



**CRISIS MANGAEMENT AND
ALARM SYSTEM USING ANDROID
APPLICATION**



By

M.ASHWIN

412511205010

S.MUKESH KUMAR

412511205052

S.SIVA HARI GANESAN

412511205087

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION TECHNOLOGY

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

SRI SAIRAM ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY :: CHENNAI - 600 025

APRIL 2015

ANNA UNIVERSITY: 600025

BONAFIDE CERTIFICATE

Certified that this project report titled “**CRISIS MANAGEMENT AND ALARM SYSTEM**” is the bonafide work of “**SIVA HARI GANESAN S (412511205087),ASHWIN M (412511205010) and MUKESH KUMAR S (412511205052)**”, who carried out the project work under my supervision. Certified further, that to the best of my knowledge that the work reported here does not form part of any other thesis on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. T.SHEELA. B.E., M.S., PhD

DEAN [NETWORK] & HEAD

Department of Information Technology,
Sri Sai Ram Engineering College,
Chennai- 600 044.

SIGNATURE

Ms. T.SUBHA. M. tech ,(PhD)

SUPERVISOR

ASSISTANT PROFESSOR - I
Department of Information Technology,
Sri Sai Ram Engineering College,
Chennai- 600 044.

Submitted for University Project Examination held on at Sri Sai Ram Engineering College, Chennai-600 044.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our gratitude to our beloved chairman **Thiru. MJF. Ln. LEO MUTHU** for the help and advice he has shared upon us.

We express our gratitude to our CEO **Mr. J. SAI PRAKASH LEO MUTHU** and our trustee **Ms. J. SHARMILA RAJAA** for their constant encouragement in completing this project.

We express our solemn thanks to our Principal **Dr. C.V. JAYAKUMAR** for having given us spontaneous and whole hearted encouragement for completing this project.

We are indebted to our Head of the Department **Dr. T. SHEELA** for her constant support during the entire course of this project work.

We express our gratitude and sincere thanks to our guide **Ms.T.SUBHA** Assistant Professor for her valuable suggestions and constant encouragement for successful completion of this project.

Our sincere thanks to our project coordinator **Ms.S.ANITHA JEBAMANI** for her kind support in bringing out this project successfully.

Finally, we thank all the staff members of the **Department of Information Technology** and all others who contributed directly or indirectly for the successful completion of our project. We thank our almighty for his blessings to make this project a success.

ABSTRACT

Human Population is constantly under threat from natural disasters .In this project, an early warning system is being developed with capability of mobile platform can be used by various users to report Disasters. In the event of crisis proper guidance needs to be provided to the general public to prevent casualties. The civilians should be directed to the refugee center for their own safety and to avoid chaos. Lack of proper medium for Government to communicate with the civilians in distress is the primary reason to why Government fails in crowd control. Often civilians are stranded in the event of crisis due to the ignorance of evacuation or rescue plans of the Government. They need to be directed to a safe location to avoid panic among themselves. The Technical virtuoso of mobile platform is exploited to provide personal information to public. Mobile platform is picked as it provides widespread personal reach to guide civilians in the affected region. The Mobile numbers of civilians in the crisis region are extracted from VLR's of local service provider. SMS Gateway is used to interface between the system and mobile network's SMSC (Short Message Service Center). The information is passed as text messages.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ACKNOWLEDGEMENT	ii
	ABSTRACT	iii
	TABLE OF CONTENTS	iv
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
I	INTRODUCTION	
	1.1 AIM OF THE PROJECT	2
	1.2 SCOPE OF THE PROJECT	3
II	LITERATURE SURVEY	4
III	SYSTEM ANALYSIS	
	3.1 EXISTING SYSTEM	12
	3.2 PROPOSED SYSTEM	13

IV	SYSTEM REQUIREMENTS	
	4.1 HARDWARE REQUIREMENTS	16
	4.2 SOFTWARE REQUIREMENTS	16
	4.3 HARDWARE DESCRIPTION	
	4.3.1 MOBILE DEVICE	17
	4.3.2 USB CONNECTOR	17
	4.4 SOFTWARE DESCRIPTION	
	4.4.1 ANDROID STUDIO	18
	4.4.2 ANDROID SDK	19
	4.4 .3 ANDROID ADB	20
	4.4.4 XAMP SERVER	21
V	PROJECT DESCRIPTION	
	5.1 SYSTEM ARCHITECTURE	23
	5.2 MODULE DESCRIPTION	
	5.2.1 USER REGISTRATION	25
	5.2.2 REPORT CRISIS	26
	5.2.3 AUTHENTICATE CRISIS AND SEND ALARM	26
VI	IMPLEMENTATION	
	6.1 SOURCE CODE	
	6.1.1 SOFTWARE MODULE	28

VII	TESTING	
	7.1 INSTALLATION	
	7.1.1 REQUIREMENTS	51
	7.1.2 INSTALL ANDROID STUDIO	51
	7.1.3 INSTALL XAMPP SERVER	51
	7.1.4 UPDATING ANDROID SDK	52
	7.2 TEST PROCEDURE	52
	7.3 SOFTWARE TESTING AND VALIDATION	
	7.3.1 UNIT TESTING	53
	7.3.2 INTEGRATION TESTING	53
	7.3.3 SYSTEM TESTING	53
	7.4 TESTING TACTICS	54
VIII	SCREENSHOTS	
	8.1 CLIENT SIDE	
	8.1.1 LAUNCHER	56
	8.1.2 REGISTRATION	56
	8.1.3 REGISTERING INTO DATABASE	57
	8.1.4 CRISIS REPORT PAGE	57
	8.2 SERVER SIDE	
	8.2.1 DATABASE STRUCTURE	58
	8.2.2 REGISTRED DATABASE ENTRY	58
	8.2.3 XAMPP CONTROL PANEL	59
	8.2.4 CRISIS DATABASE STRUCTURE	59

IX	CONCLUSION	61
	REFERENCES	63

LIST OF FIGURES

FIGURE NO	FIGURE NAME
3.1	Proposed System in Android OS
4.1	Mobile device with Android OS
4.2	Android Architecture
5.1	Overall Block Diagram
5.2	User Registration Block Diagram
5.3	Crisis authentication and alert sending process

LIST OF ABBREVIATIONS

CMAS	Crisis Management and Alarm System
GPRS	General Packet Radio Service
SMSC	Short Message Service Center
ADB	Android Debug Bridge
GPS	Global Positioning System
ADT	Android Development Toolkit
SDK	Software Development Kit
API	Application Programming Interface
IDE	Integrated Development Environment
SMS	Short Message Service
DMS	Disaster Management Server
WMP	Wireless Messaging Platform
USB	Universal Serial Bus
OGC	Open Geospatial Consortium
C2DM	Cloud to Device Messaging

INTRODUCTION

CHAPTER-I

INTRODUCTION

1.1 AIM OF THE PROJECT

Human population is under constant threat from disasters and man-made crisis. The nature of the disaster influences the kinds of threats that humans are exposed to by physical phenomena such as extreme weather, seismic, volcanic and human activities like terrorism and riots. In the event of any such events, the general public is not provided with sufficient guidance on how to overcome the situations. Lack of proper medium for government to communicate with the civilians in distress is the primary reason as to why the government fails in its competency. Often civilians are stranded in the event of crisis due to their ignorance of the evacuation or rescue plans of the government. They need to be directed to a safe location for their own safety. This will allow the affected civilians to recuperate and cope up with the occurring disaster. Clear and concise deployment of relief messages and groups are of paramount importance to avoid panic among civilians.

In the event of a crisis, this project proposes that it would be possible to provide proper guidance to the general public to prevent casualties through passing the information through their mobile devices. The civilians can be directed to a refuge center for their own safety and provide them with adequate nourishment. The most widely used mobile platform is used because this project could have a personal reach over a larger crowd of civilians in the affected region. And also this project provides the facility for an individual to report a crisis to the proper authorities which could save a lot of mishaps.

Lack of proper medium for government to communicate with the civilians in distress is the primary reason to why government fails to control the fatalities that occur. Often civilians are stranded in the event of crisis due to their ignorance of the evacuation or rescue plans of the government. This project aims at filling the gap and ensuring that there are no discrepancies between the information passed between the government and the people.

1.2 SCOPE OF THE PROJECT

Smart phone technologies and their applications have changed various aspects of human life. In the present scenario, the percentage of individuals that carry an android device has increased tremendously. So in the event of a crisis it is highly presumable that general public in distress could be highly benefitted if proper instruction were provided to them through their mobile devices. Though in most cases the government has implemented several early disaster warning schemes and training systems, it still fails to reach the civilians due to ineffective communication systems. This project attempts to fulfill the needs of socialization in the public community regarding the early disaster planning.

Finally it creates interaction opportunities between the government relief centers and the general public; care and support to the distressed civilians are introduced.

