.....	[22]
Class Diagram for SMS Search Application.....	[23]
Database Tables for SMS Search Application.....	[23]
Dataflow Diagram for LBS API.....	[24]

Literature Survey

- ▶ Free SMS gateway and configuration. www.ozekisms.com
- ▶ What is a SMS gateway. www.developershome.com
- ▶ LBS and SMS APIs. www.aloashbei.com
- ▶ How SMS service work. www.tmcsm.com/how-does-sms-work.aspx
- ▶ How SMS service work. www.kenneyjacob.com/2007/06/19/how-sms-works
- ▶ How to find location of a cell phone electronics.howstuffworks.com/gpsphone.htm
- ▶ GSM Network: http://www.visualgsm.com/gsm_topic04.htm
- ▶ Navigation without GPS: <http://eng.xakep.ru/link/50814/>

TABLE OF CONTENTS

	Page
TITLE.....	I
DECLARATION.....	II
ACKNOWLEDGEMENTS.....	III
ABSTRACT.....	IV
LIST OF FIGURES.....	V
LITERATURE SURVEY.....	VI
TABLE OF CONTENTS.....	VII
PART I – INTRODUCTION TO LOCATION BASED SERVICE	
WHAT IS LOCATION BASED SERVICE?.....	[9]
MOTIVATION.....	[9]
PROJECT GOAL.....	[9]
HOW DOES GSM TECHNOLOGY WORK?.....	[10]
HOW DOES SMS SERVICE WORK?.....	[13]
WHAT IS SMS GATEWAY?.....	[14]
HOW TO FIND OUT PRESENT LOCATION OF A MOBILE USER.....	[15]
Finding location using cell id.....	[16]
Finding location using Grameen Phone LBS API.....	[16]
PART II – PRESENTING SOME LOCATION BASED APPLICATIONS	
SMS SEARCH (WHAT IS NEAR)	
METHODOLOGY	
Introduction.....	[18]
Project Overview.....	[18]
Description.....	[18]
Process.....	[18]
System Requirements.....	[19]
System Architecture.....	[19]

IMPLEMENTATION

Introduction.....	[21]
Inputs and Outputs.....	[21]
Sequence diagram.....	[22]
Program Flow.....	[22]
Error message Reply.....	[22]
Class diagram.....	[23]
Database tables.....	[23]
Data flow diagram of LBS API.....	[24]
Screen shots.....	[25]

REPORT ON THE SPOT

Introduction	[26]
Description.....	[26]
System Requirements.....	[27]
Program Flow.....	[27]
Screen shots.....	[27]

SMS BUZZ (LOCALIZED MARKETING)

Introduction.....	[28]
Description.....	[28]
Inputs and outputs.....	[28]
Program Flow.....	[29]
Screen shot of demo app.....	[29]

LIMITATIONS AND SOLUTIONS.....[30]**FUTURE WORK.....[31]****REFERENCES.....[32]****APPENDIXES.....[33]**

Source codes.....	[34]
-------------------	------

Part I:

Introduction to Location Based Service

WHAT IS LOCATION BASED SERVICE

A location-based service (LBS) is an information and entertainment service, accessible with mobile devices through the mobile network and utilizing the ability to make use of the geographical position of the mobile device [1.1]. A LBS application gives us service through detecting location. It has varieties usages. For example, we could get localized advertising end with vehicle tracking. Not only that we could even play location based games.

MOTIVATION

I was amazed by knowing the services a LBS application can give and felt the lacking of using it in our country. LBS services can be used in a variety of contexts, such as health, work, personal life, etc [1.2]. After Grameen Phone disclosed their LBS API to public for developing some applications I intended to make something useful for our country. Before making applications I need to go through some research work. For example I need to cover GSM network architecture, how SMS service works and as well as SMS gateway.

PROJECT GOAL

There are two main goals for this project. The first goal is to obtain a minimum knowledge of GSM architecture so that I know how the program will work under this network and also to know about how SMS service work and why I need a SMS gateway. It worked as background knowledge of making my applications.

My second goal is to develop applications using LBS API. As my aim was to make SMS based services for the lacking of GPS enable handset in our country and to make life easier for them who have not internet connection on their handset.

HOW DOES GSM TECHNOLOGY WORK

The GSM network is divided into three major systems: the switching system (SS), the base station system (BSS), and the operation and support system (OSS).

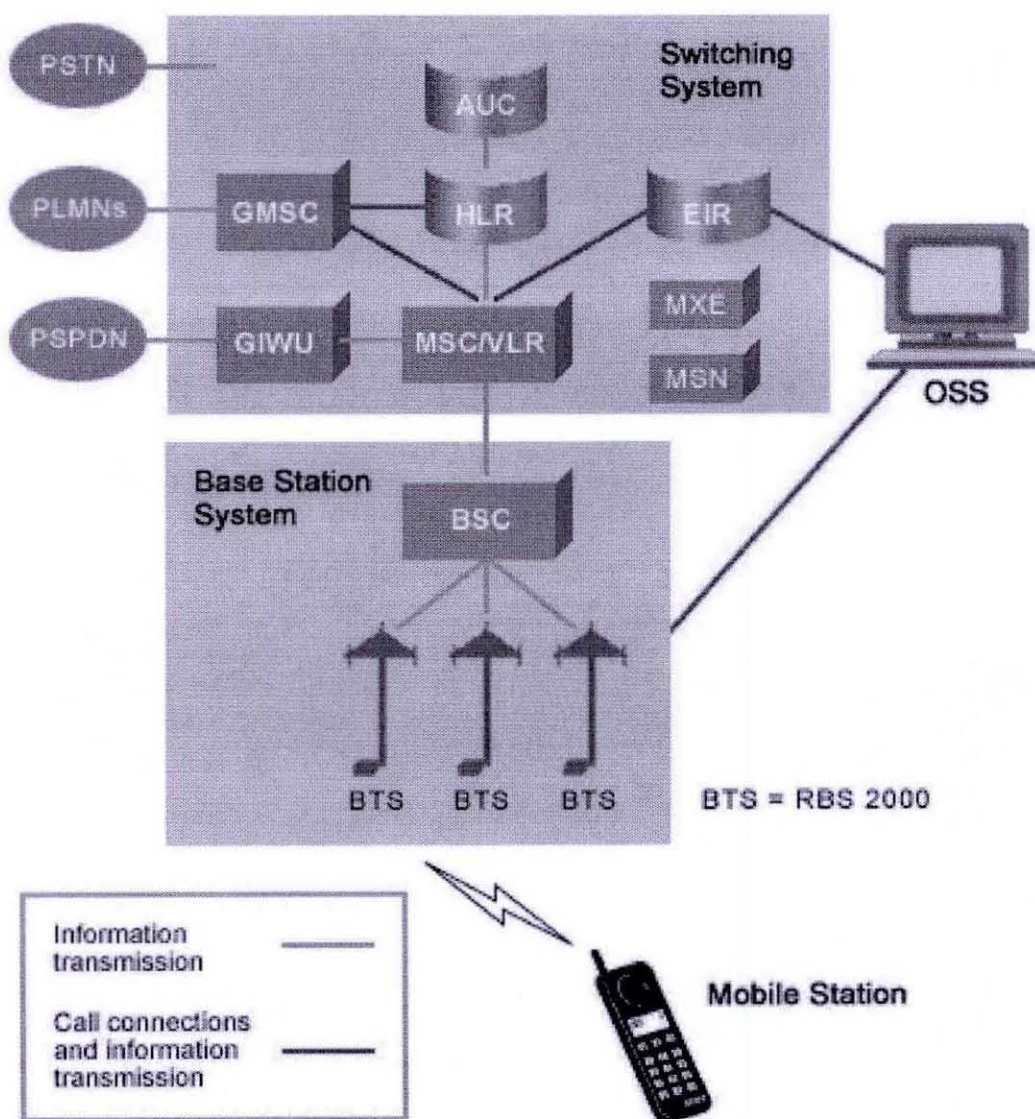


Figure 1: The basic GSM network elements.

The Switching System

The switching system (SS) is responsible for performing call processing and subscriber-related functions. The switching system includes the following functional units.

