Interloper PI Detector

IoT Application Project

An IoT based security application for detecting motion within a room, controlled by automated telegram bot. If motion is detected a buzzer would be triggered and a photo and video recording will be sent to the bot and emailed to the admin.

Fyaz Qadir Ahmed Ikram | K00237093 | Technological University of Shannon: Midlands & Midwest | Internet Systems Development | Year 4 | Technology Futures and Connected Living

Table of Contents

Application Name and Description 3
Background Summary 4
System Architecture 6
Functional Requirements and App Features 7
Non-Functional Requirements 8
Parts Required 9
Circuit Diagram 10
Configurations Required 11
System Implementation 12
Additional Feature 14
Testing and Results 15
Future Work 16
Conclusion 17
References 18
Appendices 19

Table of Figures

Figure 1 Annual Crime Figures 2020 4
Figure 2 Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030, by use case – Security and Fire Alarms (in millions) 5
Figure 3 System Architecture Diagram 6
Figure 4 Circuit Diagram 10
Figure 5 Wiring Implementation 19
Figure 6 Libraries Used 20
Figure 7 Configurations Set Up 20
Figure 8 Handle Method 21
Figure 9 Capture Method 21
Figure 10 Main Method 22
Figure 11 SendNotifcation Method 23
Figure 12 Login Screen 23
Figure 13 Main Dashboard Screen 24
Figure 14 History of Motion Detected Date&Time Stamps Screen 24
Figure 15 History of Motion Detected Images Screen 25
Figure 16 Login Screen 25
Figure 17 Dashboard Screen (Note that the Charts are just Dummy Data) 26
Figure 18 Exported Excel Screen 26
Figure 19 Garda Message notifying that a House has been Intruded 27

Application Name and Description

Application Name: Interloper PI Detector

Application Category: Home and Automation Monitoring

Application Description: Instant data feed on your mobile or email if a motion is detected in your house while you're away.

Background Summary

IoT applications run on Internet of Things devices and can be tailored to practically any sector or vertical, including healthcare, industrial automation, smart homes and buildings, automotive, and wearable technology. The scope of this project is to develop a home intrusion detection system for homeowners to prevent house burglary, around the Limerick area. According to PhoneWatch a burglary takes place in Ireland every 48 minutes. In the Limerick division there were a total of 158 burglaries in the year 2020.

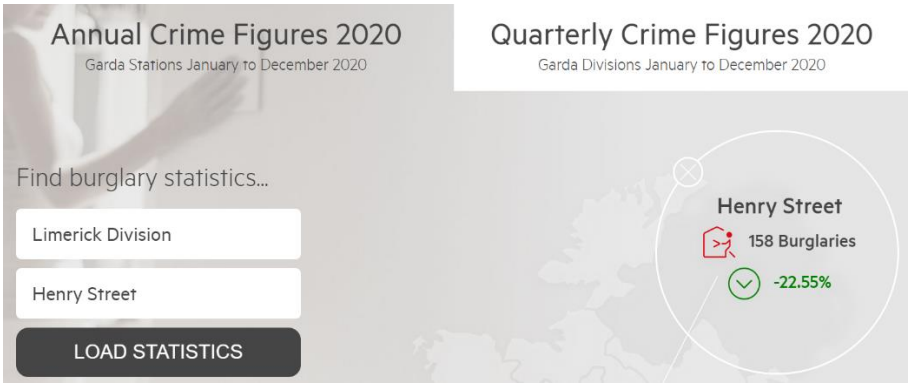

Figure 1 Annual Crime Figures 2020

Due to these reasons the Author has decided to create an IoT application to reduce house burglaries in Limerick. The value proposition of this application is a Home Intruder Detection System, to Safeguard and Protect your House. This application will capture any motion detected using 2 PIR sensors in your house when you are away and alert you by email, mobile application and also via a custom-built telegram bot, attaching a photo and video capture. When a motion is triggered, the lights would be turned on to capture a clearer view of photos and videos and a buzzer ringer to signal that there has been an intrusion, in addition to a call being made to the guards, if the program is not stopped via the push button within 10 seconds. A custom-made website also shows a history of when the motion was detected, and allows the exporting of the data, to an excel spreadsheet, to aggregate and detect patterns within the data. All data is stored using Firebase Realtime Database and Storage. The application is planned to be operated via remote measures by using a telegram bot that allows users to start and stop the program when leaving the house. In comparison to other burglar systems this system would cost approximately $100 to build, which is way cheaper and more affordable for homeowners, compared to other systems which cost around $170 - $250. The application would be initially used in the Authors house to test how the application works and functions. This IoT application can replicate itself to work in other environments too in the future, such as in hospitals, airports etc.

There are many IoT applications that are already built around the area of home security and monitoring such as Motion Sensor by Code Cartel LLC, and Smart Zone Smart Home Panel, that records video automatically when it detects movement. Most home security monitoring apps have basic features such as record pictures/videos when a motion detected. What makes the Authors application unique is that when a motion is detected, the light turns on so it can capture

a clearer video and photo capture of when the intruder was at the house, but in addition to sending a message alert to the garda when an intruder has been detected within the house. As mentioned currently the application is using firebase as the web server to save the photos and log stamps but if this system was to store multiple more files of photos and log stamps firebase would be quite expensive as the price increases by usage, the cheapest alternative for a web server would be Amazon S3 because each uploaded file has its own HTTP URL, and you can easily load it straight into your web application.

According to Statista the number of IoT connected devices for security and fire alarms in 2021 was 338.7 million is expected to grow to 552.2 million by 2024 and then by 2030 to 1047 million. This shows that there is a huge scope of IoT devices in the security and fire alarms sector. This was also one huge factor why the Author had decided to develop an application around this market within the Internet of Things.

