"University for Industry"

ADDIS ABABA SCIENCE AND TECHONOLOGY UNIVERSITY

College of Electrical and Mechanical Engineering

Departments of Electrical and Computer Engineering

(Computer Engineering Stream)

BSC. Final Year Project

Title: Computer Vision Based Authentication and Employee monitoring System

Prepared by: -

1. Getachew Getu …………….…… ETS0557/10
2. Yinebeb Tariku ……………….. ETS1263/10
3. Yosef Emyayu …………….…… ETS1310/10

Advisor name: Mr. Ashenafi Yadessa

June 20, 2022

EXAMINING COMMITTEE APPROVAL SHEET
TITLE OF THESIS: Computer Vision Based Authentication and Employee monitoring System

Student Names (ID)
Getachew Getu (ETS0557/10)
Yinebeb Tariku (ETS1263/10)
Yosef Emyayu (ETS1310/10)

Approved by the examining committee members:

	Name	Academic-Rank	Signature	Date
Advisor:	_____	_____	_____	_____
Examiner:	_____	_____	_____	_____
Examiner:	_____	_____	_____	_____

	Name	Signature	Date
DC Chairperson:	_____	_____	_____

Associate Dean for

Under Graduate Program_____ _____ _____

Declaration

We, hereby declare that, this paper titled "Computer vision-based authentication and employee monitoring system" bases on the results that we have derived ourselves throughout our research. Materials of work from researches conducted by others are mentioned in the references. Finally, it is a must to tell you how much effort we have made to do this project and come up with tangible outcomes.

Declared by (students):   **Signature Date**

   1. Getachew Getu   _____

   2. Yinebeb Tariku   _____

   3. Yosef Emyayu   _____

Approved by:

Advisor name: _____

Signature: _____

Acknowledgment

First, we are glad to thank the almighty GOD for giving us the potential to do this project. Next, our advisor Mr. Ashenafi Yadesa, he guides, support and continuous follow up our effort to come up with these points. It was his perpetual motivation and guidance that has helped us to carry out this project. We also want to thank Mr. Melaku who is a computer engineering lab assistant in AASTU and he supports us by allowing a lab room and by giving some electronics components in order to achieve this project.

Last but not least we would like to thank Prof. Ramasamy for his willingness to allow us use a computer vision lab and the equipment we need.

Abstract

Computer vision is a recently active topic which is based on image processing. Computer vision used in variety of topics from a simple feature detection up to a sophisticated system those need an image as in modern robots. In this project, we are going to use computer vision to make an authentication system for employee of a company to have an access to the campus or for specific resource like entering to a datacenter. By using the art-of-state of CV, we develop the system which screen users by using their face which is more secured and have no doubt of repetition since facial features are unique than other methods such as password authentications, those are less secured and vulnerable to attack.

Existing python built in database called SQLite is used in manage employee information such as their ID number, full name, department within a database. We deploy Raspberry PI as a main controller/ system on board computer. We have developed an interactive GUI using python's library tkinter, for the admins to manage their task those are registering new employee, train the model over the new added employee's faces, and monitoring the system overall functionality. Every employee needs to be registered via the admin's monitoring GUI and once their faces are added and trained on the system/model, the system will process either to give permission or deny request for an entrance via a servo motor which actuate the opening and closing of a door. Previously unregistered employee can ask a request to access the resource via guest mode which will processed and respond by the respective admins.

Since CV based face detection is challenged by light intensity easily and need an advanced calibration to overcome the effect, we develop the two-factor authentication. User need to provide his/her password, then the system generates a one-time password which is time bounded and the communication will process via email. Lastly after we designed and analyze the high-level system, we implement the whole system's prototype with available resources.

Key words: *Computer vision (CV), Face recognition, OpenCV, and OTP*

Table of content

List of figures

List of tables

List of Abbreviation and Acronym

CV ……………………… Computer Vision

OTP …………………. … One Time Password

LBPH…………………. Local Binary Pattern Histogram

LCD ……………………... Liquid Crystal Display

OS …………………….... Operating System

GPU …………………….....Graphics Processing Unit

GUI …………………….... Graphical User Interface

DC ……………………..... Direct Current

GPIO ……………………. General-Purpose Input/Output

USB ……………….……. Universal Serial Bus

RAM ……………………. Random Access Memory

VGI ……………………... Virtual Graphics Interface

DFG …………………….....Data Flow Graph

CSI ……………………... Camera Serial Interface

PWM ………………… ...Pulse Width Modulation

SMTP …………………... Simple Mail Transfer Protocol

Chapter One

Introduction

1.1 Background Study

Computer vision(CV) is a field of computer science that deals with replicating the complex parts of the human visual system and getting the machines to comprehend and understand the visual details present in the data [1]. CV is a very popular field that is used in almost all the domains that work with images and videos. It finds itself solving really complex problems in a simple manner and it is easy to train and work with as well. Computer vision enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and take actions or make recommendations based on that information[1]. Face detection and recognition are the most important application in computer vision.

Face recognition technology is a biometric technology, which is based on the identification of facial features of a person. Humans can recognize visual patterns all the time, and we obtain visual information through our eyes. For a computer, whether it is a picture or a video, it is a matrix of many pixels. The machine should find out what concept a certain part of the data represents within the data. For face recognition, it is necessary to distinguish who the face belongs to in the part of the data that the machines think of the face.

Face detection is the process of simply detecting the presence of a face in an image or video stream. Face detection algorithm is to find out the coordinate system of all faces in one image. Face recognition takes the faces detected from the localization phase and attempts to identify whom the face belongs to. Face detection and recognition technology can be applied to various fields such as security, biometrics, law enforcement, entertainment and personal safety to provide surveillance and tracking of people [2].

The methods used in face detection can be knowledge-based, feature-based, template matching or appearance-based. Each has advantages and disadvantages:

- ✓ Knowledge-based, or rule-based methods describe a face based on rules. The challenge of this approach is the difficulty of coming up with well-defined rules.

✓ Feature invariant methods which use features such as a person's eyes or nose to detect a face. It can be negatively affected by noise and light.

✓ Template-matching methods are based on comparing images with standard face patterns or features that have been stored previously and correlating the two to detect a face. Unfortunately, these methods do not address variations in pose, scale and shape.

✓ Appearance-based methods employ statistical analysis and machine learning to find the relevant characteristics of face images.

In today's world by using smart devices, we are making our needs smart. By following trends and updates we have to consider and remove drawbacks in existing system and add more features and updates. The study of Open-CV and its inbuilt library functions helps to generate a code that will do correct and authentic facial recognition system with new and more efficient use of hardware.

OpenCV (Open-Source Computer Vision) is a popular computer vision library started by Intel in 1999. The cross-platform library sets its focus on real-time image processing and includes patent free implementations of the latest computer vision algorithms [3].

Human body will be identified as an input within environment by capturing live video from a camera and the process will be done on captured video frames. The images will run through a processor such as raspberry pi and check with the stored database.

1.2 Statements of the Problem

Strong and multifactor authentication system are needed at the current situation in many sectors. we cannot trust the conventional methods namely a manual lock at the door, password keylock can be guessed or observed and token card can be stolen. To solve the problems, we tried to make things better and efficient. Face detection and face recognition with multifactor password authentication via computer vision is the most advance and it will provide a better authentication feature than the conventional access.

The existing attendance system requires employees to manually sign the sheet every time they attend their work. This adds more extra time to be consumed by the employee to find their name on sheet, some employees may mistakenly signed other's name and sometimes sheet may get lost. To solve the problem in manual attendance system, we used face recognition as an authentication

mechanism. After having images from a camera and once recognized, the door would be open and the attendance would be recorded.

1.3 Objective

1.3.1 General objective

The main objective of this project is to provide multifactor authentication and employee monitoring system by using computer vision.

1.3.2 Specific objective

- ➢ To design efficient face recognition system.
- ➢ To utilize the face recognition system based on OpenCV.
- ➢ To design and develop multifactor password authentication system.
- ➢ To generate and record user's data who have been access the system.
- ➢ Develop an effective two-factor authentication via email.
- ➢ To design highly reliable and efficient and automated door opening-closing mechanism.

1.4 Scope of the project

In this system, we have the following in-scopes such as the authentication and employee monitoring system with facial recognition-based attendance and authentication. For fog and smog weather conditions, we add a two-factor authentication and a guest mode also included for unregistered users. For the project implementation, we limit the scope to work only for local connection between admin and the system. This project is applicable in home's (rooms) door, data center and offices by using face recognition systems by providing more security.

The remote server connection between the admins and the system is considered as out-scope of the project. The case where employee may leave the staff once they take attendance is out of the scope. Also, a possible entrance of a person other than the registered person at time of door opening also out of the scope.

1.5 Literature review

We have discovered various papers related with the topic. The author in [4] proposed IoT based door access control using face recognition. The proposed system mainly consists of subsystems namely image capture, face detection and recognition, email notification and automatic door

access management. The captured image from a raspberry pi camera will be sent to the authorized person through email for safety purposes. In this paper the unauthorized users are not consider, to advance this we are proposed guest mode in which guest could ask a permission and get response via email notification in order to access the room.

The authors in [5] had implemented authentication system where if any person came at the door it was notified to the home owner via e-mail, then the owner could see the person standing at the door using camera from remote location. This paper has a limitation when the camera difficult to detect the face. to advance this we have introduce two factor password authentication system using keypad.

The Author in [6] have been proposed automatic attendance System Using Image Processing and deep learning algorithm. Face detection and recognition section detect face from the image capture by the camera, and the image of the face is cropped and stored. The system recognizes the images of student's faces, which have been registered manually with their names and ID in the record. The author only considers on marking of attendance, but in this paper, we have been considering on both marking attendance and door open.

The author in [7] had been proposed intelligent door lock system with face recognition to control the present barriers in conventional door locks. The system had been worked by detecting and recognizing the human face with the OpenCV library on the Arduino platform. The proposed system has been elaborated to stop robbery in extremely surveillance areas like home environment with less power consumption and more reliable standalone security device for both intruder detection and also for door security.

With the flow of technology advancement, more efficient algorithm, methods and systems are proposed and developed for robust service. In this project we have proposed computer vision-based authentication and employee monitoring system. The system works both on gate controlling and attendance marking simultaneously with an efficient face-recognition algorithm. Two factor password authentication system are also used when the face recognition system difficult to detect the humans face during fog and night conditions. From the survey, most researcher focus only on one condition either face recognition base gate control, or attendance mark system on day time or password authentication system for an authorized user or employees. In our case the guest mode

also considered via email notification. Also fog and night conditions are also considered which make the system multi-functional.

1.6 Methodology

In order to accomplish this project, we will follow a certain step and carry out different tasks. Different literature has been revised relating to this project and detail analysis and implementation is used to reach result and discussion. The general procedure in doing the proposed system is as follow in the figure below.
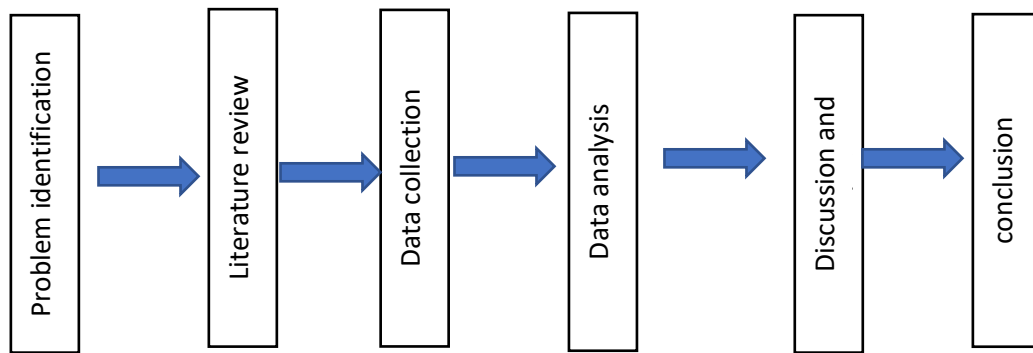

Figure 1.1 Procedures we follow in the project

In this project, we will be using the LBPH (Local Binary Pattern Histogram) face detection algorithm. This algorithm will give us more accurate results when we compare to other types of algorithms such as Fisher Face, Eigen Faces algorithms. This LBPH Algorithm will take number of images as you wish in different angles and check those all images at the time of face detection. In our case, we will be taking 30 images of a person with different angles and it will be stored in a database. We recommend to use a greater number of images with having variations to have better result, but it has tradeoff with memory and processor power. At first, it converts color images to gray scale images and then it converts into pixels for detecting this will divides the image into various pieces then it stores the values of each pixel. Images will train and store onto the database as encoded file. Pixels those grouped into low and high will takes as 0 and 1. It will be arranged into a 3 x 3 matrix format for recognizing the new images on screen compared to data base stored images. From this we divide the phases into preprocessing, feature extraction and classification phases.

The first phases are pre-processing of the image, which refers to the gathering of the image data from the camera module. But we do not need to save whole image in the dataset. We will only need a part of the face from whole captured image. For this, we will have to detect the area of the face in the image. A short code for face detection is developed. This code is also useful for other modules in the system. This detected part of the image will be cropped and saved in the data folder. Also, care has to be taken to align the images if they are shot from a different angle.

The second one is feature extraction phase, in which once we have got the images for training, we can use the image algorithm to learn on this dataset. Depending on the size of data samples, the accuracy of the classifier will vary. We applied the LBP method on image pixels by thresholding the 3×3 neighborhood of each pixel with the center value and considering the result as a binary number. The last one is classification phase which refers to testing of our face recognizer. We will do a real time video check to verify the correctness of the trained model. For a new face input, the system will first extract its features and generate binary patterns same

as we did for the training images. After its completion, the input is given to the trained recognizer to classify the image according to its training.

The designed system will follow the following steps to do the image processing.
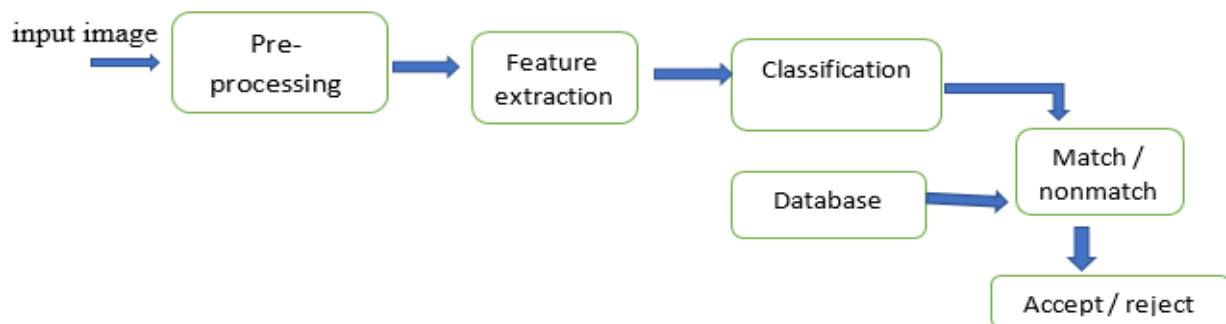
input image

Figure 1.2 flow of operation

In this chapter we have discussed overall introduction of the proposed system such as background of the study, statement of the problem, objective scope of the project, literature review and methodology. In the coming chapter we will discuss system requirement, architectural design and specifications.

