

Automated Traffic Assistance System

Aryadip Sarkar

ESL Lab, Dumdum, Kolkata

Abstract: In this paper we have described the design and implementation concept of an automated traffic assistance system. In our project we are trying to recover the common problems that everyone faces most of the days in this congested society with heavy traffic movements.

We have tried to make the support system fully automated rather than the manual control. We thought to use of the Traffic Signal Lights as the server and the vehicles moving towards the signals as clients and clients can communicate with the server via wireless communication.

Clients can make queries like the route for a specific destination, or can ask for the alternative routes and also the shortest route, as the result client can get the (route, direction as well as the distance), also they can see the map (2D and 3D). They can ask for nearest places like Hospital, ATM, Market, Shopping Mall, Food Store, Café etc. Emergency Support, is the another feature we have included, using this feature client can ask for help in any emergency situation like accident or when he or she is in trouble. Clients can get the emergency notifications or alerts from the server, by this they can be aware of the upcoming hazards in their ways.

As the wireless communication medium we have used Bluetooth.

I. Introduction

Our project is mainly divided into two parts: (I) Client Side and (II) Server Side. At the Client side (which is basically a Vehicle equipped with a device, generally an I/O system like Tablet Comp. or PC having Bluetooth device), user can communicate with the Traffic Signals (basically which is the server station). Here user can send request to the server by asking for the routes of a destination along with the shortest path and map, By the Emergency Support feature the user can send any emergency alert message to the server, here we provide some category of places by which user can search for nearby places like ATM, HOSPITAL, BANK etc. and user also can see the notifications and alerts from the server. In this project we have used various technologies and programming languages:

- Bluetooth communication.
- MYSQL, JDBC connection.
- JSP
- Python
- Inbuilt modules for python programming language:
 - a) 'Bluetooth'
 - b) 'time'
 - c) 'matplotlib'
- VPython
 - a) 'visual'

1.1 Bluetooth communication: Bluetooth is a low power consuming Radio Frequency communication system. This is the medium of communication between the client and the server.

1.2 MYSQL, JDBC connection: MySQL ("My S-Q-L" officially, but also called "My Sequel") is (as of March 2014) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius's daughter, My. The SQL phrase stands for Structured Query Language. MySQL is a popular choice of database for use in web applications, and we used WAMP (Windows, Apache, MySQL, PHP/Perl/Python) to maintain our databases.

JDBC is a Java-based data access technology (Java Standard Edition platform) from Oracle Corporation. This technology is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the JVM host environment. We used JDBC connection to access our database from JSP.

1.3 JSP: Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. JSP is similar to PHP, but it uses the Java programming language. To deploy and run Java Server Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required. We developed our pages using JSP.

1.4 Python: A widely used general purpose high-level language. Its design philosophy emphasizes code readability and its syntax allow programmers to express concepts in fewer lines of code than would be possible in languages such as C. Python supports multiple programming paradigms, including OOP and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large comprehensive standard library.

1.5 'Bluetooth': This is a module under python programming language which is use to find and manipulate the Bluetooth devices. We use this module to discover the Mac Ids of server and clients.

1.6 'time': This is a module under python programming language which gives access to the current system time. We used this module here mainly to synchronize the server and client communication process.

1.7 'matplotlib': This is a module under python programming language which is use to plot the x, y co-ordinate values to generate 2D map view.

1.8 VPython: VPython is the Python programming language plus a 3D graphics module called Visual. VPython allows users to create objects such as spheres and cones in 3D space and displays these objects in a window.

'visual': This is a module under VPython programming language which is use to plot the x, y and z co-ordinate values to generate 3D map view. We use this module to generate the 3D map view of routes.

II. Motivation

Today the harassment in traffic is the most common problem in our everyday life, it creates so many difficulties on our ways.

So, we got inspired to design this application for better assessment of traffic. In this project we are trying to support the Traffic Control System by providing automated features to make it interactive to peoples. We included various automation features inspired from the problems like:

- A person wants to know the route or the shortest route for a particular destination, the person needs to ask the traffic police or some other persons, sometimes it creates difficulty during travel or the person will be in trouble if someone is given any wrong information.
- A person is in emergency situation like accident or stuck in his way and ask for immediate support or to find the alternative route.
- A person want find nearby Market, Hospital, Park, Petrol Pump, ATM etc.
- If a car theft, the investigators faces so much difficulties to find it.
- It is difficult to know for a person of the upcoming hazards like road blocks or extensive traffic jams on their way.

