

# Desktop Assistant AI Using Python

Abeed Sayyed<sup>1</sup>, Ashpak Shaikh<sup>2</sup>, Ashish Sancheti<sup>3</sup>, Swikar Sangamner<sup>4</sup>, Prof. Jayant H Bhangale<sup>5</sup>

UG Scholar, Department of Mechanical Engineering<sup>1,2,3,4</sup>

Faculty, Department of Mechanical Engineering<sup>5</sup>

Matoshri College of Engineering and Research Centre, Eklahare, Nashik

**Abstract:** The advancement in technology over time has been unmeasurable. From the first digital computer built by Eniac having a clock speed of 100KHz to Summit developed by the US Department of Energy has a performance of 148.6 peta Flops, we have come a long way in technological advancement. In such an era of advancement if people are still struggling to interact with their machine using various input devices then it's not worth it. For this reason, many voice assistants were developed and are still being improved for better performance and efficiency. The main task of a voice assistant is to minimize the use of input devices like keyboard, mouse, touch pens, etc. This will reduce both the hardware cost and space taken by it. The Most famous application is of iPhone "SIRI" which helps the end user to communicate end user mobile with voice and it also responds to the voice commands of the user. Same kind of application is also developed by the Google that is "Google Voice Search" which is used for in Android Phones. It is named as Personal Assistant with Voice Recognition Intelligence, which takes the user input in form of voice or text and process it and returns the output in various forms like action to be performed or the search result is dictated to the end user. In addition, this proposed system can change the way of interactions between end user and the mobile devices.

**Keywords:** Desktop Assistant, Python, Machine Learning, Text to Speech, Speech to Text, Language Processing, Voice Recognition, Artificial Intelligence, Internet of Things (IoT), Virtual Assistant.

## I. 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 JARVIS. 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 JARVIS.

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. JARVIS is effortless to use. Call the wake word 'JARVIS' 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 2022. 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.

## **II. LITERATURE REVIEW**

### **2.1 Survey of Technology**

#### **A. 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.

#### **B. 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.

#### **C. 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.

#### **D. 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.

#### **E. 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 Voice Assistant The Future**

Voice search is implemented as a two-stage search procedure where string candidates generated by an automatic speech recognition (ASR) system are re-scored in order to identify the best matching entry from a potentially very large application specific database. Study provides a good example of how additional domain specific knowledge sources can be used with a domain independent ASR system to facilitate voice access to online search indices. As more data becomes available for a given speech recognition task, the natural way to improve recognition International Journal of Engineering Research and Technology. Accuracy is to train larger acoustic models. There is a nonparametric empirical

model that exploits abundant training data to directly learn pronunciation variation. Interpolating the empirical model with a parametric model yields the best performance, with a relative 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 training is likely to yield sharper distributions and a tighter fit to the data. Second, estimating word pronunciation co-occurrence counts 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.