- **Home Location Register (HLR)** – The HLR is a database used for storage and management of subscriptions. The HLR is considered the most important database, as it stores permanent data about subscribers, including a subscriber's service profile, location information, and activity status. When an individual buys a subscription from one of the PCS operators, he or she is registered in the HLR of that operator.
- **Mobile Service Switching Center (MSC)** – The MSC performs the telephony switching functions of the system. It controls calls to and from other telephone and data systems. It also performs such functions as toll ticketing, network interfacing, common channel signaling, and others.
- **Visitor Location Register (VLR)** – The VLR is a database that contains temporary information about subscribers that is needed by the MSC in order to service visiting subscribers. The VLR is always integrated with the MSC. When a mobile station roams into a new MSC area, the VLR connected to that MSC will request data about the mobile station from the HLR. Later, if the mobile station makes a call, the VLR will have the information needed for call setup without having to interrogate the HLR each time.
- **Authentication center (AUC)** – A unit called AUC provides authentication and encryption parameters that verify the user's identity and ensure the confidentiality of each call. The AUC protects network operators from different types of fraud found in today's cellular world.
- **Equipment Identity Register (EIR)** – The EIR is a database that contains information about the identity of mobile equipment that prevents calls from stolen, unauthorized, or defective mobile stations. The AUC and EIR are implemented as stand-alone nodes or as a combined AUC/EIR node.

The Base Station System (BSS)

All radio-related functions are performed in the BSS, which consists of base station controllers (BSCs) and the Base Transceiver Stations (BTSs).

- **BSC** – The BSC provides all the control functions and physical links between the MSC and BTS. It is a high capacity switch that provides functions such as handover, cell configuration data, and control of radio frequency (RF) power levels in Base Transceiver Stations. A number of BSCs are served by an MSC.

- **BTS** – The BTS handles the radio interface to the mobile station. The BTS is the radio equipment (transceivers and antennas) needed to service each cell in the network. A group of BTSs are controlled by a BSC.

The Operation and Support System

The operations and maintenance center (OMC) is connected to all equipment in the switching system and to the BSC. The implementation of OMC is called the operation and support system (OSS). The OSS is the functional entity from which the network operator monitors and controls the system. The purpose of OSS is to offer the customer cost-effective support for centralized, regional and local operational and maintenance activities that are required for a GSM network. An important function of OSS is to provide a network overview and support the maintenance activities of different operation and maintenance organizations.

Additional Functional Elements

- **Message Center (MXE)** – The MXE is a node that provides integrated voice, fax, and data messaging. Specifically, the MXE handles short message service, cell broadcast, voice mail, fax mail, e-mail and notification.
- **Mobile Service Node (MSN)** – The MSN is the node that handles the mobile intelligent network (IN) services.
- **Gateway Mobile Services Switching Center (GMSC)** – A gateway is a node used to interconnect two networks. The gateway is often implemented in an MSC. The MSC is then referred to as the GMSC.
- **GSM Interworking Unit (GIWU)** – The GIWU consists of both hardware and software that provides an interface to various networks for data communications. Through the GIWU, users can alternate between speech and data during the same call. The GIWU hardware equipment is physically located at the MSC/VL.
[4]

HOW DOES SMS SERVICE WORK

SMS stands for Short Messaging Service and it is when a text message is sent over a wireless phone system. The message then appears on the recipient's screen of their cell phone. Normally text messages can go up to 160 characters but that is for a 1 page SMS. For example, if anyone sends a SMS that has 480 characters, that means he or she will be charged for 3 SMS.

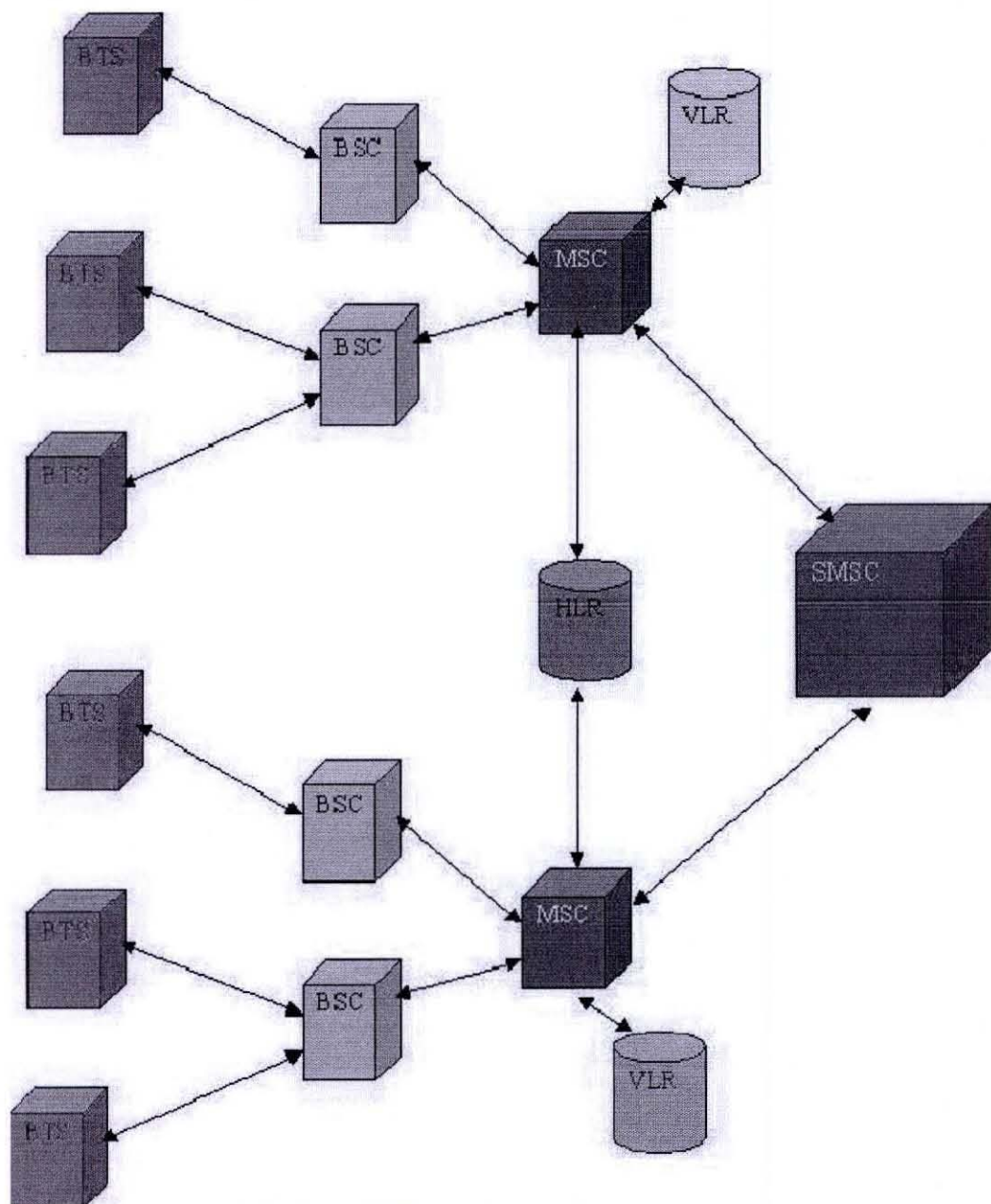


Fig2: How SMS service works

BTS – Base Transceiver Station (Antenna)
BSC – Base Station Controller
MSC – Mobile Switching center
HLR- Home Location Register
VLR – Visitor Location Register
SMSC – Short Message Service Center

When a user sends an SMS, the request is placed via the MSC. The MSC forwards the SMS to the SMSC where it gets stored. The SMSC queries the HLR to find out where the destination mobile is and forwards the message to the destination MSC if the destination mobile is available. The MSC then saves the receiver's information from the VLR and, if necessary, continues with an authentication operation. The MSC then sends the text to a mobile server. If the mobile is not available the message gets stored in the current SMSC itself. In most installations if a mobile is not available for SMS delivery the SMSC will not retry. Instead the destination MSC will inform the SMSC when the mobile comes back in range. SMS handling is a store and forward operation.

SMS has got a validity period, for which it will wait for the destination mobile to be available. After that time the SMSC will delete the message. The validity period can be set by the user. Normal validity is 1 day. [5]

WHAT IS SMS GATEWAY

One problem of SMS messaging is that SMSCs developed by different companies use their own communication protocol and most of these protocols are proprietary. For example, Nokia has an SMSC protocol called CIMD whereas another SMSC vendor, CMG, has an SMSC protocol called EMI. We cannot connect two SMSCs if they do not support a common SMSC protocol. To deal with this problem, an SMS gateway is placed between two SMSCs. This is illustrated in the following figure. The SMS gateway acts as a relay between the two SMSCs. It translates one SMSC protocol to another one. This way can be used by two different wireless carriers to interconnect their SMSCs for purposes such as enabling the exchange of inter-operator SMS messages.

