BUILDING A PERSONAL VOICE ASSISTANT USING PYTHON

Submitted in partial fulfilment of the required for the award of

Bachelors of Computer Science

by

Ananya A - 41735001

&

Akshaya J D - 41735007

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

MAY 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC I Approved by AICTE

---

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AKSHAYA J D (41735007)** who carried out the project entitled **"BUILDING A PERSONAL VOICE ASSISTANT USING PYTHON"** under our supervision from JANUARY 2023 to MAY 2023.

INTERNAL GUIDE

Ms.YUVASNEHA., M.Sc., Assistant professor

DEAN AND HEAD OF THE DEPARTMENT Dr.

REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I am, **AKSHAYA J D (41735007)** here by declare that the Project Report entitled **"BUILDING A PERSONAL VOICE ASSISTANT USING PYTHON"** done by us under the guidance of **MS.YUVASNEHA,M.Sc.,** Assistant Professor, Department of Computer **Science at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY,** Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfillment of the requirements for the award of Master of Science degree in Computer Science.

DATE: **AKSHAYA J D**

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sence of gratitude to my project guide **Ms.YUVASNEHA M.Sc.,** Assistant Professor, Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project.

AKSHAYA J D

ABSTRACT

Predictive analytics, a subset of advanced analytics, forecasts future outcomes using historical data, statistical modelling, data mining, and machine learning. Businesses employ predictive analytics to identify threats and opportunities using trends in this data, which delays adaption because the concept is used in so many different fields. Similarly, the target idea in a fraud detection programme may be one of the use case attributes named "fraudulent" with values of "yes" or "no" that signals whether a specific transaction is fraudulent. Or, a weather forecast application can have a number of target concepts like temperature, pressure, and humidity. In this paper, we offer the Recently, we predicted the silicon shortage due to a variety of reasons, including how major electrical companies handle those shortages and the usage of the predictive analytic approach applied in our project. This method specifically predicts using the input that is provided. Using the segment drift notion, we will anticipate the appropriate primacy products using two classifier algorithms, one of which is used to locate the relevant measurement materials.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

Nowadays almost all jobs are done digitally. We have Smartphones in our hands and nothing less than having the world in our hands. These days we don't even use our fingers. We are just talking about work and it is done. There are plans where we can say to the Father of the Scriptures, "I'll be late today." Text is also sent. That is the work of the Visible Assistant. It also supports specialized functions such as booking a flight, or getting the cheapest book online from various e-commerce sites and provides an order booking link, which facilitates automatic search, discovery, and online ordering services. Wise assistants based on the word need a persuasive word or a wake-up call to make the listener active, which is followed by a command. In my project the rising name is AVOKA. We have many visible assistants, such as Apple's Siri, Amazon's Alexa, and Microsoft's Cortana. In this project, the wake-up name is selected for AVOKA. Virtual Assistants can provide several services including,

- The weather.
- Scheduling appointment time.
- Trip planning.
- Play music, movies, etc.
- Indicates the time of day.
- Manage emails.
- Open applications.

CHAPTER 2

LITERATURE SURVEY

S.NO	IMPLEMENTED BY	TITLE	YEAR
1.	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	1880
2.	IBM	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.	1961
3.	Carnegie Mellon	Carnegie Mellon completed the Harpy Program. It could able to understand about 1000 words. Harpy processed speech that followed preprogrammed vocabulary, pronunciation and grammar structures.	1972
4.	Dragon	Dragon launched,the first speech recognition module for consumers for $6,ooo (Indian currency= 496.27 in current date).	1990
5.	Microsoft	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	1996

		the Microsoft Office for Windows in 1997 and in 2003 it was discontinued.	
6.	Apple	Apple introduced Siri. Voice queries, control based on gesture, focustracking and natural language user interface for answering the questions, making recommendations and perform operations by passing on the requests to as set of internet services were used in Siri.	2011
7.	Google	Google launched Google. Google Now proactively delivered information to users to predict information they might need in the form of informational cards which was based on the users' search habits and other factors. For Android and iOS, Goggle Now was a feature of Google search embedded in Google app.	2012
8.	Microsoft	Cortana was introduced by Microsoft. Cortana is a virtual assistant which uses the Bing search engine to perform tasks such as setting reminders and answering questions for the user. Depending upon the software programs and region in which its used, Cortana is currently available in English, Chinese, French, German or Italian, Portuguese, Spanish and Japanese language editions.	2013
9.	Amazon	Amazon Alexa or Alexa is a virtual assistant technology largely based on a Polish speech synthesizer named Ivona, bought by Amazon. It was first used in the Amazon Echo smart speaker and the Echo Dot, Echo Studio and Amazon Tap speakers developed by Amazon Lab126.	2014

10.	Microsoft	Cortana on windows 10 desktops and mobile devices was introduces by Microsoft. While in US Amazon officially launched Amazon Echo. Alexa Skills kit was introduced by Amazon.	2015
11.	Sound Hound	voice-powered virtual assistant app was introduced by Sound Hound, HOUND. Sound Hound is grandly known for its music recognition app, which listens to songs and identifies them. Amazon launched Amazon Echo Dot and Amazon Tap. Google introduced the Google Assistant as a part of the messaging app, Allo. In the same year the virtual assistant startup Viv was obtained by Samsung.	2016
13.	Giancarlo Iannizzotto, Lucia LoBello, AndreaNucita, Giorgio Mario Gtasso	A Vision and Speech Enabled, Customizable, Virtual Assistant for Smart Environments. Stated the software architecture for building lightweight, vision and speech enabled virtual assistant for smart phone and automation application. A complete prototype application was build featuring a realistic graphic assistant able to show facial expression and enabled with speech synthesis and recognition.	2018

14.	Deepak Shende, Ria Umahiya, Monika Raghorte, Aishwarya Bhisikar, Anup Bhange	AI based Voice Assistant Using Python. Stated the new insights of natural human-machine interaction, in which machine would learn how to understand the humans language. It also expressed the principles of functioning of voice assistants, it's main shortcomings and limitations, methos of creating local voice assistant without using cloud services is described.	2019
15.	Subhas S, Prajwal N, Siddesh S, Ullas A, Santhosh B	Artificial Intelligent-Based Voice Assistant. Stated a voice assistant gathering the audio from the microphone and get converted into text, later it sent through GTTS (Google Text To Speech.). GTTS engine will convert text into audio file in English language , then that audio sound is played using the play sound package of python programming language.	2020
16.	Nivedita Singh, Dr. Diwakar Yagyasen, Mr. Surya Vikram Singh, Gaurav	Voice Assistant Using Python. stated a voice assistant using Python which allows the user to run any type of command in linux without interating with keyboard. It performs basic tasks such as weather updates, stream music,	2021
	Kumar, Harshit Agarwal	search Wikipedia, open desktop applications etc.	

17.	Manjusha Jadhav, Krushna kalyankar, Ganesh Narkhede, Swapnil Kharose	Survey On Smart Virtual Voice Assistant. stated natural language processing algorithm that helps machines to engage in communication using natural human language in many forms. It also connects to World Wide Web to provide the results that the user required.	2022
18.	Vishal Kumar Dhanraj, Lokeshkriplani, Semal Mahajan	Research Paper On Desktop Voice Assistant.Stated working of a vice assistant without using. cloud services, which will allow the expansion of devices in the future. It can perform any kind of task in exchange of commands given by the user without any error, it will listen to the users ‘voice only and will not be activated from environment noise’.	2022

CHAPTER 3

METHODOLOGY

3.1 HARDWARE SPECIFICTION

Processor: 11th Gen Intel(R) Core (TM) i5-1135G7

Speed: 2.40GHz 2.42 GHz

RAM8.00 GB (7.73 GB usable)

Hard Disk: 256 GB

3.2 SOFTWARE SPECIFICATION

Operating system: Windows 7 / 8 / 8.1 / 10 / 11

Coding Language: PYTHON

IDE: PyCharm

3.3 EXISITING SYSTEM

In the existing system, the User has to do his work manually. If the user needs to do a search in google, he/she must turn on the computer and do research to obtain What he/she Needed The user should do all his work manually by himself like Booking an appointment, setting an alarm, doing a google search, the task assigned to the user etc... The whole part of the user’s life depends on himself and he needs to do all his work by himself in his day-to-day life. Alternative solution for the predictive algorithm Extra trees and hist gradient algorithms which is one of the predictive analysis methods.

