

**OMEGA**  
**A Friendly BOT**

by  
**Yash Lohiya (1516410311)**

**Submitted to Department of Computer Science and Engineering**  
**in partial fulfilment of requirements**  
**for the degree of**  
**BACHELOR OF TECHNOLOGY**  
**(COMPUTER SCIENCE & ENGINEERING)**

**Under the supervision of**  
**Ms. Sonam Srivastava**  
**(Assistant Professor)**

**(Pranveer Singh Institute of Technology, Kanpur)**



**Dr. APJ Abdul Kalam Technical University, Lucknow**

**May, 2019**

# **DECLARATION**

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by any other person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Signature

Name: Yash Lohiya

Roll no:1516410311

Signature

Name: Saurabh Singh

Roll no: 1516410242

Date

# CERTIFICATE

This is to certify that Project Report entitled “**Omega- A Friendly BOT**” which is submitted by **Yash Lohiya** and **Saurabh Singh** in partial fulfilment of the requirement for the award of degree B. Tech. in Department of **Computer Science and Engineering** to Pranveer Singh Institute of Technology, Kanpur which is affiliated to Dr A.P.J Abdul Kalam Technical University, Lucknow is a record of the candidate own work carried out by them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

**Date:**

# ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Dr. Vishal Nagar, Department of Computer Science & Engineering, College of Engineering, Lucknow for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant effort that our endeavours have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor M. S. Dhoni, Head of Department of Computer Science and Engineering, College of Engineering, Lucknow for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation in the completion of the project.

Signature:

Name:

Roll no:

Date:

Signature:

Name:

Roll no:

Date:

# **ABSTRACT**

Friendly BOT is an assistance for any common man and a specially challenged person. This standalone assistant interacts solely through the user's voice and actions. It helps user in complex tasks, computing arithmetic problems, searching Wikipedia and monitoring the house and individual as well as a security at home. The user calls the system by speaking a keyword ("Omega") through the microphone, following which the system signals the user (through a beep) to speak the task he wishes to accomplish. The user speaks the voice command and the system gives the output through the speakers attached to the system.

# TABLE OF CONTENTS

DECLARATION.....	ii
CERTIFICATE.....	iii
ACKNOWLEDGEMENT.....	iv
ABSTRACT.....	v
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
LIST OF SYMBOLS.....	viii
LIST OF ABBREVIATIONS.....	ix
CHAPTER 1	
1.1.	
CHAPTER 2 (FEASIBILITY STUDY)	
2.1. TECHNICAL FEASIBILITY.....	
2.2. OPERATIONAL FEASIBILITY.....	
2.3. ECONOMICAL FEASIBILITY.....	
CHAPTER 3 (REQUIREMENT SPECIFICATION)	
3.1. REQUIREMENT ELICITATION.....	
3.2. REQUIREMENT ANALYSIS.....	
3.3 REQUIREMENT GATHERING.....	
3.4. USER REQUIREMENT.....	
3.4.1. SIMPLICITY.....	
3.4.2. SAFETY, SECURITY AND PRIVACY.....	
3.5. HARWARE REQUIREMENTS.....	
3.6. SOFTWARE REQUIREMENTS.....	
3.7. FUNCTIONAL REQUIREMENTS.....	
3.8. NON-FUNCTIONAL REQUIREMENTS.....	
3.8.1. PERFORMANCE REQUIREMENTS.....	

3.8.2. SYSTEM SECURITY.....	
CHAPTER 4 (DESIGN DOCUMENTS)	
4.1. MODULAR ARCHITECTURE DESIGN (HIGH LEVEL) .....	
4.2. DETAILED DESIGN (LOW LEVEL) .....	
4.2.1. DATA FLOW DIAGRAM.....	
4.2.2. FLOW CHART.....	
4.2.3. USECASE DIAGRAM.....	
4.2.4. ER DIAGRAM.....	
CHAPTER 5 (IMPLEMENTATION DETAILS)	
5.1. SNAPSHOTS.....	
5.2. CODING.....	
CHAPTER 6 TESTING.....	
CHAPTER 7 CONCLUSION.....	
CHAPTER 8 FUTURE SCOPE OF THE PROJECT.....	
REFERENCES.....	

# LIST OF TABLES

- Table 2.1 Technical Feasibility
- Table 2.2 Operational Feasibility
- Table 2.3 Economical Feasibility

# LIST OF FIGURES



Figure 1.1 Rasp berry

Figure 4.2 Level 0 DFD

Figure 4.3 Level 1 DFD

Figure 4.4 Level 2 DFD

Figure 4.5 Flow Chart

Figure 4.6 Use Case Diagram

Figure 4.7 ER Diagram

Figure 5.1 Raspberry Pi Architecture

Figure 5.2 Raspberry Pi Pin GPIOs

Figure 5.3 Arduino Nano

Figure 5.4 Arduino Uno

Figure 5.5 IR Sensor

Figure 5.6 Omega Connect

Figure 5.7 Auto mode and web view button

Figure 5.8 Web view

## **LIST OF SYMBOLS**



# LIST OF ABBREVIATIONS

<b>RPM</b>	- Revolutions Per Minute
<b>IR</b>	- Infra Red
<b>USB</b>	- Universal Serial Bus
<b>UI</b>	- User Interface
<b>API</b>	- Application Programming Interface
<b>IDE</b>	- Integrated Development Environment
<b>HTML</b>	- Hypertext Markup Language
<b>CSS</b>	- Cascaded Style Sheet
<b>DFD</b>	- Data Flow Diagram

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1. Problem statement**

The proposed project is specially designed to sort out the problems of specially challenged people and to ease the tasks of normal people because they have to put more effort and more time in accomplishment of any task on daily basis. The problem occurs when they are alone and face problem at home, school or at any organization. The assistant mainly focuses on reducing their time consumption and increasing the efficiency of the person and task as well.

### **1.2. Introduction**

The objective of the project was to make a standalone personal assistant that can be interacted solely through the user's voice and actions. This project is a prototype for a variety of uses. It can help the user in fields like etc. as well as complex tasks like computing arithmetic problems, searching Wikipedia and face recognition for security at home. The user calls the system by speaking a keyword ("Omega") through the microphone, following which the system signals the user (through a beep) to speak the task he wishes to accomplish. The user speaks the voice command and the system gives the output through the speakers attached to the system. The core concepts used in the project are speech-to-text conversion (for understanding user input) and text-to-speech conversion (for giving output to the user) aided by artificial intelligence.

### **1.3. Motivation**

Every year on human rights day, we visit orphanage home to spend time with orphans and to help in some way. We observed that there were so many children having various problems in their daily life. That sprouted an idea of building such a thing which can help them providing comfort to do their work easily. We have also interest in home automation and it was the perfect way to use my interest for social welfare.

### **1.4. Objective and Scope:**

Relevance of the project can be in the following fields:

1. Can be used as a source of communication for specially challenged people as in voice calls, reading emails etc.
2. Central control system for the whole house providing better security
3. Source of entertainment and information for blind/visually impaired

4. Voice based calculator can be used to teach visually impaired students or it can be a game for visually sound students.
5. Security systems can use the face recognition and similarity checking.

## **1.5. Methodology:**

### **1.5.1. Conceptualization**

The “Artificial Intelligence” is the study of how to make computers to do things which at the moment people do better. This one, the process of assistant is often termed as Mundane Tasks. Now, an intelligent assistant is one kind of automation system which can assist a person’s home as well as his/her home. This can identify the mode of internet access easily which means the internet can be [even a hotspot/Wi-Fi] of any kind. As, the growth of personal computers, high speed internet, mobile phones lead to access any kind of information and can take control of any sort of appliances (home appliances). Managing personal life and home appliances are two different vigorous process. To take both vigorous process into a single one and also to increase the security and assist an individual personally AI algorithms is useful. This allows someone to make his/her house an active partner and also his/her own personal information in managing busy life. The aim of this project is to give an overview about raspberry pi based on intelligent systems and also to use voice control to control home appliances and personal life from anywhere in the world.

### **1.5.2. Implementation**

**Microphone** is used to take the audio input of the sound. This audio input when further passed through the system would be searched for keywords. These keywords are essential for the functioning of the voice command system as our modules work on the essence of searching for keywords and giving output by matching keywords as shown in below Figure.

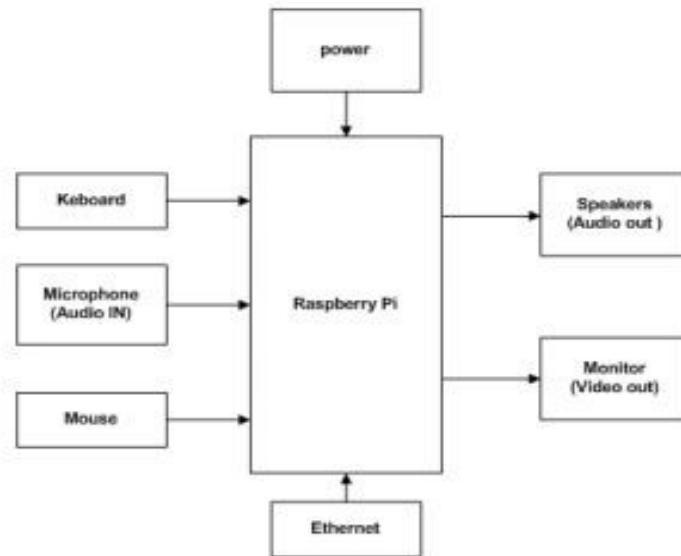


Figure 1.1 Rasp berry

**Raspberry Pi** is the heart of the voice command system as it is involved in every step of processing data to connecting components together. The Raspbian OS is mounted onto the SD card which is then loaded in the card slot to provide a functioning operating system. The Raspberry Pi needs a constant 5V, 2.1 mA power supply. This can either be provided through an AC supply using a micro USB charger or through a power bank.

**Speakers**, once the query put forward by the user has been processed, the text output of that query is converted to speech using the online text to speech converter. Now this speech which is the audio output is sent to the user using the speakers which are running on audio out.

