

Developing a Desktop Voice Assistant

Dr.PSV.Srinivasa Rao¹, A.Prem kumar², Anas Abdullah³, Ch.Rajeshwari⁴, K.Devika⁵

¹Dr.PSV.Srinivasa Rao (professor)

²A.Prem Kumar Department of Computer Science and Engineering (Joginpally b.r. Engineering College)

³Anas Abdullah Department of Computer Science and Engineering (Joginpally b.r. Engineering College)

⁴Ch.Rajeshwari Department of Computer Science and Engineering (Joginpally b.r. Engineering College)

⁵K.Devika Department of Computer Science and Engineering (Joginpally b.r. Engineering College)

ABSTRACT

The project aims to develop a personal-assistant for Laptops. Krishna draws its inspiration from virtual assistants like Cortana for Windows, and Siri for iOS but Cortana extends its services to some extent. Our project has been designed to provide a userfriendly interface for carrying out a variety of tasks by employing certain well-defined activities which made it 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 favorite IDE with the help of a single voice command. Users can interact with the assistant either through voice commands or using keyboard inputs. 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. In Python there is an API called Speech Recognition which allows us to convert speech into text. It was an interesting task to make our own assistant. By making this project, we realized that the concept of AI in every field is decreasing human effort and saving time.

Key Words: NLP(natural language processing), Speech to text conversion

1. INTRODUCTION

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 favorite IDE with the help of a single voice command. In the current scenario, advancement in technology is 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.

2. LITERATURE SURVEY

Voice Assistant has the long history. It has been in the phase of evolution since 1880.

In 1880 Alexander Graham Bell implemented further operations over Edison's phonograph, which his Volta Graphophone Company patented in 1886. Instead of foil graphophone was used, which allowed for longer recordings and higherquality playback. Edison also developed a wax version of the phonograph and both devices were used primarily for dictating letters and other documents.

In 1961 IBM introduced the IBM Shoebox, it's the first digital speech recognition tool. It recognized 16 words and digits 0 to 9. It was able perform mathematical functions and perform speech recognition.

In 1996 Microsoft introduces Clippy. Microsoft Clippy, it's also known as Clippit and officially recognized as Office Assistant, it was an intelligent user interface for Microsoft Office. It assisted the users in a number of interactive ways by appearing as a visualized character on the Office applications and offering help related to various operations of the Office Software. It was made available in the Microsoft Office for Windows in 1997 and in 2003 it was discontinued.

3.PROBLEM STATEMENT

To develop a software application that allows users to interact with their computer through voice commands, eliminating the need for manual keyboard and mouse input, providing a more efficient and accessible way to perform common desktop tasks like opening applications, searching the web, managing files, and controlling system settings, particularly for users who may have difficulty with traditional input methods

4. METHODOLOGY

A methodology for a desktop voice assistant project typically involves: defining project requirements, choosing a development platform and libraries, implementing speech recognition, natural language processing (NLP) to understand user commands,

integrating with system functionalities, building a user interface, and testing and refining the assistant.

4.1. Project Definition and Requirements Gathering:

Understand who will primarily use the assistant and what tasks they need it to perform (e.g., opening applications, setting reminders, controlling media, searching the web).

4.2 Technical Feasibility

The current state of NLP and speech recognition technology allows for accurate interpretation of user commands. Training machine learning models for personalized user experiences is feasible with available datasets. Compatibility with major platforms, including iOS and Android, is technically feasible. APIs provided by these platforms enable seamless integration of the voice assistant into various devices.

4.3 Market Feasibility

Consumer trends indicate an increasing preference for voice-activated services. The integration of voice assistants in smartphones, smart speakers, and other devices demonstrates a substantial market opportunity. While competition exists, the uniqueness of the proposed voice assistant, coupled with its personalized features, can differentiate it in the market. Continuous updates and improvements will be crucial to staying competitive.

4.4 Economic Feasibility

The cost of developing a personal voice assistant depends on factors such as software development, AI model training, and user interface design. However, the potential return on investment, given the demand for voice assistants, justifies the initial costs. Revenue can be generated through partnerships with device manufacturers, subscription models for premium features, and potentially through data analytics insights derived from user interactions (ensuring strict privacy compliance).

4.5 Legal and Ethical Feasibility

Adhering to data protection regulations is paramount. Implementing robust security measures, obtaining user consent, and anonymizing data ensure compliance with privacy laws.

5. SYSTEM DESIGN

5.1 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

- The upper part holds the name of the class.
- The middle part contains the attributes of the class.
- The bottom part gives the methods or operations the class can take or undertake.