3.4 DISADVANTAGES

- Time-consuming: Manually performing tasks such as searching online or setting appointments can be time-consuming, taking away valuable time from other activities.
- Prone to errors: Human error is inevitable when tasks are done manually, leading to mistakes in appointments, missed deadlines, or incorrect information gathered from searches.
- Limited efficiency: Human capabilities have limits, and manual tasks may not be executed as efficiently as automated processes, resulting in slower completion times and decreased productivity.
- Dependency on user availability: The user must be available to perform tasks, which can be challenging if they have a busy schedule or are unable to access their computer or device.
- Increased stress: Having to remember and manually complete various tasks can lead to increased stress and mental burden, especially when managing multiple responsibilities simultaneously.
- Risk of forgetfulness: Without automated reminders or prompts, there's a higher risk of forgetting important tasks or appointments, leading to missed opportunities or negative consequences.
- Lack of flexibility: Manual processes lack the flexibility to adapt to changing circumstances or priorities quickly, resulting in potential delays or difficulties in adjusting plans.
- Reduced accessibility: People with physical disabilities or limitations may struggle with manual tasks, limiting their ability to perform essential activities independently.
- Potential for inefficiency: The manual nature of tasks can lead to inefficiencies in resource allocation, as valuable time and effort are spent on repetitive or mundane activities instead of more strategic endeavors.
- Difficulty in tracking and organization: Without automated systems to track and organize tasks, it can be challenging to maintain an overview of responsibilities, deadlines, and completed activities, leading to confusion and disorganization.

3.5 PROPOSED SYSTEM

In order to overcome the existing problem we are making this Virtual Assistant which Makes the users day-to-day tasks simple and easy. The user is able to do his work by Adding offline functionality to the version could enhance usability, and would let users personalize their experience. Switching users in the voice assistant would be handy for shared devices, ensuring a tailored interaction for each user. It’s all about making the experience more versatile and user-friendly!

3.6 ADVANTAGES

- Time-saving: By automating day-to-day tasks, the virtual assistant saves users time, allowing them to focus on more important activities or enjoy leisure time.
- Error reduction: Automation reduces the likelihood of human error, leading to more accurate task completion and information retrieval.
- Increased efficiency: With automated processes, tasks can be completed more quickly and efficiently, boosting overall productivity and effectiveness.
- Enhanced accessibility: Users with physical disabilities or limitations can benefit from the accessibility features of the virtual assistant, making it easier for them to perform tasks independently.
- Offline functionality: Adding offline functionality enhances usability, ensuring that users can still access and utilize the virtual assistant's features even without an internet connection.
- Versatility: By making the experience more versatile, the virtual assistant can adapt to different users' needs and preferences, providing a more flexible and user-friendly experience.
- Improved organization: The virtual assistant helps users stay organized by keeping track of tasks, appointments, and deadlines, providing reminders and notifications as needed.
- Streamlined workflow: With the virtual assistant handling routine tasks, users can streamline their workflow and focus on higher-value activities, leading to greater overall efficiency and satisfaction.

