EEE 495

Electrical and Electronics Engineering Design II

Project Title: Multifunctional Smart Home using IoT Technology

High Level Design Document

Team 4

Ahsan Mehmood - 21903575

Fahad Waseem Butt - 21801356

Maaz Ud Din - 21901258

Payam Sedighiani - 21801298

23 April 2023

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Purpose of the Document

The aim of this document is to provide a comprehensive collection of details regarding the product; this includes an overview of the product as well as its workings, functionality, features, technical information, hardware and software information.

1.2 Product Definition

The aim to be accomplished by this product is to develop a smart home that provides multiple features for the sake of both security and quality of life improvement, allowing the residents with more time to spend more productively elsewhere.

1.3 Audience for the Document

This main target audience this document is intended for is engineers who are involved in understanding the technical workings of the product, along with all its functionality. It explains the functional and nonfunctional requirements for the system, as well as the architecture of the software and hardware designs used, so that engineers may gain insight into the specific details involving the workings of the product.

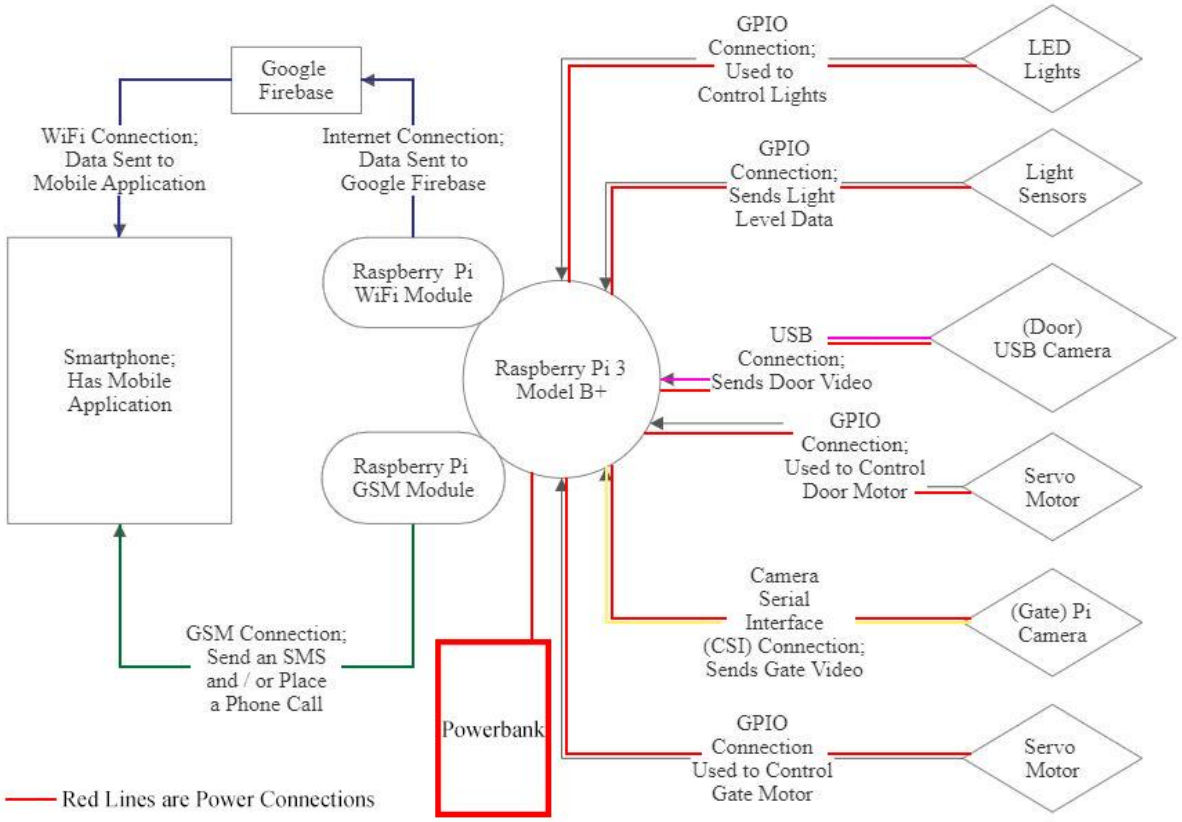
1.4 Overview of the Product

The main customer base for this product are homeowners, or people living in a rented home; hence the main application of the product is in domestic use. The users are required to have a stable internet connection to be able to use the product properly. The application itself will have a login system, so the user will need to make an account with a username and password to be able to use it for the smart home. Furthermore, for both the door camera and gate camera systems, the user will need to register to the system the faces of the people living in the house and the licence plates of the cars they use respectively. The automatic lights will need to be manually turned on from the smartphone application for them to have functionality as intended. The user will also need to register a phone number to which calls will be placed in emergency situations. The goal of the product is to provide the user with both security and comfort in the sense that there is time saved using these automated systems to be used productively elsewhere.

