AProject/Dissertation Review-
ETE Report on

AI BASED PERSONAL DIGITAL ASSISTANT

Submitted in partial fulfillment of

the requirement for the award of

thedegree of

B.Tech In Computer
Science and Engineering

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervisionof

Mr.Sandeep Bhatia

Professor

Submitted

BY-

ABHISHEK ARYAN

19SCSE1010264

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, GALGOTIAS UNIVERSITY, GREATER NOIDA INDIA MONTH, YEAR

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination Abhishek Aryan has been held on __________________________________and his/her work is recommended for the award of Btech in Computer Science and Engineering.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:
Place: Greater Noida

Acknowledgement

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **“AI BASED PERSONAL DIGITAL ASSISTANT”** in partial fulfillment of the requirements for the award ofthe Btech in Computer Science and Engineering submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the periodof month, Year to Month and Year, under the supervision of Name… Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

ABHISHEK ARYAN(19SCSE1010264)

This is to certify that the above statement made by the candidates is correct to the best of myknowledge.

Supervisor Name

Mr.SandeepBhatia

Designation Asst. Professor

ABSTRACT

The project aims to develop a personal-assistant for windows system. Sophia drawn its inspiration from virtual assistant like Cortana for Windows and Google Assistant from Android, and Siri for iOS. Users can interact with the assistant either through voice commands.

As a personal assistant, Sophia assists the end-user with day-to-day activities like general human conversation, searching queries on Google, Bing or yahoo, searching for videos, retrieving images from NASA, live weather conditions, word meanings, searching for medicine details, reminding the user about the scheduled events and tasks. The user commands are analysed with the help of machine learning to give an optimal solution.

Keywords: Personal Assistant, Windows System, Automation, AI.

TABLE OF CONTENTS

INTRODUCTION

This project is based on windows system and provide personal assistant using voice recognition. This program includes the functions and services of: set alarm, music player service, checking weather, Google searching engine, Wikipedia searching engine.

This project is originated from a popular application from Windows called “Cortana”. This is very interesting, easy going and convenient, with wide real world usage and large developing potential. For instance, the voice assistance is very useful for personal assistants, direction guides or driving, helps among the disabled community, and so on.

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.

Cortana is a virtual assistant developed by Microsoft which uses the Bing search engine to perform tasks such as setting reminders and answering questions for the user. Cortana is currently available in English, Portuguese, French, German, Italian, Spanish, Chinese, and Japanese language editions, depending on the software platform and region in which it is used.

BACKGROUND

There already exist a number of desktop virtual assistants. A few examples of current virtual assistants available in market are discussed in this section along with the tasks they can provide and their drawbacks.

SIRI from Apple :

SIRI is personal assistant software that interfaces with the user thru voice interface, recognizes commands and acts on them. It learns to adapt to user's speech and thus improves voice recognition over time. It also tries to converse with the user when it does not identify the user request. It integrates with calendar, contacts and music library applications on the device and also integrates with GPS and camera on the device. It uses location, temporal, social and task based contexts, to personalize the agent behavior specifically to the user at a given point of time.

Supported Tasks

- Call someone from my contacts list • Launch an application on my iPhone.
- Send a text message to someone.
- Set up a meeting on my calendar for 9am tomorrow.
- Set an alarm for 5am tomorrow morning.
- Play a specific song in my iTunes library.
- Enter a new note

Drawback SIRI does not maintain a knowledge database of its own and its understanding comes from the information captured in domain models and data models.

AIM AND PURPOSE

According to the overall description in the context, the purpose of the project is to develop an windows application that provides an intelligent voice assistant with the functionalities as alarm, set reminder, music play, checking weather, searching engine (Google, Wikipedia), Nasa news. This project is focusing on the windows development over the voice control (recognition, generate and analyze corresponding commands, intelligent responses automatically), Google products and relevant APIs (Google weather, Google search and etc), Wikipedia API and Nasa API, Wolframalpha(For science) API. Speech-To-Text, Text-To-Speech technology. As all those functionalities and services for the project have been explained, the main structure and construction of the project has been basically illustrated with its goals.

PRODUCT GOALS AND OBJECTIVES

Currently, the project aims to provide the windows Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, nasa space news, and many others but also help in automation of various activities.

One of the main advantages of voice searches is their rapidity. In fact, voice is reputed to be four times faster than a written search: whereas we can write about 40 words per minute, we are capable of speaking around 150 during the same period of time15. In this respect, the ability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

PURPOSE, SCOPE

PURPOSE

Purpose of virtual assistant is to being capable of voice interaction, music playback, making to-do lists, setting alarms, playing song on local computer and providing weather, sports, and other real-time information, such as news. Virtual assistants enable users to speak natural language voice commands in order to operate the device.

There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized, it's clear that we are moving towards less screen interaction.

SCOPE

Presently, Sophia is being developed as an automation tool and virtual assistant. Among the Various roles played by Sophia are: 1. Search Engine with voice interactions 2. Space news from NASA from any date. 3. Reminder and To-Do application. 4. Weather Forecasting Application.

SYSTEM ANALYSIS

HARDWARE AND SOFTWARE REQUIREMENTS

The software is designed to be light-weighted so that it doesn't be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software requirement for virtual assistant.

Hardware:

- Pentium-pro processor or later.
- RAM 512MB or more.

Software:

- Windows 7(32-bit) or above.
- Python 2.7 or later.
- Chrome Driver.
- Any Text Editor.

SURVEY OF TECHNOLOGY

Python

Python is 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 Sophia, libraries used are speechrecognition to recognize voice, Pyttsx 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.

Pyttsx

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for textto-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.

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.

Wolfram|Alpha

The Wolfram|Alpha Full Results API provides a **web-based API** allowing the computational and presentation capabilities of Wolfram|Alpha to be integrated into web, mobile, desktop and enterprise applications. The API allows clients to submit free-form queries similar to the queries one might enter at the Wolfram|Alpha website, and for the computed results to be returned in a variety of formats. It is implemented in a standard REST protocol using HTTP GET requests.

