Final Report

EECS 221 Internet of Things

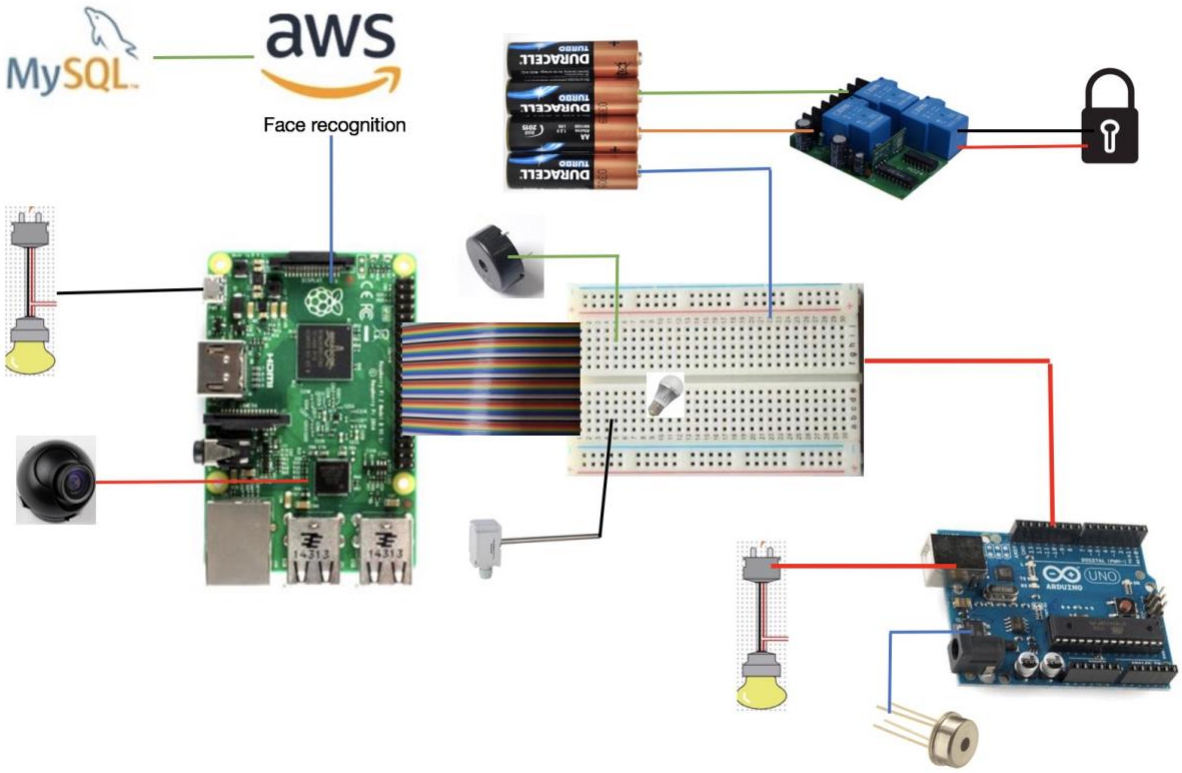
Title: Housekeeper Door System

Le Yu	48286965
Yuanzhe Li	14563990
Jiehui Zhang	21363170

Table of Content

1 Introduction

Our project is called housekeeper door system. The idea is that we designed an intelligent home IoT system which includes outlets, switches, front door camera and other sensors that can be connected to local network. On top of that, we build a chatbot running on the Facebook Messenger, serving as an exclusive home assistant who helps our users to take control of all the IoTs in their homes by talking directly to the chatbot through natural language. Moreover, we have our servers running on the cloud which aggregate all the information from home IoTs and based on the information, they can do facial recognition, automatically open the front door and data and user analysis.