### Flow of Events in Voice Command System

First, when the user starts the system, he uses a microphone to send in the input. Basically, what it does is that it takes sound input from the user and it is fed to the computer to process it further. Then, that sound input is fed to the speech to text converter, which converts audio input to text output which is recognizable by the computer and can also be processed by it. Then that text is parsed and searched for keywords. Our voice command system is built around the system of keywords where it searches the text for key words to match. And once key words are matched then it gives the relevant output. This output is in the form of text. This is then converted to speech output using a text to speech converter which involves using an optical character recognition system. OCR categorizes and identifies the text and then the text to speech engine converts it to the audio output. This output is

transmitted via the speakers which are connected to the audio jack of the raspberry pi.

### **1.5.3. Experimentation**

Our project is dealing with the home automation, image recognition and BOT functioning. In home automation, we have performed certain experiments on electric bulb, fans and few home appliances and it was perfectly running. Image recognition has been checked on numerous people and identification of faces was pretty well.

## CHAPTER 2

# FEASIBILITY STUDY

The feasibility study is important while developing any project. As the name suggests the feasibility study ensures that whether the project is legally or technically feasible. It also ensures that whether the project is successfully implemented or not as it was carefully planned. A feasibility study is the initial design stage of any project.

A feasibility study aims to objectify and rationally uncover the strengths and weakness of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

In this study we are going to analyze the cost of the system and that the system developed is feasibility in every aspect or not, by showing how the technological amalgam is affecting the developed system. In this we are going to see whether the system is economically enough to be used at homes, further we are also going to see what problems can be undergone by this system while being operational in nature that is going to tell what types of problems can be faced at the user's level.

Thus, the feasibility of the system is as follows that will be covering the following aspects of the system, which are as follows

### 2.1. Technical feasibility

In technical feasibility we have checked whether the required technology for developing the system is available or not, whether the resources to develop the system are available also it checks for the software and the hardware requirements. The developed system can be easily maintained; because the system is installed on an adaptable technology hence it is technically feasible.

Table 2.1 Technical Feasibility

S. No.	PARAMETERS ON WHICH THE TECHNICAL FEASIBILITY IS CHECKED	IS IT FEASIBLE OR NOT?
1.	<b>Technologies Used:</b> Android(Java & PHP), PyCharm	YES
2.	<b>Software Required:</b>	YES



	<b>(IDEs/APIs/Libraries/Programming)</b>  Arduino IDE, Sublime Text, OpenCV, Flask, Snowboy, RPi.GPIO, Google Version API, Volley, SQLite, HTML, Javascript, CSS, Python	
3.	<b>Hardware required:</b>  Raspberry Pi Board, Arduino(Uno, Nano), Bluetooth Module(HC-06), Motor Driver Board, PiCamera, 3xUltrasonic Sensor(HC-SR04), IR sensor, Motors(100 RPM), Boost Converter(XL6009), USB Microphone, Speaker, 2x LED Matrix, Wheels, 10000mAh Power Bank	YES

## 2.2. Operational feasibility

Operational feasibility relates to whether the participant is able to handle the new system or not. Factors that are needed to be considered include technical skills of the employee, as well as whether or not training is necessary for the staff and what material is required for the training such as training manual. In this study we also check whether the developed system is working as it is supposed to be or not and whether it is user friendly or not.

Table 2.2 Operational Feasibility

S. No.	PARAMETERS ON WHICH THE TECHNICAL FEASIBILITY IS CHECKED	IS IT FEASIBLE OR NOT?
1.	<b>Technical Skill:</b> The end user need not be skilled to operate the system. Even a layman can operate the device.	YES
2.	<b>Operation Performed:</b> The system is working as it is supposed to. The functioning of the system is as expected earlier.	YES

## 2.3. Economical feasibility

Economically feasibility is the process of identifying the financial benefits and cost associated with the project being developed. Thus, in this we are discussing the cost of the project that is being developed and that whether it is economically feasibility or not.

Table 2.3 Economical Feasibility

S. No.	TYPE OF COST	TOOL NAME	TOTAL PRICE(IN RS.)
1.	Hardware	The whole embedded sys hardware	Rs.5000
2.	Software	PyCharm Android IDE Notepad++	Free
3.	Documentation		Rs.300
4.	Total		Rs.5300

Thus, our project Omega- A Friendly BOT is technically and economically feasible. It is also operationally feasible.

# CHAPTER 3

## REQUIREMENT SPECIFICATION

### 3.1. Requirement Elicitation

Requirement elicitation is the practice of collecting the requirement of a system from, users, customers and other stakeholders. The practice is also sometimes referred to as “**requirement gathering**”.

Requirement elicitation is non-trivial because you can never be sure you get all requirements from the user and customer by just asking them what the system should do or not do (safety and reliability). Requirements elicitation practices include interviews, questionnaires, user observation, workshops, brainstorming, use cases, role playing and prototyping.

Before requirements can be analyzed, modelled, or specified they must be gathered through an elicitation process. Requirements elicitation is a part of the requirements engineering process, usually followed by analysis and specification of the requirements.

Commonly used elicitation processes are the stakeholder meetings or interviews. For example, an important first meeting could be between software engineers and customers where they discuss their perspective of the requirements.

### 3.2. Requirement Analysis

Requirement analysis encompasses those tasks that go into determining the needs or condition to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

### 3.3. Requirement Gathering

Requirements gathering is an essential part of any project management. Understanding fully what a project will deliver is critical to its success. Requirements gathering sounds like common sense, but surprisingly, it's an area that is given far too little attention.

### **3.4. User Requirement**

This chapter categorizes requirements for creating applications at homes. The list of requirements provides guidance for tool developers from an industry perspective. A composition tool should either fulfil these requirements inherently or provide means to solutions developers (e.g. installers, home owners) to relevant aspects with little effort.

#### **3.4.1. Simplicity**

Simplicity describes the complexity of application development. It involves the interaction between the system and the application developer.

- **Learning:** Targeting usually untrained home end users the composition tool must be easy to learn and simple to use.
- **Building/Changing:** Experienced or trained users should be able to quickly develop or modify even complex applications.
- **Levels of abstraction:** Providing multiple layers of abstraction allows to hide implementation details to end users and to expose them to more advanced developers.

#### **3.4.2. Security, Safety and Privacy**

**Process Safety:** Unsafe applications negatively impact devices or the environment in a way which is not foreseen by the developer and must be predicated to ensure process safety.

**Confidentiality:** Information of the system should not be visible to anyone except for a defined group of people.

**Authentication and Authorization:** Enabling confidentiality requires fine grained authentication and authorization mechanisms to access processes, devices and services.

### **3.5. Hardware Requirements**

1. Raspberry Pi Board
2. Arduino (Uno, Nano)
3. Bluetooth Module (HC-06)

4. Motor Driver Board
5. Pi Camera
6. 3xUltrasonic Sensor(HC-SR04)
7. IR sensor, Motors (100 RPM)
8. Boost Converter(XL6009)
9. USB Microphone
10. Speaker
11. 2x LED Matrix
12. Wheels
13. 10000mAh Power Bank

### **3.6. Software Requirements**

1. Python idle
2. Android Studio
3. Sublime text
4. Arduino IDE

### **3.7. Functional Requirements**

1. Movement of BOT for ordering some stuff
2. Monitoring and Intruder detection
3. Playing beats and rhythms
4. Wikipedia search by voice or speaking

### **3.8. Non-Functional Requirements**

#### **3.8.1. Performance Requirements**

Performance requirements define how well the system performs certain functions under specific conditions. BOT stands perfect on various parameters like speed of response, execution time of assigned task etc.

### **3.8.2. System Security**

The BOT acts only after listening the keyword “Omega”. Specially for the physically challenged people BOT sends continuous Gmail notification to the end user to let him know by the pictures whether everything is alright.

The administrator will have all rights to operate the BOT from android application while being at the different place at that particular moment.

# **CHAPTER 4**

## **DESIGN DOCUMENTS**

### **4.1. Modular Architecture Design (High Level)**

## 4.2 Detailed Design (Low Level)

### 4.2.1. Data Flow Diagram(DFD)

#### Level-0 (Context Diagram)

In this level, we have depicted the basic functioning of BOT. Our project has mainly the user (common man or specially challenged) and Omega, the BOT itself. An intruder has also been shown in this level.

User puts any kind of query before the BOT and it performs the action and responds accordingly. If there comes any intruder, it detects and warn the administrator.

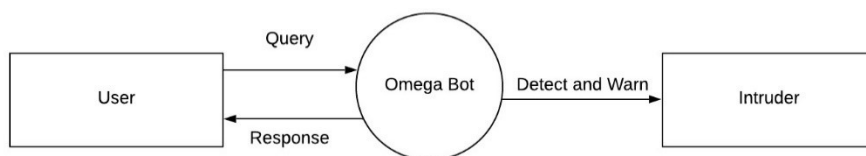


Figure 4.2 0-Level DFD

#### Level-1 DFD

It's has been derived with reference to the context diagram. First level DFD shows the main processes within the BOT system.

In level 1, query passes through a query processor. This query acts like an input to the Omega. If there is a case of monitoring, admin request for video to check whether everything is fine or not. As the query passes and admin request for the video, in the same backward way user gets the desired output.



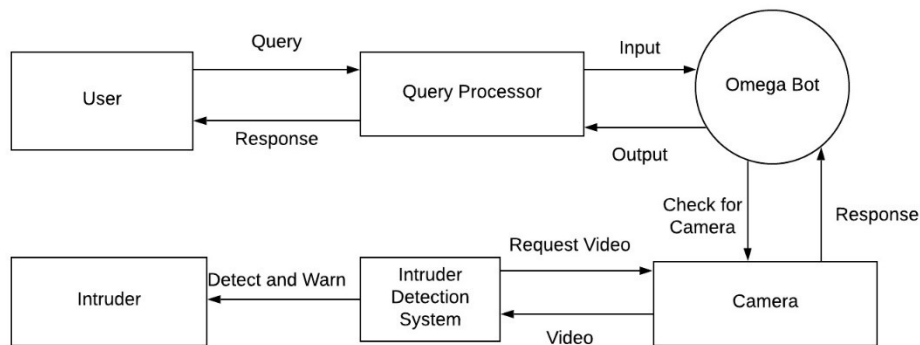


Figure 4.3 Level 1 DFD

## Level 2 DFD

In level 2 DFD, every functions offer more detailed look of the processes. It can be used to plan or record the specific makeup of a system.

