EEE 495

Electrical and Electronics Engineering Design II

Project Title: Multifunctional Smart Home using IoT Technology

Final Report

Team 4

Ahsan Mehmood - 21903575

Fahad Waseem Butt - 21801356

Maaz Ud Din - 21901258

Payam Sedighiani - 21801298

10 May 2023

TABLE OF CONTENTS

1.0 EXECUTIVE SUMMARY

1.1 Problem Considered

With the rise of automation and technological advances, smart homes are an example of Internet of Things (IoT) technology that is growing exceedingly popular in the modern day. It is an example of IoT in that it is centred around devices working in specific applications in a home [1]. The devices are connected to the internet and are connected to each other, and give the user certain information or services based on the application, and this is usually tied together through a central controlling system as a mobile app where the user can control the specifics of how they want the smart home to work [1]. The smart home systems in the modern age are growing increasingly intelligent, and can be built with the capacity to make decisions and learn about the user's habits through patterns [1].

Automation results in the luxury of saving time or improving the lifestyle of the persons benefiting from the technology. As such, smart homes are a technology that suits a wide range of audiences, with there being quite a lot of use of it possible in especially domestic environments, but also in some corporate settings [2]. There is also additional security provided when utilising what modern technology has to offer.

1.2 Proposed Solution

The goal of this project was to solve a problem with the use of the Internet of Things (IoT) methodology with multiple devices communicating with each other through the Internet to accomplish a bigger task. The Smart Home was intended to comprise several systems working at their unique tasks to form the IoT technology. A Door Cam (Door Camera) system was to be implemented which was to work to recognise faces of people entering the house. The people living in the house were to have their faces registered and a face identification method was to let them in with no issues. When an unwelcome visitor was identified, the owner was to immediately be sent a notification and a phone call. On the other hand, if unknown guests showed up, the owner was to be sent a notification, and could check a video feed of the camera so as to decide to let the visitors inside or not. A Gate Cam (Gate Camera) system was to be present that was to have the owner's licence plate number registered and would be welcomed in. If an unknown car showed up, then the owner was to be sent a notification on a smartphone app and a phone call. The lights in the house were to work with an automation feature as when sensors detected low light levels, the lights would turn on, as well as having a time based activation. With all of the systems mentioned prior being connected to the Internet, they will all be accessible through a Mobile App, which will act as a hub. All features such as activating and deactivating the systems in the house, registering faces and licence plates, and even manually turning the lights on and off [2].

1.3 Achieved Results

All key functional and nonfunctional requirements of the product were met in the project, this includes the Door Camera, Gate Camera and Automatic Lights Systems, as well as the Mobile Application. Most of the additional requirements were also met, with the exception of some such as the live feed from the cameras to the mobile application. In this implementation of the product, the key limitation was that the main controller, the Raspberry Pi 3 Model B+. Because the processing power of the controller unit was the limiting factor, implementations such as the live feed from the cameras were not feasible to implement in addition with all the other computation-heavy systems which involved reading and identification using the cameras.