Figure 1: Voice Assistant The Future

### III. SYSTEM DESIGN

#### 3.1 ER Diagram

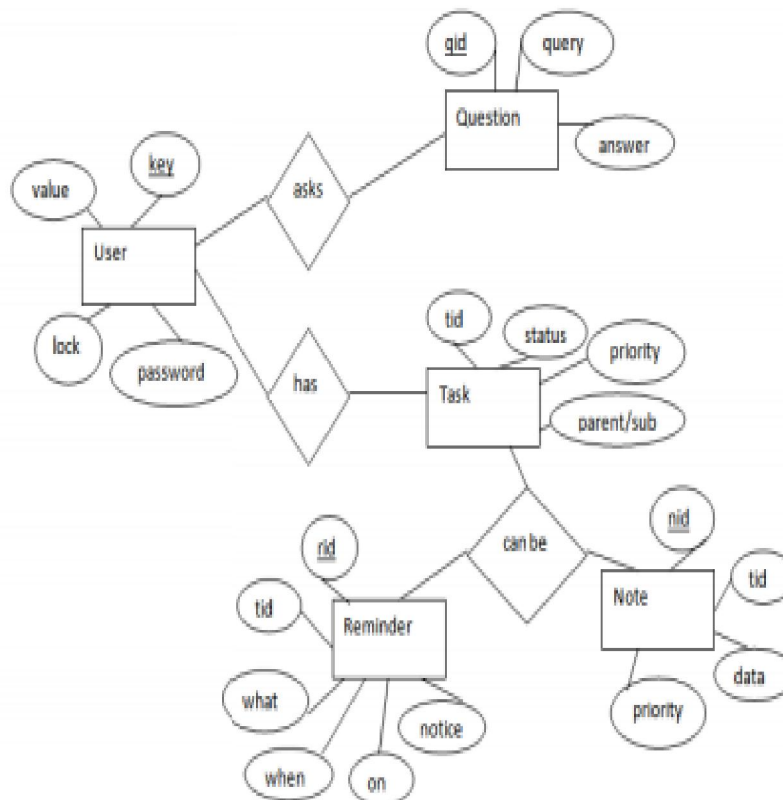
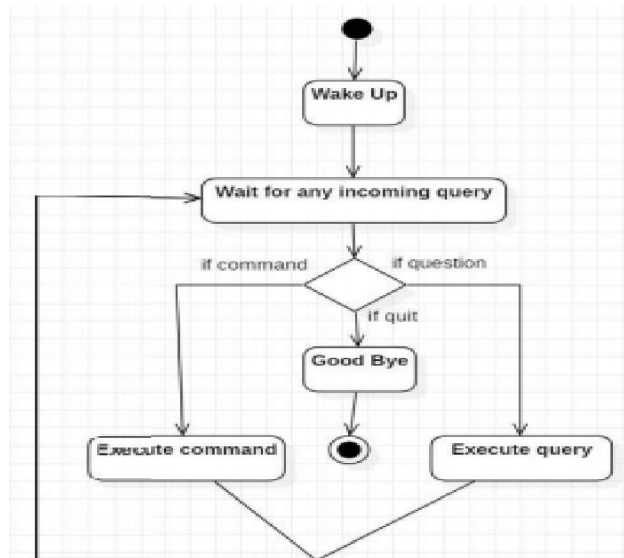


Figure 2: ER Diagram

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 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.

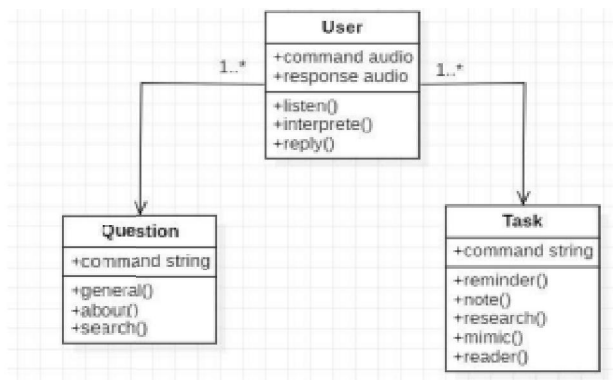
### 3.2 Activity Diagram



**Figure 3: 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.

### 3.3. Class Diagram

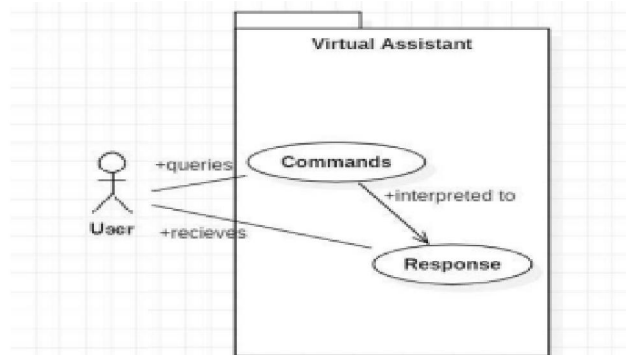


**Figure 4: Class Diagram**

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.

