Chapter – 1

INTRODUCTION

INTRODUCTION

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

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

This system is designed to be used efficiently on desktops. Personal assistant software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user. G ONE is effortless to use. Call the wake word 'G ONE' followed by the command. And within seconds, it gets executed.

Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 50% of searches will be via voice by 2020.Virtual assistants are turning out to be smarter than ever. Allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver personalized responses.

This project was started on the premise that there is sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

1.1 BACKGROUND

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 a location and time context. It records user inputs and converts them into commands, and monitors current stack of user tasks to proactively suggest actions while considering any changes in the environment. It also presents information based on the context of the user, as well as filter information to the user based on its learned understanding of the priority of that information.

Supported Tasks

Reminders

Email

Calendar, Google Calendar

Outlook

Evernote

Facebook, LinkedIn

News Feeds

Drawback

Will take some time to put all of the to-do items in – you could spend more time putting the entries in than actually doing the revision.

1.2 OBJECTIVES

Main objective of building personal assistant software (a virtual assistant) is using semantic data sources available on the web, user generated content and providing knowledge from knowledge databases. The main purpose of an intelligent virtual assistant is to answer questions that users may have. This may be done in a business environment, for example, on the business website, with a chat interface. On the mobile platform, the intelligent virtual assistant is available as a call-button operated service where a voice asks the user “What can I do for you?” and then responds to verbal input.

Virtual assistants can tremendously save you time. We spend hours in online research and then making the report in our terms of understanding. G ONE can do that for you. Provide a topic for research and continue with your tasks while G ONE does the research. Another difficult task is to remember test dates, birthdates or anniversaries. It comes with a surprise when you enter the class and realize it is class test today. Just tell G ONE in advance about your tests and she reminds you well in advance so you can prepare for the test.

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

1.3 PURPOSE, SCOPE AND APPILCABILITY

Purpose

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

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

Scope

Voice assistants will continue to offer more ***individualized*** experiences as they get better at differentiating between voices. However, it's not just developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for their specific brand. They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface.

Applicability

The mass adoption of artificial intelligence in users' everyday lives is also fuelling the shift towards voice. The number of IoT devices such as smart thermostats and speakers are giving voice assistants more utility in a connected user's life. Smart speakers are the number one way we are seeing voice being used. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 5 years.

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

Chapter-2

SURVEY OF TECHNOLOGY

Python:

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

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

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

DBpedia:

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

Quepy:

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

Pyttsx:

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for textto-speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

Speech Recognition:

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

SQLite:

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

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

Chapter-3

REQUIREMENT AND ANALYSIS

REQUIREMENT AND ANALYSIS

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

3.1 Problem definition

Usually, user needs to manually manage multiple sets of applications to complete one task. For example, a user trying to make a travel plan needs to check for airport codes for nearby airports and then check travel sites for tickets between combinations of airports to reach the destination. There is need of a system that can manage tasks effortlessly.

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

When a virtual assistant is not able to answer questions accurately, it’s because it lacks the proper context or doesn’t understand the intent of the question. Its ability to answer questions relevantly only happens with rigorous optimization, involving both humans and machine learning. Continuously ensuring solid quality control strategies will also help manage the risk of the virtual assistant learning undesired bad behaviors. They require large amount of information to be fed in order for it to work efficiently.

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

3.2 REQUIREMENT SPECIFICATION:

Personal assistant software is required to act as an interface into the digital world by understanding user requests or commands and then translating into actions or recommendations based on agent’s understanding of the world.

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

Voice recognition software enables hands free use of the applications, lets users to query or command the agent through voice interface. This helps users to have access to the agent while performing other tasks and thus enhances value of the system itself. G ONE also have ubiquitous connectivity through Wi-Fi or LAN connection, enabling distributed applications that can leverage other APIs exposed on the web without a need to store them locally.

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

Providing information such as weather, facts from e.g. Wikipedia etc.

Set an alarm or make to-do lists and shopping lists.

Remind you of birthdays and meetings.

Play music from streaming services such as Saavn and Gaana.

Play videos, TV shows or movies on televisions, streaming from e.g. Netflix or Hotstar.

Book tickets for shows, travel and movies.

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. Five 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 G ONE, 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 read out problems for system and get answers.

Economic 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, G ONE won’t cost too much.

Organizational feasibility: This shows the management and organizational structure of the project. This project is not 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.

Cultural feasibility: It deals with compatibility of the project with cultural environment. Virtual assistant is built in accordance with the general culture. The project is named G ONE so as to represent Indian culture without undermining local beliefs.

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.

3.3 HARDWARE AND SOFTWARE REQUIREMENTS

The software is designed to be light-weighted so that it doesn’t be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software requirement for virtual assistant.

Hardware:

Pentium-pro processor or later.

RAM 512MB or more.

Software:

Windows 7(32-bit) or above.

Python 2.7 or later

Chrome Driver

Selenium Web Automation

SQLite

Chapter-4

SYSTEM DESIGN

4.1 ER DIAGRAM

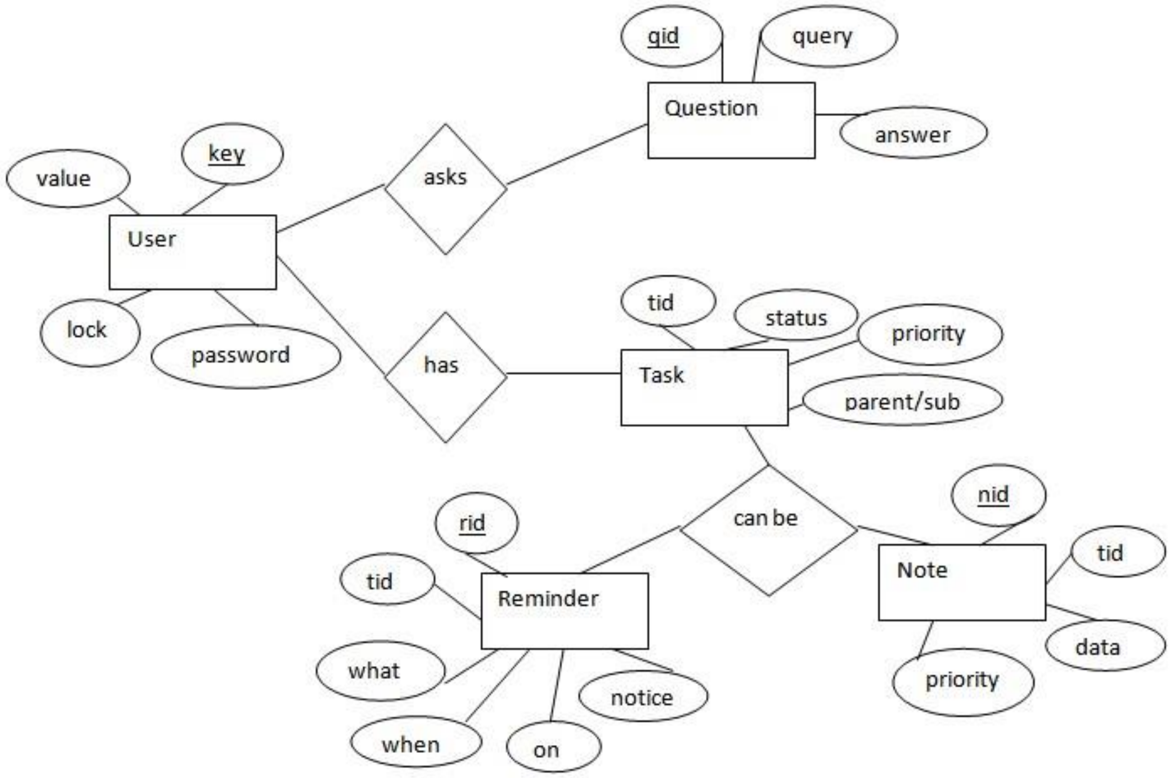


Figure 4.1 ER Diagram

The above diagram shows entities and their relationship for a virtual assistant system. We have a user of a system who can have their keys and values. It can be used to store any information about the user. Say, for key “name” value can be “Jim”. For some keys user might like to keep secure. There he can enable lock and set a password (voice clip).

Single user can ask multiple questions. Each question will be given ID to get recognized along with the query and its corresponding answer. User can also be having n number of tasks. These should have their own unique id and status i.e. their current state. A task should also have a priority value and its category whether it is a parent task or child task of an older task.

4.2 ACTIVITY DIAGRAM

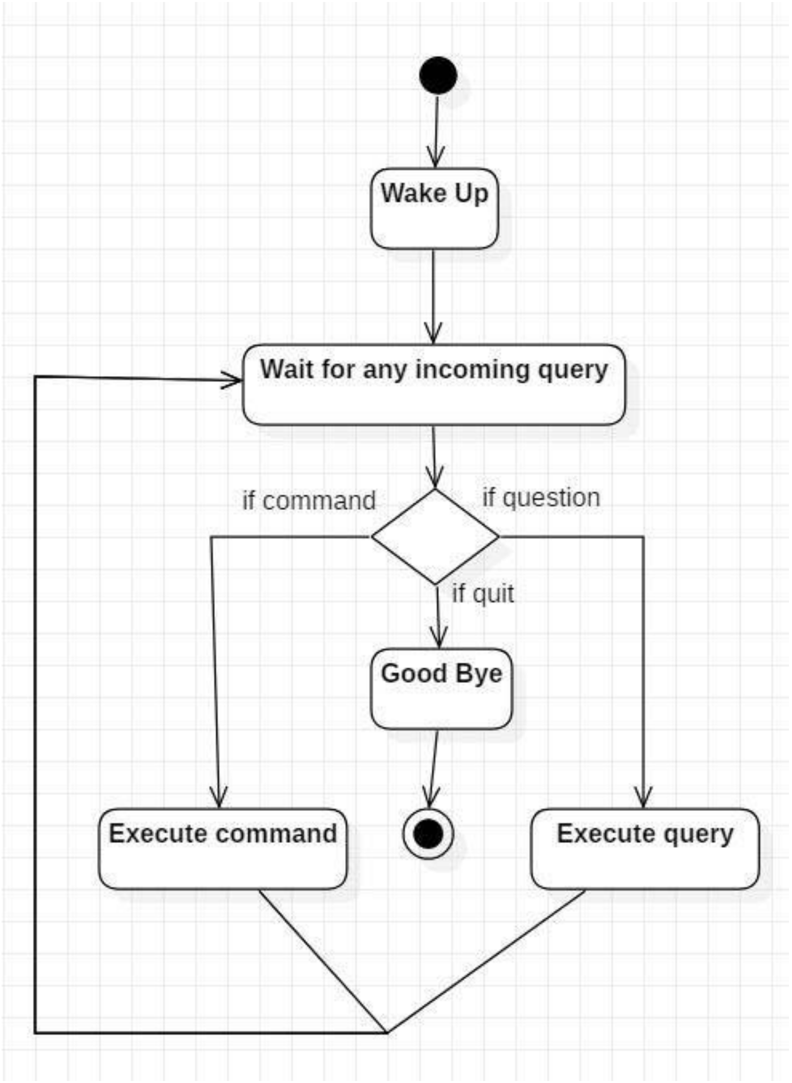


Figure 4.2 Activity Diagram

Initially, the system is in idle mode. As it receives any wake-up call it begins execution. The received command is identified whether it is a questionnaire or a task to be performed. Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits for another command. This loop continues unless it receives quit command. At that moment, it goes back to sleep.