CHAPTER 4

SYSTEM DESIGN

4.1 ARCHITECTURE DIAGRAM

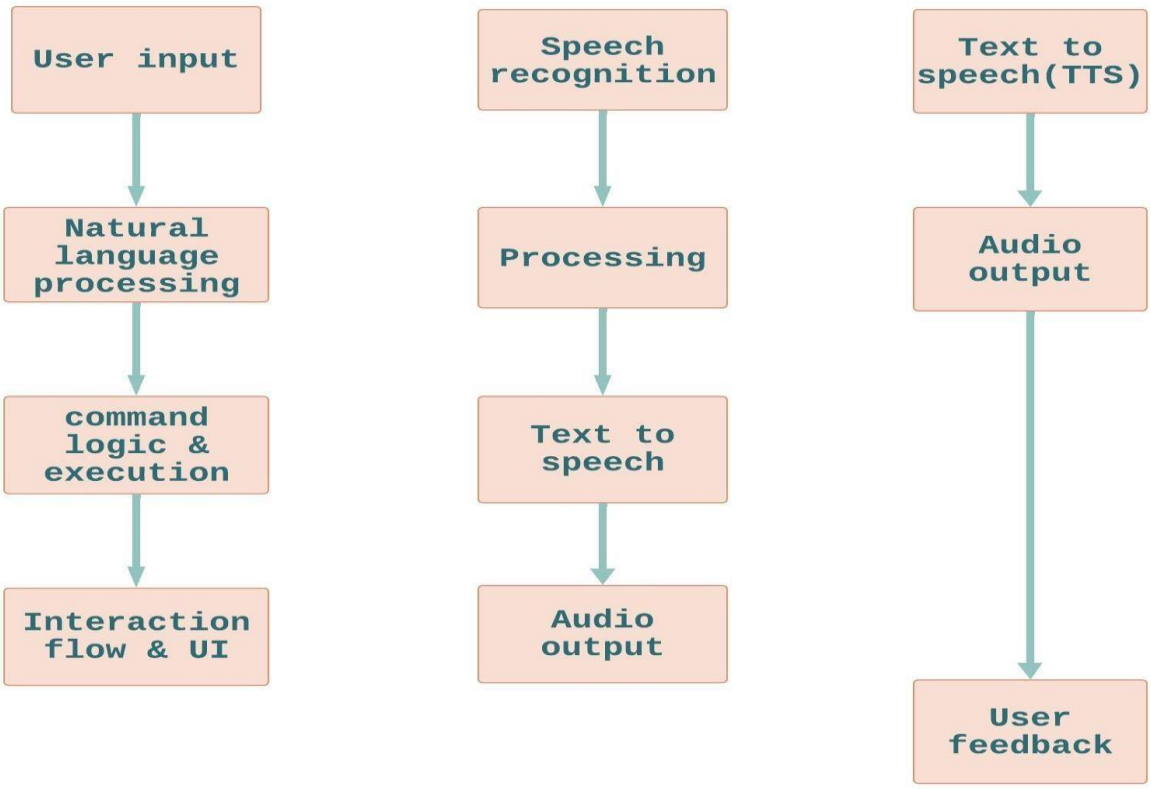


Fig.4.1. Architecture diagram

4.2 ACTIVITY DIAGRAM

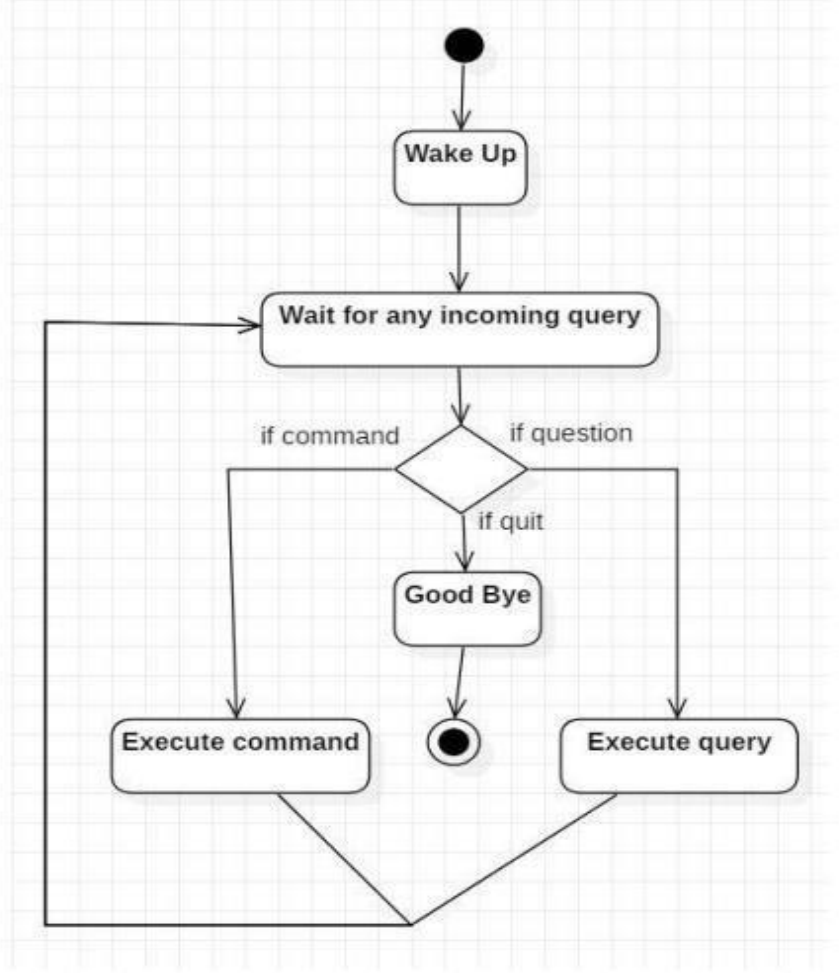


Fig.4.2. Activity Diagram

Initially, the system is in idle mode. As it receives any wake up cal 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 CASE DIAGRAM:

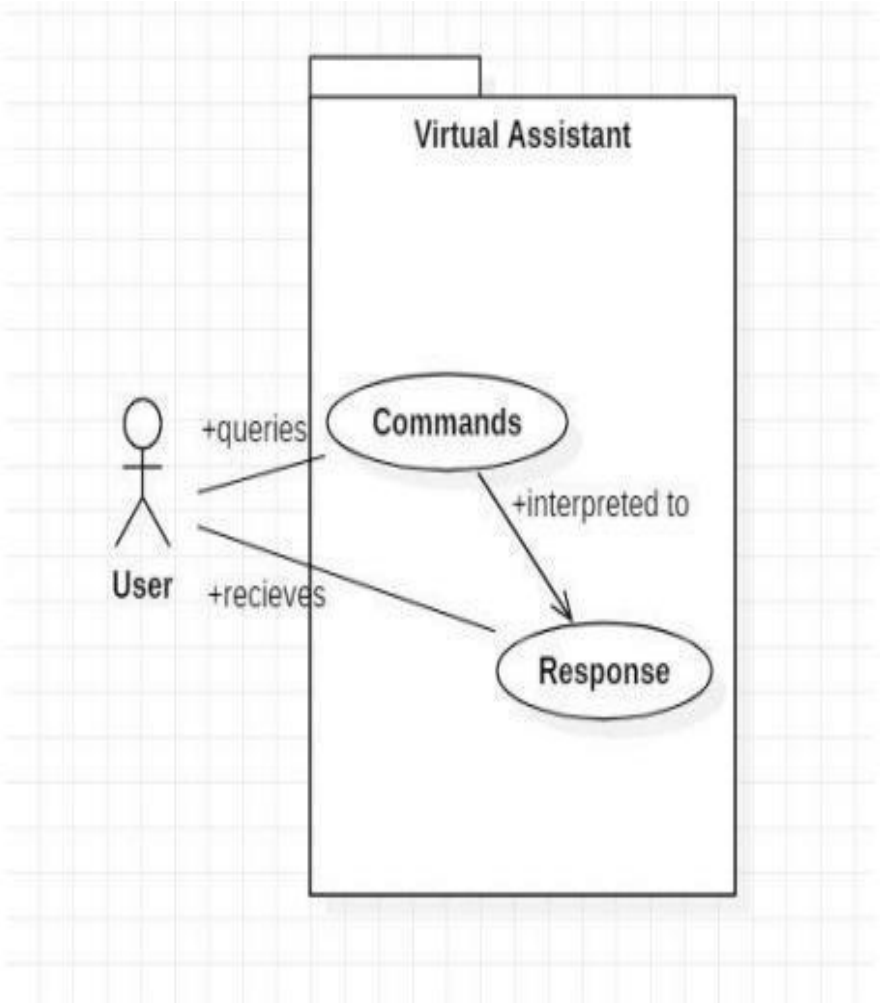


Fig.4.3. Case diagram

- 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.

4.4 DATA FLOW DIAGRAM

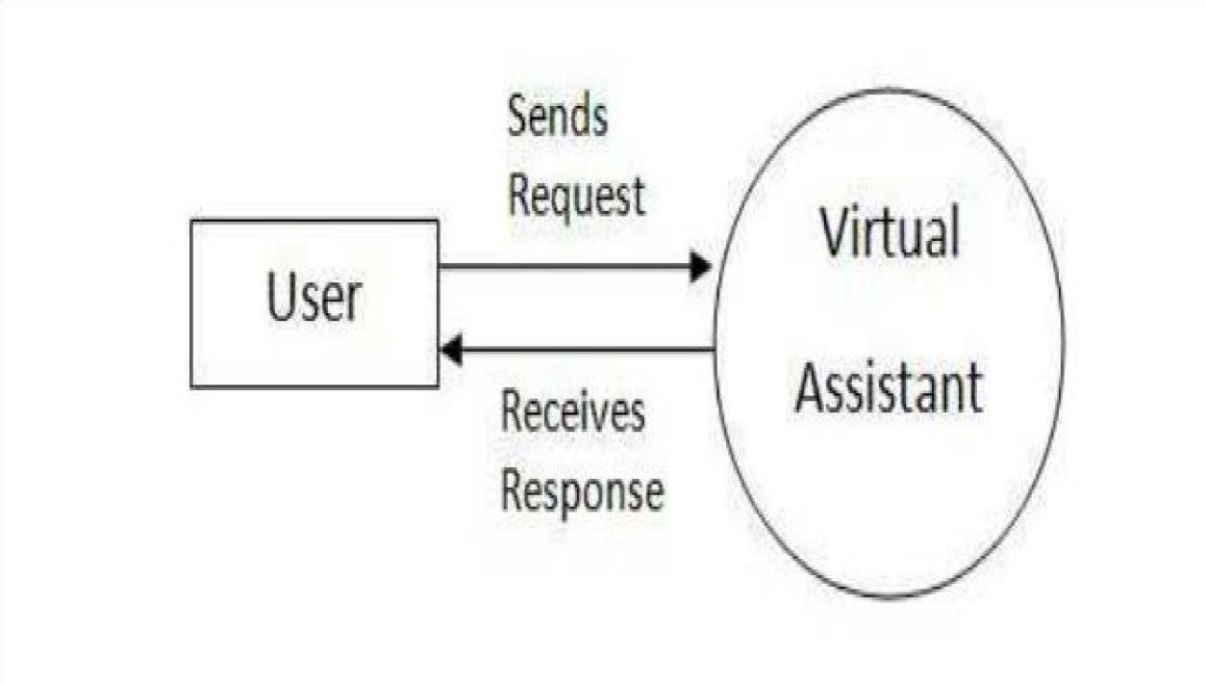


Fig.4.4. (a) Dataflow diagram (level 0)