2.0 PRODUCT TECHNICAL SPECIFICATIONS

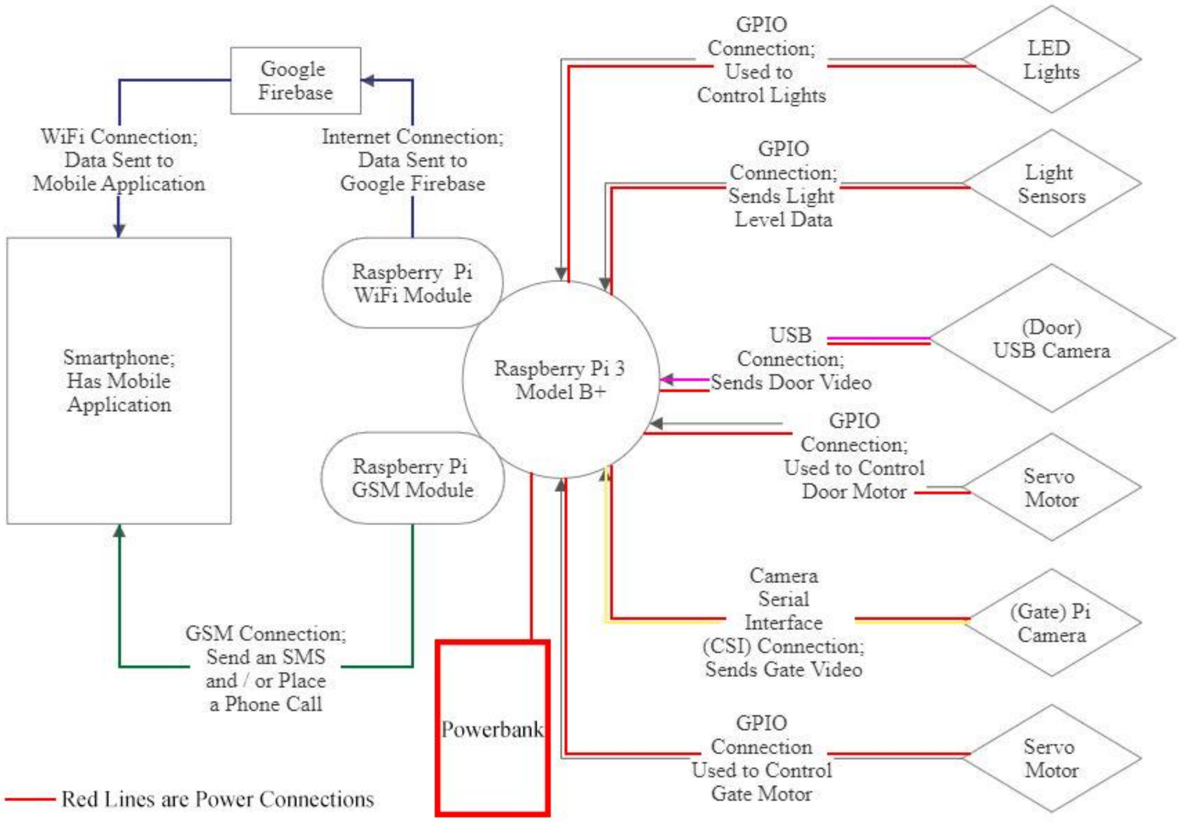
The product had a multitude of features all accessible by the user through the smartphone application [3]. The features were as follows:

- A Door Cam (Door Camera) System
  - A camera will recognise faces of people entering the house.
  - The people living in the house will have their faces registered and a face identification method will let them in with no issues, done so by the servo motor.
  - When an unwelcome visitor is identified, the owner is immediately sent a notification and a phone call.
  - If unknown guests show up, the owner is given a notification, and can decide to let the visitors in or not through manually opening the door with the mobile application.
- A Gate Cam (Gate Camera) System
  - A camera will recognise licence plates of cars entering the house.
  - The cars owned by people living in the house will have their licence plates registered and a licence plate identification method will let them in with no issues, done so by the servo motor.
  - When an unwelcome car is identified, the owner is immediately sent a notification and a phone call.
  - If unknown cars show up, the owner is given a notification, and can decide to let the cars in or not through manually opening the gate with the mobile application.
- Automatic Lights
  - A light in the house will work with an automation feature as when sensors detect low light levels, the lights will turn on.
  - Times can be manually set in the app to have a light turn on and off.
  - The smartphone app can also be used to turn both the lights off and on.
- Smartphone Application
  - The hub for all systems in the smart home.
  - It will be able to activate and deactivate the systems in the house.
  - The app can be used to register faces and licence plates.

3.0 PRODUCT DESIGN

3.1 Hardware Design

A power bank would power the system and all the components will be powered by the Raspberry Pi 3 Model B+. The information in this section is referenced from the High Level Design Document [4], with references being placed in this document again where necessary.



3.1.1 *Light Sensor Hardware*

4 PIN LDR sensors are used for the automatic lighting system. The light sensors detect the amount of light in the room and if the level of light falls below a certain amount then the lights are turned off. GPIO pins of the raspberry PI are used to connect the light sensor [7]. A python script is used to control the lights and the sensor. A capacitor is connected to the circuit so that the resistance of the LDR can be measured and used to turn the lights on or turn them off.

3.1.2 *Cameras Hardware*

In the design for IoT smart house, there are two cameras and since the Raspberry Pi board has only one CSI port, one Pi camera is used for the door camera and for the gate camera a USB camera is set. For the USB Camera, one of the USB ports of the raspberry pi module will be used. And the steps are as the following:

- Turn off the raspberry pi module
- Insert the USB camera in one of the USB ports
- Make sure the camera is properly inserted in the USB port

After inserting the USB camera properly, the camera can be used by following the steps below [R3.1.7]:

1. First download the “fswebcam” package
2. Using bash terminal or python write 1susb in the terminal

The Pi Camera can be used in a similar fashion when connected to the CSI port.

