

**SRM IST, NCR CAMPUS, MODINAGAR**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**PRACTICAL FILE**

**Object Oriented Analysis & Design**

**(PCA20C07J)**

**IInd Year, III Semester**

**Session: 2022-23**

**Submitted To: Submitted By:**

**Dr. Preeti Bala Tripathi Jatin Sharma**

**(Assistant Professor) REG NO –RA2132241030008**

**MCA**

**SRM INSTITUTE OF SCIENCE & TECHNOLOGY NCR CAMPUS, MODINAGAR**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**Register No:**

**BONAFIDE CERTIFICATE**

Certified to be the bonafide record of the work done by **Jatin Sharma** of MCA, Second year, 3nd Semester for the award of MCA degree course in the Department of Computer Applicationsin **Object Oriented Analysis and Design** during the Academic year-2022-23 .

**LAB IN-CHARGE HEAD OF DEPARTMENT**

*Submitted for the university examination held on\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**INTERNAL EXAMINER-I INTERNAL EXAMINER-II**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **TITLE OF EXPERIMENT** | **DATE** | **PAGE NO.** | **SIGNATURE**  **/REMARKS** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Experiment-1**

**Aim: The problem identified**

A virtual assistant is simply an assistant who works remotely. As technological advancements in the 1990s brought reliable internet to more homes, businesses realized they didn't have to bring an employee into the office to get work done, and this led to the first virtual assistants.

Today the developments of artificial intelligence (AI) systems that can organize a natural human- machine interaction (through voice, communication, gestures, facial expressions, etc.) are gaining in popularity. One of the most studied and popular was the direction of interaction, based on the understanding of the machine by the machine of the natural human language. It is no longer a human who learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behavior and trying to become his personalized assistant.

A Voice Assistant is a piece of software that communicates to the user audibly, and responds to spoken commands. It's technology like Google Home, Siri and Alexa that can be used to literally talk to a computer, a smartphone, or another device. It can perform tasks or services for an individual based on commands or questions. Users can ask their assistants questions, media play back via voice, and manage other basic tasks such as news, jocks, facts and search information on Wikipedia and google with verbal commands.

**Experiment-2**

**Aim: The solution to tackle that problem identified.**

The main objective of this project is to build a program that will be able to service to humans like a personal assistant. It will provide number of features of day-to-day life, so we don’t have to type all the basic question on our web browser or like we want to listen some song but I have to type the name of that song but through voice assistance I don’t have to do this, I simply say my assistance to play song and It will do it. There is no task which cans we done by voice command.

In this modern era, day to day life became smarter and interlinked with technology. We already know some voice assistance like google, Siri. Etc. This project works on voice input and gives output through voice and displays the text on the screen. The main agenda of our voice assistance makes people smart and give instant and computed results. The voice assistance takes the voice input through our microphone (Bluetooth and wired microphone) and it converts our voice into computer understandable language gives the required solutions and answers which are asked by the user. This assistance connects with the World Wide Web to provide results that the user has questioned. Natural Language processing algorithm helps computer machines to engage in communication using natural human language in many forms.

**Experiment-3**

**Aim: SRS Document**

**Software Requirement Specification (SRS) Format** as name suggests, is complete specification and description of requirements of software that needs to be fulfilled for successful development of software system. These requirements can be functional as well as non-functional depending upon type of requirement. The interaction between different customers and contractor is done because its necessary to fully understand needs of customers.

**Tools/ Platform Hardware & Software Requirements**

Software Requirement-

PythonV3, Speech Recognition, pyttsx3, pyjokes, datetime, selenium – webdriver, requests, random, randfacts, time, os, webbrowser (Package)

Python installed with Vs-Code (IDE)

Platform-

Windows 8.1 or above

Hardware Requirements-

Minimun-i3 processor,4GB RAM,1GB Space at least Recommeneded-i5 processor,8 GB RAM,10 GB ROM

Structure of Project

Step1-

User gives command to Computer

Step2-

Computer converts that command to Text

Step 3:

According to text now it search and find suitable match for that command

Step 4:

It will show the output

**Project Module Description:**

In this Project physically disable person or the person having less knowledge about Technology or how to access the devices like Laptop or mobile, can easily access the device with their voice or speech command. This application includes the functions and services such as: location services, music player service, checking weather, Google search, Wikipedia search, Funfacts. The list below indicates the information and the requirements of each individual function.