Here, we provide the information on the main aim and objectives, scope of the project, purpose of the project and an overall view of what all are implemented in this project. It also provides information about the basic project features, the benefits and constraints of the entire project.

LITERATURE SURVEY

CHAPTER II

LITERATURE SURVEY

2.1 IMPLEMENTATION OF LOCATION AWARENESS AND SHARING SYSTEM BASED ON GPS AND GPRS USING JEM2.

AUTHORS: Chandra A, Jain S, Qadeer M A

This document provides detail on the implementation of a location awareness system which gives the user's current location, sends this location using SMS (Short Message Service) plus sharing location with friends and family and view them on Google maps. Users can also take benefit of this application in emergency situations by using emergency feature of this application. To get the location coordinates, application is using GPS (Global Positioning System) as location provider. The application design has five parts: a mobile client, a web server, a database, GPS system and a map service. A mobile client which consists of a mobile and GPS receiver finds the location of the user to get aware of his location. In order to share this location the mobile client sends this location to the web server from where other users can get this location if they have the authentication provided by the user. The mobile client is implemented using J2ME which is one of the most promising software platforms for mobile devices. Sun representatives assert that 18 to 20 million mobile phones support the J2ME platform. Analysts predict that within the next few years, this technology will become omnipresent. According to Gartner Group estimates, in 2006, approximately 80 percent of mobile phones will support Java. The web server programming is done using PHP and database is maintained by MYSQL.

2.2 LOCATION BASED EARLY DISASTER WARNING AND EVACUATION SYSTEM ON MOBILE PHONES USING OPENSTREETMAP.

AUTHORS: Rahman K M, Alam T, Chowdhury M

Due to geographic position and climate change, Bangladesh is one of most vulnerable countries to natural disasters. The country also lacks effective disaster preparedness system to confront natural disasters. In addition, a tourist or a blind people may face difficulties in finding safe area or shelter place prior to the occurrence of natural disasters. For this reason, they have proposed a location based early disaster warning and evacuation system for both normal and blind people using OpenStreetMap (OSM). The system is implemented on android mobile phone because of the burgeoning growth of smart phones in Bangladesh. So, their system comprises a third-party server named Disaster Management Server (DMS), android device with our application installed on the user system. The local weather office updates the disaster (tsunami, cyclone or flood) data on DMS. Device user registers on Android Cloud to Device Messaging (C2DM) server to get automatic notification of upcoming disaster otherwise user gets manual notification. The user communicates with DMS to have updated data sending the current position obtained by GPS or network provider. The probable disaster affected area is determined by ray casting algorithm. When our application recognizes the user in probable disaster zone then application will disseminate visual and audio disaster warning and evacuation guideline including shortest path of shelter or safe zone on the map of the application. Evacuation progress is also tracked using DMS and national identification.

2.3 A RECONFIGURABLE OPEN GEOSMS MOBILE CLIENT APP DESIGN FOR ANDROID SMARTPHONES.

AUTHORS: Heo G, Genong Yu, LipingDj

A geo tagged Short Message Service (SMS), such as those following Open Geospatial Consortium (OGC) Open Geo SMS (OGS) standard, is useful for many applications that require location based services field work. In OGS specification, a geo tag is used as a hyperlink for retrieving detail information of given target at given location. This feature can give service providers to design more various and useful service scenarios, and these requirements make importantly to develop vendor specific Apps for realizing these service scenarios. A design pattern for OGS mobile client Apps will help to develop the Apps with relatively less effort. This paper focuses on a design for an OGS mobile client App and shows a Connector-Translator-Launcher (CTL) flow for smartphone environments. The CTL flow based mobile client App is based on Model-View-Controller (MVC) model. It applies pluggable modules for connecting geospatial services pointed by each geo tag and translating geospatial content retrieved from such services. The App displays details information specified by a geo tag following the OGS. The Android-based mobile client App was developed for OGC's OGC Web Services, Phase 8 (OWS-8). It was applied as a mobile tracking client in geospatial web service test bed scenarios for the Observation Fusion (Tracking) thread. Two pluggable modules were developed for the scenario application. One is a connector module for connecting to a track server, and another is a translator module for translating track information for displaying on map.

2.4 DEVELOPING AN EMAIL VIOLATION PREVENTION MODEL BASED ON WEB 3.0 TECHNOLOGY.

AUTHORS :MohDYusop, Zulkefli, Mohamad FirdausChe.

Email is one of the most important Internet applications for most computer users. The usage of email is increasing from time to time. However, together with this growth comes a variety of problems, such as increase of spam and the widely spread of computer worms via emails. The current scenario poses a challenge on ways to manage email efficiently, especially in avoiding the sending and receiving of spam email. Recent development in web technology such as Semantic Web provides an infrastructure that enables web pages, databases, and services to both consume and produce data on the web. Application developer can use this information to search, filter, and prepare information in new and exciting ways to assist the web user. With these features, Semantic Email, one of Semantic Web application, is believed to be able to help control spam by using semantic filtering and filing of email. This paper reports the current practice of email violation prevention method by Internet Service Providers in Malaysia, and later proposes a measure to prevent email violation in Malaysia through Web3.0 technology. The development of an email violation prevention model based on Web 3.0 elements signifies the theoretical contribution of this study.

2.5 REAL TIME GPS SOFTWARE RECEIVER WITH NEW FAST SIGNAL TRACKING METHOD.

AUTHORS :Sonowal N, YadayR,Kannan S.

The paper describes an approach of implementing software based GPS receiver acquisition and tracking for LI C/A signal on ADSP SHARC. The main objective of this paper is bring out fast GPS signal tracking method using which more number of channels(satellites) can be tracked on a fixed and limited resources DSP platform. Further, GPS acquisition, Carrier Pull-in and C/A Code prediction modules are discussed. The processing for the algorithms are done using sign bit of the ADC.A time domain based approach without early and late codes is used for Tracking GPS signal.

2.6 A SHORT MESSAGE SERVICE FOR CAMPUS WIDE INFORMATION DELIVERY.

AUTHORS : Mohammad M A, Norhayati A.

This paper is about software development that is based on short messaging service (SMS) system for delivering messages to students. The students' information is held on a database that can be accessed through graphical user interface (GUI). This software allows the lecturer to maintain a list of students with their respective mobile phone numbers and send message to them. The messages and thus the information must be able to be updated, edited, added or removed. Since this project is using Simple wire's wireless text-messaging platform, the message will be sent to the Simple wire's Wireless Messaging Platform (WMP) and then to the respective mobile phone. Simple wire acts as a gateway between the Internet and mobile device i.e. respective

Network Operator, e.g. Celcom, Maxis, TMTouch, Digi and Time Cell. The software is developed using Visual Basic 6.0 with query based on the Structured Query Language (SQL) and ActiveX Data Object 2.0 Library (ADO). The database is held on Microsoft Access 97. The interface between Simple wire's Wireless Message Protocol servers and an External Messaging Entity is developed using Simple wire ActiveX SMS Software Development Kit (SDK) 2.4.2 for Visual Basic.

Here gives the literature survey done during the building of CMAS application. It consists of three main applications already existing. Here, the comparison of these three applications is done in terms of the features and drawbacks with the CMAS application. This is important as it gives a clear picture of how much to the extent is the CMAS application applicable.

SYSTEM ANALYSIS

CHAPTER III

SYSTEM ANALYSIS

3.1 EXSISTING SYSTEM

The existing system does not support any specific feature that enables any form of notification or warning to be passed between the general public and the government. The Government can communicate with the general public only through social media like News. There is no available method by which a civilian can report a crisis immediately to the concerned authority. The country also lacks effective disaster preparedness system to confront natural disasters. In addition, a tourist or a blind people may face difficulties in finding safe area or shelter place prior to the occurrence of natural disasters.

DISADVANTAGES

- There is no personal means of communication between Government and general public.
- Mass Communication Causes unnecessary panic and chaos.
- People who are not accessible to mass communication are left stranded.
- DWAS involves huge maintenance cost.
- There is no available method by which a civilian can report a crisis immediately to both the concerned authority and alert concerned people.

3.2 PROPOSED SYSTEM

The technical virtuoso of the mobile platform is exploited to provide personal information to the public. In the event of a crisis, this project proposes that it would be possible to provide proper guidance to the general public to prevent casualties through passing the information through their mobile devices. The civilians can be directed to a refuge center for their own safety and provide them with adequate nourishment. As soon as a crisis is reported, the server system monitors all the visitor and home user number in the affected region. The mobile numbers of the civilians in the crisis region are extracted from the VLR's of local service providers. Then the SMS gateway is used to interface between the system and mobile network's SMSC (Short Message Service Center) to send the relief messages throughout the region to all users guiding the civilians to a secure and protected location.

