Faculty of Electrical and Electronic Engineering Technology

DEVELOPMENT OF VOICE-ASSISTED SURVEILLANCE SYSTEM FOR VISUALLY IMPAIRED PERSON USING SINGLE-BOARD COMPUTER

MOHAMMAD FIRHAD BIN ABU BAKAR

Bachelor of Computer Engineering Technology (Computer Systems) with Honours

2022

DEVELOPMENT OF VOICE-ASSISTED SURVEILLANCE SYSTEM FOR VISUALLY IMPAIRED PERSON USING SINGLE-BOARD COMPUTER

MOHAMMAD FIRHAD BIN ABU BAKAR

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Computer Engineering Technology (Computer Systems) with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSIITI TEKNIKAL MALAYSIA MELAK

2022

DECLARATION

I declare that this project report entitled "Development of Voice-Assisted Surveillance System for Visually Impaired Using Single-Board Computer**"** is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : *Firhad*

Student Name : MOHAMMAD FIRHAD BAKAR

Date : 15 January 2023

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Engineering Technology (Computer Systems) with Honours.

Signature : .....

Supervisor Name : TS. MOHD SAAD BIN HAMID

Date : 20 / 02 / 2023

DEDICATION

To my beloved family, partner and friends for their invaluable guidance, encouragement, and support throughout my journey. Their wisdom and insight has been instrumental in shaping my professional development and personal growth. I am deeply grateful for the time they have dedicated to me and the countless lessons they have taught me. This dedication is a small token of my appreciation for everything they have done for me, and a testament to the profound impact they have had on my life.

ABSTRACT

The purpose of home security is to ensure our safety at home. In the absence of auditory or physical contact, the visually impaired homeowner is limited to recognizing known people in their home, which puts their safety at risk. So, the main objective of this project is to develop a home surveillance system that can recognize predetermined faces and inform the visually impaired owner via an audio signal. The system also allows the visually impaired homeowner to access and operate the programmed device without the need for visual clues since users are guided through the application by a voice assistant. The flow of this project is images are been identified from a single-board computer through the camera that sends information to the Android application. Android Studio is to design the user interface for people with the visually impaired along with the command person's name. HAAR classified applies for face detection to execute several algorithms that process recognition. Firebase is a database where the information will be collected from both devices and the Raspberry Pi is used as a central processing unit for a variety of projects and applications. This project focuses on face recognition, voice assistant, and home automation.

Keyword: security, visually impaired, surveillance, face recognition, audio signal, voice assistant.

ABSTRAK

Tujuan utama keselamatan rumah ialah untuk memastikan keadaan keselamatan sejagat di kawasan rumah. Tanpa adanya sentuhan pendengaran atau fizikal, orang yang kurang jelas penglihatan menghadapi penglihatan yang terhad di kawasan perumahan meraka, ini juga membahayakan keselematan mereka. Objektif utama projek ini adalah untuk membangunkan sistem kawasan perumahan yang boleh mengecam muka yang telah ditetapkan dan akan beri maklumat kepada pemilik terjejas penglihatan melalui isyarat audio. Sistem ini juga membenarkan pemilik rumah terjejas penglihatan untuk mengakses dan mengendalikan peranti yang diprogramkan tanpa memerlukan petunjuk visual memandangkan pengguna dibimbing melalui aplikasi oleh pembantu suara. Pengaliran sistem ini dimana gambar yang diperoleh dari komputer papan tunggal melalui camera akan menghantar maklumat ke aplikasi Android kemudian mengeluarkan suara untuk penama. HAAR kelas digunakan untuk mengimbas muka untuk dilaksanakan sebahagian algoritma untuk pengesahan dan juga Raspberry Pi digunakan sebagai unit pemprosesan pusat untuk pelbagai projek dan aplikasi. Tujuan utama projek ini untuk pengesahan muka, pembantu suara, dan automasi rumah.

ACKNOWLEDGEMENT

In the Name of Allah, the Most Merciful, the Most Compassionate, Alhamdulillah all praises belongs to Almighty Allah, the Lord of the worlds and prayers and peace be upon Muhammad His servant and messenger.

I would like to express my sincere gratitude to myself who have supported me throughout the journey of this project. The countless hours of research, the late nights and early mornings, the brainstorming sessions and the problem solving, all of it has paid off in the end.

I would like to acknowledge the unwavering support of my family, friends and colleagues who have provided me with the encouragement and motivation to pursue this endeavour.

I am also grateful to my professors and mentors who have provided me with the knowledge and guidance that was essential in completing this project. I would also like to acknowledge the support of the organizations, who provided me with the resources and facilities necessary to complete this project.

I am thankful for Muhd Farieez who have supported me through thick and thin, guided me and encouraged me to pursue my passions, I am truly blessed. His support have been a constant source of inspiration and motivation, and I am eternally thankful for all that he have done for me.

Thank you all for your contributions and support, this project would not have been possible without you

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

INTRODUCTION

This chapter will study about the background of the project, problem statement, objectives, scope of the project, and the project outline.

1.1 Research Background

Blindness and vision impairment impact 37 million and 124 million people globally, respectively, according to the World Health Organization [1]. Many visually impaired people face lots of difficulties doing some tasks in daily life. According to studies, the disability of visual might affect a person's quality of life by limiting social interactions and interdependence along with security in their surroundings [2]. Protecting privacy, security, and safety in both the actual and virtual worlds are key issues that practically everyone encounters. However, for other groups, such as the visually impaired, these issues are particularly severe. Even though there are several types of tools available to overcome these challenges, they are still not sufficient because they require an external assistant

Throughout these studies, implementing a voice-assisted surveillance system probably could help them interact with people around them. The internet and user interface are blind to the visually impaired, but to avoid this, provide an end-to-end voice-based software solution for the visually impaired that allows them to use the devices with minimum to no keystrokes. Instead of utilizing a keyboard, the user will click the button to speak the orders to make them acknowledge the actions. The program then converts the text-to-speech, which will be the command to be performed based on the Android application.

1.2 Problem statement

Sight and vision are vital because they provide us with beauty and knowledge of the world. They also protect us from any consequences. Both views work together to aware of the hazards. The person's vision assists them in recognizing the meaning of the situation and jumping to conclusions. They tend to get more risks than other normal people. Having to struggle to recognize mankind is probably one of the primary challenges for people with visual impairment. Home is not a safe place if a lacks security which could be dangerous when no monitoring system can notify them.

Figure 1 Visual impaired sight [3]

To overcome this issue, face recognition is one of the ways as an external assistant to verify or identify the subject image, video, or audio-visual aspect of faces. Detecting the face would make the software analyze the structure of faces and convert the image into data. In terms of adding the speech or voice assistant makes the user easier to know the actual subject and give the information through audio signals.

1.3 Project Objective

The main objective for this project:

1. To develop a mobile application based on a surveillance monitoring system using a single-board computer.
2. To develop a home automation application that controls several devices using internet access.
3. To implement voice assistance features for visual impaired person.
4. To perform validation and verification of the surveillance monitoring tools and home automation

1.4 Scope of Project

The scope of this project focuses on using a single-board computer and Android application to implement face recognition, voice assistant, and system surveillance which allows visually impaired people to communicate with devices. The appearance of face recognition and system surveillance will be able to detect the person and surroundings. In addition, the single board as a core, by installing the Android application on the phone, voice commands may be delivered from the phone. The single-board computer is connected to the mobile device through a cloud database. The single-board computer receives these commands and processes them based on the apps. The smartphone will notify if there is any presence of humans around them if the camera detected the face by using a voice assistant.

1.4 Project Outline

This report is divided into five chapters that detail the “Development of voice assisted surveillance system for the visually impaired person using single-board computer” project's implementation.

The background of the development of the voice-assisted surveillance system for the visually impaired person using a single-board computer is presented in the first chapter. A problem statement is given, along with a list of objectives to be met in

order to address the difficulties. This chapter discusses the study scope and project outline.

The literature review is based on the second chapter. This section contains a discussion of relevant research conducted by researchers based on the project's implementation and functionality. A comparison of the projects is carried out in order to determine the fundamental concept and theory, and offer a wide picture of the essence of execution that will be suitable for this research.

The methodology to execute this project is detailed in the third chapter. The approach is accomplished by following a set of procedures to build this project while adhering to the given objectives. In addition, a flowchart is created to show how this project system works in its entirety.

The specifics of the outcomes acquired from the performance of this project are presented in the fourth chapter. In addition, in this chapter, the discussion of the analysis based on the project's outcomes and discoveries is clearly finished.

The final chapter essentially wraps up and highlights the important points, as well as determines if the project's output met the stated goals. Finally, there will be a part in this chapter that provides recommendations for future improvements to this project using emerging technologies.

LITERATURE REVIEW

This is the literature review of the studies and research based on the important information and details for this project was gathered from many resources such as articles, journals, books, and the internet regarding the related previous study. Hence, this chapter's studies begin with a surveillance system using face recognition, and the audio signal. It is essential to carry out this study on these topics because they are the project's primary purpose. Besides that, the medium of this project involves single-board computer technology which is important to learn the principles of technology to have a clear view of the scope.

2.1 Introduction

Generally, a surveillance system consists of a camera system, monitors or display devices. Cameras come with a wide range of design options such as analog or digital. This system can be used at a structure or property either in indoors or outdoors. To record all of the time, only record in reaction to movement, or only record at specified times of the day can be programmed by the system.

According to [4] object detection, face recognition, and face identification are a combination of multiple techniques that the surveillance system technologies can implement in order to identify intruders and monitor target regions. Face recognition has received a lot of interest because of its various uses in confidential information, security policy enforcement, and inspection.

Sector	Applying
Security	Access Control. Boarding system. Workstation access.
Law Enforcement	Forensics. Analysis.
Database Investigation	National Identity Document. Vehicles Registration. Licensed drivers.
Inspection	Monitoring CCTV Portal control

Table 1 Face recognition application [5]

Figure 2 Surveillance system devices [6]

2.2 Face Recognition

Face recognition is a vision system capability for identifying people based on their facial features. Other than that, it will provide information when determining a human face presence belonging to the size, position, and location of each facial structure. To discover face identity, the face is retrieved and compared to a known face detection when the identity characteristic appears [7]. Furthermore, there are several components of facial recognition in the following sequence which are image collecting, image pre-processing, feature extraction, matching, and combining will be integrated with hardware cameras and computing devices.

2.2.1 OpenCV HAAR Cascade Classifier Algorithm

OpenCV or known as Open-Source Computer Vision Library often uses HAAR cascade classifier for human face detection. This technique was first presented by [8]. The given image is inserted to determine features by HAAR. The sum of pixels under the black rectangle will be subtracted the sum of the pixels under the white rectangle is calculated to derive a single value for each feature.

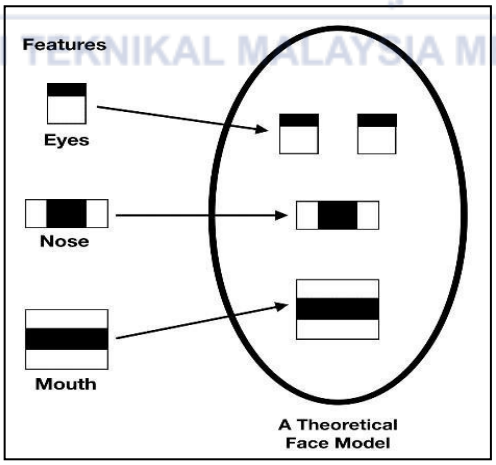


Figure 3 HAAR features [9]

The HAAR features begin the checking image for the detection of the face starting from the top left corner until the end of the face from the bottom right corner in the image. There is the calculation for every pixel as shown in (1).

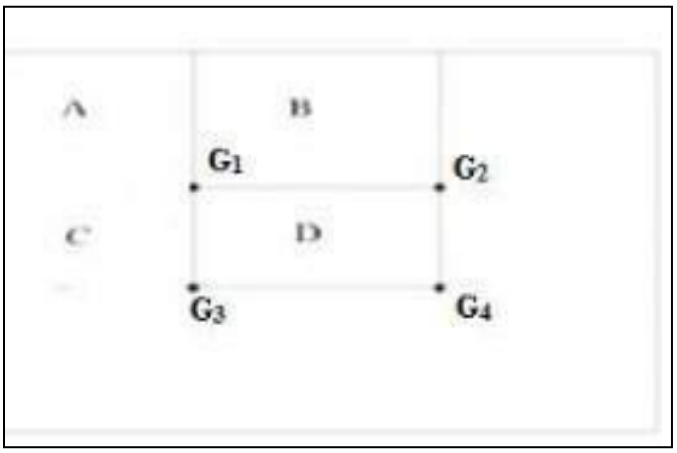


Figure 4 Diagram of HAAR features [8]

The Viola-Jones method begins analyzing these attributes in the provided picture with a base window size of 24x24. All potential HAAR feature properties could be unrealistic to count 160,000 and above features in the window regarding the properties such as location, scale, and type.

As a result, the AdaBoost algorithm would be the solution to overcome this issue. AdaBoost is a machine learning method capable of identifying the best features. This kind of method gathered some of the weak learners into the strong learners.

2.2.2 Histogram of Oriented Gradient Algorithm

Figure 5 Histogram of oriented gradient parameters [10]

The Histogram of Oriented Gradients was introduced by Navneet Dalal and Bill Triggs [11]. It is one of the face recognition feature descriptors usually used to extract features from image data pixel by pixel for image processing, mainly for object detection. In addition, it will change the picture to greyscale images. To extract unique features, every image must have a particular gradient orientation. The image will break down into smaller regions for each region. After that, the gradients and orientation would be calculated. The principle behind the algorithm is that local object appearance and shape by the distribution of intensity gradient or edge directions within the image can be described.

