**I’M HERE**

A

MINOR PROJECT REPORT

*Submitted for the partial fulfillment of*

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

***Submitted by***

Ayan Chakravorty (9912103402)

Pushpanjay Kumar (9912103407)

Vivek Patel (9912103426)

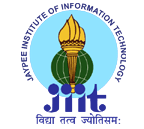
**Under the Guidance**

**of**

Mr. Bansidhar Joshi

(Dept. of CSE/IT)

DECEMBER 2014



**Department of Computer Science and Engineering/IT**

**Jaypee Institute of Information Technology, Noida**

**ACKNOWLEDGEMENT**

I would like to place on record my deep sense of gratitude to Mr. Bansidhar Joshi , faculty, Jaypee Institute of Information Technology, India for his generous guidance, useful suggestions, continuous encouragement and supervision throughout the course of present work.

I also wish to extend my thanks to all the faculties of CSE department for their insightful comments and constructive suggestions to improve the quality of this project work.

**Signature(s) of Students**

Ayan Chakravorty (9912103402) F-1

Pushpanjay Kumar (9912103407) F-1

Vivek Patel (9912103426) F-1

**ABSTRACT**

I’M HERE is an android based project to locate the current position of user’s android device to use that location in emergency situation. What make this different from the other Google apps is that this application will include a Google maps overlay, which enables user to give information and draw items on top of a Google maps tile. The increasing popularity of text messaging has increased the speed and ease of coordinating social events. However, location based GPS social services are currently limited in scope and expensive to utilize. This will implement an application that will be run on a network of GPS-enabled cellular phones using the android sdk. Through some manner of communication, the phones will trade the user's location information and then the application will utilize the Google maps API to create an interactive mapping of the established clients. This will allow any user who has a GPS phone running Android to take advantage of this service without any additional fees levied against him.

**Table of Contents**

1. Introduction…………………………………………05
2. Background study………………………………….. 05
3. Requirement Analysis……………………………….07
   1. Software
   2. Hardware
   3. Functional Requirements
   4. Non Functional Requirements
   5. Diagrams
      1. Use Case Diagram………………………….12
      2. Data Flow Diagram

* Level 0……………………………………13
* Level 1……………………………………14
  + 1. Schema Diagram……………………………..15

1. Implementation ………………………………………...16
2. Testing………………………………………………….17
   1. Black Box
   2. White Box
3. Limitations Of Software Model………………………….35
4. Gantt Chart……………………………………………….37
5. Conclusion……………………………………………...38
6. Future Work…………………………………………….39
7. References...…………………………………………….40

**1. INTRODUCTION**

I’M HERE is an Android based project to locate Friends or Family in the contact

list based on locations and proximities.

All you have to do is to get the app installed on the phone of every user and then add them to user’s contact list, as soon as it is done, we can easily see the exact location coordinates of the second user’s device in real time and integrated chat module helps the users to communicate with each other. This app can be used for sharing each other’s location, finding a lost device and can also be used as a safety tool for people in unsuitable environment etc. This application is set to gain popularity among all age groups due to its diverse use.

**2. BACKGROUND STUDY**

Research Papers:

**A).**

**Title:** Locating friends and family using mobile phones with global positioning system (GPS)

**Author:** G.B Al-Suwaidi and M.J Zemerly.

**Year of Publish**: 2009

The main objective of this work is to design and implement a client server system that helps users to locate their family members and receive alerts when friends are nearby. The mobile application was implemented using java where the most recent APIs and other older APIs were combined together in order to make the application reliable on all types of mobiles. The type of the database used in the system was MySQL

**B.)**

**Title: Location Based Services on Smart Phone Using Android Application**

**Author:** Prof. Seema Vanjire, Unmesh Kanchan, Ganesh Shitole, Pradnyesh Patil

**Year of Publication:** 2014

The motivation for every location based information system is: ―To assist with the exact information, at right place in real time with personalized setup and location sensitiveness. In this era we are dealing with smartphones. All the information must be available in his mobile device and also in user customized format. Our project Android application for location based service on mobile is based on the Android Operating System, GPS technology & Java technology (J2EE). Android operating system & Java technology emerges as one of the hottest topic in field of information technology. The Android operating system is comprised of a virtual machine that runs on the Linux kernel, plus APIs, and a collection of built-in applications. A location-based service (LBS) is a mobile application that is dependent on the location of a mobile device. A positioning component is usually needed in a LBS application to determine the location of user's mobile device. Most of the current LBS services do not require users to input location manually, like giving zip code or street name.

# 3. REQUIREMENT ANALYSIS

# 3.1 Software:

* **Users** Android OS 2.3 or above
* **Development End** eclipse IDE

## 3.2 Hardware:

* Android device
* GPS Receiving capability (A-GPS)
* GPRS Content Provider
* Internal memory or Micro SD card of capacity at least 512 MB.

## 3.3 Functional Requirements:

1).The system should allow new user to register with their mobile number, username and password and update the entered data in the database.

2).The system should display all the screens to get the information about the user contacts information and the Google map feeder configuration.

3).User information includes name, contact number whereas the Google map feeder configuration include services usage and all other screens that are going to establish the connectivity between the Android application and the Google map feeder.

4).Given all the details the system should establish a connection between the Android application and the configured online Google map feeder.

5). After establishing the connection between the Android phone and the database the system should download all the contacts from contacts list and the contacts should be of any number.

6). If text box left empty, show error message for entering data in remaining field.

7). When SOS button is pressed, the application should be able to access the intent feature of SMS.

8). The application should be able to access the created database smoothly.

9). The application should be able to access the GPS service of the phone to retrieve the exact location of the phone.