We have tried to provide solutions for all the above problems in our application.

Instead of internet connection we thought of using the Bluetooth for communication. It may be faster and affordable way for the people to install in their car. Every person traveling through the traffic signals will be benefited by all the features, in most cases an Ambulance in an emergency situation will be benefited by finding alternative routes or shortest route and by the emergency support. Moreover, Bluetooth is a low power consuming Radio Frequency communication system. Thus, it gives us benefit overall.

III. Methodology

In the following section we are going to describe the methodology of the project we have done.

3.1 Design of Client Side: At Client side the working of this system can be divided into two parts: a) Listen mode b) Client mode.

At first user can start the service by clicking the START button on the client GUI page (basically a web page designed using JSP). After the system started, at first the client system will be in Listen mode.

- **Client mode:** When user will request for any service by clicking the respective buttons, the system will immediately shift into the Client mode to send request to the server.

In this mode user can use the following services:

- Emergency Support:** In this section by clicking the Emergency Support button user can send the emergency request message to the server.
 - Path search:** In the Path Search section user can request for all paths including the shortest path and Map view for a specific destination.
 - Nearby Place Search:** In this section user can request direction and distance for nearby places by clicking the respective buttons labeled by the place names.
- **Listen mode:** After send any request to the server the Client system will automatically shift into the Listen mode. In this mode user can receive any acknowledge message and respective results of the requests, from server. When the system will be in listen mode, will automatically receive the notifications from the server and will be display in the notification area of the client GUI page.

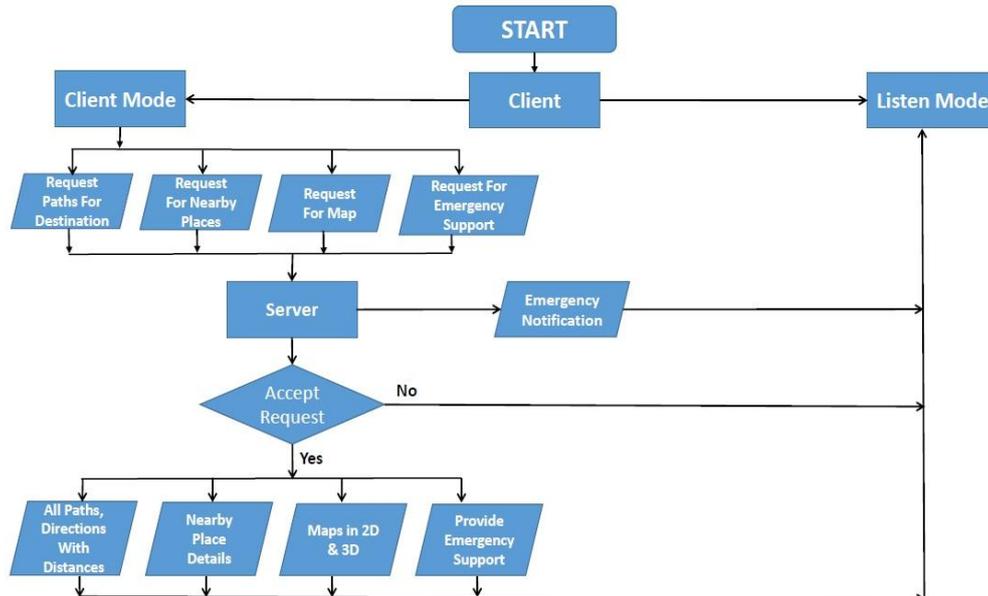


Fig. 1: Flow diagram of Client side system.

3.2 Design of Server Side: At server side the design of the system can be divided into two parts: a) Listen mode b) Client mode.

