PERSONAL SECURITY SYSTEM AND COMMUNITY SAFETY ANALYSIS

FINAL YEAR PROJECT UG 2019

By

Khizar Tariq

Usama Ahmad

Abdullah

Hassan Masood

Supervised By
Lecturer Danish Ilyas

Submitted to the Faculty of Department of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E Degree in Electrical Telecom Engineering

JUNE 2019

ABSTRACT

PERSONAL SECURITY SYSTEM AND COMMUNITY SAFETY ANALYSIS

Human Security is turning into a major issue even in this cutting edge period of innovation. Each individual venturing out of his home in some way or another feels insecure, particularly ladies all through the world. Pakistan is a developing country and ladies are having an imperative impact towards its improvement by taking a shot at key positions in various public and private sectors yet sadly this has likewise rendered our ladies insecure. World Health Organization claims that 35% of women face unethical harassment around the globe. But we seem to have no such mechanism to avoid such situations and their quick reporting. So this project reiterates on creating a device which can inform the police and victim's relatives just on a click which would enable the police to know that a person is in danger and would help track the victim to minimize the damage. The device has two modules, one is the device itself consisting of GSM and GPS which will send the alert message containing the location information of the victim to the Police, and the second one is the android application which would show the location of the incident occurrence and victim to the nearest Police. The Police can track the victim by clicking on the link which would directly open the android application especially designed for this purpose. The application creates a database of every incident being reported. The principle preferred standpoint of this framework is that the client does not require a Smartphone dissimilar to different applications that have been developed before. Furthermore, utilization of refined segments guarantees exactness and makes it robust.

CERTIFICATE OF CORRECTNESS AND APPROVAL

It is hereby certified that information contained in this thesis titled "Personal Security System & Community Safety analysis" carried out by 1) Khizar Tariq 2) Usama Ahmad 3) Abdullah 4) Hassan Masood with plagiarism percentage of 5% under the vigilant supervision of Lecturer Danish Ilyas in partial fulfillment of B.E Degree in Telecom Engineering is correct and approved.

Approved By

_____

Lecturer Danish Ilyas
Project Supervisor
Department of Electrical Engineering
Military College of Signals
National University of Sciences and Technology


بسم الله الرحمن الرحيم

DECLARATION

The work, or any of its portion, written in this thesis has not been submitted for provision of any other award or qualification, either at this institute or elsewhere.

DEDICATION

In the name of ALLAH All Mighty, the Most Gracious, the Most Merciful and Benevolent. Each individual members of this Final Year Project group dedicates this work and its thesis to his parents and the project supervisor who rendered most effective and sincere guidance throughout each and every phase of this project.

ACKNOWLEDGEMENTS

We would like to thank ALLAH All Mighty for bestowing upon us,His countless and incessant blessings. Whatever we have achieved , we owe it to Him, in totality. We are also highly obliged to our batchmates, acquaintences and relatives for their unflinching support which has helped us achieved all the milestones of our lives. Parents who provided support through their prayers and well wishes. We are extremely thankful to our Project Supervisor, Lecturer Danish Ilyas from Military College of Signals (MCS) who provided us moral support and encouraged us throughout the development of the project besides providing us with valuable technical help and guidance

TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION……1

1.1 Overview……2

1.2 Problem Statement……3

1.3 Objective……3

1.4 Project Scope……3

1.5 Organization of Document……4

CHAPTER 2 – LITERATURE REVIEW……5

2.1 Similar Research/Projects……6
2.2 Various Development Options……6
2.3 Various Development Aspects……7
2.3.1 GSM……7
2.3.2 GPS……7
2.3.3 Raspberry Pi……8

CHAPTER 3 – DESIGN REQUIREMENTS……9

3.1 SIM 808 Module……10
3.1.1 Pin Configuration……10
3.2 Raspberry Pi……11
3.2.1 Raspberry Pi 3 B Technical Specifications……12
3.3 Finger Print Sensor……13
3.3.1 Features……14
3.3.2 Specifications……15
3.4 SD Card……16
3.4.1 SD Card Types……16
3.5 USB to TTL Converter……17
3.5.1 Features……17

CHAPTER 4 – APPLICATION REQUIREMENTS……19

4.1 Android Studio……20
4.2 Firebase Database……20
4.2.1 Services……21
4.2.2 Limitations……21
4.3 Working in the Project……21

CHAPTER 5 – DESIGN METHODOLOGY……23

5.1 Explanation……25
5.1.1 Working of Push Button……25
5.1.2 Working of Finger Print Sensor……26
5.1.3 Working of Raspberry Pi……26
5.1.4 Working of SIM 808……26
5.1.5 Working of Application……27
5.2 Centralization……28
5.3 Community Analysis……28

CHAPTER 6 – HARDWARE DESIGN……30

6.1 Integration of Push Button with Raspberry Pi……31
6.2 Integration of Finger Print Sensor with Raspberry Pi ……31
6.3 USB to TTL Converter……31
6.4 Integration of SIM 808 with Raspberry Pi ……31
6.5 Integration of Battery with Raspberry Pi ……32
6.6 Final Product……32
6.7 Precautions……33

CHAPTER 7 – FUTURE WORK, APPLICATIONS AND CONCLUSION…….....34

7.1 Future Work...……...……35
7.2 Applications...……...……35
7.3 Conclusion.……35

BIBLIOGRAPHY…….……....37

APPENDICES **40**

Appendix - A 41

Appendix - B 47

Appendix - C 53

Appendix - D 65

Appendix - E 67

Appendix - F 69

Appendix - G 70

List of Figures

Figure 3.1 - SIM 808 Module…….……..11

Figure 3.2 – Pin Configuration of Raspberry Pi…….……..…..13

Figure 3.3 – Pin Configuration of FPC 1020 …….……14

Figure 3.4 - SD Card …….……...17

Figure 3.5 – USB to TTL Converter……...18

Figure 4.1 – Coordinates saved in Firebase …….…...…….22

Figure 5.1 – Block Diagram of Smart Security System……..…….……24

Figure 5.2 – Main Activity…….……..27

Figure 5.3 – Map Activity…….…...28

Figure 5.4 – Community Analysis…….….29

Figure 6.1 – Components Integrated in Belt……...…….….32

List of Tables

Table – 3.1: Description of Pin Configuration of FPC 1020 15

Table – 6.1: Connection of Components 33

Table – E.1: Components Used 68

Table – F.1: Budget 69

Table – G.1: Timeline 70

CHAPTER: 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

Personal Security System is a unique device to timely report any unfortunate event that may occur anywhere anytime. In Pakistan, the crime-rate is increasing every day and no steps have been taken for their quick reporting.

This device would not only generates the alert but will also organize a report that would help agencies spying on suspicious activities so that measures can be taken before hand. This would be a valuable contribution in recognizing the nature of crimes and reducing them to a great extent.

1.1 Overview

In this day and age, ladies have faced a noteworthy issue as they can't venture out of their home at some random time because of physical/sexual maltreatment and a dread of viciousness . Indeed, even in the 21st century where the innovation is quickly developing and new devices are developed yet at the same time ladies' and young ladies are confronting issues . It is to remember that the project's core target is women but every one of any age group who feels himself unsecure can benefit from this device. [1]

We are for the most part mindful of significance of security concerns for ladies, yet we should examine that they ought to be appropriately ensured of security. Ladies are not as physically fit as men therefore, in a crisis circumstance some assistance would be help for them. In case someone in some odd situation ,this gadget with you will watch them and can decrease their hazard and bring help when they need it. There are a few similar applications developed in other cases which need mobile phones but they are not efficient as the criminals are smart enough to first grab the phones and render the person unsafe but this sophisticated mechanism would be hidden under the clothes and not easy to grab.

1.2 Problem Statement

This project addresses following problem statement:

- Human security is turning into a noteworthy issue all through the world. There are such crimes like kidnapping, harassment and robbery that occur on daily routine and nothing much has been done yet for the reporting and eradication and reporting.
- No such gadget is being used that can quickly report the police and can track the location of the victim so as to limit the harm.
- So personal security system will guarantee quick lodging of the complaint and assist the police to take necessary actions against the crooks.

1.3 Objectives

This project is based on following objectives:

- To develop smart security device which can enable user to report occurrence of various types of crimes to nearby police stations in no time.
- To develop application for police which can utilize the crime reporting data to show geographical statistics for analysis.
- To use the present technology for enhancing social welfare by providing a low cost device for timely and reliable communication
- To create a database of every user of a device to keep his/her record, so to avoid from any dangerous situation before hand by alerting the user of the most recent crimes happened near by him/her

1.4 Project Scope

This project consists of two modules:

- Personal security module

  Alert will be sent with this module using GSM and GPS to take necessary actions. This is hardware used by the victim.

- Community analysis module

  Each incident reported through security module will be resolved and recorded in the database to make analysis of the high crime rate areas. This consists of application being used by Police to see the location of victim.

1.5 Organization of Thesis

The thesis consists of Seven chapters numbered from 1 to 7. Chapter 1 is about the Introduction of the project giving a picture to the reader of what is the purpose of the project and the scope. Chapter 2 is the Literature review. All the study and help taken for the development from the already existing papers of similar projects is sighted in this chapter. Chapter 3 is the Design Requirements. It gives all the information about the components used for the development, their technical specifications and features. Chapter 4 is about the Application Requirements, giving information about all the requirements and their details that are used for the development of the mobile application. Chapter 5 is the Design Methodology. This explains the working and the sequence of processes of each individual component. Chapter 6 is the Hardware Design. It explains the design of the security belt along with all the connections. Chapter 7 is the Future Work, Application and the Conclusion. This chapter explains the applications of the project and further work that can be done for the improvements.