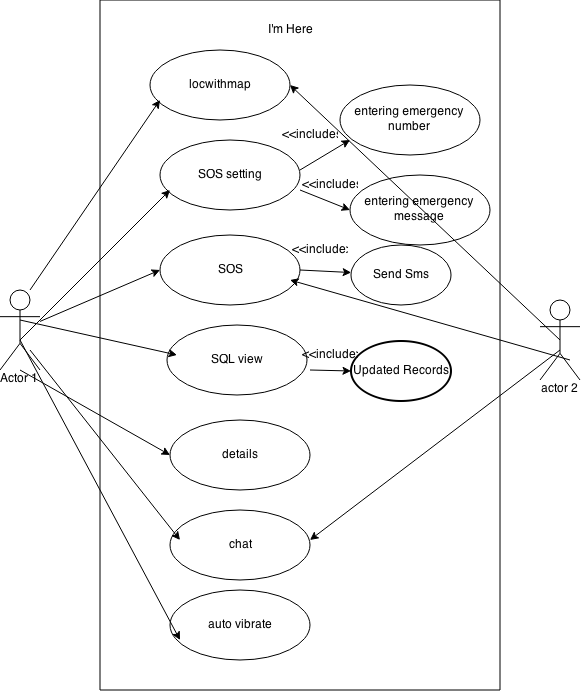
## 3.4 Non Functional Requirements:

Some Performance requirements identified are listed below:

1. The database shall be able to accommodate a minimum of 5000 records of users.
2. User should be able to sign in and connect within 20 sec.
3. The user must be able to access their account 24 hours a day, seven days a week.
4. The system shall respond to user commands within 500 milliseconds.
5. Proper sized picture must be available.

**3.5 DIAGRAMS:**

3.5.1USE CASE DIAGRAM**:**



3.5.2 DATA FLOW DIAGRAMS:

**NO**

**NO**

**YES**

**YES**

**YES**

**if GPS is on**

**if wi-fi is on**

Print: Can’t get current location

**NO**

**if GPRS is on**

Collect current Coordinates(Lat,lng)

**Figure 1: Flow chart showing the procedures of getting current position**

**Data caching**

**Search for a place**

**route**

**Spot nearby**

**Search for a person**

**map**

**My location**

**Figure 2: abstract view of user interface menu**

XML file

Other person’s location

Spot near the coordinates (text description & image)

Mark on the map

**Input my**

**location**

**YES**

**NO**

HTTP request

(Google geocoding API)

SAX(simple API for XML)

Coordinates (lat,lng)

(

Collect current location using phone

**Catch in the memory**

**Figure 3: Flow diagram showing the processes of search function**

Cache in the memory

Spot near the point (text descrption & image)

Perl file

GEO coordinates (lat,lng)

Get the screen coordinates(X,Y)

Insert the point by touching the screen

convert

HTTP request (GSW)

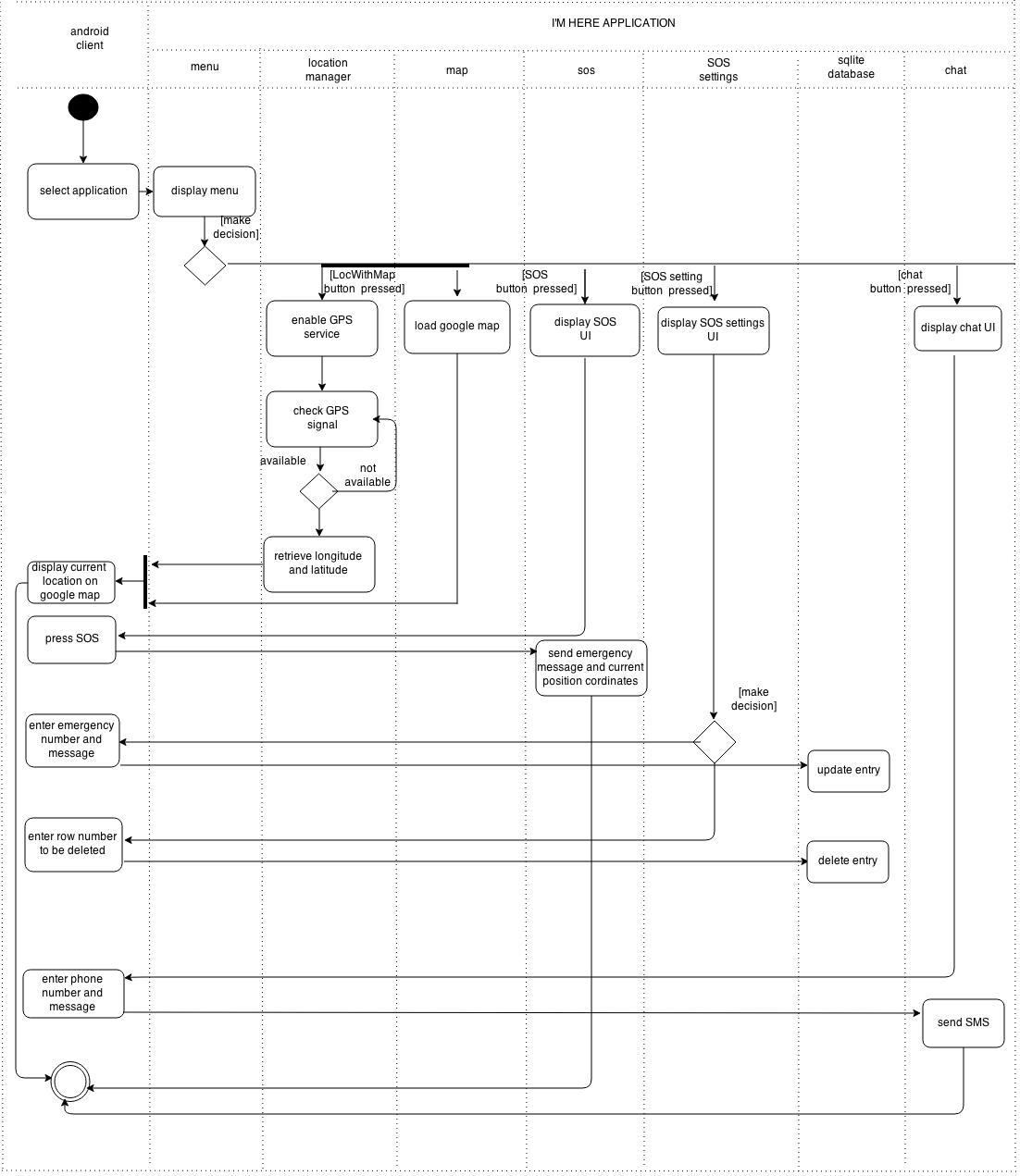
SAX (simple API for XML)