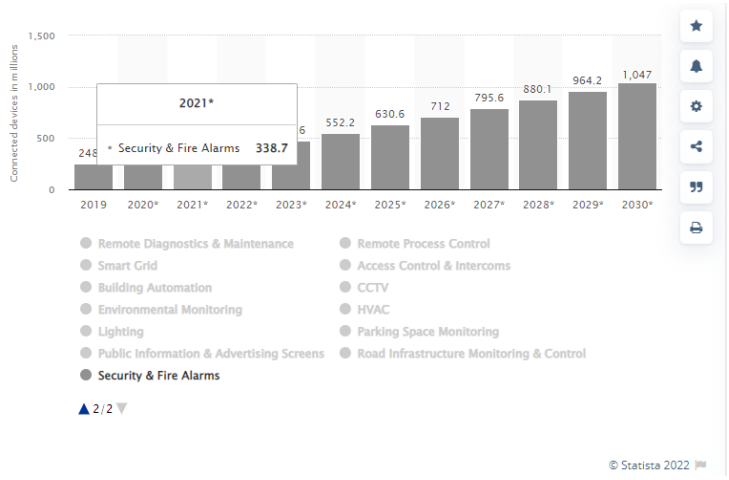

Figure 2 Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030, by use case – Security and Fire Alarms (in millions)

Also, in addition, according to the IEEE Xplore "Internet of Things is composed of things that have unique identities and are connected to each other over internet. It is simply connecting and monitoring various devices and sensors through Internet. This paved the way for home automation and monitoring which makes human life more comfortable and secured." This shows that the demand for IoT applications is increasing and becoming more popular among the common folk.

System Architecture

The system architecture diagram is shown below. The Author aims to create this application with the guidelines per the system architecture diagram.

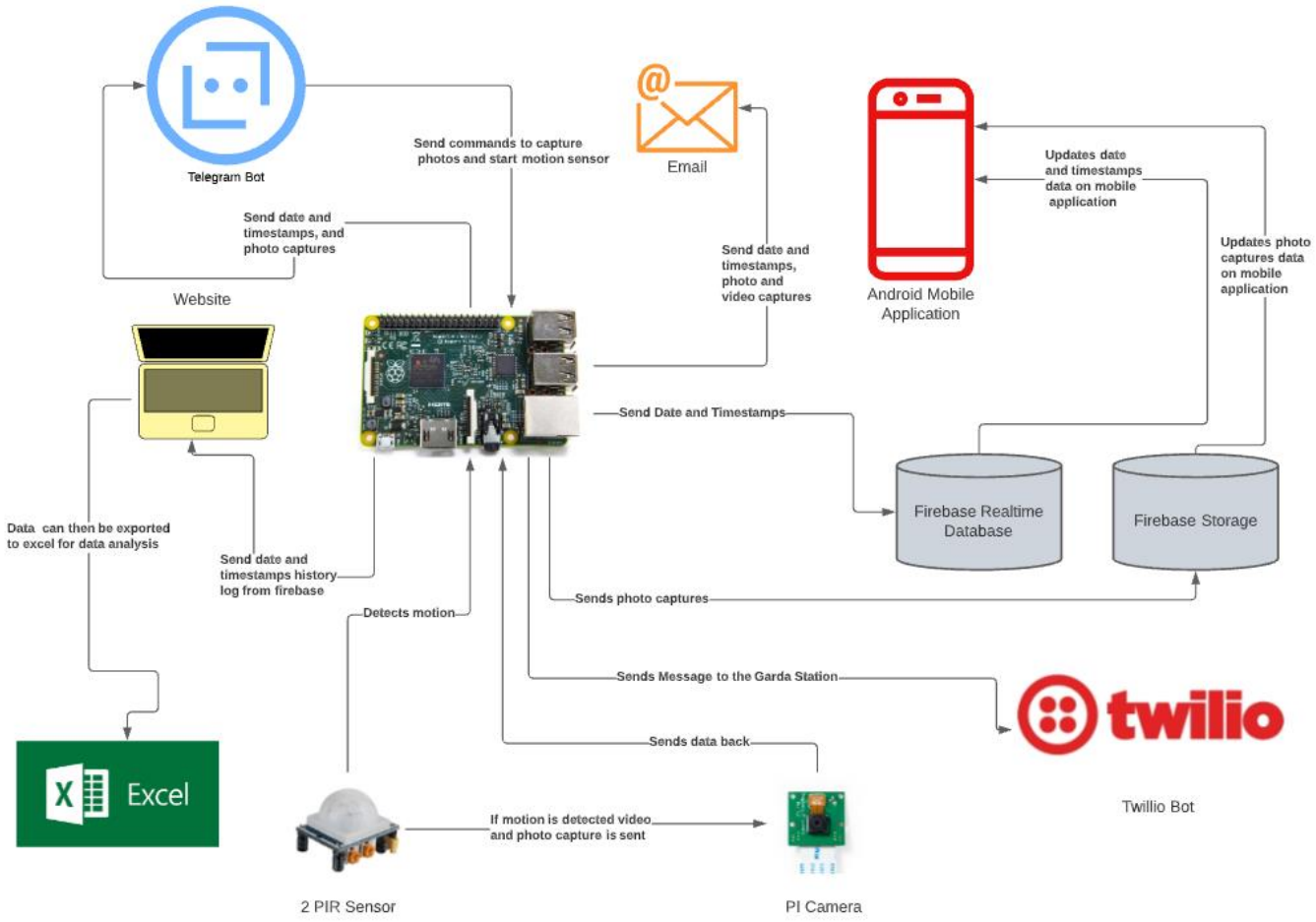

Figure 3 System Architecture Diagram

Functional Requirements and App Features

This IoT application would include the following features:

1. Telegram bot integration to take pictures of the area, to start the motion detection program to monitor If motion is detected within a specific room and to stop the program
2. Buzzer trigger and light to identify that a motion has been detected, and a clear image of the intruder and also a button switch to disable the alarm, if button is not pushed the program would send a message that there is an intruder to the Garda Station
3. Video and photo capture sent via email and telegram bot integration if a motion is detected, and button is not pushed.
4. Firebase Realtime Database integration to store motion detection sensor triggering history date and timestamps sent from the IoT device
5. Firebase Storage integration to store images of when the motion detection sensor was triggered
6. Interloper PI Detector mobile application with authentication functionality to only allow authorized users to use the mobile application, to view motion detection sensor triggering history date and timestamps, and photo captures sent from the Firebase Realtime Database and Firebase Storage
7. Website displaying the history log date of when the motion was detected, which can be exported to excel for data extraction and visualization using charts, for deep data analysis.