CHAPTER: 2

LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

2.1 Similar Research/Projects

The presented project derived inspiration from the research presented by Ashwini.P. in [2].

This paper recommends viewpoint to utilize innovation to ensure safety of humans. The system when initiated, tracks the location of the victims utilizing GPS (Global Positioning System) and sends crisis messages utilizing GSM (Global System for Mobile Communication), to three crisis contacts. The main advantage of this system is that the user does not require a Smartphone not at all like different applications that have been developed before. The system furnishes with every one of the highlights which will investigate every possibility to help the victim in any sort of crisis circumstances.

In this paper, following shortcomings are observed:

1. This is not available in Pakistan.
2. This is Arduino based which has lengthy processing time and is not much reliable.
3. The system is not centralized.
4. It focuses on just alerting the relatives and does not cater for minimizing the damage.

To overcome these limitations, the concept of centralization, quick reporting and analysis is introduced in the presented personal security system. Therefore the presented project provides basis with prototype proof of concept which can validate development and utility of such a device in our country Pakistan. This device would be the only device in Pakistan.

2.2 Various Development Options

The authors of [3]&[4] explain the data logging process used to estimate the path. It explains the need to use different algorithms for path estimation because vast use of GPS hand-held devices has increased the amount of data on the routes of people as they walk, drive or otherwise go about journeys. This information has been utilized in various ways - for instance the Open Street Map venture (The Open Street Map Foundation 2007) [5]. One characteristic of projects such as this is that there will be several GPS tracks for the same length of road. In general, repeatedly measuring something and taking the average of measurements leads to a more accurate result.

The question addressed here is “is it possible to ’average’ GPS tracks and if so, does this lead to a better estimate of road location?”. For this purpose Principle Curve analysis and Line simplification algorithms are used to estimate the path. But these are quite time taking and requires much processing and are much complex. The smart security project does not use such complex algorithms instead it uses A* algorithm to estimate the shortest path and alternative routes in real-time. A* algorithm is a updated type of Breadth first search. It avoids the costly route and picks the most encouraging route. It is an extremely brilliant algorithm. It is utilized to approximate the shortest path in real-life situations.

2.3 Various Development Aspects

Detailed literature survey for establishing sound understanding of various technologies involved in the presented project was done explained below.

2.3.1 GSM

GSM is 2nd generation cellular systems known as Global Systems for Mobile Communications. It is used for voice calls and short message service. It does not incorporate a system for sending data packets to internet.

The system to send data packets was first introduced in GPRS but it has low speed and less bandwidth. It used the TDMA technology. An advancement came in the wireless communication system with the emergence of UMTS also known as 3G. In this both voice and data packets can be send with the little change in GSM architecture. It uses Node B instead of BTS and has separate path for voice and data. It uses WCDMA technology and offers high speed and bandwidth.[6]

2.3.2 GPS

GPS is a satellite-based navigation system which was created by U.S. Department of Defense. It is a part of NAVSTAR satellite program. It gives us the location and time information to a GPS in all climate conditions, anywhere throughout the world [7]. The technical specifications can be studied from [8].

2.3.3 Raspberry Pi

Embedded Technology is advancing all around quickly as far as rapid multi core processors, high sensor incorporation with chips, Low power utilization and low footprints, embedded security features, powerful guidance sets, simple to utilize improvement devices, Real time execution and so forth. Along these lines, receiving these most recent advancements to make world class items and arrangements is the greatest current pattern in embedded system development. A microcontroller is an IC that is used in embedded system. It has processor, memory and I/O peripherals. It is used in many applications such as automobiles, medical applications, industrial processing etc. Micro Controllers are usually 4 bit, 8 bit or 16 bit processors whereas raspberry pi can be think of entire computer with CPU, RAM, GPU, Wi-Fi and Ethernet and USB Controller. Raspberry Pi can perform much more advance tasks and it needs an operating system to run on it contrary to microcontrollers that are programmed to perform only a specific operation. [9]

The operating systems that are usually used in raspberry pi are RASPBIAN, PIDORA, OPENELEC, RASPBMC, RISC OS, and ARCH LINUX. All these operating systems are Linux based. [10]

CHAPTER: 3

DESIGN REQUIREMENTS

CHAPTER 3

DESIGN REQUIREMENTS

This chapter gives a detailed information of the components used and explains the working of each. SIM 808, Raspberry Pi, Finger Print Sensor, USB to TTL converter and SD cards are the main parts of the belt. The pin configuration, technical specifications and all the relevant information is included. It is necessary to understand what the components do and their purpose to have a clear insight of the project.

3.1 SIM 808 Module

SIM808 module is a GSM and GPS two-in-one capacity module. It works on GSM/GPRS Quad-Band system and joins GPS innovation for satellite route. It highlights ultra-low power utilization in rest mode and incorporated with charging circuit for Li-Ion batteries, that influence it to get a very long reserve time and advantageous for undertakings that utilization battery-powered Li-Ion battery. It has high GPS get affectability with 22 tracking and 66 acquisition recipient channels. Furthermore, it additionally underpins A-GPS that accessible for indoor limitation.The module is constrained by AT direction by means of UART and supports 3.3V and 5V voltage level. It uses Quad-band frequencies of 850/900/1800/1900MHz. It is shown in figure 3.1

3.1.1. Pin Configuration:

PWR: this is delicate power switch for the module, you can press it for 2 seconds to power on or off the module.

RI: this pin will reveal to you whether the module is on and is there any calls and messages received. It will be pulled to high state when the module is on. Furthermore, it will change to low for 120ms when a call or message is detected.

RST: this is reset pin. Naturally it has a high pull up. This pin is used in the event in the module is behaving abnormally to its default settings.

RXD/TXD: Serial port, the module utilizes it to send and receive commands and information. TXD is output, and RXD is input. They can be associated with the 3.3V and 5V level.

VIO: this is the reference rationale level for sequential port of the module, the information voltage relies upon the consistent dimension of the micro controller you use. In the event that you utilize a 5V micro controller like Arduino, you ought to have it be 5V, and a 3V rationale micro controller you should set it to 3V.

BAT: this pin is associated with Li-Ion battery pin, in the event that you need to utilize just a single Li-Ion battery as your capacity supply, you can utilize it to drive you microcontroller.

DTR: this is wake up pin for module in rest mode. As a matter of course it has a high draw up, and you can set the module into rest mode by AT order "AT+CSCLK=1". Meanwhile, the sequential port will be debilitated. The sequential port will be dynamic after DTR stick is dismantled to low dimension for about 50ms. [11]

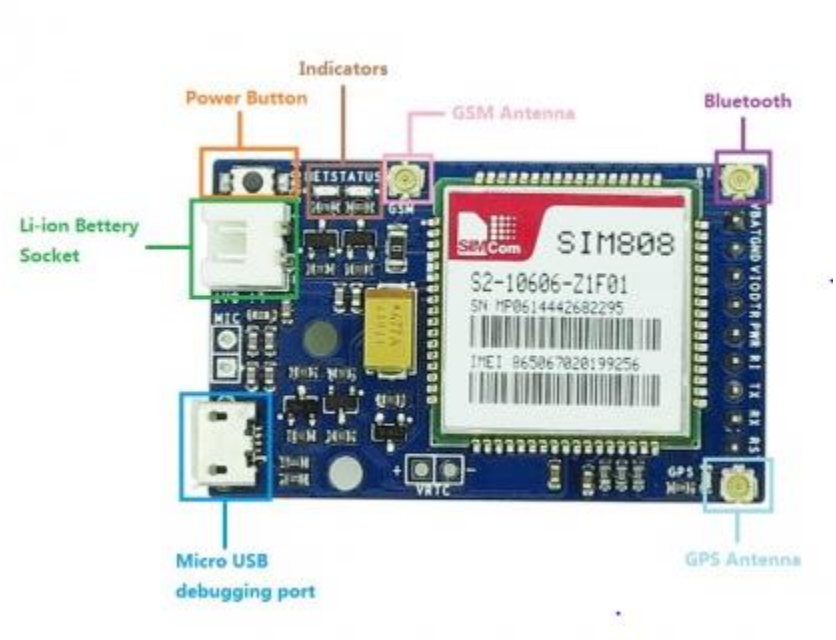


Figure 3.1: SIM 808 Module

3.2 Raspberry Pi

The Raspberry Pi is a single board computers, developed by Raspberry Pi Foundation. They are amazingly of small size and are capable of performing all the functions that is expected from a PC. The Raspberry Pi is a cost effective PC that runs Linux, yet it likewise gives a lot of GPIO (broadly useful information/yield) sticks that enable you to control electronic segments for physical processing and investigate the Internet of Things (IoT). Everywhere throughout the

world, individuals use Raspberry Pi to pick up programming aptitudes, fabricate equipment ventures, do home computerization, and even use them in mechanical applications. The Raspberry Pi propelled in 2012, and there have been a few emphases and varieties introduced since. The first Pi had a single core 700MHz CPU and simply 256MB RAM, and the most recent model has a quad-core 1.4GHz CPU with 1GB RAM [12].