SYSTEM DESIGN

Module Division:

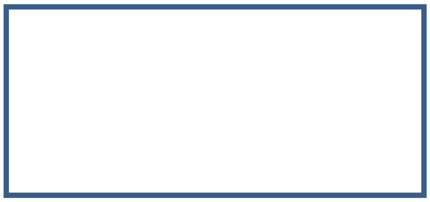
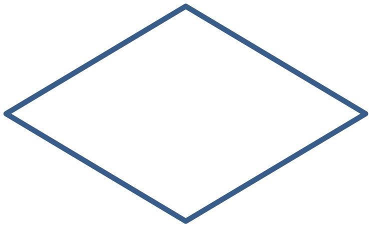
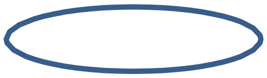
1. **User Interaction:**
   In the user interaction user can communicate with assistant via voice and give the command to perform the task.

2. **Virtual Assistant:**
   When the user give the command to assistant then task is automate by assistant. Example like: command is "Today temperature outside." Then the assistant fetch the data from the internet and return the data like "Temperature outside is 30°C"

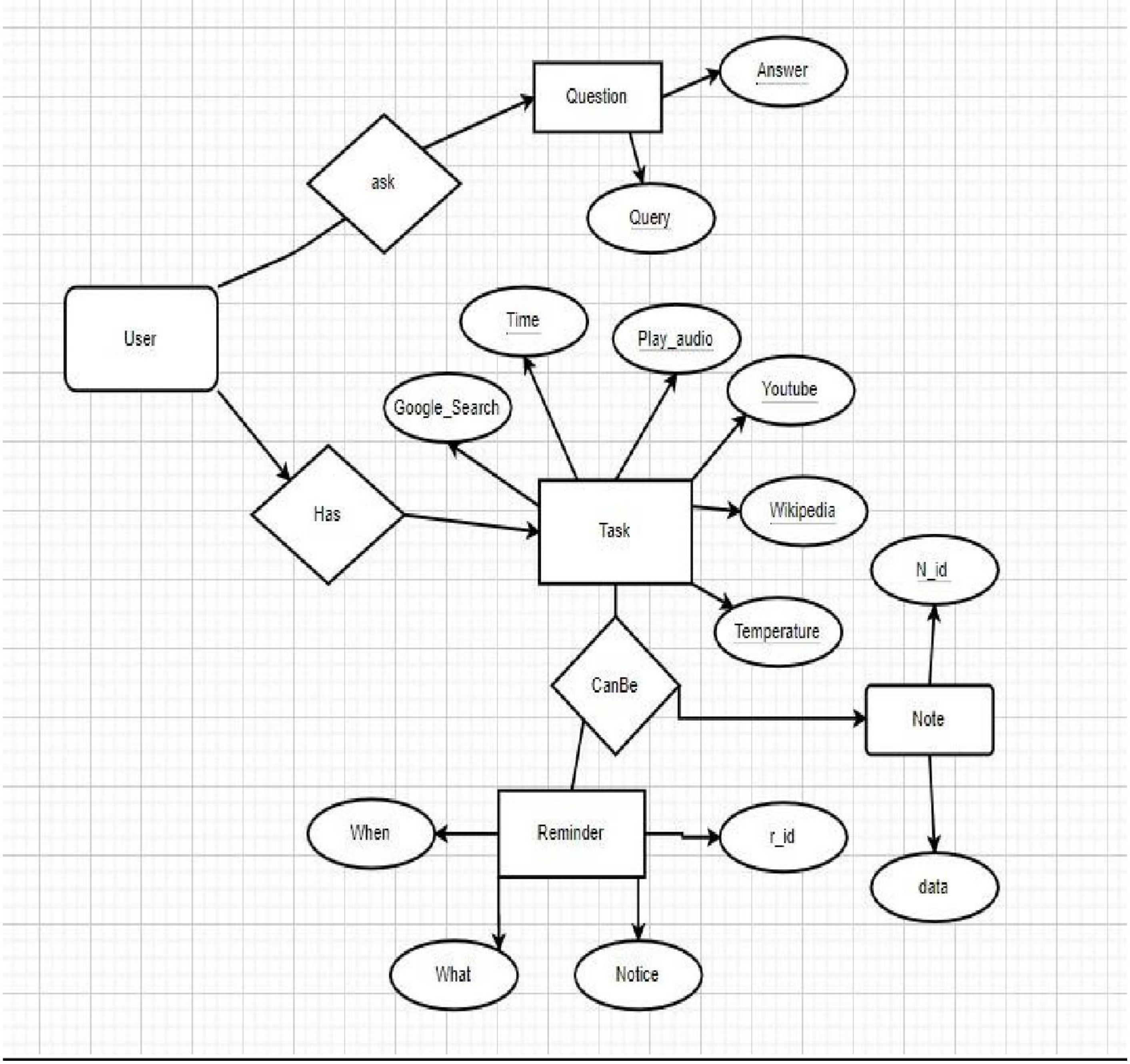
E-R Diagram

An entity-relationship (ER) diagram is a specialized graphic that illustrates the ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes .The utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user. In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

