DEVELOPMENT OF AN ANDROID-BASED MOBILE APPLICATION FOR PERSONAL SAFETY

By

Sabuj Biswas

POST GRADUATE DIPLOMA IN INFORMATION AND COMMUNICATION TECHNOLOGY

Institute of Information and Communication Technology (IICT)

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY (BUET

September 2017

The project titled "DEVELOPMENT OF AN ANDROID-BASED MOBILE APPLICATION FOR PERSONAL SAFETY" submitted by Sabuj Biswas, Roll No. 0413311001, and Session April-2013 has been accepted as satisfactory in partial fulfillment of the requirement for the Post Graduate Diploma in Information and Communication Technology (ICT) on 25 September 2017.

Board of Examiners

1.	Dr. Md. Rubaiyat Hossain Mondal Associate Professor Institute of information and communication technology BUET, Dhaka-1000, Bangladesh.	Chairman (Supervisor)
2.	Dr. Md. Liakot Ali Professor Institute of information and communication technology BUET, Dhaka-1000, Bangladesh.	Member
3.	Dr. Hossen Asiful Mustafa Assistant Professor Institute of information and communication technology BUET, Dhaka-1000, Bangladesh.	Member

Candidate's Declaration

It is hereby declared that this project or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Sabuj biswas

SabujBiswas
ID: 0413311001

Dedicated to My Mother “Radha”

TABLE OF CONTENT

ACKNOWLEDGEMENT VII

ABSTRACT VIII

CHAPTER I 1

Introduction 1

1.1 Background and Present State of The Problem 1

1.2 ObjectivesWith Specific aims and Possible Outcome 1

1.3 Outline of Methodology/Experimental Design 2

1.4 Related Research 3

1.5 Method and Materials 4

CHAPTER II 6

2.1 Data Flow Diagram (DFD) 7

2.2 Software Building Requirement 10

2.3 Application Functional Process 15

2.4 Methodology 23

2.5 Database Design 23

2.6 Requirement Specifications 23

2.7 Use Case Diagram 24

CHAPTER III 26

Implimentation of the project.....26

3.1 SharedPreferences.....26

3.2 XML.....27

3.3 PHPMailer.....27

3.4 LocationManager 28

3.5 Base64 Encoding 29

3.6 Base64 Decoding 29

3.7 Service.....29

CHAPTER IV 30

Conclusion and Future works 30

4.1 Important features of the software 30

4.2 Limitations of the software 30

4.3 Future Work 30

REFERENCES 31

APPENDIX.....33

TABLE OF FIGURES

FIGURE 2.0 : USE CASE DIAGRAM 6

FIGURE 2.1: DATA FLOW DIAGRAM FOR SYSTEM MODULES 7

FIGURE 2.2: DATA FLOW DIAGRAM FOR MODULES RELATIONSHIP 8

FIGURE 2.3:FRONTEND AND BACKEND DIFFERENTION.....10

FIGURE 2.4 : ANDROID SOFTWARE STACK 11

FIGURE 2.5 : FLOW CHART FOR APPLICATION FUNCTIONALITY 17

FIGURE 2.6 : USER INTERFACE FOR EMERGENCY INFORMATION SETUP 18

FIGURE 2.7 : SENT INFORMATION CONFIRMATION SCREEN 19

FIGURE 2.8 : RECEIVED INFORMATION SCREEN IN MOBILE.....20

FIGURE 2.9 : RECEIVED INFORMATION SCREEN IN EMAIL.....21

FIGURE 2.10 : VICTIM LOCATION FOUND AFTER CLICKING ON THE LINK.....22

FIGURE 2.11: DATABASE DESIGN.....23

Acknowledgement

At first I would like to convey my gratitude to Almighty GOD for giving me the opportunity to accomplish this project. I would like to thank my supervisor Dr. Md. Rubaiyat Hossain Mondal, Associate Professor, Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh. He has assigned me an interesting and useful area, which has an extensive range of relevance in real world. He has provided all sorts of support regarding the project work. Without his proper guidance, advice, continual encouragement and active involvement in this process of this work, it would have not been feasible.

I am indebted to all the teachers, officers and staffs of Information and Communication Technology (IICT) for giving me their kind support and information during the study. I am also grateful to my parents whose continuous support all over my life has brought me this far in my career.

Finally, I tender special thanks to the Almighty that I have been successful in my effort to complete the study.

Abstract

Android is a java based operating system which runs on the Linux 2.6 kernel. It is lightweight and full featured. Android applications are developed using Java and can be ported to new platform easily thereby fostering huge number of useful mobile applications. This project describes the development of a Safety Triggering application. The application has target users who surprisingly fall into a situations where instant communication of their whereabouts becomes indispensable to be informed to certain authorized persons at remote end. The main purpose of this application is personal safety. When we feel that we are in emergency situation, for example travelling alone at night time, we can use this application. With this application, on one click we can send our location, text messages, image to our family members and to any police station continuously until we stop with hide button.In this project, an emergency notification application for mobile devices is designed. This project contains the detail survey of existing systems to overcome common problems of having manual intervention while reporting emergency. This scheme automates the process of emergency detection and reporting in real time. It works in three steps which are the detection of emergency, the data collection/processing and the reporting to the friends/family members. Smart phones upload that emergency along with other information like current location tracked by global positioning system (GPS) on phone, mobile number, incident time, video and image of the incident to the designated email addresses and mobile numbers. In the application, the position function of a GPS and an easy-to-use interface capable of sending emergency notification messages or recording systems are included. The application design has five parts: a mobile client, an application server, a database, a GPS system and a map service. A mobile client consisting of a mobile and a GPS receiver finds the location of the user to get aware of his location. Users can take benefit of this application in emergency situations by using emergency feature of this application.

CHAPTER-I

Introduction

1.1 Background and present state of the problem :

Safety of people when travelling alone particularly at night is a concern [1-4]. Unfortunately there are a growing number of crimes taking place each day. In many cases, it becomes difficult to find the trace of the incidents and the criminals. With the conventional mobile phones, a person in danger makes a call to his/her friends or relatives and explains the position the problem. But, this system has a number of drawbacks as people often do not have the chance to explain the incident in real time over phone. The proper use automatic notifications in smart phone applications are helpful to identify miscreant or incident happening places [3], [5]. Such an application will improve the life of all citizens by offering them a higher degree of security in our society. In most of the existing mobile phone applications in the market, a user can send global positioning system (GPS) location and distress messages. However, some of these do not have video capturing option [6-8] or automatic siren system. Having all these safety features free in a mobile application will be useful. This project focuses on the development of a full-featured Save Our Souls (SOS) safety application [4-9] in android platform.

1.2 Objectives with specific aims and possible outcome

The objective of this project is to develop an Android mobile application with the following features:

- To create a personal profile of the user and store the names, email addresses and mobile phone numbers of the emergency contact persons.
- To send user location and send messages automatically to designated email addresses and mobile phones when the phone is triggered.
- To set off a siren or warning sound when the user wants.

- To automatically send image and video of the incident to designated persons via emails and multimedia messaging services (MMS) when the panic button of the phone is pressed.
- When the person feels insecure at the dangerous zone at that time he/she can press the emergency button or shake the mobile phone then the application will be started, automatically the application will take photo of that locationare also will be activated. GPS will calculate the latitude and longitude co-ordinates of that area. Then the application will send the image, text messages and GPS location link to the server. Server will then sends the location link,messages, photo to the specified email address. GSM module will send SMS which contains the latitude and longitude co-ordinates, photo link which stored in server to the numbers such as family, friends, police station and neighbors which are already stored in this emergency button application.

The outcome of this project is a mobile application to notify friends and family members in emergency situations.

1.3 Outline of Methodology/Experimental Design:

First of all, requirement analysis will be done regarding the important safety features and then the specifications will be written. An appropriate design will be done and then the software will be developed with Android studio. This Android application will contain two - side implementation: one is for the victim and the other is for the helper(s). A victim will send the information to a central server which will then divert it to the clients/helpers who will receive the emergency messages. To develop this application, Android Studio Integrated development environment (IDE) and languages like XML, JAVA will be used. The project will utilize the location-based mobile service design on Android operating system, the Short Message Service (SMS), and multimedia messaging services (MMS) functionality and the camera functionality of the smart phone. In order to fetch the current position of the user, it is needed to obtain the longitude and latitude coordinate values of the geographical location. This will be done by the build-in GPS of the android phone using the

GPS Application Program Interface (API), and then the location data will be sent as SMS using an Android tool termed as SMS Manager Class. Shared preferences memory of Android will be used to store data in the Android phone. PHP will be used for dynamic scripting and MYSQL will be used as database at the server.

1.4 Related Research

Nicole Westmarland et al [11] discussed on protecting women‟s safety. The main purpose of their study is to explore the use of smartphone‟s in relation to domestic and sexual violence. In report [10] violence against women is a global public health problem. The report also details the effects of violence on women‟s reproductive and mental health. In [12] the authors seek to place questions of surveillance technologies into a theoretical framework that foregrounds the challenges that new surveillance technologies pose to antiviolence movements. Specifically they address the impact of surveillance technologies in the practice of violence and some proposed solutions, and consider the ways that surveillance technologies are used disproportionately in the criminalization of marginalized groups. In [13], the author discussed that the technology is used in circumstances of intimate terrorism. It will examine how technology is used as a batterer's tool in exerting coercive control over a victim. It will also look at the changes in the laws as the legal system strives to keep pace with the rapid advancement of technology. In particular, the recent use of GPS monitoring of intimate terrorists has been analyzed. This analysis will identify some of the problems associated with the on-going legal changes. However, the same advancements may also allow perpetrators to adapt and/or escalate their offending behavior [14]. In a US study on domestic violence and information technology, Dimond et al [15] found that users of mobiles and social networks were making trade-offs between potential harm (e.g., continued abuse) and benefit (e.g., support). Worldwide, the sales of new smartphones are forecast to reach 700 million by 2015 [16]. In the UK, 92% of adults personally own or use a mobile phone and 39% use their mobile handset to access the internet [17]. Furthermore, 40% of UK adults own a smart phone, and tablet ownership jumped from 2% to 11% in just 12 months. Based on the advanced mobile and software capabilities, the smart phone has become a ubiquitous everyday communication device, offering access to powerful computational software that is often a multiplatform and sensor rich[18]. In the study [19],

the authors provided insight into the opportunities and challenges involved in delivering health related behavioral interventions through smart phone apps. The findings suggested a number of valued features and characteristics that app developers may wish to consider when creating health behavior apps. Findings also highlighted several major challenges that appeared the need to further consideration and research to ensure the development of effective and well-accepted behavior change apps. In report [20] by Vodafone on Connected Women, how mobile can support women‟s economic and social empowerment. The use of mobiles improves the women‟s access to literacy, banking, health, empowerment and business opportunities.

