GOVERNMENT OF KARNATAKA
DEPARTMENT OF TECHNICAL EDUCATION

GOVERNMENT POLYTECHNIC KARWAR

Karwar-581301

2019-20

A PROJECT REPORT ON
"J.A.R.V.I.S"
(Just A Rather Very Intelligent System**)**

Submitted to partial fulfillment for the award of the diploma in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Ms. SAHANA. R. DURGEKAR (114CS17035)

Mr. SUDEEP. AGER (114CS17052)

Ms. SULAKSHA. S. PADTI (114CS17053)

Ms. TEJASWINI. U. GUNAGI (114CS17059)

Ms. VAISHNAVI. P. NAIK (114CS17060)

Under the guidance of
Mr. VINAY BHAT

Lecturer
Department of
Computer Science and Engineering

Year of submission: March 2020

GOVERNMENT OF KARNATAKA
DEPARTMENT OF TECHNICAL EDUCATION

GOVERNMENT POLYTECHNIC KARWAR

Karwar-581301

2019-20

A PROJECT REPORT ON

““J.A.R.V.I.S”

(Just A Rather Very Intelligent System)

Submitted to partial fulfillment for the award of the diploma in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Ms. SAHANA. R. DURGEKAR (114CS17035)

Mr. SUDEEP. AGER (114CS17052)

Ms. SULAKSHA. S. PADTI (114CS17053)

Ms. TEJASWINI. U. GUNAGI (114CS17059)

Ms. VAISHNAVI. P. NAIK (114CS17060)

Under the guidance of
Mr. VINAY BHAT

HOD
Department of
Computer Science and Engineering

CANDIDATE'S DECLARATION

We, **Sahana. Durgekar, Sudeep. Ager, Sulaksha.Padti, Tejaswini. Gunagi, Vaishnavi. Naik** the student of **Diploma in Computer Science and Engineering Department** bearing Register Number **114CS17035, 114CS17052, 114CS17053, 114CS17059, 114CS17060** of **Government Polytechnic Karwar**, hereby declare that, we owe full responsibility for the information, results and conclusions provided in this project work titled **"J.A.R.V.I.S"** submitted to **Board of Technical Examinations, Government of Karnataka** for the award of **Diploma in Computer Science and Engineering.**

To the best of our knowledge, this project work has not been submitted in part or full elsewhere in any other institution/organization for the award of any certificate/diploma/degree. We have completely taken care in acknowledging the contribution of others in this academic work. We further declare that in case of any violation of intellectual property rights and particulars declared, found at any stage, we as the candidate will be solely responsible for the same.

Date:

Place:

Signature of candidates:

Name: Sahana. Durgekar
Reg.No: 114CS17035

Name: Sudeep. Ager
Reg.No: 114CS17052

Name: Sulaksha.Padti
Reg.No: 114CS17053

Name: Tejaswini. Gunagi
Reg.No: 114CS17059

Name: Vaishnavi. Naik
Reg.No: 114CS17060

GOVERNMENT OF KARNATAKA

DEPARTMENT OF TECHNICAL EDUCATION

GOVERNMENT POLYTECHNIC KARWAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2019-20

BONAFIDE CERTIFICATE

Certified that this project report "**J.A.R.V.I.S**" is the bonafide work of **Ms.Sahana. Durgekar, Mr.Sudeep.Ager, Ms.Sulaksha.Padti, Ms.Tejaswini.Gunagi, Ms.Vaishnavi.Naik** bearing Register Number **114CS17035, 114CS17052, 114CS17053, 114CS17059, 114CS17060,** of this institution who carried out the project work under my supervision.

Project Guide

Mr.Vinay Bhat
Department of
Computer Science and Engineering

Head of Department

Mr.Vinay Bhat
Department of
Computer Science and Engineering

GOVERNMENT OF KARNATAKA

DEPARTMENT OF TECHNICAL EDUCATION

GOVERNMENT POLYTECHNIC KARWAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2019-20

CERTIFICATE

Certified that this project report entitled "**J.A.R.V.I.S**" which is being submitted by **Ms.Sahana. Durgekar, Mr. Sudeep. Ager, Ms. Sulaksha.Padti, Ms.Tejaswini. Gunagi, Ms.Vaishnavi. Naik** Register Number, **114CS17035, 114CS17052, 114CS17053, 114CS17059, 114CS17060** a bonafide student **of government polytechnic karwar** in partial fulfillment for the award of **Diploma in Computer Science and Engineering** during the year **2019-20** is record of students indicated for internal Assessment have been incorporated in the Report and one copy of it own work carried out under my/our guidance. It is certified that all corrections/suggestions being deposited in the polytechnic library.

The project report has been approved in partial fulfillment for the said degree as per academic regulations of Government Polytechnic Karwar..

Project Guide
Mr.Vinay Bhat
Department of
Computer Science and Engineering

Head of Department
Mr.Vinay Bhat
Department of
Computer Science and Engineering

Principal
Shri.V.M.Hegde
Govt.Polytechnic Karwar

Name and signature Examiner

1 _____

2 _____

ACKNOWLEDGEMENT

We the students of final year diploma in Computer Science and Engineering of Govt. Polytechnic Karwar avail this opportunities to express our heartfelt gratitude to our lecturers, principal and all those people who were supportive enough to guide us throughout the completion of the project.

We owe our sincere thanks to our HOD and Guide **Mr**. **Vinay Bhat HOD Department of Computer Science and Engineering** for this esteem corporation in successful completion of the project.

We express our sincere gratitude to our **Mr. Sayyan Shaikh, Ms. Rashmi Vernekar** for their guidance and support, in making the project execute at its best.

We also thank **Shri.V. M. Hegde,** Principal of Govt. Polytechnic Karwar for providing us a platform to do this project.

We also thank **Mr. Mohemmed Akbar Shaikh, DOLFIN SOFTS KARWAR** for providing us a platform to do this project.

Finally we thank to all those who are in directly involved for successful completion of project well in time.

Ms.**Sahana. Durgekar**
Mr. Sudeep. Ager
Ms. Sulaksha.Padti
Ms.Tejaswini. Gunagi
Ms.Vaishnavi. Naik

ABSTRACT

This project is aimed at developing a desktop application for easy to use. **JARVIS** is a digital life assistant, based on python language. It assists with your daily life by opening application as we command, browsing information on the internet, searching for specified locations on the google maps, adjusting brightness etc. It is the new way of computing which provide computing power without touching the computer system. In this project it also provides the power of computing with the help of voice i.e., SPEAKING and System will also talk back to you like Humans in the real world. This application is a miracle in computer interaction field by which we can say good by to mouse and keyboard.

J.A.R.V.I.S

(Just A Rather Very Intelligent System**)**

CONTENTS

CHAPTER: 1

INTRODUCTION

Project **J.A.R.V.I.S** –work on the basis of VOICE RECOGNITION. It recognizes the voice spoken by human with the help of some predefine commands and perform tasks as required by the user.

It requires Microsoft Windows like Windows 8, 7

It is open source software. Jarvis assists with your daily life by voice command, informing you of the latest news headlines, the forecast of the weather for the upcoming days.

1.1 Purpose: The purpose of this project is to build a desktop application that will be able to service to humans like a personal assistant. This is an interesting concept and many people around the globe are working it. Today, time and security are the two main things to which people are more sensitive, no one has the time to spoil; nobody would like their security breach, and this project is mainly for those kinds of people.

1.2 Scope: The scope of this project is to help handicapped peoples who are unable to turn on/off gadgets, people who don't know much about computer systems and also in schools, colleges, offices, hospitals laboratories and many other organizations as a better way of interacting with computer system.

CHAPTER: 2

OBJECTIVES

The main objective of our project is to implement all these features:

- Talk to JARVIS: talks back like humans so that one don't get bored.
- Touch less: Without touching the screen.
- General searches: Search for any known answers or searches.
- Dynamic conversations: Jarvis is dynamic in conversations with you. It has a variety of answers for variety of questions. If it doesn't know any, it learns them overtime or searches the net.
- Computer termination: Log off, shutdown and restart by saying the command.
- Custom commands: Add custom commands of your own choice.
- Play video: Opens and plays video files.
- Play Music: Opens and plays Music files.
- Social commands: Shows a list box of greetings and other commands.
- Weather commands: Responds with weather and temperature updates.
- Web commands: Open web based commands.

  And many more.

CHAPTER: 3

TOOLS OR ENVIRONMENT USED

3.1 Hardware Requirements:

1. Processor : Pentium 4 or Hiegher.
2. Ram : 500MB +
3. Hdd : 20GB
4. Keyboard : 104 keys
5. Mouse : 2 buttons/3 buttons
6. Monitor : VGA/SVGA
7. Additional device : Microphone enabled headphone or a simple MIC;
   Speakers.
   USB 2.0 PORT;

3.2 Software Requirements:

1. Windows : Any Windows (Python is Platform Independent)
2. Front end framework : Tkinter
3. Back end : Python programming language (Editor used: Python VsCode Editor)

CHAPTER: 4

ANALYSIS DOCUMENT

4.1 Information Retrieval

As this program includes the functions and services of search online information, Open files, Weather Information, location services, music player service, Mathematics, Wikipedia searching engine, Date and time, hibernate, shutdown and restart your PC, adjust brightness, send e-mail, open recent activities running on your windows system, tell you about system information. The list below indicates the information and the requirements of each individual function.

The program has two modes to well fetch the services and functions. The program will start with voice mode as its primary mode to provide the voice assistant, but the user can select switching to the text mode if he or she is not well working with the voice mode or the surrounds don't support the voice recognition well.