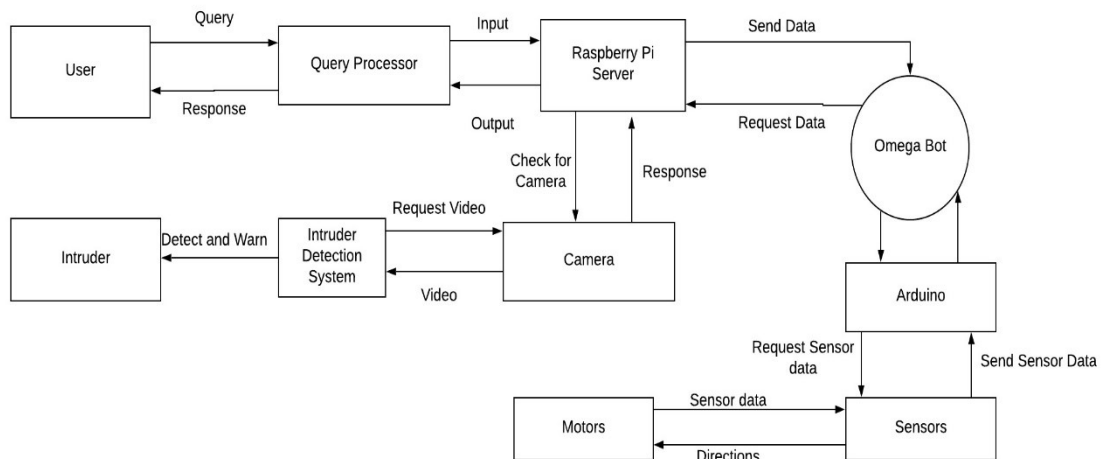


Figure 4.4 Level 2 DFD

## 4.2.2. Flow Chart

A flowchart is a type of diagram that represents an algorithm, workflow or process. It shows diagrammatic representation of how actually the BOT works. It is just flow

of the BOT's processes. Here, the flow has been represented through four switch cases of BOT's main functioning mechanism.

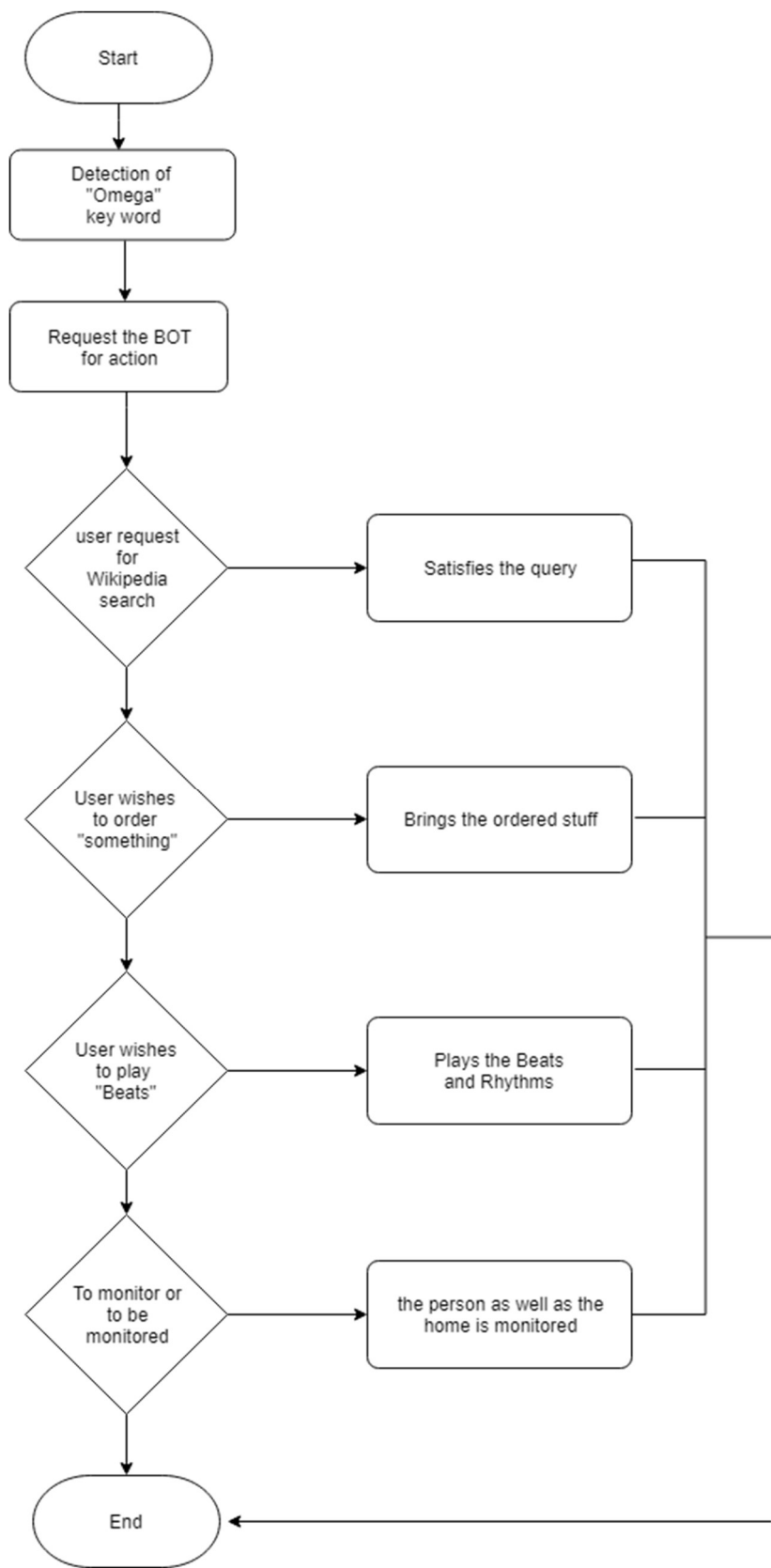


Figure 4.4 Flow Chart

#### 4.2.3. Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

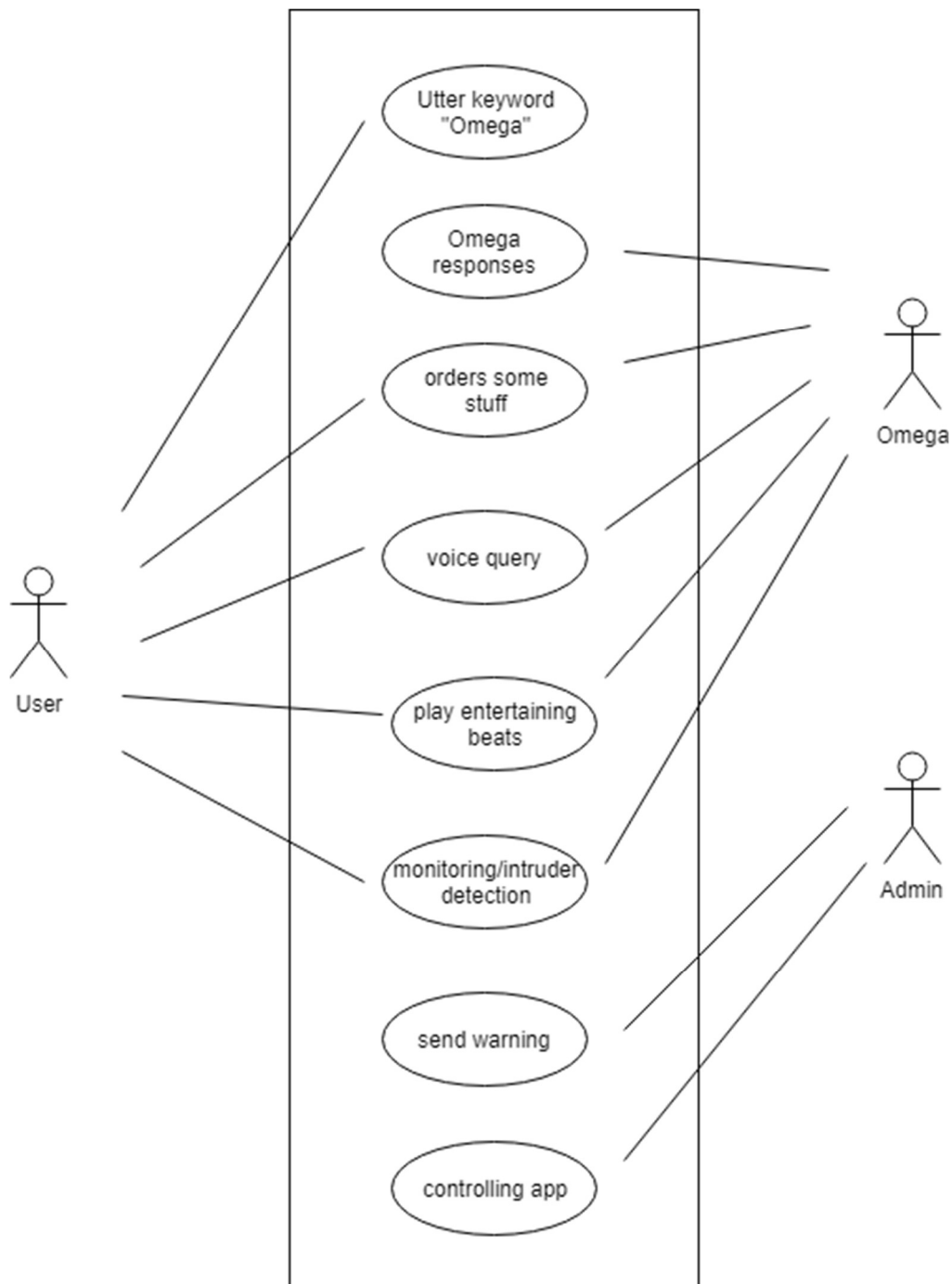


Figure 4.6 Use Case Diagram

#### 4.2.4. Entity Relationship (ER) Diagram

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. Here, we have described how Omega and user are inter-related in terms of functionality.

