

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in

Information Technology

By

SELCIYA NAYAGAM.M
(Reg. No. 37120066)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI-60011.TAMILNADU.

JANUARY 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of SELCIYA NAYAGAM.M (Reg. No. 37120066) who underwent the professional training in "WOMEN SAFETY APP" under our supervision from July 2020 to September 2020.

Internal Guide

Dr.JABEZ.J,M.E., Ph.D.,

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held

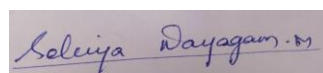
on _____

Internal Examiner

External Examiner

DECLARATION

I, SELCIYA NAYAGAM.M (Reg. No. 37120066) hereby declare that the Professional Training Report on “**WOMEN SAFETY APP**” done by me under the guidance of Dr. JABEZ.J , M.E., Ph.D., at Sathyabama Institute Of Science And Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of technology degree in Information Technology.

A rectangular box containing a handwritten signature in blue ink. The signature appears to read 'Selciya Nayagam.M'.

DATE:

SIGNATURE OF THE CANDIDATE

PLACE:

ACKNOWLEDGEMENT

The satisfaction and elation that accompany the successful completion of any task would be incomplete without the mention of the people who have made it possible. It is our great privilege to express our gratitude and respect to all those who have guided me during the course of my Professional Training.

First and foremost, we would express our sincere gratitude to our beloved Founder Chancellor **Col. Dr. JEPPIAAR, M.A., B.L., Ph.D.**, I extend my sincere thanks to our Chancellor **Dr. Mariazeena Johnson, B.E., M.B.A., M.Phil., Ph.D.**, and the President **Dr. Marie Johnson, B.E., M.B.A., M.Phil., Ph.D.**, and for providing me the necessary facilities for the completion of the professional training. I also acknowledge our Vice Chancellor **Dr. T.Sasipraba, M.E., Ph.D.**, and the Pro Vice Chancellor **Dr. E.LOGASHANMUGAM, M.E., Ph.D.**, for their constant support and endorsement.

I would like to express our gratitude to our Registrar **Dr. S. S. Rau, MBA., Ph.D.** and Controller of Examinations **Dr. Igni Sabasti Prabu M.E., Ph.D.**, for their valuable support offered to complete my professional training successfully.

I like to express our gratitude to **Dr.T.Sasikala, M.E., Ph.D.**, Dean of School of Computing and **R. Subhashini, M.E., Ph.D.**, Head of the Department of Information Technology, Sathyabama Institute Of Science and Technology for having been a constant source of support and encouragement for the completion of the professional training. I would also like to express our sincere thanks to our internal guide **Dr. JABEZ.J ,M.E., Ph.D.**, who guided me in the preparation of the report. Also, I thank the **Almighty** and my **Parents** for supporting me in the completion of the professional Training.

TRAINING CERTIFICATE



ABSTRACT

WOMEN SAFETY APP

Women's security is a critical issue in today's world and it's very much needed for every individual to be acting over such an issue. This document describes a GPS based "Women Security System" that provides the combination of GPS device as well as provide alerts and messages with an emergency button trigger whenever somebody is in trouble They might not have so much time, all that they have to do is generate a distress emergency signal by shaking up their phone. Our system provides a realizable, cost effective solution to problem detection. Nowadays due to recently happened cases such as rape by drivers or colleagues, burglary etc., women security, especially women security has become the foremost priority of the world. System uses the Global Positioning System (GPS) technology to find out the location of women. The information of women position provided by the device can be viewed on Google maps using Internet or specialized software. The companies are looking for-ward to the security problem and requires a system that will efficiently evaluate the problem of women security working in night shifts, traveling alone. We focus on the proposed model that can be used to deal with the problem of security issue of women using GPS based tracking system.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	vi
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION 1.1 Introduction	1
2	SYSTEM REQUIREMENTS 2.1 Software Requirements 2.2 Hardware Requirements	2
3	DESIGNS 3.1 System Design 3.2 Data Flow Diagram 3.3 Flow Charts	3
4	IMPLEMENTATION 4.1 Implementation Of Project 4.2 Post Implementation 4.3 Software Maintenance	8
5	RESULT 5.1 Output	10

