A Mobile Phone based Intelligent Scoring Approach for Assessment of Critical Illness

Kaavya Rekanar (940521-7184) , Siva Venkata Prasad Patta (931221-7137), Sushmitha Donthula (940303-8483)

M.Sc. Computer Science

Blekinge Institute of Technololgy, Gräsvik Campus, Karlskrona, Sweden

kare15@bth.se, sipa15@bth.se , sudo15@bth.se

Abstract

We are designing a mobile phone based intelligent scoring approach for the assessment of critical illness that would send an SMS (a text messaging service component of phone) to an artificially intelligent agent, notifying it about the accident and thus, by accurately using the mobile number and tower location the accident location is identified. After finding the exact location, the necessary services can be hurried to the spot.

This report describes the problem domain and the implementation of the prototype, showing how the system operates via software, hardware with/ without human agents and its advantages in real world. The decision support system model and design has been specifically discussed in detail.

I. Introduction

In the modern world, people have become used to the idea of sending and receiving information almost instantaneously. The same text messaging service can be utilized in emergency situations to save someone from sticky situations.

Accordingly, a wide range of organizations including colleges and universities are now partners with third-party providers that promise to improve physical security by rapidly delivering such messages [1]. Accidents have become prevalent in recent times due to many reasons; most of which are traffic (in big cities) and weather conditions, drunk careless driving.

Most of the people are stressed out and overstrained [2] after accidents even if they are not hurt physically. Consequently, they may face some difficulty in reporting the accident to the police and civil defense, or they may provide them with inaccurate information about the location of the accident [2].

Some of the deaths in accidents can also be attributed to the long response time required for help to reach an accident location. The victims of an accident may be in need of urgent treatment and a delay in response time can increase the severity of the accident. Moreover, even if they were able to provide the necessary information it may take them some time to deliver it to the human counterpart. Hence it will take the police and civil defense more time to reach the accident location in the appropriate time to rescue people.

Existing System:

Devices like Jordan have been developed in automobiles to help people when they are in danger; but this requires a lot of extra equipment in the vehicles which is expensive and requires a whole new setup. This did not really appeal to many people as it meant that any person who uses this system should be aware of all the technical aspects involved in it's maintenance and even if that meant a new help line, it is actually hinting at remodeling the old vehicle to a newer model, which is not very convincing to an average person.

Proposed System:

The party in need will send a message to service center using their mobile device. An artificially intelligent agent called Traffic Controlling Agent (TCA) using a GSM modem will receive this message.

Using the mobile number and tower location from the party's message, location of help is decided by identifying the place where latitude and longitude meet each other. This concept is explained in the Project Design section. Using that location, the agent tries to identify the type of emergency and alert the police, fire service and medical emergency services in the respective area. This will reduce the time required for help to reach the accident location.

II. Division of Labor

The work of developing the project has been divided and the tasks to perform have been allocated depending on the tasks each of them were best at. The following are table outlines the main load of the tasks conducted by each team member.

Lawrence E. Henesey: Project Supervisor

Surname	Forename	Project Duty*
Rekanar	Kaavya	A-D, F-H
Patta	Siva Venkata Prasad	B, C, F, H
Donthula	Sushmitha	C, E, F, H

Table 1: * Project duties for the allotted tasks 1 to 10

A- Manager
B- Designer
C- Programmer
D- Documenter of the report
E- Logger (team session events)
F- Presenter

G- Viewer/reviewer (proof-reading)
H- Reviser (project revisions)

Task done for the project:

I. Database creation and data management issues
II. Algorithm
III. App Development
IV. Connecting tasks I, II and III: Coding to make this relationship possible
V. GUI's, portable and central communication interface for all the components of the project
VI. Bonus work to implement wireless communication between the agents involved
VII. Database updates supporting the interactive I/O data from the user (victim)
VIII. Validating our output
IX. Presenting out project
X. Documenting and logging

III. Project Analysis

a. Background

Cellular networks are increasingly becoming the primary means of communication during emergencies. Riding the widely held perception that text message is a reliable method of rapidly distributing messages, a large number of colleges, universities, and municipalities have spent millions of dollars to deploy third-party EAS over cellular systems. However, this is a security incident response. Recovery mechanism simply does not work as advertised. Through modeling, a series of experiments and collaborating evidence from real-world tests, we have shown that these networks cannot meet the 10 minute alert goal mandated by the public EAS charter and the WARN Act [1]. Moreover, we have demonstrated that the extra text messaging traffic generated by third-party EAS will cause congestion in the network and may potentially block upward of 80 percent of normal requests, potentially including calls between emergency responders or the public to 9-1-1 services [1]. Accordingly, it is critical that legislators [1], technologists, and the general public understand the fundamental limitations of this mechanism to safeguard physical security [1] and public safety and that future solutions are thoroughly evaluated before they are deployed.

b. Problem Definition

To determine whether there exists a mismatch between the [1] current cellular text-messaging infrastructures and third party EAS, it is necessary to observe such system during an emergency. However, because large-scale physical security incidents are rare, we apply a number of modeling techniques [1] to help characterize such events.

c. Problem Objectives

The main objectives of this project are:

- Reduce the time required to report an accident and to determine its location more precisely.
- Reduce deaths, which are happening simply due to the fact that help is not able to reach them within the required time.

d. Problem Solution and Scope

- Understanding packet loss behavior since loss can have [3] a significant impact on the performance of both TCP- and UDP-based applications.
- Then the message will be received by the service center (Traffic Controlling Agent) using GSM modem.
- Using the mobile number and tower location [3] the Agent identifies the accident location using latitude and longitude.
- Then, it analyzes the type of the accident and gives appropriate location to the respective nearby service (police, fire or ambulance).

e. System criteria

System Interfaces:

Short Message Service (SMS) is a text messaging service [2] component of phone, web, or mobile communication systems, using standardized communications protocols [2] that allow the exchange of short text messages between fixed line or mobile phone devices.

SMS text messaging is the most widely used data application in the world, with 3.6 billion active users, or 78% of all mobile phone subscribers [2]. The term SMS is used as a synonym for all types of short text messaging as well as the user activity itself in many parts of the world. SMS is also being used as a form of direct marketing known as SMS marketing.

SMS is used on modern handsets originated from radio telegraphy in radio memo pagers using standardized phone protocols [2] and later defined as a part of the Global System for Mobile Communications (GSM) series of standards in 1985 as a means of sending messages of up to 160 characters, to and from GSM mobile handsets [2]. Since then, support for the service has expanded to include other mobile technologies such

as ANSI CDMA networks and Digital AMPS, as well as satellite and Landline networks. Most SMS messages are mobile-to-mobile text messages though the standard supports other types of broadcast messaging as well.

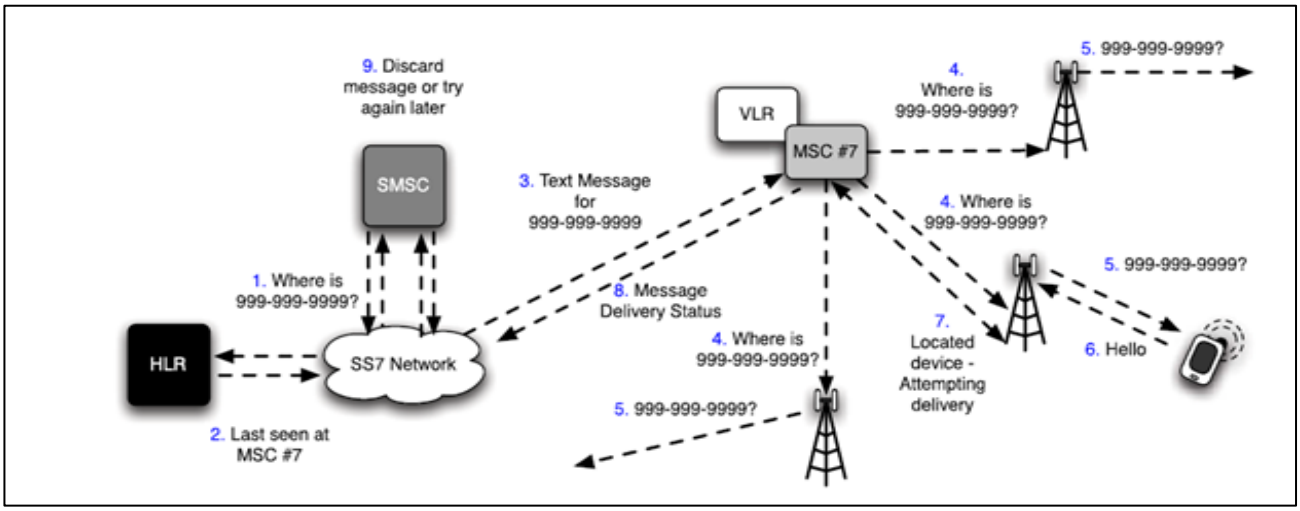


Figure 1: The working of the product to be constructed. The steps indicate the procedures, which are to be followed in order to achieve the target of helping the injured person at the spot of emergency

