

Expanding Horizons

DESKTOP VOICE ASSISTANT

Submitted By

Mr. Gunaji Panchal

Mr. Shuja Ansari

Ms. Nirupama Barahate

Ms. Sima Sarnaik

For the Degree of

Bachelor of Engineering

(Computer Science & Engineering)

**DR. BABASAHEB AMBEDKAR MARATHWADA UNIVERSITY,
CHHATRAPATI SAMBHAJINAGAR**



Department of Computer Science & Engineering

International Centre of Excellence in Engineering and Management

Chhatrapati Sambhajnagar Maharashtra State, India

2022-2023

Expanding Horizons

DESKTOP VOICE ASSISTANT

Submitted By

Mr. Gunaji Panchal

Mr. Shuja Ansari

Ms. Nirupama Barahate

Ms. Sima Sarnaik

For the Degree of

Bachelor of Engineering

(Computer Science & Engineering)

Guided By

Prof. Khan S. H.

Department of Computer Science & Engineering

International Centre of Excellence in Engineering and Management

Chhatrapati Sambhajnagar Maharashtra State, India

2022-2023

CERTIFICATE

This is to certify that the project entitled “**Desktop voice Assistant**” submitted by **Gunaji Panchal, Shuja Ansari, Nirupama Barahate, Sima Sarnaik**. Have completed work under my supervision and guidance for the award of **Degree of Bachelor of Engineering (B.E) in Computer Science & Engineering** Department of International Centre of Excellence in Engineering & Management, as per the requirement of the Dr. Babasaheb Ambedkar Marathwada University Chhatrapati Sambhajinagar, during the academic year 2022-2023.

Place: Chhatrapati Sambhajinagar

Date: 26th May 2023

Prof. Khan S. H

Guide

Computer Science & Engineering
Engineering

Prof. N.S. Magar

Head

Computer Science &

Dr.R.S.Jahagirdar

Director

International Centre of Excellence in Engineering and Management,
Chhatrapati Sambhajinagar

Name & Signature of External Examiners with Date:

1. _____

2. _____

ABSTRACT

The project aims to develop a personal-assistant for Linux-based systems. Jarvis draws its inspiration from virtual assistants like Cortana for Windows, and Siri for iOS.

It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands.

Users can interact with the assistant either through voice commands or using keyboard input. As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in google, Bing or yahoo, searching for videos, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and tasks.

The user statements/commands are analysed with the help of machine learning to give an optimal solution.

CONTENTS

1. INTRODUCTION	1
1.1 Objective	3
1.2 Purpose	3
1.3 Scope	3
1.4 Problem Identification	3
1.4.1 Purpose	4
1.4.2 Product Goals And Objectives	4
1.4.3 Product Description	4
1.5 Methodology	5
 2. LITERATURE SURVEY	 7
2.1 Survey Of Technology	7
2.1.1 Python	7
2.1.2 Quepy	7
2.1.3 Pyttsx3	8
2.1.4 Speech Recognition	8
2.1.5 SQLite	8
2.2 System Requirements	8
2.2.1 Software Requirements	8
2.2.2 Hardware Requirements	9
 3. SYSTEM DEVEOPMENT	 10
3.1 Use Case Diagram	11
3.2 Class Diagram	12
3.3 Sequence Diagram	13
3.4 Collaboration Diagram	14
3.5 Data Flow Diagram	15
3.6 Flow Chart	17
3.7 Activity Diagram	18
 4. PERFORMANCE ANALYSIS	 19
4.1How A Virtual Assistant Works?	20

4.2 Why To Use Jarvis ?	21
4.3 Deployment	24
4.4 Technologies Used	24
4.4.1 Frontend Framework	24
4.4.2 Backend Stack	24
4.4.3 Database	24
5.SYSTEM TESTING	25
5.1 Selenium Automation Tools	26
5.2 Limitations	27
6.CONCLUSION	28
6.1 Conclusion	28
6.2 Future Scope	28
6.3 Applications	29
7. REFERENCES	30
8.ACKNOWLEDGEMENT	31

1. INTRODUCTION

1. INTRODUCTION

A virtual assistant is a technology based on artificial intelligence. The software uses a device's microphone to receive voice requests while the voice output takes place at the speaker. But the most exciting thing happens between these two actions.

It is a combination of several different technologies: voice recognition, voice analysis and language processing.

It is completely developed using one of the most powerful language python.



Artificial Intelligence when used with machines, it shows us the capability of thinking like humans.

In this, a computer system is designed in such a way that typically requires interaction from human. As we know Python is an emerging language so it becomes easy to write a script for Voice Assistant in Python.

The instructions for the assistant can be handled as per the requirement of user. Speech recognition is the Alexa, Siri, etc.

In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to send emails without typing any word, Searching on Google without opening the browser, and performing many other daily tasks like playing music, opening your favourite IDE with the help of a single voice command.

In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By

making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. As the voice assistant is using Artificial Intelligence hence the result that it is providing are highly accurate and efficient.

The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task.

The assistant is no less than a human assistant but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focuses on the time complexities and reduces time. The functionalities include, It can send emails, It can read PDF, It can send text on WhatsApp, It can open command prompt, your favourite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube ,etc., in a web browser, It can give weather forecast, It can give desktop reminders of your choice.

It can have some basic conversation. Tools and technologies used are PyCharm IDE for making this project, and I created all py files in PyCharm. Along with this I used following modules and libraries in my project. pyttsx3, Speech Recognition, Date time, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, PyQt etc.

I have created a live GUI for interacting with the JARVIS as it gives a design and interesting look while having the conversation.

1.1 OBJECTIVES :

Currently, the project aims to provide the Linux Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, vocabulary help and many others but also help in automation of various activities. In the long run, we aim to develop a complete server assistant, by automating the entire server management process - deployment, backups, auto-scaling, logging, monitoring and make it smart enough to act as a replacement for a 6 general server administrator.

1.2 PURPOSE :

This Software aims at developing a personal assistant for Linux- based systems. The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech or text. It will ease most of the work of the user as a complete task can be done on a single command. Jarvis draws its inspiration from Virtual assistants like Cortana for Windows and Siri for iOS. Users can interact with the assistant either through voice commands or keyboard input.

1.3 SCOPE:

Presently, Jarvis is being developed as an automation tool and virtual assistant. Among the Various roles played by Jarvis are:

1. Medical diagnosis with Medicine aid.
2. Reminder and To-Do application.
3. Vocabulary App to show meanings and correct spelling errors.
4. Weather Forecasting Application. There shall be proper Documentation available on its Official Github repository for making further development easy and we aim to release our virtual assistant as an Open Source Software where modifications and contributions by the community are warmly welcomed.

1.4 PROBLEM IDENTIFICATION :

We are all well aware about Cortana, Siri, Google Assistant and many other virtual assistants which are designed to aid the tasks of users in Windows, Android and iOS platforms. But to our surprise, there's no such virtual assistant available for the paradise of Developers i.e. Linux platform.

1.4.1 Purpose :

This Software aims at developing a personal assistant for Linux-based systems. The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech or text. It will ease most of the work of the user as a complete task can be done on a single command. Jarvis draws its inspiration from Virtual assistants like Cortana for Windows and Siri for iOS. Users can interact with the assistant either through voice commands or keyboard input.

1.4.2 Product Goals And Objectives :

Currently, the project aims to provide the Linux Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, vocabulary help and many others but also help in automation of various activities. In the long run, we aim to develop a complete server assistant, by automating the entire server management process - deployment, backups, auto-scaling, logging, monitoring and make it smart enough to act as a replacement for a 6 general server administrator.

