A Project Report

on

DESKTOP VOICE ASSISTANT SYSTEM FOR VISUALLY IMPAIRED INDIVIDUDALS

Submitted in partial fulfilment of the

requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

Under The Supervision of

Name of Supervisor : Dr. Dileep Kumar Yadav

Professor, Galgotias University

Submitted by **VISHAL BISHT(18SCSE1060010)**

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

DECEMBER,2021

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **" DESKTOP VOICE ASSISTANT SYSTEM FOR VISUALLY IMPAIRED INDIVIDUDALS"** in partial fulfilment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September, 2021 to December, 2021, under the supervision of Dr Dileep Kumar Yadav, Professor, Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

VISHAL BISHT, 18SCSE1060010

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Dileep Kumar Yadav
Professor, SCSE, GU

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **VISHAL BISHT** (18SCSE1060010) , has been held on **__18-12-2021__** and his work is recommended for the award of Bachelor of Technology in Computer Science and Engineering:

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: 18 December, 2021

Place: Greater Noida

TABLE OF CONTENTS

S.No	Particulars	Page No
1	Abstract	5
2	Introduction	6-8
3	Literature Reviews/Comparative study	9-10
4	Problem Formulation	11
5	Proposed work	12
6	Tools used	13
7	Modules used	14-18
8	Project Architecture	19-22
9	Result	22-26
10	Research Paper	26-30

TITLE- DESKTOP VOICE ASSISTANT SYSTEM

I. ABSTRACT

In the era of Technology, everyone aspires to live a more comfortable, convenient, and tech-savvy. In this hustle, everyone requires a personal assistant to help them with their everyday tasks.

A personal voice assistant is software that can carry out various tasks and give services to a person based on their spoken commands. This is accomplished by a synchronous process that involves the speech patterns recognition and subsequent creation of synthetic speech, followed by the operations of command given. These assistants allow users to automate operations such as texting, mailing, emergency contact dialling, giving home remedies for diseases and media playback etc. Also they might be emotionally active. People are developing reliant on technology as it advances, and the computer or laptop is one of the most widely used platforms. We all want to make using these computers more pleasant and easy to use. The usual way to provide a command to the computer is to type it in, but a more convenient approach is to speak it.

Giving voice input is useful not just for regular individuals, but also for those who are visually handicapped. unable to provide input using a keyboard. Only India has 18.4 million blind individuals, accounting for one-fourth of all the people worldwide. India is home to one in every three blind people. This necessitates the use of a voice assistant that can not only receive commands via speech but also execute the required instructions and provide output by voice or any other method.

II. INTRODUCTION

In the era of Technological Development and Advancement , it's our duty to make technology more simplified and usable. Many new features are being added to mobile phones and desktop computers as technology advances. We need a quicker and more reliable mode of input to use them in a more convenient and enjoyable manner. In our voice assistant project, we use voice commands to enter data into the system, which requires the usage of a microphone to convert acoustic energy to electrical energy. Our voice assistant is a computer program that can conduct tasks and perform operations for a individual based on human verbal instructions, such as using human voice commands and the voice assistant responding with synthetic voice. Users may use voice commands to ask their assistants questions, control home automation devices, and playback media, as well as perform other basic tasks like as email, to-do lists, open or stop any programme, and send messages to anyone on WhatsApp. The following are some of the fundamental tasks that may be accomplished with the assistance of a voice assistant: -

• Reading the newspaper

•Getting mail updates

• Getting weather updates

• Searching the internet

• Listening to music or watching a video

• Running any software or application.

These are only instances; we can do a lot more depending on our needs.

With this project our goal is not only to simplify the things for normal individuals but also to the impaired one's. This computer application will help many disabled individuals, to assist them while performing some basic tasks in their Desktop. According to the World Health Organization, 1 billion people distance vision impairment due to uncorrected refractive error (89 million), glaucoma (7.7 million), corneal opacities (4.2 million), diabetic retinopathy (3.9 million), cataract (94 million) and trachoma (2 million), as well as near vision impairment due to uncorrected presbyopia (826 million).So, this project will help and assist many of such impaired individuals and will make their technological life easier.

