

Mini Project report on

Personal Desktop Assistant

by

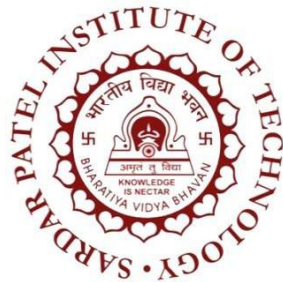
Simren Dubey [2019450014]

Siddharth Bhalerao [2019450005]

Under the guidance of

Dr. Neti Desai

Prof. Nikhita Mangaonkar



Department of Master of Computer Applications

Sardar Patel Institute of Technology

Autonomous Institute Affiliated to Mumbai University

2019-20

CERTIFICATE OF APPROVAL

This is to certify that the following students

Simren Dubey [2019450014]

Siddharth Bhalerao [2019450005]

Have satisfactorily carried out work on the project entitled

“Personal Desktop Assistant ”

Towards the fulfillment of mini project,
as laid down by Sardar Patel Institute of Technology
during the year 2019-20.

Project Guide 1
(Dr. Neti Desai)

Project Guide 2
(Prof. Nikhita Mangaonkar)

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Simren Dubey [2019450014]

Siddharth Bhalerao [2019450005]

Have successfully completed the Project report on “***Personal Desktop Assistant***”, which is found to be satisfactory and is approved

At

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI.

INTERNAL EXAMINER

EXTERNAL EXAMINER

Head of Department
(Dr. Pooja Raundale)

Principal
(Dr. Y. S. Rao)

Serial no.	Topic	Page no.
	Abstract.....	i
	Objectives.....	ii
	List of Figures.....	iii
	List of Tables.....	iv
1	Introduction.....	5
1.1	Problem Definition.....	5
1.2	Objective and Scope.....	5
1.3	System Requirements.....	6
2	SRS and Design	7
2.1	Introduction.....	7
2.2	Overall Description.....	7
2.3	System Features.....	7
2.4	External Interface Requirements	7
2.5	Nonfunctional Requirements	7
3	Project Analysis and Design.....	9
3.1	Methodologies Accepted.....	9
3.2	Architectural Design.....	10
3.3	Flow Diagram.....	11
4	Project Implementation and Testing.....	16
4.1	Code with reference to design.....	17
4.2	Snapshot of UI.....	27
4.3	Test Cases and Report.....	31

5	Documentation & Installation.....	32
6	System Maintenance	33
7	Future Enhancements	33
8	Limitations	34
9	Conclusion	34
10	Bibliography.....	35
11	Glossary of Terms.....	34

ABSTRACT:

A virtual assistant is a software agent that can perform tasks or services for an individual. Sometimes the term "chat bot" is used to refer to virtual assistants generally or specifically those accessed by online chat. This Project is a technical brief on Virtual Assistant technology and its opportunities and challenges in different areas. The project focuses on virtual assistant types and structural elements of a virtual assistant system. In this project we tried to implement virtual virtual Assistant features. We made this project based on the basic features of what an Assistant can do make thing easy for us.

OBJECTIVES:

- Provide assistant to user.
- Assistant will automatically perform the task given by user (in voice).
- Assistant will ease the work of user.
- It will save the time of user.

Fig.no	Figure name	Page no
3.1.1	Waterfall Model	6
3.2.1	Use Case Diagram	12
3.2.2	E-R Diagram	13
3.2.3	Flowchart	14
3.2.4	Gantt Chart	14
4.2.5	Pert Chart	20

Table.no	Table name	Page no
1.3.1	Hardware requirements	6
4.2.1	Snapshot of User App	28
4.3.1	Test Case – Log-in	31

1. INTRODUCTION

1.1 PROBLEM DEFINITION:

There are several vice assistant like Alexa, siri. Creating personal desktop assistant for everyone is our goal. This assistant will ease the work for user and save time. This assistant will do the work as commanded by the user.

1.2 OBJECTIVES AND SCOPE:

1.2.1 OBJECTIVES:

1. To make an interactive interface for the users.
2. To provide individual Personal Desktop Assistant for everyone.
3. To have different features in Desktop application for users.

1.2.2 SCOPE:

1. This assistant can be made advance in future.
2. It can be used to add various others features for Entertainment, study, Information, FAQ etc.
3. The possibility of designing is endless.
4. This assistant can be used by multiple user on daily bases.

1.3 SYSTEM REQUIREMENTS:

1.3.1 HARDWARE REQUIREMENTS:

Processor: Dual Core Processor and above.

RAM: 512MB and above.

Storage: Minimum Hard Disk Space.

1.3.2 SOFTWARE REQUIREMENTS:

Operating System: Windows OS

Software: python 3.7

2. PROJECT REQUIREMENT SPECIFICATION [SRS]

2.1 INTRODUCTION

This software works on voice command of the user. This is exclusively designed for the user to do ease their work and save time.

Keeping in mind about the daily requirements of the user and daily activity of the user this software is designed accordingly.

2.2 OVERALL DISCRPTION:

This assistant is name as Blue for helping people. This assistant will take the command from user in form of voice and then perform the task.

In today's the world the most important thing for anyone is time to save the time, what people need is that their small work to be done easily without any wastage of time. This assistant is designed keeping that in mind.

This project will not only save time but will also save the energy of a person to do some task and will give mental peace because their certain task is done by the assistant.

This assistant can do be accessed by any user as it only requires the voice command. Any one using the computer or laptop can access this and make this assistant do the task.

One of the major requirement for this assistant is the internet connectivity as this assistant is the real time assistant which works on currently happening.

Few modules like News, temperature, location are good example of how this assistant works on real time.

Features Included In this Assistant are:

1. Open a web page for example open Google, YouTube
2. Open application for example open vs Code or net-beans
3. Tells the current weather.
4. Tells the current date and time
5. Set an alarm

6. Tells you news for today
7. Tells you the current location
8. Opens any Wikipedia article.
9. Tells you features of Blue.
10. Tells you joke.
11. Opens any map location.
12. Gives you your internet speed.