### 3.4 Use Case Diagram

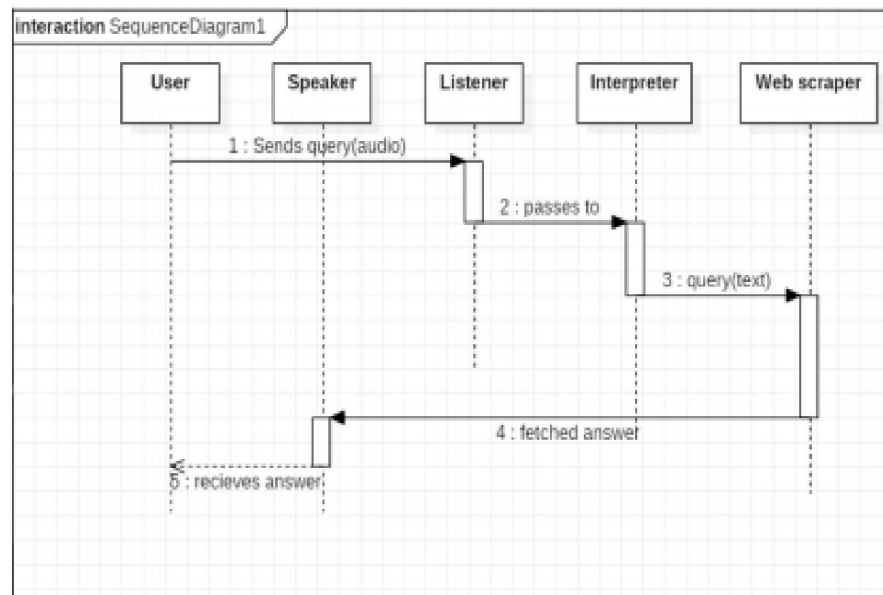


**Figure 5: Use Case Diagram**

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.

### 3.5 Sequence Diagram

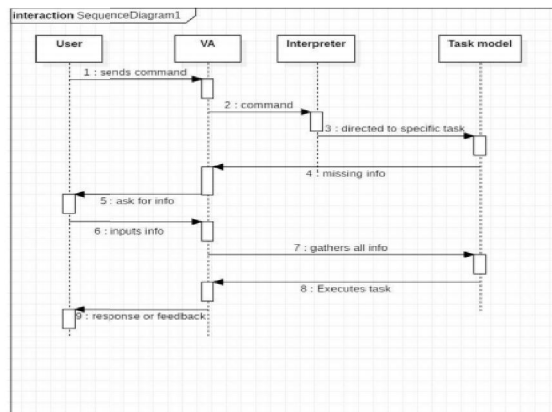
#### A. Sequence diagram for Query-Response



**Figure 6: 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.

## B. Sequence diagram for Task Execution

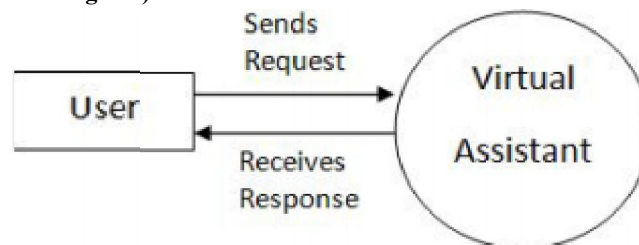


**Figure 7:** Sequence diagram for Task Execution

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 executor. 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.