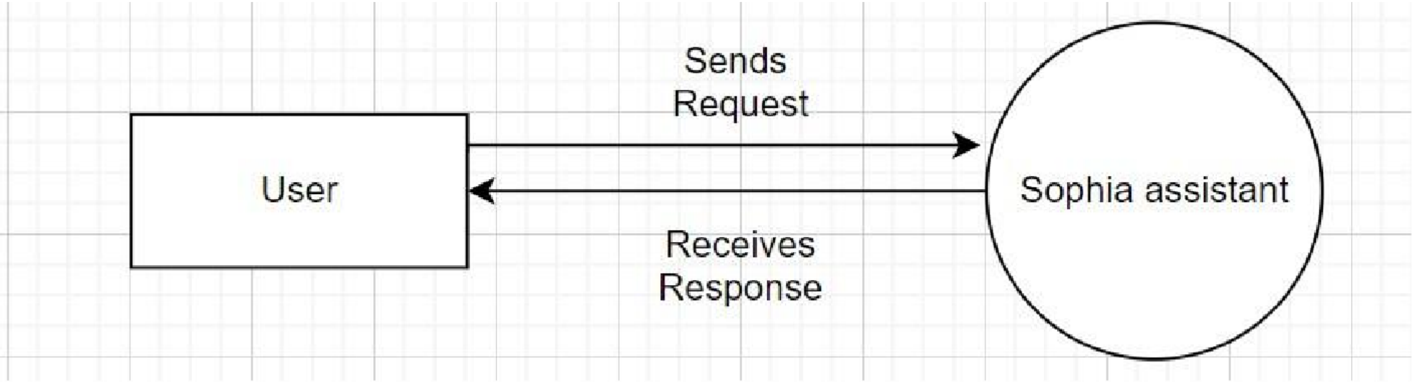
Name	Notation	Description
Entity		Entity is represented by a box within the ERD. Entities are abstract concepts, each representing one or more instances of the concept in question. Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity
Relationship		Relationships are represented by Diamonds. A relationship is a named collection or association between entities or used to relate to two or more entities with some common attributes or meaningful interaction between the objects.
Attributes		Attributes are represented by Oval. An attribute is a single data item related to a database object.

		The database schema associates one or more attributes with each database entity

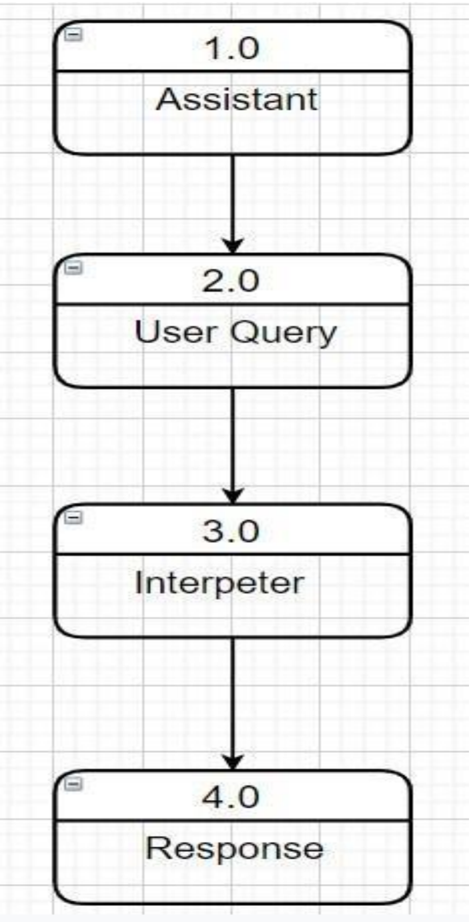


Data Flow Diagram/UML

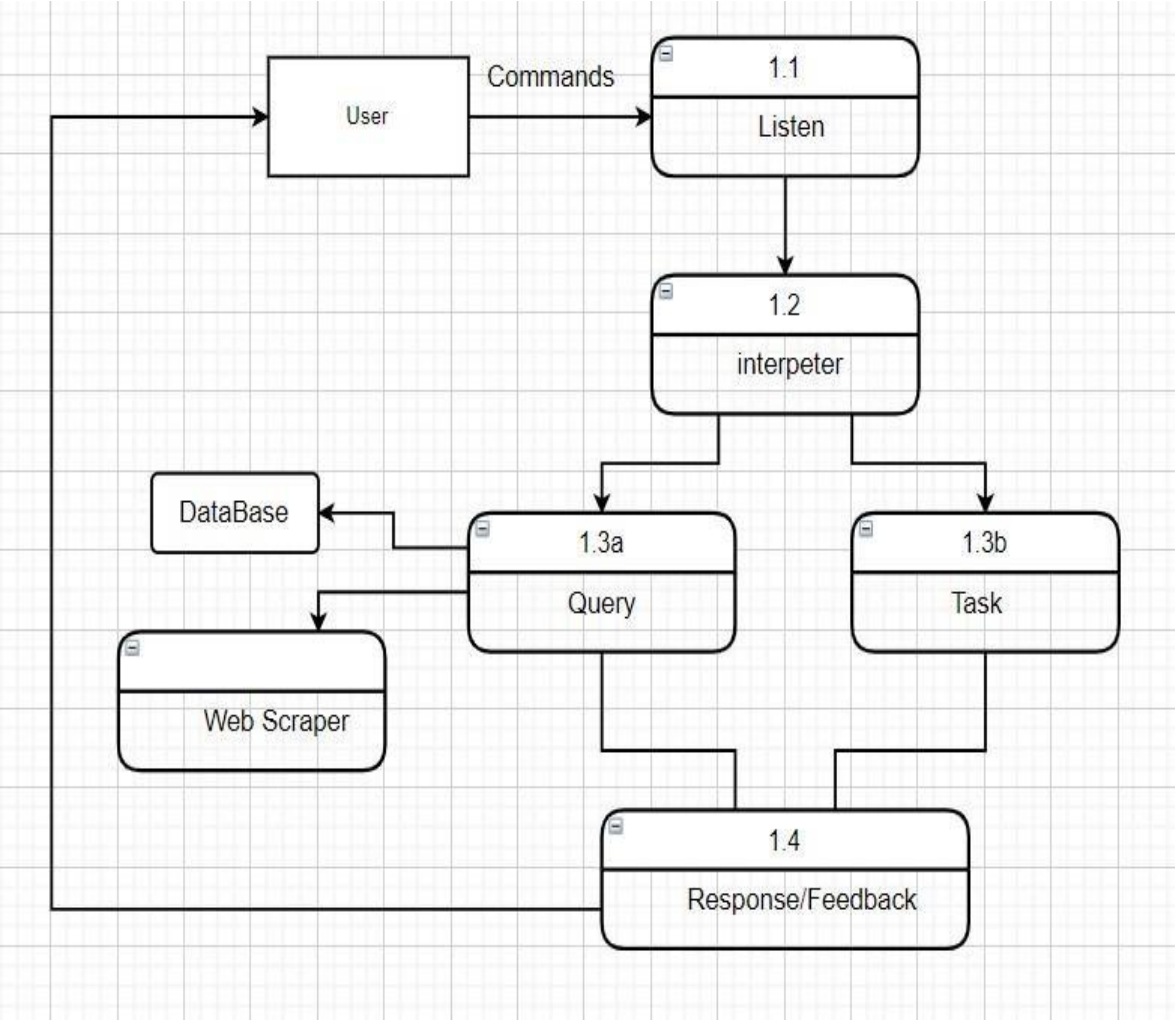
The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a Flowchart. **DFD 0 level**



