

# **A WAP BROWSER-BASED GIS APPLICATION FOR MOBILE HANDHELD DEVICES ALONG WITH SWING-BASED CLIENT TRACKING SYSTEM**

## **TECHNICAL REPORT**

Centre for Mobile Computing and Communication, Jadavpur University  
Kolkata-700098.



**ADWAY MITRA , ARPAN ROY,ARIJIT KHAN AND PRADIP K. DAS.**

## CONTENTS

1. INTRODUCTION.....	3
2. RELATED WORKS.....	3
3. DESCRIPTION OF APPLICATION AS SEEN BY END-USER.....	4
4. DIAGRAMS.....	5
5. SOFTWARE IMPLEMENTATION DETAILS.....	7
6. CONCLUSIONS.....	12
7. REFERENCES.....	12

## **I. INTRODUCTION**

Geographical Information System is an emerging field of study. Such systems are used to collect, model, store and analyze information describing physical properties of the geographical world. GIS data can be both spatial and non-spatial. In this work, one of our aims is to provide spatial GIS data to users of GPRS and GPS-enabled mobile phones, as and when requested by them. It is particularly important for a person who has come to a new place and has no idea about the location of different places of interest and amenities available there. On running this application, the user of the handheld device is to select what type of facility he wants to get information about (like Educational Institute, Medical Center, Shopping Center, and Terminal for Railways, Buses etc). For this we employ a distributed (server-client) architecture, where a central server is used to store all the relevant information in a database, and vend them out to users on request. This is accomplished using Java Servlets and GPRS connectivity for mobile devices. Section II provides a survey of all the related works. Section IIIA describes the application in detail, while the software implementation is provided in Section IVA. Another part of the work described here allows the server administrator to track the movements of clients who continuously use the service mentioned above. This tracking can be done either in real-time or as a simulation, based on past records. The tracking is done on maps downloaded dynamically from Yahoo! Map Image API. The application is described in detail in Section IIIB, and the software implementation discussed in Section IVB.

## **II. RELATED WORKS**

An instance of one of the earliest non-cartographic GIS inspired system is Air Quality Information Dissemination-APNEE [7] in Europe via SMS or WAP based services in mobiles. Closer to home, at CMCC (JU) an SMS based locational service [4] was developed for assisting mobile users. It is a 2-tier server-client architecture with a 'pull' based client service. It works only in a PDA with a specific version of operating systems for mobiles (Windows Mobile 2003 in this case). Server-client exchange of metadata is mediated by the Short Message System (SMS). The frontend or the client side is a user interface coded in Embedded Visual C++. This interface uses the inbuilt GPS tracking infrastructure in PDAs (namely HP iPAQ) to create an SMS and send it to the server. The back end or the server side contains a database server with data of the GSM addresses of mobile devices requesting service and a

second database for spatial data. A Java application in the server polls continuously to detect SMS from the client. Once the SMS containing the GPS data reaches the server, the application parses the String of locational data to extract the latitude and longitude. It then evokes the database query class which compares the latitude and longitude with the database entries to find the location nearest to it and the resultant data is sent back to the client in the form of an SMS.

### **III. DESCRIPTION OF THE APPLICATION AS SEEN BY END-USER**

#### **A. The GIS service for Mobile Handheld Devices**

The GIS facility can be used by a user who has subscribed to use this service called WIDGIS (Wireless Internet-based Distributed Geographical Information System). The user's handheld device must have GPRS connectivity, and also GPS signal-tracking software. These facilities are available in HP iPAQ, Nokia N-70 (and some higher models), Blackberry and several other devices. When the user logs on to the GIS website, he is prompted to enter his current latitude and longitude, (which can be known from the GPS-tracking capability of the handheld device), and also what facility he is looking for. The page asks for a username (usually the mobile number) and password. When these data are submitted, a module running on the server checks the {username, password} pair against a table containing such pairs for all the subscribed users. If a match is found, the Servlet processes the data submitted, queries the database, and returns stored information about the location of the user's choice that is closest to him. For example, if the user had inquired about railway station, he is informed about the nearest railway station. The information is written by the Servlet by writing into an HTML page. The page also contains two links- one that points to a Yahoo! Map highlighting the site that matches the user's choice, and the other pointing to another Yahoo! Map highlighting the site about which the user is being informed. On the other hand, if the subscription module finds that the person attempting to use the service is not a valid subscriber, it displays another page (in HTML) prompting the user to enter his details and credit card number to become a subscriber.