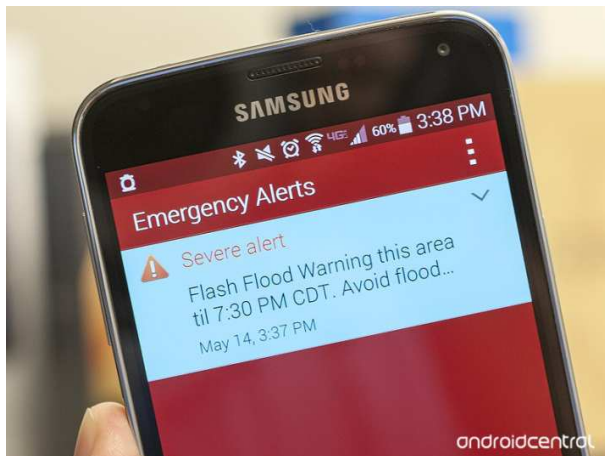


Figure 3.1 Proposed System in Android OS

The information is passed as text messages which are formed by the relief committee depending on the crisis in hand.

ADVANTAGES

- Provide a personal reach to pass information between the government and the public.
- Provide Maintenance free system
- Cost effective
- Make delivery of relief messages assured.

Here explains the working and the benefits of the existing system, definitions of the proposed project and the proposed application. It clearly gives the information about what are all provided additionally with the existing system to implement the proposed application. It also provides the list of all the features that are established upon the installation of the application into the mobile device.

SYSTEM REQUIREMENTS

CHAPTER VI

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- Personal Computer
- Mobile device.
- Mobile USB Connector

4.2 SOFTWARE REQUIREMENTS

- Android Studio
- Android SDK
- Android ADB
- Xampp server
- Notepad++
- Windows 7
- Php 5.6.3
- Mysql 5.6

4.3 HARDWARE DESCRIPTION

4.3.1 MOBILE

A mobile device is also referred to as hand held devices. Hand held device or hand held computer is a pint-sized computing device and usually come with a touch or non-touch display screen and sometimes, even a mini keyboard. The mobile device should support the Android Operating System for performing multi-tasking, good user interface and rich notifications.



Figure 4.1 Mobile device with Android OS

4.3.2 USB CONNECTOR

Universal Serial Bus (USB) is an industry standard that defines the cables, connectors and communication protocols used in a bus for connection, communication and power supply between computers and electronic devices. Connecting a USB device to a computer is simple -- you find the USB connector

on the back of your machine and plug the USB connector into it. If it's a new device, the operating system auto-detects it and asks for the driver disk. If the device has already been installed, the computer activates it and starts talking to it. USB devices can be connected and disconnected at any time. Many USB devices come with their own built-in cable, and the cable has an "A" connection on it. If not, then the device has a socket on it that accepts a USB "B" connector.

4.4 SOFTWARE DESCRIPTION

4.4.1. ANDROID STUDIO

Android Studio is the official IDE for Android application development, based on IntelliJ IDEA. On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Flexible Gradle-based build system
- Build variants and multiple apk file generation
- Code templates to help you build common app features
- Rich layout editor with support for drag and drop theme editing
- lint tools to catch performance, usability, version compatibility, and other problems
- ProGuard and app-signing capabilities
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

4.4.2 ANDROID SDK

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. A software development kit that enables developers to create applications for the Android platform. The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications. Applications are written using the Java programming language and run on Dalvik, a custom virtual machine designed for embedded use which runs on top of a Linux kernel.

FEATURES

- ✓ Application framework enabling reuse and replacement of components
- ✓ GSM Telephony (hardware dependent)
- ✓ EDGE, 3G, and Wi-Fi (hardware dependent)
- ✓ GPS, compass, and accelerometer (hardware dependent)
- ✓ Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Android Studio

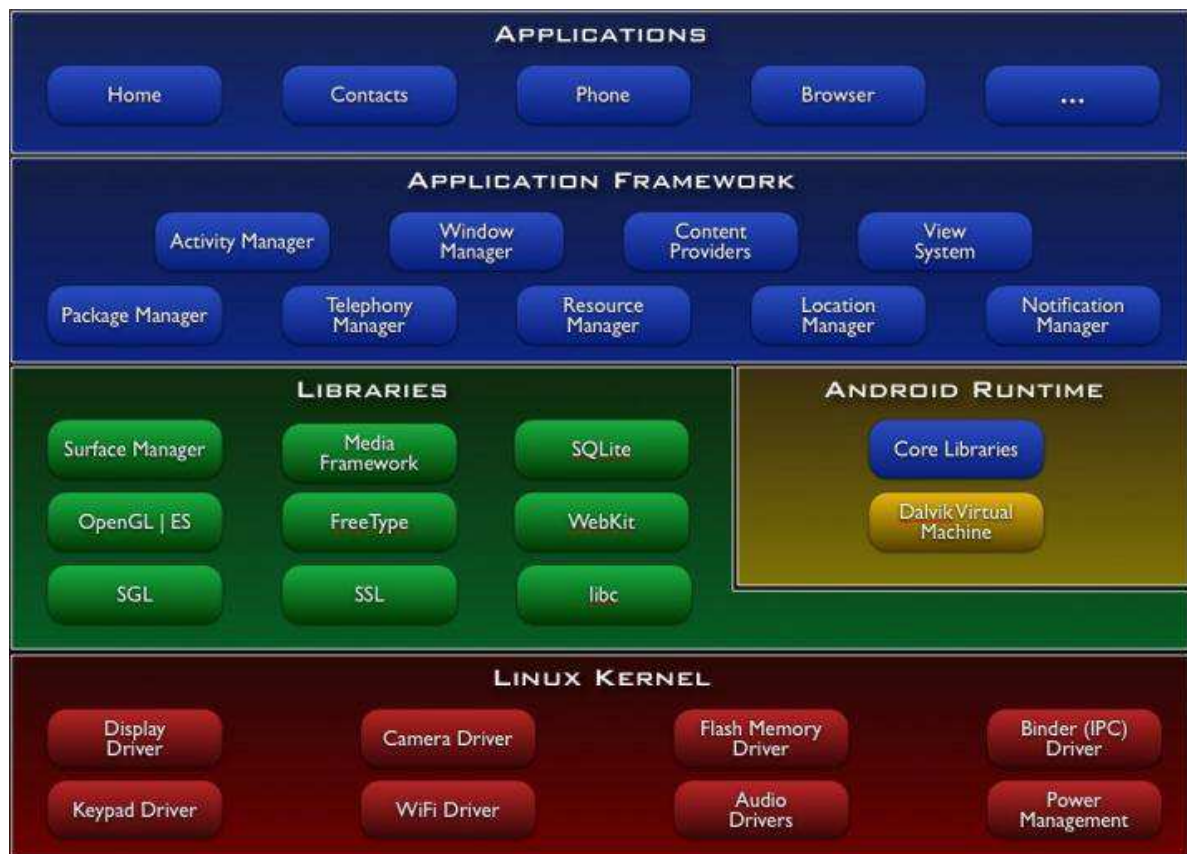


Figure 4.2 Android Architecture

4.4.2 ANDROID ADB

Android Debug Bridge (ADB) is a versatile command line tool that lets you communicate with an emulator instance or connected Android-powered device. It is a client-server program that includes three components:

A client, which runs on your development machine. You can invoke a client from a shell by issuing an adb command. Other Android tools such as the ADT plugin and DDMS also create adb clients.

A server, which runs as a background process on your development machine. The server manages communication between the client and the adb daemon running on an emulator or device. A daemon, which runs as a background process on each emulator or device instance.

4.4.3 XAMPP SERVER

Xampp version 3.2.1 is an easy to install Apache distribution containing MySQL, PHP and Perl. It contains: Apache, MySQL, PHP & PEAR, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, gdbm, zlib, expat, Sablotron, libxml, Ming, Webalizer, pdf class, ncurses, mod_perl, FreeTDS, gettext, mcrypt, mhash, eAccelerator, SQLite and IMAP C-Client.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress. It is also possible to connect to localhost via FTP with an HTML editor.

The default FTP user is "newuser", the default FTP password is "wampp". The default MySQL user is "root" while there is no default MySQL password.

PROJECT DESCRIPTION

CHAPTER V

PROJECT DESCRIPTION

5.1 SYSTEM ARCHITECTURE:

The System architecture diagram shows below provides the overall description of the project CMAS. The entire structure can be split into three major portions. Primarily, the project begins with each user registering the CMAS application into their android mobile devices. The registration provides an unique identification for each user corresponding to his device and location. This registration ensures that the registered user receives the early disaster warning and evacuation instruction properly.

