TOKYO - THE VOICE ASSISTANT

PROJECT REPORT

Project submitted in partial fulfillment of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

MD ABDUL MUSAVVIR (O161764)
TALLAPAKA MANASA (O162041)
NOSSAM VIGNATHA (O161530)
ERUKALA ANUSHA (O161005)
KONAPURI SURESH (O161714)

Under the supervision of

Mr. N. Mallikarjuna

Dept. of Computer Science and Engineering ,
RGUKT Ongole Campus

April 2020-2021

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE
TECHNOLOGIES
DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is certify that the project report entitled “**Tokyo The Voice Assistant**” Submitted by **Md Abdul Musavvir**, Roll No : **O161764**, **Tallapaka Manasa** Roll No : **O162041**, **Nossam Vignatha**, Roll No : **O161530**, **Erukala Anusha** Roll No: **O161005**, **Konapuri Suresh**, Roll No : **O161714**, to the Department of Computer Science and Engineering, Rajiv Gandhi University of Knowledge Technologies , Ongole Campus, during the academic year 2020-2021 is a partial fulfillment for the award of Under graduate degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a bonafide record carried out by him under my supervision. The project has fulfilled all the requirements as per as regulations of this institute and in my opinion reached the standard for submission.

Mr. P. Ananth Rao,
Head Of The Department,
Dept of CSE,
RGUKT Ongole Campus,
Kadapa – 516329.

Mr. N. Mallikarjuna
Assistant Professor(C),
Dept of CSE,
RGUKT Ongole Campus,
Kadapa – 516329.

Date:
Place: Rk Valley.

ACKNOWLEDGEMENT

We take this opportunity to express my profound gratitude and deep regards to my teacher **Mr. N. Mallikarjuna** for their exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by them time to time shall carry me a long way in the journey of life on which I am about to embark.

We have equally taken efforts in this project. We are also grateful to our Head Of The Department, **Mr. P. Ananth Rao** sir, for their consistent support and assistance.

Finally, last but by no means least; also to everyone in the university it was great sharing premises with all of you during last five years.

Thanks for all your encouragement!

ABSTRACT

We have come a long way in technological advancement. In such an era of advancement if people are still struggling to interact with their machine using various input devices then its not worth it. For this reason, many voice assistants were developed and are still being improved for better performance and efficiency.

Voice control is a major growing feature that change the way people can live. The voice assistant is commonly being used in smart phones and laptops. AI based Voice Assistants are the operating systems that can recognize human voice and respond via integrated voice.Speech recognition is the process of converting audio into text. SpeechRecognition allow us to convert audio into text for further processing. In this article, we will look at converting large or long audio files into text using the SpeechRecognition API in python.

The Voice Assistant will gather the audio from the microphone and then convert that into text.Later, it is sent through GTTS(Google Text To Speech).GTTS Engine will convert text into audio file in english Language, then that audio is played using playsound packages of python programming Languages.

In this project we Created a Voice activated Personal Assistant developed using Python. This Assistant currently works online and performs basic tasks like Stream Music,Search Wikipedia,Open System installed Applications etc..The functionality of the current system is limited to working online only.

CONTENTS

1. INTRODUCTION:

[1]In the 21st century, human interaction is being replaced by automation very quickly. One of the main reasons for this change is performance. There's a drastic change in technology rather than advancement. In today's world, we train our machine's to do their tasks by themselves or to think like humans using technologies like Machine Learning, Neural Networks, etc. Now in the current era, we can talk to our machines with the help of virtual assistants. There are companies like Google, Apple, Microsoft, etc with virtual assistants like Google Now, Siri, Cortana, etc. which helps their users to control their machine by just giving input in the form of voice.These types of virtual assistants are very useful for old age, blind & physically challenged people, children, etc. by making sure that the interaction with the machine is not a challenge anymore for people. Even blind people who couldn't see the machine can interact with it using their voice only.

With the use of voice assistants, we can automate the task easily, just give the input to the machine in the speech form and all the asks will be done by it from converting your speech into text form to taking out keywords from that text and execute the query to give results to the user.

For building any voice based assistant you need two main functions. One for listening to your commands and another to respond to your commands. Along with these two core functions, you need the customized instructions that you will feed your assistant.

The first step is to install and import all the necessary libraries. Use pip3[i] install to install the libraries before importing them. Following are some of the key libraries used in this program:

- The SpeechRecognition library allows Python to access audio from your system's microphone, transcribe the audio, and save it.
- Google's text-to-speech package, gTTS[ii] converts your audio questions to text. The response from the look-up function that you write for fetching answer to the question
- Speech is converted to an audio phrase by gTTS[ii]. This package interfaces with Google Translate's API[iii].

- Playsound package is used to give voice to the answer. Playsound allows Python to play MP3[iv] files.
- Web browser package provides a high-level interface that allows displaying Web-based pages to users. Subprocess is another option for displaying web pages. However, for using this you need to install and provide the browser-specific web driver.
- Wikipedia is used to fetch a variety of information from the Wikipedia website.

1.1 Project Aim and Objective :-

The Objective of our project is to create a personal voice assistant using python.This Assistant currently works online and performs basic tasks like Stream Music,Search Wikipedia,Open System installed Applications etc..The functionality of the current system is limited to working online only. The Voice Assistant will gather the audio from the microphone and then convert that into text. Later, it is sent through GTTS(Google Text To Speech).GTTS[ii] Engine will convert text into audio file in english Language, then that audio is played using playsound packages of python programming Languages.

1.2 Motivation:

The main motivation of this project is to learn the python programming language. There are companies like Google, Apple, Microsoft, etc with virtual assistants like Google Now, Siri, Cortana, etc. which helps their users to control their machine by just giving input in the form of voice.So we created a personal voice assistant which works online and performs basic tasks like serach wikipedia,opensystem installed applications,stream music etc..With the use of voice assistants, we can automate the task easily.

2. LITERATURE SURVEY:

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

With the use of voice assistants, we can automate the task easily, just give the input to the machine in the speech form and all the tasks will be done by it from converting your speech into text form to taking out keywords from that text and execute the query to give results to the user.

2.1 Existing System(Google Now, Siri, Alexa):

- For Global usage
- Works only for register users
- It consumes more network data
- Maintenance cost is more
- Doesn’t work on battery
- Voice recognition uses a lot of memory

2.2 Proposed System(Tokyo: The Voice Assistant):

- For Local usage within the organization or Institution.
- List of problems related to specific class .
- Features supported in current version include Opening system installed applications, opening anything on the web browser, search on Wikipedia, Play Music, Complaint Bot System.
- General Information about the University such as Library timings, Time table, Care Takers of Dormitories and etc..

3.SYSTEM ANALYSIS:

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. It involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore requirements analysis means to analyze, document, validate and manage software or system requirements.

High-quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to a facilitate system design.

3.1 Requirement Specification:

3.1.1 Functional Requirements:

- Linux Distribution
- Proper Internet Connection
- Python 2.7 and above
- Google Text-to-Speech(gTTS)
- Web browsers like Mozilla Firefox, Chrome
- Receive audio Stream
- Speech Recognition
- Language(Text) Processing
- UI[v] and Voice Output

3.1.2 Non-Functional Requirements

Performance Requirements

- System should recognize to any voices in his list without any fault.
- With Ideal condition, system response should be fast and error-free.
- System performance shall not decrease with time or by usage.

Security Requirements:

- Change the data is only allowed to admins and forbidden to any user.
- Program run without web, that is mean Protected from hackers.

3.1.3 Hardware Requirements

- ROM: 120GB or higher.
- Processor: Intel CoreTM i3 2.53 GHz / Above or AMD Multi-core Processor with single Thread.
- Graphics: Intel/Nviia/ATI.
- RAM: 2GB or higher.

3.1.4 Software requirements

- Operating system : Linux and Unix.
- Developing Environment : Terminal.

4.SYSTEM ENVIRONMENT:

4.1 Python:

[8]Python is a simple, general purpose, high level, and object-oriented programming language. Python is an interpreted scripting language also. Guido Van Rossum is known as the founder of Python programming. Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles. Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

Python is a general-purpose, popular programming language and it is used in almost every technical field. The various areas of Python use are given below.

- Desktop Applications
- Mobile Applications
- Software Development
- Artificial Intelligence
- Web Applications
- Image Processing Applications
- Speech Recognitions

4.2 System Modules:

4.2.1 Speech Recognition:

The []speech recognition module used the program is Google's Speech Recognition API[iii] which is imported in python using the command "import speech_recognition as sr".This module is used to recognize the voice which is given as input by the user. This is a free API[iii] that is provided and supported by Google. This is a very light API[iii] that helps in reducing the size of our application.

4.2.2 WebBrowser:

[6]This module allows the system to display web-based information to users. For example, the user wants to open any website and he gives input as "Open Google". The input is processed using the web browser module and the user gets a browser with google opened in it. The browser which will be used is the default set web browser.

4.2.3 GTTS:

The Google Text to Speech API[iii] commonly known as the gTTS API. gTTS is a very easy to use tool which converts the text entered, into audio which can be saved as a mp3 file.The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. The speech can be delivered in any one of the two available audio speeds, fast or slow. However, as of the latest update, it is not possible to change the voice of the generated audio.

4.2.4 Wikipedia:

Wikipedia is a library in python which it possible for the virtual assistant to process the queries regarding Wikipedia and display the results to users. This is an online library and needs an internet connection to fetch the results. The number of lines that the user wants to get as a result can be set manually.

4.2.5 Date Time:

The DateTime module is imported to support the functionality of the date and time. For example, the user wants to know the current date and time or the user wants to schedule a task at a certain time. In short this module supports classes to manipulate date and time and perform operations according to it only. This is an essential module, especially in tasks where we want to keep a track of time. This module is very small in size and helps to control the size of our program.

4.2.6 Os Module:

OS[vi] Module provides an operating system dependent functionalities. If we want to perform operations on files like reading, writing, or manipulate paths, all these types of functionalities are available in an OS module. All the operations available raise an error “OSError” in case of any error like invalid names, paths, or arguments which may be incorrect or correct but just no accepted by the operating system.

4.2.7 Play Sound:

The playsound module contains only one thing - the function (also named) playsound. It requires one argument - the path to the file with the sound you’d like to play.This may be a local file, or a URL[vii]. There’s an optional second argument, block, which is set to True by default. Setting it to False makes the function run asynchronously.

4.2.8 Face Recognition:

[3]Recognize and manipulate faces from Python or from the command line with the world’s simplest face recognition library. Built using dlib’s[viii] state-of-the-art face recognition built with deep learning. This also provides a simple face_recognition command line tool that lets you do face recognition on a folder of images from the command line.

4.2.9 CV2:

[5]OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

4.2.10 Json:

JSON stands for JavaScript Object Notation. JSON is a lightweight format for storing and transporting data JSON is often used when data is sent from a server to a web page JSON is "self-describing" and easy to understand.