Chapter Two

System Requirements, Architectural Design and Specifications

In this chapter we will discuss system requirement, architectural design and specifications. To perform projects full functional requirement with an efficient performance, setting specification and requirements are mandatory. Before purchasing a software program or hardware device, one can check system's requirements to make sure the product is compatible with his/her application.

2.1. System requirements

System requirement are the required specification that a device must have in order to perform its operation. This includes the minimum and recommended requirements of both the hardware and the software. If some-one use components with bellow recommended specification, there might be got unexpected result and the system will be nonfunctional. The hardware and software requirements include operating power, performance of operating system to operate some applications, storage or Memory unit, Input output port to work with peripheral device, and minimum GPU for displaying result.

2.1.1. Functional requirements

In our system, we grouped the following functional requirements the system need to be meet:

- ✓ Face capturing by the raspberry pi camera and then it saved on the raspberry pi memory.
- ✓ Face detection - if there is a human near to the camera, then it will be detected.
- ✓ Face recognition- if the detected man is registered on the system, then the man will be recognized by face recognition algorithm and it will allow the access.
- ✓ Send notification to Admin- if the guest wants to access the room, they must send a request to admin via emails in order to get the password.
- ✓ Email verification to user- if there is low light intensity, fog and difficult to detect the face then the system allows the user to access the system via two factor authentications.
- ✓ Automatically open the door when the authenticated person wants to access the system and the door will be closed after users enter to the room.
- ✓ The LCD will display an instruction based on user's request.

- ✓ Accepting input via keypad when the camera system is non-functional.
- ✓ Sensing the presence of human near to the camera system.

2.1.2. Non – functional requirements

We also grouped the following non-functional requirements to the system:
- ✓ Graphical user interface.
- ✓ Announcing employee when they are removed from the system.
- ✓ Safety and protective measures we aim to integrate including sensors for this purpose.

2.1.3. Hardware and Software requirements

In this section we mention hardware and software requirements of the system. we mentioned only the name along their model of the components both hardware and software those we are going to use in the project. There detail spec will be mentioned in the Engineering design and spec section on chapter three.

Hardware requirement

- ✓ **Raspberry PI3**(Model B V1.2)**:** Raspberry pi that contain camera slot, minimum of two USB port, 1GB RAM, multiple GPIO pin, micro-USB port and VGI port. Operating power is 5V with 2.5A.
- ✓ **Pi Camera v2.1:** a PI camera with a minimum lens size is 25x24x9, weight is 3g and its resolution is 8MP. The recommended resolution is using high resolution like 8MP and above.
- ✓ **Servo motor:** operating voltage will be 5V to 12V, inertial ratio is 3:1 or less, high performance servo drives have auto-tuning, but most servo motor manufacturers recommend that the inertia ratio be kept to 10:1 or less, although there are many applications that operate successfully at much higher ratios.
- ✓ **LCD display:** LCD with operating voltage 4.7V to 5V, and contains Data pin for interfacing to raspberry pi, and contrast pin for adjusting the level of brightness.
- ✓ **PIR sensor:** cover a minimum range 0.5 meter or above, with high accuracy
- ✓ **4x4 Keypad matrix:** support both alphabet (A, B, C, D), digits (0-9) and some special characters like * and #.
- ✓ Bread board.

- ✓ 5volt DC battery source.
- ✓ Wires and resistors.
- ✓ Desktop computer's monitor: for displaying purpose.

Software requirement

- ✓ **Operating system:** Debian Raspberry PI OS-32 bit and above.
- ✓ **Python:** Python environment with its library like OpenCV, pillow, NumPy and python GUI.
- ✓ PyCharm IDE: used as an editor for the codes.
- ✓ Windows OS | portable with Linux too: with minimum window 7, but the recommended is windows 10 and above.

Specifically, from the python environment, we use the following library

- o **OpenCV**: OpenCV is an image and video processing library and is used for image and video analysis, like facial detection.
- o **NumPy**: NumPy is the fundamental package for scientific computing in Python which provides a multidimensional array object. we used it to convert our images into some form of an array so that we can store the model that has been trained.
- o **Haar Cascade:** Haar-Cascade is basically a classifier which is used to detect the objects for which it has been trained. The result is an XML file which stores the trained result. Simply the Haar-Cascade is trained by superimposing the positive image over a set of negative images. We need haar-cascade frontal face recognizer to classify faces into groups for recognition purpose.

2.2. Architectural Design and Specifications

In this section we will discuss the architectural design and specifications of the system.

2.2.1. Introduction

We used a data-flow graph (DFG) for representing/ modeling connection of physical components and sequence functions used in the system. DFG is a collection of arcs and nodes in which the nodes are either places where variables are assigned or used, and the arcs show the relationship

between the places where a variable is assigned and where the assigned value is subsequently used [8].

Below is a general system model, having an input, the system itself and the outputs.
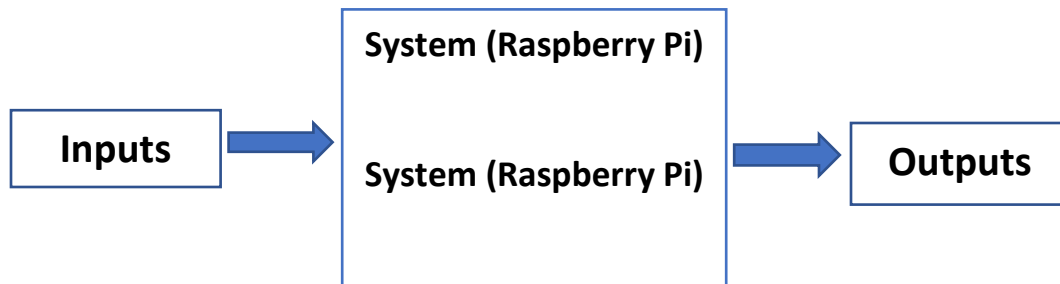

Figure 2.1 general system description

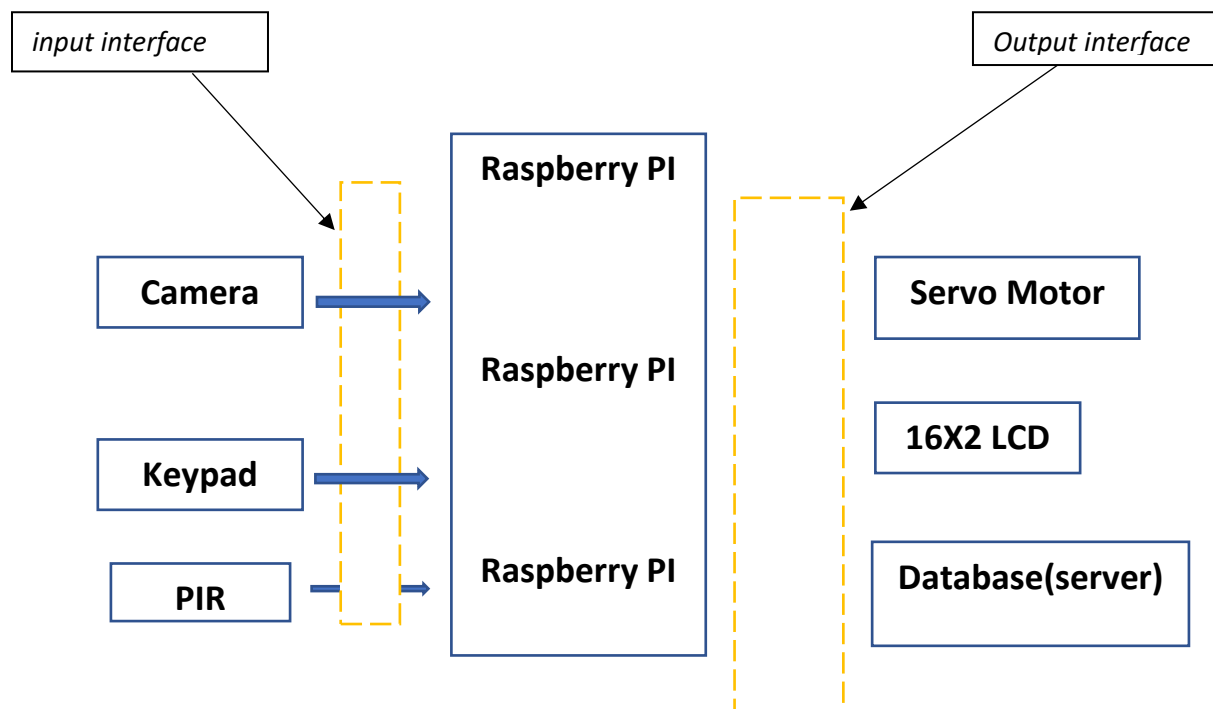A bit detail of the main system components is shown below.

Figure 2.2 our system description -on the graph

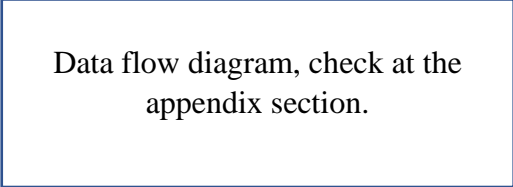### 2.2.2. Systems (Raspberry PI) detail architecture

In this section, we try to express the relationship and sequence of components as well as functions (those run on the system) interaction, along the data type they send during a communication by using DFG. Rectangular shape is used to represent hardware components such as servo motor, while elliptical shape used to represent the software - functions(methods).

Detail about the data flow diagram

As usual we consider to start from the left side where an employee sends his facial data and to be recognized by the system, or get an access to a protected resource after he get a confirmation by entering a onetime password (OTP) via a keypad. Once a user interfaced the input, his data such as face image or passwords- those are grouped into alphanumeric data will accepted and processed by going to the methods found in the main controller which again contains   more than one method. Once the input are accepted the main.py python script will identify   the method where those incoming input data should intend for. The methods inside the main controller are main.py which is a super controller for the system, guest.py which manage the guest mode request by receiving a request from someone, send to admin email and send back admins review and so on, otp.py which will manage the two-factor password authentication procedure. when there is no threshold light for the PI camera this python script will process the authentication via a password in which the system and the user communicate via an email address.

In the controller, the input data will go over a series of conditions, controls and loops for recursive actions and after a while an output data will forward to the output section. To mention this with a typical example, if a registered employee face data much with the one stored in the database, a data/command will send to the motor interface which again trigger the motor to rotate and open the door for a while. Even though the data flow diagram is depicted as an end to end system, the detail and real flow of the data is dynamic- at sometimes it may go to the beginning and do recursive processes. Such details are elaborated in the algorithm design section ion chapter three. Between a sequential two nodes, we tried to mention the data that is passed over. For data those are known specifically face data between the camera and the controller and some gross data those not mentioned explicitly due to the data passed particularly. Here one can think the data that should go over is the destinations node method parameters , so we can think of it like this one. Between

the main controller and the hardware components, there is a layer (*explicitly shown with a dot rectangle in fig 2.2 above*) called an interface which act as a link between the two ends. Systems detail data flow diagram is attached in the appendix with A3 paper, please go and refer the

Data flow diagram, check at the
appendix section.

Figure 2.3 signal flow graph

In this chapter we have discussed system requirement, architectural design and specifications. In the coming chapter we will discuss the engineering design and detail specifications of the system. Also detail algorithm design, system's hardware connection and mechanical design will be covered

Chapter Three

Engineering Design and Detail Specification

In this chapter we will mention detail engineering design and calculations along detail engineering specification of components. And also, we will have mentioned the operating power of each component and discussing how the algorithm works by indicating the flow chart of user registration, training and face recognition system. Finally, the electrical diagram are drawn by using fritzing software.

3.1 Engineering design and detail specification

In this section, we will cover hardware detail engineering specifications and related calculations. The calculation includes doing it via a simple logic like, if the raspberry pi sends an output of 3.3-5 volt, then it is not enough to run a 12v motor, so we have to interface the Raspberry pi and the motor via a relay module (one which may integrate with a motor drive or a separate one). It may also go to design a component from a scratch by measuring all detail input outputs like doing a feedback control system for a particular purpose, it needs doing basics calculation on transistors, amplifiers and so on.

1. **Raspberry pi**

The Raspberry Pi is a microprocessor like a single board computer and can host the operating system. The Raspberry Pi runs Linux distribution OS, but it also provides a set of GPIO (general purpose input/output) pins, allowing you to control electronic components for physical computing and explore the Internet of Things (IoT).

Raspberry Pi OS is a free operating system based on Debian, optimized for the Raspberry Pi hardware, and is the recommended operating system for normal use on a Raspberry Pi. In this project we have used third generation quad core raspberry pi 3 model B V1.2, which have 40 GPIO pins, Camera Serial Interface (CSI) camera port for connecting a Raspberry Pi camera, Display Serial Interface (DSI) display port for connecting a Raspberry Pi touchscreen display, Micro SD port for loading your operating system and storing data, and Upgraded switched Micro USB power source.

A Raspberry Pi must be powered with a compatible power supply. Raspberry pi 3 model B work fine on 5volt with 2.5A. The board consists of two 5V pins, two 3V pins, and 9 ground pins (0V), which are unconfigurable.

A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the extreme right edge of the board. GPIO pins do not have a specific function and can be customized using the software. A GPIO pin that is set as an input will allow a signal to be received by the Raspberry Pi that is sent by a device connected to this pin. A voltage between 1.8V and 3.3V will be read by the Raspberry Pi as HIGH and if the voltage is lower than 1.8V will be read as LOW. On the other hand, a GPIO pin set as an output pin sends the voltage signal as high (3.3V) or low (0V). When this pin is set to HIGH, the voltage at the output is 3.3V and when set to LOW, the output voltage is 0V. The raspberry pi contains both software and hardware pulse width modulation. Software pulse width modulation is available in all pins, but hardware pulse width is available on specific GPIO pins like GPIO12, GPIO13, GPIO18, and GPIO19.