Non-Functional Requirements

The Author has also taken into consideration the following non-functional requirements to develop and plan in the future for this IoT application which are:

- **Availability** – The Author has decided for this application to be operational 24/7 and have minimum idle time by reducing the number of bugs within the application in the future
- **Performance** – The Author has also considered allowing multiple users access this application simultaneously and having efficient and fast response times when using the IoT application in the future
- **Security** – The Author has also implemented security and safety requirements to secure the data in the cloud by using firebase cloud database and storages, and also by using firebase authentication to allow only authorized personal to view the data on the Interloper PI Detector mobile application, in addition to securing the website with authentication functionality.
- **Usability** – The Author has also incorporated a friendly and ease of use interaction framework using telegram bot that allows the users to seamlessly interact with the IoT application

Parts Required

The Parts that were required for the application were:

1. 1 Raspberry Pi 4
2. 1 Active Buzzer
3. 1 Red LED
4. 1 Push Button
5. 2 PIR Sensor
6. 1 220 Ω Resistor
7. Several Jumper Wires

Circuit Diagram

This is a Circuit Diagram created from fritzing of the Interloper PI Detector IoT application.

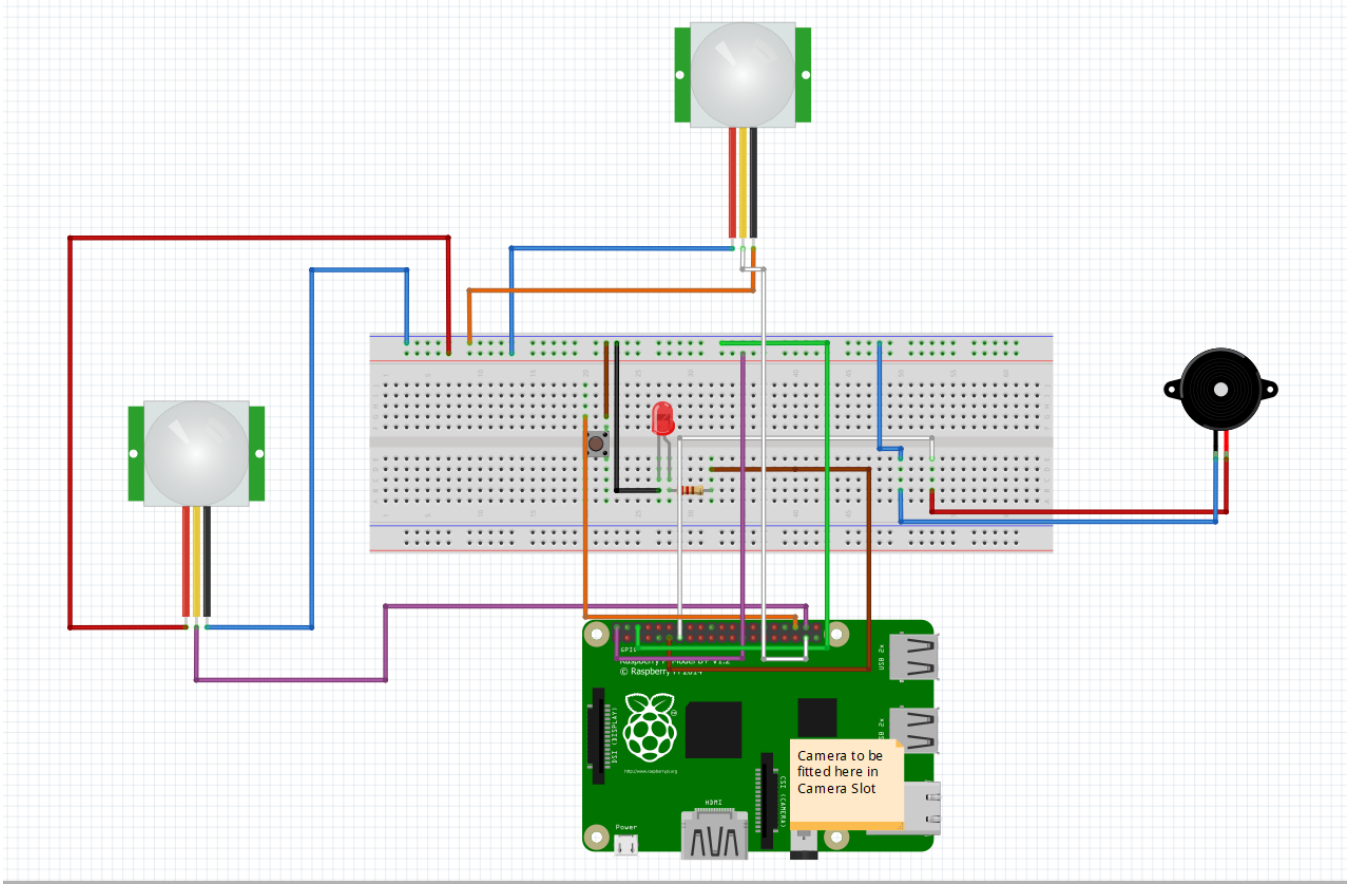

Figure 4 Circuit Diagram

Configurations Required

Configurations that were needed for this application is listed below in order:

1. sudo apt-get update
2. sudo apt-get upgrade
3. sudo raspi-config
4. Enable Camera
5. sudo reboot
6. sudo apt-get install python-picamera
7. sudo pip3.6 install telepot
8. sudo pip install telepot
9. sudo pip install pyTelegramBotAPI
10. sudo pip install git+https://github.com/eternnoir/pyTelegramBotAPI.git
11. sudo pip3 install telepot
12. sudo apt install -y gpac
13. sudo reboot
14. sudo apt full-upgrade
15. sudo reboot
16. sudo install pyrebase
17. sudo reboot
18. sudo pip3 install twilio

System Implementation

Wiring Implementation (Please See Appendix A for Wiring Diagram)

1. First the Author planned the diagram using the Fritzing software to plan out how the wiring would be implemented, this required constant changes and updates to the diagram before trying to implement it, to the actual raspberry pi and breadboard with the several parts
2. After successfully planning out the diagram using fritzing. The Author then started implementing the circuit diagram by then looking and referencing the fritzing diagram created previously. Once the wiring was complete the Author ran several simple programs to check if all the separate parts (buttons, LED's, buzzer) being used was working.

