

Python Virtual Assistant

A MINI-PROJECT REPORT

Submitted by

ASIBUL HASAN - 410619104004

MD HASAN RAJA - 410619104011

MD MUZAKKIR HOSSEN KHAN - 410619104014

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

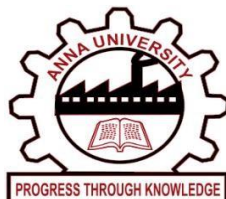


DHAANISH AHMED COLLEGE OF ENGINEERING,

PADAPPAI, CHENNAI - 601 301.

ANNA UNIVERSITY: CHENNAI 600 025.

MAY 2022



BONAFIDE CERTIFICATE

Certified that this mini-project report “**Python Virtual Assistant**” is the bonafide work of **ASIBUL HASAN(410619104004)**, **MD HASAN RAJA (410619104011)**, **MD MUZAKKIR HOSSEN KHAN(410619104014)** who carried out the project work under my supervision.

SIGNATURE

Mr P. VELAVAN

Associate Professor

HEAD OF THE DEPARTMENT

SIGNATURE

Mr S.MANIMARAN

Assistant Professor

SUPERVISOR

Computer Science and Engineering,
Dhaanish Ahmed College of Engineering,
Padappai, Chennai – 601 301

Computer Science and Engineering,
Dhaanish Ahmed College of Engineering,
Padappai, Chennai – 601 301.

Submitted for the Anna University Project viva voce held on _____

**INTERNAL EXAMINER
ACKNOWLEDGEMENT**

EXTERNAL EXAMINER

We thank our almighty God and our beloved parents for their consistent support and encouragement provided from the beginning of our project work and till its completion.

We express our heart-felt gratitude to our Chairman **ALHAJ K.MOOSA**, and to our Secretary **Mr. M. KADARSHAH,B.A,M.B.A**, for providing necessary facilities for the successful completion of our project.

We take the privilege to express our indebted sense of gratitude to our beloved Principal **Dr. G.UMA GOWRI, B.E,M.Tech. Ph.D,MISTE, MIETE** and our Director **Dr.P.PARAMASIVAN,Ph.D,MPhil,MSC**, Dhaanish Ahmed College of Engineering for granting permission to undertake the project in our college.

We would express our sincere thanks to our Head of the Department Prof **P.VELAVAN,M.E,[Ph.D]** Associate Professor for his kind permission to carry out the project and our Project Coordinator **Mr.G.RAJASEKARAN,M.E** Associate Professor, for the encouragement given to complete the project.

We would like to extend our thanks to our Project Supervisor Mr. **S.MANIMARAN,M.E**, Assistant Professor of the Department, for his excellent guidance to complete the project.

We are extremely thankful to our faculty Members of the Department of Computer Science and Engineering, for their valuable support throughout this project. We also thank our Family and Friends for their moral.

Finally we thank one and all those who have rendered help directly or indirectly at various stages of the project

ABSTRACT

In modern times, everyday life has become smarter and more sophisticated. We already know some voice services like google, and Siri. etc. Now in our voice support system, it can work like automatic chrome, open a social media website in a web browser, tell you the time and you can ask him to tell you from Wikipedia, etc. This project works by entering voice and rendering voice output and displaying text on the screen. Our main voice help agenda makes people smarter and deliver faster results with a computer. Voice Help captures voice input with our microphone and transforms our voice into understandable computer language providing the necessary solutions and answers that the user asks. This service is linked to the World Wide Web to provide the results the user has requested. The Natural Language Processing algorithm enables computer systems to engage in communication using the natural human language in many ways.

| SI No. | Title | Page No. |
|---------------|-----------------------------------|-----------------|
| 1. | Introduction | 6 |
| 1.1 | Background | 7 |
| 1.2 | Objectives | 9 |
| 1.3 | Purpose, Scope and Applicability | 10 |
| 2. | Survey of Technology | 11 |
| 3. | Requirement and Analysis | 13 |
| 3.1 | Problem Definition | 13 |
| 3.2 | Requirement Specification | 14 |
| 3.3 | Software and Hardware Requirement | 15 |
| 4. | Existing And Proposed System | 16 |
| 5. | Conclusion And Future Scope | 18 |
| 6. | System Design | 19 |
| 7. | Coding/Screenshot | 27 |
| 8. | Reference | 38 |

Chapter 1

INTRODUCTION

In today's era almost all tasks are digitalized. We have Smartphone in hands and it is nothing less than having world at your fingertips. These days we aren't even using fingers. We just speak of the task and it is done. There exist systems where we can say Text Dad, "I'll be late today." And the text is sent. That is the task of a Virtual Assistant. It also supports specialized task such as booking a flight, or finding cheapest book online from various ecommerce sites and then providing an interface to book an order are helping automate search, discovery and online order operations.

Virtual Assistants are software programs that help you ease your day-to-day tasks, such as showing weather report, creating reminders, making shopping lists etc. They can take commands via text (online chat bots) or by voice. Voice based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. For my project the wake word is JIA. We have so many virtual assistants, such as Apple's Siri, Amazon's Alexa and Microsoft's Cortana.

Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 50% of searches will be via voice by 2020.

Virtual assistants are turning out to be smarter than ever. Allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver personalized responses. This project was started on the premise that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

1.1 BACKGROUND

SIRI

There already exist a number of desktop virtual assistants. A few examples of current virtual assistants available in market are discussed in this section along with the tasks they can provide and their drawbacks.

SIRI from Apple. SIRI is personal assistant software that interfaces with the user thru voice interface, recognizes commands and acts on them. It learns to adapt to user's speech and thus improves voice recognition over time. It also tries to converse with the user when it does not identify the user request.

It integrates with calendar, contacts and music library applications on the device and also integrates with GPS and camera on the device. It uses location, temporal, social and task-based contexts, to personalize the agent behavior specifically to the user at a given point of time.

Supported Tasks

- Call someone from my contacts list
- Launch an application on my iPhone
- Send a text message to someone
- Set up a meeting on my calendar for 9am tomorrow
- Play a specific song in my iTunes library
- Enter a new note

Drawback

SIRI does not maintain a knowledge database of its own and its understanding comes from the information captured in domain models and data models.

ReQall

ReQall is personal assistant software that runs on smartphones running Apple iOS or Google Android operating system. It helps user to recall notes as well as tasks within a location and time context. It records user inputs and converts them into commands, and monitors current stack of user tasks to proactively suggest actions while considering any changes in the environment. It also presents information based on the context of the user, as well as filter information to the user based on its learned understanding of the priority of that information.

