

## Chapter 1

# INTRODUCTION

This project presents an alert system for PROBLEM detection using common commercially available electronic devices to both detect the PROBLEM and alert authorities. We use an Android based smart phone with an integrated tri-axial accelerometer.

Data from the accelerometer is evaluated with several threshold values and position data to determine a PROB. The threshold is adaptive based on user provided parameters such as: height, weight, and level of activity. The application adapts to unique movements that a phone experiences as opposed to similar systems which require users to mount accelerometers to their chest or trunk. If a PROB is suspected a notification is raised requiring the user's response. If the user does not respond, the system alerts pre-specified social contacts with an informational message via SMS. If a contact responds the system commits an audible notification, automatically connects, and enables the speakerphone. If a social contact confirms a PROB, an appropriate emergency service is alerted. Our system provides a realizable, cost effective solution to PROB detection using a simple graphical interface while not overwhelming the user with uncomfortable sensors.

I PROB is a very powerful software especially developed for the safety of girls, when ever somebody is in trouble they don't have to sit and find contacts or find ways to send SMS, or message the near ones. They might not have so much time, all that they have to do is shake the smart phone above the threshold value, vigorously ., immediately an message alert is sent to the persons mom, dad and whoever they wish to, if their guardians also have a Smartphone. Even though if it is in silent mode. When a message called ALERT is received it automatically changes its profile to general, and gives a voice notification YOUR SON / DAUGHTER IS IN TROUBLE PLEASE HELP.... PLEASE HELP.... PLEASE HELP.... REPEATEDLY AS A RING TONE until they listen and stop it.

If they want to find where their ward is all that they have to send is LOC as an SMS to their smart phone, it will respond with the current location of their ward.

If parents want to track their ward, they have to send TRACK message as a SMS to their ward's Smartphone, it will respond with the location every 5 minute once, which is stored and gets connected with the GOOGLE MAPS using GPRS and plot the ROUTE in live.

## 1.1 Definitions

- **The Android Open Source Project (AOSP):-** The Android Open Source Project (AOSP) is an initiative created to guide development of the Android mobile platform. The Android platform consists of the operating system (OS), middleware and integral mobile applications
- **Integrated development environment (IDE):-** An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.
- **Linux Phone Standards Forum (LiPS) or the Open Mobile Alliance(OMA):-** Android runs on an open source Linux kernel
- **Eclipse Software Development Kit(SDK) :-** SDK is a set of tools that can be used to develop software applications targeting a specific platform, it includes the Eclipse Java Development Tools (JDT)
- **Dalvik virtual machine (DVM):-** Dalvik is a part of the software stack that makes up the Android platform.
- **Global Position System (GPS):-** The Global Positioning System (GPS) is a satellite-based navigation system made up of at least 24 satellites.it Determines the current position of any object.

## 1.2 Project Outline

The report is arranged in the following way

### **Chapter 1:** Introduction

A brief introduction of the project. Project related definitions i.e java and its features, android and its features and apache tomcat and its features.

### **Chapter 2:** Survey of Literature

This chapter provides a review of the literature which includes the system study, proposed work, problem statement, existing system, proposed system and scope of the project.

### **Chapter 3: System Requirement Specification**

This chapter provides system requirement specifications such as functional and non functional requirements, hardware and software requirement.

### **Chapter 4: System Design**

This chapter consists of the system design which includes design overview, system architecture, dataflow diagrams, use-case diagrams, class diagrams, sequence diagrams, activity diagrams and modules of the project.

### **Chapter 5: Implementation**

This chapter includes the implementation of the project. It includes the steps for implementing, issues faced during implementation and also the algorithms used in the project.

### **Chapter 6: Testing**

This chapter gives the outline of all testing methods that are carried out in this project, it gives a brief description of the test environment. The testing process includes unit testing, integration testing, functional and system testing.

### **Chapter 7: Results**

This chapter has the snapshots of the project that shows the working process of the project.

### **Chapter 8: Conclusion**

This chapter includes the conclusion of the project.

## Chapter 2

# REVIEW OF LITERATURE

## 2.1 System Study

### 2.1.1 Java

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems Java platform [2][3]. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture[1][4]. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere"[2][3]. Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995.. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java, GNU Classpath and Dalvik [1][4].

Sun Microsystems released the first public implementation as Java 1.0 in 1995. It promised "Write Once, Run Anywhere" [1][4], providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular.

## **Java Features**

### **A) Platform Independent**

The concept of Write-once-run-anywhere (known as the Platform independent) is one of the important key feature of java language that makes java as the most powerful language. Not even a single language is idle to this feature but java is closer to this feature.

### **B) Simple**

There are various features that make the java as a simple language. Programs are easy to write and debug because java does not use the pointers explicitly. It is much harder to write the java programs that can crash the system but we cannot say about the other programming languages.

### **C) Object-Oriented**

To be an Object Oriented language, any language must follow at least the four characteristics.

- **Inheritance:** It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding the additional features as needed.
- **Encapsulation:** It is the mechanism of combining the information and providing the abstraction.
- **Polymorphism:** As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.
- **Dynamic binding:** Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

As the languages like Objective C, C++ fulfils the above four characteristics yet they are not fully object oriented languages because they are structured as well as object oriented languages. But Java is fully Object Oriented language as object is at the outer most level of data structure. No stand-alone methods, constants and variables are there in java.

#### **D) Robust**

Java has the strong memory allocation and automatic garbage collection mechanism. It provides the powerful exception handling and type checking mechanism as compare to other programming languages. Compiler checks the program whether there any error and interpreter checks any run time error and makes the system secure from crash. All of the above features make the java language robust.

#### **E) Distributed**

The widely used protocols like HTTP and FTP are developed in java. Internet programmers can call functions on these protocols and can get access the files from any remote machine on the internet rather than writing codes on their local system.

#### **F) Portable**

The feature Write-once-run-anywhere makes the java language portable provided that the system must have interpreter for the JVM. Java also has the standard data size irrespective of operating system or the processor. These features make the java as a portable language.

#### **G) Dynamic**

While executing the java program the user can get the required files dynamically from a local drive or from a computer thousands of miles away from the user just by connecting with the Internet.

#### **H) Secure**

Java does not use memory pointers explicitly. All the programs in java are run under an area known as the sand box. Security manager determines the accessibility options of a class like reading and writing a file to the local disk.

#### **I) Performance**

Java uses native code usage, and lightweight process called threads. In the beginning interpretation of byte code resulted the performance slow but the advance version of JVM uses the adaptive and just in time compilation technique that improves the performance.

## **J) Multithreaded**

As we all know several features of Java like Secure, Robust, Portable, dynamic etc; you will be more delighted to know another feature of Java which is Multithreaded. Java is also a multithreaded programming language. Multithreading means a single program having different threads executing independently at the same time. Multiple threads execute instructions according to the program code in a process or a program. Multithreading works the similar way as multiple processes run on one computer.

Multithreading programming is a very interesting concept in Java. In multithreaded programs not even a single thread disturbs the execution of other thread. Threads are obtained from the pool of available ready to run threads and they run on the system CPUs. This is how Multithreading works in Java which you will soon come to know in details in later chapters.

## **K) Interpreted**

We all know that Java is an interpreted language as well. With an interpreted language such as Java, programs run directly from the source code. The interpreter program reads the source code and translates it on the fly into computations. Thus, Java as an interpreted language depends on an interpreter program. The source code to be written and distributed is platform independent [1][4].

## 2.1.2 Apache Tomcat

Apache Tomcat (or Jakarta Tomcat or simply Tomcat)[2][4] is an open source servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run.

Tomcat should not be confused with the Apache web server, which is a C implementation of an HTTP web server; these two web servers [2][3] are not bundled together. Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files.

### Features

- Implements the Servlet 2.4 and JSP 2.0 specifications.
- Reduced garbage collection, improved performance and scalability.
- Native Windows and UNIX wrappers for platform integration.
- Faster JSP parsing.

## 2.1.3 Android

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Android [1] is a software platform and operating system for mobile devices based on the Linux operating system [2][3] and developed by Google and the Open Handset Alliance. It allows developers to write managed code in a Java-like language that utilizes Google-developed Java libraries, but does not support programs developed in native code.

The unveiling of the Android platform on 5 November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 34 hardware, software and telecom companies devoted to advancing open standards for mobile devices. When released in 2008, most of the Android platform will be made available under the Apache free-software and open-source license.



## **Features**

- **Application Framework**

It is used to write applications for Android. Unlike other embedded mobile environments, Android[1] applications are all equal, for instance, an applications which come with the phone are no different than those that any developer writes. The framework is supported by numerous open source libraries such as openssl, SQLite and libc[1][2].

- **Dalvik Virtual Machine**

It is extremely low-memory based virtual machine, which was designed especially for Android to run on embedded systems and work well in low power situations. It is also tuned to the CPU attributes. The Dalvik VM creates a special file format (.DEX) that is created through build time post processing. Conversion between Java classes and .DEX format is done by included “dx” tool.

- **Integrated Browser**

Google made a right choice on choosing WebKit as open source web browser. They added a two pass layout and frame flattening. Two pass layout loads a page without waiting for blocking elements, such as external CSS or external JavaScript and after a while renders again with all resources downloaded to the device. Frame flattening converts founded frames into single one and loads into the browser. These features increase speed and usability browsing the internet via mobile phone.