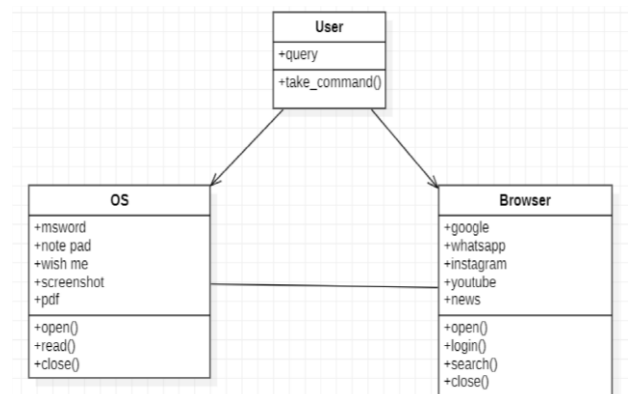


Fig 1.system design

5.2 Sequence Diagram

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

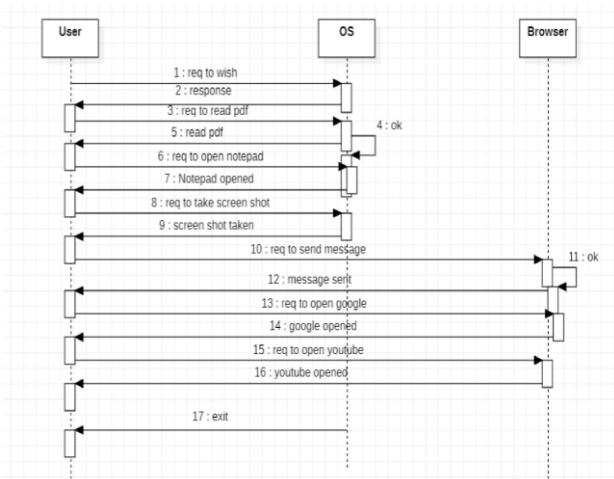


Fig 2.Sequence Diagram

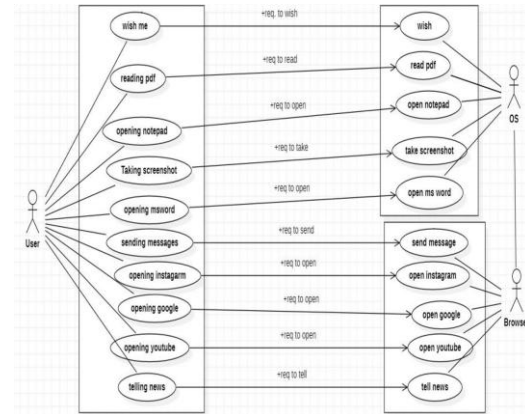


Fig 4.Use case diagram

5.3 Collaboration Diagram

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.

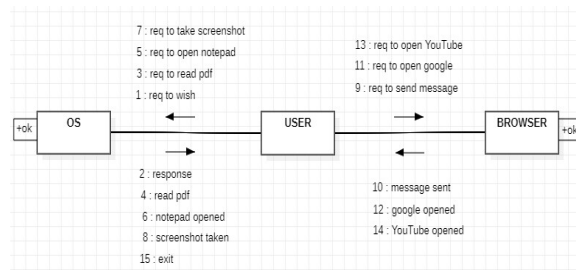


Fig 3.Collaboration Diagram

5.4 Use case diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

5.5 E-R Diagram

ER Model stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well-designed database. The ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database

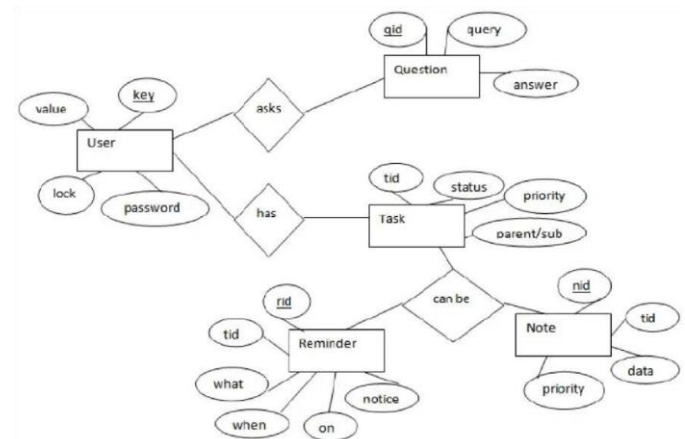


Fig 5.E-R diagram

6. RESULTS

```
C:\Users\nsiva\Desktop>python krishna.py
listening....
Recognizing....
user said : wake up
Good evening
2023-11-26
it's 08:20 PM
I am your krishna . please tell me how can i help you
listening....
Recognizing....
say that again please
listening....
Recognizing....
```

