STARK

Rasumalla Niharika
Student
Department of Information Technology
B.V. Raju Institute of Technology
Affiliated to JNTUH
Vishnupur, Narsapur, Medak,
Telangana State, India
19211a1296@bvrit.ac.in

Ravuri Vijay
Student
Department of Information Technology
B.V. Raju Institute of Technology
Affiliated to JNTUH
Vishnupur, Narsapur, Medak,
Telangana State, India
19211a1297@bvrit.ac.in

K.Bhima
Associate Professor
Department of Information Technology
B.V. Raju Institute of Technology
Affiliated to JNTUH
Vishnupur, Narsapur, Medak,
Telangana State, India
bhima.k@bvrit.ac.in

Abstract

The Motive of stark is to make an effective voice assistant which performs actions based on user command .Google also created the most well-known programme, "Google Voice Search," which is utilised with Android phones. However, this application mostly utilises internet connections. Nonetheless, our proposed system may function both with and without an internet connection.

It is known as a Personal Assistant with Voice Recognition Intelligence, and it processes voice or text input from the user to provide output in a variety of ways, such as dictating the user's next course of action or a search result. It was created to offer a user-friendly interface for performing a number of operations by using specific, clearly defined instructions.

Keywords: *Speech Recognition, Android phone.*

I.INTRODUCTION

When artificial intelligence is applied to machines, it demonstrates to us the potential to think like humans. In this, a computer system is created in such a way that it frequently needs human intervention. As Python is a relatively new language, writing a voice assistant script in it is simple. The user's requirements can be taken into account when handling the assistant's instructions. The Alexa, Siri, etc., use speech recognition. Voice recognition is an API in Python that enables text transcription of spoken words. Making my own helper was a

fun project. It became simpler to send emails without typing a word and to do Google searches With the aid of a single voice command, you may open the browser and carry out several other daily operations like playing music and launching your preferred IDE. Today's technology have advanced to the point that they can accomplish any task just as successfully as we can, if not more so. Through creating this project, I came to understand how the idea of AI is saving time and reducing human effort in every industry.

Existing System:

- We are familiar with many existing voice assistants like Alexa, Siri, Google Assistant, Cortana which uses concept of voice recognition.
- Commands can be given and questions asked of the voice assistant which can perform the tasks or services requested.
- These assistants are no less than a human assistant but we can say that they are more effective and efficient to perform any task. The algorithm used to make these assistant focuses on the time complexities and reduces time.

Proposed system:

- It was an interesting task to make my own assistant.
- It became easier to perform daily tasks like playing music, opening your favorite IDE, News, Notepad, Games , shutdown, restart and many with the help of a single voice command.
- Can easily Launch applications.
- High secure because it is personal voice assistant.
- Cost Effective.

PyCharm is the IDE employed in this project. PyCharm was used to create each and every Python file, and this IDE made it simple to install all required packages. The following modules and libraries were utilised for this project: Speech Recognition, Python, Datetime, Wikipedia, Smtplib, pywhatkit, pyjokes, pyPDF2, pyautogui, and pyQt are a few examples. In order to have a dialogue with JARVIS that has a design and an intriguing appearance, I designed a live GUI.

II.IMPLEMENTATION

In this Implementation we used different algorithm where these can also be used in homobots, mini alexa in some mini pojects also.

The implementation phases includes

→Requirements gathering

→Analysis

→Data Collecion

→Modelling/Design

→Code Implementation

The UML DIAGRAMS are as follows

FIG-1:USECASE DIAGRAM

The above usecase diagram represents the enhanced application flow.Usecases diagram are one of the five diagrams in UML Diagrams.It shows a set of Usecases,Actors and their Relationship.It also contains notes and constraints.

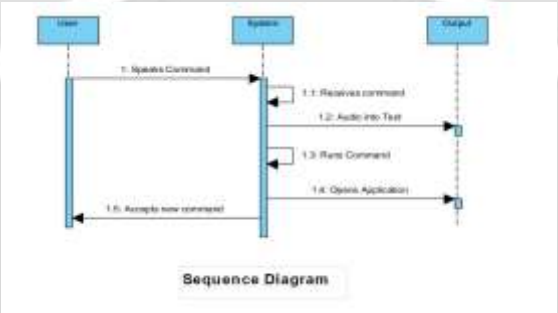

FIG-2:SEQUENCE DIAGRAM

III.RELATED WORK

The Work flow first began with the design and the architecture of the work.The architecture of our work flow is as follows.The architecture is common for our previous application and our enhanced application.The Stark lies in the start it doesn't need any trigger function.then it ask for the input then it takes the input ,once it received then it performs actions based on the user command. Then it generate output according to the input. And finally it exit the command.