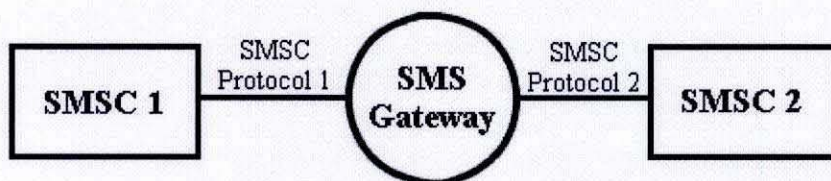


Fig 3: A SMS gateway acts as a relay between two SMS centers.

To connect to an SMS gateway, we can use an SMSC protocol such as SMPP and CIMD. Some SMS gateways support an HTTP / HTTPS interface. HTTP / HTTPS is easier to use than SMSC protocols. The drawback is that there may be fewer SMS features to use. For example, an SMS gateway may not support the sending of picture messages through the HTTP / HTTPS interface. [2]

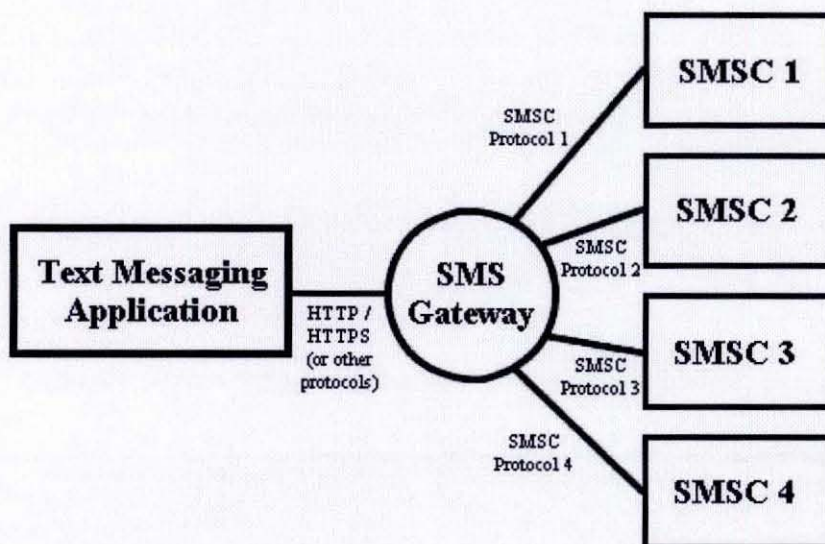


Fig 4: A SMS text messaging application connects to SMSCs through a SMS gateway.

HOW TO FIND OUT PRESENT LOCATION OF A MOBILE USER

A cell phone is basically a sophisticated two-way radio. Towers and base stations, arranged into a network of cells, send and receive radio signals. Cell phones contain low-power transmitters that let them communicate with the nearest tower.

As we travel, we move from one cell to another, and the base stations monitor the strength of our phone's signal. As we move toward the edge of one cell, our signal strength diminishes. At the same time, the base station in the cell we are approaching notices the strength of our signal increasing. As we move from cell to cell, the towers transfer our signal from one to the next. In remote locations, towers may be so far apart that they can't provide a consistent signal. Even when towers are plentiful, mountains and tall buildings can interrupt their signals. Sometimes people have a hard time getting clear signals inside buildings, especially in elevators.

Even without a GPS receiver, our cell phone can provide information about our location. A computer can determine our location based on measurements of our signal, such as:

- Its angle of approach to the cell towers
- How long it takes the signal to travel to multiple towers
- The strength of our signal when it reaches the towers

Since obstacles like trees and buildings can affect how long it takes our signal to travel to a tower, this method is often less accurate than a GPS measurement. In order to determine our location, a GPS receiver has to determine:

- The locations of at least three satellites above us
- And where we are in relation to those satellites

The receiver then uses trilateration to determine our exact location. Basically, it draws a sphere around each of three satellites it can locate. These three spheres intersect in two points -- one is in space, and one is on the ground. The point on the ground at which the three spheres intersect is our location.

A GPS receiver has to have a clear line of sight to the satellite to operate, so dense tree cover and buildings can keep it from getting a fix on our location. GPS receivers and cell phones have a lot in common, and both are very popular. [6]

Finding location using cell id

There are two steps involved in using CellID to find a device's location:

- Obtain the CellID of the base station to which the device is connected.
- Resolve the CellID into latitude and longitude.

To obtain the CellID of a Windows Mobile device, we have to use the RIL (Radio Interface Layer) API in the Windows CE operating system. The Radio Interface Layer (RIL) provides an interface that handles the communication between the CellCore system software and the radio hardware.

Once the CellID is obtained, it is sent back to a secret API at Google Maps at www.google.com/glm/mmap, which will then return the latitude and longitude to which the CellID corresponds. Apparently, Google Maps has a huge database containing the entire world's CellIDs [3]. (Not sure about Bangladesh)

Finding location using Grameen Phone LBS API

Here, in my project I used Grameen Phone LBS API to find out present latitude and longitude number of a mobile user.

PART II
PRESENTING SOME LOCATION BASED APPLICATIONS
SMS SEARCH (WHAT IS NEAR)

METHODOLOGY

A. PROJECT OVERVIEW

SMS Search Application: What is Near

Description:

User can make queries by sending text message to a particular number and can get the address of nearby Hospitals, Hotels, Police Stations, Restaurants, Schools, Cafés, Shopping Malls and Bazaars.

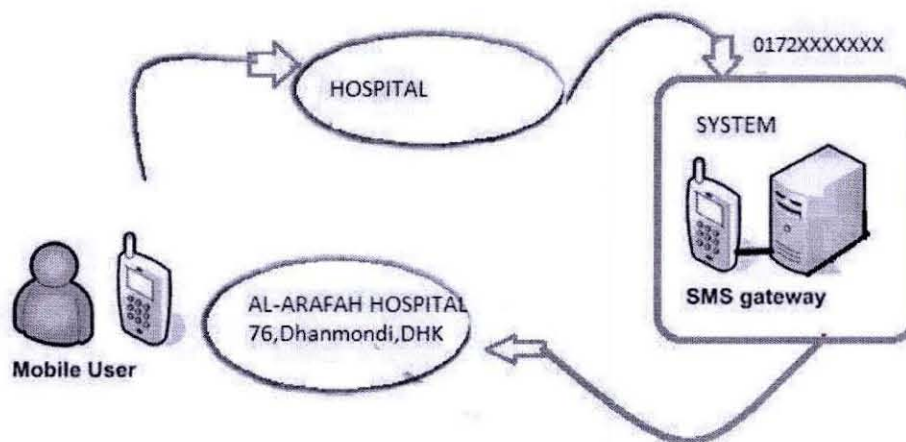


Fig 5: Diagram of Location based SMS Search Application

Process:

After sending text message to a particular number, the program will find out the present longitude and latitude number of the user then it will make the database queries into the system according to the user requirements and hence, will forward the reply text message to the user mobile hand set.

Hardware Requirements for the System

SIM Card
GSM Modem or Mobile hand set,
Data cable,
PC

Software Requirements for the System

SMS Gateway software
Database Server

User Requirements

Mobile hand set
SIM Card
User account of www.aloashbei.com

System Architecture

This mobile application is using SMS gateway software to forward the SMS to the application server. Then this SMS is handled by the JAVA and MySQL. This application used Grameen Phone LBS API to find out current location.

Gateway

Ozeki Message Server 6 - SMS gateway is a powerful, flexible SMS Gateway application that enables any applications to send/receive SMS messages to mobile devices with any computer. It has an easy to use user interface, and an excellent internal architecture. The application can use a GSM mobile phone attached to the PC with a phone-to-PC data cable to transmit and receive the messages. For example if an SMS message needs to be sent, it can be inserted into a database table used for outgoing messages. The Ozeki Message Server monitors this table and delivers the message. The Message Server puts all received SMS in another database table used for incoming messages.

MySQL

This implementation uses a MySQL database to maintain location database as well as information about hospitals, colleges, mosque, embassy etc. A typical database is organized into several tables. Each of which have many rows of data. The database can

be queried using SQL to retrieve a single or multiple rows from one or several tables within the database. The database can be readily accessed by JAVA application.

Grameen Phone LBS API

This application uses Grameen Phone LBS API to find out present latitude and longitude value of a mobile user.