1.4.3 Product Description :

As a personal assistant, Jarvis assists the end-user with day-to-day activities like general human conversation, searching queries in various search engines like Google, Bing or Yahoo, searching for videos, retrieving images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and reminding the user about the scheduled events and tasks. The user statements/commands are analysed with the help of machine learning to give an optimal solution.

1.5 METHODOLOGY :

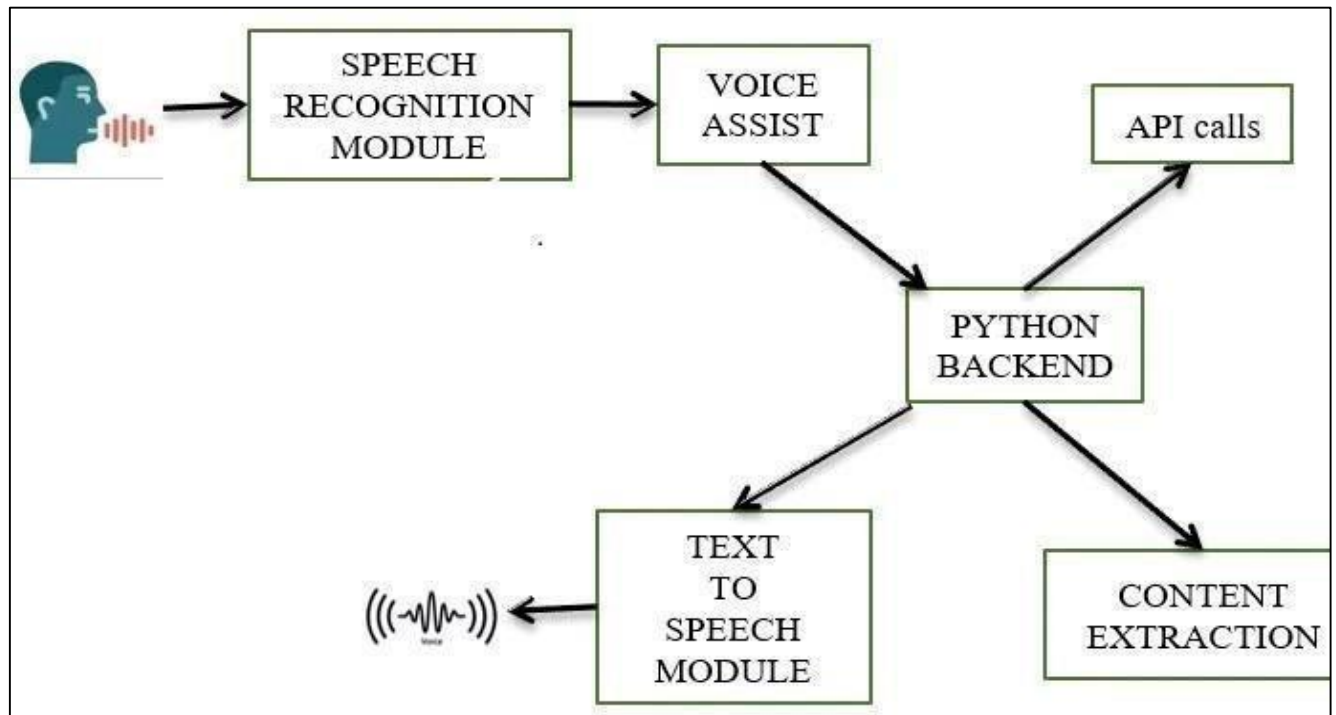


Fig:-Speech Recognition module

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

Python Backend

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

API calls

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

Content Extraction

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

Text-to-speech module

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

2. LITERATURE SURVEY

A major element in building a system is the selection of compatible software since the software in the markets is experiencing in geometric progression. Selected software should be acceptable by the one user as well as it should be feasible for the system. A major element in building a system is the selection of compatible hardware since the hardware in the market is experiencing in geometric progression selected hardware should be acceptable by the one user as well as it should be feasible for the system.

2.1 SURVEY OF TECHNOLOGY

2.1.1 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 JARVIS, libraries used are speech recognition to recognize voice, Pyttsx3 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.

2.1.2 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.

2.1.3 Pyttsx3

Pyttsx3 stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

2.1.4 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.

2.1.5 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 (Structured Query Language) 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 distribution 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.

2.2 SYSTEM REQUIREMENTS :

2.2.1 Software Requirements :

- Linux Distribution
- Proper Internet Connection
- Github Credentials
- Docker installed

- Python 2.7
- Heroku CLI
- Mplayer for voice support (Text-to-Speech)
- Chromium-based browser, like Chrome, Edge
- Heroku Credentials
- Node JS with npm

2.2.2 Hardware Requirements :

1. Windows System Desktop
2. Android System

3. SYSTEM DEVELOPMENT

In This model that exploits abundant training data to directly learn pronunciation variation. Inter with a parametric model yields the best performance, with arelative improvement of 5.2% in WER over the baseline.

There are a number of ways in which this work could be extended. First, closer integration with acoustic model is likely to yield sharper distributions and a tighter fit to the data. Second, estimating word pronunciation cocounts in semi-supervised fashion (e.g. through word recognition instead of forced alignment) would broaden its applicability to a wide range of speech genres and tasks.

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.

