ARDUINO BASED INFRARED INTRUSION DETECTION AND MOTION SENSOR DEVICE WITH ANDROID APP

Sidney Nyasha Richards

(R151895G)

ARDUINO BASED INFRARED INTRUSION DETECTION AND MOTION SENSOR DEVICE WITH ANDROID APP

By

SIDNEY NYASHA RICHARDS (R151895G)

Submitted in partial fulfilment of the requirements for the degree of

BSc Honours Computer Science

Department of Computer Science and Information Systems

in the

Faculty of Science and Technology

at the

Midlands State University

Gweru

May 2019

Supervisor: **Mr M. Giyane**

ABSTRACT

Intrusion detection is a series of processes which involves the detection, warning and notification of the arrival of an unwanted entity within the vicinity of installed sensors and the execution of necessary actions required to be performed so as to alert the members of the asset to be monitored and increase the security of the building simultaneously. The growth of intrusion detection systems is largely attributable to a decrease in costs plus enhancements to the technology in terms of additional miniaturization and software optimization that, together, have made this type of security more accessible. In line with this notion this study was carried out to develop a working prototype that uses the ultrasonic sensor for intrusion detection which is then linked with other components to come up with a robust prototype. The current technological developments, notably the improvements in precision and reduced costs of developing and running intrusion detection systems, have served very much as key drivers for the implementation of such systems in many industries. The ideas implemented were inferred from previous models which used intrusion detection devices as a tool for constant monitoring of assets. This document highlights all the necessary steps undergone in the development of the prototype from the planning phase to the operation phase. The intrusion detection prototype which is composed mainly of an ultrasonic sensor, 4x4 keypad membrane and an LCD is interfaced with an android application to enable remote control of such systems which was not there before. The future scope of this study is mainly centred on having the system protect itself from the intruder. By using intrusion detection devices, institutions believe they will be able to provide the level of security necessary to reduce the risk of theft, while also reducing operational risks through more efficient constant monitoring devices.

DECLARATION

I, **Sidney Nyasha Richards,** hereby declare that I am the sole author of this dissertation. I authorize the **Midlands State University** to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

Signature: ……………………………………………. Date: ………………………………..

APPROVAL

This dissertation, entitled **"Arduino Based Infrared Intrusion Detection and Motion Sensor Device With Android App"** by **Sidney Nyasha Richards** meets the regulations governing the award of the degree of **BSc Computer Science** of the **Midlands State University,** and is approved for its contribution to knowledge and literary presentation.

Supervisor's Signature: ……………………………………………….

Date: ………………………………………………………………..

ACKNOWLEDGEMENTS

I would like to give my sincere gratitude to all the people that have influenced me directly and indirectly to get to this level in my academic endeavours. My greatest gratitude and love will and always go to GOD for blessing me with numerous gifts that I am truly and eternally grateful for.

I thank my family for all their love, support and encouragement they have given me throughout this whole process. My Friend's and fellow classmate's whose assistance was invaluable.

Lastly and most importantly I would like to thank Mr M Giyane and all the department lecture staff for their support and assistant's in the development and completion of this dissertation.

.

DEDICATION

This dissertation is dedicated to all researchers with interests in Intrusion Detection Systems. I hope it serves as a source for the further development of this field.

TABLE OF CONTENTS

LIST OF ACRONYMS

GSM-GLOBAL SYSTEM FOR MOBILE COMMUNICATION

WIFI-WIRELESS FIDELITY

SMS-SHORT MESSAGE SERVICE

IDE-INTERGRATED DEVELOPMENT ENVIRONMENT

MIT-MASSACHEUTS INSTITUTE OF TECHNOLOGY

UART-UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER

GUI-GRAPHICAL USER INTERFACE

IEEE- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

IWHSS- LOW POWER INTELLIGENT WIRELESS AND HOME SECURITY SYSTEM

LCD-LIQUID CRYSTAL DISPLAY

ASCII-AMERICAN STANDARD

LED-LIGHT EMITTING DIODE

CCTV-CLOSED CIRCUIT TELEVISION

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Introduction

The aim of this study is to design and make an intrusion detection device that would assure constant monitoring is achieved. It is designed intently with the reason that it might enable property holders to live in their homes with total guarantee that while they are unaware their homes/premises will dependably be protected and under consistent checking. The chapter will focus on the various types of sensors to be applied as well as the supporting components. The utilization of an intrusion detection system is neglected by the vast majority however with a steady ascent in theft cases, property holders are now, more than ever in need of intrusion detection systems that can diminish unnoticed interruptions for the occupants. This is one way that can help the occupants, to deter intruders and in also with catching the culprit. In such scenario's, it is hence desirable to have an intrusion detection system built up that is in charge of helping the occupants protect their valuables.

The Arduino-based Intrusion detection system will unquestionably be productive in the protection and surveillance of premises with the aid of alarms to caution the approaching intruder. The final product is a system that alerts when an interruption occurs thus quickly enabling occupants to take action and protect their assets.

1.2 Background of Study

Existing intrusion systems in current use in the industry and homes are now obsolete, lack sophistication and do not give greater control to the end-user. Ramadan (2010) stated that the primary function of these systems which is to detect intrusion and then trigger an alarm is a noble concept but this is now obsolete as a single emergency response team is contacted to arrive at the site. The background of study identified some deficiencies in existing systems so it has been identified that instead of having one emergency response team being reached out to respond to an emergency why not have contact with multiple emergency response teams at once and this should include those who are very near the site.

Furthermore, due to the mere fact that people are attached to their mobile devices why not include greater control of the system via the mobile device instead of having to use the keypad which is linked to the intrusion system via wiring. Snehal and Jadhav (2010) stated that this is obsolete technology and hence there is need to enhance and personalize such systems. The researcher realized that since the android operating system is most popular mobile platform there is need to create an android application which will link to the system either via Bluetooth, GSM technology and even WIFI. The need to include greater remote control of intrusion systems gave rise to the creation of this study. This lack of personalization on current obsolete existing systems has driven the researcher to include more sophisticating features to the current existing systems. The aim of the research is to incorporate personalization to an intrusion system with a goal to include more control to the end-user, give notification to not one but at least three emergency response teams at once and enable remote control of the system without requiring use of keypads.

It was identified that in the existing obsolete intrusion systems putting a wrong a pin in the system a lot of times the system would not act on it rather the system would just continue buzzing and this not smart technology hence there was need to look into that and improve the system. The working concept would have an ultrasonic sensor facing the outside and a PIR sensor facing the inside of the building. The ultrasonic sensor detects intrusion using sound and PIR using infrared heat signatures. When the device is started it gives the intruder twenty seconds to move from the vicinity of sensors after that the alarm is activated when an intrusion is detected. This will trigger the buzzer to start ringing and this will not stop until the user inputs the password for it to stop. Failure to input the correct pin after three attempts would result in the intrusion system automatically sending a notification to the listed emergency response teams. The GSM/Bluetooth module will trigger sending of an SMS to the owner's mobile phone whereby there would be an android application that communicates to the system via wireless Bluetooth technology.

1.3 Problem Definition

According to Davidson (2001), problem definition is the identification of varying inhibiting factors that force the development of solutions. The solution is to create an intrusion system to hint/alert about the entry of an intruder. Using different sensors which include the pyro-electric infrared motion sensor and the ultrasonic sensor the approach of an entity within the calibrated distances of the sensors would trigger emergency response. The detection or sensitivity of the sensor would be adjusted depending on the preferences of the owner of the premises so as to accommodate different varying intrusion ranges. The current intrusion systems are operating in a rather obsolete fashion whereby there is use of simple sensors which ring when they detect motion. They are ineffective as they do not give a notification to the owners as well as the emergency response team. Hence in order to curb that the study further improved existing systems with the use of a panic button and android application which give a greater amount of personalization which the core issue.

1.4 Aim

This is defined as the benchmark that is set to be attained. An Arduino based infrared intrusion detection system and motion sensor device based on incorporating the concept of infrared heat sensing. In this device a passive infra-red sensor which measures change in heat energy of the incoming object will be used as the sensing element or detector.

1.5 Objectives Formulation

Objectives are set thresholds which are to be met/achieved. The passive infrared intrusion detection and motion sensor device should have the following:

- To have the sound/heat signatures developed by an approaching entity automatically picked up by the ultrasonic sensor when in vicinity
- To allow immediate automatic triggering of the active piezo buzzer when an intrusion is detected
- To have the triggered alarm switched off wirelessly via the android application or via the use of the membrane keypad after it had been triggered
- To allow the user to contact and notify at least three emergency response teams at once via the android application
- To have a panic button in situations whereby intrusion occurs whilst inside the premises the pressing of the distress button should trigger sending of an SMS to the listed emergency response teams

1.6 Instruments and methods

Methods and instruments are essentially the tools which are to be used in the development of the system. These include the Integrated Development Environment (IDE), which is the factory for the source code, Arduino components for example the micro-controller as well as various supporting software's.

1.7 Software

- **MIT App Inventor**-This is the Integrated development environment that would be used to write the code and design the user interface for the android application. It enables the creation of simple android applications for those with little or no background in the android language.

- **Arduino Genuino** -This software package is the factory where the source code for the system components resides. It shall be used to write the C++ language which will then be compiled and uploaded into the Mega2560 micro-controller.

1.8 Programming Languages

- C++/C programming language

- Android language

1.9 Arduino hardware components

The following is a rundown of the components to be used in the creation of the intrusion detection device:

- 4.7k Potential meter
- LCD display
- Mega 2560 (Arduino)
- Breadboard
- Breadboard jumper wires
- Male to Male Headers
- Active Buzzer Alarm Module
- Ultrasonic sensor

1.10 Proposed Budget

According to Davidson (2009), a budget is an ordered list of units and their accompanying costs which are used to estimate the likely total expenditure to be incurred. The proposed budget is a list of the items that are required for the system to be developed. The system to be developed is hardware biased and it consists of the following units as shown below.

Table 1.1 Proposed budget

Item	**Amount**
Ultrasonic sensor	US$ 9.00
Arduino Mega2560	US$ 30.00
4x4 Keypad membrane	US$ 2.00
Pyro electric infrared sensor	US$ 7.00
20x2 LCD display	US$ 10.00
Breadboard and power supply	US$ 10.00
Mini breadboard	US$ 5.00
Light Emitting Diode	US$ 1.00
Active Piezo Buzzer	US$ 3.00
Potentiometer	US$ 2.00
Total	**US$ 79.00**

1.11 Time Plan

According to Dawson (2012), a time plan is a sequenced table or diagram which shows each activity to be done and its corresponding timeframe. The activities include introduction, literature review, methodology, results discussion, coding as well as user interface design. The table below shows the time plan and length of activities.

Table 1.2 Time Plan

	week 1	week 2	week 3	week 4	week 5	week 6
Introduction						
Literature review						
Methodology						
Results and Discussion						
Summary and Conclusion						

1.12 Justification and rationale

Justification for the creation of this system rises from the mere fact that current existing intrusion systems lack personalization features thus rendering them redundant and obsolete. To improve them the research has highlighted the need for the introduction of a panic button and an android application which would enables remote access and control of the whole device. This personalization eventually leads to a robust modern intrusion system which gives greater emphasis on control. Comparison with early systems shows that they just act on what they detect and they do not offer other options to the detected entity. Furthermore, this give the researcher an opportunity to link theoretical aspects of programming with hardware for use in real-time.

1.13 Conclusion

This chapter focused on the introduction of the current capabilities of existing intrusion systems. It highlighted their shortcomings as well as the tweaks and features to be introduced in order to improve them. The main goal highlighted is the personalization of such existing systems. To sum up the passive infrared intrusion detection and motion sensor device will finally allow for the integration hardware and software elements. This system has the potential for commercial viability as intrusion detection systems are critical at a time where possible physical security breaches are very high these days.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

The chapter gives comparisons and correlations of systems already existing that are similar to the one to be developed. The chapter will focus on the literature review encompassing related work analysis and the gaps identified. Attempts will be made to include modifications or improvements in current systems so as to come up with a radically different approach that will make better systems.

