RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

Internship Report

on

Personal Voice Assistant

INT411: Intra Institutional Internship

TEAM MEMBERS

Name	USN
PK Muhammad Suarim	1MS21AD035
Yashraj Verma	1MS21AD062
Anuritha L	1MS21AD012

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)
MSR Nagar, MSRIT Post, Bangalore-560054
September – October, 2022

INTERNSHIP ASSESSMENT

Python Programming

INT411: Intra Institutional Internship

SL No.	Component	Maximum Marks	Marks Obtained
1	Continuous Evaluation	50	
2	Presentation	20	
3	Report	30	
Total Marks		100	

Signature of Student Coordinator

Signature of Faculty

RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that Mr./Ms. _____, a student of Bachelor of Engineering, bearing USN: _____, has successfully completed, 30 Hours: from 29.09.2022 to 15.10.2022 Intra Institutional Internship in ----------, from the Department of ----------,M S Ramaiah Institute of Technology, Bangalore.

Signature of Faculty Coordinator **Head of the Department**

TABLE OF CONTENTS

Introduction

General Introduction

Intelligent personal assistants (IPAs) such as Alexa, Siri, and Google Assistant have been becoming more and more popular and making people's daily lives convenient. IPAs can fulfill users' request by answering questions ranging from weather to stock price. To enrich the user experience, a large amount of third-party (3P) voice apps (aka skills) have been developed. These voice apps extend IPAs built-in capabilities to better serve customers. They can perform operations such as ordering food, playing a game, or helping a user sleep by playing soothing sounds. The supported 3P skills can number up to hundreds of thousands.

Intelligent personal assistants understand user's request using spoken language understanding (SLU) system. The request goes through a series of components to get a response, as illustrated in Figure 1. The first component is automatic speech recognition (ASR), which converts speech to its transcription also called utterance. At the second stage, the utterance is interpreted by the natural language understanding (NLU) system. NLU as the critical component of SLU interprets the meaning of an utterance by using several natural language processing (NLP) technologies including domain classifier (DC), intent classifier (IC), and named entity recognition (NER). The DC determines which domain should process the request. The IC predicts what the user wants to do from the list of intent types of the identified domain. NER, or slot tagging, finds the entity (i.e., person, place, or thing) in the utterance and tags it as a particular entity type (i.e., city, song). For example, given an utterance “play Million Reasons by Lady Gaga.,” the DC predicts Music as the domain. The IC predicts the intent as PlayMusic. Finally, NER identifies the slot-value pairs as SongName:Million Reasons and Artist:Lady Gaga. After NLU, the arbiter is responsible to select the most relevant voice app (skill) for a given NLU interpretation { Music, PlayMusic, SongName:Million Reasons, Artist:Lady Gaga}. Sometimes the arbiter may fail to find a relevant skill that can handle the user request. It could be a system error such as ASR error, NLU error. Another reason could be that the feature requested by the user is not supported yet by the dialog system or the requested content is not found such as music, video, book, and recipe. To reduce customer friction and recover the conversation, we propose a skill recommender system that proactively suggests 3P skills to users for unhandled requests, even if the users are not aware of the skills.

Figure 1

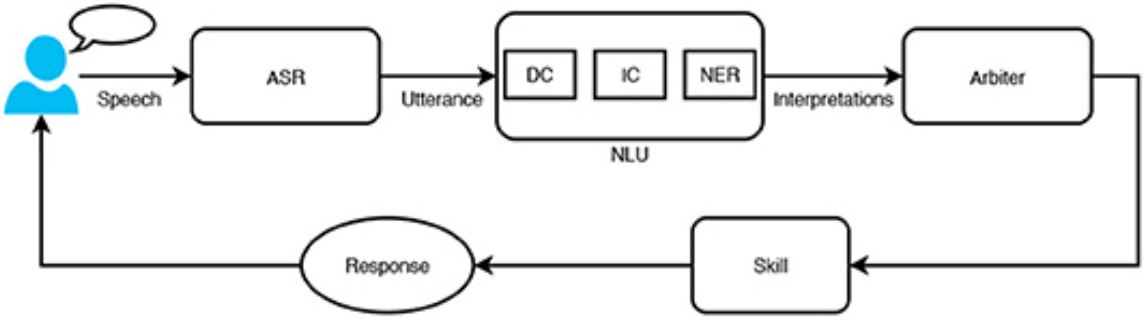


FIGURE 1. A high-level overview of an intelligent personal assistant (IPA).

In today's era almost all tasks are digitalized. We have Smartphone in hands and it is nothing less than having world at your finger tips. 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 e- commerce 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.

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.

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 behaviours. They require large amount of information to be fed in order for it to work efficiently.

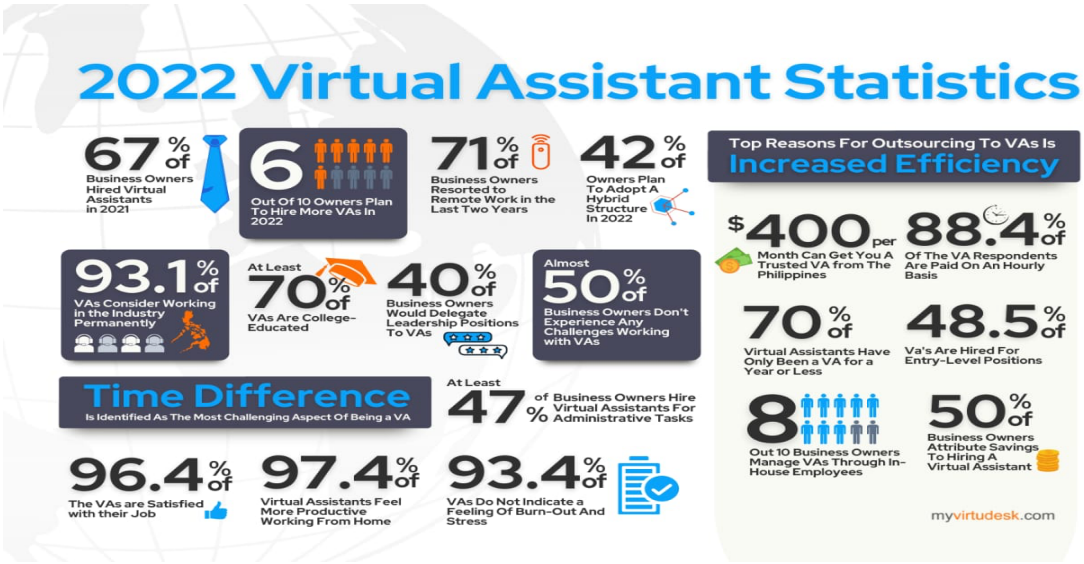
Virtual assistant should be able to model complex task dependencies and use these models to recommend optimized plans for the user. It needs to be tested for finding optimum paths when a task has multiple sub-tasks and each sub-task can have its own sub-tasks. In such a case there can be multiple solutions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

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.