HISTORY OF VOICE ASSISTANT

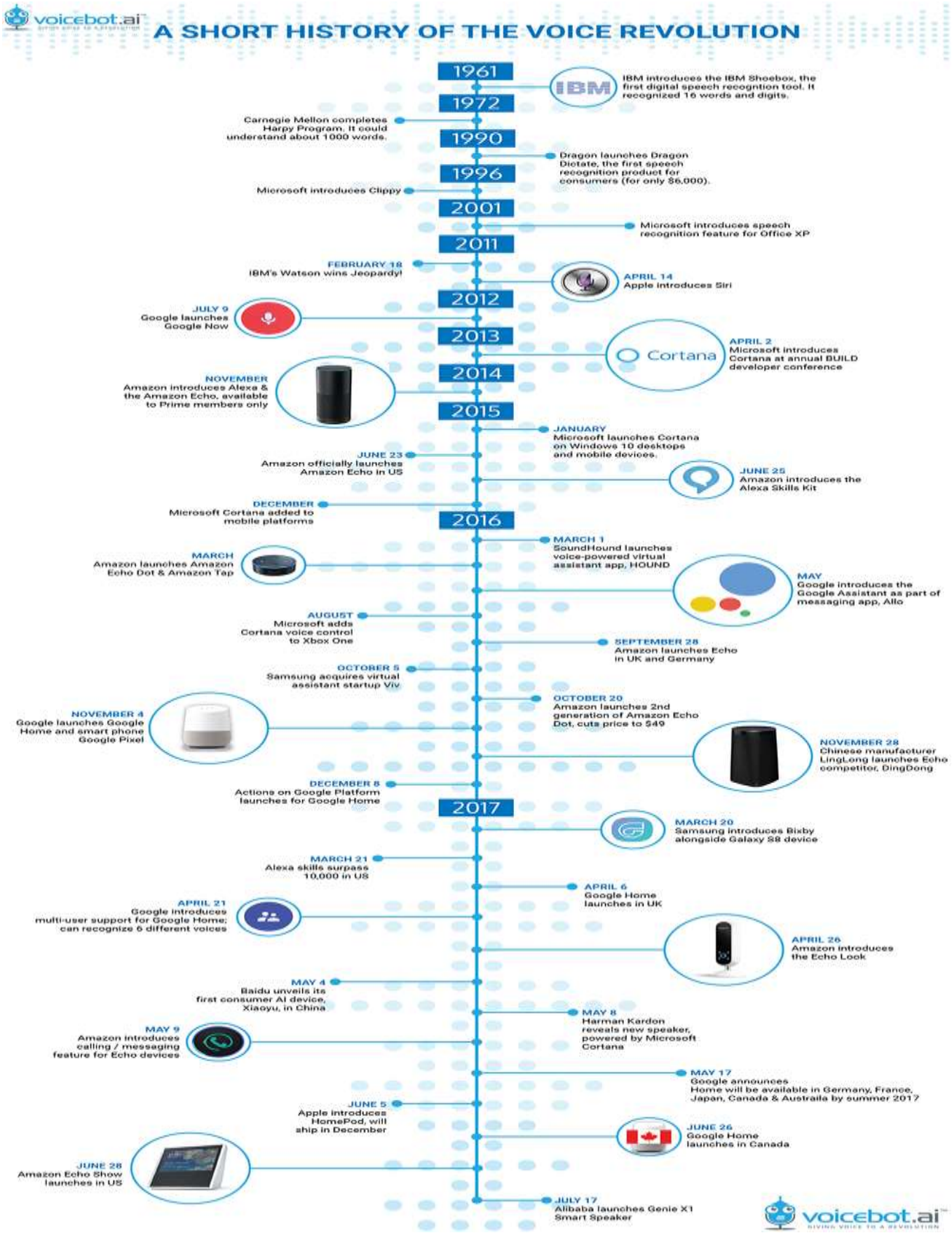


Fig. 1. History timeline of voice assistant

THE ORIGIN OF VOICE ASSISTANT

Voice assistants have a very long history that actually goes back over 100 years, which might seem surprising as apps such as Siri have only been released within the past ten years. The very first voice activated product was released in 1922 as Radio Rex. This toy was very simple, wherein a toy dog would stay inside a dog house until the user exclaimed its name, "Rex" at which point it would jump out of the house. This was all done by an electromagnet tuned to the frequency similar to the vowel found in the word Rex, and predated modern computers by over 20 years. At the 1952 World's fair, Audrey was announced by Bell Labs. The Automatic Digit Recognizer was not a small simple device however, its casing stood six feet tall just to house all the materials required to recognize ten numbers! IBM began their long history of voice assistants in 1962 at the World's Fair in Seattle when IBM Shoebox was announced. This device was able to recognize digits 0-9 and six simple commands such as, "plus, minus" so the device could be used as a simple calculator. Its name referred to its size, similar to the average shoebox, and contained a microphone connected to three audio filters to match the electric frequencies of what was being said and matched it with already assigned values for each digit. Darpa then funded five years of speech recognition R&D in 1971, known as the Speech Understanding Research (SUR) Program. One of the biggest innovations to come out if this was Carnegie Mellon's Harpy, which was capable of understanding over 1,000 words. The next decade led to amazing progress and research in the speech recognition field, leading most voice recognition devices from understanding a few hundred words to understanding thousands, and slowly making their way into consumers homes. Then, in 1990, Dragon Dictate was introduced to consumers homes for the shocking price of $9,000! This was the first consumer oriented speech recognition program designed for home PC's. The user could dictate to the computer one word at a time, pausing in between each word waiting for the computer to process before they could move on. Seven years later, Dragon NaturallySpeaking was released and it brought more natural conversation, able to understand continuous speech at a maximum of 100 words per minute and a much lower price tag of $695. In 1994, Simon by IBM was the first smart voice assistant. Simon was a PDA, and really, the first smartphone in history, considering it predates HTC's Droid by practically 25 years! In 2008, when Android was first released, Google had slowly started rolling out voice search for its Google mobile apps on various platforms, with a dedicated Google Voice Search Application being released in 2011. This led to more and more advanced features, eventually leading to Google now and Google Voice Assistant. Then, this was followed by Siri in 2010. Developed by SRI International with speech recognition provided by Nuance Communications, the original app was released in 2010 on the iOS App Store and was acquired two months later by Apple. Then, with the release of the iPhone 4s, Siri was officially released as an integrated voice assistant within iOS. Since then, Siri has made its way to every Apple device available and has linked all the devices together in a single ecosystem. Shortly after Siri was first developed, IBM Watson is announced publicly in 2011. Watson was named after the founder of IBM, and was originally conceived in 2006 to beat humans at a game of Jeopardy. Now, Watson is one of the most intelligent, naturally speaking computer systems available. Amazon Alexa is then announced in 2015. It's name being inspired by the Library of Alexandria and also the hard

consonant “X” in the name, helping with more accurate voice recognition. With Alexa, the Echo line of smart devices are announced to bring smart integration to consumers homes for an inexpensive route. Alan is finally publicly announced in 2017 to take the Enterprise Application world by storm. Being first born as “Synqq”, Alan is created by the minds behind “Qik”, the very first video messaging and conferencing mobile app. Alan is the first voice AI platform aimed at enterprise applications, so while it can be found in many consumer applications, it is designed for enterprises to be able to develop and integrate quickly and efficiently! At the bottom of the post we’ve included a Timeline to summarize the history of voice assistants!