DFD Level 1

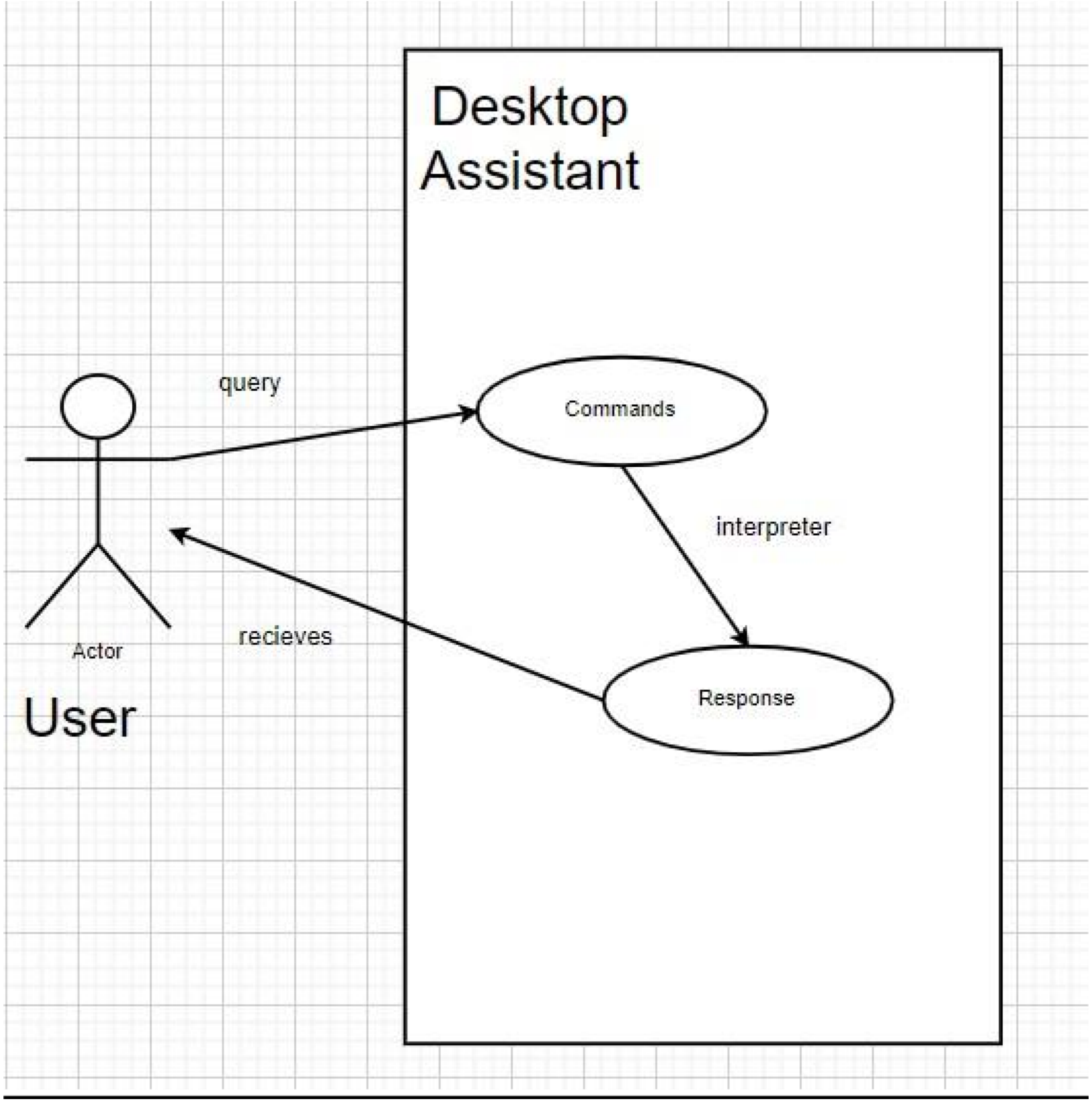


DFD Level 2


User
Commands
1.1
Listen
1.2
interpeter
DataBase
1.3a
Query
1.3b
Task
Web Scraper
1.4
Response/Feedback