Supported Tasks

- Reminders
- Email
- Calendar, Google Calendar
- Outlook
- Evernote
- Facebook, LinkedIn
- News Feeds

Drawback

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.

1.2 OBJECTIVES

Main objective of building personal assistant software (a virtual assistant) is using semantic data sources available on the web, user generated content and providing knowledge from knowledge databases. The main purpose of an intelligent virtual assistant is to answer questions that users may have. This may be done in a business environment, for example, on the business website, with a chat interface.

On the mobile platform, the intelligent virtual assistant is available as a call-button operated service where a voice asks the user “What can I do for you?” and then responds to verbal input. Virtual assistants can tremendously save you time. We spend hours in online research and then making the report in our terms of understanding. JIA can do that for you.

Provide a topic for research and continue with your tasks while JIA does the research. Another difficult task is to remember test dates, birthdates or anniversaries. It comes with a surprise when you enter the class and realize it is class test today. Just tell JIA in advance about your tests and she reminds you well in advance so you can prepare for the test. One of the main advantages of voice searches is their rapidity.

In fact, voice is reputed to be four times faster than a written search: whereas we can write about 40 words per minute, we are capable of speaking around 150 during the same period of time¹⁵. In this respect, the ability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

1.3 PURPOSE, SCOPE AND APPLICABILITY

Purpose

Purpose of virtual assistant is to being capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Virtual assistants enable users to speak natural language voice commands in order to operate the device and its apps.

Scope

Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices. However, it's not just developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for their specific brand.

They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface.

Applicability

The mass adoption of artificial intelligence in users' everyday lives is also fueling the shift towards voice. The number of IoT devices such as smart thermostats and speakers are giving voice assistants more utility in a connected user's life. Smart speakers are the number one way we are seeing voice being usually reserved for the rich and famous. used.

The use of virtual assistants can also enhance the system of IoT (Internet of Things). Twenty years from now, Microsoft and its competitors will be offering personal digital assistants that will offer the services of a full-time employee

Chapter 2

SURVEY OF TECHNOLOGY

Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages.

Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For JIA, libraries used are speech recognition to recognize voice for text to speech, selenium for web automation etc.

Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time, and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

pyttsx

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline. **Speech Recognition** This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

Speech Recognition

This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc.

PyAudio

This library provides **Python bindings for PortAudio**, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms. PyAudio is inspired by: 1.pyPortAudio/fastaudio: Python bindings for PortAudio v18 API. 2.tkSnack: cross-platform sound toolkit for Tcl/Tk and Python.

Pyjokes

Pyjokes is a **python library** that is used to create one-line jokes for programmers. Informally, it can also be referred as a fun python library which is pretty simple to use. Let us see how you can actually use it to perform the required task.

Chapter3

REQUIREMENT AND ANALYSIS

System Analysis is about complete understanding of existing systems and finding where the existing system fails. The solution is determined to resolve issues in the proposed system. It defines the system. The system is divided into smaller parts. Their functions and inter relation of these modules are studied in system analysis. The complete analysis is followed below.

3.1 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. There are number of people who have issues in voice recognition. These systems can understand English phrases but they fail to recognize in our accent. Our way of pronunciation is way distinct from theirs. Also, they are easy to use on mobile devices than desktop systems. There is need of a virtual assistant that can understand English in Indian accent and work on desktop system.

When a virtual assistant is not able to answer questions accurately, it's because it lacks the proper context or doesn't understand the intent of the question. Its

ability to answer questions relevantly only happens with rigorous optimization, involving both humans and machine learning. Continuously ensuring solid quality control strategies will also help manage the risk of the virtual assistant learning undesired bad behaviors.

They require large amount of information to be fed in order for it to work efficiently. Virtual assistant should be able to model complex task dependencies and use these models to recommend optimized plans for the user. It needs to be tested for finding optimum paths when a task has multiple sub-tasks and each sub-task can have its own sub-tasks. In such a case there can be multiple solutions to paths, and the it should be able to consider user preferences, other active tasks, priorities in order to recommend a particular plan

3.2 REQUIREMENT SPECIFICATION

Personal assistant software is required to act as an interface into the digital world by understanding user requests or commands and then translating into actions or recommendations based on agent's understanding of the world. JIA focuses on relieving the user of entering text input and using voice as primary means of user input. Agent then applies voice recognition algorithms to this input and records the input.

It then use this input to call one of the personal information management applications such as task list or calendar to record a new entry or to search about it on search engines like Google, Bing or Yahoo etc. Focus is on capturing the user input through voice, recognizing the input and then executing the tasks if the agent understands the task. Software takes this input in natural language, and so makes it easier for the user to input what he or she desires to be done.

Voice recognition software enables hands free use of the applications, lets users to query or command the agent through voice interface. This helps users to have access to the agent while performing other tasks and thus enhances value of the system itself. JIA also have ubiquitous connectivity through Wi-Fi or LAN

connection, enabling distributed applications that can leverage other APIs exposed on the web without a need to store them locally.

Virtual assistants must provide a wide variety of services. These include:

- Providing information such as news, facts from e.g. Wikipedia etc.
- Providing date and time.
- We can use it like chatbot.
- Providing google search and playing video in youtube.
- Providing user location.
- Providing entertainment like jokes.

3.3 HARDWARE AND SOFTWARE 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.**

Software:

- **Windows 7(32-bit) or above.**
- **Python 2.7 or late**
- **vs code 1.52 or above**

Chapter4

EXISTING AND PROPOSED SYSTEM

Existing System

Most existing projects use only speech recognition using emotional networks. Although their systems are relatively accurate, they are not for real use and are not suitable for any real use. There are a few basic methods they use:

1.Context-aware computing: Context-aware computing is a category of programs that can sense their physical location and adapt accordingly. These can be used to identify words spoken by people with different characteristics. It may also spell out words that may have been mispronounced.