In this Project Raspberry Pi 3 B has been used. It is the earliest model of the third-age Raspberry Pi. It supplanted the Raspberry Pi 2 Model B in February 2016.The operating system used in Pi is Raspbian Linux. To access Pi either a separate monitor screen can be used or can be accessed on laptop screen by creating a ssh session using an application called putty. It is shown in figure in fig 3.2

3.2.1 Raspberry Pi 3B Technical Specification

Following are the technical specifications of Raspberry Pi 3 B

- BCM 2387 chipset developed by Broadcom
- Cortex A53 processor of ARM family with 1.2 GHz clock Quad core
- Maximum RAM is 1 GB
- Four USB ports are provided in the package
- A stereo output is also available accessible with 4 poles stereo output
- Video output can be accessed with a composite video port
- High Definition Multimedia Interface is also available
- Provides Ethernet connection using 10/100 Base Ethernet socket
- Raspberry Pi camera can also be connected using CSI Camera Port
- OS is loaded using Micro SD card and for storing information
- Touch screen display is connected to DSI display Port
- The CPU used is of 64 bits
- Supports IEEE 802.11 Wireless LAN standard
- Supports Bluetooth 4.1

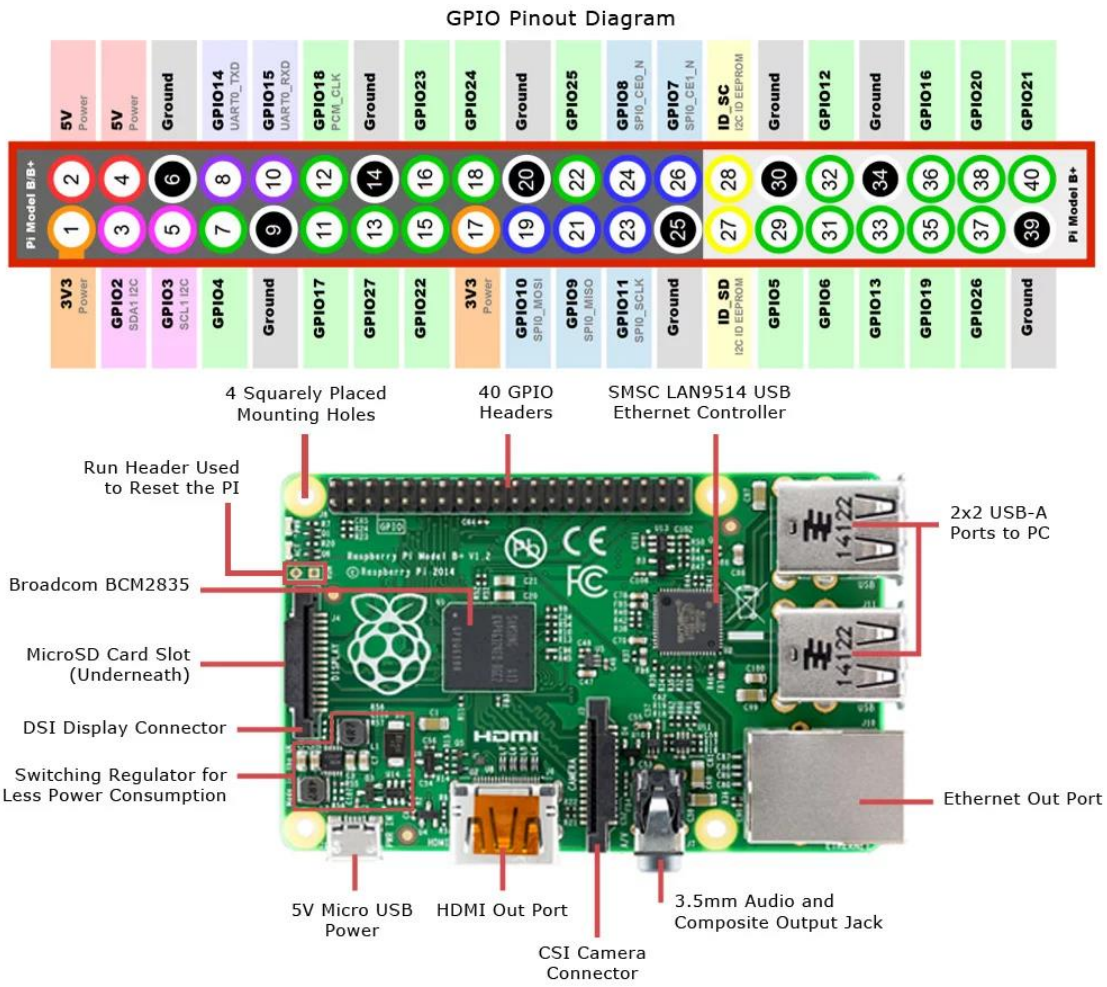


Figure 3.2: Pin Configuration of Raspberry Pi [13]

3.3 Finger Print Sensor

This is a unique mark contact sensor pack base on FPC1020 contact unique finger impression sensor, it is created with a finger impression sensor FPC1020 and a control board.

The unique mark sensor's mode is FPC1020 from Fingerprint Cards, which has been generally utilized on cell phones. It depends on capacitive sensor innovation. It conveys top tier picture quality with 256 dim scale esteems in each and every programmable pixel. Regardless of dry, wet or grimy finger, it can peruse with its 3D pixel detecting innovation. The control board depends on an ARM® Cortex®-M processor, it has a pre-consumed program inside for handling the UART directions. Through this board, we can straight forwardly drive the unique

mark sensor by means of sequential port. It is shown in fig 3.3. Pin configuration is shown in Table 3.1

3.3.1 Features

Following are the features of Finger Print sensor FPC 1020

- The interface is Grove compatible
- It stores 256 true Gray scale values in every pixel and works best for class imaging
- The fingerprint template is stored in 200 bytes
- Offers 1:N identification i.e. One-to-Many
- Offers 1:1 verification i.e. One-to-One
- Fingerprint features are automatically updated\
- Allows settings related to security
- More than 10 million fingerprint placements are capable protected using Robust protective coating

Figure 3.3: Pin Configuration of FPC 1020

3.3.2 Specifications

Following are the technical features of Fingerprint Sensor 1020

- 508 dots per inch resolution is offered by it
- 100 fingerprints can be stored at a time
- Fingerprints are verified in less than 0.45 sec
- Fingerprints identification time is also less than 0.45 sec
- The False Accept rate is less than 0.0001%
- The False Reject Rate is less than 0.01%
- Default 19200 bps rate using UART serial interface
- Various Baud Rates i.e 9600,19200,38400,57600,115200 bps
- Less than 50 mA current is required for the operation
- In standby condition the less than 10uA current is required
- The supply voltage is 5V DC
- 2.8 – 5V DC Digitial I/O voltage
- Operational temperature is -20 C – 60 C

NAME	I/O	DESCRIPTION
V-TOUCH	POWER	Power for finger detected function, 5V or 3.3V
TOUCH	OUTPUT	Output high(3.3V) when finger detected, otherwise low
VCC	POWER	5V power input
UART_TX	OUTPUT	Transmitter of TTL serial
UART_RX	INPUT	Receiver of TTL serial
GND	POWER	Power Ground

Table 3.1: Description of Pin Configuration of FPC 1020 [14]

3.4 SD Card

A Secure Digital (SD) card is a memory card for storing huge amounts of data and stores the data digitally. It is called secure card as the information stored in it is never lost and remains intact as long as you want.

A Secure Digital card is about the degree of a postage stamp and weighs around two grams. It is near in size to a MMC, yet tinier than increasingly prepared memory card types, for instance, a Smart Media card or CompactFlash card. A SD card incorporates a high data conversion standard and low battery usage, which are both fundamental examinations for flexible contraptions. A SD card uses streak memory to give nonvolatile limit, which suggests a power source isn't required to hold set away data.

Both MMC and SD cards give encryption abilities to tied down substance to ensure secure movement of copyrighted material, for instance, electronic music, video and computerized books .It is shown in fig 3.4

3.4.1 SD Card Types

Secure Digital card advancement consolidates the going with:

- SD Standard Capacity. Limits as for SDSC cards keep running from 128 megabytes to 2 GB. The default plan for these cards is FAT16 (File Allocation Table 16).
- SD High Capacity. In perspective on the SDA 2.0 detail, limits concerning SDHC cards keep running from 4 GB to 32 GB. The default position for these cards is FAT32.
- SD extended Capacity. In perspective on the SDA 3.0 assurance, limits concerning SDXC stretch out from 64 GB to 2 terabytes. The default bunch for these cards is ex FAT (Extended FAT).
- SD Input Output. SDIO cards combine I/O limits with data amassing. Beginning at 2016, the basic courses of action are full or little scale gauge SDHC and SDXC cards. [15]

Figure 3.4: SD Card

3.5 USB to TTL Convertor (PL2303)

It is used to convert serial port into USB port for serial communication. We are using PL2303 convertor in our project. It is shown in figure 3.5

3.5.1 Features [16]

Features of USB to TTL Converter are as below:

- Supports serial communication
- It has digital 5V operational level
- External power supply it works on is 5V
- It is also used for ADSL or Router firmware update
- Supports serial communication for GPS also
- Serial debugging tool
- Various satellite machines are upgraded using this
- USB to TTL converter supports different operating systems such as Windows7,XP etc