2.3 SYSTEM FEATURES:

1. Information: This system provides all the real time information which user asks for without any problem which is accurate and on point.
2. Access: This assistant can be accessed by any one using the system there is no limitation to that and no restrictions for specific person with voice can access the system.
3. Design: This design is a basic user friendly design and will get the output in voice
4. Cost: This system is cost friendly and does not cost anything to the user.

2.4 FUNCTIONAL REQUIREMENT AND NON-FUNCTIONAL REQUIREMENTS:

FUNCTIONAL REQUIREMENT:

- i) Voice command
- ii) Speech to text
- iii) Assistant to help

NON-FUNCTIONAL REQUIREMENTS:

- iv) Usability.
- v) Efficiency
- vi) Performance
- vii) Reliability
- viii) Simplicity

3. PROJECT ANALYSIS AND DESIGN

3.1 METHODOLOGIES ACCEPTED:

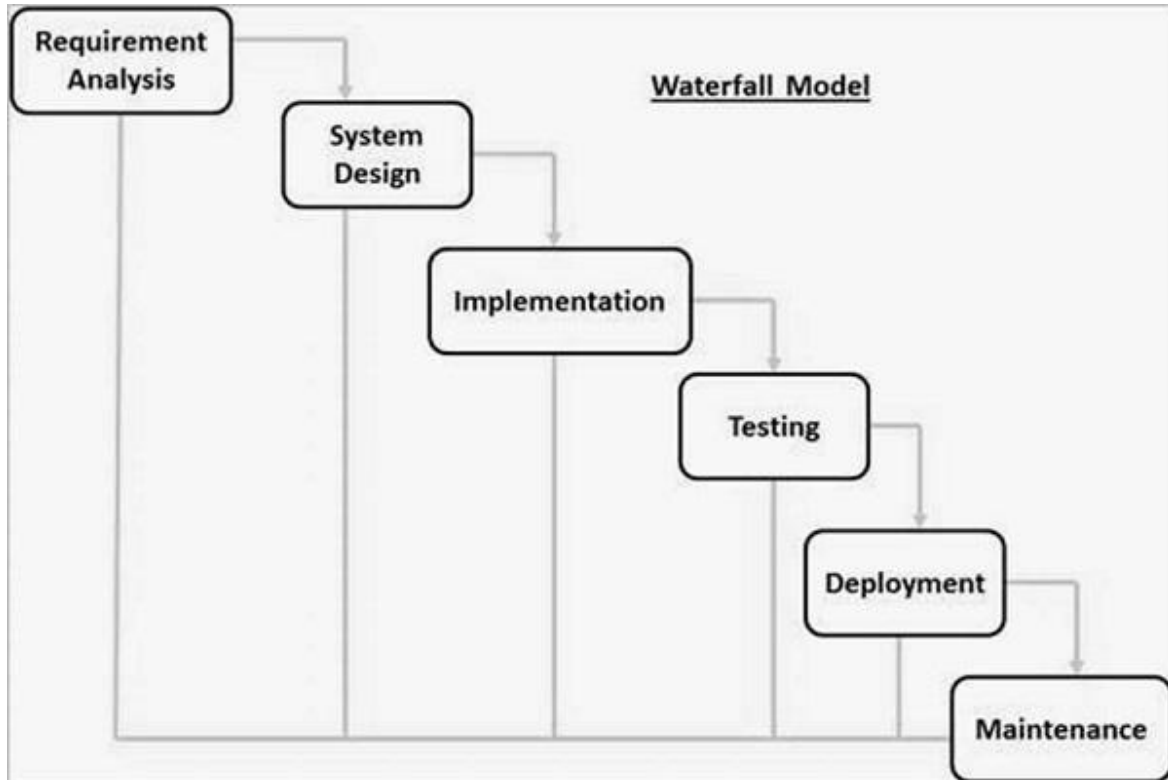
WATERFALL MODEL:

The Waterfall Model is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. In this project All the features have to be developed one after the another as the dependency depends on previous feature.

The Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap. Like wise if our one phase of execution of a feature is completed the next will be continued.

This model was considered as the best one because:

- 1) Requirements are very well documented, clear and fixed.
- 2) Product definition is stable..
- 4) There are no ambiguous requirements.
- 5) Ample resources with required expertise are available to support the product.
- 6) The project is short.