4.3 CLASS DIAGRAM

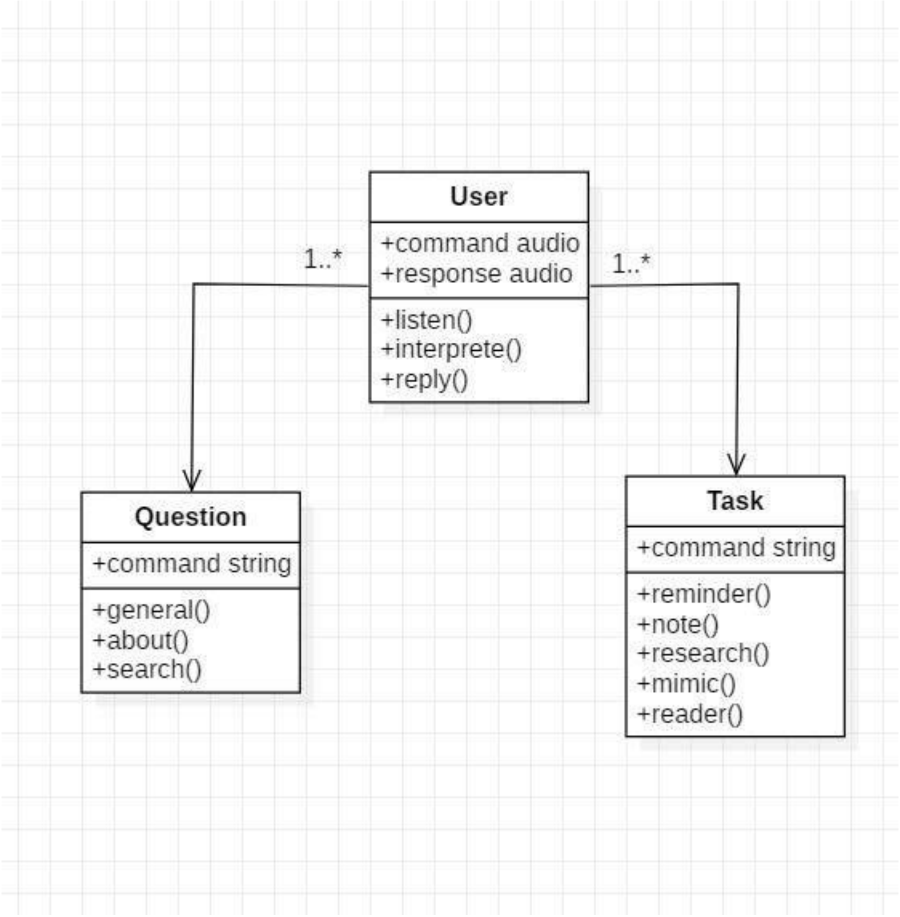


Figure 4.3 Class Diagram

The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly. Question class has the command in string form as it is interpreted by interpret class. It sends it to general or about or search function based on its identification.

The task class also has interpreted command in string format. It has various functions like reminder, note, mimic, research and reader.