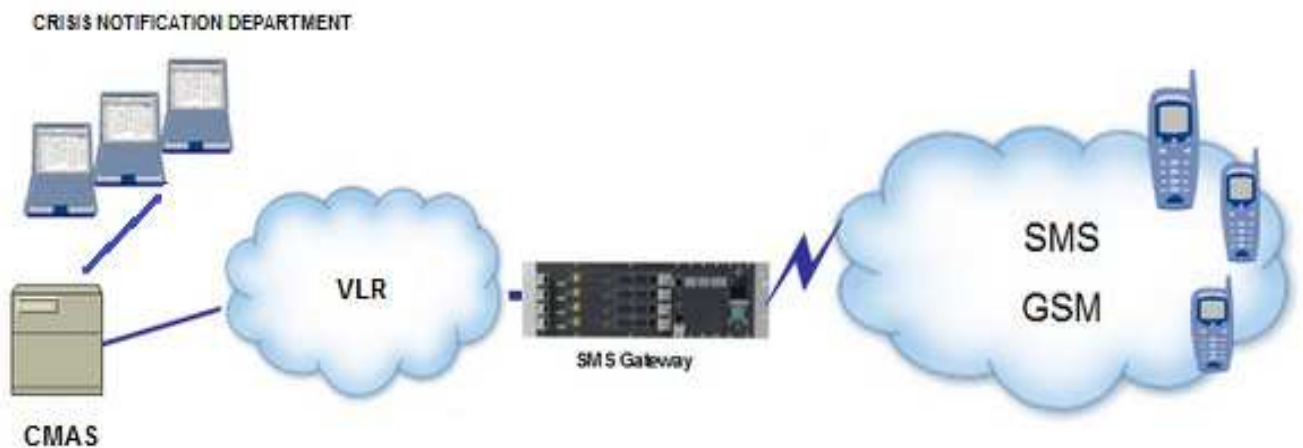


Figure 5.1 Overall Block Diagram

The second stage of the project is where the major distinction occurs regarding whether the user wishes to report a crisis or at the receiving end for crisis instructions. At this stage the user is provided with the facility to report any crisis currently occurring. In which case the responsible authorities receive the report and authenticate its validity. If the report was a spam, the user who sent is

put under severe scrutiny from the officials. The user personal information regarding the user name, phone number, address, location...etcare passed along with the report message sent. The higher officials at the relief centers could anytime view the user credentials. In case the report received turns out to be true, the relief centers are counted on to take the necessary actions such as putting together an immediate team of officials to think of solutions to successfully handle the crisis and minimizing the number of casualties. Appropriate messages depending upon the crisis is designed and the message is further edited to remove mistakes making it clear and concise. The finalized message is put into the server and forwarded to all the user who are currently present in the crisis region. This is possible by the use of sms gateways whose dedicated function is to do pass short messages to mobile devices across the region using the information of all the registered user mobile number.

Thus the architecture diagram provides the basic information regarding each module and the detailed description regarding each division of the architecture is provided in the module description as follows.

5.2 MODULE DESCRIPTION:

The project mainly contains of 3 modules, each of which enable this system enact a proper interface between the public and the government relief centers. The modules are given as follows:

- User registration module
- Report Crisis
- Authenticate Crisis and send Alarm

5.2.1 User registration module:

This module is used to register the mobile number of the user to the database. It uses php as middle tier to connect the database. The mainactivity class contains EditText element to get the user's mobile number and a button that triggers register class's execute method.

Register class is used to display the progress window while registration is in progress. JsonParser class is used to create database connection and register user's mobile number. It receives the response object from the database as json object it parses the object to retrieve the success of registration.

Cofig.php used to set up database credentials.

Connect.php used to connect to database

Register.php used to insert into database

MODULE 1: USER REGISTRATION

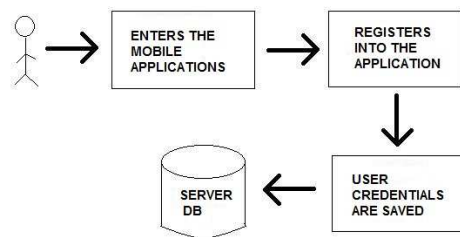


Figure 5.2 User Registration Block Diagram

5.2.2 Report Crisis:

In the event of any crisis, this module will enable the user to report the event to a nearby government facility. The Government facility in turn can make the necessary steps to handle the situation. These action are accomplished by booting an application window in which the user can enter the crisis occurred. The Gps present in the device can notify the center on the coordinates of the device. DisplayActivity.java contains editText element to get crisis from user GPS is used to detect the location of user and is used to send crisis report along with the location and mobile number to the webserver.

5.2.3 Authenticate Crisis and Send Alarm:

The crisis reported by user is verified and authenticated for its validity. Then the registered mobiles are sent a push notification to activate the app and get the location of the registered users. These users are in turn reported of the crisis in the affected region through alarm message which are sent via sms gateways.

MODULE 3: AUTHENTICATE CRISIS AND SEND ALARM

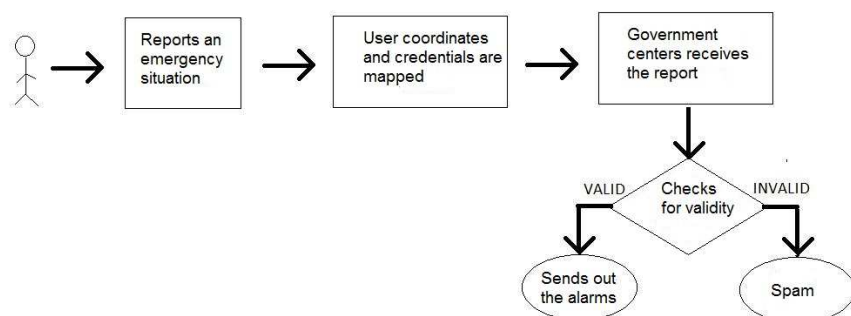


Figure 5.3 Crisis authentication and alert sending process

IMPLEMENTATION

CHAPTER VI

IMPLEMENTATION

6.1.SOURCE CODE

6.1.1 SOFTWARE MODULES

The User Interface that registers user mobile number to XAMPP database through the android SDK.

USER INTERFACE

JsonParser.java

```
package com.gpsapp.shg.gpsapplicationtest.register;

import android.util.Log;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;

/**
 * Created by HP on 01-03-2015.
 */
public class JSONParser
{
    static InputStream inputStream=null;
    static JSONObject jsonObject=null;
    static String json=" ";
    public JSONParser()
    {
    }
}
```

```

public JSONObject makeHttpRequest(String url, String method,
    List<NameValuePair> parameters)
{
    try
    {
        if (method == "POST")
        {
            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);
            httpPost.setEntity(new UrlEncodedFormEntity(parameters));

            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            InputStream inputStream = httpEntity.getContent();
        }
        elseif (method == "GET")
        {
            DefaultHttpClient httpClient = new DefaultHttpClient();
            String paramString = URLEncodedUtils.format(parameters, "utf-
8");
            url += "?" + paramString;
            HttpGet httpGet = new HttpGet(url);

            HttpResponse httpResponse = httpClient.execute(httpGet);
            HttpEntity httpEntity = httpResponse.getEntity();
            InputStream inputStream = httpEntity.getContent();
        }
        catch (UnsupportedEncodingException e)
        {
            e.printStackTrace();
        }
        catch (ClientProtocolException e)
        {
            e.printStackTrace();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
    try
    {
        BufferedReader reader = new BufferedReader(new
            InputStreamReader(inputStream, "iso-8859-1"), 8);
        StringBuilder stringBuilder = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null)
        {
            stringBuilder.append(line + "\n");
        }
        inputStream.close();
        json = stringBuilder.toString();
    }
}

```

```

        catch(UnsupportedEncodingException e)
        {
            e.printStackTrace();
        }
        catch(IOException e)
        {
            e.printStackTrace();
        }
        try
        {
            Log.e("My Error", "JSON init started");
            jsonObject=newJSONObject(json);
            Log.e("My Error", "JSON initialised");
        }
        catch(JSONException e)
        {
            Log.e("My Error line 110", e.getLocalizedMessage());
            e.printStackTrace();
        }
        return jsonObject;
    }
}

```

UserInterfaceActivity.java

```

package com.gpsapp.shg.gpsapplicationtest;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.location.Location;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.gpsapp.shg.gpsapplicationtest.register.CMAS;
import com.gpsapp.shg.gpsapplicationtest.register.DBOperation;
import com.gpsapp.shg.gpsapplicationtest.register.GPSTracker;
import com.gpsapp.shg.gpsapplicationtest.register.JSONParser;

import org.apache.http.NameValuePair;