2.0 HARDWARE DESIGN

2.1 Hardware Block Diagram

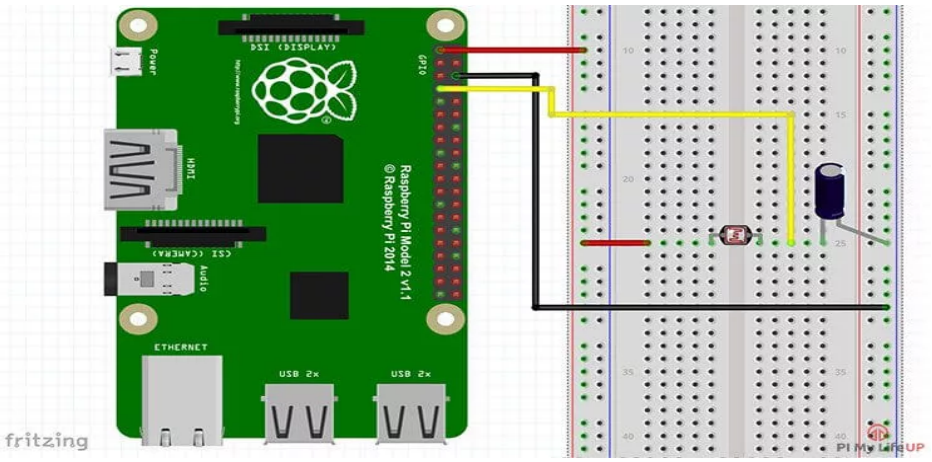
A power bank would power the system and all the components will be powered by the Raspberry Pi 3 Model B+.



2.2 Automatic Lights System Hardware

2.2.1 Light Sensors

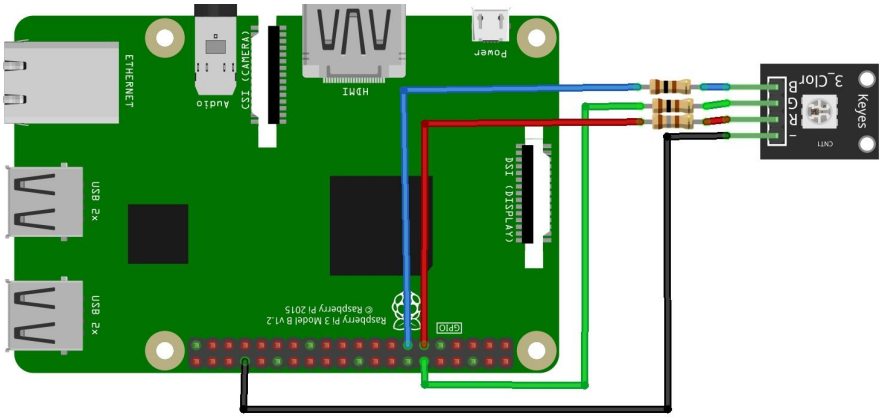
4 PIN LDR sensors are used for the automatic lighting system. The light sensors detect the amount of light in the room and if the level of light falls below a certain amount then the lights are turned off. GPIO pins of the raspberry PI are used to connect the light sensor. The connections that are required can be seen below. The following image was found online and is referenced [1].



We use a python script to control the lights and the sensor. A capacitor is connected to the circuit so that we can measure the resistance of the LDR and use that to turn the lights on or turn them off.

2.2.2 SMD LEDs

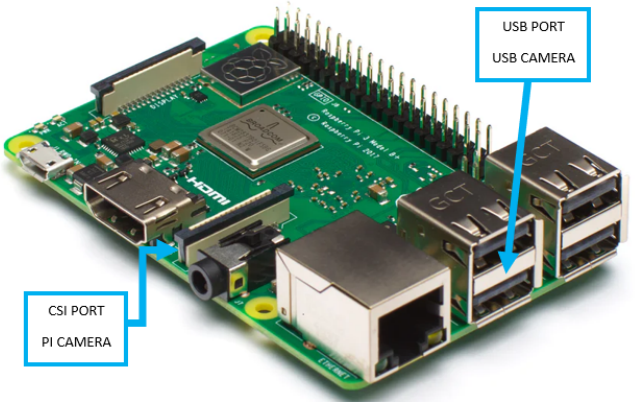
Surface mounted device (SMD) LED 2835 will be used in the lighting system, in front of the door camera and the gate camera. SMDs will be used because of their versatility, efficiency, size and ease of use. Due to their small size, it will be perfect for the scale of the product and will be easier to install in the smart home. As the camera modules need ample light for facial and licence plate recognition, the lights will be used there as well for better illumination at night. A general setup is in the following figure. The following image was found online and is referenced [2].



Here, limiting resistors will be used at the RGB terminals to avoid burnout due overvoltage. A python script will be written to control these LED lights in the light system. This python script will consider a timeframe to turn on the lights. Additionally, some lights will also depend on the user's command and the light sensors in the automatic light system of the smart home, to be turned on and off according to the light intensity recorded by the sensors [3].

2.3 Camera Modules

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 gate camera and for the door camera a USB camera is going to be set. The following image was found online and is referenced [4].



2.4 Door Camera System Hardware

2.4.1 USB Camera

For the USB Camera, one of the USB ports of the raspberry pi module will be used. And the steps are as the following:

1. Turn off the raspberry pi module
2. Insert the USB camera in one of the USB ports
3. Make sure the camera is properly inserted in the USB port

The following image was found online and is referenced [5].