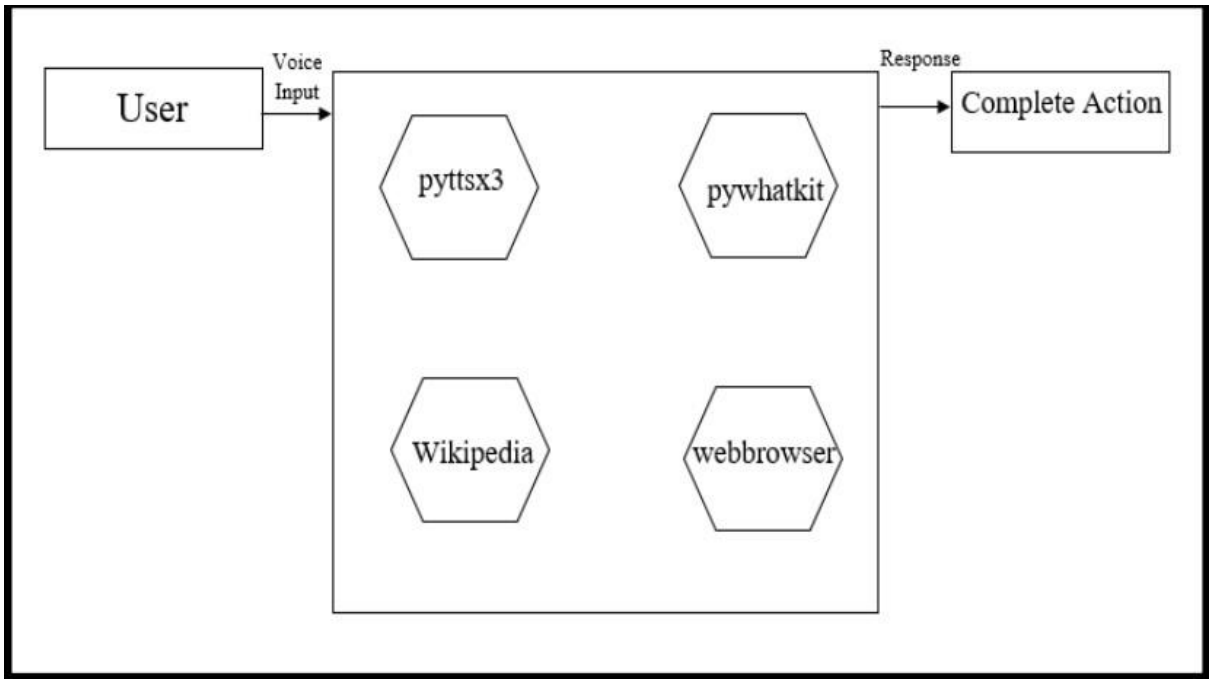


Fig. 4.4. (b) Dataflow diagram (level 1)

DFD is graphical representation of system which give detail information about data flow between input and output. As level increases it elaborates detail information about data flow.

4.5 COMPONENT DIAGRAM

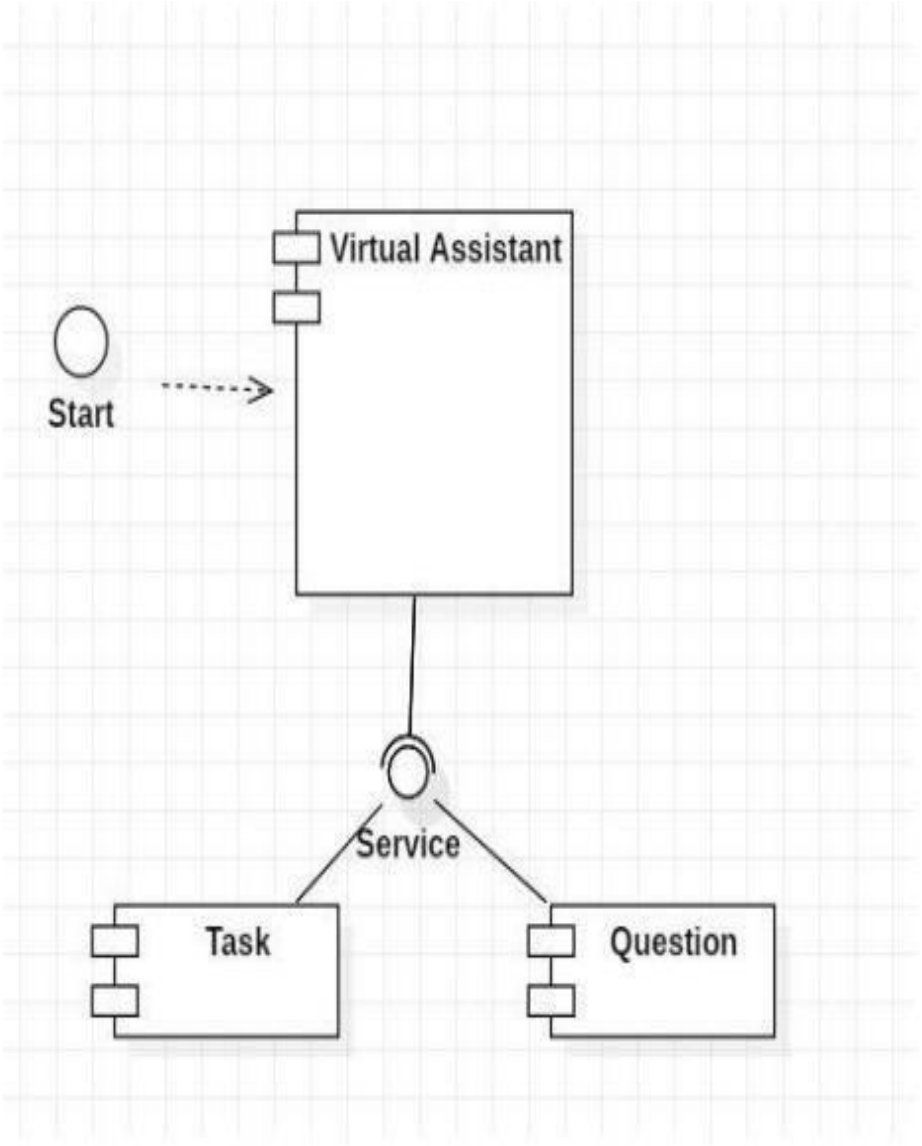


Fig.4.5 Component diagram

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

4.6 SEQUENCE DIAGRAM

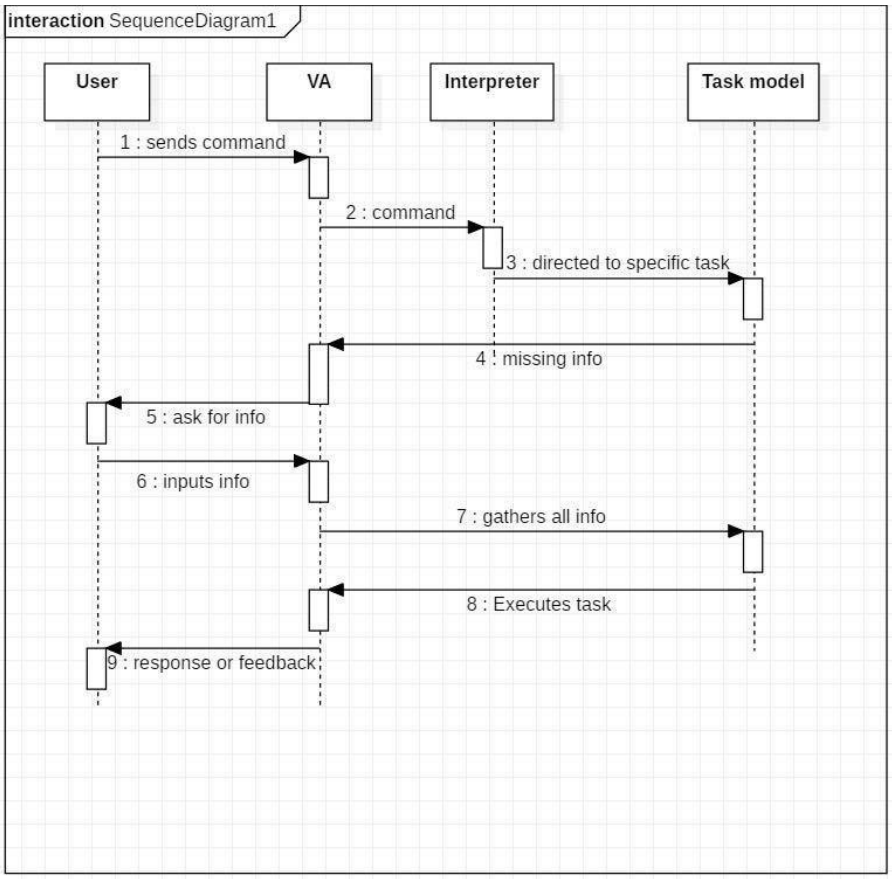


Fig. 4.6. Sequence diagram

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.