- **Optimized Graphics**

As Android has 2D graphics library and 3D graphics based on OpenGL ES 1.0, possibly we will see great applications like Google Earth and spectacular games like Second Life, which come on Linux version. At this moment, the shooting legendary 3D game Doom was presented using Android on the mobile phone.

- **SQLite**

Extremely small (~500kb) relational database management system, which is integrated in Android. It is based on function calls and single file, where all definitions, tables and data are stored. This simple design is more than suitable for a platform such as Android.

- **Handset Layouts**

The platform is adaptable to both larger, VGA, 2D graphics library, 3D graphics library based on OpenGL ES 1.0 specifications, traditional smart phone layouts. An underlying 2D graphics engine is also included. Surface Manager manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications

- **Data Storage**

SQLite is used for structured data storage. SQLite is a powerful and lightweight relational database engine available to all applications.

- **Connectivity**

Android supports a wide variety of connectivity technologies including GSM, CDMA, Bluetooth, EDGE, EVDO, 3G and Wi-Fi.

- **Messaging**

SMS, MMS, and XMPP are available forms of messaging including threaded text messaging.

- **Web Browser**

The web browser available in Android is based on the open-source WebKit application framework. It includes LibWebCore which is a modern web browser engine which powers both the Android browser and an embeddable web view.

- **Java Virtual Machine**

Software written in Java can be compiled into Dalvik byte codes and executed in the Dalvik virtual machine, which is a specialized VM implementation designed for mobile device use, although not technically a standard Java Virtual Machine.

- **Media Support**

Android will support advanced audio/video/still media formats such as MPEG-4, H.264, MP3, and AAC, AMR, JPEG, PNG, GIF.

- **Additional Hardware Support**

Android is fully capable of utilizing video/still cameras, touch screens, GPS, compasses, accelerometers, and accelerated 3D graphics.

- **Development Environment**

Includes a device emulator, tools for debugging, memory and performance profiling, a plug-in for the Eclipse IDE. There are a number of hardware dependent features, for instance, a huge media and connections support, GPS, improved support for Camera and simply GSM telephony. A great work was done for the developers to start work with Android using device emulator, tools for debugging and plug-in for Eclipse IDE.

## 2.2 Proposed Work

### 2.2.1 Problem Statement

In existing systems, initial action has to be triggered by victim which is often not possible in such situations. So, the problem is to build a system that works autonomously or by easy actions.

### 2.2.2 Existing System

- **SHE (Society Harnessing Equipment):** It is a clothing fixed with an electronic device. This clothing has an electric circuit that can generate 3800kV which can help the sufferer escape. In case of multiple attacks it can send around 80 electric shocks.
- **ILA security:** The co-founders of this system, have designed three personal alarms that can shock and disorient potential attackers and hence protect the victim from hazardous situations.
- **AESHS (Advanced Electronics System for Human Safety)** It is a device that helps track the location of the victim when attacked using GPS facility.
- **VithU app:** This is an emergency app initiated by a popular Indian crime television series “Gumrah” aired on Channel [V]. When the power button of the Smartphone is pressed twice consecutively, it begins sending alert messages with a link of the location of the user every two minutes to the contacts.
- **Smart Belt:** This system is designed with a portable device which resembles a normal belt. It consists of Arduino Board, screaming alarm and pressure sensors. When the threshold of the pressure sensor crosses, the device will be activated automatically. The screaming alarm unit will be activated and send sirens asking help.

The main drawback of these applications and services is that the initial action has to be triggered by the victim which often in situation like these doesn't happen. So the emphasis is to build a solution that works autonomously in situations encountered.

### 2.2.3 Proposed System

- Approach is to use the number one fastest growing device which billions of people already own, a programmable cellular phone.
- Exploits a greater range of communication capabilities and integrated hardware and software features.
- Touch screen response and voice recognition, common to smart phones, provide a reliable interface for the user. By using interfaces that are similar to applications the user frequently uses, the rare interaction with the Iprob detection software should be familiar.

## 2.3 Scope of the project

- Our approach is to use the number one fastest growing device which billions of people already own, a programmable cellular phone
- It also exploits a greater range of communication capabilities and integrated hardware and software features.
- Touch screen response and voice recognition, common to smart phones, provide a reliable interface for the user. By using interfaces that are similar to applications the user frequently uses, the rare interaction with the Iprob detection software should be familiar.

When a problem is detected, we first communicate with the user. If the user does not respond, we attempt to contact members in his/her social network. If either fail or the social contact confirms a problem, the system alerts an emergency service (police).

- This also has an additional feature i.e. a circuit to prevent chain snatching, if a thief tries to pull or cut a chain, he may get a shock for breaking the chain and alert messages will be sent to the related people.

## **Chapter 3**

# **SYSTEM REQUIREMENTS SPECIFICATION**

## **1. Functional Requirements**

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the general approach taken by the organization when writing requirements. The functional system requirements describe the system function in detail, its inputs and outputs, exceptions and so on.

Functional requirements are as follows:

- To provide security to women who is the victim.
- This project helps the guardian to know the condition of the victim.
- It will help to reduce the harassment and rape cases.

## **2. Non-Functional Requirements**

Non functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as capabilities of I/O devices and the data representations used in system interfaces.

The non functional requirements are as follows:

- It sends information to people in near vicinity requesting public attention via wearable device.
- Helps to track the co-ordinates of the victim using GPS.
- When the alert message is sent to guardian, it turns the phone from silent to general with a voice message alerting them.
- It is feasible, reliable and of less cost.

### **3. Hardware Requirements**

- Processor: ARM or QUALCOMM Processor (32 bit).
- Ram: 128MB or More.
- Hard Disk: Minimum 200MB.
- Android Mobile Phone.

### **4. Software Requirements**

- Android SDK 1.5 or More.
- Eclipse IDE.
- Programming language JAVA and XML.
- Operating System Android (Linux Kernel).

## Chapter 4

# SYSTEM DESIGN

### 4.1 Design Overview

System design is the process of defining the architecture, components, modules, interface and data for a system to satisfy specified requirements. System design is an application of system's theory to product development. There is some overlap with the disciplines of the system's analysis, system's architecture and system's engineering. If the broader topic of product development "Blends the perspective of marketing, design and manufacturing into a single approach to product development", then design is the act of taking the marketing information and creating the design of the product to be manufactured. System's design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

### 4.2 System Architecture

The diagram shows the major components and technologies used in the project. Each section is described in more detail below.

#### 1. Hardware layer

This layer depicts all the major hardware components used in the project.

- Microcontroller
- GSM
- Buzzer
- Vibrator

#### 2. Application layer

The application layer consists of the programming languages used in the hardware and the applications.

#### 3. Android layer

In this layer, we are mainly explaining about the operating system Android and its features.

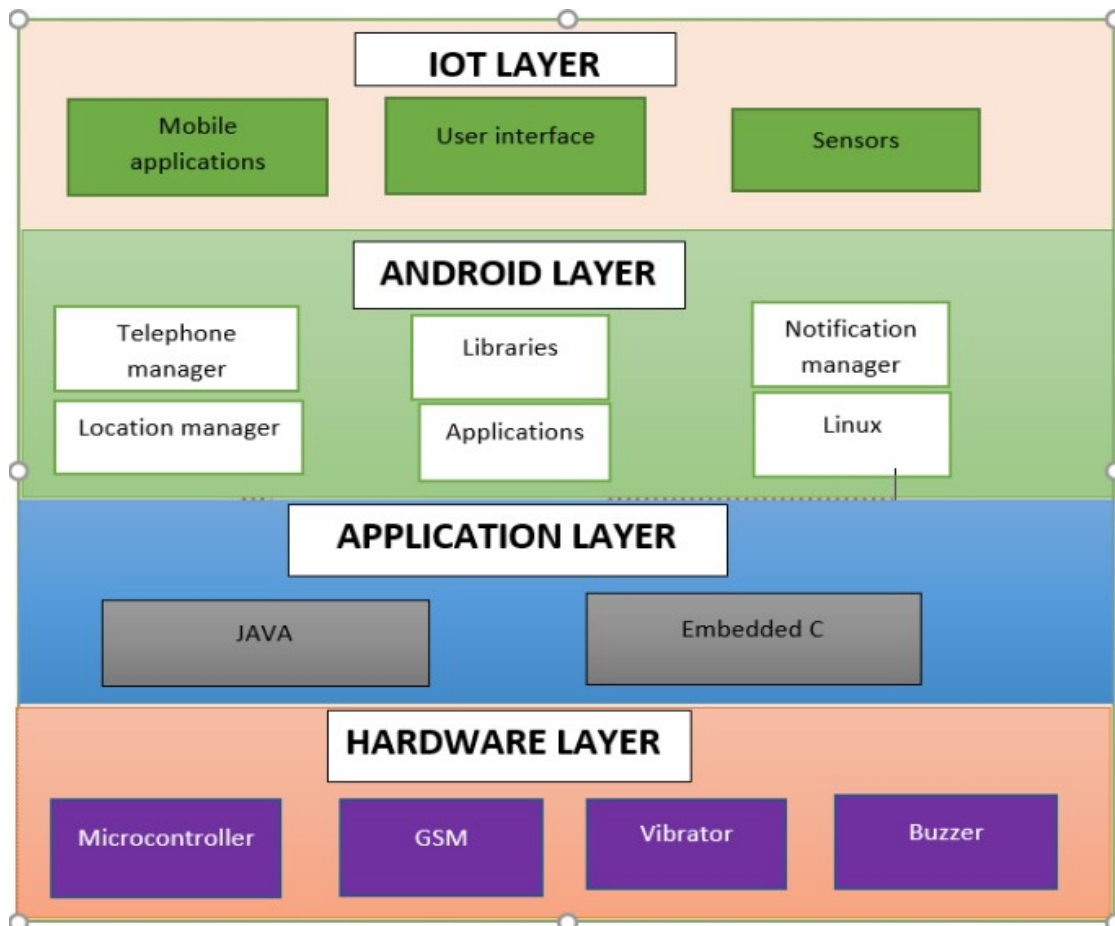


There is telephone manager, notification manager, location manager and libraries which mainly come under application framework written in java language. It is a toolkit that all applications use, ones which come with mobile device like contacts or SMS box.