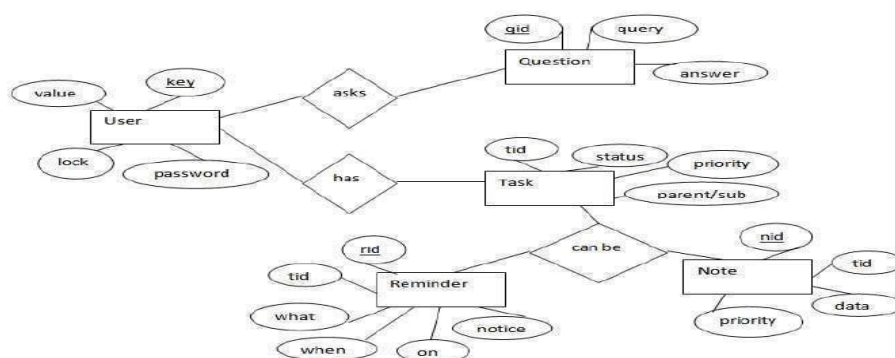


Figure 1: ER Diagram

3.1 USE CASE DIAGRAM:

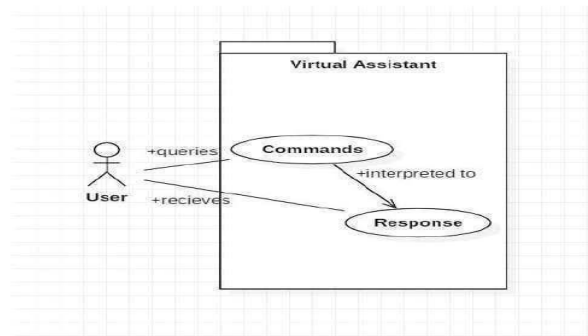


Figure 2: Use Case Diagram

3.2 CLASS DIAGRAM:

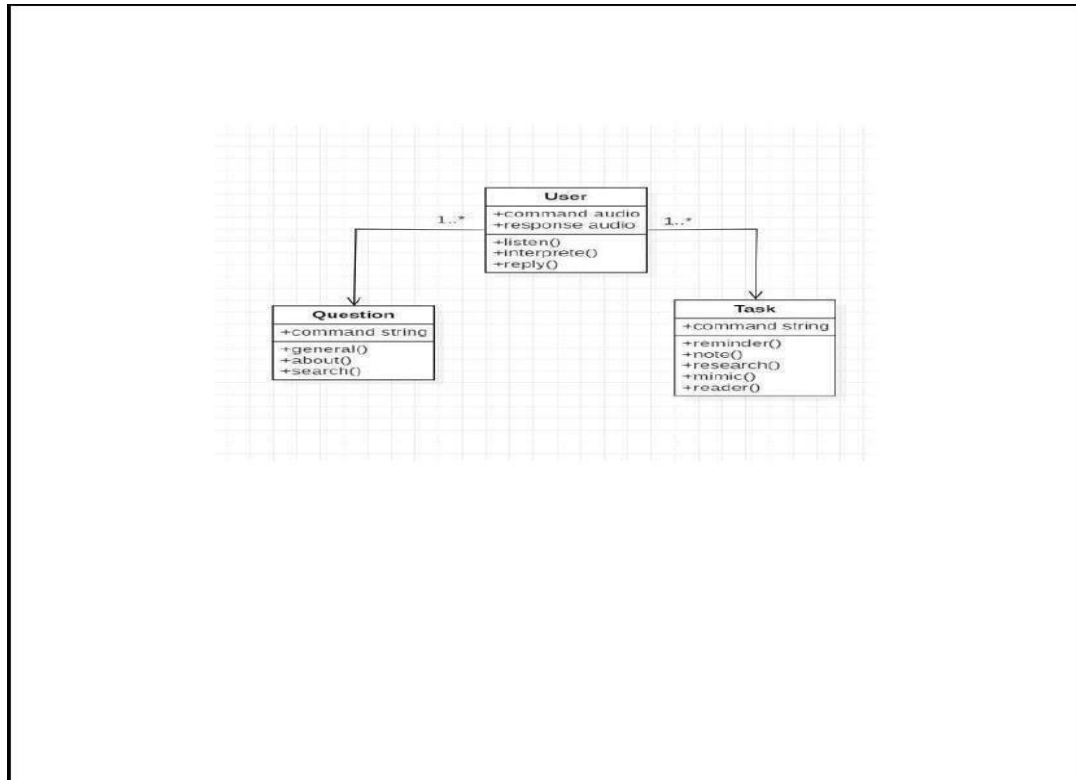


Figure 3: Class Diagram

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

3.3 SEQUENCE DIAGRAM:

Sequence diagram for Query-Response

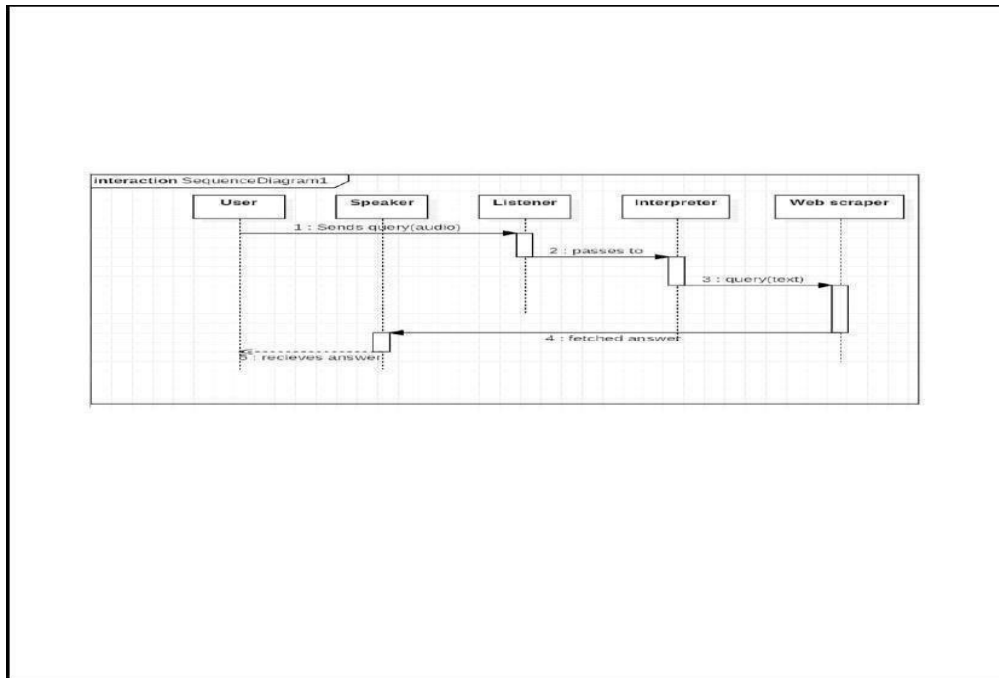


Figure 4: Sequence Diagram

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.