UML DIAGRAM

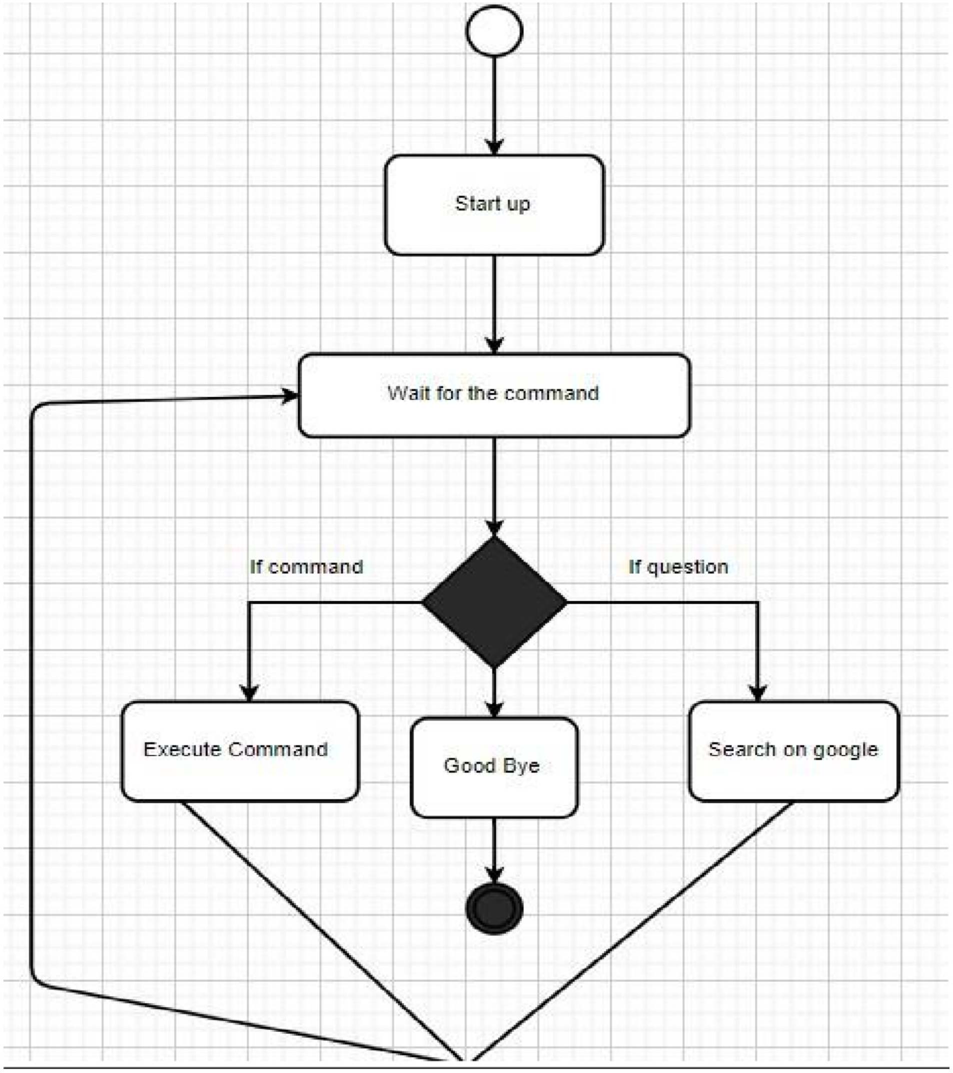
A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality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



ACTIVITY DIAGRAM

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

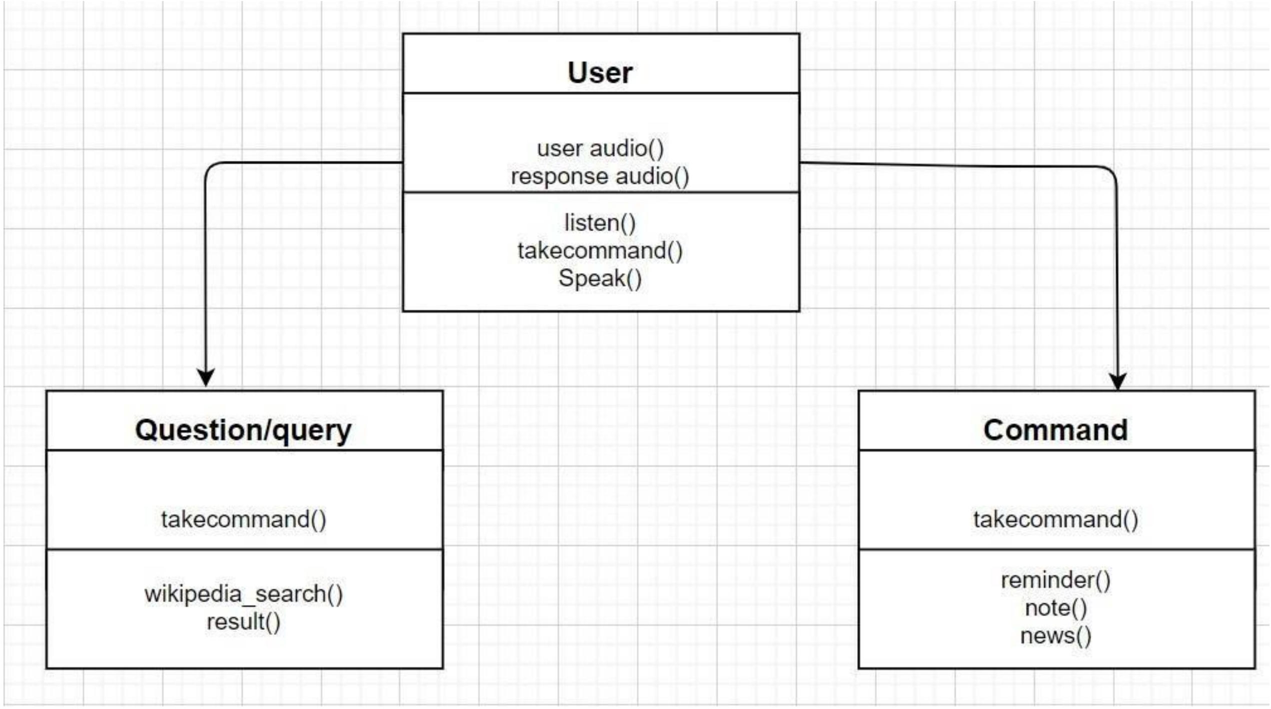
Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.



CLASS DIAGRAM

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.