User Interfaces:

The Short Message Service – Point to Point (SMS-PP) was originally defined in GSM recommendation [1] 03.40, which is now maintained in 3GPP as TS 23.040. GSM 03.41 (now 3GPP TS 23.041) defines the Short Message Service – Cell Broadcast (SMS-CB) [1], which allows messages (advertising, public information, etc.) to be broadcast to all mobile users in a specified geographical area [1].

Messages are sent to a Short message service center (SMSC) which provides a "store and forward" mechanism [2]. It attempts to send messages to the SMSC's recipients, and if a recipient is not reachable, the SMSC queues the message for later retry [2].

Some SMSCs also provide a "forward and forget" option where transmission is tried only once [3]. Both mobile terminated (MT, for messages sent *to* a mobile handset) and mobile originating (MO, for those sent *from* the mobile handset) operations are supported [3]. Message delivery is "best effort", so there are no guarantees that a message will actually be delivered to its recipient, but delay or complete loss of a message is uncommon, typically affecting less than 5% of [3] messages. Some providers allow users to request delivery reports, either via the SMS settings of most modern phones, or by prefixing each message with *0# or *N# [3]. However, the exact meaning of confirmations varies from reaching the network [1], to being queued for sending [2], to being sent, to receiving a confirmation of receipt from the target device [3], and users are often not informed of the specific type of success being reported [3].

Communications Interface:

Larger content (concatenated SMS, multipart or segmented SMS, or "long SMS") can be sent using multiple messages, in which case each message will start with a user data header (UDH) containing segmentation information. Since UDH is part of the payload, the number of available characters per segment is lower: 153 for 7-bit encoding [2], 134 for 8-bit encoding [2] and 67 for 16-bit encoding [2]. The receiving handset is then responsible for reassembling the message often billed as equivalent to multiple SMS messages. Some providers have offered length-oriented pricing schemes for messages, however, the phenomenon is disappearing.

Memory Constraints:

- RAM minimum requirement is 512 MB.
- Hard Disk minimum requirement is 80GB.

Product Functions:

Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies.

Apache Tomcat software requirements: The Apache Tomcat servlet container and JSP environment from the Jakarta Apache site. The latest version of the Apache Web server for Windows binary distribution of the Apache server for versions.

User Characteristics:

The arrangement of all the menu items is done using this layout which is being horizontally arranged and easy to view and identify each icon uniquely.

Basically home page has all menu items arranged at the center of the page to identify and operate easily.

Constraints or Non- Functional Requirements:

According to the requirements, all the fields are put in a particular order for quick identification. Username and password are placed beneath the title where everyone can recognize very easily.

This automatically reduces the ambiguity and leads to the flexibility. Every page has its own description where beginner can also understand the flow of information.

In systems engineering and requirements engineering, a non-functional requirement [4] is a requirement that specifies criteria that can be used to judge the operation of a system [4], rather than specific behavior or functions. The plan for implementing *functional* requirements is detailed in the system *design [4]*. The plan for implementing *non-functional* requirements [4] is detailed in the system *architecture.*

In general, functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be [4]. Functional requirements are usually in the form of "system shall do <requirement>" [4], while non-functional requirements are "system shall be <requirement>" [4].

Fixed satellite services: Fixed satellite services handle billions of voice [2], data and video transmission tasks across all countries and continents between certain points on the Earth's surface [2].

Mobile satellite system: Mobile satellite systems help connect remote regions, vehicles, ships, people and aircraft [2] to other arts of the world and/or other mobile or stationary communications units, in addition to serving as navigation [2] systems.

Performance:

The software application response time, is an aspect of software quality that is important in human-computer interactions. The performance of any computer system can be evaluated in measurable, technical terms, using one or more of the metrics listed above. This way the performance can be

i. Compared relative to other systems or the same system before/after changes
ii. defined in absolute terms, e.g. for fulfilling contractual obligation

Functions:

i. Clients sends the message
ii. SMS is received by GSM modem
iii. Data is stored in database and server performs its operation to make a service call for reporting
iv. GPS determines the longitude and latitude (geographical) and after finding the location EAS makes a call to the corresponding service.

Logical Database Requirements:

Microsoft database is been used by MySQL. Enterprise also provides a comprehensive knowledge-base library that includes hundreds of technical articles resolving difficult problems on popular database topics such as performance, replication, and migration.

Organising the specific requirements:

External Interface Requirements-

Input- Short Message Service (SMS) is a text messaging service component of phone, web, or mobile communication systems, using standardized communications that allow the exchange of short text messages between fixed line or mobile phone.

Output: Accurately using mobile number and tower location they identify accident location using latitudes and longitudes.

User Interfaces-

This application include GUI standards or product family style guides that are to be followed, screen layout constraints, buttons and functions that will appear on every screen, error message display standards, and so on. This application provides every user a comfort to make their search possible in a smooth manner.

Hardware Interfaces-

Main Processor	: >2GHz
RAM	: 512
Hard Disk	: 80 GB

Ports	: 1 Serial Port	Language	: Java, JSP, Servlets
GSM Modem		Web Server	: Tomcat
Operating System	: Windows XP	Database	: MySQL

JAVA Supported (jar) GPRS and GPS enabled mobile phone.

Software Interfaces:

Functional Requirements-

There are four modules in this project. They are as follows:

1. Client Module.
2. GPS Module.
3. Server Module.
4. GSM and Database Module.

Client Module: The mobile requests its location from the positioning system periodically and sends it through the communication network to the server. A location-based service is an information service that can be accessed using the mobile device through the mobile network and utilizes the ability to make use of geographical positions of the mobile device.

GPS- Global Positioning System: Every time a mobile phone updates the user's location to the server, it requests the location of the user from the GPS installed. The GPS determines the longitude and latitude (geographical) and sends them through the mobile phone.

Server: When the server receives a user alert message, the artificially intelligent agent designed, i.e., the Traffic Monitoring Center (TMC) will check the type of message and determines the kind of emergency. Keeping tabs on the public service centers from the area from where the alert message has been received does this job. The required help is directed to the location later.

GSM and Database: The database contains all the users subscribed in the service with their receiving message. On receiving the message, the TMC does its job.

GSM is a specialized modem, which accepts a SIM card and is operated over a subscription to a mobile operator, just like a mobile phone. This will help the agent in locating the client more precisely.