PURPOSE, SCOPE AND APPILCABILITY

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.

There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized, it's clear that we are moving towards less screen interaction.

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 used. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years.

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 usually reserved for the rich and famous.

SURVEY OF TECHNOLOGY

Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It isa 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 $1/5^{th}$ 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, Pyttsx 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.

DBpedia

Knowledge bases are playing an increasingly important role in enhancing the intelligence of Web and enterprise search and in supporting information integration. The DBpedia leverages this gigantic source of knowledge by extracting structured information from Wikipedia and by making this information accessible on the Web. The DBpedia knowledge base has several advantages over existing knowledge bases: it covers many domains; it represents real community agreement; it automatically evolves as Wikipedia changes, and it is truly multilingual. The DBpedia knowledge base allows you to ask quite surprising queries against Wikipedia for instance “Give me all cities in New Jersey with more than 10,000 inhabitants” or “Give me all Italian musicians from the 18^{th} century”.

Quepy

Quepy is a python framework to transform natural language questions to queries in a database query language. It can be easily customized to different kinds of questions in natural language and database queries. So, with little coding you can build your own system for natural language access to your database.

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.

SQLite

SQLite is a capable library, providing an in-process relational database for efficient storage of small-to-medium-sized data sets. It supports most of the common features of SQL with few exceptions. Best of all, most Python users do not need to install anything to get started working with SQLite, as the standard library in most distributions ships with the sqlite3 module.

SQLite runs embedded in memory alongside your application, allowing you to easily extend SQLite with your own Python code. SQLite provides quite a few hooks, a reasonable subset of which are implemented by the standard library database driver.

Project Deliverables

- Open and close camera
- Plays Youtube videos
- Enables google search
- Gives the weather report of city
- Interactive modules making it user friendly
- Send emails
- Open Instagram login page
- Has Games to kill boredom
- Set Alarm
- Use Calculator
- Display Date and Time
- Set a timer
- Open a website

Methodology of Virtual Assistant Using Python

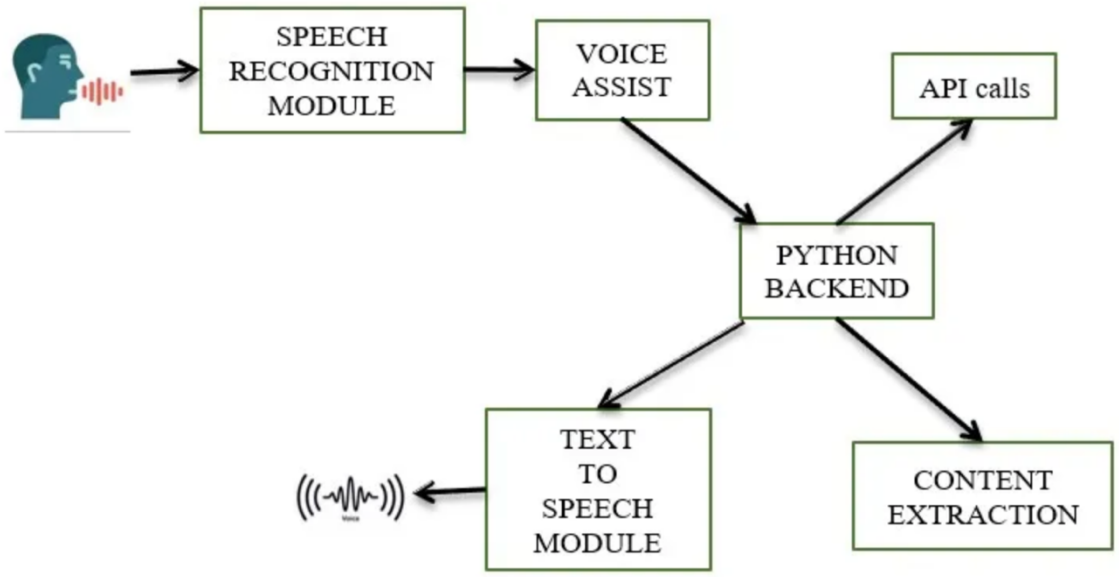


Fig 3 Detailed workflow

Speech Recognition module

The system uses Google's online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information centre from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

Python Backend:

The python backend gets the output from the speech recognition module and then identifies whether the command or the speech output is an API Call and Context Extraction. The output is then sent back to the python backend to give the required output to the user.

API calls

API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is a messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.

Content Extraction

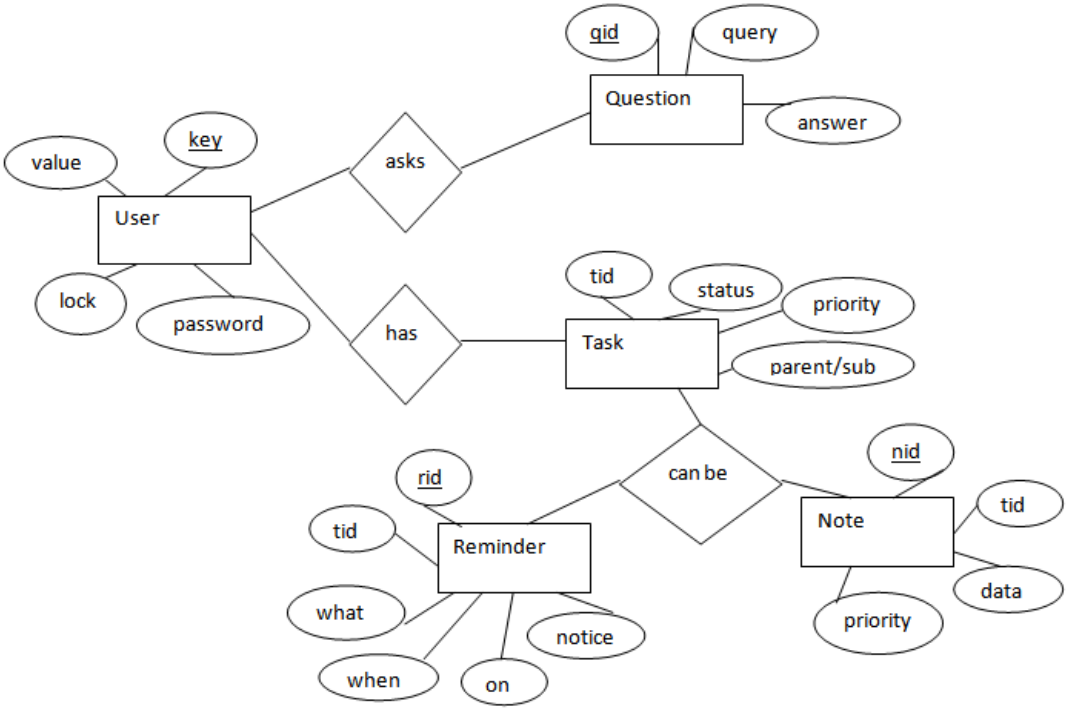
Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most cases, this activity concerns processing human language texts using natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as context extraction TEST RESULTS.

Text-to-speech module

Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

SYSTEM DESIGN

ER DIAGRAM

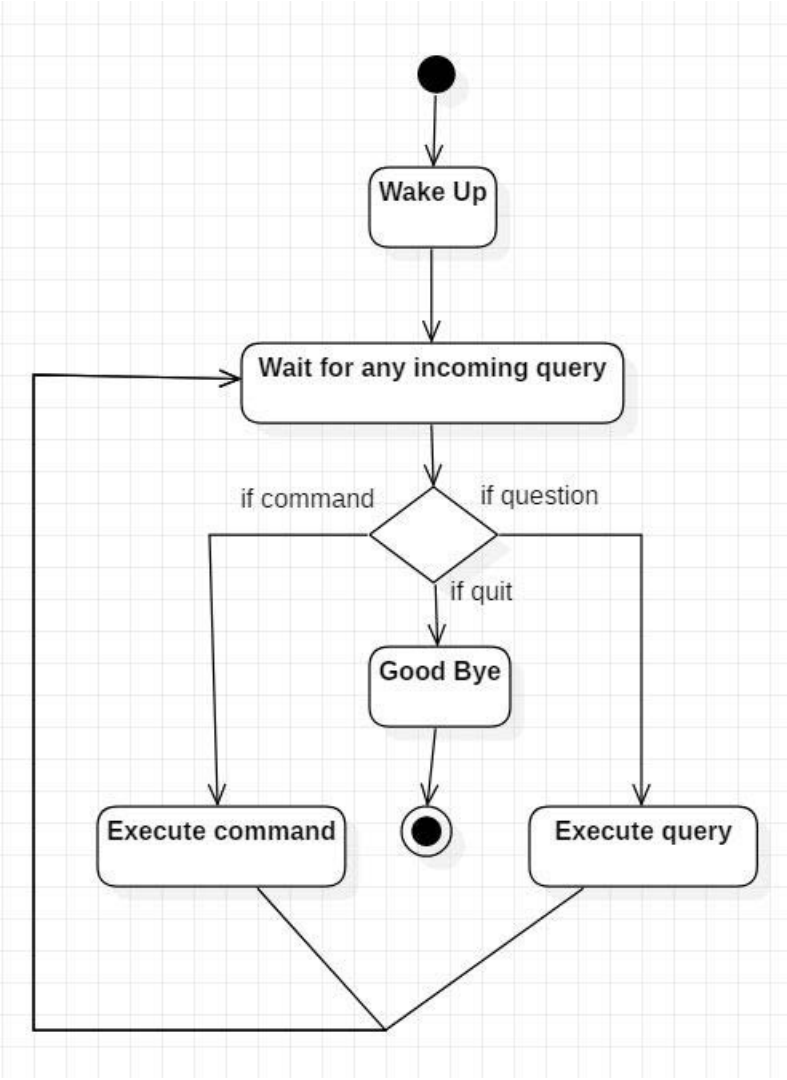


The above diagram shows entities and their relationship for a virtual assistant system. We have a user of a system who can have their keys and values. It can be used to store any information about the user. Say, for key “name” value can be “Jim”. For some keys user might like to keep secure. There he 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. A task should also have a priority value and its category whether it is a parent task or child task of an older task.

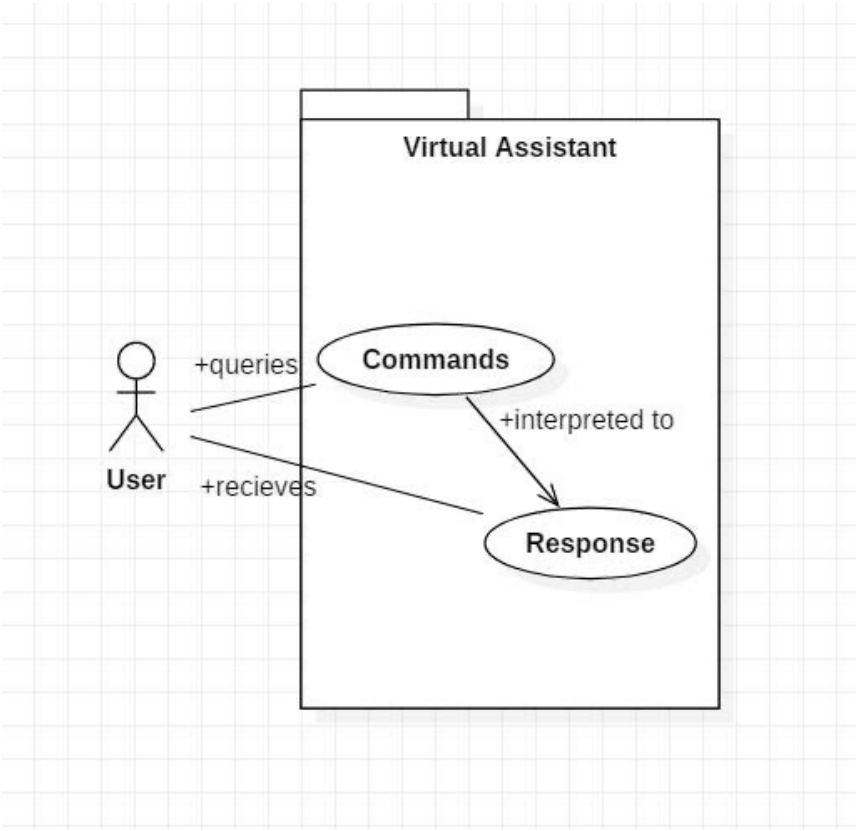
ACTIVITY DIAGRAM

Initially, the system is in idle mode. As it receives any wake up cal 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.



USE CASE DIAGRAM

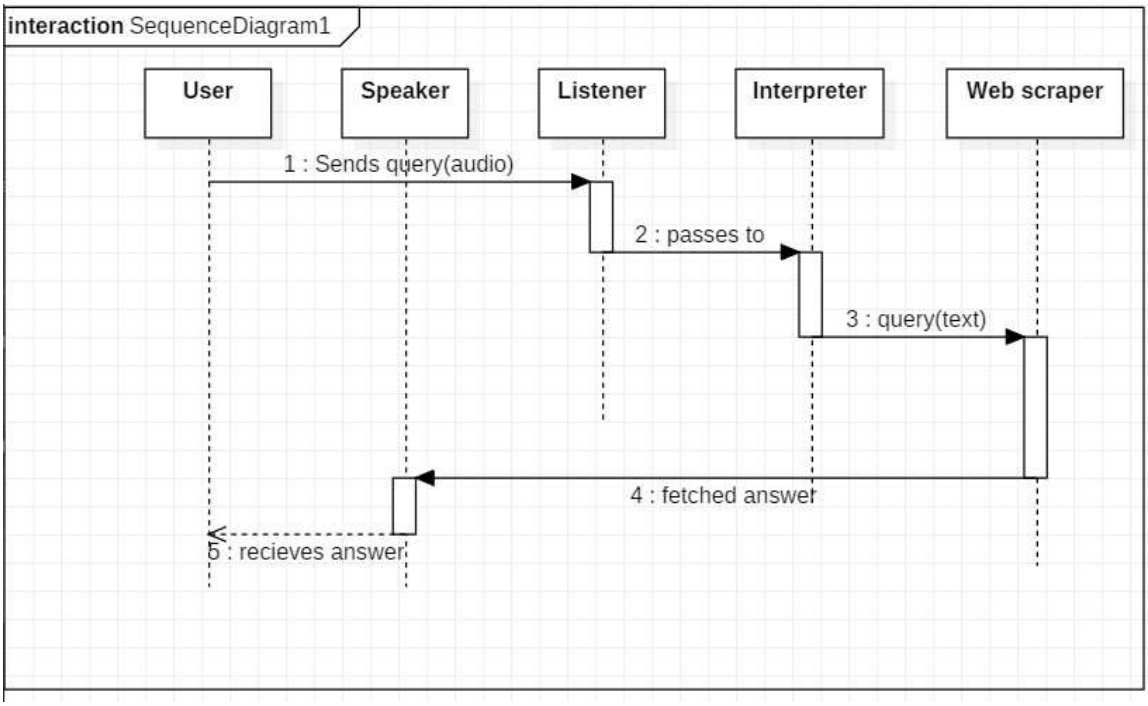
In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user.



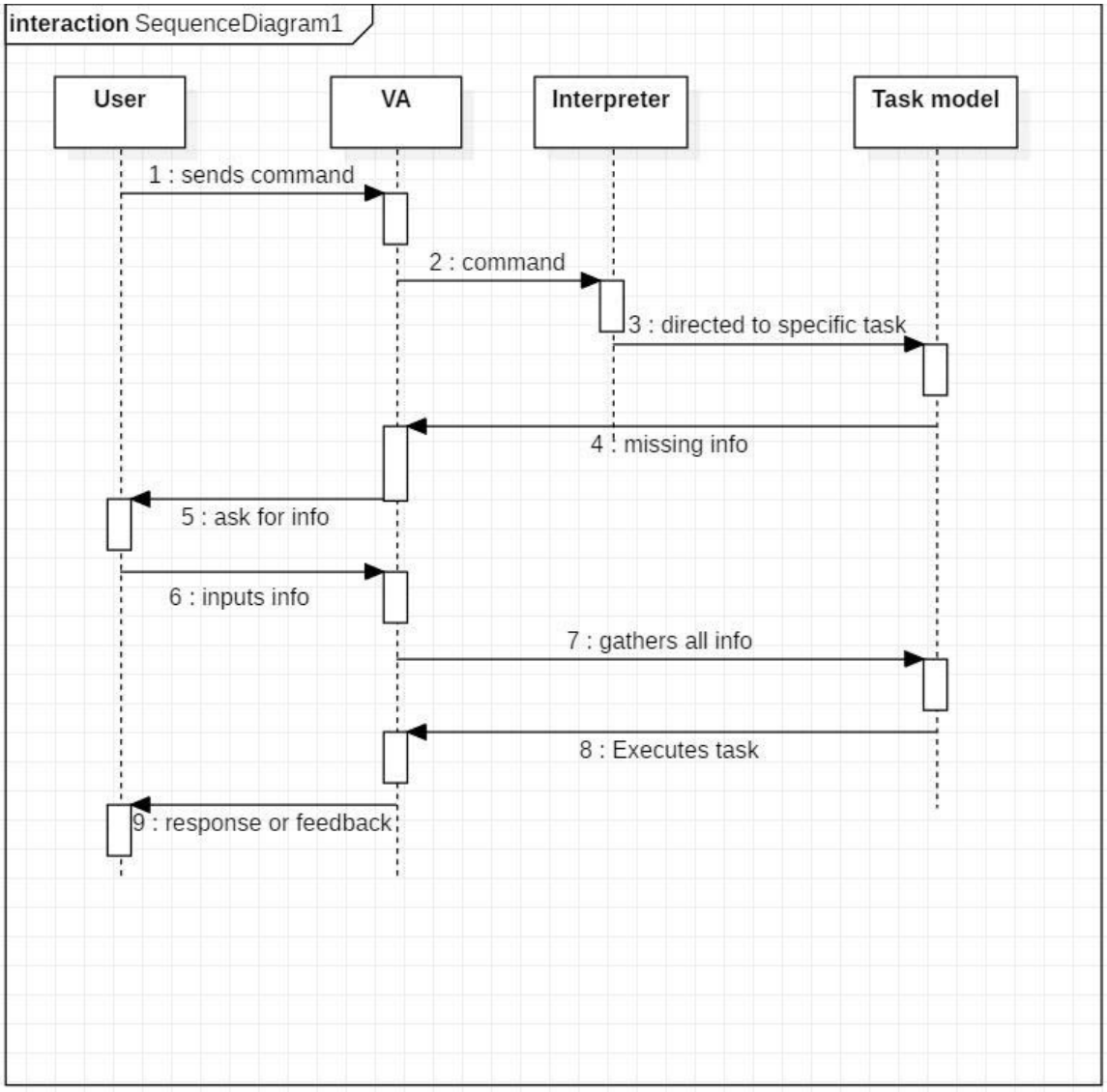
SEQUENCE DIAGRAM

SequencediagramforQuery-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.



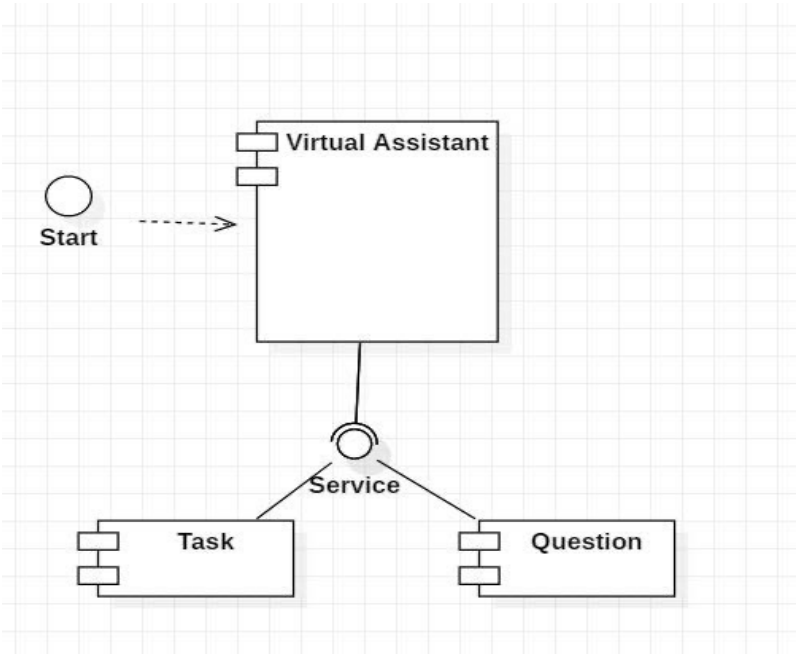
Sequence diagram for Task Execution



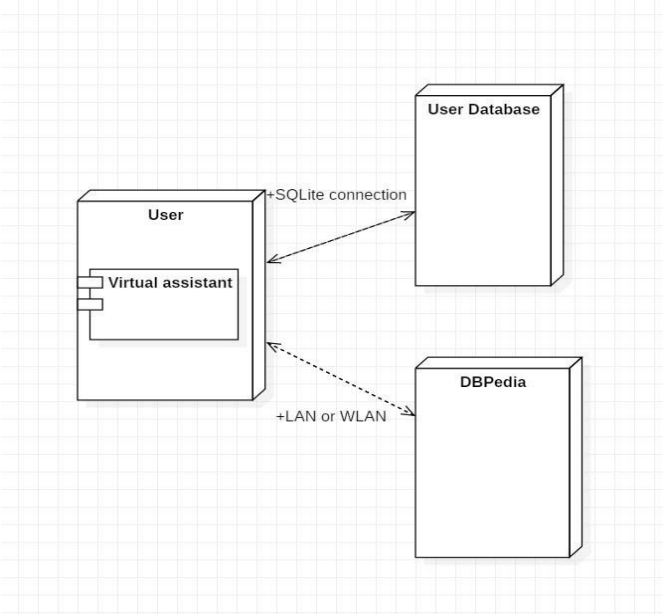
The user sends command to virtual assistant in audio form. The command is passed to the interpreter. It identifies what the user has asked and directs it to task executer. If the task is missing some info, the virtual assistant asks user back about it. The received information is sent back to task and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

COMPONENT DIAGRAM

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



DEPLOYMENT DIAGRAM



Project code

Importing all the necessary modules

```
from _tkinter import *
from http import server
import random
from datetime import datetime
import pywhatkit as kt
from subprocess import call
import subprocess
import cv2
import pyttsx3
import speech_recognition as tp
import smtplib
import time
import webbrowser
import requests
from pprint import pprint
import bs4 as bs4
ct = datetime.now()
```

- **Subprocess:-** This module is used to get system subprocess details used in various commands i.e Shutdown, Sleep, etc. This module comes built-in with Python.
- **Pyttsx3:-** This module is used for the conversion of text to speech in program it works offline. To install this module type the below command in the terminal.
  pip install pyttsx3
- **Tkinter:-** This module is used for building GUI and comes inbuilt with Python. This module comes built-in with Python.
- **Wikipedia:-** As we all know Wikipedia is a great source of knowledge just like GeeksforGeeks we have used the Wikipedia module to get information from Wikipedia or to perform a Wikipedia search. To install this module type the below command in the terminal.
- pip install wikipedia
- **Speech Recognition:-** Since we're building an Application of voice assistant, one of the most important things in this is that your assistant recognizes your voice (means what you want to say/ ask). To install this module type the below command in the terminal.
- pip install SpeechRecognition
- **Web browser:-** To perform Web Search. This module comes built-in with Python.
- **Ecapture:-** To capture images from your Camera. To install this module type the below command in the terminal.
- pip install ecapture
- **Datetime:-** Date and Time are used to showing Date and Time. This module comes built-in with Python.
- **Random:**-Python module random is an in-built module of python which is used to generate random numbers .This module is used to perform a random action such as generating random numbers, print random value from a list, etc.
- **Pywhatkit:**- Python offers numerous inbuilt libraries to ease our work. Among them **pywhatkit**is a Python library for sending WhatsApp messages at a certain time, it has several other features too.
  Following are some features of pywhatkit module:

  1. Play a YouTube video.

2. Perform a Google Search.
3. Get information on a particular topic

- **Smtplib:-**Python provides smtplib module which is used for sending emails using the Simple Mail Transfer Protocols(SMTP).
- **Instabot:-** Instabot is a Python module, which not only implements the wrapper over the Instagram API, but also various useful methods, such as login to an Instagram account, getting users from the data, getting media from users or from hashtags, comments, likes, follows, and uploading photos.By the way, we hardly recommend you not to use your own account if you don't need your personal data or don't want to promote your account. To get a quick understanding of how the bot works, let's start with trying out some of the bot functions.

Defining Methods:

```
def countdown(y):
    while y > 0:
        print(y)
        eng.say(str(y))
        eng.runAndWait()
        y = y-1
        time.sleep(1)
    print('bot->times up')
    eng.say('times up')
    eng.runAndWait()
```

The above module works as a timer

```
def cam():
    print('bot->opening camera')
    eng.say("opening camera")
    eng.runAndWait()
    cam = cv2.VideoCapture(0)
    count = 0
    while True:
        suc, f = cam.read()
        if suc:
            cv2.imshow("web cam", f)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
        if cv2.waitKey(1) & 0xFF == ord(' '):
            img_name = f'image{count}.png'
            cv2.imwrite(img_name, f)
            count += 1
    cam.release()
    cv2.destroyAllWindows()
```

The above method is used to access the camera. Opens the camera on voice command and closes the camera when 'q' is presses. The picture can be captured by pressing spacebar.

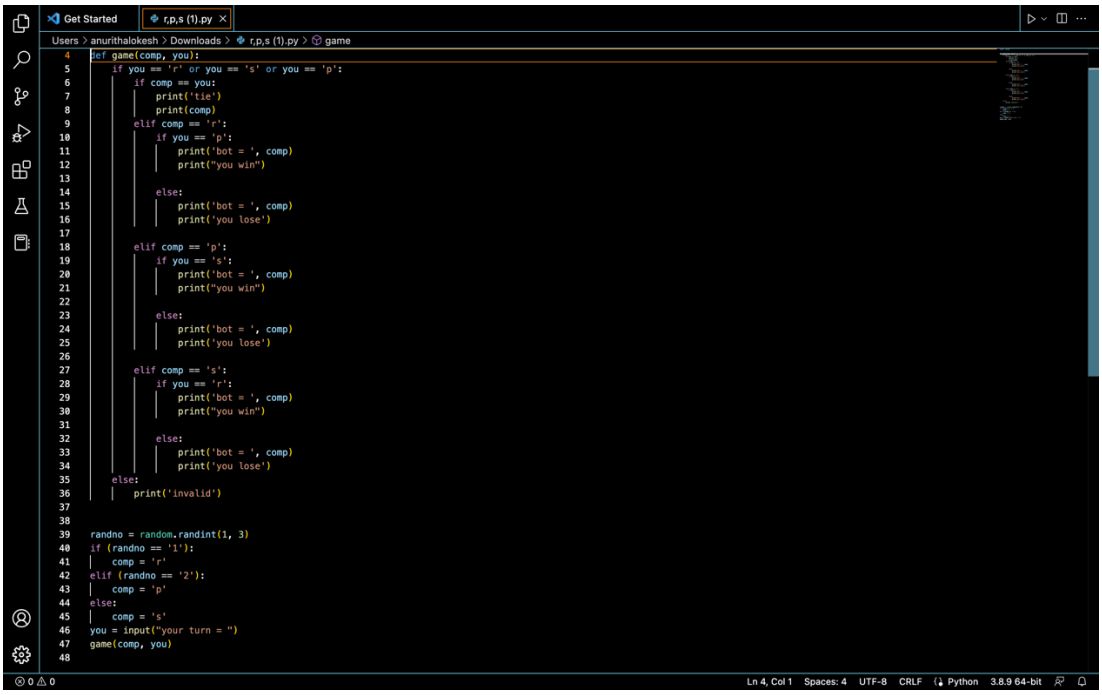
```
def sear():
    z = ui.split()
    z.remove('search')
    # print(z)
    t = ""
    for i in z:
        t += i+' '
        # print(t)
    print(f'bot->searching {t}in google')
    eng.say(f'searching {t} in google')
    eng.runAndWait()
    kt.search(t)
```

```python
def play():
    z = ui.split()
    # print(z)
    z.remove('play')
    # print(z)
    t = ""
    for i in z:
        t += i+' '
    print(f"bot->playing {t}on youtube")
    eng.say(f'playing {t}on youtube')
    eng.runAndWait()
    kt.playonyt("https://www.youtube.com/results?search_query="+t)

```

The above module is used to open youtube using the keyword play .The voice command given is stored as a string, when the word play is encountered the main function directs it to this module where all text connected to speech is stored in a string and truncates the word ‘play’ and stores the rest in a string.This string is passed to the youtube page to search for the video that the user would like to play.

```python
def open_py_file():
    call(['python', 'r,p,s.py'])

```

The above module calls the file containing a game of rock paper and scissors



The above image contains code for the rock paper scissors game.Where the user can play against the computer.

```python
def mail():
    server = smtplib.SMTP("smtp.gmail.com", 587)
    server.ehlo()
    server.starttls()
    server.ehlo()
    server.login('muhammadsuarim@gmail.com', 'gpqaspqgvzzfhbps')
    tad = input('inpput to address = ')
    subject = input('enter subject = ')
    body = input('enter body = ')
    msg = f'subject:{subject}\n\n{body}'

    server.sendmail(from_addr='muhammadsuraim@gmail.com',
                    to_addrs=tad, msg=msg)
```

The above module is used to send mail to a desired mail id and message of the mail con also be typed by the user. The module automatically logins in to the mail and sends the mail to the specified email address.

```python
def web():
    z = ui.split()
    # print(z)
    z.remove('open')
    # print(z)
    t = ""
    for i in z:
        t += i
    print(f'bot->opening {t}')
    eng.say(f'opening {t}')
    eng.runAndWait()
    webbrowser.open(f'https://www.{t}.com/')
```

The above module is used to open a desired website. The keyword used is open ,the voice command is converted to text and stored in a string the keyword ‘open’ is removed from the string and rest of the string is used to open the desired web page.

Calling all the modules:

```python
eng = pyttsx3.init('sapi5')
voices = eng.getProperty('voices')
# print(voices[0].id)
eng.setProperty('voice', voices[1].id)
```

There are two voices available and one of voices is selected on random when the code is run.

```python
gr = ['hello', 'hi', 'whats up']
gs = ['hi', 'hello', 'nice to meet you', good()]
er = ['bye', 'see you later', 'talk to you later']
es = ['take care see you later', 'bye bye', 'bye']
```

The above image contains lists which are randomized and called to greet the user.

```
while 1:
    r = tp.Recognizer()
    with tp.Microphone() as source:
        print("user->", end='')
        r.adjust_for_ambient_noise(source, duration=.2)
        aa = r.listen(source)
        ui = ''
        try:
            ui = r.recognize_google(aa, language='en-in')
            ui.lower()
            print(ui)
        except:
            print("try again")
            continue
    z = ui.split()
    if ui in gr:
        tt = random.choice(gs)
        print('bot->', tt)
        eng.say(tt)
        eng.runAndWait()
    elif 'time' in ui.split():
        eng.say(ct.hour)
        eng.say(ct.minute)
        eng.runAndWait()
    elif 'name' in ui.split() and 'your' in ui:
        name()
    elif ui in er:
        pp = random.choice(es)
        print('bot -> '+pp)
        eng.say(pp)
        eng.runAndWait()
        break
    elif 'search' in ui.split():
        sear()
    elif 'timer' in ui.split():
        yu = int(input('enter count = '))
        countdown(yu)
    elif 'game' in ui.split():
        open_py_file()
    elif 'play' in ui.split():
        play()
    elif 'camera' in ui.split():
        cam()
    elif 'mail' in ui.split():
        mail()
```

```
    elif 'mail' in ui.split():
        mail()
    elif 'calculator' in ui.split() and 'open' in ui.split():
        cal()
    elif 'open' in ui.split():
        web()
    else:
        print(f'bot->heres a google search for {ui}')
        eng.say(f'heres a google search for {ui}')
        eng.runAndWait()
        kt.search(ui)
```

The above part of code is used to convert speech to text and compares the keywords and checks the appropriate module to execute, it displays try again if not found.

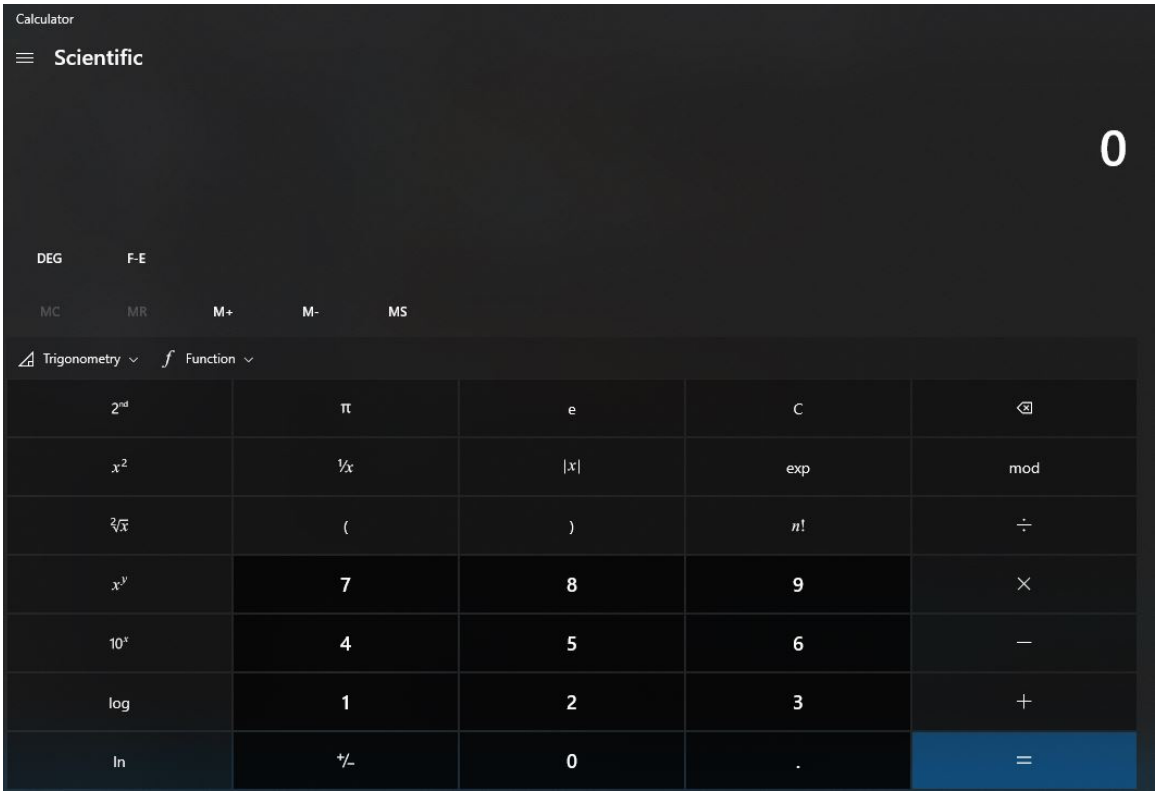
Result and Output:

Greets the user

```
user->hello
bot-> nice to meet you
```

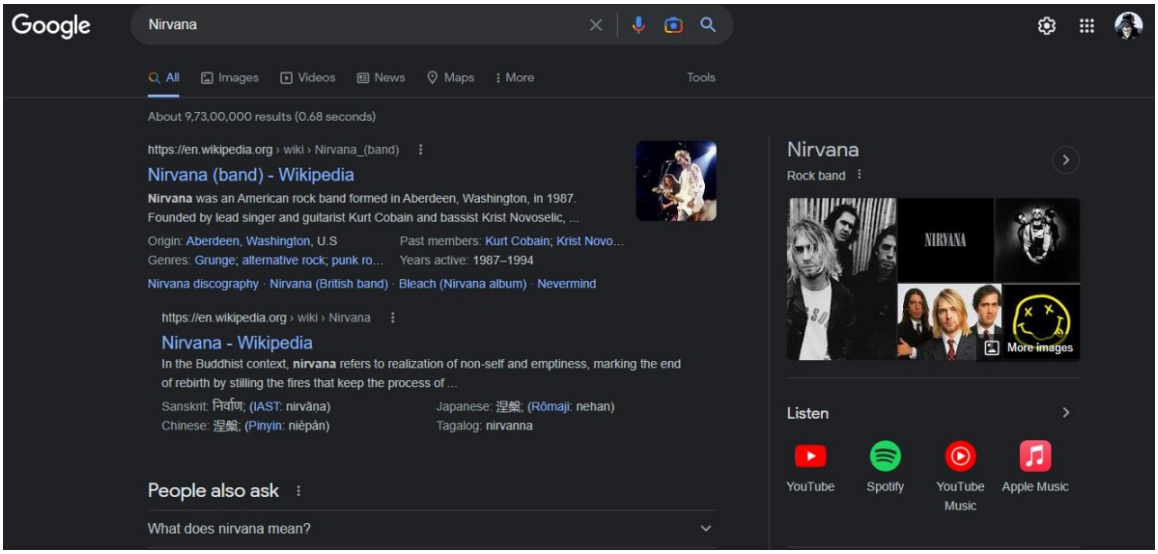
Used to open calculator

```
user->open calculator
bot->opening calculator
```



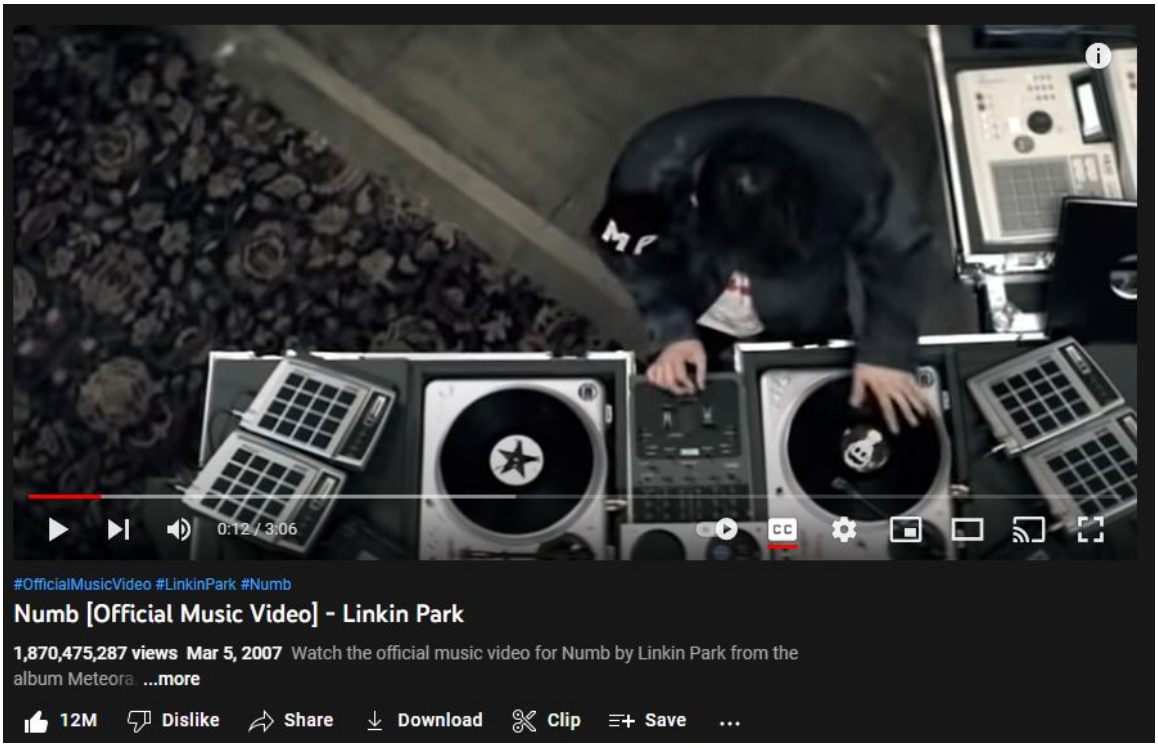
Google search

```
user->search Nirvana
bot->searching Nirvana in google
```



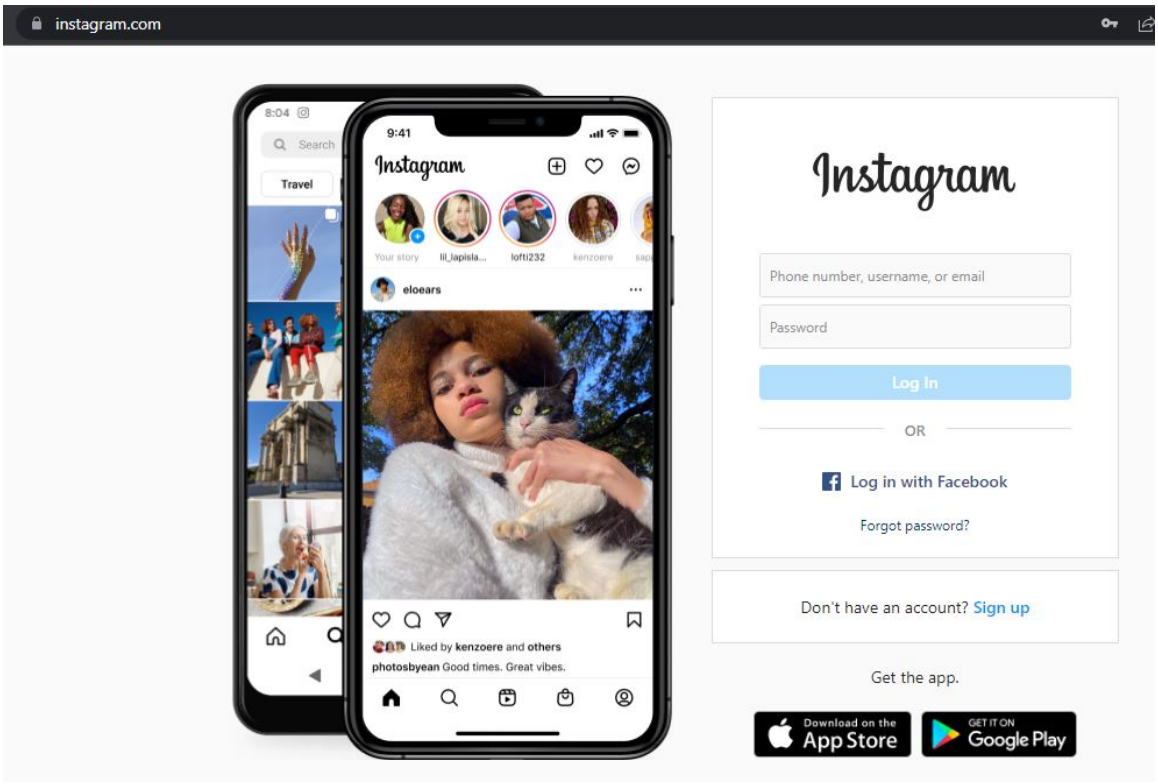
Play Youtube video

```
user->play Linkin Park Numb
bot->playing Linkin Park Numb on youtube
```



Opens Instagram login page

```
user->open Instagram
bot->opening instagram
```



Used to send mail

Prints try again when the command is not recognized

```
user->try again
user->try again
```

Conclusion

Advantages:

- Gives Handsfree experience
- Eyes free
- Minimal Effort
- Easy to use
- Easy for children
- Helps the elderly who may not know how to type
- Helps the visually disabled by assisting them in basic tasks

One thing is for sure. Personal voice assistance technology is here to stay. Just the simple thought of talking to a device to get some tasks done is an appealing innovation that presents multiple opportunities, most notably for businesses. The voice talking technology is poised to continue to shift consumer behaviour, and it, if necessary, for businesses to prepare to meet consumer needs. Getting into the voice technology space today is bound to make a huge difference and give your brand an edge in a highly competitive market.

Limitations :

- Offers only a limited numbers of features
- May not always recognize the command as dialects may vary
- Currently only recognizes one language English
- Can send mail to only one person at a time
- Code has to be run every time to access a new feature

Future Improvements:

- Offer a lot more features such as booking air tickets
- GPS voice assistant
- Offer recognition in a lot more languages that are commonly spoken
- Make code simple, quick and easier to run
- Allow the addition of attachments in mail

References

https://www.oberlo.in/blog/voice-search-statistics

https://www.geeksforgeeks.org/voice-assistant-using-python/

https://www.codewithharry.com/videos/python-tutorials-for-absolute-beginners-120/