3.1.3 Servo Motor Hardware

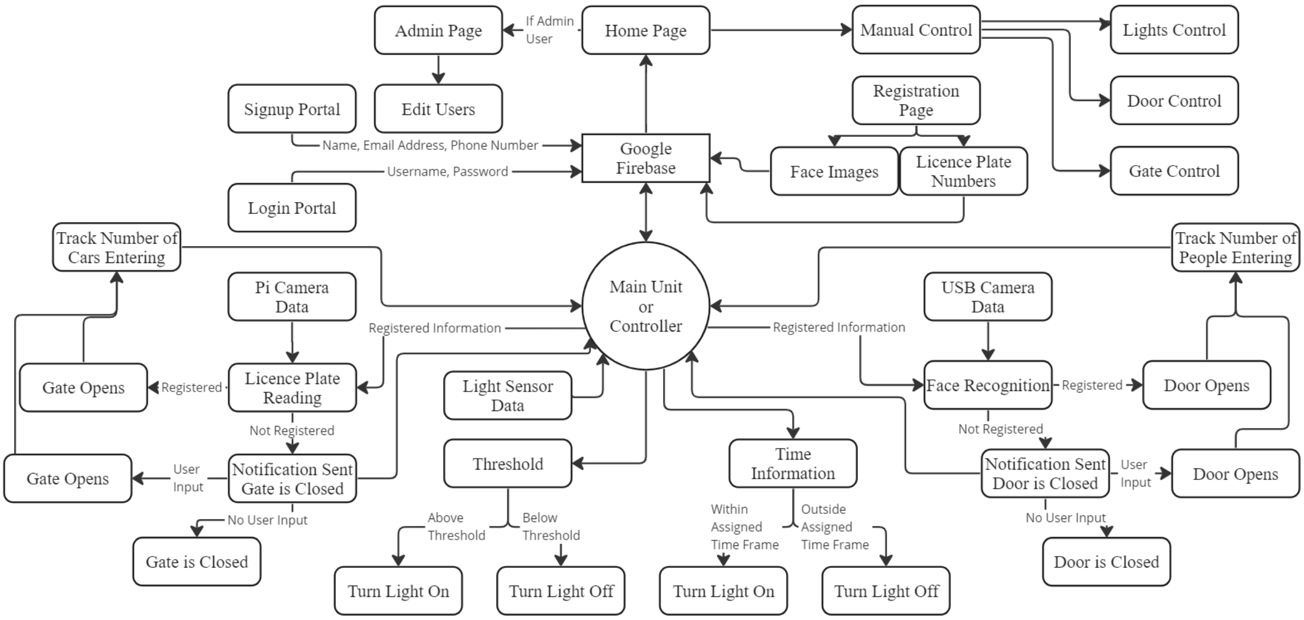
The MG90S Servo Motor is used for the functioning of the door. If the Door Camera sees a registered face then the door is opened using the motion of the servo motor. Similarly, for the Gate Camera, the MG90S servo motor is used to open the house gate. Male to Female wires are used to connect the servo motor with the microprocessor. PWM signals are sent from one GPIO pin to the servo motor. The servo motor is a DC motor which rotates itself upon getting different frequency signals. The electrical signal sent to the motor is called Pulse Width Modulation (PWM). We just send an ON signal for a certain amount of time and then an off signal for a certain amount of time. The length of time for which the signal is on defines the angle the servo motor will rotate. The servo motor connected to Raspberry PI can be seen below. The following image was found online and is referenced [8]. A python script was used to send a signal to the Raspberry PI to control the servo motor. The GPIO module was used for this purpose.

3.1.4 Mobile Application Hardware

Basic Android devices can be used for using the software. The mobile application needs to be installed and access to the internet is required using WiFi or Mobile Data for the software to work and interact with firebase and other working systems.

3.2 Software Design

The information in this section is referenced from the High Level Design Document [4], with references being placed in this document again where necessary.



3.2.1 OS Setup

The Raspberry Pi is set up by following the steps below:

- Raspberry OS will be installed on a microSD card using the Raspberry Pi imager.
- The MicroSD card will be inserted into the Raspberry Pi.
- The Raspberry Pi will be connected to the laptop with an ethernet cable.
- To use the laptop as a screen for the Raspberry Pi, PuTTY, a terminal emulator will be used with the ssh protocol for the ethernet port.
- VNC Viewer, a controller to remotely operate a computer, will be used to operate on the Raspberry Pi.