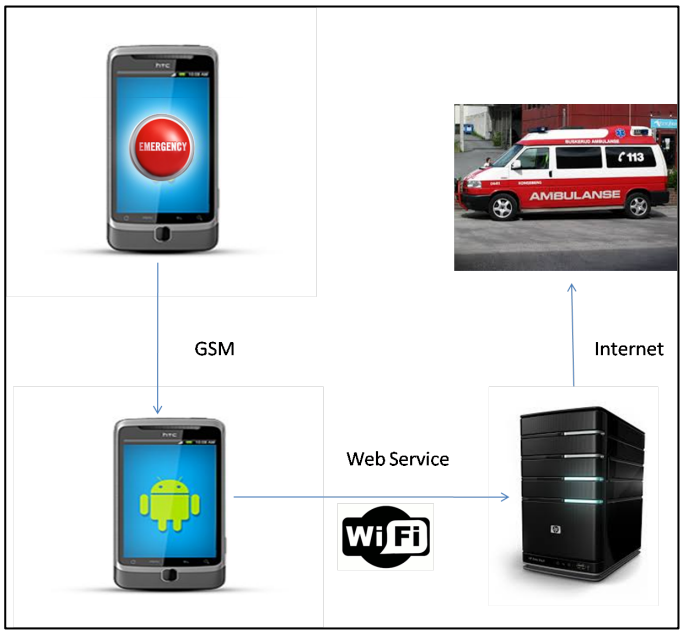


Figure 2: The working of all the four modules as a system

Software System Attributes:

- ***Authenitication*:** It is done through login password where it allows only valid users.
- ***Security:*** The web application which is being developed is free from worms, threats, virus etc. The user has to take care by not giving his/her password to others.
- ***Maintainability:*** MYSQL database stores all the information, so that user required information will be retrieved and displayed on the monitor.

f. SWOT Analysis

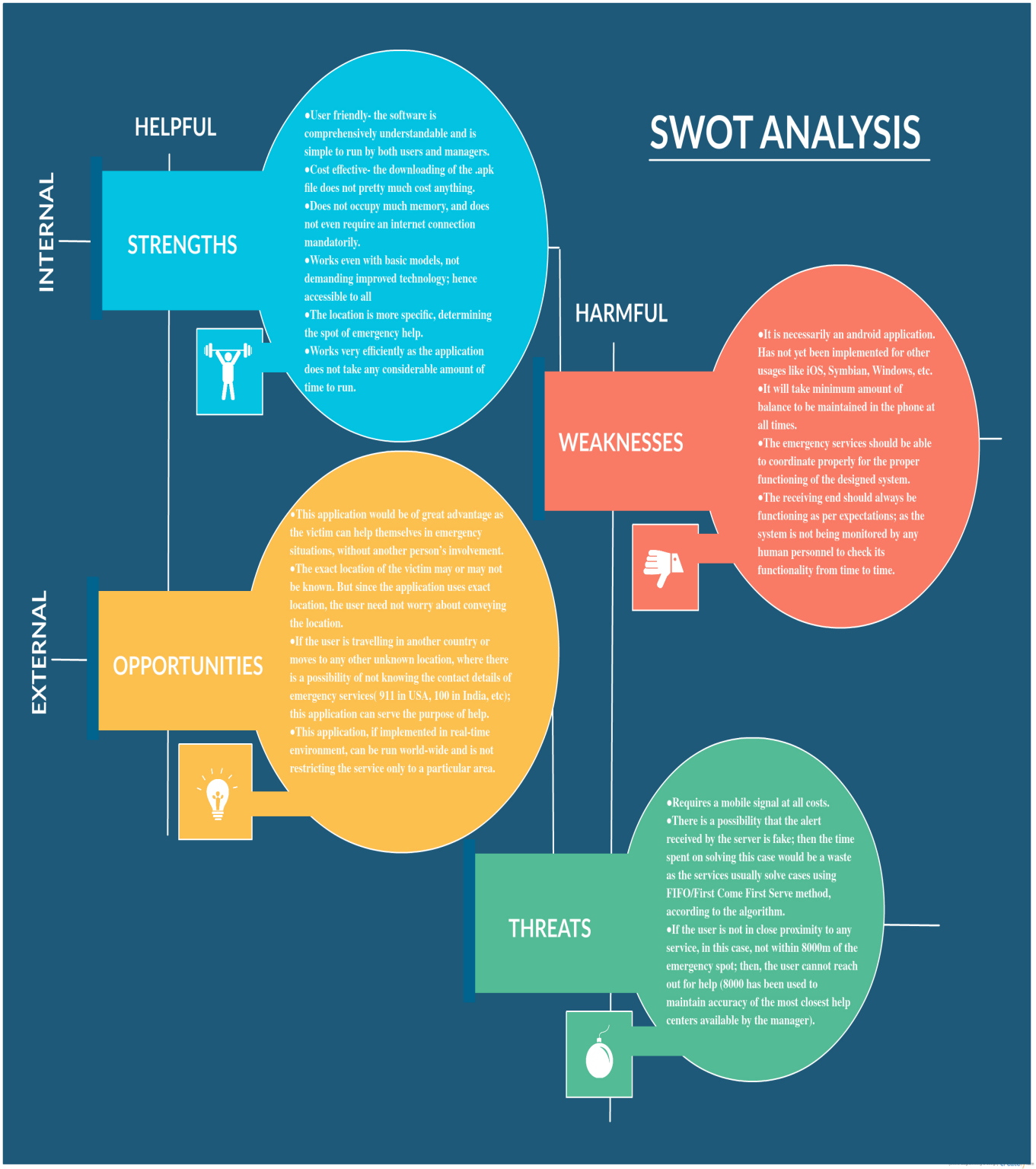


Figure 3: SWOT Analysis

IV. Project Design

Analytical Design

Use Case Diagram-

Use case is a description of the behavior of the system. Written from the user point of view, a use case captures the visible sequence of events that a system goes through in response to a single user response.
The use case diagram (Figure 4) here, describes the behavior of the model being developed according to the design. The Client and the Server are the actors here. The use cases- Incident, GSM, Sent message, Find location, GPS co-ordinates, Find incident and Related service- are a visual representation of distinct business functionality in the system.

Use case name	Overall usecase	
Use case id	1	
Participating Actors	Third party, server	
Pre Condition	In this application third party send sms to service center	
Typical course of events (main flow)	Actor action	System response
Step 1	Third party sends sms	System responses to user actions
Step 2	Traces the location through latitude and longitude values	Searches for data database
Step 3	Server makes service call	Provides helpline services
Post condition	A mobile is able to send a message	
Non functional Requirements	A mobile should be android supported	

Table 2 : Use Case table

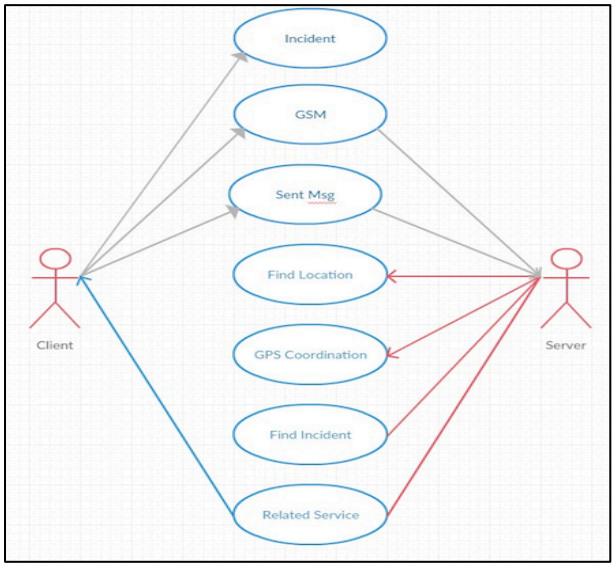


Figure 4: Use Case diagram

Activity Diagram-

Activity Diagram depicts the activities which make up the system. The activities are listed accordingly in a sequential order.
Figure 5 shows the Activity Diagram in which all the proceedings in the system can be clearly understood.