3.4 COLLABORATION DIAGRAM:

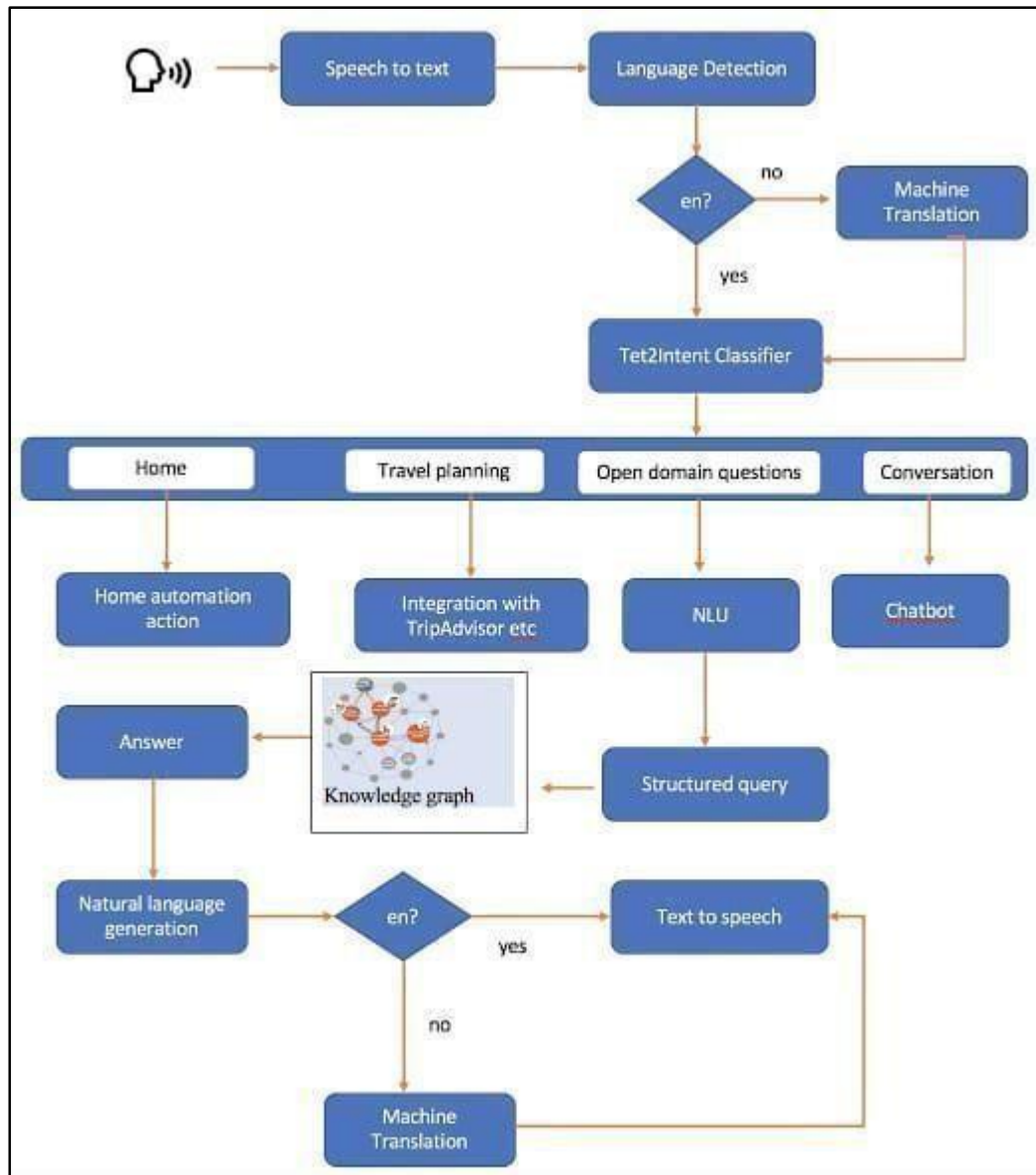


Figure 5: Collaboration Diagram

3.5 DATA FLOW DIAGRAM:

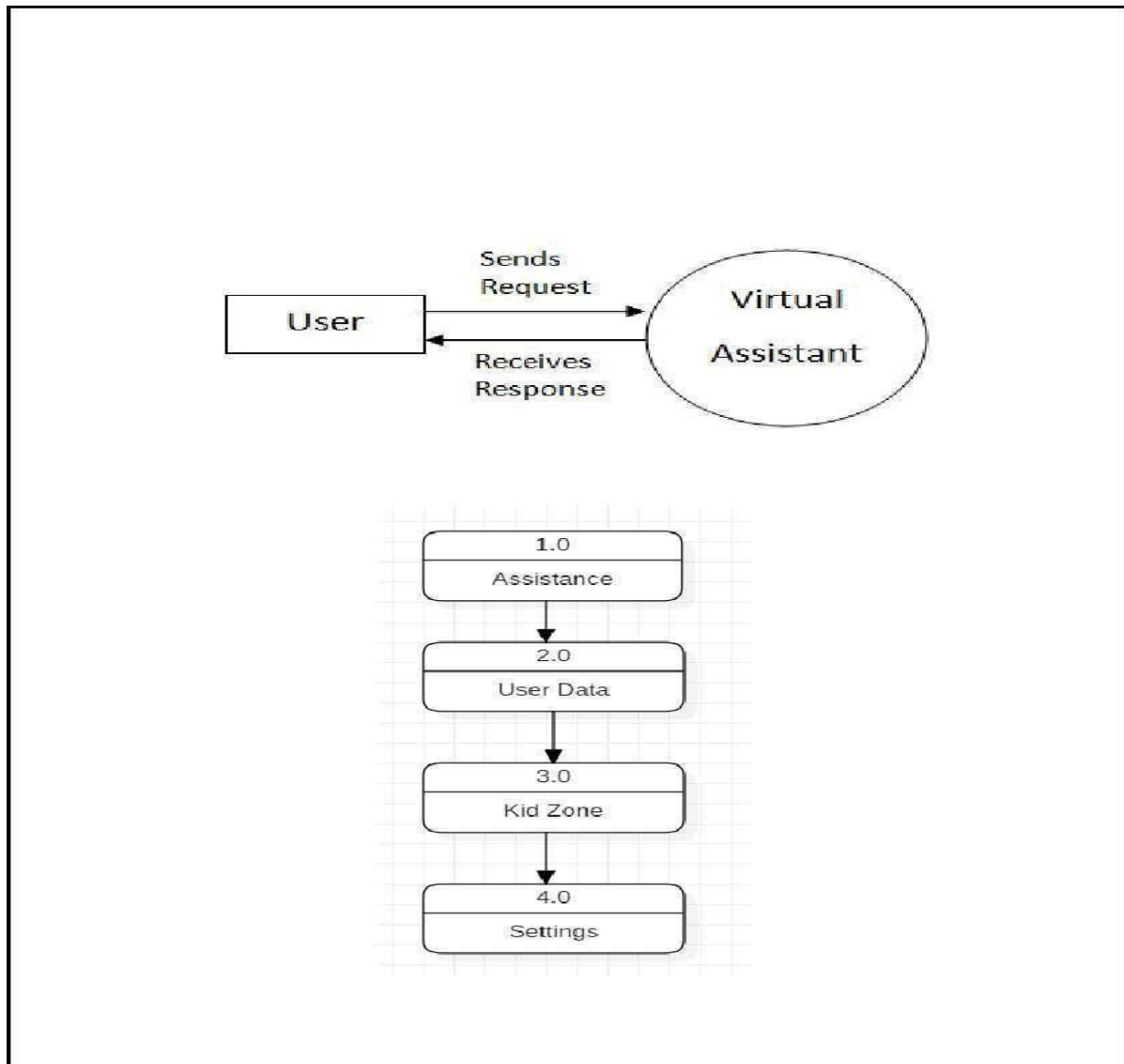


Figure 6: Data Flow Diagram

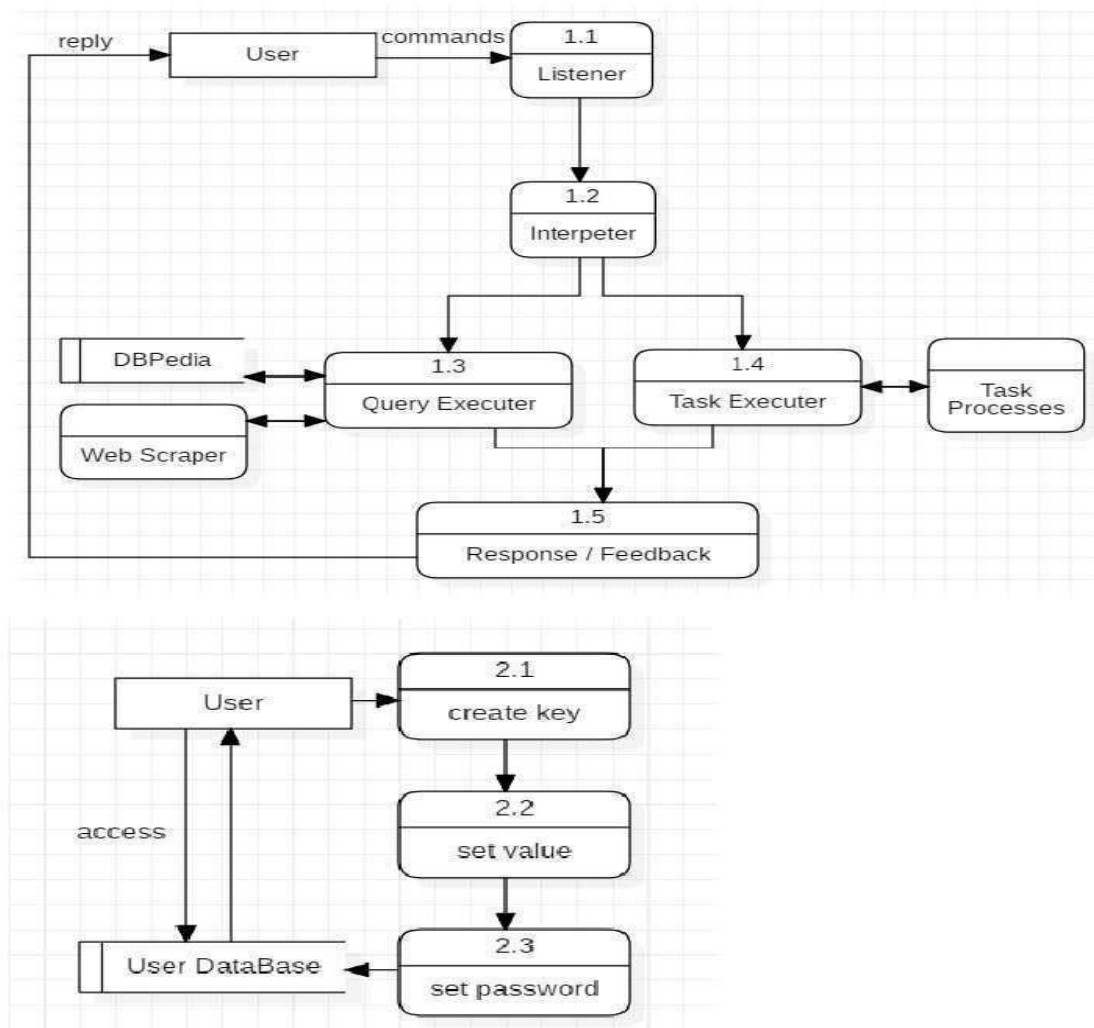


Figure 7: Data Flow Diagram

3.6 FLOW CHART:

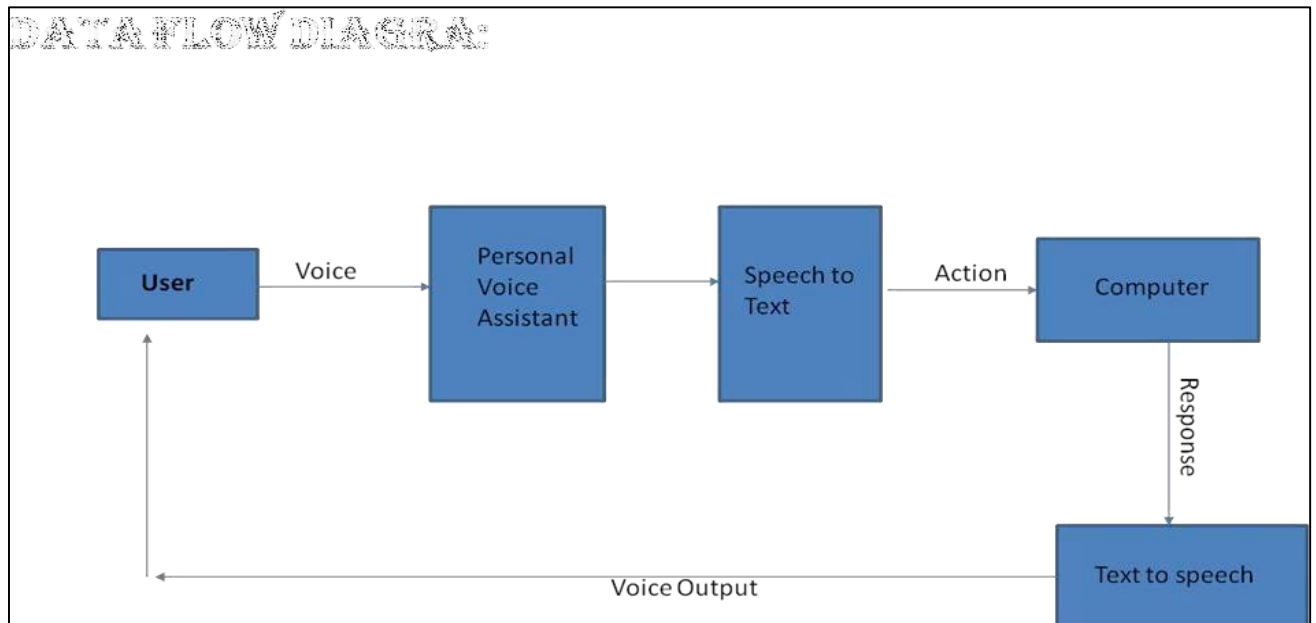


Figure 8: Flow Chart

3.7ACTIVITY DIAGRAM:

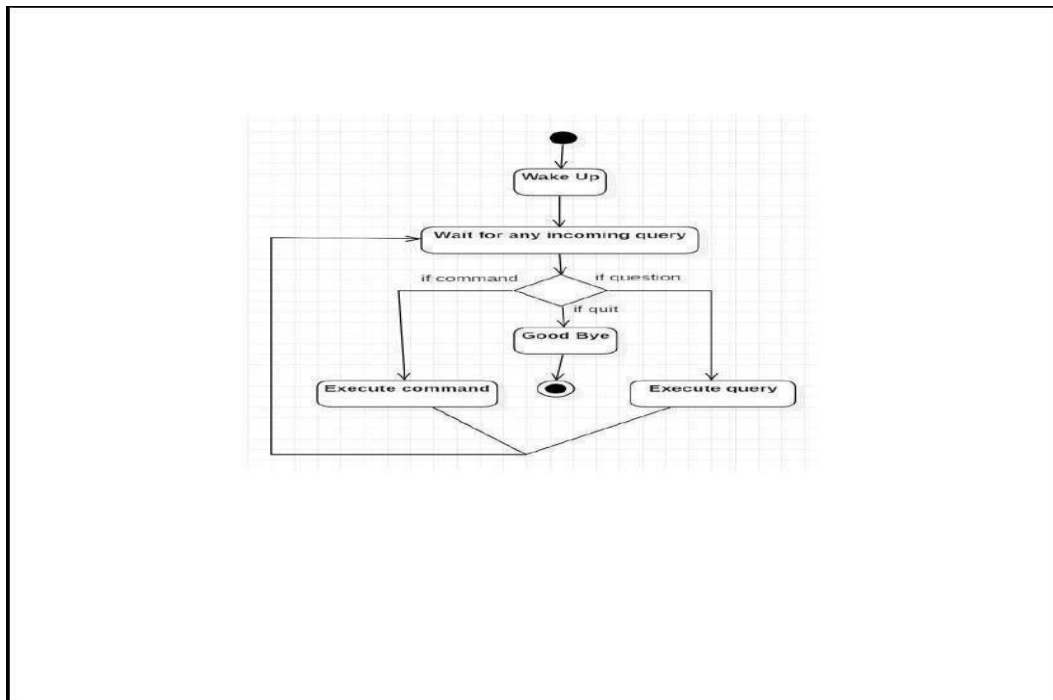


Fig 9: Activity Diagram

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 key’s 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.PERFORMANCE ANALYSIS

Virtual Assistant use Natural Language Processing (N.L.P) to match user text or voice input to executable commands

A virtual assistant is a self-employed worker who specializes in offering administrative services to clients from a remote location, usually a home office.

Typical tasks a virtual assistant might perform include scheduling appointments, making phone calls, making travel arrangements, and managing email accounts.