1.5 Methods and Materials

There might be a situation in which the person has to travel alone a long distance at an odd hour and perhaps even by public transport and may face some danger. At such a time, a personal safety app might not only be wise to have easy access to, it might also give you a lot of confidence needed. When men/women had an accident in the late night and there are no one to help and to take care of them. In such situations, the person will not be able to tell the situation that he/she facing. At such a time, a personal safety app might not only be wise to have easy access to, it might also give you a lot of much needed confidence.Moreover, they do not know the basic first aid details and to know the person where the incident has happened. To escape from the un-wanted meetings we do not know the way to escape from that meeting because we do not know the fake calls working. There are no mobile applications for the person‟s safety, if the person is in danger the people have to make a call to his friends or relatives and have to be explaining the position and problem and generally we do not know the first aid details.

Some drawbacks in the existing system are discussed below: The person who is in danger (he/she) can‟t explain and show his/her position and situation.

- The person does not know the first aid details.
- In proposed system with the push of one button, people can alert selected contacts that the person is in danger and share the location. With this personal safety app, one will not fell alone. The personal safety application requires the name and number of the person who is to be contacted in times of emergency. Users can add multiple people in the emergency

contacts list. These are the people who will receive notifications or SMS in case of an emergency. All it requires is the user's action to trigger an SOS button provided and it shoots messages as fast as the device can manage. Once the SOS button is hit, the people in the emergency contacts will get a message like: I am in an emergency; followed by another message, which has the exact or approximate GPS location of the cell phone. The user can also send video and image of the situation. This app also provides necessary first-aid measures that should be taken at the time of emergency situations.

Features:

Let the family and friends know that the person is in danger and where he/she is? The system will declare an emergency whenever

- The person senses a danger, when he/shecan disengage the emergency. Provide necessary first-aid measures that should be taken at the time of some dangerous situations.
- Let family and friends know the person‟s path to the destination.
- Creates emergency contacts that will be notified by default of all the actions the person makes in the application.
- The person can send video, image and SMS of the situation.

CHAPTER-II
System Study and Requirement Analysis

This project was developed following a number of stages including **Data Flow Diagram (DFD)**, ***Software Building Requirement***, **Methodology**, **Database Design**, **Requirement specification**,and **Use case diagram**. These stages are described in the following.

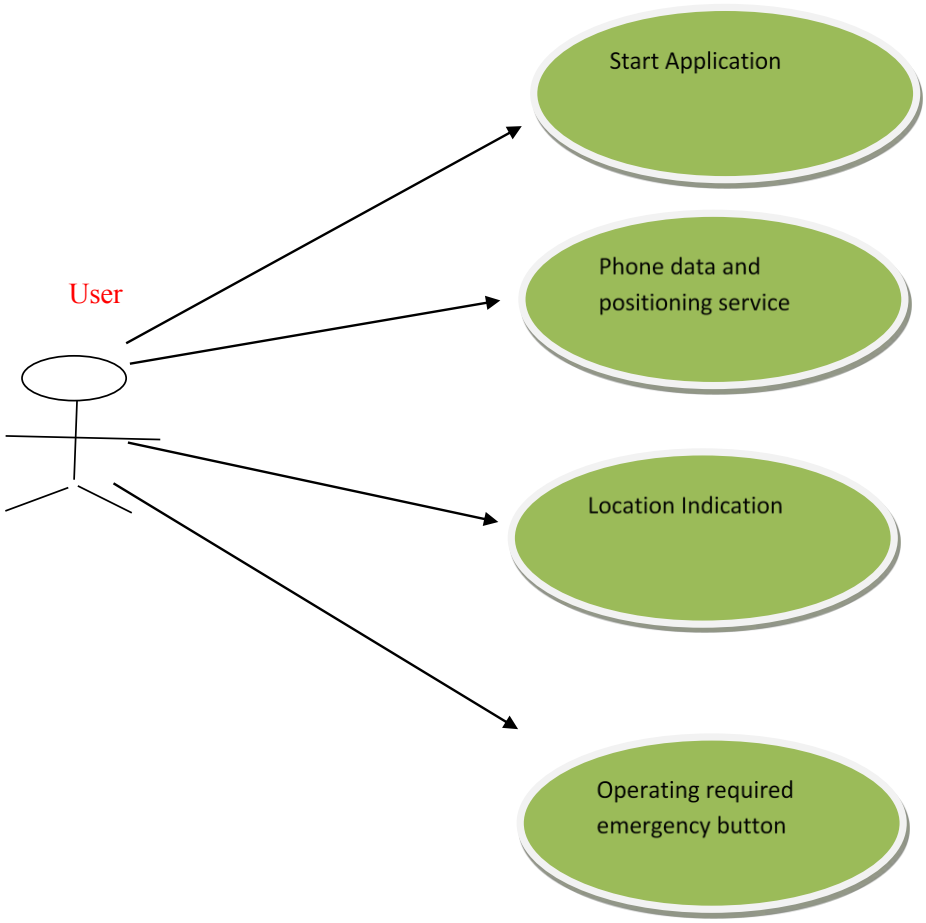


Figure 2.0 Use Case Diagram

2.1 Data Flow Diagram (DFD)

Data Flow Diagrams are a graphical tool used to describe and analyze the movement of data through a system. DFD‟s are used to capture the essential feature of both existing real system and future physical implementation of the system. The DFD is a graphical technique that depicts the information flow and the transforms that are applies as data move from input to the output.

Level-0 DFD shows outline of the system models in Figure 2.1.

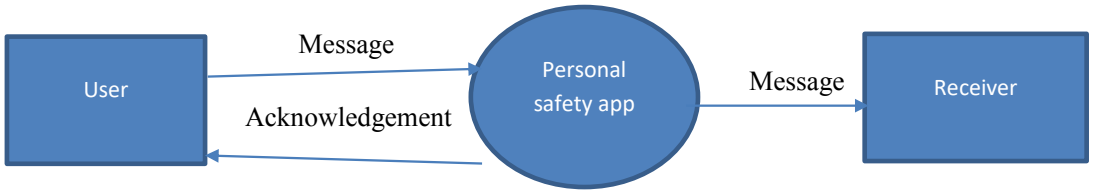


Figure 2.1 Data flow diagram for system models

Level-1 DFD

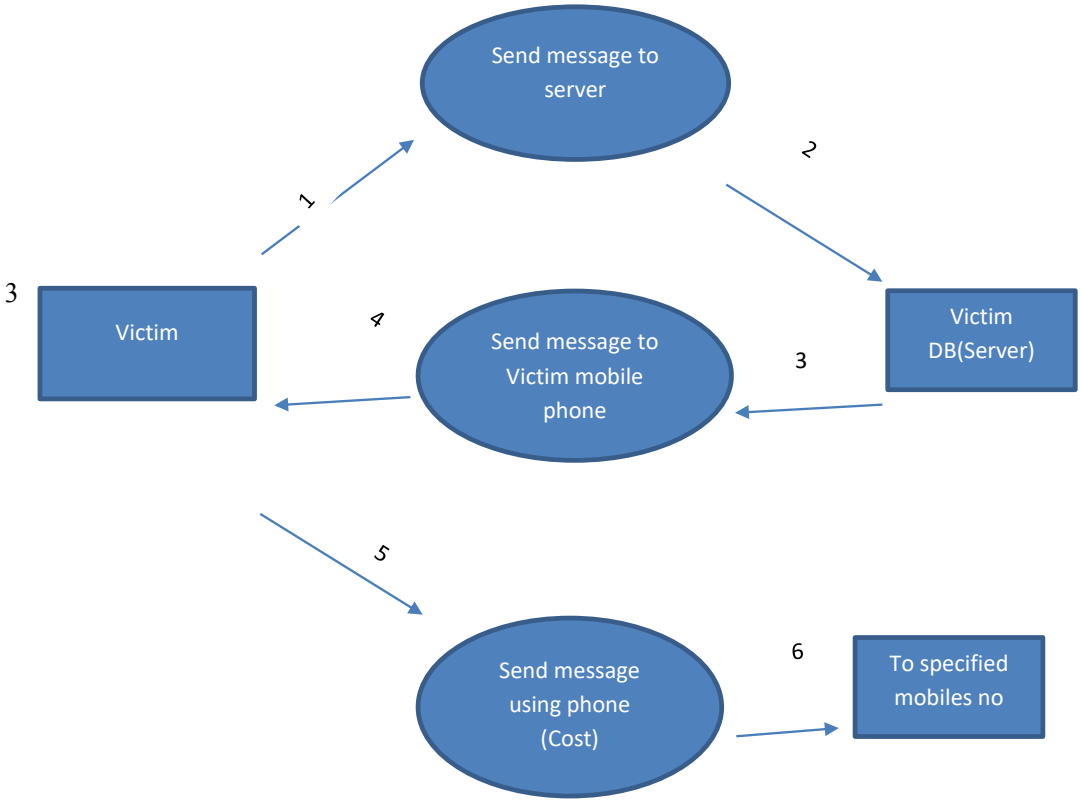
This shows the separation of all external modules, relationship between those modules and the application. This is shown in Figure 2.2.

Figure 2.2 Data flow diagram for modules relationship

Level-2 DFD

This differentiates the modules frontend and backend data flow in Figure 2.3.