#### 4. IOT layer

The IOT layer provides a network where the software embedded devices will be enabled to connect and exchange data with other smart devices. IOT allows objects to be sensed or controlled remotely across existing network infrastructure. Some of the general and key characteristics identified in IOT are as follows:

- a. Intelligence
- b. Connectivity
- c. Dynamic Nature
- d. Security



**Figure 4.1 System Architecture**

### 4.3 Data Flow Diagrams

A data flow diagram (DFD) is a graphical representation of the "flow" of data through in information system, modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design)

#### 4.3.1 Data Flow Diagram - Level 0

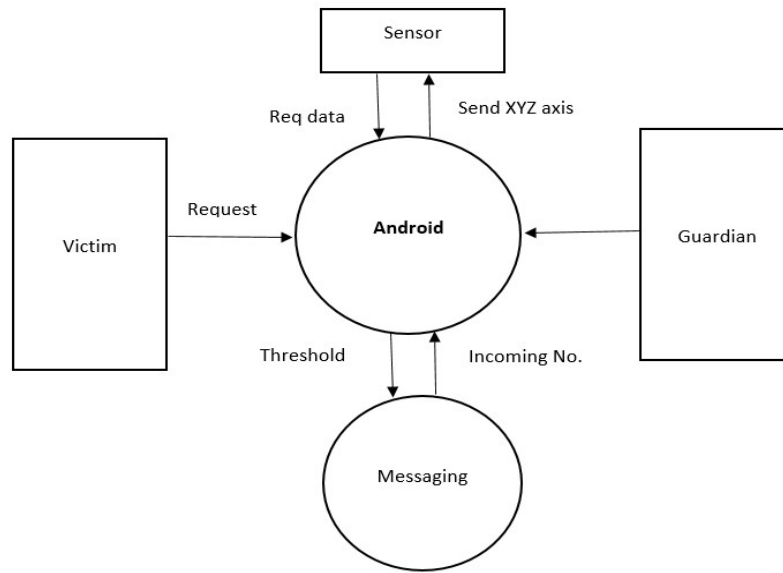


Figure 4.2 DFD-Level 0

### 4.3.2 Data Flow Diagram - Level 1

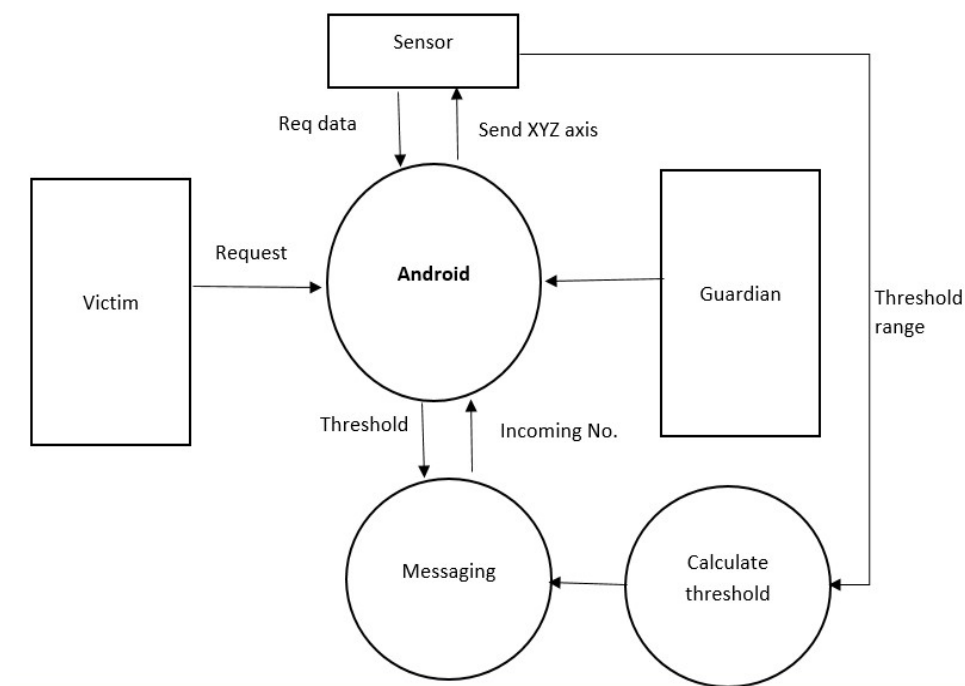


Figure 4.3 DFD-Level 1

### 4.3.3 Data Flow Diagram - Level 2

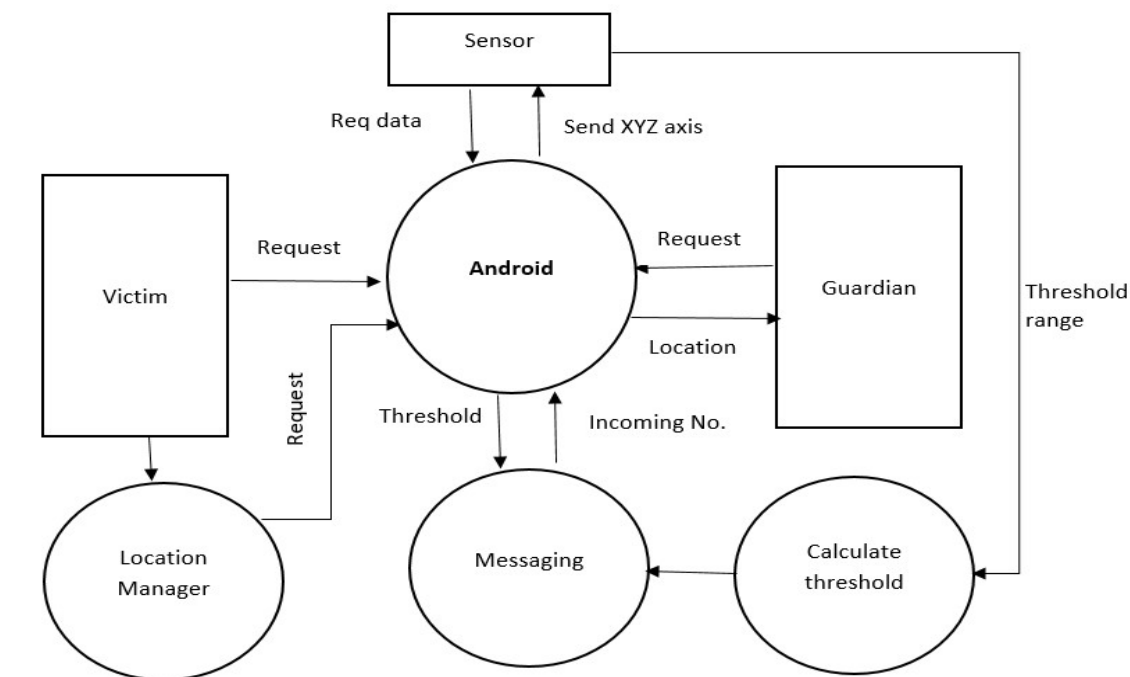


Figure 4.4 DFD-Level 2

### 4.3.4 Data Flow Diagram - Level 3

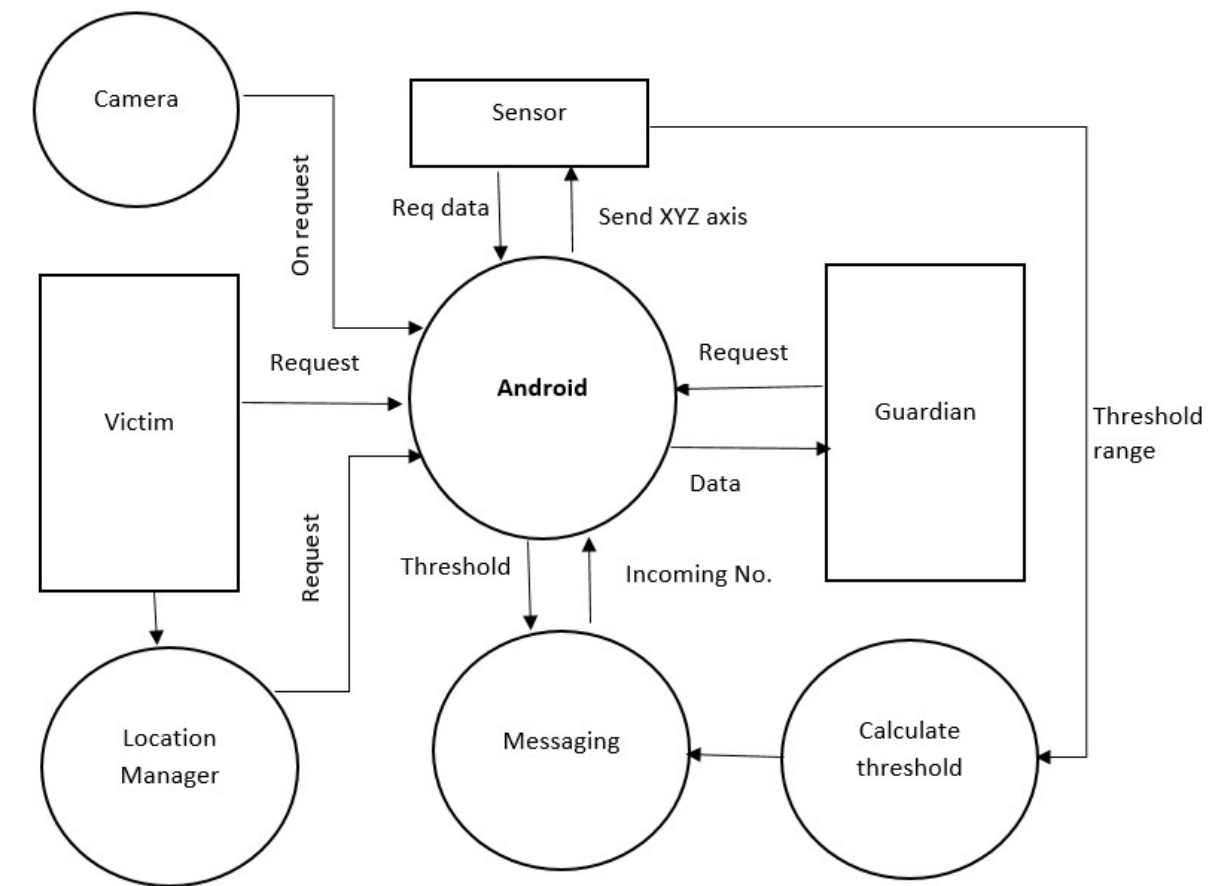


Figure 4.5 DFD-Level 3

## 4.4 Use Case Diagrams

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. It has been said that "Use case diagrams are the blueprints for your system".