- **Listen mode:** Initially the server will start in Listen mode. In this mode server can receive all the requests and messages from the Clients. After receiving requested data from Clients, will be recognized by the internal process and result based on the type of Client’s requests will be generated.
- **Client mode:** After generating the respective results, the server system will immediately be shift into Client mode to send result to specific clients by recognizing the mac-ids and immediately shift into the Listen mode. In this mode Operators also can send notifications or alerts like ‘G.T road blocked, take the alternative root’ to clients. After send the notifications server system will automatically be shift into Listen mode.

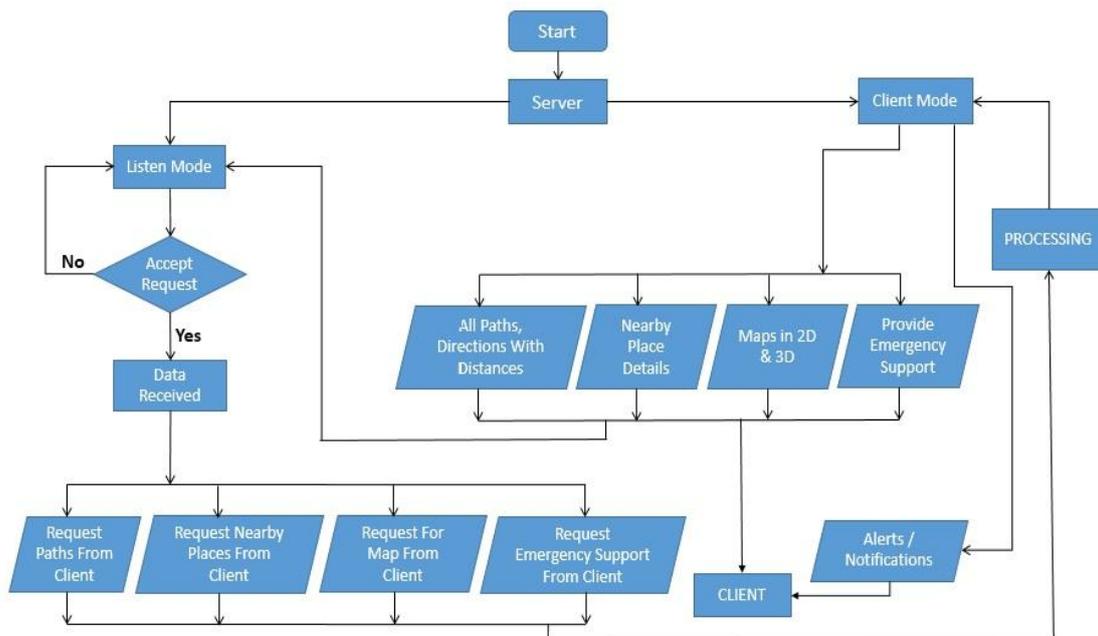


Fig.2:Flow diagram of server side system.