Figure 3.5: USB to TTL Convertor

CHAPTER: 4

APPLICATION REQUIREMENTS

CHAPTER 4

APPLICATION REQUIREMENTS

The application is Android based it is capable of running on all the versions of Android OS. The following are used for the designing of application

4.1 ANDROID STUDIO

It is the integrated development environment for Google Android Application Development. In Android Studio, JAVA language is used. Java Development kit (JDK) is used for developing java code and Java Runtime Environment (JRE) used to run Java code. Every activity in Android studio contains two files, source code and XML file. Source code contains the JAVA code in which different classes are imported and XML file defines the layout of the activity. It is free and available for different operating systems like Windows, macOS and Linux. It replaces the previous tool which was used for android development Eclipse Android Development. Since May 2019 the preferred language by Google for android development is Kotlin but still supporting different languages. Different versions have been released of Android Studio, the latest version is Android Studio 3.4. This version provides the following features

- It supports Gradle Build
- It provides support for Android Wear apps
- It has built-in support for Google Cloud Platform, Firebase and Google App Engine
- Android Studio has option for virtual device emulator to run and debug errors
- Easy layout editor
- Automatically notifying the error

4.2 FIREBASE DATABASE:

Firebase Database is online storage for storing information and notification for android

development. It is free and easily acquired globally. Firebase has 18 different products which are used by around 1.5 million apps. Firebase startup was started by James Tamplin and Andrew Lee in 2011. Firebase's first product was Real-time Database, it is an API that syncs apps data across Android, Web devices and iOS and store it on Firebase's cloud. In 2014, Firebase acquired by Google. The coordinates saved in Firebase are shown in fig 4.1

4.2.1 Services

Firebase provide the following services

- Firebase Authentication
- Cloud Messaging
- Real-time Database
- Cloud Fire store
- Storing and Hosting
- ML Kit

4.2.2 Limitations

Firebase has some limitations as follows

- Limited to 50 connections and 100 mb storage
- User don't host the data, Firebase does
- No reporting tools
- It uses JSON, so it would be difficult to migrate

4.3 Working in the project:

In this project, Real-time database is used. To store data, a reference is created to a point where the data is going to be stored. This reference is used to store the data in a child by calling a function of SetValue (). One reference can be used to store values in one parent child where it refers. Storing the data is easy as compared to fetching data from the firebase. To fetch data, another reference is created to a specific child whose value is desired. To fetch value, addListenerForSingleValueEvent () function is used through the reference. In this function, a loop is created in which Data snapshot is passed which carries all data which is going to be

fetched, then a reference of data snapshot is created to fetch the data by calling function of getValue (). In this project, two parent child are created named as Users and Analysis. In Users, only latitude and longitude are stored to locate the victim. In Analysis, coordinates of every crime are stored with specific "id" to perform the community analysis which is shown in fig below

Figure 4.1 Coordinates saved in Firebase

CHAPTER: 5

DESIGN METHODOLOGY

CHAPTER 5

DESIGN METHODOLOGY

In this chapter, application and hardware approach is discussed used for the development of the project. A block diagram is shown below in fig 5.1 to demonstrate the methodology.

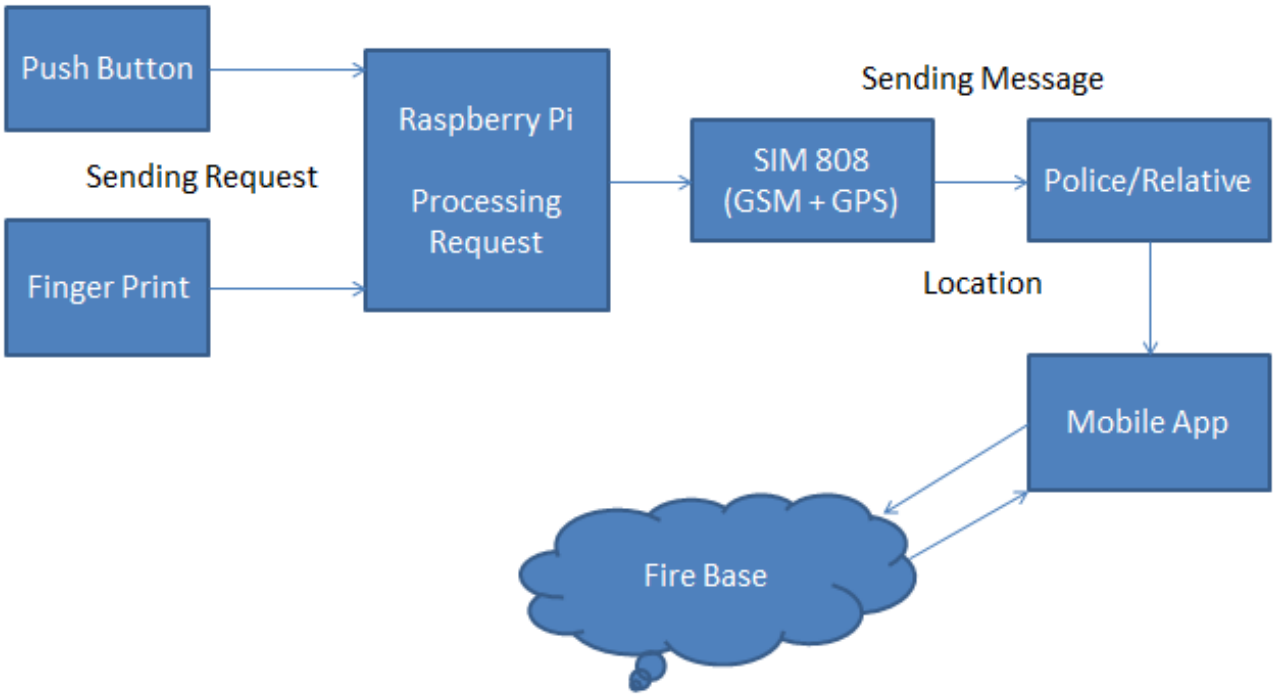


Figure 5.1: Block Diagram of Smart Security System

Personal Security System is a device for alerting the police instantly, so that police can do its job without any delay ensuring quick response which would ultimately minimize the loss in case of crime activity which is potentially dangerous to ones' personal security.The person would be able to send its location to the police stations that are near to him using the centralized system. The concerned police station can see his location using the application specialized designed for this purpose. The block diagram is explained below:

The system firstly requires the person to authenticate himself by pressing his/her thumb on the fingerprint. As soon as the person is authenticated, raspberry pi starts gathering the location and

store in its database for centralization purpose. Push button is used to send the current location to the police. When push button is pressed a requested is initiated to Raspberry Pi to send the location. SIM 808 gathers the location through GPS and sends it through GSM to the numbers saved in the Pi. As soon as the message having the location coordinates arrives the mobile phone the application installed auto-fetches the co-ordinates and shows this location on the map. As the location is updated after every certain period only the first coordinate sent by one user is updated in firebase and similarly for other users, these first location are shown on the maps in the form of red marker showing the details of crime area.

It includes the following steps:

- GPS data logging using SIM808 module
- Interface Raspberry Pi to allow GSM to send specific format SMS to specific number.
- Configuring and interfacing of fingerprint sensor and push button with Raspberry Pi.
- Using centralization method which will send message to the nearest police personnel.
- Interface the android app with Google maps API and SMS sending/receiving.
- Link android app so that it can take information from formatted SMS and show corresponding indication on the map.
- Add features into the application so that it can show, with geographical map, areas of high occurrence of crimes.

5.1 Explanation

The working of each component is explained step by step below:

5.1.1 Working of Push Button

A push button is used to power on the device. Once the device is powered on it automatically updates the current location of the user. It creates the database of all the routes, the user travels through for the day. The location of the route is determined in the real time which tells the time, the user was on particular location. This unique feature helps in determining the sensitive hours in which the user may be vulnerable. The location is updated after a specific interval.

5.1.2 Working of Finger Print Sensor

A finger print sensor is used to authenticate the user and to generate an alert message containing the real time location to the nearby police stations using the centralizing system by pressing the thumb on finger print sensor, as soon as he feels he/she is in danger and encounters any undesirable situation

5.1.3 Working of Raspberry Pi

All relevant information for emergency contacts and database for real-time location record using GPS logging is maintained here. The Raspberry pi is a stand-alone device like minicomputer. The operating system used in it is Raspbian. It is the main component which can be called the brain of the project. All the components such as SIM 808, Fingerprint Sensor, Push Button are integrated to the raspberry pi. It drives all the components simultaneously. A power source of supplies the necessary voltage to drive Pi and SIM808.

All the programming is done in Pi which makes it to control all the devices, automating all the processes. 808 uses AT Commands for its working. These necessary AT commands are provided by Raspberry Pi.

As soon as the fingerprint sensor is pressed, raspberry pi authenticates the user by comparing the finger patterns saved already. If the patterns do not match the database in the raspberry pi, it won't generate the alert. This provision prevents illegal use of the device. When the fingerprint got the pattern and that pattern matches the database, it means the user is authenticated and it will generate the alert

5.1.4 Working of SIM 808

Sim 808 is basically the communication and positioning module of the project.Sim808 module consists of GSM, Bluetooth, GPRS, GPS. It has 2 sockets of antenna. One socket for GSM antenna and other for GPS antenna. It also has a slot of SIM on the backside of the 808 module. A valid SIM is inserted into the module to send the messages.