6	CONCLUSION	12
	REFERENCE	13
	APPENDIX	14
	A. SOURCE CODE	
	B. SCREEN SHOTS	

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.2.1	Data Flow Level 0	4
3.2.2	Data Flow Level 1	4
3.3.1	Track me	5
3.3.2	Where are you	6
3.3.3	Distress signal	7
5.1.1	Layout of app	10
5.1.2	Contacts entered	10
5.1.3	Message received	11

LIST OF ABBREVIATION

IDLE	-	INTEGRATED DEVELOPMENT ENVIRONMENT
API	-	APPLICATION PROGRAMMING INTERFACE
NLP	-	NATURAL LANGUAGE PROCESSING

CHAPTER 1

INTRODUCTION

1.1 Introduction

Women are adept at mobilizing diverse groups for a common cause. They often work across ethnic, religious, political, and cultural divides to promote peace. We are all aware about importance of safety of women's but we must realize that they should be properly protected. Women's are not as physically strong as men, in an emergency situation a helping hand would be a relief for them. The best way to minimize your chances of becoming a victim of violent crime (robbery, sexual assault, rape, domestic violence) is to identify and call on resources to help you out of dangerous situations. Whether you're in immediate trouble or get separated from friends during a night out and don't know how to get home, having these apps on your phone can reduce your risk and bring assistance when you need it. Although several were originally developed for students to reduce the risk of sexual assault on campus, they are suitable for all women in the light of recent outrage in Delhi which shook the nation and woke us to the safety issues for our daughters, people are gearing up in different ways to fight back. A host of new apps have been developed to provide security systems to women on their phone.

1.2 Outline

Here we introduce an app which ensures the safety of women. This helps to identify and call on resources to help the one out of dangerous situations. These reduce risk and bring assistance when we need it and help us to identify the location of the one in danger. This app designed to provide security to women main purpose of this app to provide the awareness on the time of critical situation for women. Generally user can activate this service by adding the emergency contacts using the emergency contacts icon in the app. While in emergency the user would have to shake up his/her handset, after that a distress signal(SOS) will automatically got generated from the user end and send SMS to those contacts which has been saved at the time of registration. The SMS contains your message and your exact location.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS:

- Operating System - Android
- API Level - 14 or higher.
- Disk Usage – 20-50 Mb

2.2 HARDWARE REQUIREMENTS:

- Processor : Snapdragon, Dual Core.
- Memory Space : 50 Mb
- RAM : 512 MB.
- GPS enabled Android Phone

CHAPTER 3

DESIGNS

3.1 SYSTEM DESIGN

In System design the design functions and operations are described in detail, including screen layouts, business rules, process diagrams and other documentations. The output of this stage will describe the new system as a collection of modules or subsystems. The design stage take as its initial input the requirements identified in the approved requirements document.

The logical design of our system pertains to an abstract representation of data flows, inputs and outputs of the system. In the context of systems design, modeling can undertake the following forms, including:

- Data Flow Diagrams
- Flow Charts

The physical design relates to the actual input and output process of the system. This is laid down in terms of how data is input into our system, how it is verified/authenticated, how it is processed, and how it is displayed as output.

3.2 DATA FLOW DIAGRAM

Fig : 3.2.1 Level 0

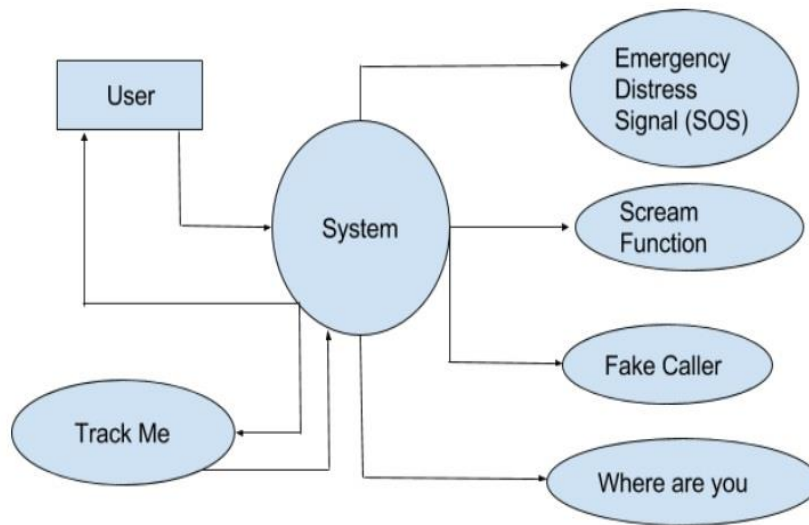
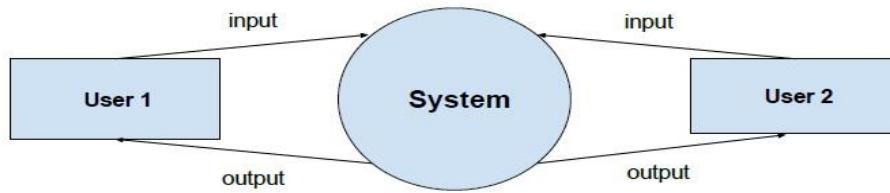