2.2.3 Support Vector Machines

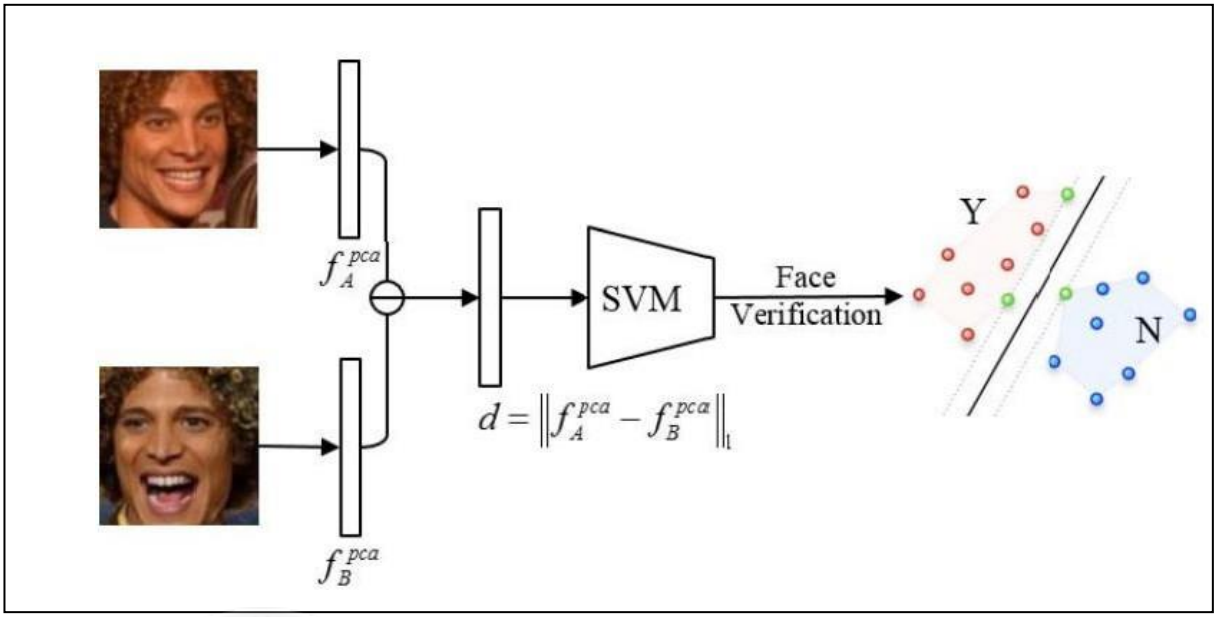


Figure 6 Support vector machines [25]

Support Vector Machines (SVMs) can be used for face recognition by first training the SVM with a set of label face images, where the label corresponds to the identity of the person in the image. Once trained, the SVM can then be used to classify new, unlabelled images of faces. In a typical SVM-based face recognition system, the input to the SVM is a feature vector extracted from the image. This feature vector could be based on various image features such as the intensity of pixels, edges, or the geometric structure of the face. The SVM then finds the best boundary, or hyperplane, that separates the different classes of faces in the feature space. During the recognition stage, a new face image is transformed into a feature vector and then classified by the SVM according to which side of the hyperplane it falls on. The class label corresponding to the side of the hyperplane the feature vector falls on is then taken as the predicted identity of the person in the face image. SVM is a powerful technique for face recognition because it can handle high-dimensional data, is robust to noise, and can be used with non-linear decision boundaries. Support vector machines (SVMs) are developed to address a traditional two class pattern recognition problem. By altering the way an SVM classifier interprets its output and creating a

representation of facial images that is compatible with a two-class issue, which adapts SVM to face recognition [25].

2.3 Voice Assistant

A voice assistant as the fundamental of voice control allows user interface operation as hand-free of a digital device that listens to particular vocal commands and returns relevant information or performs specified actions as desired by the user using speech recognition, voice synthesis, and language processing algorithm. In another word, it can deliver relevant information based on particular instructions, also referred to as intentions.

Nevertheless, [13] voice assistants can create a sense of connection, comprehension, and enjoyment. Researchers in [7] perform similar work, proposing and validating a theoretical model for voice assistants. Task attraction, social attraction, and physical attractiveness are the important elements influencing gadget adoption. It also helps the user easy to interact with the system and more user-friendly.

Areas	Microsoft Cortana	Google Home	Amazon Alexa
Skill	Microsoft has recently deleted third-party services.	It only applies to a few services	It offers a variety of services, such as cab booking and monitoring bank balances. Users can also adjust the needed expertise.
Voice accuracy and speed	Has considerably harder trouble hearing what you're saying than the other two. Has a slow reaction time when responding to inquiries.	A user's voice can control it. "Ok, Google" is the word that wakes it awake. It will not detect voice instructions if have a certain accent.	It outperforms both Cortana and Google Home.
Features and capabilities	Cortana utilizes Microsoft's Bing search engine to conduct searches and reply to information requests.	Use of Google Assistant for information searches and responses using Google	Alexa uses the Bing search engine to conduct searches and reply to information requests

Table 2 Voice assistant areas [14]

2.4 Related Previous Project

The study of previous relevant projects that are primarily focused on voice-assisted surveillance systems is vital to have a good understanding of the project and to getting fundamental pieces of information to meet the project's objectives. For the visually impaired, this project includes a voice-assisted surveillance system. As a result, this section will analyze five past projects that use a similar strategy and have a comparable goal to accomplish the main goal of this project.

2.4.1 CNN-Based Object Recognition and Tracking System for Visually Impaired.

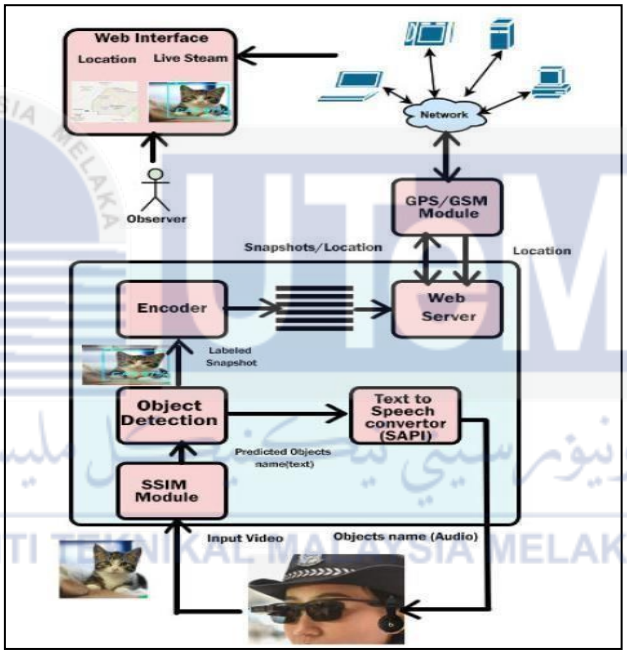


Figure 7 Flow diagram of CNN-Based Object Recognition and Tracking System for Visually Impaired [15]

Based on the project that was done by [15] has made a clear vision of various methods to object recognition and tracking system. The authors are using Raspberry pi as a mini-computer for digital signal processing which provides 4G mobile data and also as a GSM module for GPS positioning information. Other than that, the camera is used for a captured live feed from the video camera and sent to the object detection and recognition module. After that, the text-to-speech converter module would pronounce the name through headphones regarding the

current video frame. The Joint Photographic Expert Group encoding camera shot label. The server database saves the user's location with the encoded image. The rest of the family members can monitor or track their movement through a web interface.

2.4.2 Face Recognition using MATLAB

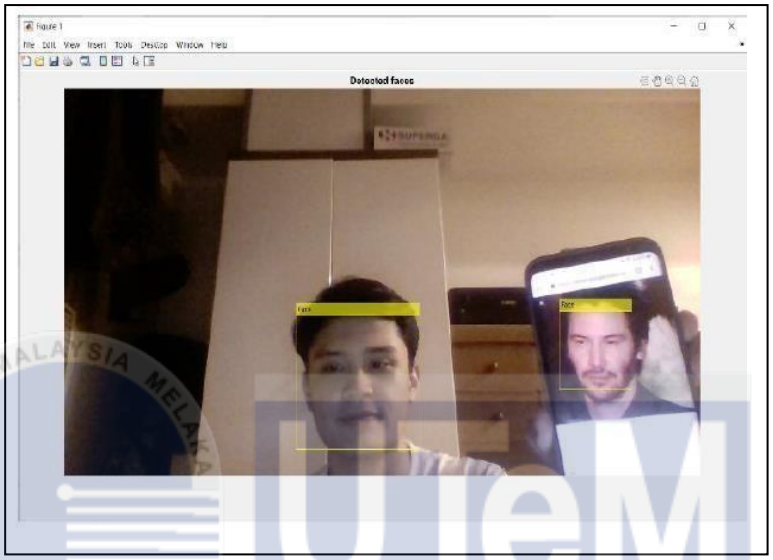


Figure 8 Output of the Face Recognition using MATLAB [16]

According to [16] the facial recognition program's digital communications system begins with the hardware. The hardware for this system is a laptop with a built-in webcam that will take a photo of the subject to be analyzed. After the shot is taken, it is digitally processed on the computer. The digital copies of the images would be saved in a database and used later. The application was created to capture a photograph, assess focus, and crop around the subject's face. After the database is updated with photographs of a subject's face, the application converts the photos into vectors so that they may be easily recognized by the computer. Following the procedure, the computer turns all of the photographs in the database into 2D vectors. The computer is now ready to receive input of a face to be inspected once it has a database of 2D vectors. A photo of a face will be used as the computer's input, and the computer will turn the photo into 2D vectors. After that, the computer must determine the 2D vector in the database that is the most comparable to the 2D vector in the input. The photo with the most similarities to the input would now be

identified as the comparable face, and the photographs would be shown in 3D image form for the user to inspect. This project, will be using a histogram of oriented diagram algorithm classified.

2.4.3 An efficient framework using visual recognition for IoT based smart city surveillance

As stated in [17] module training and store dataset are utilized for high-end servers. For the usual dataset in the cloud or inside IoT devices saved the computed features and the trained model. The model is deployed in real-time using the IoT gadget thus, CCTV for real-time input is connected to the gadget. The device's trained model will serve as an identification module for a variety of applications, including industrial security surveillance, vehicle security, and employee recognition by face. The IoT gadget will display on the smartphone to get information.

Figure 9 Proposed architecture an efficient framework using visual recognition for IoT based smart city surveillance [17]

2.4.4 A cloud-based monitoring system via face recognition using Gabor and CS-LBP features

By [18] based on the project flow, the distributed client is installed for the face detection module. When the client computer detects an image of faces, the data automatically upload to the cloud storage. The data contains face count, client IP address, and image ID. The data visualization interface provides the system to display person distribution statutes. An HTTP interface as an administrator for a face recognition module must input a specific face image that needs to be localized. The image would transmit to the cloud for all the clients. Face recognition algorithm function whether the person in the face image is detected by their cameras. The recognition result will be transferred to the cloud.

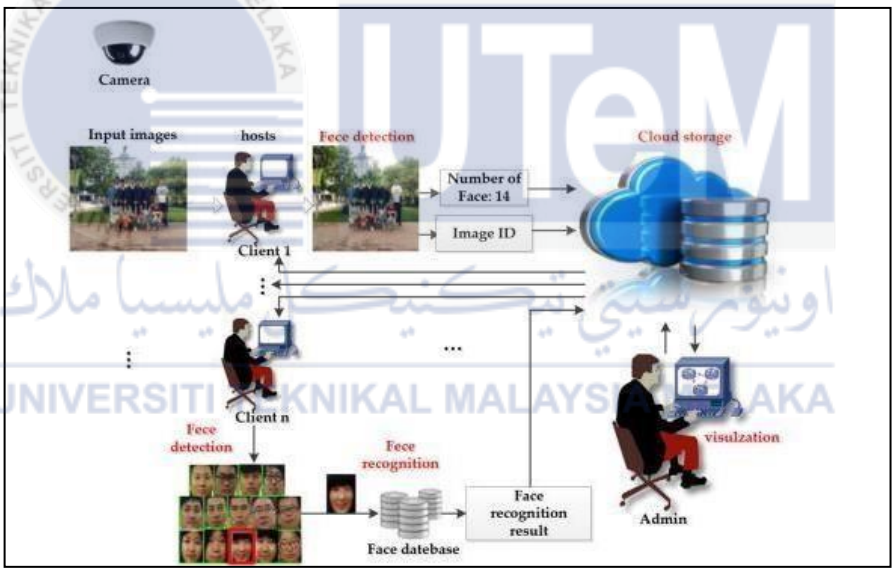


Figure 10 A cloud-based monitoring system via face recognition using Gabor and CS-LBP features [18]

2.4.5 Face Recognition Authenticated Voice Assistant System for the disabled

Based on the [19] wanted to propose a system is being developed to keep blind, elderly, and crippled persons engaged and working independently. The authentication mechanism is a face recognition system built with OpenCV. Authenticated users can use the virtual assistant system, which uses voice instructions instead of the traditional keyboard and mouse to do tasks. This system takes a voice command as an input request through a microphone and, depending on the request, performs web automation with Selenium, web scraping to get any information from a Wikipedia article, machine reading comprehension with the BERT model, requests for the current time using the time module, or serial communication to control the inbuilt LED of an Arduino board. The text-to-speech library converts the contents of the screen into speech. The user's request is responded to in the form of a voice.

Figure 11 Flow diagram of Face Recognition Authenticated Voice Assistant System for the disabled [19]

2.5 Sample formulas

G1=A, G2=A+B, G3=A+C, G4=A+B+C+D

G1+ G4 -G2-G3=A+A+B+C+D-A-B-A-C=D

According to the HAAR features under face recognition sub topic page 9

(1)

2.6 Comparison Parameter

No	Author	Algorithm	Classifier	Application, device and mechanism	Advantage and Disadvantage
1	[15]	Deep Learning	Convolution Neuron Network	- GSM Module - Raspberry pi - SAPI module - Web-Based - Android Studio - OpenCV	• **Advantage:** -Detecting most object along path - Listen to object name. - Detecting location. • **Disadvantage:** - Detecting road curbs. - Changes in the surface of roads.
2	[16]	Machine Learning	Histogram of oriented diagram	- Database - MATLAB	• **Advantage:** - Detect more faces at the same time. • **Disadvantage:** -Surrounding with have reflection will reduce accuracy.

3	[17]	Machine Learning	Linear Binary Pattern	- Raspberry pi - Cloud database - IoT Gadget - OpenCV	• **Advantage:** -Performing visual recognition efficiently with minimum feature size in minimum timespan. - Optimum power consumption. • **Disadvantage:** -Sometimes the features which are dropped or neglected were redundant and have no impact on accuracy.
4	[18]	Deep Learning	Linear Binary Pattern	- Cloud Web-Based - Java Native Interface - MATLAB - MyEclipse - OpenStreetMap	• **Advantage:** - This method is more robust to illumination variations and has lower complexity. • **Disadvantage**: - Low speed recognition
5	[19]	Machine Learning		- OpenCV - Arduino - Microphone - Selenium	• **Advantage**: - It can identify user requests exactly with high accuracy and performing • **Disadvantage** - Not using cross-platform - Limitation language

Table 3 Comparison Parameter

2.7 Summary

After a thorough observation and reading have done based on the previous related project. It is understandable most projects use raspberry pi commonly to detect and identify the subject as an improvement from the previous project. In addition, the project concept of Raspberry pi can be implemented in this project due to its practical specifications. The research on voice-assisted surveillance system-based projects assists in achieving the objectives of the development of monitoring and home automation.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter explains the methods used to create the project and reach the objectives. This chapter is divided into research design, process flow elaboration, and hardware specification. To verify that the project's flow remains consistent, detailed research on the used hardware was conducted to better understand how to handle it and the appropriate model to use for this project. This chapter is also important for getting a broad overview of the project flowchart. The process flow is detailed, and the hardware specifications will be detailed after that. Last but not least, a diagram showing the project's relationship is presented and briefly explained in this chapter.