The raspberry pi is programmed with AT Commands which initiates the module 808 to gather the current location of the user using GPS and send the co-ordinates (longitude and latitude) in a specific format message to the nearby police station using GSM module.

5.1.5 Working of Application

Raspberry Pi send message through SIM808 which contain alert message as well as coordinates in a special format. As message received, the app will automatically fetch coordinates from the message with specific id number. There is save button in the display, by pressing that button the coordinates will be saved in Firebase Database. Firebase is basically real-time online storage for storing information and notification in android. After the coordinates saved in firebase, press the Proceed button which will lead to the Map Activity. Map activity will fetch the coordinates from the base and place a marker on that location. Main activity is shown in fig 5.2 while map activity is shown in fig 5.3.

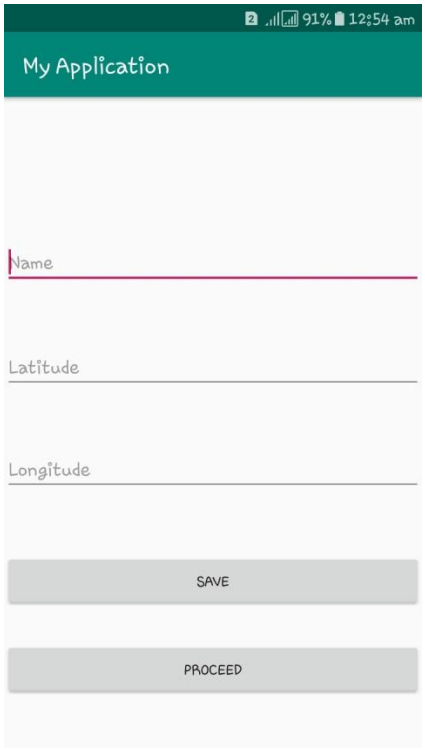


Figure 5.2: Main Activity

Figure 5.3: Map Activity

5.2 Centralization

The raspberry pi is centralized by storing the emergency cell phone numbers along with the corresponding co-ordinates of the police station in the database. When the user presses the fingerprint then a request send to the raspberry pi which in turn initiates a request to 808 module to get the current co-ordinates. These co-ordinates will be compared with the co-ordinates stored in the database of all the police stations and the message is sent to the police station having minimum distance with the co-ordinates gathered by 808.

5.3 Community Analysis

Another part of the application is the analysis of crime for the people. Analysis will show all the crimes happened and reported through the smart security belt. People can identify which path is better for them to reach their destination. This is done by storing the first location from all the crimes as the message is received. When anyone presses the Analysis button, it will lead to same map activity. But this time it will fetch all the coordinates from the firebase and show markers on all the locations. It is to be remembered that the markers show the location at the instant the

fingerprint sensor is pressed and first message which is sent, as the location is updated after every particular instant these updated locations are not shown. This community analysis is an amazing feature which will also help police determine more vulnerable areas and arrangements can be made to cater for beforehand. Screenshot of community analysis aspect of the developed app is shown in fig 5.4

Figure 5.4: Community Analysis

CHAPTER: 6

HARDWARE DESIGN

CHAPTER 6

HARDWARE DESIGN

In hardware design, how the components are integrated to each other and what is the final product of hardware is discussed. The integration of following components with Raspberry Pi (which is the central part of device) is given below:

6.1 Integration of Push Button with Raspberry Pi

The GND pin of push button is connected to the ground pin of Raspberry Pi. The P1 pin of push button is connected to the 17 pin of Raspberry Pi. The P2 pin of push button is connected to the 18 pin of Raspberry Pi. These Push buttons were connected with the help of jumper wires, no soldering was required yet.

6.2 Integration of Finger Print Sensor with Raspberry Pi

The Fingerprint sensor is integrated with the raspberry Pi with the help of USB to TTL converter.

6.3 USB TO TTL Converter

USB to TTL adapter is used to allow devices to communicate serially with each other. Since, Pi's serial ports i.e pin no. 4 & 5 were being used for other purpose so it was decided to use USB to TTL converter for integrating FPS with Raspberry Pi.

6.4 Integration of SIM 808 with Raspberry Pi

The Rx pin of SIM 808 is connected to the Tx pin of Raspberry Pi. The Tx pin of SIM 808 is connected to the Rx pin of Raspberry Pi. The GND pin of SIM 808 is connected to the GND pin of Raspberry Pi. These connection were soldered with the Raspberry Pi and SIM808, because these connections serve as the main communication for us, Jumper wires can help that too, but Jumper wires can adhere the communication of SIM808 and Raspberry Pi.

6.5 Integration of Battery with Raspberry Pi

The battery is connected to the adapter that converts the 15V from the battery into 5V/3A each for raspberry pi and SIM 808.The connections of adapter with the battery are soldered, and from adapter wires are soldered that can power up raspberry Pi and SIM808 at a time.

6.6 Final Product

The end product is in the form of belt. This belt would be tied on abdomen so that it is hidden and the criminals can't see them. All the security system is inside the belt. The user will only see a push button and fingerprint sensor outside as only these two are to be used by the owner of the belt.

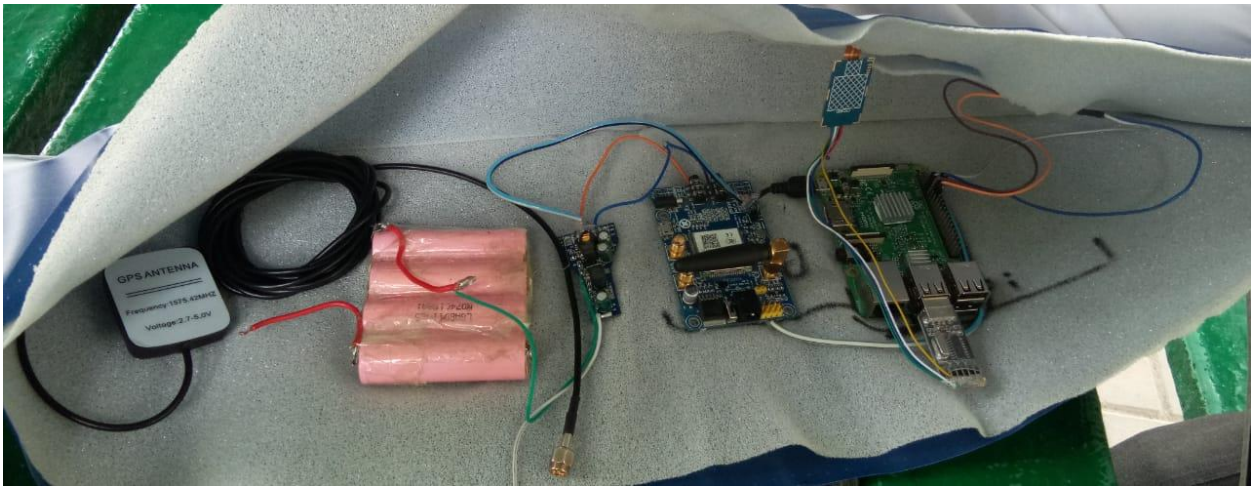


Figure 6.1 : Components integrated in BELT

Raspberry Pi	SIM808	Finger Print Sensor	Push buttons
Tx	Rx	-	-
RX	TX	-	-

GND	**GND**	**-**	**GND**
17	**-**	**-**	**P1**
18	**-**	**-**	**P2**
4	**-**	**-**	**-**
USB PORT	-	**USB**	**-**

Table 6.1 : Connection of components

6.7 Precautions:

1. Raspberry Pi should not be given more than 5V/3A, it may damage the Pi,
2. Heat sinks should be used for the proper functioning of Pi,
3. SIM808 can operate in two different parameters. i.e. 5V/3A and 12V/1A. these are the suitable working calculations for SIM 808.
4. USB to TTL should not be given more than 5V to operate.
5. FPS needs 5V to operate, more than that can damage the sensor.

This Product cannot be used under water, it will damage the whole circuitry

CHAPTER: 7

FUTURE WORK, APPLICATIONS AND CONCLUSION

CHAPTER 7

FUTURE WORK, APPLICATIONS AND CONCLUSION

7.1 Future Work

Personal Security System is a foundation stone towards revolutionizing the concept of security of the human beings. With this the people will be safer than they currently are. The future work mainly focus on the enhancement of the features and minimizing the existing problems.
Further improvements in this product can be :

- The size of the device can be made more compact and easy to handle by using SMT (Surface Mount Technology).
- A complete health monitoring system can also be introduced with the separate centralized system and android application to rapidly send health report to the doctors and relatives.
- This system lacks on the spot safety. As a future work shockwave sensor can also be integrated for self-defense.
- Voice recording and video recording technologies can also be integrated.

7.2 Applications

- Safe community oriented projects
- High profile person security measures
- Big data based crime analysis for police
- Vigilant response campaign for law and security enforcing agencies.

7.3 Conclusion

In this project, we have developed a smart security system using Sim 808 module for sending location while designing application utilizing android software which will help in

- Fast reporting

- Community analysis
- Excluding the need of smartphones

It has two components one is the security module and the other one is community analysis module. Security module is the physical device that will be with the user used to send his/her current location and the other module will get this location and shows it in the map along with the feature of community analysis to guide the users of the device of the high crime rate areas, so that they should choose the path accordingly. Besides this the community analysis will enable the police determining these areas and take security measures there beforehand.