(a). Victim user first will send data to server.



(b). Process of sending victim user's data to concern mobile number.

(c). Process of sending victim user's data to concern email address.

Figure 2.3 Frontend and backend differentiation

2.2 Software Building Requirement

A. Functional Requirement

This specification is used to specify the requirements for the initial implementation of the system and update the system in future.The software requirement specification bridges the gap between client/user and the system developer. This is the document that describes the user needs accurately.

B. Performance Requirement

This document will provide general description of the project product perspective, and overview of requirement, general constraint and user view of the product while using. In additional will also provide the specific requirement and functional needs for this project such as interface, functional and performance requirements. The purpose of this software requirement specification is to properly document the requirement of the user necessary in order to build this application.

C. Software Requirement

This system compromises an Android Operating System, using Java as a core language, with Android SDK 2.3 asits version, implementation of front end is done by XML and MySql as back end, the documentation of this system is done using MS Office.

D. Android Platform

Android is an open source, Linux-based software stack created for a wide array of devices and form factors. Android operating system is a stack of software components which is roughly divided into five sections.The following diagram shows the major components of the Android platform.

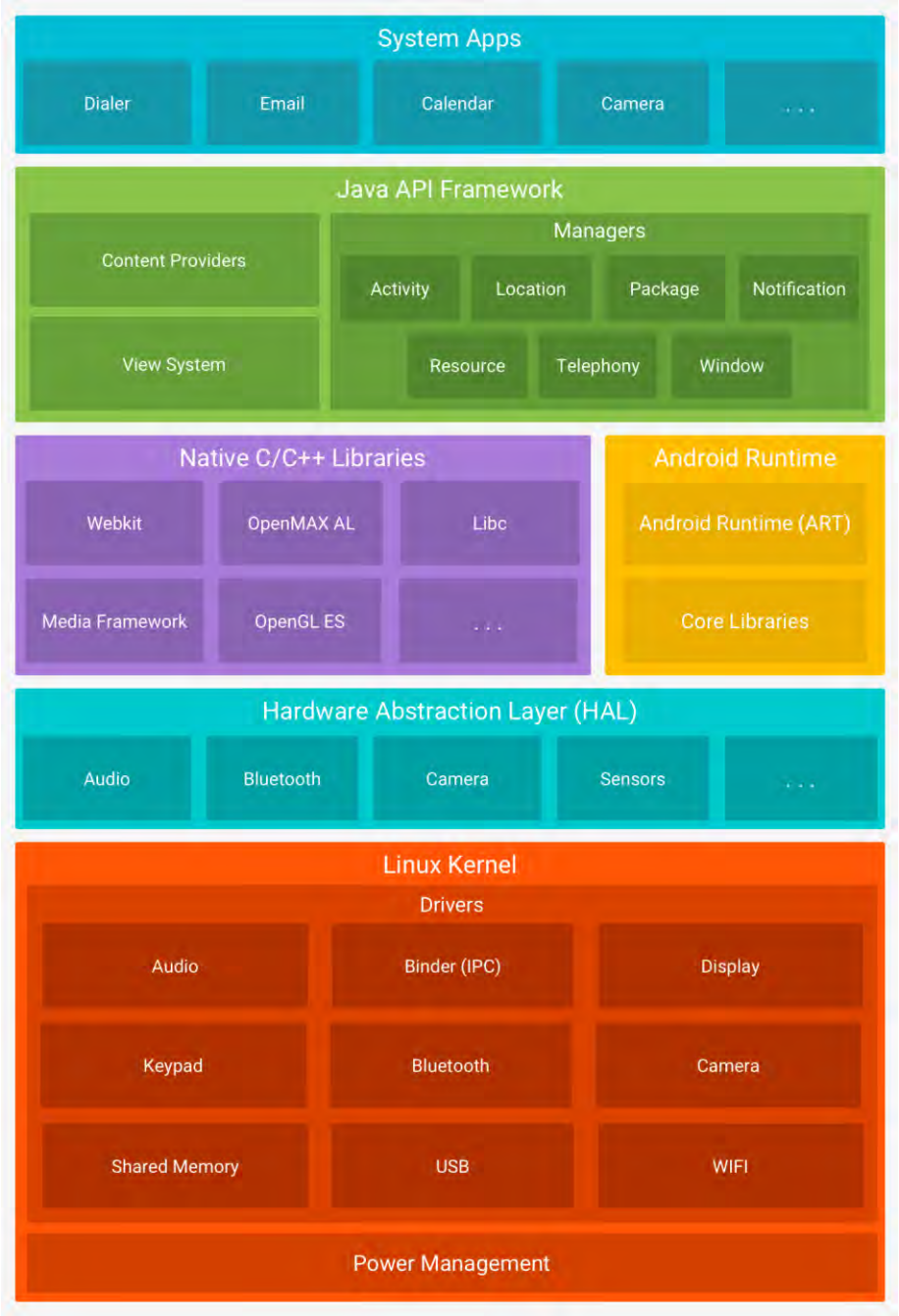


Figure 2.4. The Android software stack.

E. The Linux Kernel

The foundation of the Android platform is the Linux kernel. For example, the Android Runtime (ART) relies on the Linux kernel for underlying functionalities such as threading and low-level memory management.

Using a Linux kernel allows Android to take advantage of key security features and allows device manufacturers to develop hardware drivers for a well-known kernel.

F. Hardware Abstraction Layer(HAL)

The hardware abstraction layer (HAL) provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework. The HAL consists of multiple library modules, each of which implements an interface for a specific type of hardware component, such as the camera or bluetooth module. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component.

G. Android Runtime

For devices running Android version 5.0 (API level 21) or higher, each app runs in its own process and with its own instance of the Android Runtime (ART). ART is written to run multiple virtual machines on low-memory devices by executing DEX files, a bytecode format designed specially for Android that's optimized for minimal memory footprint. Build toolchains, such as Jack, compile Java sources into DEX bytecode, which can run on the Android platform.

Some of the major features of ART include the following:

- Ahead-of-time (AOT) and just-in-time (JIT) compilation
- Optimized garbage collection (GC)
- Better debugging support, including a dedicated sampling profiler, detailed diagnostic exceptions and crash reporting, and the ability to set watchpoints to monitor specific fields

Prior to Android version 5.0 (API level 21), Dalvik was the Android runtime. If your app runs well on ART, then it should work on Dalvik as well, but the reverse may not be true.

Android also includes a set of core runtime libraries that provide most of the functionality of the Java programming language, including some Java 8 language features, that the Java API framework uses.

H. Native C/C++ Libraries

Many core Android system components and services, such as ART and HAL, are built from native code that require native libraries written in C and C++. The Android platform provides Java framework APIs to expose the functionality of some of these native libraries to apps. For example, you can access OpenGL ES through the Android framework‟s Java OpenGL API to add support for drawing and manipulating 2D and 3D graphics in your app.

If you are developing an app that requires C or C++ code, you can use the Android NDK to access some of these native platform libraries directly from your native code.

I. Java API Framework

The entire feature-set of the Android OS is available to you through APIs written in the Java language. These APIs form the building blocks you need to create Android apps by simplifying the reuse of core, modular system components and services, which include the following:

- A rich and extensible View System you can use to build an app‟s UI, including lists, grids, text boxes, buttons, and even an embeddable web browser.
- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files.
- A Notification Manager that enables all apps to display custom alerts in the status bar
- An Activity Manager that manages the lifecycle of apps and provides a common navigation back stack.
- Content Providers that enable apps to access data from other apps, such as the Contacts app, or to share their own data.

Developers have full access to the same framework APIs that Android system apps use.

J. System Apps

Android comes with a set of core apps for email, SMS messaging, calendars, internet browsing, contacts, and more. Apps included with the platform have no special status among the apps the user chooses to install. So a third-party app can become the user's default web browser, SMS messenger, or even the default keyboard (some exceptions apply, such as the system's Settings app).

The system apps function both as apps for users and to provide key capabilities that developers can access from their own app. For example, if one‟s app would like to

deliver an SMS message, he/she does not need to build that functionality oneself. He/she can instead invoke whichever SMS app is already installed to deliver a message to the recipient he/she specifies.

K. Android SDK-API 23

Android is an operating system based on Linux with a Java programming interface. The Android Software Development Kit (Android SDK) provides all necessary tools to develop Android applications. This includes a compiler, debugger and a device emulator, as well as its own virtual machine to run Android programs. Android is primarily developed by Google. Android allows background processing, provides a rich user interface library, supports 2-D and 3-D graphics using the OpenGL libraries, access to the file system and provides an embedded SQLite database. Android application consists of different components and can re-use components of other applications. This leads to *the* concept of a task in Android; an application can re-use other Android components to archive a task.

L. Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

 A flexible Gradle-based build system

 A fast and feature-rich emulator

 A unified environment where you can develop for all Android devices

 Instant Run to push changes to your running app without building a new APK

 Code templates and GitHub integration to help you build common app features and import sample code

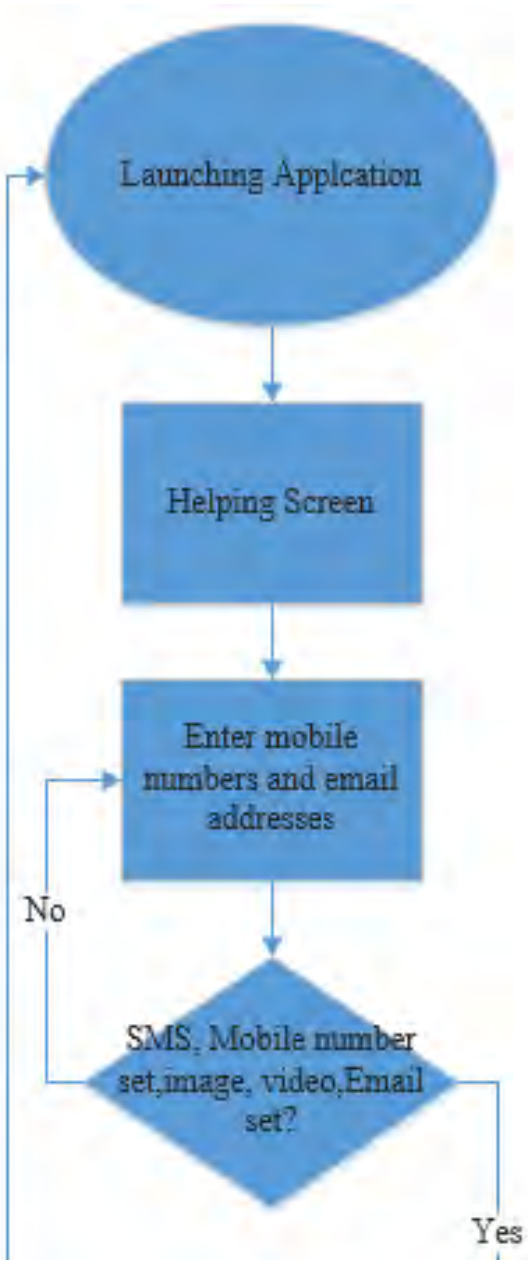
 Extensive testing tools and frameworks

 Lint tools to catch performance, usability, version compatibility, and other problems

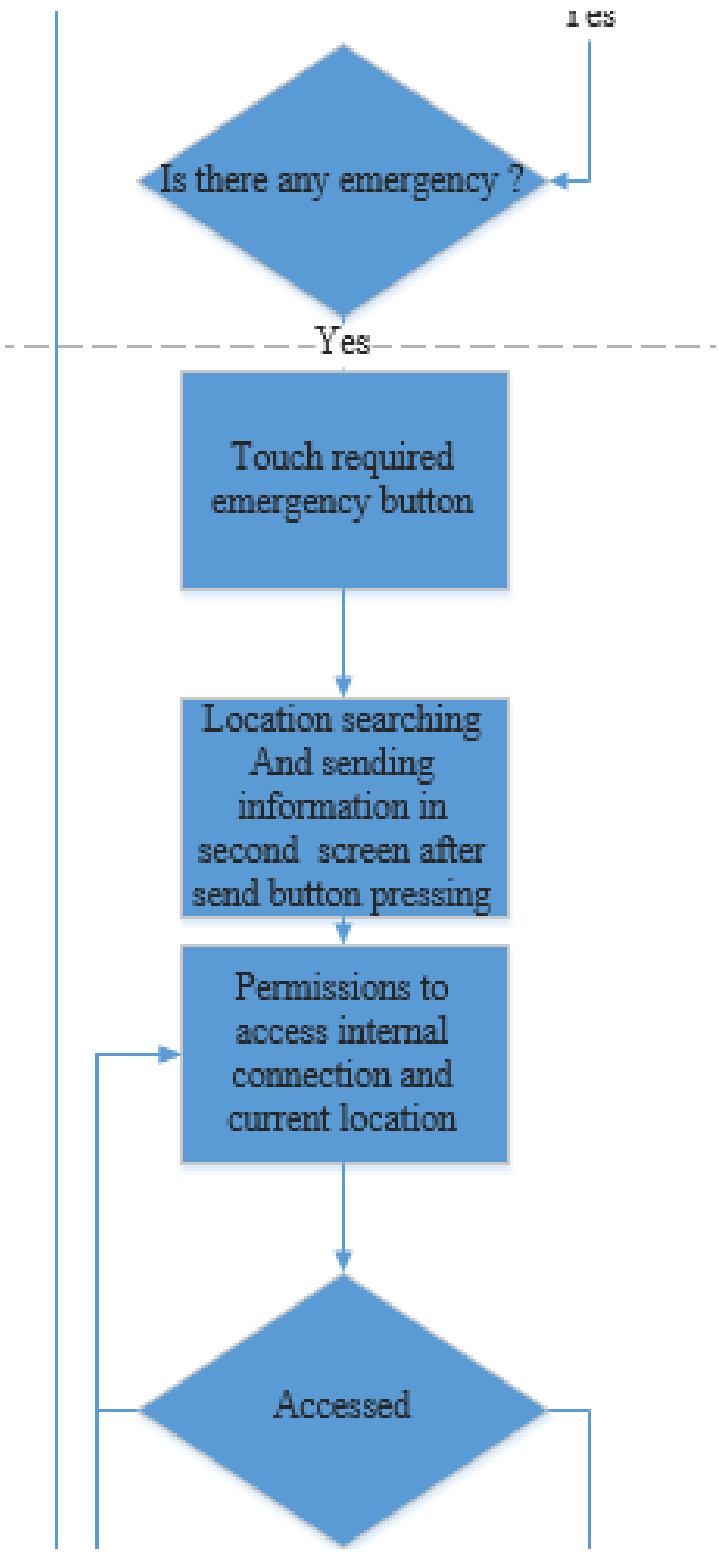
 C++ and NDK support

 Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

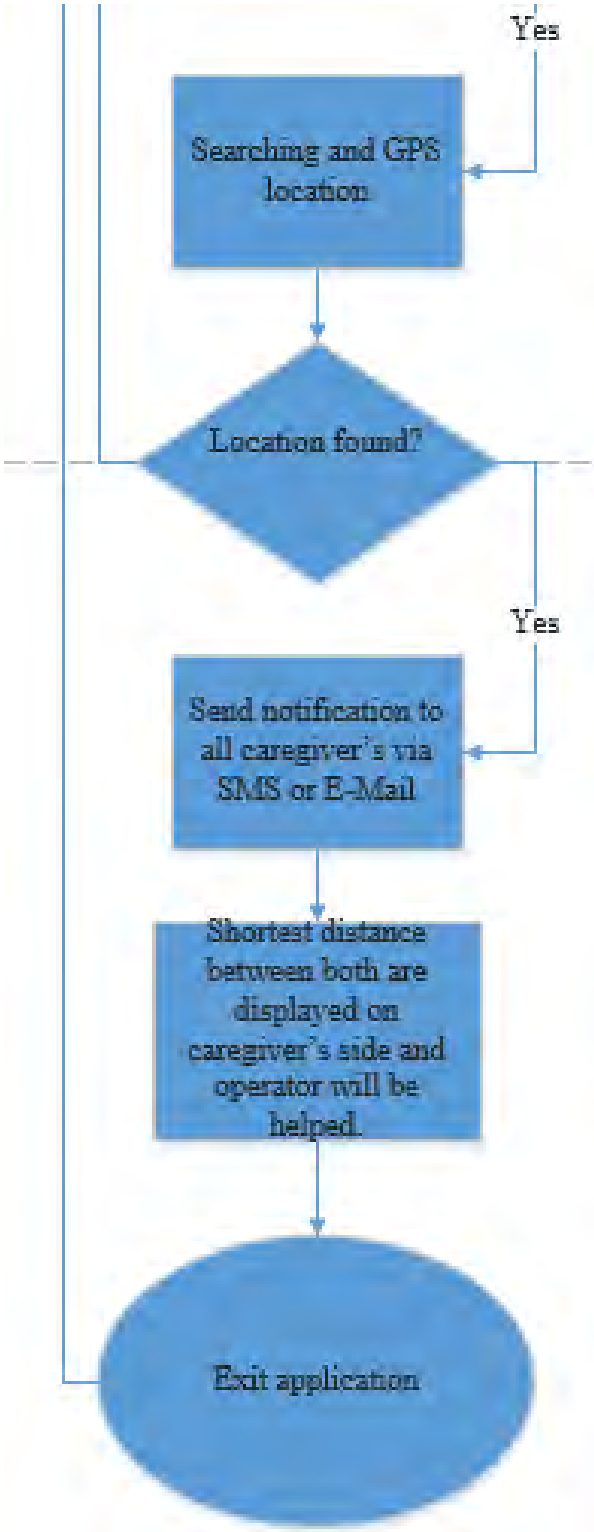
2.3. Application functional Process



Part 1


Yes
Is there any emergency ?
Yes
Touch required emergency button
Location searching And sending information in second screen after send button pressing
Permissions to access internal connection and current location
Accessed

Part 2



Part 3

Figure 2.5 Flow chart for the application functionality

This is user interface where user input mobile number, email address, message, video and photo. When the user press to send button, then the second interface will appear as seen in Figure 2.6.

Figure 2.6 User interface for emergency information set up.

This interface is the sencond activity where location and SMS, Email sent information will appear.

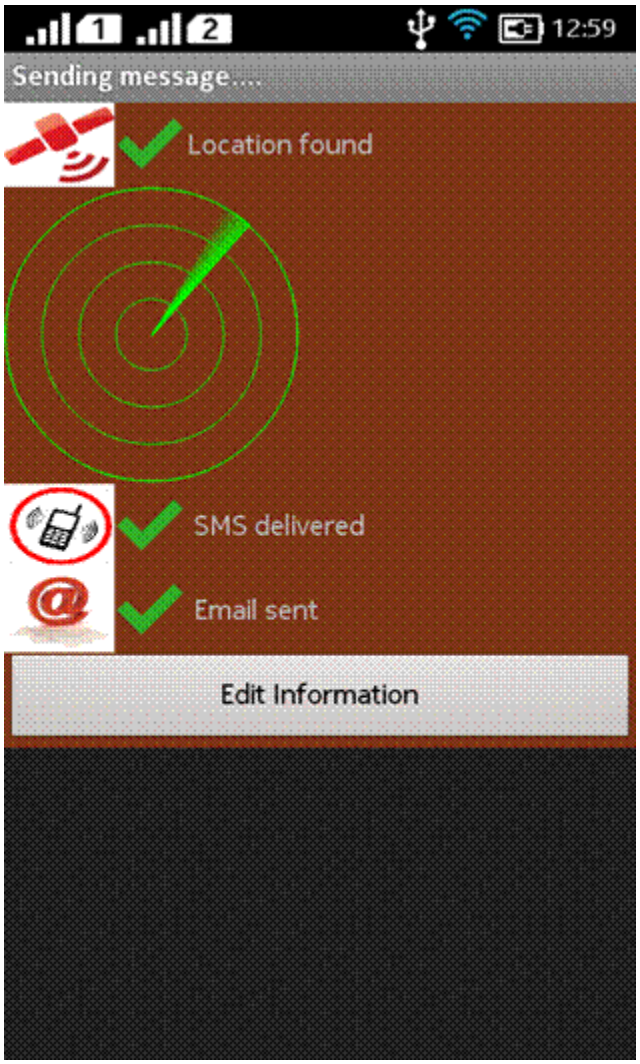


Figure 2.7 Sent information confirmation screen

This is mobile device message inbox screen where message has been received.