III. LITERATURE SURVEY

The creation of voice assistants began in 1962 at the World's Fair in Seattle, when IBM displayed a gadget dubbed Shoebox IBM that could detect spoken numerals and subsequently do calculations and return them by lighting lights with the numbers 0-9 labeled on them. It was able to recognize a total of 16 words. [1]According to J. B. Allen, language is the most important medium of communication, and voice is its primary interface. As a human-machine interface, voice signals were transformed into analogue and digital waveform that a machine could understand. [2]Miss. Priyanka V. Mhamunkar and colleagues presented a technology that would allow individuals to retrieve word meanings using synthesized sounds. [3]Speech analysis is commonly conducted in tandem with pitch analysis, as indicated by B. S. Atal and L. R. Rabiner et al. Based on signal dimensions, the study developed a pattern recognition technique for identifying whether a given slice of a speech signal should be classified as voiced speech, unvoiced speech, or silence. The technique's biggest limitation is the requirement to run the algorithm on a specified set of dimensions and under specific recording conditions. [4]Speech is the most common means of communication among humans, according to V. Radha and C. Vimala et al. Humans would prefer to communicate with machines using speech because it is the most advanced method. As a result, automated speech recognition has gained a lot of traction. The most common speech recognition techniques are Dynamic Time Warping (DTW) and HMM. Mel Frequency Cepstrum Coefficients (MFCC) were used for speech feature mining because they provide a group of distinctive vectors of speech waveform. MFCC has been shown to be more exact and real than rest characteristic mining techniques in voice recognition in previous research. The work was conducted in MATLAB, and the results show that is capable of detecting words with a good accuracy. [5]Tong Lai Yu and Santhrushna Gande have created an Android solution that uses open source services to assist programmers who face physical issues while developing. On the client side, they employed the producer-consumer paradigm to integrate operations that may be performed using an Android handset. [6]Omyonga Kevin and Kasamani Bernard Shibwabo claim to have created an app that can carry out spoken orders without the need for an internet connection, saving us money on data. Their technique is faster than many strong engines like Alexa and Cortana since it does not require an internet connection. [7]T. Schultz and A. Waiel et al. highlighted that when speech technology products proliferate over the world, the inability to adapt to new destination languages becomes a useful worry. As a result, the study focuses on the question of how to quickly and efficiently port large vocabulary continuous speech recognition (LVCSR) systems. In the context of the Global Phone project, which examines LVCSR methods in 15 languages, the research needs to evaluate acoustic models for a novel destination language using speech information from different source languages, but only limited data from the destination language identification outcomes using language-dependent, independent, and language-adaptive acoustic models are described and debated. [8] A morphological analyser for most Indian language NLP applications was described by Mugdha Bapat, Pushpak Bhattacharyya, and others. Through- out the research, they

described and estimated a Marathi morphological analyser. The study found that Marathi has a high level of correctness, with consistent inflectional standards when using Finite State Systems to demonstrate language in a sophisticated fashion. Because Marathi has complicated morphotactic, grouping of post places and the formation of FSA is one of the most important aids.[9] Sutar Shekhar and a group of academics have collaborated on an application that uses speech to execute the majority of the system's functions, as well as the ability to send a message using their voice command, to aid visually impaired individuals. They want to keep improving their software so that it can eventually recognize a variety of local languages, such as Bengali and a number of dialects of the common Hindi language.

In my project work I have chosen Python as the programming language since it is one of the most robust, with huge libraries and with the use of pyttsx3 and voice recognition API's, development becomes easier and more accurate. To assist visually impaired persons, our software always repeats the command that the user enters into the system, so that the user knows if they entered the correct command or not. On the other hand, it continues to listen and respond to the user's needs until the user decides to stop.

IV. **PROBLEM FORMULATION**

As in introduction I have discussed about product is made to help the visually impaired peoples and also ease the User Experience of the users, while working on the desktop. Earlier using system even for a basic tasks was bit difficult for the specially abled individuals, they uses traditional braille scripted keyboard to these short tasks, but with the help of voice assistant system they can perform all day to day tasks very easily and simplified manner

V. PROPOSED WORK

Our work is based on NLP(Natural Language Processing) for that we used python module Speech recognition to recognize user's speech and convert into string and for desktop assistant response speech we have used module pyttsx3.

Initially, the system will continue to listen for commands, and the amount of time spent listening is variable and can be adjusted to meet the user's needs. If the system cannot obtain information from the user, it will prompt the user to repeat the process until the desired number of times is reached. Playing music, sending emails, sending texts, searching Wikipedia, opening system-installed applications, opening anything in a web browser, and other features are supported in the current version.

Fig.2 Working of voice assistant

VI. TOOLS USED

1.Visual Studio Code: Visual Studio Code is a simplified code editor that supports development operations such as debugging, task execution, and version control. It aims to provide only the tools a developer requires for a quick code-build-debug cycle, leaving more complex workflows to full-featured IDEs like Visual Studio IDE.

2. PYTHON programming language: Python is a high-level general-purpose programming language that is interpreted. With its use of significant indentation, its design philosophy emphasises code readability. Its language constructs and object-oriented approach are intended to assist programmers in writing clear, logical code for small and large-scale projects.Python is garbage-collected and dynamically typed. It supports a variety of programming paradigms, including structured (especially procedural), object-oriented, and functional programming. Because of its extensive standard library, it is frequently referred to as a "batteries included" language.

VII. MODULES USED

We used the Python programming language to complete the work. We used the python module as follows:

pyttsx3 sapi5 API- to convert the output string to speech in order to provide a better user experience. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3.Engine instance. During construction, the engine initializes a pyttsx3.driver.DriverProxy object responsible for loading a speech engine driver implementation from the pyttsx3.drivers module. After construction, an application uses the engine object to register and unregister event call-backs; produce and stop speech; get and set speech engine properties; and start and stop event loops.

The processing of text into speech is done by an 'engine'. Text to speech engines work through the following steps:

1. **Text analysis**: interpret things like numbers as words (e.g. '1' becomes 'one') and identify segments such as phrases, clauses and sentences
2. **Linguistic Analysis**: such as identifying intonation and duration of speech
3. **Wave form generation**: the symbolic representation of speech is converted into sound.