#### **B. The Client-tracking Application**

The other part of the application is a client-tracking system rooted at the server. It is basically a Java Swing application. As the client uses the GIS service, he posts the latitudes and longitudes of his positions at periodic intervals, which get recorded in a file at the server computer, along with the time at which these were received. There is also a database of all valid subscribers at the server end. Whenever a client subscribes, his username gets recorded in this database. Similarly, when the subscription period of a client expires, his username is automatically deleted from the database. The server administrator can also manually add or remove clients from this list by opening a Server Administrator Module. To track the movements of clients, the administrator has to select the clients (at most 4 in the pilot system) that he wants to track simultaneously. Tracking can be either in High Resolution Mode or in Natural Resolution

mode. In Natural Resolution Mode, the software downloads a Yahoo! Map highlighting the position of the client, and blocks till he sends another {latitude, longitude}, for which another appropriate Yahoo! Map is downloaded if required. In the High Resolution Mode, the software traces out the path in greater details, by interpolating the intermediate points and highlighting them in the map at regular intervals. This mode is more appropriate when the tracking is not done in real-time, but by a simulation based on past data.

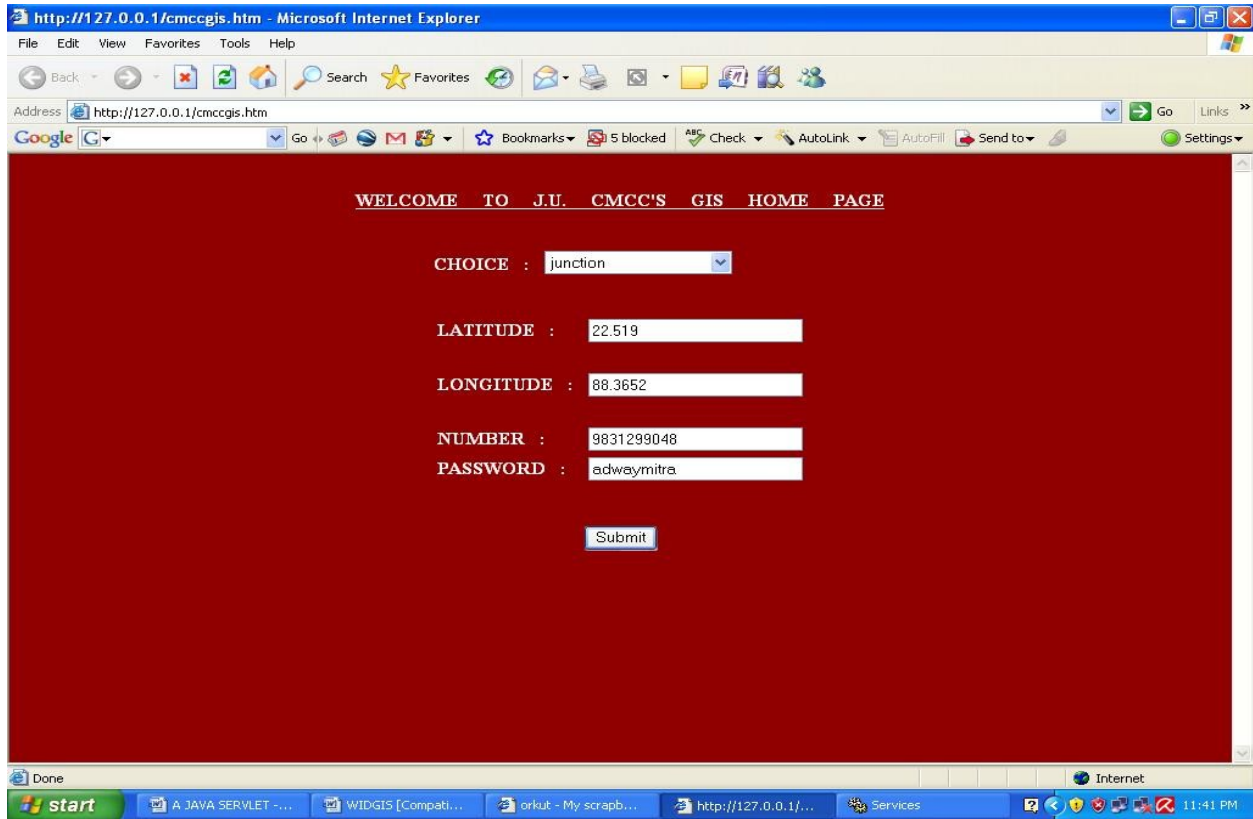


Figure 1: The GIS data submission page

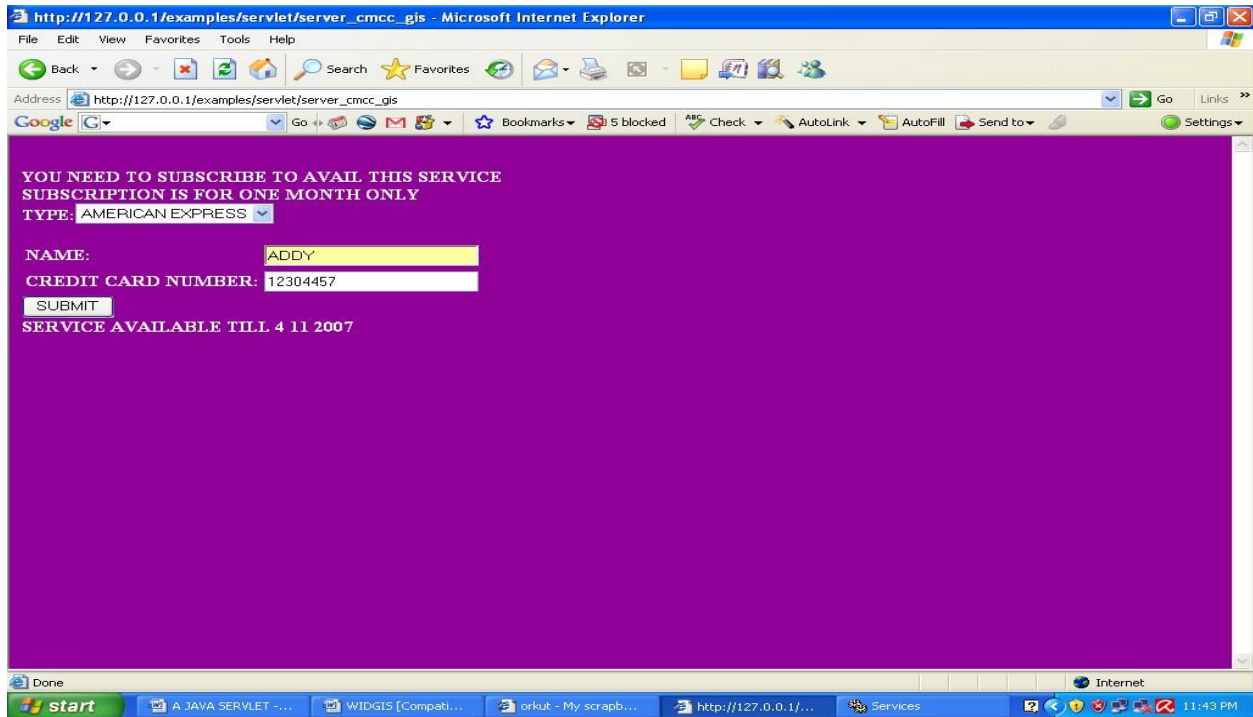


Figure 2: The Subscription Form

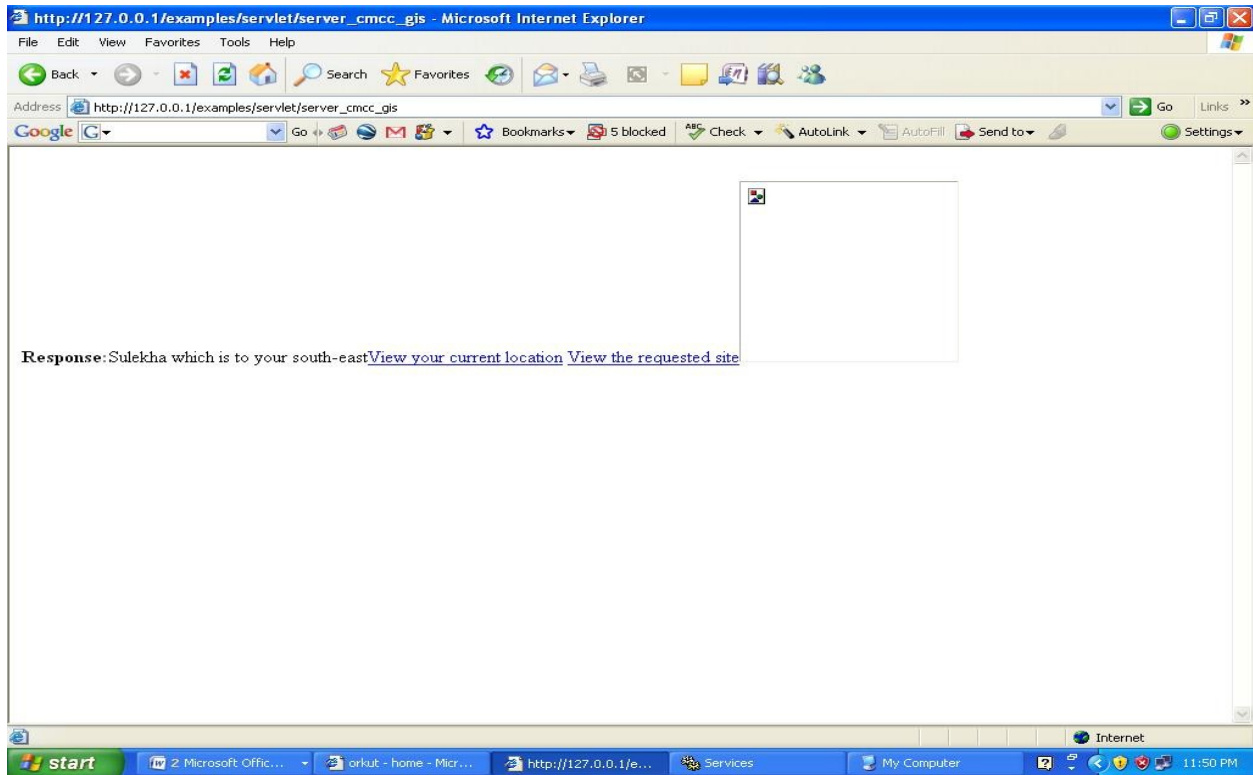