**Figure 4: Flow diagram showing the processes of assigning location**

3.5.3 SCHEMA DIAGRAM:

|  |  |  |
| --- | --- | --- |
| **Key id** | **Contact** | **Message** |

## 3.5.4 Activity diagram



**4. IMPLEMENTATION**

**Introduction**

This document describes the project implementation for developing an Android Application. The project implements Android SDK, SQLite, JAVA and standard XML. The project will be capable of running on Android OS (API 16 and above), although, the project is designed primarily around Android 4.1.2 (API 16). The interface for the project will provide a point of attraction for the User’s.

**Flow Chart: Client**

Client

Splash Screen

Main Screen

Menu

Location with Map

SOS

SOS Setting

Database View

Splash Screen appears for 4 seconds.

Client visits the Main Screen.

Client scrolls

The Menu looking for suitable Option.

Client can view their on location (coordinate) along with their Position on Google Map.

Client can send immediate emergency Message along with their current location to their Dear one’s when in uncomfortable Environment.

Client can Update, Modify the Emergency Contacts and Messages in the Database.

Client can see the Database.

Chat

Client will be able to chat with another Person in Real time.

**Modules**

**Splash Screen**

The Splash Screen of application will appear for 4 seconds and then will disappear leading us to the main screen of the application. The Splash Screen will show details about the application name and is just for interface purpose. Splash Screen is called by method startActivity() which is in Intent class of android.content.Intent Package.



Splash Screen

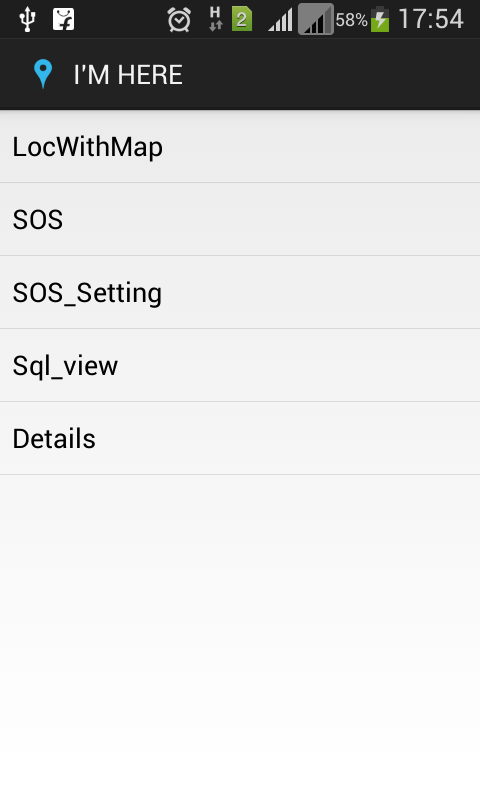
**Main Screen and Menu**

Introduction: The main screen of the application and is the entry point for all other screens contained in the application. The client will be able to browse through the menu from this point.

Details: The main screen and menu is developed in JAVA using the Interface ListActivity of API package. The menu will contain links to the modules. The layout of the page is based on setListAdapter. Each ListAdapter will contain a link to a module.

Database Tables: The menu is also linked to a database table to directly view the current database with interfering with any module.

Error Handling: The menu will handle errors by attempting to catch the Exception and inform the client type of Exception.