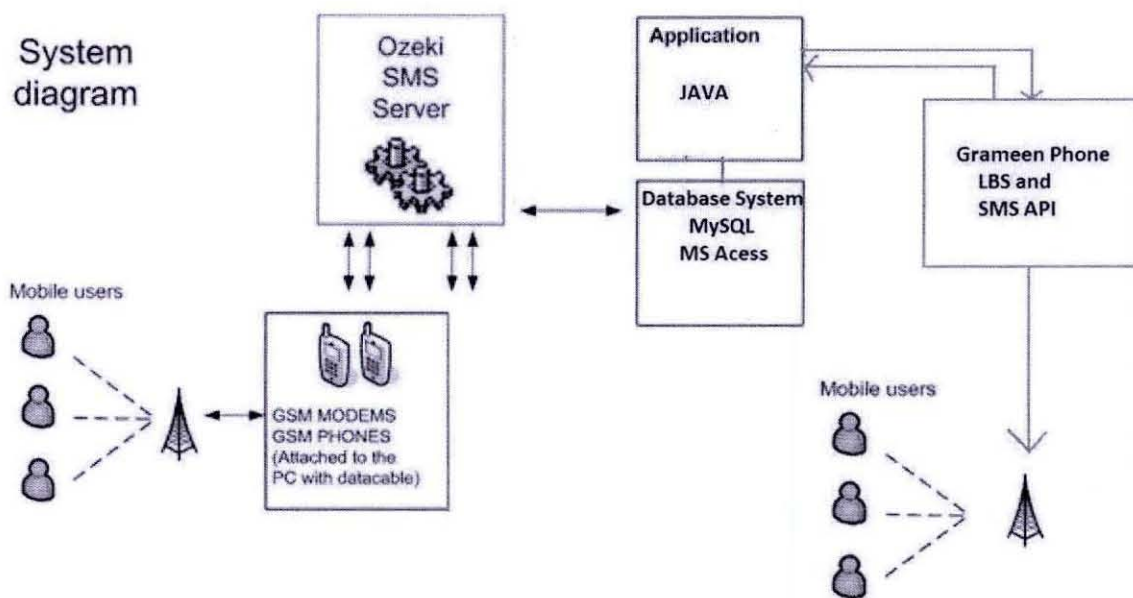


Fig 6: System diagram for the application

JAVA

This application is written in JAVA programming language. It uses the NetBeans 6.8 IDE.

MS Access

MS Access is used to store incoming text messages. This is also connected to sms gateway. So when any sms comes the Ozeki Message Server monitors this table and delivers the message to the application.

IMPLEMENTATION

Introduction

This application starts with sending a text message in a particular GSM mobile number which is connected to the sms gateway. Then the sms is saved in MS Access database system. From there the application program gets that message and tokenizes it into parts and calls LBS API to know the current latitude and longitude value of that user. After getting those values it sends SQL queries into MySQL database to find out the particular information according to user's sms. When it gets the information it then calls the SMS API to send back the information to the user.

The features of this application are given below:

- Nearby location name
- Nearby location address
- Nearby location distance
- And direction to reach there

Inputs and Outputs:

User: (aloashbei) userid password university

Reply: you are close to mohakhali BRAC University at Mohakhali total distance 0 km.

OR

User: (aloashbei) userid password university mohakhali

Reply: you are close to mohakhali BRAC University at Mohakhali total distance 0 km.

Program works in two ways:

1. One can use his/her aloashbei login information, so that the program can identify his/her present location.

Or

2. One can choose the area name given in the application. (in this version I just gave 5 areas for example: gulshan, mohakhali, tejaon, baridhara, boshundhara)

After that one needs to give the category name of what they are looking for such as hospital, university, police station, embassy, college, and mosque.

The Sequence Diagram of the application

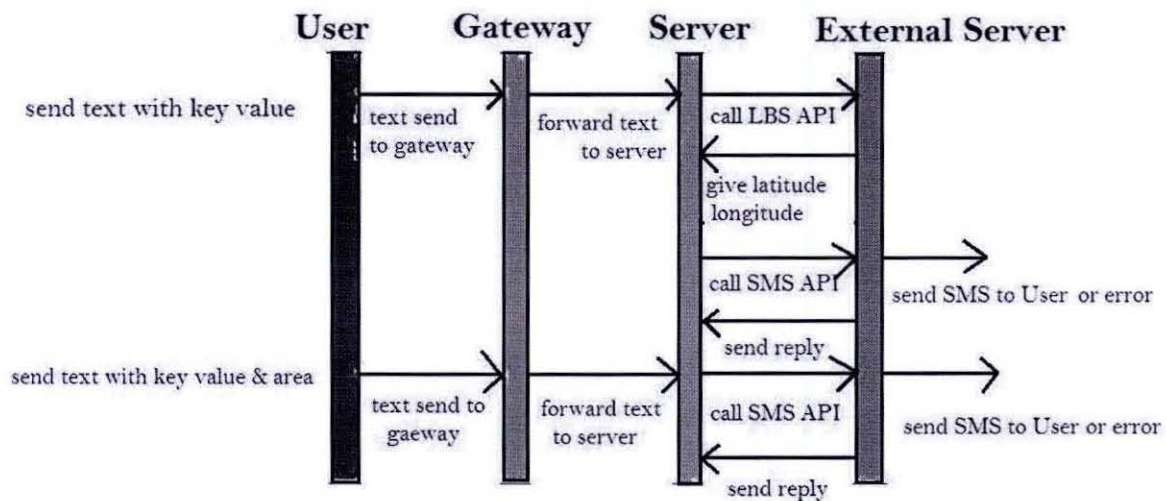


Fig 7: sequence diagram

Program Flow

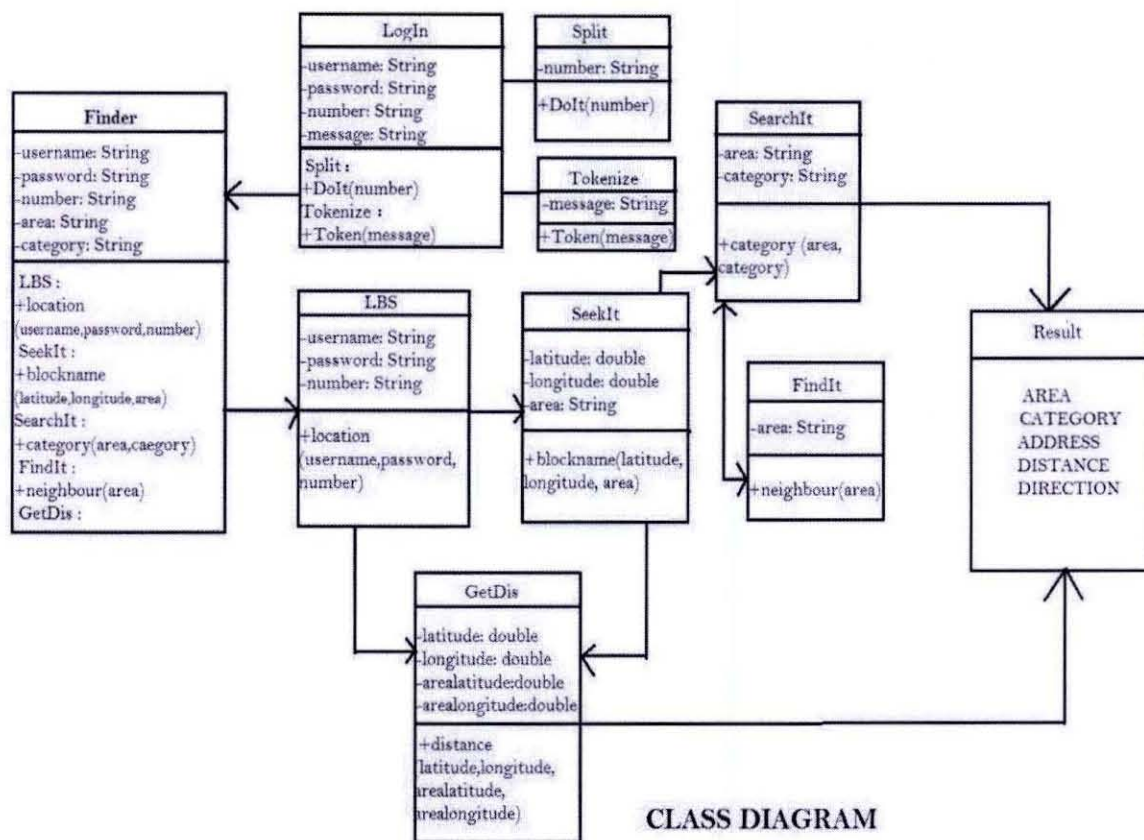
If anyone chooses "area" gulshan and "category" embassy then the program will search embassy near gulshan. If it does not find embassy at gulshan then it will look at its neighbor area for example "baridhara" to find out the match.