3.2 Study Design

The project aims to develop a voice-assisted surveillance system for visually impaired people. Essentially, Raspberry pi as a single board is used as the main component of the project which is Raspberry pi 3 model B. Besides, a camera is used for surveillance systems and face recognition thus smartphone is used for monitoring along with sending an audio signal to the devices. Moreover, this project consists of a face recognition algorithm while OpenCV-Python is a Python binding library for solving computer vision challenges. Other than that, this project allows the user to control several devices such as home automation.

3.3 Project Overview

Planning	Data collection
	Hardware and software requirements
Implementing	A point of inspection
	Implement to the project
Analysis	Observe and evaluate the performance
	Determine the outcome.

Table 4 Sequence of project overview

3.4 Planning

This final project's planning must be done well to ensure that all of the project's requirements can be met within the time frame provided. All project activities are carried out by the week's schedule and previous preparation.

3.5 Data Collection

Data collection is an important part of the project's completion. Before the project can be completed, data from all sources should be collected to identify the hardware and software requirements. To learn more about this project, all of the data collected from journals, textbooks, and research papers are applied. The comprehension of this project is increasing as a result of the material obtained from the internet and the library.

3.5.1 Cost estimation

Component	Quantity	Price
Raspberry Pi 3B+	1	RM 350.00
Webcam, Fantech	1	RM 110.00
Bulb	1	RM 1.50
Fan	1	RM 15.00
Buzzer	1	RM 3.00
Relay 4 channel	1	RM 8.00
		RM 487.50

Table 5 Cost estimation

3.6 Flowchart

The most important part of ensuring that this project succeeds is project planning. Proper planning should be emphasized to ensure that the project is completed and that the issues encountered throughout the project are minimized. To accomplish the project on time, it is necessary to create a methodical flow chart.

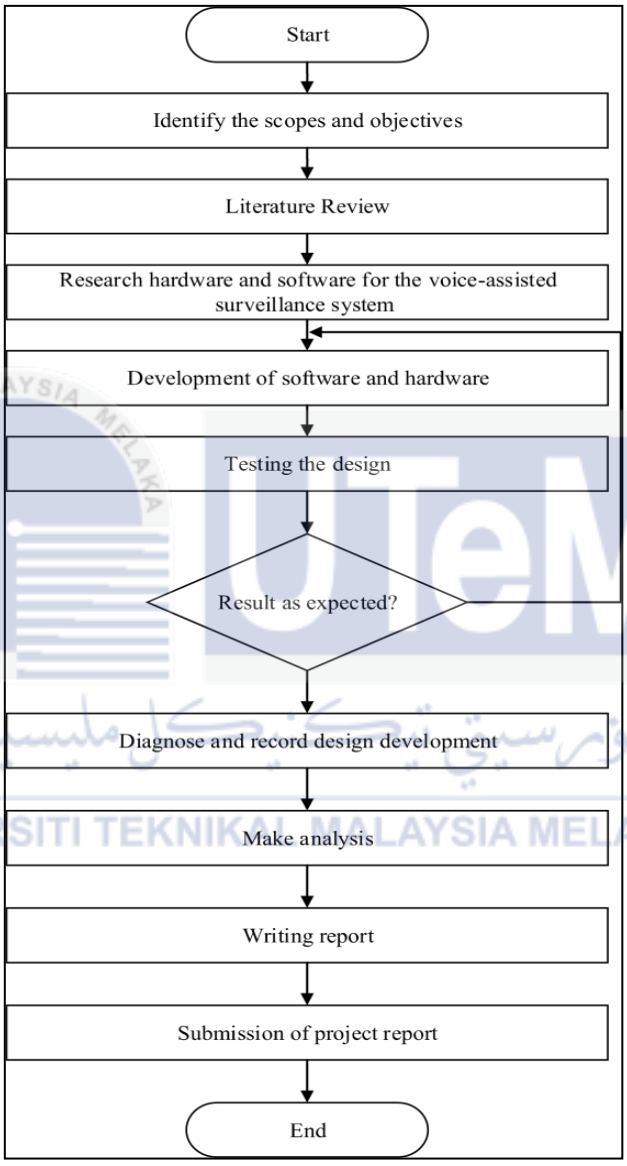


Figure 12 Flowchart of the project.

3.7 System Design

3.7.1 Face recognition flowchart

The user needs to turn on the system to operate the program. The image acquisition from the webcam camera to store in the dataset. After that, the system needs to detect faces using HAAR classification, then it will extract the features based on the training model using either the histogram of gradients (HOG) or support vector machines (SVM). The features match the facial recognition to display the person's name. It keeps repeating until the end of the program.

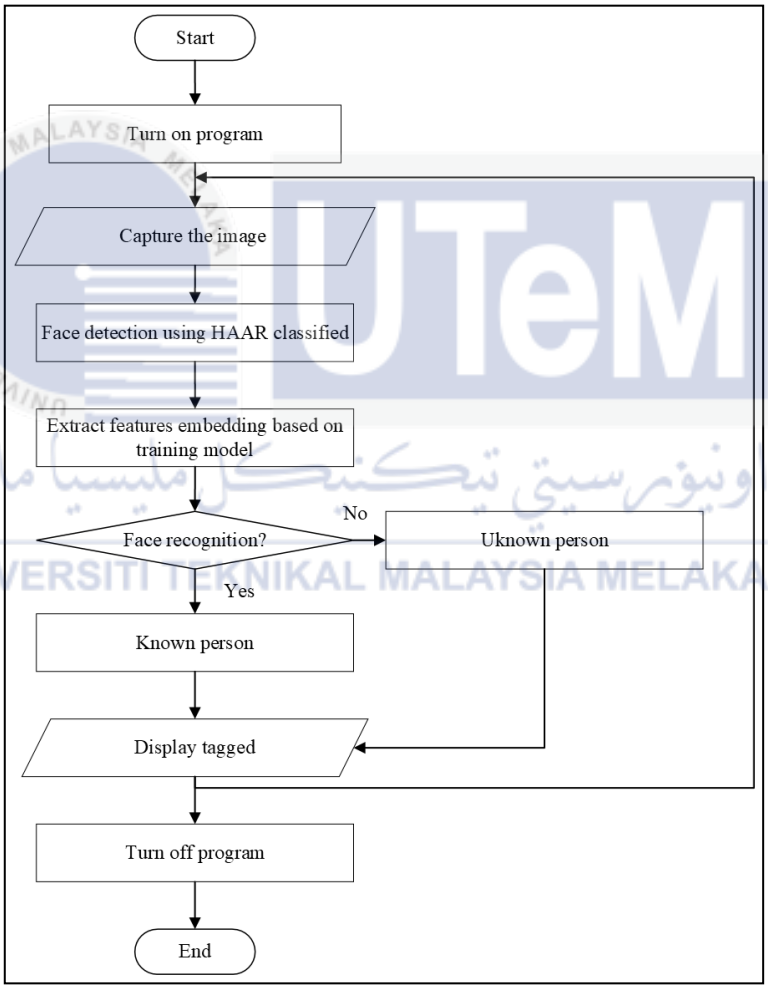


Figure 13 Face recognition flowchart of the project

3.7.2 Remote control flowchart

The user will open the application and click the remote button to use these features. Every button consists of a turn-on and turn-off button to utilize the GPIO to function. The input data is stored at Firebase for the Raspberry Pi to receive the information and execute it on the hardware devices. In this session, Firebase will read the API keys to enable the users to interact with Raspberry Pi. Furthermore, a relay is used to switch each GPIO to operate either on or off. It keeps repeating until the user turns off the application.

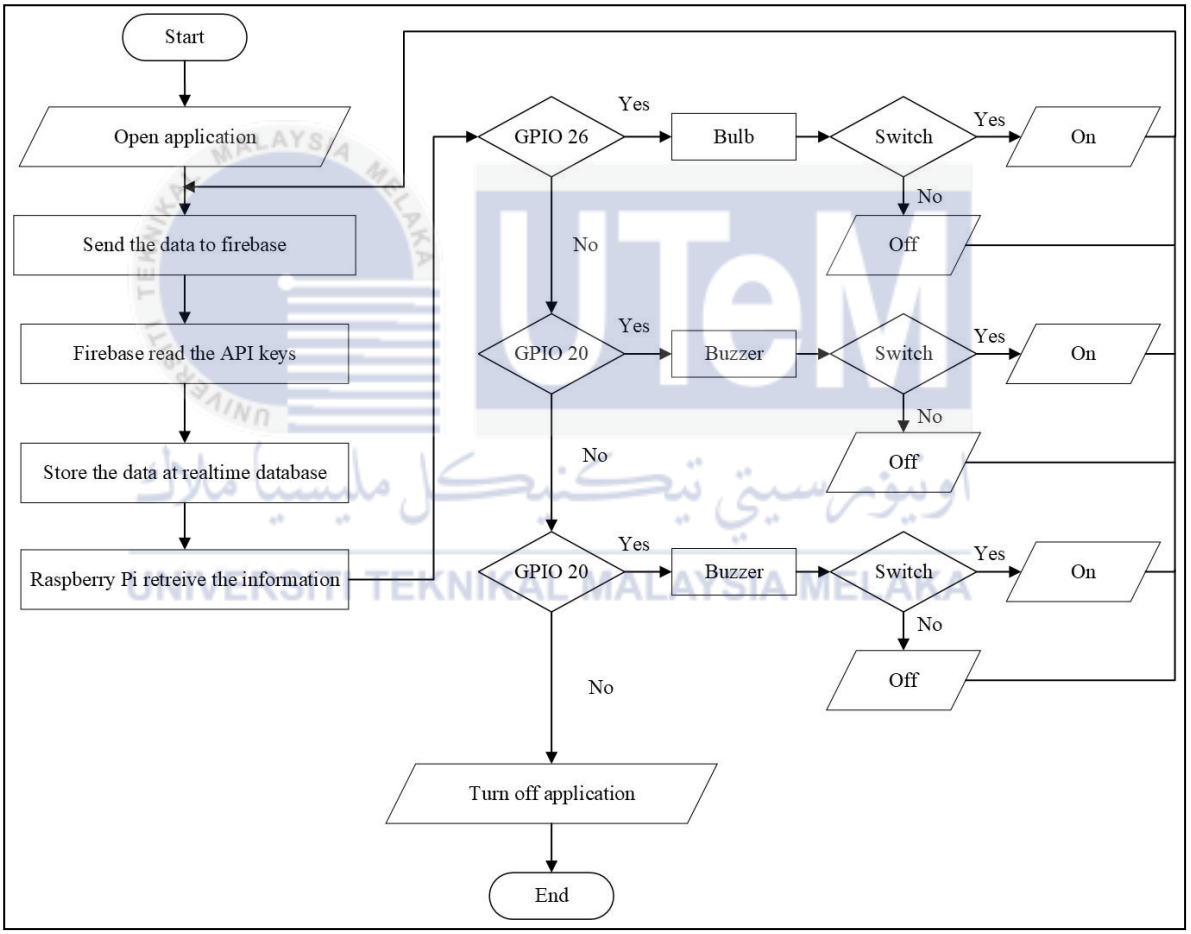


Figure 14 Remote control flowchart of the project

3.7.3 Remote control flowchart

The user needs to open the application to know a person present. The image will retrieve the image data at firebase after the Raspberry Pi captures the image of face recognition. Other than that, the user needs to click the speech button for a voice command of the person's name. This is the way the user know the presence of someone.