Fig : 3.2.2 Level 1

3.3 FLOW CHART

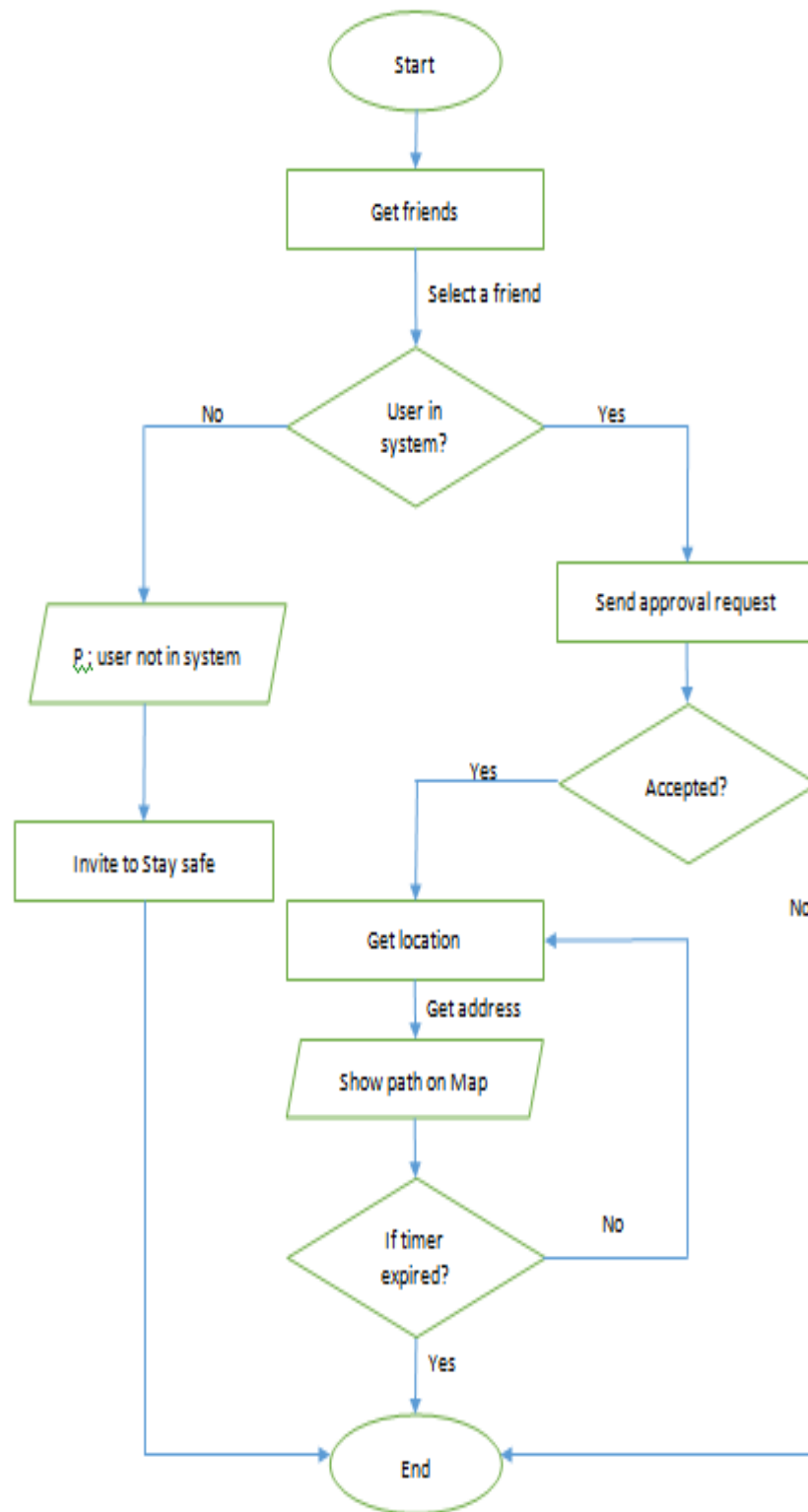