For the raspberry pi and for the other devices, we summarized the detail spec in table following the descriptions.
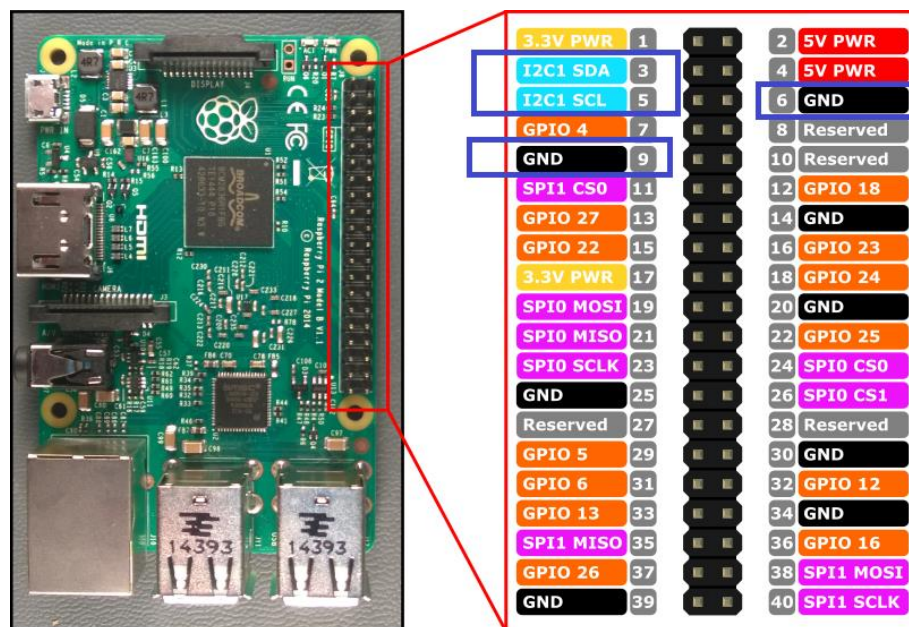

Figure 3.1 raspberry pi and its pin configuration

14

2. Liquid Crystal Display (LCD)

A liquid crystal display (LCD) is a type of display technology that makes use of liquid crystals that open or close when stimulated by an electric current. Out of various types of LCD, we choice 16×2 LCD which is cheap and easily to program. The 16x2 LCD 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. So, it will have (16×2=32) 32 characters in total. The operating voltage of this LCD is 4.7V to 5.3V and current consumption is 1mA without backlight Alphanumeric.

A 16×2 LCD has two registers such as data register and command register. The RS (register select) is mainly used to change from one register to another. When the register set is '0', then it is known as command register. Similarly, when the register set is '1', then it is known as data register.

- The command register is used to store the instructions of command which are given to the display. So that predefined tasks can be performed such as clearing the display, initializing, set the cursor place, and display control.
- The data register is used to store the information which is to be exhibited on the LCD screen. Here, the ASCII value of the character is the information which is to be exhibited on the screen of LCD. Whenever we send the information to LCD, it transmits to the data register, and then the process will be starting there. When register set =1, then the data register will be selected.

Figure 3.2 Liquid crystal display (16x2 LCD)

3. Relay module

A relay is an electrically switch, which is operated by electromagnet. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit.

When the relay is de-energized, the sets of contacts that were closed, open and breaks the connection and vice versa if the contacts were open. The operating voltage of relay in door application is 12V or 24V and current is 500mA or 250mA respectively.
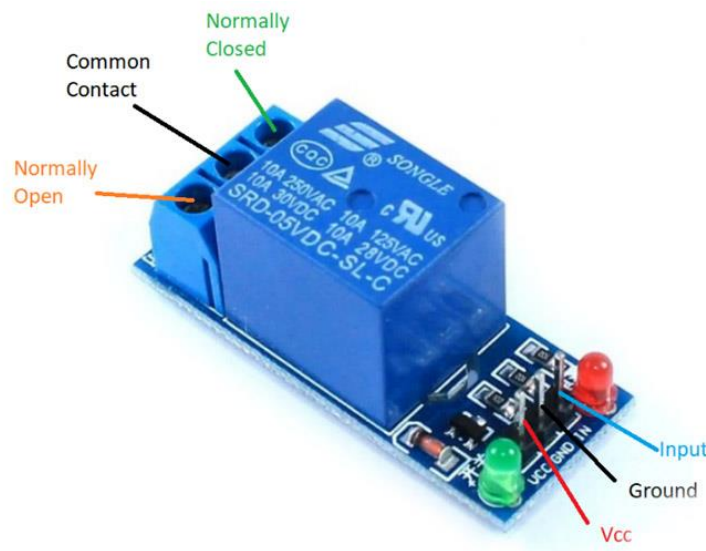

Figure 3.3 One channel relay modules

4. Servo motor

Servo motor is a combination of DC motor, position control system and gears. The motor has shaft, the rotation or the angular position of which is controlled by the signal that appears on the signal line of the servo motor. The servo motor has three wires connected to it. Two of three wires of the servo motor are the power pins and the third pin is the signal pin to which the signal is connected in such a way that the pattern of the signal controls the angular position of the shaft of the Servo Motor. we are surveying weights of many international doors, so the weight range is between 5kg to 40 kg. Stall torque is the torque produced by a mechanical device whose output rotational speed is zero. It may also mean the torque load that causes the output rotational speed of a device to become zero. A 9.4kg/cm Servo motor should be able to lift 9.4kg if the load is suspended 1cm

away from the motors shaft, the greater the distance the lesser the weight carrying capacity. For this project demonstration we have used SG90 Servo Motor. It is one of the popular, cheapest and 180-degree servo. Its operating voltage is 4.8V

The rotation angle of the servo motor is controlled by applying a Pulse Width Modulation (PWM) signal to it. By varying the width of the PWM signal, we can change the rotation angle and direction of the motor. In servo, we have a control system which takes the PWM signal from Signal pin. It decodes the signal and gets the duty ratio from it. After that, it compares the ratio to the predefined positions values. If there is a difference in the values, it adjusts the position of the servo accordingly. The frequency of PWM signal for SG90 is 50Hz.

PWM is a type of signal which is obtained from Raspberry Pi Pico board. The output signal is a square waveform which at a particular instance is either high or low. If we are using a 3.3V power supply then the PWM signal would be either high which is 3.3V or low which is 0V. The 'on time is the duration till which the signal stays high and the 'off time' is the duration till which it stays low.

$$\text{Duty-cycle} = \frac{on\ time\ of\ signal}{total\ time\ period\ of\ signal}$$

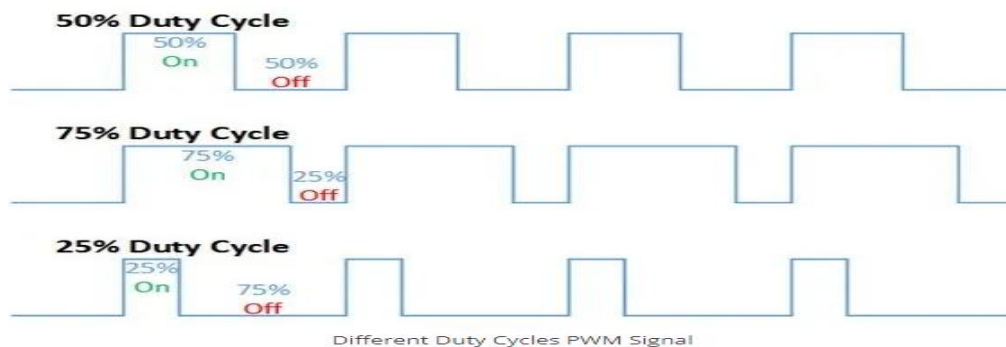Different Duty Cycles PWM Signal

Figure 3.4 a typical SG90 servo motor

Inertia ratio is the most important factor for sizing servo motor. It can be used as a measure of how well the motor is able to control the acceleration and deceleration of the load. Inertia ratio is defined as the ratio between the inertia of payload to the inertia of the motor.

$$\text{Inertia ratio} = \frac{JL}{JM},$$ where JL is the inertia of load, and JM is the inertia of the motor

Manufacturers typically provide the inertia value of the motor, so we need calculate the inertia of the load. To determine the inertia of a screw-driven load, the effect of the screw's lead must be considered.

$$J = m \cdot \left(\frac{l_S}{2\pi} \right)^2$$

Inertia of load driven by screw

where, m is mass of the load and Ls is load of screw

An inertia ratio that is too low means the motor is likely oversized, leading to higher than necessary cost and energy usage. An inertia ratio that is too high means the motor will have a difficult time controlling the load, which results in resonance and causes the system to overshoot its target parameter (position, velocity or torque).

5. PI camera module

The Pi camera is a portable light weight camera that supports Raspberry Pi. It communicates with Pi using the mobile industry processor interface (MIPI) camera serial interface protocol. It is normally used for capturing videos in the system. Apart from these modules Pi can also use normal USB webcams that are used along with computer. We are using 8MP resolution camera. It's capable of 3280 x 2464-pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video. But we recommend to use camera above 8MB resolution for sake of image quality which has effect on later processes.
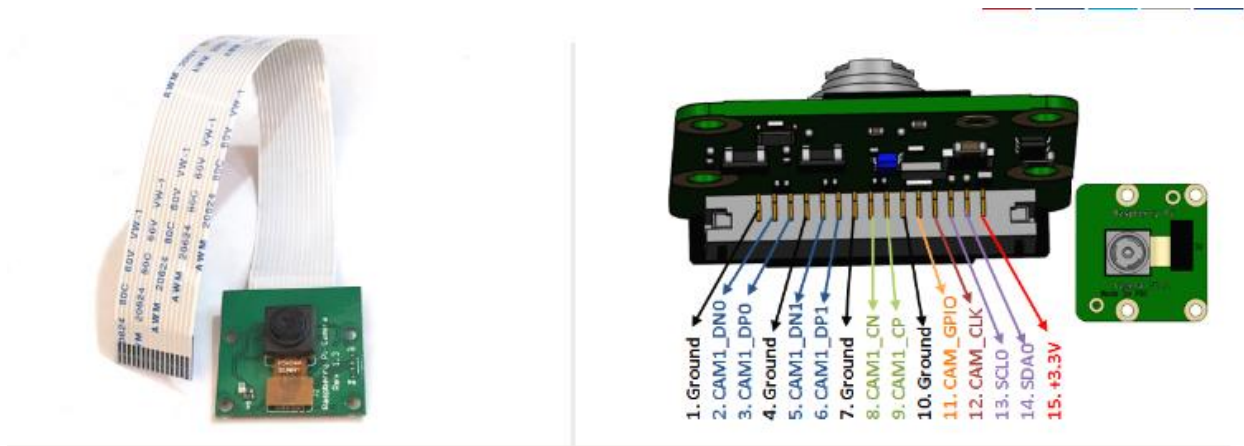
Figure 3.5 camera module and its pin configuration

6. PIR sensor

A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. PIR sensor detects a human being moving around within approximately 10m from the sensor. This is an average value, as the actual detection range is between 5m and 12m. The operating voltage of PIR sensor For a D.C. works from 3.3V to 5V.

The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.
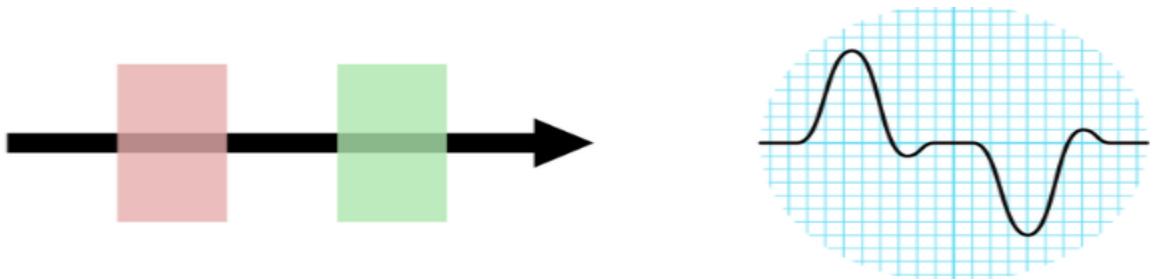

Figure 3.6   a.   Heat source movement   b. output signal

7. 4 x 4 Alpha-Numeric keypad

The 4 x4 keypad module consists of 16 keys, these Keys are organized in a matrix of rows and columns. Normally there is no connection between rows and columns. When we will press a key, then a row and a column make contact. When we will Pressing a button shorts one of the row lines to one of the column lines, allowing current to flow between them. For example, when key 'Button 1' is pressed, column 1 and row 1 are shorted, then the character is sent to the required output.
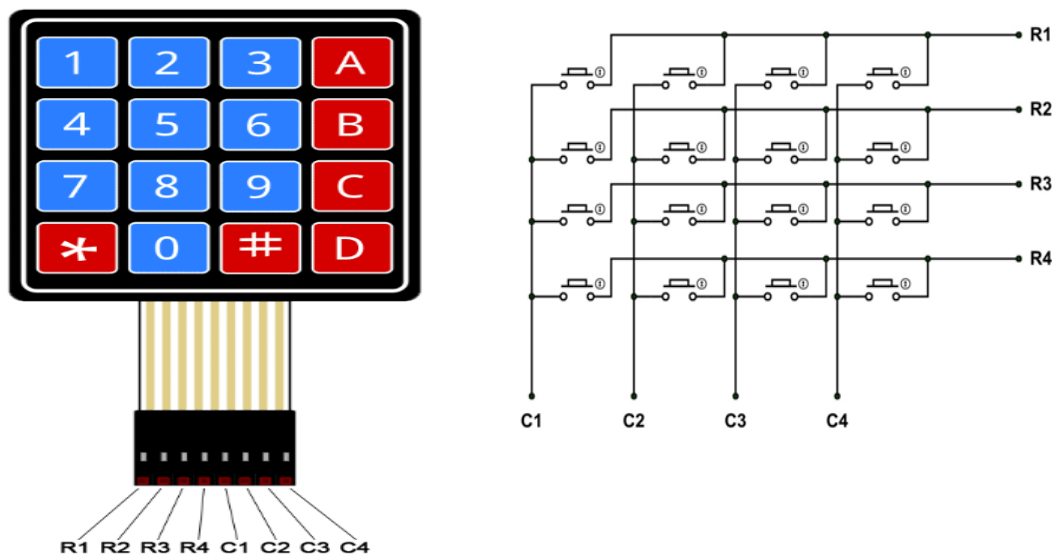

Figure 3.7 4x4 alphanumeric keypad and its internal structure

8. Screen Display-Monitor

A display device is need to show and display the designed GUI for the admins in a central room. A simple desktop monitor or any screen display devices those can support HDMI cable can be used. In our demo, we are using desktop's monitor as a display. To specify, a monitor having a moderate resolution level and screen size of minimum 15inch is enough.