2.2 Related Work Analysis

According to Davidson (2014), related work analysis is a study that is done to compare and contrast earlier made versions of a system that is to be developed. It includes the study of the system's components, schematic connections as well as the similarity in functionality. Attempts shall be made to unearth gaps and provide solutions to these identified gaps so as to create a robust system that has new features.

2.3 UART Based Detection System

Narayanan, Khadke and Megalingam (2015) proposed an intrusion system that utilized environment sensors. These sensor nodes consisted of an active piezo buzzer, a presence detecting circuit as well as a break-in camera. A Universal Asynchronous Receiver/Transmitter (UART) was used as the communication link between the end-user's device and the hardware. It consisted of a Graphical User Interface (GUI) which enabled the option of capturing a photograph of the intruder as well as an option of sending a distress alarm.

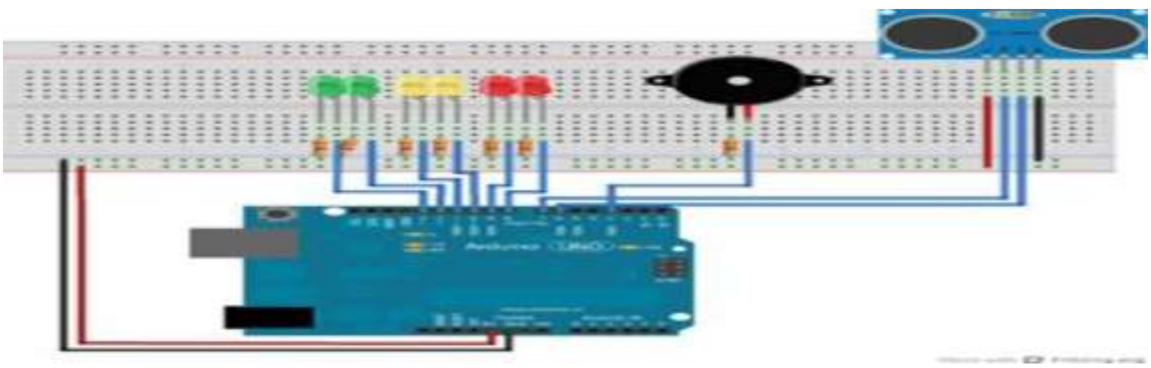

Fig 2.0 Universal Asynchronous Receiver Detection System

Source: Researchgate.inc (2019)

2.4 Smart Home System Using ZIGBEE

Palaniappan and Chuah (2015) proposed a smart home system using ZIGBEE as the wireless network to link all the components. This type of network technology is based on the IEEE 802.15.4 low rate wireless Personal Area Network standard. To enable interfacing with the multiple devices for the system a PIC16F877A controller was applied as it is relatively cheap and can easily be re-programmed. The control of the devices was achieved via the use of the PIC controller and it also sent sensor values to the computer via a ZIGBEE module. ZIGBEE was used because of its lower power consumption as compared to Bluetooth which is also relatively faster at data rates. Thus because of lower power consumption twenty-four hours of monitoring were achieved using very small amounts of power. At the personal computer terminal LABVIEW was used to control the GSM and ZIGBEE operations. When activated the system detects fire and theft scenarios resulting in turning of the active piezo buzzer and sending an emergency SMS via the GSM module. It is also able to save energy by switching off lights when the room is unoccupied.

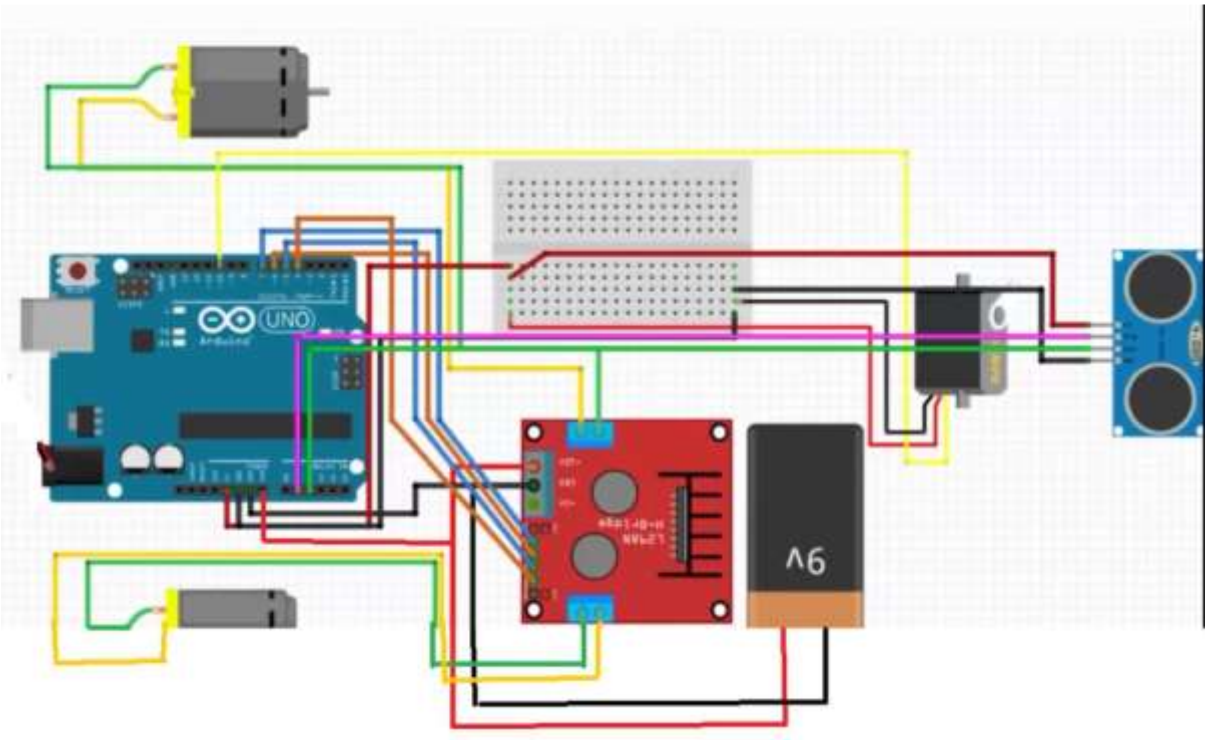

Fig 2.1 Smart Home System Using ZIGBEE

Source: Researchgate.inc (2019)

2.5 Remote Laboratory Monitoring System

Palaniappan (2015) proposed an embedded intrusion system which was capable of monitoring a laboratory environment remotely. Remote monitoring of a laboratory is necessary as it allows the laboratory owners to know the environment status of the laboratory. Receptive sensors which are able to pick up elements such as fire, leaking gas and smoke are installed and these then link to a micro-controller. The micro-controller is responsible for triggering a buzzer in situations whereby either of those listed elements are sensed. Using a GSM module, a voice alarm and an alert message are sent to the remote area which then requires a response team to survey the situation.

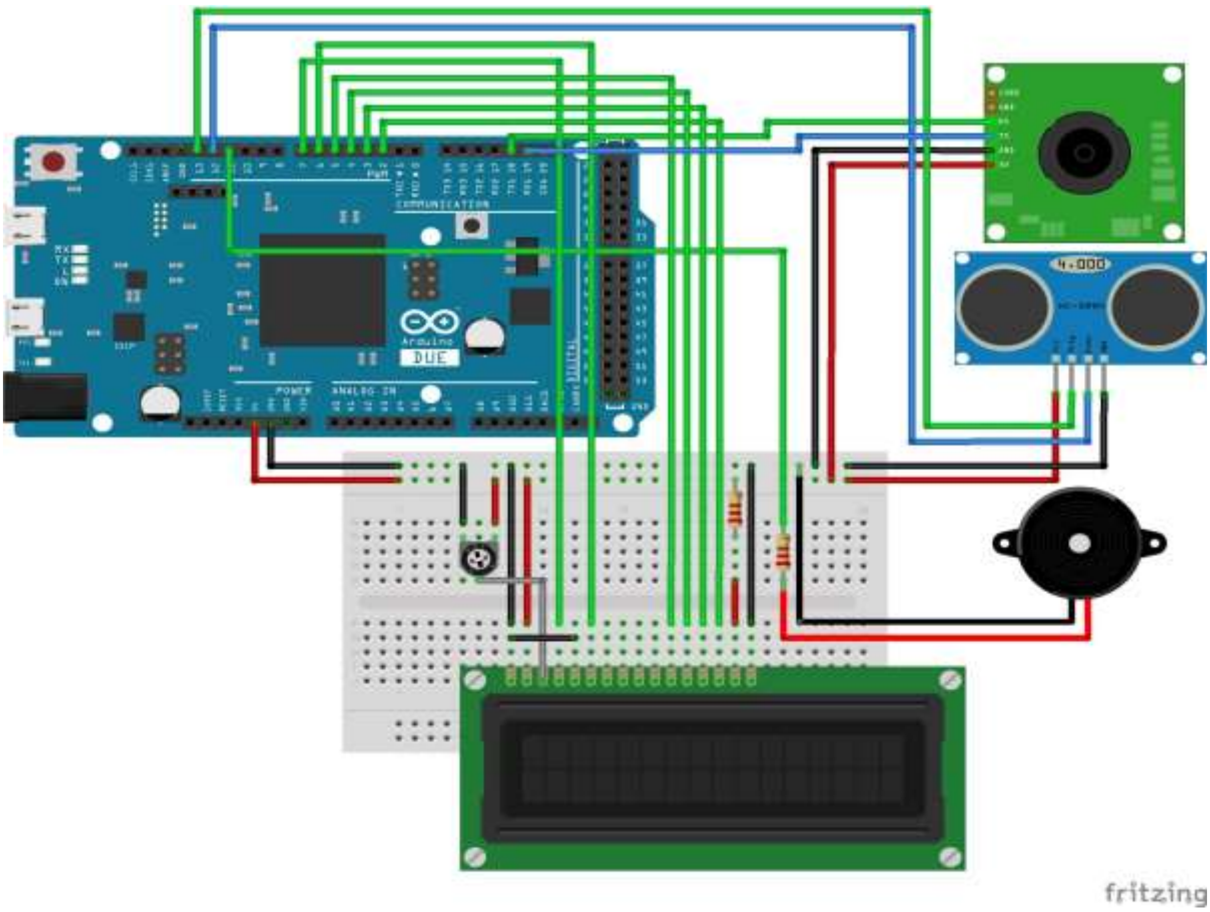

Fig 2.2 Remote Laboratory Monitoring System

Source: Researchgate.inc (2019)

2.6 Low Power Intelligent, Wireless and Home Security System (IWHSS)

Megalingam et al. (2015) proposed an intrusion detection system that used a low amount of power input which was termed Low Power Intelligent, Wireless and Home Security System(IWHSS) dedicated to elders confined within their homes. The IWHSS services can be conveniently availed within the room in the event that a stranger is attempting to enter the room. The intrusion system is capable of identifying the stranger and it can also enable informing of the police once activated. Furthermore, an emergency alarm activates so as to inform neighbours of a possible emergency situation. It includes features which enable the elders to start a conversation with the strangers as well as a platform to see them. The PIC 16F877A micro-controller is interfaced directly with a GPRS Modem hence the low power requirement making it efficient. Furthermore, the system includes the services of the Image Capture module, Radio Frequency modules as well as the IR-TSOP modules. Simulation of the system software was achieved using PROTEUS and MPLAB IDE.