3.3 Example: Here we have described one the features (Nearby Place Search).

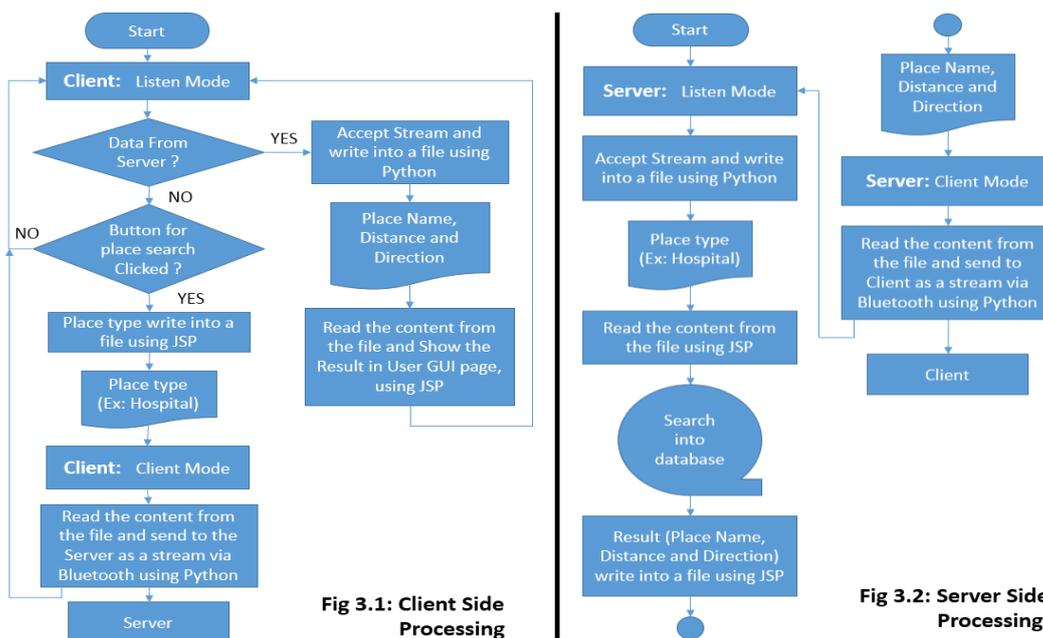


Fig 3.1: Client Side Processing

Fig 3.2: Server Side Processing

Fig.3:Flow diagram for Nearby Place Search.

IV. Algorithm

• Client Side Processing

Step 1: Start.

Step2: Mode=Client Mode or Mode=Listen Mode.

Step3: If Mode='Client Mode' then go to**Step4**.Else go to**Step25**.

Step4: Service='Request Path for Destination' or
Service='Request for Nearby Places' or
Service='Request for Map' or
Service='Request for Emergency Support'.

Step5: If Service='Request for Emergency Support' then go to **Step6**.
Else go to **Step 10**.

Step6: Waiting for Server Acceptance.

Step7: If Server Accept request successfully then go to **Step8**.
Else go to**Step9**.

Step8: Receive the acknowledgement Message & go to **Step 25**.

Step 9:Go to**Step 25**.

Step10: If Service='Request Path for Destination' then go to **Step11**.
Else go to**Step 15**.

Step 11: Waiting for Server Acceptance.

Step 12: If Server Accept request successfully then go to **Step 13**.
Else go to **Step14**.

Step 13: Receive all Paths, Directions with Distances from Server& got to **Step25**.

Step14:Go to**Step 25**.

Step15: If Service='Request for Nearby Places' then go to **Step 16**
Else go to**Step 20**

Step 16: Waiting for Server Acceptance

Step17: If Server Accept request successfully then go to **Step 18**
Else go to **Step 19**

Step 18: View nearby places details with direction & go to **Step25**

Step 19:Go to**Step 25**

Step 20:If Service='Request for Map view' then go to **Step 21**
Else go to**Step 25**

Step 21: Waiting for Server Acceptance

Step 22: if Server Accept request successfully then go to **Step 23**
Else go to **Step 24**

Step 23: view 2D & 3D map & go to **Step 25**

Step 24: Go to **Step25**

Step 25: Mode=Listen Mode & waiting for notifications or results.

Step26: Go to **Step2**

- **Server Side Processing**

Step 1: Start

Step 2: Mode=Listen Mode or Mode=Client Mode

Step 3: If Mode=Listen then go to **Step 4**

Else go to step

Step 4: If request accepted then go to **Step5.**

Else go to **Step3.**

Step5: Receive the data.

Step6: If data ='Request for path' then go to **Step7.**

Else go to **Step 11.**

Step 7: Processing for required path.

Step 8: Generate the output for all path, direction, and distance with shortest path.

Step 9: Switch to client mode.

Step 10: Send the output to client and go to **Step 2.**

Step 11:If data ='Request for nearby place' then go to **Step 12.**

Else go to **Step 16.**

Step12: Process the request.

Step 13: Generate the output for nearby places details.

Step 14: Switch to client mode.

Step 15: Send the output to client and go to **Step 2.**

Step 16: If data ='Request for maps' then go to **Step 17.**

Else go to **Step 22.**

Step18: Processing for this map view.

Step19: Generate the output for 2D & 3D map.

Step 20: Switch to client mode.

Step 21: Send the output to client and go to **Step 2.**

Step 22: If data ='Request for emergency support' then go to **Step 23.**

Else go to **Step 2.**

Step 23: Switch to client mode.

Step 24: Send the acknowledgements to client and go to **Step 2.**

Step 25: If mode='Client mode' then go to **Step26.**

Step 26: Send the alert or notification to client & go to **Step 2.**

V. Results

Here goes some of the screen shots of our project which shows the working of our project.