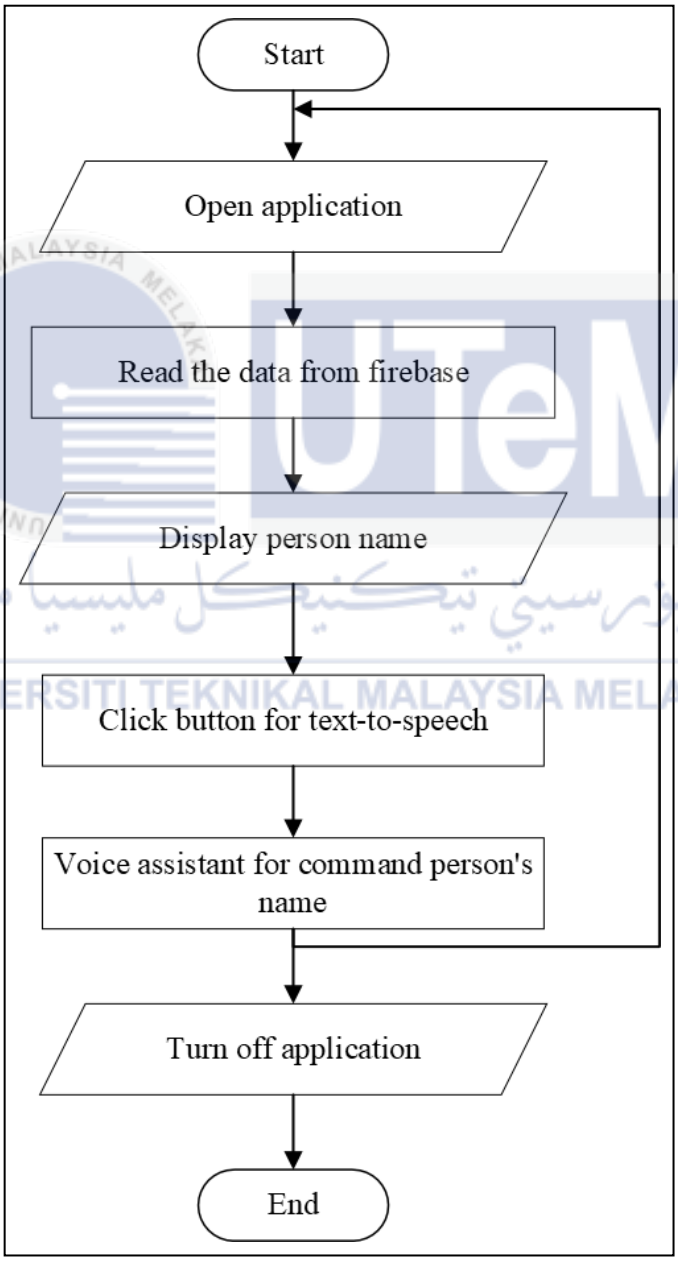


Figure 15 Software flowchart

3.7.4 New data for dataset flowchart

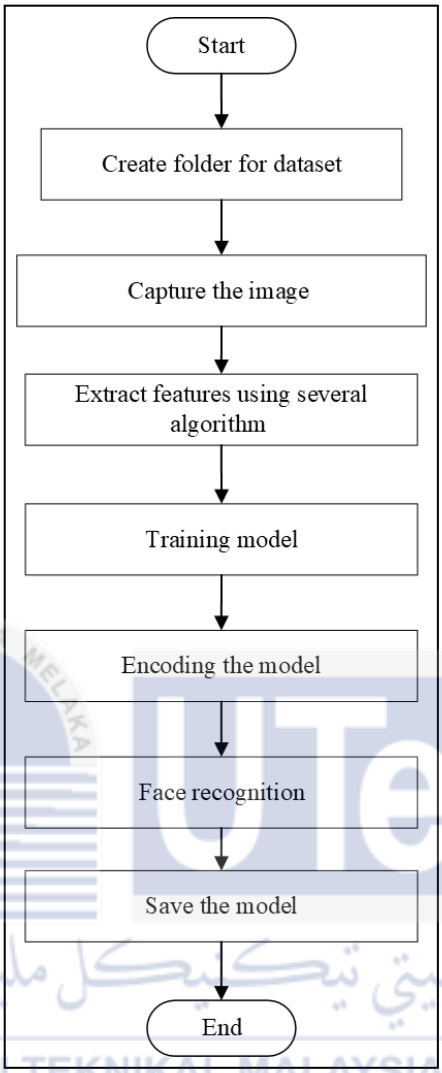


Figure 16 New data for dataset recognition flowchart

3.7.5 Block diagram

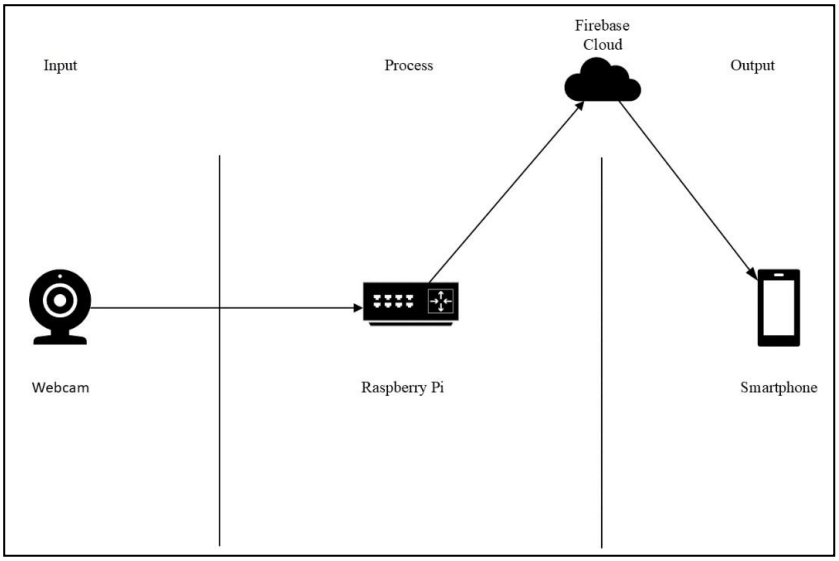


Figure 17 Block diagram of the face recognition

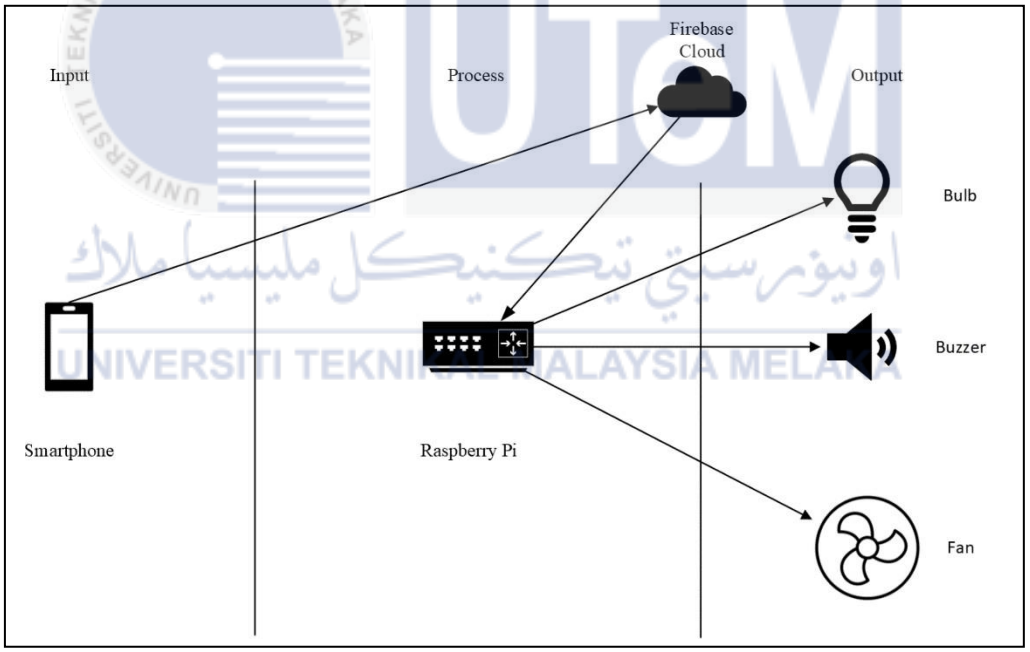


Figure 18 Block diagram of the home automation

3.7.6 Schematic diagram

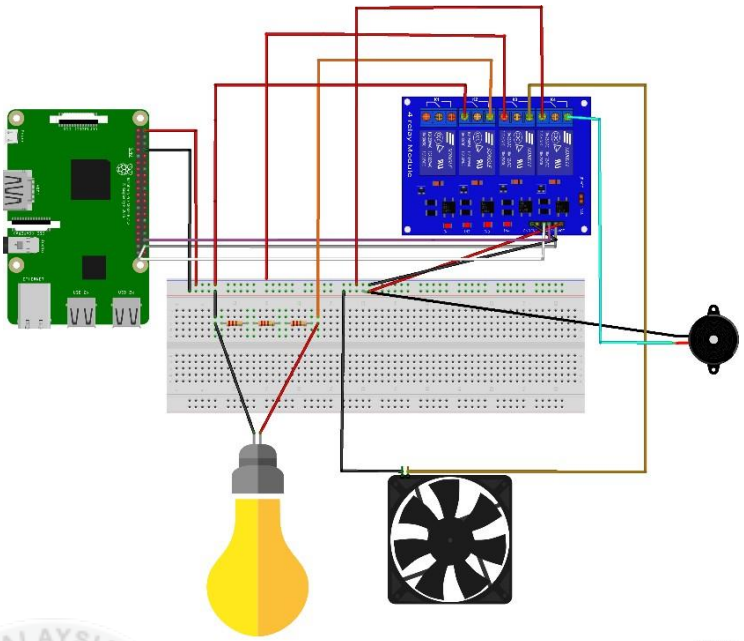


Figure 19 Hardware schematic

3.7.7 Hardware design

Figure 20 Hardware design

3.8 Hardware Specification

3.8.1 Raspberry Pi

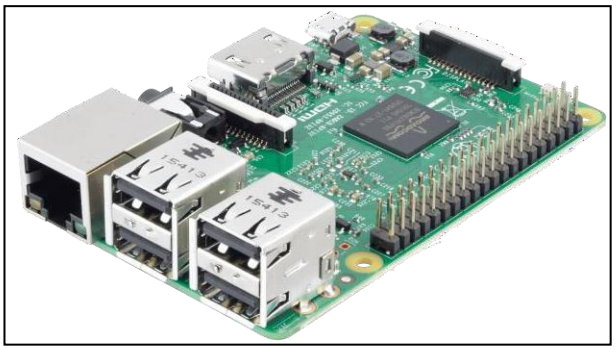


Figure 21 Raspberry Pi [26]

The Raspberry Pi is the device's main component. It functions as a minicomputer. It is involved in all actions, from the beginning when the user provides input to the end when the outcome is provided to the user. This is where all data processing takes happens [19]. It also has wireless LAN and Bluetooth connections, making it the ideal alternative for powerful connected designs. This component will receive the image from the camera and processing to detect any faces that are captured. To recognize people, OpenCV-Python is used to enable the features. Besides that, The Raspberry Pi has a set of General Purpose Input/Output (GPIO) pins that can be controlled using software to perform a variety of tasks. These pins can be configured as inputs or outputs and can be used to read sensors, control LEDs, drive motors, and more. On the other hand, the Raspberry Pi process the image processing for face detection which consists of dataset, detection algorithm, training the model, integration with firebase to store the data, and also for accuracy and adjustment.

3.8.2 Webcam

Figure 22 Webcam [27]

Over the past ten years, the number of webcams connected to the Internet has increased. Webcams are a common example of the Internet of Things (IoT) devices [28]. A webcam is a device that captures video and audio and sends it to a computer or other device for live streaming or recording. Webcams come in different forms, such as USB-connected devices or built-in cameras on laptops, tablets, and smartphones. They are commonly used for video conferencing, live streaming, video calling, and remote monitoring. The webcam is a input for capturing the several images of peoples.

3.8.3 Bulb

Figure 23 Bulb 2.5V

A 2.5V bulb refers to a light bulb that operates at a voltage of 2.5 volts. This type of bulb is likely a low-voltage incandescent bulb, which is typically used in applications such as automotive lighting, flashlights, and other portable devices. These bulbs are designed to be energy-efficient and long-lasting, and they can typically be powered by batteries or other low-voltage power sources. It will be connected to GPIO 26.


```
def LED():

    database = firebase.database()


    LED = database.child("LED")
    result = LED.child("Value").get().val()

    if result == 1:
        GPIO.output(led, GPIO.HIGH)
        print("LED on")
    if result == 0:
        GPIO.output(led, GPIO.LOW)
        print("LED off")
```

Table 6 Code for bulb

3.8.4 Fan

Figure 24 Fan [29]

A 5V fan is a type of cooling fan that operates at a voltage of 5 volts. These types of fans are commonly used in electronic devices such as computers, servers, and other devices that generate heat. It is designed to be energy-efficient and quiet while providing adequate cooling. It will be connected to GPIO 16.

```
def FAN():
    database = firebase.database()
    FAN = database.child("Fan")
    result2 = FAN.child("Value").get().val()

    if result2 == 1:
        GPIO.output(fan, GPIO.HIGH)
        print("FAN on")
    if result2 == 0:
        GPIO.output(fan, GPIO.LOW)
        print("FAN off")
```

Table 7 Code for fan

3.8.5 Buzzer

Figure 25 Buzzer [30]

A buzzer can be connected to a Raspberry Pi, a small single-board computer, to create a variety of projects that involve generating sound. Once the buzzer is connected, you can use programming languages such as Python to control the buzzer and generate different sounds. The RPi.GPIO library can be used to control the GPIO pins on the Raspberry Pi, allowing it to turn the buzzer on and off, and adjust the volume of the sound. It will be connected to GPIO 20.

```
def BUZZER():
    database = firebase.database()
    BUZZER = database.child("Buzzer")
    result1 = BUZZER.child("Value").get().val()

    if result1 == 1:
       GPIO.output(buzzer, GPIO.HIGH)
       print("BUZZER on")
    if result1 == 0:
       GPIO.output(buzzer, GPIO.LOW)
       print("BUZZER off")
```

Table 8 Code for buzzer

3.8.6 Hardware installation

Based on the figure below, GPIO is connected to the relay to enable several devices to execute and display the output. The bulb is connected to GPIO 26, the buzzer is connected to GPIO 20 along with a fan that is connected to GPIO 16. The data that receive from firebase allows the devices to run all those hardware.

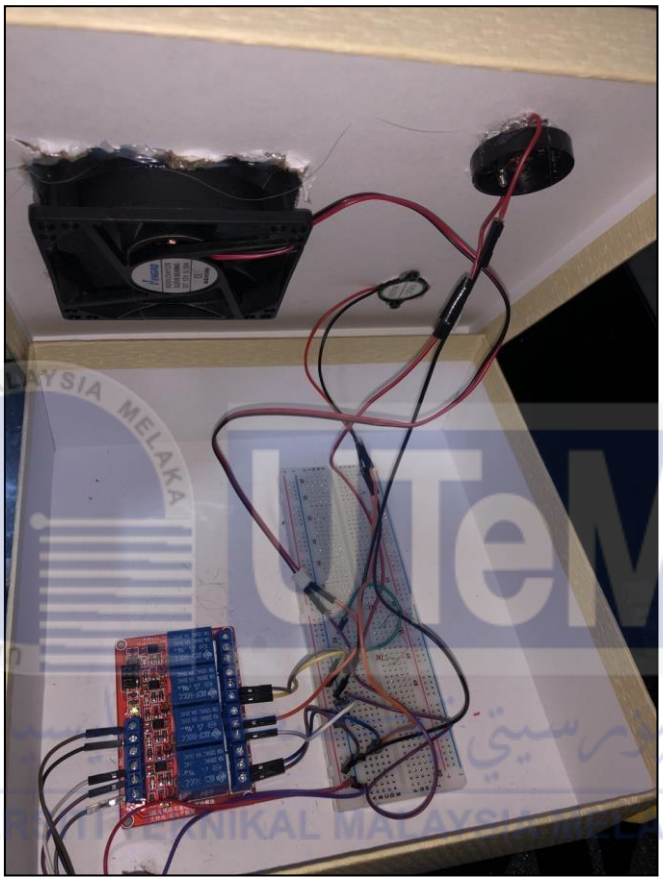


Figure 26 Hardware installation for home automation

3.9 Platform Specification

The purpose of using core software for implementing this study is to function intended. Additionally, the use of this software as a platform to build projects in this study.

3.9.1 Android Studio

Figure 27 Android Studio [31]

Android Studio is to develop a Graphical User Interface for this project. To receive information from Raspberry pi. Android application is used for monitoring tools. In this application where a voice assistant is implemented. To have interaction between devices and a person, a text-to-speech module is introduced. The user will not be using keystrokes but instead using the button to interact with the application. It will much easier to communicate with the visually impaired. Other than that, it makes the interface responsive to different devices, such as mobile phones and tablets, to allow for remote control

3.9.2 Firebase

Figure 28 Firebase [32]

Firebase is a Google-backed application development to store databases either cloud or in real-time. This software can connect Android and Raspberry Pi to get information from both devices. The Firebase will get the real-time video from Raspberry Pi and send or retrieve it to Android Application and several hardware devices.