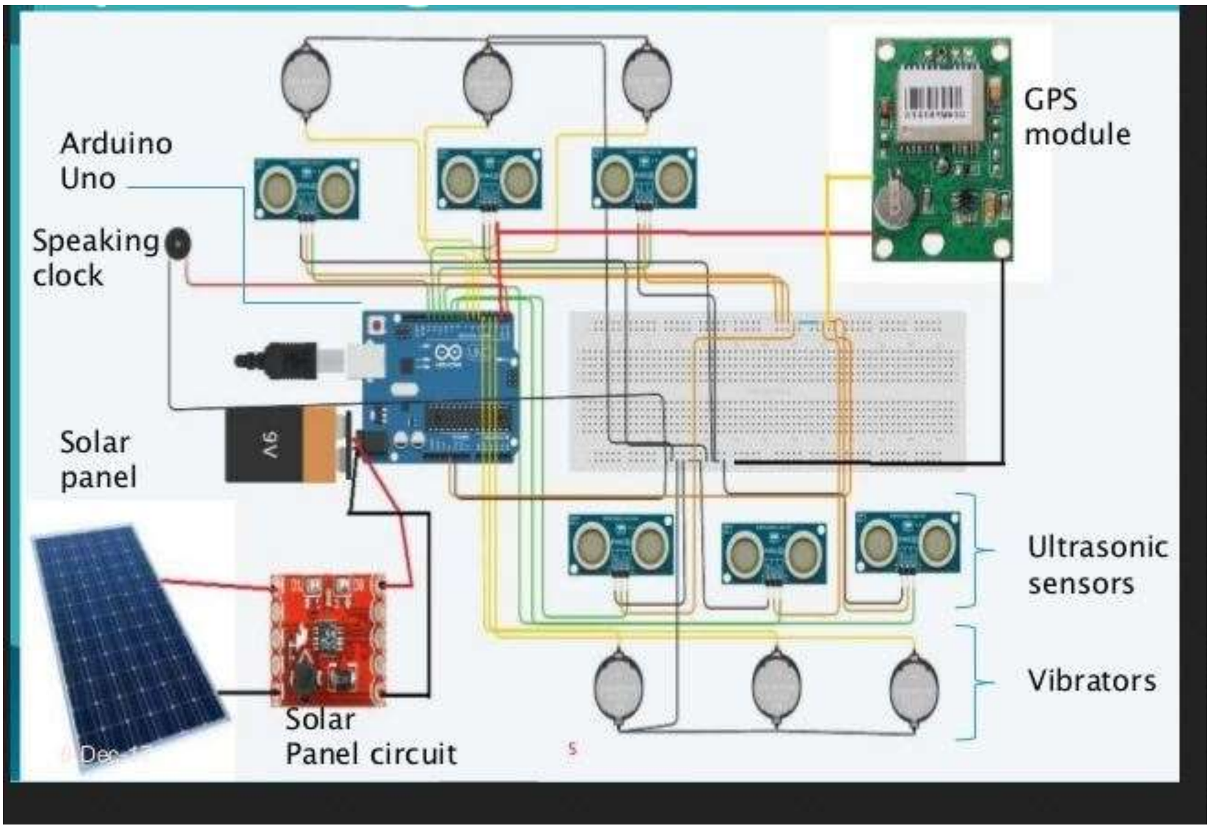

Fig 2.3 Low Power Intelligent Wireless and Home Security System

Source: Researchgate.inc (2019)

Chuah et al. (2015) mentioned intrusions to assets has primarily been based on having sensors which activate the moment an entity approaches the sensor's detecting distance. Simple sensors such as the ultrasonic sensor would be placed in a single direction whereby an intruder is expected to approach from . These obsolete inefficient intrusion systems are not effective as they do not offer additional features to help protect assets. Existing and early intrusion systems included the use of simple current breaking switches to stop the circuit from being complete. There had to have an emergency response team to hear the activated alarm. This is not necessary in the proposed system as notifications are sent to alert them. The device that is being proposed is fundamentally similar to the obsolete intrusion system. The functions are almost similar in that there are motion sensors but the one being proposed offers quite a lot more tweaks that opens a whole new level of security. New features or additions to the obsolete system include the use of a Bluetooth module to send warning texts to the emergency response team or neighbours. In addition, there is the use of an android application which incorporates greater remote control using wireless Bluetooth technology. To further improve the device there has been the addition of a panic button which when pressed would immediately contact the police or emergency response team.

2.7 Evaluation of Current Existing Systems

The previous intrusion detection system would for the most part be based on the use of simple movement recognition sensors. An alarm would be activated immediately when an intrusion would have been detected. This represents a vast majority of simple entry warning systems that includes the basic sensors and a loud warning sound. Depending on the end-user's budget and needs security companies offer intercoms, automated gates, CCTV systems and heat signature cameras. Looking at the existing intrusion systems they satisfy the minimum thresholds of detect and alert but this is minimalistic and it does not provide strong security. Furthermore, this device possibly sounds a caution if any movement has been detected inside yet it doesn't give the appropriate response concerning whom from this interruption was coming from. Such a system might be activated obviously by criminals generally but then again, youngsters or other relatives or by their pet or some arbitrary nuisance. This may then ensnare that it makes pointless false alerts as it is not an intelligent system. Thus it is not as successful as it very well may be for a security system as there are quite a lot of features missing for greater control and assurance of security.

2.8 Gaps Identified

According to Yates (2006), identifying gaps involves looking at the deficiencies that a system has and includes solutions that should be included so as to satisfy evolving demands. After a thorough analysis was carried out it was identified that there are various setbacks of most current intrusion systems in place. The gaps that the study identified include:

- Existing systems are only able to alert a single emergency response team
- Existing systems can only be controlled via a fixed keypad
- No remote control of system e.g. via an android mobile phone
- Existing systems are not able to distinguish between humans and pets
- The system cannot protect itself from being damaged

2.9 Proposed Intrusion Detection System

Dawson (2001) stated that a proposal is a description of identified solutions that are ought to be included so as to come up with a robust holistic system. The device that is being proposed is fundamentally similar in terms of hardware to the obsolete intrusion system but there are now more additions and tweaks to the user interface as well as how the system communicates wirelessly with mobile devices. The functions are almost similar in that there are motion sensors but the one being proposed offers quite a lot more tweaks that opens a whole new level of security.

New features or additions to the obsolete system include the use of a Bluetooth module to send warning texts to the emergency response team or neighbours. In addition, there is the use of an android application which incorporates greater remote control using wireless Bluetooth technology. To further improve the device there has been the addition of a panic button which when pressed would immediately contact the police or emergency response team.

2.10 Benefits of proposed system

- It gives a more elevated level of security as it uses two different types of sensors one which sensors heat and another which sensors sound.
- Greater control of system achieved via linking with an android application
- Enables to contact multiple emergency response teams at once
- Applies Internet of Things specifications via the use of Bluetooth to control the system

2.11 Drawbacks of Proposed System

- Costlier to create as it requires more hardware
- Higher number of components somewhat means a more expensive end product for the client.
- In the event that a response team is absent it therefore entails that no one can act on the warning.
- Arduino is open source thus implying that anyone with requisite knowledge can manipulate the system

2.12 Foreknowledge on Project

To start working with Arduino components one needs to have a background in physics and electric circuits. Furthermore, knowledge of the C/C++ programming language is essential as it is the language that is understood by Arduino components. The making of Arduino gadgets is based on the open source environment. It allows one to create powerful IOT devices using simple code. For one to start working with Arduino components experience with working with development environments is essential. The Arduino IDE is relatively similar to the other IDE for languages such as Java, C++, PHP etc. To enable the components to work one requires the basic declaration of variables of the components. Simulation software can be used to determine whether the connected components can communicate and work successfully. The intrusion detection device also requires use of android which will be used to create the application that will communicate with the system via wireless Bluetooth technology.

Using the knowledge acquired in the modules Electricity and magnetism, Electric Circuits the setting up of the circuitry is now possible otherwise it would have been an inconceivable task. The modules introduced me to physics as well as basic knowledge of electric circuits. To gain a thorough understanding in the creation of this device one requires background knowledge in the different types of sensors that are in use in the Arduino world. It is hoped that commercial viability will be seen by established security companies as this would be a home grown solution.

2.13 Conclusion

In this chapter a correlation study was done with existing systems. To justify the viability of this new system a great number of modifications or additions were made so as to achieve greater control and efficiency of the current existing systems. Greater control of the device has been exhibited which is a requisite requirement. Comparisons were made and it has been exhibited that the new modifications or additions would give greater efficiency, control, security and assurance for the end-user.

CHAPTER 3: ANALYSIS PHASE

3.1 Introduction

The chapter gives a description and focus on the components to be utilized in the creation of the device. It includes a description of each of the components as well as their functionality which is of fundamental importance to the working of the system. The Arduino Mega2560, pyro-electric infrared motion sensor, ultrasonic sensor, Membrane Keypad, Breadboard, Jumper Wires, Bluetooth module, ultrasonic sensor among other peripherals.

3.2 Hardware components

Johnson (2014) defines hardware as any physical entity that is to be used in the creation of the system. It is basically a component that is to perform a function or link with other circuit components so that voltage can move throughout the system. The intrusion detection system is to be built using Arduino components as well as basic electronic circuit components. Knowledge of circuit connections and specifications is essential for the researcher to be able to come up with a functioning system.

3.3 Arduino Mega 2560

The Arduino Mega 2560 is a micro-controller which is the brain of the whole system. A diagram of the component is shown below. It controls all the input processes it and gives relevant output. It consists of analogue and digital input and output pins. The digital output and input pins are 54. There are also 14 PWM outputs. On the analogue inputs they are 16 whilst 4 are hardware serial ports.

Figure 3.1: Arduino Mega 2560

Source: Arduino.inc (2019)

On the top part is a reset button as well as a 16MHz crystal oscillator. Further, there is a supporting power jack and an alternative USB connection port. For the microcontroller to power on one can use an adapter or just plug in to a USB computer port.

3.4 Arduino Mega 2560 Specifications

- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limit) 6-20V
- Digital I/O Pins 54 (of which 15 provide PWM output)
- Analog Input Pins 16
- DC Current per I/O Pin 20 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 256 KB of which 8 KB used by bootloader

3.5 Liquid Crystal Display Module

The Liquid Crystal Display is essentially the output device for the user to view. The LCD is applied in a wide variety of circuits to act as a display device. The type to be used is a 16*2 in terms of screen size so it implies that it is capable of exhibiting 16 characters in a series of 2 lines one on the top and the other below. A diagram of the component is shown below.

Figure 3.2: Liquid Crystal Display Module

Source: Arduino.inc (2019)

The pixel matrix for this LCD is 5*7. For it to work in data transfer it uses the data register and the command register. When an input command is executed all command instructions get stored by the command register. Examples of command registers include setting the contrast, adjusting character display size etc. In addition, the LCD is capable of storing data through the use of the data registers. All characters are displayed in the ASCII notation.

3.6 Liquid Crystal Display Specifications

- Operating Voltage is 4.7V to 5.3V
- Current consumption is 1mA without backlight
- Alphanumeric LCD display module, meaning can display alphabets and numbers
- Consists of two rows and each row can print 16 characters.
- Each character is built by a 5×8 pixel box
- Can work on both 8-bit and 4-bit mode
- It can also display any custom generated characters
- Available in Green and Blue Backlight

3.7 Bluetooth Module

The HC-05 is a class one master slave Bluetooth module designed for full duplex communication between Bluetooth enabled devices. It uses serial input which is then transmitted over the air. A diagram of the component is shown below.

Figure 3.3: Bluetooth Module

Source: Arduino.inc (2019)

The wireless data that would have been received is converted by the serial interface module which then allows it to be in the form of binary form. It accepts all code which is not specific to the different microcontrollers available. Due to the fact that it requires logic level power supply it is recommended to apply a logic level converter before use on the circuit board.

3.8 Bluetooth Module Specifications

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK (Gaussian Frequency Shift Keying)
- Emission power: ≤4dBm, Class 2
- Sensitivity: ≤-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA

3.9 Breadboard

Among the other components to be used in the setting up of the intrusion system is the breadboard which is used to make temporary connections during the circuit build up process. It allows power to be distributed and as well regulate the amount each component receives.

Figure 3.4: Breadboard

Source: Arduino.inc (2019)

The breadboard consists of terminals which are basically holes which allow the circuit builder to make positive and negative connections depending on the circuit setup. Below the holes of the breadboard is a series of parallel metal strips which allow current/charge to move throughout the circuit.

3.10 Breadboard Specifications

- 2.2" x 3.4" (5.5 cm x 8.5 cm)
- 9.7mm(0.38in) thick, including sticky foam on the bottom
- Weight: 38.9g(1.27oz)

3.11 Active Piezo Buzzer

The component that is to be used to produce a warning sound is the buzzer. This component works on the principles of disturbances caused by the motions of the piezo crystals. These piezo crystal obey the principles of the piezoelectric effect. A diagram of the component is shown below.