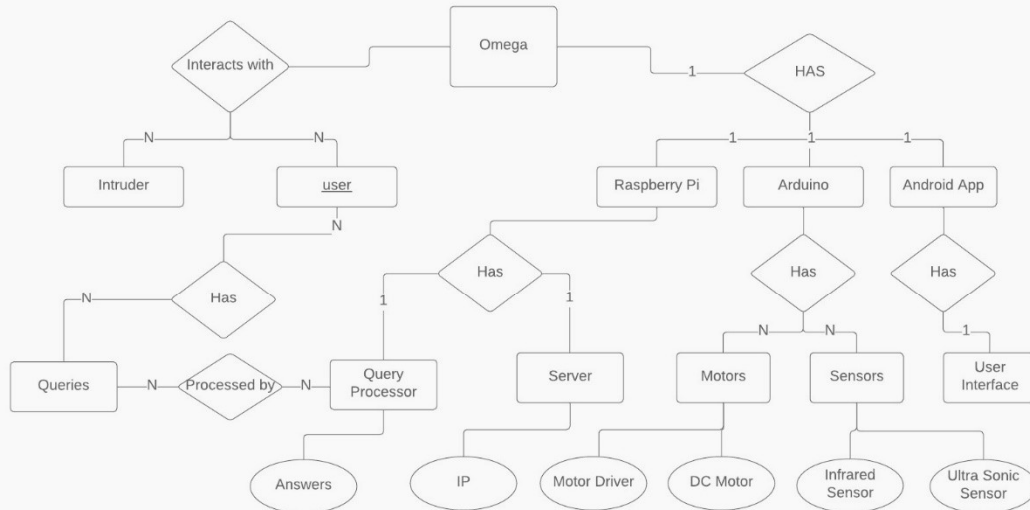


Figure 4.7 ER Diagram

## CHAPTER 5

# IMPLEMENTATION DETAILS

## 5.1. Snapshots

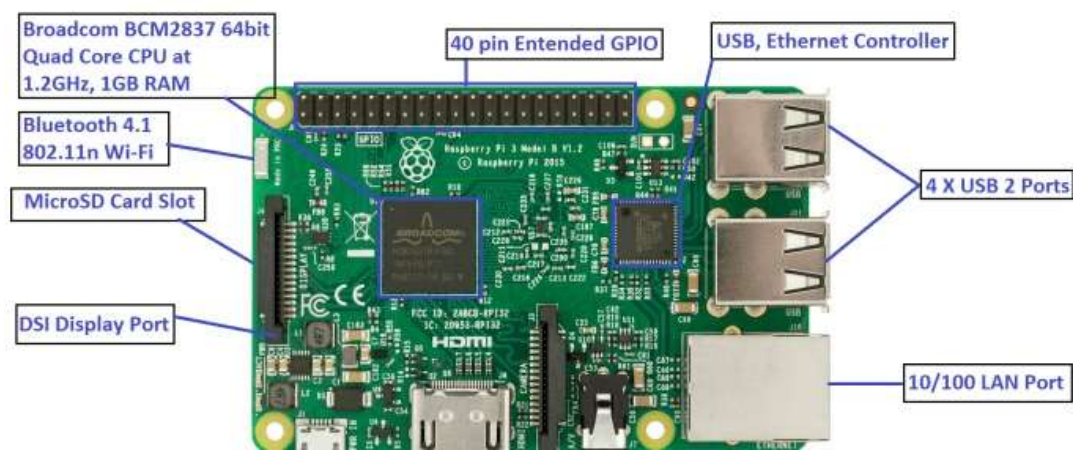


Figure 5.1 Raspberry Pi Architecture

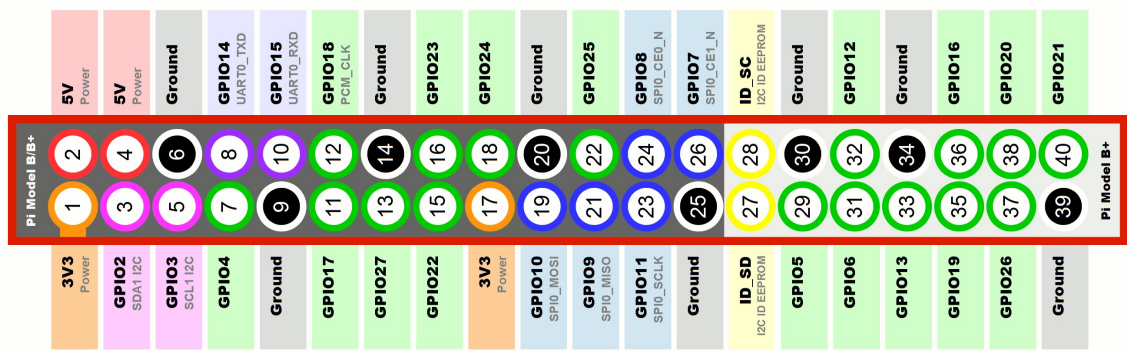


Figure 5.2 Raspberry Pin GPIOs

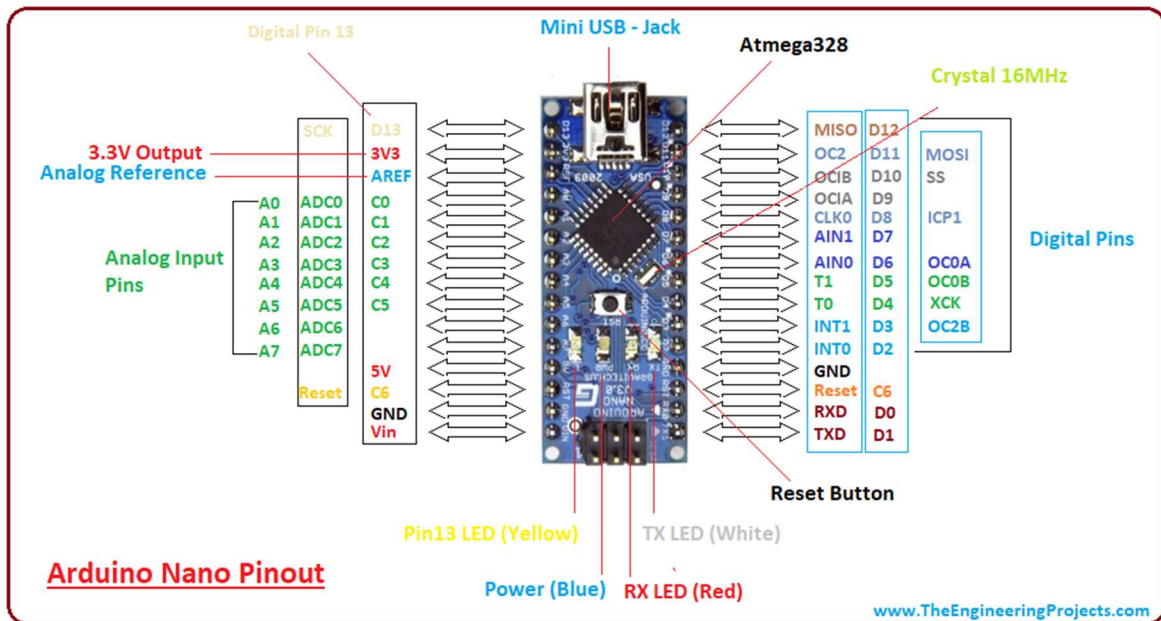


Figure 5.3 Arduino Nano

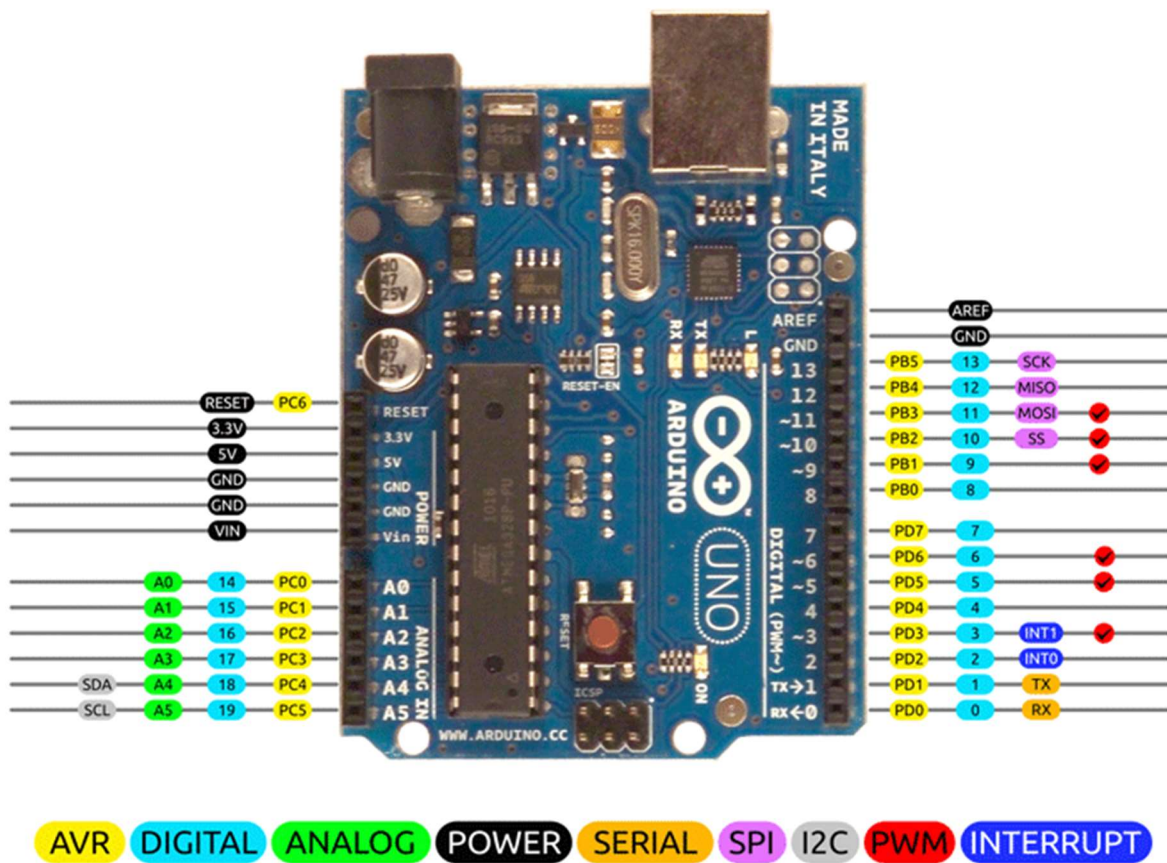


Figure 5.4 Arduino Uno

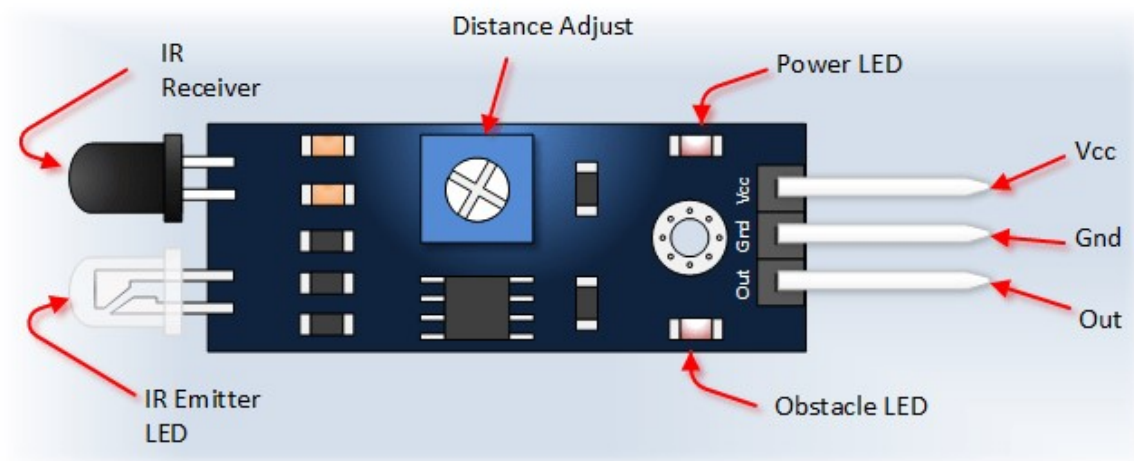


Figure 5.5 IR sensor



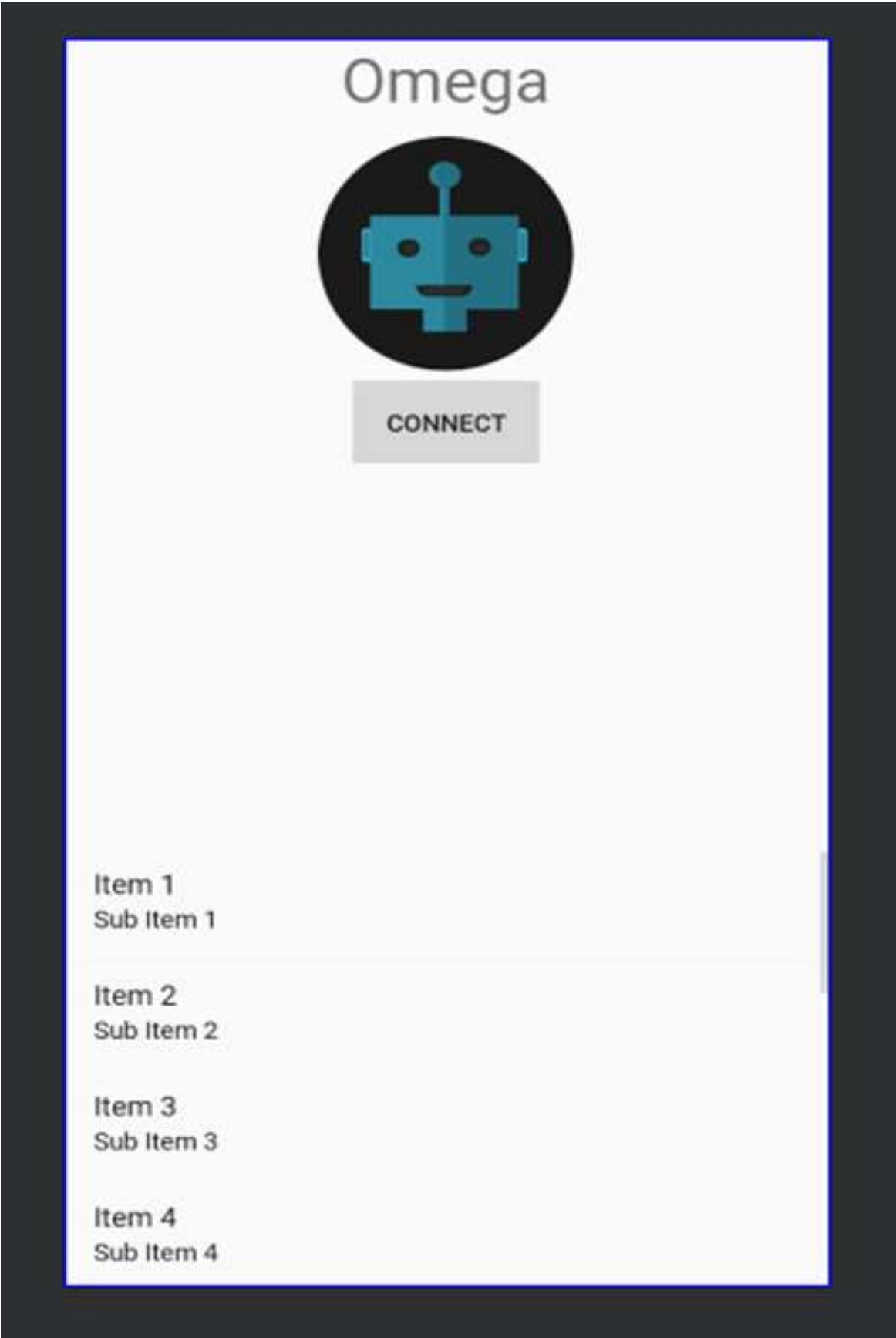


Figure 5.6 Omega Connect

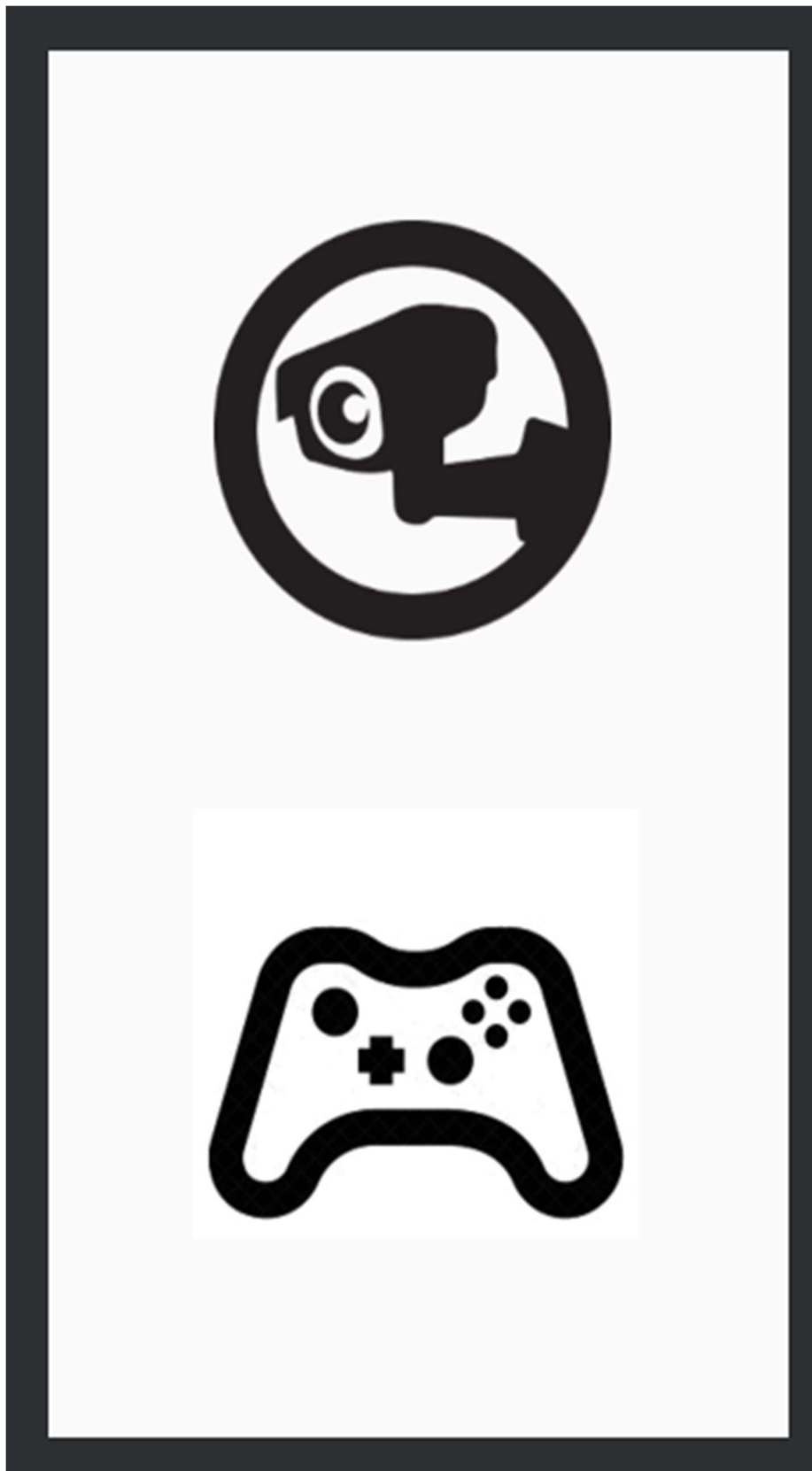


Figure 5.7 Auto mode and Web View Button

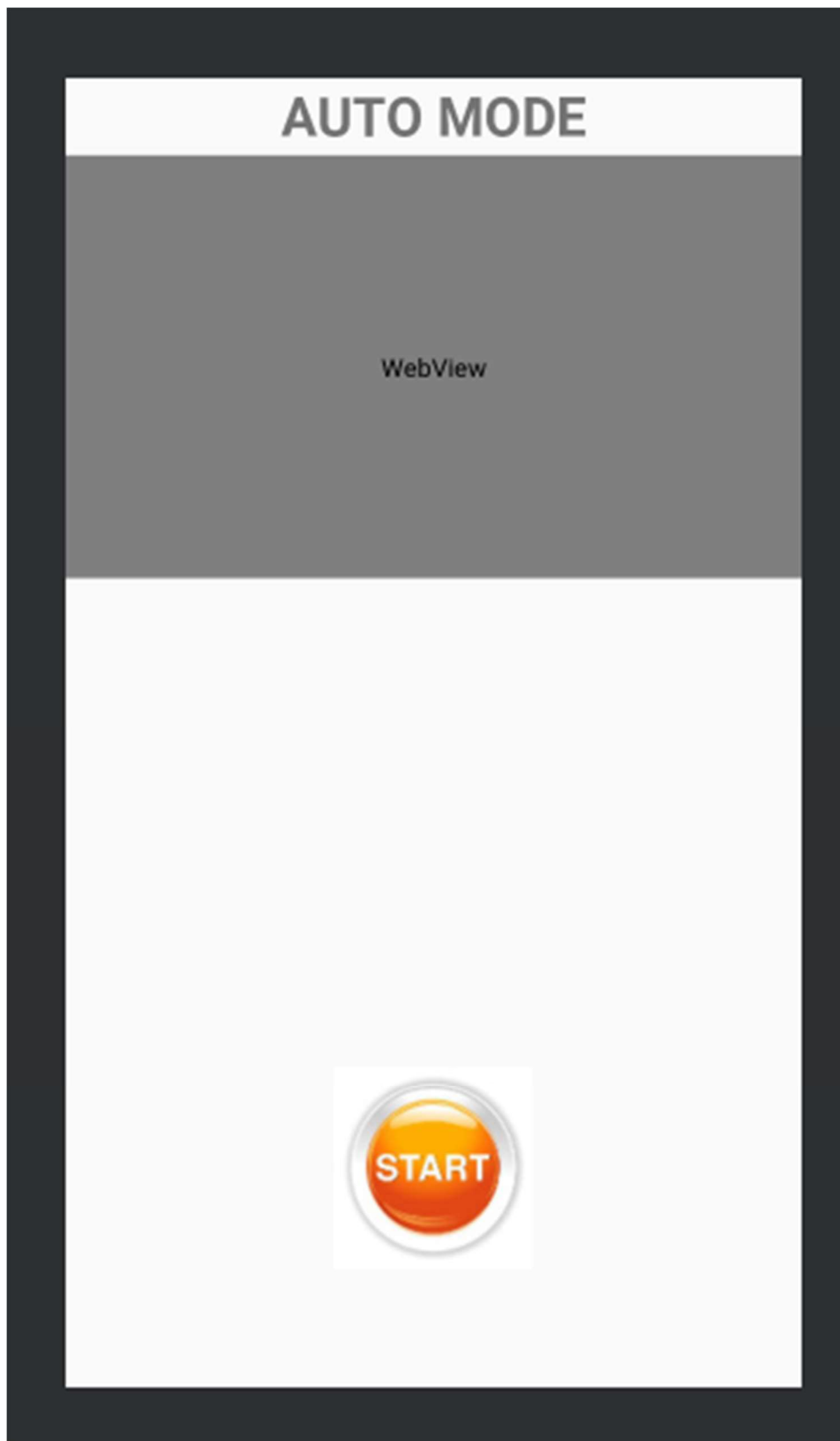


Figure 5.8 Web View

## 5.2. Coding

### (a). Face recognition

```
import cv2
import os
import numpy as np
from imutils.video.pivideostream import PiVideoStream
import imutils
import time
from PIL import Image

def prepare_dataset():
    detector =
cv2.CascadeClassifier("cascades/lbpcascade.xml")
    vs=PiVideoStream().start()
    time.sleep(2.0)
    sample=0
    while(sample!=20):
        frame=np.flip(vs.read(),0)
        gray_frame=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
        frame=np.array(frame)

        face=detector.detectMultiScale(gray_frame,scaleFactor=1.
3,minNeighbors=5)
        for(x,y,w,h) in face:

            cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
            sample=sample+1

            cv2.imwrite("DataSet/User."+str(12)+"."+str(sample)+".jpg",gray_frame[y:y+w,x:x+h])
            cv2.imshow('frame',frame)
            cv2.waitKey(10)
    vs.stop()
    cv2.destroyAllWindows()

def train_on_dataset():
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    path='DataSet'
    imagePaths=[os.path.join(path,f) for f in
os.listdir(path)]
    faces=[]
    IDs=[]
    for impath in imagePaths:
        faceImg=Image.open(impath)
        faceNp=np.array(faceImg,'uint8')
        ID = int(os.path.split(impath)[-1].split('.')[1])

        faces.append(faceNp)
        IDs.append(ID)
    print(IDs)
    IDNp=np.array(IDs)
```

```

        recognizer.train(faces, IDNp)
        recognizer.save('TrainingFiles/trainedData.yml')
        cv2.destroyAllWindows()

def face_recognition():
    detector =
cv2.CascadeClassifier("cascades/lbpcascade.xml")
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read("TrainingFiles/trainedData.yml")
    vs=PiVideoStream().start()
    time.sleep(2.0)
    while(True):
        frame=np.flip(vs.read(),0)
        gray_frame=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
        frame=np.array(frame)

        face=detector.detectMultiScale(gray_frame,scaleFactor=1.