4.2.11 Numpy:

[4]NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation.

5. SYSTEM DESIGN:

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

5.1 Program Design:

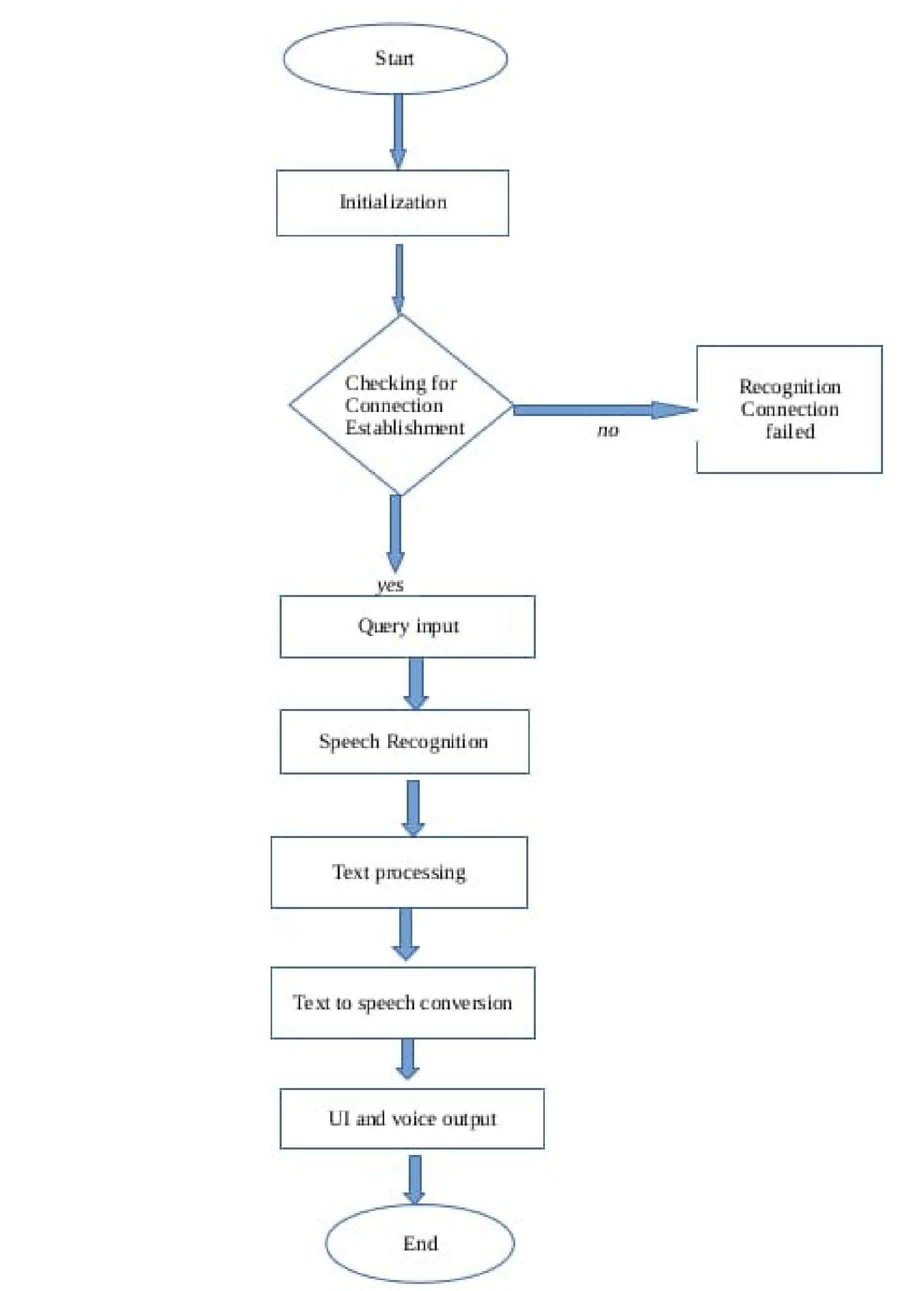

Start
Initialization
Checking for Connection Establishment
no
Recognition Connection failed
yes
Query input
Speech Recognition
Text processing
Text to speech conversion
UI and voice output
End

Figure 5.1 Algorithm Diagram

5.2 Data Flow Diagram:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.It shows how data enters and leaves the system, what changes the information, and where data is stored.The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

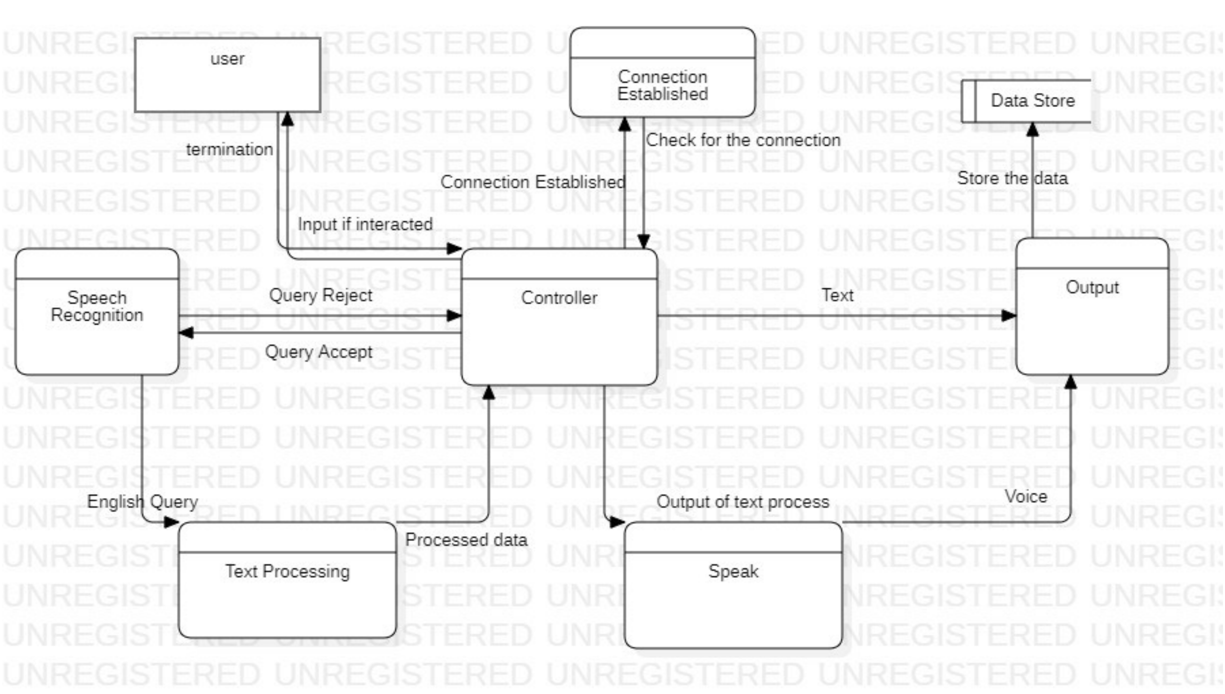


Figure 5.2 Data Flow Diagram

5.3 UML Diagrams:

Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML[ix] is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language, it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. Its been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

Types Of Uml Diagrams :

There are Several types of UML diagrams and each one of them serves a different purpose regardless of whether it is being designed before the implementation or after(as part of documentation).

The two most board categories that encompass all other types are **Behavioral.** UML diagram and **Structural** UML diagram.As the name suggests, some UML diagrams try to analyze and depict the structure of a system or process, whereas other describe the behaviour of the system, its actors, and its building components. The different types are broken down as follows:

Behavioral UML Diagram

- Activity Diagram
- Use Case Diagram
- Communication Diagram
- Sequence Diagram

Structural UML Diagram

- Class Diagram
- Object Diagram

- Component Diagram
- Deployment Diagram

5.3.1 Use Case 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.

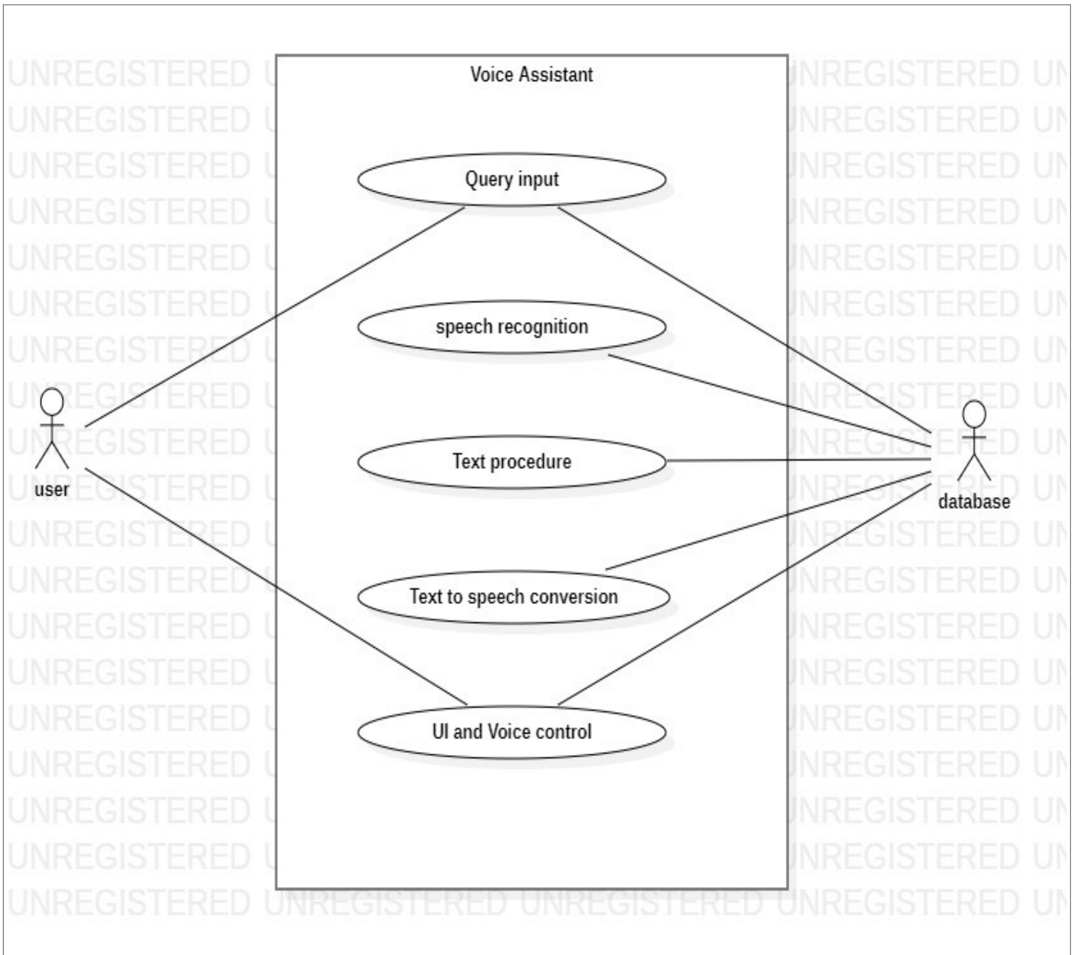


Figure 5.3.1: Use Case Diagram

5.3.2 Sequence Diagram:

A sequence diagram is a model that describe how groups of objects collobrate in some behaviour over time and capturing the behaviour of a single use case. It shows the interactions between the objects in terms of messages exchanged over time. Sequence diagrams are a popular dynamic modeling solution in UML[ix] because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