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 INTRODUCTION TO PYTHON

About Python

- Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatics in the Netherlands as a successor to ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989; Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator for Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coughlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project. Guido van Rossum has since then withdrawn his nomination for the 2020 Steering council.

- Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode.

- Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

- Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported.
- Python 3.9.2 and 3.8.8 were expeditedas all versions of Python (including 2.7) had security issues, leading to possible remote code execution and web cache poisoning.

Key points

- Readability: Python syntax is easy to read and understand, promoting code clarity and maintainability.
- Expressiveness: Python allows developers to express concepts in fewer lines of code compared to other languages
- Interpreted: Python code is executed line by line by the Python interpreter, making development and debugging easier.
- Dynamically Typed: Python is dynamically typed, meaning variable types are determined at runtime.
- Cross-platform: Python code can run on various operating systems without modification.
- Extensible: Python can be easily extended with modules and libraries written in other languages.
- High-level Data Structures: Python provides built-in support for high-level data structures such as lists, dictionaries, and tuples.
- Object-oriented: Python supports object-oriented programming, allowing developers to organize their code into classes and objects.
- Standard Library: Python comes with a comprehensive standard library, providing many useful modules and functions out of the box.
- Community and Ecosystem: Python has a large and active community, with a vast ecosystem of third-party libraries and frameworks.

Python Syntax compared to other programming languages

- Python uses indentation to indicate code blocks, eliminating the need for curly braces or keywords like "begin" and "end".
- Python syntax is clean and concise, making it easy to read and write.
- It has a minimalistic approach to syntax, reducing the cognitive load on developers.
- Python uses dynamic typing, allowing for flexible and expressive code.

- Compared to languages like C++ or Java, Python syntax is less verbose.
- Python has a more natural language-like syntax, making it accessible to beginners.
- Python supports both procedural and object-oriented programming paradigms seamlessly.
- Python has a rich set of built-in data types and functions, reducing the need for external libraries.

Importance of python to the Internet

Python is a general-purpose language — sometimes referred to as utilitarian — which is designed to be simple to read and write. The point that it's not a complex language is important. The designers placed less of an emphasis on conventional syntax, which makes it easier to work with, even for non-programmers or developers. Furthermore, because it's considered truly universal and used to meet various development needs, it's a language that offers a lot of options to programmers in general. If they begin working with

Python for one job or career, they can easily jump to another, even if it's in an unrelated industry. The language is used for system operations, web development, server and administrative tools, deployment, scientific modelling and much more .But, surprisingly, many developers don't pick up Python as their primary language. Because it's so easy to use and learn, they choose it as a second or third language. This may be another reason why it's so popular among developers .Plus, it just so happens that one of the biggest tech companies in the world — Google — uses the language for a number of their applications. They even have a developer portal devoted to Python, with free classes offered including exercises, lecture videos and more. In addition, the rise in the use of the Django framework for web development and a decline in popularity of PHP has also contributed to Python's success.

Design of python

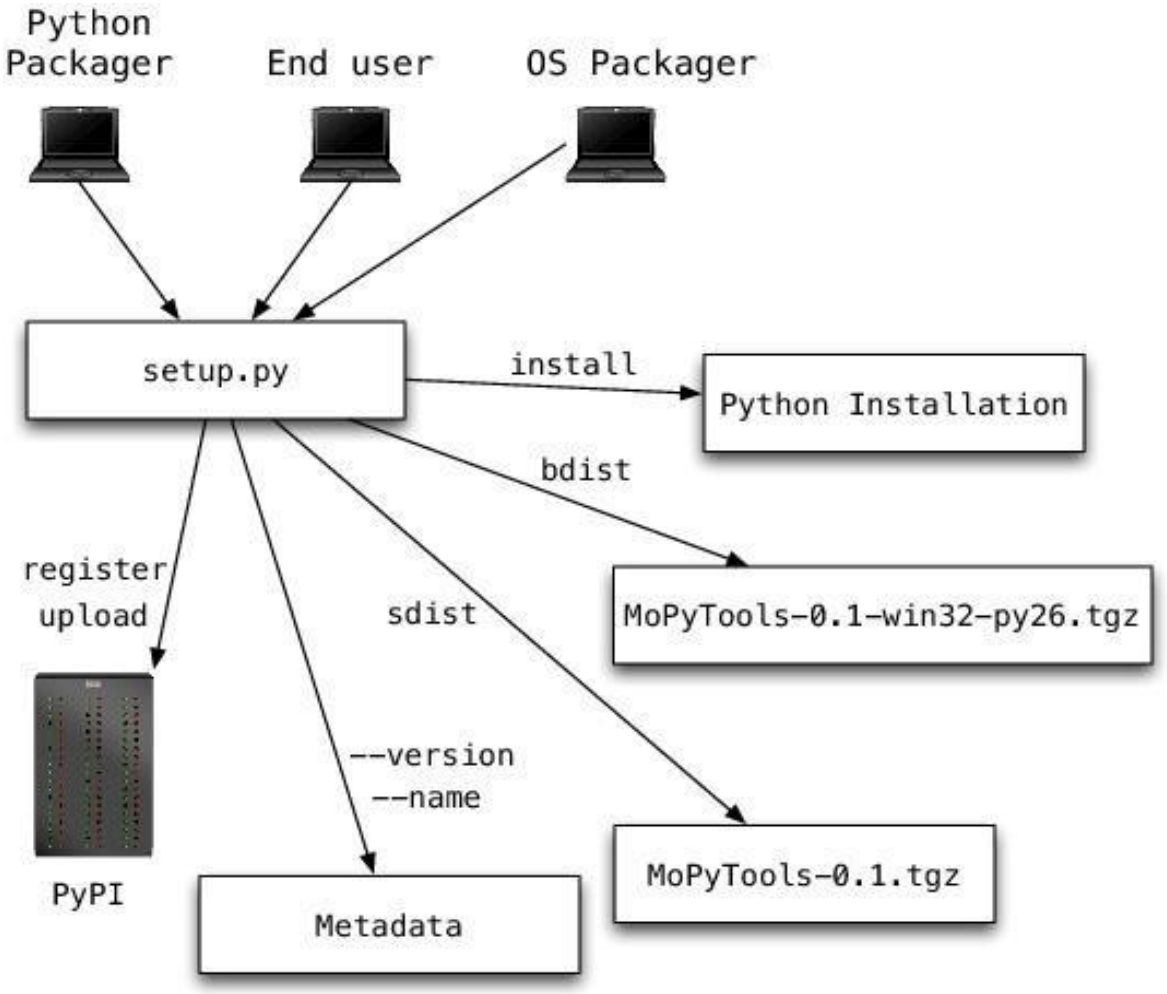


Fig.5.1 design of python

PyPy's Python Interpreter is written in Python and implements the full Python language. This interpreter very closely emulates the behaviour of Python. It contains the following key components:

- a bytecode compiler responsible for producing Python code objects from the source code of a user application;
- a bytecode evaluator responsible for interpreting Python code objects;

- a standard object space, responsible for creating and manipulating the Python objects seen by the application.

The bytecode compiler is the pre-processing phase that produces a compact bytecode format via a chain of flexible passes (tokenizer, lexer, parser, abstract syntax tree builder, bytecode generator). The bytecode evaluator interprets this bytecode. It does most of its work by delegating all actual manipulations of user objects to the object space. The latter can be thought of as the library of built-in types. It defines the implementation of the user objects, like integers and lists, as well as the operations between them, like addition or truthvaluetesting. This division between bytecode evaluator and object space gives a lot of flexibility. One can plug in different object spaces to get different or enriched behaviours of the Python objects.