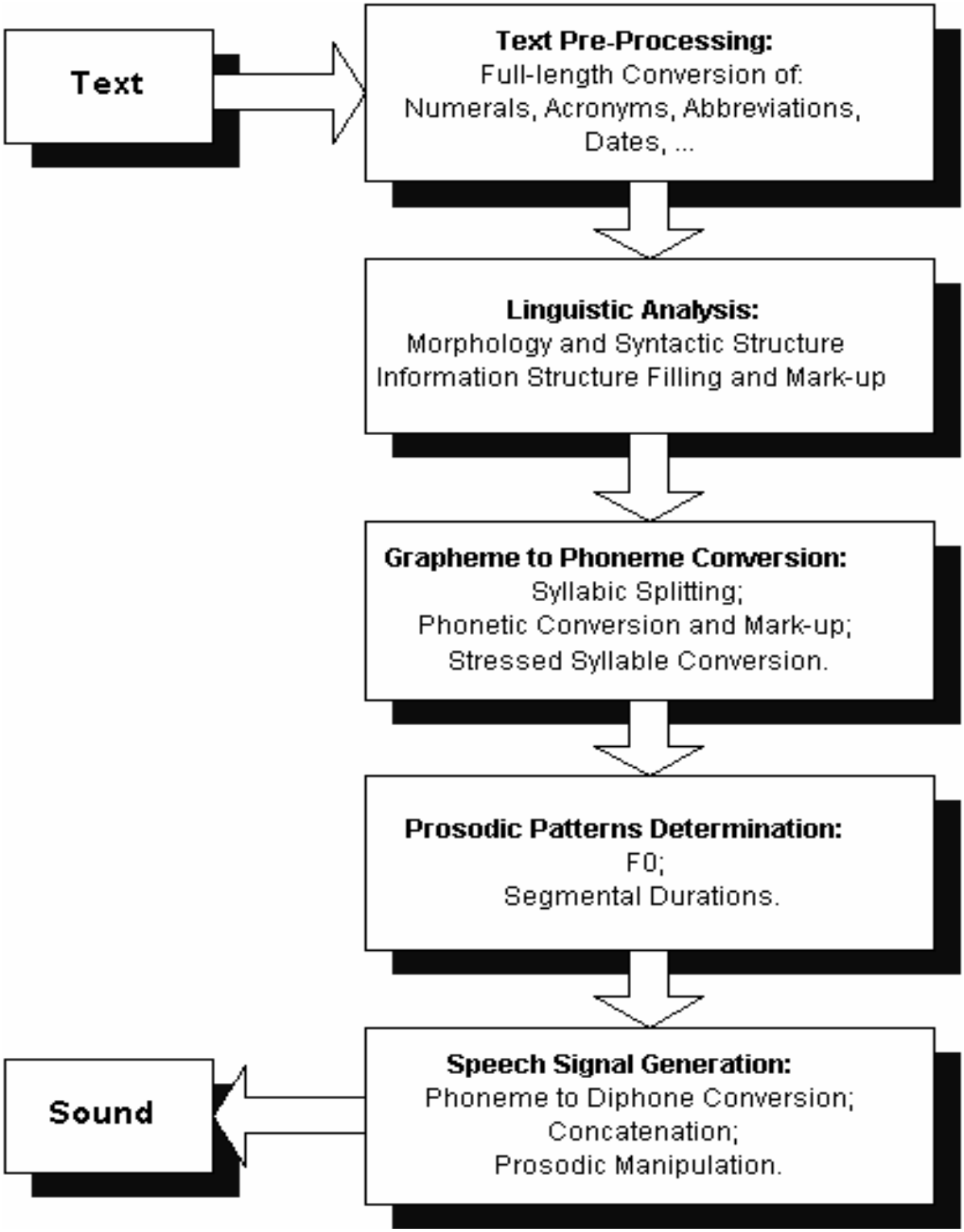


Fig.3. Working of pyttsx3 API

Speech recognition- to recognize user speech and convert them into string. Speech recognition, also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, is a capability which enables a program to process human speech into a written format. While it's commonly confused with voice recognition, speech recognition focuses on the translation of speech from a verbal format to a text one whereas voice recognition just seeks to identify an individual user's voice.

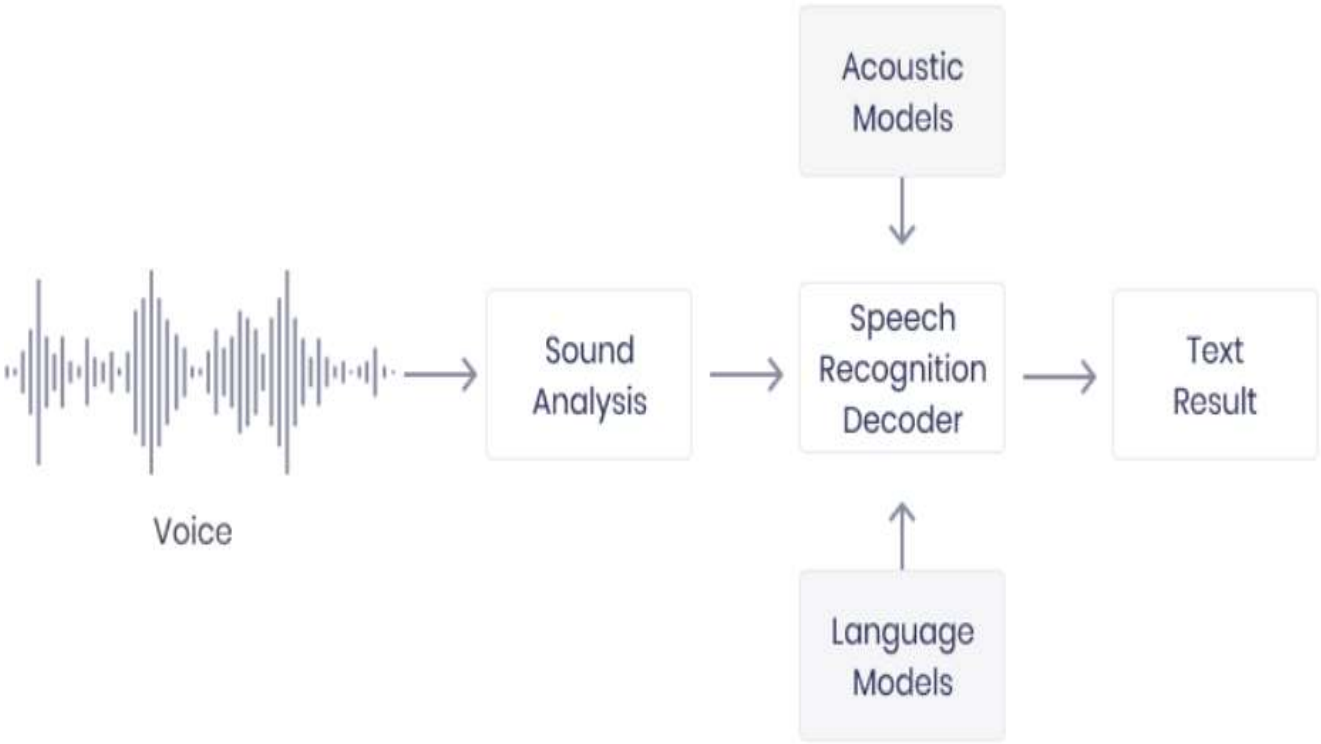


Fig.4. Working of Speech recognition

Datetime- The Python Datetime module provides classes for working with dates and times. These classes offer a variety of functions for working with dates, times, and time intervals. In Python, date and datetime are objects, so when you manipulate them, you are actually manipulating objects rather than strings or timestamps.