```



```

import org.apache.http.message.BasicNameValuePair;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends ActionBarActivity implements View.OnClickListener {
    public final static String EXTRA_MESSAGE = "com.gpsapp.shg.MESSAGE";
    private ProgressDialog progressDialog;
    JSONParser jsonParser = new JSONParser();
    private int success;
    private DBOperation dbOperation = DBOperation.getInstance();
    private static Context mainContext;
    private String message, latitude, longitude;
    private EditText textMobileNumber;
    private Button btnRegister;
    private static String
        registerUrl = "http://192.168.1.6/GPSJSON/register.php";
    private static final String successTag = "success";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        {
            final String PREFS_NAME = "PrefsFile";
            SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
            if (!dbOperation.isUserRegistered())
            {
                //here the app is being launched for first time, launch your screen

                settings.edit().putBoolean("app_first_time", false).commit();
            }
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            textMobileNumber = (EditText) findViewById(R.id.edit_message);
            btnRegister = (Button) findViewById(R.id.register);
        }
        else
        {
            super.onCreate(savedInstanceState);
            //setContentView(R.layout.activity_display_message);
            Intent intent = new Intent(MainActivity.this,
                DisplayMessageActivity.class);
            startActivity(intent);
            finish();
        }
    }
    @Override
    public void onClick(View view) {
        {
            if (view.getId() == R.id.register)
            {
                mainContext = this;
                new RegisterMobile().execute();
            }
        }
    }
}

```

```
}
}
```

```

@Override
public boolean onCreateOptionsMenu(Menu menu){
    // Inflate the menu; this adds items to the action bar if it is present.

    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.

    int id = item.getItemId();
    //noinspectionSimplifiableIfStatement
    if(id == R.id.action_settings){
        return true;
    }

    return super.onOptionsItemSelected(item);
}

public void sendMessage(View view)
{
    if(view.getId() == R.id.register)
    {
        mContext = this;
        Location location = new GPSTracker().getLocation();
        latitude = Double.toString(location.getLatitude());
        longitude = Double.toString(location.getLongitude());
        dbOperation.registerUser(textMobileNumber.getText().
            toString(), latitude, longitude);
        new RegisterMobile().execute();
    }
}

class RegisterMobile extends AsyncTask<String, String, String>
{
    String mobileNumber = textMobileNumber.getText().toString();
    @Override
    protected void onPreExecute()
    {
        super.onPreExecute();
        progressDialog = new ProgressDialog(MainActivity.this);
        progressDialog.setMessage("Registering New Mobile
            Number (" + mobileNumber + ") ...");
        progressDialog.setIndeterminate(false);
        progressDialog.setCancelable(true);
        progressDialog.show();
    }
    protected String doInBackground(String... args)

```

```

{
    List<NameValuePair> parameters=newArrayList<NameValuePair>();
parameters.add(new

        BasicNameValuePair("MobileNumber",mobileNumber));
parameters.add(newBasicNameValuePair("latitude",latitude));


        parameters.add(newBasicNameValuePair("longitude",longitude));
JSONObjectjsonObject=
        jsonParser.makeHttpRequest(registerUrl,"GET",parameters);
try
{
    success=jsonObject.getInt(successTag);
    message=jsonObject.getString("message");
    Log.e("My Error",message+success);
}
catch(JSONException e)
{
    e.printStackTrace();
}
returnnull;
}
protectedvoidonPostExecute(String url)
{
    progressDialog.dismiss();
    if(success==1)
    {
        Log.e("My Error","Success");
        Toast.makeText(getApplicationContext(),"New Mobile Number
            Saved....",Toast.LENGTH_LONG).show();
            Intent intent=newIntent(mainContext,
                DisplayMessageActivity.class);
        EditTexteditText=(EditText)findViewById(R.id.edit_message);
            String message =editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
        finish();
    }
    else
    {
        Toast.makeText(getApplicationContext(),"New Mobile Number Failed
            To Save....",Toast.LENGTH_LONG).show();
    }
}
}
}
}
}

```

DBAccess.java

```
package com.gpsapp.shg.gpsapplicationtest.register;

/**
 * Created by HP on 08-03-2015.
 */
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DBAccess extends SQLiteOpenHelper
{
    /*      table for storing the user details      */
    public static final String USER_TABLE="userTable";
    public static final String USER_TABLE_COLUMN_1_MOBILE="mobilenumber";
    public static final String USER_TABLE_COLUMN_2_LATITUDE="latitude";
    public static final String USER_TABLE_COLUMN_3_LONGITUDE="longitude";
    /*      database file name and version      */
    private static final String DATABASE_NAME="CMAS.db";
    private static final int DATABASE_VERSION=1;

    /*create table sql query string for USER_TABLE */
    public static final String CREATE_USER_TABLE="create table "+USER_TABLE+"
        (" +USER_TABLE_COLUMN_1_MOBILE+" text primary key,"+
        USER_TABLE_COLUMN_2_LATITUDE+"
        text,"+USER_TABLE_COLUMN_3_LONGITUDE+" text);"
    public DBAccess(Context context)
    {
        super(context,DATABASE_NAME,null,DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase)
    {
        sqLiteDatabase.execSQL(CREATE_USER_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase,int oldVersion,int
        newVersion)
    {
        Log.w(DBAccess.class.getName(),"Upgrading database from the old version
            "+oldVersion+
            " to the new version "+newVersion+" ,which will destroy the old data");
        sqLiteDatabase.execSQL("drop table if exists "+USER_TABLE);
    }
}
```

DBOperation.java

```
package com.gpsapp.shg.gpsapplicationtest.register;

import android.content.ContentResolver;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;

import java.math.BigInteger;

/**
 * Created by HP on 08-03-2015.
 */

public class DBOperation
{
    private static SQLiteDatabase database;
    private static DBAccess dbAccess;
    private static DBOperation dbOperation = new DBOperation(CMAS.getContext());
    private ContentResolver contentResolver;

    public DBOperation(Context context)
    {
        dbAccess = new DBAccess(context);
        contentResolver = context.getContentResolver();
    }

    /**
     * @return the instance of the DBOperation class
     */
    public static DBOperation getInstance()
    {
        return dbOperation;
    }

    /**
     * @return the instance of SQLiteDatabase
     */
    public static SQLiteDatabase getDatabase()
    {
        return database;
    }

    /**
     * Opens the database connection
     */
    public void openConnection()
    {
        if (database == null || !database.isOpen())
        {
            database = dbAccess.getWritableDatabase();
        }
    }
}
```

```

}
/*
    * Closes the database connection
    */
public void closeConnection()
{
    database.close();
}

/*
    * @param mobile number of the user
    * Inserts the user details into the USER_TABLE
    */

public void registerUser(String mobileNumber, String latitude, String longitude)
{
    try
    {
        openConnection();
        ContentValues values = new ContentValues();

        values.put(DBAccess.USER_TABLE_COLUMN_1_MOBILE, mobileNumber);
        values.put(DBAccess.USER_TABLE_COLUMN_2_LATITUDE, latitude);
        values.put(DBAccess.USER_TABLE_COLUMN_3_LONGITUDE, longitude);
        database.insert(DBAccess.USER_TABLE, null, values);
    }
    finally
    {
        //closeConnection();
    }
}

/*
    * @return true if there is any user registered
    * @return false if there is no user registered
    */

public boolean isUserRegistered()
{
    openConnection();
    boolean operationResult = false;
    Cursor cursor = null;
    try
    {
        cursor = database.rawQuery("select
                                   " + DBAccess.USER_TABLE_COLUMN_1_MOBILE + " from
                                   " + DBAccess.USER_TABLE,
                                   null);
        if (cursor.moveToNext())
        {
            operationResult = true;
        }
        else
        {
            operationResult = false;
        }
    }
}

```

```

    }
    }
finally
{
    if(cursor!=null)
    {
        cursor.close();
        //closeConnection();
    }
}
return operationResult;
}

    /*
    * @return the mobile number of the user
    */

public String getMobileNumber()
{
    String mobileNumber=" ";
    Cursor cursor=null;
try
{
    openConnection();

        cursor=database.rawQuery("select
        "+DBAccess.USER_TABLE_COLUMN_1_MOBILE+" from
        "+DBAccess.USER_TABLE,null);
if(cursor.moveToNext())
{
mobileNumber=cursor.getString(0);
}
else
{
mobileNumber=null;
}
}
finally
{
    if(cursor!=null)
    {
        cursor.close();
    }
    //closeConnection();
}
return mobileNumber;
}
public String getLatitude()
{
    String latitude=" ";
    Cursor cursor=null;
try
{
    openConnection();

```

```

cursor=database.rawQuery("select
                        "+DBAccess.USER_TABLE_COLUMN_2_LATITUDE+" from
                        "+DBAccess.USER_TABLE,null);

if(cursor.moveToNext())
{
latitude=cursor.getString(0);
}
else
{
latitude=null;
}
}
finally
{
if(cursor!=null)
{
cursor.close();
}
//closeConnection();
}
return latitude;
}

    public String getLongitude()
    {
        String longitude="";
        Cursor cursor=null;
    try
    {
openConnection();
cursor=database.rawQuery("select
                        "+DBAccess.USER_TABLE_COLUMN_3_LONGITUDE+"
                        "+DBAccess.USER_TABLE,null);
from
if(cursor.moveToNext())
{
longitude=cursor.getString(0);
}
else
{
longitude=null;
}
}
finally
{
if(cursor!=null)
{
cursor.close();
}
//closeConnection();
}
return longitude;
}

    /*
    * @param latitude of location
    * @param longitude of location
    * Updates location of the user