Fig 5. Output results

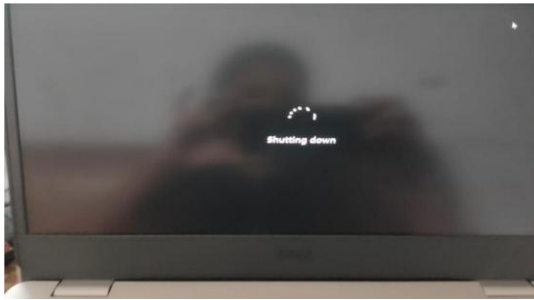


Fig 6.shutting down system

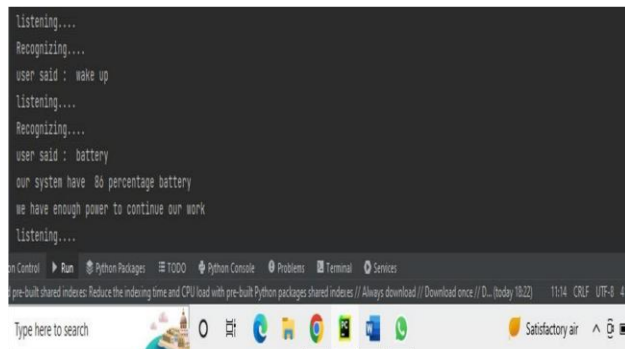
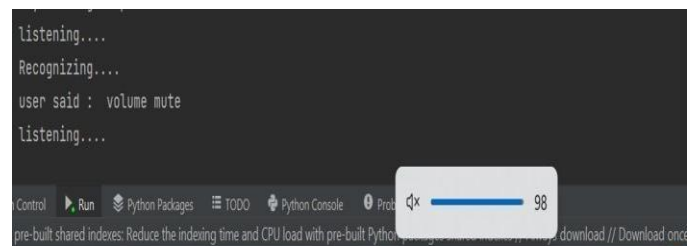
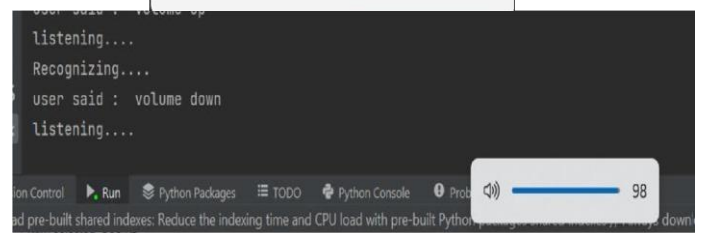
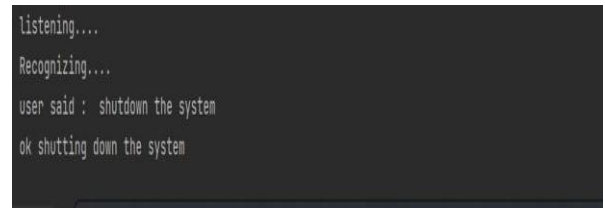


Fig 7.Telling battery percentage

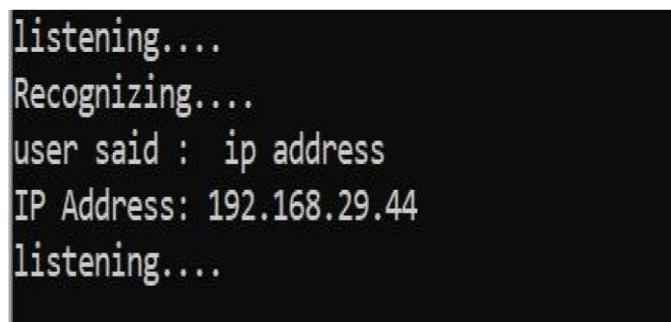


Fig 8. Telling IP address

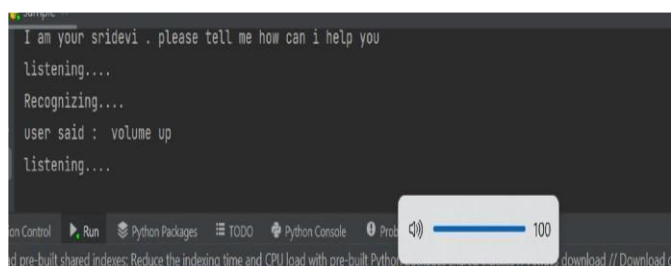


Fig 9. Volume up down and mute

7.CONCLUSION

Voice-Controlled Devices uses Natural Language Processing to process the language spoken by the human to understand and process the query and respond to the human with the result. It eases most of the tasks of the user like searching the web and opening applications on desktop In the process of making this project we realized that the concept of Artificial Intelligence in every field is decreasing human efforts and saving a lot of time.

REFERENCES

- [1] Rabiner Lawrence, Juang Bing-Hwang. Fundamentals of Speech Recognition Prentice Hall , New Jersey, 1993, ISBN 0-13-015157-2 2.
- [2] Deller John R., Jr., Hansen John J.L., Proakis John G. ,Discrete-Time Processing of Speech Signals, IEEE Press, ISBN 0-7803-5386-2 3.

[3]Hayes H. Monson,Statistical Digital Signal Processing and Modeling, John Wiley & Sons Inc. , Toronto, 1996, ISBN 0-471-59431-8 4.

[4]Proakis John G., Manolakis Dimitris G.,Digital Signal Processing, principles, algorithms, and applications, Third Edition, Prentice Hall , New Jersey, 1996, ISBN 0-13- 394338-9 5.

[5]Ashish Jain,Hohn Harris,Speaker identification using MFCC and HMM based techniques,university Of Florida,April 25,2004.

[6]<http://www.cse.unsw.edu.au/~waleed/phd/html/node38.html> , downloaded on 2 Oct 2012.