Again, if anyone chooses "area" tejgaon and "category" embassy then again if it does not find embassy at tejgaon it will look at its neighbor area, (here its mohakhali) but if it does not find embassy in its neighbor then it will tell user no match found in its neighbor. It is just because the program sets like this way. User will get information what is near to his or her.

Error Message Reply

1. If anyone does not provide any information like no login information.
2. If anyone chooses such category which does not match his or her present location and also not in his or her neighbor areas (which is set in the program) then user will get an error message.
3. If Grameen phone aloashbei server error happens.

Class Diagram for SMS Search (WhatIsNear)



Database Tables for SMS Search (WhatIsNear)

LogIn	
username	String
password	String
message	String

Block	
ID	int
Flati	double
Llati	double
Block	String
Flong	double
Llong	double

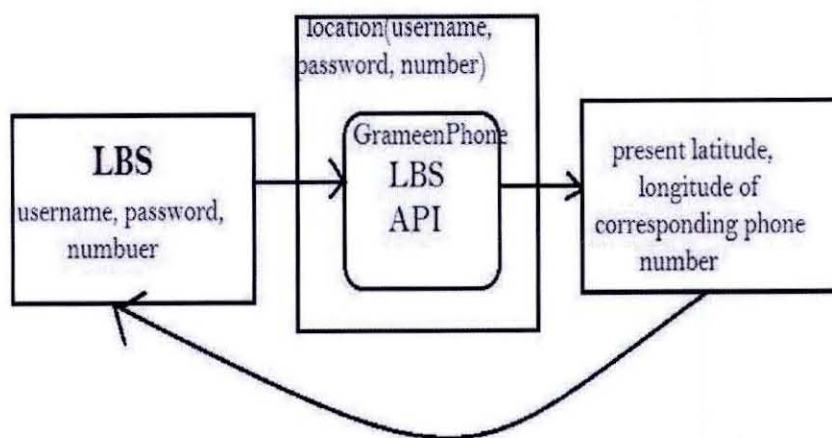
Database table for LogIn and Block

Area	
ID	int
latitude	double
longitude	double
category	String
name	String
address	String
neighbour	String
ndirection	String

loc	
ID	int
total	double

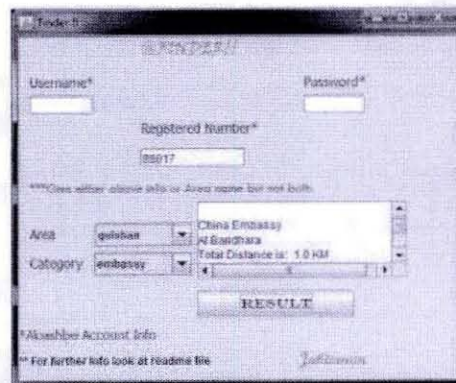
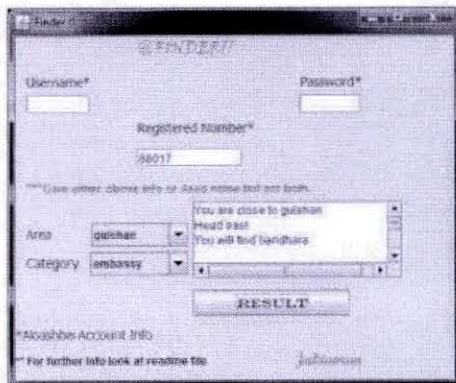
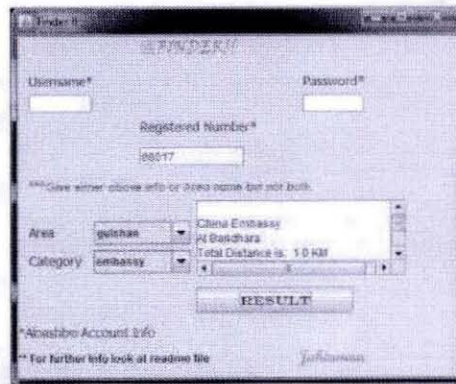
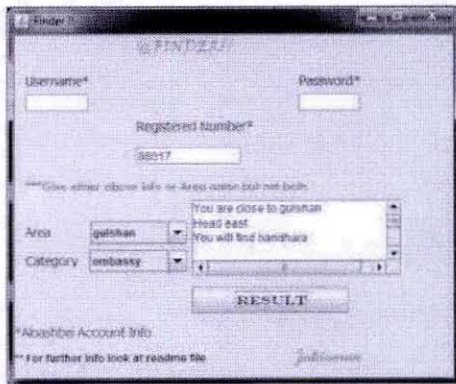
Database tables for Area and loc

Data Flow Diagrams for getting location using LBS API



Data flow diagram for getting location

Screenshots of Desktop based SMS Search application



REPORT ON THE SPOT

Introduction

Description

This is an application which will be helpful for sending report to the legal authority and it's a location based application which means in the case of emergency one can send report on the spot and the application will find out user's present location and forward his or her message to the nearest police station number based on its database. For that no one need to know the number of any police stations.

This concept can be applicable for any similar scenario. For example, to make a report on a Fire Service Office. In that case, users can send their addresses as a report and this SMS will be forwarded to the nearest Fire Service Station.

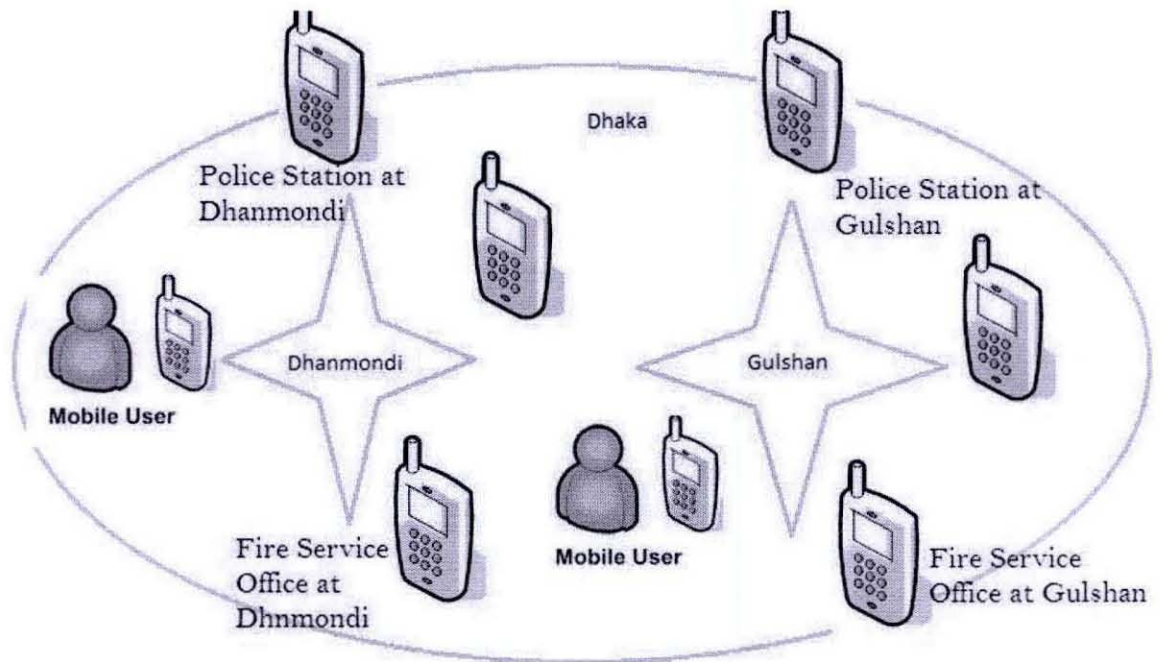


Fig 8: diagram of report on the spot

System Requirements:

- JAVA
- MYSQL
- LBS API
- SMS API

Program Flow for the Desktop based application:

- User sends SMS
- Program uses LBS API to locate user's location
- It matches through its database to find out the area
- Then it finds out the particular Police Station number on that area
- Forward that message using SMS API to that particular number

Screen shot



SMS BUZZ (LOCALIZED MARKETING).

Introduction

Description

SMS BUZZ is a SMS based advertising application. It is a location based service also. That means, user of a particular location can send SMS and get the updates of any products or services which are given by a particular area.