9. Server Computer

We were doing and thinking on how we can make the project more relevant by making it expandable and following we were searching on how we can manage and do all logical operations performed on the existing -demo level inside a remote, powerful, more efficient server computer. This will solve the processing capability of the raspberry pi for hundred thousand of users and we are approaching to implement a remote server. For this scalable scenario, a separate remote server that will manage and do tasks those were done in the raspberry pi such as storing data including

faces and metadata including user name, user id, email and others on remote server. Both admins and the raspberry pi will access the remote server. and each data modification will synchronized. Since both are not in one room-the usual one, there need to have a network where both ends will communicate over. The protocol over the network will depend on the organizations, it is better if the network which is going to created is compatible and inclusive within the existing resources. Below diagram shows the way in which the local raspberry pi and remote servers are communicated.
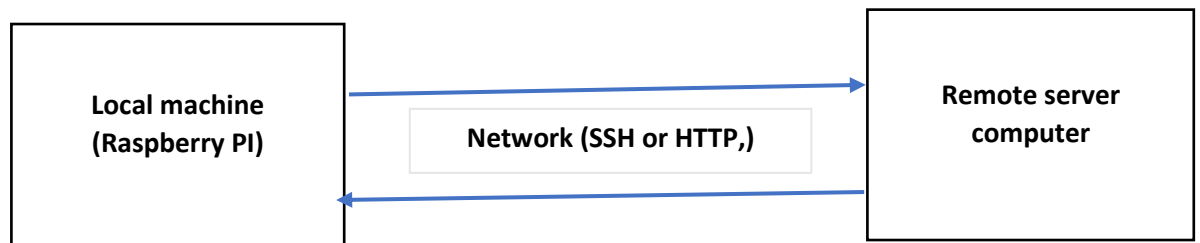
```
┌─────────────────────┐                                    ┌─────────────────────┐
│                     │  ──────────────────────────▶      │                     │
│   Local machine     │                                    │   Remote server     │
│   (Raspberry PI)     │      Network (SSH or HTTP,)        │   computer          │
│                     │  ◀──────────────────────────      │                     │
└─────────────────────┘                                    └─────────────────────┘
```

Figure 3.8 communication between the local device raspberry pi and the remote server via network

we can summaries the device spec as shown table in the next page, in table one for the design level and table two for the implementation part:

Device name	Specification
Raspberry pi	- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU - Support: Ethernet cable, - Full-size HDMI CSI (15-pin Camera Serial Interface) camera port for connecting a camera - RAM: 1GB and more, processor - Micro USB-2 power port (up to 2.5A) - Micro SD port 4GB and above - Supply voltage 5V - 40-pin extended GPIO - DSI (Display Serial Interface) display port for connecting a touchscreen display - 4 USB 2-port - Video: Supports 1080p @30fps, 720p @60fps and 640x480p 60/90 Recording
LCD	- Operating Voltage is 4.7V to 5.3V - Current consumption is 1mA without backlight - Can work on both 8-bit and 4-bit mode - display alphabets and numbers
Camera	- Resolution: 3280 * 1944 /8MP and above color camera module - Supports: 1080p, 720p and 480p
Relay	- supply 5v and above - maximum contact voltage – 250VAC or 30VDC - maximum current – 10A - Relay is activated 70mA
PIR sensor	- Range 2-4 meter - Good accuracy - Level of output: high 3.3V, low 0V - Operating voltage 5V
Servo motor	- Stall torque min: (5V) 21Kg/cm - Stall torque min: (6.8V) 25Kg/cm - Operating speed: 0.15 s/60º (5 V), 0.16 s/60º (6.8 V)
keypad	- 4x4 alphanumeric - Maximum Voltage rating 24V - Maximum Current rating 30mA - Numbers of Pin 8

Table 3.1 hardware specification fort the design-level

Device name	Specification
Raspberry pi	- Similar with the above.
LCD	- Similar with the above specification - 16x2
Camera	- Resolution: 2592 * 1944 / 4MP
PIR sensor	- Similar with the above,
Servo motor	- Operating voltage 4.8V - Weight 9gram - Torque 1.8kg-cm
keypad	- Similar with the above,

Table 3.2 Hardware specification (for the implementation-prototype).

3.2 Algorithm Design

Algorithm is a set of procedures in order to do some tasks. Algorithms are never written to support a particular programming code. We used a flow chart to describe the algorithm of the system where the set of procedures the system will go through to do the expected tasks. The end task the algorithm aims to destined is either recognizing an employee via his face and by going several steps, after recognition done, the door will open and the user/employee is authenticated and the algorithm ends at this point. In another perspective, if we are considering a registration and training phase of the model for new user/ employee existed, the algorithm aimed destination after going several steps, model training is done and a trainer file will generate which is face's encoded data file and the algorithm ends at this point. Since it is easy to recognize and understand the flow and relations, we used flowchart over pseudocode.

Flow chart for Registration with training and for the recognition process of the system is give below. The drawing is done on flowchart-draw.io, online[1].

---

[1] Flowchart-draw.io link: https://app.diagrams.net/

Flow chart for Registration and training

Figure 3.9 flow chart for registration and training
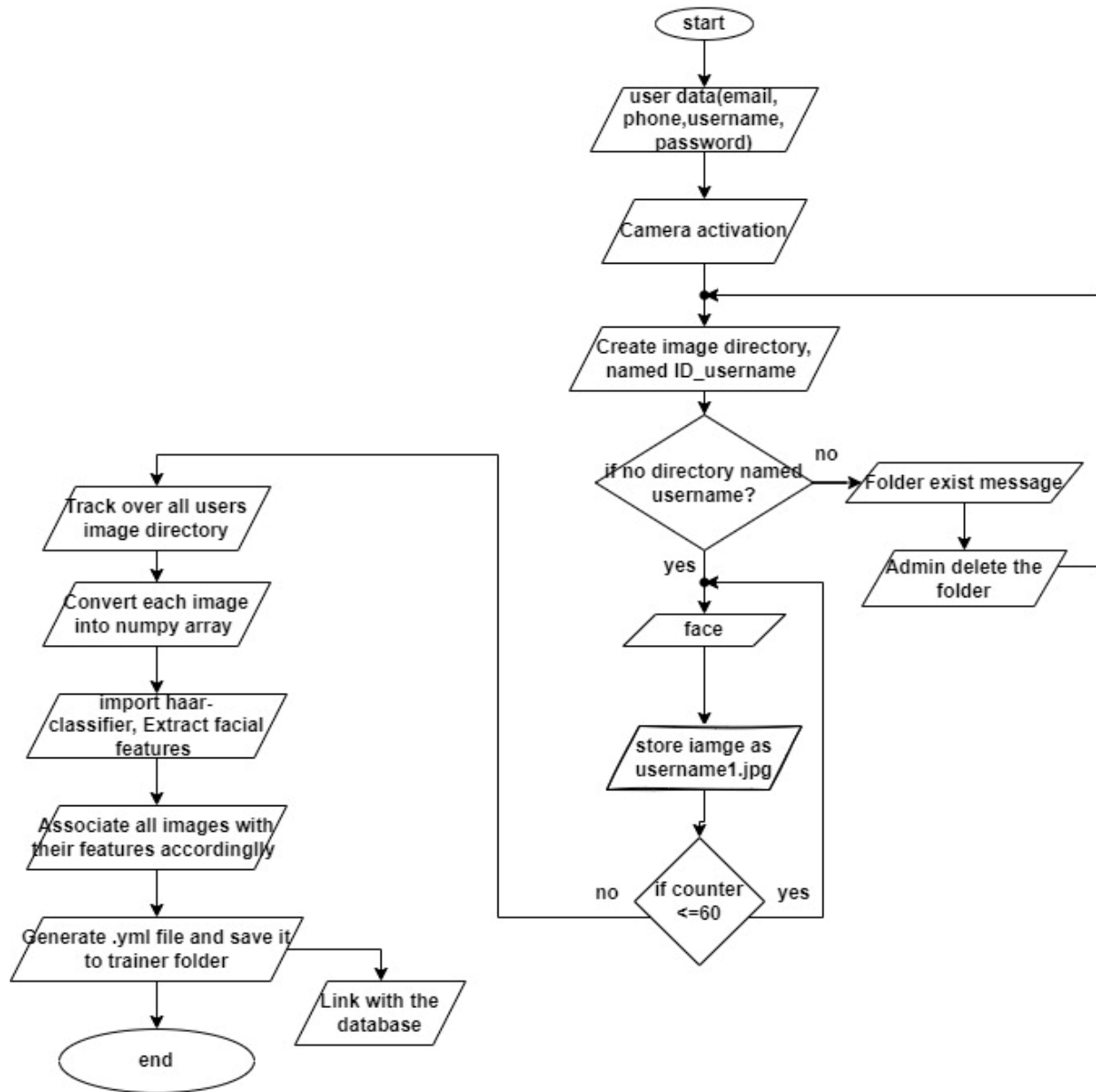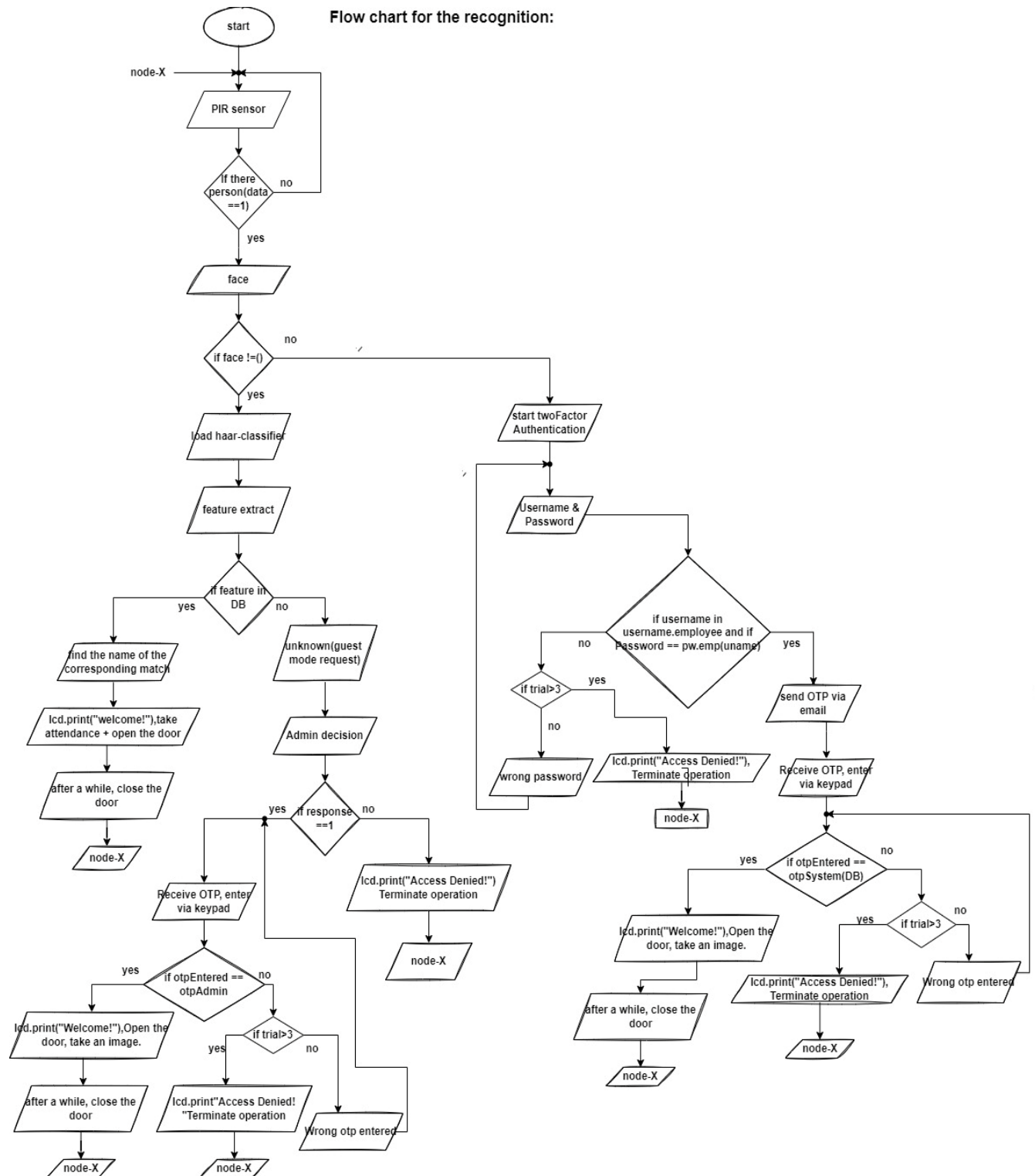
Flow chart for the recognition:

Figure 3.10 flow chart for face-recognition

3.3 System's hardware/ circuit configuration(implementation)

Below we try to draw the possible hardware component connection and their procedure for the designed system as shown below by using a fritzing drawing application.
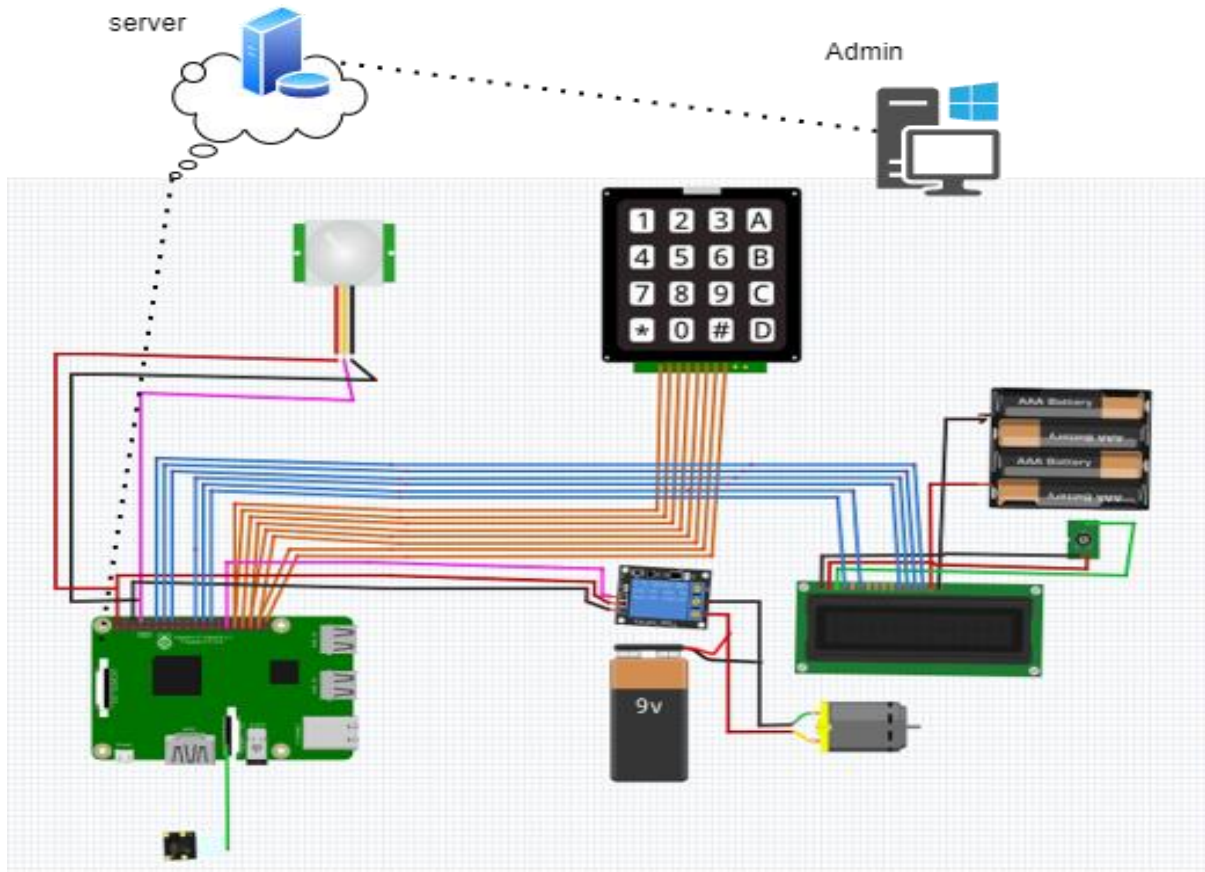

Figure 3.11 System's circuit connection (drawn in fritzing).