SOURCE CODE

```
import pyttsx3 #pip install pyttsx3
import speech_recognition as sr #pip install speechRecognition
import datetime import time
import wikipedia #pip install wikipedia
import webbrowser import os import
random import pywhatkit import pyjokes
from pytube import YouTube import
keyboard import pyautogui import
requests from PIL import Image import
requests
from bs4 import BeautifulSoup
import wolframalpha import
whatsapp


assistant = pyttsx3.init('sapi5') voices
= assistant.getProperty('voices')
# print(voices[1].id)
assistant.setProperty('voice', voices[1].id)


def speak(audio):
    """[summary]
                The audio is that which you are speak take as
string[str] and then speak
    Args:         audio
([str]]): [none]
    """         print(f":
{audio}")     print("
")
assistant.say(audio)
assistant.runAndWait() def
takeCommand():
command = sr.Recognizer()
   # with sr.Microphone(device_index=1) as source:
with sr.Microphone() as source:
print("Listening...")
```

```
command.pause_threshold = 1          audio = 
command.listen(source)      try:            
print("Recognizing...")
        query = command.recognize_google(audio, language='en-in')
print(f"User said: {query}\n")
     except Exception as e:
print("Say that again please...")
return "None"     return query


def startup():
    """[startup] this function will start when the sophia is run
every time."""
    speak("Initializing Sophia")
    speak("Starting all systems applications")
speak("Installing and checking all drivers")
speak("Checking the internet connection")
speak("All systems have been activated")
hour = int(datetime.datetime.now().hour)     if
hour>=0 and hour<=12:         speak("Good
Morning")     elif hour>12 and hour<18:
speak("Good afternoon")     else:
speak("Good evening")
    strTime = datetime.datetime.now().strftime('%H:%M:%S')
speak(f"Currently it is {strTime}")
    speak("I am sophia. Online and ready sir. Please tell me how may
I help you")
 def screenshot():     speak("ok done, tell me
the name of file.")     f_name= takeCommand()
file_name= f_name+".png"
    path = 
"C:\\Users\\wozzy\\OneDrive\\Desktop\\Project\\miniProjectm
ain\\miniProject-main\\Database\\screenshot\\"+file_name
pyauto = pyautogui.screenshot()     pyauto.save(path)
    os.startfile("C:\\Users\\wozzy\\OneDrive\\Desktop\\Project\\mini
Project-main\\miniProject-main\\Database\\screenshot\\"+file_name)
speak("Here your screenshot")
 def nasa_news(Date):
speak("Extracting data from nasa")
print("Extracting data from nasa")
```

```
    api_key = "ZMGdewvFuBq4SQfZJzCuNtrzqrnaPWkabDahrbpZ"
url =
"https://api.nasa.gov/planetary/apod?api_key="+str(api_key)
Params = {'date':str(Date)}     r =
requests.get(url,params=Params)
    Data =r.json()
    Info = Data['explanation']
Title = Data['title']
print(Info)     print(Title)
    Img_url = Data['url']     Img_r
= requests.get(Img_url)
file_name = str(Date)+'.jpg'
with open(file_name,'wb') as f:
        f.write(Img_r.content)
path1 =
"C:\\Users\\wozzy\\OneDrive\\Desktop\\Project\\miniProjectmain\\
miniProject-main\\Database\\Photos\\"+str(file_name)     path2 =
"
C:\\Users\\wozzy\\OneDrive\\Desktop\\Project\\miniProjectm
ain\\miniProject-main\\Database\\Photos\\"+str(file_name)
os.rename(path1, path2)     img = Image.open(path2)
img.show()
    speak(f"Title is:{Title}")
speak(f"according to nasa:{Info}")
 def wolfram(query):     api_key ="K93RPX-
WWEK9JLAY8"     requester =
wolframalpha.Client(api_key) requested =
requester.query(query) try:
        Answer = next(requested.results).text
return Answer except:         print("An string
value is not answerable.")         speak("An string
value is not answerable.")
 def
calculator(query):
term = str(query)
    term = term.replace('calculate ','')     term =
term.replace('multiply ','*')     term =
term.replace('plus ','+')     term =
term.replace('minus ','-')     term =
term.replace('divide ','/')      term =
term.replace('into ','*')      term =
term.replace('sophia ','')      final = str(term)
try:             result = wolfram(final)
print(result)          speak(f"Answer is: {result}")
except:            print("An string value is not
answerable.")          speak("An string value is not
answerable.")
 def Temp(query):      tmp = str(query)
tmp = tmp.replace("sophia ","")      tmp =
tmp.replace("in ","")      tmp =
tmp.replace("what is the ","")      tmp =
tmp.replace("temperature ","")
tmp_query = str(tmp)      if 'outside' in
tmp_query:          var1="temperature in
Noida"          answer=wolfram(var1)
speak(f"{var1} is {answer}")      else:
var2="temperature in " + tmp_query
ans=wolfram(var2) speak(f"{var2} is
{ans}")
if___name___== "__main__":
    # startup()      while True:
try:              query =
takeCommand().lower()
            # Logic for executing tasks based on query
if "hello" in query:
                speak("hello sir how can i help.")
                         elif "how are you" in query:
speak("I am fine sir What about you")
                         elif
"fine" in query:
speak("ok, sir")
                         elif "bye" in query:
speak("bye sir, you can call me any time")
break
             elif "time" in
query:                      strTime =
datetime.datetime.now().strftime('%H:%M:%S')
speak(f"Currently it is {strTime}")
```

```
                         elif 'wikipedia' in query:
try:                     speak('Searching On Wikipedia...')
query = query.replace("wikipedia", "")                        results
= wikipedia.summary(query, sentences=2)
speak("According to Wikipedia")                        print(results)
speak("Can I read This.")
read_or_not=takeCommand()                         if 'yes' in
read_or_not:
                        speak(results)
if 'no' in read_or_not:
pass                     except Exception as e:
speak("Can not find result")

            elif 'open amazon' in query or 'shop online' in query:
            webbrowser.open("https://www.amazon.com")
            print("opening amazon")
                speak("opening amazon")
                         elif 'open flipkart' in query:
webbrowser.open("https://www.flipkart.com")
                print("opening flipkart")
speak("opening flipkart")
                         elif 'open ebay' in query:
webbrowser.open("https://www.ebay.com")
print("opening ebay")                 speak("opening
ebay")
                         elif "shutdown"
in query:
speak("shutting down")
os.system('shutdown -s')

            elif 'make you' in query or 'created you' in query or
'develop you' in query:                 ans_m = " For your
information Akash, Sachin and Vishwajeet Develop me!."
print(ans_m)                 speak(ans_m)
             elif "who are you" in query or "about you" in query
or
"your details" in query:                 about = "I am Sophia. I am