Fig: 4 shows the client GUI page where in the Notification Area user got the Alert Message from the Server.

Fig: 5 shows that user sending the Emergency Help Request to the Server.

Fig: 6 shows the 'Nearby Place Menu' where user can search for verious types of places on need.

Fig: 7 shows the Result of search for the place Hospital.

Fig: 8 shows the Map View of the Paths search by the user.

Fig: 9 shows the Server receive the Emergency message from the Client.

Fig: 10 shows that from Server Emergency Alert message send to the Clients.

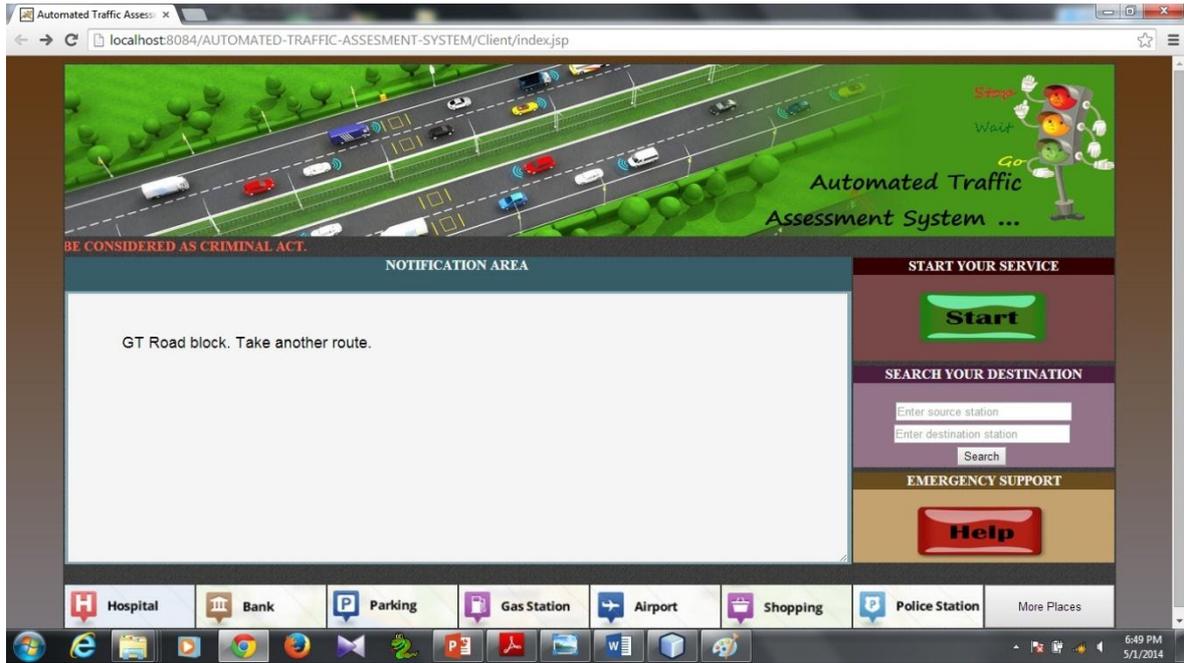


Fig 4: Client got alert from Server.