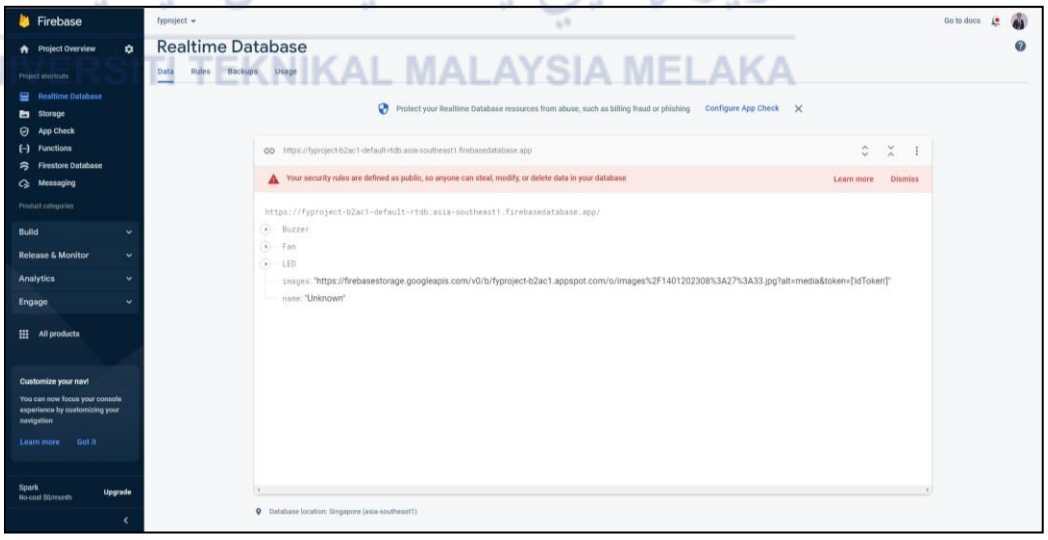


Figure 29 Interface of firebase database

3.9.3 Graphical user interface design

This interface is provided for the visually impaired. Every button has text-to-speech that enables the voice command features. The image will display based on pallets ImageView and the person's name will show in TextView. To enable voice assistance, speech button is implemented.

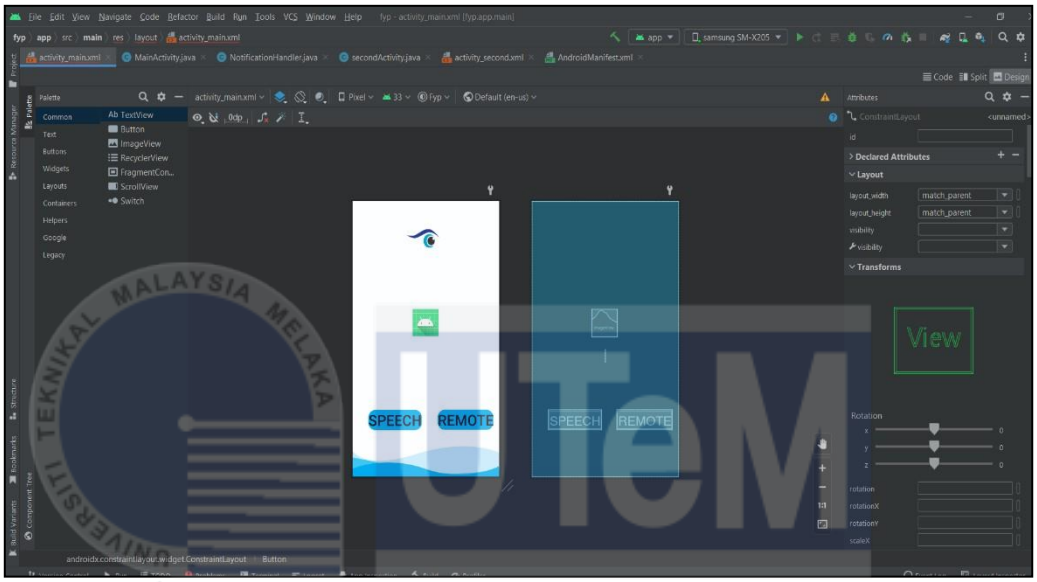


Figure 30 Monitoring graphical user interface

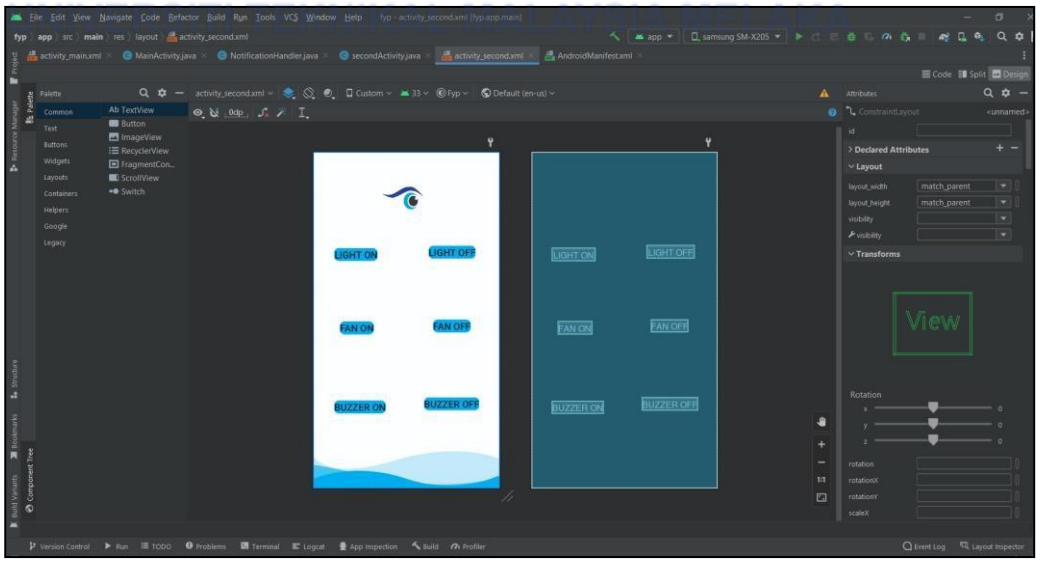


Figure 31 Home automation graphical user interface

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This part shows the result of the project. To get the project functioning as expected and following the requirement, every output from any device will be collected and recorded. Using Android Studio as a user interface, Raspberry pi as image processing and a cloud database as storage would make this system work well.

4.2 Graphical user interface

Creating a GUI for visually impaired users that utilize voice commands can be achieved by using speech recognition technology. This can allow users to interact with the application by clicking the speech button. The application can then interpret the voice commands and perform the corresponding actions. It can be integrated with the Firebase's database to fetch data based on user actions. It is also important to consider providing audio feedback to confirm that a command has been sent and executed correctly. Additionally, the interface is more personalized and intuitive for users.

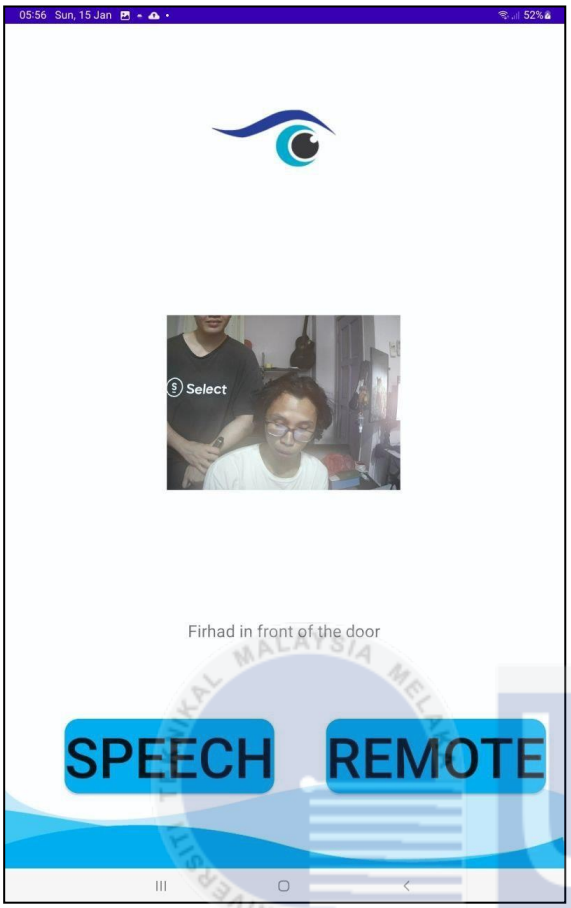


Figure 32 Graphical User Interface

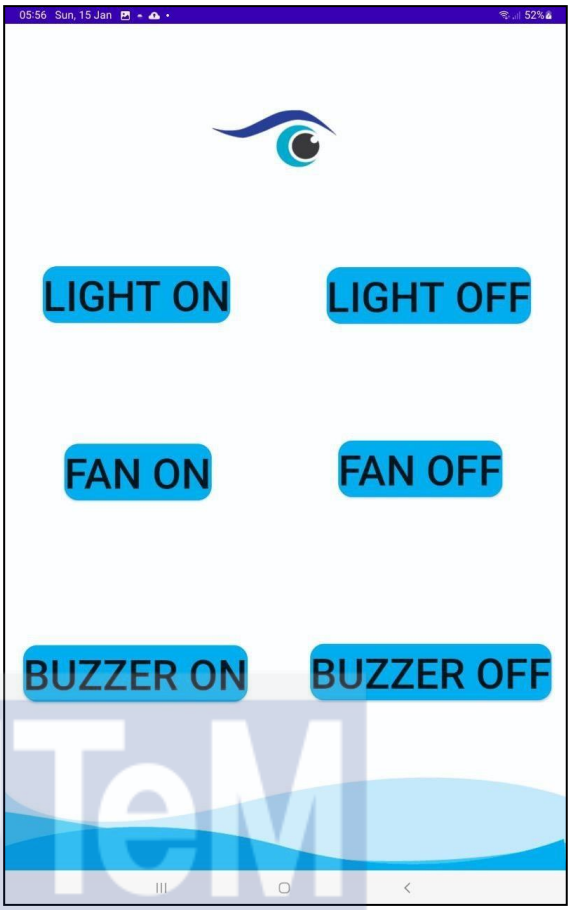


Figure 33 Home automation interface

4.3 Face Recognition

The face recognition system is a measure of how well the system is able to correctly identify or verify the identities of the individuals in a given dataset. There are different ways to calculate the accuracy of a face recognition system, but a common method is to use a sample of labelled data, where the true identities of the individuals are known.

A sample for face recognition can be divided into two main sets, the training set and the test set. The training set is used to train the system, by providing it with a set of labelled data, where the true identities of the individuals are known. The test set, on the other hand, is used to evaluate the system, by providing it with a different set of labelled data, where the true identities of the individuals are also known, but the system has not seen before. The several dataset have been collected for determined the accuracy while time taken for face recognition to recognize is recorded for performance.

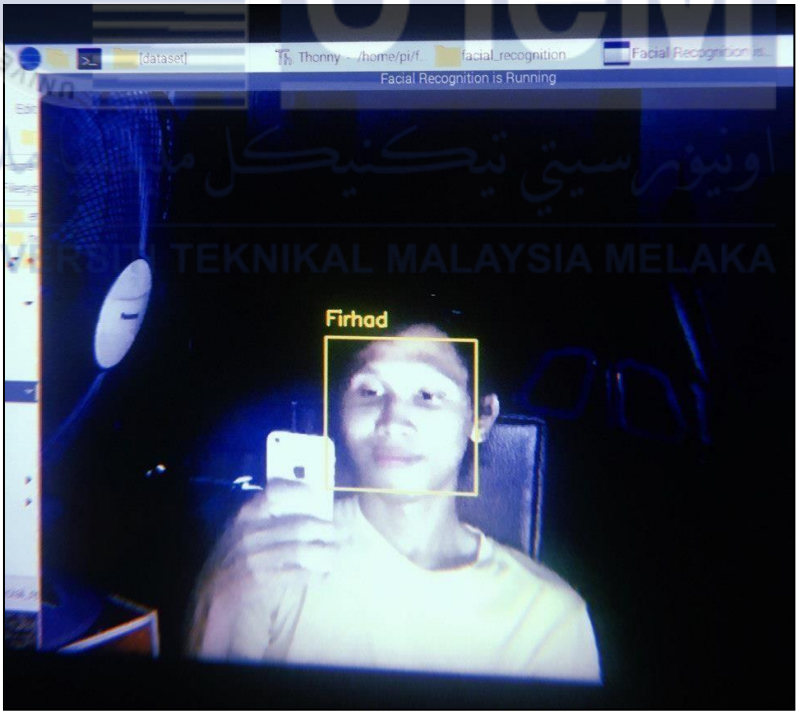


Figure 34 Face recognition known person

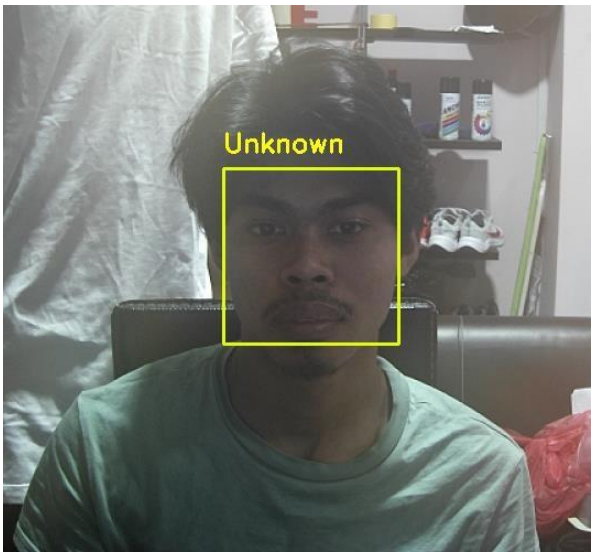


Figure 35 Face recognition for unknown person

```
boxes = face_recognition.face_locations(rgb,
    model="svm"+"hog")
```

Table 9 Training model

This line is where image processing is executed and which model is used. There are several types of model which is histogram of gradient (HOG), support virtual machines (SVM), and convolution neuron network (CNN). Raspberry Pi 3B+ is only available for HOG and SVM regarding the process and performance of the CPU is limited.