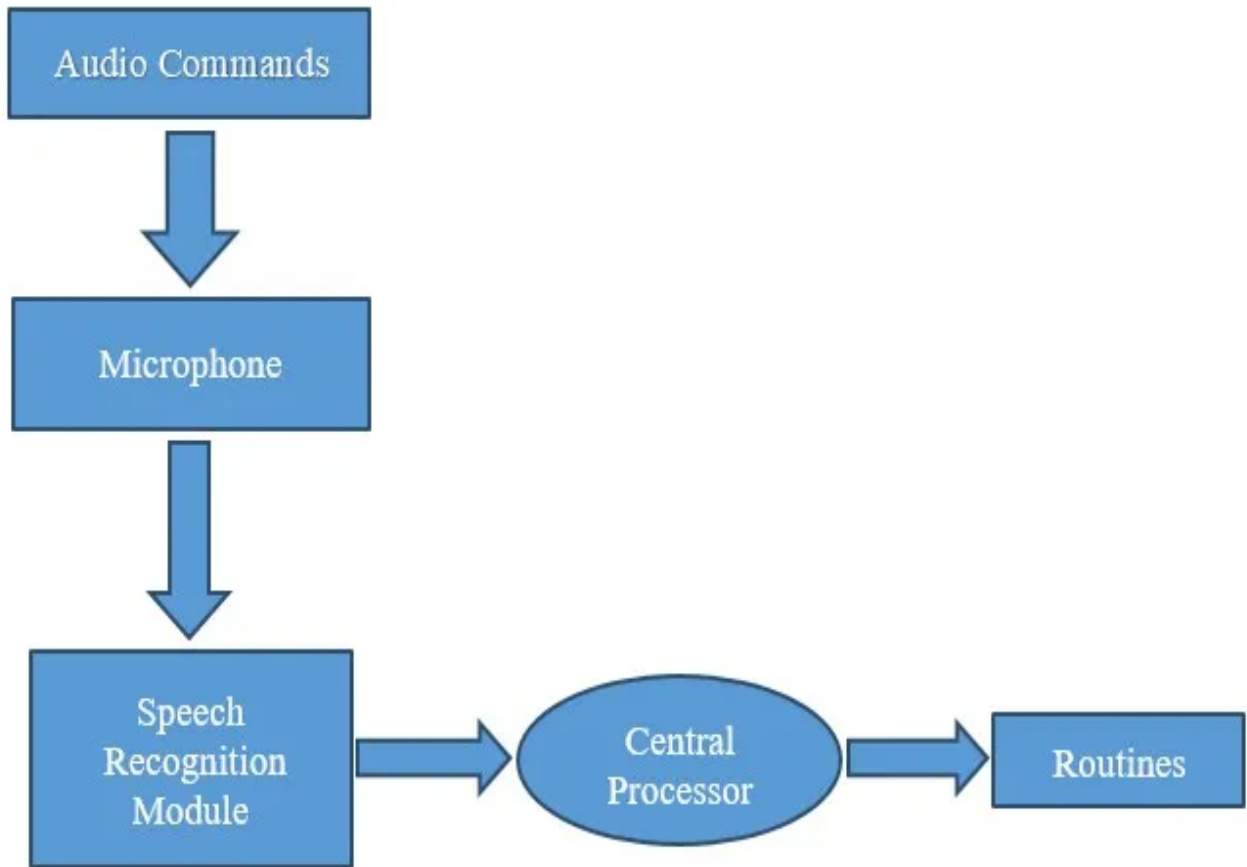
2.MFCC: MFCC refers to MelFrequency Cepstral Coefficients. MFC (Melfrequency Cepstrum) is a collection of this coefficient. It is equal to the short-term energy spectrum of sound. These can be used to hear sound variations to detect the various variables needed for voice recognition.

3.NLP: Natural Language Programming is a branch of Artificial Intelligence responsible for computer interactions and human languages. It focuses on programming computers so that they can process large amounts of data in native languages. This concept is used to familiarize a computer with a variety of words in a given language and to recognize them when spoken.

Proposed System

The work started with analyzing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer's internal files, etc. This is an empirical qualitative study, based on reading above mentioned literature and testing their examples.

Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Voice Assistant.



Chapter5

CONCLUSION AND FUTURE SCOPE

CONCLUSION

Virtual Personal Assistants are a very effective way to organize your program. There are now many Smart Personal Digital Assistant apps available on the market for various device platforms. These new Software apps work much better than PDA devices as they provide all the features of your smartphone.

VPAs are also more reliable than Personal Assistants because VPAs are portable and you can use them at any time. And they have more information than any assistant as they are connected to the internet.

FUTURE SCOPE

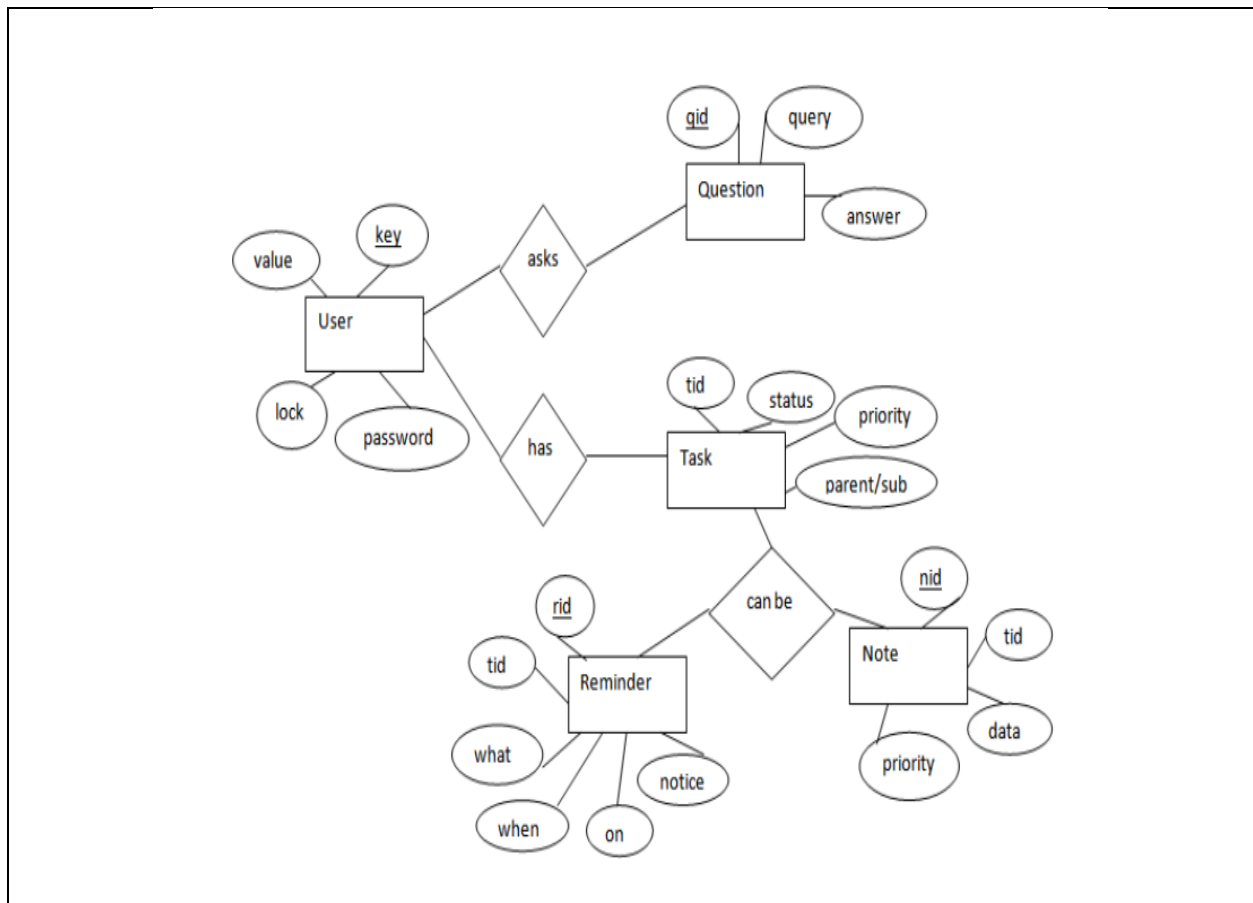
The next step will be to remove as much hardware as possible. With the ingenuity of the VA present in the clouds, being pulled in, and pushing its way into our lives with many devices in our bodies and our offices, homes, and cars.