Wikipedia- In order to extract data from Wikipedia, we must first install the Python Wikipedia library, which wraps the official Wikipedia API.

Web browser- The web browser module in Python is a useful web browser controller. It provides a high-level interface for users to view Web-based documents.A webbrowser can also be used as a command-line tool. It takes a URL as an argument

and has the following optional parameters: If possible, -n opens the URL in a new browser window, and -t opens the URL in a new browser tab.

os- In Python, the OS module provides functions for interacting with the operating system. Python's standard utility modules include OS. This module provides a portable way to access operating system-specific functionality. Many functions for interacting with the file system are included in the *os* and *os.path* modules.

Random- The Python Random module is a built-in Python module that is used to generate random numbers. Because these are pseudo-random numbers, they are not truly random. This module can be used to generate random numbers, print a random value from a list or string, and so on.

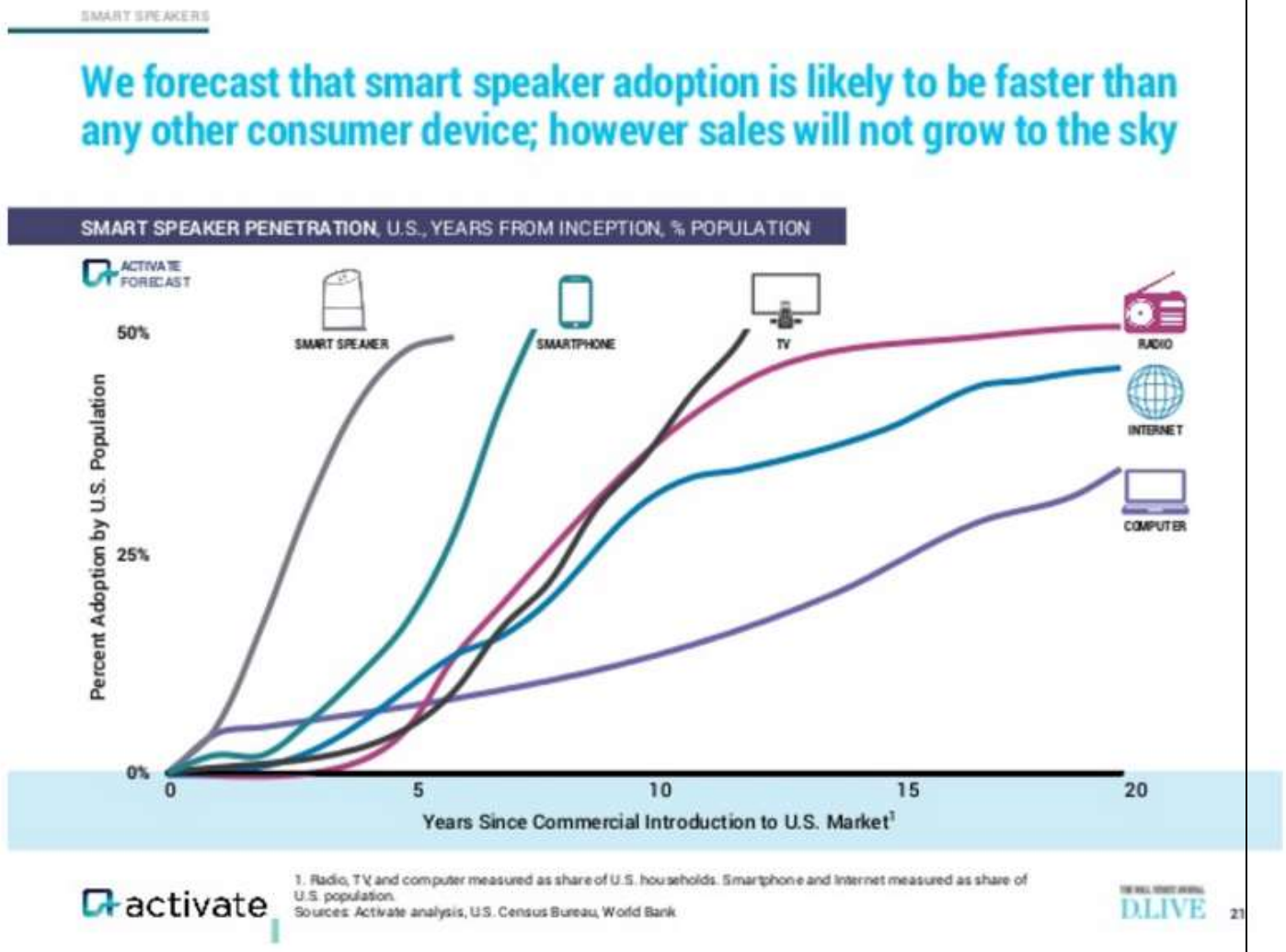
Twilio.rest-. Twilio is a web application programming interface (API) that software developers can use to add communications such as phone calling, messaging, video and two-factor authentication into their Python applications. etc.and we created many functions that will be executed on the user's command.

```
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import os
import random
import smtpd
from twilio.rest import Client
import phonenumbers
```

Fig.5. Modules used in voice assistant

VIII. FUTURE SCOPE

The future of voice assistants can be defined in a variety of ways. There is still a lot to be done in terms of interoperability and integration with other devices and systems. Another dimension would be the redundant use of wake words at the start of each command. Individuality of findings is also a major issue. But, for all intents and purposes, these technologies have a bright future.