1.1 Motivation

With the most popular technologies like artificial intelligence, machine learning, natural language processing, facial recognition and so on developing so fast, we want to utilize these technologies to build some cool things. Also, with the idea of Internet of Things coming up, intelligent home system has become an indeed popular market, a lot of companies who focus on providing these sensors and control system. However, these central controls or apps is in a way that is not so intelligent where user needs to understand the system and find out which button is doing the right things with the system becomes more and more complicated. As the ultimate goal of computer-user interaction is through natural language communication which largely increases the user-friendliness and scalability. As a result, we plan to integrate some of these technologies, for instance, using natural language processing and machine learning to build up chatbot which

serves as a home assistant who takes care all the control and data; using cloud computing to do facial recognition for family members and making decisions based on the information collected like automatic door opening and so on.

1.2 Background

Nowadays, artificial intelligence has been a heated topic among computer science field. Many big companies, such as Microsoft, Google, Facebook, paid a lot of effort and money in this area. Among those techniques in artificial intelligence, chatbot is becoming more and more popular and is widely used in many applications. As we known, lots of startup companies were purchased by big companies due to their excellent, fascinating and novel chatbot applications. Therefore, we decided to integrate this new technique in our mobile device to provide users natural and easy accessed interaction experience.

Besides, face recognition technology is becoming matured these years. It provides an authentication method without touching or memorizing. It also allows to add more face images in an easy way to learn more identifications of people. There are many face recognition libraries which provide reliable and fast enough face recognition methods. Here, we chose OpenCV, a well-known, stable, widely used, high performance open source image processing library, which contains powerful image processing functions and face recognition algorithms in it.

Moreover, as sensor performance and network speed improved, IoTs have been used in a number of places, including people's home. These low priced, low energy cost, and small sized sensors in IoTs attract many people to develop applications for them, which can be used in everywhere and anytime. Home automation is one of the important applications in IoTs, and it brings a lot of conveniences to us in our daily life. Consider these following scenarios: (1) We forget to bring our key when we come back home; (2) A friend comes to our home but we are far away from the door, or even not at home; (3) A stranger comes in front of the door and we want to know what he does. Our group decide to solve these problems by upgrading current door system.

1.3 Approach & Innovation

We integrate some of the most popular technologies now being widely used in the industry, face recognition, natural language processing and machine learning to provide people with all the conveniences these outstanding technologies can bring to their lives. The whole system is intelligent, user-friendly and the most important, the flexible machine learning mechanism enables it to develop with time when we gather more and more from the users. It is personalized and secure, for which we encrypt all the information where Internet communication happens. Also we changed our app interfaces from building a native app to use chatbot as human – computer interaction. We add the chatbot module which deals with all the text or voice message received from our end users, recognizing their intent and performing the corresponding actions. Moreover, we set up the whole system on the Amazon Web Service(AWS) instead of just local environment, which brings new problems and challenges. However, we managed to solve most of the problems and stated new problems we found during the process.

Still, some further details and improvements need us to implement in the future to make this system work perfectly. For more details, see "Concern" paragraph at the end of each part.

This final report is organized as follows: section 2 describes the whole system architecture from the highest level; section 3 describes the design and interactions between the whole server systems; section 4 illustrates the proper functions and connections of the hardware systems, especially the home IoTs; section 5 concludes the whole housekeeper door system.

2 System Architecture

In this section, we will describe how we decompose the system architecture of our project, including logic view diagram, network communication, system vertical decomposition diagram, deployment diagram, system requirement, and how we ensure the security of our system.

