

A  
Mini Project  
On

## **Desktop Assistance**

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

K.THARAKARAM(197R1A05L5)  
K.THARUN KUMAR (197R1A05M2)

Under the Guidance of

**DR.SOMA SHEKAR**

(Assistant Professor)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE,  
New Delhi) Recognized Under Section 2(f) & 12(B) of the UGC Act. 1956, Kandlakoya (V),  
Medchal Road, Hyderabad-501401.

**2019-2023**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



### CERTIFICATE

This is to certify that the project entitled “**Desktop Assistance**” being submitted by K.THARAKA RAM(197R1A05L5) & K.THARUN KUMAR (197R1A05M2) in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**DR.SOMA SHEKAR**  
(Assistant Professor)  
INTERNAL GUIDE

**Dr. A. Raji Reddy**  
DIRECTOR

**Dr. K. Srujan Raju**  
HOD

**EXTERNAL EXAMINER**

**Submitted for viva voice Examination held on**

## ACKNOWLEDGEMENT

A part from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **DR.SOMA SHEKAR**, Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Dr. M . Subha Mastan Rao & J. Narasimharao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju**, Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**K.THARAKARAM**

**(197R1A05L5)**

**K.THARUN KUMAR**

**(197R1A05M2)**

## **ABSTRACT**

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient.

The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time.

## **LIST OF FIGURES/TABLES**

<b>FIGURE NO</b>	<b>FIGURE NAME</b>	<b>PAGE NO</b>
Figure 3.1	Project Architecture of Desktop Assistance	6
Figure 3.2	Use Case Diagram for Desktop Assistance	7
Figure 3.3	Class Diagram for Desktop Assistance	8
Figure 3.4	Sequence diagram for Desktop Assistance	9
Figure 3.5	Activity diagram for Desktop Assistance	10
Figure 5.1	Input for Google search	23
Figure 5.2	Output for Google search	24
Figure 5.3	Input to open Microsoft Office	25
Figure 5.4	Output to open Microsoft Office	26
Figure 5.5	Input to play music	27
Figure 5.6	Output to play music	28

# TABLE OF CONTENTS

## ABSTRACT

## LIST OF FIGURES

### 1. INTRODUCTION

1.1 PROJECT SCOPE 1

1.2 PROJECT PURPOSE 1

1.3 PROJECT FEATURES 1

### 2. SYSTEM ANALYSIS 2

2.1 PROBLEM DEFINITION 2

2.2 EXISTING SYSTEM 2

2.2.1 DISADVANTAGES OF THE EXISTING SYSTEM 3

2.3 PROPOSED SYSTEM 3

2.3.1 ADVANTAGES OF PROPOSED SYSTEM 3

2.4 FEASIBILITY STUDY 4

2.4.1 ECONOMIC FEASIBILITY 4

2.4.2 TECHNICAL FEASIBILITY 4

2.4.3 BEHAVIORAL FEASIBILITY 4

2.5 HARDWARE & SOFTWARE REQUIREMENTS 5

2.5.1 HARDWARE REQUIREMENTS 5

2.5.2 SOFTWARE REQUIREMENTS 5

### 3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE 6

3.2 USE CASE DIAGRAM 7

3.3 CLASS DIAGRAM 8

3.4 SEQUENCE DIAGRAM 9

3.5 ACTIVITY DIAGRAM 10

### 4. IMPLEMENTATION 11

4.1 SAMPLE CODE 11

### 5. RESULT 23

<b>6. TESTING</b>	29
6.1 INTRODUCTION TO TESTING	29
6.2 TYPES OF TESTING	29
6.2.1 UNIT TESTING	29
6.2.2 INTEGRATION TESTING	30
6.2.3 FUNCTIONAL TESTING	30
6.3 TEST CASES	30
6.3.1 TEST CASE 1	31
6.3.2 TEST CASE 2	31
6.3.2 TEST CASE 3	31
<b>7.CONCLUSION &amp; FUTURE SCOPE</b>	
7.1 PROJECT CONCLUSION	32
7.2 FUTURE SCOPE	32
<b>8.BIBILOGRAPHY</b>	
8.1 REFERENCES	33
8.2 GITHUB LINK	33

# **1. INTRODUCTION**



# **1. INTRODUCTION**

## **1.1 PROJECT SCOPE**

Artificial Intelligence when used with machines, it shows us the capability of thinking like humans. In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time.

## **1.2 PROJECT PURPOSE**

As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time

## **1.3 PROJECT FEATURES**

The functionalities include , It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice. It can have some basic conversation

## **2. SYSTEM ANALYSIS**

## **2. SYSTEM ANALYSIS**

### **SYSTEM ANALYSIS**

The IDE used in this project is PyCharm. All the python files were created in PyCharm and all the necessary packages were easily installable in this IDE. For this project following modules and libraries were used i.e. pyttsx3, SpeechRecognition, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, PyQt etc. I have created a live GUI for interacting with the JARVIS as it gives a design and interesting look while having the conversation.

#### **2.1 PROBLEM DEFINITION**

Usually, user needs to manually manage multiple sets of applications to complete one task. For example, a user trying to make a travel plan needs to check for airport codes for nearby airports and then check travel sites for tickets between combinations of airports to reach the destination. There is need of a system that can manage tasks effortlessly. We already have multiple virtual assistants. But we hardly use it.

#### **2.2 EXISTING SYSTEM**

We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of language processing, and voice recognition. They listen the command given by the user as per their requirements and perform that specific function in a very efficient and effective manner.

### **2.2.1 DISADVANTAGES OF EXISTING SYSTEM**

SIRI does not maintain a knowledge database of its own and its understanding comes from the information captured in domain models and data models. Will take some time to put all of the to-do items in – you could spend more time putting the entries in than actually doing the revision.

## **2.3 PROPOSED SYSTEM**

It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command.

### **2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM**

Human effort and saving time. Functionalities of this project include, It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast.

## **2.4 FEASIBILITY STUDY**

Feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed system. Five types of feasibility study are taken into consideration.

### **2.4.1 ECONOMIC FEASIBILITY**

Here, we find the total cost and benefit of the proposed system over current system. For this project, the main cost is documentation cost. User also would have to pay for microphone and speakers. Again, they are cheap and available. As far as maintenance is concerned, JARVIS won't cost too much

### **2.4.2 TECHNICAL FEASIBILITY**

It includes finding out technologies for the project, both hardware and software. For virtual assistant, user must have microphone to convey their message and a speaker to listen when system speaks. These are very cheap now a days and everyone generally possess them. Besides, system may or maynot be needed internet or connection. While using JARVIS, make sure you have a steady internet connection if it is available. It is also not anissue in this era where almost every home or office has Wi-Fi.