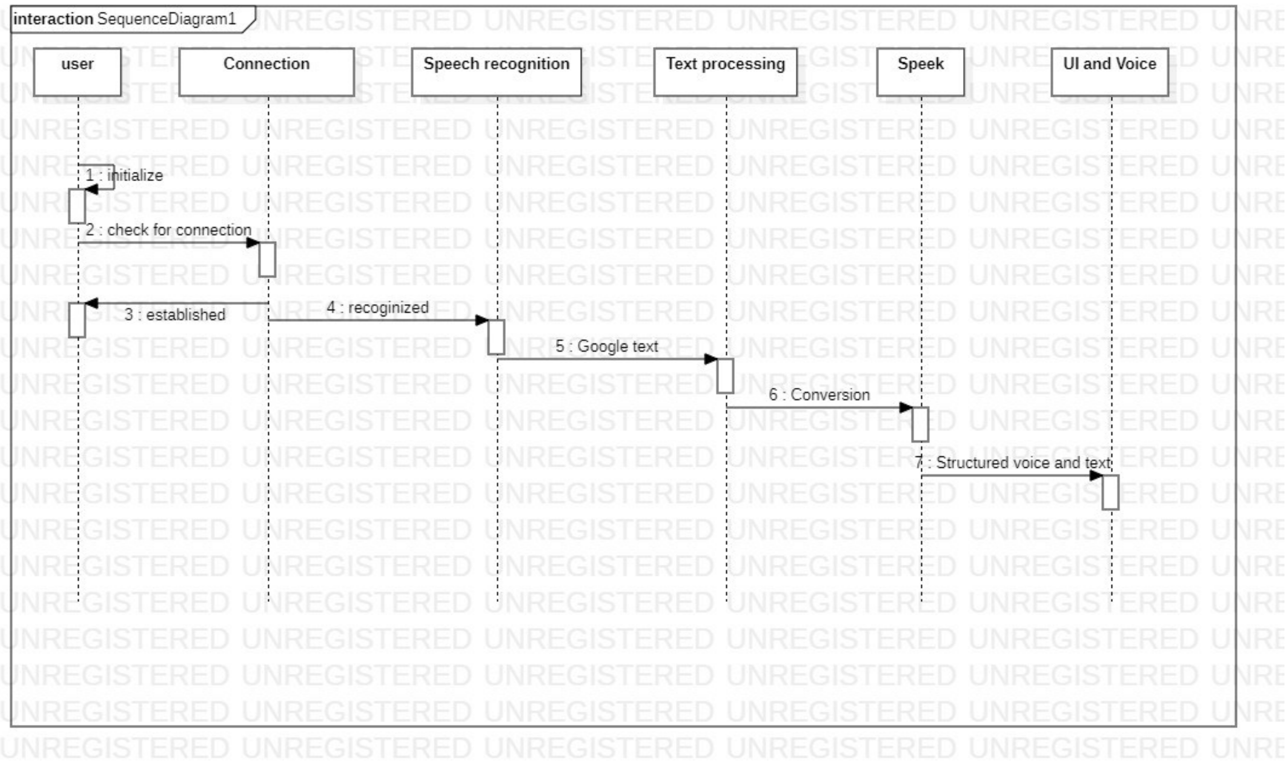


Figure 5.3.2 Sequence Diagram

5.3.3 Activity Diagram:

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system.

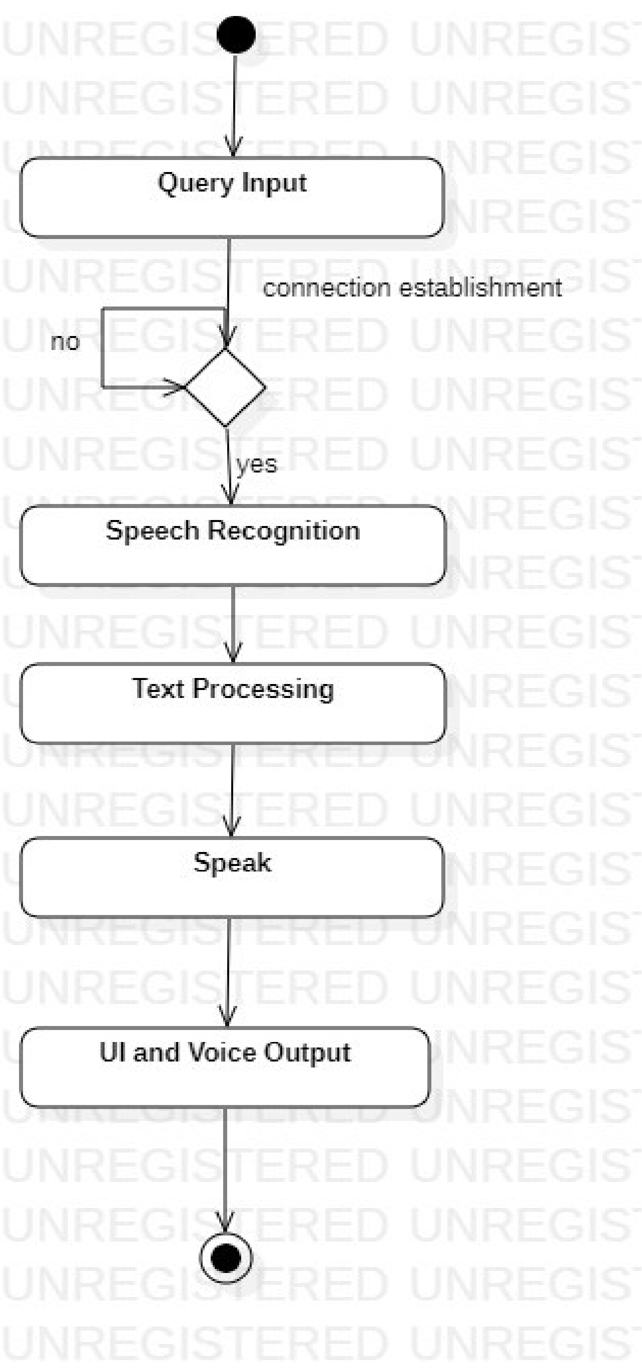


Figure 5.3.3: Activity Diagram

5.3.4 Communication Diagram:

A communication diagram is made up of three parts: objects (also called participants), the communication links, and the messages that can be passed along those links. Communication diagrams show the same information as a sequence diagram but may be more difficult to read. In order to show time ordering, you must indicate a sequence number and describe the message. A communication diagram emphasizes the organization of objects, whereas a sequence diagram emphasizes the time ordering of messages. A communication diagram will show a path to indicate how one object is linked to another.

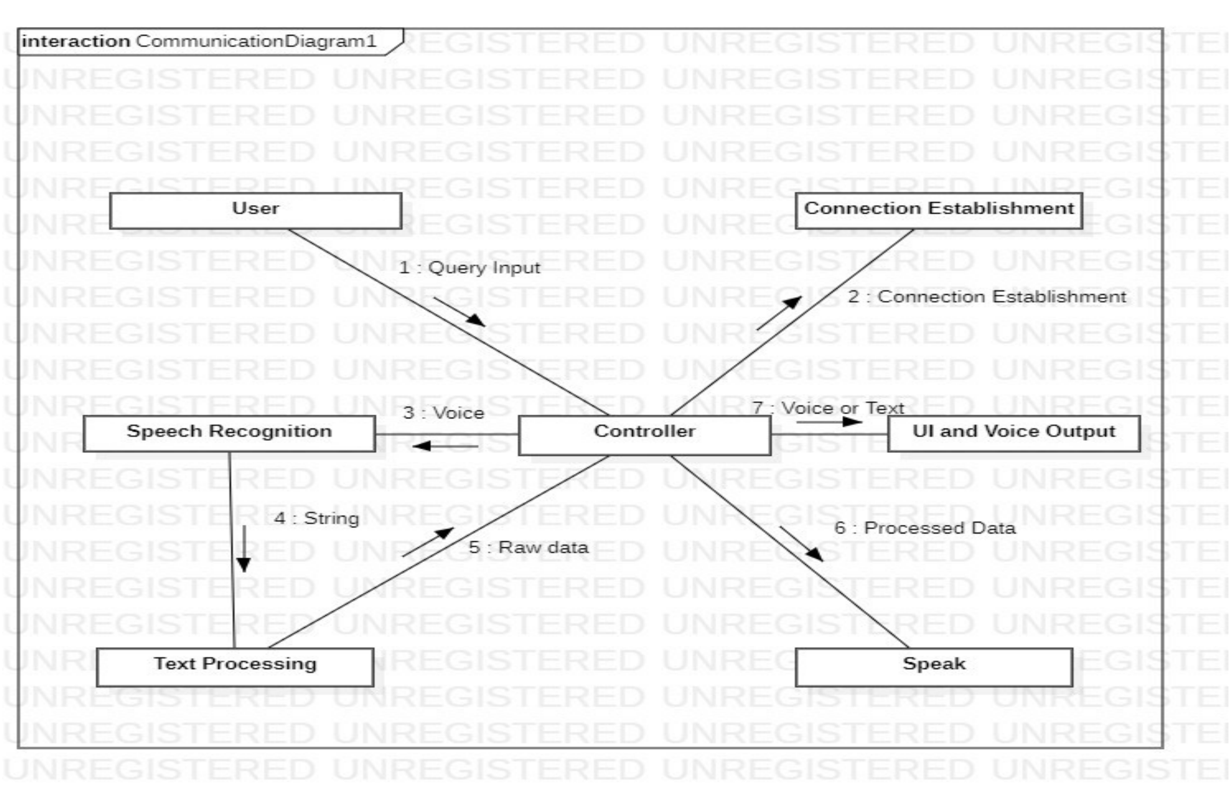


Figure 5.3.4: Communication Diagram

5.3.5 Component Diagram:

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.