4.4 USE CASE DIAGRAM

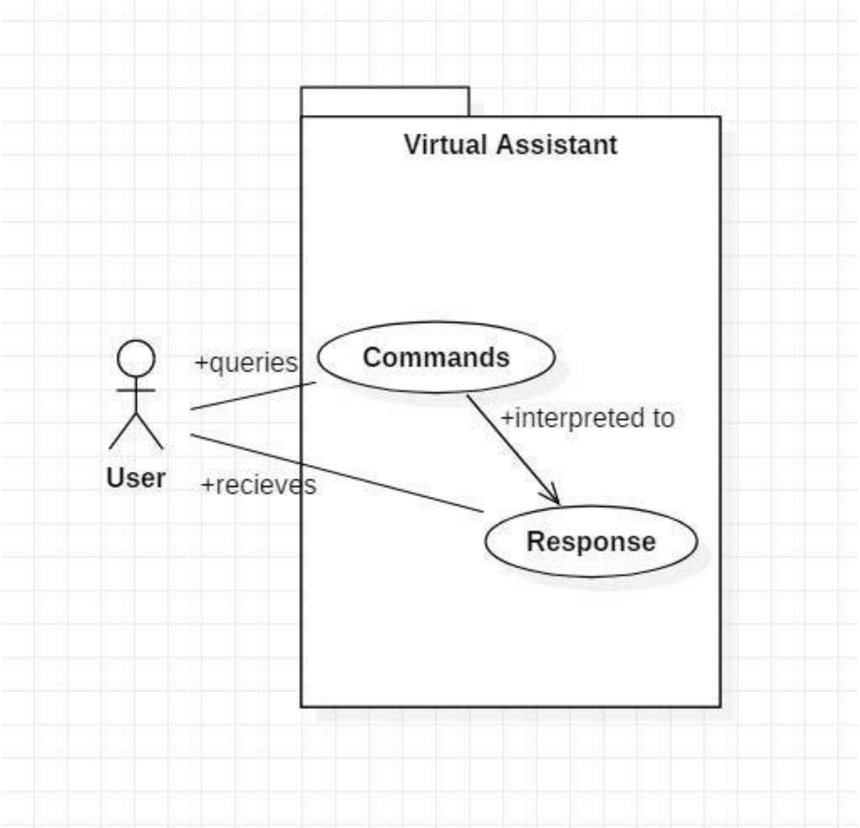


Figure 4.4 Use Case Diagram

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

4.5 SEQUENCE DIAGRAM

4.5.1 Sequence diagram for Query-Response

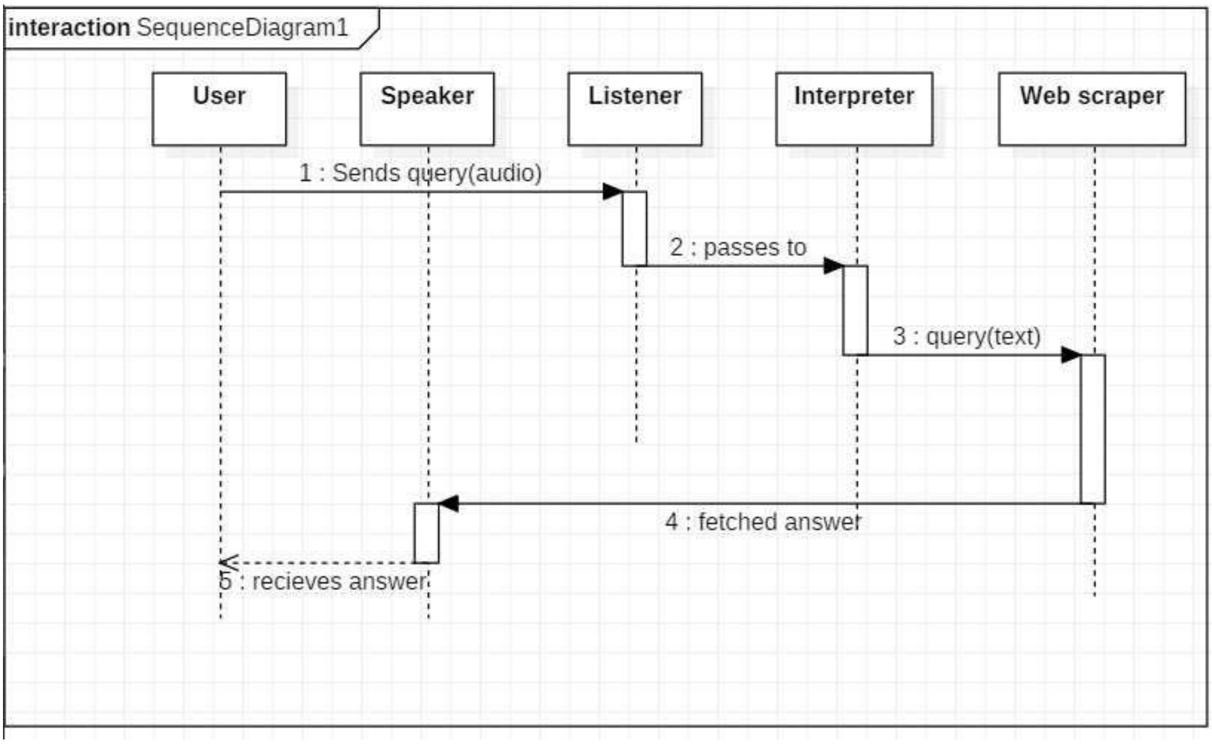


Figure 4.5.1 Sequence diagram for Query-Response

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

4.5.2 Sequence diagram for Task Execution

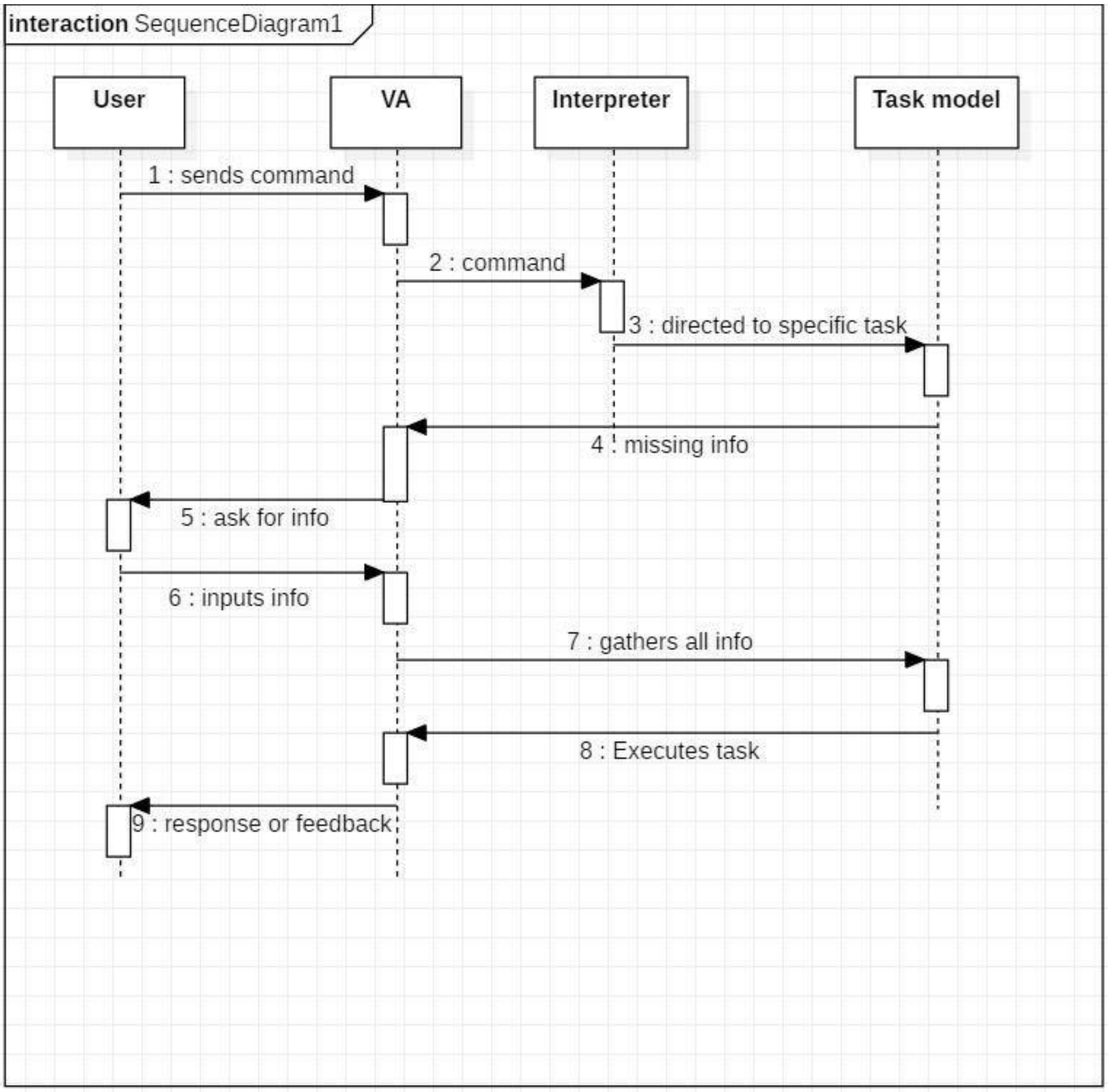


Figure 4.5.2 Sequence diagram for task execution

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

4.6 DATA FLOW DIAGRAM

4.6.1 DFD Level 0 (Context Level Diagram)

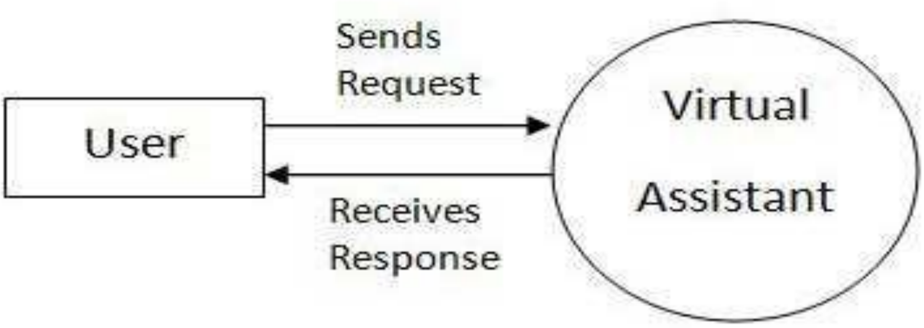


Figure 4.6.1 DFD Level 0

4.6.2 DFD Level 1

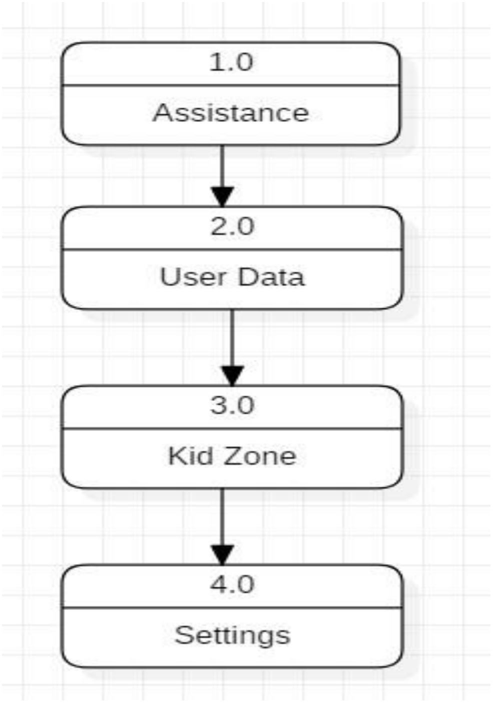


Figure 4.6.2 DFD Level 1

4.6.3 DFD Level 2

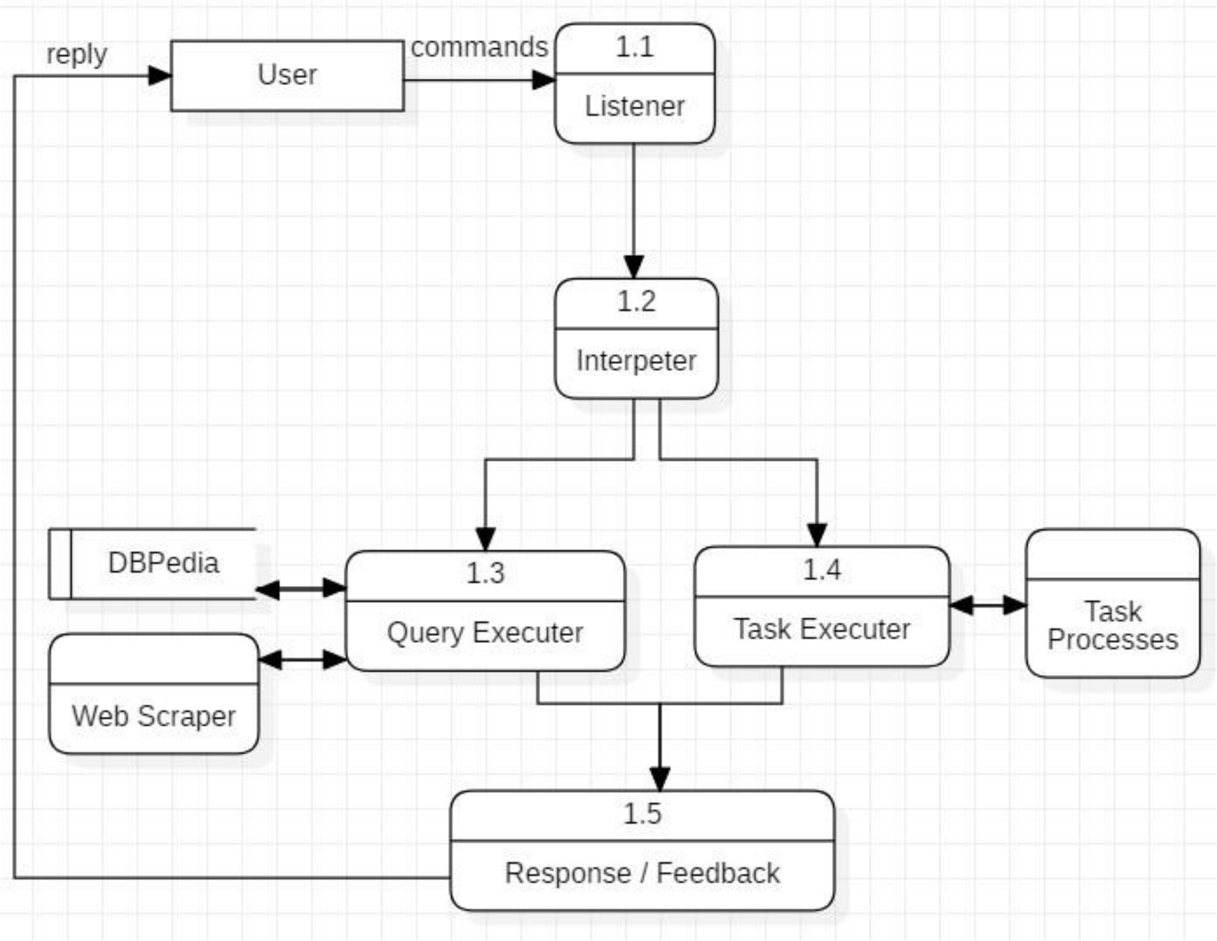


Figure 4.6.3 Data Flow in Assistance

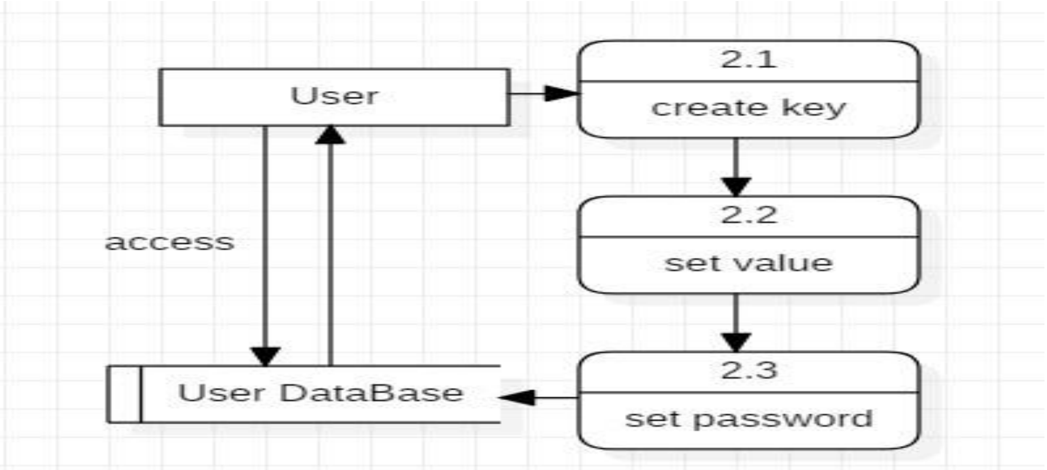


Figure 4.6.4 Managing User Data

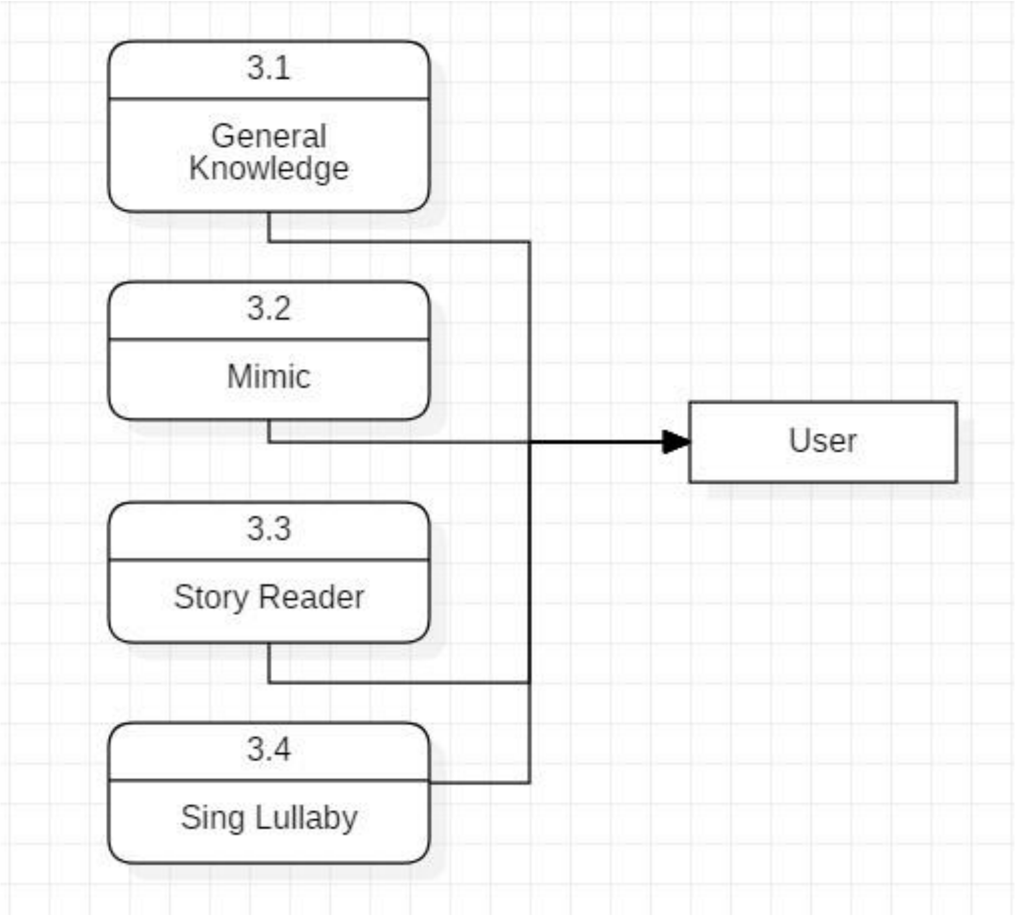


Figure 4.6.5 Data Flow in Kid Zone

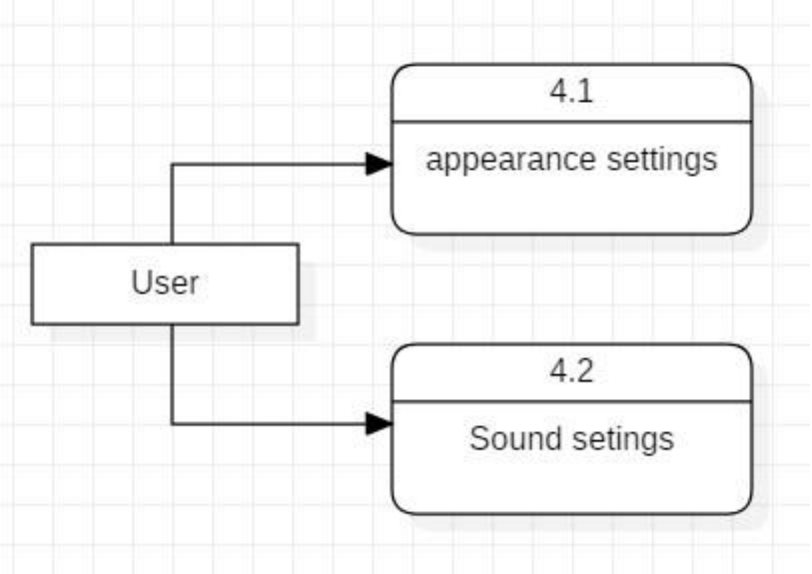


Figure 1.6.6 Settings of virtual Assistant

4.7 COMPONENT DIAGRAM

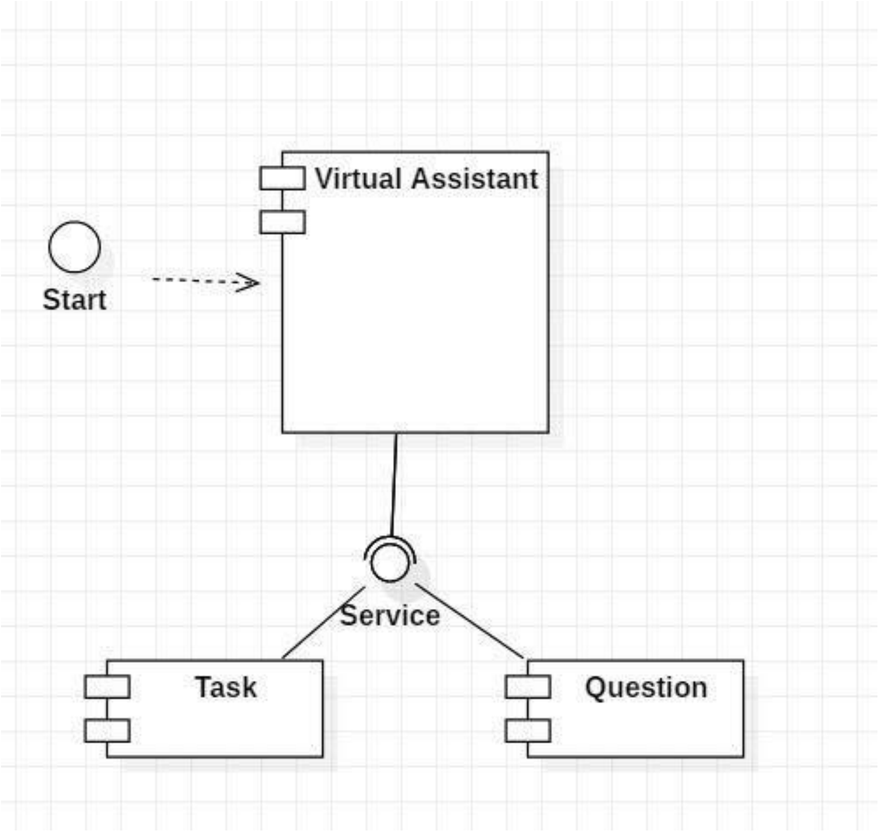


Figure 4.7 Component Diagram

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

4.8 DEPLOYMENT DIAGRAM

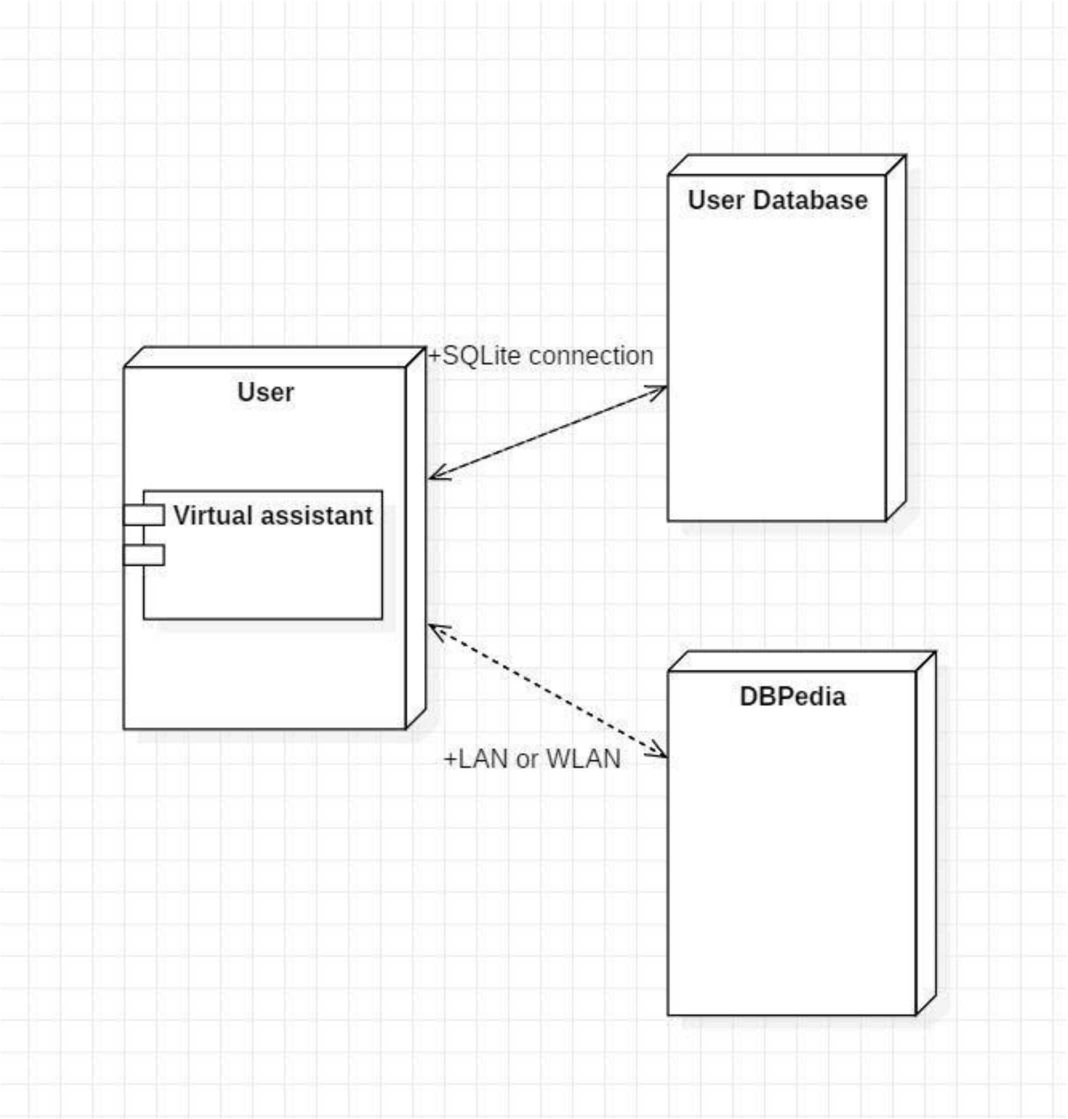


Figure 4.8 Deployment Diagram

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

4.9 DATA DICTIONARY

User

Key	Text
Value	Text
Lock	Boolean
Password	Text

Question

Qid	Integer PRIMARY KEY
Query	Text
Answer	Text

Task

Tid	Integer PRIMARY KEY
Status	Text (Active/Waiting/Stopped)
Level	Text (Parent/Sub)
Priority	Integer

Reminder

Rid	Integer PRIMARY KEY
Tid	Integer FOREIGN KEY
What	Text
When	Time
On	Date
Notify before	Time

Note

Nid	Integer PRIMARY KEY
Tid	Integer FOREIGN KEY
Data	Text
Priority	Integer

4.10 TEST CASE DESIGN

- **Test Case 1**

Test Title: Response Time

Test ID: T1

Test Priority: High

Test Objective: To make sure that the system respond back time is efficient.

Description:

Time is very critical in a voice based system. As we are not typing inputs, we are speaking them. The system must also reply in a moment. User must get instant response of the query made.

- **Test Case 2**

Test Title: Accuracy

Test ID: T2

Test Priority: High

Test Objective: To assure that answers retrieved by system are accurate as per gathered data.

Description:

A virtual assistant system is mainly used to get precise answers to any question asked. Getting answer in a moment is of no use if the answer is not correct. Accuracy is of utmost importance in a virtual assistant system.

- **Test Case 3**

Test Title: Approximation

Test ID: t3

Test priority: Moderate

Test Objective: To check approximate answers about calculations.

Description:

There are times when mathematical calculation requires approximate value. For example, if someone asks for value of PI the system must respond with approximate value and not the accurate value. Getting exact value in such cases is undesirable.