3,minNeighbors=5)
        for(x,y,w,h) in face:

            cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0),2)

            Id,conf=recognizer.predict(gray_frame[y:y+h,x:x+w])
                print(Id," ",conf)
                cv2.imshow('frame',frame)
                if cv2.waitKey(10) & 0xFF==ord('q'):
                    break
    vs.stop()
    cv2.destroyAllWindows()

if __name__=='__main__':
    prepare_dataset()
    train_on_dataset()
    face_recognition()

```

## **(b). test.py**

```

import cv2
import numpy as np
import copy
import math
#from appscript import app

# Environment:
# OS      : Mac OS EL Capitan
# python: 3.5
# opencv: 2.4.13

# parameters
cap_region_x_begin=0.5 # start point/total width

```

```

cap_region_y_end=0.8 # start point/total width
threshold = 60 # BINARY threshold
blurValue = 41 # GaussianBlur parameter
bgSubThreshold = 50
learningRate = 0

# variables
isBgCaptured = 0 # bool, whether the background captured
triggerSwitch = False # if true, keyborad simulator works

def printThreshold(thr):
    print("! Changed threshold to "+str(thr))

def removeBG(frame):
    fgmask = bgModel.apply(frame, learningRate=learningRate)
    # kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,
    (3, 3))
    # res = cv2.morphologyEx(fgmask, cv2.MORPH_OPEN, kernel)

    kernel = np.ones((3, 3), np.uint8)
    fgmask = cv2.erode(fgmask, kernel, iterations=1)
    res = cv2.bitwise_and(frame, frame, mask=fgmask)
    return res

def calculateFingers(res, drawing): # -> finished bool, cnt:
finger count
    # convexity defect
    hull = cv2.convexHull(res, returnPoints=False)
    if len(hull) > 3:
        defects = cv2.convexityDefects(res, hull)
        if type(defects) != type(None): # avoid crashing.
            (BUG not found)

            cnt = 0
            for i in range(defects.shape[0]): # calculate the
angle
                s, e, f, d = defects[i][0]
                start = tuple(res[s][0])
                end = tuple(res[e][0])
                far = tuple(res[f][0])
                a = math.sqrt((end[0] - start[0]) ** 2 +
(end[1] - start[1]) ** 2)
                b = math.sqrt((far[0] - start[0]) ** 2 +
(far[1] - start[1]) ** 2)
                c = math.sqrt((end[0] - far[0]) ** 2 + (end[1]
- far[1]) ** 2)
                angle = math.acos((b ** 2 + c ** 2 - a ** 2) /
(2 * b * c)) # cosine theorem

```

```

        if angle <= math.pi / 2: # angle less than 90
degree, treat as fingers
            cnt += 1
            cv2.circle(drawing, far, 8, [211, 84, 0],
-1)

            return True, cnt
        return False, 0

# Camera
camera = cv2.VideoCapture(0)