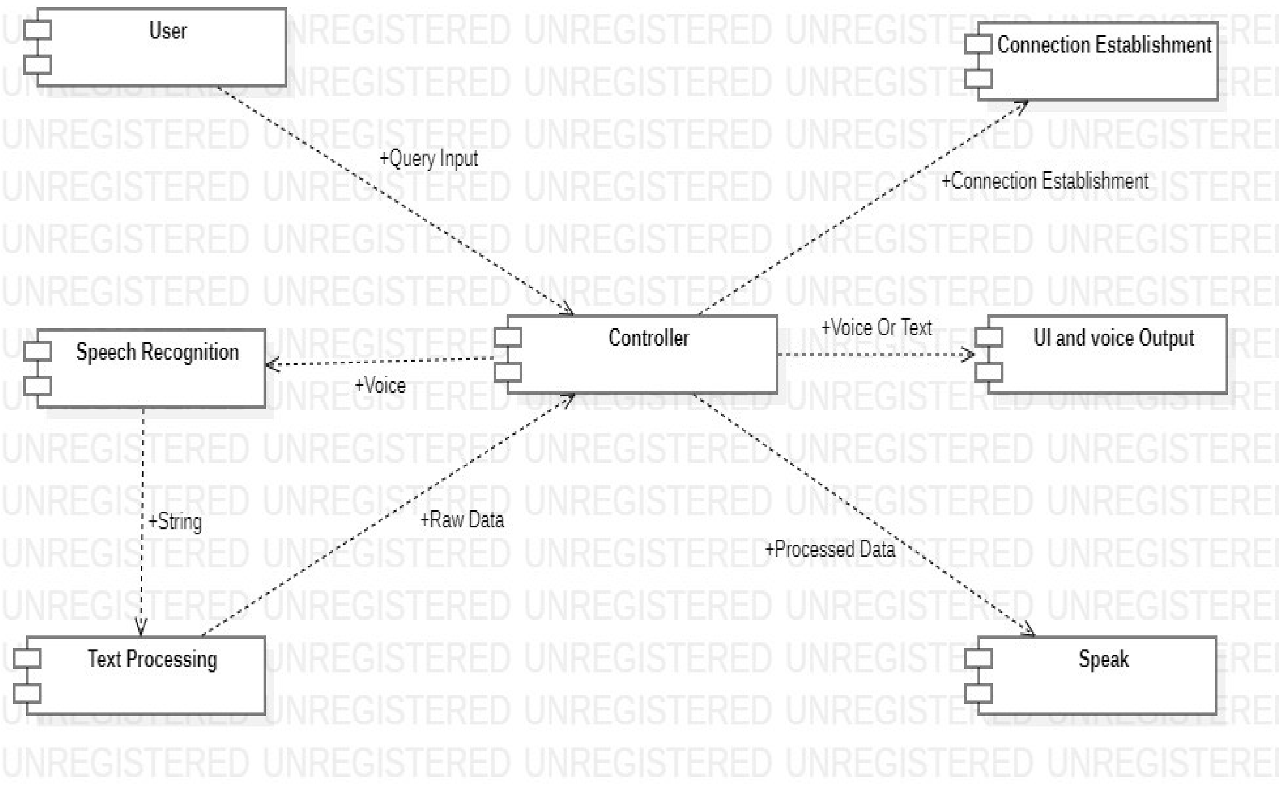


Figure 5.3.5: Component Diagram

5.3.6 Deployment Diagram:

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships. It as certains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

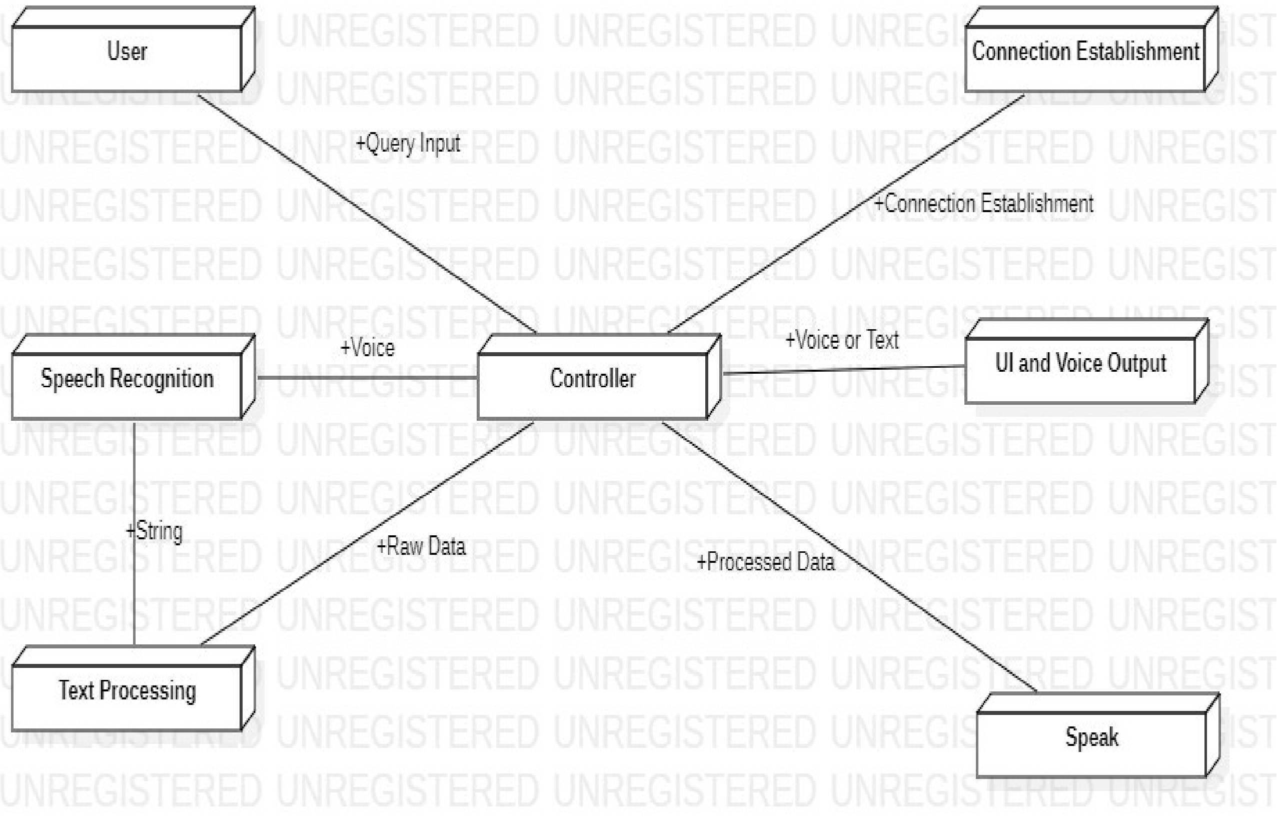


Figure 5.3.6 Deployment Diagram

5.3.7 Object Diagram:

Object diagrams are dependent on the class diagram as they are derived from the class diagram. It represents an instance of a class diagram. The objects help in portraying a static view of an object-oriented system at a specific instant. Both the object and class diagram are similar to some extent; the only difference is that the class diagram provides an abstract view of a system. It helps in visualizing a particular functionality of a system.

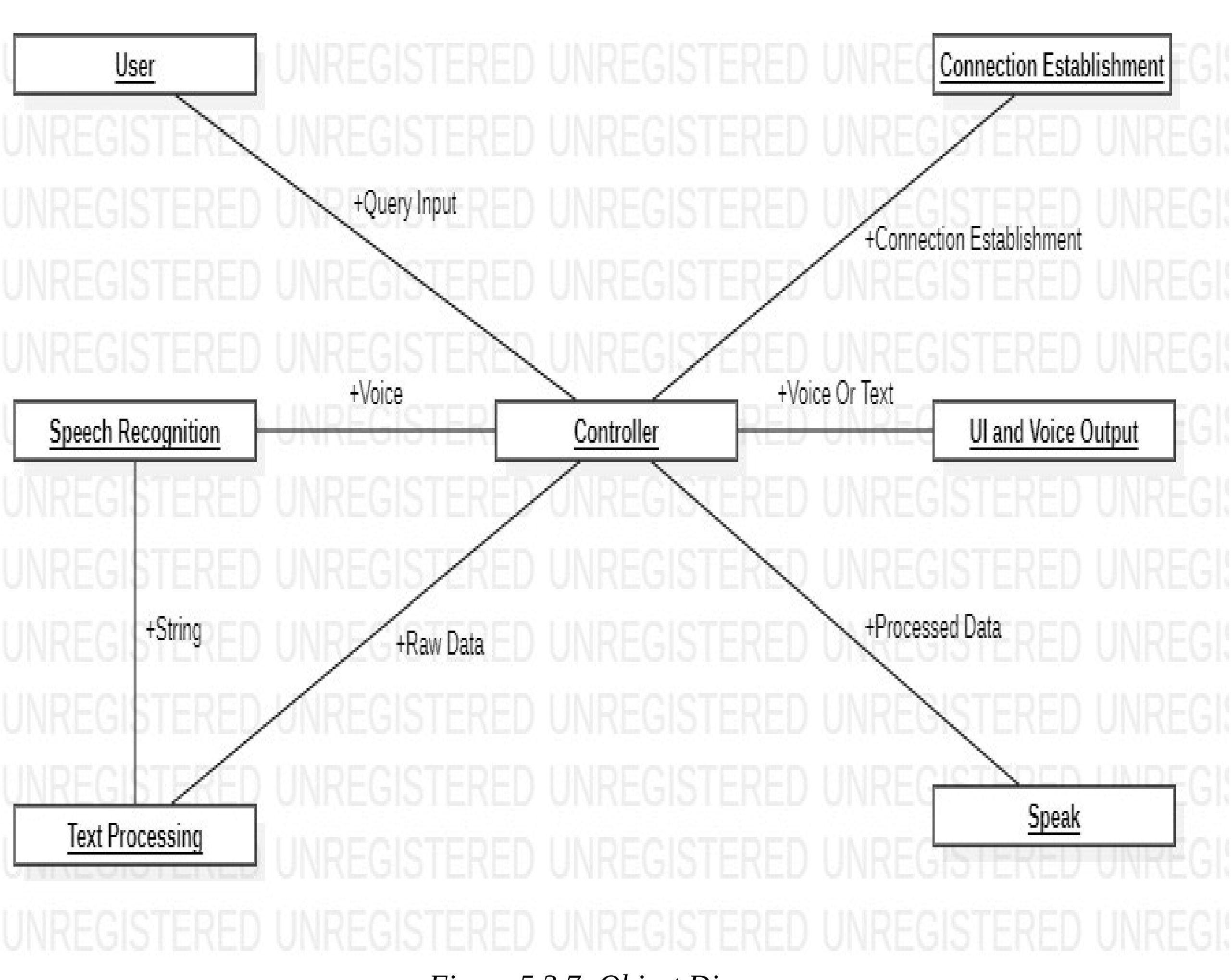


Figure 5.3.7: Object Diagram

5.3.8 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.