### **2.4.3 BEHAVIORAL FEASIBILITY**

It is the ease and simplicity of operation of proposed system. System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost everyone. Kids who still don't know to write can readout problems for system and get answers.

## **2.5 HARDWARE & SOFTWARE REQUIREMENTS**

### **2.5.1 HARDWARE REQUIREMENTS:**

The software is designed to be light-weighted so that it doesn't be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software requirement for virtual assistant.

Hardware:

- Pentium-pro processor or later.
- RAM 512MB or more.

### **2.5.2 SOFTWARE REQUIREMENTS:**

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

Windows 7(32-bit) or above.

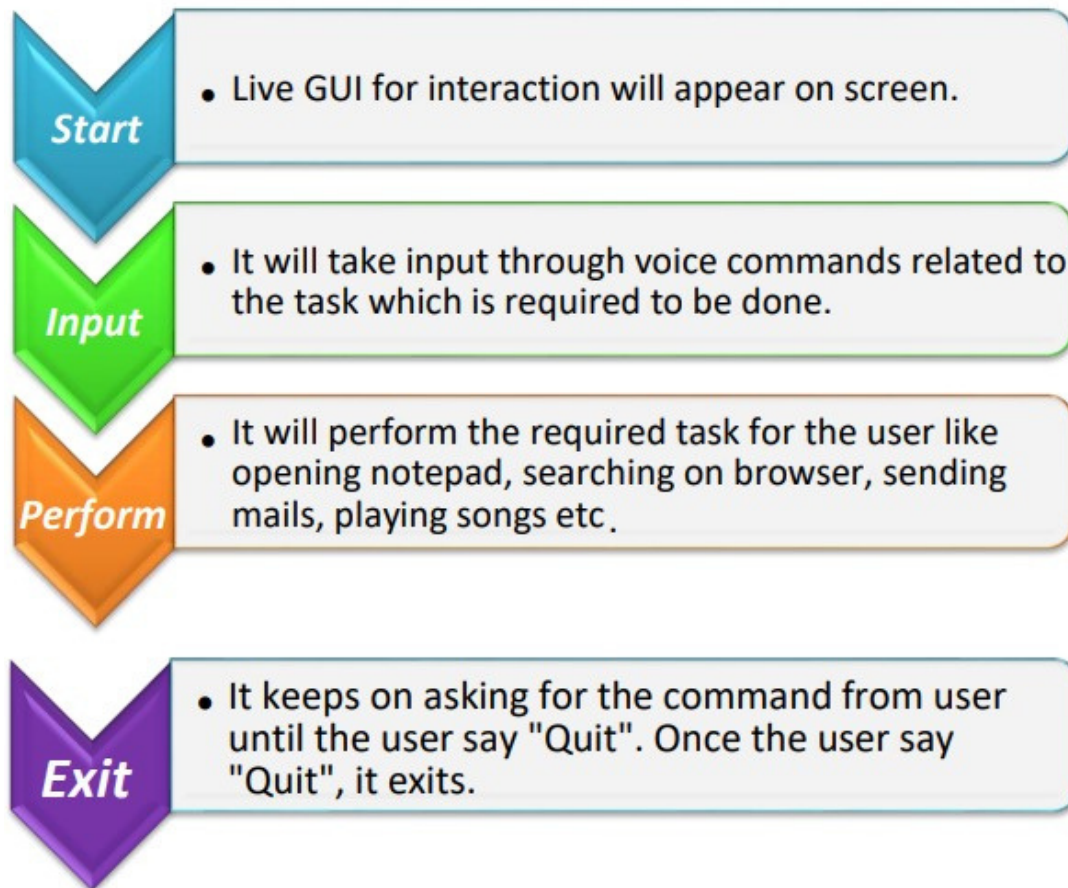
- Python 2.7 or later
- Chrome Driver
- Selenium Web Automation
- SQLite

### **3. ARCHITECTURE**

### 3.ARCHITECTURE

#### 3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from start to exit



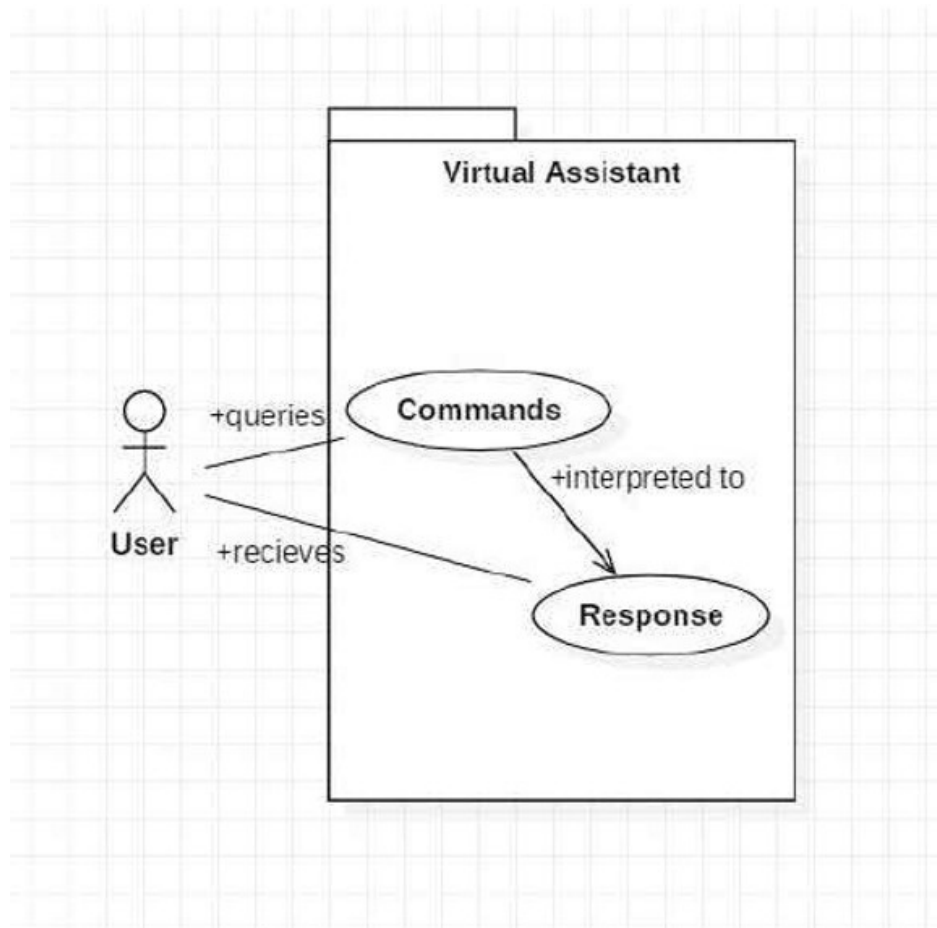
3.1 Project Architecture of Desktop Assistance



## 3.2 USE CASE DIAGRAM

In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user.