4.4 Accuracy parameters

Algorithm / Condition	HOG	SVM	HOG + SVM
Light	78%	85%	96%
Dark	71%	78%	85%

Table 10 Accuracy of face recognition

Based on the table above, there are 14 people are test for the sample for recognition. For HOG there are 3 out of 14 people are recognise as the known person, while SVM have 2 people and HOG + SVM only one person. On the other hand, face recognition technology is affected by lighting conditions because the algorithms used to detect and identify faces rely on certain features, such as the eyes, nose, and mouth, to be visible and clearly defined. In low light conditions, these features may be difficult to discern, which can make it harder for the technology to accurately identify a person. Additionally, shadows and glare from lights can also create issues for face recognition systems. On the other hand, in well-lit environments, the technology is likely to perform better as the facial features are more clearly visible. HOG is a feature descriptor used in computer vision for object detection tasks such as face detection. It is particularly effective at capturing the shape and structure of an object. SVM is a supervised machine learning algorithm that can be used for classification and regression tasks. It is particularly effective when the data is linearly separable. When HOG is combined with SVM, it is typically used as a feature descriptor in the SVM algorithm. This combination is often used in object detection tasks because HOG is effective at capturing the shape and structure of an object and SVM is effective at classifying the objects. The accuracy of HOG + SVM combination depends on several factors such as the dataset, the quality of the images, the parameters used for the HOG descriptor, and the parameters used for the SVM classifier. However, in general, HOG + SVM is considered to be

an effective combination. Accuracy can be measured using this formula based on below.

$$= \frac{\text{Tp}}{\text{Tn}} \times 100 \quad (2)$$

Figure 36 Light condition

Figure 37 Dark condition

```
matches = face_recognition.compare_faces(data["encodings"],
    encoding)
name = "Unknown"
```

Table 11 Code for image data compare with current image

4.5 Performance

The performance of a face recognition system on a Raspberry Pi will depend on several factors, including the model of Raspberry Pi being used, the quality of the camera, the resolution of the images, and the complexity of the face recognition algorithm. Raspberry Pi has a limited processing power and memory compared to a full-fledged computer. Therefore, using a more powerful Raspberry Pi model such as Raspberry Pi 4 with a better CPU and memory will lead to better performance. The quality of the camera used also plays a significant role in the performance of the system. A high-resolution camera will provide better quality images, which will make it easier for the algorithm to detect and recognize faces. The complexity of the algorithm used also affects the performance of the system. Simple algorithms such as HAAR cascade tend to be faster, but may not be as

accurate as more complex algorithms like deep learning-based methods. In general, the performance of a face recognition system on a Raspberry Pi will be lower than on a full-fledged computer with more powerful hardware. However, with proper optimization and the use of efficient algorithms, a face recognition system can still be implemented on a Raspberry Pi with acceptable performance for certain use cases.

Size frame	Time taken for detection (s)	Elapse time for display the program (s)
200	0.92	3.44
500	1.21	11.59
800	2.47	12.20
1000	3.34	14.28
1200	4.01	16.28

Table 12 Performance based on size frame

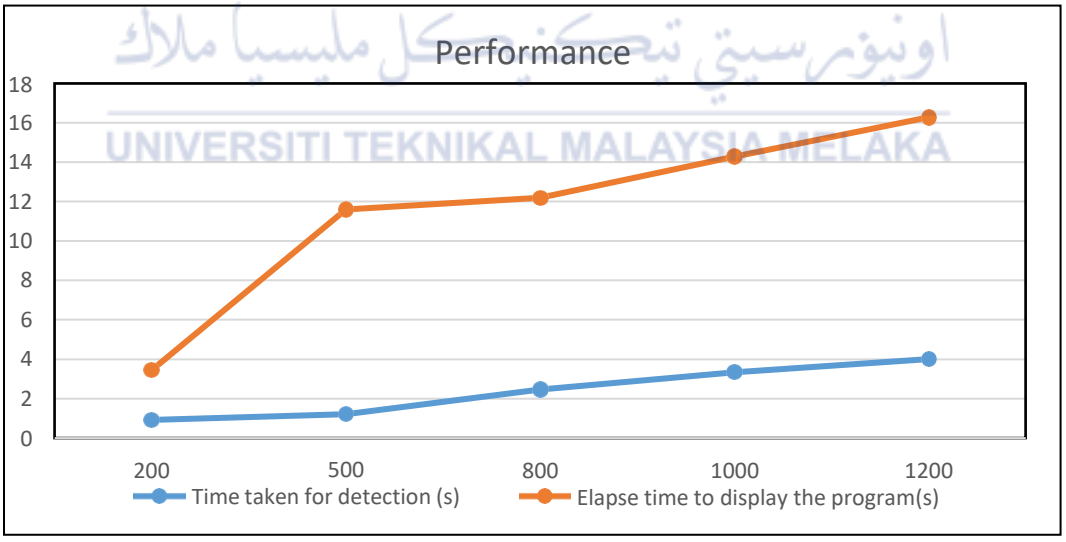


Figure 38 Graph of the performance

4.6 Hardware output

When the button is clicked the Raspberry Pi will receive the data and send it to GPIO to give the instruction on which GPIO has selected either bulb, fan, or buzzer. Moreover, the button produces a voice command to make the user understand which button is clicked.

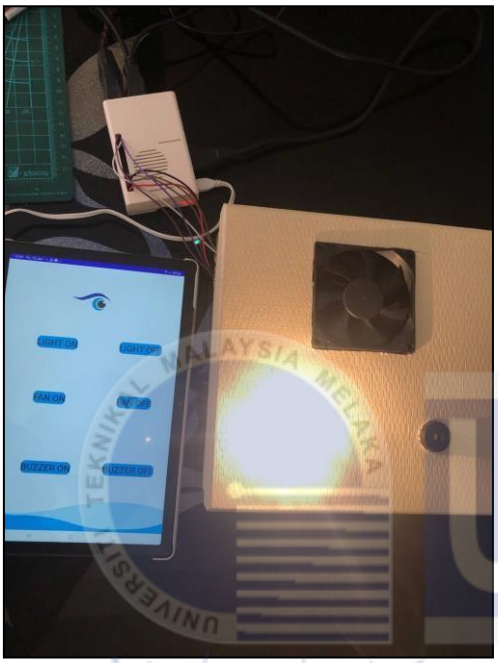


Figure 39 Bulb On

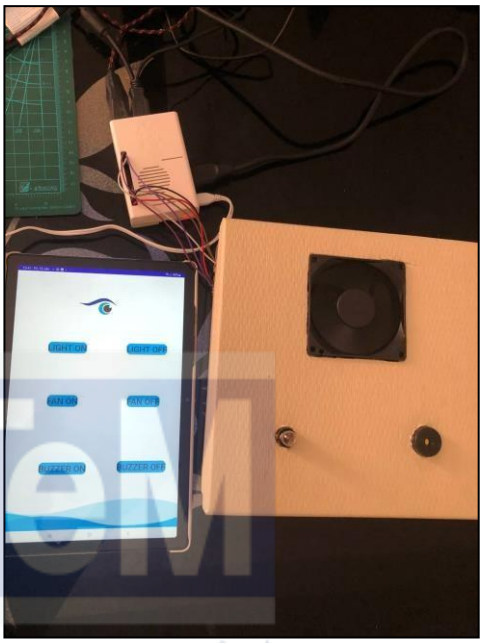


Figure 40 Fan on

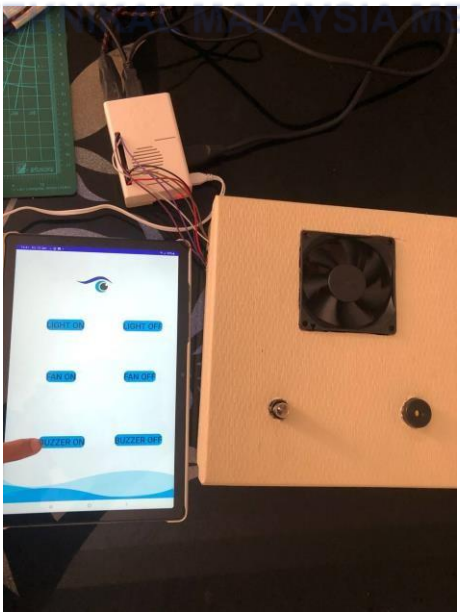


Figure 41 Buzzer on

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The system successfully developed a mobile application based on a surveillance monitoring system using a single-board computer which consists of Raspberry Pi along with OpenCV-Python for face recognition and a smartphone as an output device. Furthermore, Raspberry Pi can provide the necessary hardware and software to create a fully-functioning home automation system. With its small form factor, low power consumption, and versatility. It can be used as a remote control for hardware devices. Besides that, the application delivers a voice assistant for the user to understand the usage of the system.

Based on chapter 3 the planning and implementation have been made to determine the result for every objective. Chapter 4 requires recorded data that has been analyzed and tested. The new registration of the known person was created for two datasets to check the accuracy. On the other hand, the condition of the lighting might be affected as well. Several samples determined the accuracy by using a specific algorithm. Other than that, the performance of face recognition is also based on the speed of processing memory. Raspberry Pi 3B+ has limited capabilities rather to 4B+ which has better processing memory and capabilities. The Android application was developed for the visually impaired to monitor and control particular hardware. Additionally, voice-assistant for the visually impaired is applied in Android that allows users to listen to the action or acknowledge someone's presence by using a smartphone.

In conclusion, last but not least, to achieve the objective, the result, and analysis have been taken to prove the validation and verification of monitoring tools and home automation. The project achieved the objective to ensure the system is safe and make life easier for visually impaired people.

5.2 Future work

Future work for a home automation system using a Raspberry Pi could include incorporating sound capabilities, such as voice commands or playing music. This could be achieved by adding a microphone and speaker to the Raspberry Pi, and using machine learning algorithms to process the audio input. Another potential improvement would be to automatically open the program controlling the home automation system when the Raspberry Pi is powered on, providing a more seamless user experience. Additionally, a physical button could be added to the system to initiate the training of machine learning models, allowing for more efficient updates to the system's capabilities. These enhancements would add more convenience and usability to the home automation system, making it even more efficient and user-friendly.

References

[1] M. Khorrami-Nejad, A. Sarabandi, and F. Askarizadeh, "The Impact of Visual Impairment on Quality of Life," 2016. [Online]. Available: https://www.researchgate.net/publication/315689148

[2] Institute of Electrical and Electronics Engineers, *2017 2nd International Conference on Image, Vision and Computing (ICIVC 2017) : August June 24, 2017, Chengdu, China.*

[3] "See The World Through The Eyes Of A Person With Visual Impairment | HuffPost Life." https://www.huffpost.com/archive/au/entry/see-the-worldthrough-the-eyes-of-a-person-with-visual-impairmen_a_23008911 (accessed Jun. 01, 2022).

[4] D. Kim, S. Park, J. Kim, J. Y. Bang, and S. Jung, "Stabilized adaptive sampling control for reliable real-time learning-based surveillance systems," *Journal of Communications and Networks*, vol. 23, no. 2, pp. 129–137, May 2021, doi: 10.23919/jcn.2021.000009.

[5] N. Hazim Barnouti, S. Sameer Mahmood Al-Dabbagh, and W. Esam Matti, "ISSN : 2249-0868 Foundation of Computer Science FCS," 2016. [Online]. Available: www.ijais.org

[6] "7 Solutions for Powering a Video Surveillance System." https://blog.tripplite.com/7-solutions-for-powering-protecting-andcontrolling-your-video-surveillance-system (accessed Jun. 03, 2022).

[7] S. Han and H. Yang, "Understanding adoption of intelligent personal assistants: A parasocial relationship perspective," *Industrial Management and Data Systems*, vol. 118, no. 3, pp. 618–636, 2018, doi: 10.1108/IMDS05-2017-0214.

[8] V. Singh, V. Shokeen, and B. Singh, "FACE DETECTION BY HAAR CASCADE CLASSIFIER WITH SIMPLE AND COMPLEX BACKGROUNDS IMAGES USING OPENCV IMPLEMENTATION."

[Online]. Available: http://www.ijates.com

[9] "Face Detection Using OpenCV With Haar Cascade Classifiers | by Vincent Tabora | Becoming Human: Artificial Intelligence Magazine." https://becominghuman.ai/face-detection-using-opencv-with-haar-cascadeclassifiers-941dbb25177 (accessed Jun. 04, 2022).

[10] "Using Histogram of Oriented Gradients (HOG) for Object Detection." https://iq.opengenus.org/object-detection-with-histogram-of-orientedgradients-hog/ (accessed Jun. 06, 2022).

[11] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection." [Online]. Available: http://lear.inrialpes.fr

[12] "Understanding Face Recognition using LBPH algorithm - Analytics Vidhya." https://www.analyticsvidhya.com/blog/2021/07/understandingface-recognition-using-lbph-algorithm/ (accessed Jun. 04, 2022).

[13] C. W. (Chloe) Ki, E. Cho, and J. E. Lee, "Can an intelligent personal assistant (IPA) be your friend? Para-friendship development mechanism between IPAs and their users," *Computers in Human Behavior*, vol. 111, p. 106412, Oct. 2020, doi: 10.1016/J.CHB.2020.106412.

[14] F. Ashiq *et al.*, "CNN-Based Object Recognition and Tracking System to Assist Visually Impaired People," *IEEE Access*, vol. 10, pp. 14819–14834, 2022, doi: 10.1109/ACCESS.2022.3148036.

[15] A. D. M. Africa, A. J. A. Abello, Z. G. Gacuya, I. K. A. Naco, and V. A. R. Valdes, "Face recognition using MATLAB," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 4, pp. 1110–1116, Jul. 2019, doi: 10.30534/ijatcse/2019/17842019.

[16] M. Kumar, K. S. Raju, D. Kumar, N. Goyal, S. Verma, and A. Singh, "An efficient framework using visual recognition for IoT based smart city surveillance," *Multimedia Tools and Applications*, vol. 80, no. 20, pp.

31277–31295, Aug. 2021, doi: 10.1007/s11042-020-10471-x.

[17] C. Li, W. Wei, J. Li, and W. Song, "A cloud-based monitoring system via face recognition using Gabor and CS-LBP features," *Journal of Supercomputing*, vol. 73, no. 4, pp. 1532–1546, Apr. 2017, doi: 10.1007/s11227-016-1840-6.

[18] A. D. Post Graduate Scholar and S. P. Associate Professor, "Face Recognition Authenticated Voice Assistant System for the disabled; Face Recognition Authenticated Voice Assistant System for the disabled," 2021, doi: 10.1109/ICESC51422.2021.9533017.