Python Raspberry PI Coding Implementation (Please See Appendix B for Code Snippets)

1. First imported all libraries that was going to be used and installed the packages
2. Setup all the configurations for the bots, firebase, and wiring
3. Then coded the handle method to manage what happens when a certain command is pressed
4. Then coded the capture method which manages if the user wants to take a photo of the room at any given specific time
5. Then coded the main method which handled the monitoring using the motion PIR sensor, to track if a motion is detected, and if so, the protocols (.i.e., email notification, messaging the Garda) that would occur when the PIR sensor is triggered
6. Then finally coded the sendNotifcation method which sends a video recording and message to the telegram bot when the motion sensor is triggered
7. The Author then also tried implementing facial recognition, to detect if the motion triggered was by a homeowner or not in the application by using OpenCV and Imutils but due to having other issues with the base system, and facial recognition to work coherently, the Author had then decided to leave the facial recognition as part of future work.

Mobile Application Coding Implementation (Please See Appendix C for the Mobile App)

1. The Author then proceeded to code the mobile application which pulls the data from firebase and displays the data(images and date time stamps) on the mobile application, with login functionality to allow only authorized users to access the data

Web Application Coding Implementation (Please See Appendix D for the Web Dashboard)

1. The Author then proceeded to build a website that displays all the data being logged from the Firebase Database, and display it in a table layout using JavaScript, HTML and CSS, the data can then also be exported to excel in a table format allowing the user to see patterns within the different data sets

Additional Feature

The Author has implemented the following additional features within this IoT application which are:

1. **Mobile Application:**

   The first additional feature that the Author has incorporated into this IoT mobile application was creating a mobile application that allows homeowners to view a history of when the motion sensor was triggered in a time and date stamp format. In addition to displaying a list of images displaying when the motion sensor was triggered. The data is Realtime and gives immediate data updates as all the data is stored in Firebase. The mobile application also has authentication which allows only authorized users to use it.

2. **Web Dashboard:**

   The second additional feature that was incorporated by the Author was a web dashboard that displays the date and time stamps when the motion was detected. This web dashboard also allows users to download the data and export to excel to view the data for deep data analysis and predictive model visualizations, to see patterns and trends within the data.

Testing and Results

The Author has tested the application individually and also tested it by installing and demoing the IoT application to other family members, in different rooms. The feedback that the Author's family had provided to the Autor in regard to the mobile application was the application was very smooth and easy to use and was very user friendly, however they mentioned a few areas that I could improve my application which was to have a notification message on the mobile application to notify homeowners when a new motion has been detected and been logged into the application. Another feature that was recommended was to have a stats and predictive model visualizations on the web dashboards to allow homeowners to see the patterns and trends that occur when an intruder is in the house, for example the times when intruders attack the house, and the frequency of how many burglaries happen around the area. The Author has taken into consideration of their feedback and would try to be implemented within the next update of the application. But for purposes of the scope of the project, the Author has decided to consider this into the future work and try to implement these changes, in the next update.

Future Work

The Author aims to further develop this IoT application in the future. Two additional features that the Author wishes to incorporate into this application are:

1. **Data Visualization Charts**

   One feature that the Author would like to implement into their IoT application would be to create automatic AI Based visualizations when the homeowners visit the web dashboard and also when trying to export the data to excel, from the website which allows them to see a detailed report automatically generated on the motion sensing triggering details of the house within a specific time frame that can be selected by the user.

2. **Machine Learning & Artificial Intelligence**

   Another feature that the Author would like to implement into their IoT application in the future is the use of Machine Learning and Artificial Intelligence. This would be used instead of a button to deactivate the alarm. Machine Learning and Artificial Learning would be able to automatically detect if the person entering the house is a resident, and if they are, the alarm would be deactivated, if not the alarm would be installed and turned on.

Conclusion

To conclude this report, the Author can say that they have learned the process of developing an IoT based system and also learned the best practices and techniques when trying to build a unique IoT application from scratch. The Author has also learnt various technologies that could be incorporated into modern IoT application systems, in addition to understanding the fundamental concepts of IoT based application systems and how to help shape the modern world around us.

References

Background Research Material

1. Arduino Blog. 2022. Introducing the Arduino UNO Mini Limited Edition: Pre-orders now open. [online] Available at: <https://blog.arduino.cc/2021/11/24/introducing-the-arduino-uno-mini-limited-edition-pre-orders-now-open/> [Accessed 26 February 2022].
2. Tom's Hardware. 2022. Raspberry Pi's Ninth Birthday: 9 Things You Might Not Know. [online] Available at: <https://www.tomshardware.com/news/raspberry-pi-9th-birthday> [Accessed 26 February 2022].
3. 2022. [online] Available at: <https://all3dp.com/2/best-raspberry-pi-iot-projects-using-raspberry-pi/> [Accessed 26 February 2022].
4. Digiteum. 2022. Difference Between Raspberry Pi and Arduino: Which Is Better for IoT Project — Digiteum. [online] Available at: <https://www.digiteum.com/comparing-arduino-raspberry-pi-iot/> [Accessed 26 February 2022].
5. Statista. 2022. IoT connected devices by use case 2030 | Statista. [online] Available at: <https://www.statista.com/statistics/1194701/iot-connected-devices-use-case/> [Accessed 26 February 2022].
6. Findstack. 2022. 21+ Internet of Things Statistics, Facts & Trends for 2022. [online] Available at: <https://findstack.com/internet-of-things-statistics/> [Accessed 26 February 2022].
7. Watters, A., 2022. 30 Internet of Things Stats & Facts for 2022. [online] Default. Available at: <https://connect.comptia.org/blog/internet-of-things-stats-facts> [Accessed 26 February 2022].

Prototyping:

1. Please refer to the attached fritzing image and file in the design folder within the zipped file.

Technical Research/ Documentation

1. Telepot.readthedocs.io. 2022. Introduction — telepot 12.7 documentation. [online] Available at: <https://telepot.readthedocs.io/en/latest/> [Accessed 2 March 2022].
2. Iotdesignpro.com. 2022. How to Send SMTP Email using Raspberry Pi. [online] Available at: <https://iotdesignpro.com/projects/sending-smtp-email-using-raspberry-pi> [Accessed 2 March 2022].
3. Youtube.com. 2022. [online] Available at: <https://www.youtube.com/watch?v=2G1d3q8MlsI> [Accessed 2 March 2022].
4. Youtube.com. 2022. [online] Available at: <https://www.youtube.com/watch?v=By-pDkZ1Zws> [Accessed 2 March 2022].
5. Firebase. 2022. Installation & Setup in JavaScript | Firebase Documentation. [online] Available at: <https://firebase.google.com/docs/database/web/start> [Accessed 2 March 2022].
6. Android Developers. 2022. Documentation | Android Developers. [online] Available at: <https://developer.android.com/docs> [Accessed 2 March 2022].
7. IEEE Internet of Things Journal. 2022. IEEE Internet of Things Journal - IEEE Internet of Things Journal. [online] Available at: <https://ieee-iotj.org/> [Accessed 13 April 2022].
8. Ieee-iotj.org. 2022. [online] Available at: <https://ieee-iotj.org/wp-content/uploads/2022/01/SI-on-intrusion-detection-for-IoT-CFP.pdf> [Accessed 13 April 2022].