Note: There might include a few more test cases and these test cases are also subject to change with the final software development.

Chapter -5

Introduction To Main Library

5.1 pyttsx3 library:

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. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech. The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows. It supports three TTS

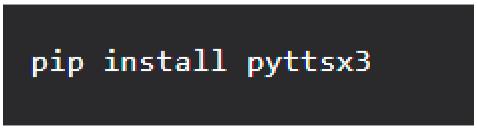
engines:

sapi5 – SAPI5 on Windows

nsss – NSSpeechSynthesizer on Mac OS X

espeak – eSpeak on every other platform

Installation To install the pyttsx3 module, first of all, you have to open the terminal and write



```
pip install pyttsx3
```

5.2 OS library module:

The OS module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system-dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.

For Import:

```
import os
```

5.3 datetime module:

In Python, date and time are not a data type of their own, but a module named datetime can be imported to work with the date as well as time. Python Datetime module comes built into Python, so there is no need to install it externally.

The Date Time module is categorized into 6 main classes –

date – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.

time – An idealized time, independent of any particular day, assuming that every day has exactly 24*60*60 seconds. Its attributes are hour, minute, second, microsecond, and tzinfo.

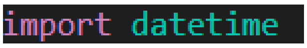
datetime – It's a combination of date and time along with the attribute's year, month, day, hour, minute, second, microsecond, and tzinfo.

timedelta – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.

tzinfo – It provides time zone information objects.

timezone – A class that implements the tzinfo abstract base class as a fixed offset from the UTC

For import this module:



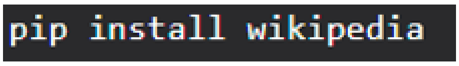
```
import datetime
```

5.4 Wikipedia module:

The Internet is the single largest source of information, and therefore it is important to know how to fetch data from various sources. And with Wikipedia being one of the largest and most popular sources for information on the Internet. Wikipedia is a multilingual online encyclopaedia created and maintained as an open collaboration project by a community of volunteer editors using a wiki-based editing system. In this article, we will see how to use Python's Wikipedia module to fetch a variety of information from the Wikipedia website.

Installation:

In order to extract data from Wikipedia, we must first install the Python Wikipedia library, which wraps the official Wikipedia API. This can be done by entering the command below in your command prompt or terminal



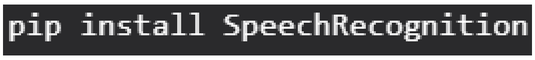
```
pip install wikipedia
```

Syntax: Wikipedia.Summary (title, sentences)

5.5 speech recognition module:

Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc. This article aims to provide an introduction on how to make use of the Speech Recognition and pyttsx3 library of Python.

Installation required:



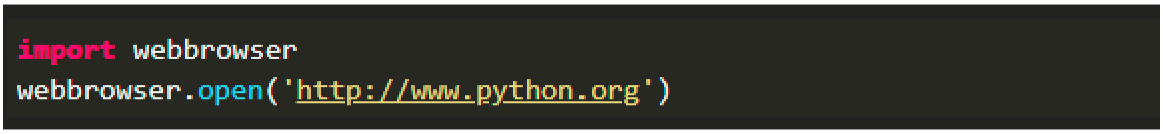
```
pip install SpeechRecognition
```

5.6 web browser module:

In Python, web browser module is a convenient web browser controller. It provides a high-level interface that allows displaying Web-based documents to users.

Web browser can also be used as a CLI tool. It accepts a URL as the argument with the following optional parameters: -n opens the URL in a new browser window, if possible, and -t opens the URL in a new browser tab.

The web browser module can be used to launch a browser in a platform-independent manner as shown below:



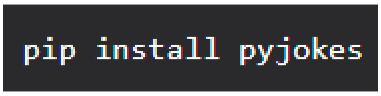
```python
import webbrowser
webbrowser.open('http://www.python.org')
```

5.7 pyjokes module

Python supports creation of random jokes using one of its libraries. Let us explore it a little more, Pyjokes is a python library that is used to create one-line jokes for programmers. Informally, it can also be referred as a fun python library which is pretty simple to use. Let us see how you can actually use it to perform the required task

Installation:

You can simply install it using pip with the following command:



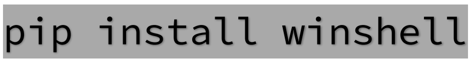
```
pip install pyjokes
```

5.8winshell module:

The winshell module is a light wrapper around the Windows shell functionality. It includes convenience functions for accessing special folders, for using the shell's file copy, rename & delete functionality, and a certain amount of support for structured storage.

Installation:

You can simply install it using pip with the following command:



```
pip install winshell
```

5.9 Play sound module:

In this article, we will see how to play sound in Python using some of the most popular audio libraries. We will learn about the various methods for playing sound.

The playsound module contains only a single function named playsound().

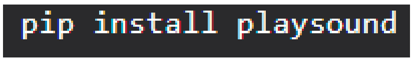
It requires one argument: the path to the file with the sound we have to play. It can be a local file, or a URL.

There’s an optional second argument, block, which is set to True by default. We can set it to False for making the function run asynchronously.

It works with both WAV and MP3 files.

Installation:

You can simply install it using pip with the following command:



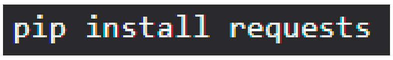
```
pip install playsound
```

5.10 Requests module:

Requests library is one of the integral parts of Python for making HTTP requests to a specified URL. Whether it be REST APIs or Web Scrapping, requests is must to be learned for proceeding further with these technologies. When one makes a request to a URI, it returns a response. Python requests provides inbuilt functionalities for managing both the request and response.