The proposed design will deal with critical issues faced by people, especially women who work in odd hours or by anyone who feels himself vulnerable to such activities .While the society may or may not change its mind set but this device will help the humanity feel more secure and independent.

BIBLIOGRAPHY

[1] Vaibhav A. Alone. "A Study Based On Women Security System" International Journal of Science, Engineering and Technology Research (IJSETR)

[Accessed June 20, 2018]

[2] Ashwini .P. Thaware . " A Safety Device for Women's Security Using GSM/GPS" International Journal on Recent and Innovation Trends in Computing and Communication

[Accessed June 20, 2018]

[3] Brunsdon, Chris. "Path estimation from GPS tracks." *Proceedings of the 9th International Conference on GeoComputation*. National Centre for Geocomputation, Maynooth University., 2007.

[Accessed August 24, 2018]

[4] Taylor, George. "Line simplification algorithms." *Retrieved 15 April 2005* (2005).

[Accessed August 24, 2018]

[5] The Open Street Map Foundation, 2007

Available: http://www. openstreetmap.org

[Accessed August 24, 2018]

[6] Haykin, S. "Moher. Modern Wireless Communications. Editorial Pearson Prentice Hall." *Ontario, Canada, Febrero*(2005).

[Accessed September 12, 2018]

[7] Parthasarathy, Jagannathan. "Positioning and navigation system using gps." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science* 36.Part 6 (2006): 208-212.

[Accessed October 7, 2018]

[8] GPS Module Data Sheet by SKYLAB

Available: http://sensorembedded.com/product_extra_files/gps.pdf

[Accessed October 7, 2018]

[9] Mauricio González, Liliana González, Jaime Echeverri, Miguel Aristizábal, Germán Urrego, Ana Lucía Pérez, "Method to determine optimal hardware platforms in Human Centered Computing based on non functional requirements analysis", *Information Systems and Technologies (CISTI) 2014 9th Iberian Conference on*, pp. 1-6, 2014.

[Accessed December 18, 2018]

[10] Chaudhari, Harshada. "Raspberry pi technology: A review." *International Journal of Innovative and Emerging Research in Engineering* 2.3 (2015): 83-87.
[Accessed December 18, 2018]

[11] Detailed Pin Configuration of SIM 808
Available: https://www.itead.cc/wiki/SIM808_GSM/GPRS/GPS_Module
[Accessed May 15, 2019]

[12] Introduction to Raspberry Pi
Available: https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/
[Accessed May 16, 2019]

[13] GPIO Pins of Raspberry Pi
Available: https://www.raspberrypi.org/documentation/usage/gpio/
[Accessed May 16, 2019]

[14] Pin Configuration of FPC1020
Available: https://www.shenzhen2u.com/Capacitive-Fingerprint-Sensor-FPC1020.html
[Accessed May 17, 2019]

[15] Types of SD Card
Available: https://searchstorage.techtarget.com/definition/Secure-Digital-card
[Accessed May 18, 2019]

[16] Features of USB to TTL Convertor
Available: https://www.rhydolabz.com/interfacing-modules-c-80/pl2303-usb-to-ttl-module-p-1244.html
[Accessed May 19, 2019]

APPENDICES

APPENDIX A

Application Code:

Main Activity:

This activity will demonstrate how the coordinates will be fetched from the message and saved in firebase.

Code:

```
package com.example.myapplication;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
```

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener,SmsListener {
    private static final int REQUEST_ID_MULTIPLE_PERMISSIONS = 1;
    private DatabaseReference mDatabase,mD;
    private Button btnsave;
    private Button btnproceed;
    private EditText editTextName;
    private  EditText editTextLatitude;
    private EditText editTextLongitude;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnproceed=(Button) findViewById(R.id.btnproceed);
        mDatabase= FirebaseDatabase.getInstance().getReference().child("Users");
        mD= FirebaseDatabase.getInstance().getReference().child("Analysis");
        editTextName=(EditText)findViewById(R.id.editTextName);
        editTextLatitude=(EditText)findViewById(R.id.editTextLatitude);
        editTextLongitude=(EditText)findViewById(R.id.editTextLongitude);
        btnsave=(Button)findViewById(R.id.btnsave);
        btnsave.setOnClickListener(this);
        checkAndRequestPermissions();
        SmsReceiver.bindListener(this);

        btnproceed.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v){
                Intent i=new Intent(MainActivity.this,MapsActivity.class);
                i.putExtra("key",1);
                startActivity(i);
            }
```

```
    });
}
private void saveUserInformation(){
    //int id=Integer.parseInt(editTextName.getText().toString().trim());
    double latitude= Double.parseDouble(editTextLatitude.getText().toString().trim());
    double longitude= Double.parseDouble(editTextLongitude.getText().toString().trim());
    UserInformation userInformation=new UserInformation(latitude,longitude);
    mDatabase.child("Users").setValue(userInformation);
    Toast.makeText(this,"Saved",Toast.LENGTH_LONG).show();
}

@Override
public void onClick(View view) {
    if(view==btnproceed){
        Intent i=new Intent(MainActivity.this,MapsActivity.class);
        i.putExtra("key",1);
        startActivity(i);
    }
    if(view==btnsave){
        saveUserInformation();
        editTextName.getText().clear();
        editTextLatitude.getText().clear();
        editTextLongitude.getText().clear();
    }

}
private boolean checkAndRequestPermissions()
{
    int sms = ContextCompat.checkSelfPermission(this, Manifest.permission.READ_SMS);

    if (sms != PackageManager.PERMISSION_GRANTED)
```

```
        {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_SMS}, REQUEST_ID_MULTIPLE_PERMISSIONS);
            return false;
        }
        return true;
    }

    @Override
    public void messageReceived(String messageText) {
        boolean flag=false;
        for(int i=0;i<messageText.length();i++){
            if(messageText.charAt(i)=='/'){
                flag=true;
            }
        }
        try {
            if(flag==true){
                String[] alpha=messageText.split("/",2);
                String a=alpha[0];
                String[] beta=a.split(":",2);
                String b=beta[0];
                String c=beta[1];
                String[] gamma=b.split(",",2);
                String[] tetra=c.split(";",2);
                String d=gamma[0];
                String e=gamma[1];
                String f=tetra[0];
                String g=tetra[1];
                final int id=Integer.parseInt(f);
```

```
final double latitude =Double.parseDouble(d);
final double longitude =Double.parseDouble(e);
editTextLatitude.setText(String.valueOf(latitude));
editTextLongitude.setText(String.valueOf(longitude));
editTextName.setText(String.valueOf(id));
int identifier=Integer.parseInt(g);
if (identifier==1) {
   mD.addListenerForSingleValueEvent(new ValueEventListener() {
      @Override
      public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
         if(dataSnapshot.exists()){
            for(DataSnapshot s:dataSnapshot.getChildren()){
               Info user=s.getValue(Info.class);
               ArrayList<Integer> list;
               list = new ArrayList<>();
               list.add(user.id);
               if (list.contains(id)) {
                  Info info = new Info(id, latitude, longitude);
                  mD.child("User" +id).setValue(info);
               } else {

                  Info info = new Info(id, latitude, longitude);
                  mD.child("User" + id).setValue(info);
               }
               list.clear();
            }

         }else{
            Info info=new Info(id,latitude,longitude);
            mD.child("User"+id).setValue(info);
         }
```

```
                }

                @Override
                public void onCancelled(@NonNull DatabaseError databaseError) {

                }
            });
        }
    }

    }catch (Exception e){

    }
  }
}
```

APPENDIX B

Application Code:

Maps Activity:

this activity will show the location of the victim on the map. On the other hand it also show analysis for the community.

Code:

```
package com.example.myapplication;

import android.location.Location;
import android.location.LocationListener;
import android.support.annotation.NonNull;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.Toast;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.firebase.FirebaseApp;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
```

```
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class MapsActivity extends AppCompatActivity implements OnMapReadyCallback,
LocationListener,GoogleMap.OnMarkerClickListener {

    private GoogleMap mMap;
    private ChildEventListener mChildEventListener;
    private DatabaseReference mUsers, mAnalysis;
    Marker marker;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_maps);

        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
        ChildEventListener mChildEventListener;
        mUsers = FirebaseDatabase.getInstance().getReference("Analysis");
        mAnalysis=FirebaseDatabase.getInstance().getReference("Users");
        mUsers.push().setValue(marker);
        mAnalysis.push().setValue(marker);
    }
```

```
@Override
public void onMapReady(GoogleMap googleMap) {
   int key = getIntent().getExtras().getInt("key");
   Toast.makeText(this, String.valueOf(key), Toast.LENGTH_SHORT).show();
   mMap = googleMap;
   googleMap.setOnMarkerClickListener(this);
   googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
   if (key == 0){
      mUsers.addListenerForSingleValueEvent(new ValueEventListener() {
         @Override
         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for (DataSnapshot s : dataSnapshot.getChildren()) {
               Info user = s.getValue(Info.class);

               LatLng location1 = new LatLng(user.latitude, user.longitude);
               LatLng location2 = new LatLng(user.latitude, user.longitude);

               MarkerOptions markerOptions1 = new MarkerOptions();
               MarkerOptions markerOptions2 = new MarkerOptions();



               // Setting the position for the marker
               markerOptions1.position(location1);
               markerOptions2.position(location2);