Debugging and Error Fixes:

1. Stack Overflow. 2022. Stack Overflow - Where Developers Learn, Share, & Build Careers. [online] Available at: <https://stackoverflow.com> [Accessed 26 February 2022].

Appendices

Appendix A: Wiring of IoT System [Page 12]

- Below, is a diagram of the actual wiring and implementation of the circuit diagram.

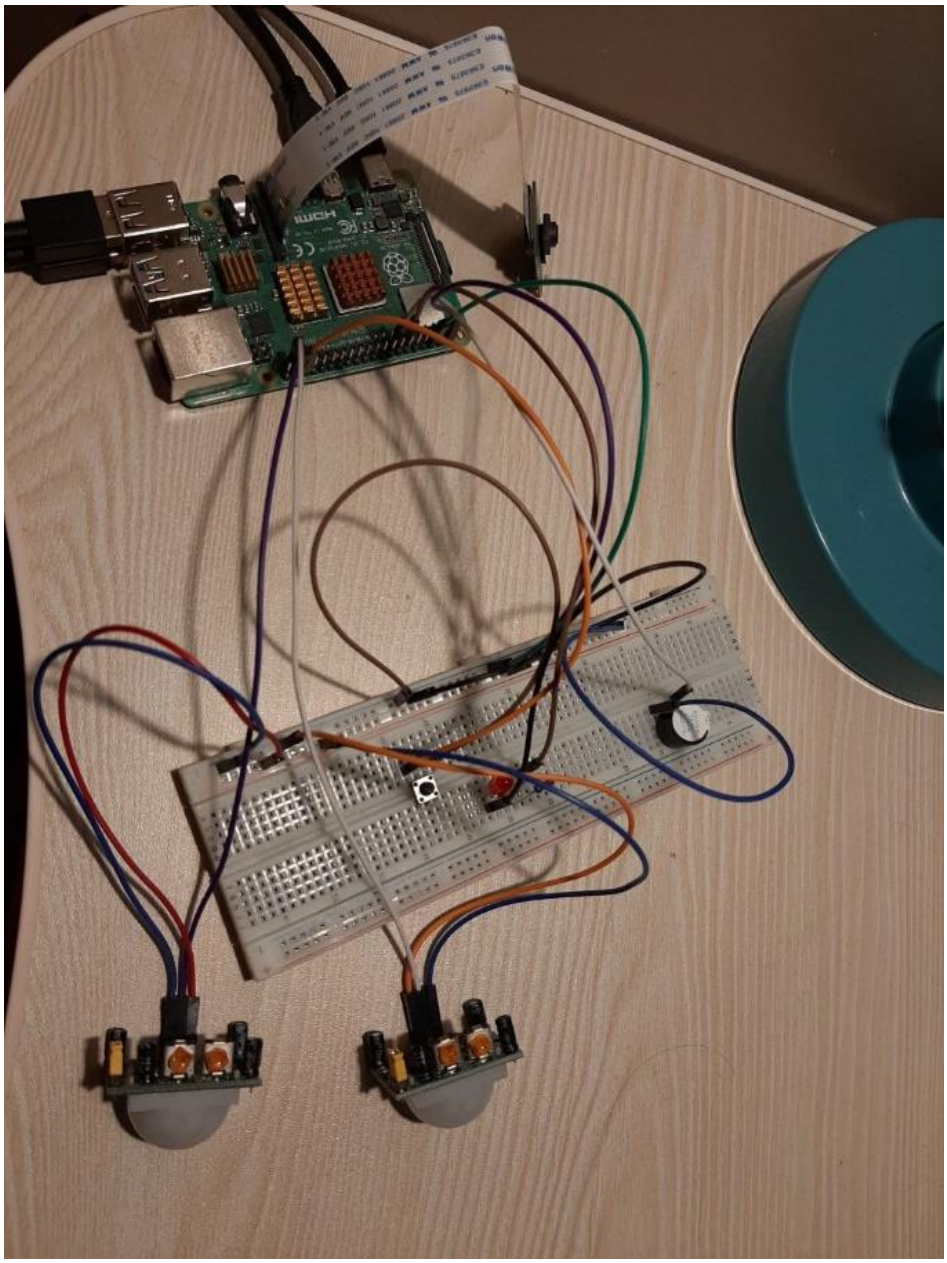

Figure 5 Wiring Implementation

Appendix B: Code Snippets for Methods [Page 12]

- Libraries initialization

```
# Libraries and API's Imported
from gpiozero import MotionSensor, Button, LED, Buzzer
from picamera import PiCamera
import picamera
from email.mime.image import MIMEImage
from email.mime.multipart import MIMEMultipart
from subprocess import call
import os
import RPi.GPIO as GPIO
import email.mime.application
import datetime
import smtplib
from time import sleep
import telepot
import sys
from telepot.loop import MessageLoop
import time
import pyrebase
from twilio.rest import Client
```

Figure 6 Libraries Used

- Configurations Set Up

```
#Twillio Account SID and Token Setup
account_sid = "AC836c1cf4753582fd0c9a125a0b230f4e"
auth_token = "3c82962f494cace9bec24faf46818bdd"

#Firebase Configiration Code to Link to Firebase Project
firebaseConfig = {
    'apiKey': "AIzaSyC0k72cW3c9269zuVHXwVk12RV5TX_ozZ4",
    'authDomain': "interloper-pi-detector.firebaseapp.com",
    'databaseURL': "https://interloper-pi-detector-default-rtdb.firebaseio.com",
    'projectId': "interloper-pi-detector",
    'storageBucket': "interloper-pi-detector.appspot.com",
    'messagingSenderId': "203002723801",
    'appId': "1:203002723801:web:31a494a6d82ae7a368edc8",
    'measurementId': "G-68NLMPZM6K"
}

#Intilizate Firebase Database and Firebase Storage
firebase = pyrebase.initialize_app(firebaseConfig)
storage = firebase.storage()
database = firebase.database()

#Created objects to point to the LED, Button, Buzzer, pirSensor sensor and PI Camera.
# For Example when MotionSensor(20) it means that the MotionSensor is connected to GPIO 20
pirSensor = MotionSensor(20)
pirSensorTwo = MotionSensor(26)
camera = PiCamera()
led = LED(17)
button = Button(16)
buzzer = Buzzer(27)

motion = 0
motionChange = 0
```