Figure 3.5: Active Piezo Buzzer

Source: Arduino.inc (2019)

They work by using the variations in pressure caused by the distribution of electric potential in them. There are many various applications of the piezo buzzer and they can be applied in sensor input or a switching action as well as the identification of a counter signal. When a circuit is complete regardless of the amount of current supplied the action of the piezo buzzer is to produce a sound. What results in a sound is the action of the two piezo crystals caused by the movement of one of the crystals to the other. The range of the piezo buzzer is 1 to 4 KHz.

3.12 Active Piezo Buzzer Specifications

- Rated Voltage 5V
- Operating Voltage 4-8V
- Rated Current (max) mA ≤ 32 Min.
- Sound Output at 10cm dB ≤85
- Coil Resistance 16Ω ±4
- Resonant Frequency Hz 2300± 300
- Operating Temperature deg C -20~+45

3.13 Membrane keypad

Input and selection of options displayed on the liquid crystal display a membrane keypad is to be used. It consists of four rows and four columns with a selection of numbers that range from zero to nine. Letters range from A to D. A diagram of the component is shown below.

Figure 3.6: Membrane keypad

Source: Arduino.inc (2019)

It works on the fact that pressure which is applied to the surface of a button links to a corresponding number or character input. The membrane keypad looks like any other regular keypad with the exception that it is very thin and works on the principle of pressure applied on a flat surface.

3.14 Membrane Keypad Specifications

- Maximum Rating: 24 VDC, 30 mA
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions: Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)

3.15 Ultrasonic Sensor

The detection of entities is to be achieved using the ultrasonic sensor. It functions on the principle of sound waves which bounce back to the origin. A diagram of the component is shown below.

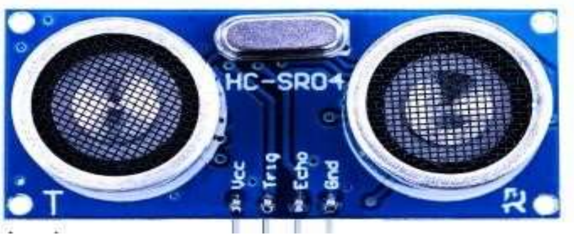

Figure 3.7: Ultrasonic Sensor

Source: Arduino.inc (2019)

For the ultrasonic sensor to work it emits a sound wave at frequencies which have a higher probability of enabling bounce back thus this is how it measures distance. When the bounced back wave reaches the ultrasonic sensor it measures the amount of time taken hence giving a figure of the distance of the entity from where the ultrasonic sensor is positioned.

3.16 Ultrasonic Sensor Specifications

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

3.19 Light Emitting Diode

This component is to be used to indicate that the panic button has been pressed as well as show that the system is connected to a power source. When the panic button is pressed it gives off a red colour. A diagram of the component is shown below.

Figure 3.9: Light Emitting Diode

Source: Arduino.inc (2019)

It works by combining the colour hues coming from the three separate wires connected together. These wires have different frequency ranges meaning they give off different colours in the microwave spectrum when exposed to light/current.

3.20 LED Specifications

- 5mm Round Standard Directivity
- UV Resistant Eproxy
- Forward Current (IF): 30mA
- Forward Voltage (VF): 1.8V to 2.4V
- Reverse Voltage: 5V
- Operating Temperature: -30℃ to +85℃
- Storage Temperature: -40℃ to +100℃
- Luminous Intensity: 20mcd

3.21 Jumper Wires

For current to move throughout the system jumper wires are required. These are simply electric conductors that allow the linking of the different components and for current to move through. A diagram of the component is shown below.

Figure 3.10: Jumper Wires

Source: Arduino.inc (2019)

They are either inserted, clipped or magnetically linked with the other components. They are quick to dismantle hence they are appropriate with breadboards. The colour coding used in jumper wires is not important as it is used to just educe confusion when wiring a circuit board.

3.22 Circuit Design

The intrusion detection system has been designed in such a way that it has to be activated by choosing the activation option. In this state neither of the sensors will be active. The liquid crystal display however would still remain open. The diagram to be seen below is an illustration of how the components are to be connected. It shows the connections as well as their positions.

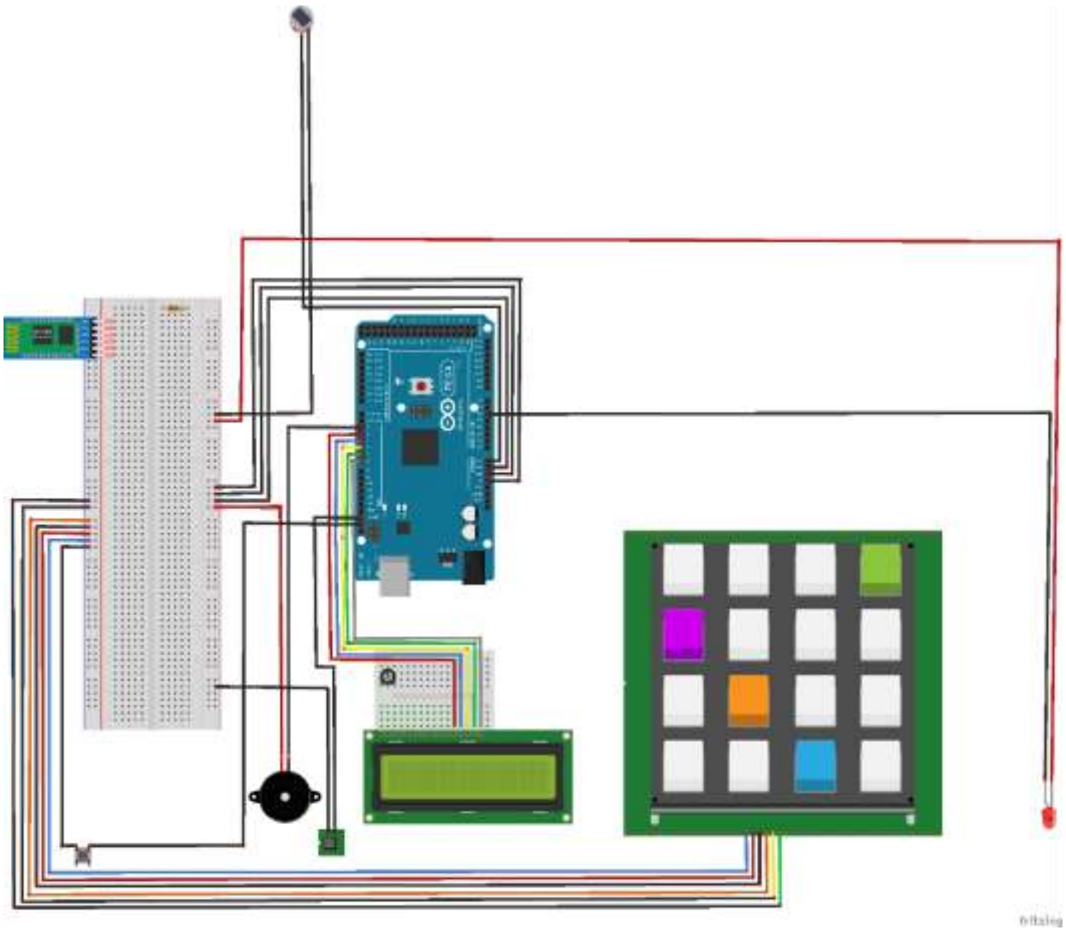

Figure 3.11 Diagram of circuit

3.23 Conclusion

This chapter focused on giving a thorough description of the components to be used as well as their operating specifications. It has provided pictorial views of the components as well as detailed explanations of their operating environments, voltage inputs, working ranges as well as their use specific to the system.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Introduction

The Arduino-based intrusion detection system has had a thorough system review in the previous chapters and what is left is to implement it in different varying practical situations. There are multiple ways to test the system and there exist a different range of tools and options available so as to illustrate the internal workings of the system. This simulation and implementation chapter will focus on the logic design, programming logic, as well as the syntactical errors that would have occurred during the development of the system.

4.2 Implementation

According to Pressman (2015), implementation is the actual migration of the system from the theoretical phase to actual practical system use in real-time. The intrusion detection system is modelled in such a way that it takes input from the pyro-electric infrared motion sensor, ultrasonic sensor, android application as well as the keypad. Once the activation button is pressed it can either be de-activated via the input of a correct pin or through the use of an android application. The pyro-electric infrared motion sensor faces outside whilst the ultrasonic sensor faces inside. In actual implementation the system requires two input keypads one for the outside guest and the other for the occupants inside the room. Furthermore, more sensors are required to cover three hundred and sixty degrees of the whole room for maximum security.

The implementation of the intrusion detector is based on having the Arduino Mega2560 micro-processor as the central brain for all the components. It is the controller of the sensors, LCD, membrane keypad and other components. Working cohesively in a real-time environment it takes data from its environs and displays it on the LCD. Depending on the asset to be protected the actual system is supposed to have two keypads to activate the system from both inside and the outside. In addition, there should be three hundred and sixty-degree geographical coverage of the asset to be monitored.

4.3 Testing

As stated by Richards (2015) testing is the subjection of a recently compiled system to check whether it is performing as coded. The system went through a series of tests to see if the objectives were met and if the highlighted modifications were included. Furthermore, the system will be validated to allow proper input from end-user.

4.4 Testing types

Shown below are the various testing types that can be applied to the intrusion system so as to achieve end-user requests.

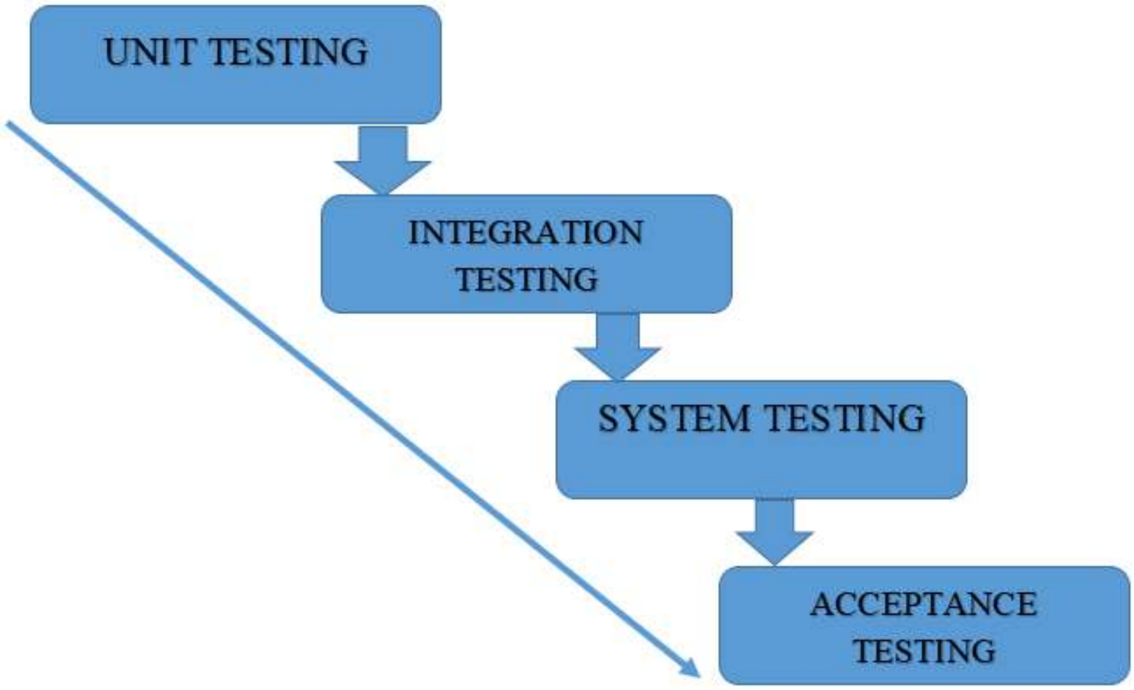

Fig 4.1 Testing types

Since the intrusion detection system is composed of hardware and software elements various testing methods can be applied to the hardware as well as the software in use. The sensors which are controlled by the Arduino Mega2560 micro-controller require real-time testing with human entities whilst the android application requires white box testing to check if each line of code is performing according to the required specifications.