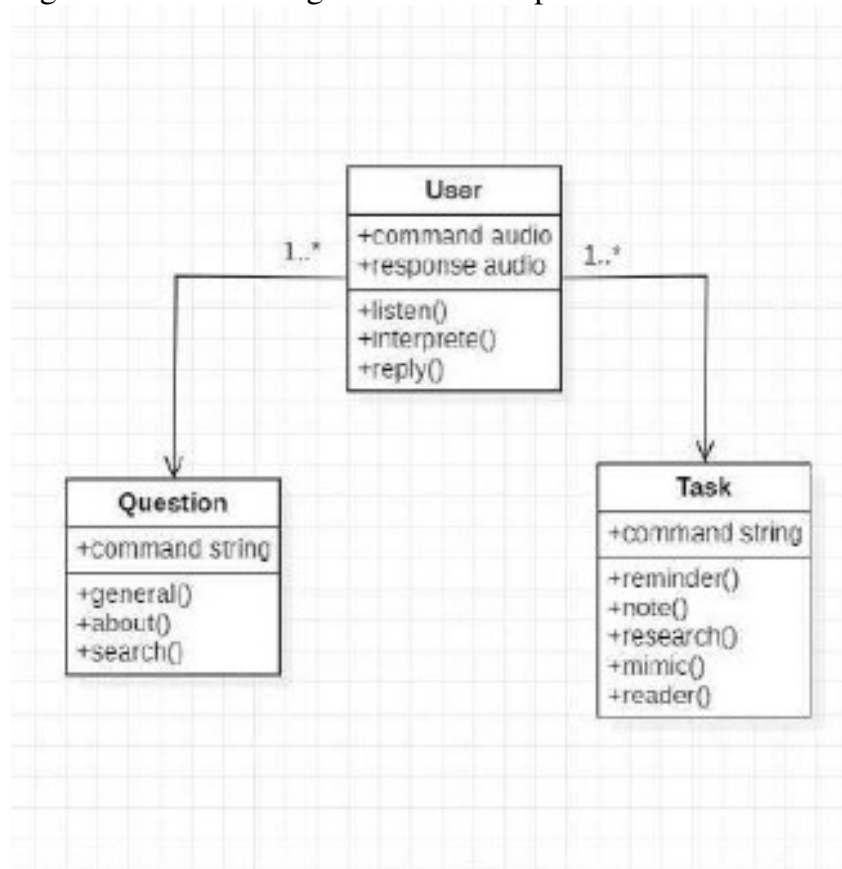
Figure 3.2:Use Case Diagram of Desktop Assistance



### 3.3 CLASS DIAGRAM

The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly. Question class has the command in string format it is interpreted by interpret class. It sends it to general or about or search function based on its identification. The task class also has interpreted command in string format. It has various functions like reminder, note, mimic, research and reader.

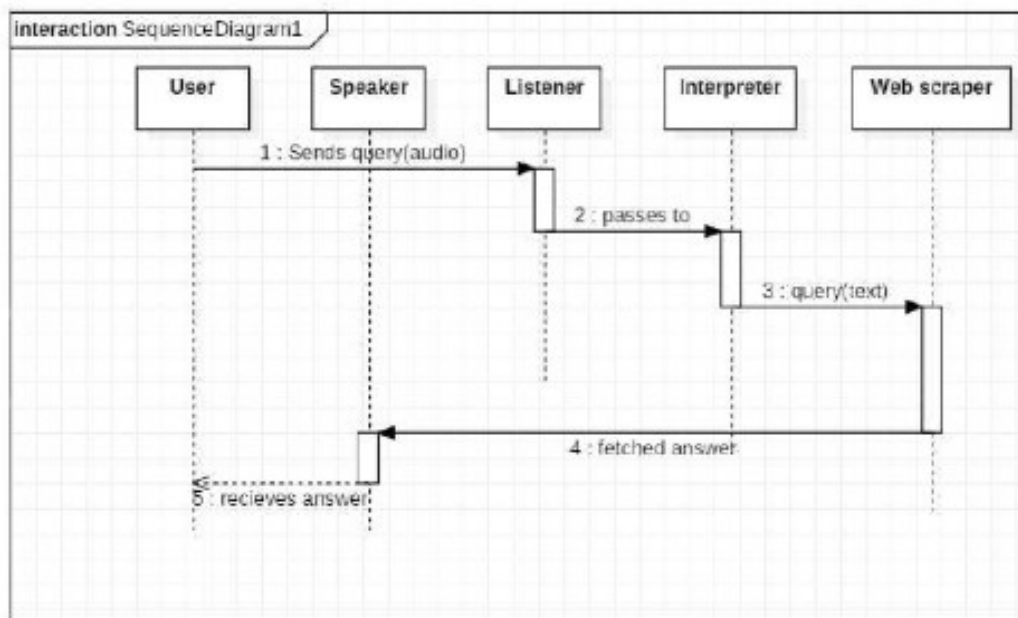
Figure 3.3: Class Diagram for Desktop Assistance



### 3.4 SEQUENCE DIAGRAM

The above sequence diagram shows how an answer asked by the user is being fetched from internet. The audio query is interpreted and sent to Web scraper. The web scraper searches and finds the answer. It is then sent back to speaker, where it speaks the answer to user.