Your VA will always tell you about suggestions and take orders, and you will know more about yourself than you know yourself. We can expect this device to be installed and permanent.

Chapter6

SYSTEM DESIGN

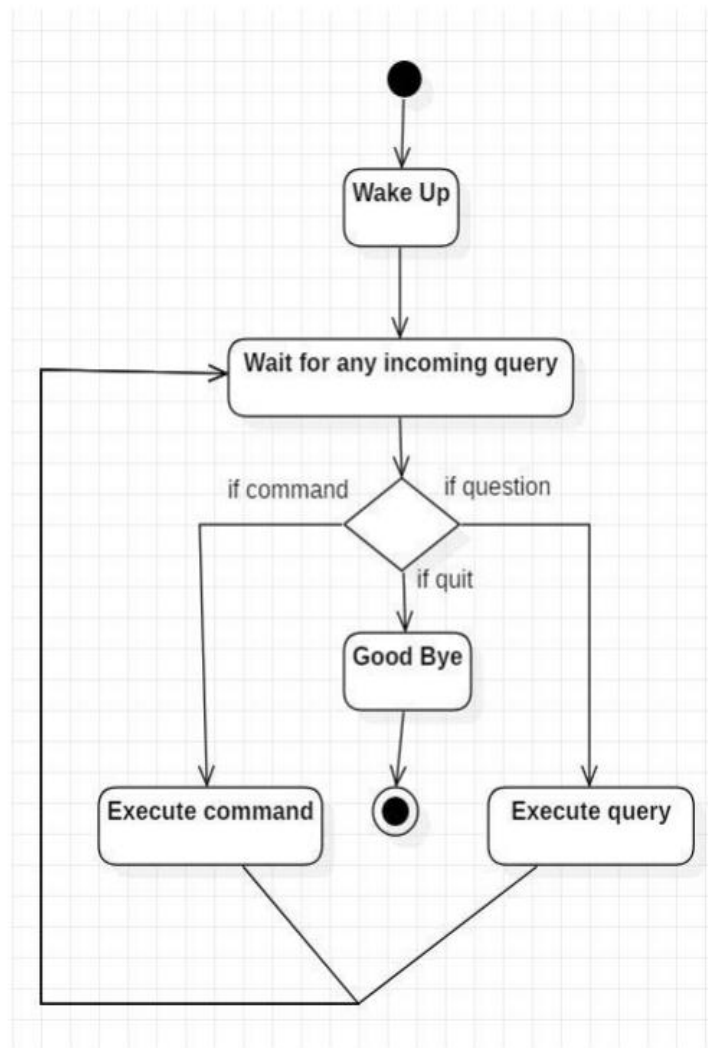
1.ER DIAGRAM



The above diagram shows entities and their relationship for a virtual assistant system. We have a user of a system who can have their keys and values. It can be used to store any information about the user. Say, for key “name” value can be “Jim”. For some keys user might like to keep secure. There he can enable lock and set a password (voice clip). Single user can ask multiple questions. Each question will be given ID to get recognized along with the query and its corresponding

answer. User can also be having n number of tasks. These should have their own unique id and status i.e. their current state.

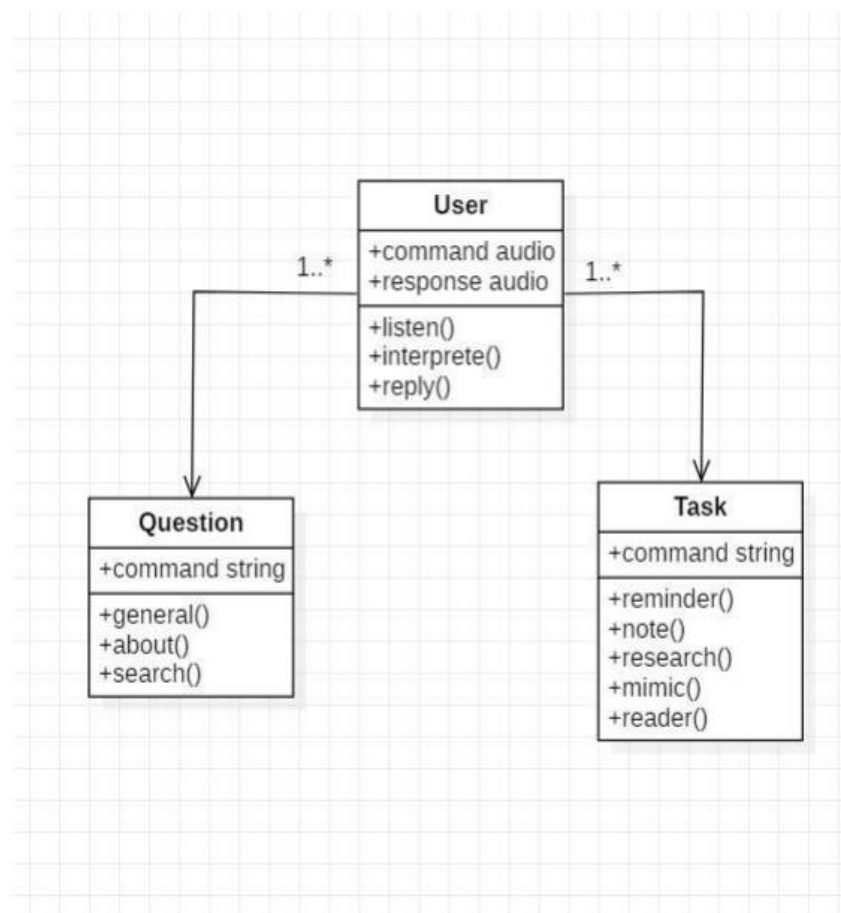
2. ACTIVITY DIAGRAM



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.