camera.set(10,200)
cv2.namedWindow('trackbar')
cv2.createTrackbar('trh1', 'trackbar', threshold, 100,
printThreshold)

while camera.isOpened():
    ret, frame = camera.read()
    threshold = cv2.getTrackbarPos('trh1', 'trackbar')
    frame = cv2.bilateralFilter(frame, 5, 50, 100) #
smoothing filter
    frame = cv2.flip(frame, 1) # flip the frame horizontally
    cv2.rectangle(frame, (int(cap_region_x_begin *
frame.shape[1]), 0),
                    (frame.shape[1], int(cap_region_y_end *
frame.shape[0])), (255, 0, 0), 2)
    cv2.imshow('original', frame)
    print("opennn")

    # Main operation
    if isBgCaptured == 1: # this part wont run until
background captured
        img = removeBG(frame)
        img = img[0:int(cap_region_y_end * frame.shape[0]),
                    int(cap_region_x_begin *
frame.shape[1]):frame.shape[1]] # clip the ROI
        cv2.imshow('mask', img)

        # convert the image into binary image
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        blur = cv2.GaussianBlur(gray, (blurValue, blurValue),
0)

        cv2.imshow('blur', blur)
        ret, thresh = cv2.threshold(blur, threshold, 255,
cv2.THRESH_BINARY)
        cv2.imshow('ori', thresh)

        # get the coutours
        thresh1 = copy.deepcopy(thresh)

```

```

_, contours, hierarchy = cv2.findContours(thresh1,
cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
length = len(contours)
maxArea = -1
if length > 0:
    for i in range(length): # find the biggest
contour (according to area)
        temp = contours[i]
        area = cv2.contourArea(temp)
        if area > maxArea:
            maxArea = area
            ci = i

    res = contours[ci]
    hull = cv2.convexHull(res)
    drawing = np.zeros(img.shape, np.uint8)
    cv2.drawContours(drawing, [res], 0, (0, 255, 0),
2)
    cv2.drawContours(drawing, [hull], 0, (0, 0, 255),
3)

    isFinishCal, cnt = calculateFingers(res, drawing)
    if triggerSwitch is True:
        if isFinishCal is True and cnt <= 2:
            print (cnt)
            #app('System Events').keystroke(' ') #
simulate pressing blank space

    cv2.imshow('output', drawing)

# Keyboard OP
k = cv2.waitKey(10)
if k == 27: # press ESC to exit
    break
elif k == ord('b'): # press 'b' to capture the background
    bgModel = cv2.createBackgroundSubtractorMOG2(0,
bgSubThreshold)
    isBgCaptured = 1
    print( '!!!Background Captured!!!')
elif k == ord('r'): # press 'r' to reset the background
    bgModel = None
    triggerSwitch = False
    isBgCaptured = 0
    print ('!!!Reset BackGround!!!')
elif k == ord('n'):
    triggerSwitch = True
    print ('!!!Trigger On!!!')