For example, a Hair Salon Shop can give the updates of its new hair cut price rate or a Medical Center can give updates of its available services and so on.

For Producer:

To give this advertisement one needs to send a SMS with the Company Name and Category like name **Hair Salon Shop** and category **Price Rate** in a particular number. Then this SMS will be saved in database with the shop's location and later when someone sends query from that area to know all-around updates he or she will get this updates by sorting categories.

For Customer:

To get the updates which are subdivided by areas one needs to send SMS and just need to mention the category. First, the program will find out user's current location and will give him or her recent updates based on the area where he or she is now. User can also mention the area and category to know the updates of that area.

Input for Producers:

Hair Salon Shop, Cut and Blow Dry: 200tk

(It will be saved in a database and later will be forwarded to a particular user.)

Input for Customers:

Dhanmondi (mention the area name is optional) Hair Salon

Output for Customers:

And after sending this query user will get back the SMS of

Hair Salon Shop, Cut and Blow Dry: 200tk

Program Flow:

- Producer will send SMS by advertising its products or services
- Program will find out sender's location by LBS API and will save this SMS in a database
- Later when any customer near that location will send query to get particular updates
- Then the Program will forward those SMS which are created by Producers and which are saved in its (Program) database.

Screen Shot of demo app

The screenshot shows a web application window titled "SMS BUZZ". The interface includes the following elements:

- Username*** and **Password*** input fields.
- Registered Number*** input field with the value "9917".
- Category** input field.
- BUZZ** input field.
- POST** and **FIND** buttons.
- Footer text: "Microsoft Internet Explorer" and "johannes".

LIMITATION AND SOLUTION

Limitations exist in my application due to inaccurate latitude and longitude given by the LBS API and also the Database system can be improved by enriching with data and strong relation between them.

Limitations in current tools and technologies

Inaccuracy in LBS API:

This application uses Grameen phone LBS API and that API does not guarantee accuracy in finding latitude and longitude value of a mobile user.

External server error:

It can also give a threat to my application. As this application uses external server for LBS and SMS APIs so this program also depends on this server.

Authentication:

Perhaps, this is the major drawback of the system. The Grameen phone LBS and SMS APIs do not allow all users to use their APIs rather than who are registered in this www.aloashbei.com site.

Maximum hit count:

Users cannot use this APIs for maximum 20 times in a day.

SMS characters limitation:

For security reason, only first 50 characters can be sent.

Solutions:

Grameen phone is working on their APIs to improve their system. In future, may be they can change their security policy such a way so that program flow does not hamper.

Current application limitations:**Little data at this time:**

Application database does not cover entire Dhaka city or whole country. It just uses five areas of Dhaka city for testing purpose.

Unable to Map the entire city:

Limitation lies due to inaccurate mapping of the city. The problem is to draw accurate boundary between the areas.

Limited functions:

At this time the features of this application are only limited to give nearby name, address, location and direction of a place.

Limited search:

User only can search what is near to him or her but not for the other areas. For example, if user is in Tejgaon and searches embassy he or she will not get the result unless he or she is in the Baridhara.

Very limited driving direction:

Due to less data information, program does not give full driving direction at this stage.

Solutions

There are many places for improvement. First of all, application like this needs huge database system and good relations between data tables. I only emphasis on system architecture that is how sms and web server can interact and just used some data tables for testing purpose. All the limitations of this application except current tools and technologies depend on how program database is created and manipulated. So improvement of this part will gradually make this project big, reliable and useful.

FUTURE WORK

The future work will focus the improvement of the existing system by giving more features in it. Some are mentioned below:

Cover the entire Dhaka city:

In future this system will cover the entire Dhaka city not only just few areas.

Good mapping system:

The mapping system of the existing system is very poor. So it needs tremendous improvement and consistent boundary system between areas. The program will work on it.

Unlimited search option:

In future, users will not only just get the information of nearby places rather than the entire city by giving proper categories. There will be help option for making better search.

Unlimited features:

This application will add more features in it. For example, it will be able to show not just the places but also about the people or friends around the user. This application will also include the localized marketing feature in it. That means at the same time user will find out what is selling and buying around him or her.

Driving and walking direction:

In future, this application will give full fledged driving and as well as walking direction to the user.

REFERENCES

[1.1] Definition of LBS: "Foundations of Location Based Services", Stefan Steiniger, Moritz Neun and Alistair Edwardes, University of Zurich

[1.2] usages of LBS: Deuker, André (2008), "Del 11.2: Mobility and LBS", *FIDIS Deliverables 11* (2), <http://www.fidis.net/resources/deliverables/mobility-and-identity/>

[2] How SMS service works www.developershome.com

[3] Cell ID for windows mobile: <http://www.devx.com/wireless/Article/39709/0/page/1>

[4] GSM Network: http://www.visualgsm.com/gsm_topic04.htm

[5] How sms service works www.kenneyjacob.com/2007/06/19/how-sms-works

[6] To find out present location electronics.howstuffworks.com/gps-phone.htm

APPANDIXES

Source codes for SMS Search application:

The source codes used for this application are given below. It has several classes which are highlighted properly. One can match these codes by looking at class diagram I have included in this report.

Some Class descriptions are given below:

Split: To remove the "+" sign from user number.

Tokenize: To tokenize the message into parts.

LBS: For calculating the longitude and latitude value.

SeekIt: For finding out the area names.

SearchIt: To search in the category table for exact match.

GetDis: To calculate the distance.

FindIt: To find out the neighbor areas.

SMS: To send back sms to user.

Main class of the Program:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package msaccess;

import java.sql.*;

/**
 *
 * @author User
 */
public class Main {

    /**
     * @param args the command line arguments
     */

    public static void main(String[] args) throws InterruptedException {

        // TODO code application logic here

        String val = null;
        String val2 = null;
        String num = null;
        int count = 15;
        int id = 16;
        int x = 0;
```

```
try {  
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  
    String url = "jdbc:odbc:DRIVER={Microsoft Access Driver  
(*;.mdb)};DBQ=C:\\Users\\User\\Documents\\sms.mdb";  
    Connection con = DriverManager.getConnection(url);  
    System.out.println("Connected!");  
  
    Statement st = con.createStatement();  
  
    while (x < 10) {  
        System.out.println("ID is: " + id);  
        Thread.sleep(20000);  
  
        ResultSet rr = st.executeQuery("select sender,msg from ozekimessagein where id =  
"+id+"");  
        while (rr.next()) {  
            count++;  
            System.out.println("Count is: " + count);  
  
            String d = rr.getString(1);  
            val = d;  
            System.out.println("Number = " + val);  
  
            String e = rr.getString(2);  
            val2 = e;
```

```
System.out.println("Message = " + val2);
```

```
Split split = new Split();  
num = split.doit(val);  
System.out.println("Check from main " + num);
```

```
Tokenize tokenize = new Tokenize();  
tokenize.Token(val2, num);  
}
```

```
if (id == count) {  
    id++;  
}  
x++;  
}
```

```
st.close();  
con.close();
```

```
} catch (SQLException e) {
```

```
        System.out.println("SQL Exception: " + e.toString());
    } catch (ClassNotFoundException cE) {
        System.out.println("Class Not Found Exception: "
            + cE.toString());
    }
}
}
```

Split Class:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package msaccess;

/**
 *
 * @author User
 */
public class Split {

    public String doit(String val) {
        String s3 = val;
        String[] temp = null;
        temp = s3.split("\\|+");
    }
}
```

```
        // temp = s3.split("-");
        String value = temp[1];
        System.out.println("Check " + value);
        return value;
    }
}

Tokenize class:
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package msaccess;

import java.util.StringTokenizer;

/**
 *
 * @author User
 */
public class Tokenize {

    public void Token(String value,String num) {
        // TODO code application logic here
        int idx = 0;
        int tokenCount;
        String words[] = new String[255];
```

```
// String message="The text of the message to be scanned.";
String message = value;
StringTokenizer st = new StringTokenizer(message);
tokenCount = st.countTokens();
System.out.println("Number of tokens = " + tokenCount);

while (st.hasMoreTokens()) // is there stuff to get?
{
    words[idx] = st.nextToken();
    idx++;
}
for (idx = 0; idx < tokenCount; idx++) {
    System.out.println(words[idx]);
}

String user = words[0];
System.out.println("User ID is: " + user);

String password = words[1];
System.out.println("Password is : " + password);

String category = words[2];
System.out.println("Category is: " + category);
```

```
String area = words[3];  
System.out.println("Area is: " + area);
```

```
Finder finder = new Finder();  
finder.findmain(user,password,num,category,area);  
}  
}
```