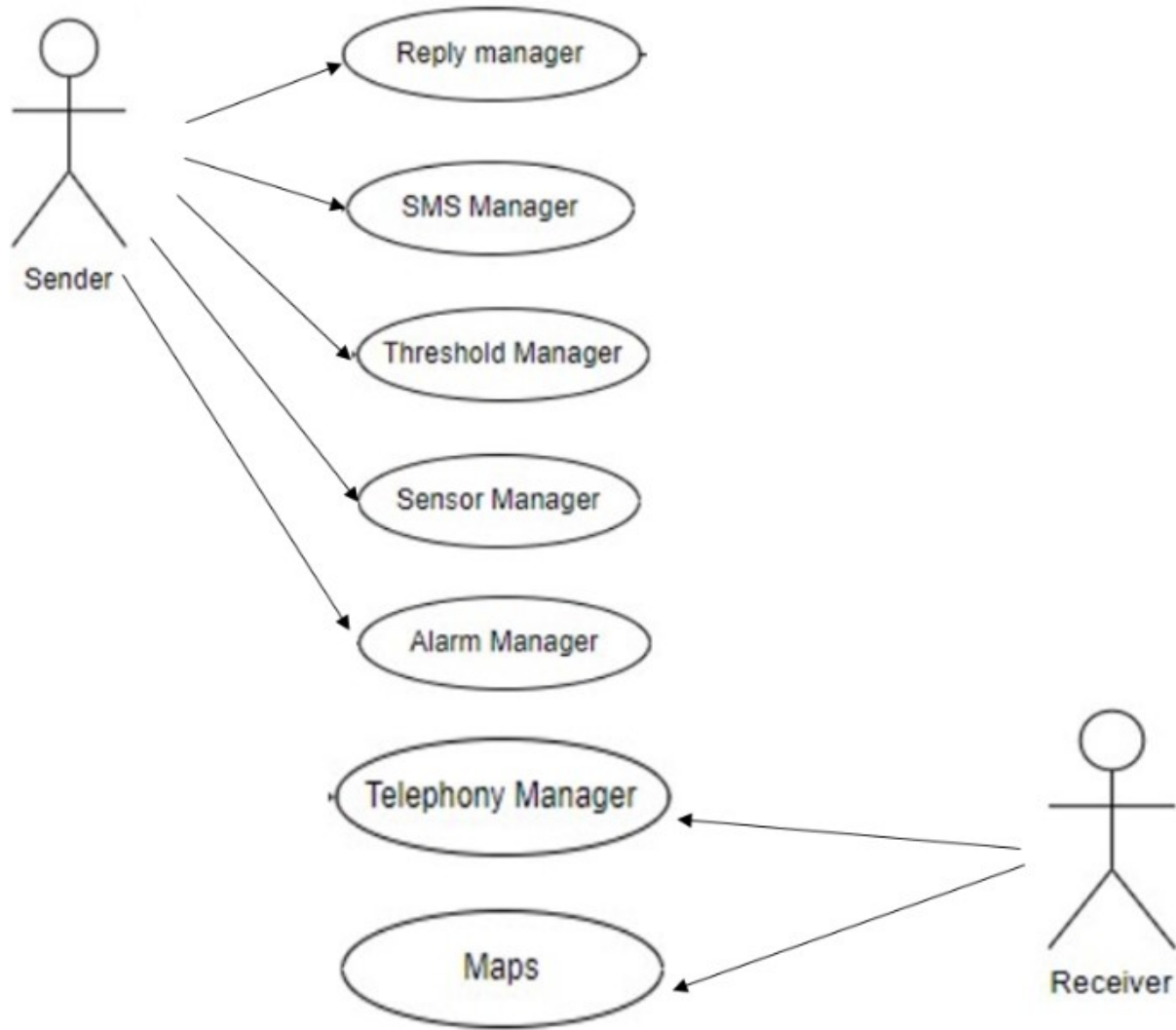


Figure 4.6 Use Case diagram

## 4.5 Class Diagrams

Class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

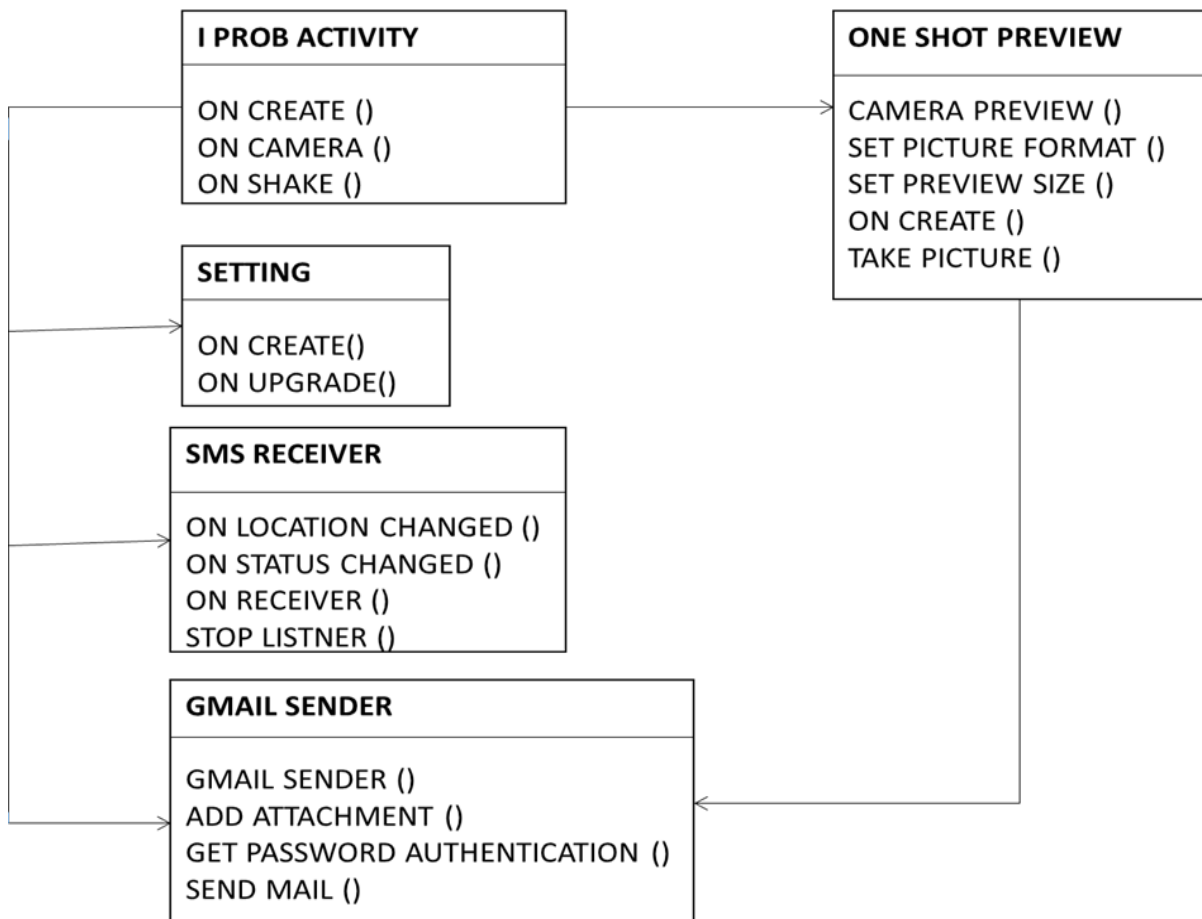


Figure 4.7 Class diagram for Camera

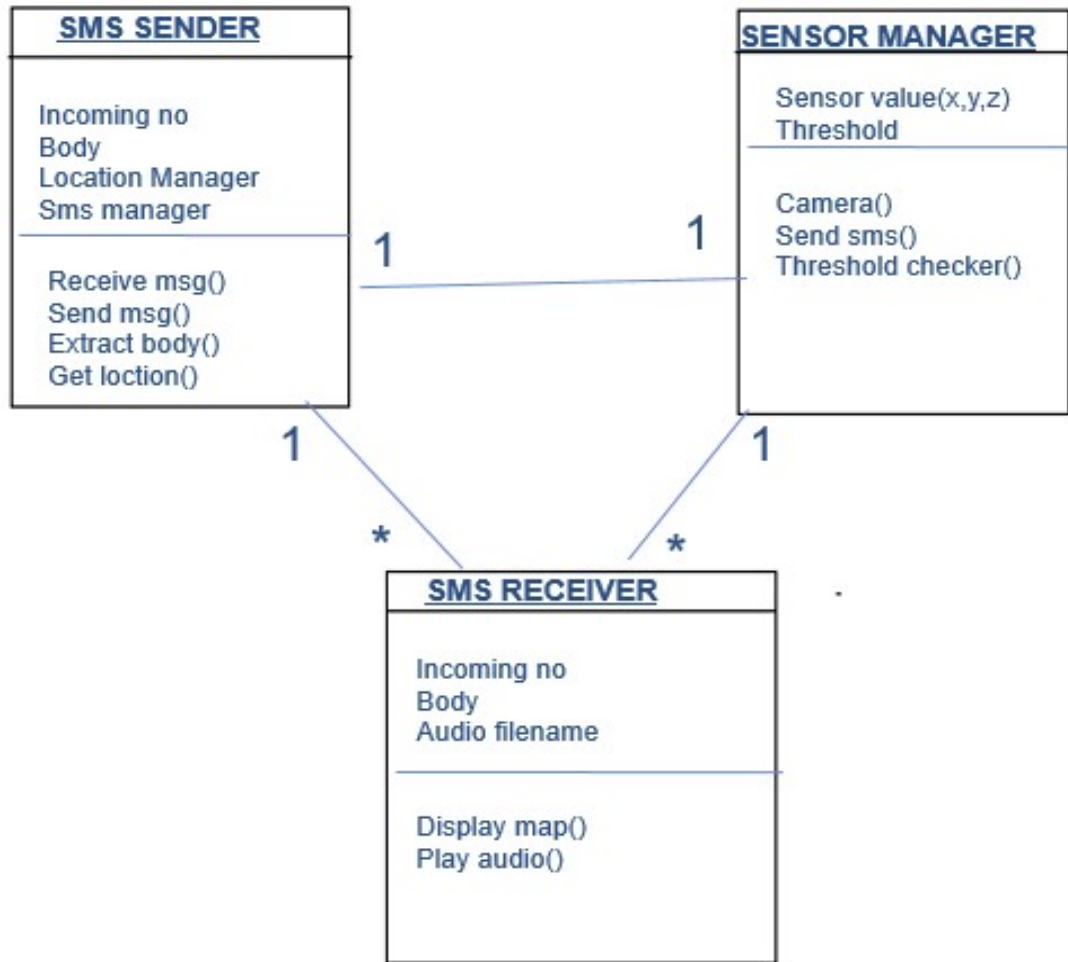


Figure 4.8 Class diagram for Message

## 4.6 Sequence Diagrams

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

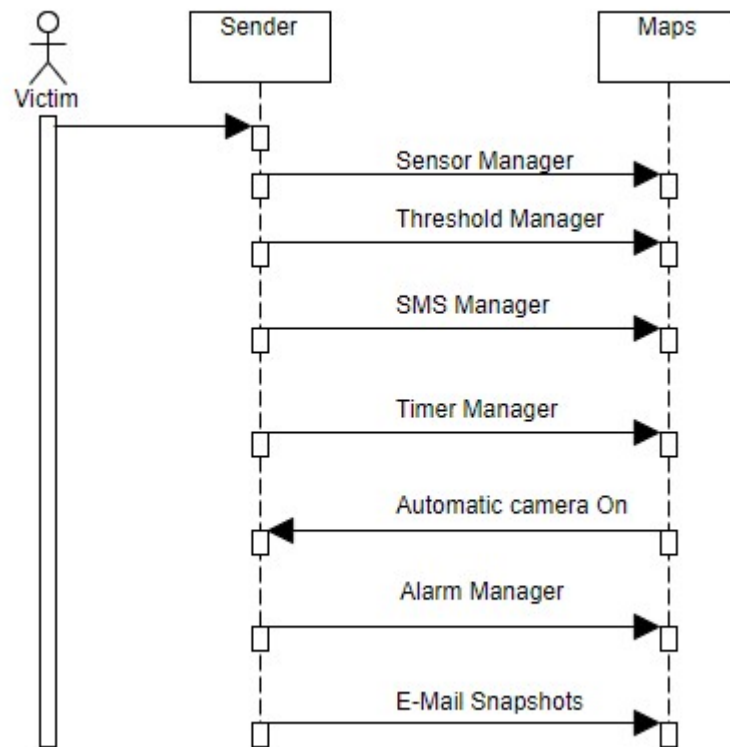


Figure 4.9 Sequence diagram for Sender



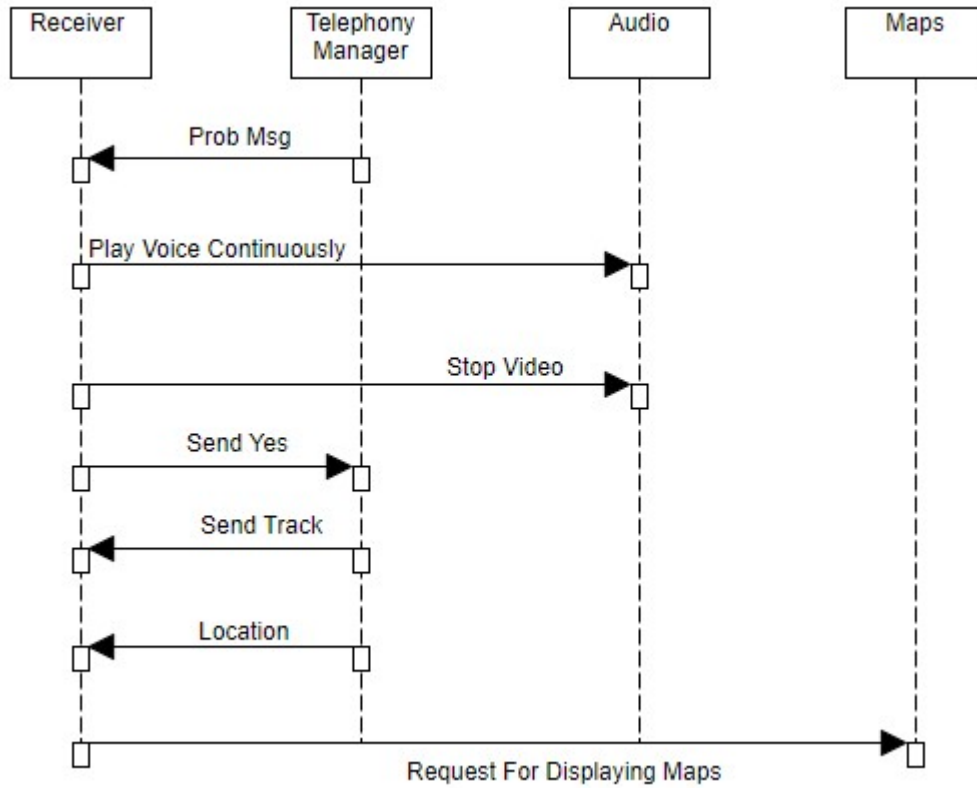


Figure 4.10 Sequence diagram for Receiver

## 4.8 Activity Diagrams

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

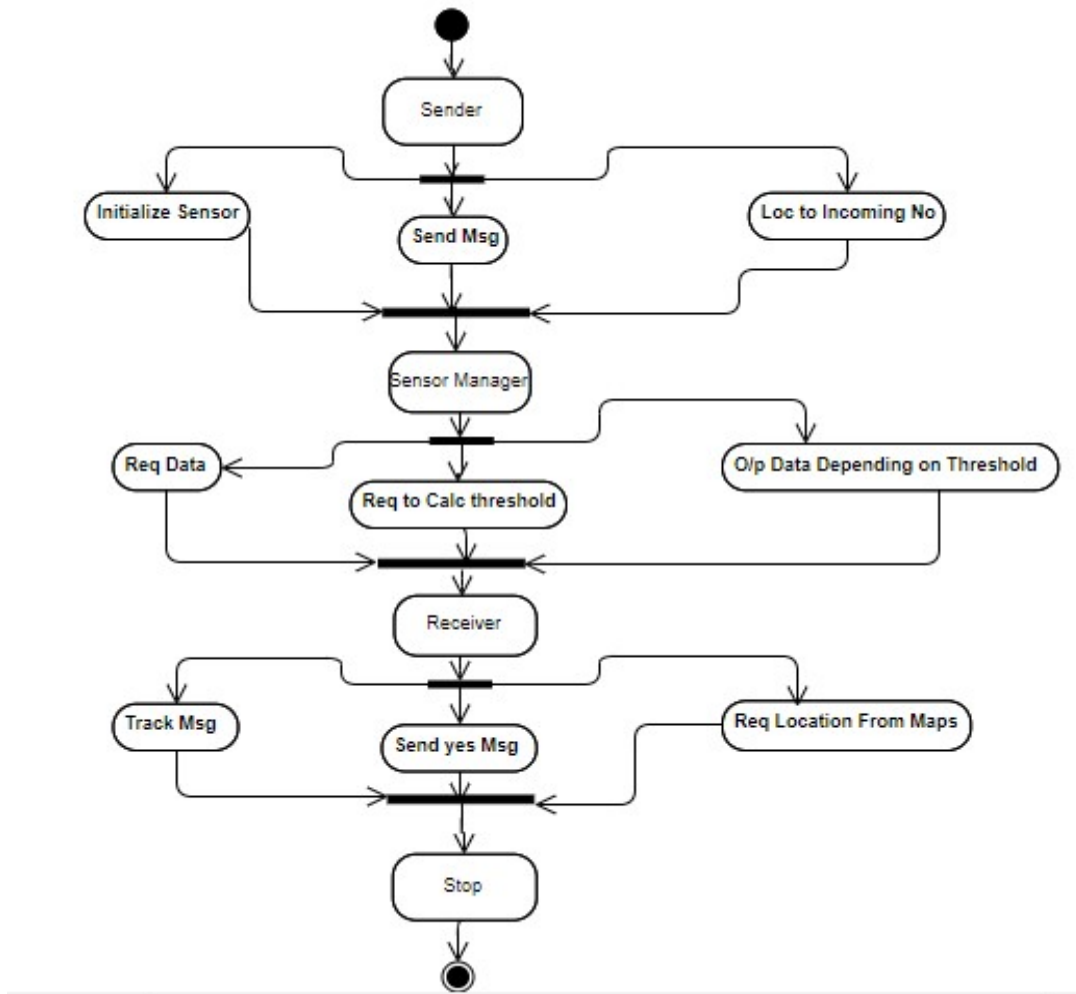


Figure 4.11 Activity diagram

## 4.9 Modules of the Project

- Sensor Manager Module
- Threshold Manager Module
- Contact Manager Module
- SMS Manager Module

### 4.9.1 Module 1

**Module Name:-** Sensor Manager Module

**Functionality:-** A sensor is a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument

**Input:-**Movement of the device

**Output:-**After sensing the acceleration and gravity induced forces the application will get started based on predefined specifications.

**Algorithm:-** Sensor Value Reader

### 4.9.2 Module 2

**Module Name:-** Threshold Manager Module

**Functionality:-** A threshold is a test of some variable against some value, with a report when the threshold value is exceeded. So it scales to the largest networks, with little traffic overhead

**Input:-**Values given by Accelerometer in terms of meter per second square.

**Output:-** - If the threshold value exceeds the vibration will be detected or else no

**Algorithm:-**Iprob detection or Shake algorithm

### 4.9.3 Module 3

**Module Name:-** Contact and SMS Manager Module

**Functionality:-** This module explains how to add contact to the social contact list. The user is given the option of adding more than one social contact to the list. SMS/MMS Manager contains a powerful rule editor which can be used to automate message processing.

**Input:-** Output from threshold manager module

**Output:-** When the fall is detected the message is sent to all the contacts one by one and React to events like message being received or connection going down.

**Algorithm:-**Message Alert algorithm

### 4.9.4 Module 4

**Module Name:-** GPS Manager Module

**Functionality:-** Tracks the location of the victim when requested by guardians

**Input:-** accelerometer values more than threshold values

**Output:** React to events like message being received or connection going down. Reply to SMS, MMS, and e-mail with any type of message and location of the victim will be displayed to guardian using google maps.

## Chapter 5

# IMPLEMENTATION

Implementation is the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled. The implementation of a system is done after the development effort is completed.

### 5.1 Steps for Implementation

#### 5.1.1 Creating a New Project

To create a new project:

1. Select **File > New > Project**.
2. Select **Android > Android Project**, and click **Next**.
3. Select the contents for the project:
  - a. Enter a *Project Name*. This will be the name of the folder where your project is created.
  - b. Under Contents, select **Create new project in workspace**. Select your project workspace location.
  - c. Under Target, select an Android target to be used as the project's Build Target. The Build Target specifies which Android platform you'd like your application built against.
  - d. Under Properties, fill this all necessary fields.
    - Enter an *Application name*. This is the human-readable title for your application- the name that will appear on the Android device.
    - Enter a *Package name*. This is the package namespace (following the same rules as for packages in the Java programming language) where all your source code will reside.
    - Select *Create Activity* (optional, of course, but common) and enter a name for your main Activity class.
    - Enter a *Min SDK Version*. This is an integer that indicates the minimum API Level required to properly run your application.
4. Click Finish

### 5.1.2 Procedure to Run the Application

Before we run our application on the Android Emulator, you **must** create an Android Virtual Device (AVD). An AVD is a configuration that specifies the Android platform to be used on the emulator.

- **Creating an AVD**

Here's the basic procedure to create an AVD:

1. Open a command-line (eg."Command Prompt" application on Windows, or "Terminal" on Mac/Linux) and navigate to our SDK package's tools/ directory.
2. First, we need to select a Deployment Target. To view available targets, execute: This will output a list of available Android targets.
3. Create a new AVD using your selected Deployment Target. Execute: That's it; our AVD is ready.

- **Run the Application**

To run (or debug) your application, select **Run > Run** (or **Run > Debug**) from the Eclipse main menu. The ADT plug in will automatically create a default launch configuration for the project. When you choose to run or debug your application, Eclipse will perform the following:

1. Compile the project (if there have been changes since the last build).
2. Create a default launch configuration (if one does not already exist for the project).
3. Install and start the application on an emulator or device.

## 5.2 Implementation Issues

The implementation phase of software development is concerned with translating design specifications into source code. The primary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified and so that debugging testing and modification are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity clarity and elegance are the hallmarks of good programs and these characteristics have been implemented in each program module.

The goals of implementation are as follows.

- Minimize the memory required.
- Maximize output readability.