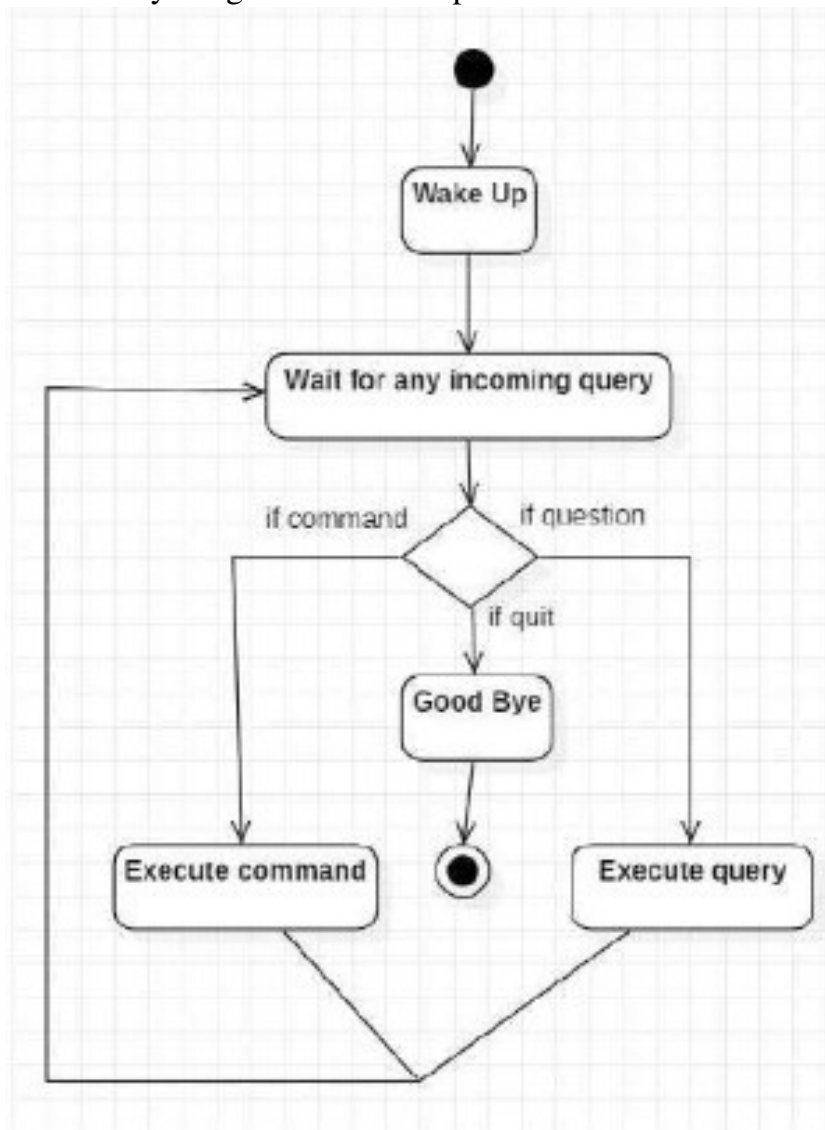
Figure 3.4: Sequence Diagram for Desktop Assistance



### 3.5 ACTIVITY DIAGRAM

Initially, the system is in idle mode. As it receives any wake up call it begins execution. The received command is identified whether it is a questionnaire or a task to be performed. Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits for another command. This loop continues unless it receives quit command. At that moment, it goes back to sleep.

Figure 3.5: Activity Diagram for Desktop Assistance



## **4.IMPLEMENTATION**

## 4.1 SAMPLE CODE

```

import datetime
import os
import smtplib
import subprocess
import sys
import time
import webbrowser
import cv2
import psutil
import pyjokes
import pyttsx3
import pywhatkit as kit
import requests
import speech_recognition as sr
import speedtest
import wikipedia
import winshell
import wolframalpha
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtCore import *
from PyQt5.QtCore import QDate, Qt, QTime, QTimer
from PyQt5.QtGui import *
from PyQt5.QtGui import QMovie
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUiType
from twilio.rest import Client
from coraUI import Ui_MainWindow
engine = pyttsx3.init('sapi5')
engine.setProperty('rate', 140)
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
def speak(audio):
    engine.say(audio)
    print(f"stark : {audio}")
    engine.runAndWait()
def wishMe():
    hour = int(datetime.datetime.now().hour)
    if 0 <= hour < 12:

```

```

speak("Good Morning ")
elif 12 <= hour < 18:
speak("Good Afternoon ")
else:
speak("Good Evening ")
assname = "stark"
speak("I am your Assistant " + assname + " how can i help you")
def sendEmail(to, content):
server = smtplib.SMTP('smtp.gmail.com', 587)
server.ehlo()
server.starttls()
server.login('nikhilgattu9@gmail.com', '#Nikhilkumar2000')
server.sendmail('nikhilgattu9@gmail.com', to, content)
server.close()
def news():
main_url = 'http://newsapi.org/v2/topheadlines?
sources=techcrunch&apikey=06416390f6324ef8af466e86cb99b4e5'
main_page = requests.get(main_url).json()
articles = main_page["articles"]
head = []
day = ["first", "second", "third", "fourth", "fifth"]
for i in articles:
head.append(i["title"])
for i in range(len(day)):
speak(f'today's {day[i]} news is: {head[i]}')
def playgame(self):
speak("Welcome to Tic Tac Toe game!")
theBoard = {'7': '', '8': '', '9': '',
'4': '', '5': '', '6': '',
'1': '', '2': '', '3': ''}
board_keys = []
for key in theBoard:
board_keys.append(key)
def printBoard(board):
print(" " +
board['7'] + '|' + board['8'] + '|' + board['9'])
print(" "+'-+--+')
print(" " +
board['4'] + '|' + board['5'] + '|' + board['6'])

```

```

print(" "+'-+--+')
print(" " +
board['1'] + '|' + board['2'] + '|' + board['3'])
def game():
    turn = 'X'
    count = 0
    for i in range(10):
        printBoard(theBoard)
        print("It's your turn," + turn + " Move to which place?")
        speak("It's your turn " + turn + " Move to which place?")
        move = input()
        if move.isdigit():
            if theBoard[move] == ' ':
                theBoard[move] = turn
                count += 1
            else:
                print("That place is already filled.")
                speak("That place is already filled.")
                continue
            if count >= 5:
                if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top
                    printBoard(theBoard)
                    print("\nGame Over.\n")
                    print(" ** " + turn + " won. **")
                    speak(turn + " won.")
                    break
                # across the middle
                elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ':
                    printBoard(theBoard)
                    print("\nGame Over.\n")
                    print(" ** " + turn + " won. **")
                    speak(turn + " won.")
                    break
                # across the bottom
                elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ':
                    printBoard(theBoard)
                    print("\nGame Over.\n")
                    print(" ** " + turn + " won. **")

```



```

speak(turn + " won.")
break
# down the left side
elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ':
    printBoard(theBoard)
    print("\nGame Over.\n")
    print(" ** " + turn + " won. **")
    speak(turn + " won.")
    break
elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle
    printBoard(theBoard)
    print("\nGame Over.\n")
    print(" ** " + turn + " won. **")
    speak(turn + " won.")
    break
# down the right side
elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ':
    printBoard(theBoard)
    print("\nGame Over.\n")
    print(" ** " + turn + " won. **")
    speak(turn + " won.")
    break
elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal
    printBoard(theBoard)
    print("\nGame Over.\n")
    print(" ** " + turn + " won. **")
    speak(turn + " won.")
    break
elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # diagonal
    printBoard(theBoard)
    print("\nGame Over.\n")
    print(" ** " + turn + " won. **")
    speak(turn + " won.")
    break
if count == 9:
    print("\nGame Over.\n")
    print("It's a Tie!!")
if turn == 'X':