After inserting the pi camera properly, the camera can be used by following the steps below:

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

The following image was found online and is referenced [6].

```
(main) pi@jolpi146:~ $ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 046d:0825 Logitech, Inc. Webcam C270
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

If the above result is seen then the USB Camera is ready.

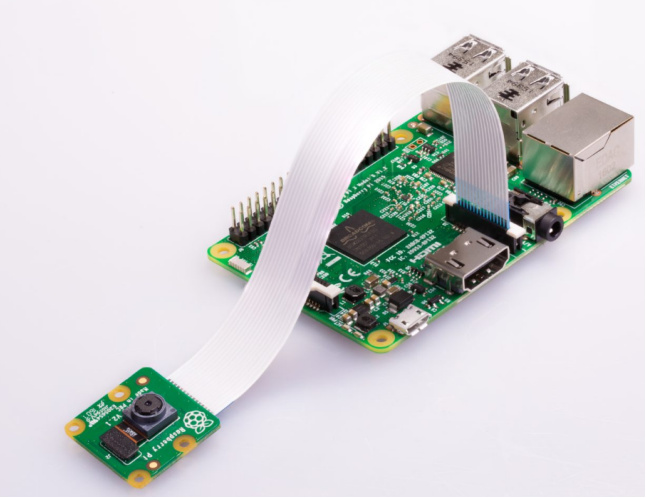
2.5 Gate Camera System Hardware

2.5.1 Pi Camera

The configuration steps for the Pi Camera are as the following [7]:

1. Turn off the raspberry pi module
2. Insert the pi camera in the CSI port by pulling the clip
3. Make sure the ribbon cable is set properly in the CSI port
4. Put the plastic clip back where it was initially

The following image was found online and is referenced [8].

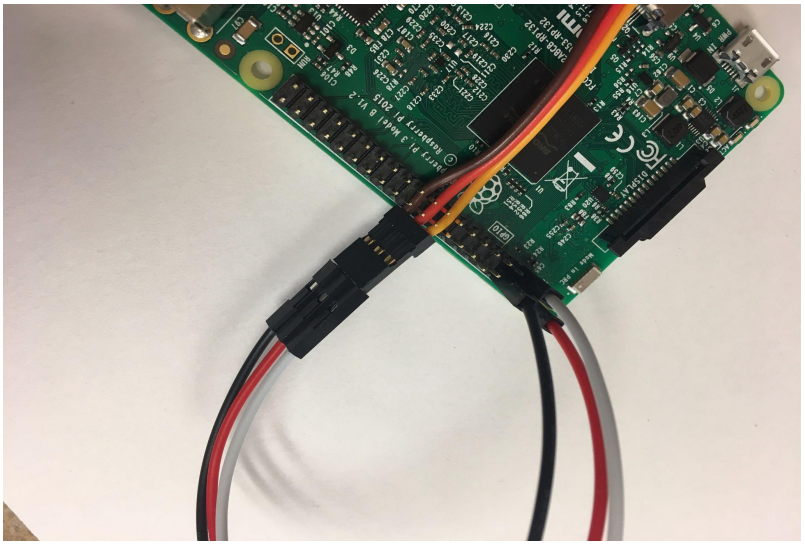


After inserting the pi camera properly, the camera can be used by following the steps below:

1. Go to homepage and then go to preferences
2. Click on setup configurations
3. Go to interfaces and enable the camera option

2.6 Servo Motor for Door and Gate Systems

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 [9].



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.

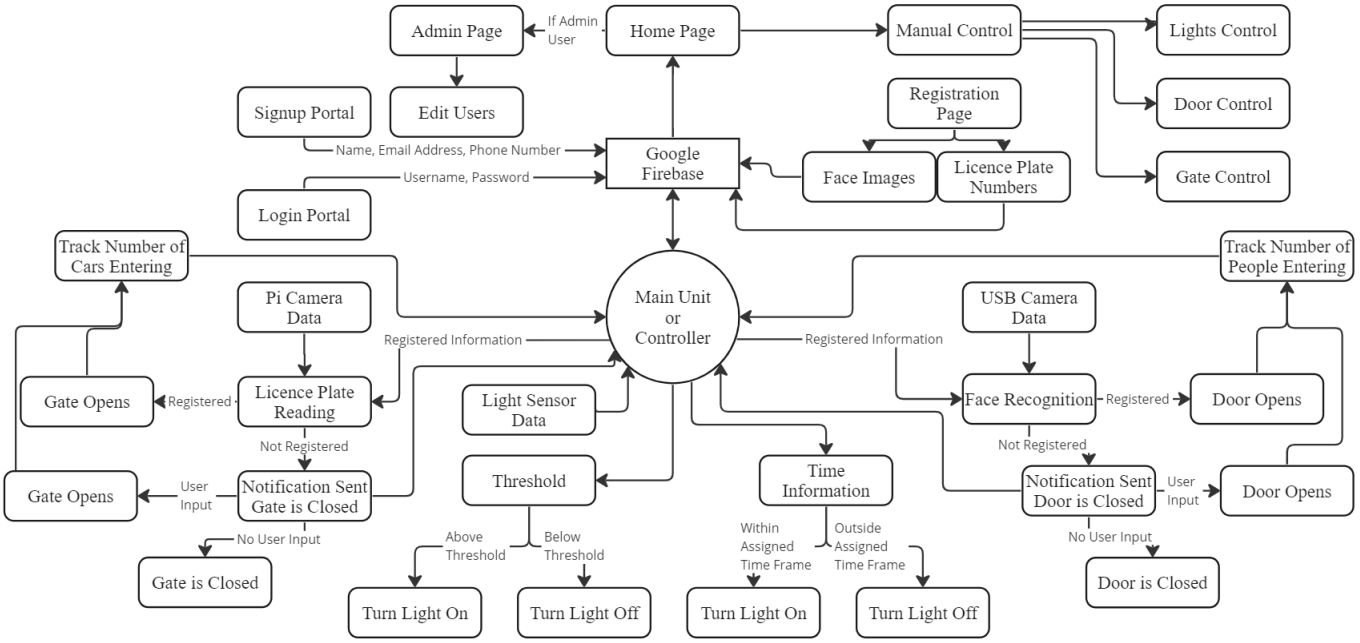
2.7 Mobile Application Hardware

2.7.1 Smartphone

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.

3.0 SOFTWARE DESIGN

3.1 Software Block Diagram



3.2 OS Setup

The raspberry pi will be set up by following the steps below:

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

3.3 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.

3.4 Door Camera System Software

3.4.1 Face Recognition with USB Camera

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. For getting access to these images the following libraries are used:

1. OS
2. CV2
3. Numpy

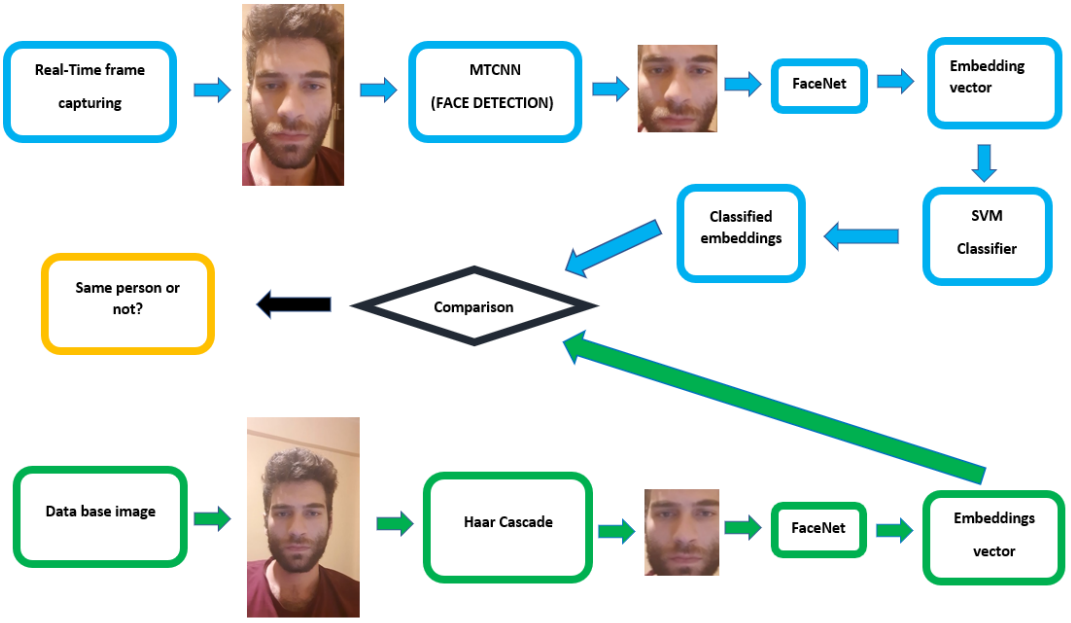
Then for the training part the following libraries are necessary to be downloaded:

1. Tensorflow
2. Matplotlib
3. CV2
4. Zlib
5. MTCNN
6. FaceNet
7. Sklearn
8. Pickle

Next, for the real time detection the following libraries are needed to be downloaded:

1. Time
2. Imutils
3. Argparse

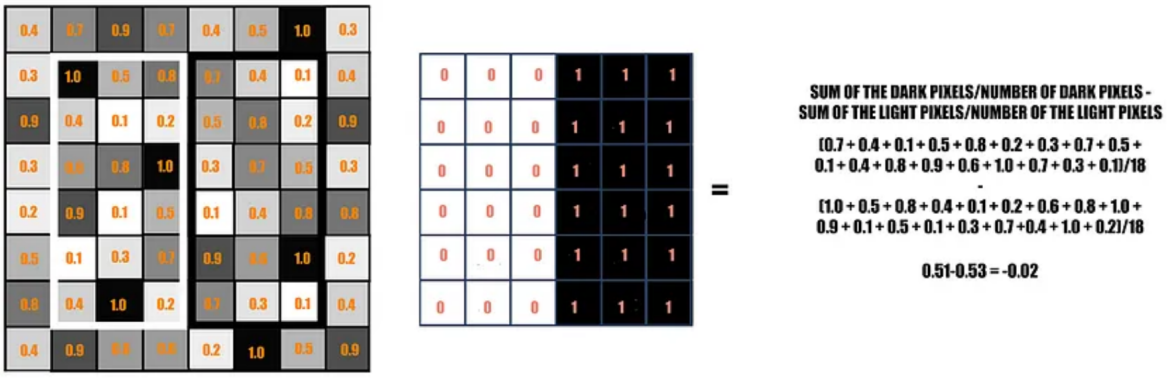
3.4.2 Face Recognition Algorithm



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. The following sections are the steps taken for achieving the facial recognition algorithm.

3.4.2.1 Detecting Faces