```

**(c). camera.py**



```

import cv2
from imutils.video.pivideostream import PiVideoStream
import imutils
import time
import numpy as np

class VideoCamera(object):
    def __init__(self, flip = False):
        self.vs = PiVideoStream().start()
        self.flip = flip
        time.sleep(2.0)

    def __del__(self):
        self.vs.stop()

    def flip_if_needed(self, frame):
        if self.flip:
            return np.flip(frame, 0)
        return frame

    def get_frame(self):
        frame = self.flip_if_needed(self.vs.read())
        ret, jpeg = cv2.imencode('.png', frame)
        return jpeg.tobytes()

    def get_object(self, classifier):
        found_objects = False
        frame = self.flip_if_needed(self.vs.read()).copy()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        objects = classifier.detectMultiScale(
            gray,
            scaleFactor=1.1,
            minNeighbors=5,
            minSize=(30, 30),
            flags=cv2.CASCADE_SCALE_IMAGE
        )

        if len(objects) > 0:
            found_objects = True

        # Draw a rectangle around the objects
        for (x, y, w, h) in objects:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0,
255, 0), 2)

        ret, jpeg = cv2.imencode('.png', frame)
        return (jpeg.tobytes(), found_objects)

```

#### (d). main.py

```
import cv2
import sys
from mail import sendEmail
from flask import Flask, render_template, Response
from camera import VideoCamera
from flask_basicauth import BasicAuth
import time
import threading

email_update_interval = 0 # sends an email only once in this
time interval
video_camera = VideoCamera(flip=True) # creates a camera
object, flip vertically
object_classifier =
cv2.CascadeClassifier("models/frontalface.xml") # an opencv
classifier

# App Globals (do not edit)
app = Flask(__name__)
app.config['BASIC_AUTH_USERNAME'] = 'hh'
app.config['BASIC_AUTH_PASSWORD'] = 'hh'
app.config['BASIC_AUTH_FORCE'] = True

basic_auth = BasicAuth(app)
last_epoch = 0
def check_for_objects():
    global last_epoch
    while True:
        try:
            frame, found_obj =
video_camera.get_object(object_classifier)
            if found_obj and (time.time() - last_epoch)
> email_update_interval:
                last_epoch = time.time()
                print("Sending email...")
                sendEmail(frame)
                print("done!")
        except:
            print("Error sending email: ",
sys.exc_info()[0])
@app.route('/')
@basic_auth.required
def index():
    return render_template('index.html')

def gen(camera):
    while True:
        frame=camera.get_frame()
```

```

        yield (b'--frame\r\n'b'Content-Type:
image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(gen(video_camera),
                    mimetype='multipart/x-mixed-replace;
boundary=frame')

if __name__ == '__main__':
    t = threading.Thread(target=check_for_objects, args=())
    t.daemon = True
    t.start()
    app.run(host='0.0.0.0',port=8080, debug=False)

```

### (e). Android App- Controlling of BOT

```

package com.project.kshitiz.ardublu;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.KeyEvent;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class SecurityFeed extends AppCompatActivity {
    WebView webapp;

    private class MyWebViewClient extends WebViewClient {
        private MyWebViewClient() {

        }

        public boolean shouldOverrideUrlLoading(WebView view,
String url) {
            if
(Uri.parse(url).getHost().equals("192.168.1.100")) {
                return false;
            }
            SecurityFeed.this.startActivity(new
Intent("android.intent.action.VIEW", Uri.parse(url)));
            return true;
        }
    }

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

        getWindow().requestFeature(2);
        setContentView((int)
C0190R.layout.activity_security_feed);
        this.webapp = (WebView)
findViewById(C0190R.id.webapp);
        Activity activity = this;
        this.webapp.setWebViewClient(new MyWebViewClient());
        WebSettings webSettings = this.webapp.getSettings();
        webSettings.setBuiltInZoomControls(true);
        webSettings.setJavaScriptEnabled(true);
        webSettings.setLoadWithOverviewMode(true);
        webSettings.setAllowContentAccess(true);
        webSettings.setUseWideViewPort(true);
        this.webapp.loadUrl("http://192.168.1.100:8080/cam");
        this.webapp.setPadding(0, 0, 0, 0);
        this.webapp.setInitialScale(325);
    }

    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode != 4 || !this.webapp.canGoBack()) {
            return super.onKeyDown(keyCode, event);
        }
        this.webapp.goBack();
        return true;
    }
}

```

### **(f). ConnectionSetUP.java**

```

package com.project.kshitiz.ardublu;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Set;

public class ConnectionSetup extends AppCompatActivity {
    private BluetoothAdapter BA;
    ArrayList BTlist;

```

```

        final int REQUEST_BT_CONST = 10;
        ArrayAdapter adapter;
        /* renamed from: b */
        Button f9b;
        ListView lv;

        /* renamed from:
com.project.kshitiz.ardublu.ConnectionSetup$1 */
        class C01851 implements OnClickListener {
            C01851() {
            }

            public void onClick(View v) {
                ConnectionSetup.this.ConnectToOmega();
            }
        }

        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView((int) C0190R.layout.activity_main);
            this.f9b = (Button) findViewById(C0190R.id.button);
            this.BA = BluetoothAdapter.getDefaultAdapter();
            this.lv = (ListView) findViewById(C0190R.id.listView);
            this.f9b.setOnClickListener(new C01851());
        }

        public void ConnectToOmega() {
            if (this.BA == null) {
                Toast.makeText(this, "Device does not supports
Bluetooth", 0).show();
                return;
            }
            if (!this.BA.isEnabled()) {
                startActivityForResult(new
Intent("android.bluetooth.adapter.action.REQUEST_ENABLE"),
10);
            }
            listBTDevices();
        }

        protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
            if (requestCode != 10) {
                Toast.makeText(getApplicationContext(), "Error
Occured!", 1).show();
            } else if (resultCode == -1) {
                try {
                    Toast.makeText(getApplicationContext(),
"Bluetooth on", 1).show();
                } catch (Exception e) {
                }
            }
        }

```

```

    }
}

public void listBTDevices() {
    Set<BluetoothDevice> pairedDevices =
this.BA.getBondedDevices();
    this.BTlist = new ArrayList();
    String ADDR = null;
    for (BluetoothDevice bt : pairedDevices) {
        ArrayList arrayList = this.BTlist;
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append(bt.getName());
        stringBuilder.append(" ");
        stringBuilder.append(bt.getAddress());
        arrayList.add(stringBuilder.toString());
        if (bt.getName().equals("HC-05")) {
            ADDR = bt.getAddress();
        }
    }
    Toast.makeText(getApplicationContext(), "Showing
Paired Devices", 0).show();
    this.adapter = new ArrayAdapter(this, 17367043,
this.BTlist);
    this.lv.setAdapter(this.adapter);
    if (ADDR != null) {
        Intent i = new Intent(this, Controls.class);
        stringBuilder = new StringBuilder();
        stringBuilder.append("");
        stringBuilder.append(ADDR);
        i.putExtra("btAddress", stringBuilder.toString());
        Toast.makeText(this, "I m here", 0).show();
        startActivity(i);
        return;
    }
    Toast.makeText(this, "First Pair up with HC-05",
0).show();
}
}

```

### **(g). controls.java**

```

package com.project.kshitiz.ardublu;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.os.AsyncTask;
import android.os.Bundle;

```

```

import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.KeyEvent;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnTouchListener;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.ImageButton;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import java.io.IOException;
import java.util.UUID;
import okhttp3.MediaType;
import okhttp3.OkHttpClient;
import okhttp3.Request.Builder;
import okhttp3.RequestBody;

public class Controls extends AppCompatActivity {
    public static final MediaType JSON =
    MediaType.parse("application/json; charset=utf-8");
    static final UUID myUUID = UUID.fromString("00001101-0000-
1000-8000-00805F9B34FB");
    WebView FTV;
    String Mode;
    String address;
    BluetoothAdapter ba;
    OkHttpClient client;
    ImageButton down;
    private boolean isBtConnected = false;
    ImageButton left;
    BluetoothDevice mmDevice;
    BluetoothSocket mmSocket = null;
    TextView mode;
    ProgressBar pb;
    ImageButton right;
    ImageButton start;
    ImageButton stop;
    ImageButton up;

    /* renamed from: com.project.kshitiz.ardubluе.Controls$1
    */
    class C01861 implements OnClickListener {
        C01861() {
        }

        public void onClick(View v) {
            try {
                Controls.this.mmSocket.close();
            }
        }
    }