- Maximize source text readability.
- Minimize the number of source statements.
- Minimize development time.

## 5.3 Algorithms

### 5.3.1 Algorithm 1

#### **Sensor value reader algorithm:-**

**Step1:** Start the sensor reading function

**Step2:** If (reading is greater than or less than threshold)

    Sense the acceleration of value change

    Else

    Start sensing again

    Goto Step1

**Step3:** If (acceleration in high)

    Prob is detected

    Else

    Start sensing again

**Step4:** Exit

### **5.3.2 Algorithm 2**

#### **Iprob detection Algorithm or Shake algorithm:-**

**Step1:** Start

**Step2:** Sense the Sensor data to maximum changes

**Step3:** If (Reading is crosses the threshold specified)

    Prob is detected

**Step4:** Start the timer and wait for the response

    If (user responds positively)

        Prob detected is true.

        Alert is sent.

    If (user responds negatively)

        Prob detected is false

        Stop the timer and restart the application

    If (user does not respond)

        Prob is detected

        Alert is sent

**Step5:** Exit



### 5.3.3 Algorithm 3

#### **Message Alert Algorithm:-**

**Step1:** Prompt user to respond after a prob is detected

**Step2:** Start the timer and wait for the response

    If (user responds positively)

        Prob detected is true.

        Goto Step3.

    If (user responds negatively)

        Prob detected is false

        Stop the timer and restart the application

    If (user does not respond)

        Prob is detected

        Goto Step3

**Step3:** Search for the social records from the contact list

    Send the pre-stored alert message

    Wait for response from the contacts in the list

    If (got reply)

        Call is made to the replied contact

        Enable the speaker.

**Step4:** Exit

## Chapter 6

# TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## 6.1 Test Environment

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested. Process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

White box and black box testing are terms used to describe the point of view a test engineer takes when designing test cases. Black box being an external view of the test object and white box being an internal view. Software testing is partly intuitive, but largely systematic. Good testing involves much more than just running the program a few times to see whether it works. Thorough analysis of the program under test, backed by a broad knowledge of testing techniques and tools are prerequisites to systematic testing.

Software Testing is the process of executing software in a controlled manner; in order to answer the question “Does this software behave as specified?” Software testing is used in association with Verification and Validation. Verification is the checking of or testing of items, including software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques as reviews, inspections, walk-through.