Source:Activate Forecast

Fig.6. Popularity over the time

IX. PROJECT ARCHITECTURE

DATAFLOW DIAGRAM:

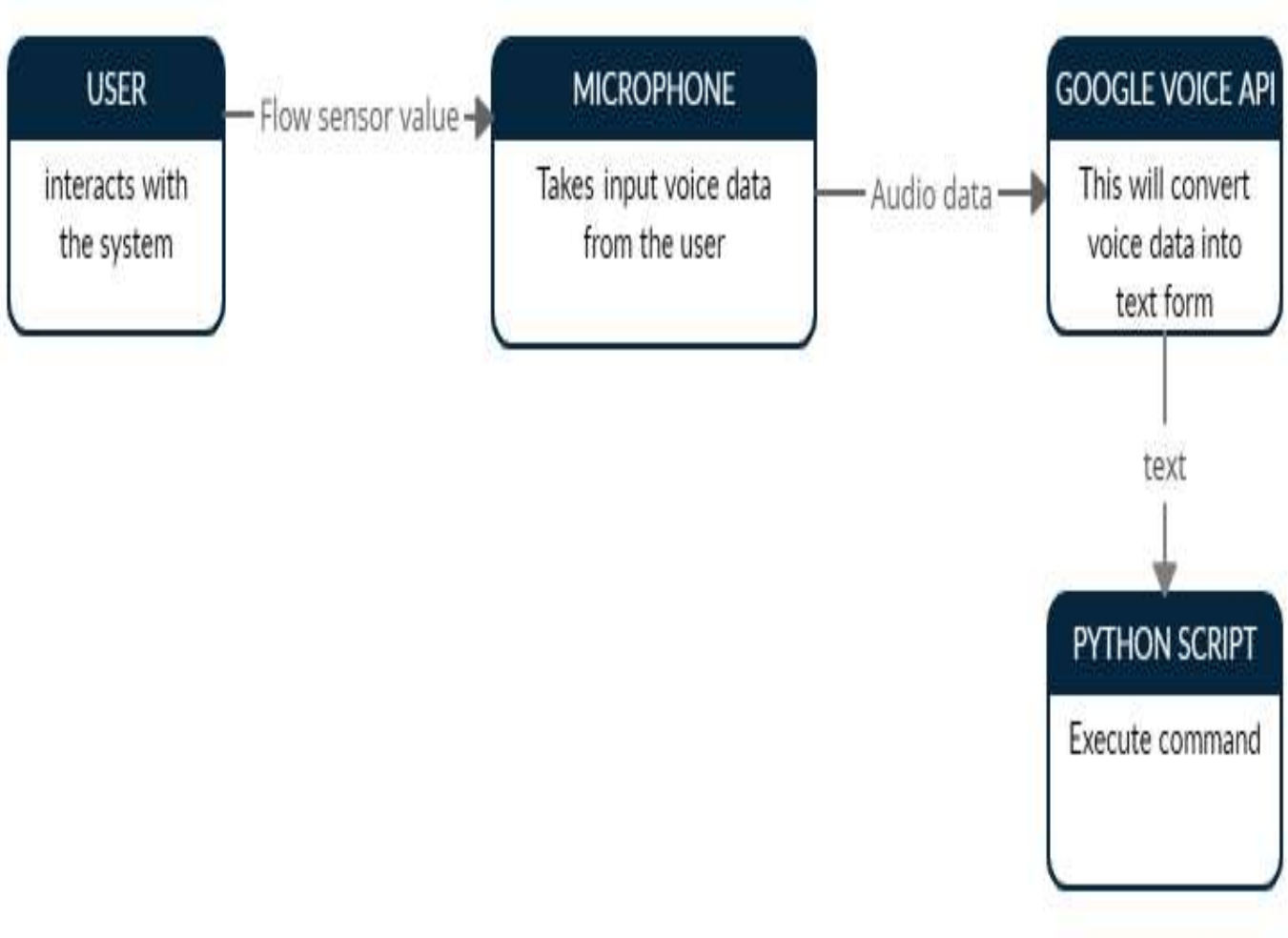


Fig7. DFD of voice assistant

In the above figure I have drawn the dataflow diagram of the voice assistant system in which user voice data flows to microphone, then from microphone audio data is transferred to Google's API ,which then understand the command , then executes the order accordingly.

UML Use Case Diagram:

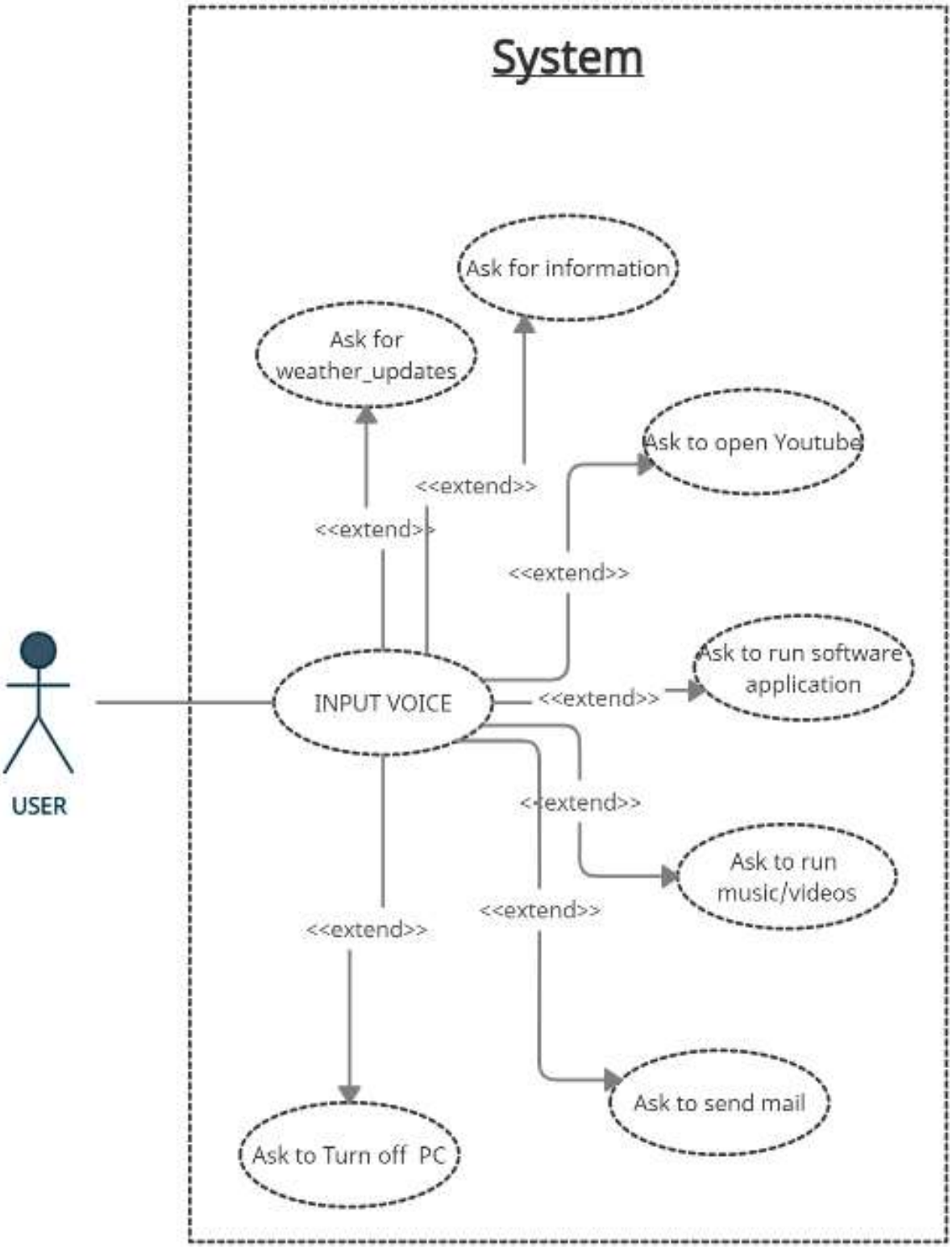


Fig.8. USECASE Diagram of voice assistant

FLOWCHART

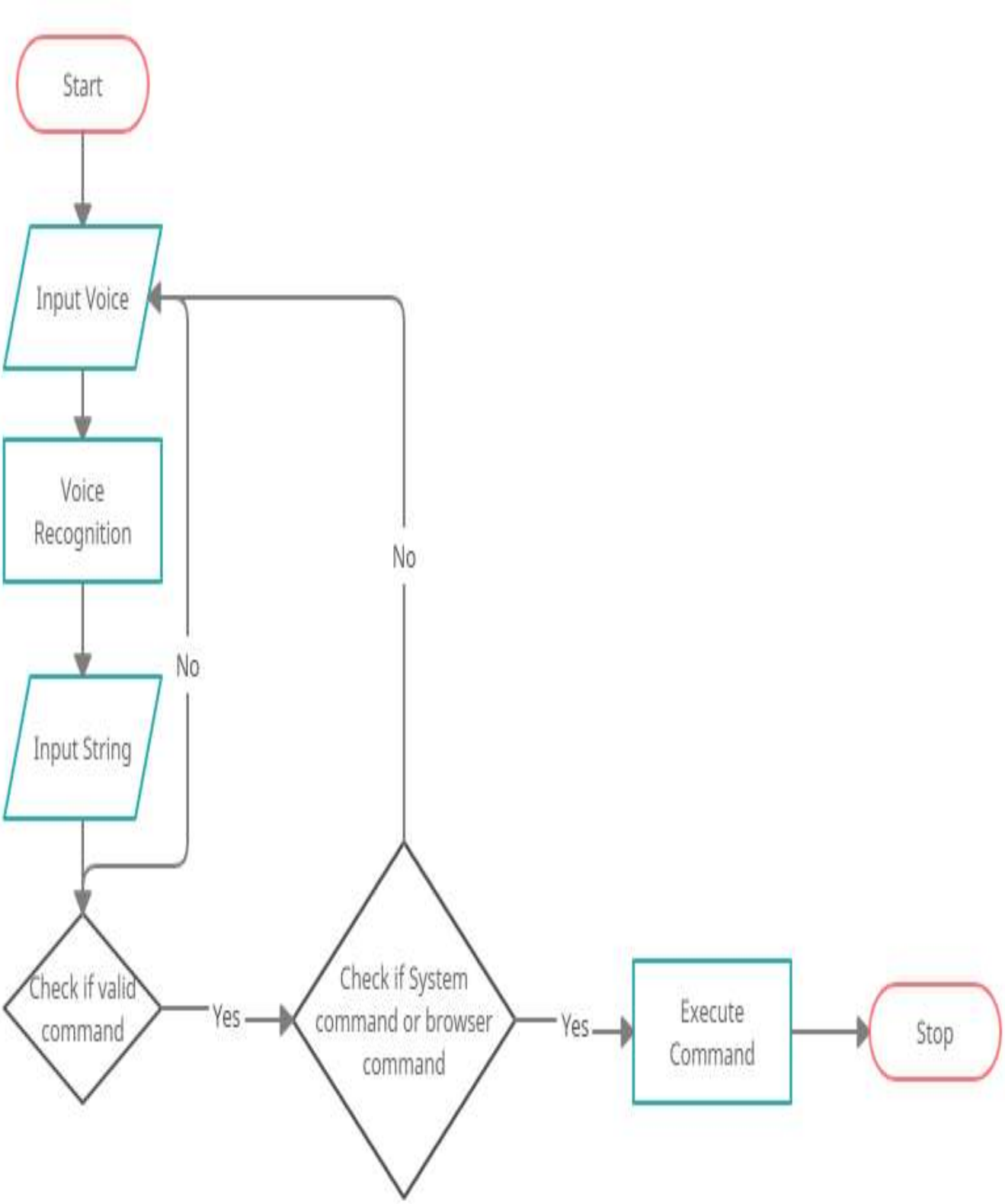


Fig.9. Flowchart of voice assistant

X. RESULT AND CONCLUSION

```
def takeCommand():      #takes microphone input from user and give string output
    r=sr.Recognizer()  #helps to recognize audio
    with sr.Microphone() as source:
        print("LISTENING.....")
        r.energy_threshold=10000 #mini energy to grab  the sound
        r.pause_threshold=1 #sec of non speaking audio before phrase is considered complete(max GAP in speaking to take it )
        audio=r.listen(source)

    try:
        print("RECOGNIZING.....")
        query=r.recognize_google(audio,language="en-in")
        print(f"Master Vishal said->{query}\n")
    except Exception as e:
        print("say that again please!")
        return("None") #print None if any problem occure
    return query
```