Here, the left bottom green device is the raspberry pi, the lines in between devices are numbers of wires those are going to draw between for data/signal exchanges. The left top sensor is a passive infrared ray sensor which used to detect the presence of a person near the camera. In the right side, a 4x4 keypad matrix, a relay module, DC (5volt) battery, LCD along a potentiometer for adjusting its brightness. In the between a bread board is placed.

3.4 Mechanical Design

Below we try to draw the door system where the system is going to operate on. We draw it by using draw.io software.
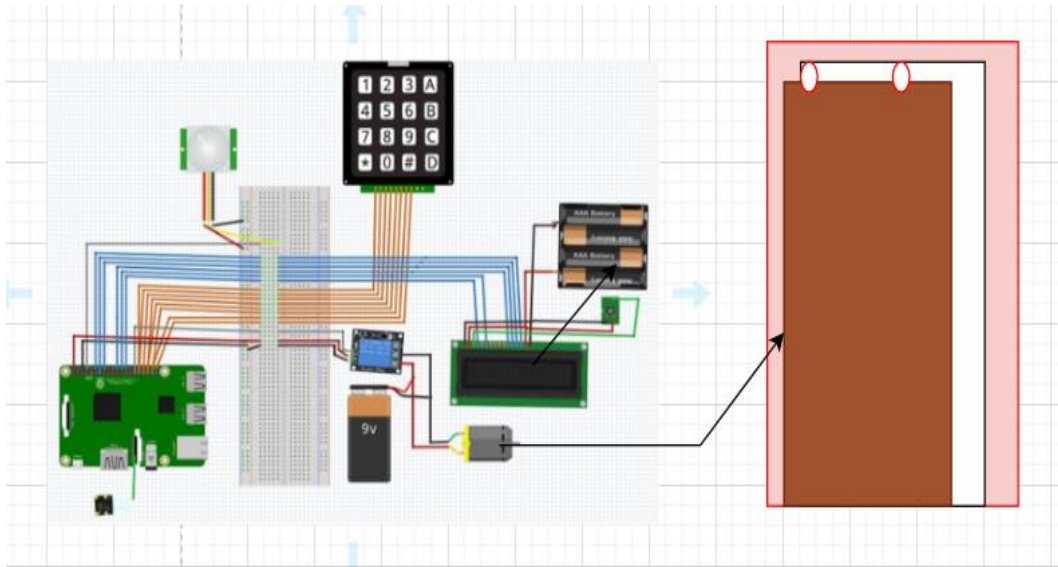

Figure 3.12 mechanical design of the door system

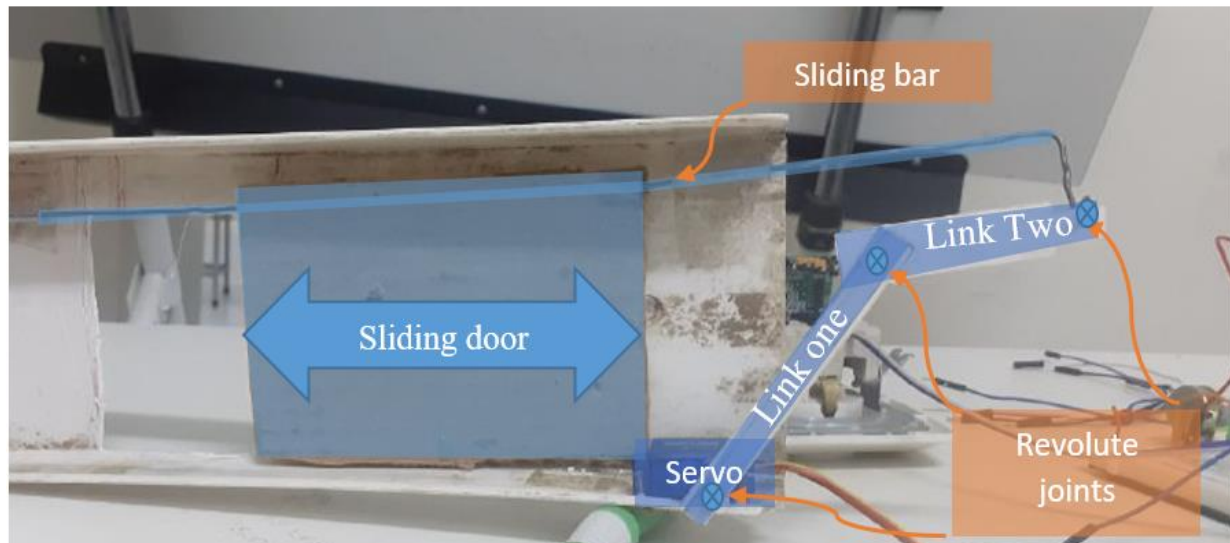A more clear and real door which shows the closing and opening mechanism is show below.

Figure 3.13 the mechanical design for controlling door movement

Working operation

Initially before the motor gets command form the microcontroller, the door is closed and the first link that is attached to the servo motor inclined at an angle of 45^0 as shown in diagrammatic picture bellow.

Figure 3.14 initial position of the door (door is closed)

When the servo motor rotates in the clockwise direction, the first link will also rotate in the same direction as the motor, the rotation of the first link pushed the second link and the second link pulls the sliding bar that holds the door then the door will starts sliding to the right

Figure 3.15 the sliding door start to open

Final, the motor continues rotating and the link attached to the motor riches at angle of 135^0 the door will fully opened.
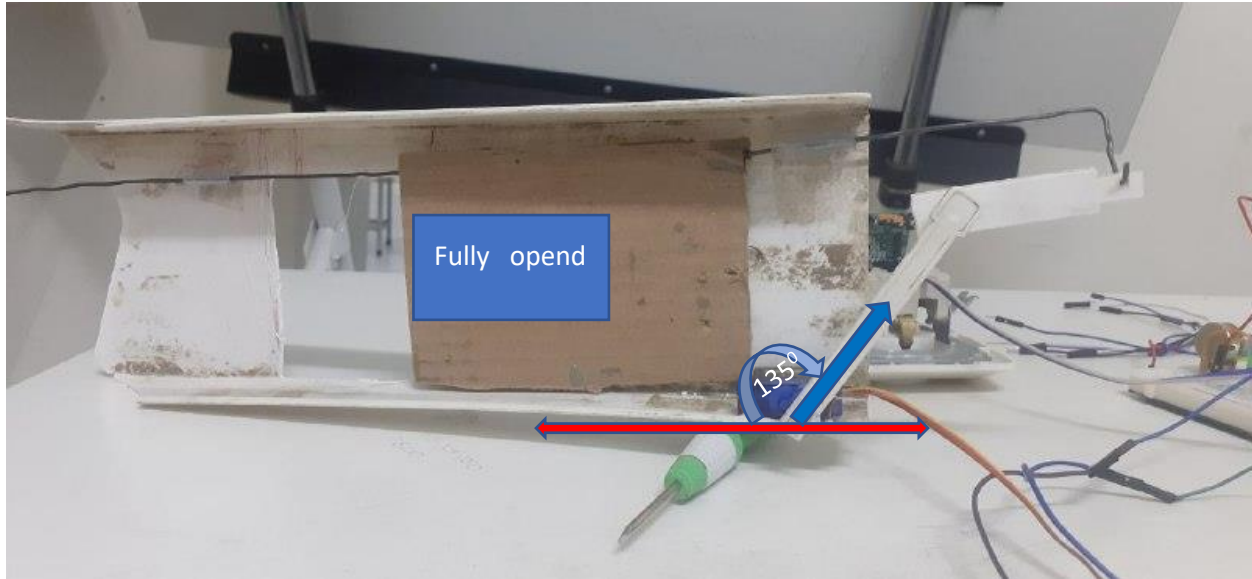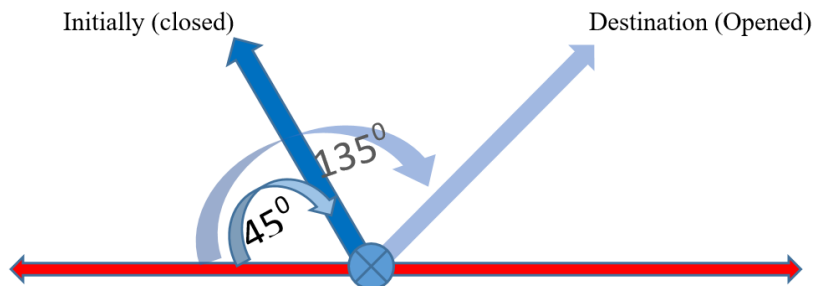
Figure 3.16 the door reaches final position (opened)

Once the link attached to the motor reaches 135^0 from the horizontal, the servo motor will finish its rotation in the clockwise direction and it is ready for the anticlockwise direction to close the door. For closing of the door, a similar operation will be followed in the anticlockwise rotation of motor

Figur 3.17 initial to final movement of the firist link

In this chapter we have discussed the system's engineering design, detail specifications, algorithm design and mechanical design. In the coming chapter we will discuss implementation, testing and discussions. In the implementation section we cover component connection of the circuitry and detail code explanations.

Chapter Four

Implementation, testing and Discussion

In this chapter we will elaborate the topics about our implementation of the designed system, the detail procedure for the operation to be followed and the detail behavior on which the data flow diagrams and the algorithms designed in the previous chapter will be entertained with a real input data and each phase of the algorithm and conditions will be testified in a logical coherency. At each stage, we will test and document the results and along each result and procedures, each phase will be discussed with a detail illustration.

4.1 Implementation

The entire system is made up of raspberry pi, LCD display, PIR sensor, Pi camera module, Relay module, servo moto, keypad, electrical circuit board and the gate system. The raspberry pi a single board circuit (SBC) microprocessor that control all activities inside the system.

The system contains two phases. The first phase is so called registration and training phase and the second phase is called recognition phase. during the first phase the system's admin registers the user's information user's such as Identification number, full name, gender, and email address. After filling the form, the admin clicks the camera activating button (take image button), then the PI camera capture 30 images and store it on image directory. After registering the user, the system admin trains the model over the images found in the image directory over the whole registered users face. The training phase will take the image on the image directory as an input and by having the haar-classifier, it will extract a feature for each face images and once it correlates and associate, an encoded face data will be stored and save in a trainer file. This generated .yml file will be used in the second phase for the recognition stage to compare the images extracted data. For every new user registration, a training need to have performed, else the system will have no the new user's facial data and will probably know as "Unknown!".

When we come to face recognition phase, the PIR camera sense the presence of human and if it is present, the PI camera is trigger and start detect humans face. If the detected face is matched with the face data stored in the trainer file – which is considered as it saved on a database, then the name

of the recognized person is attached with his/her face. during this time, an LCD display's method will be called and passed a string - 'the door is open, welcome Dear user!!' at that time the system main controller wills send a command to the raspberry PI GPIO pin where the motor drive is listening a data, and will forward a command to open rotate the motor and the door will opened for a while. Again, attendance is also marked by including, full name and id of attended person with attended date and time. If the detected person is unknown then, the person must ask a guest mode and send request to admin via email, and then the admin give a permission to access the door, unless the access is denied. The trainer file is used as a reference during face recognition system.

For the sake of resources, time and budget constraint, we are going to implement only the local system which consist of the whole electrical components presented in the electrical connection as given before in the prototype's circuit connectivity.

The raspberry pi which is the central system, also known as the brain of the system. It is used to store data, control the I/O pins, monitoring users. The PIR sensor is used to activate a camera system by detecting the presence of humans within a specified range. Since raspberry pi's GPIO pins release a maximum of 3.3 volt only, and insufficient to run an external servo motor and for the sake of raspberry pi's safety due to back electromagnetic force, a relay module is used to switch the servo motor during closing and opening the gate. The alphanumeric keypad is an input of the system that is used to enter a string in to the system. Such strings can be user's password and username, admins password and username, or a onetime password given by the admin. The LCD display is an output of the system that used to display some instructions to the user.

The LCD is interfaced with the raspberry pi via implicit pull up/down resistors those are built inside the PI and only an external potentiometer is used to adjust the brightness of the display. As depicted in the algorithm for the recognition scenario, the very beginning task/process goes is the PIR sensor detection for the presence of a human around the gate. The connection for the IR is, it gets a 5volt input and the data pin is connected to raspberry PI's GPIO pin number 17. The code is written in a way to send a data and activate the camera then the whole system to detect, recognize and do the rest functionalities. Once the system is started, it will grape a frame from the employee face and check face's extracted feature with the one stored in the database, after making a series of procedures, the system will validate either the employee has to entered or not. The gripped

31

image will be passed over the haar - classifier and a feature will be extracted. If a match found, the PI will send a command data to the motor interface and the motor; will opened for a while and allow the employee to entered. If the face take is out of the database facial data, the user will get a guest mode request. This is nothing but, for a user who has no registered yet on the system gets a trial to get access after having a permission from am respective admin of the system. The code is written to send a request from the employee to the admin, sending the response from admin to the employee via a simple mail transfer protocol (SMTP)- email services. Here, we are only reliable on the security which google provide in the communication protocol which is secured shell transmission (SSL) and for the case those need a greater security on such communication, another encryption technique can be deployed on it.