After the images are extracted from the database, Haar_cascade library is used to detect faces and extract only the face contours which leads to a higher accuracy and prevents extra data. The following image is referenced [10]



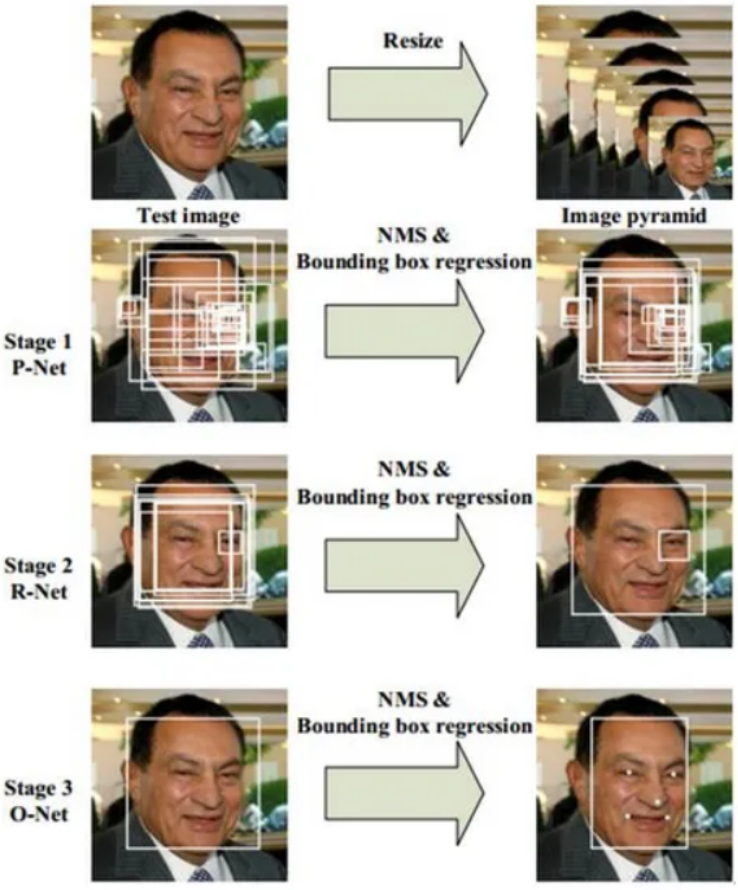
The above image shows how Haar features are extracted from an image. In the first rectangle from left, which represents an image, each pixel has a value between 0.0 to 1.0. The middle white and black rectangle is a Haar kernel which takes all the light pixels to the white

side and all the dark pixels to the black side. Lastly, in the most left part of the above image, the formula for Haar calculation is given in which it is calculated by taking the difference of the average of the dark pixels from the average of the light pixels in their corresponding region.

Finally, if the Haar value is close to 1 then there is an edge detected using the Haar feature [10].

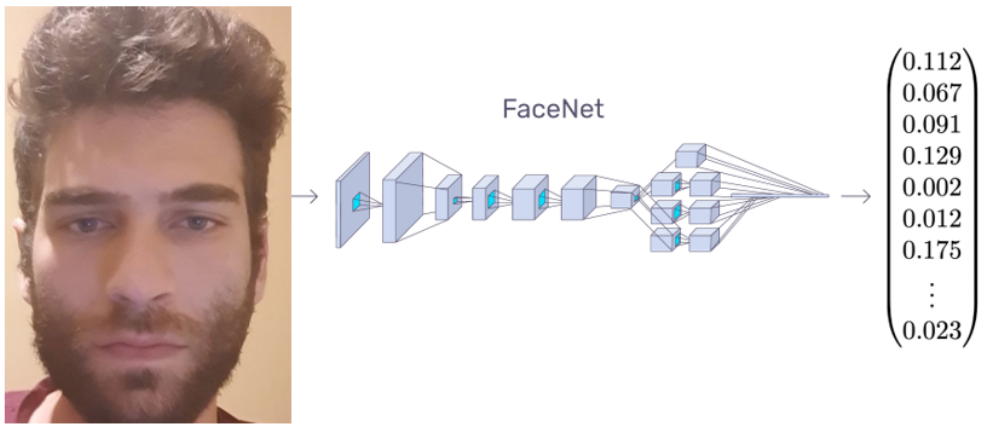
3.4.2.2 Vectorizing Faces and Labelling

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. The following image is referenced [11].



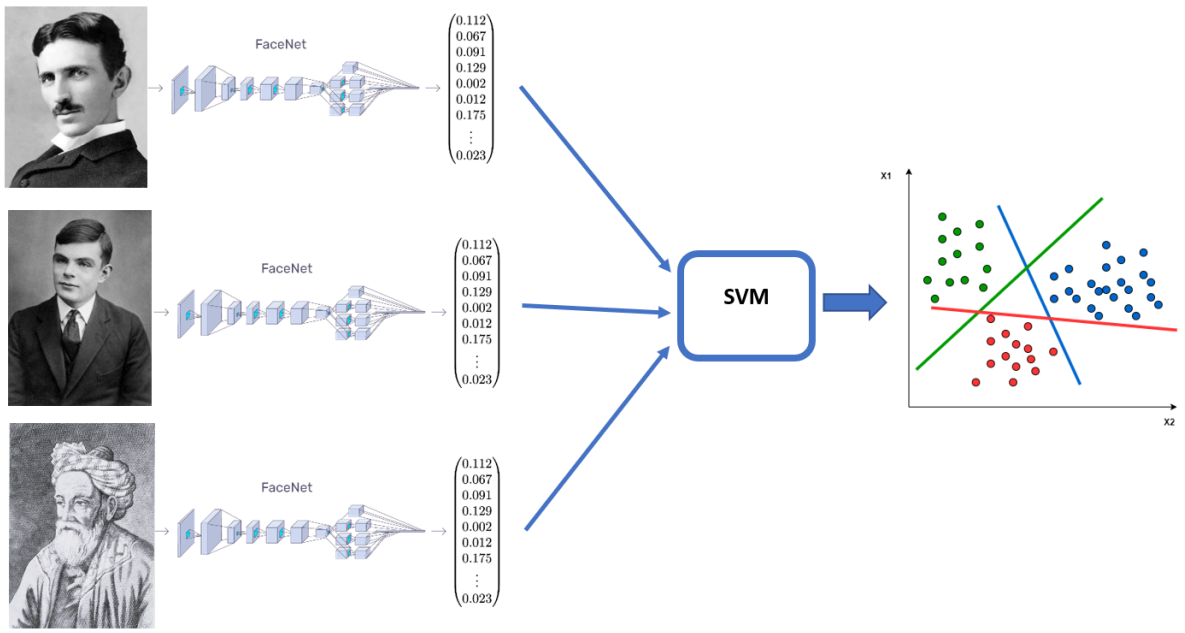
The above figure shows the steps of MTCNN ,in which, first the images are resized many times to be able to detect faces in different scales. Then, first detection is done by the Proposal-Net network. The threshold in this network is set to be low and hence there will be many false positives even after Non maximum suppression and bounding box regression. In the next network, which is Refine-Net, the detections are refined to get the most precise bounding boxes. Finally, in the last network, Output-Net, the final refinement is performed to get the most accurate detections with the most precise bounding box [11].

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 [12].



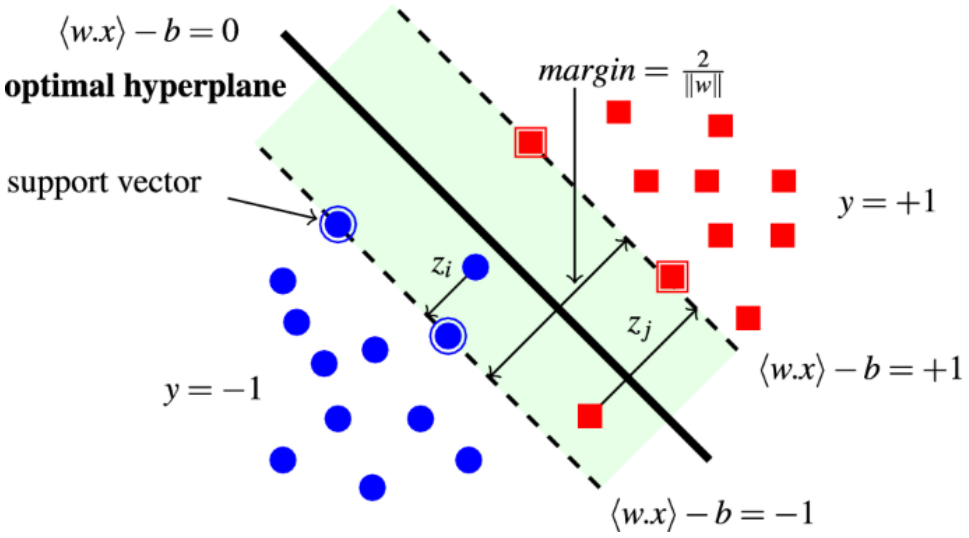
3.4.2.3 Training (Classification)

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 step and we can think of each person's face as a different class.



The used model is a linear SVM model in which for each class the hyperplanes will be found using the formulas stated in the below image and each hyperplane states the best line to

separate the data points for each class. Each of the hyperplanes are chosen such that the margins are maximised. The following figure illustrates the algorithm [13]:



3.4.2.4 Real-Time face recognition

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 as we can see a test sample of the algorithm in the below image:

3.4.2.5 FPS Improving

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.

3.5 Gate Camera System Software