2.1 Logic view and network communication

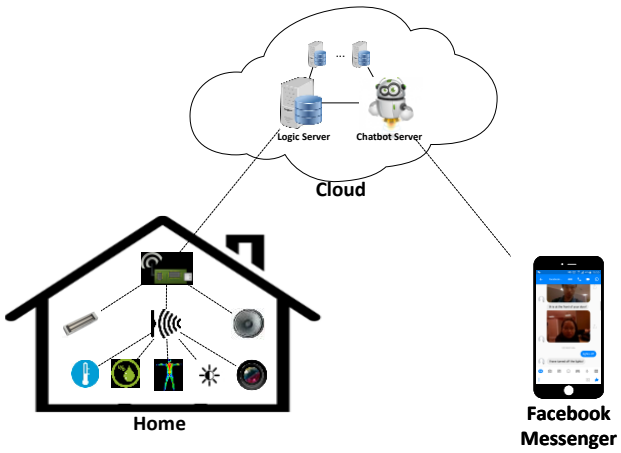


Figure 1 Logic View

Above is the highest abstract level of diagram of our system which includes cloud running on AWS, home IoTs and Facebook Messenger on user mobiles. Home IoTs typically collect data from the home of the user, receive instructions from the server on the cloud and perform corresponding actions. The cloud processes all the information from the users and home IoTs, use the powerful cloud computing potential to make decisions and analyze data. The mobile app is typically a Facebook Messenger serving as a user interfaces. The mobile device will connect with messenger webhook server by HTTP Post through HTTPS protocol. The servers on Amazon AWS cloud, including push server, chatbot server, business server, and database server, will also connect with each other through HTTPS protocol. The Raspberry Pi at home will connect with business server through HTTPS protocol by using Wi-Fi mainly. As for those sensors at home, they will connect with Raspberry Pi through ZigBee protocol to keep the characteristics of low energy cost and mobility.

2.2 System vertical decomposition

Our system architecture which is divided into three layers: client, server and hardware. We use the typical Client – Server(CS) architecture to build up the whole system and the Subscribe – Publish architecture for message passing between each layer.

A) Client: It is only a Facebook Messenger interfaces, through which end users send text or voice message to the Chabot registered on Facebook. Our back – end then deals with the natural language and performs the corresponding actions and give back results.
B) Server: The back – end server side is further decomposed into four servers: Chatbot Server which is responsible for natural language understanding of the message from the user, managing training set and machine learning; Business Server which is responsible for all the business logics, typically receiving instructions from the Chatbot Server, performing corresponding actions, sending instructions to the hardware components and sending back execution results to the users; Push Server which is responsible for message passing between Server and Hardware especially the Raspberry Pi; Database Server which is responsible for all the information related to the end users.
C) Hardware: The hardware layer is controlled in a centralized way by the Raspberry Pi, on which we run the Pi Camera module, facial detection interfaces, message passing interfaces, GPIO interfaces, ZigBee interfaces and Arduino.

2.3 Two working mode

HouseKeeper door system has two modes. One is Keeper mode: detect house outer states and check it is anyone break in; second is waiter mode: check outside situation. Camera is working under both modes. Host could check could check sensors data from cellphone via sending request to Facebook account.

Under the keeper system, host is in house. Temperature sensor works to detecting temperature in house regularly. If the temperature is too high, it means it is firing in house. The door will open immediately. Also since our HouseKeeper system is extendable, gas and smoke sensors could further be added into HouseKeeper system. Those two sensors could help to detect fire and make this system more robust. What's more, if air conditioning is added into our system, houseKeeper system could adjust the indoor temperature from temperature data.

Under the keeper system, host is out of house. No person in house. Pressure sensor works to detecting pressure on window regularly. If pressure on window is too high, it means there is someone touching housing window. There may be a bad condition that robber tries to come in. Once the pressure on window is too high, this situation will be transmitted to host via Facebook. Buzzer will alert to warn robber.

2.4 Deployment and system requirements

Facebook Messenger is a world widely used instant message APP, which can run on Android, iOS, and web platforms. We use Facebook Messenger to provide a natural language talking interface for our users. Our servers are all running on Amazon Web Services, including an EC2 instance and a RDS instance. It ensures us to provide users services with high security, high performance and high quality. As for our home device, we use a Raspberry Pi and a main control program to control all the sensors in the home.
System requirement:
Mobile device: Android smart phone or iPhone with Android 4.0+ or iOS 7.0+
Server: Amazon Linux + Apache 2.4.25 + OpenCV 2.4 + MySQL + Node.js + Python Flask + Mosquitto
Home device: Raspberry Pi 3 + Raspbian + OpenCV 2.4