               // Setting the title for the marker.
               // This will be displayed on taping the marker
               markerOptions1.title(location1.latitude + " : " + location1.longitude);
               markerOptions2.title(location2.latitude + " : " + location2.longitude);
```

```
                // Clears the previously touched position
                //  mMap.clear();

                // Animating to the touched position
                mMap.animateCamera(CameraUpdateFactory.newLatLng(location1));
                mMap.animateCamera(CameraUpdateFactory.newLatLng(location2));


                // Placing a marker on the touched position
                mMap.addMarker(markerOptions1);
                mMap.addMarker(markerOptions2);


                mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(location1,
16.0f));
                mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(location2,
16.0f));

            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}
if(key==1) {
    mAnalysis.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
```

```
            for (DataSnapshot s : dataSnapshot.getChildren()) {
                UserInformation user = s.getValue(UserInformation.class);

                LatLng location = new LatLng(user.latitude, user.longitude);

                MarkerOptions markerOptions = new MarkerOptions();

                // Setting the position for the marker
                markerOptions.position(location);

                // Setting the title for the marker.
                // This will be displayed on taping the marker
                markerOptions.title(location.latitude + " : " + location.longitude);

                // Clears the previously touched position
                mMap.clear();

                // Animating to the touched position
                mMap.animateCamera(CameraUpdateFactory.newLatLng(location));

                // Placing a marker on the touched position
                mMap.addMarker(markerOptions);

                mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(location,
16.0f));
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
```

```
            }
        });
    }
}

@Override
public void onLocationChanged(Location location) {

}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {

}

@Override
public void onProviderEnabled(String provider) {

}

@Override
public void onProviderDisabled(String provider) {

}

@Override
public boolean onMarkerClick(Marker marker) {
    return false;
}}
```

APPENDIX C

Hardware Code:

```
from pyfingerprint.pyfingerprint import PyFingerprint
from time   import sleep
import   RPi  .GPIO   as   GPIO
import   datetime
import   serial
import   json
import   math
import   hashlib
import   time
currentDT=  datetime.datetime.now  ()
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup  (18 ,   GPIO.IN , pull_up_down = GPIO.PUD_UP)
GPIO.setup  (17,   GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup  (4,   GPIO.OUT)
f=""
ser   =   serial.Serial  (
port='/dev/ttyS0',
baudrate = 9600,
parity=serial.PARITY_NONE,
stopbits  =  serial.STOPBITS_ONE  ,
bytesize  =  serial.EIGHTBITS  ,
timeout  =5
)
#Database for Police Personnels
validNumber = ""
defaultNumber = "+923355857907"
latitudeArray = {}
firstArray = []
```

```
secondArray = {}
secondArray["secondArray"] = [ 33.584799 , 73.093443 , "+923337705655" ]
firstArray.append(secondArray)
secondArray = {}
secondArray["secondArray"] = [ 33.582632 , 73.097815 , "+923164308696" ]
firstArray.append(secondArray)
secondArray = {}
secondArray["secondArray"] = [ 33.578610 , 73.099945 , "+923164308696" ]
firstArray.append(secondArray)
secondArray = {}
secondArray["secondArray"] = [ 33.583346 , 73.104251 , "+923164308696" ]
firstArray.append(secondArray)
secondArray = {}
secondArray["secondArray"] = [ 33.580634 , 73.104243 , "+923355857907" ]
firstArray.append(secondArray)
latitudeArray["mainArray"] = firstArray
dataJsonInfo        = json.dumps(latitudeArray)
parsedDataJsonInfo  = json.loads(dataJsonInfo)
def button2():
if GPIO.input(17) == GPIO.LOW :
print 'Button is Pressed First Time'
while True:
time.sleep(0.01)
if GPIO.input(17) == GPIO.LOW:
print 'Button is Pressed 2nd time'
return 1
def Vib1():
GPIO.output(4, True)
time.sleep(1)
GPIO.output(4, False)
time.sleep(1)
```

```
def Vib2():
  GPIO.output  (4,   True)
   time.sleep(2)
  GPIO.output  (4,   False)
  time.sleep(2)
def Vib3():
    GPIO.output(4, True)
  time.sleep(3)
  GPIO.output(4,   False)
  time.sleep(3)

"""def loop2():
s = 0
while True:
if button2() == 1:
while True:
s = s + 1
time.sleep(1)
print s
if s == 10:
if button2()==1:
s = 0
s = s + 1
time.sleep(1)
print s
if s == 10:
LocationMessage()
else:
LocationMessage()
time.sleep(0.1)
print 'Button Not Pressed'
```

```
"""
# This methods intends to detach GPS module in normal mode
# @parameter : NULL
# @returns   : boolean (as per state)
def detachGPS():
 ser.write("AT+CGNSPWR=0\r\n")
 sleep(1)
while(True):
response =   ser.readline()
if "OK" in response:
print("OK Found in Detached")
break
if "OK" in response:
print("Detached Successfully..")
return True
else:
return False
# This methods intends to invoke the module in GPS mode
# and will give acknowledgement whether module is working
# or not
# @paramters : NULL
# @returns   : boolean (as per state)
def attachGPS():
  response = ' '
 ser.write("AT+CGNSPWR=1\r\n")
 sleep(2)
 response =   ser.readline()
       ("First Response is:\t" + response)
while(True):
if "OK" in response:
print("Ok Found..")
```

```
break
else:
response = ser.readline()
print("While is :\t" + response)
   ser.write("AT+CGNSTST=1\r\n")
   time.sleep  (2)
   response = ' '
while(True):
if "OK" in response:
print("Ok Found..")
break
else:
response = ser.readline()
print("While is :\t" + response)
return True
# This function intends to get the values of GPS Response only
# from the GPS Module and Will return the value whose 18 index
# value is A (which actually is valid one)
# @parameter : NULL
# @return    : String
def getGpsResponse():
while(True):
recievedBuffer = ser.readline()
print("Received Line is:\t" + str(recievedBuffer))
if len(recievedBuffer) < 20:
continue;
elif recievedBuffer[18] == "A":
break;
print("Desired Response is:\t" + recievedBuffer)
return recievedBuffer
# This fucntion intends to parse the values of GPS from the
```

```
# String returned from the "getGpsResponse()" method
# @parameter : String from above mentioned Method
# @return    : String as formation of (Latitude : value , Longitude : value)
def getGpsFromResponse(responseString):
splitedString       = responseString.rsplit(",")
print(splitedString)
longitude       = float(splitedString[5])
longitudeDirection  = splitedString[6]
latitude        = float(splitedString[3])
latitudeDirection   = splitedString[4]
degreesLatitude     = math.floor(latitude / 100)
minutesLatitude     = latitude - ( 100 * degreesLatitude)
minutesLatitude        /= 60
degreesLatitude        += minutesLatitude
if (latitudeDirection == "S"):
degreesLatitude *=  -1
latitude        = degreesLatitude
degreesLongitude    = math.floor(longitude / 100)
minutesLongitude    = longitude - (100 * degreesLongitude)
minutesLongitude       /= 60
degreesLongitude       += minutesLongitude
if (longitudeDirection == "W"):
degreesLongitude *= -1
longitude       = degreesLongitude
gpsInfo   = "{ \"Latitude\" : \"%f\" , \"LatitudeDirection\" : \"%s\"  , \"Longitude\" : \"%f\" ,
\"LongitudeDirection\" : \"%s\" }" %(latitude, latitudeDirection , longitude,
longitudeDirection)
print("GPS INFO (%s)"%gpsInfo)
return gpsInfo
def makeDataBase(responseJson):
print("Response Json:\t" + str(responseJson))
```

```
parsedJson = json.loads(responseJson)
longitude  = parsedJson["Longitude"]
latitude   = parsedJson["Latitude"]
f=open('index.txt','a')
f.write('\nLatitude : ' + latitude + ' Longitude : ' + longitude + ' ' + currentDT.strftime("%Y-
%m-%d %H:%M:%S"))
f.close()
def makeDataBase2():
#count += 1
attachGPS()
responseString = getGpsResponse()
gpsInfo = getGpsFromResponse(responseString)
makeDataBase(gpsInfo)
print("Response Json:\t" + str(responseJson))
parsedJson = json.loads(responseJson)
longitude  = parsedJson["Longitude"]
latitude   = parsedJson["Latitude"]
detachGPS()
f=open('index2.txt','a')
f.write( '-' + '\nLatitude : ' + latitude + ' Longitude : ' + longitude + ' ' +
currentDT.strftime("%Y-%m-%d %H:%M:%S"))
f.close()
def sendMessage(number , message):
ser.write("AT\r\n")
sleep(1)
responseMessage = ser.readline()
print("First Response in Send Message is:\t" + responseMessage)
while(True):
if "OK" in responseMessage:
break
responseMessage = ser.readline()
```

```
if "OK" in responseMessage:
ser.write("AT+CFUN=1\r\n")
sleep(1)
while(True):
if "OK" in responseMessage:
break
responseMessage = ser.readline()
      ("First OK..")
print("Second Response in Send Message" + responseMessage)
if "OK" in responseMessage:
 ser.write("AT+CMGF=1\r\n")
 sleep  (1)
 while(True):
if "OK" in responseMessage:
break
responseMessage = ser.readline()
print("Second OK")
print("Third Response in Send Message" + responseMessage)
if "OK" in responseMessage:
ser.write("AT+CMGS=\"%s\"\r\n" %number)
sleep(1)
sendThis = message.encode('raw_unicode_escape')
ser.write("%s" %sendThis)
sleep(1)
ser.write(chr(26))
print("Message Sent..")
sleep(3)
def findCloseAreaNumber(compareValueLatitude , compareValueLongitude):
latitudeArrayDifferences = []
longitudeArrayDifferences = []
counter = 0
```