When the PIR's data shows a presence of a man in front of the camera, but no face detected, the case can be for two reasons, one it may not have enough brightness level which the camera needs to work properly. The second and most rare scenario is when the user faced the camera in inappropriate position. We only consider and integrate the first- at night time and in a summer season, a fogy whether cause the brightness to dimension. During this case, the system will trigger the users to gain access to the resource once they're authenticated via a two-factor authentication. As we brief the registration phases implementation, a new user will give his email, password, username during his registration. So, the system will prompt the user to follow his email and once he requests this OTP means of authentication, the system will generate a one-time password and send to his email and if here entered he correct username and password related to his email address, he will get access to enter the door once the entered passcode is similar to that generate by the system within a short period of times. The generated OTP will be stored in a database and once it will used, the system clears it. But in our implementation, we only store the OTP in a script file that is on the run on a variable and the best and scalable way id the one which stored in a database.

4.1.1 Component connection through GPIO pins

When we come to the implementation detail, the PIR sensor, the PI camera, the 4x4 matrix keypad, LCD, relay module, and the desktop are connected to the central unit called raspberry pi. The raspberry pi camera's 15 pins are connected to the raspberry pi camera slot port.

The PIR's ground pin connected to the GND pin of raspberry pi, the VCC pin of PIR is connected the VCC of raspberry pi and the third pin is data pin, which is powered by the 3.3V raspberry pi GPIO5 and send the presence of human to raspberry pi by using this data pin.

The 4x4 matrix keypad's row R1, R2, R3, R4 are connected to GPIO26, GPIO19, GPIO13, GPIO6 pins of PI and the column C1, C2, C3, C4 are connected to GPIO12, GPIO16, GPIO20, GPIO21 pins of raspberry pi respectively. The users are communicating or send a steam of character to the system through these pins.

For the LCD, first, connect pin GND and K of the LCD to GND and VDD and A pin to 5V supply. Then connect a 10KΩ Potentiometer to V0 pin of the LCD, which is the contrast adjustment pin. The two control pins of the LCD i.e., RS and E are connected to GPIO7 and GPIO 8 respectively. Now, the data pins of the LCD, since we are configuring the LCD in 4-bit mode, we need only 4 data pins (D4 to D7). D4 of LCD is connected to GPIO25, D5 to GPIO24, D6 to GPIO23 and D7 to GPIO18.

Connecting electric motor directly to the raspberry pi may cause a damage due to high current in the motor, we need to insert relay module between the raspberry pi and the servomotor. The output of the raspberry pi GPIO 22 is connected to the signal pin (IN) of the relay, so the microcontroller controls the relay via this GPIO pin. the Ground pin of relay is connected to the ground pin of the raspberry pi and the 5V VCC pin is connected to 5V VCC of raspberry pi. When we come to the other side of the relay module (output side), the normal open pin is connected to VCC of power supply (5V or above) and the common of relay is connected to the VCC of servo motor.

The ground pin of the servomotor is connected to ground pin of power supply. On the other hand, the signal pin of a servo motor is directly connected to the PWM port of the raspberry GPIO17. A servo motor is controllable through PWM. Specifically, its arm position depends on the width of the pulse applied to it. Once a match face is found, the Raspberry PI's GPIO will send a command to the motor drive (relay) and the motor dive (relay) will switch the relay to run the servo motor.

Below is summary of component connection

Component		Raspberry pi pin
PI Camera		CSI camera slot
PIR sensor	VCC	VCC
	GND	GND
	Data pin	GPIO5
LCD	VCC	VCC
	GND	GND
	RS	GPIO7
	E	GPIO8
	D4	GPIO25
	D5	GPIO24
	D6	GPIO23
	D7	GPIO18
servo	Data pin	GPIO17

components		Raspberry pi pin
keypad	row1	GPIO26
	Row2	GPIO19
	Row3	GPIO13
	Row4	GPIO6
	column1	GPIO12
	Column2	GPIO16
	Column3	GPIO20
	Column4	GPIO21
relay	Data pin	GPIO22

Table 4.1 component pin connection

4.1.2   GUI and python Scripts uses

Python has a lot of GUI frameworks. Tkinter is a graphical user interface (GUI) library for Python, used to make buttons, desktop apps, image and text area features with an abstractive feature. The system has an interactive User interface as mentioned in a screenshot below by using python-tkinter. The detail on how the GUI is accomplished will explained in the code's description in the coming section.

Following, we are going to explicitly mention and explain the working procedure and the implementation by having a reference with codes we wrote. Here we will take only a snippet of code for illustration purpose. The whole project code is protected for copy right and members interest to expand and do the project in the coming time with the scalable version, hence any interested party can contact us and we can work together on it under some conditions.

To tell the whole with a bird view, the project codes are organized in a separate python script to manage and handle easily by calling a method via their class's instantiate- object. We have the following python scripts:

- ✓ Headshot.py
- ✓ Recognition.py
- ✓ Train_model.py
- ✓ Database.py

- ✓ Otp.py
- ✓ Guest.py
- ✓ Admin.py
- ✓ Main.py

4.1.3 Detail code explanation (implementation)

Here for all the python scripts, we import various python's libraries those are related to computer vision and some others those help to manipulates data. These includes python's tool – OpenCV, for image processing and performing computer vision tasks. We install the package into the raspberry pi python environment, and once installed we used it in a python script by importing it as its aliasing name called "cv2".

Import cv2

Other libraries like OS- which used to manipulate file management and other related tasks, time, SQLite, tkinter, and raspberry pi's GPIO are imported in their respective scripts. The detail is attached in each script below.

The **OS module** in Python provides functions for interacting with the operating system. This module provides a portable way of using operating system-dependent functionality like manipulating directories/paths.

TKinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Time module is used to manage time related tasks.

GPIO pins are imported to declare the pins and related functions in the python script.

SQLite provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language.

Below we discussed each python scripts as follow:

1. *Headshot.py*

This python script's main task is to initiate the camera and store/grape images from an employee's face by the help of the OpenCV package. Inside, there is one class named *Headshot*, and two functions named *__init__* which used to consider the folders and the scripts as a python package for later usage by importing it and *start_headshot*, which contains all the detail to take image from the face. From this script, a snippets code which loop over the frames of person's face and putting in an array of data which will be stored later in a .jpg file is given below:

```python
while True:
    ret, frame = cam.read()
    faces = self.classifier.detectMultiScale(frame, 1.3, 5)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.imshow("AASTU -> Capturing of your photo...", frame)
        img_tobeSaved = gray[y:y + h, x:x + w]
```

And a code to save a gripped image into image directory:

```python
if faces != ():
    imagePath = "Dataset/Employee." + str(self.employId) + "." + str(img_counter) + ".jpg"
    isWriten = cv2.imwrite(imagePath, img_tobeSaved)
    if isWriten:
        print(f'{imagePath} is writen')
    img_counter += 1
```

The tkinter window need to be closed one a script finished the task as:

> cam.release()   #destroying the videocam.
>
> cv2.destroyAllWindows()   #destroying the started tkinter windows.

2. *Train_model.py*

In this script, additional packages such as NumPy which is a Python library used for working with arrays; threading, pandas which used for data manipulation and analysis in particular operations for manipulating numerical tables and time series; PIL python image library which adds support for opening, manipulating, and saving many different images file formats.

The script has a class named Trian and methods are code to perform training model by looking over the whole image directory and finally generating a trainer file. With the help of the OS module, images will be traversed from their respective directory. A snipe code which trains the images on the haar-cascaded classifier and saving the result into .yml is given below:

```python
Ids, faces = self.getImagesWithID(self.path)
self.recognizer.train(faces, Ids)
self.recognizer.save('recognizer/trainingData.yml')
cv2.destroyAllWindows()
```

3. Register.py

In this script, the main task performed is registering new employee by receiving his information like name, gender, email and etc. by using the GUI which relay on the TKinter library. The hardcode for both the logic and the for the GUI is also goes here. The script has a code which validate the users input like passwords and it also ensure the passwords are confirmed before saving with a single trial.

4. facial_reg.py

This is all about recognition- identifying either an employee is a known person or not. By taking a face – which is extended from the start Headshote class. Once the match is done, this method will send commands to the main controller to do following tasks, opening a door or denying the request. Since the comparison needs the database, we import the databse.py script- package as:

from Database import Database

Below we attached a snipe of code which loops on the employee face data and by extracting features from it, comparing along the recognizer from the database encoded file-trainer, and finally

displaying the image with his/her name on the tk windows, which gives an access to the employee. Else it displays, no match found on the tk windows along the face.

```
while True:
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 3)
        ids, conf = recognizer.predict(gray[y:y + h, x:x + w])
        db = Database()
        if conf < 50:
            recognizedId = ids
            name = db.getEmployee(ids)[1]
            cv2.putText(img, name, (x + 2, y + h - 5), cv2.FONT_HERSHEY_SIMPLEX, 1, (150, 255, 0), 2)
        else:
            cv2.putText(img, 'No Match', (x + 2, y + h - 5), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
    cv2.imshow('Face Recognizer', img)
```

5. *mailing.py*

This script is all about managing the two-factor authentication which will communicate between the system itself and the user via an email protocol – SMTP. For such purpose, we import the necessary python's packages called *smtplib* which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP and *email* which is used to manage email messages.

Again, since we need to interact with the data stored in the database, we import the database class here too. First the user will be screened via his username and email password, once he passed this stage of verification, user send request for an OTP means of authentication, the system generate an o and will send to his /her email and the user will enter this token via the keypad and he will get access to the door system. The one-time password is expected to returned within 30 seconds, else it will be expired and the session will be terminated.

A snipe code for a method named *verify* which verify either the entered password is similar with the one generated with in the specified time for the OTP to be valid is attached below.

```python
def verify(self, otpe, otp, age):
    now = time.time()
    if now<=age:
        if otpe == otp:
            print("successful")
        else:
            print('InvalidOTP')
    else:
        print("OTP is expired")
    pass
```

6. keypad.py

This script manipulates the users input like password and username and one-time password for authentication purpose. It has imported the Raspberry Pi's GPIO pins to listen and to write data to the pins via the keypad matrixes. The code below ensures the configuration of the input pins to use the internal pull-down resistors. C1,C2,C3, and C4 are variables those store pin numbers.

```python
GPIO.setup(C1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

7. lcd.py

This script will manage all task those are related to display some alphanumeric data on the 16x2 LCD. As in the keypad.py file, PI's GPIO pins are imported for the same reasons. Once you initialize the LCD pins and related configurations in the raspberry pi, methods such as LCD strings are used to display a string on the LCD display. The snipe code is attached below:

```
def lcd_string(self, message, line):
    if line ==1:
        line = self.LCD_LINE_1
    elif line ==2:
        line = self.LCD_LINE_2
    # Send string to display
    print("line:", line, message)   #for testing/tracking
    message = message.ljust(self.LCD_WIDTH, " ")
    self.lcd_display(line, self.LCD_CMD)
    for i in range(self.LCD_WIDTH):
        self.lcd_display(ord(message[i]), self.LCD_CHR)
```

8. main.py

This script consists of the main part where the starter of the whole system is coded. It mainly contains the admins login page. It asks admin to give his password and username to get access to the system with admin level privileges. The GUI also embedded here in this script too for the login windows. The passwords and the username are considered to be retrieved from a database, but for our case, we just check from a dictionary-python's datatype, which work with value-pair combinations of username and passwords. A snipe of code which checks the username and the password given by the admin with the one stored in the dictionary is attached below. If they are similar, the login page will call the main systems window- from the masterclass.

```
if self.uname.get() in admins.keys() and self.passkey.get() in admins.values():
    master = MasterClass()
    window1.destroy()
    master.showMain()
else:
    tk.Label(frame, text="user name or password incorrect", font=('times', 10, 'bold'),
                fg="red").grid(row=3, column=0, columnspan=2)
```

9. masterwindow.py

This script file is the central and most basic part of the project where most of the admin's GUI is coded, main tasks like registering new employee, and then training the model were called from a separate window from this script.

40

Here, the script also code in order to show attendance in the front windows, along a GUI button those are linked to a separate window to delete and to see the detail including employee lists. Below is a code to send a notification message as the account of the user is removed from the system by the admins for some reason. The code invoked the mailing class to send the notification via an email.

```python
def removed(self):
    db = Database()
    subject = "Account Removed"
    messages = "Since your account is permanently deleted by System admins,you can't access the system
    header = " your Account is removed by admin of the system"
    email = db.getEmployee(int(self.id.get()))[5]

    #db.removeEmployee(int(self.id.get()))
    mailing = Mailing()
    mailing.sentMail(email,subject,header,messages)
```

10. databse.py

The database python script is particularly coded to manage database related tasks only. The script performs tasks including storing employee's primitive data such as name, password, email etc. It also used to see and check a previously stored data by retrieving from the database via the SQLite-supported query language. Below code create an employee table, if not exist, adding a new employee table.

```python
def insertToEmployee(self, trainer):
    connection = sqlite3.connect('trainer.db')
    cursor = connection.cursor()
    cursor.execute('''CREATE TABLE IF NOT EXISTS employee(
                    id INTEGER PRIMARY KEY AUTOINCREMENT,
                    firstName TEXT NOT NULL,
                    lastName TEXT NOT NULL,
                    gender TEXT NOT NULL,

                    department TEXT,

                    email TEXT NOT NULL,
                    userId TEXT NOT NULL,
                    password TEXT NOT NULL,

                    registerdDate TEXT)
    ''')
```

Below a snipe of code is given which will retrieve the data of the employee, particularly only his/her name where his id match with the one passed with the method's arguments.