2.5 Security

We enforce strict security control since the security concerns of the home IoTs and potential privacy problems. All the communications via internet are encrypted using asymmetric encryption, typically in the form of public and private key for decode and encode messages. The server enforces the security criteria designed by Amazon which ensures the security of the main servers. For the server service running on the AWS, we use the Apache which has strong enough defense against common attacks. For the information stored on the MySQL database, we currently only enforce the authentication by using password which is not absolute secure. However, MySQL is run on the server which means the only machine can access the database is the server itself, thus adopting the security provided by Amazon. Later on, we plan to add the disk encryption to encrypt all the data and information stored on the disks, for example, MacAfee for commercial use.

3 Server

In this section, we will describe some details on our server, including push server, chatbot server, business server, and database.

3.1 Push Server

The use of this push server is to address the problem of communication between the raspberry and our Business Server since the Raspberry Pi doesn't have a public IP, the server cannot actively connect to it. One way of doing the communication between server and Raspberry Pi is to use Pulling where Raspberry Pi continuously send http request to the server to get new notifications. However, by doing so, the raspberry pi needs to send http requests frequently which is not energy and resources efficient. As a result, we plan to use another standard way which is to use push. After research on that, we found out that the MQTT protocol developed by IBM is a suitable way in our situation where we aimed at home sensors and controls.

3.2 Chatbot Server

The chatbot server is mainly responsible for reading the input text or voice message from the users and translates them into machine understandable instruction that can be performed accordingly to the lower hardware level. This sub server consists of two subsystems. The first is natural language understanding and the second is the main logic systems.

This system receives a HTTP POST. It first decodes all the information and then passes the text part to the NLP understanding system and waits for its answer. When the NLP system returns the results of the recognition, the main logic system will perform the corresponding actions, typically sending a HTTP POST request to the Business Server and waiting for the results. At last, it forms a JSON formatting data according to the response from the Business Server and sends it back to the Facebook Messenger Webhook as a response.
The communication and execution between different servers and subsystems are shown in the following sequence diagram.

3.3 Business Server

Our business server is deployed on an Amazon AWS EC2 instance with Amazon Linux. It can provide users all day long stable service with high throughput, high security, high performance and high quality. The business server contains three subsystems: face recognition system, main logic system, and user information management.

3.3.1 Face recognition system

The face recognition system on our business server is used to identify the man who stands in front of the door: whether he is home owner, a friend of home owner, or just a stranger. It can provide the result in three seconds and pass that result to the main logic system. The required process for Raspberry Pi could be like this:

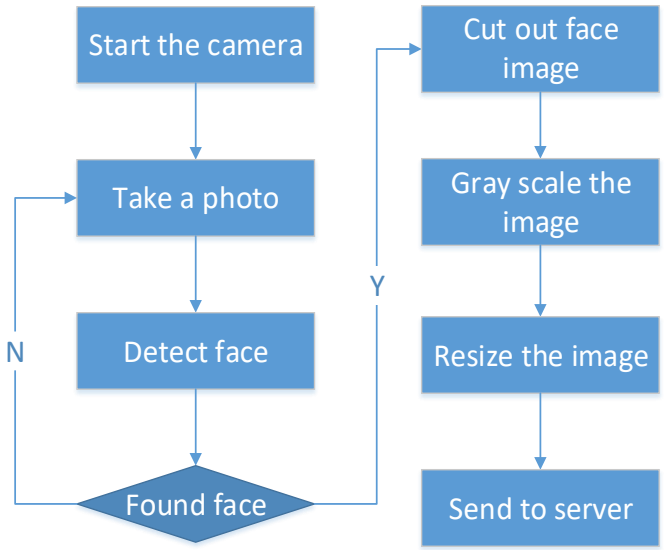


Figure 2 Flow Chart of Detecting Face

3.3.2 Process of Face Recognition

Generally, we use OpenCV library in our face recognition system. OpenCV is an open source, powerful and widely used image processing library. It provides three different kinds of face recognition algorithms for us to use: Eigenfaces, Fisherfaces, and Local Binary Patterns Histograms. All of these algorithms use the following steps to predict the face:

To obtain a better prediction result, we use all these three face recognition algorithms in our system and let them provide their predictions independently. Then we combine their results according to their prediction confidence and give out a final result.