4.5 Unit testing

This type of test checks to see if each unit of code is performing the function for which it was written to do. It includes various elements such as validation to see if end-user input is valid. White box testing was carried out on the android application saving the details of the various emergency response teams so as to achieve valid user input.

4.6 Integration Testing

In this type of testing multiple units are combined together and subjected to tests so as to see if can function cohesively (Leonard and Bryan, 2015). End-users and coders are active in this type of testing. All the components were connected together via a Bluetooth connection using the HC-05 Bluetooth module. A diagram is shown below.

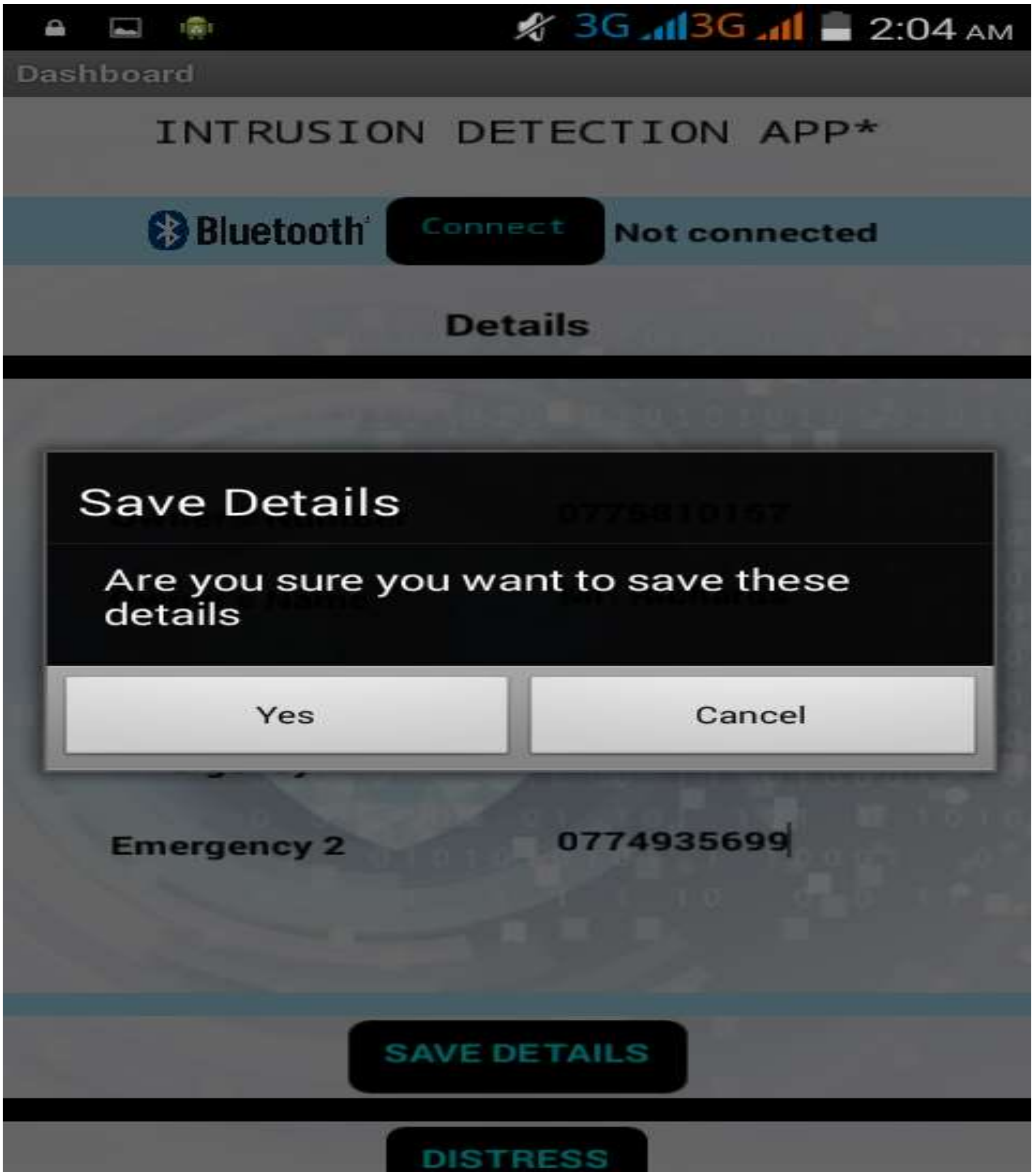

Fig 4.2 Save details notification

4.7 System testing

Hector (2016) stated that system testing is the bundling up of the whole system individual sub units so as to see the integrity as well as the efficiency of the system. Black box as well as white box testing methods are subjected to the system. After thorough system tests it was identified that the system performed well and was error free. A diagram is shown below

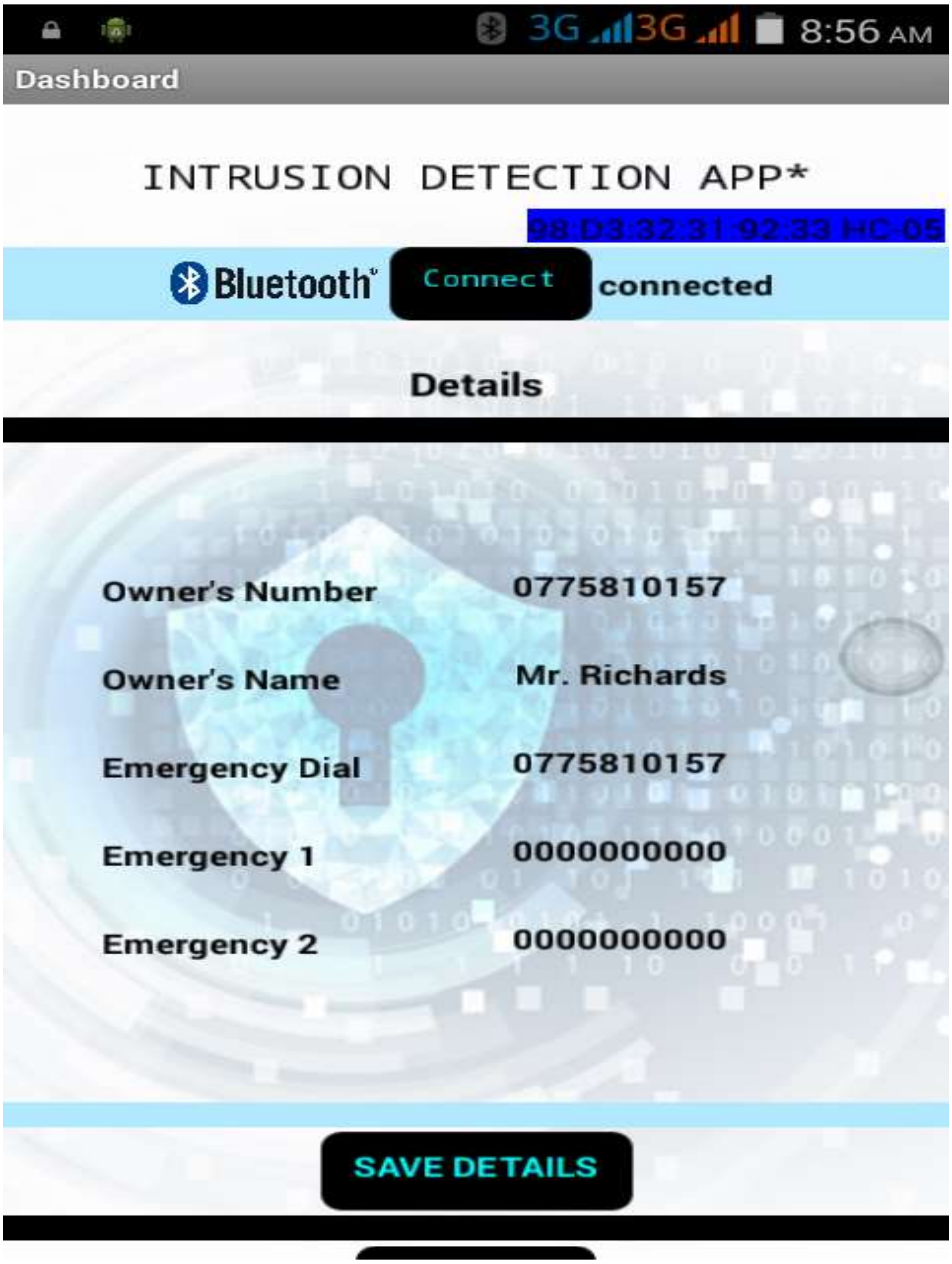

Fig 4.3 Bluetooth Connected Notification

4.8 Verification and validation

According to Johns (1992), verification is a process undertaken to check whether the recently completed system is functioning as coded and that units of the program are working cohesively. Validation on the other hand is a process of accepting valid end-user input as the system restricts the end user to enter a certain type of value for example being restricted to enter an integer value only. A diagram is shown below.

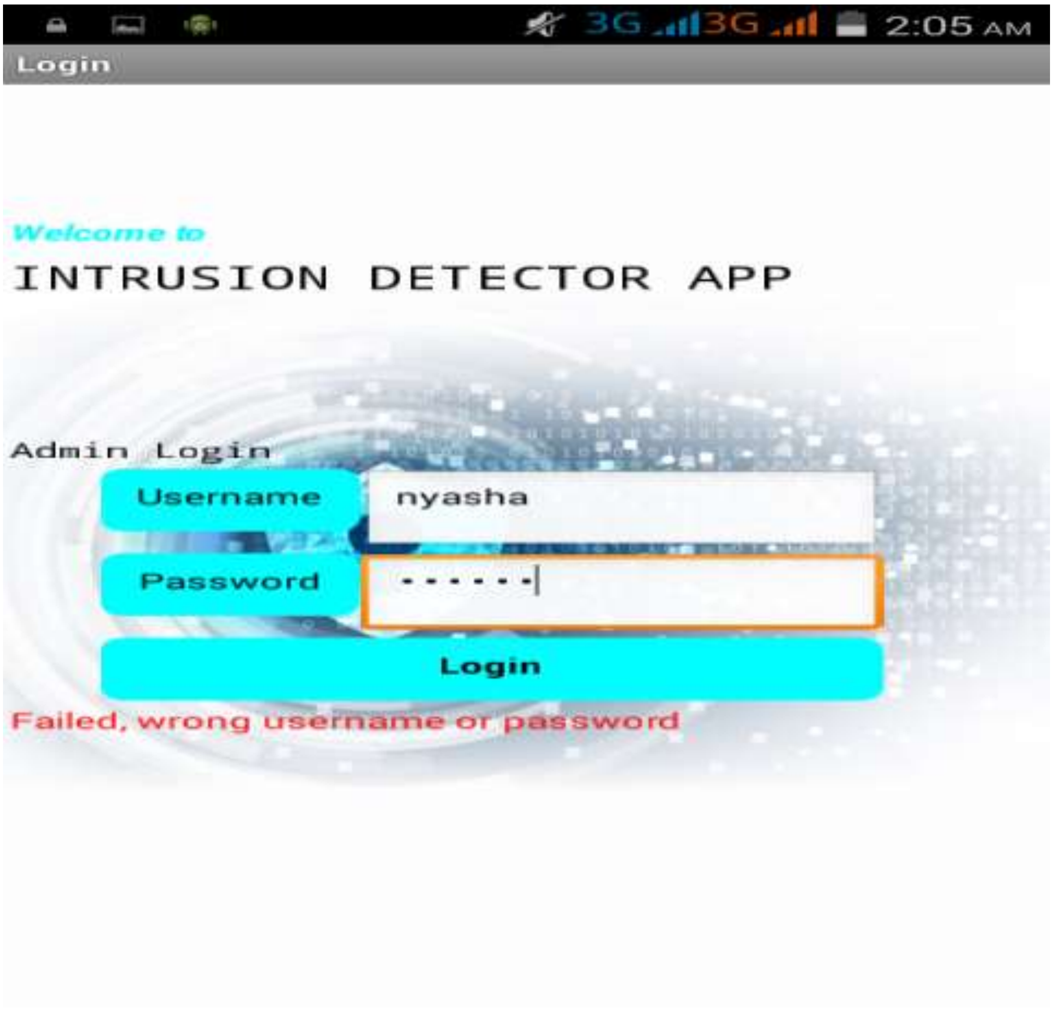

Fig 4.4 Failed Login Attempt Form

On a successful login attempt the form displayed below would then appear. The form has input fields for the owner's number, owner's name, first emergency dial as well the two other emergency dials. In addition, it consists of a save details button as well as the text field which enables the owner to input the master pin to turn off the alarm wirelessly via Bluetooth.