```

```

turn = 'O'
else:
turn = 'X'
else:
print("please enter a valid number")
speak("please enter a valid number")
restart = input("Do want to play Again?(y/n)")
if restart == "y" or restart == "Y":
for key in board_keys:
theBoard[key] = " "
game()
game()
def get_num(self):
dic = {"send it to arawind": 9381146833, "send it to sivaiah sir": 9505838400, "send it
to nikhil":
8019997494}
speak('whom do you want to send the message ')
inpp = self.takecommand()
print(inpp)
if inpp in dic.keys():
a = dic[inpp]
return a
elif "no thanks" in inpp:
return "none"
else:
speak('there is no contact named ' + inpp)
return get_num(self)
# def search_wikihow(query, m=10, lang="en"):
# return list(wikiHow.search(query, m, lang))
class MainThread(QThread):
def _init_(self):
super(MainThread, self)._init_()
def takecommand(self):
r = sr.Recognizer()
with sr.Microphone() as source:
print('listening...')
r.pause_threshold = 1
r.adjust_for_ambient_noise(source)
audio = r.listen(source, phrase_time_limit=5)
try:
print('Recognizing...')
query = r.recognize_google(audio, language='en-in')

```

```

print(f"Human: {query}")
except Exception:
    return "none"
query = query.lower()
return query
def run(self):
    # speak("please say wakeup to continue..")
    # while True:
    # self.query = self.takecommand()
    # if "wake" in self.query or "hello" in self.query:
    self.pertask()
    def pertask(self):
        wishMe()
        while True:
            self.query = self.takecommand()
            if 'open notepad' in self.query:
                npath = "C:\\Windows\\notepad.exe"
                os.startfile(npath)
            elif 'game' in self.query:
                playgame(self)
            elif 'close notepad' in self.query:
                speak("closing notepad")
                os.system("taskkill /f /im notepad.exe")
            elif 'close command prompt' in self.query:
                speak("closing CMD")
                os.system("taskkill /f /im cmd.exe")
            elif 'open command prompt' in self.query:
                npath = "C:\\Windows\\System32\\cmd.exe"
                os.startfile(npath)
            elif 'play' in self.query:
                song = self.query.replace('play', "")
                speak('playing' + song)
                print(song)
                kit.playonyt(song)
            elif 'hello' in self.query or 'hey' in self.query:
                speak("hello there how are you!")
            elif 'who made you' in self.query or 'who created you' in self.query:
                speak("I have been created by team stark.")

```

```

elif 'who are you' in self.query:
    speak('iam stark, I can perform various task\'s')
elif 'tell me a joke' in self.query or 'joke' in self.query:
    speak(pyjokes.get_joke())
elif 'how are you' in self.query:
    speak("I am fine, Thank you")
elif 'fine' in self.query or "good" in self.query:
    speak("It's good to know that your fine")
elif 'wikipedia' in self.query:
    qquery = self.query.replace("wikipedia", "")
    print(qquery)
    info = wikipedia.summary(qquery, sentences=1)
    print(info)
    speak('According to wikipedia ' + info)
elif 'open youtube' in self.query:
    speak('here you go')
    webbrowser.open("https://www.youtube.com")
elif 'open google' in self.query:
    speak('what should i have to search ')
    cm = self.takecommand().lower()
    webbrowser.open(f"{cm}")
elif 'send a message' in self.query:
    timef = datetime.datetime.now().strftime('%I:%M %p')
    timee = datetime.datetime.now().strftime('%H:%M')
    s1 = str(timee[0]) + str(timee[1])
    s2 = str(timee[3] + timee[4])
    if s2[0] == '0':
        s2 = s2[1]
    elif 'AM' in timef and s1 == '12':
        s1 = '00'
    speak('what i have to send ')
    mess = self.takecommand()
    number = get_num(self)
    if number != "none":
        speak('message sending..')
        kit.sendwhatmsg('+91' + str(number), mess,
            int(s1), int(s2) + 2)
    else:
        pass
elif 'send a email' in self.query:
    try:

```

```

speak("What should I say?")
content = self.takecommand().lower()
to = input("enter email : ")
sendEmail(to, content)
speak("Email has been sent !")
except Exception as e:
    print(e)
speak("I am not able to send this email")
elif "write a note" in self.query:
    speak("What should i write, ")
    note = self.takecommand()
    file = open('jarvis', 'w')
    speak("Should i include date and time")
    snfm = self.takecommand()
    if 'yes' in snfm or 'sure' in snfm:
        timee = datetime.datetime.now().strftime('%I:%M %p')
        file.write(timee)
        file.write(" :- ")
        file.write(note)
        speak('notes taken!')
    else:
        file.write(note)
        speak('notes taken!')
elif "calculate" in self.query:
    app_id = "E4AJ9J-WGXTR9AGEX"
    client = wolframalpha.Client(app_id)
    ind = self.query.lower().split().index("calculate")
    qqquery = self.query.split()[ind + 1:]
    res = client.query(" ".join(qqquery))
    answer = next(res.results).text
    speak("The answer is " + answer)
elif "stark what is" in self.query or "stark who is" in self.query or "birth" in self.query:
    try:
        app_id = "E4AJ9J-WGXTR9AGEX"
        client = wolframalpha.Client(app_id)
        ind = self.query.lower().split().index("is")
        qqquery = self.query.split()[ind + 1:]
        res = client.query(" ".join(qqquery))
        answer = next(res.results).text

```

```

speak(answer)
except Exception:
speak("unable to fetch the information")
elif 'show me the notes' in self.query:
speak("Showing Notes")
file = open("jarvis", "r")
s = file.read()
print(s)
speak(s)
elif "restart" in self.query:
subprocess.call(["shutdown", "/r"])
exit()
elif 'empty recycle bin' in self.query:
winshell.recycle_bin().empty(confirm=False, show_progress=True, sound=True)
speak("Recycle Bin Recycled")
elif "hibernate" in self.query or "sleep" in self.query:
speak("Hibernating")
subprocess.call(["shutdown", "/h"])
exit()
elif "log off" in self.query or "shutdown" in self.query:
speak("Make sure all the application are closed before sign-out")
time.sleep(10)
subprocess.call(["shutdown", "/l"])
exit()
elif 'no thanks' in self.query or 'thank you' in self.query:
speak('have a good day, thank you')
exit()
elif 'news today' in self.query or 'news' in self.query:
speak('please wait , fetching the latest news')
news()
elif "open camera" in self.query:
cap = cv2.VideoCapture(0)
while True:
ret, img = cap.read()
cv2.imshow('webcam', img)
k = cv2.waitKey(50)
if k == 27:
break
cap.release()
cv2.destroyAllWindows()
# elif "set alarm" in self.query:

```

```

# nn=int(datetime.datetime.now().hour)
# if nn==22:
# music_dir="
# songs=os.listdir(music_dir)
# os.startfile(os.path.join(music_dir,songs[0]))
elif 'send sms' in self.query:
speak('what should i have to send')
msg = self.takecommand()
account_sid = 'AC1016d7c21e071e441350f7c9d774cd33'
auth_token = '9294e3235de29d9831295d5e93edcd9e'
client = Client(account_sid, auth_token)
message = client.messages \
.create(
body=msg,
from_='+14157797080',
to='+918019997494'
)
print(message.sid)
speak('message sent..')
elif 'where are we' in self.query:
speak('let me check!')
try:
ipadd = requests.get('https://api.ipify.org').text
url = 'https://get.geojs.io/v1/ip/geo/' + ipadd + '.json'
geo_requests = requests.get(url)
geo_data = geo_requests.json()
city = geo_data['city']
state = geo_data['region']
country = geo_data['country']
speak(
f" we are in state {state} of {city} city of {country} country")
except Exception:
speak('unable to find location ')
pass
elif "weather" in self.query:
key = "4265b08b2aae7efa2e1c4456706c258d"
weather_url = "http://api.openweathermap.org/data/2.5/weather?"
ind = self.query.split().index("in")
location = self.query.split()[ind + 1:]
location = "".join(location)
url = weather_url + "appid=" + key + "&q=" + location

```

```

temperature = weather["temp"]
temperature = temperature - 273.15
humidity = weather["humidity"]
weather_response = " The temperature in Celcius is " + \
str(temperature) + " The humidity is " + str(humidity)
speak(weather_response)
else:
speak("City Not Found")
elif "where is" in self.query:
ind = self.query.lower().split().index("is")
location = self.query.split()[ind + 1:]
url = "https://www.google.com/maps/place/" + "".join(location)
speak("This is where " + str(location[0]) + " is.")
webbrowser.open(url)
# elif "activate how to do" in self.query:
# from pywikihow import search_wikihow
# speak("activated how to do mod")
# while True:
# speak("tell me what you want to know")
# how = self.takecommand()
# try:
# if "exit" in how or "no thanks" in how:
# speak("ok how to do mode is closed")
# break
# else:
# m = 1
# h = search_wikihow(how, m)
# assert len(h) == 1
# h[0].print()
# speak(h[0].summary)
# except Exception:
# speak("sorry something went wrong")
# break
elif "laptop percentage" in self.query:
battery = psutil.sensors_battery()
per = battery.percent
speak(str(per)+" percent")
elif "internet speed" in self.query:
st = speedtest.Speedtest()
dl = st.download()
ul = st.upload()
speak("download speed is: "+str(dl) +

```



```

" upload speed is " + str(ul))
else:
    print("\n")
startExecution = MainThread()
class Main(QMainWindow):
    def __init__(self):
        super().__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.pushButton.clicked.connect(self.startTask)
        self.ui.pushButton_2.clicked.connect(self.close)
    def startTask(self):
        self.ui.movie = QtGui.QMovie(
            "../Downloads/Iron man wallpaper - Imgur.gif")
        self.ui.label.setMovie(self.ui.movie)
        self.ui.movie.start()
        self.ui.movie = QtGui.QMovie(
            "../Downloads/Jarvis Loading Screen on Make a GIF.gif")
        self.ui.label_2.setMovie(self.ui.movie)
        self.ui.movie.start()
        timer = QTimer(self)
        timer.timeout.connect(self.showTime)
        timer.start(1000)
        startExecution.start()
    def showTime(self):
        current_time = QTime.currentTime()
        current_date = QDate.currentDate()
        label_time = current_time.toString('hh:mm:ss')
        label_date = current_date.toString(Qt.ISODate)
        self.ui.textBrowser.setText(label_date)
        self.ui.textBrowser_2.setText(label_time)
app = QApplication(sys.argv)
stark = Main()
stark.show()
exit(app.exec_())

```

## **5.RESULTS**

## 5.RESULT

Figure 5.1 Input for Google search

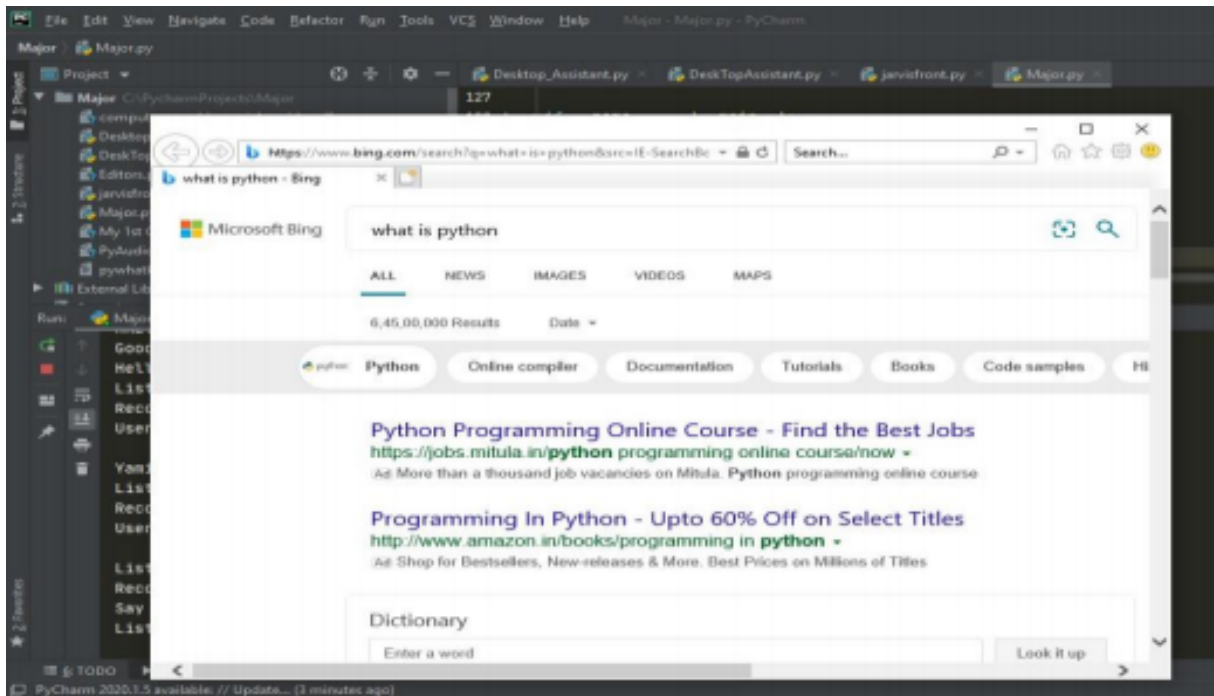
```
C:\Python36\python.exe C:/PycharmProjects/Major/Major.py
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_DAVID_11
Good Morning!
Hello!, I am Zira. Please tell me how may I help you
Listening...
Recognizing...
User said: open Google

Yamini, what should I search on google?
Listening...
Recognizing...
User said: what is python
```

In the above FIGURE 5.1 The input is to open the Google search is given to ZIRA and the assistant is responded as what should I search on google?.