A virtual Assistance can provide a number of benefits to your business. When budgets are tight, hiring a virtual PA can be an effective way to cut business costs. By outsourcing your PA activity, your business can reduce (or remove completely) some of the costs associated with a full-time employee, such as benefits, training and utility.

In this project we can play song, search information on Wikipedia and Google, search location on map, also get daily news, also listen jocks and facts and also send WhatsApp message to our contact and also found about temperature outside and there will be many more future possibility’s.

**MODULE WORKED ON IN THIS PROJECT:**

Speech Recognition:

Library for performing speech recognition, with support for several engines and APIs, online and offline. Speech recognition software is a computer program that’s trained to take the input of human speech, interpret it, and transcribe it into text.

PYTTSX3(Python text toSpeech):

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech.

Selenium:

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others. A browser-driver then executes these scripts on a browser-Instance on your device.

Request:

Requests is an Apache2 Licensed HTTP library, written in Python. It is designed to be used by humans to interact with the language. This means you don’t have to manually add query strings to URLs, or form-encode your POST data. Don’t worry if that made no sense to you. It will in due time. Requests will allow you to send HTTP/1.1 requests using Python. With it, you can add content like headers, form data, multipart files, and parameters via simple Python libraries. It also allows you to access the response data of Python in the same way.

Random:

This module implements pseudo-random number generators for various distributions. For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement.

RandFact:

Randfacts is a python library that generates random facts. You can user and facts.get\_fact() to return a random fun fact. Disclaimer: Facts are not guaranteed to be true.

Pyjokes:

Python supports creation of random jokes using one of its libraries. Let us explore it a little more, Pyjokes is a python library that is used to create one-line jokes for programmers. Informally, it can also be referred as a funpython library which is pretty simple to use. Let us see how you can actually use it to perform the required task.

Date-Time:

This python module is helpful in representing Date and time on any place or time zone. It has number of functionalities like day, time, month, time zone, second, minutes, hour, etc.

Web browser:

This module support web browser and it is help us to access and show our output on web browser. Like we use packet in our program which shows website, with the help of this packet we can show answer or functionality of a package on Web browser.

Pywhatkit:

It a Python library with various helpful features. It is an easy-to-use library which does not requires you to do some additional setup. pywhatkit is a Python library for sending WhatsApp messages at a certain time, it has several other features too.

Beautiful Soup:

It is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work. We are using this to find out the temperature of any location.

**Experiment-4**

**Aim: Identify which use case diagram is to be made related to the software for the problem identified**

**Use Case:**

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

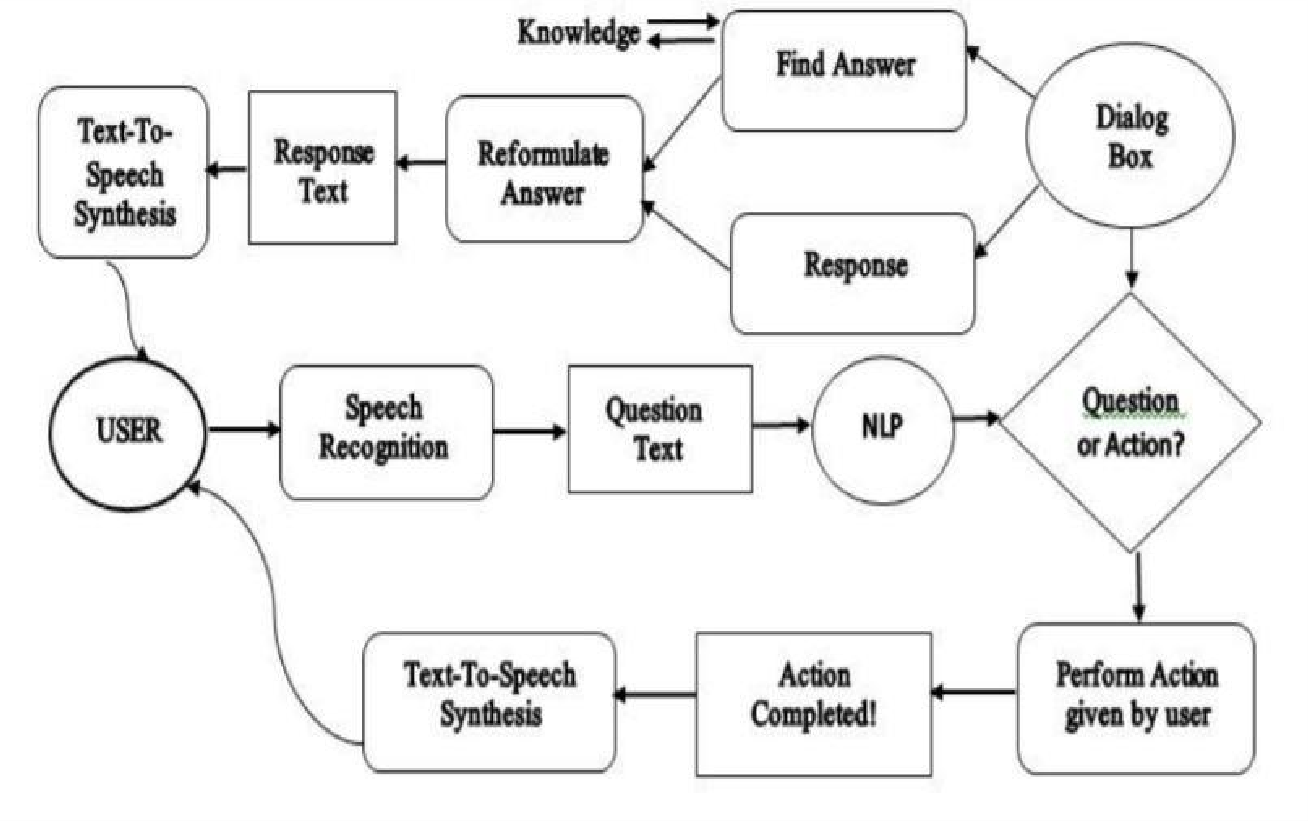
It has main 3 factors:

**Actors :** A use case diagram shows the interaction between the system and entities external to the system. These external entities are referred to as actors. Actors represent roles which may include human users, external hardware or other systems.

**Use Cases:** A use case is a single unit of meaningful work. It provides a high-level view of behavior observable to someone or something outside the system. The notation for a use case is an ellipse.

**System Boundary:** A system boundary **is**a rectangle that you can draw in a use-case diagram to separate the use cases that are internal to a system from the actors that are external to the system**.**

## PROJECT DESIGN:

Data modeling:

**Use Case Diagram**

**Experiment-5**

**Aim: Make Class Diagram and Object Diagram but differentiate which is associated with class and which is not.**

**Class diagrams:**

Class diagrams are the main building blocks of every object-oriented method. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams.

The main purpose to use class diagrams are:

* This is the only UML that can appropriately depict various aspects of the OOPs concept.
* Proper design and analysis of applications can be faster and efficient.
* It is the base for deployment and component diagram.

**Object Diagram:**

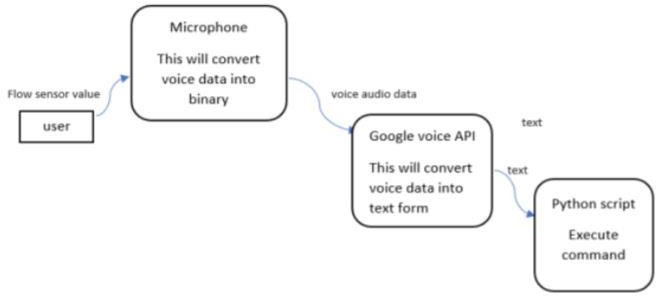
An ObjectDiagram can be referred to as a screenshot of the instances in a system and the relationship that exists between them. Since object diagrams depict behaviour when objects have been instantiated, we are able to study the behavior of the system at a particular instant. Object diagrams are vital to portray and understand functional requirements of the system.

In other words, “An object diagram in the Unified Modeling Language (UML), is a diagram that shows a completeorpartialview of the structure of a modeled system ataspecifictime**.**”