****

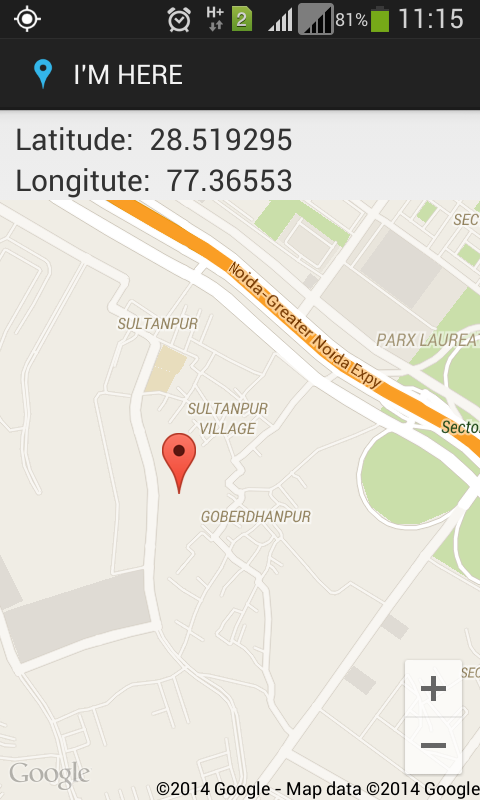
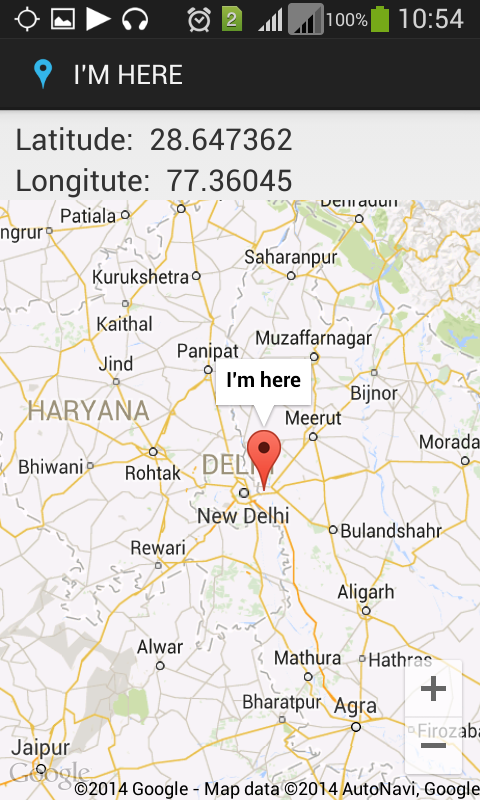
**Menu Screen**

**Location with Map**

Introduction: The Location with Map module will allow the client to view their current location in coordinate form (Latitude and Longitude) as well as on the Google Map.

Details:

* Client will have to enable the GPS service on their GPS-Enabled Android Smartphone and within few seconds they will be able to see their exact coordinates in the display frame and on the Google Map with a pointer on their Location.
* The layout of screen is developed in XML with LinearLayout format supporting two textviews for latitude and longitude and a Fragment for Google Map.
* The position through GPS is extracted via **LocationListener** interface which comes in android.location.LocationListener package.
* The method of the LocationListener interface, onLocationChanged( ) gets invoked whenever there is a change in the position of device and through location.getLatitude( ) and location.getLongitude( ) method latitude and longitude is assigned to local variable.
* **LocationManager** data type enable the option of selecting best location provider and variable of Criteria type allows to select the best provider. LocationManager also checks whether GPS Service is available on the device or not.
* **Google Map** is integrated in fragment via Google Play service which has to be externally attached to the project to enable the application to use the various Google Map API methods.
* **Google Map API** is only accessible after registering the developer’s system at the Google Developer Console and registering the project to generate Map API key which is used in the Manifest of the project.
* **OpenGL ES** is used to provide 2D and 3D graphics, to display the Google Map on the fragment.



Location On The Map

**SOS**

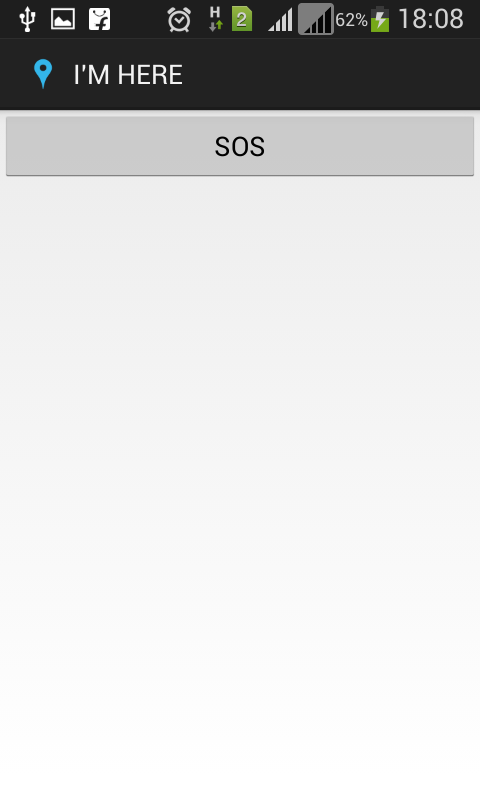
Introduction: The SOS module will allow the client to send immediate emergency message and their exact position to their dear one’s with just a click.

Details:

* Client have to store the emergency contacts and messages in the database. SQLite is used here to store the entries.
* OnClickListener interface is used to catch the event and perform the SOS operation. Emergency contact and message is accessed from the database.
* SmsManager type variable is used to invoke sendTextMessage( ) method, which is present in android.telephony.SmsManager package.

Database Tables: The database is also linked to directly access the contacts and Messages stored by the client.

Error Handling: The error will be handled by attempting to catch the Exception and inform the client, the type of Exception.



SOS Screen

**SOS Setting and Database**

Introduction: In this screen client can modify, update, remove or just view the entries from the SOS database, .i.e. client can modify their emergency contacts and messages

Details:

* This Screen provide an interface to interact with the database. TextView,
* EditText and Buttons are used to handle all the operations on the database.
* SQLiteOpenHelper class of android.databse.sqlite.SQLiteOpenHelper package is used to create new databases and database table using Dbhelper( ) method.
* onCreate() method is used to create first entry in the Database and the subsequent entry is made using onUpgrade( ) method.
* Databasename.execSQL() method is used to write an operation which is to be performed on the database.
* Cursor type variable is used scroll the database to access any entry.
* SQLiteDatabase type variable of android.database.sqlite.SQLiteDatabase package is used in creating the Database.