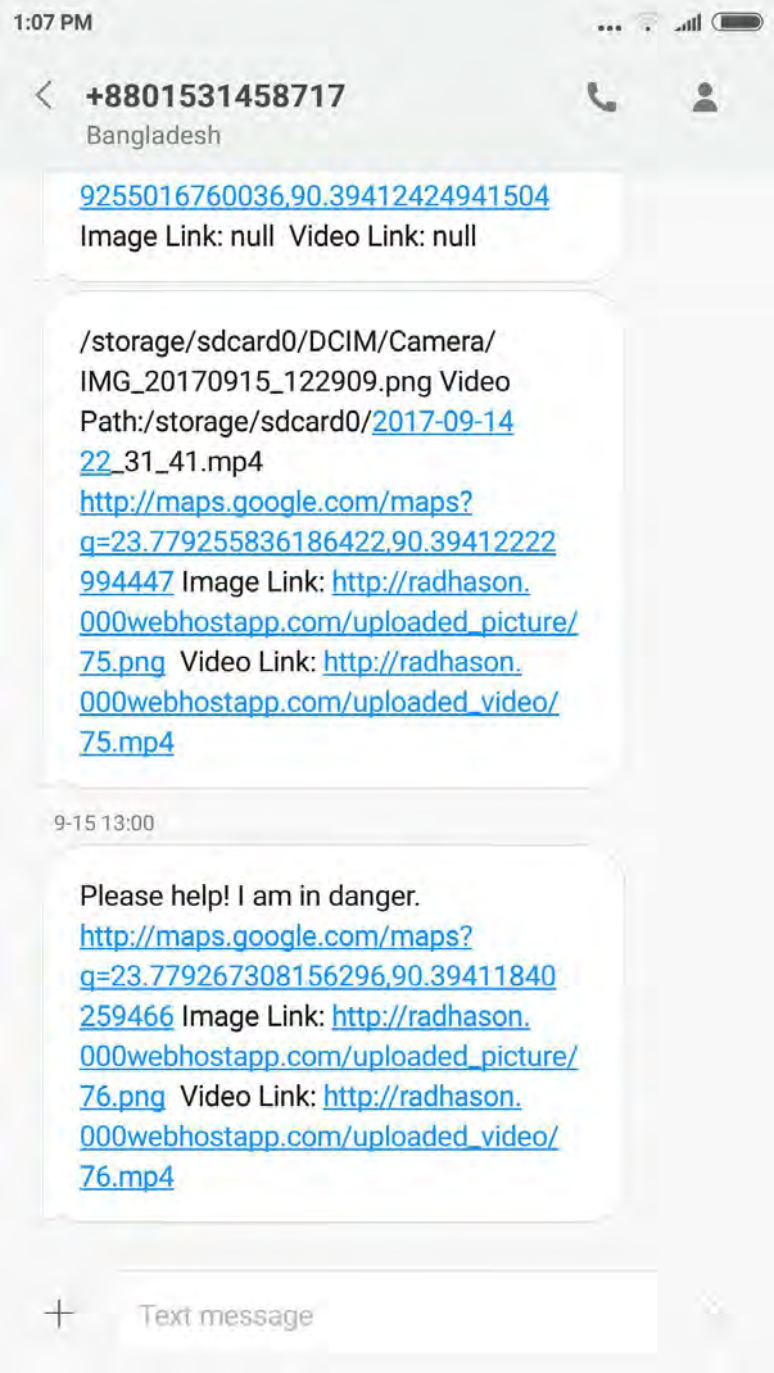


Figure 2.8 Received information screen in mobile

This is inbox of email address of the receiver end where received message has been message has been shown.

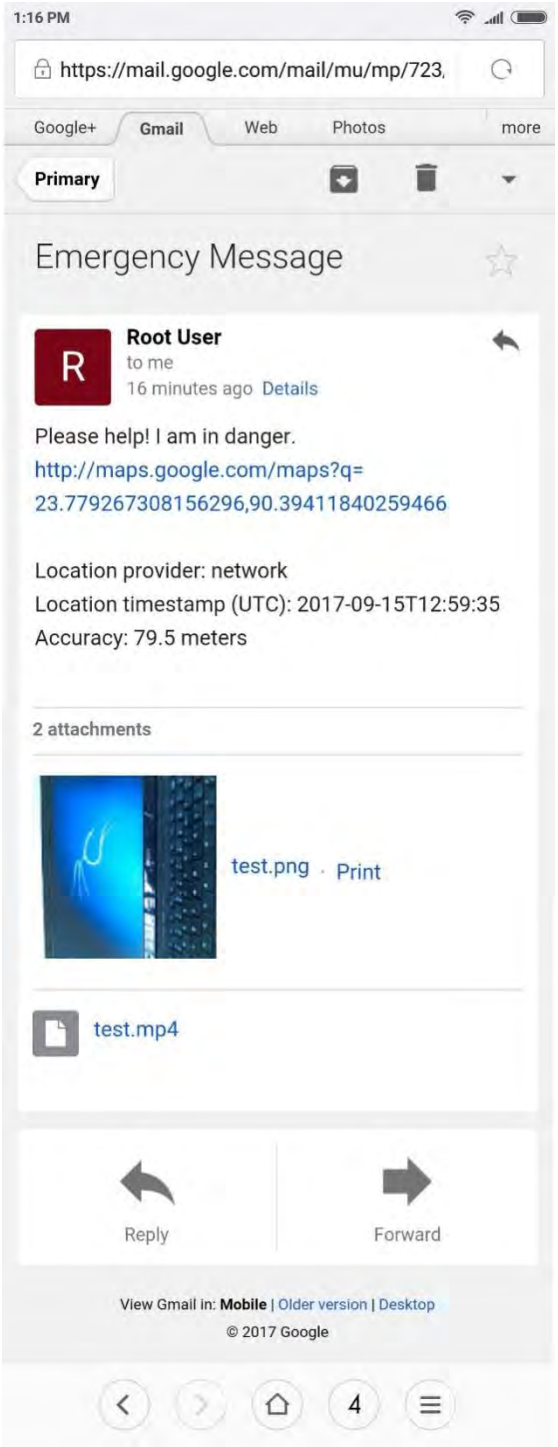


Figure 2.9 Received information screen in email

This is the user location after clicking the victim sent location link. This is shown in Figure 2.10.

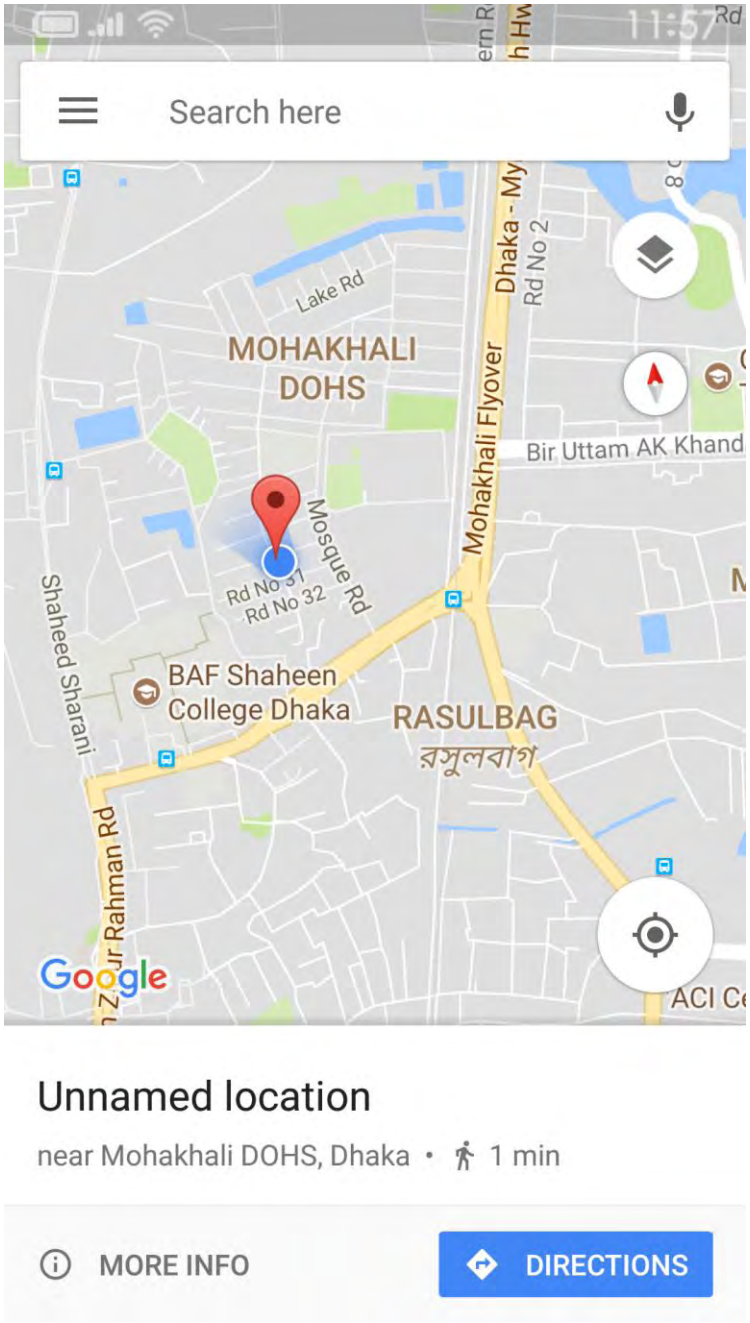


Figure 2.10 Victim location found after clicking on the link.