3.2.2 Automatic Lights System Software

The controller reads light intensity as the input data from the LDR sensor. Based on the threshold set in the code, the controller decides whether to turn the light on or off. Output is then sent to the GPIO pins according to this decision. For the time dependent light, a timeframe is set in the code and the clock of the Raspberry Pi is taken as an input using the time library in python. It is checked whether the time of the day is within the set timeframe, which is set manually. If it is, then the light turns on, otherwise it stays off.

3.2.3 Door Camera System Software

For the face recognition algorithm to be used and interfaced with Raspberry Pi 3 Model B+ some setups and libraries are necessary. All of the algorithms in this part are implemented using Python 3. The first step is getting access to the gathered images from the users which are saved in the database.

The used algorithm is a mixture of multiple different algorithms to get the most accurate results considering the speed of FPS. The general flowchart of the entire system's algorithm is given in the above figure. There are two main starting parts. First the data that is used for training is processed and then the other data which is the real time data for real time face recognition. Both of the data are going through different and similar steps.

After the images are extracted from the database, Haar_cascade library [9] is used to detect faces and extract only the face contours which leads to a higher accuracy and prevents extra data.

After the face images are extracted, we need to again detect the faces in the images to extract their embeddings and vectorize them. In this part, the MTCNN (MultiTask Cascaded Convolutional Neural Network) algorithm is used for the detection of the faces to get the best vector values. This choice is made because the MTCNN algorithm is one of the most accurate and robust algorithms for face detection [10].

In this step all of the detected faces will be encoded using FaceNet. Each vector is called an embedded face. The FaceNet algorithm extracts 128 important features of the face as a vector. Basically, each face's important features are embedded in a vector [11].

Lastly, the SVM(Support Vector Machine) model, which is a very powerful supervised learning model for classification(SVC) and comparison between different classes, is used to classify faces' embeddings derived from previous steps and we can think of each person's face as a different class [12].

After the classification of each face is done, each real time frame is captured and the faces are detected by MTCNN which was explained previously. Then, after vectorizing the faces, if we get a face vector we compare it with the classified face vectors in the training part(and also the unknown class which consist of random faces of different people) If they are similar with the given threshold then the person is recognized, if not then the person is unknown.

Since in all of the steps of face recognition many computational blocks take place, in the real time there will be a speed issue if no FPS improving techniques are used. One of the techniques used is to only do face recognition only for the frames where the faces are detected and pass a few frames after you detect a frame without computing any of the models. This will cause a lot of improvements. Another technique is to start and stop the capturing of data at the same time with the above technique so we prevent the calculations and computations of extra frames.

3.2.4 Gate Camera System Software

For the licence plate reading method, it is all done in Python 3, some setup and installation of libraries is necessary. For the reading method, the following steps are undertaken:

- The Pi Camera is initialised with a continuous capture.
- A Bilateral Filter is applied to the image on each frame of the capture.

- Then the Canny Edge Method is used on the filtered frame to detect edges so as to find the licence plate.
- The Pytesseract library [13] is used to read and store the detected characters of the licence plate, reading the result of the frame after the Canny Edge Method.
- The characters of the licence plate read are compared with those stored in the database, and based on a match, the method decides if the car is known or unknown.

3.2.5 Mobile Application Software

React Native is used to set up and program the mobile application which is a framework of JavaScript Programming Language. Integration with Firebase is also done using React Native and JavaScript.

4.0 LIST OF COMPONENTS AND COSTS

The following table contains the components used and/or purchased for this implementation and their costs. Some personal items were used as components so their cost is not included as the market price for those was different when they were purchased for personal use.

Component	**Cost (TL)**
Raspberry Pi 3 Model B+	3,163.58
Pi Camera	225.97
Servo Motors x2	162.70
LDR (Light Sensor) x3	43.39
SMD LED (1) x3	1.36
SMD LED (2) x3	0.75
Demo House	211.90
USB Camera	None (Personal Item Used)
Powerbank	None (Personal Item Used)
LED Lights	None (Used From Lab)

5.0 TASK DISTRIBUTION

Ahsan Mehmood:

- Mobile Application.

- Setting up Google Firebase for the product.

Fahad Waseem Butt:

- Gate Camera.
- Implementation of licence plate reading method.

Maaz Ud Din:

- Automatic Lights.
- Implementation of automatic and manual lights method.

Payam Sedighiani:

- Door Camera.
- Implementation of face identification method.