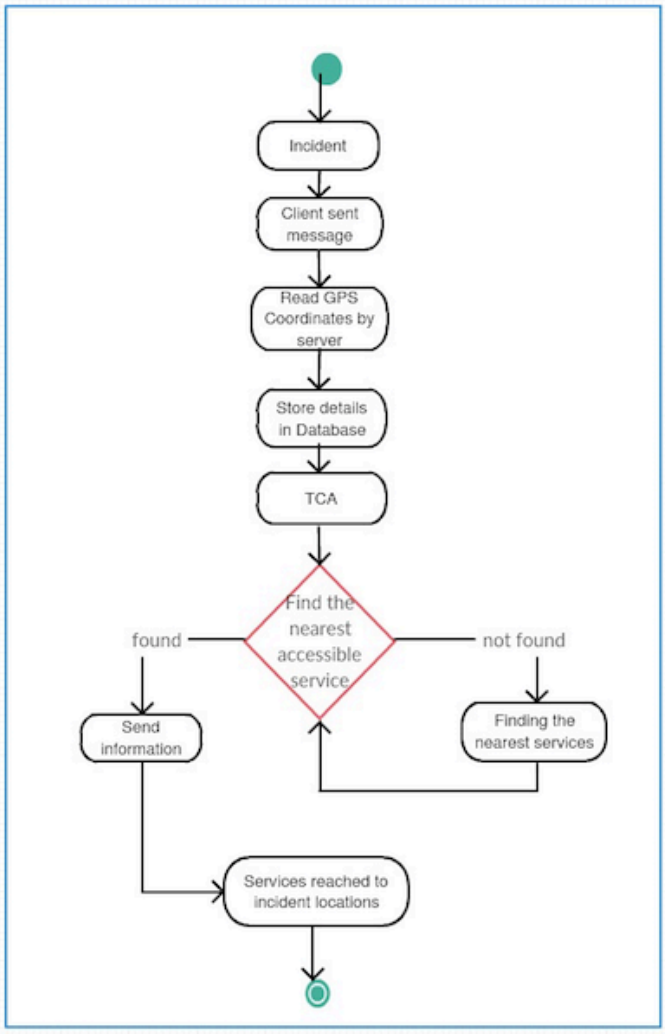


Figure 5: Activity Diagram

Interface Design

Sequence Diagram

The interactions among classes in terms of an exchange of messages over time is shown in the sequence diagram.

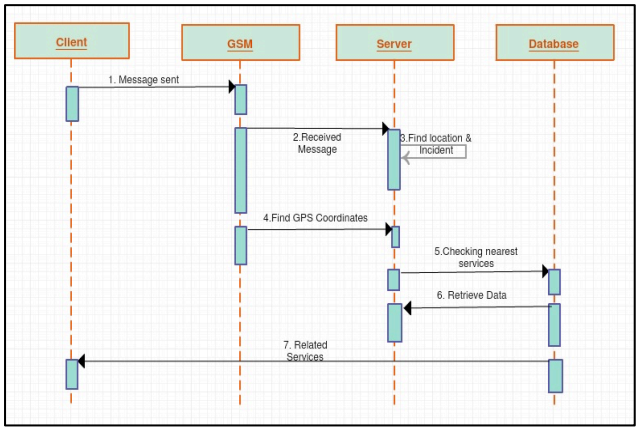


Figure 6: Sequence Diagram

Collaboration Diagram

Collaboration diagrams represent interactions between objects as a series of sequenced messages. These diagrams describe both the static structure and dynamic behavior of the system.

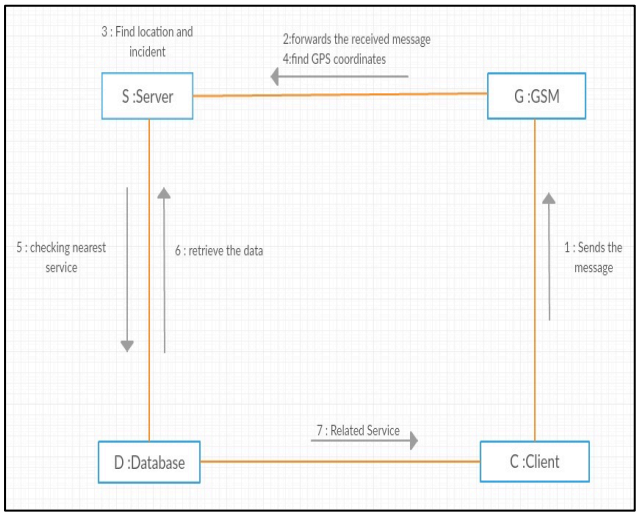


Figure 7: Collaboration Diagram

Data Design

Class Diagram

The class diagram comprises of five major classes:

- Client
- GSM
- Database
- Server
- Services.

The objects and functionalities for these classes are clearly listed.

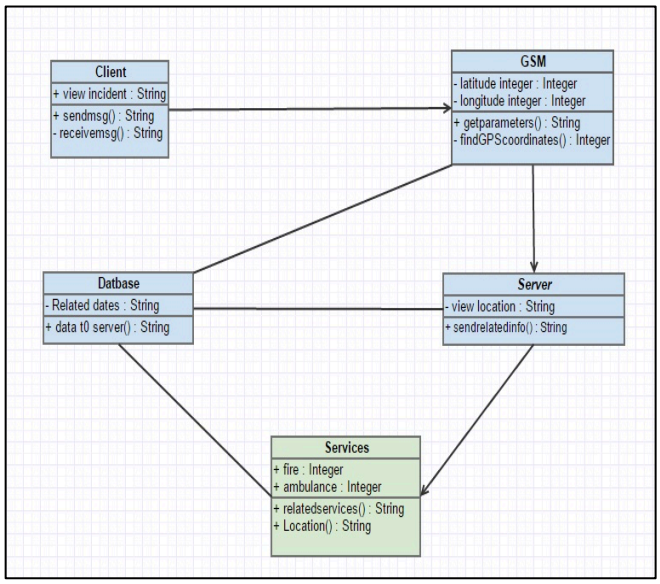


Figure 8: Class Diagram

Component Diagram

This diagram depicts the organisation of components, including source code, run-time code and executables in the application.

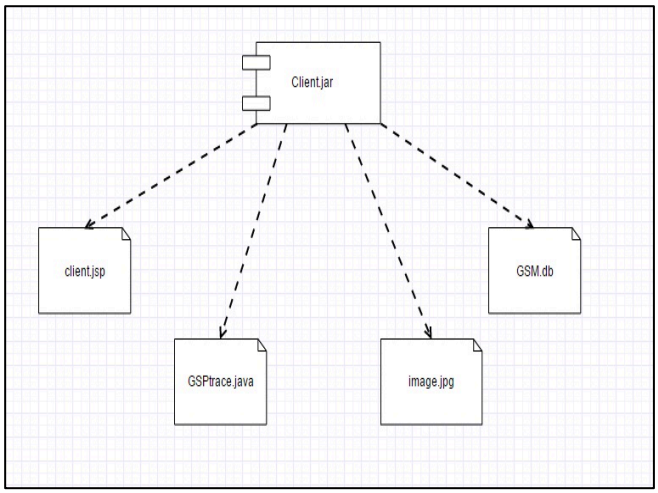


Figure 9: Component Diagram

Deployment Diagram

This diagram visualizes the topology of the physical components of a system where components are deployed.

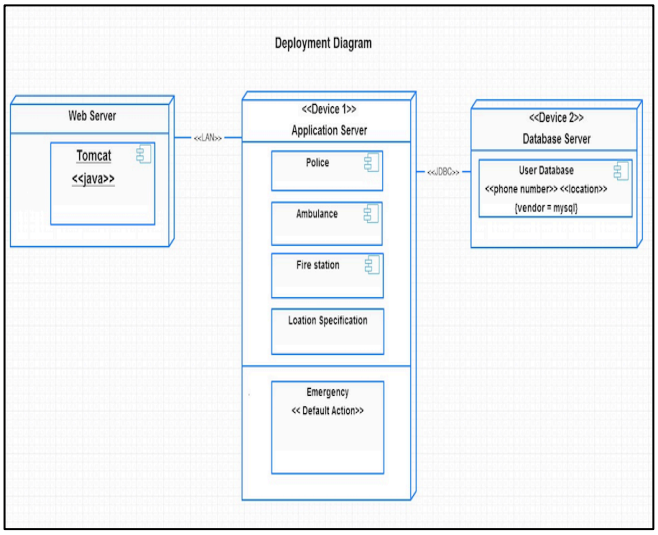


Figure 10: Deployment diagram

V. Project Implementation

The system has been implemented based on **Simon's model of decision making** where the following phases have been used for the complete development.