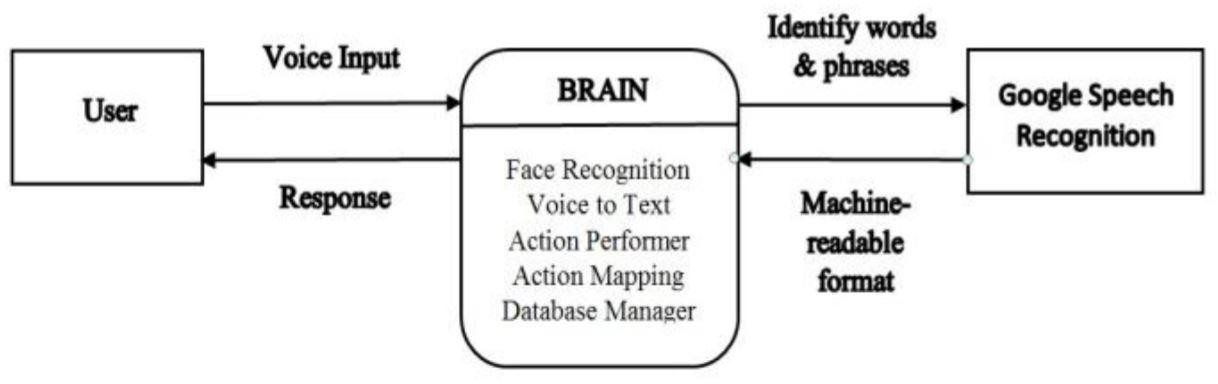
**Difference between an Object and a Class Diagram –**

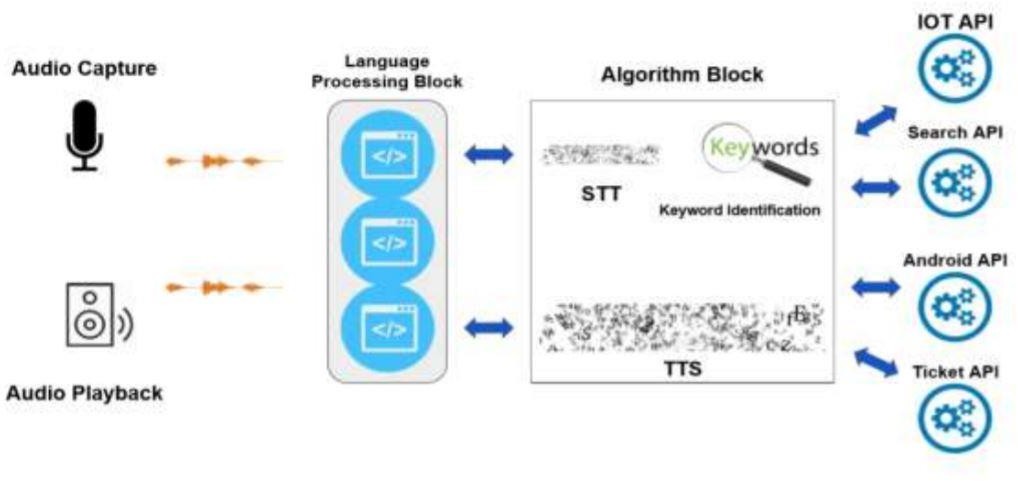
An object diagram is similar to a class diagram except it shows the instances of classes in the system. We depict actual classifiers and their relationships making the use of class diagrams. On the other hand, an Object Diagram represents specific instances of classes and relationships between them at a point of time.