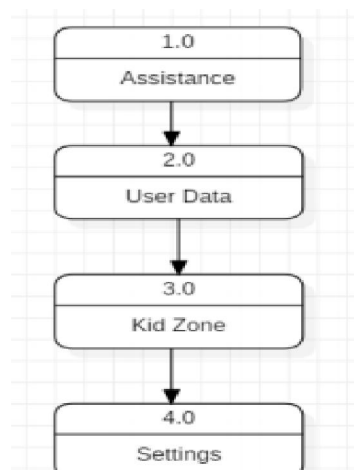
## 3.6 Data Flow Diagram

### A. DFD Level 0 (Context Level Diagram)



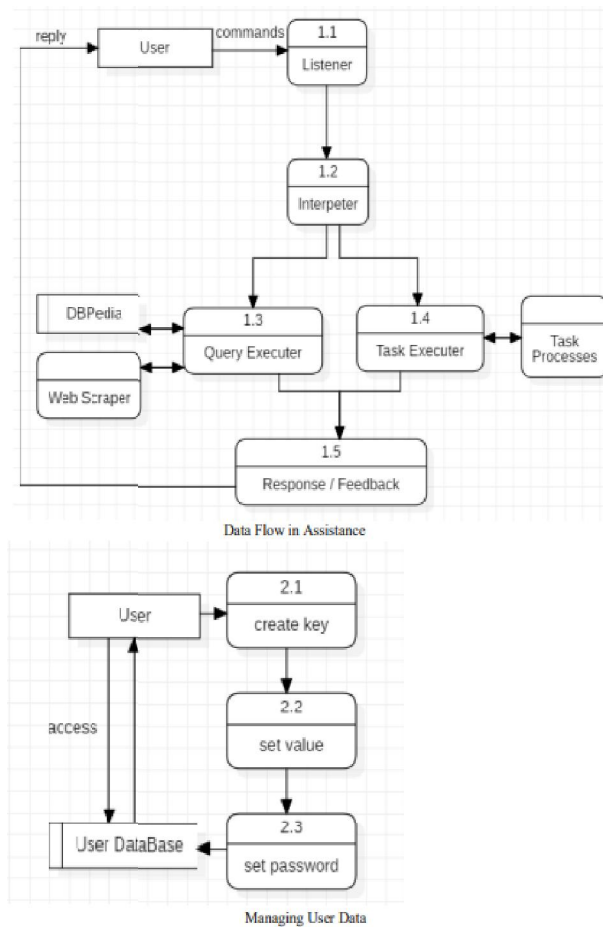
**Figure 8:** Data Flow Diagram L0

### B. DFD Level 1



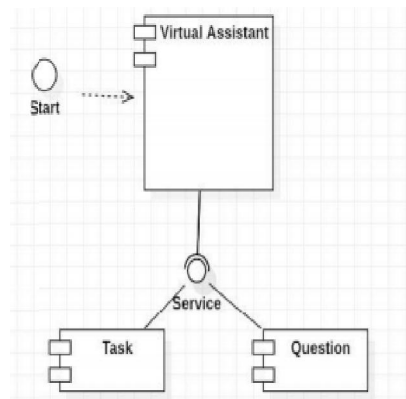
**Figure 9:** Data Flow Diagram L1

### C. DFD Level 2



**Figure 10: Data Flow Diagram L2**

### 3.7 Component Diagram



**Figure 11: Component Diagram**

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



#### **IV. RESULT AND DISCUSSION**

##### **4.1 Test Case Design**

###### **A. Test Case 1**

Test Title: Response Time

Test ID: T 1

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.

###### **B. Test Case 2**

Test Title: Accuracy

Test ID: T 2

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

###### **C. Test Case 3**

Test Title: Approximation

Test ID: T 3

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.

#### **V. CONCLUSION**

In this paper we have discussed a Voice Activated Personal Assistant developed using python. This assistant currently works online and performs basic tasks like weather updates, stream music, search Wikipedia, open desktop applications, etc. The functionality of the current system is limited to working online only. The upcoming updates of this assistant will have machine learning incorporated in the system which will result in better suggestions with IoT to control the nearby devices similar to what Amazon's Alexa does.

#### **VI. FUTURE SCOPE**

The virtual assistants which are currently available are fast and responsive but we still have to go a long way. The understanding and reliability of the current systems need to be improved a lot. The assistants available nowadays are still not reliable in critical scenarios. The future of these assistants will have the virtual assistants incorporated with Artificial Intelligence which includes Machine Learning, Neural Networks, etc. and IoT. With the incorporation of these technologies, we will be able to achieve new heights. What the virtual assistants can achieve is much beyond what we have achieved till now. Most of us have seen Jarvis, that is a virtual assistant developed by iron man which is although fictional but this has set new standards of what we can achieve using voice-activated virtual assistants.



**REFERENCES**

- [1] [www.stackoverflow.com](http://www.stackoverflow.com)
- [2] [www.pythonprogramming.net](http://www.pythonprogramming.net)
- [3] [www.codecademy.com](http://www.codecademy.com)
- [4] [www.tutorialspoint.com](http://www.tutorialspoint.com)
- [5] [www.google.co.in](http://www.google.co.in)
- [6] [www.python.org](http://www.python.org)
- [7] Python Programming - Kiran Gurbani
- [8] Learning Python - Mark Lutz
- [9] CS Dojo
- [10] edureka!
- [11] Codewithharry
- [12] Designing Personal Assistant Software for Task Management using Semantic Web Technologies and Knowledge Databases, Purushotham Botla
- [13] Python code for Artificial Intelligence: Foundations of Computational Agents, David L. Poole and Alan K. Mackworth