Figure 7 Configurations Set Up

- Handle Method

```
#Method to Handle bot commands for the application
def handle(msg):
    global telegramCommandText
    global chat_id
    global sendPhotoCapture

    chat_id = msg['chat']['id']
    telegramCommandText = msg['text']

    #Displays the ID of the Telegram Bot Command
    print('Message received from ' + str(chat_id))

    #If the /start is typed on the telegram bot it starts the motion detection system
    if telegramCommandText == '/start':
        #Message Displayed to the Telegram Bot
        bot.sendMessage(chat_id, 'Start Command Intiated. Interloper PI Detector Activated.')
        #Method to detect motion in the house
        main()

    #If the /photo is typed on the telegram bot it captures a photo
    elif telegramCommandText == '/photo':
        sendPhotoCapture = True
        #Message Displayed to the Telegram Bot
        bot.sendMessage(chat_id, 'Photo Command Activated.')
        sendPhotoCapture = False
        #Method to capture photo from the camera
        capture()

    #If the /quit is typed on the telegram bot it quits the program
    elif telegramCommandText == '/quit':
        #Message Displayed to the Telegram Bot
        bot.sendMessage(chat_id, 'The Program is now shutdown')
        #Exits the Program
        sys.exit()

    #If the user types another string on the telegram bot it sends a message saying invalid command.
    else:
        bot.sendMessage(chat_id, 'Invalid Command.')
```

Figure 8 Handle Method

- Capture Method

```
#Method to capture photo and send to the Telegram Bot (/photo Command Function)
def capture():
    print("Captruing Photo")
    #Stores the image locally first in the desktop directory
    photoCaptured1 = '/home/pi/Desktop/image.png'
    #Camera rotation to 180 degrees
    camera.rotation = 180
    #Captures image
    camera.capture(photoCaptured1)
    print("Sending Photo to " + str(chat_id))
    #Sends Photo capture to the Telegram Bot
    bot.sendPhoto(chat_id, photo = open('/home/pi/Desktop/image.png', 'rb'))
```

Figure 9 Capture Method

- Main Method (Only top bit of code showing. Please review code file to see full code)

```python
#Method to start the motion detetction(/start Command Function)
def main():
    #Intialized all the variables and set the Alarm state to True
    AlarmState = True
    global chat_id
    global motion
    global motionChange
    global telegramCommandText
    global sendPhotoCapture

    try:
        #While the Command is True
        while True:
            #If Alarm State is True, LED is off and waits execution of the current thread for 1 second
            if AlarmState == True:
                led.off()
                sleep(1)
                #If motion is detected a message would be displayed that there has been motion detetced and turns on the LED and Beeps
                if pirSensor.motion_detected or pirSensorTwo.motion_detected:
                    print("Motion Detected")
                    led.on()
                    buzzer.beep()
                    #Waits execution of the current thread for a 0.2 seconds
                    sleep(0.2)
                    #Then after that turns off the buzzer
                    buzzer.off()
                    #Waits for 10 seconds for user to push button
                    button.wait_for_press(timeout=10)

                    #If button is pressed within 10 seconds the program exits and turns off the LED
                    if button.is_pressed:
                        print("\nThe System Has Been Exited!")
                        bot.sendMessage(chat_id, 'The System Has Been Exited!')
                        led.off()
                        sys.exit()
```

Figure 10 Main Method

- Send Notification to Bot Method

```
#Method to send notification to the Telegram Bot that a motion has beeen detected
def sendNotification(motion):

    #Intiliazing the global Telegram Bot Chat ID
    global chat_id

    #If the motion = 1
    if motion == 1:
        #Creates a filename with the datetime timestamp as the file name
        filename = "./video_" + (time.strftime("%y%b%d_%H%M%S"))
        #Set the camera resoltion to 640x480
        camera.resolution = (640,480)
        #Sets the camera resolution to 180 degrees
        camera.rotation = 180
        #Start the camera recording using the filename created and with the .h264 extension to record video as raspberry pi uses the default .h264 extension to record videos
        camera.start_recording(filename + ".h264")
        #Waits execution of the current thread for 1 second
        sleep(1)
        #Records video for 20 seconds
        camera.wait_recording(20)
        #finishes recording after 20 seconds.
        camera.stop_recording()
        #Splits and creates an individual playable video file (.mp4) from the original (.h624) video extension
        command = "MP4Box -add " + filename + '.h264' + " " + filename + '.mp4'
        print(command)

        #Converts the video from .h264 extension to the .mp4 video extension
        call([command], shell=True)
        bot.sendVideo(chat_id, video = open(filename + '.mp4', 'rb'))
        bot.sendMessage(chat_id, 'The motion sensor is triggered!')
        bot.sendMessage(chat_id, str(datetime.datetime.now()))
```

Figure 11 SendNotifcation Method

Appendix C: Mobile Application [Page 12]