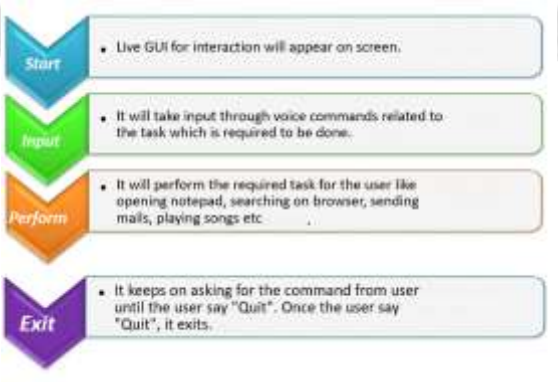

FIG-3:ARCHITECTURE

```
class Main(QMainWindow):
 def _init_(self):
 super()._init_()
```

```
 self.ui = Ui_MainWindow()
 self.ui.setupUi(self)
 self.ui.pushButton.clicked.connect(self.startTask)
 self.ui.pushButton_2.clicked.connect(self.close)
```

“VDESK”

Dept. of IT, BVRIT, Narsapur. 39

```
 def startTask(self):
 self.ui.movie = QtGui.QMovie(
 "../../Downloads/Iron man wallpaper - Imgur.gif")
 self.ui.label.setMovie(self.ui.movie)
 self.ui.movie.start()
 self.ui.movie = QtGui.QMovie(
 "../../Downloads/Jarvis Loading Screen on Make a GIF.gif")
 self.ui.label_2.setMovie(self.ui.movie)
 self.ui.movie.start()
 timer = QTimer(self)
 timer.timeout.connect(self.showTime)
 timer.start(1000)
 startExecution.start()
 def showTime(self):
 current_time = QTime.currentTime()
 current_date = QDate.currentDate()
 label_time = current_time.toString('hh:mm:ss')
 label_date = current_date.toString(Qt.ISODate)
 self.ui.textBrowser.setText(label_date)
 self.ui.textBrowser_2.setText(label_time)
app = QApplication(sys.argv)
stark = Main()
stark.show()
exit(app.exec_()
```

Above code is the sample code for the stark

IV.RESULTS

Now, lets us look how these stark look like by implementing the code the result is as follows:

FIG-1 MAIN VOICE PAGE

Here we have to give the input Eg: if we want to play swami swami song then we have give command play swami swami song , the result will be like this.

FIG-2 PLAYING SONG

If we give command open notepad then ,it opens notepad like this we have many features for this as we discussed in the introduction part.

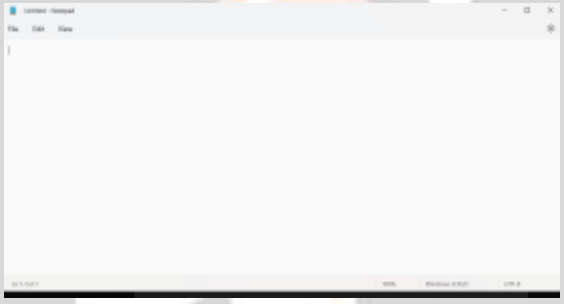

FIG-3 OPEN NOTEPAD

We can have conversation with that and we can play game also .

FIG-4 PLAY GAME

FIG-5 SENDING MAIL

V.CONCLUSION

Stark 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. And we want to add more functionalities and

This should also be handled in the normal phones also not only in the touch phones.

VI.REFERENCES

[1]: https://www.geeksforgeeks.org Voice-based intelligent virtual assistance for Windows
[2] - Nevon Projects Prateek Joshi
[3]. Python and artificial intelligence.
[4]Emotion Based Music Recommendation System- by R. Saranya, 2009.
[5] "Interacting with computers via voice automated speech detection and synthesis," Douglas, D.O. Shaughnessy, 2003