Fig : 3.3.1 Track me

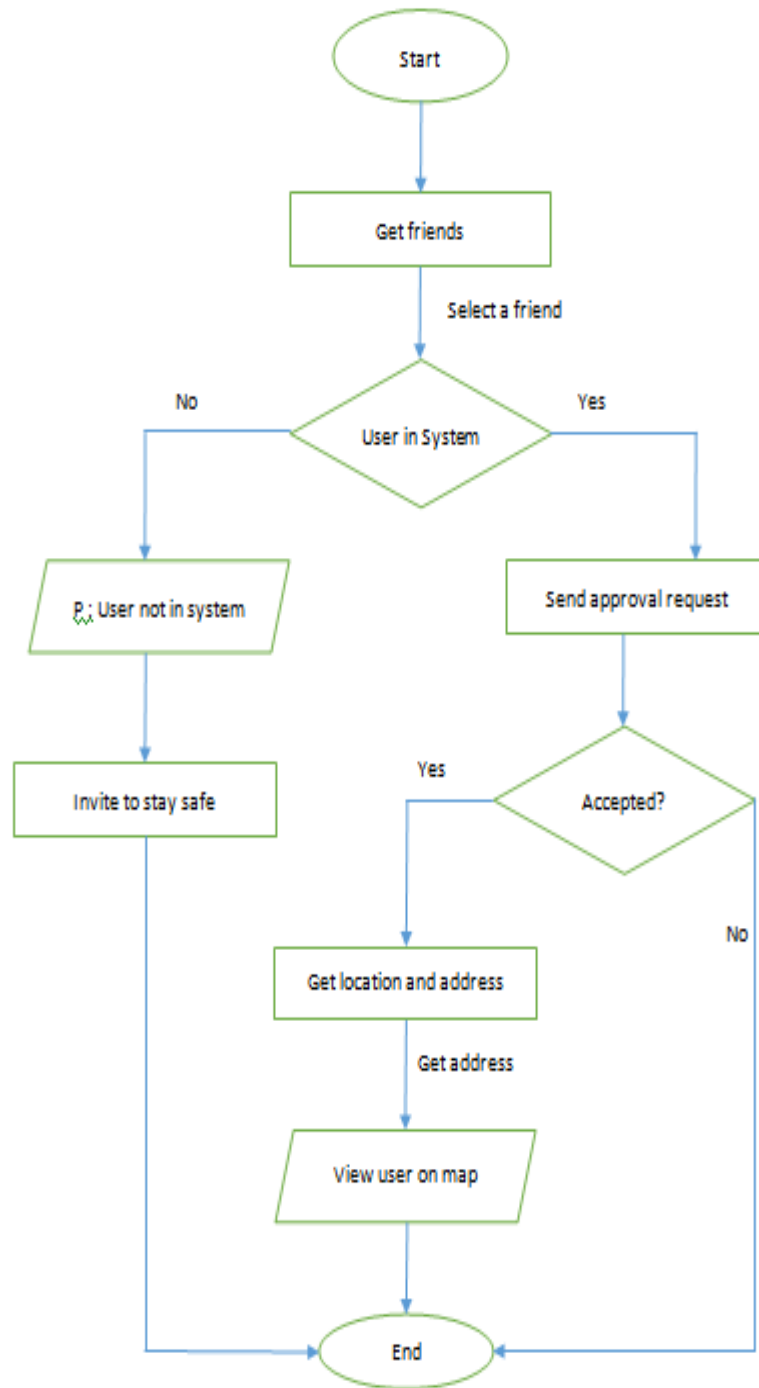