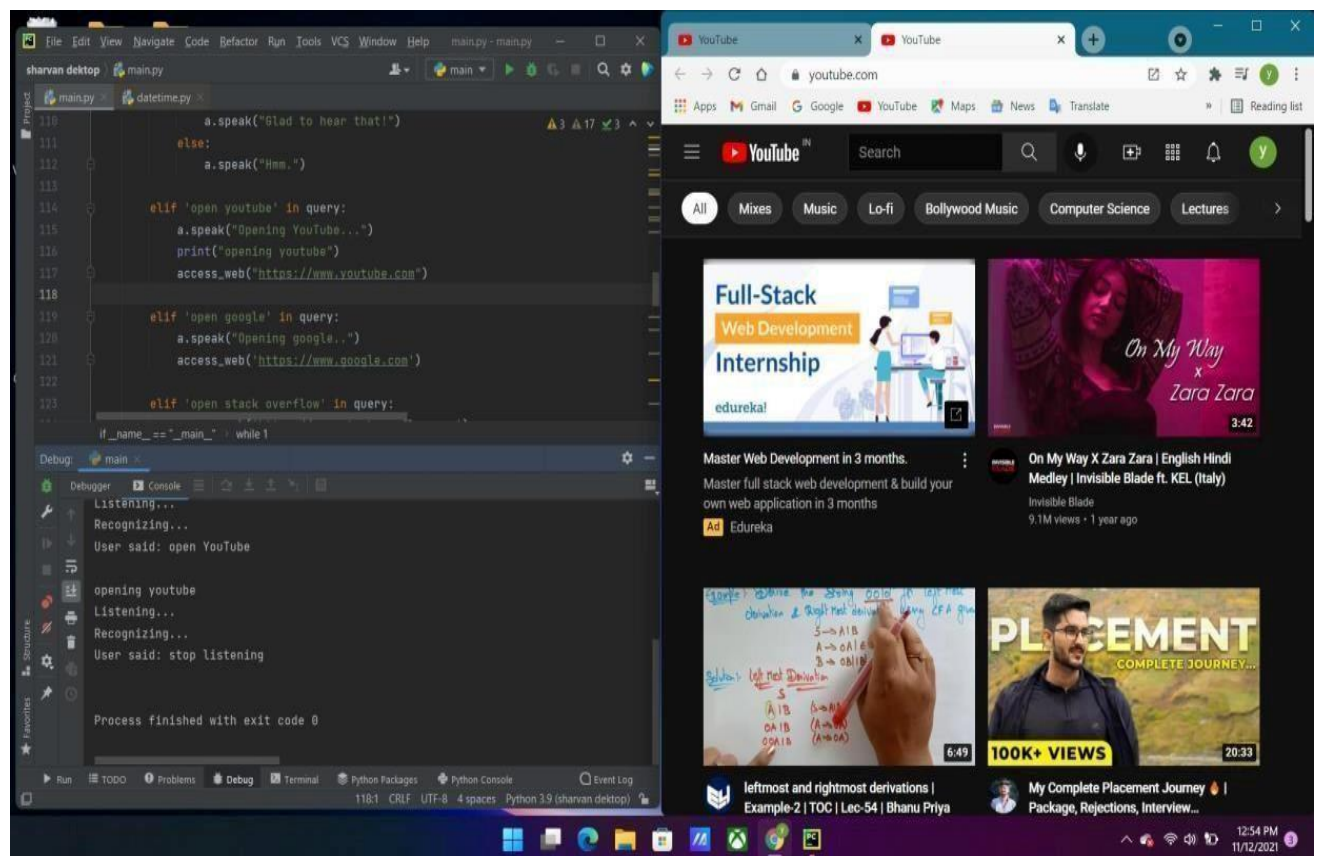


Fig 10: Input and Output for YouTube search

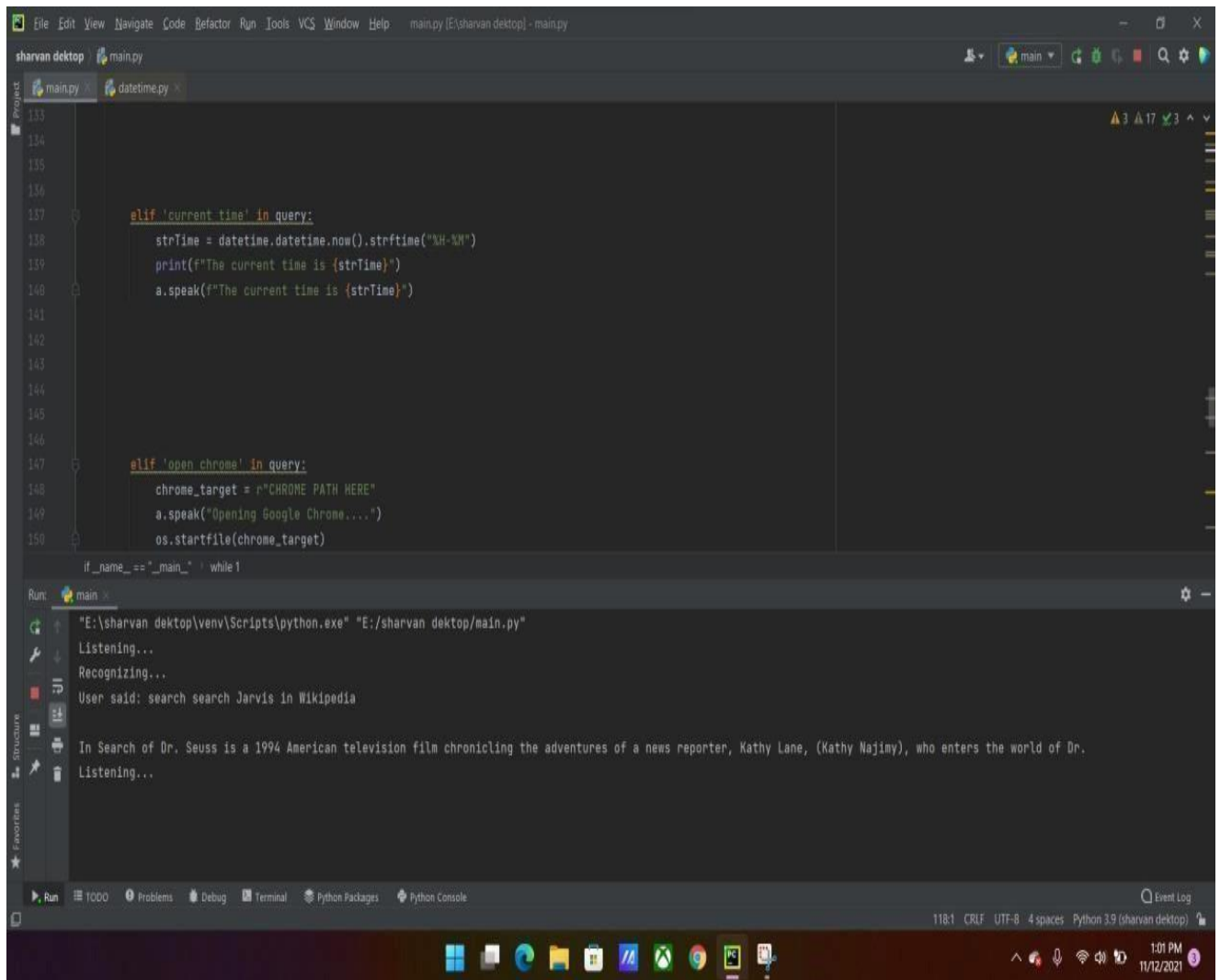


Fig 11: Input and Output for Introduction

4.1 HOW A VIRTUAL ASSISTANT WORKS?

Virtual assistants have become more prominent as small businesses and start-ups rely on virtual offices to keep costs down and businesses of all sizes increase their use of the internet for daily operations. Because a virtual assistant is an independent contractor, a business does not have to provide the same benefits or pay the same taxes that it would for a full-time employee.

Also, since the virtual assistant works offsite, there is no need for a desk or other workspace at the company's office. A virtual assistant is expected to pay for and provide their own computer equipment, commonly used software programs, and high-speed Internet service.