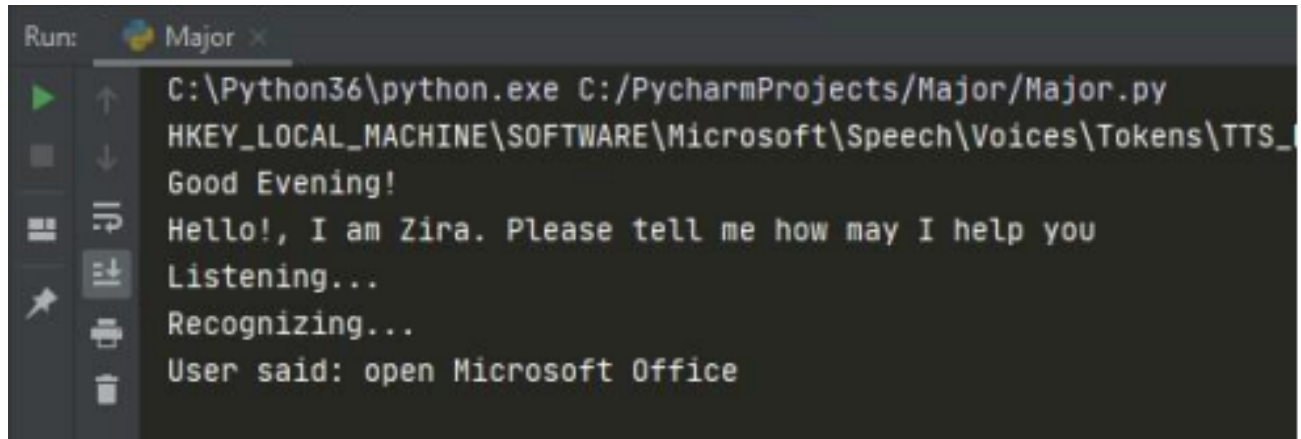
And the user asked the Assistant what is python?

Figure 5.2 Output for Google search



In the above FIGURE 5.2 The output of the google search for what is python?  
and the ZIRA gave the output of the google search.

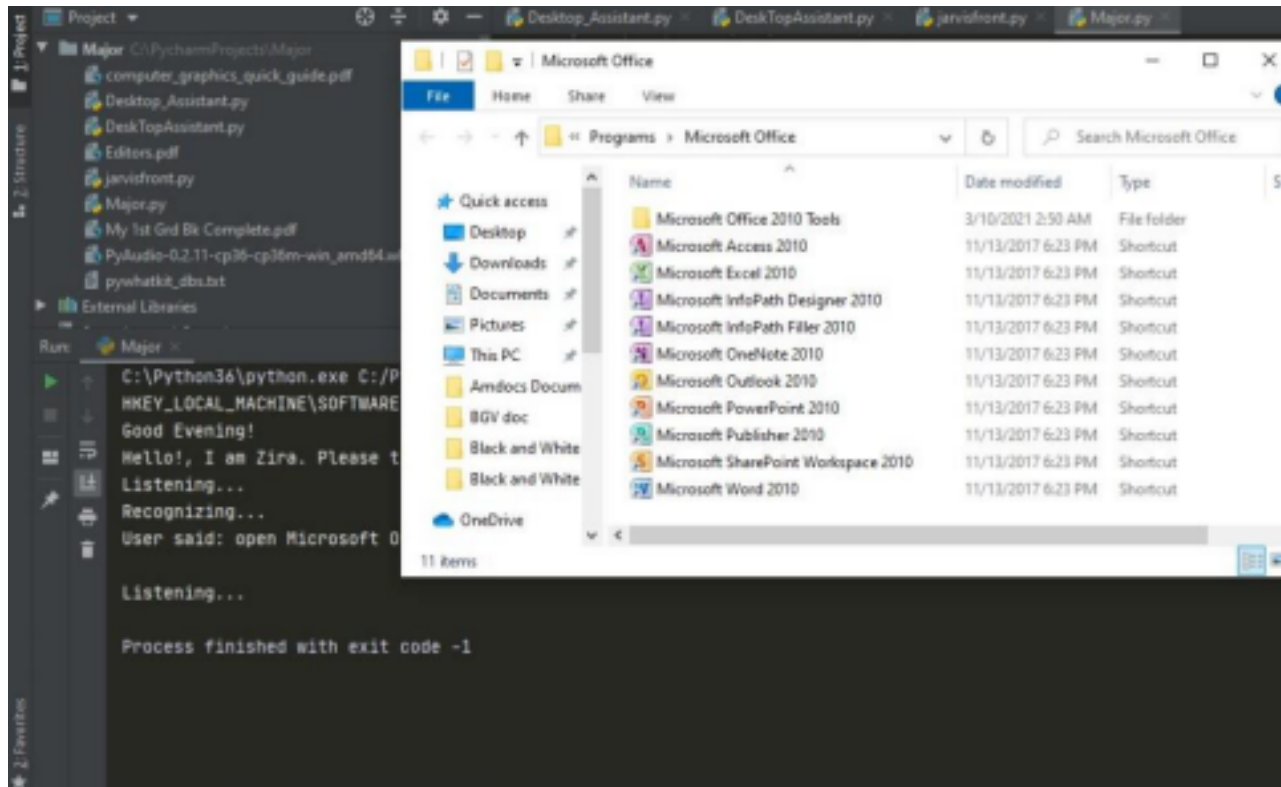
Figure 5.3 Input to open Microsoft Office

A screenshot of a Python console window titled 'Run: Major'. The console shows the execution of a Python script. The output text is as follows:

```
C:\Python36\python.exe C:/PycharmProjects/Major/Major.py
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_
Good Evening!
Hello!, I am Zira. Please tell me how may I help you
Listening...
Recognizing...
User said: open Microsoft Office
```