```

```

        } catch (IOException e) {
            e.printStackTrace();
        }
        try {
            Controls.this.mmSocket.close();
        } catch (IOException e2) {
            e2.printStackTrace();
        }
        Controls.this.Mode = "{\\"mode\\":\\"auto\\"}";
        new setMode().execute(new Void[0]);
    }
}

/* renamed from: com.project.kshitiz.ardubluetooth.Controls$2
*/
class C01872 implements OnClickListener {
    C01872() {

        public void onClick(View v) {
            Controls.this.pb.setVisibility(0);
            Controls.this.ba =
BluetoothAdapter.getDefaultAdapter();
            Controls.this.mmDevice =
Controls.this.ba.getRemoteDevice(Controls.this.address);
            Controls.this.mode.setText("CONTROL MODE");
            new btSocket().execute(new Void[0]);
            Controls.this.Mode = "{\\"mode\\":\\"control\\"}";
            new setMode().execute(new Void[0]);
        }
    }

    private class MyTouchListener implements OnTouchListener {
        boolean turn;

        private MyTouchListener() {
            this.turn = true;
        }

        public boolean onTouch(View v, MotionEvent event) {
            int eventaction = event.getAction();
            int val = 5;
            if (eventaction == 0 && this.turn) {
                this.turn = true ^ this.turn;
                switch (v.getId()) {
                    case C0190R.id.down:
                        val = 2;
                        break;
                    case C0190R.id.left:
                        val = 3;
                        break;
                }
            }
        }
    }
}

```



```

        case C0190R.id.right:
            val = 4;
            break;
        case C0190R.id.stop:
            val = 5;
            break;
        case C0190R.id.up:
            val = 1;
            break;
        default:
            break;
    }
    if (Controls.this.mmSocket != null) {
        try {

Controls.this.mmSocket.getOutputStream().write((byte) val);
            Log.d("DONE", "ACTION");
        } catch (IOException e) {
            Log.e("ERROR", "Error writing data");
        }
    }
    } else if (eventaction == 1 && !this.turn) {
        this.turn = true ^ this.turn;
        if (Controls.this.mmSocket != null) {
            try {

Controls.this.mmSocket.getOutputStream().write((byte) 5);
                Log.d("DONE", "ACTION");
            } catch (IOException e2) {
                Log.e("ERROR", "Error writing data");
            }
        }
    }
    return false;
}

private class btSocket extends AsyncTask<Void, Void, Void>
{
    private boolean ConnectSuccess;

    private btSocket() {
        this.ConnectSuccess = true;
    }

    protected void onPreExecute() {
        super.onPreExecute();
        Controls.this.pb.setVisibility(0);
        Toast.makeText(Controls.this, "Establishing
Connection", 0).show();
    }
}

```

```

        protected Void doInBackground(Void... voids) {
            try {
                if (Controls.this.mmSocket == null ||
!Controls.this.isBtConnected) {
                    Controls.this.ba.cancelDiscovery();
                    Controls.this.mmSocket =
Controls.this.mmDevice.createRfcommSocketToServiceRecord(Contr
ols.myUUID);
                    Controls.this.mmSocket.connect();
                }
            } catch (Exception e) {
                this.ConnectSuccess = false;
            }
            return null;
        }

        protected void onPostExecute(Void aVoid) {
            super.onPostExecute(aVoid);
            if (this.ConnectSuccess) {
                Toast.makeText(Controls.this, "Connected",
0).show();
                Controls.this.isBtConnected = true;
            } else {
                Toast.makeText(Controls.this, "Connection
Failed", 0).show();
                Controls.this.finish();
            }
            Controls.this.pb.setVisibility(4);
        }
    }

    private class setMode extends AsyncTask<Void, Void,
String> {
        private setMode() {

        }

        protected String doInBackground(Void... voids) {
            String response = "";
            try {
                return
Controls.this.post("http://192.168.1.100:8080/controls",
Controls.this.Mode);
            } catch (IOException e) {
                e.printStackTrace();
                return response;
            }
        }

        protected void onPostExecute(String s) {
            super.onPostExecute(s);

```

```

        if (s.equals("OK Control")) {
            Controls.this.start.setVisibility(8);
            Controls.this.stop.setVisibility(0);
            Controls.this.up.setVisibility(0);
            Controls.this.down.setVisibility(0);
            Controls.this.left.setVisibility(0);
            Controls.this.right.setVisibility(0);
            Controls.this.FTV.setVisibility(0);
            WebSettings webSettings =
Controls.this.FTV.getSettings();
            webSettings.setBuiltInZoomControls(true);
            webSettings.setJavaScriptEnabled(true);
            webSettings.setLoadWithOverviewMode(true);
            webSettings.setAllowContentAccess(true);
            webSettings.setUseWideViewPort(true);
            Controls.this.FTV.setPadding(0, 0, 0, 0);
            Controls.this.FTV.setInitialScale(250);

Controls.this.FTV.loadUrl("http://192.168.1.100:8080/cam");
        } else if (s.equals("OK Auto")) {
            Controls.this.up.setVisibility(8);
            Controls.this.down.setVisibility(8);
            Controls.this.left.setVisibility(8);
            Controls.this.right.setVisibility(8);
            Controls.this.stop.setVisibility(8);
            Controls.this.start.setVisibility(0);
            Controls.this.FTV.destroy();
            Controls.this.FTV.setVisibility(8);
        } else {
            Toast.makeText(Controls.this, "Unable to start
Control Mode", 0).show();
        }
        Controls.this.pb.setVisibility(8);
    }
}

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getWindow().requestFeature(2);
    setContentView((int) C0190R.layout.activity_controls);
    this.address =
getIntent().getStringExtra("btAddress");
    this.pb = (ProgressBar)
findViewById(C0190R.id.progressBar);
    this.pb.setVisibility(8);
    this.FTV = (WebView) findViewById(C0190R.id.FTV);
    this.FTV.setVisibility(8);
    this.client = new OkHttpClient();
    this.mode = (TextView) findViewById(C0190R.id.mode);
    this.up = (ImageButton) findViewById(C0190R.id.up);
}

```

```

        this.down = (ImageButton)
findViewById(C0190R.id.down);
        this.left = (ImageButton)
findViewById(C0190R.id.left);
        this.right = (ImageButton)
findViewById(C0190R.id.right);
        this.stop = (ImageButton)
findViewById(C0190R.id.stop);
        this.start = (ImageButton)
findViewById(C0190R.id.startImg);
        this.up.setVisibility(8);
        this.down.setVisibility(8);
        this.left.setVisibility(8);
        this.right.setVisibility(8);
        this.stop.setVisibility(8);
        this.stop.setOnClickListener(new C01861());
        this.start.setOnClickListener(new C01872());
        this.up.setOnTouchListener(new MyTouchListener());
        this.down.setOnTouchListener(new MyTouchListener());
        this.left.setOnTouchListener(new MyTouchListener());
        this.right.setOnTouchListener(new MyTouchListener());
        this.stop.setOnTouchListener(new MyTouchListener());
    }

    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if (keyCode != 4 || !this.FTV.canGoBack()) {
            return super.onKeyDown(keyCode, event);
        }
        this.FTV.goBack();
        return true;
    }

    String post(String url, String json) throws IOException {
        return this.client.newCall(new
Builder().url(url).post(RequestBody.create(JSON,
json)).addHeader("Content-Type",
"application/json").addHeader("cache-control", "no-
cache").build()).execute().body().string();
    }
}

```

## **(h). Dashboard.java**

```

package com.project.kshitiz.ardublu;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

```

```

import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageButton;

public class Dashboardo extends AppCompatActivity {
    ImageButton controls;
    ImageButton securityFeed;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int)
C0190R.layout.activity_dash_board);
        final Activity activity = this;
        this.securityFeed = (ImageButton)
findViewById(C0190R.id.security);
        this.controls = (ImageButton)
findViewById(C0190R.id.control);
        this.securityFeed.setOnClickListener(new
OnClickListener() {
            public void onClick(View v) {
                Dashboardo.this.startActivity(new
Intent(activity, SecurityFeed.class));
            }
        });
        this.controls.setOnClickListener(new OnClickListener()
{
            public void onClick(View v) {
                Dashboardo.this.startActivity(new
Intent(activity, ConnectionSetup.class));
            }
        });
    }
}

```

# **CHAPTER 6**

## **TESTING**

# **CHAPTER 7**

## **CONCLUSION**

## **CHAPTER 8**

### **FUTURE SCOPE OF THE PROJECT**

This is the era where everyone is in rush and wants everything to be done at the correct time and in accurate way. This project Omega is basically for assisting people in their daily task whether they are common people or specially challenged.

If we consider the second aspect for specially challenged people, this is something like boon for them. Omega will assist them in various perspective. If the user is challenged in walking, he needs to just say “Omega, bring a glass of water” and it will fetch for him. If the user is challenged in writing or texting, he can send the voice command “Omega, what is Algorithm?” and the user will get instant Wikipedia answer.

Apart from this, in future it can help people in various ways like opening the door by just commanding Omega, identifying someone standing across the door, handling home appliances for example switching on/off the fans or bulbs etc., and a number of things.



# REFERENCES

1. Dahl, George E., et al. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition." *Audio, Speech, and Language Processing, IEEE Transactions on* 20.1 (2012): 30-42.
  2. Chelba, Ciprian, et al. "Large scale language modeling in automatic speech recognition." *arXiv preprint arXiv:1210.8440* (2012).
  3. Schultz, Tanja, Ngoc Thang Vu, and Tim Schlippe. "GlobalPhone: A multilingual text & speech database in 20 languages." *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013.
  4. Tokuda, Keiichi, et al. "Speech synthesis based on hidden Markov models." *Proceedings of the IEEE* 101.5 (2013): 1234-1252.
- International Journal of Scientific & Engineering Research Volume 8, Issue 11, November-2017 1615 ISSN 2229-5518 IJSER © 2017 <http://www.ijser.org>
5. Singh, Bhupinder, Neha Kapur, and Puneet Kaur. "Speech recognition with hidden Markov model: a review." *International Journal of Advanced Research in Computer Science and Software Engineering* 2.3 (2012).
  6. Lamere, Paul, et al. "The CMU SPHINX-4 speech recognition system." *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*. Vol. 1. 2003.
  7. Black, Alan W., and Kevin A. Lenzo. "Flite: a small fast run-time synthesis engine." *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*. 2001.
  8. Lee, Chin-Hui, Frank K. Soong, and Kuldip Paliwal, eds. *Automatic speech and speaker recognition: advanced topics*. Vol. 355. Springer Science & Business Media, 2012.
  9. Elaine Rich, Kelvin Knight, Shiva Shankar B-Nair, "Artificial Intelligence", MC Graw Hill Publication, "2013".
  10. Steven Hickson, "Voice Control and Face Recognition on the Raspberry Pi" [online], Available at <http://stevenhickson.blogspot.com>.