6.0 PROFESSIONAL AND ETHICAL ISSUES

6.1 Software Design

When considering the software design of the project i.e. the coding aspect of it, the following libraries/modules were used in Python 3, as with labels for being usable as a product:

- Tensorflow
- Matplotlib.pyplot
- Cv2
- Os
- Numpy
- Zlib
- Mtcnn.mtcnn
- Keras_facenet
- Sklearn
- Pickle
- Firebase_admin
- Urllib
- PIL
- Pytesseract
- Imutils
- Difflib
- Pyrebase
- Base64
- Gpiozero
- Time
- Requests
- Datetime
- Twilio.rest

- RPi.GPIO
- Flask
- Flask_cors
- Socket
- Subprocess
- Future
- Argparse

And for the mobile application, the React framework in JavaScript was used, along with open source CSS packages.

Of these, most of the libraries/modules are open source and have no licensing issues. The following are libraries/modules that have conditions in commercial use as compared to use in a single university project:

- Twilio: It requires that the distributor (direct customer) of the product purchases twilio, and both the direct customer and third party users (users of the product) follow twilio's terms of service [14].
- Firebase_admin: "Redistribution and use in source and binary forms, with or without modification, are permitted provided that" the conditions listed in the licence are met [15].
- Gpiozero: "Redistribution and use in source and binary forms, with or without modification, are permitted provided that" the conditions listed in the licence are met [16].

Aside from this, Google Firebase has a pricing plan [17] when considering the scale it would be used, as well as terms of service to follow [18], when not in the context of a university project.

6.2 Hardware Design

The parts (subassemblies) of the product are listed in the section "4.0 LIST OF COMPONENTS AND COSTS" where it can be noted that almost all of the components are hardware that can be purchased and is free to use in any context, be it academic or commercial without any licencing.

However, the controller unit, the Raspberry Pi 3 Model B+ itself requires a licence from Raspberry Pi. Since the product is heavily reliant on a Raspberry Pi board, any sales would need to be made with an official and legal model of the board, and for this purpose, the Raspberry Pi brand and logo would also need to be used. A standard, non-transferable licence may be obtained from Raspberry Pi for this purpose [19]. In addition, for the sale of the product in the context that is not of a university product, the licensing terms [20] need to be followed.

7.0 CONCLUSIONS

In the working of this project, an organised procedure was followed with the steps taken being documents to cross verify that the working of the project was going as intended from the beginning. The documentation being:

- Project Proposal
- Product Specifications Document
- High Level Design Document
- Test Plan
- Final Report.

The project proposal was the initial conception of the idea that was shaped and developed to make into a working product at the end. The product specifications document was written to narrow down how the subsystems of the product were intended to work and determine a direction of the functionalities to be expected at the end. The high level design document gave a look at the hardware and software designs used in the project such that the key components were identified and showed how they implemented the specifications. The test plan was, in essence, a checklist formed to see which of the specifications laid out in the previous document were followed through on, which specifications had an acceptable rate of accomplishment, and which specifications were not met (or not met to an acceptable standard). And finally, this final report document is a summation of everything accomplished in the project. The working methodology followed an approach where the idea was conceived at a high level, with the work going to lower and more complex levels until the product was designed. Then the lower level, complex subcomponents were methodically reassembled back to a high level and tested at each stage. This was further facilitated by how the course had a progress check system with the Teaching Assistant present to take almost weekly checks on what meaningful updates were implemented into the product's construction.

The methodology seems to be similar to how products are conceived as ideas in the industry, divided into subprojects (or smaller tasks) to be divided for different teams to work on, and eventually, the results made from each team are joined together to shape into the product that the customer or investor had initially conceived of as an idea.

When it comes to accomplishment, the majority of the declared and assigned goals in this project were accomplished, with the exception of just a few, such as the live video feed being accessible from the smartphone application. The product itself was rather slow, as when the systems were run on a standard laptop, there was a very significant speed increase. But given how the Raspberry Pi 3 Model B+ is a very old board with limited processing power, even being able to run all the proposed systems on it, simultaneously at that, and with an almost acceptable speed to live implementations is a satisfactory accomplishment. Naturally, due to the nature of the project and how there was a limited timeframe to accomplish it, there was room for improvement. The most significant factor that would help improve the product would be the use of a higher end controller unit, with more processing power, which would be available to use with a higher budget.