3.5.1 Licence Plate Recognition with Pi Camera

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

1. OS
2. CV2
3. Imutils
4. Numpy
5. Pytesseract
6. Image from PIL
7. Matplotlib

For the reading method, the following steps are undertaken:

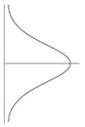
1. The Pi Camera is initialised with a continuous capture.
2. A Bilateral Filter (it is meant to reduce noise and smoothen images whilst also preserving the edges in the image [14]) is applied. The following image was taken from the internet and is referenced [14].

$$BF[I]_p = \frac{1}{W_p} \sum_{q \epsilon S} G_{\sigma_s}(||p-q||) G_{\sigma_r}(|I_p - I_q|) I_q$$

Normalization Factor

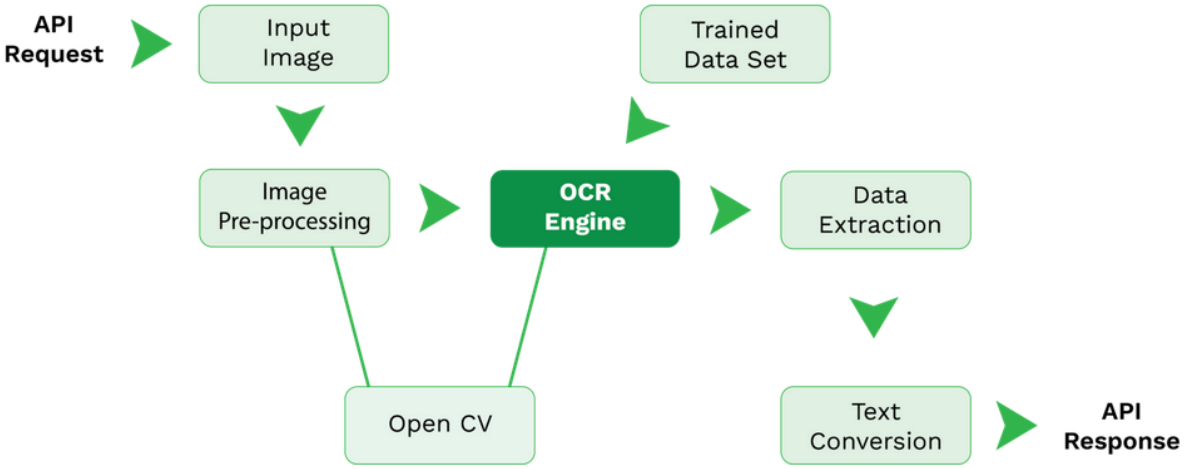
Space Weight

Range Weight



3. Then the Canny Edge Method (it makes use of a multi-stage method to find edges in an image [15]) is used to detect edges so as to find the licence plate. The following image was taken from the internet and is referenced [15].

4. The Pytesseract library (it is an open source Optical Character Recognition engine that makes use of Long Short-Term Memory neural networks to find and get text from visual media [16]) is used to read and store the detected characters of the licence plate. The following image was taken from the internet and is referenced [16].



3.6 Mobile Application Software

3.6.1 Technical Details

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.

3.6.2 User Sign-up

Users can enter their email and password and name so that they can be registered to be able to use the application. This data is stored on firebase and the user is then directed to the homepage.

3.6.3 User Log-in

Already registered users can login to use the application. These can be members of the household. Email and Password need to be entered so that the user can get access to the application. The email and password are checked with the existing dataset on firebase.

3.6.4 Home Page

Once logged in, users will be directed to the home page where they can see the option of giving manual inputs to the gates, uploading their images or licence plate numbers and logging out of the page. The admin user can see the path to the page where they can edit current users.

3.6.5 Upload Data Page

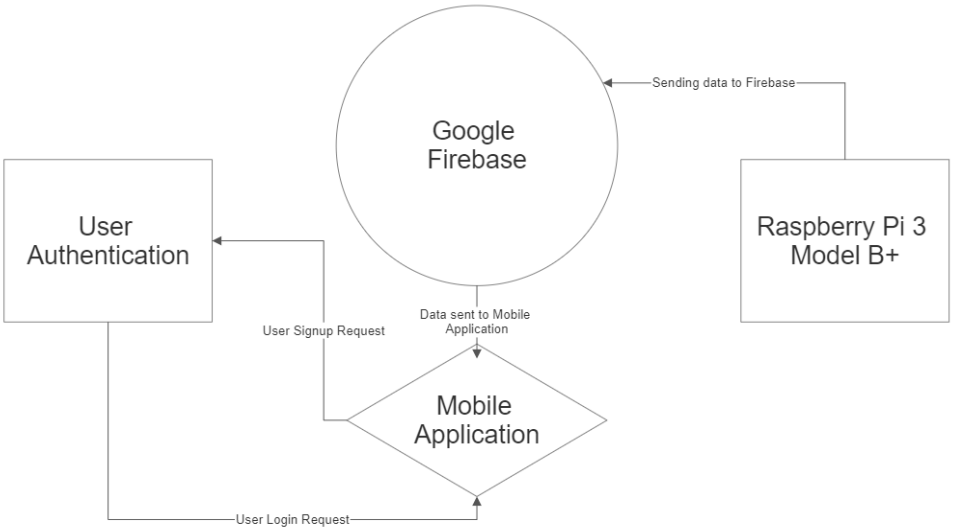
Users will be given two options, whether to upload images from phone storage or from the smartphone camera. This image will be stored on the firebase database.

3.6.6 Manual Input Page

The user here can give manual input to the door and give manual input to the gate to open it.