3.3.3 Design of Modules

The main logic system is the core of our business server. It will process the data comes from face recognition system and user information management system, and run the main functions of the business server.

The user can send server several orders to Order module. There are several orders that are available in our system: 'OpenDoor', 'TurnOnLight', 'TurnOffLight', 'HomeStatus', 'ViewPhoto',

'ViewVideo', 'SendAlert', and 'AddFace'. We can use these orders to open the door at home, turn on / off light, check home status (including home temperature, humidity, smog, etc.), view photo / video in front of the door, turn on the buzzer, and add more face images to the training set. These orders will be sent to Raspberry Pi through Notification module.

3.4 Database

We deployed our database on an Amazon AWS RDS instance. It can provide users all day long stable service with high throughput, high security, high performance and high quality. Besides, our database will be backup automatically daily, so that even if sometimes some critical situations happen, we can restore users' data in short time with minimum loss. For all tables in the database, we follow 3NF rule to design these tables to make our data structures clear, concise, and robust.

Currently, this is just a basic design of the database for our project, which satisfies all the demands of our project. In the future, we could add more details in it. For instance, we could add an authority table, which distinguishes whether the owner is parent of the family or just a kid. A parent can have all privileges of his home system, such as adding / removing friends, while a kid cannot, due to some security reasons. Furthermore, we could also add a table to record strangers' information. For example, if a stranger frequently comes to the home, we should add his records and his face images in our database, and inform owners every time he comes.

Name	Homes			
Explanation	Store home information			
Columns	rid	int	Primary key	Identify each Raspberry Pi
	homename	varchar		The name of each Raspberry Pi

Name	RaspOwner			
Explanation	Store the owner relationships for each home			
Columns	rid	int	Primary key	Identify each Raspberry Pi
	uid	int	Primary key	Identify each user

Name	Users			
Explanation	Store user information			
Columns	uid	int	Primary key	Identify each user
	username	varchar		The name of each user
	fid	int		Facebook ID for each user

Name	Buffer			
Explanation	Store the temporary data received from mobile device			
Columns	bid	int	Primary key	Identify each buffer
	uid	int	Foreign key	Identify each user
	type	int		Buffer type
	content	varchar		Buffer content

Name	Faces			
Explanation	Store the training set for face recognition			
Columns	faceid	int	Primary key	Identifies the face of different images, a person can have multiple face images
	uid	int	Foreign key	Identify each user
	path	int		Face image file path

For more details, see face recognition system.

Name	Friends			
Explanation	Store the friend relationships for each home			
Columns	rid	int	Primary key	Identify each Raspberry Pi
	uid	int	Primary key	Identify each user

4 Hardware

4.1 Sensors

4.1.1 Motion detect sensor

We use HC-SR501 PIR motion detector to detect human body. HC-SR501 is based on infrared technology. Its detecting range is about 120 degrees. We adjust its detecting distance to the maximum value (around 7 meters). We also adjusted its delay time to minimum value (around 0.5 seconds). In the hardware board, we choose repeatable trigger, which means if there is human activity in its sensing range, the output will always remain high until the people left. And if there is nobody, the output will remain low. In Raspberry Pi module, a flag value is used to mark the sensor's status. We only take pictures when the value jumps from low to high. In this case, we can avoid taking a lot of photos if someone keeps standing in the sensing range. The sensor's detecting frequency is set to 2 times per second when there is nobody and 1 time per 6 seconds when it detects someone.