Phase 1: Intelligence phase

For the goal of the system to be achieved, certain challenges had to be overcome; out of which the emergency calling procedures for a telecommunication system had to be figured as a primary need.
Challenge 2- The next challenge was to develop the agent such that it projects the importance of the message and diverts appropriate service to the spot of emergency.
Challenge 3- To reduce the time of travel for emergency aid to reach the identified location.

Phase 2: Design Phase

The system's working is generated, developed and analysed in this phase. The steps involved in the process are:
After conceptualizing the problem, and analyzing the major milestones in the system's development, which includes both the server and Traffic Controlling Agent (TCA), the design has been constructed.
Since the design of the system is developed on Operational Planning and Control, it's nature is classified as Quantitative.
Operational Planning and Control- This system is focused on efficient and effective working in the performance of necessary tasks.
The efficiency of the system is determined where the help service reaches the exact location of need. The effectiveness is determined on how fast the service is able to reach given the close proximity to the needed location.
Structured problem because a routine has been set and the solution is more or less repetitive in every case. A well-defined decision making procedure is followed by the system when it is working, i.e., a defined set of input is given and a defined set of output is expected based on the circumstances in which the agent works.

Alternative solutions

- To reduce the time of working for TCA in case of extreme emergencies, the Internet messaging facility (Skype) can be ignored and a direct text message can be sent. The reason for this is, a direct text message is not specifically using the server immediately. Though, the processing is saved, it is not done on the spot which saves up to 0.10 seconds of the client's time for every proceeding which can sometimes make a huge difference.
- If there is no help service available in the closest proximity of the set radius (i.e., 8000), which is responding at that moment, the TCA suggests other near services to the client immediately.

The model developed is described clearly in the Model section.

Phase 3: Choice phase

Based on the model developed, the choice phase mainly focuses on the proposed solution and the choices the agent makes for every anticipated proceeding.
The solution developed is based on Analytical techniques and suitable algorithms.
The analytical technique is involved where a decision is taken by the TCA in selecting the emergency service required at the site.
Algorithms are involved where Operational Planning and controlling is operating. The TCA precisely works on the set algorithms for the four modules as described in the Pseudo code section.
The robustness in the system can be analyzed on the speed of the network. The traffic should be controlled and diverted effectively in order to have the fastest output.

Phase 4: Implementation phase

In this phase, we have to implement the proposed solution (i.e., as described in the model/ Design phase). The major concerns in this phase have been listed below;

- *Dealing with resistance to change*-According to the design we cannot actually implement the model and test it as this would involve the Police and other emergency services which could turn into a crime for providing false alert to the authority. Hence, we have set up a random phone number to which the call could be placed (one of the team member's personal number). This way, the implementation of the proposed model can be done without causing any disturbances.

- *User Training*-The client/ user has to be trained before hand in order to check the implementation of the project. The installation of the application has to be known to the public for successful implementation.
- *Upper management support*-This is the part where this project is actually implemented in real-time and the emergency services work towards the success of the operation.

Phase 5: Feedback and monitoring phase

In this phase, an evaluation of all the decisions made in the previous phases is done and the proficiency of the system is determined, which can be performed in the following manner.

- *Evaluate decision processes*-The decision taken by the agent on sending the appropriate service to the required location is evaluated based on the time taken to decide by the system, the total throughput time (including execution) and the ability of the agent to correctly assess the situation in hand.
- *Evaluate entire system*-The entire system can only be evaluated if the project is implemented in real time environment. However, an evaluation with the random number placed can be done just to be sure that all the modules are in perfect sync with each other.
- *Continuous updates and fine-tuning of decisions made earlier*-The project can be evaluated by the working of agent considering different scenarios. As this project is developed such that it can be used worldwide, we can assess the decisions made by the agent considering the location, weather and cultural differences, as well.
  The decisions made during the designing of the system can be validated in the most unsurpassed manner this way.

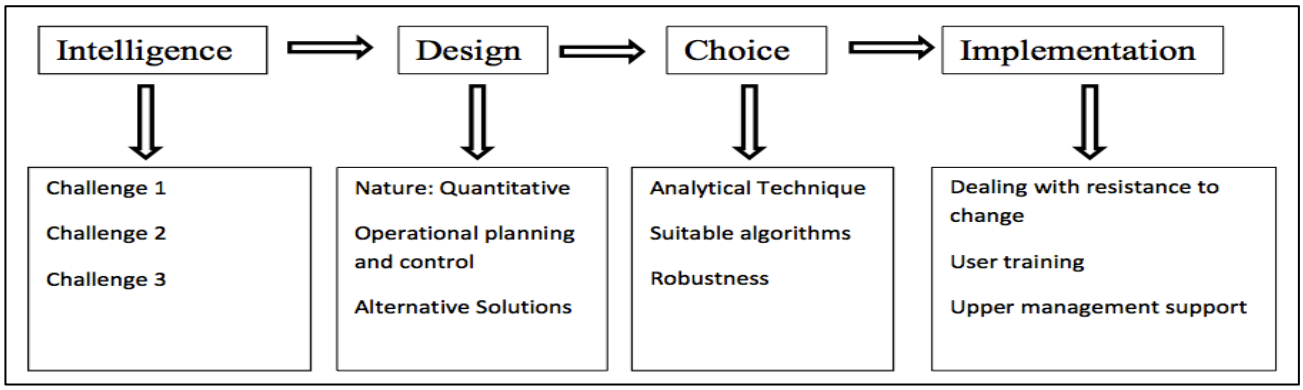


Figure 11: Simon's model of decision making

Taxonomy of Decision Support System used

The typical steps any decision support system would work on are (a) Accessing all information assets, (b) comparative data figures, (c) Projected figures based on new assumptions or data, (d) consequences of different alternatives made in decisions using past experiences.
The different types of DSS used to develop the system using these steps are:

- *Knowledge driven DSS*-The agent always follows a pre-set procedure for all the actions. Therefore, the system works on this type of DSS
- *Data driven DSS*-All the requests by the user to the agent are stored in the database, which are accessible using Query techniques. The data is stored in Tomcat server and the queries are processed using MySQL Workbench.
- *Communication driven DSS*-The server uses this type of DSS to a certain extent as the communication is only in one direction, i.e., the agent is communicating with all the needed emergency services.

Incorporated models

The system built based on the design developed incorporates the following models;

- *Analytical Model*-The agent will analyze the requirement of service at the location and divert them accordingly.
- *Operational Model*- The analysed requirement is sent a notification by the agent to reach the location in order to provide necessary help.
- *Tactical Model*-This model is involved where the agent automatically searches for nearest emergency service centers within a set radius (i.e., 8000) and then diverts them to the location.

Pseudocodes

Module 1- User application

This is the application where user requests the server module. An sms is sent to the server, which contains latitude, longitude and type of emergency. There is also a default option where the user can send the text without bothering to mention about the emergency. The TCA will figure out the issue with it's intelligence.

```
import com.loopj.android.http.RequestParams;

public class MainActivity extends Activity {
        Context mContext;
        // flag for Internet connection status
        Boolean isInternetPresent = false;

        //Emergency Team Name
        private String emergencyTeam;
```

Module 2: Server Module
The server module is further divided into 2 sub modules: server application in the mobile and the server

Work done in server application:
The work done here is listening to the text sent by the client, and figuring out the emergency service stations located closest to the accident spot.
We are calling the google api to provide the nearest emergency services using location and request type through json string. The search is done by setting the radius to around 8km. Using gson library, we parse the location of the given string.
The key is generated by the manager.