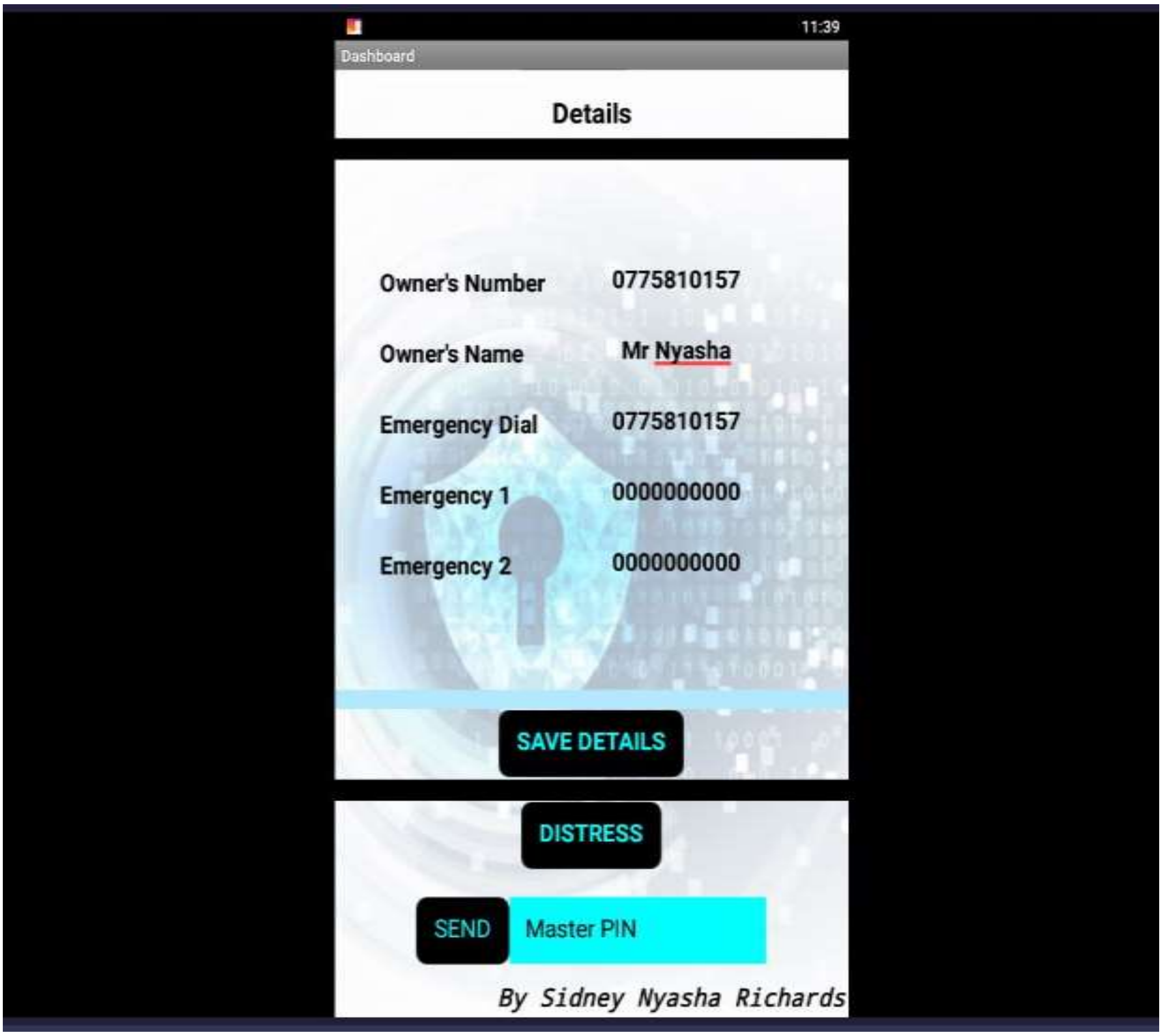

Fig 4.5 Android application user interface

Once the end-user inputs his/her details as well as those of the emergency response teams and saves them. The clicking of the distress button triggers the sending of SMS's to three separate teams. Furthermore, in case the alarm is activated it can be switched off wirelessly via inputting the master pin.

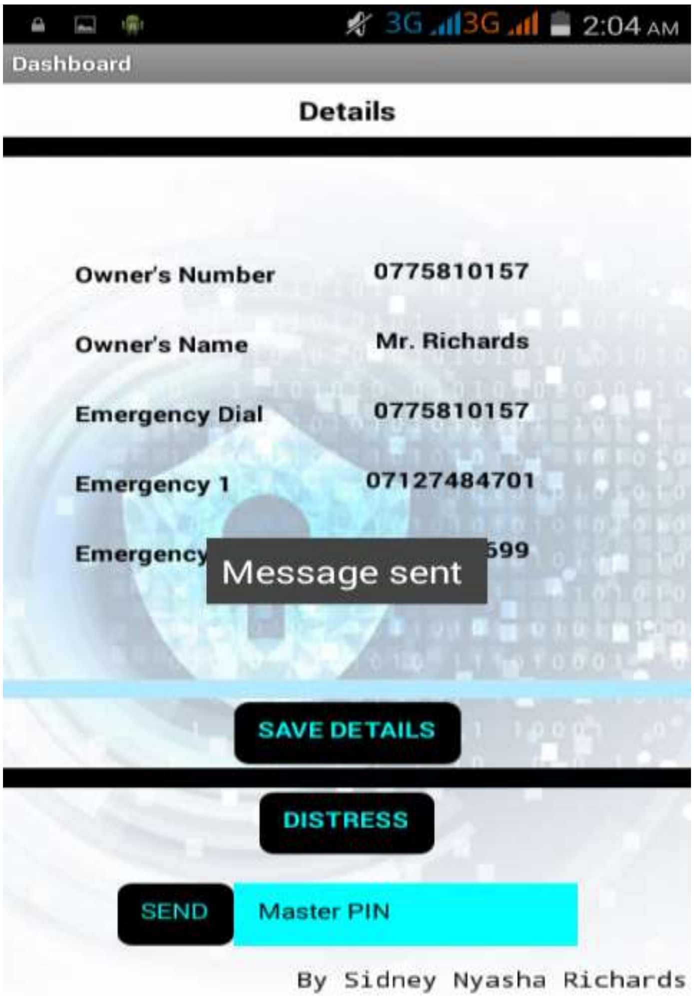

Fig 4.6 Message Sent Notification

4.9 Acceptance testing

Acceptance testing involves giving the system to the client the end-users of the system. These included home owners room guards so that they could check if there were any errors. This type of testing requires no background knowledge of the code. This is the final stage before the system is rolled out to everyone.

The Infrared Intrusion Detection System and Motion Sensor device underwent various verification and validation steps. In these steps the following stated objectives in the first chapter were achieved as evidenced below:

- **Objective number one:** To have the sound/heat signatures developed by the approaching entity automatically detected by the ultrasonic sensor when in vicinity

- **Objective number two:** To automatically trigger the active piezo buzzer when an intrusion is detected.

Fig 4.7 Alarm activated notification

4.10 Explanation

The first and second objectives were met in the sense that an intrusion was detected thus it was displayed on the LCD with the word alarm via the triggering of the active piezo buzzer as well as the prompt to put a password via the word pass**.**

- **Objective number three:** To have the triggered/activated alarm turned off wirelessly via the android application or via the use of the membrane keypad after it had been triggered
- **Objective number four:** To have the ability to contact and notify at least three emergency response teams at once via the android application

Fig 4.8 Message sent notification

4.11 Explanation

These objectives have been achieved as shown on the android application that a message has been sent. If the distress button is clicked objective number four is achieved and if the master pin textbox if filled with the correct pin objective number three is achieved.

- **Objective number five:** To have a panic button in situations whereby intrusion occurs whilst inside the premises the pressing of the distress button should trigger sending of an SMS to the listed emergency response teams

4.12 Explanation

This objective has been achieved and it can be demonstrated practically by pressing the button. Once it is pressed this triggers the Bluetooth module to invoke message sending via the android application hence it is achieved.

Fig 4.9 Device Frontal View

4.13 System Flow Chart

According to Murdoch (2016), a system flow chart is a diagrammatic presentation of the flow of a system's functions. It includes the start, mid-point as well as endpoints of how the system operates. The diagram below shows a system flowchart for the intrusion detection system.

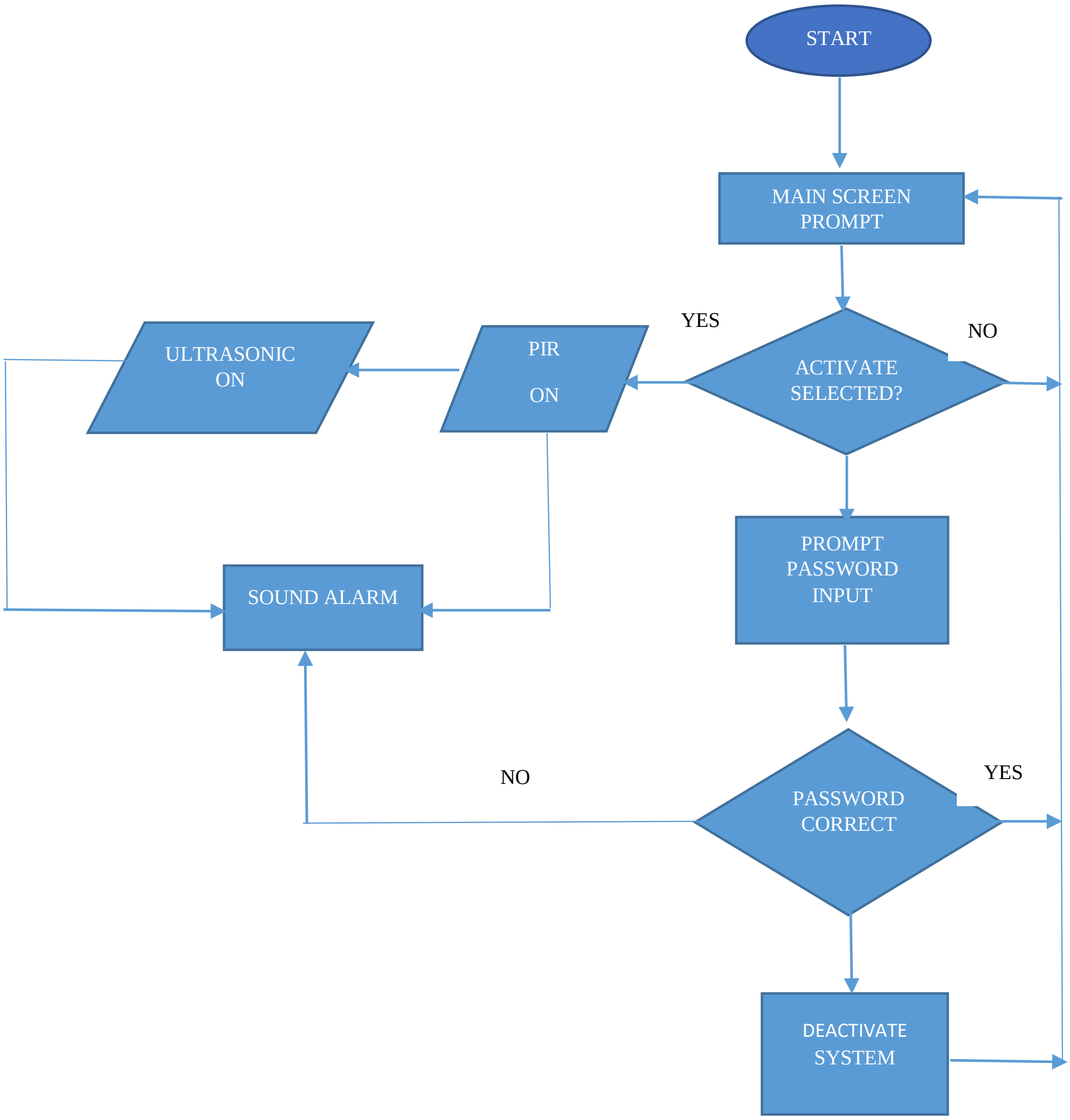

Fig 4.10 System Flow Chart

4.14 Code snippet of Membrane Keypad

Below is a diagram which shows the code snippet which shows how the characters of the membrane keypad are declared and set to enable it to function as expected. Furthermore, a detailed connection schematic is displayed in the diagram.