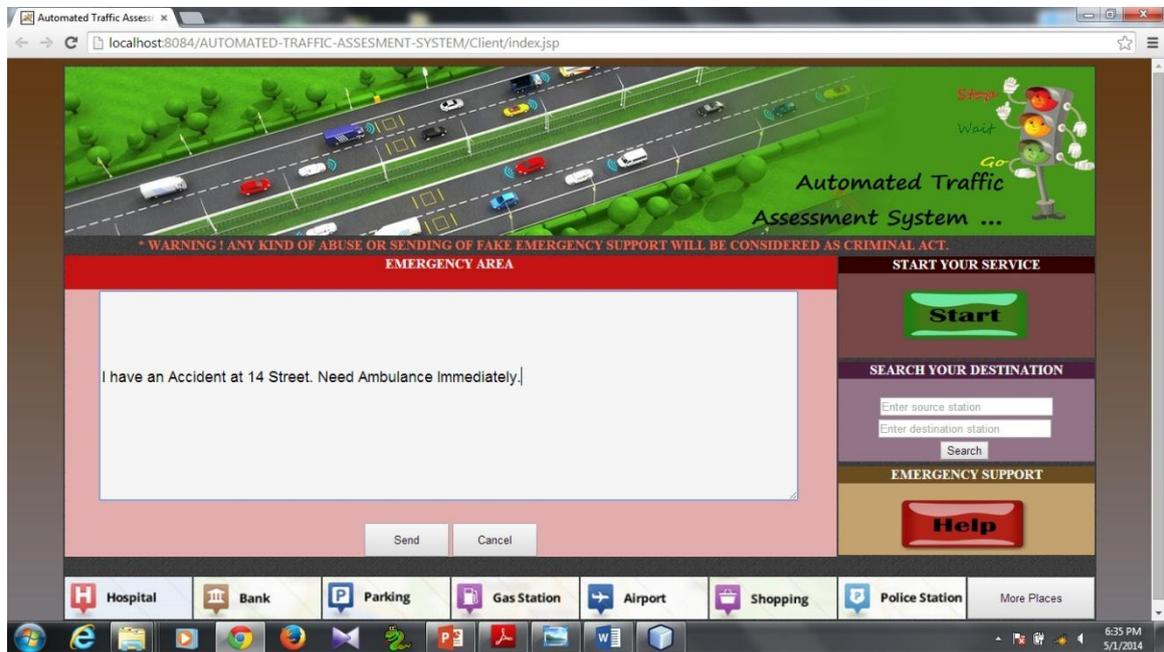


Fig 5: Client Sending Emergency Message to Server.

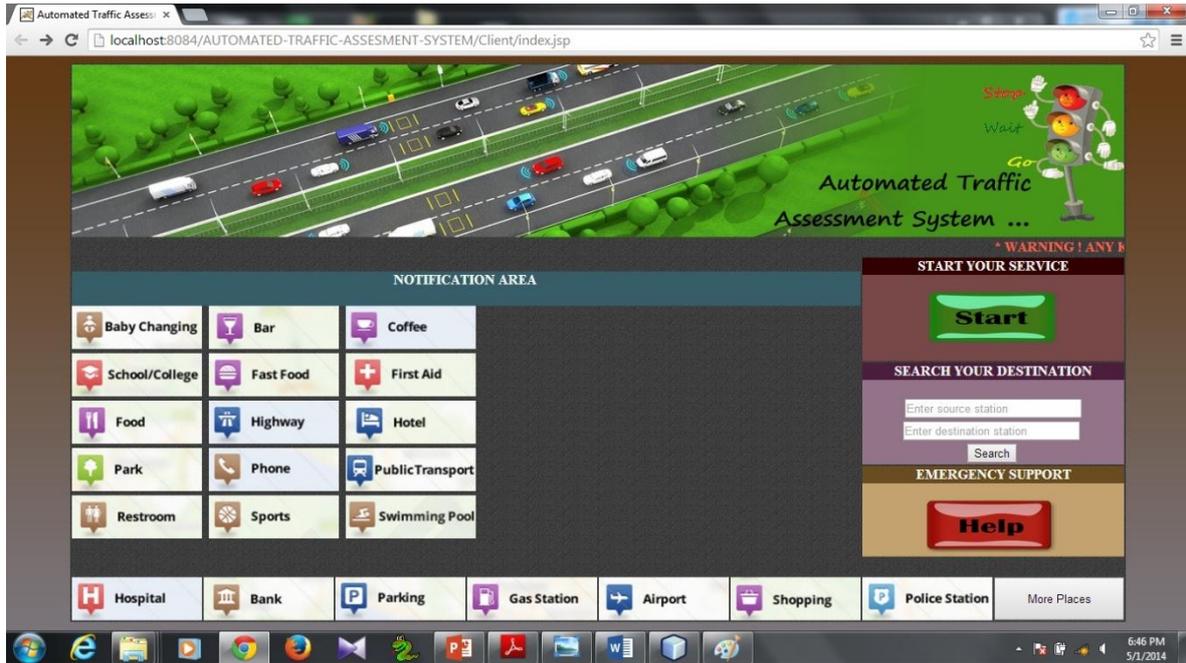


Fig 6: Place Search Menu.