Database Tables:

Table Name: DATABASE\_TABLE

Table Fields: KEY\_ROWID

KEY\_CONTACT

KEY\_MESSAGE

Code:

db.execSQL("CREATE TABLE " + *DATABASE\_TABLE* +" (" +

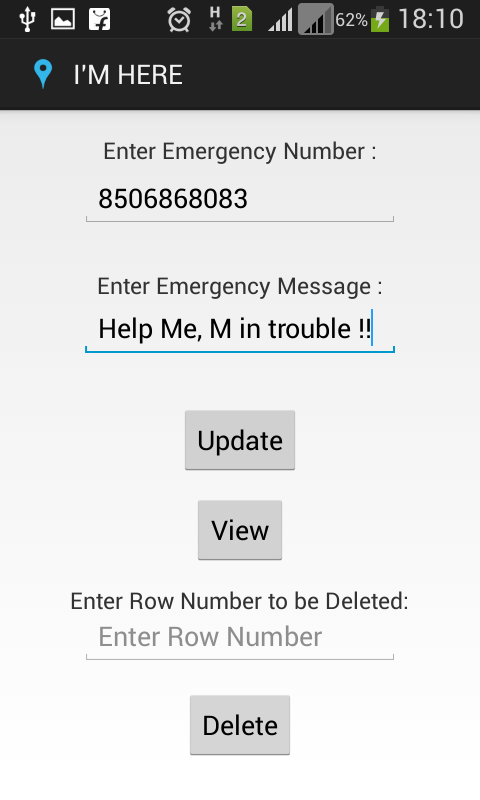
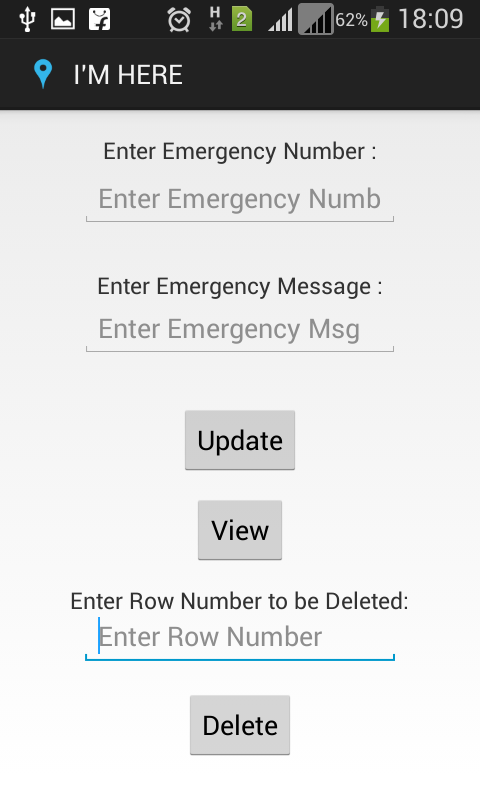
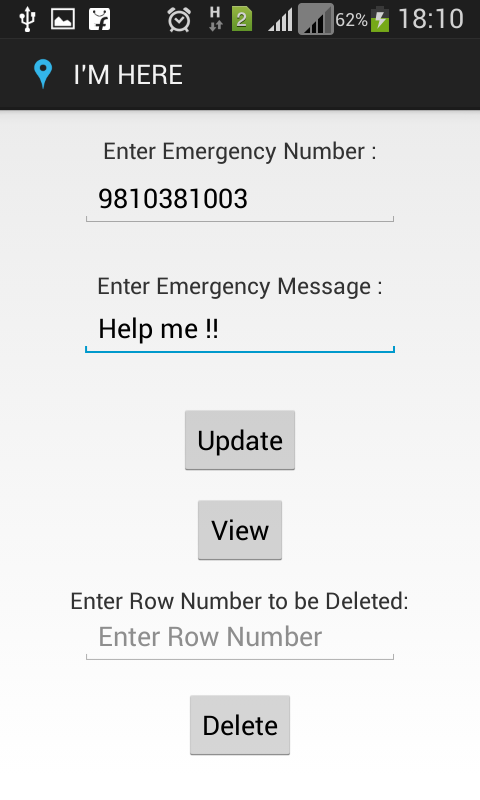
*KEY\_ROWID* + "INTEGER PRIMARY KEY AUTOINCREMENT," +

*KEY\_CONTACT*+" TEXT NOT NULL, "+

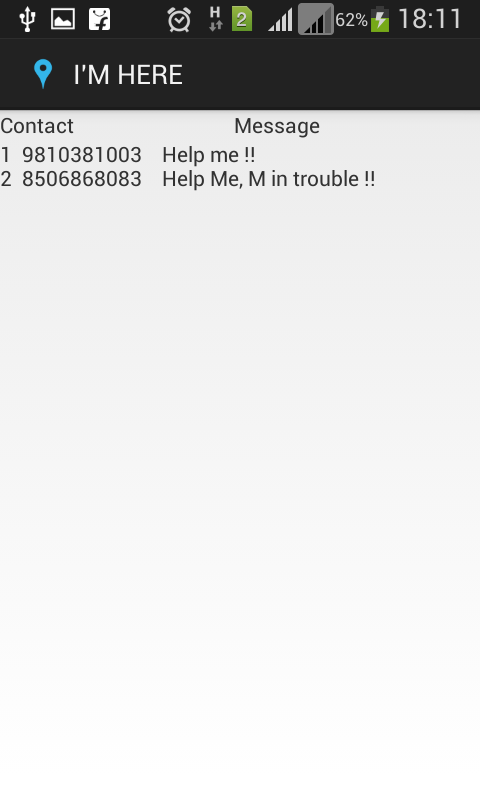
*KEY\_MESSAGE*+" TEXT NOT NULL);"

);

Error Handling: Incomplete or partial data will be dropped from the database. All fields need to be filled in. The errors in accessing, creating, modifying, deleting entries in the database are handled by throwing and catching SQLException type Exception which is imported from android.database. SQLException.