Figure 3: The Response Page

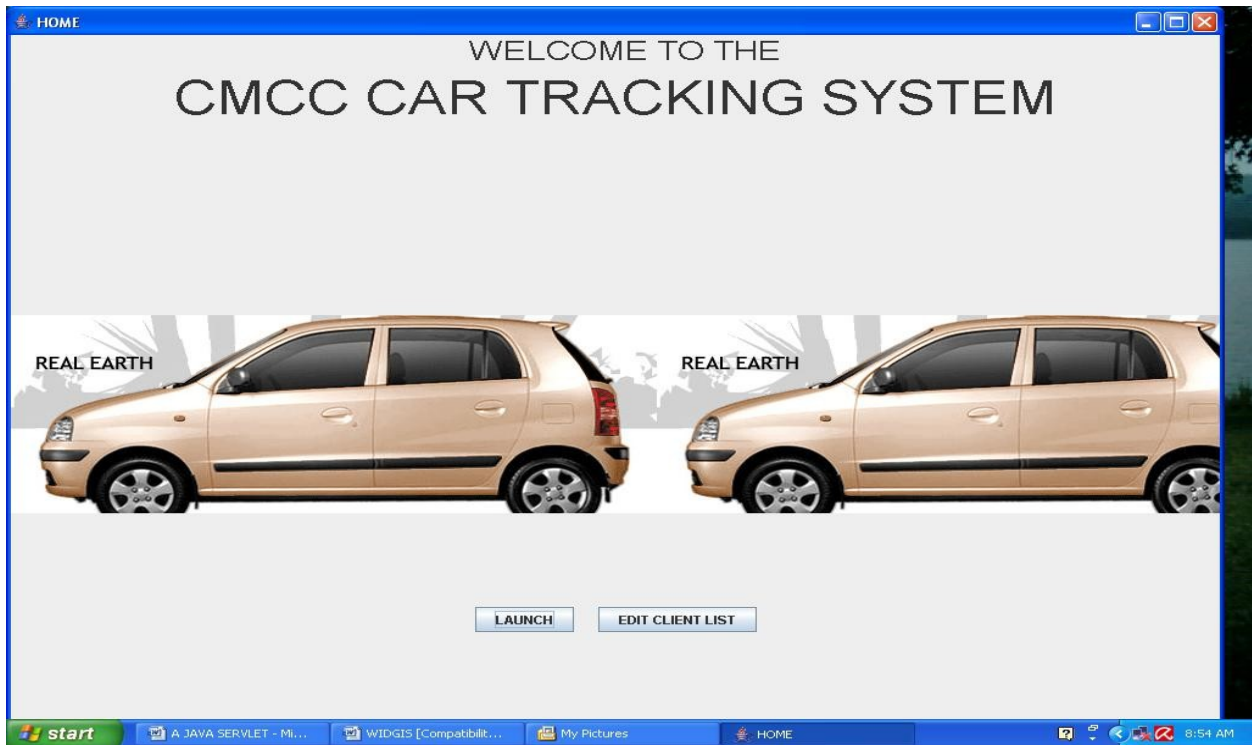


Figure 4: The Client-Tracking System Home Page

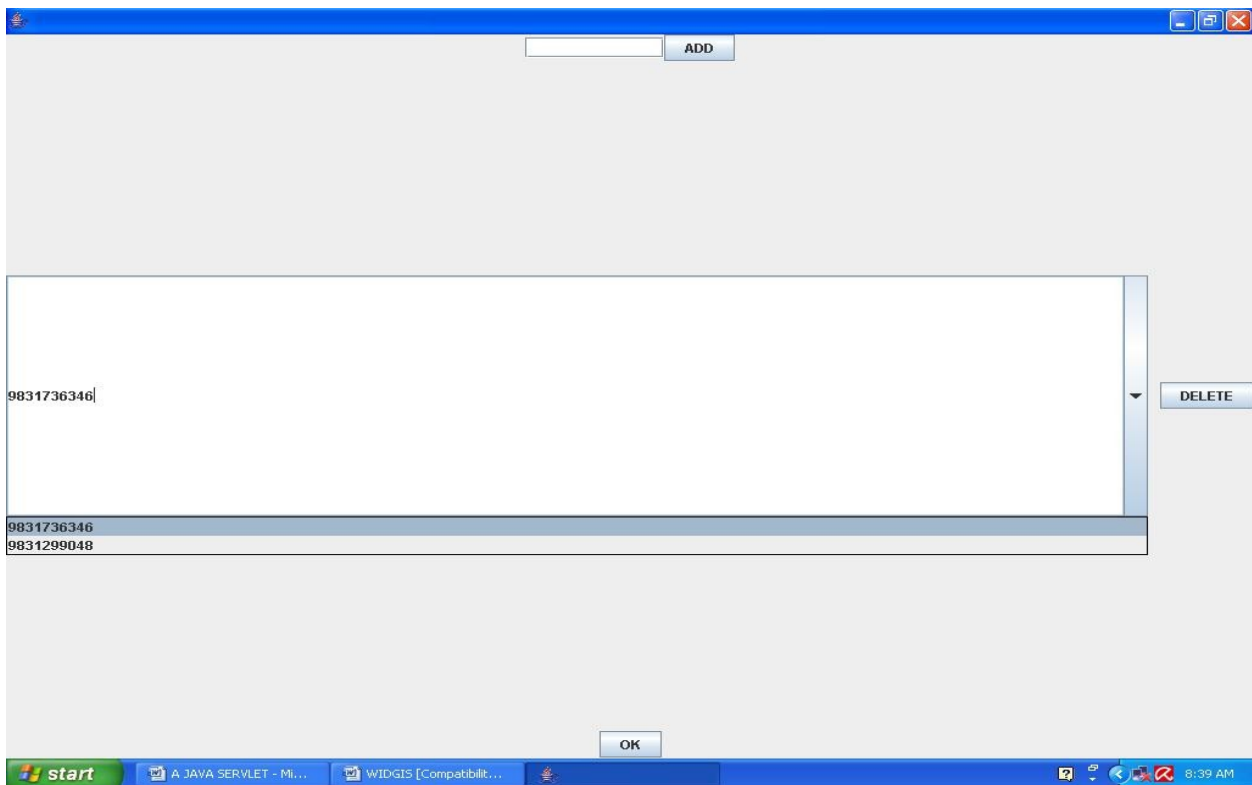


Figure 5: The Server Administrative Module

## IV. SOFTWARE IMPLEMENTATION DETAILS

### A. WIDGIS SOFTWARE

The Location Based Textual Information System (LBTIS) comprises two main modules: -

1) **Client-Server Data Pull Module (CSDPM)**:- The primary module in the LBTIS is the 'Pull' [3] based half-duplex data flow module. It has two sub modules, the data sending module and data receiving module.