```python
def getEmployee(self, id):
    connection = sqlite3.connect('trainer.db')
    cursor = connection.cursor()
    cursor.execute(fr"SELECT *FROM employee WHERE id ={id}")
    row = cursor.fetchall()
    connection.commit()
    connection.close()
    return row[0]
```

Having all these implementation details, the circuit we are connecting and wiring in the implementation is shown in the following attached pictures.

Implementations circuit connection, for clarity drawn on fritzing app.
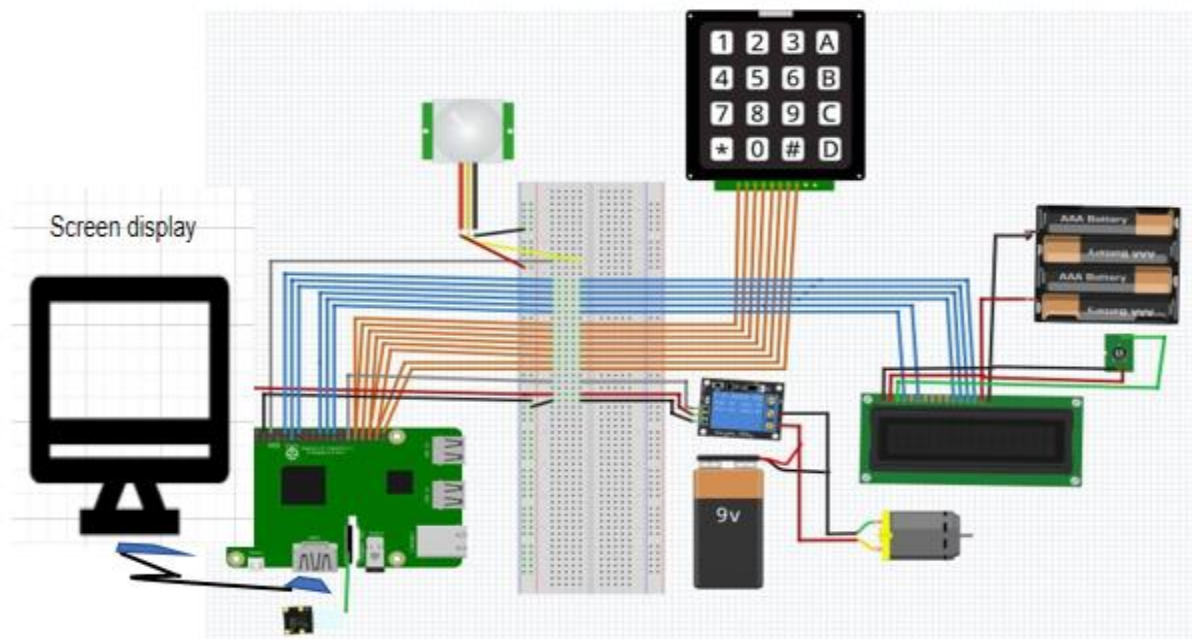
a) sample implementetion circuit connectivity
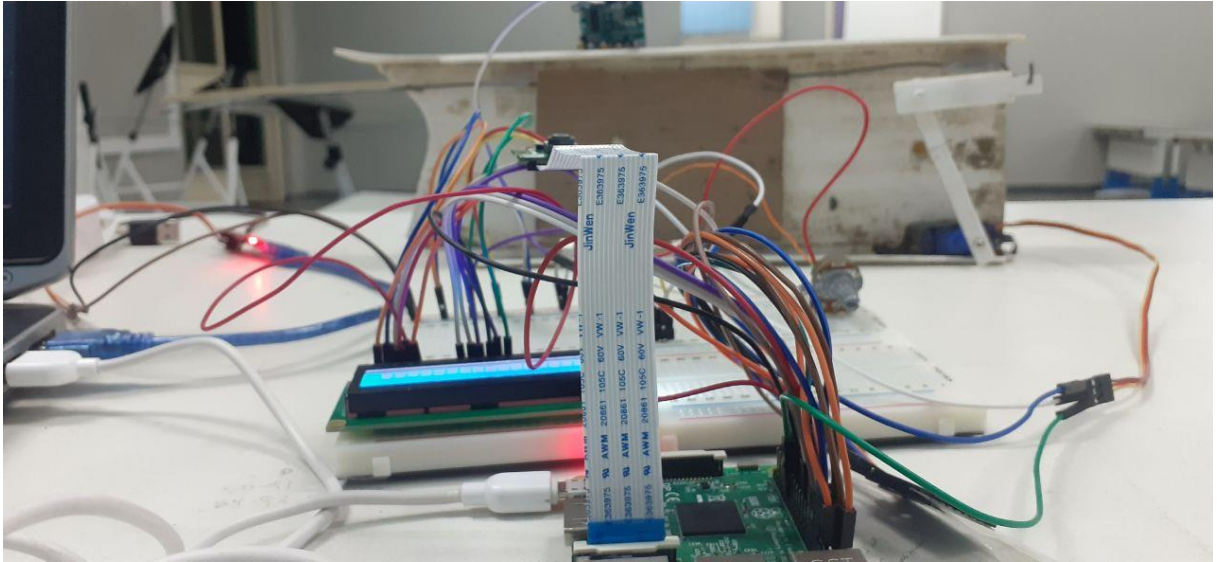
b) sample hardware implementation

Figure 4.1. Connection of Keypad, LCD, and PI camera with the Raspberry Pi. [a) drawn on fritzing, b) image taken by our phone camera.]

Graphical User interface screenshots

In this section, we attached the results we got in the graphical user interface when the project runs. From the very beginning the admin page seems as shown in the screenshot below:
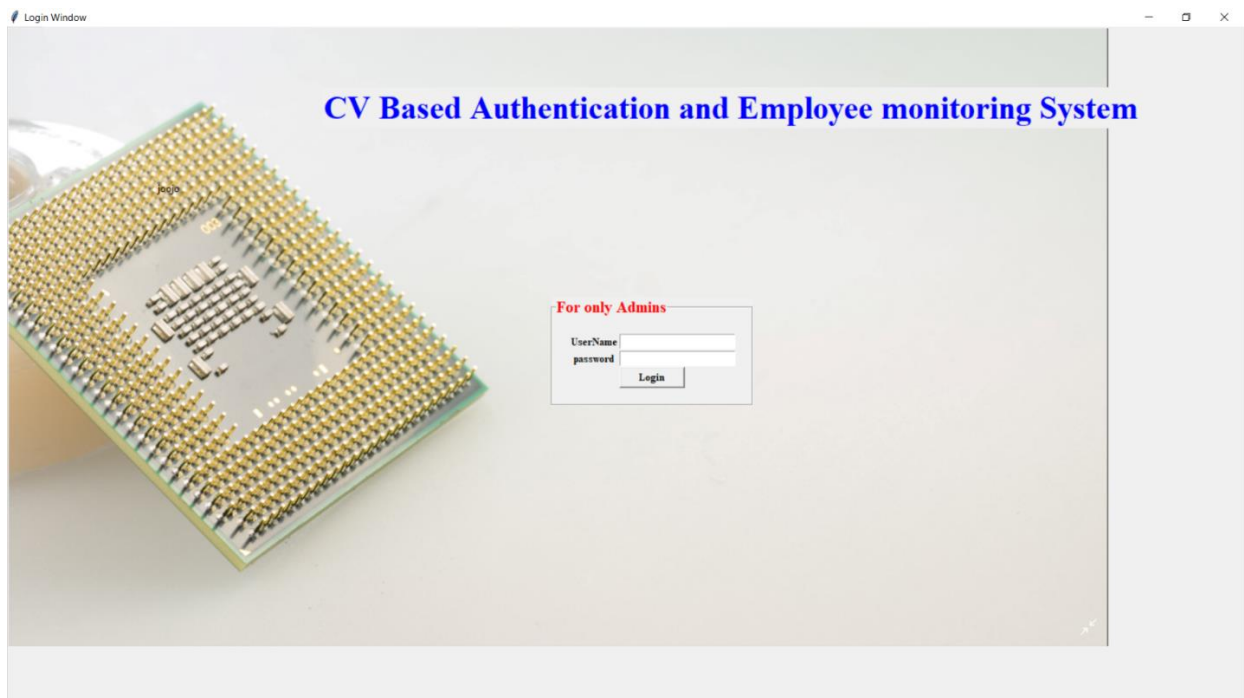
Figure 4.2. admin login page

Below, the screenshot shows the home windows where New Registration, total list, remove, generate attendance, generate report are linked via a button. Each button is linked with a separate window for detail show.
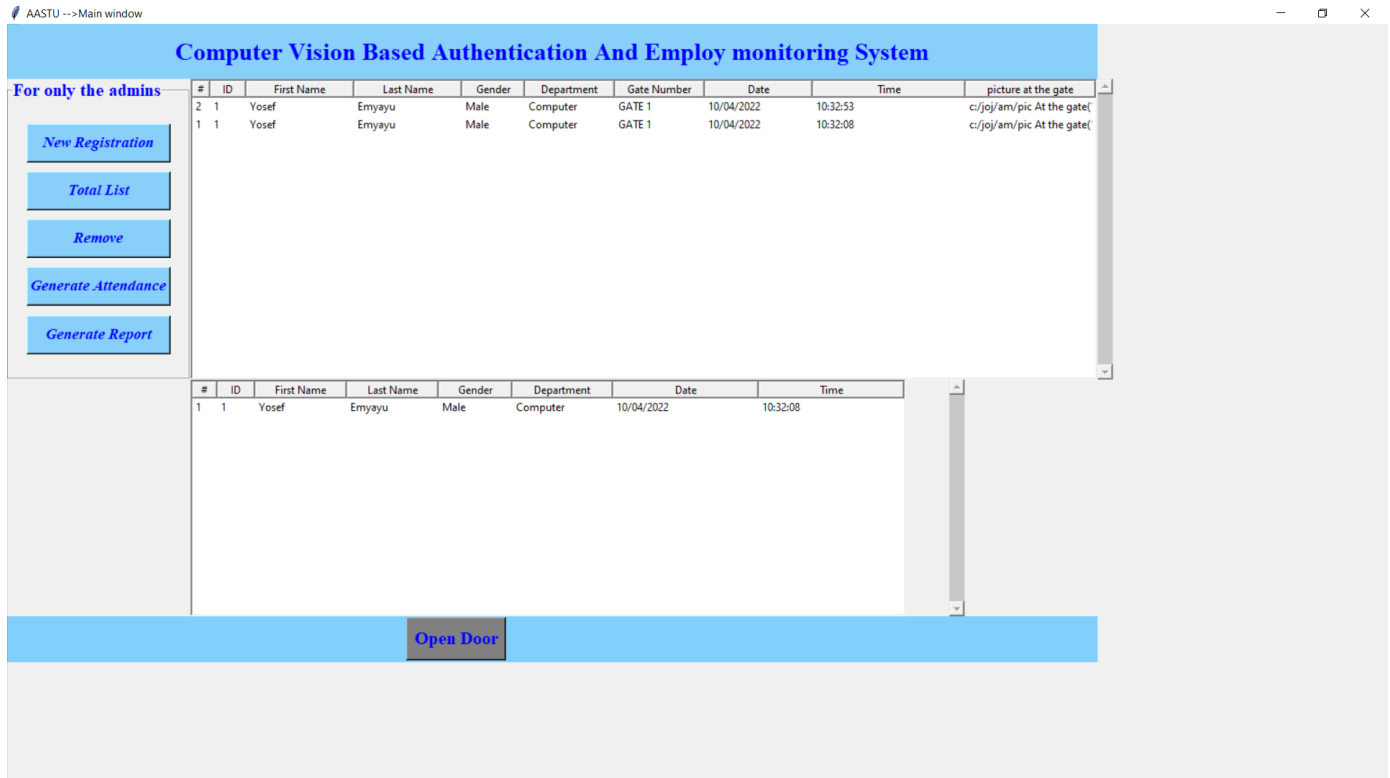
Figure 4.3. Home windows

4.2 Testing and Discussion

In this section, we will mention and present the results we got by testing the designed system via the described implementations with real input data.

User registration

The admin register users in to the system by filling the form as shown in the figure below. After filling the user's name, email, Id, the admin clicks "take photo button", in order to capture the users profile.in this case 30 image are captured and saved on the directory Dataset with a file name the employee index.

Figure 4.4 new employee registration window

Training the model

Figure 4.5 training the model

Face recognition
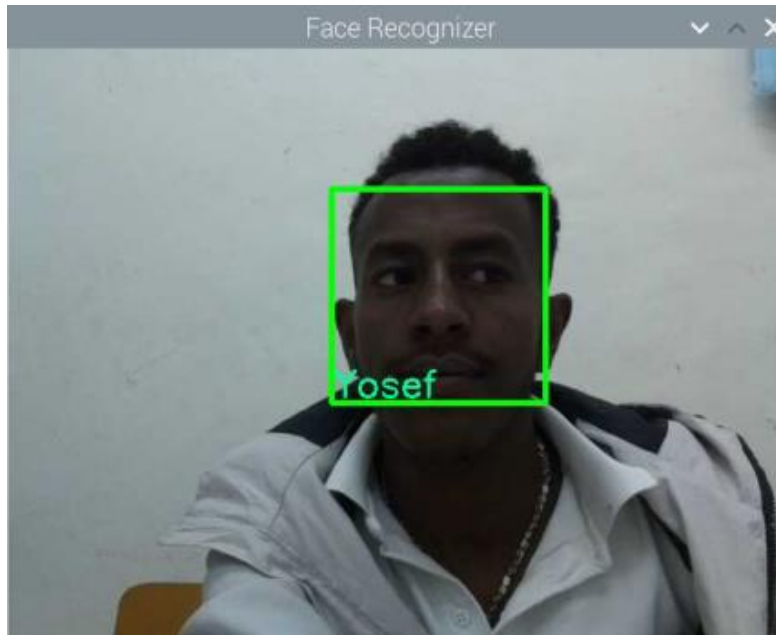
a).
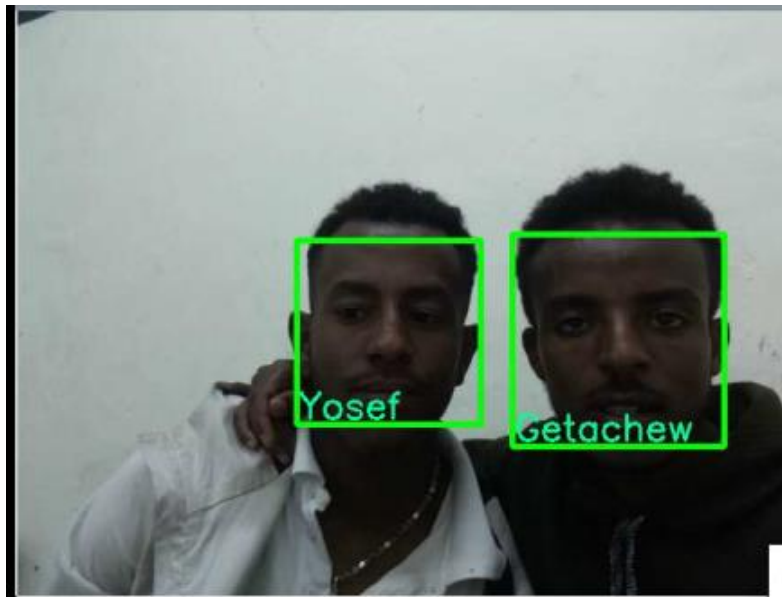
b).

Figure 4.6 Person face recognized, single person(a) and two people(b)

Email verification (OTP) during fog weather condition
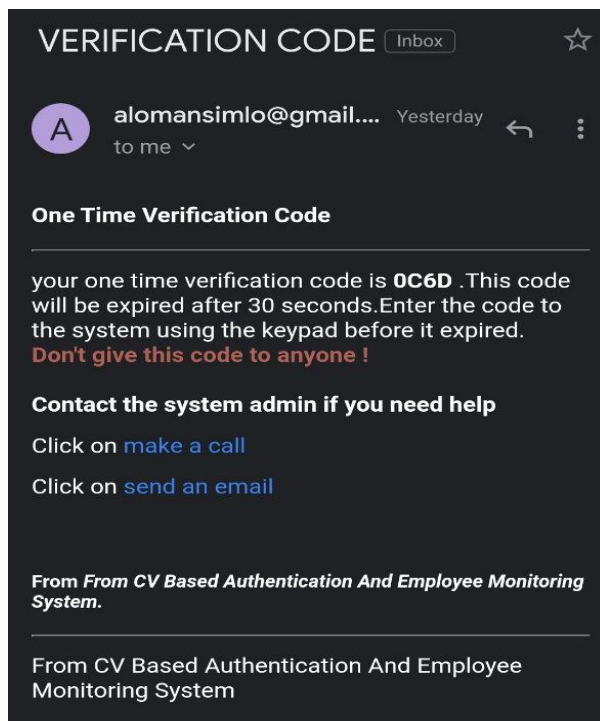
Figure 4.7 Email verification (OTP) during fog weather condition

Guest mode email notification
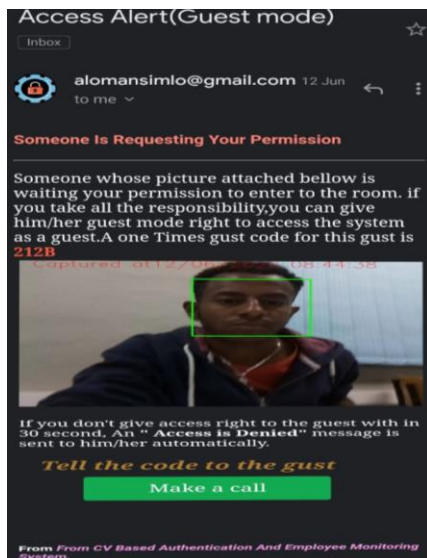
Figure 4.8 Guest mode email notification

The case when admin remove the user from the system
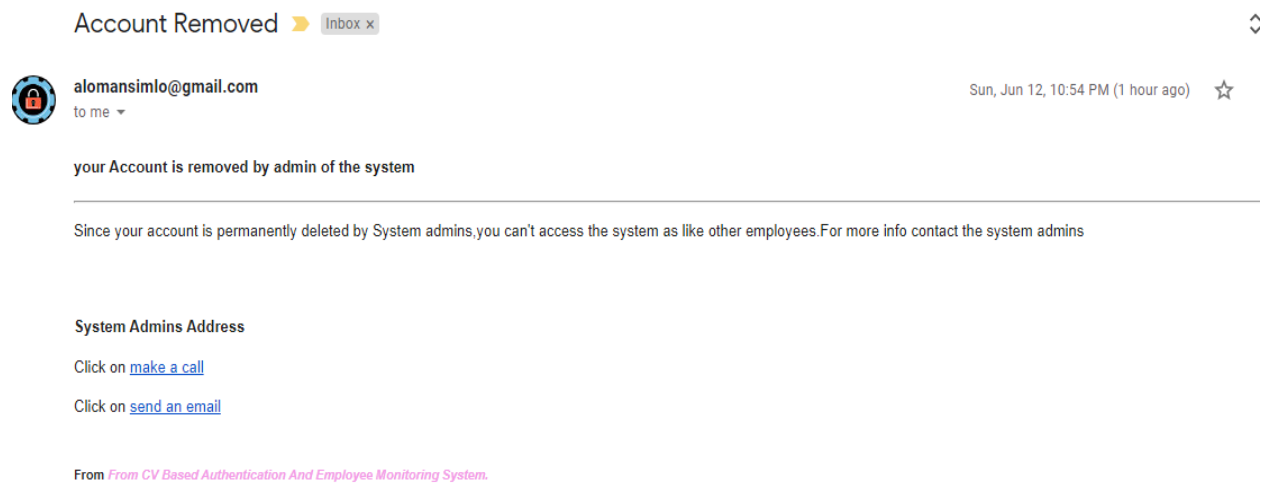
Figure 4.9 admin remove the user

In this chapter we have covered implementation, testing and discussion of the system. In the last chapter, we will cover conclusion and future works .In the future work we included a possible gaps and limitations where the project can be improved by addressing those gaps.

Chapter five

Conclusion and future works

5.1   Conclusion

In this paper we have implemented computer vision-based authentication and employee monitoring system. the authentication is based on face detection and recognition by taking the face images from the streaming video with raspberry pi camera. The user will access the system when they are recognized. The project has user mode and guest mode authentications. The guest needs to ask a request from the admin with an email notification to get access grant. For employee monitoring, the system automatically generates an attendance when the user accesses the room or offices. This will minimize the drawback of manual attendance system.

We have implemented two factor password authentication system with email notifications when there is a fog or smog weather conditions are occurred. In the methodology the training image are stored on the database. we have used local binary pattern histogram LBPH algorithm for face recognition system. the algorithm is work based on a local binary operator, designed to recognize both the side and front face of a human. Finally, for the testing case, when the motion is detected, the camera is initialized. The faces are compared with the previously trained image. If the face is Match, then the system will be recognized, the door is opened and the attendance is generated simultaneously. We had come across designing, developing and testing a face recognition system which is computer vision-based authentication and employee monitoring system.

5.2   Future work

For the future work, we hope the gaps we mentioned above and some limitations we got can be integrated and done in the future including the remote server. To recap, the remote server is a scenario which is an improved version over the project we are implemented which uses a separate remote server that will manage and do tasks those were done in the raspberry pi for efficiency. Storing data such as faces and metadata including user name, user id, email and others on remote server. Both admins and the raspberry pi will access the remote server and each data modification will synchronized. The raspberry pi still will do the main processing task such as the image processing part in detecting human face.

 Level of light intensity to detect an image via the camera is very challenging task, it may be considered as the hard part of computer vision, even we tried to calibrate and fix the issue with some techniques, still it needs another level of improvement to detect and grippe an image from face in any light intensity environment. In the future, there may be a special camera for face

recognition, which can improve the image quality and solve the problems of image filtering, image reconstruction, denoising etc. We can also use 3D technology to supplement 2D images to solve some problems such as rotation and occlusion.

References

[1] "computer vision,," [Online]. Available: https://www.ibm.com/topics/computer-vision.

[2] "Face recognition,," [Online]. Available: https://customers.pyimagesearch.com/lesson-sample-what-is-face-recognition/.

[3] "openCV," [Online]. Available: https://opencv.org/about/.

[4] H. H. Prasad, *International Research Journal of Engineering and Technology ,* p. 4, May 2019.

[5] Pavan Reddy Punnam, Dr.Munaswamy Pidugu, "Design of an Embedded Surveillance System," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),* vol. 7, p. 6, May - June 2018.

[6] Sharvani Yedulapuraml, Rajeshwarrao Arabelli, Kommabatla Mahender, Chintoju Sidhardha, "Automatic Door Lock System by Face Recognition," *IOP Conf. Series: Materials Science and Engineering 981 (2020) 032036,* p. 8, 2021.

[7] K. Tarun Reddy, K. Murali, M. Samba Murthy, G. Pavan, "Intelligent Door Lock System with Face Recognition," *International Journal for Research in Applied Science and Engineering Technology IJRASET,* vol. 8, no. V May 2020- Available at www.ijraset.com, p. 10, 2020.

[8] "Data Flow Graph," [Online]. Available: https://www.sciencedirect.com/topics/computer-science/data-flow-graph.

Appendix

The code for main.py python script [full project code is available from group members, upon request]:

code start here