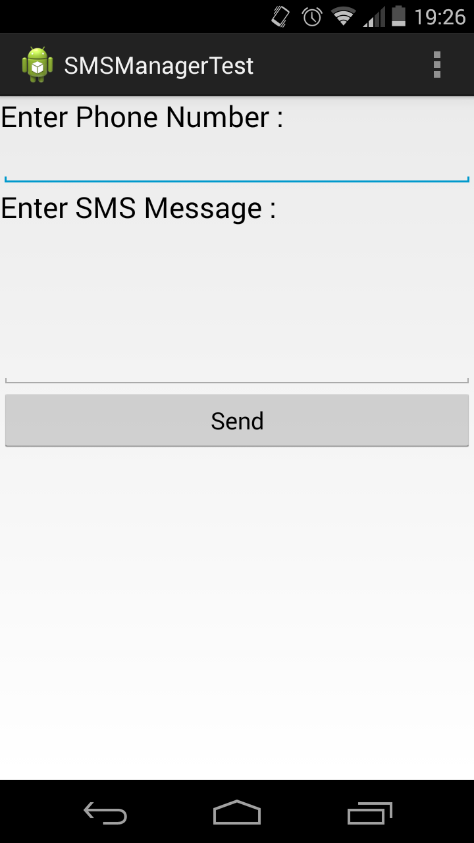


SOS Setting and Updating Database



Database Screen

**Chat Module**



1. Send attention message text:

*Perform predefined action according to alert message and abort broadcasting.*

1. Send text other than attention word:

*Allow broadcasting*

1. Get latitude and longitude of device and integrate them in the message being sent.
2. Send device location to sender of SMS.
3. Exit Application

**Fig:** Chat interface

**Details**

**a. Broadcast receiver that alerts application when each new SMS arrived.**

This module decides which action has to perform when attention word matches with the default keyword being set. If it is matched then it starts activity which enables device ringing. If attention word matches with the keyword for location then it starts activity which retrieves location of device and sends information to the sender of SMS. At the same time it aborts message broadcasting so that message can’t be reached to inbox of native messaging application.

If attention word is not matched with the specified key word than it simply allow broadcasting so that message can be reached to inbox of native messaging application.

**Step 1:** START

**Step 2:** SMS received.

**Step 3:** Checks attention word.

**Step 4:** If attention word matches with the default message then starts ringing activity and abort broadcasting.

**Step 5:** If attention word matches with location keyword then starts ringing activity and abort broadcasting.

**Step 6:** If attention word not matched then allow broadcasting.

**Step 7:** End

### 

### 

### b. Get location And Acknowledges user

Received attention word text which is the location keyword being set.

In this module we provide the functionality of getting location details of device and the same will be sent to user. Location recognizes the keyword, retrieves latitude and longitude of device, integrates the location in message and the same will be sent to sender of SMS.

**Step 1:** START

**Step 2:** Checks that internet is available.

**Step 3:** If internet is available then get location details from Network Provider.

**Step 4:** If internet is not available then Checks is GPS turned on.

**Step 5:** If GPS is available then get location details.

**Step 6:** Send location information to user.

**Step 7:** Pass on the location to the message

**Step 8:** Send the message to other user.

**Step 9:** End.

**Sound Profile**

This is a module for automatic profile switching as per location

Working:



Fig: Architecture design

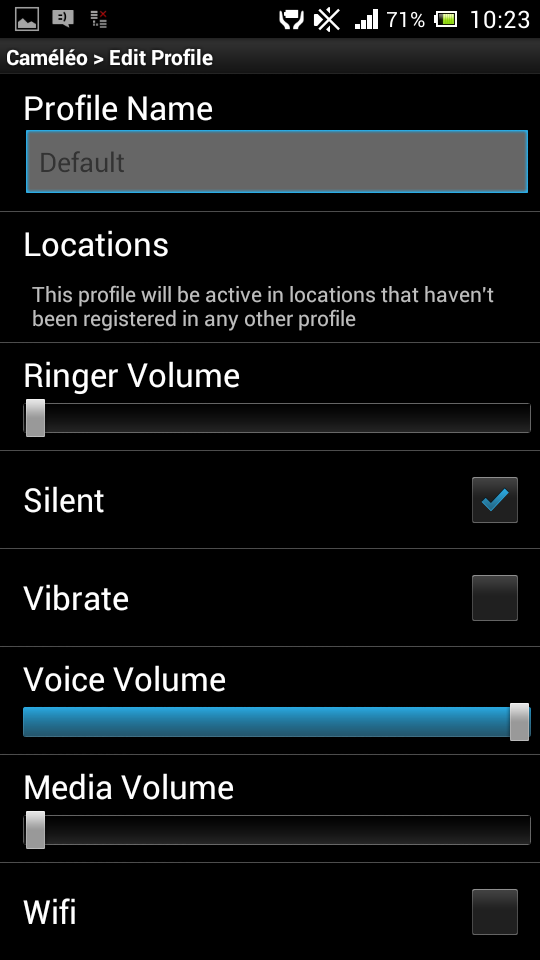


Fig: profile menu

1. Default Mode Switching

In Default Mode flow of processes will be as follows.