Fig.10. Code of voice assistant speech recognition

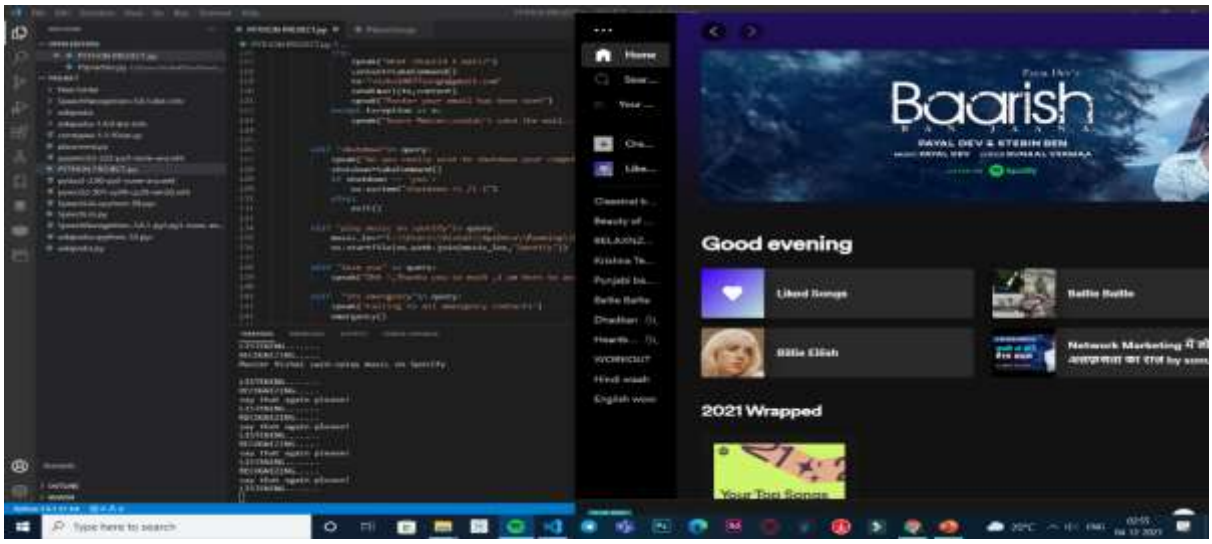


Fig.11. Code of voice assistant

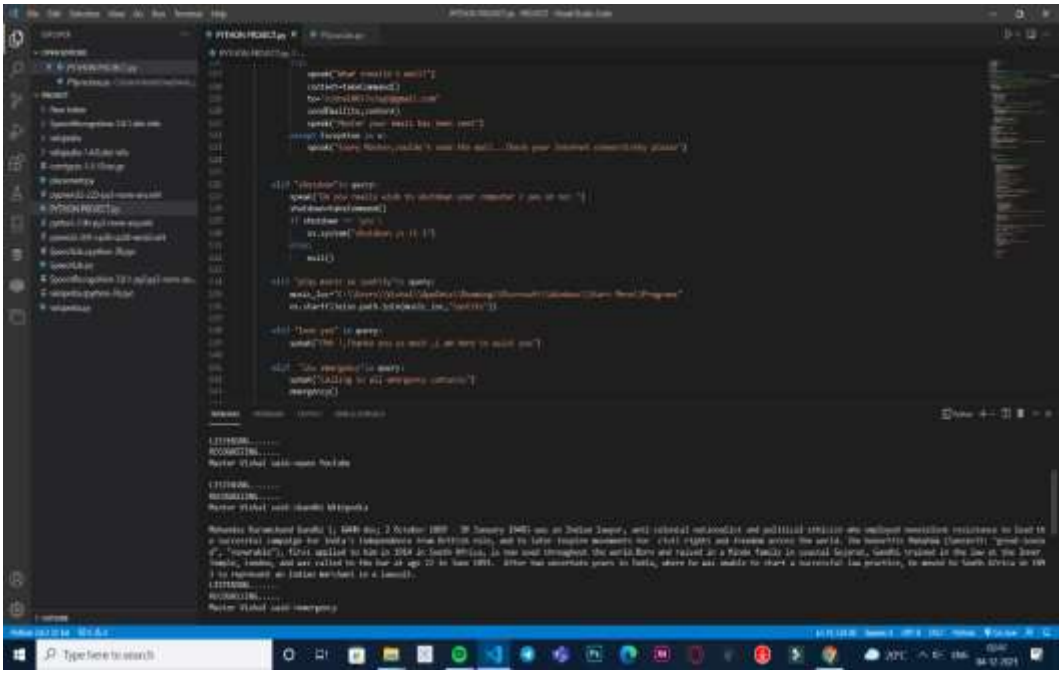


Fig.12. Code EXECUTION

Fig.13. Assistant calling during emergency command

References

[1] J. B. Allen, "From lord rayleigh to shannon: How do humans decode speech," *International Conference on Acoustics, Speech and Signal Processing,* 2002.

[2] M. p. v. B. M. k. S. &. N. L. Mhamunkarr, "Android application to get word meaning through voice," *International journal of Advance Research in Computer Engineering & Technology (IJARCET),* vol. 2, no. 2, pp. 572-572, 2013.

[3] L. R. R. B S Atal, "A pattern recognition approach to voiced unvoiced-silence classification with applications to speech recognition," *Acoustics, Speech and Signal Processing,IEEE Transactions,* vol. 24, no. 3, pp. 201-2012, 1976.

[4] V. C Radha, "A review on speech recognition challenges and approaches," *doaj.org,* vol. 2, no. 1, pp. 1-7, 2012.

[5] S. T L Yu, "An open -Source Based Speech Recognition Android Application for Helping Handicapped Students Writing Programs," *International Conference on Wireless Networks (ICWN),The Steering Committee of the International Conferennce on Wireless networks.,* pp. 71-71, 2015.

[6] K. O. B K Shibwabo, "The application of realtime voice recognition to control critical mobile device operations," 2015.

[7] A. W. T Schultz, "Language independent and language adaptive acoustic modeling for speech recognition," *Speech Communication,* vol. 35, no. 1, pp. 31-51, 2001.

[8] H. M.Bapat, "A paradigm-based finite state morphological analyzer for marathi," *Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP),* pp. 26-34, 2010.

[9] P. S. K. N. D. L. Sutar Shekhar, "International Journal of Advance research in computer Science and Management Studies," *International Journal of Advance research in computer Science and Management Studies,* 2015.