[19] P. Singh, P. Nayak, A. Datta, D. Sani, G. Raghav, and R. Tejpal, "Voice Control Device using Raspberry Pi," in *Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019*, Apr. 2019, pp. 723–728. doi: 10.1109/AICAI.2019.8701409.

[20] J. Jommuangbut and K. Sritrakulchai, "Development of the Human Following Robot Control System Using HD Webcam."

[21] X. Xing, G. Xu, B. Cai, C. Qing, and X. Xu, "Face verification based on feature transfer via PCA-SVM framework," in *Proceedings - 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017*, Jan. 2018, vol. 2018-January, pp. 1086–1091. doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.166.

[22] "Raspberry Pi 3 Single-board computer Linux, pi, electronics, computer, hDMI png | PNGWing." https://www.pngwing.com/en/free-png-zfjrv (accessed Jan. 13, 2023).

[23] "Luminous C30 2K QHD Webcam | Fantech." https://fantechworld.com/luminous-c30/ (accessed Jan. 13, 2023).

[24] X. Feng, Q. Li, H. Wang, and L. Sun, *Open access to the Proceedings of the 27th USENIX Security Symposium is sponsored by USENIX. Acquisitional Rule-based*

Engine for Discovering Internet-of-Thing Devices Acquisitional Rule-based Engine for Discovering Internet-of-Thing Devices. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/feng

[25] X. Xing, G. Xu, B. Cai, C. Qing, and X. Xu, "Face verification based on feature transfer via PCA-SVM framework," in *Proceedings - 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017*, Jan. 2018, vol. 2018-January, pp. 1086–1091. doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.166.

[26] "Raspberry Pi 3 Single-board computer Linux, pi, electronics, computer, hDMI png | PNGWing." https://www.pngwing.com/en/free-png-zfjrv (accessed Jan. 13, 2023).

[27] "Luminous C30 2K QHD Webcam | Fantech." https://fantechworld.com/luminous-c30/ (accessed Jan. 13, 2023).

[28] X. Feng, Q. Li, H. Wang, and L. Sun, *Open access to the Proceedings of the 27th USENIX Security Symposium is sponsored by USENIX. Acquisitional Rule-based Engine for Discovering Internet-of-Thing Devices Acquisitional Rule-based Engine for Discovering Internet-of-Thing Devices*. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/feng

[29] "MY_SEO_BAU_V1." https://www.lazada.com.my/products/dc5015-cooling-fan-fan-two-wire-oil-bearing-5cm-purifier-car-light-dc-fan-5v-12v-24v-015a-i2891741193.html (accessed Jan. 13, 2023).

[30] "CPI Piezo Buzzer Indicators - CUI Devices | Mouser." https://my.mouser.com/new/cui-devices/cui-inc-cpi-buzzer-indicators/ (accessed Jan. 13, 2023).

[31] "Android Studio Google Developers Android Oreo, vektor, logo, mobile Phones, google Daydream png | PNGWing." https://www.pngwing.com/en/free-png-nrabo (accessed Jan. 13, 2023).

[32] "Free download | Firebase Cloud Messaging Google Cloud Messaging, Api As A Service, angle, rectangle png | PNGEgg." https://www.pngegg.com/en/png-tdcmf/download (accessed Jan. 13, 2023).

APPENDICES

Appendix A

Raspberry Pi

Snapshot.py

```
import cv2

name = '' #replace with your name

cam = cv2.VideoCapture(0)

cv2.namedWindow("press space to take a photo", cv2.WINDOW_NORMAL)
cv2.resizeWindow("press space to take a photo", 500, 300)

img_counter = 0

while True:
    ret, frame = cam.read()
    if not ret:
        print("failed to grab frame")
        break
    cv2.imshow("press space to take a photo", frame)

    k = cv2.waitKey(1)
    if k%256 == 27:
        # ESC pressed
        print("Escape hit, closing...")
        break
    elif k%256 == 32:
        # SPACE pressed
        img_name = "dataset/"+ name +"/image_{}.jpg".format(img_counter)
        cv2.imwrite(img_name, frame)
        print("{} written!".format(img_name))
        img_counter += 1

cam.release()
```

Train_model.py

```
#! /usr/bin/python

# import the necessary packages
from imutils import paths
import face_recognition
#import argparse
import pickle
import cv2
import os

# our images are located in the dataset folder
print("[INFO] start processing faces...")
imagePaths = list(paths.list_images("dataset"))

# initialize the list of known encodings and known names
knownEncodings = []
knownNames = []

# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
    # extract the person name from the image path
    print("[INFO] processing image {}/{}".format(i + 1,
        len(imagePaths)))
    name = imagePath.split(os.path.sep)[-2]

    # load the input image and convert it from RGB (OpenCV ordering)
    # to dlib ordering (RGB)
    image = cv2.imread(imagePath)
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # detect the (x, y)-coordinates of the bounding boxes
    # corresponding to each face in the input image
    boxes = face_recognition.face_locations(rgb,
        model="svm"+"hog")

    # compute the facial embedding for the face
    encodings = face_recognition.face_encodings(rgb, boxes)

    # loop over the encodings
    for encoding in encodings:
        # add each encoding + name to our set of known names and
        # encodings
        knownEncodings.append(encoding)
        knownNames.append(name)

# dump the facial encodings + names to disk
print("[INFO] serializing encodings...")
data = {"encodings": knownEncodings, "names": knownNames}
f = open("encodings.pickle", "wb")
f.write(pickle.dumps(data))
f.close()
```

fyp.py

```
#! /usr/bin/python

# import the necessary packages
from imutils.video import VideoStream
from imutils.video import FPS
import face_recognition
import imutils
import pickle
import time
from time import sleep
import cv2
import requests
import pyrebase
from datetime import datetime
import os
import RPi.GPIO as GPIO



#Initialize 'currentname' to trigger only when a new person is identified.
currentname = "unknown"
#Determine faces from encodings.pickle file model created from train_model.py
encodingsP = "encodings.pickle"

#use this xml file
cascade = "haarcascade_frontalface_default.xml"


config ={                                                                    #define a dictionary na
  "apiKey": "AIzaSyD-OSJOIT7HKJudGFyL4R-djZGhHmURXb0",
  "authDomain": "fyproject-b2ac1.appspot.com",
  "databaseURL": "https://fyproject-b2ac1-default-rtdb.asia-southeast1.firebasedatabase.app/",
  "storageBucket": "fyproject-b2ac1.appspot.com"
}

firebase = pyrebase.initialize_app(config)
database = firebase.database()
storage = firebase.storage()

#function for setting up emails

# load the known faces and embeddings along with OpenCV's Haar
# cascade for face detection
print("[INFO] loading encodings + face detector...")
data = pickle.loads(open(encodingsP, "rb").read())
detector = cv2.CascadeClassifier(cascade)

# initialize the video stream and allow the camera sensor to warm up
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
# vs = VideoStream(usePiCamera=True).start()
time.sleep(2.0)

# start the FPS counter
fps = FPS().start()


# loop over frames from the video file stream
while True:
    # grab the frame from the threaded video stream and resize it
    # to 500px (to speedup processing)
```

```
frame = imutils.resize(frame, width=800)
t1 = cv2.getTickCount()

# convert the input frame from (1) BGR to grayscale (for face
# detection) and (2) from BGR to RGB (for face recognition)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

# detect faces in the grayscale frame
rects = detector.detectMultiScale(gray, scaleFactor=1.1,
    minNeighbors=5, minSize=(30, 30),
    flags=cv2.CASCADE_SCALE_IMAGE)

# OpenCV returns bounding box coordinates in (x, y, w, h) order
# but we need them in (top, right, bottom, left) order, so we
# need to do a bit of reordering
boxes = [(y, x + w, y + h, x) for (x, y, w, h) in rects]

# compute the facial embeddings for each face bounding box
encodings = face_recognition.face_encodings(rgb, boxes)
names = []

# loop over the facial embeddings
for encoding in encodings:
    # attempt to match each face in the input image to our known
    # encodings
    matches = face_recognition.compare_faces(data["encodings"],
        encoding)
    name = "Unknown"

    # check to see if we have found a match
    if True in matches:
        # find the indexes of all matched faces then initialize a
        # dictionary to count the total number of times each face
        # was matched
        matchedIdxs = [i for (i, b) in enumerate(matches) if b]
        counts = {}

        # loop over the matched indexes and maintain a count for
        # each recognized face face
        for i in matchedIdxs:
            name = data["names"][i]
            counts[name] = counts.get(name, 0) + 1

        # determine the recognized face with the largest number
        # of votes (note: in the event of an unlikely tie Python
        # will select first entry in the dictionary)
        name = max(counts, key=counts.get)

        #If someone in your dataset is identified, print their name on the screen
        if currentname != name:
            currentname = name
            print(currentname)
            #Take a picture to send in the firebase
            os.chdir ("/home/yad/face_recognition/facial_recognition/capture")
            now = datetime.now()
            dt = now.strftime("%d%m%Y%H:%M:%S")
            img_name = dt + ".jpg"
            cv2.imwrite(img_name, frame)
            print('Taking a picture.')
```

```
            # update the list of names
            names.append(name)

            #store at database
            storage.child("/images", img_name).put(img_name)
            url = storage.child("/images",img_name).get_url(['idToken'])
            database.child("images").set(url)
            database.child("name").set(currentname)

        # loop over the recognized faces
        for ((top, right, bottom, left), name) in zip(boxes, names):
            # draw the predicted face name on the image - color is in BGR
            cv2.rectangle(frame, (left, top), (right, bottom),
                (0, 255, 225), 2)
            y = top - 15 if top - 15 > 15 else top + 15
            cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
                .8, (0, 255, 255), 2)

        # display the image to our screen
        cv2.imshow("Facial Recognition is Running", frame)
        key = cv2.waitKey(1) & 0xFF


        # if the `q` key was pressed, break from the loop
        if key == ord("q"):
            break

        # update the FPS counter
        fps.update()


# stop the timer and display FPS information
fps.stop()
print("[INFO] elasped time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

officAutomation.py

```
import pyrebase
from firebase import firebase
import RPi.GPIO as GPIO
from time import sleep

led = 26
buzzer= 20
fan = 16
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(led, GPIO.OUT)
GPIO.setup(buzzer, GPIO.OUT)
GPIO.setup(fan, GPIO.OUT)


config = {
  "apiKey": "AIzaSyD-OSJOIT7HKJudGFyL4R-djZGhHmURXb0",
  "authDomain": "fyproject-b2ac1.appspot.com",
  "databaseURL": "https://fyproject-b2ac1-default-rtdb.asia-southeast1.firebasedatabase.app/",
  "storageBucket": "fyproject-b2ac1.appspot.com"
}

firebase = pyrebase.initialize_app(config)


def LED():

        database = firebase.database()


        LED = database.child("LED")
        result = LED.child("Value").get().val()

        if result == 1:
            GPIO.output(led, GPIO.HIGH)
            print("LED on")
        if result == 0:
            GPIO.output(led, GPIO.LOW)
            print("LED off")

def BUZZER():

         database = firebase.database()
         BUZZER = database.child("Buzzer")
         result1 = BUZZER.child("Value").get().val()

         if result1 == 1:
            GPIO.output(buzzer, GPIO.HIGH)
            print("BUZZER on")
         if result1 == 0:
            GPIO.output(buzzer, GPIO.LOW)
            print("BUZZER off")

def FAN():

         database = firebase.database()
         FAN = database.child("Fan")
         result2 = FAN.child("Value").get().val()

         if result2 == 1:
            GPIO.output(fan, GPIO.HIGH)
            print("FAN on")
         if result2 == 0:
            GPIO.output(fan, GPIO.LOW)
            print("FAN off")

while True:
```

```
while True:

    LED()

    BUZZER()

    FAN()

    sleep(1)
```

Android Studio

MainActivity.java

```
package com.example.fyp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

import android.app.AlarmManager;
import android.app.Notification;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.SystemClock;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;

import java.util.Locale;


public class MainActivity extends AppCompatActivity {

    private TextView personName;
    private ImageView image;
    TextToSpeech textToSpeech;

    public static String NOTIFICATION_CHANNEL_ID = "1001";
    public static String default_notification_id = "default";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        personName = (TextView) findViewById(R.id.personName);
        image = (ImageView) findViewById(R.id.image);
        Button btnConvert = findViewById(R.id.tts);
        Button button = (Button) findViewById(R.id.button);


        textToSpeech = new TextToSpeech(getApplicationContext(), status -> {
            if (status == TextToSpeech.SUCCESS) {

                textToSpeech.setLanguage(new Locale("ms", "MY"));
            }

        });

        DatabaseReference mDatabase = FirebaseDatabase.getInstance().getReference();
```

```
        mDatabase.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

                scheduleNotification(getNotification("Someone in front of the door"), 0);

                String img = dataSnapshot.child("images").getValue().toString();
                Picasso.get().load(img).into(image);

                String name = dataSnapshot.child("name").getValue().toString() + " " + "in front of the door";
                personName.setText(name);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
            }
        });

        btnConvert.setOnClickListener(v -> {
            String s = personName.getText().toString();

            int speech = textToSpeech.speak(s, TextToSpeech.QUEUE_FLUSH, null);

        });

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, secondActivity.class);
                startActivity(intent);

                String s = "Remote";

                int speech = textToSpeech.speak(s, TextToSpeech.QUEUE_FLUSH, null);

            }
        });
    }

    private void scheduleNotification (Notification notification,int delay)
    {
        Intent notificationIntent = new Intent(this, NotificationHandler.class);
        notificationIntent.putExtra(NotificationHandler.NOTIFICATIONID, 1);
        notificationIntent.putExtra(NotificationHandler.NOTIFICATION, notification);
        PendingIntent pendingIntent = PendingIntent.getBroadcast(this, 0, notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE);
        long futureMillis = SystemClock.elapsedRealtime() + delay;
        AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
        assert alarmManager != null;
        alarmManager.set(AlarmManager.ELAPSED_REALTIME_WAKEUP, futureMillis, pendingIntent);
        System.out.println("function is called");
    }

    private Notification getNotification (String content){
        NotificationCompat.Builder builder = new NotificationCompat.Builder(this, default_notification_id);

        builder.setContentTitle("Alert");
        builder.setContentText(content);
        builder.setSmallIcon(R.drawable.ic_launcher_foreground);
        builder.setAutoCancel(true);
        builder.setChannelId(NOTIFICATION_CHANNEL_ID);

        return builder.build();
    }
```


NotificationHandler.java

```
package com.example.fyp;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Build;

public class NotificationHandler extends BroadcastReceiver {

    public static String NOTIFICATIONID = "notification-id";
    public static String NOTIFICATION = "notification";