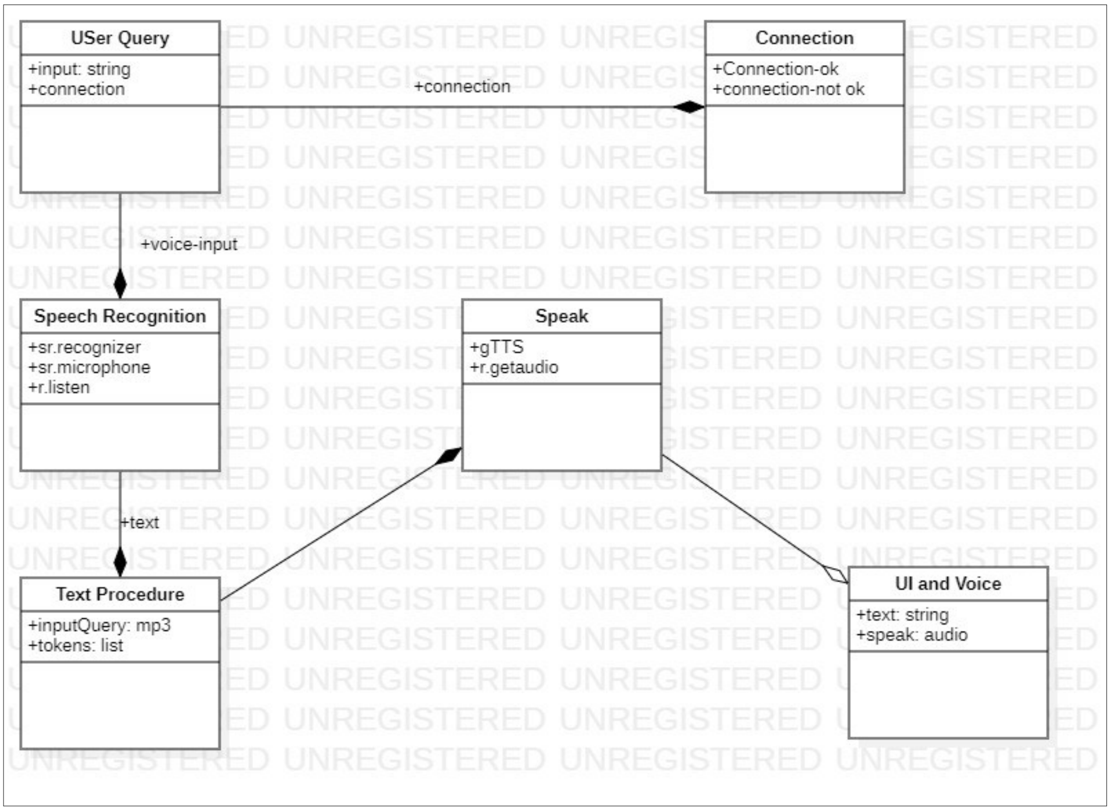


Figure 5.3.8: Class Diagram

6. SYSTEM IMPLEMENTATION :

6.1 Sample Codes:

6.1.1 Main.py:

```
import shutil

import webbrowser

import wikipedia

import sys

import os

import time

import playsound

import speech_recognition as sr

from gtts import gTTS

from datetime import datetime as day

from Attendance import takeAttendance

import subprocess

def speak(text):

    tts = gTTS(text=text, lang="en")

    filename = "abuvoice.mp3"

    tts.save(filename)

    playsound.playsound(filename)

def wishMe():

    hour = int(day.now().hour)

    if hour>= 0 and hour<12:

        speak("a very warm Good Morning Sir")

        print("\n**** Good Morning ****\n")
    elif hour>= 12 and hour<18:
        speak("a very Good Afternoon sir")
        print("\n**** Good Afternoon ****\n")

    else:
        speak("A very breeze Good Evening Sir")
        print("\n**** Good Evening ****\n")

    speak("I am your Personnel Assistant")
    print("I am your Personnel Assistant")
    speak("Tokyo 1 point o")
    print("Tokyo 1.0")

def usrname():
    speak("What should i call you sir")
    try:
        username = get_audio()
        speak("Welcome Mister")
        speak(username)
    except AssertionError as e:
        username ="ABDUL"
        speak(username)
    columns = shutil.get_terminal_size().columns
    print("#######################################".center(columns))
    print("Welcome Mr.", username.center(columns))
    print("#######################################".center(columns))
    speak("How can i Help you, Sir")
```

```
        print("How can i Help you, Sir")


def get_audio():
        r = sr.Recognizer()
        with sr.Microphone() as source:
                print("Listen ki inka poledhu.....\n")
                audio = r.listen(source)
                print("Ardham Ayindha")
                said = ""
                try:
                        print("Recognizing....\n")
                        said = r.recognize_google(audio, language ='en-in')
                        print(f'User said: {said}\n')
                except Exception as e:
                        print("Exception: Voice Not Recognized Properly" + str(e))
        return said.lower()


def note(text):
        t1 = time.localtime()
        time1 = time.strftime("%H:%M:%S", t1)
        file_name = time.ctime().replace(" ","_").replace(":","|")+"_note.txt"
        with open(file_name,"w") as f:
                f.write(text)
        subprocess.Popen(["gedit", file_name])


wishMe()
usrname()
```

```
while True :
        start = time.time()
        print("\n*************************************\n")
        print("Listening the audio: \n")
        text = get_audio()

        MANASA=["take the complaint", "write down the complaint"]
        if text in MANASA:
                speak("What is your identity number this is only for authentication")
                authen = get_audio()
                ID_NO =["45","67","233","434","091","001","879"]
                if authen in ID_NO:
                        speak("what would you like me to write down in complaint box")
                        note_text = get_audio()
                        note(note_text)
                        speak("i have made a note of that my friend")



        HELL=["Hai","hai","hi tokyo","hai tokyo how are you","hi how are you","hi tokyo how
        are you"]
        if text in HELL:
                print("Answering....")
                speak("Hello Berlin, how you doing? I am glad you have called me soon !")



        if 'wikipedia' in text:
```

```
        speak('Searching Wikipedia...')
        text = text.replace("wikipedia", "")
        results = wikipedia.summary(text, sentences = 3)
        speak("According to Wikipedia")
        print(results)
        speak(results)

if 'open youtube' in text:
        speak("Here you go to Youtube\n")
        webbrowser.open("https://www.youtube.com")

if 'open google' in text:
        speak("Here you go to Google\n")
        webbrowser.open("https://www.google.com")

if 'open stackoverflow' in text:
        speak("Here you go to Stack Over flow.Happy coding")
        webbrowser.open("htpps://www.stackoverflow.com")

if 'open spotify' in text:
        speak("Here you go to spotify.Enjoy the music")
        webbrowser.open("https://www.spotify.com")

if "will you be my gf" in text or "will you be my valentine" in text:
 speak("I'm not sure about, may be you should give me some time")

if "how are you" in text:
 speak("I'm fine, glad you me that")


if "i love you" in text:
 speak("It's hard to understand")


if "what is the time table tokyo" in text:
        if (day.today().strftime("%A") == 'Monday'):


                print("Answering....")
                speak("network technologies, business management or placement skills, unix and unix")


        if (day.today().strftime("%A") == 'Tuesday'):
                print("Answering....")
                speak("compiler design and compiler design, distributed systems and distributed systems")


        if (day.today().strftime("%A") == 'Wednesday'):
                print("Answering....")
                speak("network technologies, business management or placement skills, unix and unix")


        if (day.today().strftime("%A") == 'Thursday'):
                print("Answering....")
                speak("compiler design and compiler design, distributed systems and distributed systems")
        if (day.today().strftime("%A") == 'Friday'):
            print("Answering....")
            speak("only business management or placement skills")

        if (day.today().strftime("%A") == 'Saturday'):
            print("Answering....")
            speak("No classes Today")

        if (day.today().strftime("%A") == 'Sunday'):
            print("Answering....")
            speak("No classes Today")

    if "is library open tokyo" in text:
        t = time.localtime()
        current_time = time.strftime("%H:%M:%S", t)
        start = '08:30:00'



        end = '20:30:00'
        if start <= current_time <= end:
            print("Answering....")

            speak("Yes it is open and can study hard")
        else:
            print("Answering....")
            speak("no its not open, and study in your respective dormitory")
```

```
GIRLS=["who is the care taker for girls", "who is the caretaker for girls"]
if text in GIRLS:
        print("Answering....")
        speak("P.E.T Madam and the contact number is 9 3 4 7 2 4 4 8 1 5")

ALPHA=["who is the care taker for Alpha", "who is the caretaker for Alpha","who is the caretaker for alpha","who is the care taker for alpha"]
if text in ALPHA:
        print("Answering....")
        speak("T Mahesh Babu Sir and contact number is 9 3 8 1 2 0 7 1 5 3")

GAMMA=["who is the care taker for gama", "who is the caretaker for gama"]
if text in GAMMA:
        print("Answering....")
        speak("G Venkata Sai kishore Sir and contact number is 7036125335")

DELTA=["who is the care taker for Delta", "who is the caretaker for Delta","who is the care taker for delta","who is the caretaker for delta"]
if text in DELTA:
        print("Answering....")
        speak("S RavindranathReddy Sir and contact number is 9 zero 3 2 7 8 9 8 9 zero")
EXAMS=["when are the exams for 3rd year students", "when are the exams for third year students"]
if text in EXAMS:
        print("Answering....")
```

```
    speak("According to the Trusted Sources  Mid one is on april 9 and 10 and MId 2 is on april 14 and 15 and finally End semester will be in last week of april")

if "when are the holidays Tokyo" in text:
    print("Answering....")
    speak("According to mee seva suresh anna, after end semester you will have holidays")

if "tokyo play music" in text:
    print("Answering....")

playsound.playsound('/home/lovely/Documents/Mini_Project_Offical/py_Music/DripReport - Skechers (Official Music Video) Prod. OUHBOY.mp3', True)

ATTENDANCE = ["tokyo take attendance","take the attendance","take attendance","attendance please","attendance"]
if text in ATTENDANCE:
    print("Started the attendance\n")
    speak("Taking the attendance professor Abdul")
    speak("Students, all of you come in line one by one")
    listt=takeAttendance()

THANK = ["thankyou robot","thank you robot","thankyou","thank you","thanks namaste","namaste"]
if text in THANK:
    print("Answering....")
```

```
            speak("Ok OK OK")
            speak("Thankyou so much Berlin and come back soon for more information")
            print("**************************************\n")
            sys.exit()
    print("\nTime taken: ")
    print(time.time()-start)
```