Installing Requests

Requests installation depends on type of operating system on eis using, the basic command anywhere would be to open a command terminal and run,



```
pip install requests
```

Chapter-6

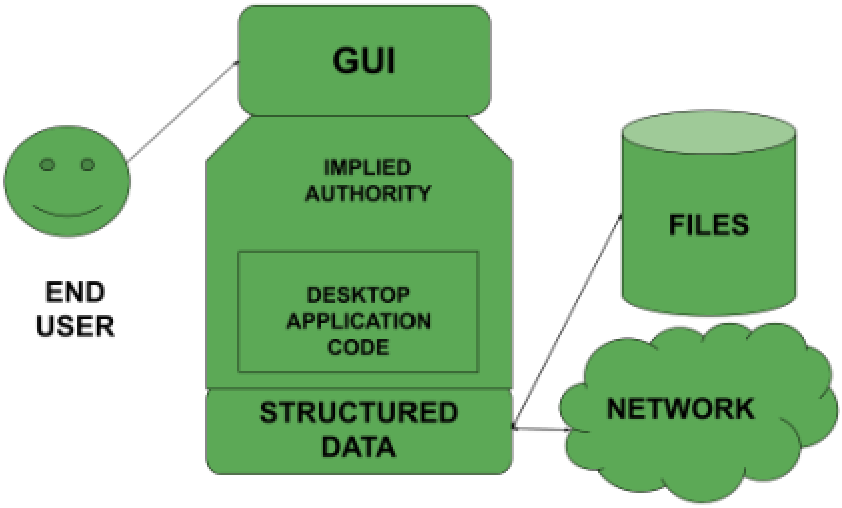
Introduction of GUI

6.1 What is GUI:

The graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation.

6.2 History of GUI

Earlier, there was no GUI so people used to interact with the command-line interface (CLI). The CLI was not that friendly to use and the end-user was not familiar with all the commands. so to bridge this gap, GUI was introduced. The main aim of the GUI was to make the applications much more user-friendly. People love when the task which they want to perform gets done easily and in an efficient manner.GUI stresses one of the most important aspects which is “ease of use”.



6.3 Characteristics

As mentioned above they are user-friendly i.e very easy to use.

A GUI consists of different characteristics such as Menu, Tabs, Pointers and many more kinds of stuff

The icons represent on the user interface represents the software or the file or some application required on the screen.

6.4 Description of GUI

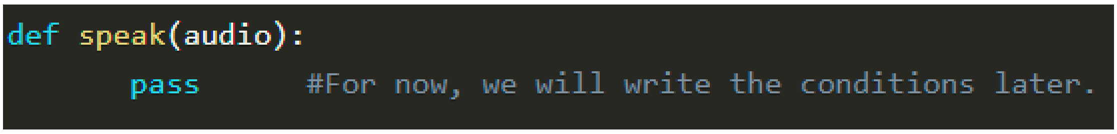
We have included gui also in our virtual assistant.it will run as animation on screen and it will looks cool and it will make assistant attractive. you can also do chances as you want if you have a knowledge of programming language.

Chapter – 7

Speak and Take command

7.1 Speak Function:

The first and foremost thing for an A.I. assistant is that it should be able to speak. To make our G-ONE talk, we will make a function called speak (). This function will take audio as an argument, and then it will pronounce it.



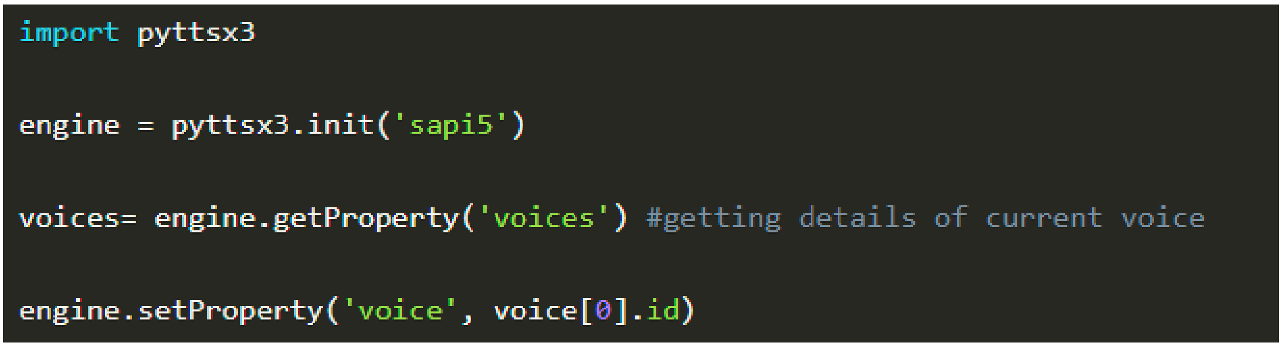
```
def speak(audio):
       pass      #For now, we will write the conditions later.
```

Now, the next thing we need is audio. We must supply audio so that we can pronounce it using the speak () function we made. We are going to install a module called pyttsx3.

7.2What is pyttsx3?

A python library that will help us to convert text to speech. In short, it is a text-to-speech library.It works offline, and it is compatible with Python 2 as well as Python 3.

Usage:



```
import pyttsx3

engine = pyttsx3.init('sapi5')

voices= engine.getProperty('voices') #getting details of current voice

engine.setProperty('voice', voice[0].id)
```

7.3What is sapi5:

Microsoft developed speech API.

Helps in synthesis and recognition of voice.

7.4 What is Voice Id?

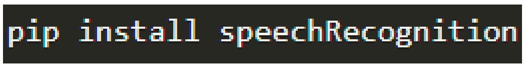
Voice id helps us to select different voices.

voice[0].id = Male voice

voice[1].id = Female voice

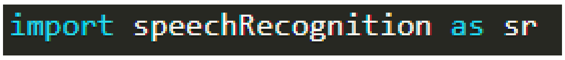
7.5 Take Command:

The next most important thing for our A.I. assistant is that it should take command with the help of the microphone of the user's system. So, now we will make a takeCommand() function. With the help of the takeCommand() function, our A.I. assistant will return a string output by taking microphone input from the user.Before defining the takeCommand() function, we need to install a module called speechRecognition. Install this module by:



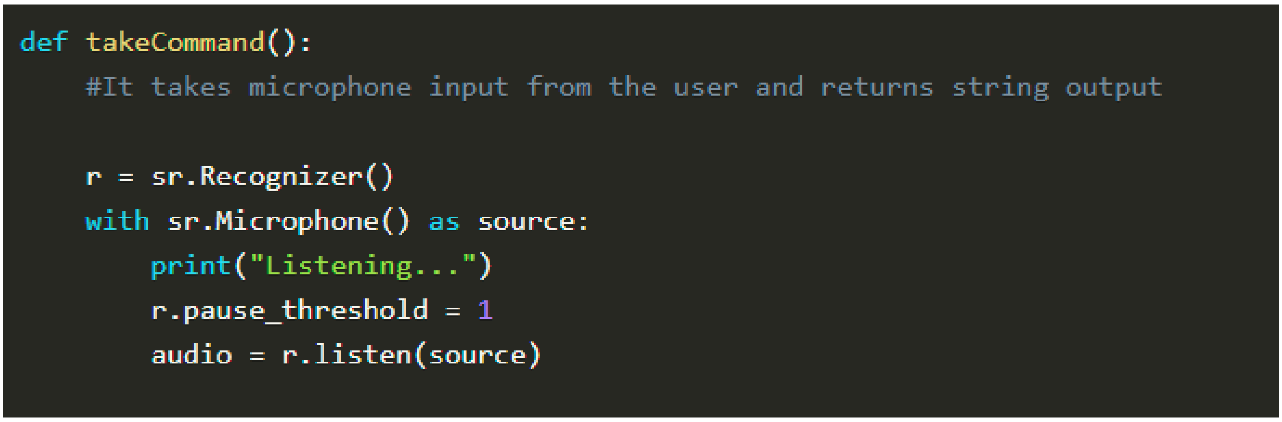
```
pip install speechRecognition
```

After successfully installing this module, import this module into the program by writing an import statement.



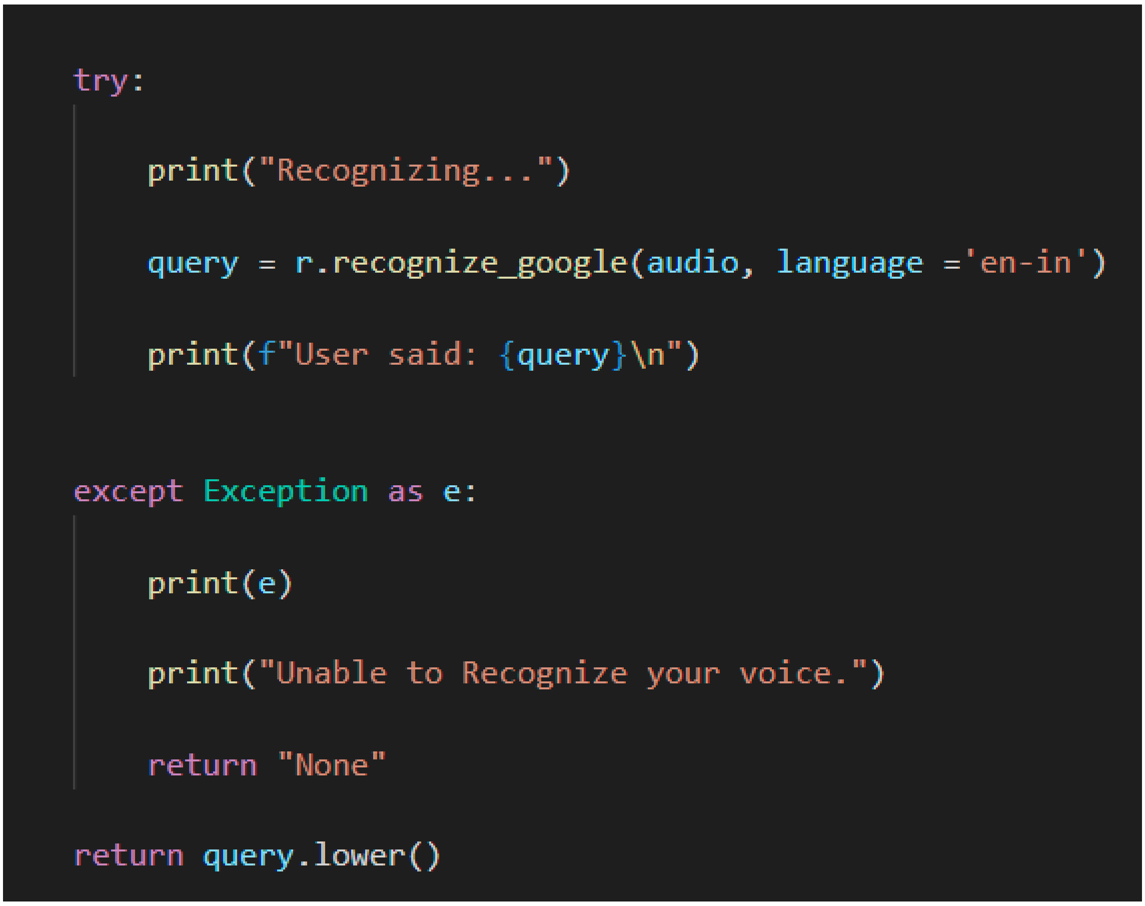
```
import speechRecognition as sr
```

Let's start coding the take Command() function :



```
def takeCommand():
    #It takes microphone input from the user and returns string output

    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)
```

We have successfully created our take Command() function. Now we are going to add a try and except block to our program to handle errors effectively.