- Login Screen

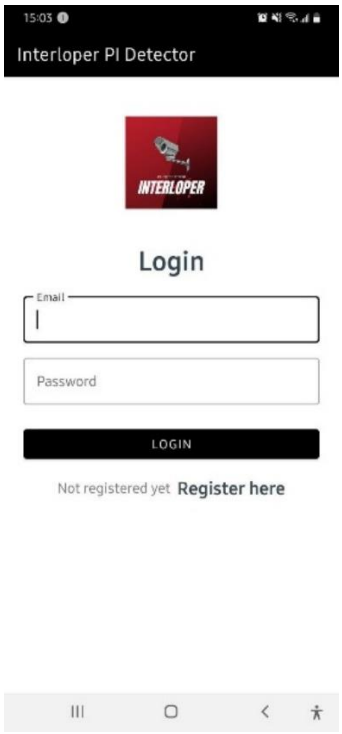

Figure 12 Login Screen

- Main Dashboard Screen

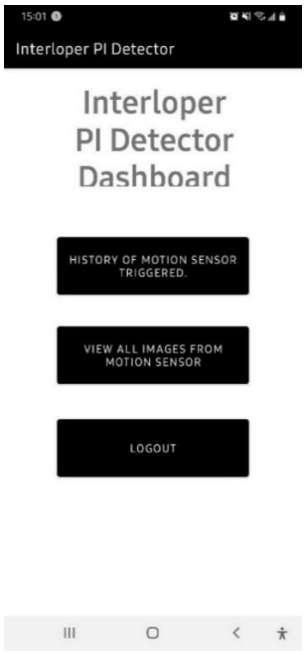

Figure 13 Main Dashboard Screen

- History of Motion Detected Date and Time Stamp History Screen

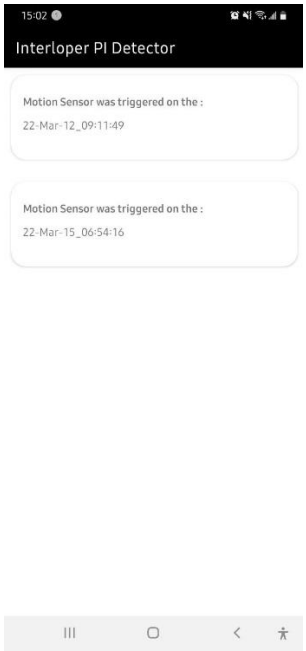

Figure 14 History of Motion Detected Date&Time Stamps Screen

- History of Motion Detected Photo Capture History Screen

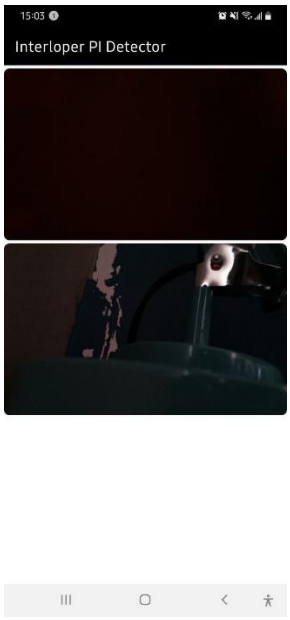

Figure 15 History of Motion Detected Images Screen

Appendix D: Website Dashboard [Page 12]