3.7 Google Firebase

The cloud database serves as a form of user account verification in that it facilitates registered accounts to signup and login to the mobile application. It also serves as a cloud storage for data to be circulated between the microprocessor and the other components in order to make decisions. The working is summarised as follows.



4.0 CONCLUSION

The overall design should work since most of the errors and possible obstacles are taken into consideration. However, due to raspberry pi's GPU and memory limitations there might be some unpredictable errors and outcomes which only can be seen during the testing session of the entire IoT project. The system is designed such that to perform the following tasks:

1. Registering users with their necessary information.
2. The automatic and manual control of the light system of the house.
3. The automatic and manual control over the main entrance door of the house.
4. The automatic and manual control over the gate door of the house.
5. The automatic and manual control over the house security assurances.

The following sections are for durability check, performance check, security check, product operation check, and life cycle sustainability.

4.1 Durability of the Product

The mobile application has no physical durability considerations, and hence, no requirement for it. When it comes to durability, the wiring should be of a good enough quality that the connections do not come loose, and that the wiring itself is in a casing where it can resist external harm. The product itself is made of components that are not waterproof, but this is a problem that can be solved by placing the components indoors, with the gate camera being provided a shelter from factors such as rain.

4.2 Product Performance

Not factoring in the noise, the components should work with the following response times.

Component	Component Response Time
Pi Camera	30 - 60 frames/second
USB Camera	30 - 60 frames/second
Servo Motor	10-20 ms
Light Sensors	2-50 ms
Microprocessor	1.2 GHz

It is difficult to factor in a nearly exact estimate for response times of the components while taking the noise into consideration, but since the microprocessor is the fastest component in the product, it can be assumed that the response time of the other components is the functional time for the product. As for the mobile application, since the product is still in an experimental stage, no specific conclusions can be made, but the goal should be to match the times to those of the hardware components to have a seamless integration of software and hardware.