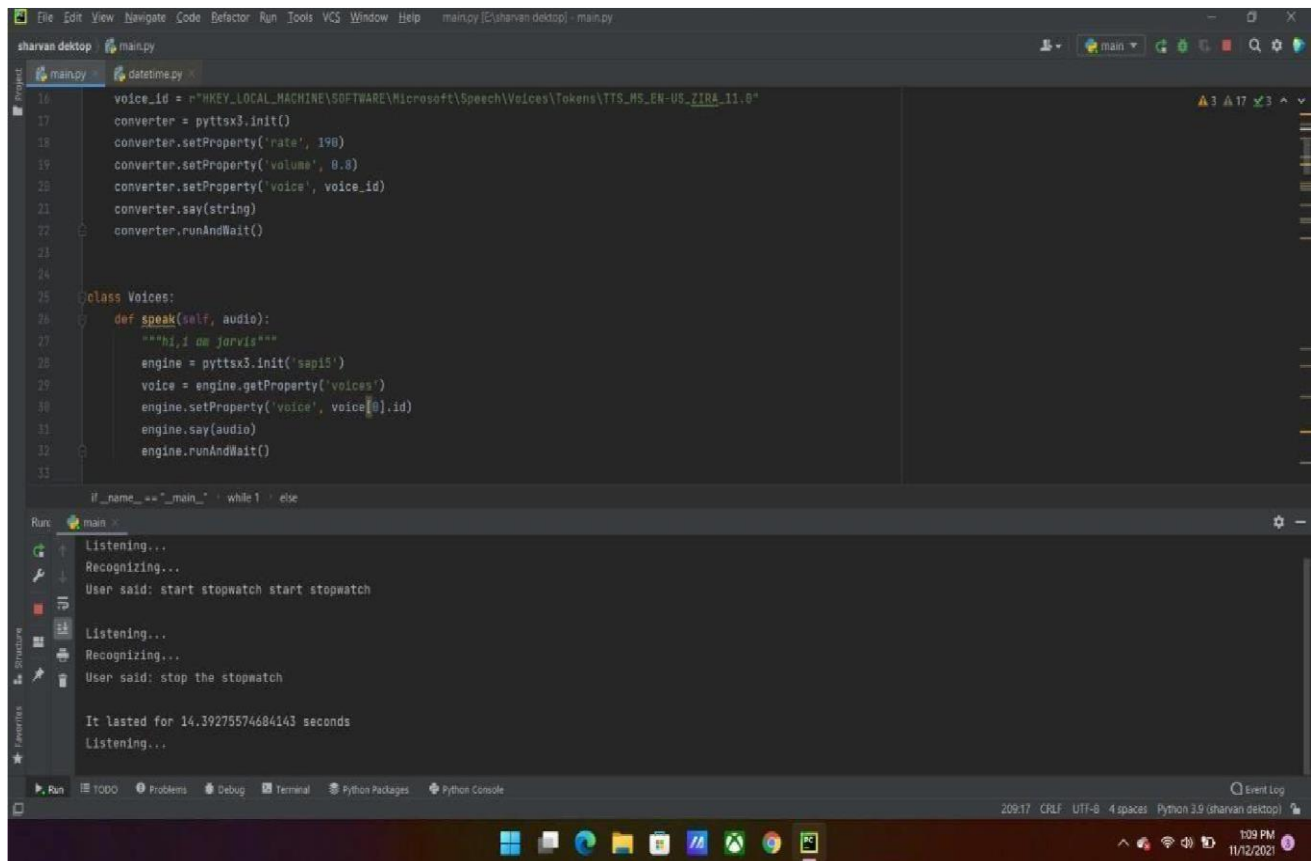


Fig 12: Input and Output for stop watch

4.2 WHY TO USE JARVIS?

1. It fulfils the lack of a virtual assistant in Linux systems.
2. It has an easy to install and use interface.
3. It accepts inputs even through voice or keyboard.
4. It automates tedious tasks like deployment, unit testing through a single command.
5. It gives live weather updates.
6. It gives advice on health.

```
34
35 a = Voices()
36
37
38 def wish():
39     """wishes the user according to time"""
40     hour = int(datetime.datetime.now().hour)
41     if hour >= 20 and hour <= 24:
42         a.speak("Night time working sir!")
43     elif hour >= 2 and hour <= 12:
44         a.speak("Good morning sir!")
45     elif hour > 12 and hour < 4:
46         a.speak("Good afternoon sir!")
47     else:
48         a.speak("Good evening sir!")
49     a.speak("How can I help you?")
50
51 male_voice()
```

Run: main
"E:\sharvan dektop\venv\Scripts\python.exe" "E:/sharvan dektop/main.py"
Listening...

Fig 13: Input and Output for current greeting

```
133
134
135
136
137 elif 'current time' in query:
138     strTime = datetime.datetime.now().strftime("%H-%M")
139     print(f"The current time is {strTime}")
140     a.speak(f"The current time is {strTime}")
141
142
143
144
145
146
147 elif 'open chrome' in query:
148     chrome_target = r"CHROME PATH HERE"
149     a.speak("Opening Google Chrome...")
150     os.startfile(chrome_target)
151
152 if __name__ == "__main__": while 1: if 'joke' in query
```

Run: main
"E:\sharvan dektop\venv\Scripts\python.exe" "E:/sharvan dektop/main.py"
Listening...
Recognizing...
User said: tell me the current time
The current time is 13:05
Listening...

Fig 14: Input and Output for current time

```

195         else:
196             a.speak("Ok changing voice")
197             Voices.speak = male_voice
198             a.speak("Hi sir! How can I help you?")
199
200     elif 'coronavirus cases' in query:
201         r = requests.get('https://api.covid19api.com/country/india/status/confirmed/live').text
202         parser = json.loads(r)
203         l = parser[1:-1]
204         today = l[0]
205         print(f"There are {today['Cases']} confirmed corona cases in India.")
206         a.speak(f"There are {today['Cases']} confirmed corona cases in India.")
207
208     else:
209         pass
210
211 if __name__ == "__main__": while 1: elif 'coronavirus cases' in que...
```

Run: main
 "E:\sharvan dektop\venv\Scripts\python.exe" "E:\sharvan dektop/main.py"
 Listening...
 Recognizing...
 User said: tell me tell me the coronavirus cases in India
 There are 34414186 confirmed corona cases in India.

Fig 15: Input and Output for corona virus cases

```

55 with sr.Microphone() as source:
56     print("Listening...")
57     r.pause_threshold = 1
58     audio = r.listen(source)
59
60 try:
61     print("Recognizing...")
62     query = r.recognize_google(audio, language='en-in')
63     print(f"User said: {query}\n")
64 except Exception as e:
65     print("Say that again please...")
66     return "None"
67 return query
68
69 if __name__ == "__main__": while 1: elif 'joke' in query
```

Run: main
 "E:\sharvan dektop\venv\Scripts\python.exe" "E:\sharvan dektop/main.py"
 Listening...
 Recognizing...
 User said: open open calculator open calculator
 Listening...

Fig 16: Input and Output for calculator

4.3DEPLOYMENT

Companies with very frequent releases may require knowledge of DevOps. For example, the company that operates an image hosting website Flickr developed a DevOps approach to support ten deployments a day. Daily deployment cycle would be much higher at organizations producing multi-focus or multi-function applications.

Daily deployment is referred to as continuous deployment or continuous delivery and has been associated with the lean start-up methodology. Professional associations and blogs posts have formed on the topic since 2009.

4.4TECHNOLOGIES USED:

4.4.1 Frontend Framework

- AngularJS

4.4.2 Backend Stack

- GO-lang
- Machine Learning
- Docker Container

4.4.3 Database

- SQLite
- Cookies

5.SYSTEM TESTING

The system testing is done on fully integrated system to check whether the requirements are matching or not. The system testing for JARVIS desktop assistant focuses on the following four parameters:

1. FUNCTIONALITY:

In this we check the functionality of the system whether the system performs the task which it was intended to do. To check the functionality each function was checked and run, if it is able to execute the required task correctly then the system passes in that particular functionality test. For example to check whether ASSISTANT can search on Google or not, as we can see in the figure 7.1, user said “Open Google”, then Jarvis asked, ”What should I search on Google?” then user said, “What is Python”, Jarvis open Google and searched for the required input.