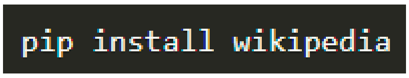
```
try:

    print("Recognizing...")

    query = r.recognize_google(audio, language ='en-in')

    print(f"User said: {query}\n")


except Exception as e:

    print(e)

    print("Unable to Recognize your voice.")

    return "None"

return query.lower()
```

Chapter-8

Features of VIRTUAL ASSISTANT

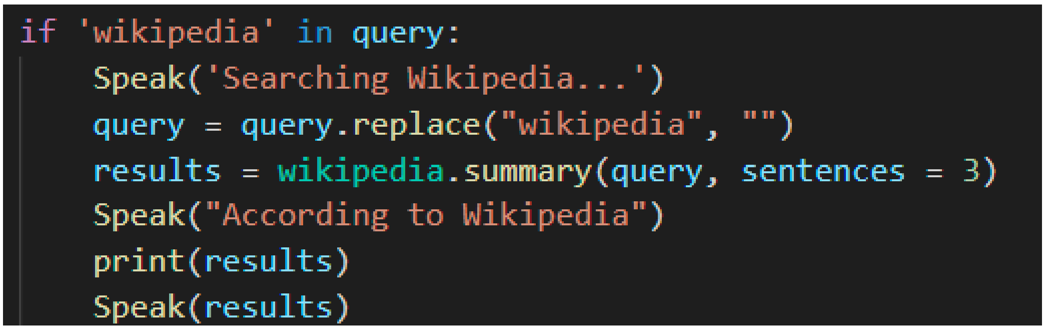
8.1 Wikipedia:

To do Wikipedia searches, we need to install and import the Wikipedia module into our program. Type the below command to install the Wikipedia module:



```
pip install wikipedia
```

After successfully installing the Wikipedia module, import it into the program by writing an import statement.

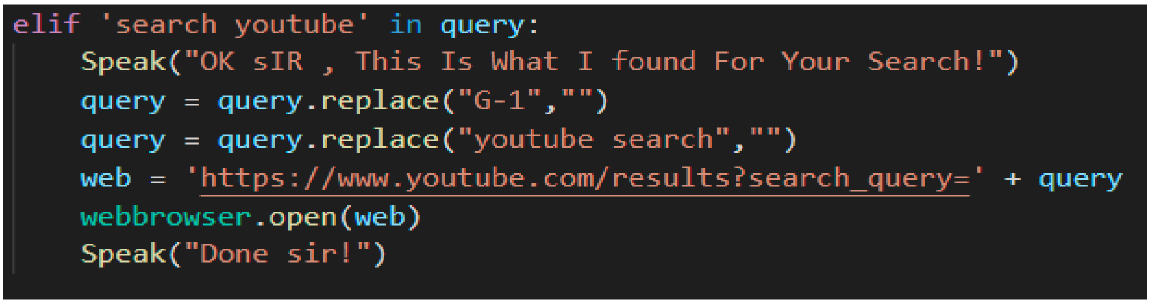


```
if 'wikipedia' in query:
    Speak('Searching Wikipedia...')
    query = query.replace("wikipedia", "")
    results = wikipedia.summary(query, sentences = 3)
    Speak("According to Wikipedia")
    print(results)
    Speak(results)
```

n the above code, we have used an if statement to check whether Wikipedia is in the user's search query or not. If Wikipedia is found in the user's search query, then three sentences from the summary of the Wikipedia page will be converted to speech with the speak function's help.

8.2 Open YouTube

To open any website, we need to import a module called **web browser**. It is an in-built module, and we do not need to install it with a pip statement; we can directly import it into our program by writing an import statement.

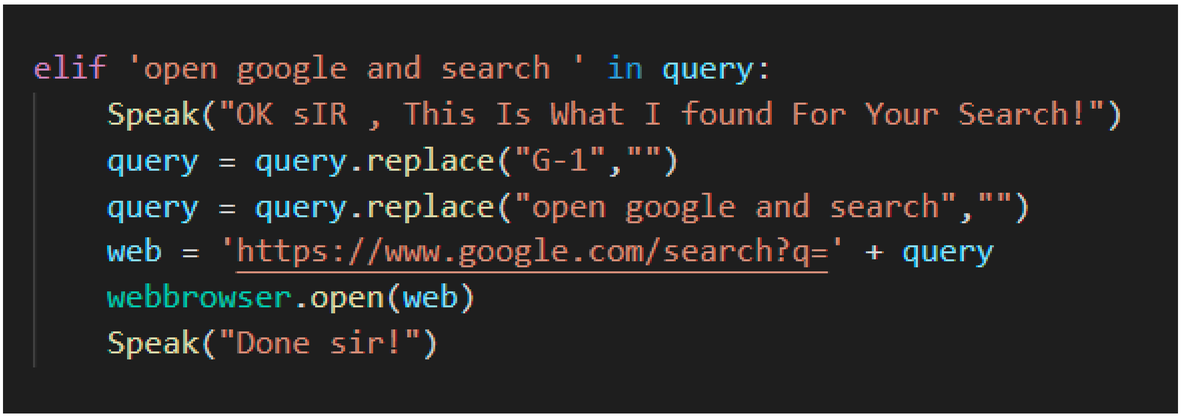


```
elif 'search youtube' in query:
    Speak("OK sIR , This Is What I found For Your Search!")
    query = query.replace("G-1","")
    query = query.replace("youtube search","")
    web = 'https://www.youtube.com/results?search_query=' + query
    webbrowser.open(web)
    Speak("Done sir!")
```

Here, we are using an elif loop to check whether YouTube is in the user's query. Let' suppose the user gives a command as “G-One” open YouTube and search …… ." So, open YouTube will be in the user's query, and the elif condition will be true.

8.3 Open GOOGLE:

To open Google site in a web-browser:

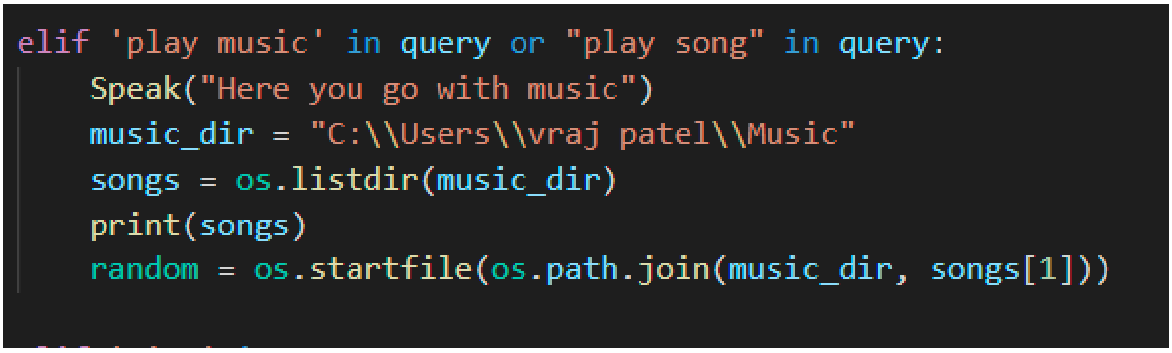


```
elif 'open google and search ' in query:
    Speak("OK sIR , This Is What I found For Your Search!")
    query = query.replace("G-1","")
    query = query.replace("open google and search","")
    web = 'https://www.google.com/search?q=' + query
    webbrowser.open(web)
    Speak("Done sir!")
```

Here, we are using an elif loop to check whether google is in the user's query. Let' suppose the user gives a command as “G-One” open google and search …… ." So, open google will be in the user's query, and the elif condition will be true.

8.4 play music:

To play music, we need to import a module called os. Import this module directly with an import statement.

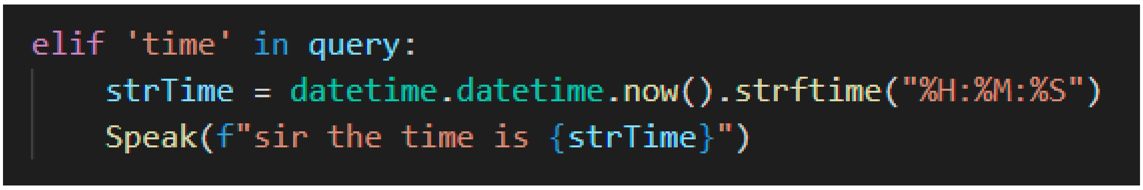


```
elif 'play music' in query or "play song" in query:
    Speak("Here you go with music")
    music_dir = "C:\\Users\\vraj patel\\Music"
    songs = os.listdir(music_dir)
    print(songs)
    random = os.startfile(os.path.join(music_dir, songs[1]))
```

In the above code, we first opened our music directory and then listed all the songs present in the directory with the os module's help. With the help of os.startfile, you can play any song of your choice. I am playing the first song in the directory. However, you can also play a random song with the help of a random module. Every time you command to play music, G-ONE will play any random song from the song directory.