4.3 Product Security

When users register accounts in the mobile applications, they would input personal information which would be stored in the database, as well as information about the smart home. This information should be kept safely in such a manner as to prevent it from being stolen by malicious actors. The light sensors are small and would be outside of plain view, so they are unlikely to be stolen or tampered with. Even then, they would be inside the smart home which would have other security measures taken by the homeowner. The door camera will automatically detect faces of people entering the house, and if an unknown person is detected, the user is sent a notification and/or call. If a malicious actor tries to dismount the camera, their face would be captured by the camera. The gate camera would be exposed at the outermost part of the smart home since it would need to get a view of cars outside and so is most susceptible to theft. A possible security feature here would be to keep an updated log of what the camera is observing in order to identify the thief. Since this component is a microprocessor , it should be kept in a safe part of the house in order to prevent a malicious actor from disconnecting it and stealing it.

4.4 Product Operations

4.4.1 Human Factors

With the Door and Gate Cameras being the most susceptible to theft, an unidentified person can possibly dismount and take them away. However, the person's face would be stored and they can be reported to the authorities, such as the police, for trespassing and theft.

4.4.2 Maintainability

As the system is a Internet of Things product with detailed hardware connections and complex software components, it is strongly not advised for a layman or user to attempt to maintain it. Instead, the original developers or trained engineers should be consulted for this task as they can better understand the code and hardware specifications. Attempts to maintain the product made by the user may permanently damage it or cause severe malfunction.

4.4.3 Reliability

The mobile application has no physical reliability considerations, and hence, no requirement for it. As for the hardware, it must be provided with a DC input, possibly through a power bank or battery, while taking into consideration the 5.1 V, 2.5 A requirement of the microprocessor. If the system is powered off by any reason, such as an accidental power outage, the user should only start it on and the system will continue to work as normal, booting itself with the SD card.