Simple

Python was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Python will oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object oriented

Python was not designed to be source-code compatible with any other language. This allowed the Python team the freedom to design with a blank state. One outcome of this was a clean, usable, pragmatic approach to objects. The object model in Python is simple and easy to extend, while simple types, such as integers, are kept as high-performance nonobjects.

Robust

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Python. Python is strictly typed language; it checks your code at compile time and runtime.

Python virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Python program, all run-time errors can and should be managed by your program

5.2 SOFTWARE REQUIREMENT SPECIFICATION

The purpose of this document is to present a detailed description of the Web application system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

PURPOSE

The purpose of this Software Requirement Specification (SRS) is to help the project. It is provided with some requirements which are used in the Transaction Mercator System. All parts; design, coding and testing will be prepared with helping of SRS. The purpose of this document is to detail the requirements placed on the Transaction Mercator System and serves as a contract between the customer and the developers as to what is to be expected of the stock exchange, and how the components of the system are working with each other with external systems.

This document will be checked by the group member's supervisor and it will corrected by members if supervisor orders.

5.3 DEVELOPERS RESPONSIBILITIES OVERVIEW

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system. • Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

5.4 FUNCTIONAL REQUIREMENTS

- Following is a list of functionalities of the browsing enabled system.
- An Activity with a UI that allows you to browser settings. Provide a second Activity that allows users to access the share with permission from the administrator. Handle activity lifecycle appropriately. A precondition for any points in this part of the grade is code that compiles and runs.
- Your application should allow a user to browse the shares, buy and sell the shares with specific metadata. The assignment requires you to create a UI for browsing and a UI for integrating the two.
- The Net beans provide a number of useful layout components, views, and tools that you may want to use to create your location browser. As with the final project, you should design your application to only use the buttons on the Key board and mouse as input. Your application should use the Key board, Mouse and keywords.

5.5 NON-FUNCTIONAL REQUIREMENTS

- The system should be supported Net beans. The member should use the System browser. Each member should have a separate system.
- The system should ask the username and password to open the application. It doesn’t permit to unregistered user to access the System.

- The system should have Role based System functions access. Approval Process has to be defined.
- The system should have Modular customization components so that they can be reused across the implementation.
- These are the mainly following:
- Secure access of confidential data. 24 X 7 availability
- Flexible service based architecture will be highly desirable for future extension

5.6 PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the required specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

5.7 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Five types of feasibility study are taken into consideration.

1. **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, 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.

2. **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.

3. **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, it won't cost too much.

4. **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.

5. **Cultural feasibility:** It deals with compatibility of the project with cultural environment. Virtual assistant is built in accordance with the general culture. 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.

CHAPTER 6

MODULES DISCRIPTION

- Speech Recognition
- Natural Language Processing (NLP)
- Text-to-Speech (TTS)
- Audio Playback
- Other Useful Libraries

6.1 Speech recognition

Description: Speech Recognition is a module that allows your program to recognize speech input from various sources, such as microphone or audio files, and convert it into text.

Functionality: It provides functions to capture audio input, process it using speech recognition algorithms, and return the recognized text.

Usage: You can use this module to enable voice commands in your applications, transcribe spoken audio to text, or build voice-controlled interfaces.

6.2 Natural Language Processing (NLP)

Description: Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and humans through natural language.

Functionality: This module includes tools and techniques for analyzing, understanding, and generating human language.

Usage: NLP can be used for tasks such as text classification, sentiment analysis, named entity recognition, machine translation, and text summarization.

6.3 Text-to-Speech (TTS)

Description: Text-to-Speech (TTS) is a module that synthesizes human-like speech from written text.

Functionality: It converts text input into spoken audio, allowing your application to produce speech output.

Usage: TTS can be used to create accessibility features for visually impaired users, develop voice-based virtual assistants, or generate audio content from text-based sources.

6.4 Audio Playback

Description: Audio Playback module enables your program to play audio files in various formats.

Functionality: It provides functions to load, play, pause, resume, and stop audio playback.

Usage: You can use this module to incorporate sound effects, background music, or voice prompts into your applications, games, or multimedia projects.

6.5 Other Useful Libraries

1.Requests:

Description: The requests module is essential for making HTTP requests in Python. It allows your voice assistant to communicate with web services and APIs to fetch data from the internet.

Usage: You can use requests to access various online resources such as weather forecasts, news updates, traffic information, or to interact with social media platforms and other web services.

Example: Your voice assistant can use requests to fetch weather information from a weather API based on the user's location or retrieve news headlines from a news website's API.

2. Wikipedia:

Description: The Wikipedia module provides an easy interface for accessing and parsing Wikipedia content. It allows your voice assistant to retrieve information from Wikipedia articles.

Usage: You can use Wikipedia to answer user queries by searching for relevant information on Wikipedia, providing quick access to encyclopedic knowledge.

Example: When the user asks a question like "Who is Albert Einstein?" or "What is Python programming language?", your voice assistant can use the wikipedia module to fetch and summarize information from relevant Wikipedia articles.

3. Datetime:

Description: The datetime module provides functions and classes for working with dates and times in Python. It allows your voice assistant to handle time-related operations, such as displaying the current time or scheduling tasks.

Usage: You can use datetime to retrieve the current date and time, calculate time differences, format dates and times, or perform date arithmetic.

Example: Your voice assistant can use datetime to greet the user with a personalized message based on the time of day (e.g., "Good morning, User!") or to remind the user of upcoming events or appointments.

5.Subprocess:

Description: The subprocess module allows your Python program to spawn new processes, execute system commands, and interact with the operating system's shell.

Usage: You can use subprocess to run external commands or scripts, launch other programs, or perform system-related tasks.

Example: Your voice assistant can use subprocess to execute system commands to control system settings, launch applications, or perform system operations such as shutting down or restarting the compu

CHAPTER 7

APPENDICES

7.1 SAMPLE CODE

```
import speech_recognition as sr
import pyttsx3 import random
import pyjokes import
webbrowser import os from
selenium import webdriver
import pywhatkit as kt import
pywhatkit import datetime
import wikipedia


def say(text):     engine = pyttsx3.init()
newVoiceRate = 145
engine.setProperty('rate', newVoiceRate)
voice = engine.getProperty('voices')
engine.setProperty('voice', voice[1].id)
engine.say(text)     engine.runAndWait()
```

```
def takecommand():
r=sr.Recognizer()     with
sr.Microphone() as source:
        r.adjust_for_ambient_noise(source,duration=1)
print("Listening....")
        r.pause_threshold=1
        audio=r.listen(source)
        try:
            print("Recognizing...")
            text=r.recognize_google(audio,language='en-in')
print(f"User said : {text}\n")         except Exception as e
:
            print(e)
            print("cannot recognize your voice")
return "None"         return text

def  respond(text):          if
"how are you" in text:
        say("I  am  doing  well,  thank  you  for  asking.")
say("How about yourself?")

    elif  'are  you  there'in  text:
say("for you! always boss")

    elif 'I am doing well'in text:
        say("thats great!. how may i assist you!")


    elif 'I am doing good'in text:
        say("thats great!. how may i assist you!")
```

```
    elif 'I am great' in text:
        say("thats great!. how may i assist you!")

    elif 'I am ok' in text:
        say("thats great!. how may i assist you!")

    elif 'I have been better' in text:
        say("thats great!. how may i assist you!")

    elif 'I am fantastic' in text:
        say("thats great!. how may i assist you!")

    elif 'I am feeling pretty good today' in text:
say("thats great!. how may i assist you!")

    elif 'I am doing alright' in text:
        say("thats great!. how may i assist you!")

    elif 'avoca'in text:
        say("Greetings!. how may i assist you!")
elif 'avoka'in text:
        say("Greetings!. how may i assist you!")
elif 'avaka'in text:
        say("Greetings!. how may i assist you!")
elif 'avaca'in text:
        say("Greetings!. how may i assist you!")
elif "tell about yourself" in text:
        say("My name is Avokaa!. I am a virtual assistant AI program. And i am here to
assist you in anyway i can")  elif "good morning" in text:  say("a very good
morning!")  say("how are you boss!")  elif "good afternoon" in text:
say("good noon boss!")  say("how shall i assist you")  elif "good evening" in
```