computer based program but i can help you lot like a your close
friend. Simple try me to give simple command ! I also play video and
song from web or online. ok Lets Start "
```

```
                 print(about)
speak(about)
              elif "your name" in query or "sweat name" in query:
    na_me = "Thanks for Asking my name. well my self !
Sophia." print(na_me)
                 speak(na_me)
             elif query == 'none':
             continue

             elif 'exit' in query or 'abort' in query or 'stop' in
query or 'bye' in query or 'quit' in query :
ex_exit = 'I feeling very sweet after meeting with you but you are
going! i am very sad'                    speak(ex_exit)
exit()
                           elif 'open
code' in query:
code_path =
"C:\\Users\\Akash\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe"
                 print("Openning VSCODE")
speak("Openning VSCODE")
os.startfile(code_path)
                           elif 'play
song' in query:
                 music_folder = "E:\\audios"
song = os.listdir(music_folder)
all_song=len(song)                 import
random
                 rand_song    =    random.randint(0,    all_song-1)
print(f"Now playing {rand_song}, enjoy beautiful song")
                 os.startfile(os.path.join(music_folder,song[rand_son
g]))             elif "search on youtube" in query:               link
= takeCommand()
                 link = query.replace("search",'')
link = query.replace("on","")                   link
= query.replace("youtube","")                   web
=
f"https://www.youtube.com/results?search_query={link}"
                 speak("Searching On YouTube")
             pywhatkit.playonyt(web)
```

```
            elif "how to" in query:
                from pywikihow import search_wikihow
                import webbrowser as web
                speak("Collecting data from the internet")
            link = query.replace('sophia','')     link =
            query.replace('how to','')     link =
            query.replace('what is','')
                link = query.replace('what do you mean by','')
max_result = 1
                how_to = search_wikihow(link,max_result)
assert len(how_to) == 1
how_to[0].print()                 speak("Can I read
this?")                 read_or_not=takeCommand()
if 'yes' in read_or_not:
                    speak(how_to[0].summary)
if 'no' in read_or_not:
pass
             elif 'remember that' in
query:                 speak("What to
remember")                 rememberMsg
=takeCommand()
                # speak("You tell me to remind you that" +
rememberMsg)                 remeber =
open("C:\\Users\\wozzy\\OneDrive\\Desktop\\Project\\miniProject-
main\\miniProject-main\\Database\\data.txt",'w')
remeber.write(rememberMsg)
remeber.close()                 speak("Done")
             elif 'what you remember' in
query:                 remeber =
open("C:\\Users\\wozzy\\OneDrive\\Desktop\\Project\\miniProjectmai
n\\miniProject-main\\Database\\data.txt")
                print(f"You tell me to remember is that
{remeber.read()}")
                speak(f"You tell me to remember is that
{remeber.read()}")

            elif 'temperature' in query:
                Temp(query)
                # speak("Tell me the place name")
                # place_name =takeCommand()
                # search=f"Temperature in {place_name}"
                # url = f"https://www.google.com/search?q={search}"
                # r = requests.get(url)
                # data = BeautifulSoup(r.text,"html.parser")
                # temperature =
data.find("div",class_="BNeawe").text
                # speak(f"Temperature is {temperature}")
                # print(f"Temperature is {temperature}")
# speak("You want to konw another city temperature?")
                # next = takeCommand()
                # if 'yes' in next:
                #         speak("Tell me the place name")
                #         place_name =takeCommand()
                #         search=f"Temperature in {place_name}"
                #         url =
f"https://www.google.com/search?q={search}"
                #         import requests
                #         from bs4 import BeautifulSoup
                #         r = requests.get(url)
                #         data = BeautifulSoup(r.text,"html.parser")
                #         temperature =
data.find("div",class_="BNeawe").text
                #         speak(f"Temperature is {temperature}")
                #         print(f"Temperature is {temperature}")
                # else:
                #   speak("No   problem!")
             elif 'calculate' in
query:
                calculator(query)
             elif 'screenshot' in
query:                 screenshot()
                    elif 'nasa'
in query:
                speak("Tell me the complete date like year, month,
date ")
                speak("tell me the year")     year =
            takeCommand() speak("Tell me the month like 1,2
            like that") month = takeCommand() speak("tell me
            the date")
                date = takeCommand()
                final_date =f"{year}-{month}-{date}"
                nasa_news(final_date)

            elif 'joke' in query:
                list_of_jokes = pyjokes.get_jokes(language="en",
category="all")                 for i in range(0, 4):
                    speak(list_of_jokes[i])
             elif 'what is' in
query:
                # ser=str(query)
                # ser=ser.replace('what is ','')
                # print(ser)
result = wolfram(query)
speak(result)
                         elif 'whatsapp message' in query:
query = query.replace("sophia","")                      query =
query.replace("send","")                     query =
query.replace("whatsapp message","")                      query =
query.replace("to","")                     name = query
                     if 'Akash Niet' in
name:                         numb =
"7909000586"
                        speak(f"What's The Message For {name}")
mess = takeCommand()
whatsapp.whatsapp(numb,mess)
                     elif 'Sachin MCA' in
name:                         numb =
"9939058215"
                        speak(f"What's The Message For {name}")
mess = takeCommand()
whatsapp.whatsapp(numb,mess)
                                         elif
'Vishwajeet' in name:
                        numb = "7488608522"
            speak(f"What's The Message For {name}")
            mess = takeCommand()
            whatsapp.whatsapp(numb,mess)
                    elif 'project group' in  name:
                        gro = "GRNrmfIs9dYEs4RRybvjUw"
                speak(f"What's The Message For {name}")
                        mess = takeCommand()
                        whatsapp.Whatspp_Grp(gro,mess)


        except Exception as e:
print(e)
            speak("I cannot recognize ")
```