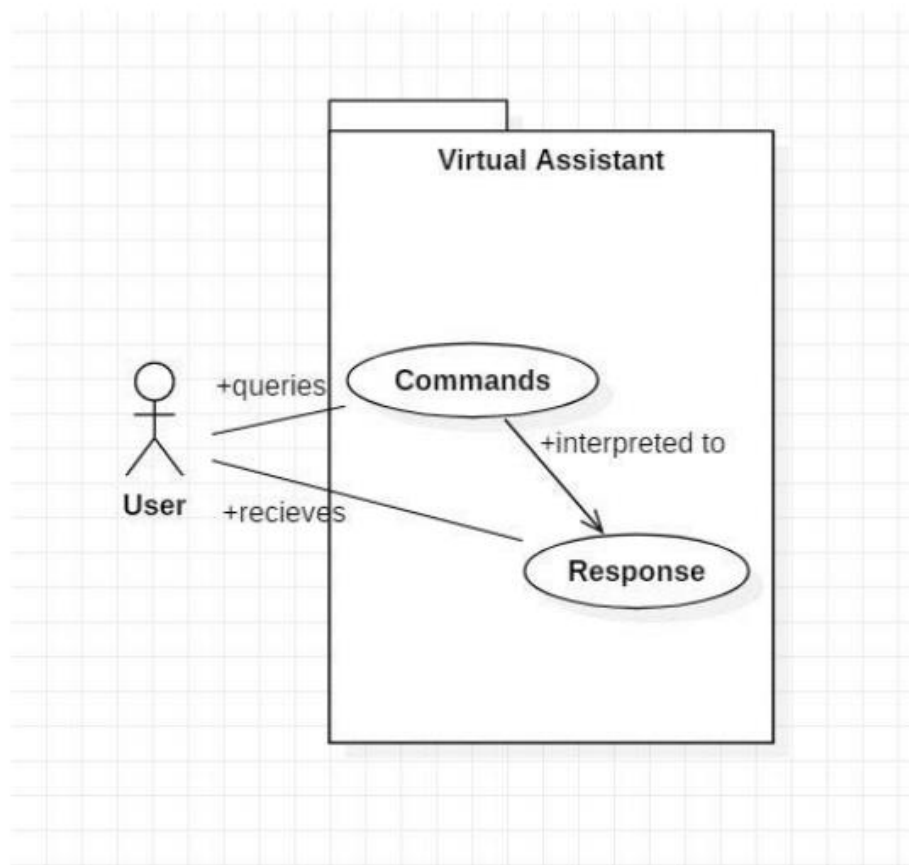
3. CLASS DIAGRAM



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 form as it is interpreted by interpret class. It sends it to

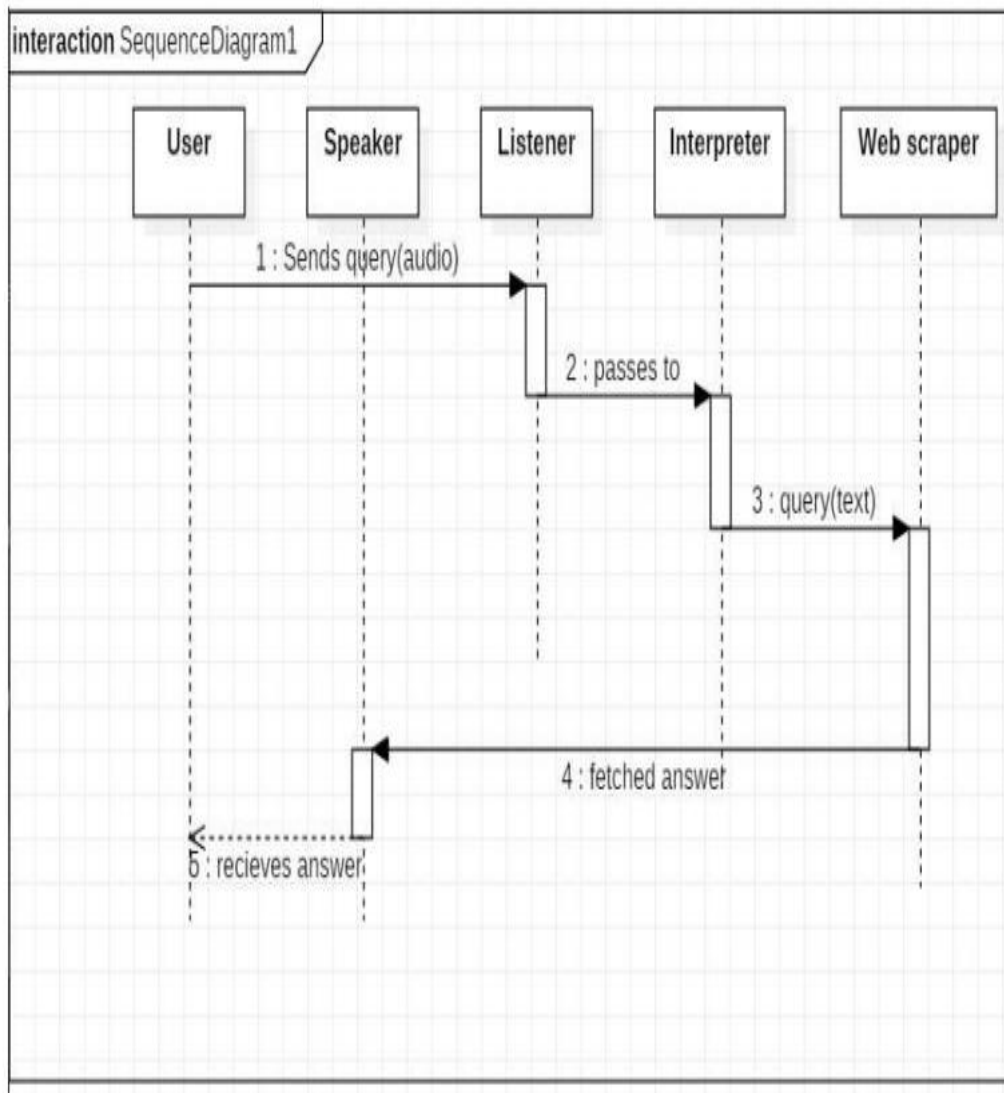
general or about or search function based on its identification. The task class also has interpreted command in string format. It has various functions like reminder, note, mimic, research and reader.