6.1.2 Attendance.py:

```
import cv2
import json
import subprocess
import numpy as np
import face_recognition
from datetime import datetime
import time
def markAttendance(name):
    with open('attendance.csv','r+' ) as f:

        myDataList = f.readlines()
        nameList = []
        for line in myDataList:
            entry = line.split(',')
            nameList.append(entry[0])
        if name not in nameList:
            now = datetime.now()
            today = datetime.today()
            d1 = today.strftime("%d/%m/%Y")
```

```
                dtString = now.strftime('%H:%M:%S')
                f.writelines(f'\n{name},{dtString},{d1}')
def takeAttendance():
    cam = cv2.VideoCapture(0)
    users_data_file = open("sharedPreference.json","r")
    data = json.load(users_data_file)
    users_data_file.close()
    known_face_names = []
    known_face_encodings= []
    for user in data["users"]:
        known_face_names.append(user["name"])
        image = face_recognition.load_image_file(user["img_path"])
        face_encode = face_recognition.face_encodings(image)[0]
        known_face_encodings.append(face_encode)
    name=""

    face_locations = []
    face_encodings = []
    face_names = []
    process_this_frame = True
    while True:
        ret, frame = cam.read()
        small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
        rgb_small_frame = small_frame[:, :, ::-1]
        if process_this_frame:
            face_locations = face_recognition.face_locations(rgb_small_frame)
            face_encodings = face_recognition.face_encodings(rgb_small_frame,
            face_locations)
            face_names = []
            for face_encoding in face_encodings:
                                                matches =
face_recognition.compare_faces(known_face_encodings, face_encoding)
                name = "Unknown"
                face_distances =
face_recognition.face_distance(known_face_encodings, face_encoding)
                best_match_index = np.argmin(face_distances)
                if matches[best_match_index]:
                    name = known_face_names[best_match_index]
                face_names.append(name)
        process_this_frame = not process_this_frame
        for (top, right, bottom, left), name in zip(face_locations, face_names):
            top *= 4
            right *= 4
            bottom *= 4
            left *= 4
            cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)

cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 255, 0), cv2.FILLED)
            font = cv2.FONT_HERSHEY_DUPLEX
            cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255,
            255), 1)
            markAttendance(name)
            cv2.imshow('Attendance Face Recognizer', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cam.release()
    cv2.destroyAllWindows()
```

6.1.3 sharedPreference.json:

```
{
    "users":[
        {
            "name":"Md Abdul Musavvir",
            "img_path": "/home/lovely/Documents/Mini_Project_Offical/py_Images/
1.png"
        },
        {
            "name":"Bill Gates",
            "img_path": "/home/lovely/Documents/Mini_Project_Offical/py_Images/
2.jpg"
        },
        {
            "name":"Nossam Vignatha",
            "img_path": "/home/lovely/Documents/Mini_Project_Offical/py_Images/
3.png"
        }
    ]
}
```

7. SYSTEM TESTING:

7.1 Unit Testing:

Unit Testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

7.1.1 Test on Attendance Module:

First we take the input from the user for further executing the program and then it recognise the speech by speech recpgnition and text processing takes place by the google text to speech and the converted speech to text is being outputed by the playsound of the Voice assistant and open the frame called attendance, where it recognise the person whether the student is present or absent and it saved it to the excel sheet called attendace.csv, which stores in the same directory.

7.1.2 Test on Complaint Box Module:

First we take the input from the user and take the input for checking the pre-defined conditions and the voice assistant will ask the private key which is already given by the professor for maintaing the security and non-breachable program for the authentication. And then the user will take the input to note down the complaint in the note.txt file and then it will shows up the saved file and it also stores in the same directory.

7.1.3 Test on Wikipedia Server Module:

Here we will take the input and if the input is present in the query which is given in the program then it will replace the wikipedia from the voice input by space and takes the remaining query to the wikipedia server and gives the information about the person or the thing.

7.1.4 Test on Time Table:

Here we will take the input and if the input is present in the query which is given in the program then it will take the localtime by the time package in python and will understand the day of the current day and it will tells the time table of the current day of execution.

7.1.5 Test on Library Timings:

Here we will take the input and if the input is present in the query which is given in the program then it will take the localtime by the time package in python and will understand the time and also checks the in timings and out timings and it will tells the status of the library.

7.2 Intergration Testing:

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing.

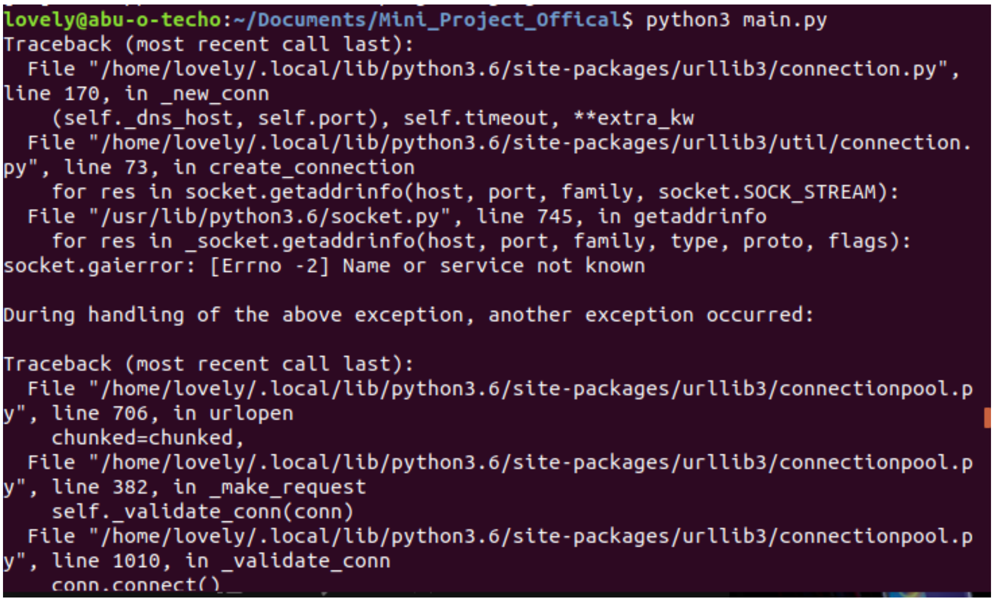
7.2.1 Test on Main Function of the program:

The programs are integrated in one module or combined and tested as a group for the overall execution of the program where, it greets the user and ask the name of the user and it speaks out “Welcome Mr. user_Name” and then we can give the input using the micrphone of the in-built system or the headphones connected to it which will then checks all the condition for the given voice string in the whole program where we specified to check.

And then it will gives the desired output what the user wants or it will raise the exception by voice not recognised properly

8. SAMPLE SCREENS:

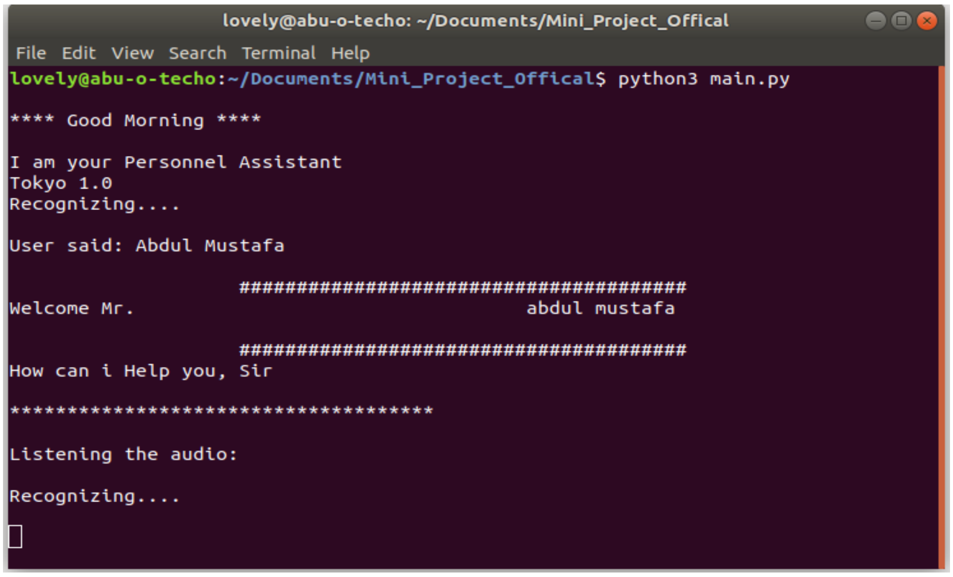
TestCase 8.1: Create connection : Here, we run the program using the command “python3 main.py” then it will show the Connection Error and tells to create the connection to access the Internet.



```
lovely@abu-o-techo:~/Documents/Mini_Project_Offical$ python3 main.py
Traceback (most recent call last):
  File "/home/lovely/.local/lib/python3.6/site-packages/urllib3/connection.py",
line 170, in _new_conn
    (self._dns_host, self.port), self.timeout, **extra_kw
  File "/home/lovely/.local/lib/python3.6/site-packages/urllib3/util/connection.
py", line 73, in create_connection
    for res in socket.getaddrinfo(host, port, family, socket.SOCK_STREAM):
  File "/usr/lib/python3.6/socket.py", line 745, in getaddrinfo
    for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
socket.gaierror: [Errno -2] Name or service not known

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/lovely/.local/lib/python3.6/site-packages/urllib3/connectionpool.p
y", line 706, in urlopen
    chunked=chunked,
  File "/home/lovely/.local/lib/python3.6/site-packages/urllib3/connectionpool.p
y", line 382, in _make_request
    self._validate_conn(conn)
  File "/home/lovely/.local/lib/python3.6/site-packages/urllib3/connectionpool.p
y", line 1010, in _validate_conn
    conn.connect()
```

Test Case 8.1 Create Connection

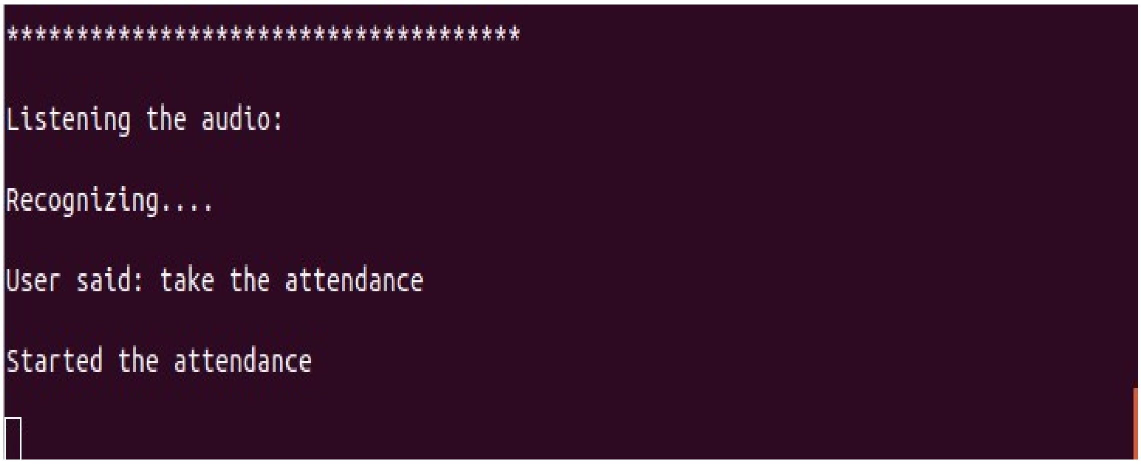
TestCase 8.2: Connection Established and Started: Connection has been successfully established and the program started by greetings. And take the name of the user before going into the actually execution.


lovely@abu-o-techo: ~/Documents/Mini_Project_Offical
File Edit View Search Terminal Help

```
lovely@abu-o-techo:~/Documents/Mini_Project_Offical$ python3 main.py

**** Good Morning ****

I am your Personnel Assistant
Tokyo 1.0
Recognizing....

User said: Abdul Mustafa

                    #######################################
Welcome Mr.                                  abdul mustafa

                    #######################################
How can i Help you, Sir

*************************************

Listening the audio:

Recognizing....