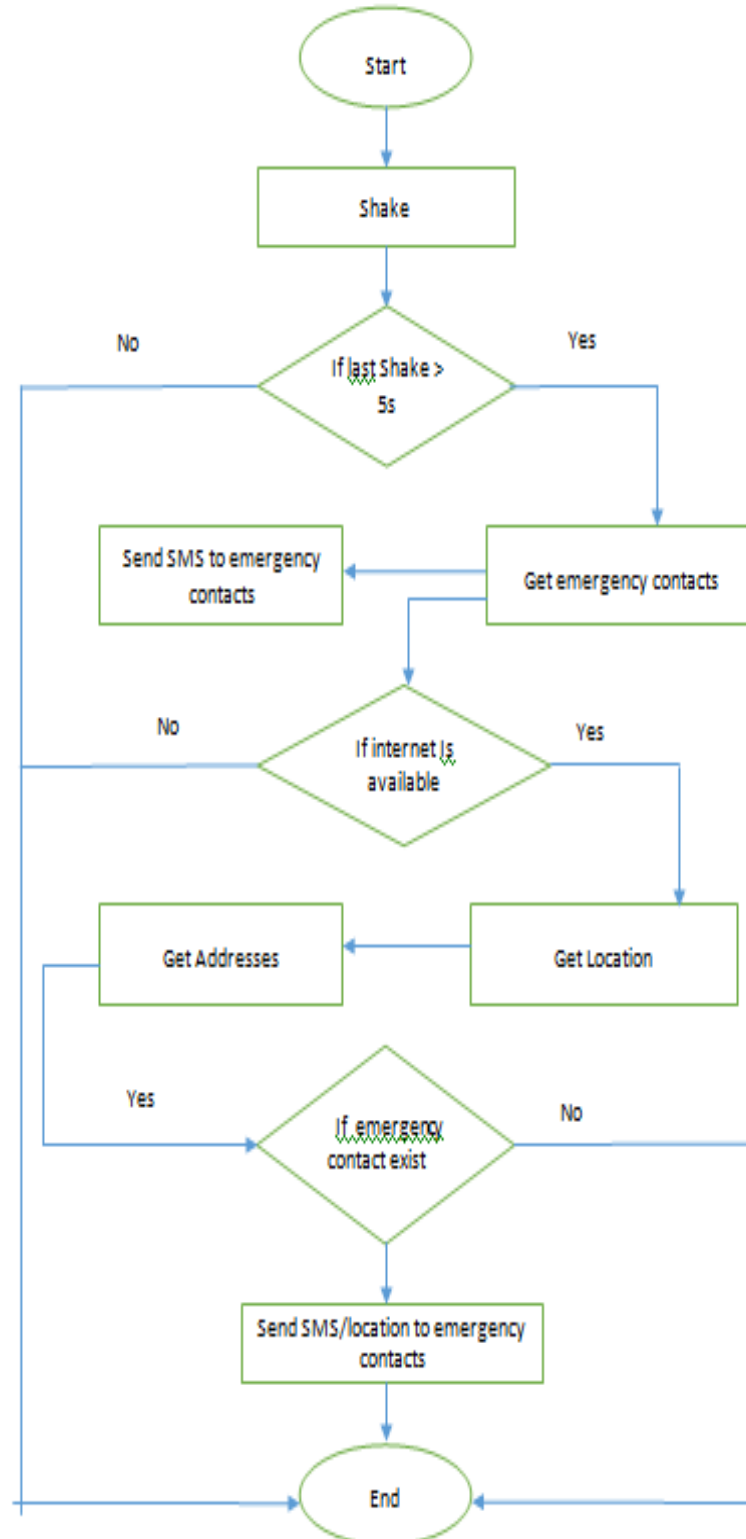