1. Finding coordinates of current location of Device

*GPS receiver (i.e. our android device) calculates the coordinates of current location using Forward Geocoding method. A software interface of Location Manager is used to get the coordinates from GPS receiver in Android Mobile Device.*

2) Getting address of location from GPS Server

*Using Reverse Geocoding method, device finds out the address of the location of device. Using Location Manager Interface device sends the coordinates of location to GPS Server as a key to find out the address of that particular location. If location entry is present in GPS Server Database, GPS Server sends back the address for that location in form of multi-line text (list). If location entry is not present GPS Server sends the NULL value.*

3) Checking if location belongs to Silent Zone

*After getting the response of GPS Server, if GPS Server sends NULL value then it means the location is not belong to Silent Zone else device will check for substring in address such as School, College, University, Hospital, Pvt. Ltd., Government of, Petrol Pump etc.*

4) Switching sound profile accordingly

*If location is not belongs to any of the Silent Zone then maintain ringer mode as ‘General’ else switch ringer mode as per setting to Complete ‘Silent Mode’ or ‘Vibrate Only Mode’ or ‘Flight Mode’.*

*B. User-Defined Mode Switching*

1) Finding location in SQLite Database

*Coordinates of current location find in Default Mode will be used in User-Defined Mode also. Using those coordinates device checks the entry made by user for that coordinates in its SQLite Database. If device founds entry in SQLite Database then device is in Silent Zone else not.*

2) Checking status of location

*If device is in Silent Zone then device checks the status for that user-defined location, whether it is activated or not. If it is activated then device is in Activated Silent Zone else not.*

3) Switching sound profile accordingly

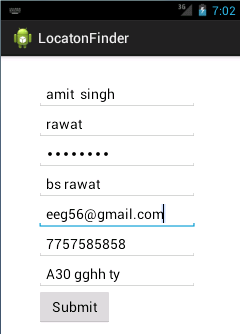
*If location is not belongs to the Activated Silent Zone then maintain ringer mode as ‘General’ else switch ringer mode as per user setting to Complete ‘Silent Mode’ or ‘Vibrate Only Mode’.*

**Distance and Location**

Introduction:

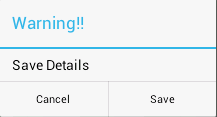
The user can find his location and friend’s location coordinates on the map and the distance between the two locations.

1. Registration of the user:

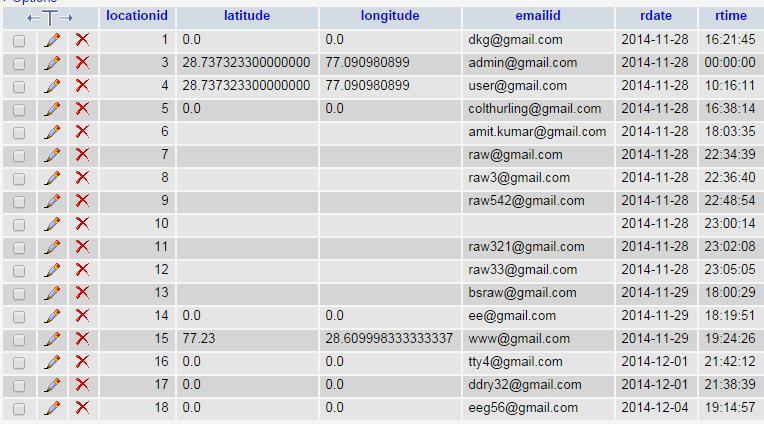


Details:

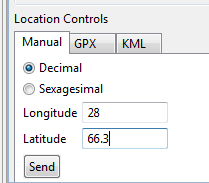
2. Submit the details:



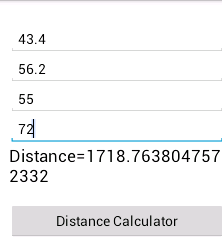
Details of the user are submitted and saved in the database:



3. Send the location of the user via ddms



4. Distance can be calculated:



5. Location of the user and friend’s location



**Summary**

This document shows flowchart diagrams of how the client can access the various modules. The modules are then described in detail. The database tables are described in detail with field names, field categories and other important information. The tables can be built with the information provided in SQLite.



**Fig:** Details

**5. TESTING**

**5.1 BLACK BOX TESTING**:

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings.

1. Usability testing

User interface testing:

1. Easy to use

.control are well defined.

.control are not wrapped /truncated/overlapped.

.control provide complete description.

.initcap

.system menu

1. Look and feel

Consistency along the application.

1. quick interface

Main screen easily accessible.

**5.1.1 current\_location:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Test Cases Description** | **Expected Output** | **Actual Output** | **Remark** |
| 1 | **User is trying to retrieve Latitude and Longitude of current location** | **Location retrieval successful** | **Location retrieval successful** | **Expected Is Same As Actual** |
| 2 | **Retrieving Latitude and Longitude without internet connection** | **Latitude and Longitude retrieval successful without map loading.** | **Latitude and Longitude retrieval successful without map loading.** | **Coordinates retrieval successful** |

**5.1.2 SOS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Test Cases Description** | **Expected Output** | **Actual Output** | **Remark** |
|  | **If the User is trying to access SOS feature.** | **Send SMS to the user defined emergency number** | **Login successful** | **Expected Is Same As Actual** |
|  | **If user is trying to login without his/her username and password** | **Login unsuccessful** | **Login unsuccessful** | **Error occurred if user is trying to login with incorrect id or password** |

**5.1.3 Chat:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Test Cases Description** | **Expected Output** | **Actual Output** | **Remark** |
|  | **Send text message with limit of 140 characters** | **Message sent and receiver gets**  **message** | **Message sent and receiver gets**  **message** | **Error shown if network not available** |