## 6.2 Unit Testing Of Modules

### 6.2.1 Module 1

**Table 6.1 Sensor Manager Module testing**

Steps	Test Data	Expected Results	Observed Results	Remarks
Step 1	Vibrating mobile below threshold	No response should be provided	No response	Pass
Step 2	Vibrating mobile above threshold	Message should be sent to the emergency contacts	Message was sent to the emergency numbers	Pass
Step 3	Vibrating mobile above threshold	Message should be sent to the emergency contacts	Message was not sent because of the network issues	Fail
Step 4	Vibrating mobile above threshold casually	Message should not be sent to the emergency contacts	Message was sent to the emergency numbers	Fail

### 6.2.2 Module 2

**Table 6.2 Threshold Manager Module testing**

Steps	Test Data	Expected Results	Observed Results	Remarks
Step 1	2500	No response should be provided	No Response	Pass
Step 2	3000	Message should be sent to the emergency contacts	Message was sent to the emergency numbers	Pass
Step 3	4000	Message should be sent to the emergency contacts	Message was sent to the emergency numbers	Pass
Step 4	4500	Message should be sent to the emergency contacts	Message was not sent because of the network issues	Fail

### 6.2.3 Module 3

**Table 6.3 Contact and SMS Module testing**

Steps	Test Data	Expected Results	Observed Results	Remarks
<b>Step 1</b>	Vibrating the device more than the threshold and the vibration is detected	Message Should be Sent to Three emergency numbers which are previously defined	Message is Sent To three emergency contacts	Pass
<b>Step 2</b>	Vibrating the device more than the threshold and the vibration is detected	Message Should be Sent to Three emergency numbers which are previously defined	Message is not Sent To three emergency contacts because of network issues	Fail
<b>Step 3</b>	Vibrating the device more than the threshold casually but the vibration is detected	Message Should not be Sent to Three emergency numbers which are previously defined	Message is Sent To three emergency contacts	Fail

### 6.2.4 Module 4

**Table 6.4 GPS Manager Module testing**

Steps	Test Data	Expected Results	Observed Results	Remarks
<b>Step 1</b>	Vibration is detected and the app gets started and guardian presses the location button	Location of the victim should be sent to the guardian	Location is sent to guardian through the link	Pass
<b>Step 2</b>	Vibration is detected and the app gets started and guardian presses the location button	Location of the victim should be sent to the guardian	Location is sent to guardian through the link but it is not the accurate location since the victims location is remote unnamed location	Pass
<b>Step 3</b>	Vibration is detected and the app gets started and guardian presses the location button	Location of the victim should be sent to the guardian	Location is sent to guardian through the link but the guardian could not access the location because of network issues	Fail
<b>Step 4</b>	Vibration is not detected because it is lesser than threshold	Location of the victim should not be sent to the guardian	Location is not sent to guardian through the link	Pass

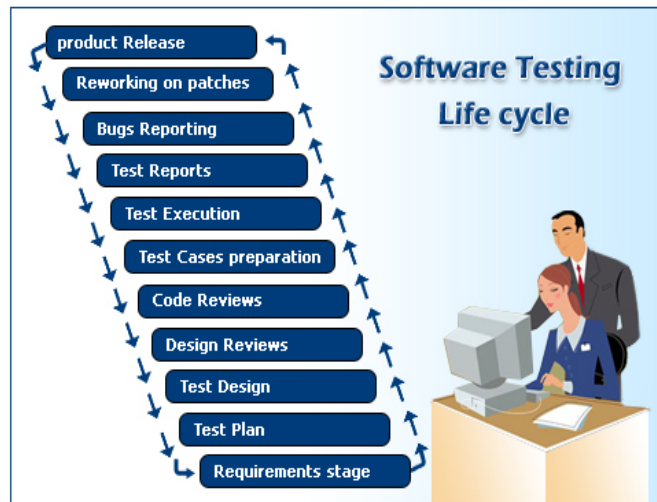
## 6.3 Integration Testing Of Modules

The phase of software testing in which individual software modules are combined and tested as a group. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing.

The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages, In this approach, all or most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing.

The phase of software testing in which individual software modules are combined and tested as a group. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing.

The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages, In this approach, all or most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing.



**Figure 6.1 Software Testing Life-Cycle**

Although testing varies between organizations, there is a cycle to testing:

- Requirements Analysis: Testing should begin in the requirements phase of the software development life cycle.
- During the design phase, testers work with developers in determining what aspects of a design are testable and under what parameter those tests work.
- Test Planning: Test Strategy, Test Plan(s), Test Bed creation.
- Test Development: Test Procedures, Test Scenarios, Test Cases, and Test Scripts to use in testing software.

The phase of software testing in which individual software modules are combined and tested as a group. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing.

The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages. In this approach, all or most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing.

Although testing varies between organizations, there is a cycle to testing:

Not all errors or defects reported must be fixed by a software development team. Some may be caused by errors in configuring the test software to match the development or production environment. Some defects can be handled by a workaround in the production environment. Others might be deferred to future releases of the software, or the deficiency might be accepted by the business user. There are yet other defects that may be rejected by the development team (of course, with due reason) if they deem it inappropriate to be called a defect.

## **6.4 System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points. The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results

## 6.5 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.



## Chapter 7

# RESULTS

This section describes the screens of the “Smart Security solutions for women based on IOT”. The snapshots are shown below for each module.

### Snapshot1

The home pages of the victim application,

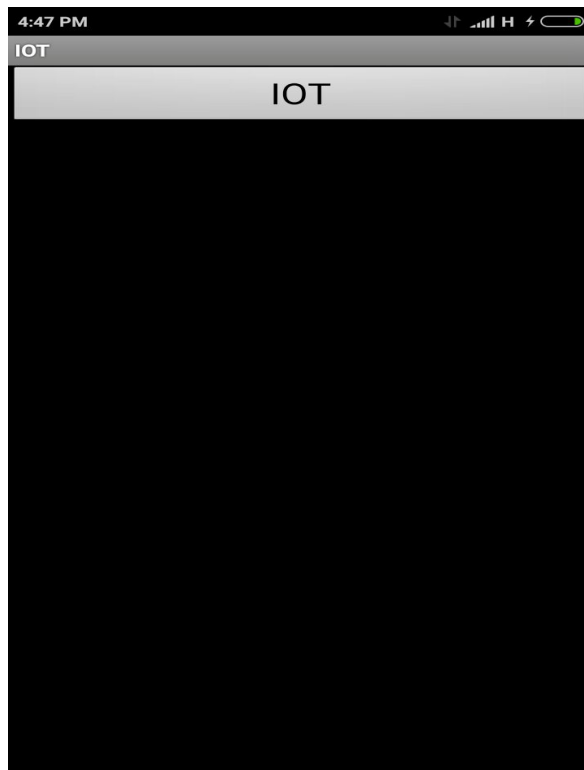
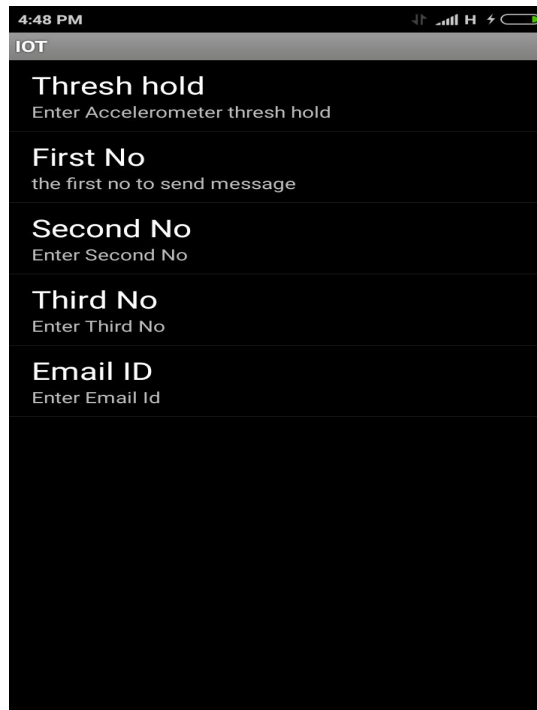


Figure 7.1 Snapshot of Login page

## Snapshot 2

Allows the victim to set the threshold and helps to store the guardian's details

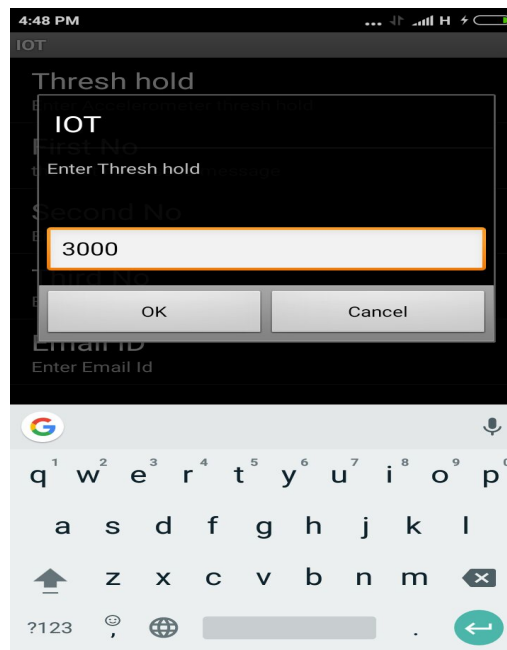


The screenshot shows a mobile application interface with a black background and white text. At the top, the status bar shows '4:48 PM' and various icons. Below the status bar, the app title 'IOT' is displayed. The main content area consists of five vertically stacked input fields, each with a title and a subtitle: 'Thresh hold' (Enter Accelerometer thresh hold), 'First No' (the first no to send message), 'Second No' (Enter Second No), 'Third No' (Enter Third No), and 'Email ID' (Enter Email Id). Each field has a corresponding input area below the subtitle.