LBS Class:

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package msaccess;
```

```
/**  
 *  
 * @author User  
 */
```

```
class LBS {
```

```
    static class Returner {
```

```
        double a = 0.0;
```

```
        double b = 0.0;
```



```
    }  
    public LBS() {  
  
    }  
  
    public static Returner Location(String val, String val2, String val3) {  
  
        double lat = 0.0;  
        double lon = 0.0;  
        String registrationID = val;  
        //"jakiamun";  
        String password = val2;  
        //"hellogp";  
        String msisdni = val3;  
        //"8801717097886";  
  
        try { // call web service operation  
            com.grameenphone.playground.WebServiceGPADPBizTalkLBSOrchestration service = new  
com.grameenphone.playground.WebServiceGPADPBizTalkLBSOrchestration();  
  
            com.grameenphone.playground.WebServiceGPADPBizTalkLBSOrchestrationSoap port =  
service.getWebServiceGPADPBizTalkLBSOrchestrationSoap();
```

```
//initialize WS operation arguments here
```

```
lbsrequest.schemas.lbs.biztalk.adp.gp.LBSRequest lbsRequest = new  
lbsrequest.schemas.lbs.biztalk.adp.gp.LBSRequest();
```

```
lbsRequest.setRegistrationID(registrationID);
```

```
lbsRequest.setPassword(password);
```

```
lbsRequest.setMsisdn(msisdn);
```

```
//process result here
```

```
lbsresponse.schemas.lbs.biztalk.adp.gp.LBSResponse result = port.requestLocation(lbsRequest);
```

```
System.out.println("Result = " + result.getLatitude() + " " + result.getLongitude() + " " +  
result.getStatus() + " " + result.getTimestamp());
```

```
lat = Double.parseDouble(result.getLatitude());
```

```
System.out.println(lat);
```

```
lon = Double.parseDouble(result.getLongitude());
```

```
System.out.println(lon);
```

```
} catch (Exception ex) {
```

```
ex.printStackTrace();
```

```
    }  
  
    Returner myReturner = new Returner();  
    myReturner.a = lat;  
    myReturner.b = lon;  
    return myReturner;  
}  
}
```

Finder Class:

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package msaccess;  
  
/**  
 *  
 * @author User  
 */  
import java.sql.*;  
  
public class Finder {  
  
    public void findmain(String val, String val2, String val3, String val5, String val4) {  
  
        double vau = 0;
```

```
double vau2 = 0;
double value = 0;
double value2 = 0;
Connection con = null;
String cate = null;
String f = null;
double g = 0.0;
double w = 0.0;
String ss = null;
String yy = null;
int digit = 0;
int spin = 1;
int id = 0;
String ngh = null;
String ND = null;
double z = 0;
String msg1 = "";
String msg2 = "";

try {

    Class.forName("com.mysql.jdbc.Driver");
    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/javasql", "root", "");
    try {
```

```
Statement st = con.createStatement();

LBS.Returner ret = LBS.Location(val, val2, val3);

value = ret.a;
value2 = ret.b;

vau = value;
vau2 = value2;

System.out.println("area is : " + val4);

SeekIt seek = new SeekIt();
SearchIt search = new SearchIt();
FindIt find = new FindIt();
f = seek.Blockname(value, value2, val4, val3, val, val2);
System.out.println("area is: " + f + " category is: " + val5 + " number is: " + val3);
if (f != null) {
    cate = search.Category(f, val5, val3, val, val2);
    digit = find.Neighbour(f);

    ngh = f;

    msg1 = "you are close to " + f;
    // ResultTextArea.append("You are close to "+ f + "\n");
```

```
while (spin <= digit) {  
  
    System.out.println("am from while");  
  
    if (cate == null) {  
        id = id + 1;  
  
        System.out.println("id number is = " + id);  
  
        ResultSet result = st.executeQuery("select neighbour,ndir from " + f + " where ID = " + id +  
" and neighbour is not null");  
        while (result.next()) {  
            String d = result.getString(1);  
            ngh = d;  
            String nn = result.getString(2);  
            ND = nn;  
        }  
  
        System.out.println("Neighbour name = " + ngh);  
  
        cate = search.Category(ngh, val5, val3, val, val2);  
        if (cate != null) // ResultTextArea.setText("You are close to: " + f + "\n" + "Head " + ND  
+ "\n" + "You will find " + ngh + "\n");  
    }
```

```
        msg1 = " You are close to " + f + " Head " + ND + " You will find " + ngh;  
    }  
  
    System.out.println("Main Category = " + cate);  
  
    }  
  
    spin = spin + 1;  
  
    }  
  
    f = ngh;  
  
    System.out.println("print me");  
  
    if (vau == 0 || vau2 == 0) {  
        ResultSet qt = st.executeQuery("select latitude, longitude from " + val4 + " where ID = 1");  
        while (qt.next()) {  
            System.out.println(qt.getDouble(1) + "\t" + qt.getDouble(2));  
            double p = qt.getDouble(1);  
            vau = p;  
            double q = qt.getDouble(2);  
            vau2 = q;  
        }  
    }  
}
```

```
System.out.println("lat,lon,val" + value + value2 + val4);
```

```
ResultSet rt = st.executeQuery("select latitude, longitude, name, address from " + f + " where  
category = " + cate + "");
```

```
while (rt.next()) {
```

```
    System.out.println(rt.getString(3) + "\t" + rt.getString(4));
```

```
    String s = rt.getString(3);
```

```
    ss = s;
```

```
    String y = rt.getString(4);
```

```
    yy = y;
```

```
    double c = rt.getDouble(1);
```

```
    g = c;
```

```
    double x = rt.getDouble(2);
```

```
    w = x;
```

```
    GetDis dis = new GetDis();
```

```
    z = dis.Distance(vau, vau2, g, w);
```

```
    // ResultTextArea.append("\n" + ss + "\n" + "At " + yy + "\n" + "Total Distance is: " + z + "  
KM" + "\n");
```

```
    msg1 += " " + ss + " At " + yy + " Distance is: " + z + " KM";
```

```
}
```



```
System.out.println("Last MSG is: " + msg1);  
// send msg1;  
SMS sms = new SMS();  
sms.send(val, val2, val3,msg1);  
}  
  
else // JOptionPane.showMessageDialog(null, "Search Not Found::Check your Given Value",  
"Error", JOptionPane.ERROR_MESSAGE);  
{  
msg2 = "Search not found,Check your given value";  
  
System.out.println("Not Found MSG is: " + msg2);  
  
// send msg2  
SMS sms = new SMS();  
sms.send(val, val2, val3,msg2);  
  
System.out.println("Search Not Found::Check your Given value");  
}  
  
st.close();  
con.close();
```

```
    } catch (SQLException s) {  
        System.out.println("SQL statement is not executed!");  
        // JOptionPane.showMessageDialog(null, "Search Not Found", "Error",  
        JOptionPane.ERROR_MESSAGE);  
    }  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
    }  
}
```

SeekIt class:

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
  
package msaccess;  
  
/**  
 *
```

```
* @author User
*/
import java.sql.*;
//import javax.swing.JOptionPane;

public class SeekIt {

    public String Blockname(double val, double val2, String val3,String user,String pass) {

        System.out.println("print the area table!");

        Connection con = null;
        PreparedStatement ste = null;

        String g = null;
        double min = 0.0;
        int a = 1;
        int b = 1;
        int d = 1;
        double lat2 = 0;
        double lon2 = 0;
        double dis = 0;
        String msg5 = "";
        String msg6 = "";
        try {
            Class.forName("com.mysql.jdbc.Driver");
```

```
con = DriverManager.getConnection("jdbc:mysql://localhost:3306/javasql", "root", "");

try {
    Statement st = con.createStatement();

    System.out.println("Latitude = " + val + ", Longitude = " + val2 + "." + "area " + val3);

    if (val != 0.0 && val2 != 0.0 && val3 == null) {

        // if (val != 0.0 && val2 != 0.0 && val3.equals("---")) {

            ResultSet res = st.executeQuery("select FLati, LLati, Block from Area where (FLati < " + val + "
and LLati > " + val + " or FLati = " + val + " or LLati = " + val + ") and (FLong < " + val2 + " and LLong > " + val2
+ " or FLong = " + val2 + " or LLong = " + val2 + ")");

            while (res.next()) {

                System.out.println(res.getString(1) + "\t" + res.getString(2) + "\t" + res.getString(3));

                String c = res.getString(3);

                g = c;

                System.out.println("am from logitude");

            }

            System.out.println("Block Name = " + g);

            if (g == null) {

                while (d < 6) {
```

```
ResultSet result = st.executeQuery("select FLati, FLong from Area where ID = " + a + "");
while (result.next()) {
    //System.out.println(result.getString(1) + "\t" + result.getString(2) + "\t" +
res.getString(3));

    double e = result.getDouble(1);
    lat2 = e;
    double f = result.getDouble(2);
    lon2 = f;
}

GetDis distance = new GetDis();
dis = distance.Distance(val, val2, lat2, lon2);

String sql = "update loc set total = " + dis + " where ID = " + b + "";
int updateCount = st.executeUpdate(sql);

a = a + 1;
b = b + 1;
d = d + 1;

}

ResultSet re = st.executeQuery("select min(total)from loc");
while (re.next()) {
    System.out.println(re.getDouble(1));
```

```
        double c = re.getDouble(1);  
        min = (min + c);  
    }  
  
    System.out.println("Minimum value is = " + min);  
  
    ResultSet rt = st.executeQuery("select Block from area where ID in (select ID from loc where  
total = (select min(total)from loc))");  
  
    while (rt.next()) {  
        System.out.println(rt.getString(1));  
        String x = rt.getString(1);  
        g = x;  
    }  
  
    System.out.println("Block Name = " + g);  
}  
}  
  
else if (val3 != null) {  
  
    //ResultSet r = st.executeQuery("select FLati, LLati, Block from Area where Block = "" + val3 +  
    "");  
  
    ste = con.prepareStatement("select FLati, LLati, Block from Area where Block = ?");  
    ste.setString(1, val3);  
    ResultSet r = ste.executeQuery();  
  
    while (r.next()) {
```

```
System.out.println(r.getString(1) + "\t" + r.getString(2) + "\t" + r.getString(3));
String c = r.getString(3);
g = c;
System.out.println("am from block name:");
}

System.out.println("Block Name1 = " + g);
} else {
    // JOptionPane.showMessageDialog(null, "Select Correct Area name or Give Your Correct
    account info but not both", "Error", JOptionPane.ERROR_MESSAGE);

    msg5 = "Give correct user id and password";
    System.out.println("msg5: "+msg5);
    // send msg5
    SMS sms = new SMS();
    sms.send(user, pass, num ,msg5);

}

// ste.close();
st.close();
con.close();

} catch (SQLException s) {
    System.out.println("Error from Block Table!");
```

```
        // JOptionPane.showMessageDialog(null, "No Area Found", "Error",
        JOptionPane.ERROR_MESSAGE);

        msg6 = "no area found";

        System.out.println("msg6: " + msg6);

    }

} catch (Exception e) {

    e.printStackTrace();

}

return g;

}

}
```

SearchIt class:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package msaccess;

.
/**
 *
 * @author User
 */
```



```
import java.sql.*;

//import javax.swing.JOptionPane;

public class SearchIt {

    public String Category(String f, String val4,String num,String user,String pass) {

        System.out.println("print the Category table!");

        Connection con = null;

        String cate = null;

        String msg7 = "";

        try {

            Class.forName("com.mysql.jdbc.Driver");

            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/javasql", "root", "");

            try {

                System.out.println("VAlue4 = " + val4);

                // Statement st = con.createStatement();

                PreparedStatement st = con.prepareStatement("select category from " + f + " where category =
?");

                st.setString(1, val4);

                // ResultSet rr = st.executeQuery("select category from "+f+" where category = '"+val4+"'");
```

```
ResultSet rr = st.executeQuery();

while (rr.next()) {
    String d = rr.getString(1);
    cate = d;
}

System.out.println("Category = " + cate);

st.close();
con.close();

} catch (SQLException s) {
    System.out.println("i am from category!");
    // JOptionPane.showMessageDialog(null, "No similar Category found in this Area::Select other
Area", "Error", JOptionPane.ERROR_MESSAGE);

    msg7 = "no similar category found";

    // send msg7

    SMS sms = new SMS();
    sms.send(user, pass, num,msg7);
```

```
        System.out.println("msg7: " + msg7);

    }

    } catch (Exception e) {
        e.printStackTrace();
    }

    return cate;

}

}

FindIt class:
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package msaccess;

/**
 *
 * @author User
 */
import java.sql.*;
```

```
//import javax.swing.JOptionPane;
```

```
public class FindIt {
```

```
    public int Neighbour(String f) {
```

```
        System.out.println("print the Neighbour table!");
```

```
        Connection con = null;
```

```
        int digit = 0;
```

```
        String msg8 = "";
```

```
        try{
```

```
            Class.forName("com.mysql.jdbc.Driver");
```

```
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/javasql", "root", "");
```

```
            try{
```

```
                Statement st = con.createStatement();
```

```
                ResultSet rr = st.executeQuery("select count(distinct neighbour) from "+f+"");
```

```
                while (rr.next()){
```

```
                    int d = rr.getInt(1);
```

```
                    digit = d;
```

```
                }
```

```
                System.out.println("Neighbour Block = "+digit);
```

```
        st.close();
        con.close();

    }

    catch (SQLException s){

        System.out.println("i am from find!");

        // JOptionPane.showMessageDialog(null, "No Nearby Match Found", "Error",
        JOptionPane.ERROR_MESSAGE);

        msg8 = "no nearby match found";

    }

}

catch (Exception e){

    e.printStackTrace();

}

return digit;

}

}

GetDis class:
/*
```

* To change this template, choose Tools | Templates

* and open the template in the editor.

*/

```
package msaccess;
```

```
/**
```

```
 *
```

```
 * @author User
```

```
 */
```

```
import static java.lang.Math.*;
```

```
public class GetDis {
```

```
    public double Distance(double val, double val2, double val3, double val4) {
```

```
        double lat1 = val;
```

```
        double lat2 = val3;
```

```
        double lon1 = val2;
```

```
        double lon2 = val4;
```

```
        double R = 6371; // km
```

```
        double dlat = Math.toRadians(lat1 - lat2);
```

```
        double dLat = abs(dlat);
```

```
        double ddlon = Math.toRadians(lon1 - lon2);
```

```

        double dLon = abs(ddlon);

        double a = Math.sin(dLat / 2) * Math.sin(dLat / 2) + Math.cos(Math.toRadians(lat1)) *
        Math.cos(Math.toRadians(lat2)) * Math.sin(dLon / 2) * Math.sin(dLon / 2);

        double h = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

        double d = R * h;

        double z = Math.round(d);

        //System.out.println("Distance is = "+d);

        System.out.println("Round distance = " + z);

        return z;

    }
}

```

SMS class:

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package msaccess;

/**
 *
 * @author User
 */
public class SMS {

    public void send(String user, String pass, String num, String msg){

```

```
String registrationID = user;

String password = pass;

String fromMSISDN = num;

String toMSISDN = num;

String smsPort = "5001";

double charge = 0.00;

String chargedParty = num;

int msgType = 4;

String contentArea = "test";

String msgcontent = msg;

try {

    com.grameenphone.playground.WebServiceGPADPBizTalkSMSOrchestrations service = new
    com.grameenphone.playground.WebServiceGPADPBizTalkSMSOrchestrations ();

    com.grameenphone.playground.WebServiceGPADPBizTalkSMSOrchestrationsSoap port =
    service.getWebServiceGPADPBizTalkSMSOrchestrationsSoap();

    com.grameenphone.sendsmsrequest_xml.SendSMSRequest sendSMSRequest = new
    com.grameenphone.sendsmsrequest_xml.SendSMSRequest();

    sendSMSRequest.setRegistrationID (registrationID);

    sendSMSRequest.setPassword (password);

    sendSMSRequest.setSourceMsisdn (fromMSISDN);

    sendSMSRequest.setDestinationMsisdn (toMSISDN);

    sendSMSRequest.setSmsPort (smsPort);

    sendSMSRequest.setMsgType (msgType);

    sendSMSRequest.setCharge (charge);
```



```
sendSMSRequest.setChargedParty (chargedParty);  
sendSMSRequest.setContentArea (contentArea);  
sendSMSRequest.setMsgContent (msgcontent);  
  
//process result here  
  
sendsmsresponse_xml.schemas.biztalk.adp.gp.SendSMSResponse result = port.sendSMS  
(sendSMSRequest);  
  
System.out.println ("Result = "+ result.getMsgID() + " " + result.getStatus());  
} catch (Exception ex) {  
ex.printStackTrace ();  
}  
}  
}
```