```

TestCase 8.2: Connection Established and Started

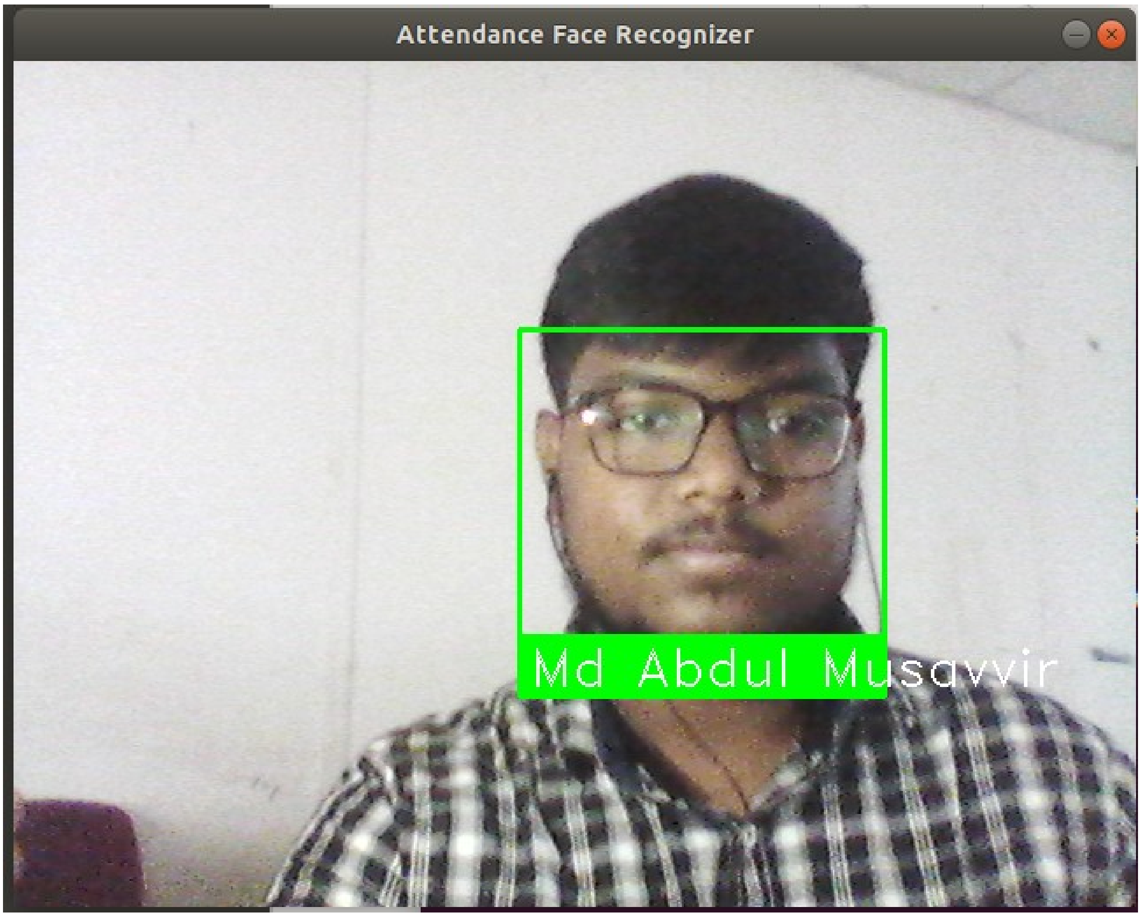
TestCase 8.3: Attendance: Here the user is going to say “take the attendance” then it will check the condition and it will redirect to the attendance.py where the attendance takes place by face recognition.



```
************************************

Listening the audio:

Recognizing....

User said: take the attendance

Started the attendance

```

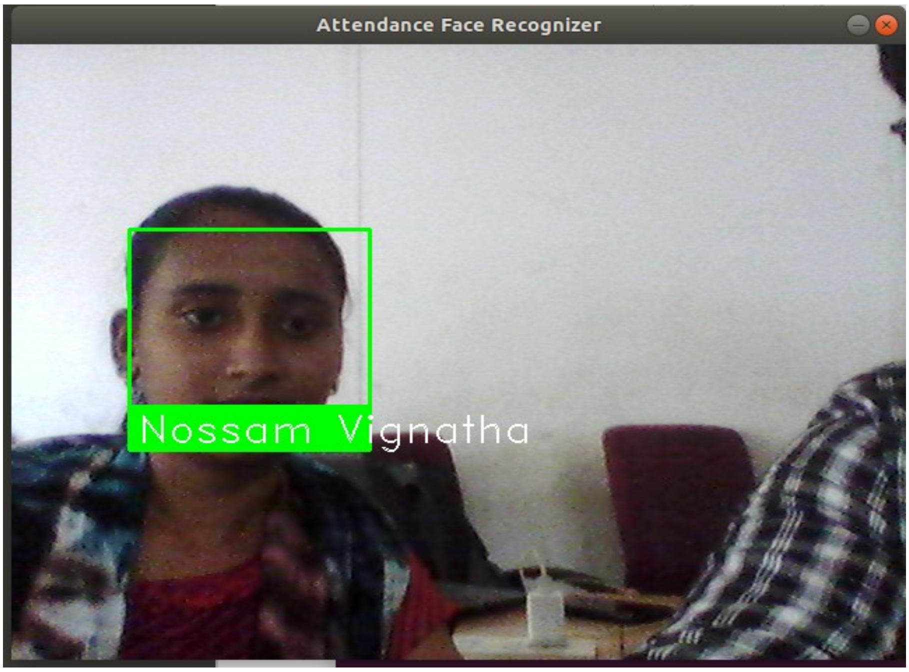
TestCase 8.3: Attendance

TestCase 8.3.1: Student1: Here, the student 1 (Md Abdul Musavvir) is on the screen which is being identified by the face recognizer.



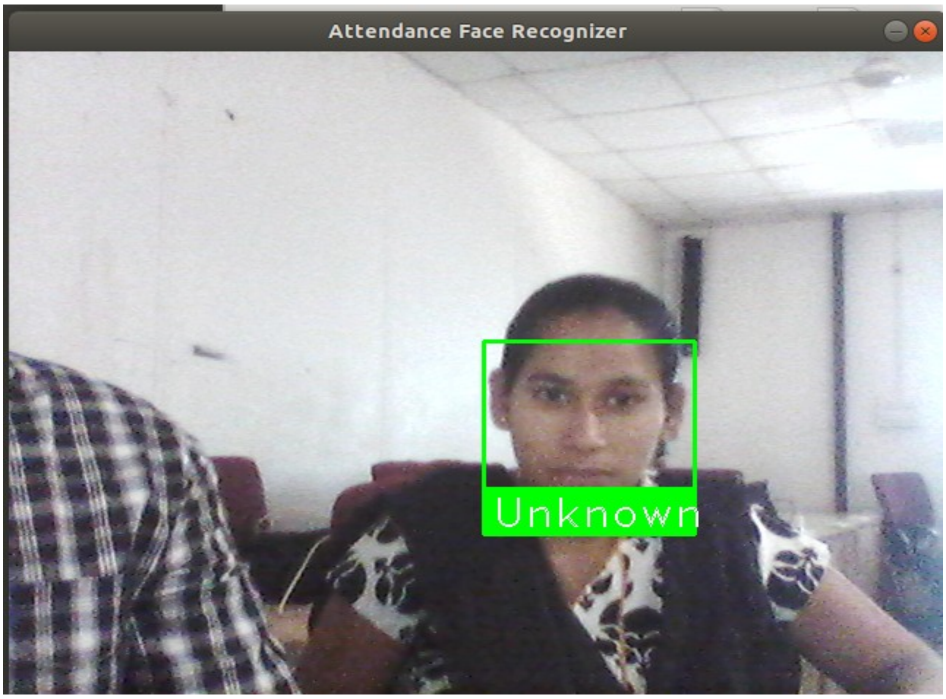
TestCase 8.3.1: Student1

TestCase 8.3.2: Student2: Here, the student 2 (Nossam Vignatha) is on the screen which is being identified by the face recognizer.



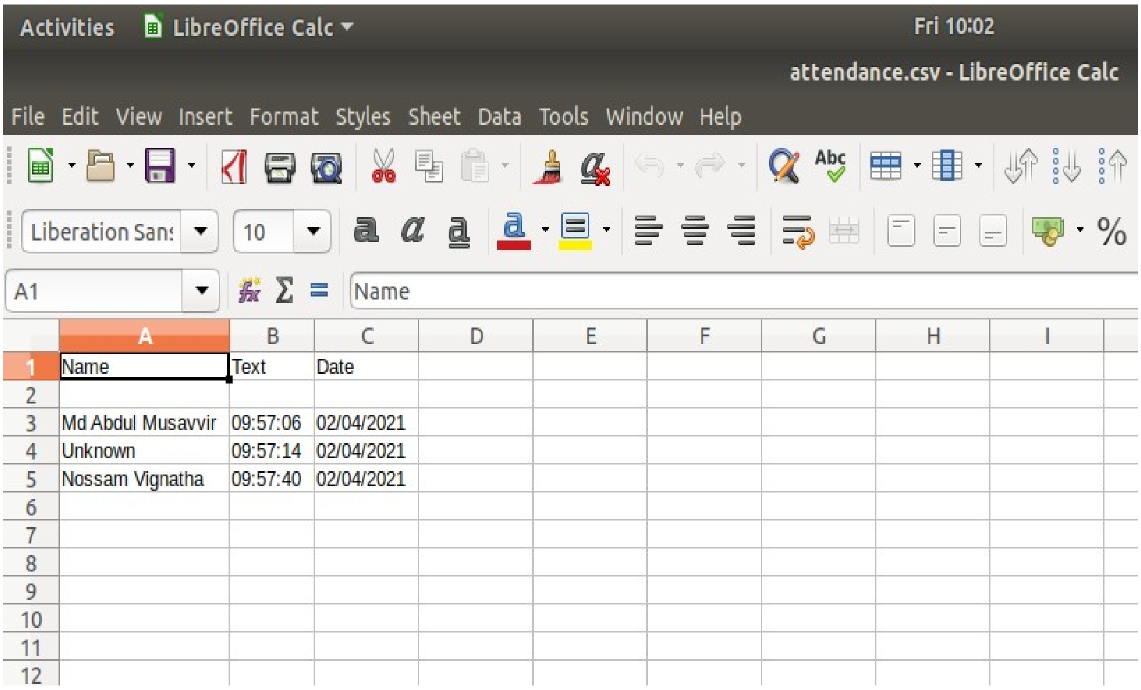
TestCase 8.3.2: Student2

TestCase 8.3.3: Unknown Person : Here, the Unknown person is on the screen which is not being identified by the face recognizer.