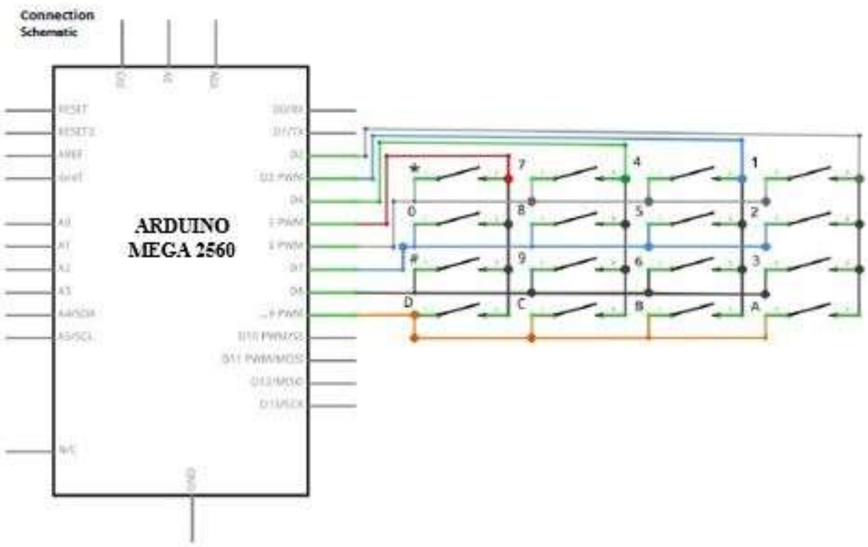

Fig 4.11 Membrane Keypad Connection Schematic (source: Arduino.inc)

4.15 Code snippet of Membrane Keypad

```
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keypressed;
//define the cymbols on the buttons of the keypads
char keyMap[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {21, 20, 19, 18};  //Row pinouts of the keypad
byte colPins[COLS] = {17, 16, 15, 14}; //Column pinouts of the keypad
Keypad myKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, ROWS, COLS);
```

4.16 Code snippet of active piezo buzzer

Below is a diagram which shows the code snippet which shows how the active piezo buzzer is set to enable it to function as expected. Furthermore, a detailed connection schematic is displayed in the diagram.

Fig 4.12 Active Piezo Buzzer Connection Schematic (source: Arduino.inc)

```
if (activateAlarm) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Alarm will be");
  lcd.setCursor(0, 1);
  lcd.print("activated in");

  int countdown = 10; // 10 seconds count down before activating the alarm
  while (countdown != 0) {
    lcd.setCursor(13, 1);
    lcd.print(countdown);
    countdown--;
    tone(buzzer, 700, 100);
    delay(1000);
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Alarm Activated!");
  initialDistance = getDistance();
  activateAlarm = false;
  alarmActivated = true;
}
```

4.17 Code snippet of ultrasonic sensor

Below is a diagram which shows the code snippet which shows how the ultrasonic sensor is set to enable it to function as expected. Furthermore, a detailed connection schematic is displayed in the diagram.

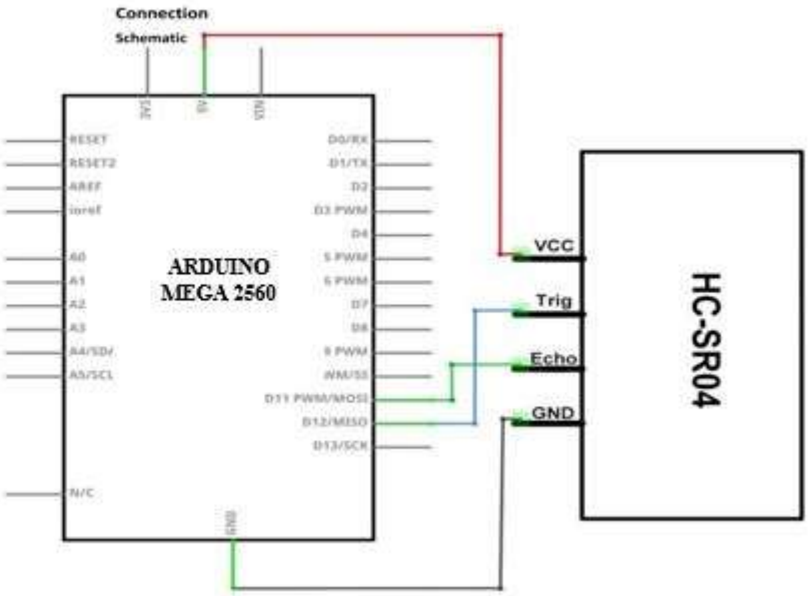

Fig 4.13 Ultrasonic Sensor Connection Schematic (source: Arduino.inc)

```
// Custom function for the Ultrasonic sensor
long getDistance() {
  //int i=10;

  //while( i<=10 ) {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;
  //sumDistance += distance;
  //}
  //int averageDistance= sumDistance/10;
  return distance;
}
```

4.18 Code Snippet of Panic Button

Below is a code snippet which shows how the panic button is coded to enable it to function as expected. The panic button is linked to the android application via the HC-05 Bluetooth module.

```
void loop() {


  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    distress = 1;
    Serial.print(distress);
    Serial.print("|");
    Serial.println(alert);
    delay(1000);
  }
  if (buttonState == LOW)
  {
    // turn LED off:
    digitalWrite(ledPin, LOW);
    distress = 0;
    Serial.print(distress);
    Serial.print("|");
    Serial.println(alert);
    //delay(1000);
  }
```

4.19 Code Snippet to Trigger Buzzer

Below is a code snippet which shows how the active piezo buzzer is coded to enable it to function as expected. The buzzer is triggered when an entity approaches the sensors vicinity. It can be switched off either by using the android application or by the membrane keypad.

```
if (alarmActivated == true) {
  sensor_value = digitalRead(sensor); // Reading sensor value from pin 10
  currentDistance = getDistance() + 10;
  alert = 1;
  if (( currentDistance < initialDistance) || (sensor_value == HIGH)) {
    tone(buzzer, 1000); // Send 1KHz sound signal
    lcd.clear();
    enterPassword();
  }
}
if (!alarmActivated) {
  alert = 0;
  if (screenOffMsg == 0 ) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("A - Activate");
    lcd.setCursor(0, 1);
    lcd.print("B - Change Pass");
    screenOffMsg = 1;
  }
  keypressed = myKeypad.getKey();
  if (keypressed == 'A') {       //If A is pressed, activate the alarm
    tone(buzzer, 1000, 200);
    activateAlarm = true;
  }
```

4.20 Conclusion

This chapter focused on the connection schematics and actual implementation of the system. It exhibited the code that runs the system as well as the circuit diagram setup. In addition, it demonstrated the actual use of the system in real-time as opposed to the previous theoretical chapters.

CHAPTER 5: SUMMARY AND CONCLUSION

5.1 Introduction

The Arduino based Infrared Intrusion Detection and Motion Sensor Device has been shown to be a success. One of the fundamental objectives which is personalization of the system has been successfully met. If applied in real-life situations the intrusion device is capable of detecting alerting and preventing unauthorized access by either humans or objects. Greater control of the system has been granted to the user as opposed to the existing obsolete systems.

5.2 Recommendations

The Arduino based intrusion detection system and motion sensor device as like any other developed system is subject to recommendations and input from the end user. The following recommendations are as follows:

1) The sensitivity of the pyro-electric infrared motion sensor should have the manufacturers of the components greatly reduce it as a mere rise in room temperature is detected and this triggers the buzzer to start ringing which makes the system difficult to operate in real-time situations.
2) Range of sensors is another issue and it is recommended that instead of having the sensors calibrated they should already come with a predefined range as it is really difficult to calibrate the range.
3) Alternate input methods such as retina eye scanners, fingerprint scanner, voice recognition sensors depending on the end user budget can be incorporated as an alternative for the membrane keypad thus greatly increasing the level of security.
4) The sensors are able to pick up any object which is approaching and this is not efficient as a bug can be picked up etc. It is recommended to have smart sensor with object recognition software to reduce annoying triggers of the alarm.
5) A mechanism should be put in place to disable an intruder for example the linking of the system with a self-firing gun/pistol.

5.3 Limitations

During the system development phase of the device the following limitations were realized/observed:

1) Sensors do not have three-hundred and sixty-degree coverage thus this implies that more sensors are required leading to an expensive and costly device to produce.
2) The pyro-electric infrared motion sensor and the ultrasonic sensor cannot differentiate between an object and a human and this leads to the creation of an annoying system as it picks up every entity within range.
3) The intrusion system has no mechanism of disabling the intruder when they enter thus assets could still be lost/stolen.

5.4 Conclusion

The Arduino Based Infrared Intrusion Detection and Motion Sensor device has been successfully developed. The improvements added to the device include a panic button which immediately notifies the emergency response team secondly, instead of being able to have one emergency response team to notify the android app has added more textboxes to add more contact numbers, thirdly instead of having to turn off the triggered alarm by keypad input one can instead make use of the android app.