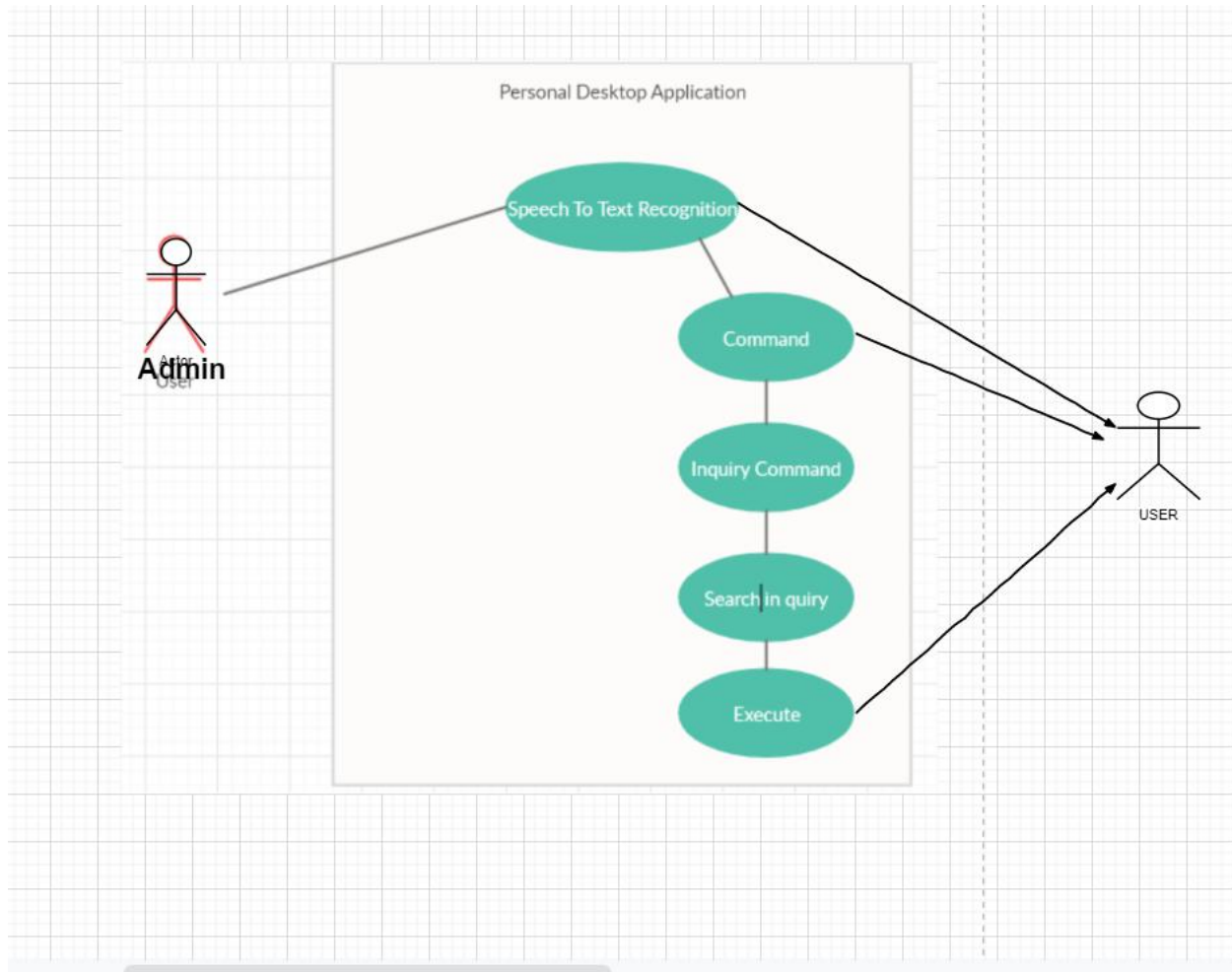


WATERFALL MODEL

3.2 DESIGN:

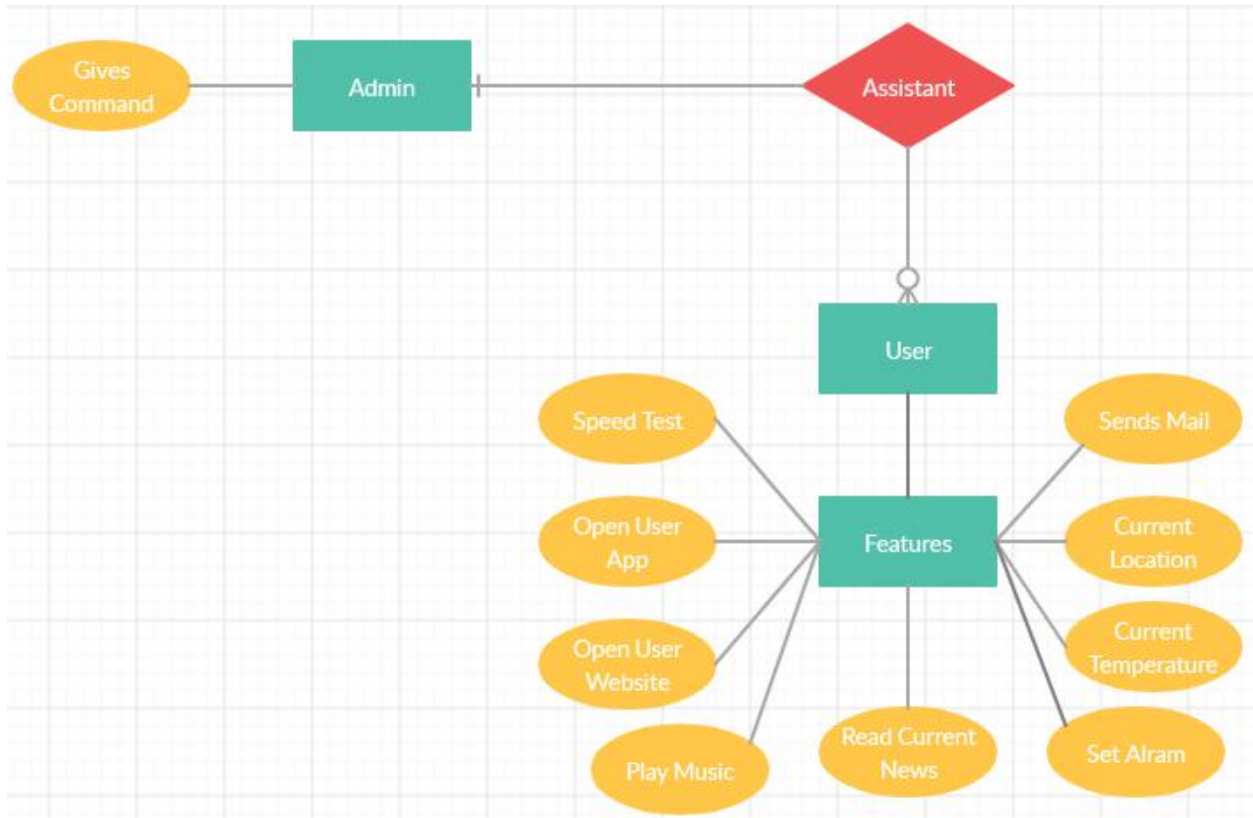
LIST OF FIGURES:

3.2.1 Use-Case Diagram



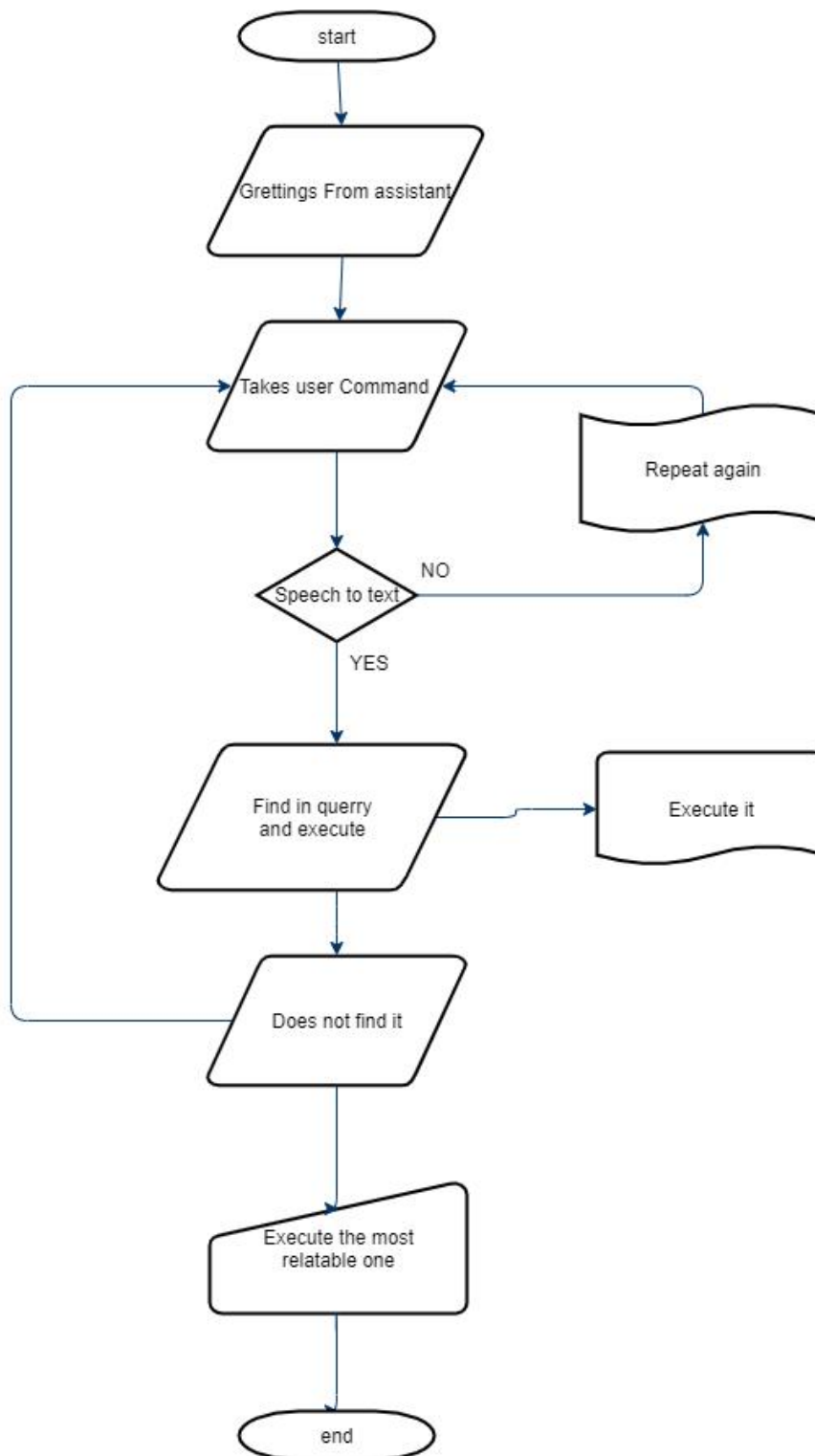
Use-case Diagram

3.2.2 ER-Diagram

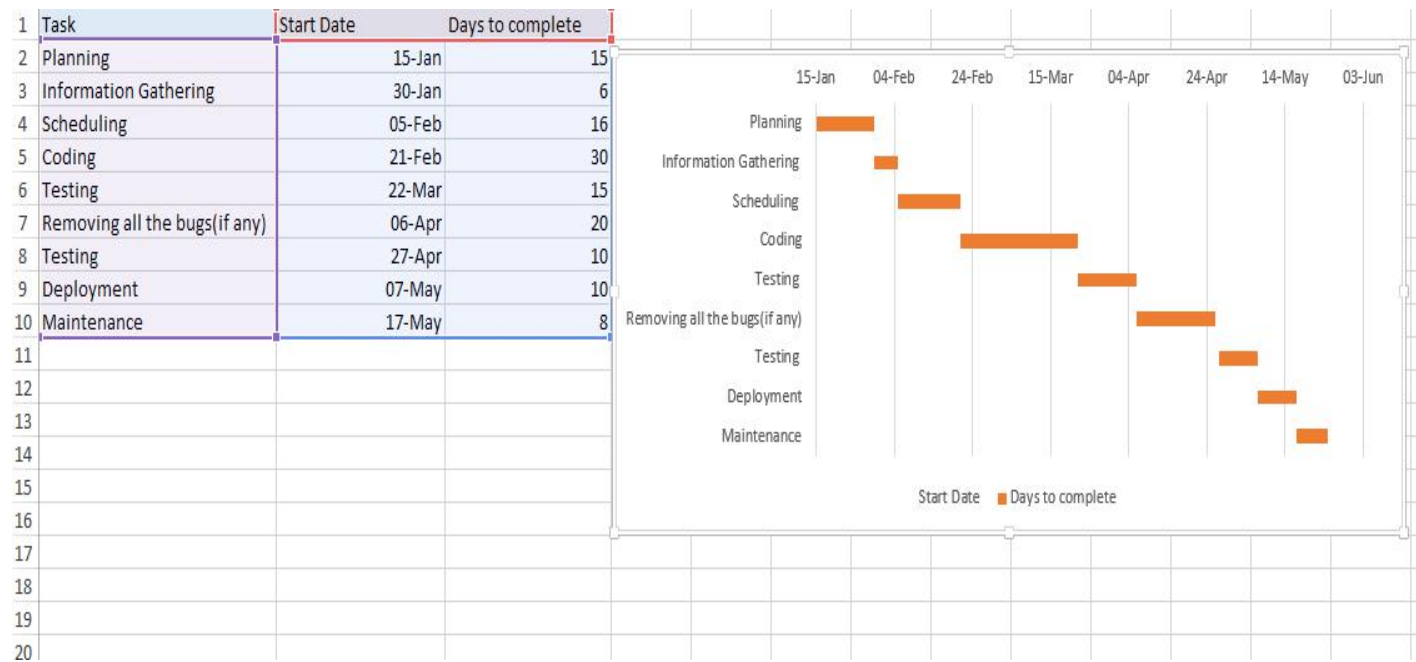


ER-Diagram

3.2.3 Flowchart:

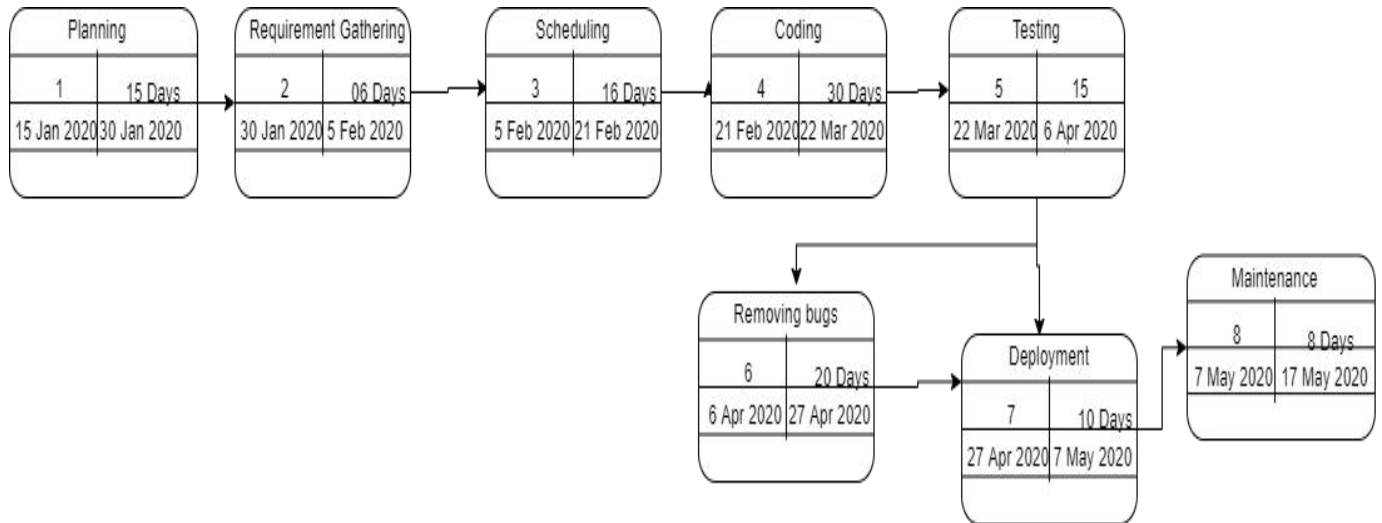


3.2.4 Gantt Chart:



Gantt chart

3.2.5 Pert Chart:



4. PROJECT IMPLEMENTATION AND TESTING

Personal Desktop Assistant is being implemented with the help of frameworks, programming languages, servers, and architecture.

- Framework used for implementation:

Anaconda3
Spyder IDE

- Programming Language Used:

Python 3.7

- Operating System Supported:

Windows OS 7 and Above.

Code:

Blue.py

```
import pyttsx3
import datetime#inbuilt
import speech_recognition as sr
import wikipedia
import webbrowser#inbuilt
import os#inbuilt
import smtplib#inbuilt
#import time#inbuilt
import requests
from bs4 import BeautifulSoup
from random import randint
import secrets
from playsound import playsound
import pyowm
import speedtest

#start
x = 'hello simran, how are you?'

def speak(t):
    engine = pyttsx3.init()
    s = engine.getProperty('voices')
    engine.setProperty('rate',160)#rate at which it will speak
    engine.setProperty('voice',s[0].id)#male or female voice
    engine.say(t)
    engine.runAndWait()

speak(x)

#greetings
def wishme():
    #greets you
```

```
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!")
```

```
speak("I am Blue. Please tell me how may I help you?")
```

```
#will listen to the voice and return what you said
```

```
def takeCommand(ask=False):
```

```
    #takes microphone input from the user and returns string output
```

```
    r= sr.Recognizer()#this class will help to recognize
```

```
    with sr.Microphone() as source:
```

```
        if ask:
```

```
            print(ask)
```

```
            print("Listening...")
```

```
            r.pause_threshold = 1 #seconds of non-speaking audio before a phrase is  
            considered complete
```

```
            audio=r.listen(source)
```

```
        try:
```

```
            print("Recognizing....")
```

```
            query = r.recognize_google(audio, language='en-in')#recognize_google  
            engine used (there are various other engine available too)
```

```
            print(f"User said: {query}\n")#f string used
```

```
        except Exception as e:
```

```
            print(e)
```

```
        print("Say that again please")
```

```
    return "None"#Just a string None
return query
```

```
def sendEmail(to, content):# to whom and what
    server = smtplib.SMTP('smtp.gmail.com',587)
    server.ehlo()
    server.starttls()
    server.login('simren.dubey@spit.ac.in','password')
    server.sendmail('simren.dubey@spit.ac.in',to,content)
    server.close()
```

```
def beautiful_soup(url):
    request = requests.get(url)
    soup = BeautifulSoup(request.text, "lxml")
    #print(soup.prettify())
    return soup
```

```
#def respond(query):
    #if "what is your name" in query:
    #    print("My name is Blue")
    #    speak("My name is Blue")
```

```
if __name__ == "__main__":
```

```
    wishme()
```

```
    # while True:
```

```
if 1:
    query = takeCommand().lower()
    #logic for execting task based on query

#Browser search
if "wikipedia" in query:
    speak('Searching wikipedia...')
    query = query.replace("wikipedia","")#replacing wikipedia from query and
making it blank
    results1 = wikipedia.summary(query, sentences=2)
    speak("According to wikipedia")
    print(results1)
    speak(results1)

elif "youtube" in query:
    webbrowser.open("youtube.com")

elif "google" in query:
    webbrowser.open("google.com")

elif "stackoverflow" in query:
    webbrowser.open("stackoverflow.com")

elif "gmail" in query:
    webbrowser.open("gmail.com")


#music in PC
elif "play music" in query:
    music_dir = "C:\\ganne"
    songs = os.listdir(music_dir)
    print(songs)#will print list of the songs

    os.startfile(os.path.join(music_dir,songs[0]))
```

```
#real time time
elif "time" in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"the time is {strTime}")
```

```
#open .exe file
elif "visual code" in query:
    codePath="C:\\Users\\simra\\AppData\\Local\\Programs\\Microsoft VS
Code\\Code.exe"
    os.startfile(codePath)
```

```
elif "netbeans" in query:
    codePath="C:\\Program Files\\NetBeans 8.0.1\\bin\\netbeans64.exe"
    os.startfile(codePath)
```

```
elif "chrome" in query:
    codePath="C:\\Program Files
(x86)\\Google\\Chrome\\Application\\chrome.exe"
    os.startfile(codePath)
```

```
elif "firefox" in query:
    codePath="C:\\Program Files\\Mozilla Firefox\\firefox.exe"
    os.startfile(codePath)
```

```
elif "dev" in query:
    codePath="C:\\Program Files (x86)\\Dev-Cpp\\devcpp.exe"
    os.startfile(codePath)
```

```
elif "quit" in query:
    speak("see you soon!")
    exit()
```



```
elif "email" in query:
```

```
    try:
```

```
        speak("What should I say?")
```

```
        content = takeCommand()
```

```
        to = "simrendubey@spit.ac.in"
```

```
        sendEmail(to, content)
```

```
        speak("Email has been send")
```

```
    except Exception as e:
```

```
        print(e)
```

```
        speak("Sorry bhai email nahi bhej sakta hu mai")
```

```
elif "name" in query:
```

```
    print("My name is Blue")
```

```
    speak("My name is Blue")
```

```
elif "jokes" in query:
```

```
    foo = ['Why did the teddy bear say no to dessert?\n\tBecause she was  
stuffed\n'
```

```
          'What did the left eye say to the right eye?\n\tBetween us, something  
smells!\n'
```

```
          'What do you get when you cross a vampire and a snowman?\n\tFrost  
bite!\n'
```

```
          'What did one plate say to the other plate\n\t?Dinner is on me\n'
```

```
          'When you look for something, why is it always in the last place you  
look?\n\tBecause when you find it, you stop looking\n'
```

```
          'What is brown, hairy and wears sunglasses?\n\tA coconut on vacation.\n'
```

```
    ]
```

```
    rando=secrets.choice(foo)
```

```
    print(rando)
```

```
    speak(rando)
```

```
elif "search" in query:
```

```
    speak("what do you want to search")
```

```
search = takeCommand("what do you want to search")
Path='https://google.com/search?q='+ search
webbrowser.get().open(Path)
speak("Here is what I found on web")
print('Here is what I found on web ' + search)
```

```
#elif "location" or "place" in query:
```

```
# speak("which place do you want to search")
# location = takeCommand("which place do you do you want to search")
# url='https://google.nl/maps/place/'+ location + '/&';
# webbrowser.get().open(url)
# speak("Here is what I found on web")
# print('Here is what I found on web ' + location)
```

```
elif "alarm" or "Wake me" in query:
```

```
    speak("What time do you want to set the alarm for")
    # os. system('clear')
    speak("Set the hour")
    alarmH = int(input("What hour do you want the alarm to ring? "))
    speak("Set the minute")
    alarmM = int(input("What minute do you want the alarm to ring? "))
    amPm = str(input("am or pm? "))
    # os. system('clear')
    print("Waiting for alarm",alarmH,alarmM,amPm)
    if (amPm == "pm"):
        alarmH = alarmH + 12
    while(1 == 1):
        if(alarmH == datetime.datetime.now().hour and
           alarmM == datetime.datetime.now().minute) :
            print("Time to wake up")
            speak("Get up")
            playsound('C:/ganne/Baby Shark.mp3')
            break
```

```
elif "news" or "headline" in query:
```

```
soup = beautiful_soup('https://news.google.com/?hl=en-IN&gl=IN&ceid=IN:en')
for headlines in soup.find_all('a', {'class': 'VDXfz'}):
    resultss=headlines.find_next('span').text
    print(resultss)
    #beautiful_soup(url)
    speak(resultss)

elif "current" or "now" in query:
    #geojs website

    raa = requests.get('https://get.geojs.io/')
    ip_request = requests.get('https://get.geojs.io/v1/ip.json')

    ipAdd = ip_request.json()['ip']
    print(ipAdd)

    url1 = 'https://get.geojs.io/v1/ip/geo/'+ipAdd+'.json'
    geo_request = requests.get(url1)
    geo_data = geo_request.json()

    #print(geo_data)
    speak("Your current contry location is")
    print(geo_data['country'])
    speak(geo_data['country'])

    speak("Your current city location is")
    print(geo_data['city'])
    speak(geo_data['city'])

    speak("Your current region location is")
    print(geo_data['region'])
    speak(geo_data['region'])

    speak("Your current location latitude is")
    print(geo_data['latitude'])
    speak(geo_data['latitude'])
```

```
speak("Your current location longitude is")
print(geo_data['longitude'])
speak(geo_data['longitude'])
```

```
speak("Your current location timezone is")
print(geo_data['timezone'])
speak(geo_data['timezone'])
```

```
elif "weather" or "temprature" in query:
    #https://home.openweathermap.org/api_keys
    speak("Blue is here to give you temprature update")

    owm = pyowm.OWM('e27d8164b3234c34502aa9a41d68040e')

    speak("Tell me the location please")

    place1 = takeCommand("Tell me the location please")
    location1 = owm.weather_at_place(place1)
    weather = location1.get_weather()
    #print(weather)

    temp = weather.get_temperature('celsius')
    humidity = weather.get_humidity()

    print(temp)
    speak("Here is the current temprature")
    speak(temp)
    print(humidity)
    speak("Here is the current humidity level")
    speak(humidity)

    #print(humidity)
    # for key,value in temp.items():
    #     print(key,value)
```

```
elif "feature" or "what can you do" in query:
    speak("Blue is always there to help you here is the list of what I can do")
    print("Blue is always there to help you here is the list of what I can do")
    speak("Open a web page for example open google,youtube")
    print("Open a web page for example open google,youtube")
    speak("Open application for example open vs Code or netbeans")
    print("Open application for example open vs Code or netbeans")
    speak("Tells the weather")
    print("Tells the weather")
    speak("Tells the current date and time")
    print("Tells the current date and time")
    speak("Set an alarm")
    print("Set an alarm")
    speak("Tells you news for today")
    print("Tells you news for today")
    speak("Tells you the current location")
    print("Tells you the current location")
    speak("I am still under development soon I will be able to do other things too")
```

```
elif "speedtest" or "speed" or "test" in query:
    ay=speedtest.Speedtest()
    print("Speed test is here")
    speak("Blue is here to give you Speed test")
```

```
    print("Upload Speed")
    speak("Upload Speed")
    speak(ay.upload())
    print(ay.upload())
```

```
    print("download speed")
    speak("download speed")
    speak(ay.download())
    print(ay.download())
```

```
    print("Ping result")
    speak("Ping result")
    speak(ay.results.ping)
    print(ay.results.ping)
```

SNAPSHOT OF USER INTERFACE:

```
In [3]: runfile('C:/Users/simra/OneDrive/Desktop/blue.py', wdir='C:/
Users/simra/OneDrive/Desktop')
Listening...
Recognizing....
User said: news

President Donal Trump on Wednesday said he has offered to arbitrate the
India-China border dispute in what is being seen as move that will irk
Beijing more ...
Chinese President Xi Jinping on Tuesday ordered the military to scale
up the battle preparedness, visualising the worst-case scenarios and
asked them to ...
India has pushed in high altitude warfare troops with support elements
to the eastern Ladakh theatre to counter Chinese People's Liberation
Army (PLA) ...
India clearly doesnt want to be drawn into a united front with those
having serious problems with China.
Tell Beijing its escalation will impact the overall relationship.
A woman on board a Shramik Special train gave birth to a baby boy at
Ratlam railway station on Wednesday. Pooja Devi, who was travelling
```

```
In [2]:

In [2]: runfile('C:/Users/simra/OneDrive/Desktop/Blue/current.py',
wdir='C:/Users/simra/OneDrive/Desktop/Blue')
Listening...
Recognizing....
User said: where am I now

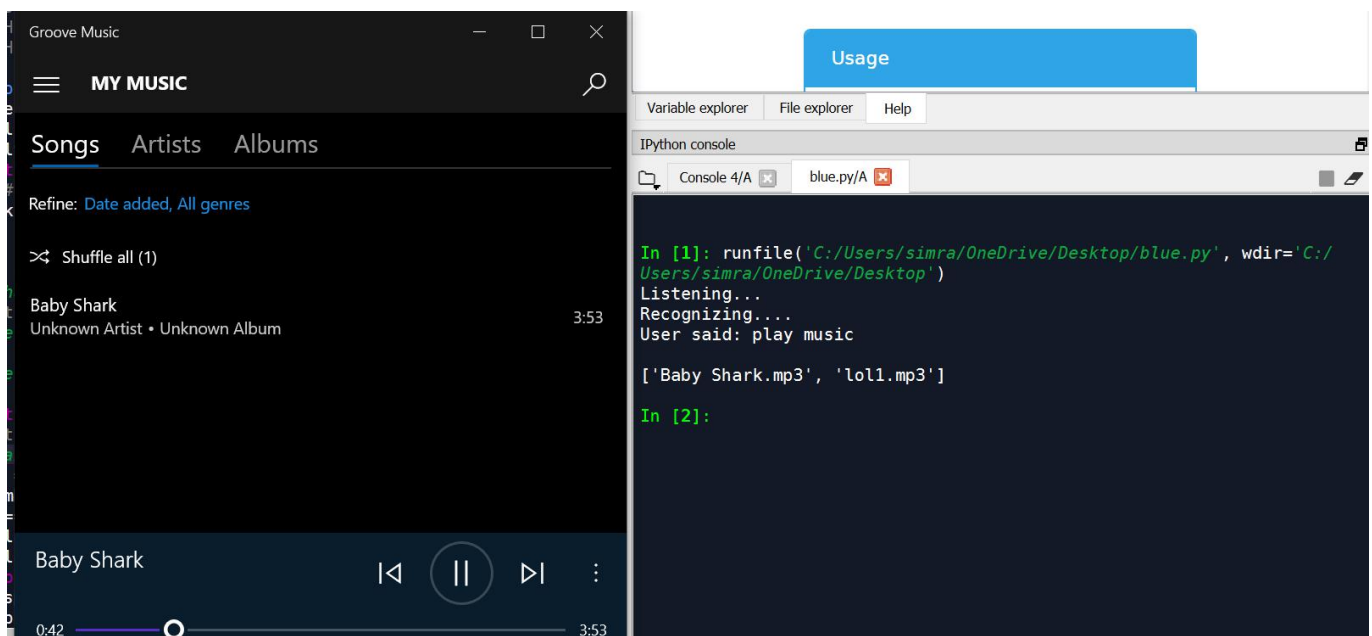
103.216.214.72
India
Mumbai
Maharashtra
19.0748
72.8856
Asia/Kolkata

In [3]:
```

```
In [3]: runfile('C:/Users/simra/OneDrive/Desktop/Blue/feature.py',  
wdir='C:/Users/simra/OneDrive/Desktop/Blue')  
Listening...  
Recognizing....  
User said: what are you feature  
  
Blue is always there to help you here is the list of what I can do  
Open a web page for example open google,youtube  
Open application for example open vs Code or netbeans  
Tells the weather  
Tells the current date and time  
Set an alarm  
Tells you news for today  
Tells you the current location  
  
In [4]:
```

```
In [4]: runfile('C:/Users/simra/OneDrive/Desktop/Blue/speed.py',  
wdir='C:/Users/simra/OneDrive/Desktop/Blue')  
Listening...  
Recognizing....  
User said: please do the speed test  
  
Speed test is here  
Upload Speed  
5224451.956581926  
download speed  
8500620.940766163  
Ping result  
14.034
```

```
In [1]: runfile('C:/Users/simra/OneDrive/Desktop/Blue/temprature.py',  
wdir='C:/Users/simra/OneDrive/Desktop/Blue')  
Listening...  
Recognizing....  
User said: give me the weather update  
  
Tell me the location please  
Listening...  
Recognizing....  
User said: Mumbai  
  
{'temp': 30.0, 'temp_max': 30.0, 'temp_min': 30.0, 'temp_kf': None}  
74
```



TEST CASES:

Test Case ID	Test Case Name	Test Data	Expected Output	Result
1	User want to open YouTube.	Speak 'open YouTube' then your voice recognize and convert into text And open YouTube on Your Screen	Display YouTube on Your Windows Screen	Pass
2	User want to Open YouTube	User speak wrong sentence or some disturbing noise are their	Error Occurred (Not understand by Desktop Assistant)	Fail
3	User want to Play Music	Speak 'Play Music' then your voice recognize and convert into text and Music Play	Show Music Player on the screen and music in on	Pass
4	User Want to Search Location	Speak 'Location' then your voice recognize and convert into text And show your location on The basis of your IP address on your screen(It display your current State, current City and Current Time on screen)	Show your Location on screen	Pass

5. DOCUMENTATION AND INSTALLATION:

5.1 INSTALLATION:

Module installation:

- 1)import pyttsx3
 ->pip install pyttsx3
- 2)import datetime#inbuilt
 INBUILT
- 3)import speech_recognition as sr
 ->pip install speechrecognition
- 4)import wikipedia
 ->pip install wikipedia
- 5)import webbrowser#inbuilt
 INBUILT
- 6)import os#inbuilt
 INBUILT
- 7)import smtplib#inbuilt
 INBUILT
- 8)import time#inbuilt
 INBUILT
- 9)import requests
 ->pip install requests
- 10)from bs4 import BeautifulSoup
 ->pip install BeautifulSoup
- 11)from random import randint
 ->pip install random
- 12)import secrets
 -> pip install secrets
- 13)from playsound import playsound
 -> pip install playsound
- 14)import pyowm
 -> pip install pyowm
- 15)import speedtest
 -> pip install speedtest

6. SYSTEM MAINTENANCE:

- ✓ System Maintenance is needed when new version of the existing software is released.
- ✓ System Maintenance should be carried out at least quarterly to check weather everything about application is working fine.
- ✓ Maintenance works is also carried out when the system fails to work properly.
- ✓ Maintenance require for update new feature in Personal Desktop Assistant.
- ✓ Maintenance require for Desktop Assistant to Listen voice properly and convert into text.

7. FUTURE ENHANCEMENT

This Project can be Enhanced in many ways as this project have lot of scope in future the possibility of adding feature to this project is endless. This assistant is use in real time. Some features like news can be more enhanced and can ask you from which newspaper or site you want news to be read for you. There can be a provision to add reminder for you like for drinking water or whatever personal work you want your assistant to do for you. This assistant can have many games for user to play and certain FAQs. This system can further send a message to someone using APIs condition those number have to be linked to the Assistant.

8. LIMITATION:

- Requires constant and stable internet connection to use Application.
- Is not fully secured there are vulnerabilities.
- You can use feature which are present in software only.
- There is no android or ios system right now. It's only run on machine as well as in software.

9. CONCLUSION:

- Throughout the history of computing, we need technology to advance and ease our life.
- We tried to implement this project for our betterment and for the goal to save time.
- This project is completely depend on Internet speed which is a hurdle.
- We learned a lot of things and about new module through this project.

10. BIBLIOGRAPHY

Here is our reference through which we created this project

- 1) <https://www.youtube.com/watch?v=Lp9Ftuq2sVI>
- 2) <https://www.youtube.com/watch?v=yVj1WELaG3I>
- 3) https://www.youtube.com/watch?v=K_WbsFrPUck
- 4) https://www.youtube.com/watch?v=Jt1w29_0J8E
- 5) <https://www.activestate.com/blog/how-to-build-a-digital-virtual-assistant-in-python/>