As a product, the Smart Home System has potential to actually be a suitable product to be sold on the market. With the knowledge obtained thus far, the pricing might be high, but there may be more methods in the industry to reduce the cost per item to be sold. Aside from that the functionality, especially considering the potential from a faster controller unit, seems to be highly acceptable. The mobile application may also be hosted online on stores such as the Google Play Store.

The project was primarily a culmination of all the knowledge obtained through courses and through an understanding of how to learn, as obtained from the university. As for new topics, those included specific technologies such as React, Twilio and the like. The approach to acquiring new knowledge was based on what functionality was needed. As the project advanced, the progression was based on covering the specifications and functionality of the product, and online research was conducted to identify and learn what methods would best assist in the implementation of the goals.

In conclusion, this project allowed for learning a structured approach to planning, designing and implementing a product that has potential to be used in the industry. Furthermore, to accomplish this, new technologies needed to be learnt and it was an interesting experience to research and understand how to implement them.

8.0 REFERENCES

[1] "Internet of Things vs. Smart Home - What's the difference?" *Livy.* [Online]. Available: https://livy-protect.com/en/iot-vs-smart-home/ [Accessed: 14 February, 2023]

[2] Team 4. "Multifunctional Smart Home using IoT Technology: Project Proposal." *Bilkent University EEE 495*, February 2023. [Online] Available: https://moodle.bilkent.edu.tr/2022-2023-spring

[3] Team 4. "Multifunctional Smart Home using IoT Technology: Product Specifications Document." *Bilkent University EEE 495*, March 2023. [Online] Available: https://moodle.bilkent.edu.tr/2022-2023-spring

[4] Team 4. "Multifunctional Smart Home using IoT Technology: High Level Design Document." *Bilkent University EEE 495*, April 2023. [Online] Available: https://moodle.bilkent.edu.tr/2022-2023-spring

[5] "Raspberry Pi Light Sensor using an LDR." *PiMyLifeUp*, October 2022. [Online]. Available: https://pimylifeup.com/raspberry-pi-light-sensor/ [Accessed: 15 April, 2023]

[7] "Working with USB webcams on your Raspberry Pi." *The Raspberry Pi Guide*, 2021. [Online]. Available: https://raspberrypi-guide.github.io/electronics/using-usb-webcams [Accessed: 13 April, 2023]

[8] "Servo Motor Control With Raspberry Pi." [Online]. *Instructables*. Available: https://www.instructables.com/Servo-Motor-Control-With-Raspberry-Pi/ [Accessed: 14 April, 2023]

[9] "Face Detection with Haar Cascade." *Towards Data Science*, December 2020. [Online]. Available: https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08 [Accessed: 19 April, 2023]

[10] "Robust face detection with MTCNN." *Towards Data Science*, February 2021. [Online]. Available: https://towardsdatascience.com/robust-face-detection-with-mtcnn-400fa81adc2e [Accessed: 19 April, 2023]

[11] "Face Recognition with FaceNet and MTCNN." *Arsfutura*. [Online]. Available: https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/ [Accessed: 20 April, 2023]

[12] "Raspberry Pi Camera Options and Usage." *Springer Link*, June 2019. [Online]. Available: https://link.springer.com/article/10.1007/s42979-019-0006-z [Accessed: 20 April, 2023]

[13] "Tesseract OCR: What is it, and why would you choose it in 2023?" *Klippa*, October 2022. [Online]. Available: https://www.klippa.com/en/blog/information/tesseract-ocr/ [Accessed: 20 April, 2023]

[14] “Twilio Terms of Service.” *Twilio*, February 2023. [Online]. Available: https://www.twilio.com/en-us/legal/tos [Accessed: 6 June, 2023]

[15] “firebase_admin/LICENSE.” *Github*, 2020. [Online]. Available: https://github.com/appsup-dart/firebase_admin/blob/master/LICENSE [Accessed: 6 June, 2023]

[16] “License.” *GPIO Zero*, 2020. [Online]. Available: https://gpiozero.readthedocs.io/en/stable/license.html [Accessed: 6 June, 2023]

[17] “Firebase Pricing.” *Firebase*. [Online]. Available: https://firebase.google.com/pricing [Accessed: 7 June, 2023]

[18] “Terms of Service for Firebase Services.” *Firebase*. [Online]. Available: https://firebase.google.com/terms [Accessed: 7 June, 2023]

[19] “Raspberry Pi trademark rules and brand guidelines.” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.com/trademark-rules/ [Accessed: 8 June, 2023]

[20] “Licensing.” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.com/licensing/ [Accessed: 8 June, 2023]