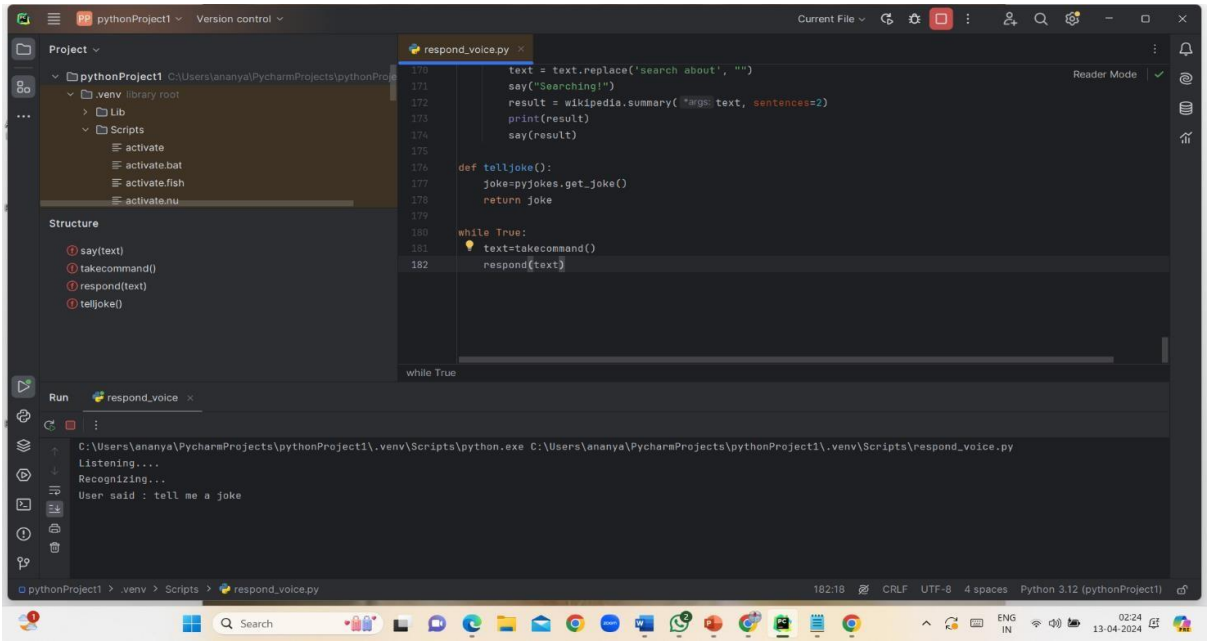
```
text:        say("good evening boss!")        say("how shall i assist you")     elif "good
night" in text:        say("good night boss!")
       say("i hope you had busy day today.")
       say("Please have a good nap!.")
elif "thanks for the help" in text:
say("you are wellcome.")        say("its
my pleasure to assist you")     elif
"thanks" in text:
       say("you are wellcome.")        say("its
my pleasure to assist you")     elif "thank
you" in text:        say("you are wellcome.")
say("its my pleasure to assist you")     elif
"tell me a joke" in text:
engine2=pyttsx3.init()        voice =
engine2.getProperty('voices')
engine2.setProperty('rate',145)
engine2.setProperty('voice', voice[1].id)
engine2.say(telljoke())
engine2.runAndWait()

   elif "tell me another joke" in text:
       engine2=pyttsx3.init()        voice =
engine2.getProperty('voices')
engine2.setProperty('rate',145)
engine2.setProperty('voice', voice[1].id)
engine2.say(telljoke())
engine2.runAndWait()

   elif    "open    browser"    in    text:
say("Sure boss! I am on it.")
```

```
        path = r"C:\Program Files\Google\Chrome\Application\chrome.exe"
os.startfile(path)

    elif "close browser" in text:
        say("Sure boss! I am on it.")
        os.startfile('C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe')
os.system("TASKKILL /F /IM chrome.exe")

    elif 'open' in text:
        text = text.replace('open', "")
if 'Notepad' in text:
            say("Sure! I am on it.")
            os.startfile('C:\\WINDOWS\\system32\\notepad.exe')

        elif 'paint' or 'Paint' in text:
say("Sure! I am on it.")
                os.startfile('C:\\WINDOWS\\system32\\mspaint.exe')

    elif 'close' in text:
        text=text.replace('close',"")
if 'Notepad'or 'notepad' in text:
            say("Sure! will close it now boss.")            path
= r"C:\WINDOWS\system32\notepad.exe"
os.system("TASKKILL /F /IM notepad.exe")
        elif 'paint' in
text:
            say("Sure! will close it now boss.")
            os.startfile('C:\\WINDOWS\\system32\\mspaint.exe')
os.system("TASKKILL /F /IM mspaint.exe")

    elif 'browse YouTube' in text:
```

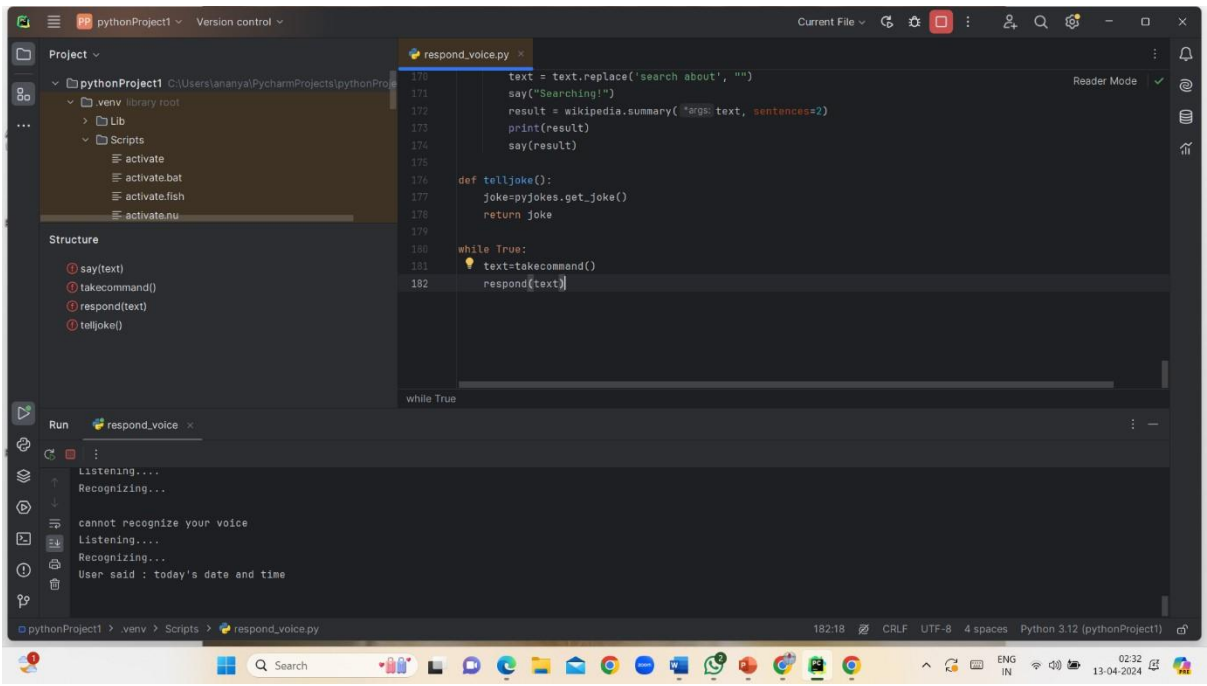
```
        text = text.replace('search Youtube', "")
say("Searching!")
        os.startfile('C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe')
url='https://www.youtube.com/'  webbrowser.open_new_tab(url)

    elif 'time' in text:
        time = datetime.datetime.now().strftime('%I:%M %p')
say('Current time is ' + time)

    elif 'date' in text:
        date = datetime.datetime.now().strftime('%Y %B %d %A')
say('Today date is ' + date)


    elif 'search about' in text:
        text = text.replace('search about', "")
say("Searching!")
        result = wikipedia.summary(text, sentences=2)
print(result)  say(result)

def telljoke():
joke=pyjokes.get_joke()  return
joke

while True:
text=takecommand()
respond(text)
```