- **Search Online Information:** The search engine enables the user to search anything on Google. The search engine will give result list back and displayed on the browser.

- **Open files or applications:** Here we use the os() module to open a file or any application. User will give input by voice command or by text to open particular file or application. The Jarvis will perform the specified command by opening the file or application.

- **Weather Information:** The user could check the weather in any place. In addition, the weather is returned with the temperature and humidity; the user could also check the weather for current day, tomorrow or in next four days.

- **Wikipedia searching engine:** The search engine enable the user to search anything on Wikipedia. The result is given back on the web browser with the searched content on Wikipedia.

- **Date and time:** Tells current date and time

- **Location services:** location services provide the functions for the user to check the current location or find the direction to a destination. The user should get an easy to understand map with the locations or routes depending on the category of the request.

- **Music player service:** The music player offers the services to the user to play a named or a randomly picked song in the pre-stored song list on the mobile phone. And it could be stopped when the user wants to terminate it.

- **Hibernate, shutdown and restart your PC:** Jarvis will hibernate, shutdown and restart your PC as commanded by the user.

- **Adjust brightness:** Alters system brightness.

- **E-mail service:** Users are able to send the mail to the person with mail address in the contacts. By giving a correct command contains the mail request keyword together with the destination person; the mail should be received by the recipient after it has been sent.

- **Open recent activities:** open recent activities running on your windows system.

- **Tell you about system information:** Jarvis will opens the performance monitor and displays the system information such as network usage, ram etc.

- **Create folder:** it will creates folder as commanded

3.2 Theory Model

The project is based on the theories related to various aspects of software engineering principles and software development model.

- **Software engineering principles**

  Extreme programming will direct the development process of the project, it focus on the development cycle of defining the requirement, corresponding design and test, integration and simplicity; during the development, there should always be working in pair programming, as well as doing the revision control, calculate the velocity and efficiency.

  Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles (time boxing), which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. The developments will on the small cycle model

repeatedly, every cycle will have analysis, design, implementation and test. Figure-1 somehow shows how to follow the XP develop model.

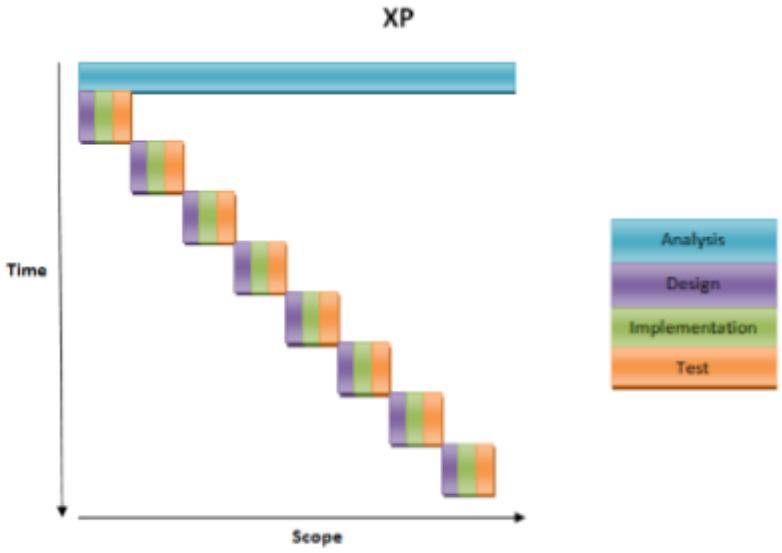


Figure-1

Project Lifecycle

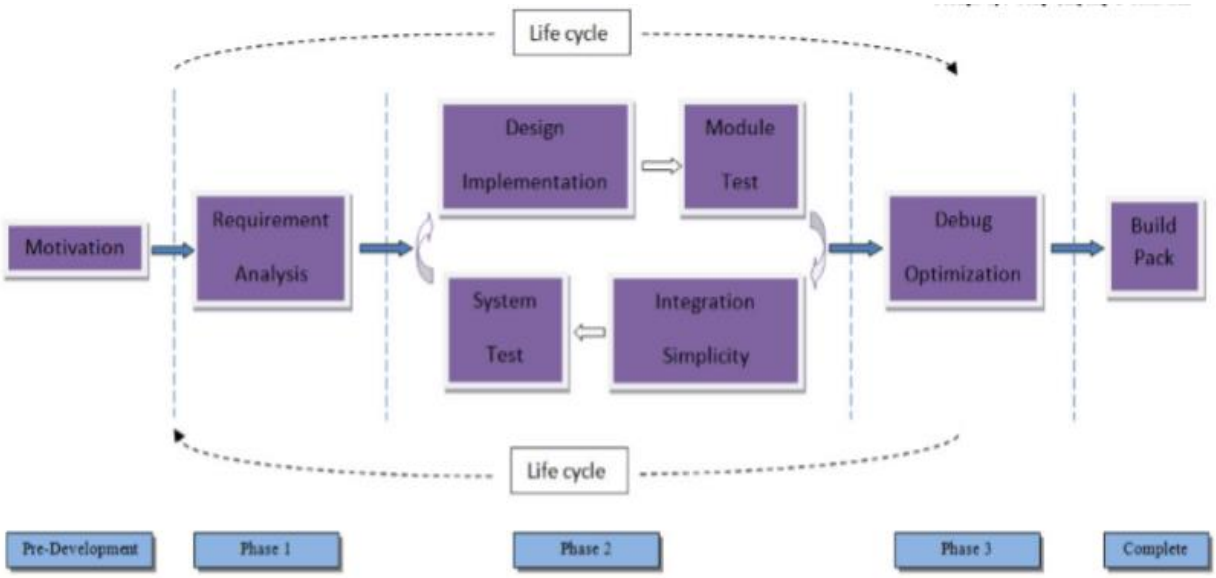


Figure-2

The Model and Flow Chart (Figure-2) describes the develop process that include all the phases in the software development life cycle. This chart is well illustrating how the project is carried out and how the development was managed. The project started with the motivation and brain storm, repeatedly implement in the developing life cycle until the system has been fully constructed.

- Brain storm, the project start with the ideas from the brain storm. Here the basic ideas and design the primary concepts, prototype of the program have been obtained.

- While the ideas has been obtained, it has been analyzed which of them can be accomplished and make sure the structure of the project.

- According to the requirements that had been identified, collected all the resources and useful references from any channel, together with the programming skills and experiences, the design items were pointed out.

- Implement each individual design item based on the planning, structure and references.

- Test each single module that has been implemented and fix the possible bugs appear in the code implementation and make sure the functions are well constructed.

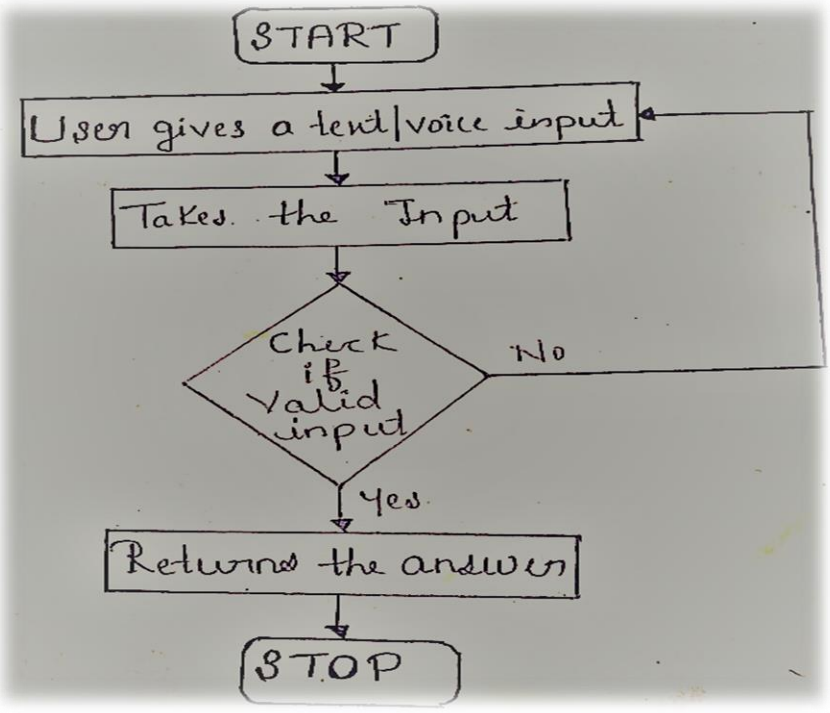
- Integrate all the individual sections to contribute to a complete system.

- Try the black and white box testing strategies to test the system, both the functional and nonfunctional logic and implementation should be verified.

4.3 Flow Chart and Diagrams

4.4.1 Flow Chart

The below figure[3] shows the flowchart for over all voice recognition process**.**



Figure[3]

4.4.2 Activity Diagram:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc as shown in below Figure[4]

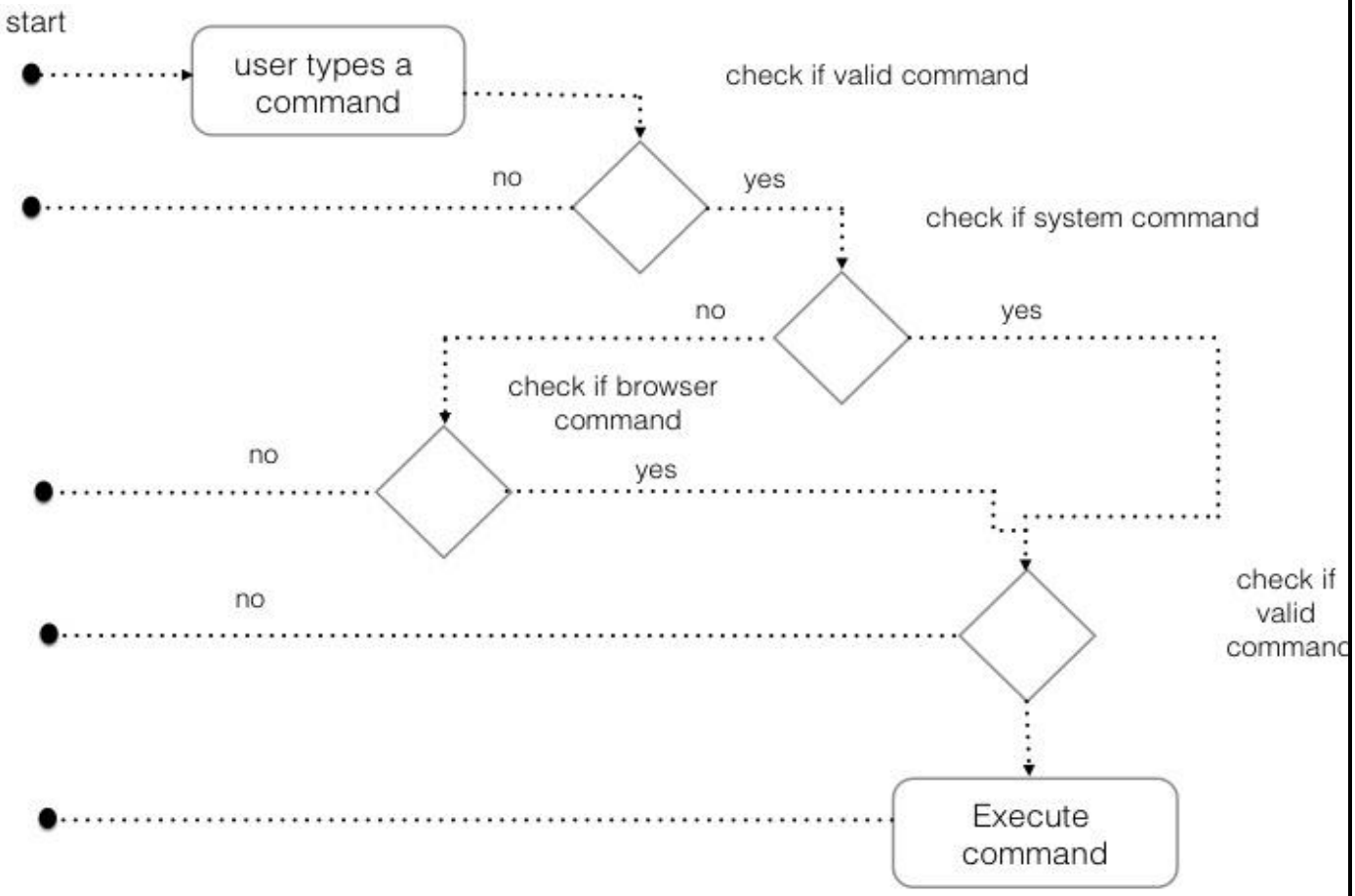
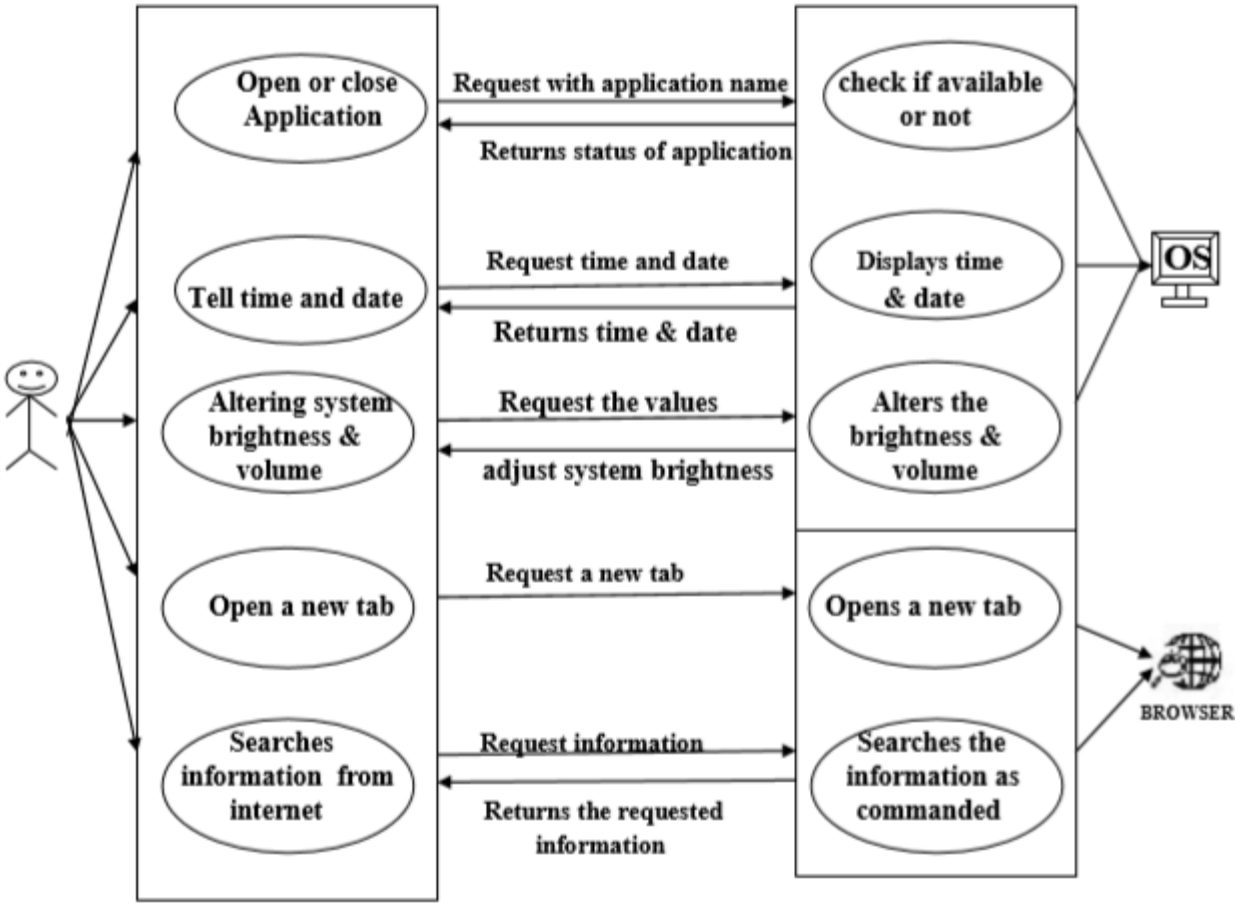


Figure [4]: Activity Diagram

4.4.4 Use Case Diagram:

A use case is a set of scenarios that describe an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors as shown in below Figure[5].



Figure[5]: Use Case Diagram

3.4 Environmental Consequences

This program is green to the environment and no pollution will be generated by the software or hardware. During the development, the process will not do any harm to the surrounding environment since it is software development on the computer.

CHAPTER: 5

DESIGN DOCUMENT

5.1 Frontend Design

Front end is designed with tkinter().

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

To Create a GUI application using Tkinter, we need to perform the following steps

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

5.2 Backend Design

The backend is designed in python programming language.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms.

Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

5.2.1 Module we need for this Desktop Assistant:

- Espeak - if you are in linux based operating system. In windows you did not have to install this espeak
- Pyttsx3
- webbrowser
- random
- datetime
- speech_recognition as sr
- wikipedia
- os
- wmi
- smtplib
- subprocess
- ctypes

5.2.2 Command to install Modules:

- **Pyttsx3**- pip install Pyttsx3
- **Webbrowser**- pip install pycopy-webbroswer
- **Random**- pip install random2
- **Datetime**- pip install datetime2
- **speech_recognition**- pip install speech_recognition
- **Wikipedia**- pip install wikipedia
- **Os**- pip install os
- **Wmi**- pip install Wmi
- **Smtplib**- pip install secure- Smtplib
- **Subprocess**- pip install subprocess.run
- **Ctypes**- pip install ctypes

5.3 User Interface:

The program should firstly be started on the desktop. The initial mode of the program is Voice mode since this program aims at making a voice assistant program. However, if there are users who prefer to operate in text mode by inputting the text manually, the text mode is also available.
After the application has been started, click on arrow icon then the chat frame will be open. There is a label named JARVIS and another label at the middle called as canvas where we write the text and another frame at bottom which consists of listening and sending button. The user should have correct voice input "command/request" to make those functions work properly.

5.4 Functioning

5.4.1 Text mode:

After entering the text in the chat frame, the submit button is clicked which calls the sumbit function. It stores the chat (i.e. that is the chat what we chat) when submit () is called, conver() also get called. Then the text which we type will be displayed at left. The command is taken and control is passed to conver() and conver() searches for the specified command which is saved to the variable called as query. If the query matches the user command, it will returns the answer and will be displayed in the label towads right else it displays the message **"Sorry master I didn't get command please say it again or try typing the command."**

5.4.2 Voice mode:

When listen button is called, listen() is called along with the takeCommand().The takeCommand() recognize user speech and convert it into text from which it is saved in variable called as query and this query is displayed on left side of canvas and the control takes query to canver(). And conver() searches for the specified command which is saved to the variable called as query. If the query matches the user command, it will return the answer and will be displayed in the label towards right else it displays the message **"Sorry master I didn't get command please say it again or try typing the command."**

CHAPTER: 6

PROGRAM CODE

main.py

```
from lispk import speak, takeCommand
from conv import *
from tkinter import *
from tkinter import ttk
from conv import conver
root = Tk()
root.geometry("600x700")
root.maxsize(600, 700)
root.minsize(600, 700)
root.title('J.A.R.V.I.S')
A = PhotoImage(file='image_5.png')  # personal assistant image in welcome frame
b = PhotoImage(file="fw.png")  # forword button
c = PhotoImage(file="rec.png")  # listen button in chat frame
d = PhotoImage(file="ssss.png")  # submit button in chat frames


def topic_to_chat():

    chat()

# when user enters text in entry widget submit function is called


def get_response():

    global label_response
    label_response = Label(frame_chats, text=answer.get(), bg="#007FFF", fg="white",
                           justify=LEFT, wraplength=300, font='Arial 12 bold')

    label_response.pack(anchor='e')

    if answer == 'Bye':
        root.destroy()


def submit():
    global chat_raw
    chat_raw = entry.get('1.0', 'end-1c')

    entry.delete('1.0', END)

    chat = chat_raw.lower()
    # chat = chat.replace(' ', '')
    global label_request

    label_request = Label(frame_chats, text=chat, bg="white", fg="black",
                          justify=LEFT, wraplength=300, font='Arial 12 bold')
```

```
    label_request.pack(anchor='w')

    global answer
    answer = StringVar()
    answer.set(conver(chat))
    get_response()


# when listen button is pressed listen() function will bi called
def listen():
    """
function for producing response of
        request of user

    """


    global label_request
    takeCommand1 = StringVar()
    takeCommand1.set(takeCommand())

    label_request = Label(frame_chats, text=takeCommand1.get(), bg="white", fg="black",
                          justify=LEFT, wraplength=300, font='Arial 12 bold')

    label_request.pack(anchor='w')
    global answer
    answer = StringVar()

    take = takeCommand1.get()

    answer.set(conver(take))
    get_response()


# def func():
#     res1 = StringVar()
#     res1.set(conver(takeCommand()))
#     print(res1.get())
#     Label(root, text=res1.get()).pack()
#     # print(res1.set(res))
""" main chat frame"""


def chat():
    frame_welcome.pack_forget()
    frame_chat = Frame(root, bg="grey", height='670',
                       width='550')

    frame_chat.pack_propagate(0)

    frame_top = Frame(frame_chat, bg="#9adcfe", height='100', width='550')
    frame_top.pack()

    label_topic = Label(frame_top, bg="#9adcfe", text="welcome to JARVIS", fg='black',
                        font='Verdana 20 bold ')
    label_topic.pack(pady='40')
```

```
    frame_spacer = Frame(frame_top, bg="black", height="10", width="550")
    frame_spacer.pack()
    bottom_frame = Frame(frame_chat, bg="black", height='100', width='550')
    bottom_frame.pack_propagate(0)
    bottom_frame.pack(side=BOTTOM)
    lis_button = Button(bottom_frame, image=c, relief="solid",
                    font='Vardana 10 bold', bg="black", activebackground="black", command=listen)
    lis_button.place(x=470, y=20)
    submit_button = Button(bottom_frame, image=d, relief="solid",
                      font='Vardana 10 bold', bg="black", activebackground="black", command=submit)
    submit_button.place(x=400, y=20)

    global entry
    entry = Text(bottom_frame, bg="white", fg="black", height='5',
                width='45', font='Verdana 10')
    entry.bind('<Return>', submit)
    entry.place(x=20, y=10)
    global frame_chats

    frame_chats = Frame(frame_chat, bg="grey", height='450',
                    width='500')
    frame_chats.pack_propagate(0)
    frame_chats.pack()

# Label(frame_chats, bg="#7B68EE").pack()

    frame_chat.pack()


# welcome frame
frame_welcome = Frame(root, bg="#7B68EE", height='670', width='550')
frame_welcome.pack_propagate(0)
frame_welcome.pack()
welcome = Label(frame_welcome, text='Welcome',
                font="Vardana 40 bold", bg="#7B68EE", fg="black")
welcome.place(x=160, y=200)

welcome_chatbot = Label(frame_welcome, text='I am JARVIS ! ',
                    font="Vardana 15 bold italic", bg="#7B68EE", fg="black")
welcome_chatbot.place(x=210, y=270)

pic_1 = Label(frame_welcome, image=A)
pic_1.place(x=-2, y=357)
Button(root, image=b, bg="#7B68EE", relief="solid",
       bd="3px solid black", activebackground="#7B68EE", command=chat).place(x=270, y=310)


# conver(takeCommand())

wishme()
root.mainloop()
speak("bye master.... have a nice day....")
```

conver.py

```
 import pyttsx3
import webbrowser
import random
import datetime
import speech_recognition as sr
import webbrowser
import wikipedia
from lispk import speak
from lispk import takeCommand
import os
import wmi
import smtplib
from datetime import date
import subprocess
import os
import ctypes
import random


def return_value(ans):
    return ans


def wishme():
    hour = int(datetime.datetime.now().hour)
    if hour >= 0 and hour < 12:
        speak("Good Morning")
    elif hour >= 12 and hour <= 16:
        speak("Good Afternoon")
    else:
        speak("Good Evening")
    speak("I am Jarvis, how may i help you.")


def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('vpnvairohi@gmail.com', 'vchampkanayya')
    server.sendmail("vaishurohi2001@gmail.com", to, content)
    server.close()


def conver(query):
    query = query.lower()
    print(query)
    if query == 'hello' or query == 'hi':
        answer = "hi am jarvis"
        speak(answer)
        return answer

    elif 'your name' in query or 'who are you' in query:
        answer = "My name is Jarvis, at your service"
```

```
        speak(answer)
        return answer

    elif 'whats up' in query or 'how are you' in query:
        answer = "I am fine  thankyou"
        speak(answer)
        return answer

    elif 'something about you' in query:
        answer = "In the Marvel Cinematic Universe  i was an AI developed by Tony Stark first primarily as being and artificial assistant in his house, and later modified to perform a lot of functions in the Mark 2 and now presently implemented by my group"
        speak(answer)
        return answer

    elif 'how will be my day' in query:
        answer = "wish that it will be good"
        speak(answer)
        return answer

    elif 'hangout jarvis' in query:
        answer = "Sorry  i would love to but i cant"
        speak(answer)
        return answer

    elif 'who am i jarvis' in query:
        answer = "you are my master"
        speak(answer)
        return answer

    elif 'can you be my friend' in query:
        answer = "Its my pleasure to be"
        speak(answer)
        return answer

    elif 'are you tired' in query:
        answer = "no master full of energy...."
        speak(answer)
        return answer

    elif 'goodbye jarvis' in query:
        answer = "goodbye have a nice day....."
        speak(answer)
        return answer

    elif 'hod of cs department' in query or 'H O D of cs department' in query or 'vinay bhat' in query:
        answer = "vinay bhat is the H O D of cs department"
        speak(answer)
        return answer

    elif 'about gpt karwar' in query:
        answer = "government polytechnic college is a polytechnic college in uttar kannada district of karnatka,india situated in karwar "
        speak(answer)
        return answer
```

```
    elif 'mechanical department hod' in query or 'hod of mechanical department' in query:
        answer = "shantaram is the H O D of mechanical department"
        speak(answer)
        return answer

    elif 'branches in gpt karwar' in query:
        answer = "there are seven branches in gpt karwar they are computer  science and  engineering,  civil engineering, mechanical engineering, automobile engineering, commercial pratice and electrical engineering"
        speak(answer)
        return answer

    elif 'sayyan shaikh' in query or 'saiyan shaikh' in query or 'saiyan sheikh'in query:
        answer = "saiyan shaikh is a lecturer of computer science department"
        speak(answer)
        return answer

    elif 'rashmi vernekar' in query:
        answer = "rashmi vernekar is a lecturer of computer science department"
        speak(answer)
        return answer

    elif 'subjects in computer science department?' in query:
        answer = "there are six subjects in first and second semester, eight subjects in third fourth and fifth semester, seven subjects in sixth semester"
        speak(answer)
        return answer

    elif 'subjects are there in first semester?' in query:
        answer = "Engineering mathematics one applied science,concepts of electrical and electronics engineering,applied science lab,basic electronic lab,basic computer skills lab"
        speak(answer)
        return answer

    elif 'subjects are there in second semester?' in query:
        answer = "Engineering mathematics two,Communication skills in English,Digital computer fundamentals,Multimedia lab,Basic web designing lab,Digital electronics lab"
        speak(answer)
        return answer

    elif 'subjects are there in third semester?' in query:
        answer = "Programming with c,Computer Organization,Database Management System ,computer network,network administration lab,dbms and GUI lab,Programming with c lab"
        speak(answer)
        return answer

    elif 'subjects are there in fourth semester?' in query:
        answer = "Data structure,Programming with Java,Operating system,Professional ethics and Indian Constitution,Tantrika Kannada and Kannada Kali,OOP with Java lab,Linux lab,Data structure lab"
        speak(answer)
        return answer

    elif 'subjects are there in fifth semester?' in query:
        answer = "Software engineering,Web programming,Design and analysis of algorithm,Green computing,Web programming lab,Design and analysis of algorithm lab,Professional practices,Project work one"
```

```
        speak(answer)
        return answer

    elif 'subjects are there in sixth semester?' in query:
        answer = "Software testing,Network security,Internet of things,Software testing lab,Implant
training,Project work two"
        speak(answer)
        return answer

    elif '.com' in query:
        os.startfile(
            "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe")
        webbrowser.open('http://www.'+query)
        answer = f'opening  '+query
        return answer

    elif 'open cmd' in query or 'open command prompt' in query:
        os.system("start cmd")
        answer = "opening command prompt"
        speak(answer)
        return answer

    elif 'camera' in query or 'open camera' in query:
        answer = "opening camera"
        speak("opening camera")
        subprocess.run('start microsoft.windows.camera:', shell=True)
        return answer

    elif 'date' in query or 'what is todays date' in query or 'whats the date today' in query:
        today = date.today()
        answer = f"Today's date is :, {today}"
        speak(answer)
        return answer

    elif 'wallpaper' in query or 'change wallpaper' in query:
        dir = "C:\\Users\\Vaishnavi\\Pictures\\wlpr\\"
        file1 = random.choice(os.listdir(dir))
        wpp1 = ('%s\%s' % (dir, file1))
        ctypes.windll.user32.SystemParametersInfoW(20, 0, r"%s " % wpp1, 0)
        answer = "your wallpaper changed"
        speak(answer)
        return(answer)

    elif 'create folder' in query:
        path = os.getcwd()
        speak("enter the folder name to create ")
        name = takeCommand()
        dir = f"{path}\\{name}"
        os.mkdir(dir)
        answer = "Folder %s created successfully" % name
        speak(answer)
        return(answer)

    elif 'find location' in query:
        speak("What location you want to find")
        print("What location you want to find")
```

```
        location = takeCommand()
        url = 'https://google.nl/maps/place/' + location + '/&amp;amp;'
        webbrowser.get().open(url)
        answer = f'Here are the result of location '+location
        speak(answer)
        return answer

    elif 'hibernate' in query:
        speak("good night master")
        answer = "good night master"
        os.system('rundll32.exe powrprof.dll,SetSuspendState 0,1,0')
        return answer

    elif 'open explorer' in query:
        answer = "opening  master....."
        speak(answer)
        print("opening  master.....")
        os.system("explorer.exe")
        return answer

    elif 'brightness' in query:
        speak("please value to increase or decrease brightness")
        brightness = takeCommand()  # percentage [0-100]
        print(brightness)
        c = wmi.WMI(namespace='wmi')

        methods = c.WmiMonitorBrightnessMethods()[0]
        methods.WmiSetBrightness(brightness, 0)
        answer = "adjusting brightness...."
        return answer

    elif 'shutdown' in query:
        answer = "Do you want to shutdown your system? (yes/no)"
        speak(answer)
        a = takeCommand()
        if 'yes' in a:
            os.system("shutdown /s /t 1")
            shut = "shuttind down...."
            return(shut)
        elif 'no' in a:
            exit()
        return answer

    elif 'restart' in query:
        answer = "Do you want to shut down your system? (yes/no)"
        speak(answer)
        a = takeCommand()
        if 'yes' in a:
            os.system("shutdown /r /t 1")
            rest = "restarting pc"
            return(rest)
        elif 'no' in a:
            exit()
        return(answer)

    elif 'wikipedia' in query:
```

```
        speak('Searching Wikipedia...')
        query = query.replace("wikipedia", "")
        answer = wikipedia.summary(query, sentences=2)
        speak("According to Wikipedia")
        # print(results)
        speak(answer)
        return answer

    elif 'music' in query:
        music_dir = 'C:\\Users\\Vaishnavi\\Music'
        songs = os.listdir(music_dir)
        os.startfile(os.path.join(music_dir, songs[0]))
        answer = "Ya sure starting....."
        speak(answer)
        return(answer)

    elif 'open gmail' in query:
        answer = "okay right on the way"
        webbrowser.open('www.gmail.com')
        speak(answer)
        return(answer)

    elif 'send' in query:
        try:
            speak("what should i send")
            print("what should i send")
            content = takeCommand()
            to = 'anupadti@gmail.com'
            sendEmail(to, content)
            speak("email sent master")
            print("email sent master")
        except Exception as e:
            print(e)

    elif 'notepad' in query or 'open notepad' in query:
        answer = "opening notepad..."
        return_value(answer)
        speak(answer)
        os.system("notepad.exe")
        return(answer)

    elif 'open recent activities' in query or 'recent activities' in query:
        answer = "letting you to recent activities....."
        os.startfile("C:\\Users\\Vaishnavi\\Recent")
        speak(answer)
        return(answer)

    elif 'firefox' in query or 'open firefox' in query:
        answer = "opening firefox..."
        os.startfile('C:\\Program Files (x86)\\Mozilla Firefox\\firefox.exe')
        speak(answer)
        return(answer)

    elif 'chrome' in query or 'open chrome' in query:
        answer = "opening google chrome"
        os.startfile(
```

```
            'C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe')
        speak(answer)
        return(answer)

    elif 'adobe photoshop' in query or 'open adobe photoshop' in query:
        answer = "opening adobe photoshop"
        os.startfile(
            'C:\\Program Files (x86)\\Adobe\\Photoshop 7.0\\Photoshop.exe')
        speak(answer)
        return(answer)

    elif 'brackets' in query or 'open brackets' in query:
        answer = "opening brackets"
        os.startfile('C:\\Program Files (x86)\\Brackets\\Brackets.exe')
        speak(answer)
        return(answer)

    elif 'wamp' in query or 'open wamp64' in query:
        answer = "opening wampserver"
        speak(answer)
        os.startfile(
            'C:\\ProgramData\\Microsoft\\Windows\\Start
Menu\\Programs\\Wampserver64\\Wampserver64.lnk')
        return(answer)

    elif 'vmware' in query or 'open vmware' in query:
        answer = "opening vmware"
        speak(answer)
        os.startfile(
            'C:\\Program Files (x86)\\VMware\\VMware Player\\vmplayer.exe')
        return(answer)

    elif 'character map' in query or 'open cmap' in query:
        answer = "opening charctermap"
        speak(answer)
        os.startfile("charmap.exe")
        return(answer)

    elif 'paint' in query or 'open paint' in query or 'open paint' in query:
        answer = "opening paint"
        speak(answer)
        os.startfile("mspaint.exe")
        return(answer)

    elif 'steps recorder' in query or 'open steps recorder' in query:
        answer = "opening step recorder"
        speak(answer)
        os.startfile("psr.exe")
        return(answer)

    elif 'player' in query or 'open wmplayer' in query:
        answer = "opening windows media player"
        speak(answer)
        os.startfile('wmplayer.exe')
        return(answer)
```

```
elif 'wordpad' in query or 'open wordpad ' in query:
    answer = "opening wordpad"
    speak(answer)
    os.startfile('wordpad.exe')
    return(answer)

elif 'quick assist' in query or 'open quick assist' in query:
    answer = "opening quickassist"
    speak(answer)
    os.startfile('quickassist.exe')
    return(answer)

elif 'component services' in query or 'open component services' in query:
    answer = "opening component service"
    speak(answer)
    os.system("comexp.msc")
    return(answer)

elif 'scheduler' in query or 'open task scheduler' in query:
    answer = "opening task scheduler"
    speak(answer)
    os.startfile("taskschd.msc ")
    return(answer)

elif 'component management' in query or 'open component management' in query:
    answer = "opening component management"
    speak(answer)
    os.startfile("compmgmt.msc")
    return(answer)

elif 'system information' in query or 'open system information' in query:
    answer = "opening system information"
    speak(answer)
    os.startfile("msinfo32.exe")
    return(answer)

elif 'firewall' in query or 'open windows defender firewall' in query:
    answer = "opening firewall settings"
    speak(answer)
    os.startfile("WF.msc")
    return(answer)

elif 'calculator' in query or 'open calcy' in query:
    answer = "opening calculator"
    speak(answer)
    os.startfile("C:\\Windows\\System32\\calc.exe")
    return(answer)

elif 'tell me about system performance' in query or 'open performance monitor' in query:
    answer = "on the way to monitor performance"
    speak(answer)
    os.startfile("perfmon.msc")
    return(answer)

elif 'control panel' in query or 'open control panel' in query:
    answer = "opening control panel"
```

```
        speak(answer)
        os.system("Control.exe")
        return(answer)

    elif 'services' in query or 'services of my system' in query:
        answer = "opening services"
        speak(answer)
        os.startfile("services.msc")
        return(answer)

    elif 'time' in query:
        answer = (datetime.datetime.now().strftime("%H:%M:%S"))
        speak(answer)
        return(answer)

    elif 'Microsoft word' in query or 'word' in query:
        answer = "Opening Microsoft Word"
        os.startfile(
            'C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Microsoft Office\\Microsoft Office
Word 2007.lnk')
        speak(answer)
        return answer

    elif 'Microsoft access' in query or 'access' in query:
        answer = "Opening Microsoft Access"
        os.startfile(
            'C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Microsoft Office\\Microsoft Office
Access 2007.lnk')
        speak(answer)
        return(answer)

    elif 'Microsoft powerpoint' in query or 'powerpoint' in query:
        answer = "Opening Microsoft Powerpoint"
        os.startfile(
            'C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Microsoft Office\\Microsoft Office
PowerPoint 2007.lnk')
        speak(answer)
        return(answer)

    elif 'Microsoft excel' in query or 'excel' in query:
        answer = "Opening Microsoft Excel"
        os.startfile(
            'C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Microsoft Office\\Microsoft Office
Excel 2007.lnk')
        speak(answer)
        return(answer)

    elif 'adobe' in query:
        os.startfile(
            "C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Acrobat Reader DC.lnk")
        answer = "opening adobe acrobat reader dc...."
        speak(answer)
        return(answer)

    elif 'exit' in query or 'stop' in query:
        ans = 'good bye master'
```

```
        speak(ans)
        exit()
        return ans

    else:
        ans = "sorry i didn't get that     say that again     or try typing the command    "
        speak(ans)
        return ans
```

lispk.py

```
import pyttsx3
import datetime
import speech_recognition as sr
import webbrowser
import wikipedia


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


def speak(audio):
    engine.say(audio)
    engine.runAndWait()


def takeCommand():

    r = sr.Recognizer()
    with sr.Microphone() as source:

        print("listening...")
        r.pause_threshold = 0.5
        r.energy_threshold = 3000
        audio = r.listen(source)

    try:

        print("recognizing...")
        query = r.recognize_google(audio, language='en-in')
        print(f'user said: {query}\n')

    # except Exception as e:
    #     print(e)
    #     print("say That again please")
    #     return "None"
 except sr.UnknownValueError:
        speak('Sorry master! I didn\'t get that!Please say it again')
        return takeCommand()
  return query
```

CHAPTER: 7

SOFTWARE TESTING

Software testing can be stated as the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by it's design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy and usability. It mainly aims at measuring specification, functionality and performance of a software program or application.

Software testing can be divided into two steps:

1. **Verification:** It refers to the set of tasks that ensure that software correctly implements a specific function.

2. **Validation:** It refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Verification: "Are we building the product right?"

Validation: "Are we building the right product?"

Software Testing can be broadly classified into two types:

1. **Manual Testing:** Manual testing includes testing a software manually, i.e., without using any automated tool or any script

2. **Automation Testing:** Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product.

Software techniques can be majorly classified into two categories:

1. **Black Box Testing:** The technique of testing in which the tester doesn't have access to the source code of the software and is conducted at the software interface without concerning with the internal logical structure of the software is known as black box testing.

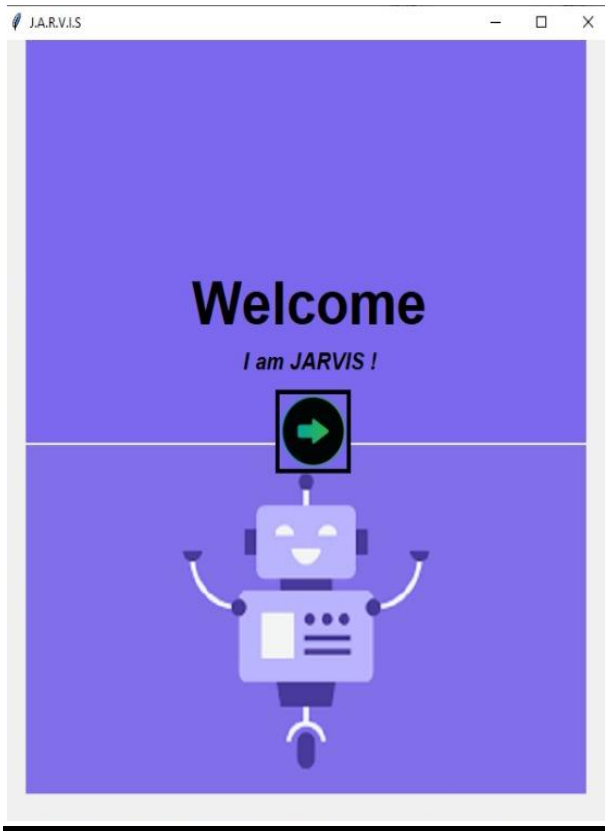
2. **White-Box Testing:** The technique of testing in which the tester is aware of the internal workings of the product, have access to it's source code and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

Different levels of software testing:

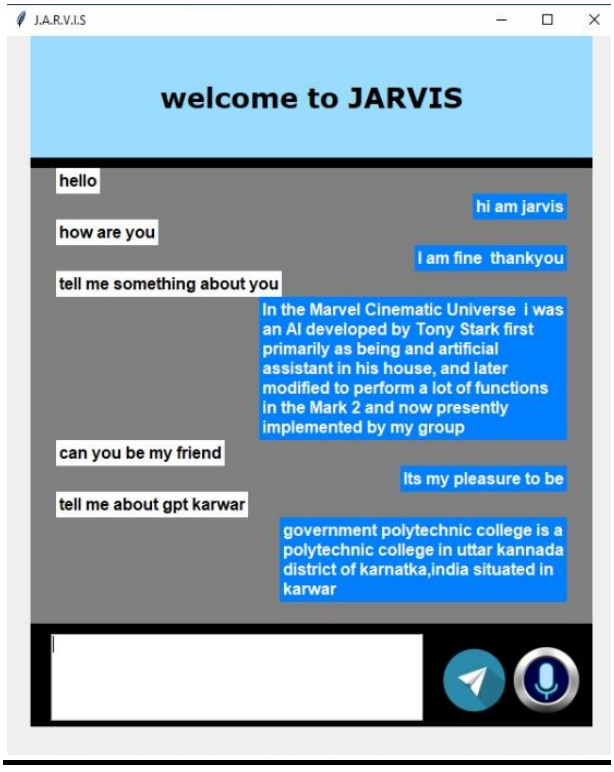
1. **Unit Testing:** A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.
2. **Integration Testing:** A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.
3. **System Testing:** A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.
4. **Acceptance Testing:** A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