**Data Flow Diagram (DFD):**

****

**CLASS DIAGRAM**

****

****

**Object Diagram**

**Experiment-6**

**Aim : Relationship between Use Case Diagram and Sequence Diagram. What actors are using here.**

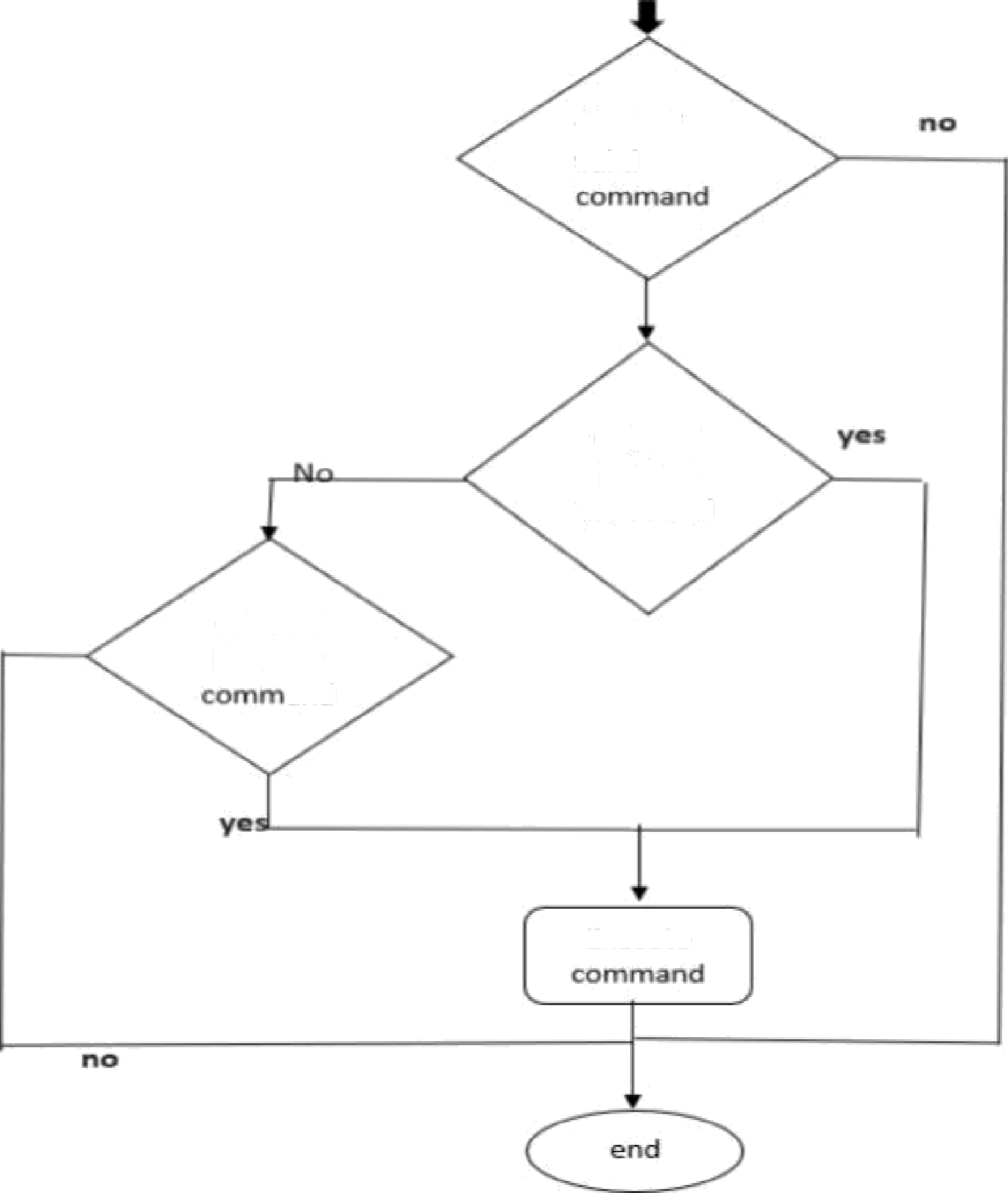
**Sequence Diagrams:**

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

**Relationship between Use Case Diagram and Sequence Diagram**

A use-case model is built and the actors are connected to use cases. Each use case represents a task in which the actor participates. For each use case, a sequence diagram is built. Each sequence diagram specifies the main interaction steps to be achieved for each task (i.e. use case).

Actors that are using in this use case and sequence diagram are



**Future of Work:**

In the future, we can also expect to see VA’s that is able to process the user commands locally on- device, rather than the data being transmitted to a datacenter. This field of artificial intelligence is called Edge AI, or simply edge computing. This is the type of technology that Tesla and other car vendors use to make their cars able to be aware of its surroundings and to make decisions based on them. Self-driving cars need to be able to make decisions quickly and cannot rely on cloud services that might lag. Therefore, the vehicles must have huge data processing abilities on the device itself. In many ways, it is the heart of future autonomous vehicles.

Further, VA’s with on-device processing abilities can be trained to detect other noises as well, such as a person falling in their home. This way, the device can assist you in other ways than just by voice commands. For instance, it can call for healthcare personnel if the person is not responding after falling, or it can function as a house alarm if it detects that a window gets broken.

Today we might view virtual assistants as simple and immature applications, but I am confident that they will be a crucial part of our future lives. The user experience will become much better, making

Interactions richer and more natural. Their ability to process information on-device will open a whole new range of opportunities. All in all, virtual assistants will become more complex ecosystems that can support you in multiple areas of your everyday lives.

**Reference:**

1. https://en.wikipedia.org/wiki/Main\_Page
2. https://www.python.org/
3. https://pypi.org/project/SpeechRecognition/
4. https://pypi.org/project/pyttsx3/
5. https://en.wikipedia.org/wiki/Virtual\_assistant
6. https://www.definedcrowd.com/success-stories/building-a-voice-assistant-model/
7. https://www.tietoevry.com/en/blog/2020/04/will-virtual-assistants-become-a-vital-part-of-the-future/