4.1.2 Pressure and temperature sensor

We use pressure and temperature sensors on Arduino. It gives analog signal output value. They read pressure data on window and temperature data in house, then send data from Arduino to raspberry pi via serial communication.

4.2 Arduino

We use Arduino UNO to control the door lock and gas sensor. The communication between Arduino UNO and Raspberry Pi is through serial protocol. The Arduino control module keeps waiting for unlock door request from Raspberry Pi. Also, it will send commands to Raspberry Pi if there should be a gas alarm. There is a thread in Raspberry Pi control module which keeps waiting for Arduino serial commands. After it receives gas alarm request, it will start the buzzer and also send an alert to server.

4.3 Interfaces

4.3.1 GPIO Interface

The GPIO mode of Raspberry Pi is BCM mode. We use connect GPIO 25 to the output pin of Buzzer sensor, GPIO 26 to the output pin of relay to control door. and GPIO 12 to the signal pin of the Motion detector. And we connect analog A0 of Arduino to the analog input of temperature sensor and connect analog A1 of Arduino to the analog input of pressure.

4.3.2 HTTP Post Interface

We use HTTP Post to post JSON message to the server. A Python library called requests is used to post the messages. After the camera takes a picture, it is transformed into base64 string form and posted to the server using HTTP. Also, after turning on/off lights, unlocking door, we post conforming messages. Home temperature humidity status and gas alarms are also sent to the server using HTTP post.

4.5 Lock

The lock is driven by 12V/1A buttery power supply. When the lock connects with the power relay, and raspberry pi set HIGH voltage, it will open immediately. The maximum time that the lock is connect to the power supply is 10s. We use relay to control the on-off of the lock. When the raspberry pi receives the command from the server to open the door, it will send the command to Arduino, Arduino will drive the relay to connect the power supply with the lock so that the door will open. In other time, the reply's switch is off.

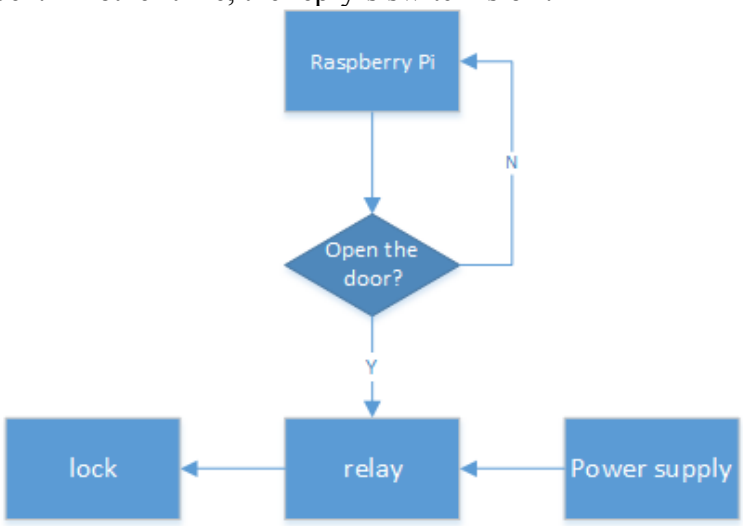


Figure 10 Flow Chart of Door Control

4.6 Camera

The camera we use has a native resolution of 5 megapixels, and has a fixed focus lens onboard. In terms of still images, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. The camera is connected to the BCM2835 processor on the Pi via the CSI bus, a higher bandwidth link which carries pixel data from the camera back to the processor. This bus travels along the ribbon cable that attaches the camera