2.4.Methodology

The methodology includes Use Cases and actions will be used to analyze the requirements and specifications, which will be followed by the actual implementation of the specified functionalities of the Emergency SOSAndroid application.

2.5.Database Design:

This is theinsert_data table structure ofdatabase id1664980_radhason which is created in this link: http://radhason.000webhostapp.com/. Here **000webhost** is the hosting company which gives hosting with a free domain name. **000webhost** has been used for hosting PHP code to get server functionality for this android project.

This table has been used to store text messages, image and video link stored in server.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	tinyint(20)			No	*None*		AUTO_INCREMENT
2	**image**	varchar(100)	utf8_unicode_ci		No	*None*		
3	**name**	varchar(50)	utf8_unicode_ci		No	*None*		
4	**video**	varchar(50)	utf8_unicode_ci		No	*None*		

Figure 2.11 Database design

2.6. Requirement Specifications

Emergency SOS Android application is to be represented in instinctive way which provides the useful functionality to display the operator‟s location, emergency buttons, emergency contacts list, image and video. Internet connection is required to senddata and if the internet is not working on the Android phone, then the application will not work.

2.7. Use Case Diagram

The following user activities are illustrated by using UML tool for the use case [6]. The four main steps for user helps are given inFigure-2.1.

(a). Start Application

The first activity of the use case is Start Application (Table 1) which represents the introduction of Android application. The primary requirement of this is an Android operating system with Internet support.

Table 1

Name	Start Android Application
Performers	Operators
Prerequisites	Android OS with Internet support
Explanation	Start-up activity

(b). Phone Data and Positing Service

This activity is Phone data and positioning service (Table 2) and it describes an attention message to turn on positing service. As when location and phone, internet data are on then the application further function.

Name	Start Android Application
Performers	Operators
Prerequisites	Location and phone, internet data are on
Explanation	Attention activity

Table 2. Phone data and positioning service.

(c). Location Indication

The activity Location indication (Table 3) of the Use Cases includes the operator current location indicated on the Google Map and the emergency buttons Fake Call, Location, Timer, Call& E-Mail are displayed to contact in case of need.

Name	Start Android Application
Performers	Operators
Prerequisites	Operator current location indication and display of Emergency buttons.
Explanation	Current location indication on Google Map

Table 3. Location Indication

(d). Operating Required Emergency Button

Fourth activity operating required emergency button (Table 4) represents the emergency situation by touching required emergency button.

Table 4. Emergency situation.

Name	Start Android Application
Performers	Operators
Prerequisites	Required Emergency button is touched
Explanation	Emergency situation is recognized

CHAPTER-III
Implementation of the project

3.1. SharedPreferences

This SharedPreferences memory is used in this application for storing email, mobile number, image, video, messages.

Interface for accessing and modifying preference data returned by getSharedPreferences(String, int). For any particular set of preferences, there is a single instance of this class that all clients share. Modifications to the preferences must go through a SharedPreferences.Editor objectto ensure the preference values remain in a consistent state and control when they are committed to storage. Objects that are returned from the various getmethods must be treated as immutable by the application.

The SharedPreferences class provides a general framework that allows you to save and retrieve persistent key-value pairs of primitive data types. You can use SharedPreferences to save any primitive data: booleans, floats, ints, longs, and strings. This data will persist across user sessions (even if your application is killed).

To get a SharedPreferences object for your application, use one of two methods:

- getSharedPreferences() - Use this if you need multiple preferences files identified by name, which you specify with the first parameter.
- getPreferences() - Use this if you need only one preferences file for your Activity. Because this will be the only preferences file for your Activity, you do not supply a name.

To write values:

1. Call edit() to get a SharedPreferences.Editor.
2. Add values with methods such as putBoolean() and putString().
3. Commit the new values with commit()

To read values, use SharedPreferences methods such as getBoolean() and getString().

The sample code for using SharedPreferences is provided in Appendix A.1

3.2. XML

XML is used for design user interface of this application.

Example:

Main.xml interface code of this application as shown in Figure 2.5 User interface for emergency information set up.

The sample code for using XML is provided in Appendix A.2

XML stands for Extensible Markup Language and is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

3.3. PHPMailer

PHPMailerlibray is used to send email to the client end. This is PHP library.

Class Features

- Probably the world's most popular code for sending email from PHP!
- Used by many open-source projects: WordPress, Drupal, 1CRM, SugarCRM, Yii, Joomla! and many more
- Integrated SMTP support - send without a local mail server
- Send emails with multiple TOs, CCs, BCCs and REPLY-TOs
- Multipart/alternative emails for mail clients that do not read HTML email
- Support for UTF-8 content and 8bit, base64, binary, and quoted-printable encodings

- SMTP authentication with LOGIN, PLAIN, NTLM, CRAM-MD5 and Google's XOAUTH2 mechanisms over SSL and TLS transports
- Error messages in 47 languages!
- DKIM and S/MIME signing support
- Compatible with PHP 5.0 and later

Many PHP developers utilize email in their code. The only PHP function that supports this is the mail() function. However, it does not provide any assistance for making use of popular features such as HTML-based emails and attachments.

The PHP mail() function usually sends via a local mail server, typically fronted by a sendmail binary on Linux, BSD and OS X platforms, however, Windows usually does not include a local mail server; PHPMailer's integrated SMTP implementation allows email sending on Windows platforms without a local mail server.

Example: This code has been used in server end:

The sample code for using PHP Mailer is provided in Appendix A.3

3.4. LocationManger:

In order to fetch the current position of the user it is needed to obtain the longitude and latitude coordinate values. All the applications should carry an AndroidManifest.xml file (with precisely that name) in its root directory, since the essential information about the application will be held by the manifest. Hence the protected part of APIs can be accessed. To expend the GPS functionality, ACCESS_FINE_LOCATION is added toget permission to Androidmanifest.xml. To obtain the coordinate values, a location manager has to be created. Location manager is the part that is responsible for creating a location based service on Android. The snippet code below shows how the location manager can be created

```
LocationManagerlocationManager=(LocationManager)context.getSystemService(Context.LOCATION_SERVICE);
```

After fetching the longitude and latitude coordinate values from GPS, a SMS message which contains the location information is generated. In order to send a SMS message, SEND_SMS permission is added to the AndroidManifest.xml file which in turn makes the application to send SMS message using sendSMS() function thereby initiating the smsManager to allow the application to send a SMS message.

3.5. Base64 Encoding:

This encoding is used to convert image and video into String.

The sample code for using Base64 Encodingis provided in Appendix A.4

3.6. Base64 Decoding:

In Server end Base64 Decoding has been used to convert String to image and video and send to specific email address.

The sample code for using Base64 Decoding is provided in Appendix A.5

3.7. Service:

Service has been used to start the application using shaking the Android device.To implement a feature that activates when the user shakes his/her Android Phone. Say you want to reload the Activity contents when user shakes the phone.To implement such functionality you need to create a listener that is related to motion and the hardware that deals with motion is called the Accelerometer.Accelerometer is a motion sensor (hardware) that is built in to your Android device that can measure motion and orientations. They provide us raw data with high precision and accuracy. Most of the Gaming Application make use of this sensor.

The sample code for using Service classes provided in Appendix A.6

CHAPTER-IV

Conclusion and Future works

4.1. Important features of the software

In this report, a simple, precise, and efficient tracking system is presented. An emergency program is designed to receive latitude and longitude coordinates automatically at start. Any time in a day, when a user encounters a danger, then he/she sends call for help to the registered numbers via one touch. This one touch is theSEND Button. User interface of designed program is illustrated in Fig.2.5. The system sends coordinate information and help text by pressing theSEND Button. The application of web-based GPS technology greatly increases the system‟s intelligence making the location monitoring more transparent and efficient in emergency situations. The application can transmit messages along with the location details in case of any emergency.The application server can interact with the mobile device and send the message accordingly. The application can provide immediate video recording and image capturing in case of any emergency situation. This will help the user for future use if any crime occurs.

The application can be launched using shaking the Android device and after auto starting the application can capture image and auto send image, SMS, video, GPS location link addresses to registered email addresses and mobile numbers.

4.2. Limitations of the software

This developed application can be used to provide security to users. It helps to efficiently increase the security and safety of all people. However, the developed application is unable to record video automatically after shaking the devices. The developed application does not effectively provide siren or warning sound when the user wants.

4.3 Future Work

The web configuration application will be created in PHP, which is suitable for modular application development. Here it is possible to configure the mobile numbers, on which will be send the alert in the triggering the warning.In future the application will be modified to provide siren or warning sound when the user wants.

References

[1] Mandapati1, S., Pamidi, S., Ambati, S., "A Mobile Based Women Safety Application (I Safe Apps)", IOSR Journal of Computer Engineering, vol. 17, no. 1, Feb. 2015.

[2] Gogoi, D., Sharma, R. K., "Android Base Emergency Alert Button", International Journal of Innovative Technology and Exploring Engineering, vol. 2, no. 4, Mar. 2013.

[3] Barge, A., Belokar, N, V., Khodave, S., Gosavi, P., "I am Secure Android Application", International Journal of Research in Engineering and Technology, vol. 3, no. 4, Apr. 2014.

[4]Thakare, R. T., Shahade, M. R., "Review On Android Based SOS Emergency Alert Button", International Journal of Research In Science & Engineering, vol. 1, May 2015.

[5]https://developer.android.com/reference/android/location/LocationManager.html, last access date 27/01/2017

[6]https://play.google.com/store/apps/details?id=com.xankle.mobile.sos&hl=en, last access date

27/01/2017.

[7] https://play.google.com/store/apps/details?id=com.kovan.sos, last access date

27/01/2017.

[8] https://play.google.com/store/apps/details?id=com.sosemergencybutton, last access date

27/01/2017.

[9]Varma, K., Baikar, P., Mangale, S., "Human Surveillance with Android based S.M.A.R.T Emergency Application", International Journal of Innovative Research in Computer and Communication Engineering , vol. 3, no. 4, Apr. 2015.