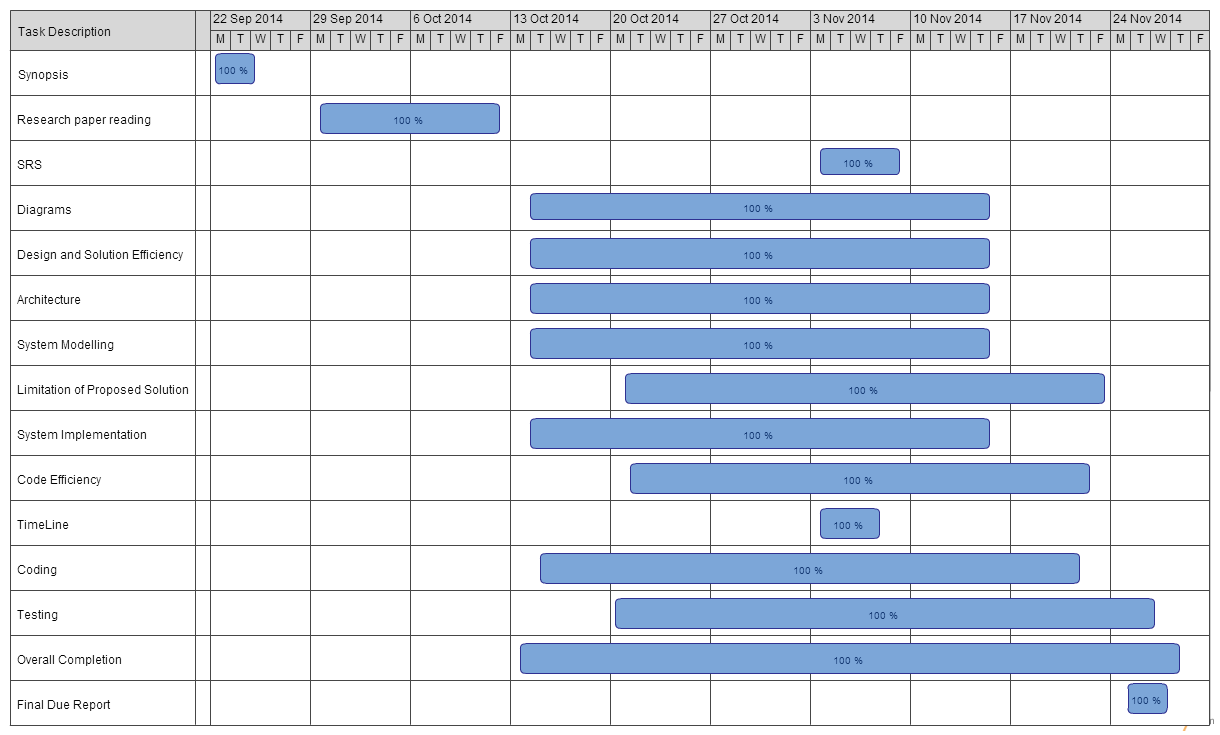
**5.1.4 Distance calculator:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S no.** | **Description** | **Expected**  **output** | **Actual output** | **Remark** |
| **1.** | **User’s location** | **Actual coordinate and location on the map** | **Actual coordinate and location on the map** | **Error shown if network not available** |
| 2. | **Friend’s location** | **Actual coordinate on the location on the map** | **Actual coordinate on the location on the map** | **Error shown if network not available** |
| 3. | **Distance between the two locations** | **Distance in meters** | **Distance in meters** | **Error if latitude and longitude are not given or not properly given** |

**6. LIMITATIONS**

* GPS system might not give specific location according to user requirement in enclosed places.
* Database security remains an issue
* Only available for android platform.
* In bad weather GPS may not give current location.

**7. GANTT CHART**

****

**8. CONCLUSION**

Android users today have a plethora of apps available at their fingertips, from text messaging and social networking to real-time video streaming and immersive 3-D gaming. Thriving developer communities has evolved to support the growing user-base, which continue to make available thousands of new apps to mobile phone users. I’M HERE is a unique and efficient application which can search the second user’s present location via coordinates and to trace the path on which his/her device moves. The additional features like sound profile changer, distance calculator, messaging and SOS will make this app gain popularity among all age groups due to its diverse use. This app can be used for sharing each other’s location, finding a lost device and can also be used as a safety tool for people in unsuitable environment etc. The application will be published free for download in the Google Play store.

**9. FUTURE WORK**

The “I’M HERE” app can be improved and extended in several ways. Here we explore some of these exciting possibilities and discuss our future plans for adding new features that build upon and compliment the current functionality.

In addition to location, we would like to add a reminder function in this app. This will allow users specify a time and calendar date when location is not sufficient to define the context and make it integrate with the built-in Google Calendar app for easy access.

Social networking features can help leverage the location based services, adding a whole new dimension of possibilities. For example, sharing of bookmarked locations between Facebook friends would make it easy for people to exchange addresses of popular restaurants, tourist attractions, hang-out destinations, etc. Bookmarks can also be extended to include geo-tagged photos, videos, reviews and articles. In addition, this app can be updated to show nearby friends on the map in real-time and send alerts when they are immediate vicinity, making it easy for people to connect with each other.

Location centric micro-blogging is another feature we would like to explore. This would allow users to blog on websites such as Twitter, in real-time, directly from their phone and share location specific information such as photos and videos of the places they visit or publish reviews of restaurants, shops, etc.

Some extra features like SIM card change notification, call/SMS filtering, lock device and wipe memory for protecting private data in case of device is stolen, etc.

**10. REFERENCES**

**1. Journal Article:**

G.B Al-Suwaidi and M.J Zemerly, “Locating friends and family using mobile phones with global positioning” Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference 10-13 May 2009

**2.** **Online:**

Android Developer Guide:

<http://developer.android.com/guide/index.html>

This website provides us a rich-content which teach us some fundamental concepts about Android development and also take us through some advance concepts related to Android application development.