4. USE CASE DIAGRAM



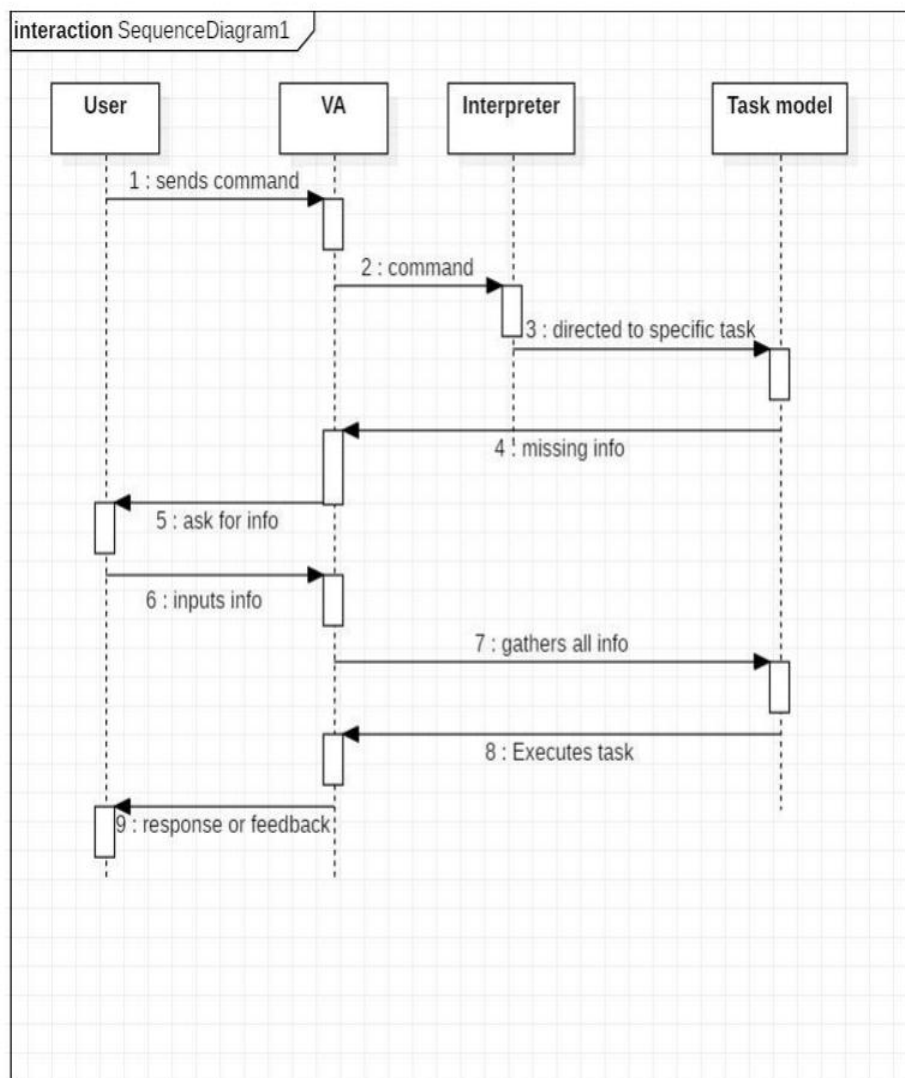
In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user.

5. SEQUENCE DIAGRAM FOR QUERY-RESPONSE



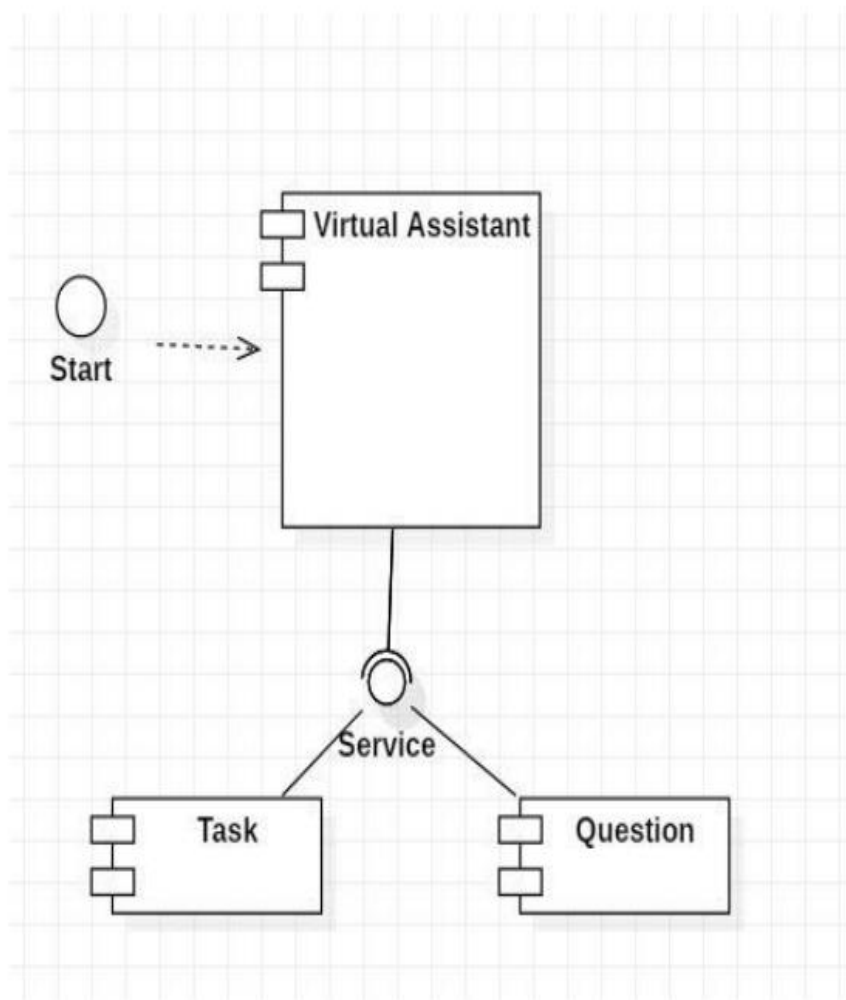
The above sequence diagram shows how an answer asked by the user is being fetched from internet. The audio query is interpreted and sent to Web scraper. The web scraper searches and finds the answer. It is then sent back to speaker, where it speaks the answer to user.