WHATSAPP CODE

```
import webbrowser as web
import time import
keyboard import
pyautogui
 def
whatsapp(number,message):
    numb = '+91' + number
    open_chat = "https://web.whatsapp.com/send?photo=" + numb +
"&text=" + message
web.open(open_chat)     time.sleep(15)
keyboard.press('enter')
 def Whatspp_Grp(group_id ,
message):
    open_chat = "https://web.whatsapp.com/accept?code=" + group_id
web.open(open_chat)     time.sleep(15)
    pyautogui.click(x=754 , y =696)
keyboard.write(message)
time.sleep(1)
keyboard.press('enter')
```

SYSTEM TESTING

UNIT TESTING

The unit testing of the source code has to be done for every individual unit of module that was developing part of the system and some errors were found for every turn and rectified. This form of testing was use to check for the behavior signified the working of the system in different environment as an independent functional unit. In the software development process Unit Tests basically test individual parts (also called as Unit) of code (mostly methods) and make it work as expected by programmer. A Unit Test is a code written by any programmer which test small pieces of functionality of big programs. Performing unit tests is always designed to be simple, A "UNIT" in this sense is the smallest component of the large code part that makes sense to test, mainly a method out of many methods of some class.

Why do we need Unit test?

One of the most valuable benefits of using Unit Tests for your development is that it may give you positive confidence that your code will work as you have expected it to work in your development process. Unit Tests always give you the certainty that it will lead to a long term development phase because with the help of unit tests you can easily know that your foundation code block is totally dependable on it.

INTEGRATION TESTING

From the individual parts to the cohesion of each part to make the system as a whole, there is need to test the working between the assembled modules of the system. The modules are integrated to makes up the entire system. The testing process is concerned with finding errors that result from unanticipated interaction between the sub-system and system component. It is also concerned with validating the system meets its functional and nonfunctional requirement. Integration tests ensure that an app's components function correctly at a level that includes the app's supporting infrastructure, such as the database, file system, and network. PYTHON supports integration tests using a unit test framework with a test web host and an in-memory test server.

Operation and Maintenance

Operation:

Checking weather: the user can use the application to check the weather for recent days in local place or specific location. He or she should say the keyword "what is weather today", then the user should get the message be presented as "today/tomorrow/the day after tomorrow" if he or she wants to get the information about the other days otherwise the application will default set the date as today, and the user can also can choose to tell about the place name "in Noida", the application will check the weather belong to that place, otherwise the place will be set as locally.

Text Message Transfer: If the user wants to use the application to send the text message WhatsApp, he or she must have a command or keyword "Send Message on WhatssApp" and a valid name, then the message will be send if the person is found in the contacts. They are different forms to send the message; the list below shows the correct command can do the message sending. "Send a message to Akash", send a message to Sachin. Then assistant ask what message want to send then say message "Let's complete project", then the program auto login whatssapp web on web in computer then first phone number corresponding to "Akash" and send the message to Akash. "Message , join the meeting", send a message to Sachin with the content "Let's complete project".

Weather check today: "What's the weather for today", the current weather condition for local place will be show.
"What's the weather in Noida" The current weather condition for Noida will be show.
Weather check other days: "What's the weather next few days", the forecast in next 4 days will be show.

Wikipedia : searching engine, whenever the user wants to search any content in Wikipedia, it is possible to do in this program by having a command contain the keyword **'Wikipedia'**. If 'Wikipedia' is detected by the program, the program will automatically give the result by search the content after 'Wikipedia' in Wikipedia. For Example:

"Android Wikipedia", the keyword 'Wikipedia' is detected, and the program will return the result by searching 'Android' on Wikipedia.

"Narendra Modi Wikipedia", the keyword 'Wikipedia' is detected, and the program will return the result by search the content after 'Wikipedia', which is 'Narendra Modi' on Wikipedia.

Music player service, the user can use this application to play songs, his or her command must contain keyword "play song" then song is play by randomly. For example:

"Play Song" – output is playing song Enjoy!

Nasa News: This is the key words where the user can ask the anything from the this nasa API like when he or she say "Nasa News" the assistant ask the full date like say date, month and year. Then is fetch the data and image.

And many more Command!

Maintenance

After the program is completed, the program still needs long term maintenance to make it available and stable to execute. The program will be test after a certain period of time and debug each of the function and possible bugs, whenever a potential bug is detected; the program needs to be refined to a better design. Meanwhile, there will update and add more data to the database to increase the database capacity. Depending on the new keywords, responses, relevant data found that could be applied in this program; the database will always be improved and can handle more and more cases.

Conclusions

As it has been previous stated, the program is mainly concerns with the techniques of windows application development, python programming, different APIs for Google products etc. The development is carried out as its primary planning which guide the work process of how to work with the program, how much time should the each of the developers spent in every week, the rescores needed for developing and how to handle the problems while it came up. The project was efficiently completed under the development model and the resources we found in early time were really useful when implementing the program.

Project experience & teamwork:

Apart from the program, we as the developers have improved a lot from the 0 level project. It is quite different from what we previously experienced in the working model, volume of tasks, and the problems we have encountered. In conclusion, we have been improved a lot from the project development, and gained development experience as well as programming skills; the most important is work as a team for a long term, challenge development.

Recommendations for Further Work

Design Improvements: No program has a perfect design without any user interface it is the same here in this program. Even though the program is completed with all the primary functions implemented and work properly, there are still many things that can be done with this program. For now it is console based project.

Additional Functions: There can be more functions which simplify our daily life and make it convenient to use. Functions as playing movies, checking stocks, exchange rate, downloading video from Youtube and uploading, automate Whatsapp, send messages on email etc.

Voice Recognition: Give the all the secret information about user. Then it recognize voice the say accordingly.

More Voice: Adding more voice option to communicate.

Translation: Translate into any language.

REFERENCE AND BIBLIOGRAPHY

Websites referred:

- www.stackoverflow.com
- www.pythonprogramming.net
- www.codecademy.com
- www.tutorialspoint.com □ www.google.co.in