Fig 7: Place Search Result from Server.

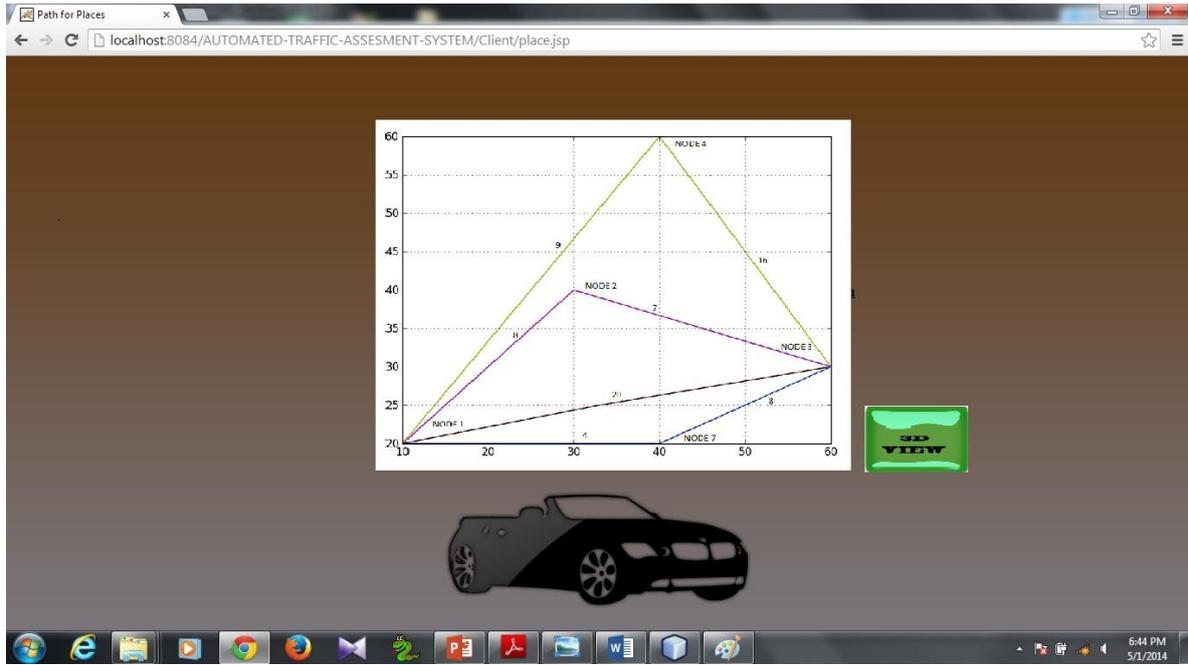


Fig 8: Map View result for Path Search.