6. SEQUENCE DIAGRAM FOR TASK EXECUTION



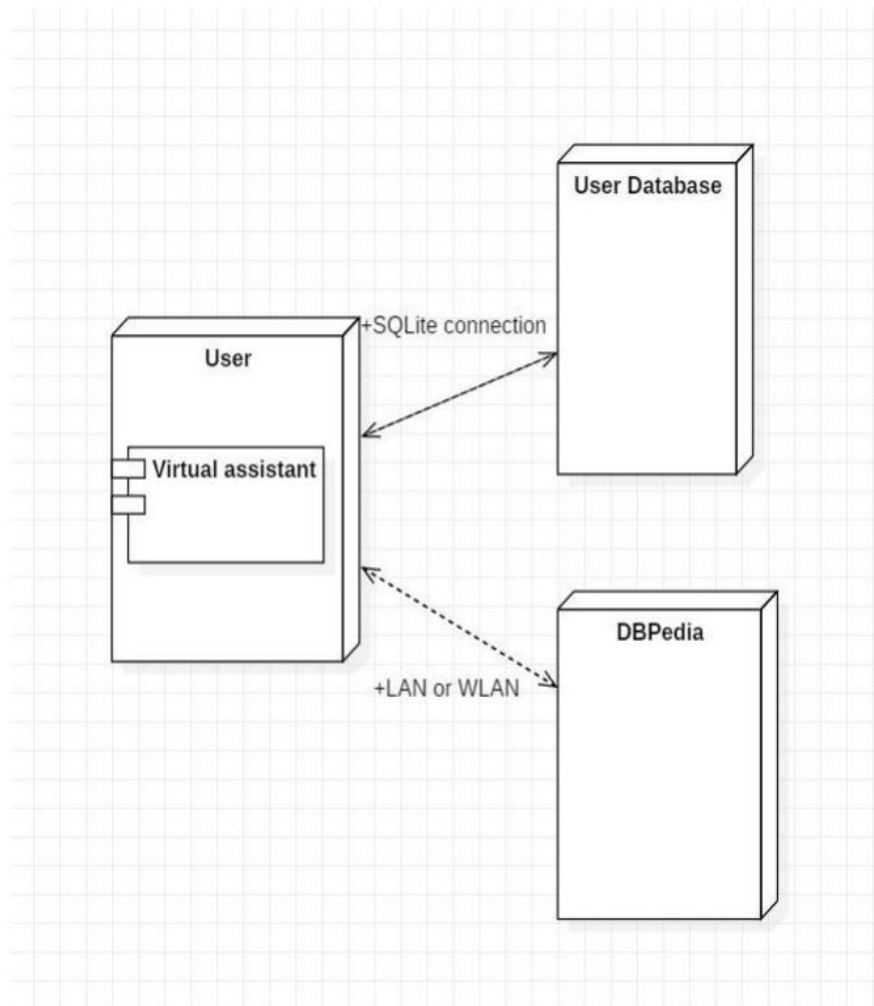
The user sends command to virtual assistant in audio form. The command is passed to the interpreter. It identifies what the user has asked and directs it to task executer. If the task is missing some info, the virtual assistant asks user back about it. The received information is sent back to task and it is accomplished. After execution feedback is sent back to user.

7. COMPONENT DIAGRAM



The main component here is the Virtual Assistant. It provides two specific service, executing Task or Answering your question.

8. DEPLOYMENT DIAGRAM



The user interacts with SQLite database using SQLite connection in Python code. The knowledge database DBPedia must be accessed via internet connection. This requires LAN or WLAN / Ethernet network.

Chapter7

CODING/SCREENSHOT

MODULE:

import audioop

import ctypes

from distutils.cmd import Command

from opcode import hasname

import shutil

import subprocess

import time

from tkinter.colorchooser import askcolor

from tkinter.filedialog import askdirectory

from urllib.request import urlopen

from xml.dom.minidom import NamedNodeMap

import pyttsx3

import speech_recognition as sr

import datetime

```
import os

import webbrowser

import webbrowser

import pyjokes

from time import ctime

from winsound import PlaySound

import speech_recognition as sr

import webbrowser

import time

import os
```

CODE:

```
engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

engine.setProperty('voice', voices[1].id)
```

```
def there_exists(terms):

    for term in terms:

        return True
```

```

r = sr.Recognizer() # initialise a recogniser

# listen for audio and convert it to text:

def record_audio(ask=False):

    with sr.Microphone() as source:

        r.energy_threshold=500

        r.adjust_for_ambient_noise(source,1.2)

        r.pause_threshold= 1

        if askcolor:

            speak(askdirectory)

    voice_data = ''

    try :

        voice_data = r.recognize_google(audioop)

    except sr.RequestError:

        speak('Sorry, the service is down')

    except sr.UnknownValueError:

        print('Recognizing..')

```

```
print(f">> {voice_data.lower()}")
```

```
return voice_data.lower()
```

```
def speak(audio):
```

```
    engine.say(audio)
```

```
    engine.runAndWait()
```

```
def wishMe():
```

```
    hour = int(datetime.datetime.now().hour)
```

```
    if hour >= 0 and hour < 12:
```

```
        speak("Good Morning Sir !")
```

```
    elif hour >= 12 and hour < 18:
```

```
        speak("Good Afternoon Sir !")
```

```
    else:
```

```
        speak("Good Evening Sir !")
```

```
assname = ("This is Bongo")
```

```
speak("I am your Assistant")
```

```
    speak(assname)
```

```
def username():
```

```
    speak("What should i call you sir")
```

```
    uname = takeCommand()
```

```
    speak("Welcome Mister")
```

```
    speak(uname)
```

```
    columns = shutil.get_terminal_size().columns
```

```
    print("*****".center(columns))
```

```
    print("Welcome Mr.", uname.center(columns))
```

```
    print("*****".center(columns))
```

```
    speak("How can i Help you, Sir")
```

```
def takeCommand():
```

```
    r = sr.Recognizer()
```

```
    with sr.Microphone() as source:
```

```
print("Listening...")  
  
r.pause_threshold = 1  
  
audio = r.listen(source)
```

try:

```
print("Recognizing...")  
  
query = r.recognize_google(audio, language='en-in')  
  
print(f"User said: {query}\n")
```

except Exception as e:

```
print(e)  
  
print("Unable to Recognize your voice.")  
  
return "None"
```