```



```

        */

publicvoidupdateUserLocation(String latitude,String longitude)
{
    try
    {
        openConnection();
        ContentValues values=newContentValues();
        values.put(DBAccess.USER_TABLE_COLUMN_2_LATITUDE, latitude);
        values.put(DBAccess.USER_TABLE_COLUMN_3_LONGITUDE,longitude);
        Log.e("DataBaseError","Test");
        database.update(DBAccess.USER_TABLE,values,null,null);
    }
    finally
    {
        //closeConnection();
    }
}
/*
    * @return true if the database is opened
    * @return false if the database is closed
    */

```

```

publicbooleanisOpen()
{
    if(database.isOpen())
    {
        returntrue;
    }
    else
    {
        returnfalse;
    }
}
}

```

CMAS.java

```

packagecom.gpsapp.shg.gpsapplicationtest.register;

importandroid.app.Application;
importandroid.content.Context;

/**
 * Created by HP on 08-03-2015.
 */

publicclass CMAS extends Application
{
    privatestatic Context context;

    @Override
    publicvoidonCreate()
    {
        super.onCreate();
        context=getApplicationContext();
    }
}

```

```

}

public static Context getContext()
{
    return context;
}
}

```

GPSTracker.java

```

package com.gpsapp.shg.gpsapplicationtest.register;

import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;

import android.os.IBinder;
import android.provider.Settings;
import android.util.Log;

/**
 * Created by HP on 08-03-2015.
 */

public class GPSTracker extends Service implements LocationListener
{
    private Location location;
    private final Context context;
    private boolean isGPSEnabled, isNetworkEnabled, canGetLocation;
    private double latitude, longitude;
    private DBOperation dbOperation = DBOperation.getInstance();
    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 1000 * 10; // 10 km
    private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 30;
    protected LocationManager locationManager;
    public GPSTracker()
    {
        context = CMAS.getContext();
        getLocation();
    }
    public Location getLocation()
    {
        try
        {
            locationManager = (LocationManager)
                context.getSystemService(LOCATION_SERVICE);
            isGPSEnabled =

```

```

        locationManager.isEnabled(LocationManager.GPS_PROVIDER);
isNetworkEnabled=locationManager.

        locationManager.isEnabled(LocationManager.NETWORK_PROVIDER);
if(isNetworkEnabled)
{
    canGetLocation=true;
        locationManager.
            requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
MIN_TIME_BW_UPDATES,MIN_DISTANCE_CHANGE_FOR_UPDATES,this);
Log.d("Location","Network Location");
if(locationManager!=null)
{
    location=locationManager.
        getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
}
if(location !=null)
{
    latitude=location.getLatitude();
    longitude=location.getLongitude();
}
}
elseif(isGPSEnabled)
{
    canGetLocation=true;

        locationManager.
            requestLocationUpdates(LocationManager.GPS_PROVIDER,MIN_TIME_B
W_UPDATES,MIN_DISTANCE_CHANGE_FOR_UPDATES,this);
Log.d("Location","GPS Location");
if(locationManager!=null)
{
    location=locationManager.
        getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
}
if(location !=null)
{
    latitude=location.getLatitude();
    longitude=location.getLongitude();
}
}
else
{
    showSettingsAlert();
}
}
catch(Exception e)
{
    e.printStackTrace();
}
return location;
}

@Override
publicvoidonLocationChanged(Location location)

```

```

{
latitude=location.getLatitude();
longitude=location.getLongitude();
dbOperation.
updateUserLocation(Double.toString(latitude),Double.toString(longitud
e));
}

@Override
publicvoidonProviderDisabled(String provider){

}

@Override
publicvoidonProviderEnabled(String provider){

}

@Override
publicvoidonStatusChanged(String provider,int status, Bundle extras){

}

@Override
publicIBinderonBind(Intent arg0){
returnnull;
}
publicvoidshowSettingsAlert()
{
AlertDialog.BuilderalertDialog=newAlertDialog.Builder(context);

        alertDialog.setTitle("GPS settings");
alertDialog.setMessage("GPS is not enabled. Do you want to go to settings
        menu?");
alertDialog.setPositiveButton("Settings",new
        DialogInterface.OnClickListener()
        {
publicvoidonClick(DialogInterfacedialog,int which)
        {
                Intent intent=new

                Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
context.startActivity(intent);
        }
        });
alertDialog.setNegativeButton("Cancel",new
        DialogInterface.OnClickListener()
        {
publicvoidonClick(DialogInterface dialog,int which)
        {
dialog.cancel();
        }
        });
alertDialog.show();
}
publicvoidstopUsingGPS()
{
if(locationManager!=null)
{

```

```

locationManager.removeUpdates(GPSTracker.this);
}
}

}

```

ReportCrisis.java

```

package com.gpsapp.shg.gpsapplicationtest;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.gpsapp.shg.gpsapplicationtest.register.DBOperation;
import com.gpsapp.shg.gpsapplicationtest.register.GPSTracker;
import com.gpsapp.shg.gpsapplicationtest.register.JSONParser;
import org.apache.http.NameValuePair;

import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;

public class DisplayMessageActivity extends ActionBarActivity implements View.OnClickListener {
    {
        public final static String EXTRA_MESSAGE = "com.gpsapp.shg.MESSAGE";
        private ProgressDialog progressDialog;
        JSONParser jsonParser = new JSONParser();
        private int success;
        private String message;
        private EditText textCrisis;
        private static String
            registerUrl = "http://192.168.1.6/GPSJSON/sendCrisis.php";
        private static final String successTag = "success";
        private DBOperation dbOperation = DBOperation.getInstance();
        private static Context mainContext;

        @Override
        protected void onCreate(Bundle savedInstanceState)

```

```

{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_display_message);
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.

    int id = item.getItemId();
    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings)
    {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

@Override
public void onClick(View view)
{
    if (view.getId() == R.id.register)
    {
        mContext = this;
        Log.e("My Error", "Entered click");
        textCrisis = (EditText) findViewById(R.id.edit_crisis);
        new SendCrisis().execute();
    }
}

public void sendCrisis(View view)
{
    if (view.getId() == R.id.send)
    {
        mContext = this;
        Log.e("My Error", "Entered click");
        textCrisis = (EditText) findViewById(R.id.edit_crisis);
        new GPSTracker().getLocation();
        new SendCrisis().execute();
    }
}

class SendCrisis extends AsyncTask<String, String, String>
{
    String mobileNumber = dbOperation.getMobileNumber();
    String crisis = textCrisis.getText().toString();
    String latitude = dbOperation.getLatitude();
    String longitude = dbOperation.getLongitude();
    @Override
    protected void onPreExecute()
    {
        super.onPreExecute();
        progressDialog = new ProgressDialog(DisplayMessageActivity.this);
        progressDialog.setMessage("Sending Crisis...");
    }
}

```

```

progressDialog.setIndeterminate(false);
progressDialog.setCancelable(true);
progressDialog.show();
}
protected String doInBackground(String...args)
{
    List<NameValuePair> parameters=newArrayList<NameValuePair>();
parameters.add(new

    BasicNameValuePair("MobileNumber",mobileNumber));
parameters.add(newBasicNameValuePair("crisisMessage",crisis));
parameters.add(newBasicNameValuePair("latitude",latitude));
parameters.add(newBasicNameValuePair("longitude",longitude));
JSONObjectjsonObject=
    jsonParser.makeHttpRequest(registerUrl,"GET",parameters);
try
{
success=jsonObject.getInt(successTag);
message=jsonObject.getString("message");
Log.e("My Error", message + success);
}
catch(JSONException e)
{
e.printStackTrace();
}
returnnull;
}
protectedvoidonPostExecute(String url)
{
progressDialog.dismiss();
if(success==1)
{
Log.e("My Error", "Success");
Toast.makeText(getApplicationContext(),"Crisis Sent...",
    Toast.LENGTH_LONG).show();
    Intent intent=newIntent(mainContext,
        DisplayMessageActivity.class);
startActivity(intent);
finish();
}
else
{
Toast.makeText(getApplicationContext(),"Sending Crisis
    Failed...",Toast.LENGTH_LONG).show();
}
}
}
}
}
}

```

Config.php

```
<?php
/* * All database connection variables */
define('DB_USER','root');
define('DB_PASSWORD','');
define('DB_DATABASE','CMAS');// database name
define('DB_SERVER','localhost');// db server
?>
```