Fig : 3.3.2 Where are you



CHAPTER 4

IMPLEMENTATION

4.1 IMPLEMENTATION OF THE PROJECT

There are three types of implementations :-

- Implementation of android system to replace a manual system.
- Implementation of a new android system to replace an existing system.
- Implementation of a modified application to replace an existing one, using the same system.

4.2 POST IMPLEMENTATION

After the implementation, the beta version of the application would be available and ready to use for all the customers. During this phase, we try to identify any bugs that may have crept in to the system despite all the previous testing done. This is also called beta testing during the Post.

4.3 SOFTWARE MAINTENANCE

Maintenance is the enigma of system development. It holds the software industry captive, tying up programming resources. Analysts and programmers spend more time maintaining programs than they writing them. Maintenance is not considered a part of software development, its activity an extremely important in the life of software product.

- **Corrective Maintenance:** - After the Implementation, Correcting the residual errors if any. If such errors are discovered, the source of it should be detected and removed. This phenomenon falls under corrective maintenance.

- Perfective Maintenance: - Sometimes changes have to be done according to the user requirements. This type of changes to the software is called perfective maintenance.
- Adaptive Maintenance: - Software often must be upgraded and enhanced to include more features and provide more services. This also requires modification of software.

CHAPTER 5

RESULT

5.1 Output:

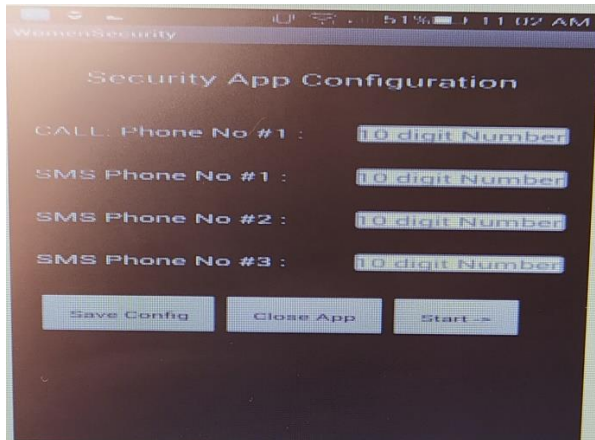


Fig : 5.1.1 Layout of app immediately after Installation.

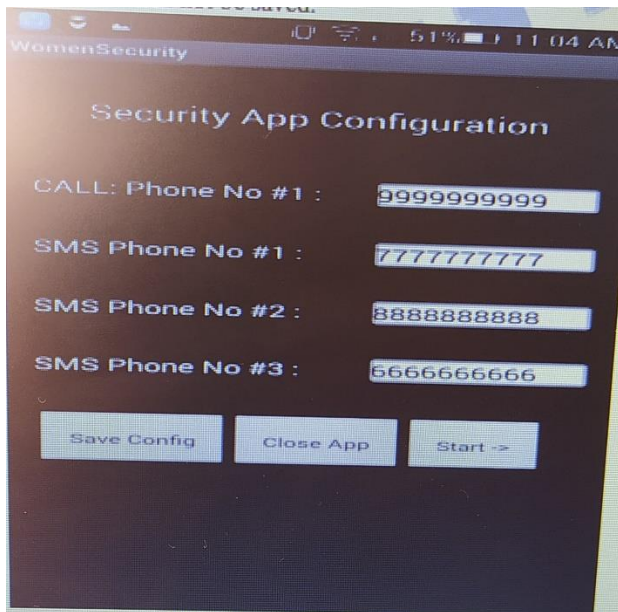


Fig : 5.1.2 Contacts entered in app.

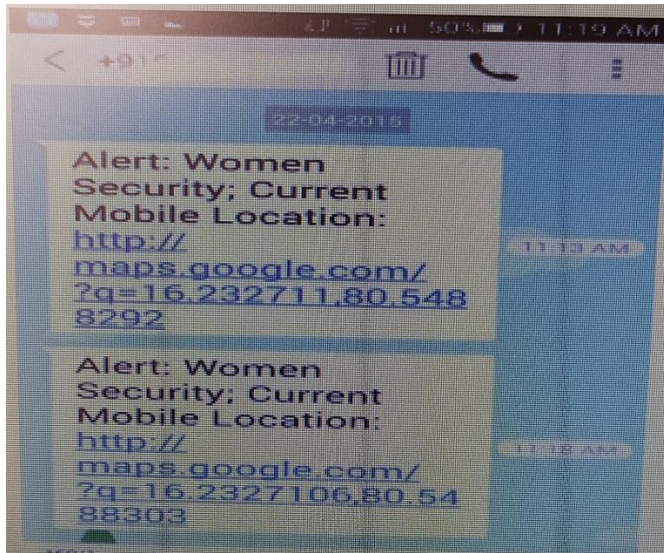


Fig : 5.1.3 Message received by one of the Contact.

CHAPTER 6

CONCLUSION

The testing results of the mentioned three sections are provided with snap or screen shots taken in various intervals of time from the root device and contact's device. Here, the root device means the device over which the rescue application is started; it means the user's device. The contact's device means the device to which the user's location information is sent continuously. For installing the application in the mobile phone, firstly in the settings, "allow non-market apps to be worked over the device" must be checked . depicts the settings of the device such that only by placing checkmark over the mentioned icon, the app will be installed on the device.