return query

```
if __name__ == '__main__':  
  
    clear = lambda: os.system('cls')  
  
    clear()
```


wishMe()

username()

while True:

query = takeCommand().lower()

if 'lock window' in query:

speak("locking the device")

ctypes.windll.user32.LockWorkStation()

elif 'open youtube' in query:

speak("Here you go to Youtube\n")

webbrowser.open("[youtube.com](https://www.youtube.com)")

elif 'lock window' in query:

speak("locking the device")

ctypes.windll.user32.LockWorkStation()

elif 'open google' in query:

speak("Here you go to Google\n")

webbrowser.open("[google.com](https://www.google.com)")

elif 'open stackoverflow' in query:

speak("Here you go to stackoverflow")

webbrowser.open("stackoverflow.com")

elif 'open facebook' in query:

speak("Here you go to facebook")

webbrowser.open("facebook.com")

elif 'open dhaanish college' in query:

speak("Here you go to dhaanish college")

webbrowser.open("dhaanish.in")

elif 'open anna university' in query:

speak("Here you go to anna university")

webbrowser.open("<https://www.annauniv.edu>")

elif 'my location' in query:

speak("your location is")

webbrowser.open("<https://google.nl/maps/place/>")

elif 'open google' in query:

speak("Here you go to google")

webbrowser.open("<https://www.google.com/>")

elif 'shutdown system' in query:

speak("Hold On a Sec ! Your system is on its way to shut down")

subprocess.call('shutdown / p /f')

elif "don't listen" in query or "stop listening" in query:

speak("for how much time you want to stop jarvis from listening commands")

a = int(takeCommand())

time.sleep(a)

print(a)

elif 'tell me a joke' in query:

```
speak(pyjokes.get_joke())
```

```
elif "who am i" in query:
```

```
speak("If you talk then definitely your human.")
```

```
elif "why i came to world" in query:
```

```
speak("Thanks to god. further It's a secret")
```

```
elif "how are you" in query:
```

```
speak("I'm fine, nice to meet you")
```

```
elif "i love you" in query:
```

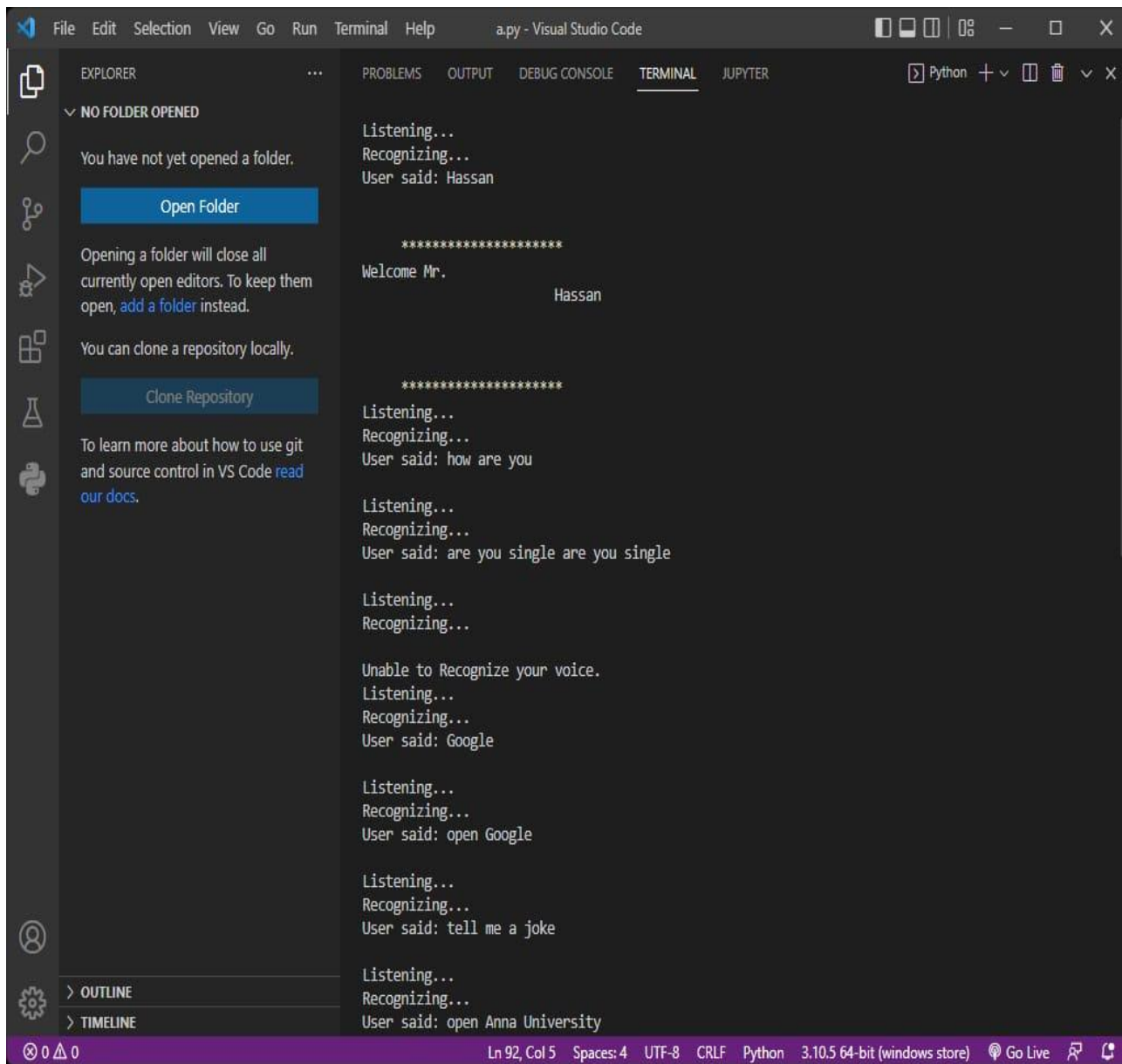
```
speak("sorry, It's hard to understand")
```

```
elif "wikipedia" in query:
```

```
webbrowser.open("wikipedia.com")
```

```
elif "bongo" in
```

SCREENSHOT:



Chapter8

REFERENCE

- **Websites referred**

❏ www.stackoverflow.com

❏ www.pythonprogramming.net

❏ www.codecademy.com

❏ www.tutorialspoint.com

❏ www.google.co.in

- **Books referred**

❏ Python Programming - Kiran Gurbani

❏ Learning Python - Mark Lutz

- **YouTube Channels referred**

❏ CS Dojo

❏ edureka!