Arduino components Code

```
/*
  Nyasha Final Year Project
  Midlands State University
*/
#include <LiquidCrystal.h> // includes the LiquidCrystal Library
#include <Keypad.h>
#define buzzer 8
#define trigPin 9
#define echoPin 10
#define sensor 6
String inString;
const int buttonPin = 7;
int buttonState = 0;
int distress = 0;
int alert = 0;
int ledPin = 13;
int sensor_value;
long duration;
int distance, initialDistance, currentDistance, i;
int screenOffMsg = 0;
String password = "1212";
String tempPassword;
boolean activated = false; // State of the alarm
boolean isActivated;
```

```
boolean activateAlarm = false;
boolean alarmActivated = false;
boolean enteredPassword; // State of the entered password to stop the alarm
boolean passChangeMode = false;
boolean passChanged = false;
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keypressed;
//define the cymbols on the buttons of the keypads
char keyMap[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {21, 20, 19, 18};  //Row pinouts of the keypad
byte colPins[COLS] = {17, 16, 15, 14}; //Column pinouts of the keypad
Keypad myKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, ROWS, COLS);
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
// Creates an LC object. Parameters: (rs, enable, d4, d5, d6, d7)
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
```

```
  pinMode(buzzer, OUTPUT); // Set buzzer as an output
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(sensor, INPUT); // configuring pin 12 as Input for PIR
}
void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    distress = 1;
    Serial.print(distress);
    Serial.print("|");
    Serial.println(alert);
    delay(1000);
  }
  if (buttonState == LOW)
  {
    // turn LED off:
    digitalWrite(ledPin, LOW);
```

```
  distress = 0;
  Serial.print(distress);
  Serial.print("|");
  Serial.println(alert);
  //delay(1000);
 }
if (activateAlarm) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Alarm will be");
  lcd.setCursor(0, 1);
  lcd.print("activated in");
  int countdown = 10; // 10 seconds count down before activating the alarm
  while (countdown != 0) {
    lcd.setCursor(13, 1);
    lcd.print(countdown);
    countdown--;
    tone(buzzer, 700, 100);
    delay(1000);
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Alarm Activated!");
  initialDistance = getDistance();
```

```
  activateAlarm = false;
  alarmActivated = true;
}
if (alarmActivated == true) {
  sensor_value = digitalRead(sensor); // Reading sensor value from pin 10
  currentDistance = getDistance() + 10;
  alert = 1;
  if (( currentDistance < initialDistance) || (sensor_value == HIGH)) {
    tone(buzzer, 1000); // Send 1KHz sound signal
    lcd.clear();
    enterPassword();
  }
}
if (!alarmActivated) {
  alert = 0;
  if (screenOffMsg == 0 ) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("A - Activate");
    lcd.setCursor(0, 1);
    lcd.print("B - Change Pass");
    screenOffMsg = 1;
  }
  keypressed = myKeypad.getKey();
```

```
if (keypressed == 'A') {      //If A is pressed, activate the alarm
  tone(buzzer, 1000, 200);
  activateAlarm = true;
}
else if (keypressed == 'B') {
  lcd.clear();
  int i = 1;
  tone(buzzer, 2000, 100);
  tempPassword = "";
  lcd.setCursor(0, 0);
  lcd.print("Current Password");
  lcd.setCursor(0, 1);
  lcd.print(">");
  passChangeMode = true;
  passChanged = true;
  while (passChanged) {
    keypressed = myKeypad.getKey();
    if (keypressed != NO_KEY) {
      if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
          keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
          keypressed == '8' || keypressed == '9' ) {
        tempPassword += keypressed;
        lcd.setCursor(i, 1);
        lcd.print("*");
```

```
    i++;
    tone(buzzer, 2000, 100);
  }
}
if (i > 5 || keypressed == '#') {
  tempPassword = "";
  i = 1;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Current Password");
  lcd.setCursor(0, 1);
  lcd.print(">");
}
if ( keypressed == '*') {
  i = 1;
  tone(buzzer, 2000, 100);
  if (password == tempPassword) {
    tempPassword = "";
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Set New Password");
    lcd.setCursor(0, 1);
    lcd.print(">");
    while (passChangeMode) {
```

```
keypressed = myKeypad.getKey();
if (keypressed != NO_KEY) {
 if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
    keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
    keypressed == '8' || keypressed == '9' ) {
  tempPassword += keypressed;
  lcd.setCursor(i, 1);
  lcd.print("*");
  i++;
  tone(buzzer, 2000, 100);
 }
}
if (i > 5 || keypressed == '#') {
 tempPassword = "";
 i = 1;
 tone(buzzer, 2000, 100);
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Set New Password");
 lcd.setCursor(0, 1);
 lcd.print(">");
}
if ( keypressed == '*') {
 i = 1;
```

```
              tone(buzzer, 2000, 100);
              password = tempPassword;
              passChangeMode = false;
              passChanged = false;
              screenOffMsg = 0;
            }
          }
        }
      }
    }
  }
 }
}
void enterPassword() {
 int count = 0;
 int k = 5;
 tempPassword = "";
 activated = true;
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print(" *** ALARM *** ");
 lcd.setCursor(0, 1);
 lcd.print("Pass>");
 while (activated) {
```

```
alert = 1;

// read the state of the pushbutton value:

buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed. If it is, the buttonState is HIGH:

if (buttonState == HIGH) {

  // turn LED on:

  digitalWrite(ledPin, HIGH);

  distress = 1;

  delay(500);

  Serial.print(distress);

  Serial.print("|");

  Serial.println(alert);

  //delay(1000);

}

if (buttonState == LOW) {

  // turn LED off:

  digitalWrite(ledPin, LOW);

  distress = 0;

  Serial.print(distress);

  Serial.print("|");

  Serial.println(alert);

  //delay(1000);

}
```

```
keypressed = myKeypad.getKey();
if (keypressed != NO_KEY) {
  if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
      keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
      keypressed == '8' || keypressed == '9' ) {
    tempPassword += keypressed;
    lcd.setCursor(k, 1);
    lcd.print("*");
    k++;
  }
}
if (k > 9 || keypressed == '#') {
  tempPassword = "";
  k = 5;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" *** ALARM *** ");
  lcd.setCursor(0, 1);
  lcd.print("Pass>");
}
if ( keypressed == '*') {
  if ( tempPassword == password ) {
    activated = false;
    alarmActivated = false;
```

```
    noTone(buzzer);
    screenOffMsg = 0;
  }
  else if (tempPassword != password) {
    count++;
    lcd.setCursor(0, 1);
    lcd.print("Wrong! Try Again");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" *** ALARM *** ");
    lcd.setCursor(0, 1);
    lcd.print("Pass>");
  }
  if (count == 3) {
    distress = 1;
    Serial.print(distress);
    Serial.print("|");
    Serial.println(alert);
    delay(1000);
    count = 0;
  }
}
if (Serial.available()) {
```

```
    inString = Serial.readString();
    //Serial.print(inString);
     if (inString=="1234") {
      activated = false;
      alarmActivated = false;
      noTone(buzzer);
      screenOffMsg = 0;
    }
  }
  }
}
// Custom function for the Ultrasonic sensor
long getDistance() {
  //int i=10;
  //while( i<=10 ) {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
```

```
  // Calculating the distance

  distance = duration * 0.034 / 2;

  //sumDistance += distance;

  //}

  //int averageDistance= sumDistance/10;

  return distance;

}
```

Android application Code

```
<EditText
   android:id = "@+id/editText2"
   android:layout_width = "wrap_content"
   android:layout_height = "wrap_content"
   android:inputType = "textPassword" />

<EditText
   android:id = "@+id/editText1"
   android:layout_width = "wrap_content"
   android:layout_height = "wrap_content"
/>
<Button
   android:id = "@+id/button1"
   android:layout_width = "wrap_content"
   android:layout_height = "wrap_content"
   android:onClick = "login"
   android:text = "@string/Login"
/>

public void login(View view){
   if(username.getText().toString().equals("admin") &&
password.getText().toString().equals("admin")){

   //correct password
   }else{
   //wrong password
}
import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;

import android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;


public class MainActivity extends Activity  {
   Button b1,b2;
   EditText ed1,ed2;

   TextView tx1;
   int counter = 3;

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

      b1 = (Button)findViewById(R.id.button);
      ed1 = (EditText)findViewById(R.id.editText);
      ed2 = (EditText)findViewById(R.id.editText2);

      b2 = (Button)findViewById(R.id.button2);
      tx1 = (TextView)findViewById(R.id.textView3);
      tx1.setVisibility(View.GONE);

      b1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            if(ed1.getText().toString().equals("admin") &&
               ed2.getText().toString().equals("admin")) {
                  Toast.makeText(getApplicationContext(),
                     "Redirecting...",Toast.LENGTH_SHORT).show();
               }else{
                  Toast.makeText(getApplicationContext(), "Wrong
                     Credentials",Toast.LENGTH_SHORT).show();

                  tx1.setVisibility(View.VISIBLE);
                  tx1.setBackgroundColor(Color.RED);
                  counter--;
                  tx1.setText(Integer.toString(counter));

                  if (counter == 0) {
                     b1.setEnabled(false);
                  }
               }
         }
      });

      b2.setOnClickListener(new View.OnClickListener() {
         @Override
```

```
      public void onClick(View v) {
        finish();
      }
    });
  }
}
<?xml version = "1.0" encoding = "utf-8"?>
<RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"
  xmlns:tools = "http://schemas.android.com/tools" android:layout_width="match_parent"
  android:layout_height = "match_parent" android:paddingLeft=
"@dimen/activity_horizontal_margin"
  android:paddingRight = "@dimen/activity_horizontal_margin"
  android:paddingTop = "@dimen/activity_vertical_margin"
  android:paddingBottom = "@dimen/activity_vertical_margin" tools:context =
".MainActivity">

  <TextView android:text = "Login" android:layout_width="wrap_content"
    android:layout_height = "wrap_content"
    android:id = "@+id/textview"
    android:textSize = "35dp"
    android:layout_alignParentTop = "true"
    android:layout_centerHorizontal = "true" />

  <TextView
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:text = "Tutorials point"
    android:id = "@+id/textView"
    android:layout_below = "@+id/textview"
    android:layout_centerHorizontal = "true"
    android:textColor = "#ff7aff24"
    android:textSize = "35dp" />

  <EditText
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:id = "@+id/editText"
    android:hint = "Enter Name"
    android:focusable = "true"
    android:textColorHighlight = "#ff7eff15"
    android:textColorHint = "#ffff25e6"
    android:layout_marginTop = "46dp"
    android:layout_below = "@+id/imageView"
    android:layout_alignParentLeft = "true"
    android:layout_alignParentStart = "true"
    android:layout_alignParentRight = "true"
    android:layout_alignParentEnd = "true" />

  <ImageView
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:id="@+id/imageView"
    android:src="@drawable/abc"
    android:layout_below="@+id/textView"
    android:layout_centerHorizontal="true" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:id="@+id/editText2"
    android:layout_below="@+id/editText"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/editText"
    android:layout_alignEnd="@+id/editText"
    android:textColorHint="#ffff299f"
    android:hint="Password" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Attempts Left:"
    android:id="@+id/textView2"
    android:layout_below="@+id/editText2"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:textSize="25dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Text"
    android:id="@+id/textView3"
    android:layout_alignTop="@+id/textView2"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignBottom="@+id/textView2"
    android:layout_toEndOf="@+id/textview"
    android:textSize="25dp"
    android:layout_toRightOf="@+id/textview" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="login"
    android:id="@+id/button"
    android:layout_alignParentBottom="true"
    android:layout_toLeftOf="@+id/textview"
```

```
        android:layout_toStartOf="@+id/textview" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cancel"
        android:id="@+id/button2"
        android:layout_alignParentBottom="true"
        android:layout_toRightOf="@+id/textview"
        android:layout_toEndOf="@+id/textview" />

</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sairamkrishna.myapplication" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

        </activity>

    </application>

</manifest>
```

REFERENCES

Kumar P.V. (2014). Automatic Calling Recognition System. *International Journal of Electronic and Electrical Engineering*, 1(7), 266-269.

Sharma A. (2014). Android Phone Controlled Robot Using Bluetooth. *International Journal of Science, Engineering and Technology Research (IJSETR)*, 1(2), 443-448.

Maqbool S. (2013). Smart Traffic Light Control and Congestion Avoidance System during Emergencies Using Arduino and ZigBee. *International Journal Advanced Research in Computer Science and Software Engineering*, 1(3), 1801-1808.

Yan M. & Shi H. (2013) Smart living using Bluetooth based android smartphone. *International Journal of Wireless & Mobile Networks (IJWMN)*, 1(5), 65-72.

Tare S.R. (2013). Review Paper on Android Operating System-A Step towards Green Environment. 1(3), 54-57.

Marquardt P. (2013). Decoding Vibrations from Nearby Keyboards Using Mobile Phone. *International Journal Advanced Research in Computer Science and Software Engineering*, 1(3), 180-184.

Gobel S. (2013). Using the Android Platform to control Robots. *Software Engineering Research Group*, 1(3), 45-57.

Wu J. (2010). Hand Motion-Based Remote Control Interface with Vibrotactile Feedback for Home Robots. *International Journal of Computer Communication and Information System (IJCCIS)*, 1(2), 1-10.

Snehal B. & Priyanka (2010). Wireless Intrusion Detection System. *International Journal of Computer Applications*, 1(5), 1-5.

Madan V. & Reddy R.N. (2012). GSM-Bluetooth based Remote Monitoring and Control System with automatic Light Controller. *International Journal of Computer applications (0975-8887), Vol. 46, no. 1*

Lange P.M & Nielsen K.L.G. (2010). Phase recognition in an operating room using sensor technology.

Winkler F (2010). Wireless Sensor networks for Human Intruder Detection. *Indian institute of Science, Bangalore*.

Eslaminejad M.R & Sookhak M (2011). A Review of Routing Mechanisms in Wireless Sensor Networks. *International Journal of Computer Science and Telecommunication*, 1(2) 7-8.

Wan L & Chen T (2009). Automobile Antitheft System Design based on GSM. *International Conference on Advanced Computer Control*, 1(9) 551-554.

Pandey K. J. & Choudhary D. (2010). Wireless Sensor Networks for Human Intruder Detection. *Journal of the Indian Institute of Science, Vol. 90*.

Ramadan M.N & Khedhar M. D. (2012). Intelligent Anti-theft and Tracking System for Automobiles. *International Journal of Machine Learning and Computing, Vol. 2, no. 1, Feb. 2012*.