    @Override
    public void onReceive(Context context, Intent intent) {
        NotificationManager notificationManager = (NotificationManager) context.getSystemService(context.NOTIFICATION_SERVICE);
        Notification notification = intent.getParcelableExtra(NOTIFICATION);
        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
            int importance = NotificationManager.IMPORTANCE_HIGH;
            NotificationChannel notificationChannel = new NotificationChannel(MainActivity.NOTIFICATION_CHANNEL_ID,"NOTIFICATION_CHANNEL_NAME",importance);
            assert notificationManager !=null;
            notificationManager.createNotificationChannel(notificationChannel);
        }

        int id = intent.getIntExtra(NOTIFICATIONID, 0);
        assert notificationManager != null;
        notificationManager.notify(id, notification);

    }
}
```

secondActivity.java

```
package com.example.fyp;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.widget.Button;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import java.util.Locale;

public class secondActivity extends AppCompatActivity {

    TextToSpeech textToSpeech;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        Button led_on = (Button) findViewById(R.id.led_on);
        Button led_off = (Button) findViewById(R.id.led_off);
        Button fan_on = (Button) findViewById(R.id.fan_on);
        Button fan_off = (Button) findViewById(R.id.fan_off);
        Button buzzer_on = (Button) findViewById(R.id.buzzer_on);
        Button buzzer_off = (Button) findViewById(R.id.buzzer_off);

        textToSpeech = new TextToSpeech(getApplicationContext(), status -> {
            if (status == TextToSpeech.SUCCESS) {

                textToSpeech.setLanguage(new Locale("ms", "MY"));
            }
        });

        DatabaseReference mDatabase = FirebaseDatabase.getInstance().getReference();

        led_on.setOnClickListener(view -> {
            mDatabase.child("LED").child("Value").setValue(1);
            String s = "Light On";
            int speech = textToSpeech.speak(s, TextToSpeech.QUEUE_FLUSH, null);
        });

        led_off.setOnClickListener(view -> {
            mDatabase.child("LED").child("Value").setValue(0);
            String s = "Light Off";
            int speech = textToSpeech.speak(s, TextToSpeech.QUEUE_FLUSH, null);
        });

        fan_on.setOnClickListener(view -> {
            mDatabase.child("Fan").child("Value").setValue(1);
            String s = "Fan On";
            int speech = textToSpeech.speak(s, TextToSpeech.QUEUE_FLUSH, null);
        });
```

```
        fan_off.setOnClickListener(view -> {
            mDatabase.child("Fan").child("Value").setValue(0);
            String s = "Fan off";
            int speech = textToSpeech.speak(s, TextToSpeech.QUEUE_FLUSH, null);
        });

        buzzer_on.setOnClickListener(view -> {
            mDatabase.child("Buzzer").child("Value").setValue(1);
            String s = "Buzzer On";
            int speech = textToSpeech.speak(s, TextToSpeech.QUEUE_FLUSH, null);
        });

        buzzer_off.setOnClickListener(view -> {
            mDatabase.child("Buzzer").child("Value").setValue(0);
            String s = "Buzzer Off";
            int speech = textToSpeech.speak(s, TextToSpeech.QUEUE_FLUSH, null);
        });


    }
}
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background"
    tools:context=".MainActivity">


    <TextView
        android:id="@+id/personName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="304dp"
        android:contentDescription="@string/display_person_name"
        android:textColorHint="#757575"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />

    <ImageView
        android:id="@+id/image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="@string/picture"
        android:src="@mipmap/ic_launcher"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.496"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.435" />

    <Button
        android:id="@+id/tts"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/btn"
        android:text="@string/speech"
        android:textSize="40dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.17"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/personName"
        app:layout_constraintVertical_bias="0.506" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/btn"
    android:text="Remote"
    android:textSize="40dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.733"
    app:layout_constraintStart_toEndOf="@+id/tts"
    app:layout_constraintTop_toBottomOf="@+id/personName"
    app:layout_constraintVertical_bias="0.505" />

androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_second.xml

```
xml version="1.0" encoding="utf-8"?>
ndroidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="@drawable/background"
  tools:context=".secondActivity">


  <Button
      android:id="@+id/led_on"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:background="@drawable/btn"
      android:text="light on"
      android:textSize="40dp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.228"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.308" />

  <Button
      android:id="@+id/led_off"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginStart="68dp"
      android:background="@drawable/btn"
      android:text="light off"
      android:textSize="40dp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.582"
      app:layout_constraintStart_toEndOf="@+id/led_on"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.308" />

  <Button
      android:id="@+id/fan_on"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="88dp"
      android:background="@drawable/btn"
      android:text="@string/fan_on"
      android:textSize="40dp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.24"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/led_on"
      app:layout_constraintVertical_bias="0.194" />
```

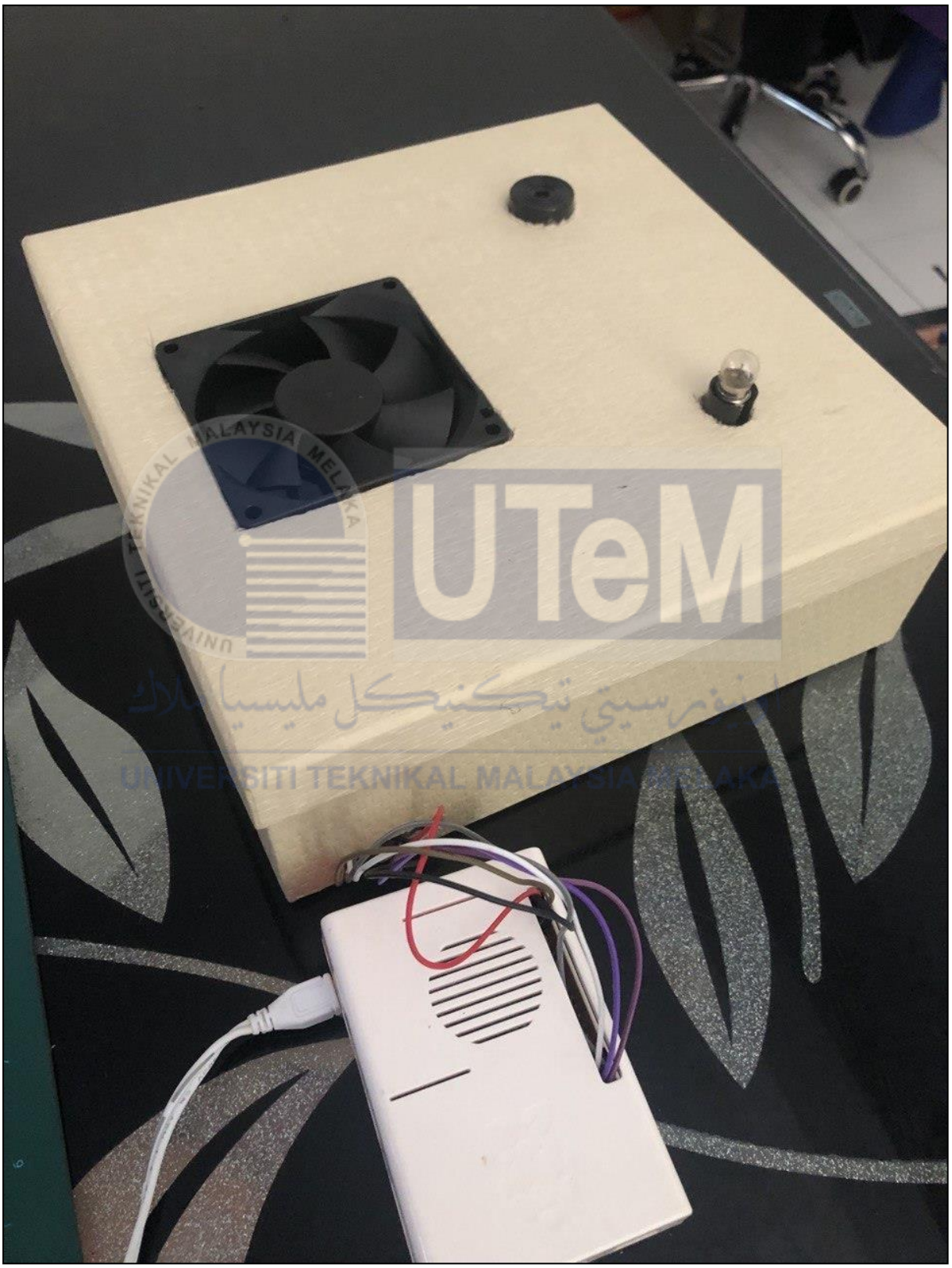
```
    <Button
        android:id="@+id/fan_off"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:background="@drawable/btn"
        android:text="@string/fan_off"
        android:textSize="40dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.656"
        app:layout_constraintStart_toEndOf="@+id/fan_on"
        app:layout_constraintTop_toBottomOf="@+id/led_off"
        app:layout_constraintVertical_bias="0.231" />

    <Button
        android:id="@+id/buzzer_on"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="76dp"
        android:background="@drawable/btn"
        android:text="@string/buzzer_on"
        android:textSize="40dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.212"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/fan_on"
        app:layout_constraintVertical_bias="0.474" />

    <Button
        android:id="@+id/buzzer_off"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/btn"
        android:text="@string/buzzer_off"
        android:textSize="40dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.646"
        app:layout_constraintStart_toEndOf="@+id/buzzer_on"
        app:layout_constraintTop_toBottomOf="@+id/fan_off"
        app:layout_constraintVertical_bias="0.524" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

AndroidManifest.xml

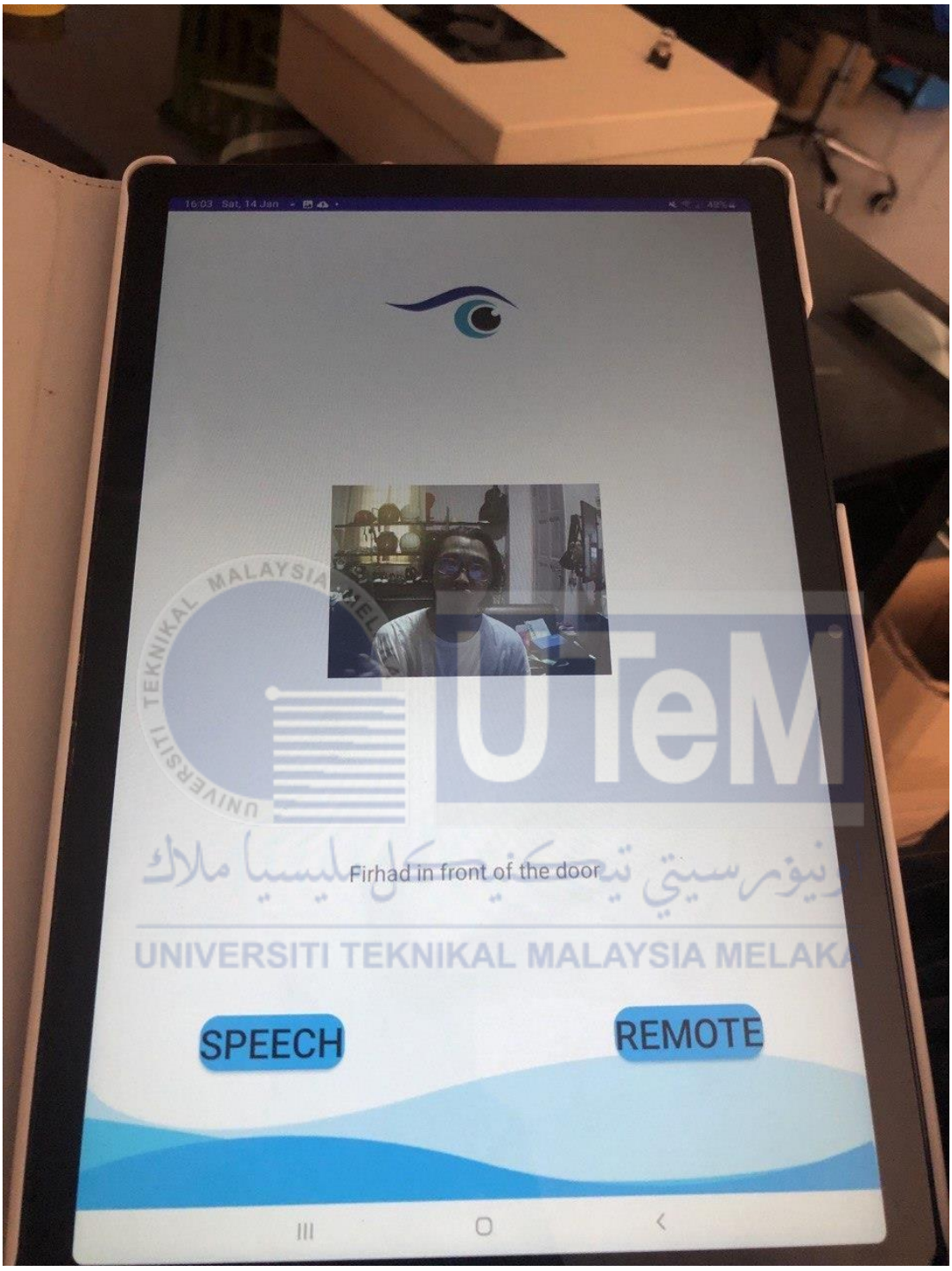
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Fyp"
        tools:targetApi="31">
        <activity
            android:name=".secondActivity"
            android:exported="false">
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>

        <receiver android:name=".NotificationHandler"/>

    </application>

</manifest>
```

Appendix B

Box



Interface



Project Task	Project Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
BDP 1 Briefing by JK, PSM, FTKEE														
Determine FYP Title														
Chapter 1: Introduction														
Study for raw materials														
Chapter 2: Literature Review														
Research for articles and journals														
Research for components and software														
Chapter 3: Methodology														
List out components and compare prices														
System testing & finalize the report														
FYP 1 documentation														
FYP 1 Presentation														

Appendix Gantt Chart BDP 1

Project Task	Project Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
BDP 2 Briefing by JK, PSM, FTKEE														
Buy and receive components														
Software simulation and PCB Layout														
Test components														
Write program code and testing														
Chapter 4 – Result														
Make connections between components and microcontroller														
Chapter 5 – Conclusion														
Prepare and finalize report														
Submit final report to supervisor														
BDP Presentation														
Submit final report with hard cover														

Appendix Gantt Chart BDP 2