Work done in server:
The server's job is to receive data. We use server and user on the same network (LAN).
For the mobile device of the server application to communicate with the server, we use ip address. The server address is entered into the mobile, but due to large network possibilities, we have to be more specific.

```
public class SmsListener extends BroadcastReceiver {

        private SharedPreferences preferences;

        // private onSmsReceive listener = null;

        public void setOnSmsReceivedListener(Context context) {
                // this.listener = (onSmsReceive) context;
        }

        @Override
        public void onReceive(Context context, Intent intent) {
                // TODO Auto-generated method stub

                if (intent.getAction()
                                .equals("android.provider.Telephony.SMS_RECEIVED")) {
                        Bundle bundle = intent.getExtras(); // ---get the SMS message passed
                                                                                        // in---

                        SmsMessage[] msgs = null;
                        String msg_from;
                        if (bundle != null) {
                                // ---retrieve the SMS message received---
                                try {
                                        Object[] pdus = (Object[]) bundle.get("pdus");
                                        msgs = new SmsMessage[pdus.length];
                                        for (int i = 0; i < msgs.length; i++) {
                                                msgs[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
                                                msg_from = msgs[i].getOriginatingAddress();
                                                String msgBody = msgs[i].getMessageBody();
                                                Log.e("test", "got msg>>>>>>>>>>>>>>>>>>>>." + "    "
                                                                + msgBody + "   " + msg_from);
                                                ;

                                                if (msgBody.contains("Emergency Request")) {
```

```
        String team="police";

        if(msgBody.contains("Crime")) team="police";
        else if (msgBody.contains("Fire")) team="fire_station";
        else if (msgBody.contains("Medical")) team="hospital";

        // start activity here

        Intent myIntent = new Intent(context,
                        MainActivity.class);
        myIntent.putExtra("msg", msgBody);
        myIntent.putExtra("num", msg_from);
        myIntent.putExtra("team", msg_from);

        myIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        context.startActivity(myIntent);
```

MainActivity.java

```
        }

        // creating GPS Class object
        gps = new GPSTracker(this);

        // check if GPS location can get
        /*
         * if (gps.canGetLocation()) { //Log.d("Your Location", "latitude:" +
         * gps.getLatitude() + ", longitude: " + gps.getLongitude()); } else {
         * // Can't get user's current location
         * alert.showAlertDialog(MainActivity.this, "GPS Status",
         * "Couldn't get location information. Please enable GPS", false); //
         * stop executing code by return return; }
         */
        // Getting listview
        lv = (ListView) findViewById(R.id.list);

        // button show on map
        btnShowOnMap = (Button) findViewById(R.id.btn_show_map);

        // calling background Async task to load Google Places
        // After getting places from Google all the data is shown in listview

        /**
         * ListItem click event On selecting a listitem SinglePlaceActivity is
         * launched
         * */
        lv.setOnItemClickListener(new OnItemClickListener() {

                @Override
                public void onItemClick(AdapterView<?> parent, View view,
                                int position, long id) {
                        // getting values from selected ListItem
                        String reference = ((TextView) view
                                        .findViewById(R.id.reference)).getText().toString();

                        // Starting new intent
                        Intent in = new Intent(getApplicationContext(),
                                        SinglePlaceActivity.class);

                        // Sending place refrence id to single place activity
                        // place refrence id used to get "Place full details"
                        in.putExtra(KEY_REFERENCE, reference);
                        startActivity(in);
                }
        });
```

Module 3: Web service module
Web service is hosted on the web server tomcat. The service is installed on port 8080. The web service connects data internally, and all the data is stored in a different location. Web server and database servers are set up on different computers to manage the load of huge user requests in real time scenario.
The default port number for database server will be 3306. Internally, an ip address is used to link both web service and database server.

```
@Path("/accident")
public class Accident {

    // HTTP Get Method
    @GET
    // Path: http://localhost/<appln-folder-name>/login/dologin
    @Path("/doinsert_records")
    // Produces JSON as response
    @Produces(MediaType.APPLICATION_JSON)
    // Query parameters are parameters: http://localhost/<appln-folder-name>/login/dologin?username=abc&password=xyz
    public String insert_records(@QueryParam("req_num") String req_num, @QueryParam("req_lat") String req_lat,
            @QueryParam("req_lng") String req_lng) throws SQLException {
        String response = "";
        {
            if (Utitlity.isNotNull(req_num)
                    && Utitlity.isNotNull(req_lat)
                    && Utitlity.isNotNull(req_lng)) {
                try {
                    if (DBConnection.insertRecord(req_num, req_lat, req_lng)) {
                        response = Utitlity.constructJSON("insert", true);
                    } else {
                        response = Utitlity.constructJSON("insert", false);
                    }
```

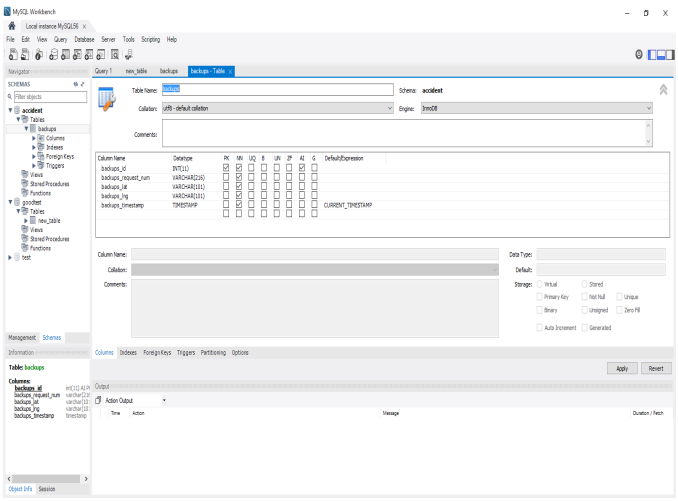
Module 4: Database module
All the data is pushed into database using jdbc, i.e., the connection is done using Java Database Connectivity.

```
public class DBConnection {

    /**
     * Method to create DB Connection
     *
     * @return
     * @throws Exception
     */
    @SuppressWarnings("finally")
    public static Connection createConnection() throws Exception {
        Connection con = null;
        try {
            Class.forName(Constants.dbClass);
            con = DriverManager.getConnection(Constants.dbUrl, Constants.dbUser, Constants.dbPwd);
        } catch (Exception e) {
            throw e;
        } finally {
            return con;
        }
    }

    /**
     * Method to check whether uname and pwd combination are correct
     *
     * @param uname
     * @param pwd
     * @return
     * @throws Exception
     */
    public static boolean checkLogin(String uname, String pwd) throws Exception {
        boolean isUserAvailable = false;
        Connection dbConn = null;
```

The database is setup using MySQL. The WAR file has been generated from netbeans, and is updated onto the web server.