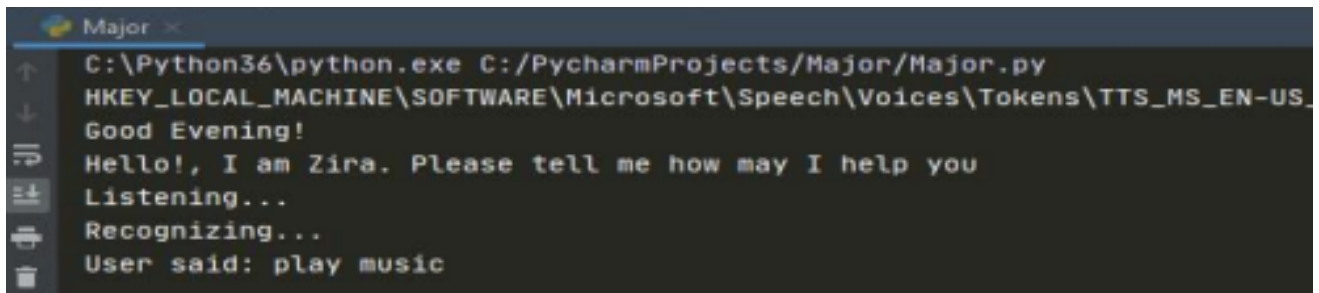
In the above FIGURE 5.3 The Input o the assistant to open the Microsoft Office

Figure 5.4 Output to open Microsoft Office



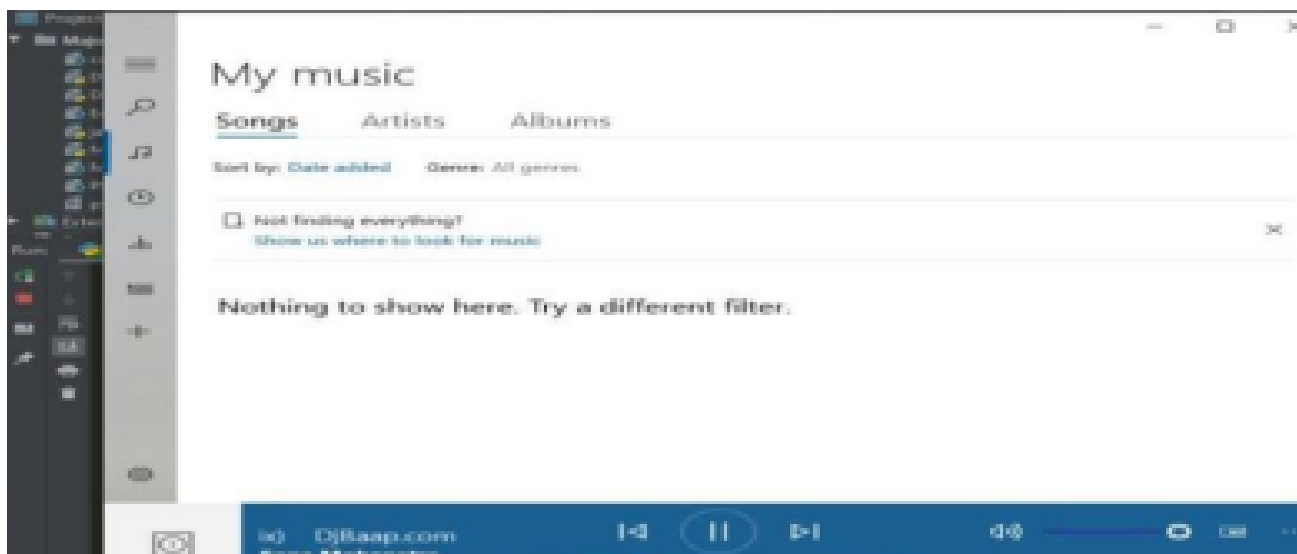
In the above FIGURE 5.4 The Input to open the Microsoft Office is done and shows the Output of the process.

Figure 5.5 Input to play music



```
Major x
C:\Python36\python.exe C:/PycharmProjects/Major/Major.py
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS_MS_EN-US_
Good Evening!
Hello!, I am Zira. Please tell me how may I help you
Listening...
Recognizing...
User said: play music
```

Figure 5.6 Output to play music





## **6.TESTING**

## **6.TESTING**

### **6.1 INTRODUCTION TO TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

### **6.2 TYPES OF TESTING**

#### **6.2.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### **6.2.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

## 6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input** : Identified classes of valid input must be accepted.
- Invalid Input** : Identified classes of invalid input must be rejected.
- Functions** : Identified functions must be exercised.
- Output** : Identified classes of application outputs Must be exercised.

Systems/Procedures:interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

## 6.3 TEST CASES

### 6.3.1 • Test Case 1

**Test Title** :Response Time

**Test ID** :T1

**Test Priority** :High

**Test Objective** :To make sure that the system respond back time is efficient.

**Description** :A virtual assistant system is mainly used to get precise answers to any question asked. Getting answer in a moment is of no use if the answer is not correct. Accuracy is of utmost importance in a virtual assistant system.

### 6.3.2 Test Case 2

**Test Title** :Accuracy

**Test ID** :T2

**Test Priority** :High

**Test Objective** :To assure that answers retrieved by system are accurate as per gathered data.

**Description** :A virtual assistant system is mainly used to get precise answers to any question asked. Getting answer in a moment is of no use if the answer is not correct. Accuracy is of utmost importance in a virtual assistant system.

### 6.3.3 Test Case 3

**Test Title:** Approximation

**Test ID:** T3

**Test priority:** Moderate

**Test Objective:** To check approximate answers about calculations.

**Description:** There are times when mathematical calculation requires approximate value. For example, if someone asks for value of PI the system must respond with approximate value and not the accurate value. Getting exact value in such cases is undesirable.

## **7.CONCLUSION**

## **7.CONCLUSION & FUTURE SCOPE**

### **7.1 PROJECT CONCLUSION**

The desktop assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, i.e. human understandable language, English. So user finds its reaction in an informed and smart way. The main application of it can be its multitasking ability. It can ask for continuous instruction one after other until the user “QUIT” it. It asks for the instruction and listen the response that is given by user without needing any trigger phase and then only executes the task.

### **7.2 FUTURE SCOPE**

- 7.2.1. Make Stark to learn more on its own and develop a new skill in it.
- 7.2.2. JARVIS android app can also be developed.
- 7.2.3. Make more Jarvis voice terminals.
- 7.2.4. Voice commands can be encrypted to maintain security

## **8. BIBLIOGRAPHY**

### **8.1 REFERENCES**

#### **Websites referred**

[www.stackoverflow.com](http://www.stackoverflow.com)  
[www.pythonprogramming.net](http://www.pythonprogramming.net)  
[www.codecademy.com](http://www.codecademy.com)  
[www.tutorialspoint.com](http://www.tutorialspoint.com)  
[www.google.co.in](http://www.google.co.in)

#### **Books referred**

Python Programming - Kiran Gurbani  
Learning Python - Mark Lutz

#### **YouTube Channels referred**

CS Dojo  
edureka!

#### **Documents referred**

Designing Personal Assistant Software for Task Management using Semantic  
Web Technologies and Knowledge Databases  
- Purushotham Botla  
Python code for Artificial Intelligence: Foundations of Computational Agents  
- David L. Poole and Alan K. Mackworth

### **8.2 GITHUB LINK**