[10]. World Health Organization, "Global and regional estimates of violence against women: prevalence and health effects of intimate partner violence and non-partner sexual violence, Geneva: World Health Organization (2013).

[11]. Nicole Westmarland, MariannHardey, et al. “Protecting Women‟s Safety? The use of smartphone „apps‟ in relation to domestic and sexual violence‟, Durham University, Durham centre for research into violence and abuse (2013).

[12]. Mason, C and Shoshana Magnet, “Surveillance studies and violence against women”, surveillance & society 10(2):105 – 118 (2012).

[13]. Justine A. Dunlap, Intimate Terrorism and Technology: There's an App for That, 7 U. Mass. L. Rev. 10 (2012).

[14]. Elliott, J. and McCartan K. The Reality of Trafficked People‟s Access to Technology. The Journal of Criminal Law 77(3): 255- 273(2013).

[15]. Dimond, J.P., Fiesler, C. and Bruckman, A.S. Domestic violence and information communication technologies, Interacting with Computers, 23(5): 413-421(2011).

[16]. Portio Research. Portio research mobile factbook 2011. Chippenham, UK. Retrieved from http://www.portioresearch.com/Portio%20Research%20Ltd%20Mobile%20Factbook%202011.pdf (2011).

[17]. Ofcom. Children and Parents: Media Use and Attitudes Report (2012).

[18]. Dennison, L., Morrison, L., Conway, G., & Yardley, L. (2013).Opportunities and Challenges for Smartphone Applications in Supporting Health Behavior Change: Qualitative Study, Journal of medical Internet research, 15(4): e86.

[19]. Qualcomm Wireless ReachTM, Vital Wave, and GSMA. Transforming Women‟s Livelihoods Through Mobile Broadband. Report, February 2014. Available at: http://www.qualcomm.com/mwomenreport

[20]. Vodafone. “Connected Women: How mobile can support women‟s economic and social empowerment”. Report, March, 2014

APPENDIX

Code Segment of the Developed Software

A.1. Code for using SharedPreferences

```
public class EmergencyData {

Context mContext;
SharedPreferencesmSettings;

public static final String PREFS_NAME = "EmergencyPrefsFile";
public static final String EMAIL = "emailAddress";
public static final String PHONE = "phoneNo";
public static final String MESSAGE = "message";
public static final String SEND_EMERGENCY = "sendEmergency";

String nameImage="name";
String video="video";
String IMAGE="image";
Bitmap bitmap;
ImageViewimgPreview;

String selectedImagePath;
String getSelectedVideoPath;
private static final String COUNT_SUFFIX = "Count";
private String name;

public EmergencyData(Context context) {
this.mContext= context;
this.mSettings= context.getSharedPreferences(PREFS_NAME, Context.MODE_PRIVATE);
  }

private intgetCount(String id) {
return mSettings.getInt(id + COUNT_SUFFIX, 1);
  }

private void setCount(SharedPreferences.Editor editor, String id, intcount) {
editor.putInt(id + COUNT_SUFFIX, count);
  }

private List<String>getList(String id) {
// NOTE: this function should always return a list with at least one
    //   string.
List<String>stringsList = new ArrayList<String>();
```

```
String lastString;
lastString = mSettings.getString(id, "");
stringsList.add(lastString);

intstringsCount = getCount(id);
for(inti = 1; i <stringsCount; i++) {
// NOTE: the first item is just "id" and then next is "id1"
        //    this is for backwards compatibility, oh well...
lastString = mSettings.getString(id + i, "");
stringsList.add(lastString);
     }

return stringsList;
   }

private void commitList(String id, List<String>stringList) {
SharedPreferences.Editor editor = mSettings.edit();
intpreviousCount = getCount(id);
inti = 0;
for (; i <stringList.size(); i++) {
        String idName;
if (i >0) {
idName = id + i;
        } else {
idName = id;
        }
editor.putString(idName, stringList.get(i));
     }

for(; i <previousCount; i++) {
// remove older irrelevant strings
editor.remove(id + i);
     }

setCount(editor, id, stringList.size());
editor.commit();
   }
public String encodeToString(){
selectedImagePath=mSettings.getString(nameImage, "");

bitmap = BitmapFactory.decodeFile(selectedImagePath);
bitmap=Bitmap.createScaledBitmap(bitmap, 100, 100, true);
ByteArrayOutputStream stream = new ByteArrayOutputStream();
bitmap.compress(Bitmap.CompressFormat.PNG, 100, stream);
byte[] byteFormat = stream.toByteArray();
     String imgString = Base64.encodeToString(byteFormat, Base64.DEFAULT);
```

```
return imgString;
  }
public List<String>getEmails() {
return getList(EMAIL);
  }
public List<String>getPhones() {
return getList(PHONE);
  }
public String getMessage() {
return mSettings.getString(MESSAGE, "");
  }
public booleangetArmEmergency() {
return mSettings.getBoolean(SEND_EMERGENCY, false);
  }
public void getImage(){


selectedImagePath=mSettings.getString(nameImage, "");
bitmap = BitmapFactory.decodeFile(selectedImagePath);
imgPreview.setImageBitmap(bitmap);

  }

public String getImagePath(){


selectedImagePath=mSettings.getString(nameImage, "");
//      bitmap = BitmapFactory.decodeFile(selectedImagePath);
//      imgPreview.setImageBitmap(bitmap);
return selectedImagePath;

   }
public  String getVideoPath(){
getSelectedVideoPath=mSettings.getString(video, "");
//      bitmap = BitmapFactory.decodeFile(selectedImagePath);
//      imgPreview.setImageBitmap(bitmap);
return getSelectedVideoPath;
  }

private void commitString(String id, String value) {
// We need an Editor object to make preference changes.
    // All objects are from android.context.Context
SharedPreferences.Editor editor = mSettings.edit();
editor.putString(id, value);
editor.commit();
  }
```

```
public void setEmails(List<String> emails) {
commitList(EMAIL, emails);
   }
public void setPhones(List<String> phones) {
commitList(PHONE, phones);
   }
public void setMessage(String message) {
commitString(MESSAGE, message);
   }

public void setArmEmergency(booleansendMessage) {
// We need an Editor object to make preference changes.
     // All objects are from android.context.Context
SharedPreferences.Editor editor = mSettings.edit();
editor.putBoolean(SEND_EMERGENCY, sendMessage);
editor.commit();
   }
public void setImage(Uri imaPath){

SharedPreferences.Editor editor = mSettings.edit();
editor.putString(nameImage, imaPath.getPath());
//     editor.putString(nameImage, imaPath.toString());
// editor.putString(IMAGE, encodeTobase64(yourbitmap));
editor.commit();
}

public void setVideoPath(Uri videoPath){

SharedPreferences.Editor editor = mSettings.edit();
editor.putString(video, videoPath.getPath());

//     editor.putString(nameImage, imaPath.toString());
// editor.putString(IMAGE, encodeTobase64(yourbitmap));
editor.commit();
   }

public static String encodeTobase64(Bitmap image) {

        String imageEncoded = null;
if (image != null) {
System.out.println(image + "Radhason");
        Bitmap immage = image;
ByteArrayOutputStreambaos = new ByteArrayOutputStream();
immage.compress(Bitmap.CompressFormat.PNG, 100, baos);
byte[] b = baos.toByteArray();
```

```
imageEncoded = Base64.encodeToString(b, Base64.DEFAULT);

Log.d("Radhason", imageEncoded);
    }
return imageEncoded;
    }



public void setName(String name) {
this.name = name;
  }

public String getName() {
return name;
  }
}
```

A.2. Code for using XML

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollViewxmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:background="@color/colorBlood"

>
<LinearLayout
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<RelativeLayout
android:orientation="horizontal"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingTop="6dp"
android:paddingRight="10dp"
>

<ImageButton
android:id="@+id/btnHelp"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
```

```
android:background="@drawable/question_small"
android:layout_alignParentRight="true"
/>
</RelativeLayout>

<ImageButton
android:id="@+id/btnEmergency"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:background="@drawable/emergency_button"
android:layout_gravity="center"
android:layout_centerHorizontal="true"
android:layout_marginBottom="15dp"
/>
<LinearLayout
android:id="@+id/linEmail"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
>
<LinearLayout
android:orientation="horizontal"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:paddingRight="10dp"
>
<ImageView
android:layout_width="wrap_content"
android:layout_height="wrap_content"

android:src="@drawable/main_email_color"

/>
<EditText
android:id="@+id/txtEmail"
android:layout_weight="1"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:inputType="textEmailAddress"
android:hint="Email"
/>

<ImageButton
android:id="@+id/btnEmailPlus"
android:layout_height="wrap_content"
android:layout_width="wrap_content"
```

```
android:background="@drawable/grey_plus"
android:layout_gravity="center"
/>
</LinearLayout>
</LinearLayout>

<!-- Note that the special padding for this TextView is to compensate for having no capital
letters -->
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:includeFontPadding="false"
android:text="- and/or -"
android:gravity="center"
android:textSize="20sp"
android:paddingTop="0sp"
android:paddingBottom="3sp"
/>

<LinearLayout
android:id="@+id/linPhoneNo"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingRight="10dp"
>
<LinearLayout
android:orientation="horizontal"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<ImageView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/phone_main_color"
/>
<EditText
android:id="@+id/txtPhoneNo"
android:layout_weight="1"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:inputType="phone"
android:hint="SMS Number"
/>
<ImageButton
android:id="@+id/btnPhoneNoPlus"
```

```
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:background="@drawable/grey_plus"
android:layout_gravity="center"
/>
</LinearLayout>
</LinearLayout>

<LinearLayout
android:id="@+id/linMessage"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:paddingRight="10dp"
>
<LinearLayout
android:orientation="horizontal"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<ImageView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/message_sms"
/>
<EditText
android:id="@+id/txtMessage"
android:hint="Message"
android:layout_width="fill_parent"
android:layout_height="120px"
android:gravity="top"
/>
</LinearLayout>
</LinearLayout>