Simulation of Results

User Application

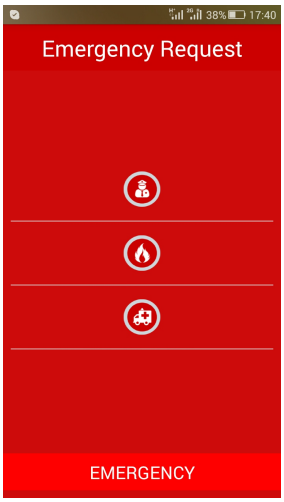


Figure 12: User app home page

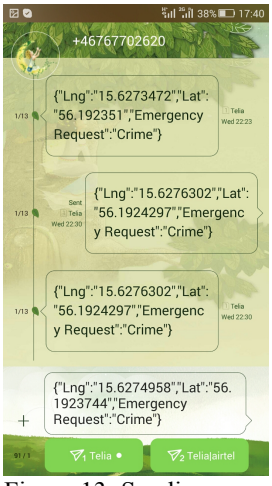


Figure 13: Sending message to server

Server Application

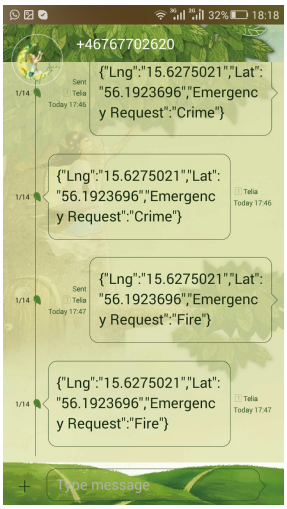


Figure 14: Receiving message from user

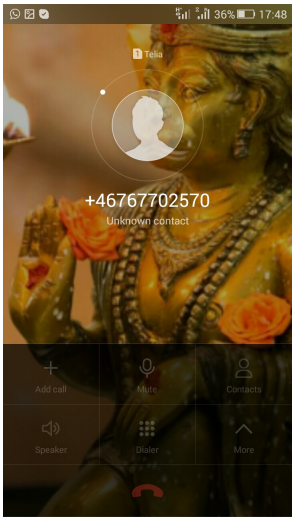


Figure 15: TCA calling the Police since the request is crime

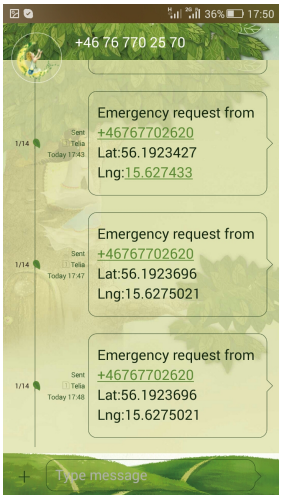


Figure 16 : TCA sending the emergency message to Police

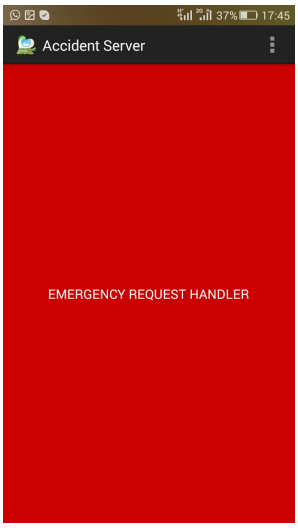


Figure 17: Server home page

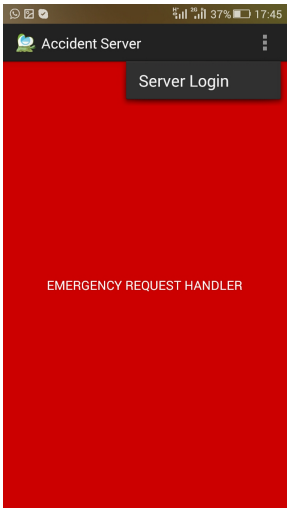


Figure 18: Server Login option in Server page

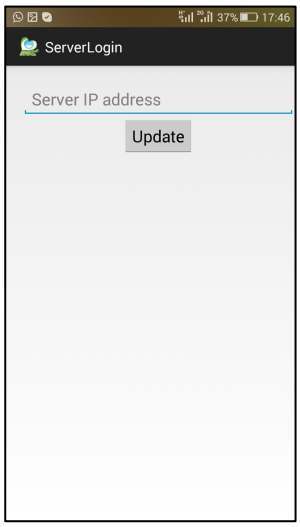


Figure 19: Entering the IP address for installation

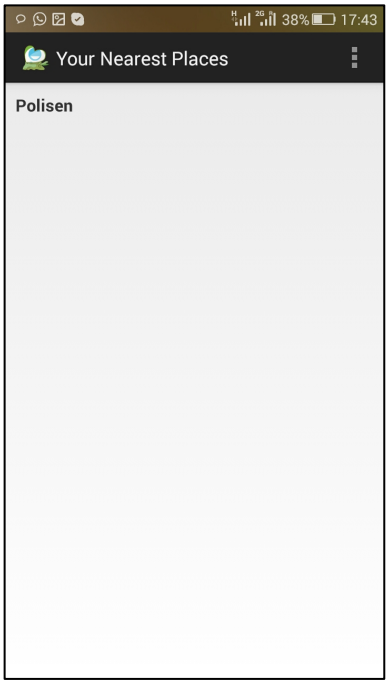


Figure 20: TCA displaying the nearest places to route services

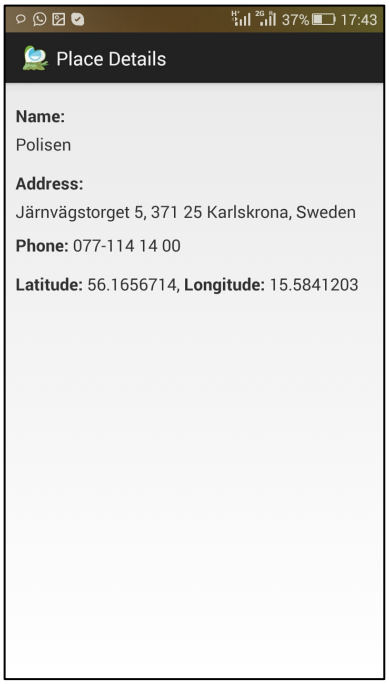


Figure 21: Details of the nearest police station Emergency Service

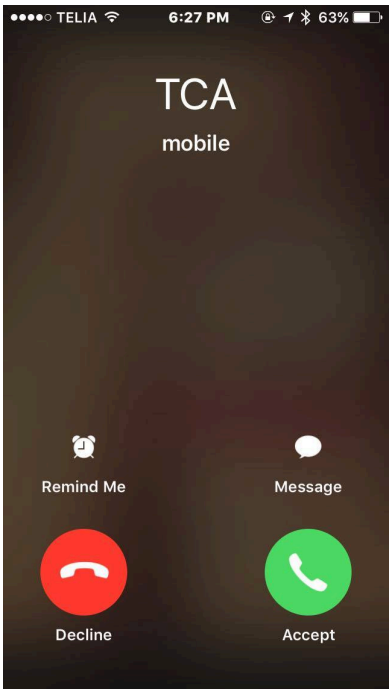


Figure 22:TCA calling Police

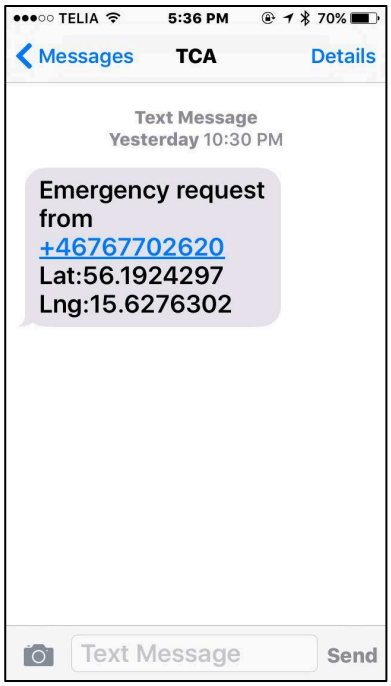


Figure 23: Emergency message sent to the Police by TCA

Server

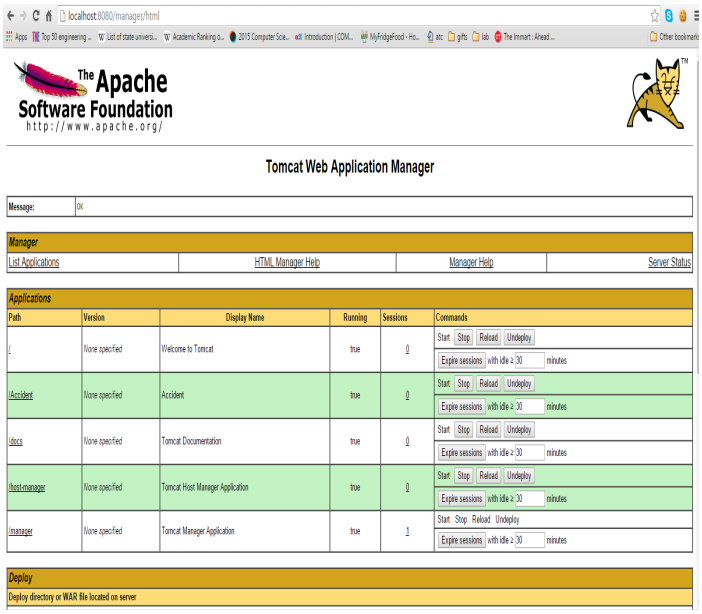


Figure 24: Working of Server

Database

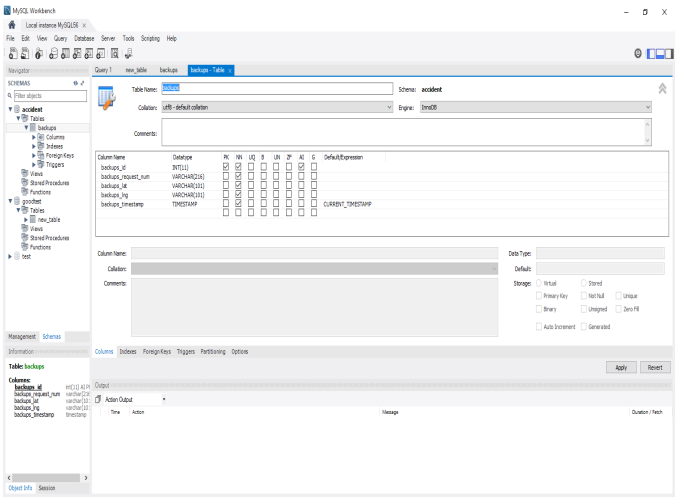


Figure 25: Database stored

VI. Project Optimisation

The project developed is an emergency support service, which means that every second of processing contains importance.
The most time occupying work is parsing of the identified location into the string.
As explained earlier, we are using json for this purpose.
XML has the same functionality; but the reason for this preference is that XML takes multiple steps to parse which occupies a time of O(mn), while json can parse the whole string in a single step which occupies a time period of O(m). Hence, keeping this in mind json has been used. Though this requires a system with higher functionality, it can counted in as an optimization since the requirement is only at the server end. In real time scenario, this will a one- time investment and will not be of much problem.

The installation of the application will take little time, which is again a single time installation. Even if the application is deleted and re-downloaded, there is no need to enter an ip address again. Everything is automatically scanned and filled in within a gap of two seconds. Thus, there is no problem of time consumption.

Though there is a lot of background work going on for the emergency service in order to actually reach the client, all the hard work is done on the server side, which makes the client side completely ignorant. The application takes only 400kb which is very less memory, thus optimizing the whole system in a user point of view.

VII. Evaluation- Verification and Validation

One of the major quality measure employed during software development is evaluation. Usually, for verification the program is executed with a set of test cases and the output for those test cases is evaluated to determine is the program is working as it is expected to.

Test cases design methods can be classified into two types:

- White Box Testing- a case designed method that uses the control structure of the procedural design to derive the test cases. This method enables the test case designer to derive a logical complexity measure of procedural design and use that measure as a basis to exercise every statement in the program.

- Black Box Testing- a behavioral testing method, which mainly focuses on the functional requirements of the software. This type of testing is attempted in order to find incorrect or missing functions, interface errors, errors in the data structures or external database access, behavior or performance errors, initialization and termination errors.

Project Specific Test Cases

Test Case	Check Field	Objectives	Expected Result
TC-001	GPS	Turn off GPS in the mobile	Should indicate that Turn ON GPS
TC-002	GPS	Turn ON GPS in the mobile	Should load the latitude and longitude value to the application
TC-003	GPRS	Turn off GPRS in the mobile	Should indicate that Turn ON GPRS
TC-004	GPRS	Turn ON GPRS in the mobile	Application gets ready to access

			internet
TC-005	Application	Launch the application in the android mobile	Application should be launched without showing any error
