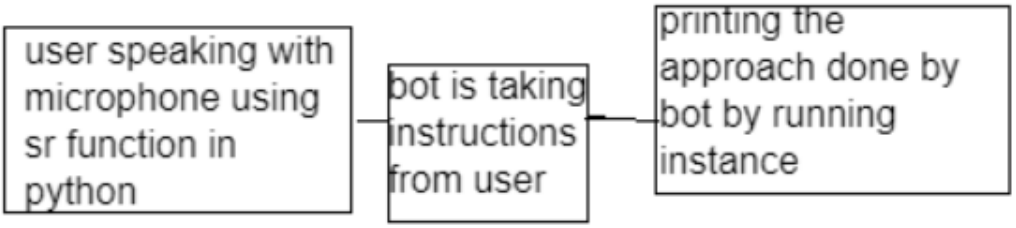
ABHIGNA -Artificial Based Human Interface Geo Neutral Architecture

Always having a thought of creating a bot wi9tht the help of ai and youtube support used a basic prototype that understands user instructions and commands the user

Diagramatic approach



Code

```
import pyttsx3
import pywhatkit
import datetime                                                    ###importing of libraries
import wikipedia
import pyjokes

listener = sr.Recognizer()
engine = pyttsx3.init()                          ##initilising the bot and setting the   voice to the
bot
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)


def talk(text):
    engine.say(text)                                    # making the bot the realising the text
    engine.runAndWait()


def take_command():
    try:
        with sr.Microphone() as source:
            print('listening...')
            voice = listener.listen(source)
            command = listener.recognize_google(voice)
            command = command.lower()
            if 'abhi' in command:
```

```
            command = command.replace('abhi', '')
            print(command)
    except:
        pass
    return command

def run_abhi():
    command = take_command()
    print(command)
    if 'play' in command:
        song = command.replace('play', '')
        talk('playing ' + song)
        pywhatkit.playonyt(song)
    elif 'time' in command:
        time = datetime.datetime.now().strftime('%I:%M %p')
        talk('Current time is ' + time)
    elif 'who  is' in command:
        person = command.replace('who is', '')
        info = wikipedia.summary(person, 1)
        print(info)
        talk(info)
    elif 'date' in command:
        talk('sorry, I have a headache')
    elif 'are you single' in command:
        talk('I am in a relationship with wifi')
    elif 'joke' in command:                                    ##instructing the bot
        talk(pyjokes.get_joke())
    else:
        talk('Please say the command again.')


while True:
    run_abhi()
```

Explanation

Importing Libraries and Initializing Components:

- Import the required libraries: speech_recognition, pyttsx3, pywhatkit, datetime, wikipedia, and pyjokes.
- Initialize the speech recognition listener and the text-to-speech engine.
- Set the voice for the text-to-speech engine.

Creating a Text-to-Speech Function:

- Define a function talk(text) to convert input text into speech using the text-to-speech engine.

Creating a Speech Recognition Function:

- Define a function take_command() that captures audio using the microphone and attempts to recognize spoken commands.
- If the command contains "abhi," remove it and return the modified command.

Creating the Main Function:

- Define a function run_abhi() to process recognized commands and provide responses.
- Call the take_command() function to get the recognized command.
- Process different command types:
    - If the command contains "play," play a song on YouTube using pywhatkit.
    - If the command contains "time," get and announce the current time.
    - If the command contains "who is," fetch a summary from Wikipedia about the mentioned person and provide the information.
    - If the command contains "date," respond with a humorous excuse.
    - If the command contains "are you single," respond humorously.
    - If the command contains "joke," generate and tell a joke using pyjokes.
    - For unrecognized commands, request the user to repeat the command.

Main Loop:

- Run an infinite loop using while True.
- Inside the loop, repeatedly call the run_abhi() function to listen for commands and provide responses.

Execution and Interaction:

- When the script is executed, it continually listens for user commands through the microphone.
- It recognizes keywords like "play," "time," "who is," etc., and responds accordingly with actions like playing a song, telling the time, fetching Wikipedia summaries, etc.
- It can also respond humorously to certain queries and generate jokes.

Note:

- The script's functionality is reliant on external services and APIs, such as Google's speech recognition and text-to-speech services, as well as Wikipedia and joke APIs.
- Errors and exceptions during microphone usage or API calls are not extensively handled in this script.

Future predictions
Want to implement in the board and wanted to add geo naturality features