CHAPTER: 8

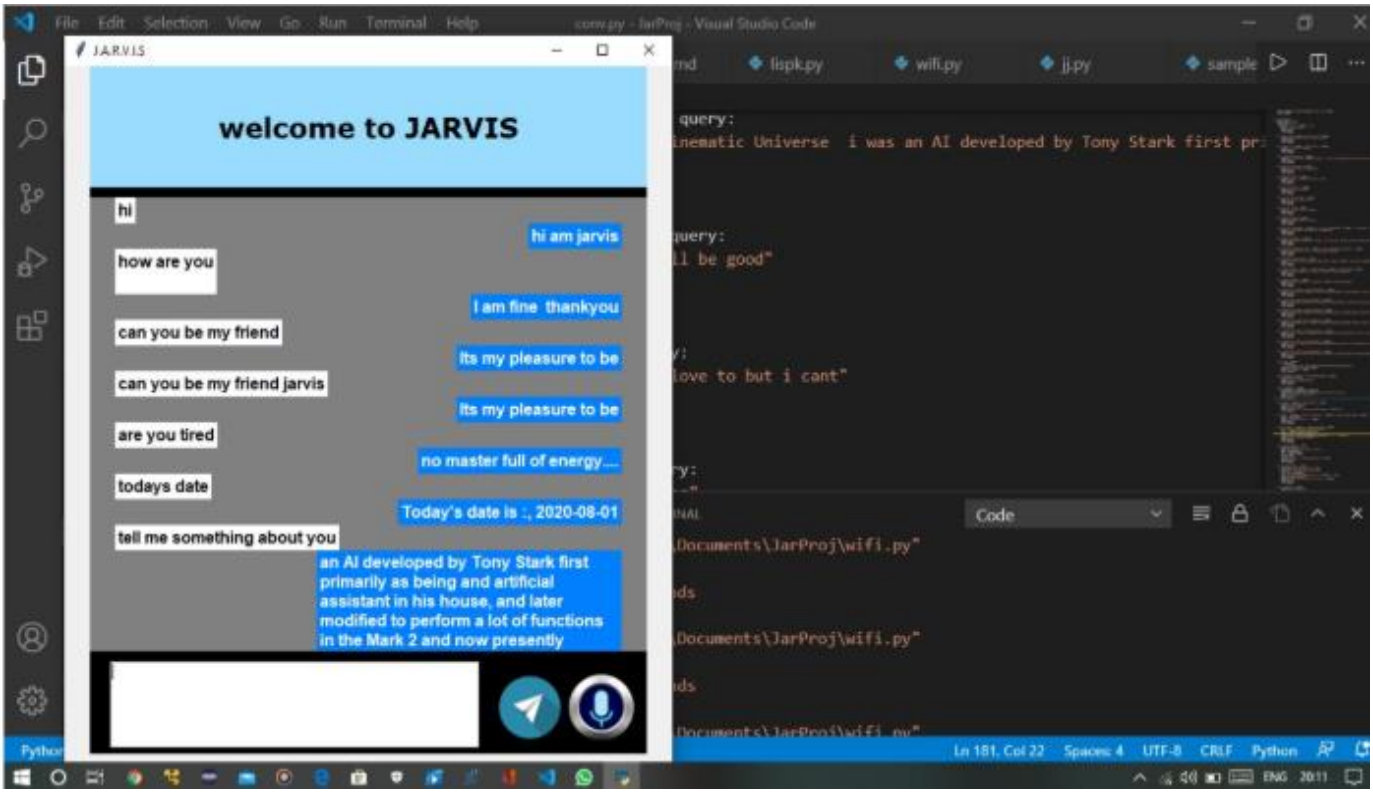
INPUT AND OUTPUT SCREENS



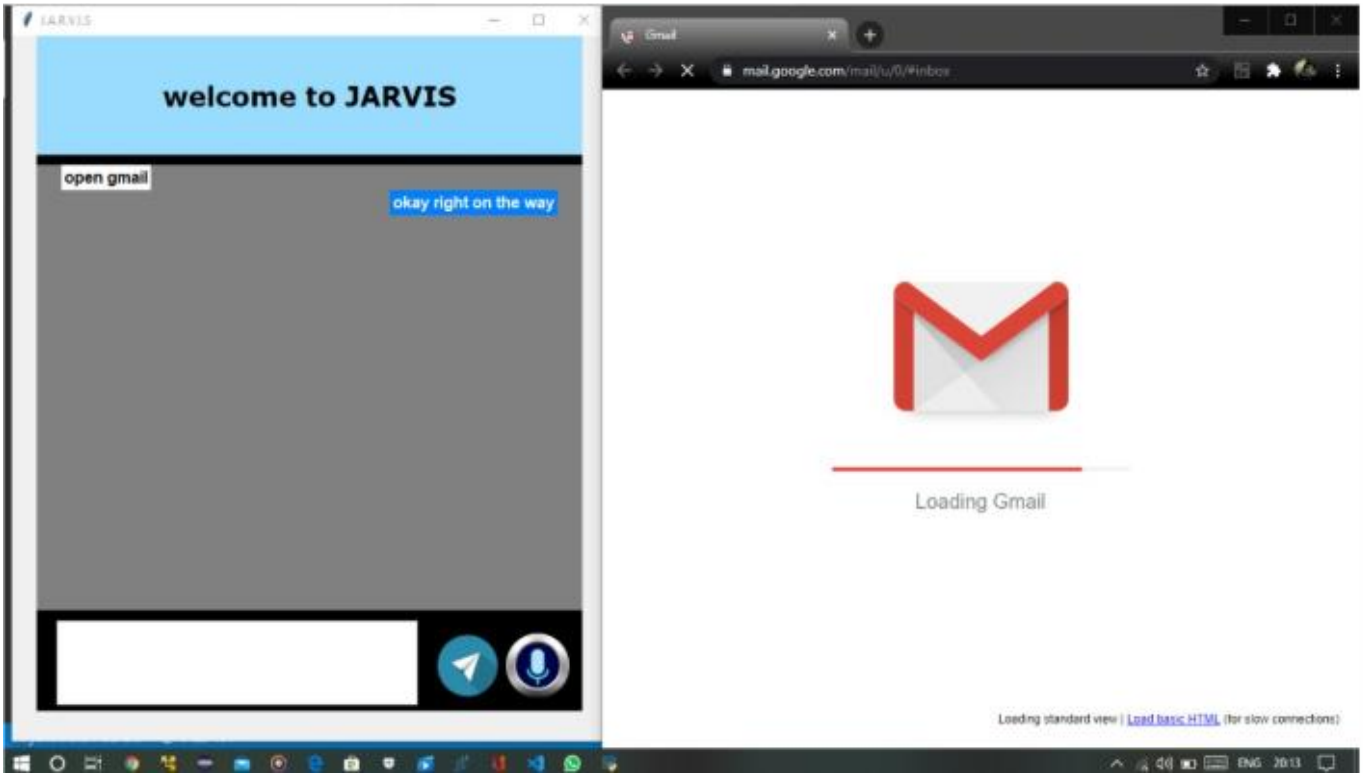
ChatFrame:



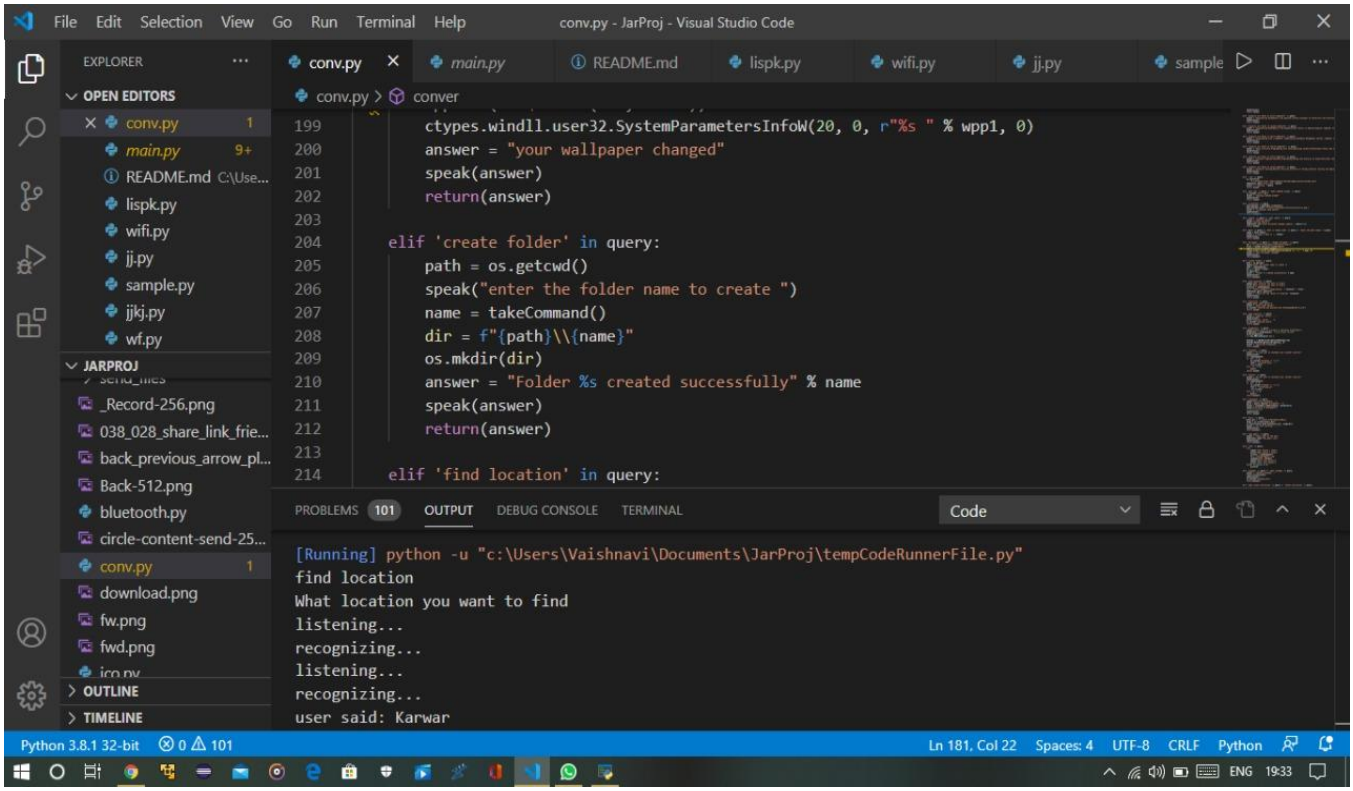
Conversation with JARVIS:

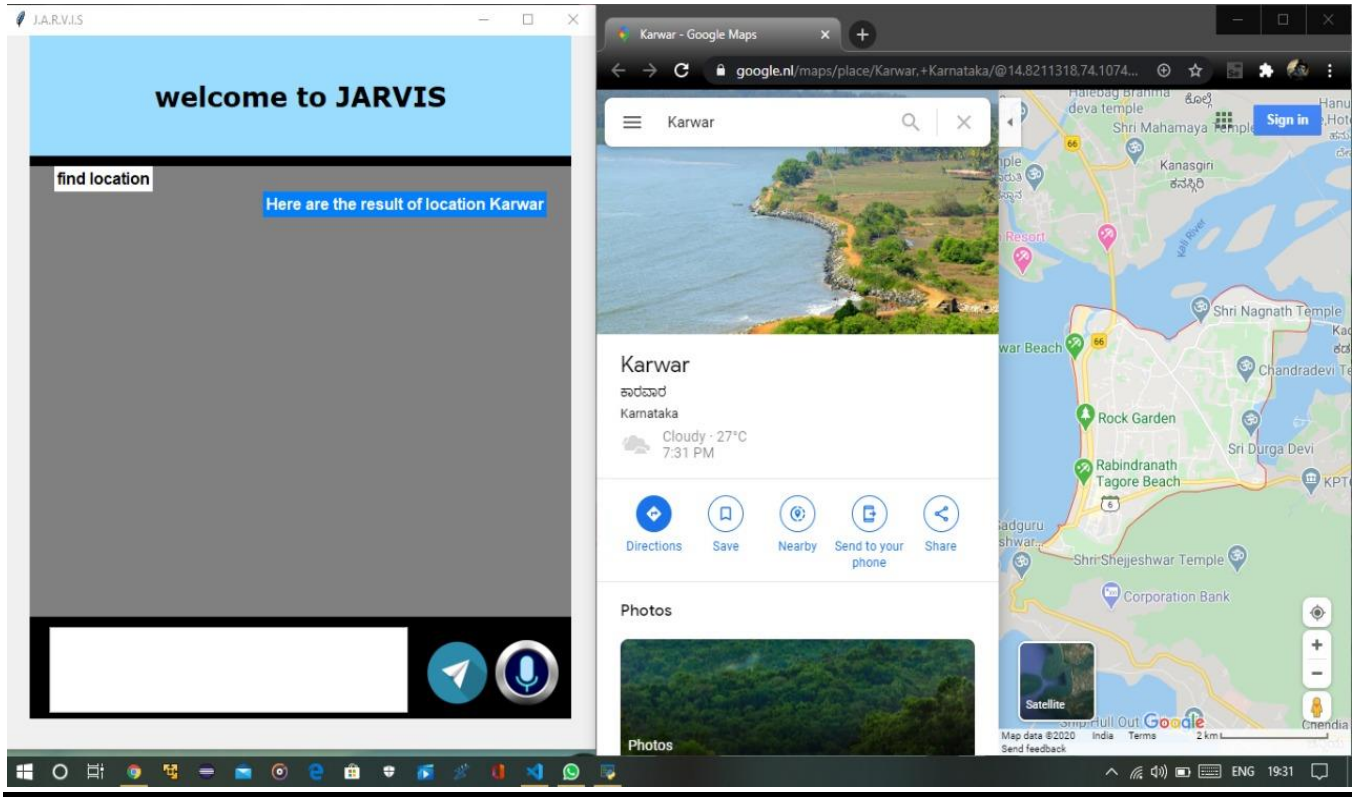


Open gmail:

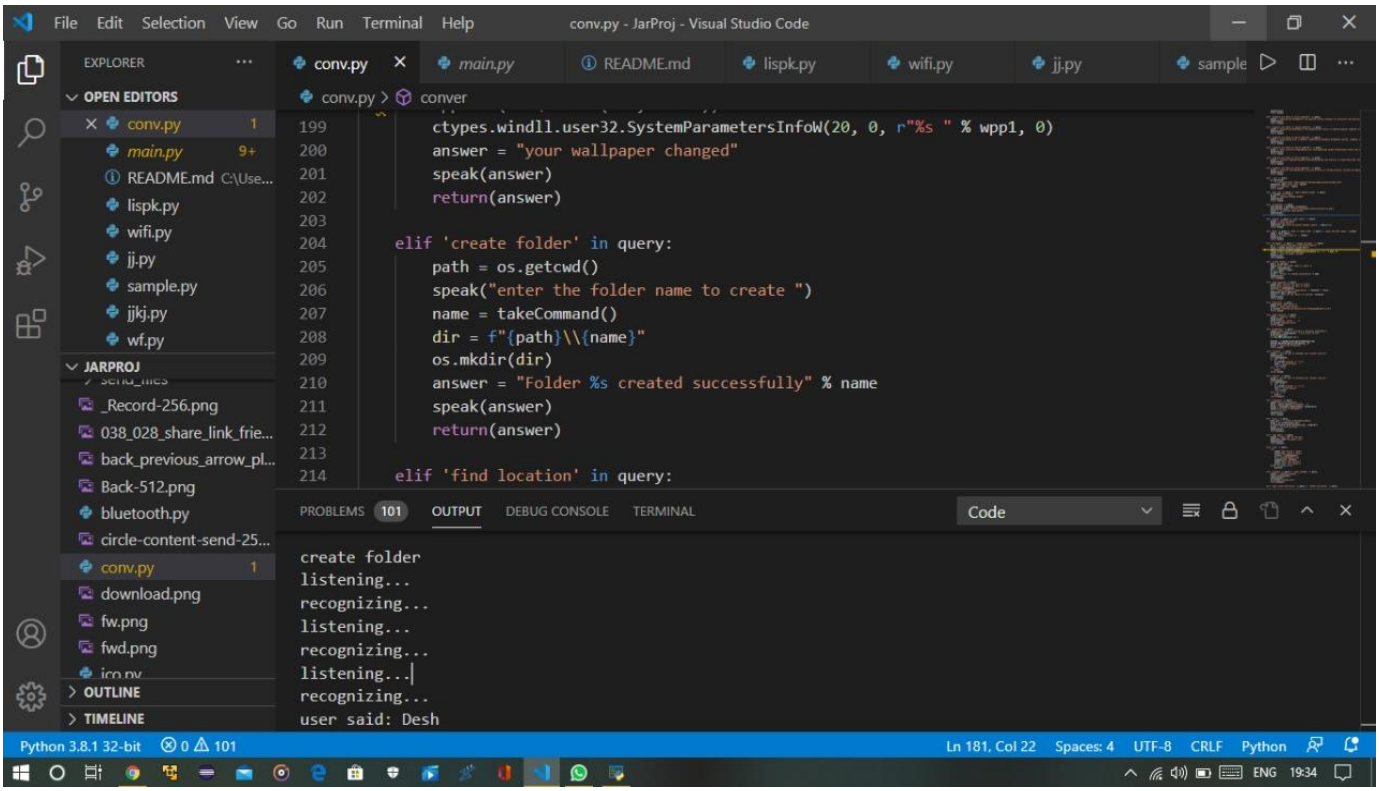


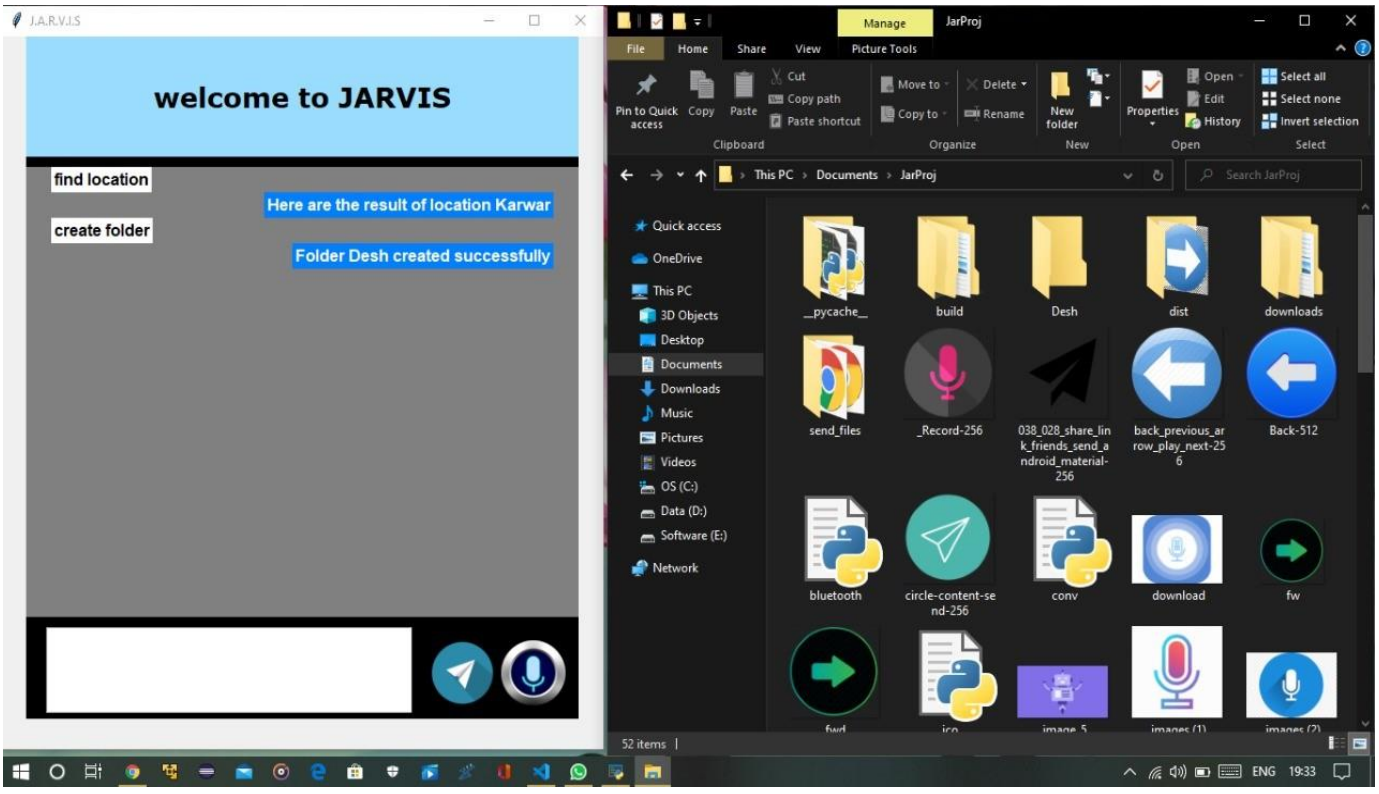
Find location:





Create Folder:





CHAPTER: 9

LIMITATIONS

- Since our software can work with a limited number of languages hence it is not language diversity friendly which makes it useless for a huge number of people who do not know the languages it can work with.

- Our software will not be able to perform well in very noisy places. Because of mixed sounds will be hard to process. You need to have a silent and calm environment in order to make JARVIS work.

CHAPTER: 10

CONCLUSION

This chapter gives an overview of the system that will be developed. The topics covered in this chapter include introduction of the current system, objectives of the proposed system, improvements that can be made in present existing system and scope of the system. This software almost saves good time same as existing system, which can be used for other useful work.

The proposed system gives a whole new way of interaction with a computer. This software changes the way of interacting with computer, without changing or affecting the existing system but with better interaction.

Finally after development of this software the power of keyboard and mouse will be in our voice and wave of our hand.

CHAPTER: 11

FUTURE APPLICATIONS

The team ran into difficulties defining the boundaries of what would be included in the project and what would not. This is because of the open-ended nature of the project. Many different ideas and pieces of functionality had potential to be incorporated into the application. The team wanted to be as open as possible when considering features, so it became difficult to say no to different parts of the system. This resulted in pieces of the system that were not developed as completely as they could have been.

Future projects should have a clearly defined scope that teams can commit to. The team believes that since the scope was so large, there was no chance of completing everything in the project, which hurt the overall motivation on the project. The new sneakers team would have benefit from a well defined scope from the beginning of the second semester. This would have increased the functionality of the completed system.

CHAPTER: 12

REFERENCES

1. docs.python.org
2. pypi.org
3. www.tutorialspoint.com
4. https://github.com/nihal111/J.A.R.V.I.S
5. https://www.codewithharry.com/videos/python-tutorials-for-absolute-beginners-120
6. techflayer.blogspot.com