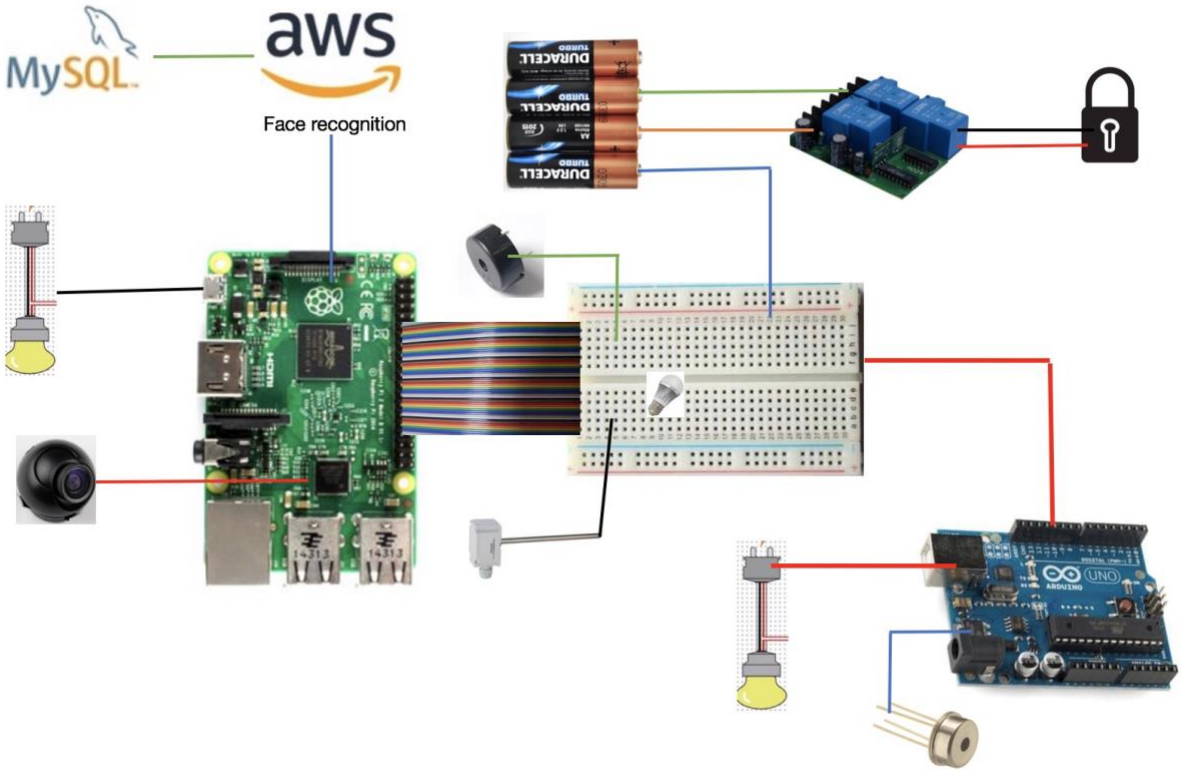
board to the Pi. After the sensors find there is a human out of the door, The Raspberry pi will open the camera and take a image. The raspberry pi will do the face recognition. It will capture all faces in the image and send them to the server.

4.7 Buzzer

The buzzer will make a sound when it is driven by 5 voltage power supply. The raspberry pi could control the buzzer via GPIO. If the raspberry pi give the buzzer high and low voltage at different rate, the buzzer could make different frequency sound. So the user could send alert to the raspberry through the chatbot if necessary and the raspberry pi will launch the buzzer to Alarm the stranger.

4.8 System diagram

Following is our whole houseKeeper door system diagram.



5 Conclusion

Our housekeeper door system uses the intelligent home idea to design and implement IoT system which includes front door camera, temperature sensor, motion detector sensor, pressure sensor, light bulbs, lock, raspberry pi, Arduino chatbot with Facebook, face recognition on AWS server and MySQL database. Housekeeper door system has two mode, keeper and waiter. On top of that, we build a chatbot running on the Facebook Messenger, serving as an exclusive home assistant who helps our users to take control of all the IoTs in their homes by talking directly to the chatbot through natural language. Moreover, we have our servers running on the AWS cloud which aggregate all the information from home IoTs and based on the information, they can do

face recognition, automatically open the front door when host comes, report visitor photo to host via Facebook. Furthermore, housekeeper door system also could detect temperature in house and pressure on window to protect host and house.