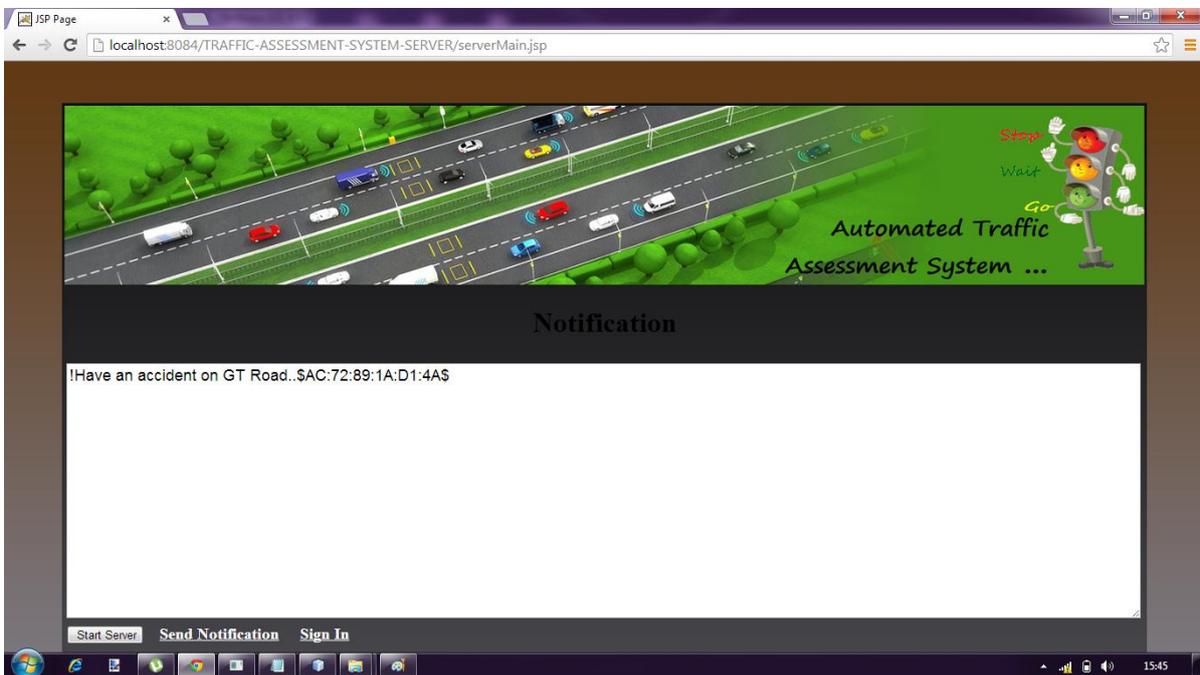


Fig 9: Server Received Emergency Message from Client.

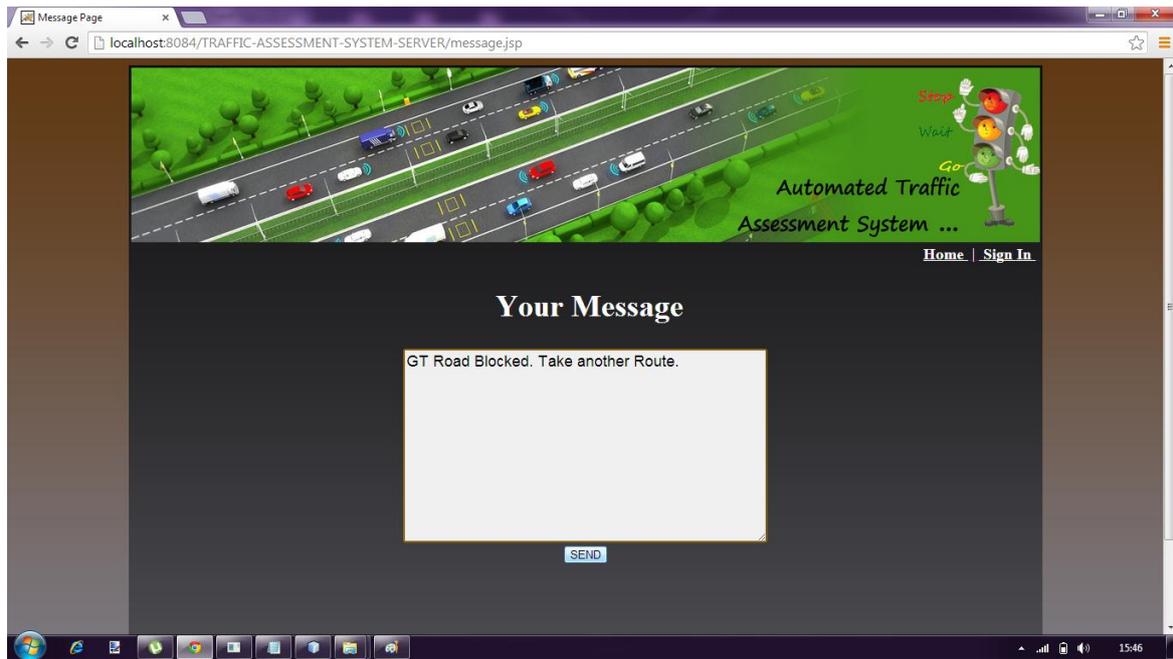


Fig 10: Server Sending Emergency Alert to Clients.

VI. Future Work

We do not have applied any encryption and decryption methods for files which are generated on the client side. So our System is less secure. We will apply this methods in future

VII. Conclusion

The design and implantation concept described in detail in the methodology section of this report. The result section shows that the system works as per our expectation. We hope that the concept and algorithms described in this report will aid the developers' community in near future.