7.2 SCREEN SHOTS

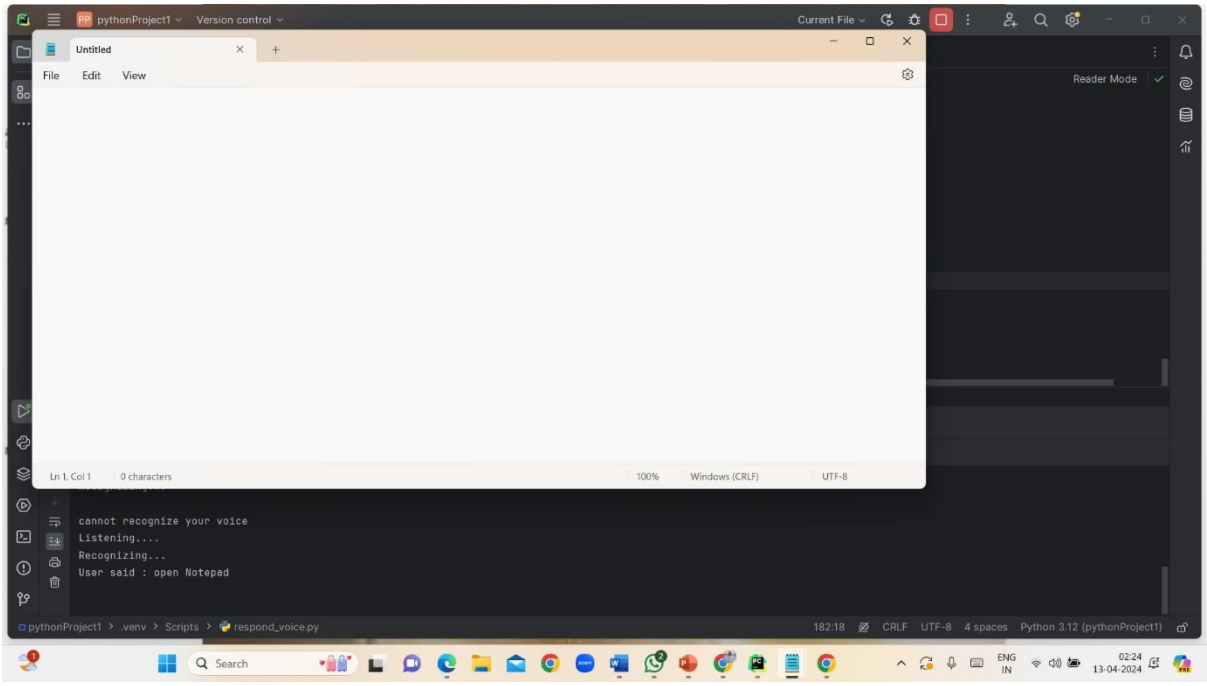
TELL A JOKE



TELLS CURRENT DATE AND TIME



OPENS NOTEPAD



CHAPTER 8

CONCLUSION AND FUTURE WORK

As stated before, "voice assistant is one of the biggest problem solver" and you can see that in the proposals with the examples that it is in fact one of the biggest problem solver of the current world. We can see that voice assistant is one of the major evolving artificial intelligence in the current world once again on seeing the proposal examples because at the past, the best feature which a voice assistant had was telling the date and searching the web and giving the results but now look at the functions that it can do so with this, we can say that it is a evolving software in the current world. The main idea is to develop the assistant even more advanced than it is now and make it the best ai in the world which will save an ample of time for its users. I would like to conclude with the statement that we will try our best and give one of the best voice assistants which we are able to.

FUTURE WORK

We are entering the era of implementing voice-activated technologies to remain relevant and competitive. Voice-activation technology is vital not only for businesses to stay relevant with their target customers, but also for internal operations. Technology may be utilized to automate human operations, saving time for everyone. Routine operations, such as sending basic emails or scheduling appointments, can be completed more quickly, with less effort, and without the use of a computer, just by employing a simple voice command. People can multitask as a result, enhancing their productivity. Furthermore, relieving employees from hours of tedious administrative tasks allows them to devote more time to strategy meetings, brainstorming sessions, and other jobs that need creativity and human interaction.

1) Sending Emails with a voice assistant:

Emails, as we all know, are very crucial for communication because they can be used for any professional contact, and the finest service for sending and receiving emails is, as we all know, GMAIL. Gmail is a Google-created free email service. Gmail can be

accessed over the web or using third-party apps that use the POP or IMAP protocols to synchronize email content.

To integrate Gmail with Voice Assistant we have to utilize Gmail API. The Gmail API allows you to access and control threads, messages, and labels in your Gmail mailbox.

2) Scheduling appointments using a voice assistant:

The demands on our time increase as our company grows. A growing number of people want to meet with us. We have a growing number of people who rely on us. We must check in on certain projects or set aside time to chat with possible business leads. There won't be enough hours in the day if we keep doing things the old way.

We need to get a better handle on our full-time schedule and devise a strategy for arranging appointments that doesn't interfere with our most critical job. By working with a virtual scheduler or, in other words, a virtual assistant, we let someone else worry about the organization and prioritize our schedule while we focus on the work.

3) Improved Interface of a voice assistant (VUI):

Voice user interfaces (VUIs) allow users to interact with a system by speaking commands. VUIs include virtual assistants like Amazon's Alexa and Apple's Siri. The real advantage of a VUI is that it allows users to interact with a product without using their hands or their eyes while focusing on anything else.

Other benefits of a Voice user interface (VUI):

Speed and Efficiency:

Hands-free interactions are possible with VUIs. This method of interaction eliminates the need to click buttons or tap on the screen. The major means of human communication is

speech. People have been using speech to form relationships for ages. As a result, solutions that allow customers to do the same are extremely valuable. Furthermore, even for experienced texters, dictating text messages has been demonstrated to be faster than typing. Hands-free interactions, at least in some circumstances, save time and boost efficiency.

Intuitiveness and convenience:

Intuitive user flow is required of high-quality VUIs, and technical advancements are expected to continue to improve the intuitiveness of voice interfaces. Compared to graphical UIs, VUIs require less cognitive effort from the user. Furthermore, everyone – from a small child to your grandmother – can communicate. As a result, VUI designers are in a better position than GUI designers, who run the danger of producing incomprehensible menus and exposing users to the agony of poor interface design. Customers are unlikely to need to be instructed on how to utilize the technology by VUI makers. People can instead ask their voice assistant for assistance.

CHAPTER 8

REFERENCES

[1] K. Noda, H. Arie, Y. Suga, T. Ogata, Multimodal integration learning of robot behavior using deep neural networks, Elsevier: Robotics and Autonomous Systems, 2014.

[2] Artificial intelligence (AI), sometimes called machine intelligence. https://en.wikipedia.org/wiki/Artificial_intelligence.

[3] Deepak Shende, RiaUmahiya, Monika Raghorte, AishwaryaBhisikar, AnupBhange, "AI Based Voice Assistant Using Python", Journal of Emerging Technologies and Innovative Research (JETIR), February 2019, Volume 6, Issue 2.

[4] J. B. Allen, "From lord rayleigh to shannon: How do humans decode speech," in International Conference on Acoustics, Speech and Signal Processing, 2002.

[5] John Levis and Ruslan Suvorov, "Automatic Speech Recognition".

[6] B.H. Juang and Lawrence R. Rabiner, "Automatic Speech Recognition - A Brief History of the Technology Development".

[7] AbhayDekate, ChaitanyaKulkarni, RohanKilledar, "Study of Voice Controlled Personal Assistant Device", International Journal of Computer Trends and Technology (IJCTT) – Volume 42 Number 1 – December 2016.