Connect.php

```
<?php
/**
 * A class file to connect to database
 */
class DB_CONNECT
{
    // constructor
    function __construct()
    {
        // connecting to database
        $this->connect();
    }
    // destructor
    function __destruct()
    {
        // closing db connection
        $this->close();
    }
}
/**
 * Function to connect with database
 */

function connect()
{
    // import database connection variables

    require_once __DIR__ . '/config.php';
    // Connecting to mysql database

    $con=mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD)or

        die(mysql_error());
    // Selecing database
    $db=mysql_select_db(DB_DATABASE)or die(mysql_error())or
        die(mysql_error());
    // returing connection cursor
    return $con;
}

/**
```



```

        * Function to close db connection
        */

        function close()
        {
// closing db connection
mysql_close();
}
}
?>

```

Register.php

```

<?php//insertnew.php
/*
 * Following code will create a new product row
 * All product details are read from HTTP GET Request
 */
// array for JSON response

$response=array();

// check for required fields

if(isset($_GET["MobileNumber"]))
{
    $MobileNumber=$_GET['MobileNumber'];
    $latitude=$_GET['latitude'];
    $longitude=$_GET['longitude'];
    // include db connect class
    require_once__DIR__.'connect.php';
    // connecting to db
    $db=new DB_CONNECT();
    // mysql inserting a new row
    $result=mysql_query("INSERT INTO
    GPSTLocation(MobileNumber,latitude,longitude)
    VALUES(' $MobileNumber',' $latitude',' $longitude')");
    // check if row inserted or not
    if($result)
    {
        // successfully inserted into database
        $response["success"]=1;
        $response["message"]="new Mobile Number saved....";
        // echoing JSON response
        echojson_encode($response);
    }
    else
    {
        // failed to insert row
        $response["success"]=0;
        $response["message"]="Oops! An error occurred.";

        // echoing JSON response
        echojson_encode($response);
    }
}

```

```

else
{
// required field is missing
$response["success"]=0;
$response["message"]="Required field(s) is missing";
// echoing JSON response
echo json_encode($response);
}
?>

```

sendCrisis.php

```

<?php

/*
 * Following code will create a new product row
 */
// array for JSON response

$response=array();
// check for required fields
if(isset($_GET["MobileNumber"])&&isset($_GET["crisisMessage"]))
{
    $MobileNumber=$_GET['MobileNumber'];
    $crisisMessage=$_GET['crisisMessage'];
    $latitude=$_GET['latitude'];
    $longitude=$_GET['longitude'];
    // include db connect class
    require_once __DIR__ . '/connect.php';
    // connecting to db
    $db=new DB_CONNECT();
    // mysql inserting a new row
    $result=mysql_query("INSERT INTO
        usercrisis(MobileNumber,crisis,latitude,longitude)
VALUES (' $MobileNumber', ' $crisisMessage', ' $latitude', ' $longitude')");
    // check if row inserted or not
    if($result)
    {
        // successfully inserted into database
        $response["success"]=1;
        $response["message"]="Crisis sent....";
        // echoing JSON response
        echo json_encode($response);
    }
    else
    {
        // failed to insert row
        $response["success"]=0;
        $response["message"]="Oops! An error occurred.";
        // echoing JSON response
        echo json_encode($response);
    }
}

else
{
// required field is missing

```

```
$response["success"]=0;
$response["message"]="Required field(s) is missing";
// echoing JSON response
echo json_encode($response);
}
```

?>

TESTING

CHAPTER -VII

INSTALLATION AND TESTING:

7.1 INSTALLATION

In order to basically use the software portion of CMAS we need to use the Xampp Server and the Android Studio along with Android ADB system files.

7.1.1 REQUIREMENTS

Development of CMAS can be viewed on a reasonably sized mobile device. For a nice experience a modern mobile device is recommended. A best configuration is Android version of 4 and above with Java Term software with Android Studio.

7.1.2 INSTALL ANDROID STUDIO

Android Studio is a freeware available on the internet which can be used to run the particular android application and can act as simulator to test run on the device. It also provides the feature to connect it directly to the particular database and hence can be easily used at any part of the program.

7.1.3 INSTALL XAMPP SERVER 3.2.1:

Xampp Server 3.2.1 is also a freeware available on the web which can be used as a database for any application that we create. The database is used to store all the details of the users who register with the CMAS application. But while installation we should ensure we run it in “Administrator” mode. Every time we start the server we should choose “Run as Administrator” option.

7.1.4 UPDATING ANDROID SDK:

SDK Tools is a downloadable component for the Android SDK. It includes the complete set of development and debugging tools for the Android SDK. It is included with [Android Studio](#). If you are already using the SDK and you want to update to the latest version of the SDK Tools, use the [SDK Manager](#) to get the update. Use the Android Studio update manager via Help-> Install new software and start the android to install to install all available components.

7.2 TEST PROCEDURE

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in it. If it is not detected before coding commences, the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough. Testing is one of the important steps in the software development phase.

7.3 SOFTWARE TESTING AND VALIDATION

It explains the various testing methods incorporated in the application to overcome the errors.

7.3.1 UNIT TESTING

In computer programming, unit testing is a software design and development method where the programmer gains confidence that individual units of source code are fit for use. Here we have tested the individual modules of our project using android studio emulator.

7.3.2 INTEGRATION TESTING

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

7.3.3 VERIFICATION AND VALIDATION

Validation is the process of ensuring that a program operates on clean, correct and useful data. We have verified with all the possible types of congestion. We also validated the system without any errors.

7.3.4 SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

7.4 TESTING TACTICS

7.4.1.BLACK BOX TESTING

Black box testing is the testing without knowledge of the internal workings of the item being tested.

7.4.1.WHITE BOX TESTING

White box testing uses an internal perspective of the system to design test cases based on the internal structure.

7.4.3.BASIS PATH TESTING

A testing mechanism proposed by McCabe. Aim is to derive a logical complexity measure of a procedural design and use this as a guide for defining a basic set of execution paths.

7.5CONTROL STRUCTURE TESTING

Control structure is a group of white- box testing methods.

1. Branch Testing
2. Condition Testing
3. Data Flow Testing
4. Loop Testing

1. Branch Testing (Decision Testing): For every decision, each branch needs to be executed at least once.

2. Condition Testing: Condition Testing is a test construction method that focuses on exercising the logical conditions in a program module.