The Sauver app icon can be placed anywhere on the home screen of the smart phone so that we can immediately touch over the application when we are in danger. Once the application is installed on the mobile for the first time, depicts the list view of contact details and grid layout of buttons: save, close and start. The first contact number that has to be entered is for calling and the other three contact numbers that are entered will be used for sending messages from the root device.

REFERENCE

- [1] Doulamis, A.; Pelekis, N.; Theodoridis, Y., "EasyTracker: An Android Application for Capturing Mobility Behavior," 2012 16th Panhellenic Conference on Informatics (PCI), vol., no., pp.357,362, 5-7 Oct. 2012 .
- [2] Saranya, J.; Selvakumar, J., "Implementation of children tracking system on android mobile terminals," 2013 IEEE International Conference on Communications and Signal Processing (ICCSP), vol., no., pp.961,965, 3-5 April 2013.
- [3] Android Developers, Location APIs. URL: <http://developer.android.com/google/play-services/location.html>
- [4] "WOMEN'S SECURITY", Android App developed by AppSoftIndia, December 17, 2013. [https:// play .google. com/store /apps/ details? id= com. Zayanin fotech. security& hl=en](https://play.google.com/store/apps/details?id=com.Zayaninfotech.security&hl=en)
- [5] "BSAFE-PERSONAL SAFETY APP", Android app developed by Bipper. Inc., March 6, 2015 <http://getbsafe.com/>
- [6] "SAFETIPIN-COMPLETE SAFETY APP" , an Android app developed in January 21, 2015. <http://safetipin.com/>
- [7] "POLIE NEARBY", Android app developed by Big Systems in 2013. [https:// play. google. com/ store /apps /details? id=com. smoketech.](https://play.google.com/store/apps/details?id=com.smoketech)

APPENDIX

A.Source code

Accelerometer Listener

```
package com.selci.womensafetyapp;
```

```
public interface AccelerometerListener {
```

```
    public void onAccelerationChanged(float x, float y, float z);
```

```
    public void onShake(float force);
```

```
}
```

Accelerometer Manager

```
package com.selci.womensafetyapp;
```

```
import java.util.List;
```

```
import android.content.Context;
```

```
import android.hardware.Sensor;
```

```
import android.hardware.SensorEvent;
```

```
import android.hardware.SensorEventListener;
```

```
import android.hardware.SensorManager;
```

```
import android.widget.Toast;
```

```
public class AccelerometerManager {
```

```
    private static Context aContext=null;
```

```
    /** Accuracy configuration */
```

```

private static float threshold = 15.0f;
private static int interval = 200;

private static Sensor sensor;
private static SensorManager sensorManager;
// you could use an OrientationListener array instead
// if you plans to use more than one listener
private static AccelerometerListener listener;

/** indicates whether or not Accelerometer Sensor is supported */
private static Boolean supported;
/** indicates whether or not Accelerometer Sensor is running */
private static boolean running = false;

/**
 * Returns true if the manager is listening to orientation changes
 */
public static boolean isListening() {
    return running;
}

/**
 * Unregisters listeners
 */
public static void stopListening() {
    running = false;
    try {
        if (sensorManager != null && sensorEventListener != null) {
            sensorManager.unregisterListener(sensorEventListener);
        }
    } catch (Exception e) {}
}

```

```

}

/**
 * Returns true if at least one Accelerometer sensor is available
 */
public static boolean isSupported(Context context) {
    aContext = context;
    if (supported == null) {
        if (aContext != null) {

            sensorManager = (SensorManager) aContext.
                getSystemService(Context.SENSOR_SERVICE);

            // Get all sensors in device
            List<Sensor> sensors = sensorManager.getSensorList(
                Sensor.TYPE_ACCELEROMETER);

            supported = new Boolean(sensors.size() > 0);

        } else {
            supported = Boolean.FALSE;
        }
    }
    return supported;
}

/**
 * Configure the listener for shaking