TC-006	Send-Button	Leave the fields empty and press SEND button	Error message should appear that " Don't leave the fields empty"
TC-007	Server Name	Type invalid server name Like: test	Error message should appear "Please enter a valid server name"
TC-008	Server Name	Type valid server name like:108	It should allow users to proceed further
TC-009	Send-Button	Fill all the fields and press send button	The alert should be forwarded to server successfully
TC-010	Link	Press link showed on the browser	The exact location of the emergency and the nearest support center should be indicated

Table 2: Test cases for alert system- White Box Testing

Test Case	Check Field	Objective	Expected Result
TC-001	Mobile	Open the application and enter the user name and password	Should give correct user name and password
TC-002	Mobile	Fill all the fields and press SEND button	The alert should be forwarded to server successfully else given correct data's
TC-003	Mobile	Turn ON GPS in the mobile	Should load the latitude and longitude value to the application
TC-004	Mobile	Turn ON GPS in the mobile	Error in the internet connection
TC-005	Server	Type valid server name	It should allow users to proceed further
TC-006	Server	Press link showed on the browser	The exact location of the emergency and the nearest support center should be indicated

Table 3: Test cases for emergency alert- Black Box Testing

Validating

Validation can be done in two ways:

- Bottom up approach-specified with the assistance of White Box testing.
- Top down approach-specified with the assistance of Black Box testing.

The validation is done using Top down approach, where a complex system is subdivided and tested individually. When the expected result matches with the obtained result, we can give a perfectly validated application.

The Java application has been tested successfully.

VIII. Future Work

There can be instances when people prank about being in an emergency situation. To avoid such cases, we can put an image confirmation clause where the seriousness of the situation can be pictured and sent to TCA, which analyzes and confirms if it is really a sticky situation where a person needs immediate help.

Another development can be done in the form of IVRS to the emergency service personnel. The agent should read out the location when a call is being placed to the personnel's phone. That would be helpful if they are travelling in traffic and cannot check the message sent by the TCA. We have done the project in both call and message but an IVRS is beyond the scope of the project.

IX. Conclusion

An atttempt has been made to develop an application fusing both artifical intelligence and android technology. The agent is given the capability of taking decisions according to the situation and act as per the previously given directions of use. This is extensively exploiting the knowledge-based decision support system and putting to use the most comprehensively used technology of texting in today's time.

X. Reference

[1]. P. Traynor, "Characterizing the Security Implications of Third-Party Emergency Alert Systems over Cellular Text Messaging Services," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 983–994, Jun. 2012.

[2]. G. Shanyora and A. K. Chaitanya, “Third-Party Emergency Alert Systems over Cellular Text Messaging Services Providing The Security Implications,” IOSR Journal of Computer Engineering, vol.12, Issue 5, pp. 85-91, Aug 2013.
[3]. J. Sommers, P. Barford, N. Duffield, and A. Ron, “Improving accuracy in end-to-end packet loss measurement,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 157–168, 2005.
[4]. M. Ahmad, N. Belloir, and J.-M. Bruel, “Modeling and Verification of Functional and Non-Functional Requirements of Ambient Self-Adaptive Systems,” *J. Syst. Softw.*, vol. 107, no. C, pp. 50–70, Sep. 2015.
[5]. A. Acker, “The Short Message Service: Standards, infrastructure and innovation,” *Telematics and Informatics*, vol. 31, no. 4, pp. 559–568, Nov. 2014.
[6]. L. Luong, J. He, K. Abhary, and L. Qiu, “A decision support system for cellular manufacturing system design,” *Computers & Industrial Engineering*, vol. 42, no. 2–4, pp. 457–470, Apr. 2002.
[7]. D. Mirchandani and R. Pakath, “Four models for a decision support system,” *Information & Management*, vol. 35, no. 1, pp. 31–42, Jan. 1999.