4.5 Product Life Cycle Sustainability

The mobile applications should be made with support for all smartphone devices available and should be updated as needed. The system is guaranteed to perform without any problem for a long term use (minimum 5 years). All the devices, integrations, and systems in the product are designed to provide reliability. However, it is recommended to get annual technical testing for durability and sustainability by the product's support team.

5.0 TEAM MEMBER RESPONSIBILITIES

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 REFERENCES

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

[2] "KY-009 3-color LED SMD module Raspberry Pi example." *Pi bits*, February 2019. [Online]. Available: http://www.pibits.net/amp/code/ky-009-3-color-led-smd-module-raspberry-pi-example.php [Accessed: 15 April, 2023]

[3] "SMD LED." [Online]. *Visual LED*. Available: https://visualled.com/en/glossary/smd-led/ [Accessed: 15 April, 2023]

[4] "Raspberry Pi 3 Model B+." *Pi Australia*. [Online]. Available: https://raspberry.piaustralia.com.au/products/raspberry-pi-3-model-b-plus [Accessed: 13 April, 2023]

[5] "Everest SC-HD01 1080p Usb Otomatik Focuslu Pc Kamera Webcam TRİPOD HEDİYE Everest SC-HD01." *Trendyol*. [Online]. Available: https://www.trendyol.com/everest/sc-hd01-1080p-usb-otomatik-focuslu-pc-kamera-webcam-tripod-hediye-p-45517194 [Accessed: 13 April, 2023]

[6] "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]

[7] "Raspberry Pi Camera Options and Usage." *Arrow*, November 2019. [Online]. Available: https://www.arrow.com/en/research-and-events/articles/raspberry-pi-camera-options-and-usage [Accessed: 13 April, 2023]

[8] "2. Getting Started." *Picamera*. [Online]. Available: https://picamera.readthedocs.io/en/release-1.13/quickstart.html [Accessed: 13 April, 2023]

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

[10] "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]

[11] "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]

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

[13] "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]

[14] "Python | Bilateral Filtering." *Geeks for Geeks*, January 2023. [Online]. Available: https://www.geeksforgeeks.org/python-bilateral-filtering/ [Accessed: 13 April, 2023]

[15] "Canny Edge Detection Step by Step in Python — Computer Vision." *Towards Data Science*, January 2019. [Online]. Available: https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123 [Accessed: 13 April, 2023]

[16] "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]