```

```

* @param threshold
*      minimum acceleration variation for considering shaking
* @param interval
*      minimum interval between to shake events
*/
public static void configure(int threshold, int interval) {
    AccelerometerManager.threshold = threshold;
    AccelerometerManager.interval = interval;
}

/**
* Registers a listener and start listening
* @param accelerometerListener
*      callback for accelerometer events
*/
public static void startListening( AccelerometerListener accelerometerListener )
{

    sensorManager = (SensorManager) aContext.
        getSystemService(Context.SENSOR_SERVICE);

    // Take all sensors in device
    List<Sensor> sensors = sensorManager.getSensorList(
        Sensor.TYPE_ACCELEROMETER);

    if (sensors.size() > 0) {

        sensor = sensors.get(0);

        // Register Accelerometer Listener
        running = sensorManager.registerListener(

```

```

        sensorEventListener, sensor,
        SensorManager.SENSOR_DELAY_GAME);

    listener = accelerometerListener;
}

}

/**
 * Configures threshold and interval
 * And registers a listener and start listening
 * @param accelerometerListener
 *      callback for accelerometer events
 * @param threshold
 *      minimum acceleration variation for considering shaking
 * @param interval
 *      minimum interval between to shake events
 */
public static void startListening(
    AccelerometerListener accelerometerListener,
    int threshold, int interval) {
    configure(threshold, interval);
    startListening(accelerometerListener);
}

/**
 * The listener that listen to events from the accelerometer listener
 */
private static SensorEventListener sensorEventListener =
    new SensorEventListener() {

```

```

private long now = 0;
private long timeDiff = 0;
private long lastUpdate = 0;
private long lastShake = 0;

private float x = 0;
private float y = 0;
private float z = 0;
private float lastX = 0;
private float lastY = 0;
private float lastZ = 0;
private float force = 0;

public void onAccuracyChanged(Sensor sensor, int accuracy) {}

public void onSensorChanged(SensorEvent event) {
    // use the event timestamp as reference
    // so the manager precision won't depends
    // on the AccelerometerListener implementation
    // processing time
    now = event.timestamp;

    x = event.values[0];
    y = event.values[1];
    z = event.values[2];

    // if not interesting in shake events
    // just remove the whole if then else block
    if (lastUpdate == 0) {
        lastUpdate = now;
    }
}

```

```

lastShake = now;
lastX = x;
lastY = y;
lastZ = z;
Toast.makeText(aContext,"No Motion detected",
    Toast.LENGTH_SHORT).show();

} else {
    timeDiff = now - lastUpdate;

    if (timeDiff > 0) {

        /*force = Math.abs(x + y + z - lastX - lastY - lastZ)
           / timeDiff;*/
        force = Math.abs(x + y + z - lastX - lastY - lastZ);

        if (Float.compare(force, threshold) >0 ) {
            //Toast.makeText(Accelerometer.getContext(),
            //(now-lastShake)+" >= "+interval, 1000).show();

            if (now - lastShake >= interval) {

                // trigger shake event
                listener.onShake(force);
            }
            else
            {
                Toast.makeText(aContext,"No Motion detected.",
                    Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

        lastShake = now;
    }
    lastX = x;
    lastY = y;
    lastZ = z;
    lastUpdate = now;
}
else
{
    Toast.makeText(aContext,"No Motion detected",
Toast.LENGTH_SHORT).show();

}
}

// trigger change event
listener.onAccelerationChanged(x, y, z);
}

```

```
};
```

Instructions

```
package com.prabhu.womensafetyapp;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.app.Activity;
```

```
import android.content.Intent;
```

```
public class Instructions extends Activity {
```

```
    @Override
```

```

        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_instructions);
        }

        public void back(View v) {
            Intent i_back=new Intent(Instructions.this,MainActivity.class);
            startActivity(i_back);

        }
    }
}

```

Main Activity

```
package com.prabhu.womensafetyapp;
```

```

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```

        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
        }
    }

```

```

        public void register(View v) {
            Intent i_register=new Intent(MainActivity.this,Register.class);

```

```
        startActivity(i_register);

    }

    public void display_no(View v) {
        Intent i_view=new Intent(MainActivity.this,Display.class);
        startActivity(i_view);

    }

    public void instruct(View v) {
        Intent i_help=new Intent(MainActivity.this,Instructions.class);
        startActivity(i_help);
    }

    public void verify(View v) {
        Intent i_verify=new Intent(MainActivity.this,Verify.class);
        startActivity(i_verify);
    }
}
```

B.SCREESHOTS:

