VIRTUAL VOICE ASSISTANT

Project report in partial fulfilment of the requirement for the award of the degree of

Bachelor of Computer Application

In

COMPUTER APPLICATION

Submitted By

ANUSHKA PODDER	University Roll No.12018004009052
DEEPTENDU CHIKI	University Roll No.12018004009061
MANISHA GANGULY	University Roll No.12018004009078
SHIVAM DHANANJAY PANDEY	University Roll No.12018004009099
SAMARPITA SAHA	University Roll No.12018004009129

Under the guidance of

Prof. Anirban Das

Department of Computer Applications

UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160

CERTIFICATE

This is to certify that the project titled **VIRTUAL VOICE ASSISTANT** submitted by ANUSHKA PODDER **(University Roll No-**12018004009052**)**, DEEPTENDU CHIKI **(University Roll No-**12018004009061**)** MANISHA GANGULY **(University Roll No-**12018004009078**),** SHIVAM DHANANJAY PANDEY **(University Roll No-**12018004009099**) and** SAMARPITA SAHA **(University Roll No-**12018004009129**) s**tudents of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, in fulfilment of requirement for the degree of Bachelor of Computer Application is a bona fide work carried out by them under the supervision and guidance of Prof.Anirban Das during 5th Semester of academic session of 2020-2021. The content of this report has not been submitted to any other university or institute for the award of any other degree.

I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

____

Prof.Anirban Das
Assistant Professor
Department of CA
UEM, Kolkata

____

Prof. Kaustav Bhattacharjee
Head of the Department
Department of CA
UEM, Kolkata

ACKNOWLEDGEMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the on going course of this project remains invaluable to us.

We are sincerely grateful to our guide Prof.Anirban Das of the Department of Computer Application, UEM, Kolkata, for his wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to Prof. Kaustav Bhattacharjee, HOD, Computer Application, UEM, Kolkata and all other departmental faculties for their ever-present assistant and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

ANUSHKA PODDER

DEEPTENDU CHIKI

MANISHA GANGULY

SHIVAM DHANANJAY PANDEY

SAMARPITA SAHA

Table of Contents

Abstract

This technology could be harnessed to create an intelligent Virtual Personal Assistant (VPA) with a focus on user-based data. It will look at examples of intelligent programs with natural language processing that are currently available, with different categories of support, and examine the potential usefulness of one specific piece of software as a VPA. This engages the ability to communicate socially through natural language processing, holding and analysing data within the context of the user. It is suggested that new technologies may soon make the idea of virtual personal assistants a reality.

Experiments conducted on this system, combined with user testing, have provided evidence that a basic program with natural language processing algorithms in the form of a VPA, with basic natural language processing and the ability to function without the need for other type of human input (or programming) may already be viable.

The Most famous application of iPhone is “SIRI” which helps the end user to communicate end user mobile with voice and it also responds to the voice commands of the user. Same kind of application is also developed by the Google that is “Google Voice Search” which is used for in Android Phones. But this Application mostly works with Internet Connections. But our Proposed System has capability to work with and without Internet Connectivity. It is named as Personal Assistant with Voice Recognition Intelligence, which takes the user input in form of voice or text and process it and returns the output in various forms like action to be performed or the search result is dictated to the end user. In addition, this proposed system can change the way of interactions between end user and the mobile devices. The system is being designed in such a way that all the services provided by the mobile devices are accessible by the end user on the user's voice commands.

Keywords: *SIRI, Google Voice Search, Mobile Device, Internet*

Chapter 1

Introduction

A virtual assistant is a software agent that can perform tasks or services for an individual based on commands or questions. Sometimes the term "Chabot" is used to refer to virtual assistants generally or specifically accessed by online chat. In some cases, online chat programs are exclusively for entertainment purposes. Some virtual assistants are able to interpret human speech and respond via synthesized voices. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal (spoken) commands. A similar concept, however with

Differences, lays under the dialogue systems. Capabilities and usage of virtual assistants are expanding rapidly, with new products entering the market and a strong emphasis on both email and voice user interfaces. Apple and Google have large installed bases of users on smartphones. Microsoft has a large installed base of Windows-based personal computers, smart-phones and smart speakers.

The complexity and accuracy of voice recognition technology and voice assistant software have grown exponentially in the last few years. Currently available voice assistant products from Apple, Amazon, Google, and 86 M. B. HOY Microsoft allow users to ask questions and issue commands to computers in natural language. There are many possible future uses of this technology, from home automation to translation to companionship and support for the elderly. However, there are also several problems with the currently available voice assistant products. Privacy and security controls will need to be improved before voice assistants should be used for anything that requires confidentiality. Librarians should monitor these products and be ready to provide assistance to their patrons with these devices. They should also explore the possibilities for providing library materials via voice assistants as the technology matures.

Chapter 2

Literature Survey

2.1 Origin

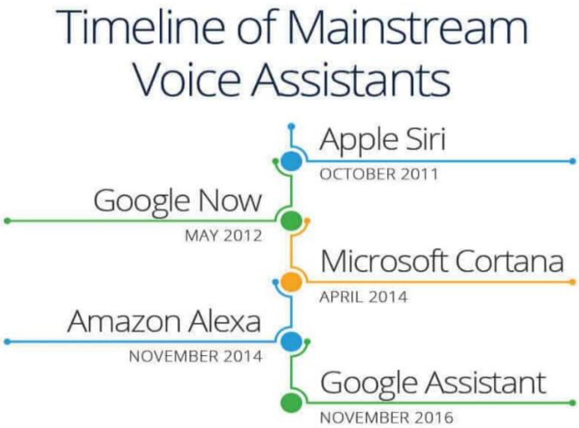


Fig 1:Mainstream Voice Assistants

At the Seattle World's Fair in 1962, IBM presented a tool called Shoebox. Throughout the 1990s, companies like IBM, Apple, and others created items that used voice recognition. Apple began building speech recognition features into its Macintosh computers with Plain Talk in 1993.
Abhay Dekate (2016) et al. presented in the Modern Era of fast-moving technology we can do things which we never thought we could do before but, to achieve and accomplish these thoughts there is a need for a platform which can automate all our tasks with ease and comfort.

Dr.Kshama V. Kulhalli (2017) et al. proposed the Most famous application of iPhone is “SIRI” which helps the end user to communicate end user mobile with voice and it also responds to the voice commands of the user. Same kind of application is also developed by the Google that is “Google Voice Search” which is used for in Android Phones. But this Application mostly works with Internet Connections.

Kishore Kumar R1 (2018) et al. presented to develop an economically effective and performance wise efficient virtual assistant using Raspberry Pi for home automation based on the concepts of Internet of Things, Speech Recognition, Natural Language Processing and Artificial Intelligence.

Rutuja V. Kukade (2018) et al. proposed there are various communication barriers for people who are blind, and they have to face various challenges.

VetonKëpuska (2018) proposed one of the goals of Artificial intelligence (AI) is the realization of natural dialogue between humans and machines. in recent years, the dialogue systems, also known as interactive conversational systems are the fastest growing area in AI. Many companies have used the dialogue systems technology to establish various kinds of Virtual Personal Assistants (VPAs) based on their applications and areas, such as Microsoft's Cortana, Apple's Siri, Amazon Alexa, Google Assistant, and Facebook's M.

Deny Nancy (2019) et al. presented in the Modern Era of fast-moving technology we can do things which we never thought we could do before but, to achieve and accomplish these thought s there is a need for a plat form which can automate all our tasks with ease and comfort. Thus we humans developed applications like Personal Voice Assistant having the ability to inter act with the surroundings just by one of the materialistic form of human inter action i.e .Human Voice.

Deepak Shende (2019) et al. presented artificial intelligence technologies are beginning to be actively used in human life, this is facilitated by the appearance and wide dissemination of the Internet of Things (IOT).

Isha S. Dubey (2019) et al. proposed about a different combination of a reading machine (OCR), virtual assistant and Domotics system using Raspberry-Pi which will be a combination of a great system. This is a helpful aid for visually impaired people and people with disabilities.

M. A. Jawale (2019) et al. proposed in today's world, many artificial intelligence applications developed using programming languages like Python, R and so on.

Tushar Gharge (2019) et al. presented the problem of user while developing a computer program. Developing a computer program is not an easy task it needs hardware resources which user have to handle. While continuous typing the code there may be possibility of injuries to the fingers of the user. To avoid the problems, we are designing a system in which the computer program can developed through the voice. The voice will be recognized by the system and that recognized words or word will be compared with the stored keywords in the database and if they are matched then that will be printed on editor and after this again by recognizing the specific keywords the program will be compiled and executed. This system will be easy to use, it reduces human efforts and the use of hardware resources. It would be surely useful for blind as well as novice plus knowledge intermittent users.

What are Voice Assistants?

The key here is voice. A voice assistant is a digital assistant that uses voice recognition, speech synthesis, and natural language processing (NLP) to provide a service through a particular application

Simply put, voice assistants are the realization of the science fiction dream of interacting with our computers by talking to them. Apple's Siri, Microsoft's Cortana, Amazon's Alexa, and Google's Assistant are all software agents that run on purpose-built speaker devices or smartphones. The software constantly listens for a key word to wake it up. Once it hears that key word, it records the user's voice and sends it to specialized server, which processes and interprets it as a command. Depending on the command, the server will supply the voice assistant with appropriate information to be read back to the user, play the media requested by the user, or complete tasks with various connected services and devices.

What Can Voice Assistants Do?

Although each currently available voice assistant has unique features, they share some similarities and are able to perform the following basic tasks:
. send and read text messages, make phone calls, and send and read email messages;
. answer basic informational queries ("What time is it? What's the weather forecast? How many ounces are in a cup?");
. set timers, alarms, and calendar entries;
. set reminders, make lists, and do basic math calculations;
. control media playback from connected services such as Amazon, Google Play, iTunes, Pandora, Netflix, and Spotify;
. control Internet-of-Things-enabled devices such as thermostats, lights, alarms, and locks; and
. tell jokes and stories.
In addition to these tasks, voice assistants can add other features, often called "skills," that expand their abilities by interfacing with other programs via voice commands

2.2 Desktop virtual assistants

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
- Set an alarm for 5am tomorrow morning
- 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 allocation 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.

Why Kolkata-based voice assistant startup Mihup feels offline is the way to go to protect privacy?

In the digital world, concerns about privacy and personal data being vulnerable to attacks and threats abound. But Kolkata-based Mihup, which offers an AI-enabled speech recognition platform for vernacular languages, believes that offline is the key to protecting privacy. The founders agree that India seems to be at greater risk compared to the rest of the world. According to the World Economic Forum's 14th edition of Global Risks Report 2019, India saw the most data breaches in 2018, compared to any other country. “To tackle the issue of privacy, offline is the solution. And that’s why we have stuck to providing offline solutions from the very start. Right from our B2C days to our pivot to B2B,” says Mihup Co-founder

2.3 Objective

The main objective of our Voice Assistant is to automate and help the user to do the basic things virtually. We generally spend lot of time in clicking and typing to do repetitive task. This problem can be solved by automating the tasks. With the help of our Virtual Assistant users can do a lot of tasks just by their voice command. Since our program takes only voice command, so it becomes very useful for physically challenged people. Despite of this, user can do simple conversation with our voice assistant. In future most of the manually done things are going to be automated using technology. Our project is intended to automate almost anything possible to help the user to save time and be efficient.

The main objective of our Voice Assistant is to automate and help the user to do the basic things virtually. We generally spend lot of time in clicking and typing to do repetitive task. This problem can be solved by automating the tasks. With the help of our Virtual Assistant users can do a lot of tasks just by their voice command. Since our program takes only voice command, so it becomes very useful for physically challenged people. Despite of this, user can do simple conversation with our voice assistant. In future most of the manually done things are going to be automated using technology. Our project is intended to automate almost anything possible to help the user to save time and be efficient.

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.

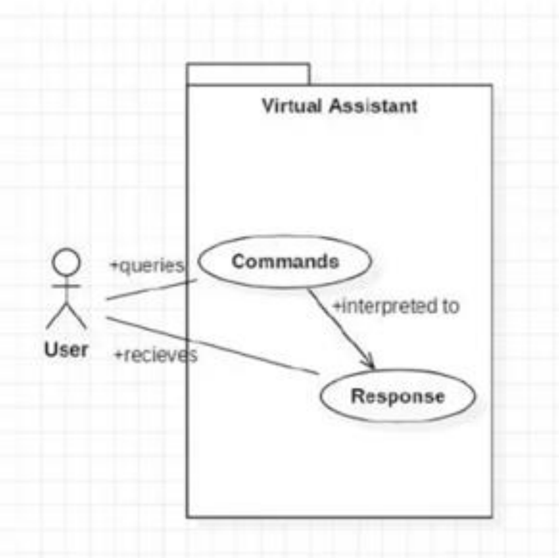


Fig 2: Basic Architecture

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. Four types of feasibility study are taken into consideration.

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 needs internet connection. While using assistant, make sure you have a steady internet connection. It is also not an issue in this era where almost every home or office has Wi-Fi.

Operational 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.

Economical 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, it won’t cost too much.

Organizational feasibility:

This shows the management and organizational structure of the project. This project is built by a team. The management tasks are all to be carried out by a single person. That won’t create any management issues and will increase the feasibility of the project.

This project is technically feasible with no external hardware requirements. Also, it is simple in operation and does not cost training or repairs. Overall feasibility study of the project reveals that the goals of the proposed system are achievable. Decision is taken to proceed with the project

Chapter 3

Problem

3.1 Problem Statement:

Our project “Virtual Assistant” is particularly based on Automation so it can be developed further according to user’s need.

Our program is written in python and it is very easy to understand. It contains some very powerful python modules like o s module which can use the window terminal to execute desired command.

Summary of the project

Virtual Assistant is a software application which is intended to automate some repetitive and time consuming task. It takes user’s voice command as it is input. We can send email, store messages, set alarm and countdown, read news and weather forecasts and much more.

Advantage of the Project

- It will help in increase the efficiency of the user.
- It does not require keyboard or mouse input.
- It can be also used by physically impaired as well as visually impaired people.
- It is news feature can be used to hear latest news from Google news API.
- Weather forecast can be listened from a very trusted weather API.
- Send email without typing anything.
- Set Alarm and set Brightness of Screen.
- Open your secret directory or most visited directories.
- Project can be easily modified with the help of it is source code and updated.

3.2 Methodology

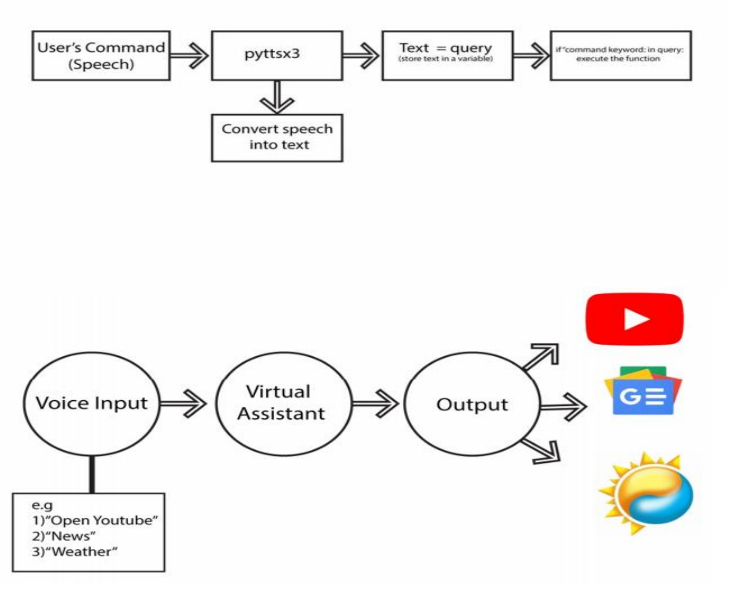


Fig 3.Methodology

Chapter 4

Problem Solution

4.1 Coding

```
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
def speak(audio):
 engine.say(audio)
 engine.runAndWait()
```

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline, and is compatible with both Python 2 and 3.

Here we have created the speak fuction with help of it to produce the output in audio format

```
def countdown(content):
 while content:
 mins, secs = divmod((content), 60)
 timer = '{:02d}:{:02d}'.format(mins, secs)
 print(timer, end="\r")
 time.sleep(1)
 content -= 1
 winsound.PlaySound("SystemQuestion", winsound.SND_ALIAS)
 speak("Completed")
```

Here countdown function takes time input from user and start timer. After completion it informs the user with an alert sound.

```
def wishMe():
 hour = int(datetime.datetime.now().hour)
 if hour >= 0 and hour < 12:
 speak("Good Morning!")
 elif hour >= 12 and hour < 18:
 speak("Good Afternoon!")
 else:
 speak("Good Evening!")
 print("I am Jarvis Sir. Please tell me how may I help you?")
 speak("I am Jarvis Sir. Please tell me how may I help you?")
```

wishMe function runs at the beginning to wish the user with appropriate greeting. In morning, it greets user by “Good Morning” and in Afternoon “Good Afternoon” and so on.

```
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("Say that again please...")
 return "None"
 return query
```

takeCommand is a very crucial part of our program. It’s function is to listen the user’s command using system’s Microphone. Try catch is used to handle the Exception.

```
def sendEmail(to, content):
 server = smtplib.SMTP('smtp.gmail.com', 587)
 server.ehlo()
 server.starttls()
 server.login('qwertyforwork@gmail.com', 'jyppeoscgghmqlbf')
 server.sendmail('qwertyforwork@gmail.com', toaddr, content)
server.close()
```

sendEmail function uses smtp(Standard mail transfer protocol) to send the email. Content of the email and recievers email address are argument of this function.

```
if 'youtube' in query:
 listening = True
 print("What should I search?")
 speak("What should I search?")
 content = takeCommand()
 url = 'https://www.youtube.com/results?search_query=' + content
 webbrowser.open(url)
 print('Done')
 speak('Done')
```

User’s input are stored in query. The above code means that if user’s instruction sentence contains word ‘youtube’, it will automatically execute everything under this if statement. Further user will be asked what to search.

```
elif "google" in query:
 listening = True
 url = 'https://www.google.com/'
 webbrowser.open(url)
 print('Done')
 speak('Done')
```

Else if user says open Google, then user is redirected to Google home page in web browser.

```
elif "calculator" in query:
 listening = True
 speak("Okay Sir, I am opening calculator")
 os.system("C:\\Windows\\System32\\calc.exe")
 elif "notepad" in query:
 listening = True
 speak("Okay Sir, I am opening notepad")
 os.system("C:\\Windows\\System32\\notepad.exe")
 elif "paint" in query:
 listening = True
 speak("Okay Sir, I am opening paint")
 os.system("C:\\Windows\\System32\\mspaint.exe")
```

To open calculator, notepad and paint, user needs to speak ‘calculator’,’notepad’, and ‘paint’ respectively. These are some daily uses.

```
elif 'time' in query:
 strTime = datetime.datetime.now().strftime("%H:%M:%S")
 speak(f"Sir, the time is {strTime}")
It speaks out the current time to the user.
 elif 'email' in query:
 try:
 speak("Whom should i sent email to?")
 to = takeCommand()
 edict = {'Shiv': 'mauryashivanand3@gmail.com',
 'Ankit': 'akumar397800@gmail.com', }
 toaddr = edict[to]
 speak("What should I say?")
 content = takeCommand()
 sendEmail(toaddr, content)
 speak("Email has been sent!")
 except Exception as e:
 print(e)
 speak("Sorry sir , I can not send this")
```

The above code asks the user receiver’s name and the subject line with content. After that message sent confirmation is received by user. If the email sending was unsuccessful then it will say.

```
“Sorry sir , I can not send this”.
 elif "how are you" in query:
 listening = True
 speak("I am fine sir")
 elif "what is your name" in query:
 listening = True
 speak("I don't have a name yet")
 elif "who are you" in query:
 listening = True
 speak("I am Virtual Assistant")
 elif "why virtual friend" in query:
 listening = True
 speak("Because i will help you to make your life simple")
```

The above line of code is to communicate with user. This type of conversation makes our Virtual Assistant more user-friendly.

```
elif "sound" in query:
 listening = True
 winsound.PlaySound("SystemQuestion", winsound.SND_ALIAS)
 speak("Done sir!")
```

You can check your speakers by saying sound. It will produce windows sound.

```
elif "message" in query:
 listening = True
 speak("What message you want to store?")
 content = takeCommand()
 speak("Alright!")
 sand.append(content)
We can store message by above code.
 elif "inbox" in query:
 listening = True
 for x in sand:
 speak(x)
 speak("do you want to delete last messege")
 query = takeCommand()
 if "yes" in query:
 sand.remove(x)
 if "no" in query:
 speak("Message not deleted")
 pass
```

To retrive the saved message, users need to say inbox. Voice Assistant will speak your message for you.

```
elif "write" in query:
 listening = True
 speak("speak")
 query = takeCommand()
 f = open("new.txt", "w")
 f.write(query)
 f.close()
```

The above code will help to write a document in .txt extension. This is good use of speech-to-text functionality.

```
elif "countdown" in query:
 listening = True
 speak("Tell me the time:")
 content = takeCommand()
 try:
 countdown(int(content))
 except:
 speak("try again")
```

The above code asks user to give time input and initiate the countdown function.

```
elif 'wikipedia' in query:
 speak('What should I search')
 query = takeCommand().lower()
 results = wikipedia.summary(query, sentences=3)
 speak("According to Wikipedia")
 print(results)
 speak(results)
```

The above code uses wikipedia module and extract out 3 paragraph and read them.

```
elif 'weather' in query:
 print("Which city Sir?")
 speak("Which city Sir?")
 api_key = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
 base_url = "http://api.openweathermap.org/data/2.5/weather?"
 city_name = takeCommand()
 print(str(city_name)+" Weather report:\n")
 complete_url = base_url + "appid=" + api_key + \
 "&q=" + city_name + "&units=metric"
 response = requests.get(complete_url)
 x = response.json()
 # print(x)
 if x["cod"] != "404":
y = x["main"]
 current_temperature = y["temp"]
 current_pressure = y["pressure"]
 current_humidiy = y["humidity"]
 z = x["weather"]
 weather_description = z[0]["description"]
 print(" Temperature (Celcius) = " +
 str(current_temperature) +
 "\n atmospheric pressure (hPa) = " +
 str(current_pressure) +
 "\n humidity () = " +
 str(current_humidiy) + "%" +
"\n description = " +
 str(weather_description))
 speak(" Temperature (in celcius unit) = " +
 str(current_temperature) +
 "\n atmospheric pressure (in hPa unit) = " +
 str(current_pressure) +
 "\n humidity (in percentage) = " +
 str(current_humidiy) +
 "\n description = " +
 str(weather_description))
 else:
 print(" City Not Found ")
```

In above code, OpenWeatherMap API has been used to get important weather information.
e.g. Atmospheric Pressure, Humidity, brief Description of current weather.

```
elif 'play music' in query:
 music_dir = 'D:\\MUSIC\\My Downloaded Music'
 songs = os.listdir(music_dir)
 # print(songs)
 os.startfile(os.path.join(music_dir, songs[0]))
```

The execution of above code plays a song from your music directory.

```
elif "news" in query:
 news_url = "https://news.google.com/news/rss"
 Client = urlopen(news_url)
 xml_page = Client.read()
 Client.close()
 count = 0
 speak("How many news headline sir? ")
 limit = takeCommand()
 soup_page = soup(xml_page, "lxml")
 news_list = soup_page.findAll("item")
 # Print news title, url and publish date
 for news in news_list:
 print(news.title.text)
 print(news.link.text)
 speak(news.title.text)
 speak(news.link.text)
 count += 1
 if count > int(limit):
 break
```

The function of the above code is to read the latest news from Google

```
elif "exit" in query:
 speak("See you later, Sir")
 exit()
```

news. It asks user number of headlines to read.

To exit from the Voice Assistant program exit command can be used.

```
elif "shutdown" in query:
 speak("Do you wish to shutdown your computer ? (yes or no): ")
 shutdown = takeCommand().lower()
 if shutdown == 'yes':
 os.system("shutdown /s /t 1")
 else:
 Pass
```

To shut down the computer we can give assistant command “shutdown”. It Prompts user if he/she really want to shutdown his/her PC.

Like Shutdown function, we have restart function which too asks user if he/she really want to restart their PC.

```
elif "sleep" in query:
 speak("Do you wish to sleep your computer ? (yes / no):
")
 shutdown = takeCommand().lower()
 if shutdown == 'yes':
 os.system("shutdown /h /t 2")
 else:
 pass
```

```
elif "alarm" in query:
 speak("Please enter time manually")
 alarm_hour = int(input("Set hour: "))
 alarm_minutes = int(input("Set minutes: "))
 am_pm = input("am or pm? ")
 winsound.PlaySound("SystemQuestion", winsound.SND_ALIAS)
 print(f"Waiting for time: {alarm_hour}:{alarm_minutes} {am_pm}")
 if am_pm == 'pm': # to convert pm to military time
 alarm_hour += 12
 elif alarm_hour == 12 and am_pm == 'am':

 alarm_hour -= 12
 else:
 pass
 while True:
 #infinite loop that runs until alarm time
if alarm_hour == datetime.datetime.now().hour and alarm_minutes ==
datetime.datetime.now().minute:
 print("\nIt's the time!")
```

We can also put our computer on sleep mode with above code.

The function of above code is to Set alarm in your voice assistant. Users have to enter time manually. When it’s time, Assistant alert the user with sound.

```
elif "brightness" in query:
 ans = "no"
 curr_brightness = sbc.get_brightness()
 while(ans != "yes"):
 speak("How much brightness do you want?")
 try:
 br = takeCommand()
 sbc.set_brightness(br)
 speak("Do you want this much ??")
 ans = takeCommand().lower()
 if (ans == "no"):
 sbc.set_brightness(curr_brightness)
 except:
 speak("Please speak clearly")
 ans = "no"
 speak("Brightness has been set to " + curr_brightness)
 print(curr_brightness)
```

With “Brightness” command the above code is executed. It sets Brightness of the screen according to user’s need. It keeps confirming until user is satisfied with the brightness intensity.

```
elif "translate" in query:
        speak("speak please")
        inputtrans = takeCommand().lower()

        translator = google_translator()

        translate_text = translator.translate(inputtrans, lang_src='en', lang_tgt='hi')

        text=translate_text
        speaktext = gTTS(text=text, lang="hi", slow=False)
        speaktext.save("captured_voice.mp3")
        os.system("start captured_voice.mp3")
        speak(translate_text)
```

With “translate” we can change the language “English” to language”Hindi”.

Chapter 5

Experimental Setup and Result Analysis

5.1 Requirement specification

Hardware and Software Requirement

Software Requirement: -

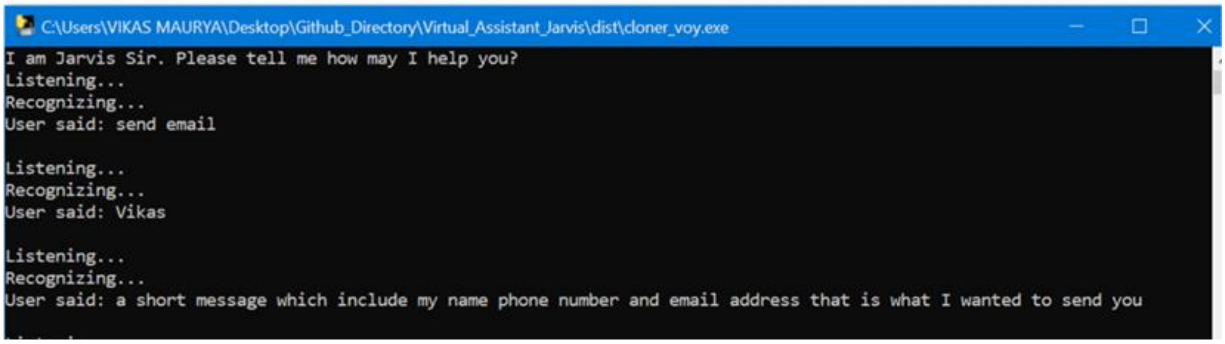
- Operating System: - Windows 10 (64-bit)
- Python 3.8
- Audio Recognition software
- Internet connection.

Hardware Requirement: -

- At least 500MB of free space.
- At least 2 GB RAM
- Mic (for Audio input)
- Speaker (for Audio Output)

5.2 Screenshots

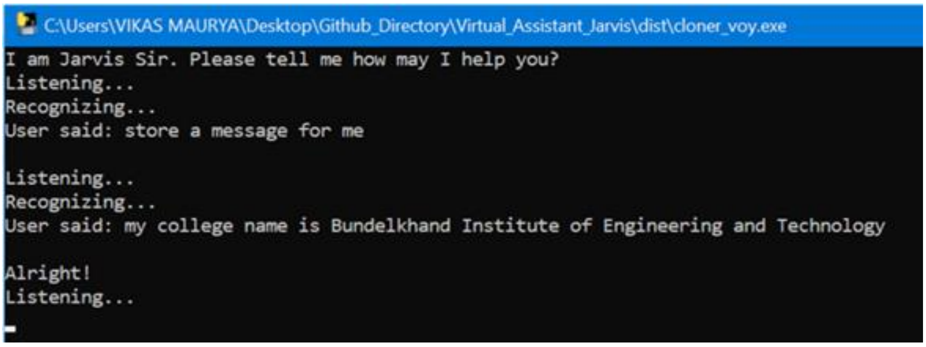
a. Sending Email


C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Jarvis\dist\cloner_voy.exe

```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: send email

Listening...
Recognizing...
User said: Vikas

Listening...
Recognizing...
User said: a short message which include my name phone number and email address that is what I wanted to send you
```

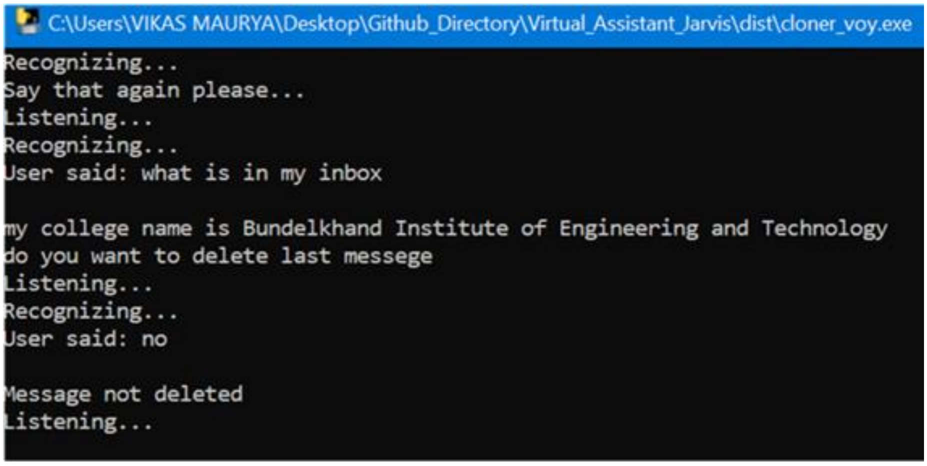
Fig 4. Sending Email

b. Store and Read Messages


C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Jarvis\dist\cloner_voy.exe

```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: store a message for me

Listening...
Recognizing...
User said: my college name is Bundelkhand Institute of Engineering and Technology

Alright!
Listening...
```

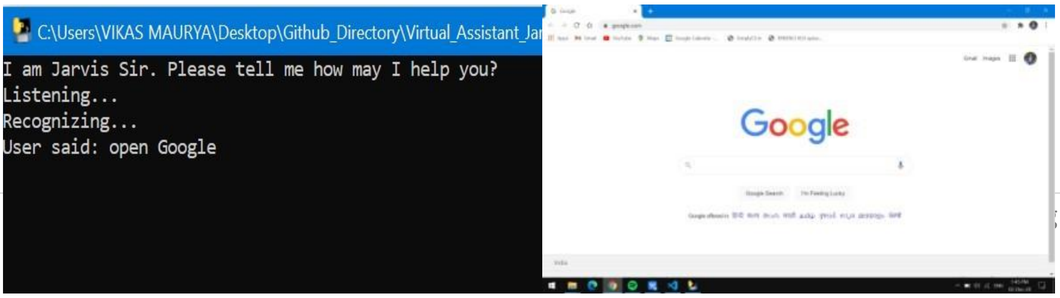
Fig 5. Storing Message


C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Jarvis\dist\cloner_voy.exe

```
Recognizing...
Say that again please...
Listening...
Recognizing...
User said: what is in my inbox

my college name is Bundelkhand Institute of Engineering and Technology
do you want to delete last messege
Listening...
Recognizing...
User said: no

Message not deleted
Listening...
```

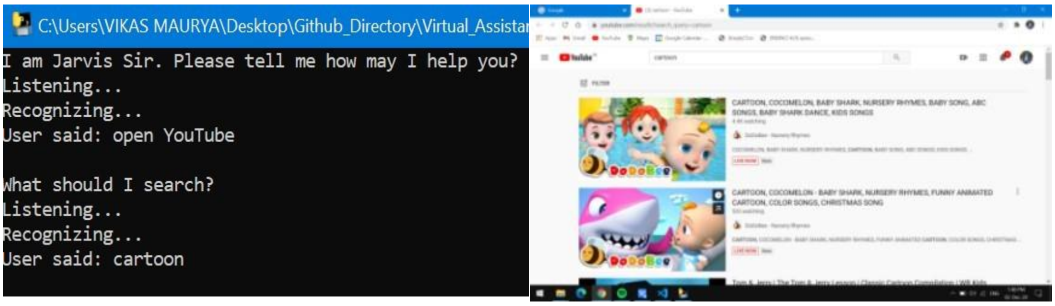
Fig 6. Reading stored message

c. Open Google and YouTube


C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Ja
Google

```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: open Google
```

Fig 7.Opening Google Search Home page

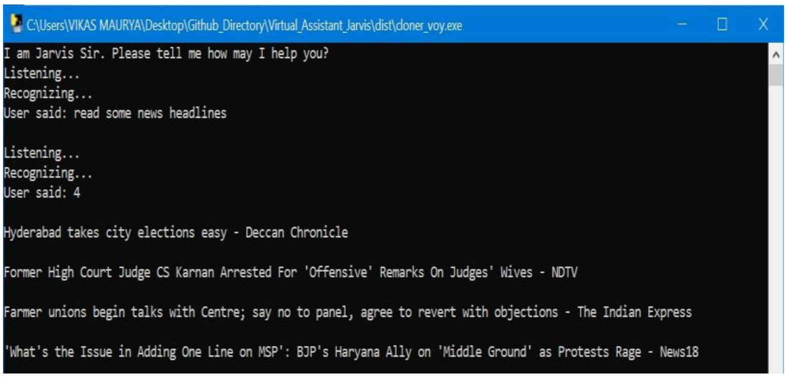


C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assista

```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: open YouTube

What should I search?
Listening...
Recognizing...
User said: cartoon
```

Fig 8. Opening YouTube with a topic

d. Read News Headlines

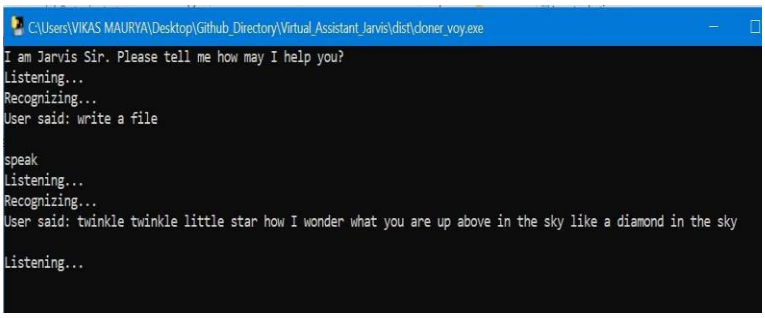


C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Jarvis\dist\doner_voy.exe

```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: read some news headlines

Listening...
Recognizing...
User said: 4

Hyderabad takes city elections easy - Deccan Chronicle

Former High Court Judge CS Karnan Arrested For 'Offensive' Remarks On Judges' Wives - NDTV

Farmer unions begin talks with Centre; say no to panel, agree to revert with objections - The Indian Express

'What's the Issue in Adding One Line on MSP': BJP's Haryana Ally on 'Middle Ground' as Protests Rage - News18
```

Fig 9.Reading news directly from Google news

e. Write a file (.txt)

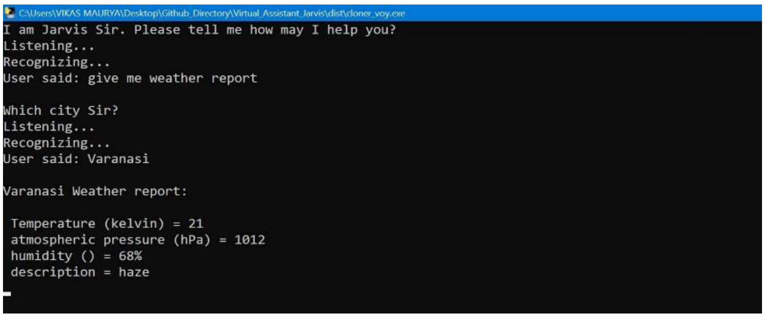


C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Jarvis\dist\doner_voy.exe

```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: write a file

speak
Listening...
Recognizing...
User said: twinkle twinkle little star how I wonder what you are up above in the sky like a diamond in the sky

Listening...
```

Fig 10.Write a file(.txt)

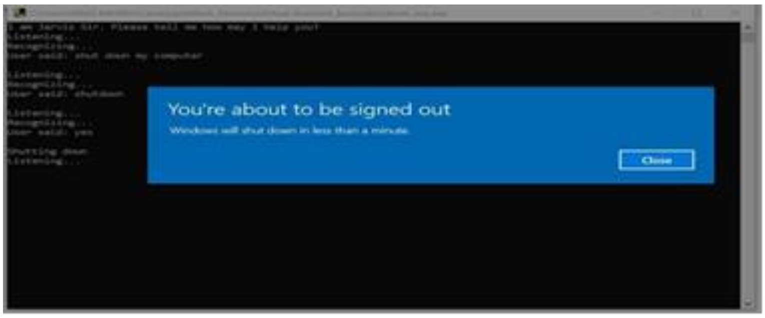
f. Weather forecast


C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Jarvis\dist\cloner_voy.exe

```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: give me weather report

Which city Sir?
Listening...
Recognizing...
User said: Varanasi

Varanasi Weather report:

 Temperature (kelvin) = 21
 atmospheric pressure (hPa) = 1012
 humidity () = 68%
 description = haze
```

Fig 11. Weather forecast

g. Power Options (Shutdown and Restart)



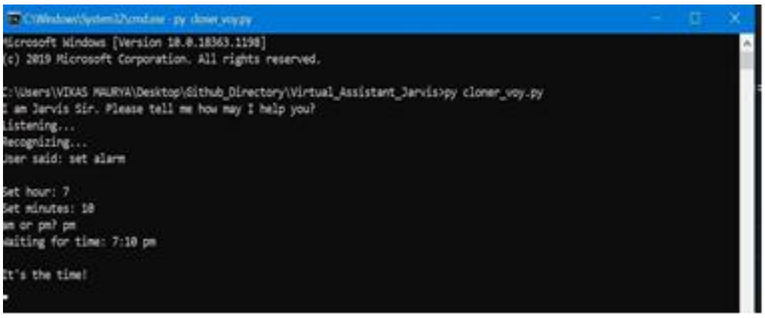
```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: shut down my computer

Listening...
Recognizing...
User said: shutdown

Listening...
Recognizing...
User said: yes

Shutting down
Listening...
```

You're about to be signed out

Windows will shut down in less than a minute.

Fig 12. Power Options (Shutdown and Restart)

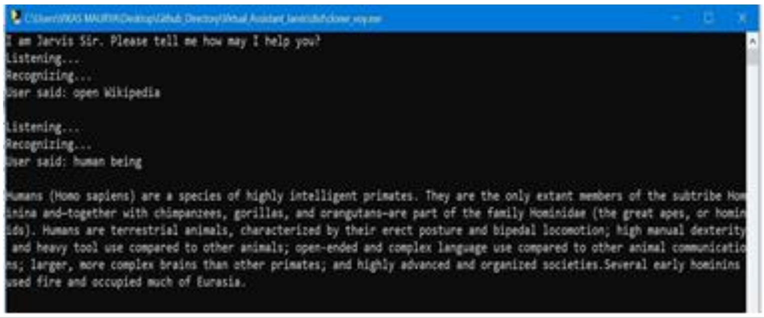
h. Alarm



```
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Jarvis>py cloner_voy.py
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: set alarm

Set hour: 7
Set minutes: 10
am or pm? pm
Waiting for time: 7:10 pm

It's the time!
```

Fig 13. Alarm

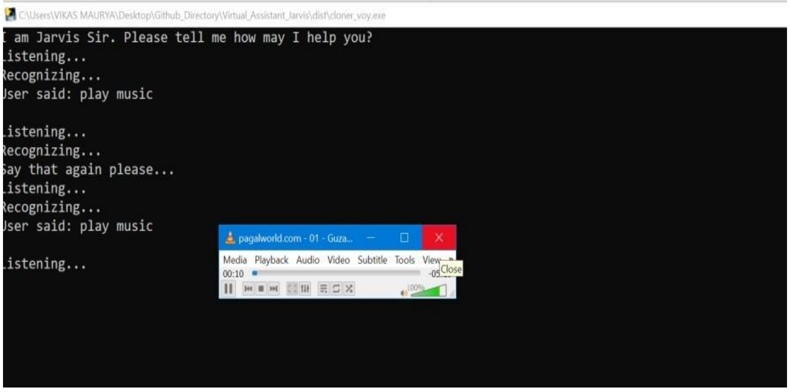
i. Wikipedia


C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Jarvis\dist\cloner_voy.exe

```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: open Wikipedia

Listening...
Recognizing...
User said: human being

Humans (Homo sapiens) are a species of highly intelligent primates. They are the only extant members of the subtribe Hom
inina and-together with chimpanzees, gorillas, and orangutans-are part of the family Hominidae (the great apes, or homin
ids). Humans are terrestrial animals, characterized by their erect posture and bipedal locomotion; high manual dexterity
 and heavy tool use compared to other animals; open-ended and complex language use compared to other animal communicatio
ns; larger, more complex brains than other primates; and highly advanced and organized societies.Several early hominins
used fire and occupied much of Eurasia.
```

Fig 14. Wikipedia

j. Play Music

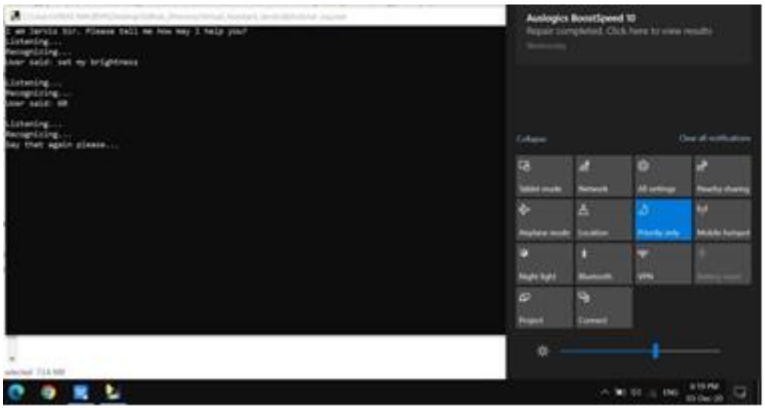


```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: play music

Listening...
Recognizing...
Say that again please...
Listening...
Recognizing...
User said: play music

Listening...
```

C:\Users\VIKAS MAURYA\Desktop\Github_Directory\Virtual_Assistant_Jarvis\dist\cloner_voy.exe
pagalworld.com - 01 - Guza...
Media Playback Audio Video Subtitle Tools View
00:10
-05
Close
100%

Fig 15. Play Music

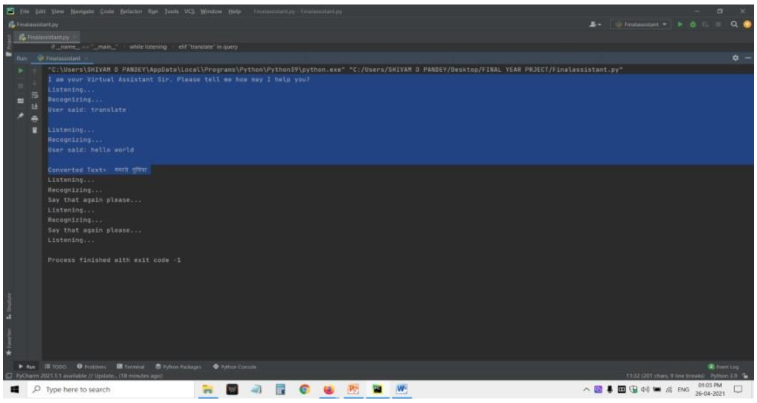
k. Set Brightness



```
I am Jarvis Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: set my brightness

Listening...
Recognizing...
User said: 60

Listening...
Recognizing...
Say that again please...
```

Fig 16: Set Brightness

l. Translate



```
"C:\Users\SHIVAM D PANDEY\AppData\Local\Programs\Python\Python39\python.exe" "C:/Users/SHIVAM D PANDEY/Desktop/FINAL YEAR PRJECT/Finalassistant.py"
I am your Virtual Assistant Sir. Please tell me how may I help you?
Listening...
Recognizing...
User said: translate

Listening...
Recognizing...
User said: hello world

Converted Text:  नमस्ते दुनिया
Listening...
Recognizing...
Say that again please...
Listening...
Recognizing...
Say that again please...
Listening...

Process finished with exit code -1
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Finalassistant.py
if __name__ == "__main__" › while listening › elif "translate" in query
Run: Finalassistant
TODO Problems Terminal Python Packages Python Console
Event Log
PyCharm 2021.1.1 available // Update... (18 minutes ago)
Type here to search

Fig 17: Translate

Chapter 6

Testing

6.1 Testing

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

6.2 Unit Testing

Unit testing is a type of software testing where individual units or components of software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a White Box testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing.

Unit testing Unit is important because software developers sometimes try saving time doing minimal unit testing and this is myth because in appropriate unit testing leads to high cost Defect fixing during System Testing, Integration Testing and even Beta Testing after application is built. If proper unit testing is done in early development, then it saves time and money in the end. Here are the key reasons to perform unit testing.

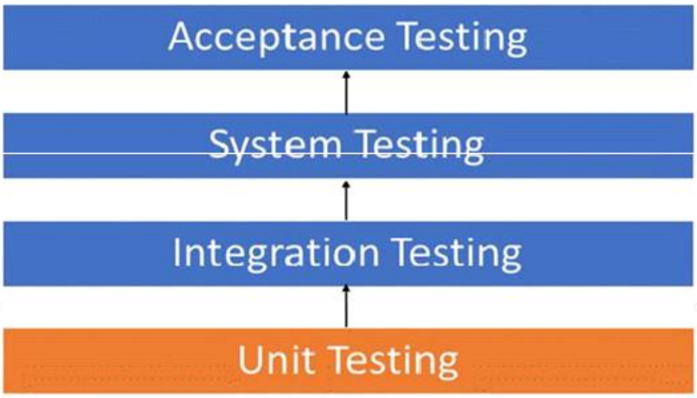


Fig 18: Unit testing

1. Unit tests help to fix bugs early in the development cycle and save costs.

2. It helps the developers to understand the code base and enables them to make changes quickly.

3. Good unit tests serve as project documentation.

4. Unit tests help with code re-use. Migrate both your code and your tests to your new project. Tweak the code until the tests run again.

6.3 Integrated Testing

Integration testing is defined as a type of testing where software modules are integrated logically

And tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

Integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as 'I & T' (Integration and Testing), 'String Testing' and sometimes 'Thread Testing'. Although each software module is unit tested, defects still exist for various reasons like-

- A Module, in general, is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration Testing becomes necessary to verify the software modules work in unity.
- At the time of module development, there are wide chances of change in requirements by the clients. These new requirements by the clients. These new requirements may not be unit tested and hence system integration Testing becomes necessary.
- Interfaces of the software modules with the database could be erroneous
- External Hardware interfaces, if any, could be erroneous
- Inadequate exception handling could cause issues

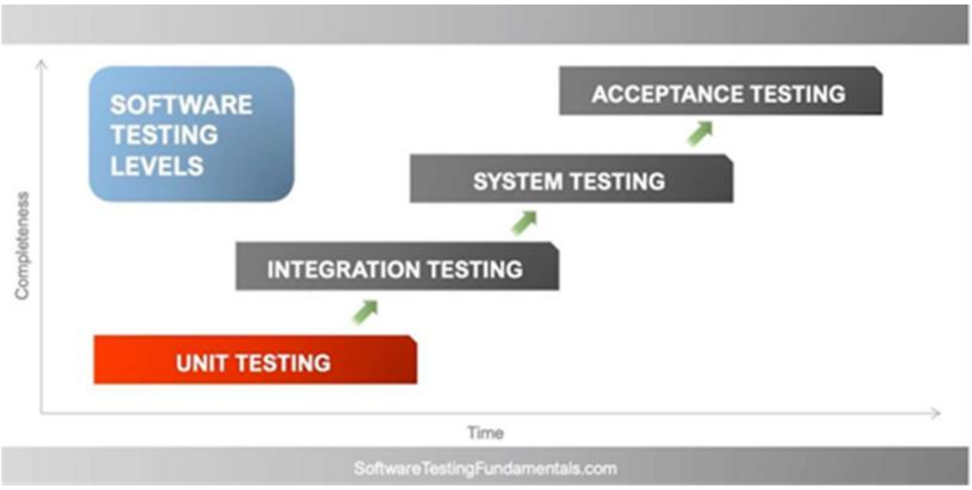


Fig 19: Integrated testing

6.4 Black box testing

Black Box Testing is a software testing method in which the functionalities of software application and tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioural Testing.

The above Black-Box can be any software system you want to test. For Example, an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

Here are the generic steps followed to carry out any type of Black Box Testing Initially, the requirements and specifications of the system are examined.

- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenarios) are chosen to verify that the SUT is able to detect them.
- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any fixed and re-tested.

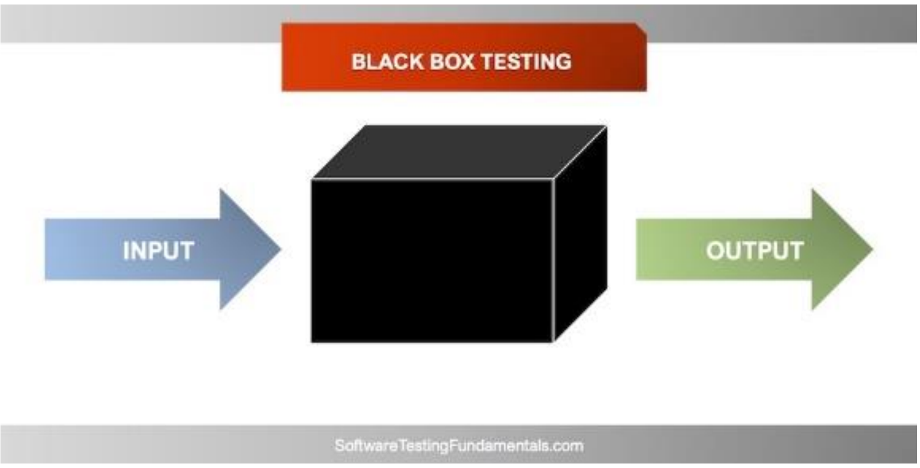


Fig 20: Black box testing

6.5 Future Scope

7 Key Predictions for the Future of Voice Assistants and AI

When voice assistants began to emerge in 2011 with the introduction of Siri, no one could have predicted that this novelty would become a driver for tech innovation. Now nearly eight years later, it’s estimated that every one in six Americans own a smart speaker (Google Home, Amazon Echo) and eMarketer forecasts that nearly 100 million smartphone users will be using voice assistants in 2020.

Brands such as Amazon, Google are continuing to fuel this trend as they compete for market share. Voice interfaces are advancing at an exponential rate in industries of all kinds, ranging from healthcare to banking, as companies are racing to release their own voice technology integrations to keep pace with consumer demand.

What’s Causing The Shift Towards Voice?

The main driver of the shift towards voice user interfaces is the changing user demands. 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.

The mass adoption of artificial intelligence in users’ everyday lives is also fuelling the shift towards voice applications. The number of IoT devices such as smart thermostats, appliances, 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; however, it only starts there. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years.

Applications of this technology are seen everywhere, so where will it take us in 2020 and beyond? We provide a high-level overview of the potential that voice has and 7 key predictions we think will take off in the coming years.

7 Key Predictions for Voice In 2020

Streamlined Conversations

Both Google and Amazon recently announced that both assistants will no longer require the use of repeated “wake” words. Previously both assistants were dependent on a wake word (Alexa or Ok, Google) to initiate a new line of conversation. For example, one would have to ask “Alexa, what’s the current temperature at the hallway thermostat?” and then have to say, “Alexa” again before requesting that the voice assistant to “set the hallway thermostat to 23 degrees.” It would be more convenient and natural for the user to say, “Alexa, what’s the current temperature at the hallway thermostat?” and then simply say “set my hallway thermostat to 23 degrees,” without requiring the wake word again, and now that’s possible.

Consumers use voice assistants in specific locations, usually while multitasking, and can either be alone or amongst a group of people when using them. Having devices that can decipher these contextual factors make a conversation more convenient and efficient with these devices, but it also shows that developers behind the technology are aiming to provide a more user-centric experience.

Compatibility and Integration

When it comes to integrating voice technology with other products, Amazon has been ahead of the game. Those who use Alexa will be familiar with the fact that the voice assistant is already integrated into a vast array of products including Samsung’s Family Hub refrigerators. Google has finally caught on and has announced Google Assistant Connect. The idea behind this technology is for manufacturers to create custom devices that serve specific functions and are integrated with the Assistant.

In 2020, we will see a greater interest in the development of voice-enabled devices. This will include an increase in mid-level devices: devices that have some assistant functionality but aren’t full-blown smart speakers. Instead, they communicate with your smart speaker, display or even perhaps your phone over Bluetooth where the processing happens on those devices. Amazon is already well on its way with an Alexa-enabled wall clock.

Search Behaviours Will Change

Voice search has been a hot topic of discussion. Visibility of voice will undoubtedly be a challenge. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface unless it is connected to the Alexa or Google Assistant app. Search behaviours, in turn, will see a big change. In fact, if tech research firm Juniper

Research is correct, voice-based ad revenue could reach $19 billion by 2022, thanks in large part to the growth of voice search apps on mobile devices.

Brands are now experiencing a shift in which touch points are transforming to listening points, and organic search will be the main way in which brands have visibility. comScore data even reveals that 50% of all search will be via voice tech by 2020.

As voice search grows in popularity, advertising agencies and marketers expect Google and Amazon will open their platforms to additional forms of paid messages.

Individualized Experiences

Voice assistants will also continue to offer more individualized experiences as they get better at differentiating between voices. Google Home is able to support up to six user accounts and detect unique voices, which allows Google Home users to customize many features. Users can ask “What’s on my calendar today?” or “tell me about my day?” and the assistant will dictate commute times, weather, and news information for individual users. It also includes features such as nicknames, work locations, payment information, and linked accounts such as Google Play, Spotify, and Netflix. Similarly, for those using Alexa, simply saying “learn my voice” will allow users to create separate voice profiles so the technology can detect who is speaking for more individualized experiences.

Voice Push Notifications

We’ve previously discussed the method of using user-centric push notifications as a means to re-engage users with your app, voice technology presents a unique means of distributing push notifications. As a way to increase user engagement and retention, push notifications simply remind users of the app and display relevant messaging to the user. Now that both Google Assistant and Amazon’s Alexa allow the user to enable spoken notifications for any third-party app that has the compatibility, users can hear notifications rather than read them. These notifications are generally related to calendar appointments or new content from core features.

Touch Interaction

CES 2019 continued to prove that voice and visual displays are merging into one seamless experience. This year Google showcased what is being called the E Ink screen. This display can show the weather, local traffic information, or calendar events. The push to bring visual and voice capabilities together allow users to further interact with the assistant.

Security Will Be A Focus

Forty-one percent of voice assistant users are concerned about trust and privacy according to a report from Microsoft. With news from Google I/O and Amazon's re:MARS conferences announcing that assistants will essentially be able to plan an entire evening, for example, find local movie times, buy tickets, book a restaurant reservation and schedule an Uber, concerns regarding payments and sensitive information are valid. Voice payments, in particular, will become more secure and convenient for users to make purchases. Speaker verification and ID will also become paramount as part of the voice assistant experience with more security being built around the user.

Why Adopt A Mobile Voice Strategy?

Mobile phones are already personalized, more so than any website. Additionally, there is very little screen space on mobile, making it more difficult for users to search, or navigate. With larger product directories and more information, voice applications enable consumers to use natural language to eliminate or reduce manual effort, making it a lot faster to accomplish tasks.

Whether that's finding out information, making a purchase, or achieving a task, voice is the new mobile experience. It's clear that brands are racing to figure out their voice strategy. With over 100 million Alexa devices being sold alone, there's a reason why businesses are looking to catch up.

New call-to-action

Voice User Interface (VUI) Will Continue To Advance

Even with just that handful of simple scenarios, it's easy to see why voice assistants are shaping up to become the hubs of our connected homes and increasingly connected lives.

Voice technology is becoming increasingly accessible to developers. For example, Amazon offers Transcribe, an automatic speech recognition (ASR) service that enables developers to add speech-to-text capability to their applications. Once the voice capability is integrated into the application, users can analyse audio files and in return, receive a text file of the transcribed speech.

Google has made moves in making Assistant more ubiquitous by opening the software development kit through Actions, which allows developers to build voice into their own products that support artificial intelligence. Another one of Google's speech-recognition products is the AI-driven Cloud Speech-to-Text tool which enables developers to convert audio to text through deep learning neural network algorithms.

This is only the beginning of voice technology as we will see major advancements in the user interface in the years to come. With the advancements in VUI, companies need to start educating themselves on how they can best leverage voice to better interact with their customers. It's important to ask what the value of adding voice will be as it doesn't always make sense for every brand to adopt. How can you provide value to your customers? How are you solving their pain points with voice? Will voice enhance the user experience or frustrate the user?

In 2020, voice-enabled apps will not only accurately understand what we are saying, but how we are saying it and the context in which the inquiry is made.

However, there are still a number of barriers that need to be overcome before voice applications will see mass adoption. Technological advances are making voice assistants more capable particularly in AI, natural language processing (NLP), and machine learning. To build a robust speech recognition experience, the artificial intelligence behind it has to become better at handling challenges such as accents and background noise. And as consumers are becoming increasingly more comfortable and reliant upon using voice to talk to their phones, cars, smart home devices, etc., voice technology will become a primary interface to the digital world and with it, expertise for voice interface design and voice app development will be in greater demand.

Voice Is The Future Of Brand Interaction And Customer Experience

Advancements in a number of industries are helping digital voice assistants become more sophisticated and useful for everyday use. Voice has now established itself as the ultimate mobile experience. A lack of skills and knowledge make it particularly hard for companies to adopt a voice strategy. There is a lot of opportunity for much deeper and much more conversational experiences with customers. The question is, is your brand willing to jump on this opportunity?

Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a White Box testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing.

6.5 Conclusion

During this thesis, I have analysed, investigated, and tested three virtual assistants. They were Amazon Alexa, Google Assistant, and Apple Siri. First I have taken a look at how these assistants work, how they have been developed, and some other facts. My next step was to analyse possible privacy and security risks. After that, I have tested the assistants in three categories: Simple questions, Conversation, and Commands. As a result of these steps, I discovered that the Google Assistant is the smartest and the best one of these three, Amazon Alexa ended up second, and Apple Siri finished in the last place. The first half of the thesis showed me what the flaws and advantages of an assistant are. According to that, I have set a few characteristics of a future perfect assistant. Those were: being offline, giving the user notification when recording audio/video, not keeping the history, and expanding the features and list of compatible devices.

In the second half, I created my virtual assistant. I have called it SIA - Simple Intelligent Assistant.

Time is precious and this project will let user save their time efficiently. Voice assistant can be personalized for a user’s day to day life. There is tremendous possibility of growth of this virtual assistant. It will reduce dependency on human resources this project is very beneficial for differently abled people due to the fact that this project can use voice or text for communicating with the user. Also this Voice Assistant can be used for personal as well as commercial use.

References

[1] https://en.wikipedia.org/wiki/Virtual_assistant

[2] https://pypi.org/project/pyttsx3/

[3] https://docs.python.org/3/library/smtplib.html

[4] https://openweathermap.org/api

[5] https://www.geeksforgeeks.org/python-winsound-module/