3. Loop Testing: It will test the loops such as Simple Loops, Nested loops, concatenated Loops, Unstructured loops.

SCREENSHOTS

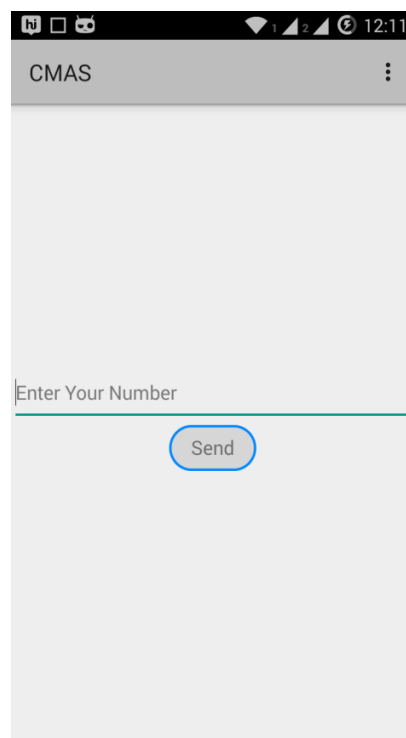
CHAPTER VIII SCREENSHOTS

8.1. CLIENT SIDE

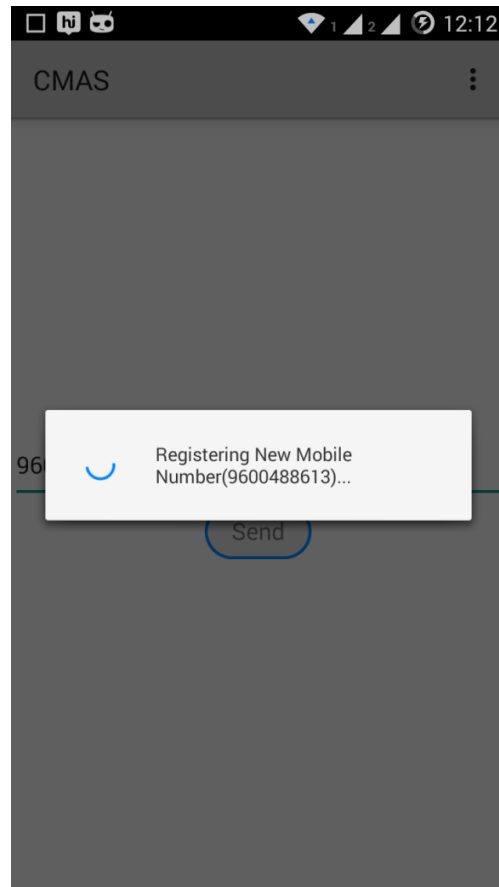
8.1.1. Launcher



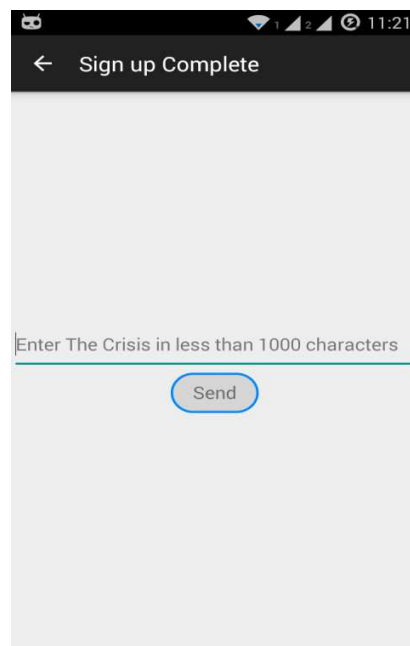
8.1.2. Registration



8.1.3. Registering into the Database:



8.1.4. Crisis Report Page:



8.2. SERVER SIDE

8.2.1. DatabaseStructure:

The screenshot shows the phpMyAdmin interface. The left sidebar displays the database structure tree with 'cmas' selected. The main panel shows the 'Structure' tab for the 'gpslocation' table. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	MobileNumber	bigint(10)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	Latitude	double			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	Longitude	double			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Below the table structure, there is an 'Add' button and a 'Go' button. The 'Information' tab is selected, showing space usage and row statistics.

Space usage		Row statistics	
Data	16 KiB	Format	Compact
Index	0 B	Collation	latin1_swedish_ci
Total	16 KiB	Creation	Mar 08, 2015 at 11:32 PM

8.2.2. Registered Database Entry:

The screenshot shows the phpMyAdmin interface. The left sidebar displays the database structure tree with 'cmas' selected. The main panel shows the 'SQL' tab for the 'gpslocation' table. The SQL query is:

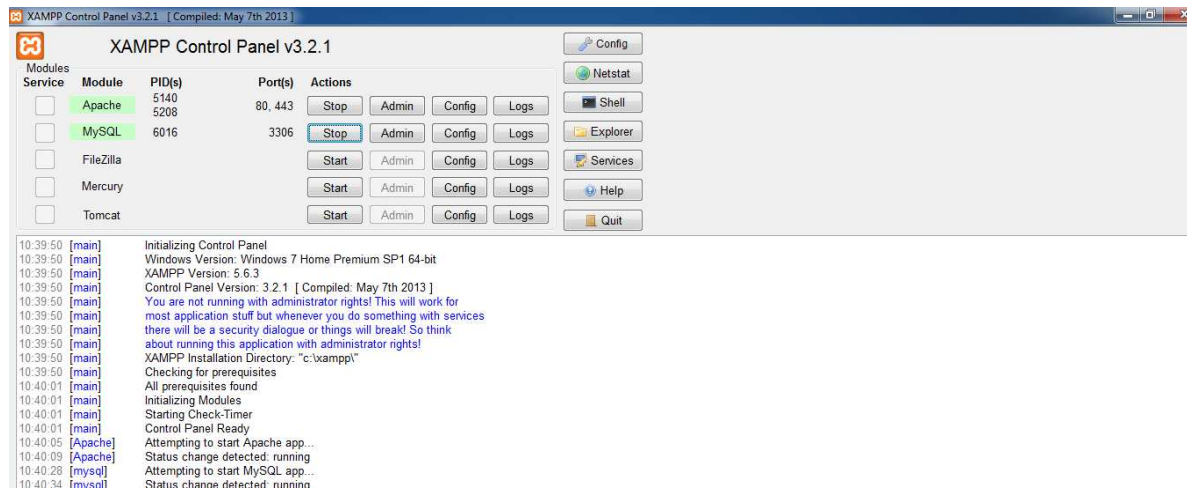
```
SELECT * FROM 'gpslocation' WHERE 1
```

The query results show 1 row. The row data is:

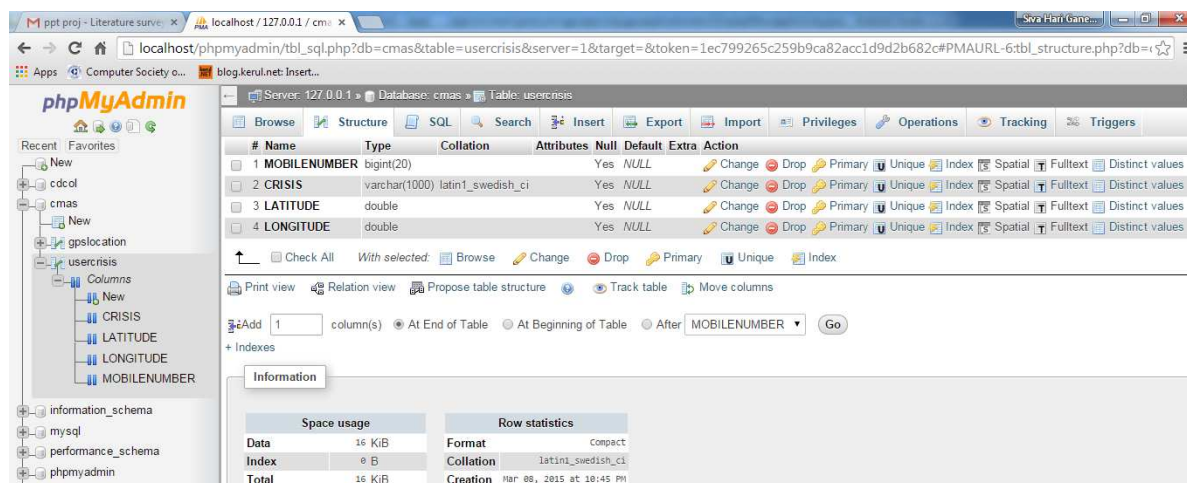
MobileNumber	Latitude	Longitude
9600488613	12.92599	80.1598751

8.2.3. Xampp Control

Panel:



8.2.4. CrisisDatabaseStructure:



8.2.5. Crisis Database Entry:

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a tree view of databases, including 'cdcol', 'cmas', 'gpslocation', 'usercrisis', 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', 'test', and 'webauth'. The 'usercrisis' table is selected under the 'cmas' database. The main panel shows the 'Browse' tab with a message: 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.' Below this, a green bar indicates 'Showing rows 0 - 0 (1 total, Query took 0.0010 seconds)'. The SQL query 'SELECT * FROM `usercrisis`' is entered in the query box. Below the query box, there are controls for 'Number of rows' (set to 25) and a 'Filter rows' search box. The 'Options' section is expanded, showing a table with the following data:

MOBILENUMBER	CRISIS	LATITUDE	LONGITUDE
9600488613	Communal Riot	12.9259833	80.1598737

Below the table, there are again controls for 'Number of rows' (set to 25) and a 'Filter rows' search box.

CONCLUSION

CHAPTER XI

CONCLUSION

The intrinsic purpose of any technology is to advance the quality of living. Technology is a light that cuts through darkness and touches lives. The development of an application that helps general public to be warned of an occurring crisis will go a long way in helping the public to act responsibly. This coupled along with the low cost technology available come together in building a system that greatly reduces the need for dependency on others social media. This application serves as an assisting aid for the general public as they are ignorant to their surrounding environment. Through this application, the government could establish a more personal reach to its civilians. It enhances the civilians access to any sort of instruction regarding the crisis. Since this works in a real time environment and also involves fairly simple logic, it has immense potential.

REFERENCES

REFERENCES

- [1] Chandra A, Jain S, Qadeer M A, "Implementation of location awareness and sharing system based on gps and gprs using J2me", Computer Research and Development (ICCRD), 2011 3rd International Conference.
- [2] Mohammad M A, Norhayati A. Anumba, "A short message service for campus wide information delivery", Telecommunication Technology, 2003. NCTT 2003 Proceedings.4th National Conference.
- [3] Rahman K M, Alam T, Chowdhury M, "Location based early disaster warning and evacuation system on mobile phones using OpenStreetMap",Open Systems (ICOS), 2012 IEEE Conference.
- [4] Heo G, Genong Yu, LipingDj, "A reconfigurable open geosms mobile client app design for android smartphones", Agro-Geoinformatics (Agro-Geoinformatics), 2013 Second International Conference.
- [5] Sonowal N, YadavR,Kannan S, "Real time gps software receiver with new signal tracking method", Radio and Wireless Symposium, 2008 IEEE.
- [6] MohDYusop, Zulkefli, Mohamad FirdausChe, "Developing an email violation prevention model based on web 3.0 technology", Science and Social Research (CSSR), 2010 International Conference.