- Login Screen

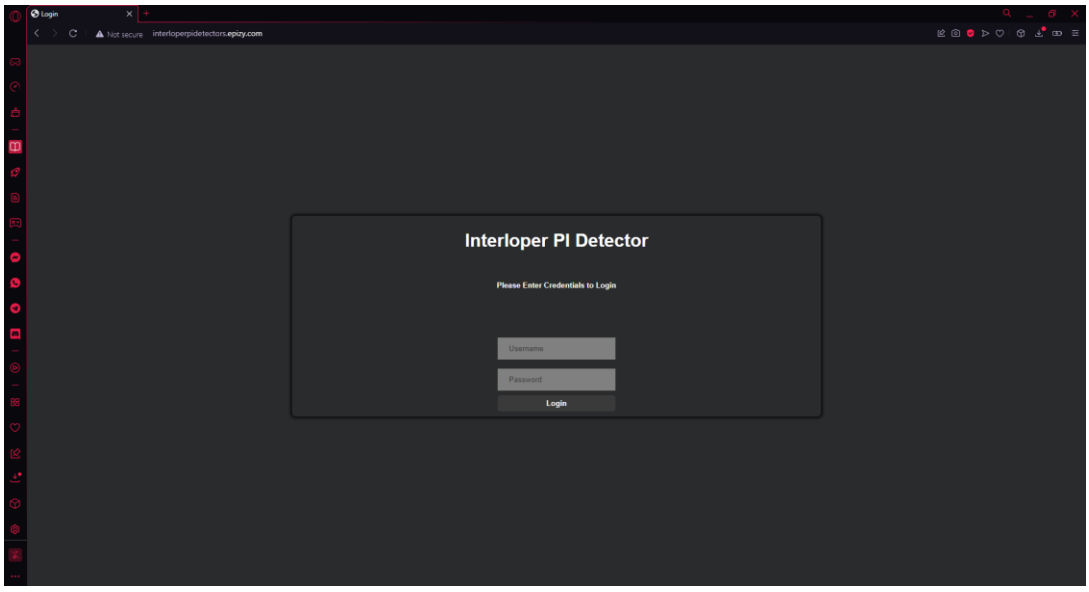

Figure 16 Login Screen

- Dashboard Screen

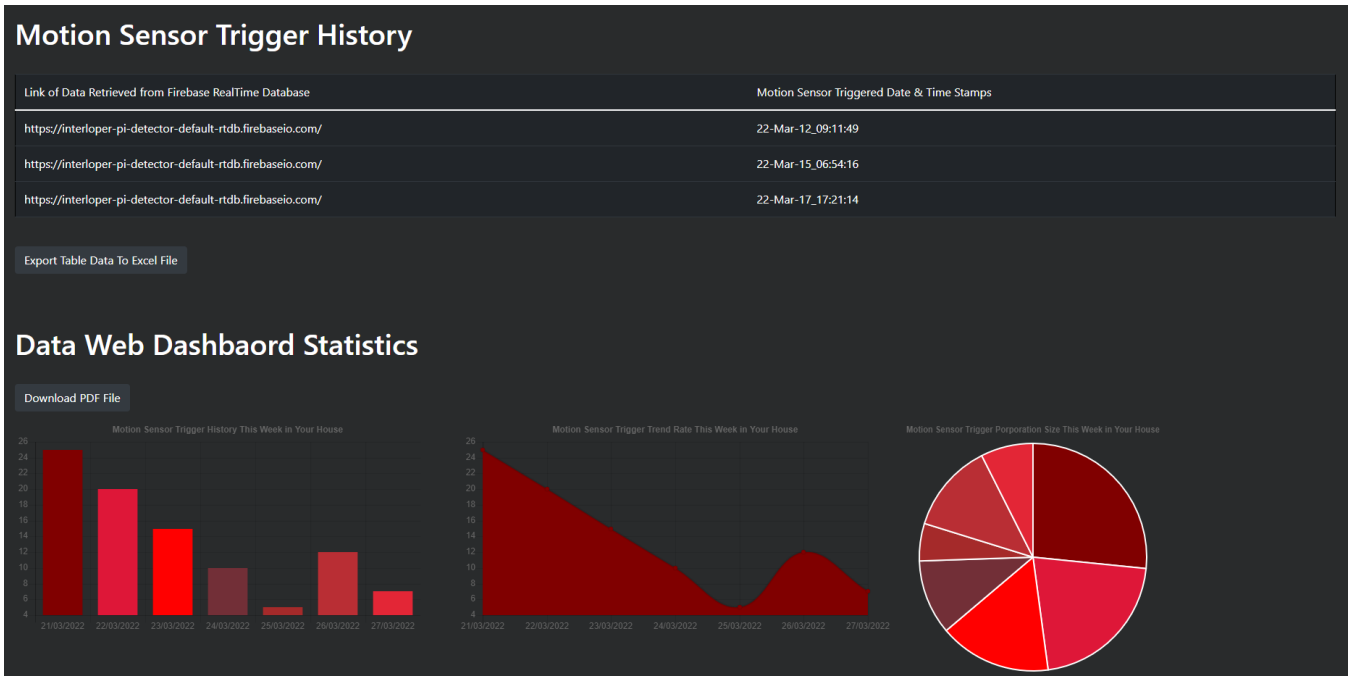

Figure 17 Dashboard Screen (Note that the Charts are just Dummy Data)

- Exported Excel Page Screen

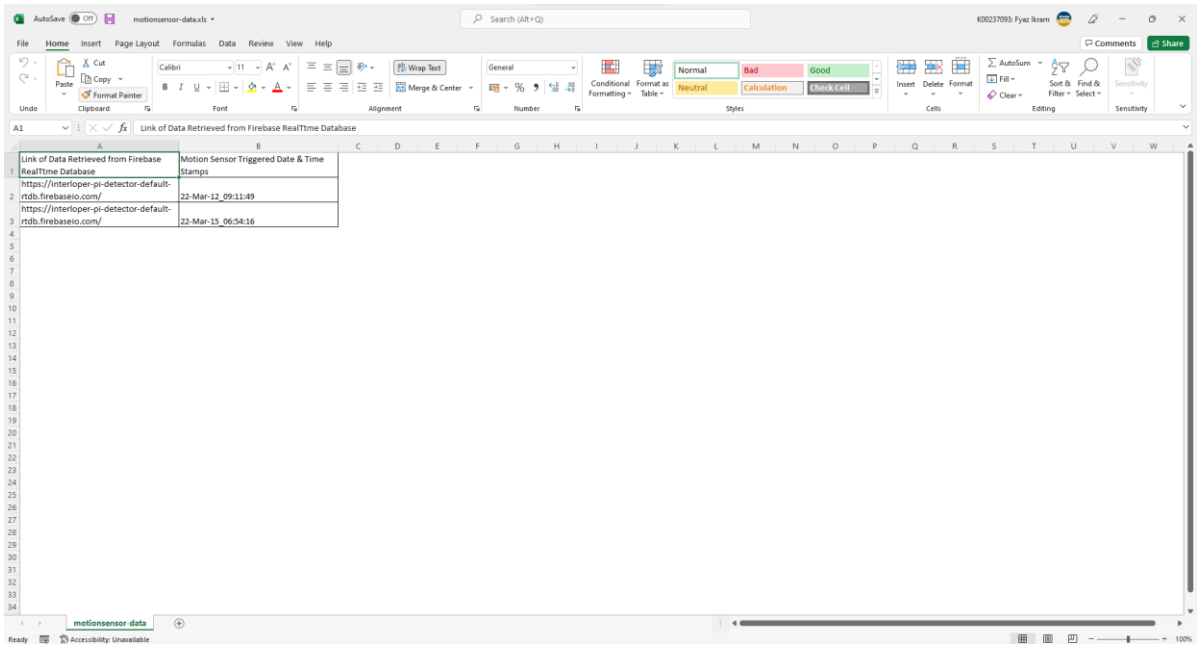

Figure 18 Exported Excel Screen

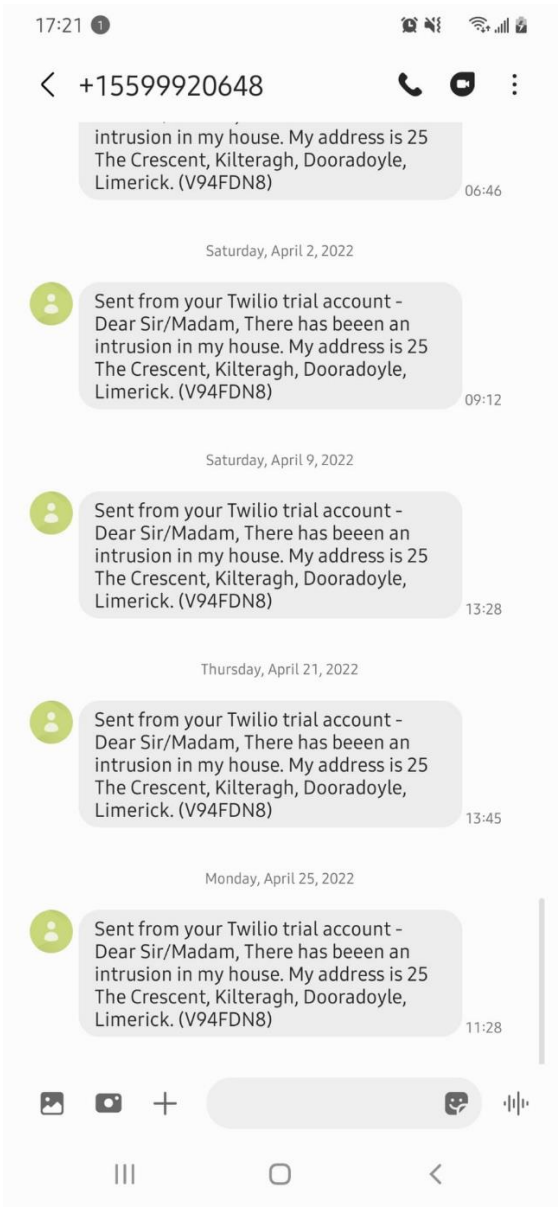

Figure 19 Garda Message notifying that a House has been Intruded