```
for i in parsedDataJsonInfo["mainArray"]:
print(parsedDataJsonInfo["mainArray"][counter]["secondArray"][0])
latitude = parsedDataJsonInfo["mainArray"][counter]["secondArray"][0]
if (latitude > compareValueLatitude):
latitudeArrayDifferences.append(latitude - compareValueLatitude)
else:
latitudeArrayDifferences.append(compareValueLatitude - latitude)
counter += 1
print("Latitude Array Differences")
print(latitudeArrayDifferences)
print("Close Value is")
print(min(latitudeArrayDifferences))
print(latitudeArrayDifferences.index(min(latitudeArrayDifferences)))
latitudeCloseIndex = latitudeArrayDifferences.index(min(latitudeArrayDifferences))
print("============================================================")
counter = 0
for i in parsedDataJsonInfo["mainArray"]:
print(parsedDataJsonInfo["mainArray"][counter]["secondArray"][1])
longitude = parsedDataJsonInfo["mainArray"][counter]["secondArray"][1]
if (longitude > compareValueLongitude):
longitudeArrayDifferences.append(longitude - compareValueLongitude)
else:
longitudeArrayDifferences.append(compareValueLongitude - longitude)
counter += 1
print("Latitude Array Differences")
print(longitudeArrayDifferences)
print("Close Value is")
print(min(longitudeArrayDifferences))
print(longitudeArrayDifferences.index(min(longitudeArrayDifferences)))
longitudeCloseIndex = longitudeArrayDifferences.index(min(longitudeArrayDifferences))
print("============================================================")
```

```
if (longitudeCloseIndex == latitudeCloseIndex):
return str(parsedDataJsonInfo["mainArray"][longitudeCloseIndex]["secondArray"][2])
else:
return str(parsedDataJsonInfo["mainArray"][latitudeCloseIndex]["secondArray"][2])
 except   Exception   as e:
 print('   Operation failed  !')
 print(   'Exception message: ' +   str(e))
 exit  (1)
 def LocationMessage():
sleep(2)
attachGPS()
responseString = getGpsResponse()
gpsInfo = getGpsFromResponse(responseString)
makeDataBase(gpsInfo)
detachGPS()
parsedJson = json.loads(gpsInfo)
longitude  = parsedJson['Longitude']
latitude   = parsedJson['Latitude']
print("Current Latitude is:\t" + str(latitude))
print("Current Longitude is:\t" + str(longitude))
validNumber = findCloseAreaNumber(float(latitude) , float(longitude))
while True:
if (counterLoop == 0):
print("This is First Time..")
message = "Hello, I am in trouble.\n Please Help me.\nMy current location is %s.\nYou can
track my location at:
https://maps.google.com/maps?q=%s,%s\n"%(gpsInfo,latitude,longitude)
counterLoop += 1
detachGPS()
if validNumber:
print("Valid Nunmber:\t" + str(validNumber))
```

```
sendMessage(validNumber , message)
else:
print("Not a Valid Number\n")
else:
attachGPS()
responseFromGPS = getGpsResponse()
gpsInfoNew = getGpsFromResponse(responseFromGPS)
jsonParsed = json.loads(gpsInfoNew)
longitudeNew = jsonParsed['Longitude']
latitudeNew  = jsonParsed['Latitude']
messageNew = "Hello, I am in trouble.\n Please Help me.\nMy current location is %s.\nYou
can track my location at:
https://maps.google.com/maps?q=%s,%s\n"%(gpsInfoNew,latitudeNew,longitudeNew)
detachGPS()
print("Current Latitude:\t" + str(latitudeNew))
print("Current Longitude:\t" + str(longitudeNew))
validNumber = findCloseAreaNumber(float(latitudeNew) , float(longitudeNew))
if validNumber:
print("Valid Number:\t" + str(validNumber))
sendMessage(validNumber , messageNew)
time.sleep(10)
counterLoop = 0
count = 0
# Starting Main Program
if Verify() == 1:
while True:
 time.sleep  (1)
if   GPIO.input(18)   ==     GPIO.LOW  :
 print("Button is Pressed")
LocationMessage()
else:
```

```
#while True:
print("Button is Not Pressed..")
time.sleep(1)
#makeDataBase2()
#time.sleep(3)
```

APPENDIX D

SYNOPSIS

Personal Security System

Extended Title: Personal Security System And Community Safety Analysis
Brief Description of The Project / Thesis with Salient Specifications: Day by day the women safety is becoming the common issue, such apps do exist, and kidnappers are equally smart to confiscate the victim's phone. Hence the strategy to switch to an independent hardware is focused in our project. Here we introduce a device which ensures the protection of women by allowing them to timely report any personal security threat to concerned authorities.
Scope of Work: The project will consist of 2 separate modules: First module is Personal Security Band, other one is Police Community Safety System. Personal security band will comprise of GPS positioning system, wireless communication system for crime/security threat reporting. User will also have a mobile application that will display the crime analysis from police departments. Police community Safety Analysis system will act as a receiver to display the position of victim, as well as they will get to know which areas are specifically related to crimes by creating a database from each crime report.
Academic Objectives : • Study and application of wireless communication and Global Positioning System • To go through the process of project development involving embedded systems
Application / End Goal Objectives: • To develop smart personal security reporting band which can enable user to report occurrence of various types of crimes wirelessly to nearby police. • To develop application for police which can utilize the crime reporting data to show geographical statistics for analysis.
Previous Work On the Subject: This project has been experimented in other countries but not in Pakistan.

Material Resources Required:

- Raspberry Pi 3 B
- Finger Print Sensor FPC1020
- USB to TTL Convertor
- SIM 808 Module
- SD Card Class 10
- Push Button
- Voltage Regulator
- Power Supply

No of Students Required : 04

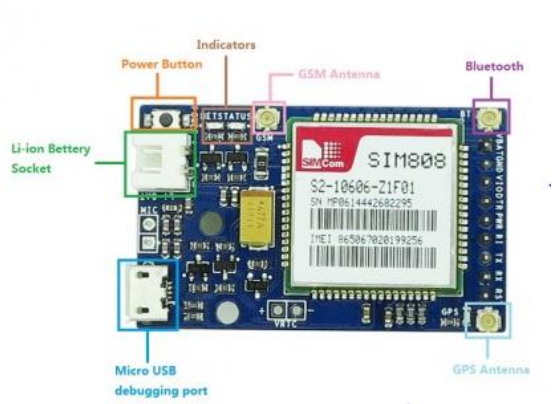
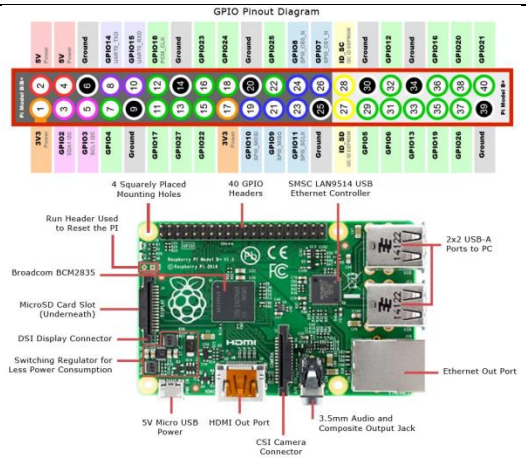
Group Members: NC Khizar Tariq
NC Hassan Masood
NC Usama Ahmad
NC Abdullah

Special Skills Required:

- Raspberry Pi OS programming
- Android development

APPENDIX E

COMPONENTS USED

Component Name	Figure
SIM 808	
Raspberry PI 3B	
Finger Print Sensor FPC 1020	

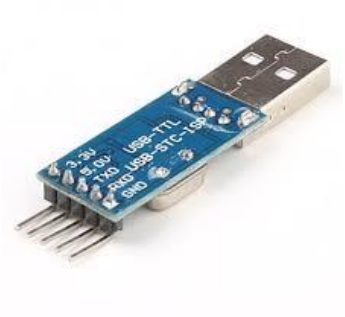
SD Card	
USB to TTL Convertor	

Table E.1: Components Used

APPENDIX F

Cost Breakdown

Sr #	Equipment	Quantity	Price (Rs)
1	Raspberry Pi 3B	1	4700/-
2	SIM808(GSM+GPS)	1	3300/-
3	FPS 1020	1	9000/-
4	SD Card Cass (10)	1	1300/-
5	Miscellaneous	-	1000/-
		Total: 19,200	

Table F.1: Budget

APPENDIX G

Plan of Action

Sr#	Date	Task	Status
1	June 2018	Project Synopsis and Approval	Completed
2	July – Oct 18	Literature Review	Completed
3	Nov-Dec 18	Hardware Implementation(Receive SMS and Location)	Completed
4	Jan 19	Creating Database GUI of Application	Completed
5	Feb-Mar 19	Interfacing FPS Centralizing the system	Completed
6	Apr 19	Analysis for the community Thesis Write-up	Completed
7	May19	Finalizing	Completed

Table G.1: Timeline