Figure 7.2 Snapshot of Welcome page

## Snapshot 3

Setting the required threshold



The screenshot shows the same mobile application interface as Figure 7.2, but with a modal dialog box open over the 'Thresh hold' field. The dialog box has a black background and white text. It contains the title 'IOT', the subtitle 'Enter Thresh hold', and a text input field with the value '3000'. Below the input field are two buttons: 'OK' and 'Cancel'. The background of the app is dimmed. At the bottom of the screen, a virtual keyboard is visible with a Google logo on the left and a microphone icon on the right.

Figure 7.3 Snapshot of threshold setting

## Snapshot 4

### Entering of guardian's details (PHONE NUMBER)

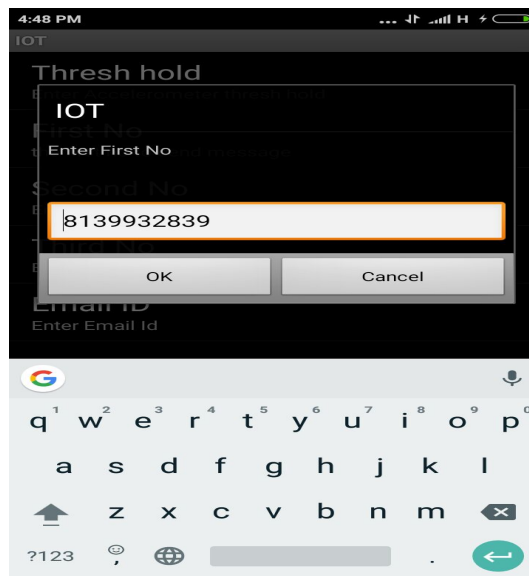


Figure 7.4 Snapshot of entering guardian numbers

## Snapshot 5

### Entering guardian mail id

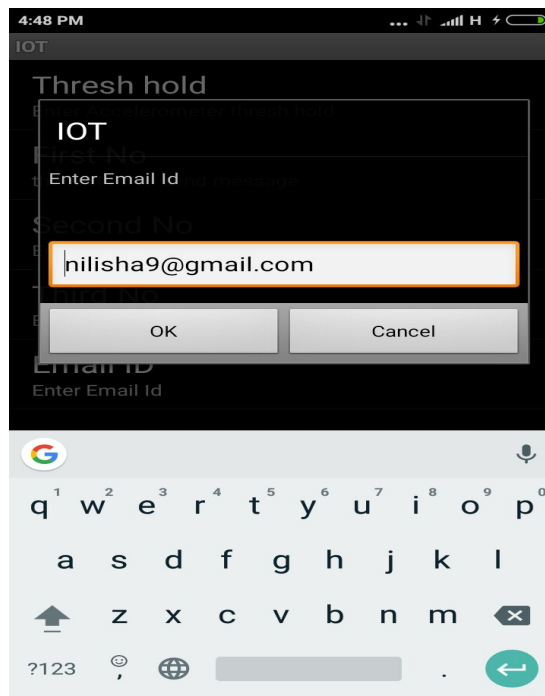


Figure 7.5 Snapshot of Entering email ID of guardian

## Snapshot 6

### Home page of guardian application

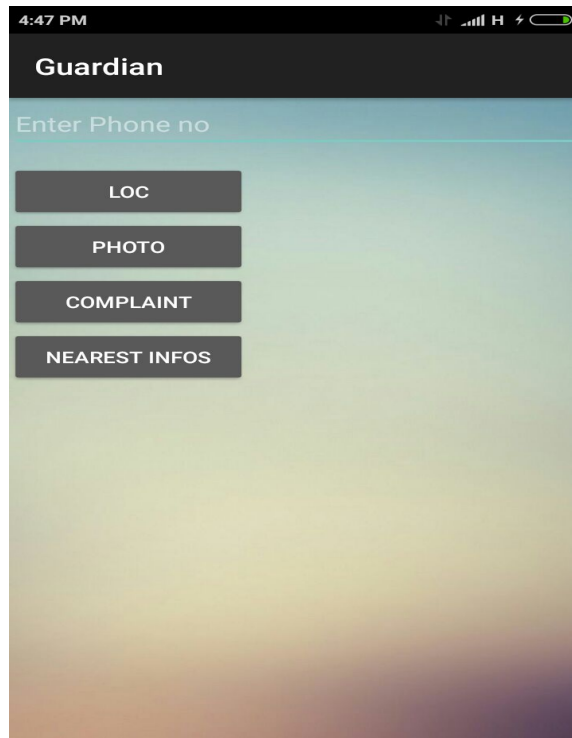


Figure 7.6 Snapshot of guardian app front page

## Snapshot 7

### Tracking the location of the victim

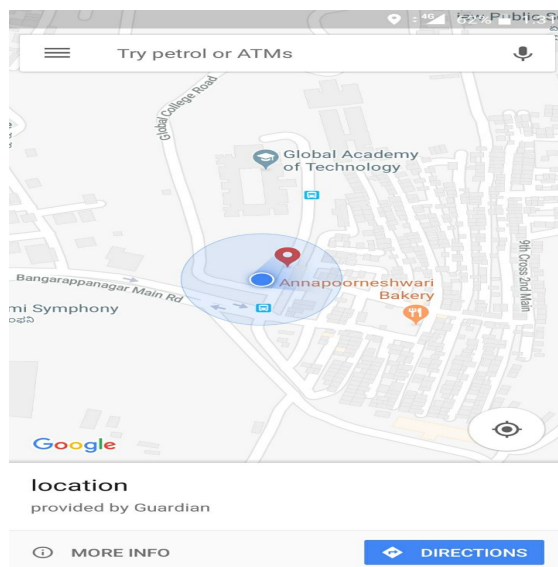


Figure 7.7 Snapshot of location tracking

## Snapshot 8

Fetching the information about nearest emergency places for the victim



Figure 7.8 Snapshot of fetching information about nearby hospitals

## Chapter 8

# CONCLUSION

There are many severe problems that arise all of a sudden, that really cause threat to life. There should be some means to both detect and alert the authorities. Developing such an Android application which is an alert system for PROBLEM detection “iPROB”, was a great experience. The person, who is in trouble-victim, can make use of our application to seek the help from his/her trust worthy people-guardian. The functionality that we provide in our application makes it very simple and easy to use. Whenever a person is in trouble all that he/she got to do is, to just give shake to their cell that has our application deployed in it. Once the shake is given a message seeking for help will be sent. There will not be any necessary to search the contact details, to create message and then send it. All that needed is an Android cell with our application deployed in it. When the guardian responds with a TRACK message, the current location of victim is found. Therefore our application will be more efficient and productive in problem detecting and alerting.

Sometimes the culprits are not punished just because of the lack of evidences. But using our application one can even register the complaint to police against the culprit. Preventive measures can be taken before some crime is committed.

## 8.1 Major contributions

In this current society, we need to make sure that we are at par with the security and safety of all the people during the trouble. This project provides an environment where the person only with his actions, can alert their parents and all the related people if they are in trouble. In addition, to prevent chain snatching, an idea is been developed which aims to reduce such cases in the future.

## 8.2Future Enhancements

Future enhancement of our application is as follows

- The number of people to seek help from can be increased.
- It can be used to help senior citizens.
- It can be used to track kids or the one who gets off the way.
- It can be used by the defense departments to track the prisoners who are found escaped.

## BIBLIOGRAPHY

- [1] A.Chou, J.Yang, B.Chelf, S.Hallem, and D.Engler. An Empirical Study of Android Operating Systems, *In Proc. of 18th ACM Symposium on Operating Systems Principles (SOSP)*, P. 73–88, 2008.
- [2] W.Gu, Z.Kalbarczyk, R.K.Iyer, and Z.Yang. Characterization of Linux Kernel Behavior, *In Proc. of International Conference on Dependable Systems and Networks (DSN)*, 2007, P.4.
- [3] Shaou-Gang Miaou, Pei-Hsu Sung, and Chia-Yuan Huang. A customized human fall detection system using omni-camera images and personal information. Pages 39–41. *Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference, April 2006*
- [4] Rick Rogers, John Combardo, Zigard Mednieks and Blake Meiko. Android Application Development, *programming with Google SDK*, May 2000.
- [5] Hock Beng Lim, “A Soldier Health Monitoring System for Military Applications,” *International Conference on Body Sensor Networks*.
- [6] Palve Pramod, “GPS Based Advanced Soldier Tracking With Emergency Messages & Communication System,” *International Journal of Advance Research in Computer Science and Management Studies Research Article*, Volume 2, Issue 6, June 2014.
- [7] Vamil B. Sangoi, “Smart security solutions,” *International Journal of Current Engineering and Technology*, Vol.4, No.5, Oct-2014.
- [8] Simon L. Cotton and William G. Scanlon, “Millimeter - wave Soldier –to- soldier communications for covert battlefield operation,” *IEEE communication Magazine*, October 2009.