i) **The Client side**: - It consists of a markup language based client interface. Pages in HTML (in case of GPRS-enabled HP Blackberry, N-70, Nokia 6600 or laptops using a standard data card for wireless Internet connectivity) or WML (in case of lower order mobiles) are opened by loading the static IP address (or Domain Name) in the micro browser of the client's mobile device. The client can request any location-based information the server offers to provide, and submit spatial data in form of the user's current latitude and longitude. Then the 'doPost' method in the Servlet deployed in the server is invoked with the client's spatial data as arguments. The Servlet then dynamically generates a HTML page containing the requested information, and returns it to the client. It is displayed by the micro browser of the client's handheld device. A section of the source code from the Servlet's 'doPost' method is presented to demonstrate the creation of markup language page as a portion of the response.

```
public void doPost(HttpServletRequest request, HttpServletResponse response ) throws ServletException, IOException{

    response.setContentType("text/html"); //or text/wml

    PrintWriter pw=response.getWriter();

    .

    .

    .

    pw.print("<html><body><bodybgcolor=#FFCFDA'><left>"+<formname=\"Form1\"\"+\"action=\"http://210.212.30.245/\"+\"examples
    /servlet/server_cmcc_gis\">");

    pw.print("<BR><B>Response:</B>"+loc+\" \" +dir);

    pw.print("<A HREF=\"+url+\">View your current location</A>");

    pw.print("<A>    </A>");

    pw.print("<A HREF=\"+url2+\">View the requested site</A>");

    pw.println("</form></body></html>");

    pw.close();

}
```

ii) **The Server side:** - This is a system deployed initially in an improvised server and later a server with global IP (IBM – p690 64-bit Regatta server in Jadavpur University in our case). Here the Servlets are deployed in Apache Tomcat 4.1 in JDK 1.6 environment. It has three HttpServlets. One of them is the GIS Servlet which deals with the receiving and sending of locational information. The other two are registration Servlets required by the client registration and login module. The connection uses HTTP, and so the Servlets are HttpServlets. The server has access to databases (designed using MySQL). It forms queries using the client's spatial data and connects to the database. The database replies with suitable textual information. The Servlet then dynamically generates the response page using the retrieved information. Database query classes are coded in Java making use of the JDBC Drivers and JDBC API.

```
Class.forName("org.gjt.mm.mysql.Driver");

Db=DriverManager.getConnection("jdbc:mysql://localhost/gis1","root","root");

String query="select * from landmarks";

DataRequest=Db.createStatement();

results=DataRequest.executeQuery(query);
```

The above code source fragment shows how a landmark nearest to the client's position is extracted from database table of the same name and sent to the client. The locational information in response consists of the name of the landmark of the type requested by the client that is nearest to him, the approximate direction in which it lies from the client's present location. All these are embedded into a single HTML page by the server. Also, there is a file where the server stores every client request. Each record consists of the following field: 1) Latitude 2) Longitude 3) ID of the client 4) date of request and 5) Time of request. These records are useful for the Server-rooted Client Tracker Application described later.

```
String s=" client="+num+" lat="+d[1]+" long="+d[0]+" date: "+dd+"/"+mm+"/"+yy+" time: "+hr+": "+min+": "+sec+" \n";

buf=s.getBytes();

OutputStream o=new FileOutputStream("C:/CMCC_TRACKER/latlong.java",true);

o.write(buf);

o.close();
```

The server also embeds in the reply HTML page links to Yahoo! Maps showing the client's position and the position of the site about which textual information is being returned. To get this URL, the server runs a module to construct a Yahoo! Map Image API command sentence that looks like this:

```
URL yahoo = new
URL("http://local.yahooapis.com/MapsService/V1/mapImage"+"?"+"appid=JLPWyvnV34FfmTBoFWRGZnhGI9Sm04PtjB0qe6vw
K3zBG91kRmCDqIq2J8rGSZYn1Es&latitude="+lat+"&longitude="+lon);

URLConnection yc = yahoo.openConnection();

BufferedReader in = new BufferedReader(new InputStreamReader(yc.getInputStream()));
```

From webpage pointed to by this URL, the URL of the appropriate map is obtained by parsing the string from the contents of the webpage. However, the servlet system installation succeeded in the Regatta server but query handling failed owing to an incompatible JDBC driver for the version of SuSE Linux used in the server.

II) **Client Registration & Sign-in Module (CRSM):-** The use of this GIS system is subscription-based. Hence a proper client identification protocol becomes imperative. WIDGIS accomplishes this by a username-password module. This authentication module starts functioning a) when unregistered client requests information from the GIS server or b) a registered user seeks locational service. If the user is new to WIDGIS a registration page opens up prompting the client to submit a username, credit card information (for registration) and a password of his choice. When an existing user logs into the system he is prompted for his username/mobile number which is then authenticated. If the authentication is successful, the user is taken into the CSDPM otherwise the system displays an exit message. The server side has a database of clients where client IDs in the form of their mobile numbers or username are stored. The CRSM module is thus designed to provide some degree of security to this GIS service.

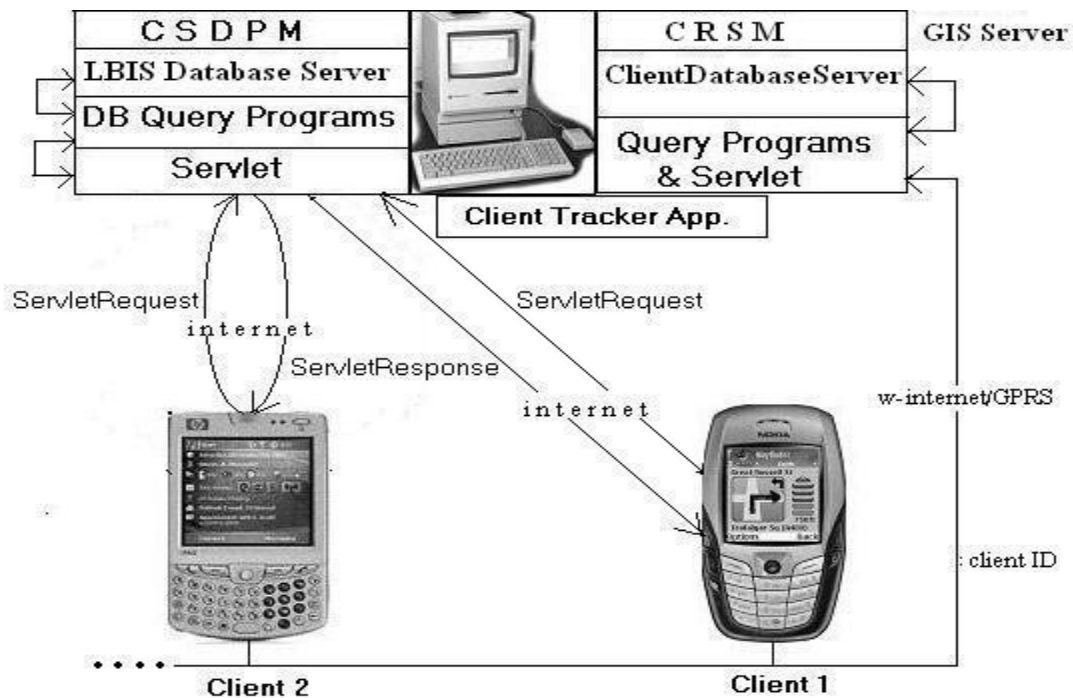


Figure 7: Architecture of LBTIS

## B. CLIENT-TRACKING SOFTWARE

This is a server-based application that performs the analysis of preconfigured spatial data in the server. It is concerned with tracking WIDGIS users on the basis of requests made by them to the WIDGIS server. This is primarily a simulation-based application where the actual path of a user is traced on the basis of spatial data provided by the client. The application has access to two-dimensional maps. The client is tracked on the basis of these scalable vector GIS data (SVG) [9] in the form of Yahoo! Map images. The software is composed of two modules viz *i) the display module and ii) the administrator module*. Both the modules are based on Java Swing API. It requires the support of javax.swing library and requires swing.jar file in JDK's lib.

The Display Module: When the application is launched in the display module, the operator is provided a list of all valid WIDGIS users. He can choose to track several users simultaneously, whose movements can be traced on the screen. Tracing can be either on-line or off-line. The latter uses a simulation on the basis of their movement records stored earlier. In either case, the application reads a record from the file (called latlong.java) and checks if the user associated with that record has been selected by the operator.

```
FileReader fr=new FileReader("latlong.java");  
  
BufferedReader br=new BufferedReader(fr);
```

A typical record stored in this file looks as follows:

```
client=9831736346 lat=22.5005 long=88.3685 date: 7/5/2007 time: 9:20:11
```

The client associated can be obtained by parsing each record. Until a record associated with one of the selected clients is found, successive records are parsed. If no more records are found, the process blocks, waiting for new information to come in from the clients.

```
String ip=lex.getClient(s);  
  
while(s!=null && client.equals(ip)==false && client1.equals(ip)==false && client2.equals(ip)==false && client3.equals(ip)==false)  
{  
  
    s=br.readLine();  
  
    ip=lex.getClient(s);  
  
}
```

If so, the application chooses the suitable map from its folder, loads it on the display page, and paints the animation showing the position of that particular user. Also, this can be done in either "NATURAL RESOLUTION" form or in "HIGH RESOLUTION" form. In the first case, the point indicated by the record (by latitude and longitude) will be animated on the suitable map. In the second case, however, the application simulates the movement of the user in greater detail, using interpolation to trace out the route between the last two query points for that user. It may so happen that those two points are not connected by a straight road. Then, to trace out the path, we also need to store information about roads in that region. As of now, we have recorded the details of three major roads in our testing region viz.- The Raja SC Mullick Road, The Eastern Metropolitan Bypass and The Rajarhat New-town Connector. The details

include the list of points (latitude, longitude) on that road, between any two of which the road may be considered a straight line. Our application, when running in the “HIGH RESOLUTION” mode, checks if the last two query points of a user are on any of the specified roads. If so, the interpolation points are animated along that road, taking into account all the “sharp turns” (i.e.- the animation is slow enough and the dots are well-paced to make out sharp turns along the road). In case no such match is found, however, the application simply animates the straight line between those two points. For animation, suitable images have to be downloaded from Yahoo! Maps. The Yahoo! Map Image API [11] command sentence is created as described in the WIDGIS section, and on obtaining the image URL, that animated image is displayed in the appletviewer.

ii) The Administrator Module: This module gives system administrator the option to manually delete or add any client into the client database.

## V. CONCLUSIONS

This type of client tracking will eventually find its use in location based tourism [10]. In fact, the LBTS (Location Based Tourism System) discussed in [6] with GPS based positional tracking is the brainchild of this LBTIS module in WIDGIS. LBIS is also one of the practical offshoots of the specific types of LBS proposed in [5]. Data delivery can be of two distinct types: - ‘push’ or ‘pull’ [1]. Using push, data items are sent out to the clients without explicit requests. In contrast, with pull data items are broadcast in response to explicit requests received from clients. Pull can be used either for unicast or for broadcast. When used for broadcast, pull is also referred to as interactive, or on-demand broadcast. The WIDGIS architecture makes use of both the concepts simultaneously. The client “pulls” information from the server by making explicit requests with his spatial information, but the server-based tracking application views this information as a “push”, since it uses the same information to simulate the movements of the client.

## REFERENCES

- [1] Demet Aksoy & Mason Sin-Fai Leung "Pull vs Push: A Quantitative Comparison for Data Broadcast" in GlobeCom 2004, IEEE Communication Society.
- [2] Jihong Guan, Shuigeng Zhou, Fuling Bian and Aoying Zhou "Mobile Agent-Based Distributed Geographic Information Systems under Internet Environment" in Database and Expert Systems Applications, 2000.
- [3] Juhong Guan, Shuigeng Zhou and Fuling Bian " A Mobile-Agent and GML Based Framework for Integrating Distributed GIS" IAPRS Volume XXXIV, Part-2, Commission II, Xi'an, August 2002

- [4] Kostas Karatzas, Gertraud Peinel, Thomas Rose "GIS-based Dissemination of Air Quality Information in APNEE", Project APNEE (Air Pollution Network for Early warning and online information Exchange in Europe) 1999-2000
- [5] Manhoi Choy, Mei-Po Kwan, and Hong Va Leong "On Real- time Distributed Geographical Database Systems" Research Report no.21, UC Santa Barbara, December 1993
- [6] <http://www.cs.ucdavis.edu/~aksoy/PLASMA/>
- [7] Pradip K. Das, Niranjana Patra "Server-based locational management system for GPS enabled vehicles" Technical Report, 2007. Centre for Mobile Computing and Communication (CMCC), Jadavpur University.
- [8] Serge Boucher and Prof. Esteban Zimányi "Software Engineering Issues in Location Based Services" Université Libre de Bruxelles, Brussels, Bou Masters Thesis 2005
- [9] Wu Binzhuo and Xia Bin "Mobile Phone GIS based on Mobile SVG" in Geoscience and Remote Sensing Symposium, 2005.
- [10] [http://www.trimble.com/mgis\\_mobilegis.shtml](http://www.trimble.com/mgis_mobilegis.shtml)
- [11] <http://developer.yahoo.com/maps/rest/V1/mapImage.html>