```python
import tkinter as tk

from PIL import Image,ImageTk

from masterWin import MasterClass

class Main:

    def login(self):

        print("testing")

        window1 = tk.Tk()

        window1.title("AASTU -->Login window")

        # get height and wodth to make the frame full screen

        width = window1.winfo_screenwidth()

        height = window1.winfo_screenheight()

        window1.geometry("%dx%d" % (width, height))

        # bg = tk.PhotoImage(file="bg.png")

        img = (Image.open("bg.png"))

        resized_image = img.resize((width, height), Image.ANTIALIAS)

        bg = ImageTk.PhotoImage(resized_image)

        cavana = tk.Canvas(window1, width=width, height=height)

        cavana.pack(fill='both', expand=True)

        cavana.create_image(0, 0, image=bg, anchor='nw')

       # cavana.create_text(200, 200, text="joojo")

        astuIcon = tk.PhotoImage(file=r"aastuIcon.png", height=500)

        # labelIcon = tk.Label(window1, image=astuIcon).pack()

        # labelTitle = tk.Label(window1, text="CV Based Authentication and Employee monitoring

            System", fg='blue',

        #                 font=('times', 30, 'bold'))

        # labelTitle.pack()

        # cavana.create_window(160+width/2, 100, window=labelTitle)
```

```python
        frame = tk.LabelFrame(window1, fg='red', text="For only the admins", padx=20, pady=20,
                    font=('times', 15, 'bold'))

        frame.pack(padx=20, pady=20)

        cavana.create_window(width/1.5, height/1.3, window=frame)

        label1 = tk.Label(frame, text="UserName", font=('times', 10, 'bold')).grid(row=0,
            column=0)

        cavana.create_window(100, 100, window=label1)

        self.uname = tk.Entry(frame, text="UserName", font=('times', 10, 'bold'))

        self.uname.grid(row=0, column=1)

        # cavana.create_window(100, 100, window=self.uname)

        label2 = tk.Label(frame, text="password", font=('times', 10, 'bold')).grid(row=1, column=0)

        cavana.create_window(100, 100, window=label2)

        self.passkey = tk.Entry(frame, show="*", font=('times', 10, 'bold'))

        self.passkey.grid(row=1, column=1)

        # cavana.create_window(100, 100, window=self.passkey)

        def loginn():

            print('login_test')

            admins = {"user_name":"password"}

            if self.uname.get() in admins.keys() and self.passkey.get()in admins.values():

                master = MasterClass()

                window1.destroy()

                master.showMain()

            else:

                label3 = tk.Label(frame, text="user name or password incorrect", font=('times', 10, 'bold'),
                        fg="red").grid(row=3, column=0, columnspan=2)

        button = tk.Button(frame, text="Login", font=('times', 10, 'bold'), padx=20,
command=loginn).grid(row=2,column =0,columnspan =2)

        window1.mainloop()

system = Main()

system.login()

### end ###
```
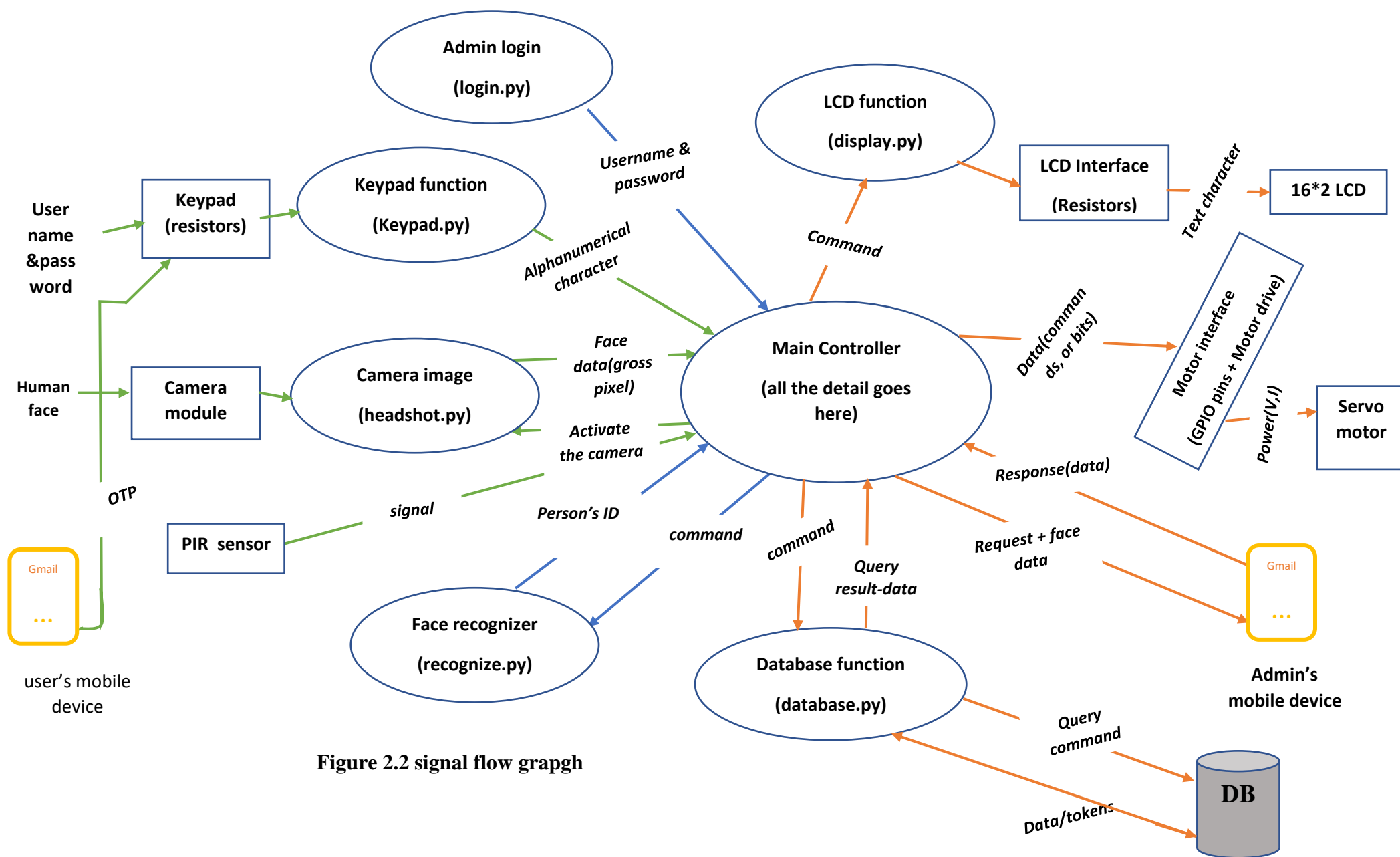
Figure 2.2 signal flow grapgh

- Admin login (login.py)
- Keypad (resistors)
- Keypad function (Keypad.py)
- User name &pass word
- Human face
- Camera module
- Camera image (headshot.py)
- PIR sensor
- Face recognizer (recognize.py)
- Main Controller (all the detail goes here)
- LCD function (display.py)
- LCD Interface (Resistors)
- 16*2 LCD
- Motor interface (GPIO pins + Motor drive)
- Servo motor
- Database function (database.py)
- Gmail ... user's mobile device
- Gmail ... Admin's mobile device
- DB

Username & password
Alphanumerical character
Face data(gross pixel)
Activate the camera
signal
Person's ID
command
Command
Data(commands, or bits)
Power(V,I)
Response(data)
Request + face data
Query result-data
OTP
Query command
Data/tokens
Text character

54