<LinearLayout
android:id="@+id/linearl"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="#ffffff"
android:orientation="vertical" >
```

```
</LinearLayout>
<FrameLayout
android:id="@+id/camera_preview"
android:layout_width="fill_parent"
android:layout_height="400dp"
>
</FrameLayout>
<LinearLayout
android:id="@+id/linearLayout1"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:gravity="center"
android:orientation="horizontal" >

<Button
android:id="@+id/mybutton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="Record"
/>

<Button
android:id="@+id/ibCapture"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="Capture"
/>
<FrameLayout
android:id="@+id/flBtnContainer"
android:layout_width="wrap_content"
android:layout_height="fill_parent">


</FrameLayout>

<Button
android:id="@+id/ibRetake"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="Retake"
android:visibility="gone" />
```

```
<Button
android:id="@+id/ibUse"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="Use"
android:visibility="gone" />

</LinearLayout>

</LinearLayout>

</ScrollView>
```

A.3. Code for using PHP Mailer

```
<?php

//require(/public_html/PHPMailer/class.phpmailer.php);

include_once( '../public_html/PHPMailerAutoload.php');

include_once( '../public_html/PHPMailer/class.phpmailer.php');

$con=mysqli_connect("mysql9.000webhost.com","a7612731_ma","Srimatiradha1","a76127
31_radha");

if (mysqli_connect_errno($con))

{

echo '{"query_result":"ERROR"}';

}
```

```
$fullName = $_POST['fullname'];

$description = $_POST['description'];

$passWord = $_POST['password'];

$phoneNumber = $_POST['phonenumber'];

$emailAddress = $_POST['emailaddress'];

$spinner=$_POST['spinner'];

$image=$_POST['image'];

$video=$_POST['videofile'];


//Create a new PHPMailer instance

$mail = new PHPMailer();


$mail->IsMail();

//$mail->setFrom($description, 'SabujBiswas(Radhason)');

$mail->AddAddress($emailAddress);

$mail->Subject = "Emergency Message";

$mail->Body = $fullName;

$var=base64_decode($video);

$fp = fopen('2013-02-21_14-52-35_968.mp4', 'w');
```

```
fwrite($fp,$var);

fclose($fp);


$videofile= "data:video/mp4; base64,$video";

$data_video= substr($videofile, strpos($videofile, ","));

$filevideo="test.mp4";

$encodingvideo = "base64";

$typevideo = "video/mp4";


$contact_image_data="$image";

$data = substr($contact_image_data, strpos($contact_image_data, ","));

$filename="test.png";

$encoding = "base64";

$type = "image/png";

if($_POST['image']!=null){

$mail->AddStringAttachment(base64_decode($data), $filename, $encoding, $type);

}

if($_POST['videofile']!=null){
```

```
$mail->AddStringAttachment(base64_decode($data_video), $filevideo, $encodingvideo ,
$typevideo);

}

if(!$mail->Send())

{

echo "Error sending: " . $mail->ErrorInfo;;

}

else

{

echo "Letter sent";

}

$result = mysqli_query($con,"INSERT INTO user (fullname, description,  phone,
email,spinner,image)

        VALUES ('$fullName', '$description', '$phoneNumber',
'$emailAddress','$spinner','$image')");



if($result == true) {

echo '{"query_result":"SUCCESS"}';

}

else{
```

```
echo '{"query_result":"FAILURE"}';

}

mysqli_close($con);

?>
```

A.4. Code for using Base64 Encoding

```
        bitmap = BitmapFactory.decodeFile(imagePath);
        bitmap = Bitmap.createScaledBitmap(bitmap, 200, 200, true);
ByteArrayOutputStream stream = new ByteArrayOutputStream();
bitmap.compress(Bitmap.CompressFormat.PNG, 100, stream);

byte[] byteFormat = stream.toByteArray();
final String image = Base64.encodeToString(byteFormat, Base64.DEFAULT);



try {
System.out.println("filepath" + videoPath);
objFileIS = new FileInputStream(videoPath);
} catch (FileNotFoundException e) {
e.printStackTrace();
}
ByteArrayOutputStreamobjByteArrayOS = new ByteArrayOutputStream();
byte[] byteBufferString = new byte[1024000];
try {
for (intreadNum; (readNum = objFileIS.read(byteBufferString)) != -1; ) {
```

```
objByteArrayOS.write(byteBufferString, 0, readNum);



  }
} catch (IOException e) {
e.printStackTrace();
}

videodata = Base64.encodeToString(byteBufferString, Base64.DEFAULT);
```

This String is then sent using sendPostRequest method.

```
param.put(„videofile",videodata);

param.put('image', image);

result = rh.sendPostRequest(urlAddress, param);
```

A.5. Code for using Base64 Decoding

```
$image=$_POST['image'];

$video=$_POST['videofile'];

$mail = new PHPMailer();


$mail->IsMail();
//$mail->setFrom($description, 'SabujBiswas(Radhason)');
$mail->AddAddress($emailAddress);
$mail->Subject = "Emergency Message";
$mail->Body = $fullName;
```

```
$var=base64_decode($video);
$fp = fopen('2013-02-21_14-52-35_968.mp4', 'w');
fwrite($fp,$var);
fclose($fp);

$videofile= "data:video/mp4; base64,$video";
$data_video= substr($videofile, strpos($videofile, ","));
$filevideo="test.mp4";
$encodingvideo = "base64";
$typevideo = "video/mp4";

$contact_image_data="$image";
$data = substr($contact_image_data, strpos($contact_image_data, ","));
$filename="test.png";
$encoding = "base64";
$type = "image/png";
if($_POST['image']!=null){
$mail->AddStringAttachment(base64_decode($data), $filename, $encoding, $type);
}
if($_POST['videofile']!=null){
$mail->AddStringAttachment(base64_decode($data_video), $filevideo, $encodingvideo ,
$typevideo);
}

if(!$mail->Send())
{
echo "Error sending: " . $mail->ErrorInfo;;
}
else
{
echo "Letter sent";
```

A.6. Code for using Service class

```
SensorManagermSensorManager=(SensorManager)getSystemService(SENSOR_SERVICE);
        List listOfSensorsOnDevice=mSensorManager.getSensorList(Sensor.TYPE_ALL);
for(int i =0; i<listOfSensorsOnDevice.size(); i++){
if(listOfSensorsOnDevice.get(i).getType()==Sensor.TYPE_ACCELEROMETER){

Toast.makeText(this,"TYPE_ACCELEROMETER sensor is available on
device",Toast.LENGTH_SHORT).show();
}
else{

Toast.makeText(this,"TYPE_ACCELEROMETER sensor is NOT available on
device",Toast.LENGTH_SHORT).show();
}
}
```

SensorManager is the class that we have to make use of to detect sensor availability. Just add the above code in the onCreate() function of the launcher Activity it check if ACCELEROMETER sensor is available.

The code has been used in this application given below.

```
package com.iictbuet.pgd0413311001;
import android.app.Notification;
import android.app.NotificationManager;
```

```
import android.app.PendingIntent;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.hardware.Camera;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Handler;
import android.os.IBinder;
import android.support.v4.app.NotificationCompat;
import android.util.Log;
import android.view.View;
import android.widget.Toast;

import static android.content.ContentValues.TAG;
import static com.iictbuet.pgd0413311001.R.id.ibCapture;
public class MyServiceextends Service implements SensorEventListener {
public SensorManagermSensorManager;
public Sensor mAccelerometer;
public float mAccel; // acceleration apart from gravity
public float mAccelCurrent; // current acceleration including gravity
public float mAccelLast; // last acceleration including gravity
@Override
public IBinderonBind(Intent intent) {
return null;
  }

@Override
public intonStartCommand(Intent intent, intflags, intstartId) {
```

```
mSensorManager= (SensorManager) getSystemService(Context.SENSOR_SERVICE);
mAccelerometer= mSensorManager
.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
mSensorManager.registerListener(this, mAccelerometer,
SensorManager.SENSOR_DELAY_UI, new Handler());
//    return START_NOT_STICKY;
return START_STICKY;
  }
@Override
public void onAccuracyChanged(Sensor sensor, intaccuracy) {
  }
@Override
public void onSensorChanged(SensorEvent event) {
EmergencyButtonActivityemergencyButtonActivity = new EmergencyButtonActivity();
float x = event.values[0];
float y = event.values[1];
float z = event.values[2];
mAccelLast= mAccelCurrent;
mAccelCurrent= (float) Math.sqrt((double) (x * x + y * y + z * z));
float delta = mAccelCurrent- mAccelLast;
mAccel= mAccel* 0.9f + delta; // perform low-cut filter

if (mAccel>11) {
EmergencyActivity.armEmergencyActivity(getApplicationContext());
System.out.println("radhasonobject"+getApplicationContext());
//        Intent myIntent = new Intent(getApplicationContext(), EmergencyActivity.class);
Intent myIntent = new Intent(getApplicationContext(), EmergencyButtonActivity.class);
myIntent.putExtra("Fire",2);

//      Intent myIntent_second = new Intent(getApplicationContext(), EmergencyActivity.class);
myIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
```

```
getApplicationContext().startActivity(myIntent);
showNotification();
    }
  }

@Override
public void onDestroy() {

super.onDestroy();
Toast.makeText(getApplicationContext(), "Now the Service has been stopped!",
Toast.LENGTH_LONG).show();
  }
private void showNotification() {
final NotificationManagermgr = (NotificationManager) this
.getSystemService(Context.NOTIFICATION_SERVICE);
NotificationCompat.Builder note = new NotificationCompat.Builder(this);
note.setContentTitle("Device Accelerometer Notification");
note.setTicker("New Message Alert!");
note.setAutoCancel(true);
// to set default sound/light/vibrate or all
note.setDefaults(Notification.DEFAULT_ALL);
PendingIntent pi = PendingIntent.getActivity(this, 0, new Intent(this,
EmergencyButtonActivity.class), 0);
// set pending intent to notification builder
note.setContentIntent(pi);
mgr.notify(101, note.build());
  }
}
```