8.5 time:

To know the current time:



```
elif 'time' in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    Speak(f"sir the time is {strTime}")
```

In the above, code we are using the datetime() function and storing the current or live system time into a variable called strTime. After storing the time in strTime, we are passing this variable as an argument in speak function. Now, the time string will be converted into speech.

8.6sendEmail:

To send an email, we need to import a module called smtplib.

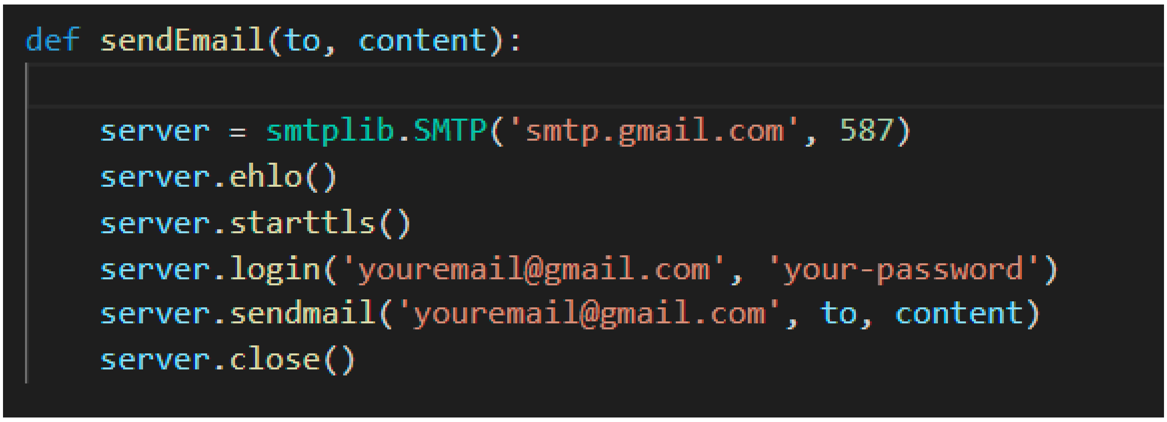
Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and route emails between mail servers. An instance method called sendmail is present in the SMTP module. This instance method allows us to send an email. It takes 3 parameters:

The sender: Email address of the sender.

The receiver: T Email of the receiver.

The message: A string message which needs to be sent to one or more than one recipient.

We will create a **sendEmail()** function, which will help us send emails to one or more than one recipient.

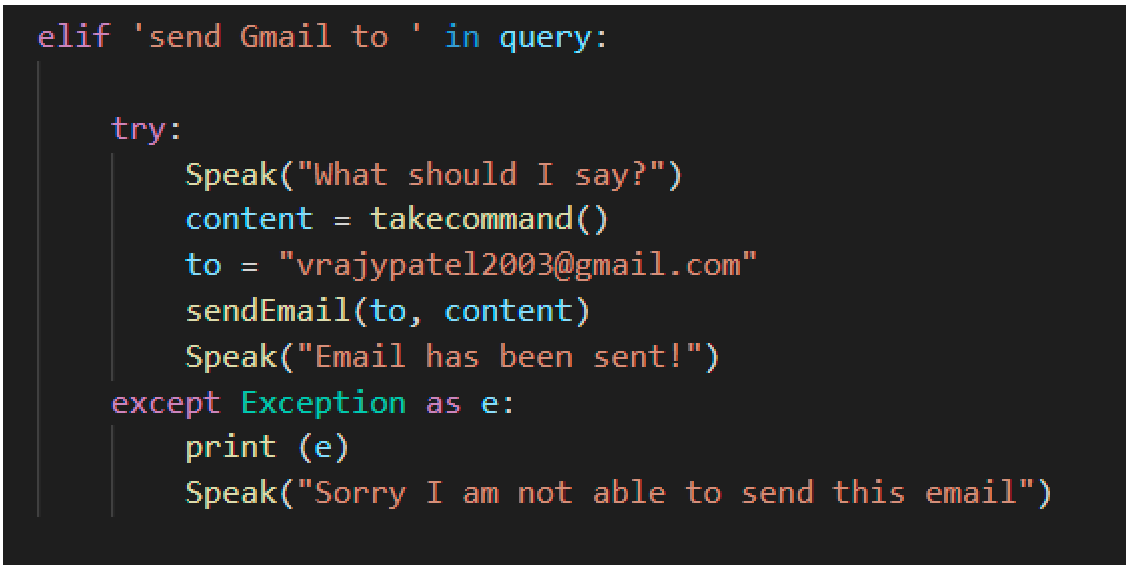


```
def sendEmail(to, content):

    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('youremail@gmail.com', 'your-password')
    server.sendmail('youremail@gmail.com', to, content)
    server.close()
```

In the above code, we are using the SMTP module, which we have already discussed above.

Note: Do not forget to 'enable the less secure apps' feature in your Gmail account. Otherwise, the sendEmail function will not work properly.

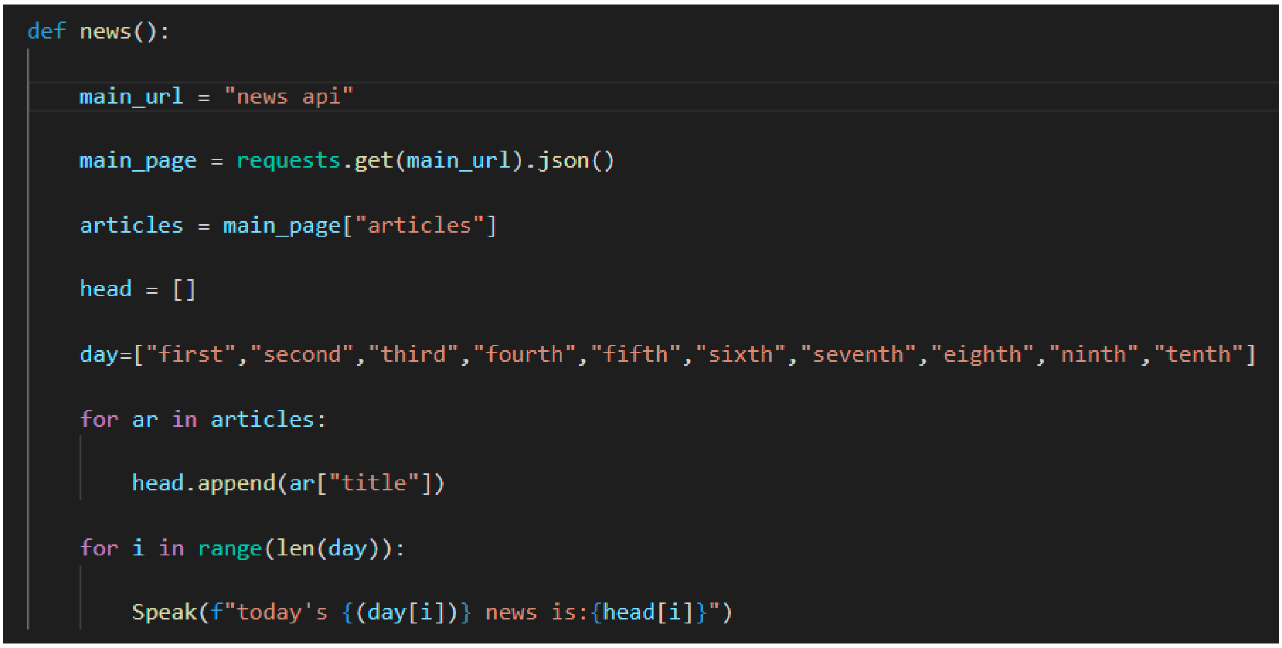
Calling sendEmail() function inside the main() function:



```
elif 'send Gmail to ' in query:

    try:
        Speak("What should I say?")
        content = takecommand()
        to = "vrajypatel2003@gmail.com"
        sendEmail(to, content)
        Speak("Email has been sent!")
    except Exception as e:
        print (e)
        Speak("Sorry I am not able to send this email")
```

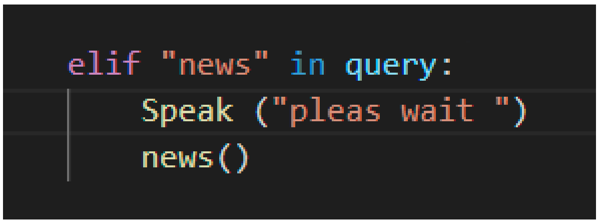
8.7news:

To fetch the latest news headlines, we'll be using NewsAPI. Signup for a free account on NewsAPI and get the API Key. Add the NEWS_API_KEY in the .env file.



```
def news():

    main_url = "news api"

    main_page = requests.get(main_url).json()

    articles = main_page["articles"]

    head = []

    day=["first","second","third","fourth","fifth","sixth","seventh","eighth","ninth","tenth"]

    for ar in articles:

        head.append(ar["title"])

    for i in range(len(day)):

        Speak(f"today's {(day[i])} news is:{head[i]}")
```

elif commend for news :



```
elif "news" in query:
    Speak ("pleas wait ")
    news()
```

Chapter-9

Conclusion

9.1 Conclusion:

• In this project “Virtual Assistant G-One” we discussed the design and implementation of Digital Assistance. The project is built using open-source software modules with PyCharm community backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easier to add additional features without disturbing current system functionalities. It not only works on human commands but also give responses to the user based on the query being asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way the user feels more comfortable and feels free to interact with the voice assistant. The Program should also eliminate any kind of unnecessary manual work required in the user life of performing every task. The entire project work on voice command.

Chapter-10

REFERENCE AND BIBLIOGRAPHY

10.1 REFERENCE AND BIBLIOGRAPHY

Websites referred

www.stackoverflow.com

www.pythonprogramming.net

www.codecademy.com

www.tutorialspoint.com

www.google.co.in

Books referred

Python Programming - Kiran Gurbani

Learning Python - Mark Lutz

YouTube Channels referred

CS Dojo

- Edureka!