2. USABILITY:

Usability of a system is checked by measuring the easiness of the software and how user friendly it is for the user to use, how it responses to each query that is being asked by the user. It makes it easier to complete any task as it automatically do it by using the essential module or libraries of Python, in a conversational interaction way. Hence any user when instruct any task to it, they feel like giving task to a human assistant because of the conversational interaction forgiving input and getting the desired output in the form of task done. The desktop assistant is reactive which means it know human language very well and understand the context that is provided by the user and gives response in the same way, i.e. human understandable language, English. So user finds its reaction in an informed and smart way. The main application of it can be its multitasking ability.

It can ask for continuous instruction one after other until the user “QUIT” it. It asks for the instruction and listen the response that is given by user without needing any trigger phase and then only executes the task

3. SECURITY:

The security testing mainly focuses on vulnerabilities and risks. As JARVIS is a local desktop application, hence there is no risk of data breaching through remote access. The software is dedicated to a specific system so when the user logs in, it will be activated.

4. STABILITY:

Stability of a system depends upon the output of the system, if the output is bounded and specific to the bounded input then the system is said to be stable. If the system works on all the poles of functionality then it is stable

5.1 SELENIUM AUTOMATION TOOL

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms.

It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications.

Testing done using Selenium tool is usually referred to as Selenium 12 Testing. Selenium is not just a single tool but a suite of software, each catering to different testing needs of an organization.

The entire Selenium Tool Suite is comprised of four components:

- Selenium IDE, a Firefox add-on that you can only use in creating relatively simple test cases and test suites.
- Selenium Remote Control, also known as Selenium 1, which is the first Selenium tool that allowed users to use programming languages in creating complex tests.
- WebDriver, the newer breakthrough that allows your test scripts to communicate directly to the browser, thereby controlling it from the OS level.
- Selenium Grid is also a tool that is used with Selenium RC to execute parallel tests across different browsers and operating systems

5.2 LIMITATIONS:

Security is somewhere an issue, there is no voice command encryption in this project. Background voice can interfere. Misinterpretation because of accents and may cause inaccurate results. JARVIS cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, “Ok Google!

6.CONCLUSIONS

6.1 CONCLUSION:

DESKTOP ASSISTANT is a very helpful voice assistant without any doubt as it saves time of the user by conversational interactions, its effectiveness and efficiency. But while working on this project, there were some limitations encountered and also realized some scope of enhancement in the future.

Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving weather forecast details, vocabulary help and medical related queries. We aim to make this project a complete server assistant and make it smart enough to act as a replacement for a general server administration.

The future plans include integrating Jarvis with mobile using React Native to provide a synchronised experience between the two connected devices. Further, in the long run, Jarvis is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Jarvis.

6.2 FUTURE ENHANCEMENTS:

Enhancement in the capacity of database or the data training sets can be done in this for more situations or the acquaintances that can be faced by JARVIS.

This would upgrade its effectiveness and the wide range ability of producing responses.

Further addition of more voices can also be done as an additional feature. So these limitations can be broken with the increase in data training sets.

The interface of the system can be improved more or we can say can be optimized. From saying more optimized it is meant that the interface can be more user friendly, comprehensive and easy to use for more percentage of users.

So the Jarvis would become more accessible and intractable.

1. Make It to learn more on its own and develop a new skill in it.
2. DESKTOP ASSISTANT android app can also be developed.
3. Make more It voice terminals.
4. Voice commands can be encrypted to maintain security

6.3 APPLICATIONS:

1) AI chatbots:

AI chatbots are computer programs that serve as virtual assistants and communicate with users through text-based interfaces on websites, social media platforms and messaging apps. These chatbots can assist customers, respond to inquiries or start a discussion with them. AI chatbots employ natural language processing (NLP) and machine learning (ML) algorithms to understand user input, produce pertinent responses and improve their performance over time by learning from these interactions.

2) Conversational agents:

Beyond text-based chatbots, conversational agents enable virtual assistants to converse in a human-like manner through both text and voice interfaces. These AI-based systems are designed to comprehend intricate linguistic patterns, discern user intent and offer responses that are appropriate in a given situation. Conversational AI can be used to build more sophisticated virtual assistants that can handle various tasks and offer a more intuitive and natural user interface.

3) AI virtual assistants:

These advanced virtual assistants can perform a variety of tasks, often across multiple devices and platforms. They frequently converse with users by using natural language through voice-based interactions. With these AI virtual assistants, users can easily and quickly access information, control smart home appliances and manage their calendars by integrating with a variety of services and applications. Amazon Alexa and Apple Siri are examples of AI virtual assistants.

REFERENCE

- [1] Alotto, F., Scidà, I., and Osello, A. (2020). “Building modeling with artificial intelligence and speech recognition for learning purpose.” Proceedings of EDULEARN20 Conference, Vol. 6. 7th.
- [2] Beirl, D., Rogers, Y., and Yuill, N. (2019). “Using voice assistant skills in family life.” ComputerSupported Collaborative Learning Conference, CSCL, Vol. 1, Inter-national Society of the Learning Sciences, Inc. 96–103.
- [3] Canbek, N. G. and Mutlu, M. E. (2016). “On the track of artificial intelligence: Learning with intelligent personal assistants.” Journal of Human Sciences, 13(1), 592–601.
- [4] Malodia, S., Islam, N., Kaur, P., and Dhir, A. (2021). “Why do people use artificial intelligence (AI)-enabled voice assistants?.” IEEE Transactions on Engineering Management.
- [5] Nasirian, F., Ahmadian, M., and Lee, O.-K. D. (2017). “Ai-based voice assistant systems: evaluating from the interaction and trust perspectives.
- [6] RAJA, K. D. P. R. A. (2020). “Jarvis ai using python.
- [7] Sangpal, R., Gawand, T., Vaykar, S., and Madhavi, N. (2019). “Jarvis: An inter-pretation of AIML with integration of gttts and python.” 2019 2nd International Con-ference on Intelligent Computing, Instrumentation and Control Technologies (ICI-CICT), Vol. 1. 486–489.
- [8] Steen, J. and Wilroth, M. (2021). “Adaptive voice control system using ai.
- [9] Terzopoulos, G. and Satratzemi, M. (2019). “Voice assistants and artificial intelligence in education.” Proceedings of the 9th Balkan Conference on Informatics. 1–6.
- [10] Tibola, L. R. and Tarouco, L. M. R. (2013). “Interoperability in virtual world.” XVIII Congreso Argentino de Ciencias de la Computación.
- [11] Vora, J., Yadav, D., Jain, R., and Gupta, J. (2021). “Jarvis: A pc voice assistant.
- [12] Nasirian et al. (2017) Malodia et al. (2021) Vora et al. (2021) Tibola and Tarouco(2013) Sangpal et al. (2019) RAJA (2020) Beirl et al. (2019) Terzopoulos and Satratzemi
- [13] (2019) Alotto et al. (2020) Steen and Wilroth (2021) Canbek and Mutlu (2016)

ACKNOWLEDGEMENT

We wish to extend my sincere gratitude to my project guide, **Prof. Khan S.H**, Lecturer, Department of Computer Science and Engineering for valuable guidance and encouragement which has been absolutely helpful in successful completion of this project.

We are indebted to **Prof. N.S Magar**, Professor and Head, Department of Computer Science and Engineering for his valuable support.

We are extremely thankful to **Dr.R.S.Jahagirdar** Director, ICEEM Chhatrapati Sambhajinagar, for providing me infrastructure facilities to work in, without which this work not have been possible.

We are also grateful to our parents and friends for their timely aid without which we wouldn't have finished our project successfully.

Name Of Student

Mr. Gunaji Panchal

Mr. Shuja Ansari

Ms. Nirupama Barahate

Ms. Sima Sarnaik