TestCase 8.3.3: Unknowm Person

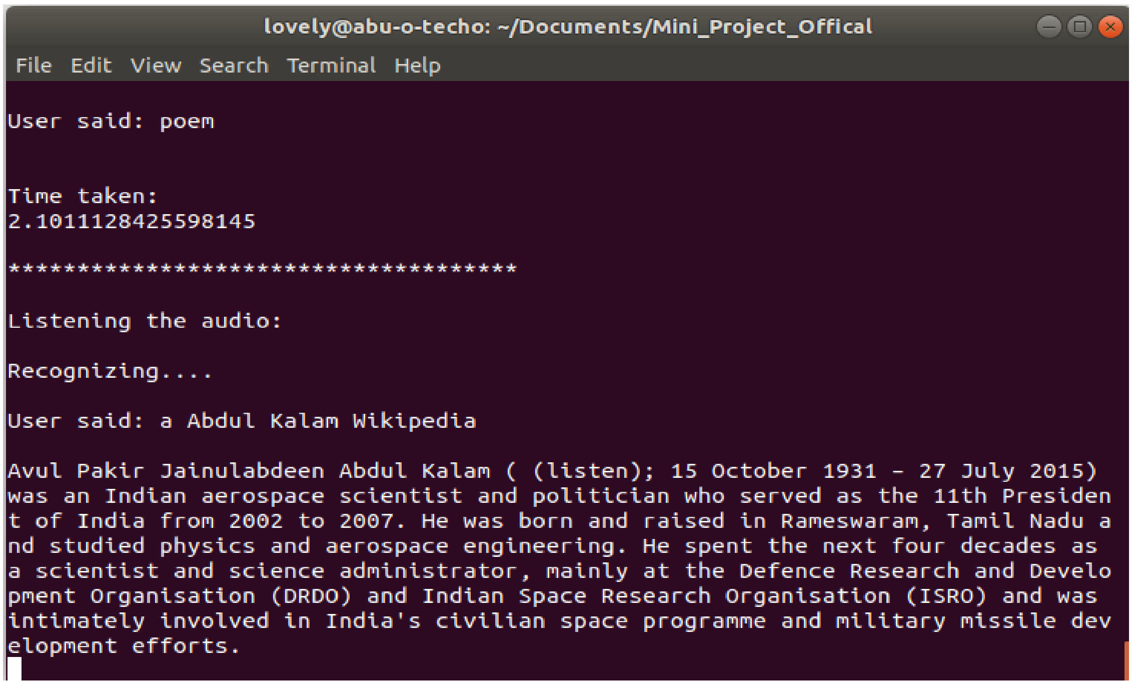
Test Case 8.4: Data Sheet: The data of the attendance is being kept stored in the same directory of the program execution by “attendance.csv” file name.


Activities
LibreOffice Calc
Fri 10:02
attendance.csv - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Liberation Sans
10
A1
Name

Name	Text	Date
Md Abdul Musavvir	09:57:06	02/04/2021
Unknown	09:57:14	02/04/2021
Nossam Vignatha	09:57:40	02/04/2021

Test Case 8.4: Data Sheet

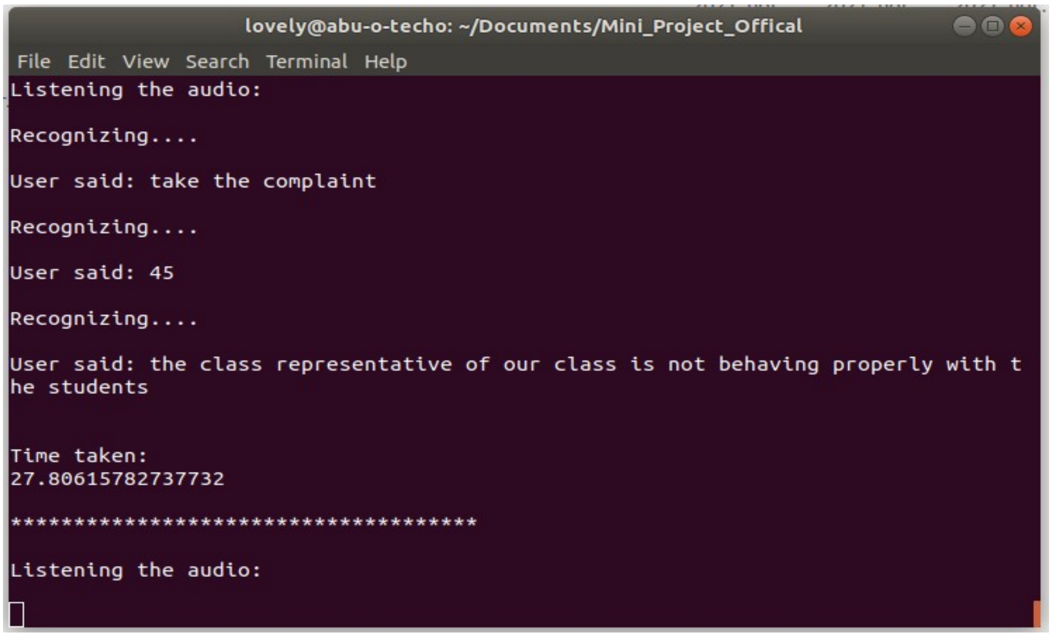
TestCase 8.5: WikiPedia: When the user said to the voice assistant about to search the abdul kalam wikipedia then it will take the input string to the wikipedia server then it will get the information about that person or thing to the controller and it will speak out.


lovely@abu-o-techo: ~/Documents/Mini_Project_Offical
File Edit View Search Terminal Help

```
User said: poem


Time taken:
2.1011128425598145

*************************************

Listening the audio:

Recognizing....

User said: a Abdul Kalam Wikipedia

Avul Pakir Jainulabdeen Abdul Kalam ( (listen); 15 October 1931 – 27 July 2015)
was an Indian aerospace scientist and politician who served as the 11th Presiden
t of India from 2002 to 2007. He was born and raised in Rameswaram, Tamil Nadu a
nd studied physics and aerospace engineering. He spent the next four decades as
a scientist and science administrator, mainly at the Defence Research and Develo
pment Organisation (DRDO) and Indian Space Research Organisation (ISRO) and was
intimately involved in India's civilian space programme and military missile dev
elopment efforts.
```

TestCase 8.5: WikiPedia

Test Case 8.6: Complaint Box: Here, the main thing is to take the complaint from the different students and for the selected students the private key is being provided and can give the complain which will return in the text format in the text file which is saved in the same directory.

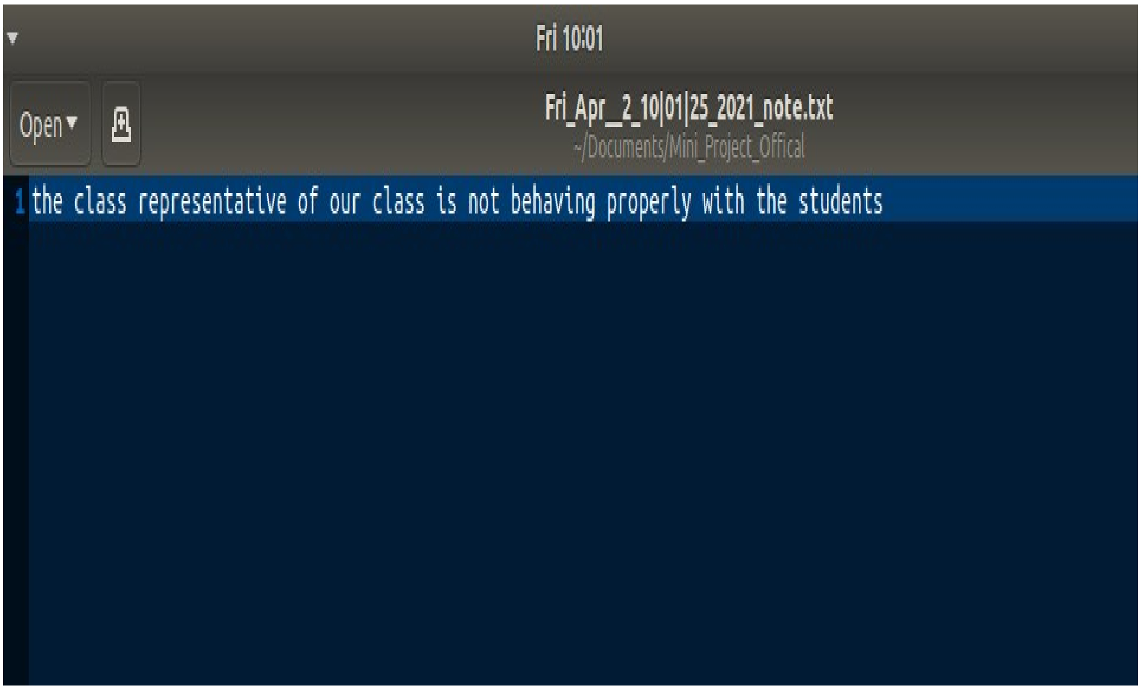


lovely@abu-o-techo: ~/Documents/Mini_Project_Offical
File Edit View Search Terminal Help

```
Listening the audio:

Recognizing....

User said: take the complaint

Recognizing....

User said: 45

Recognizing....

User said: the class representative of our class is not behaving properly with t
he students


Time taken:
27.80615782737732

*************************************

Listening the audio:
```

Test Case 8.6: Complaint Box

Test Case 8.6.1: Saved in Text File: The voice input complaint will be saved and stored in the txt(text) file.

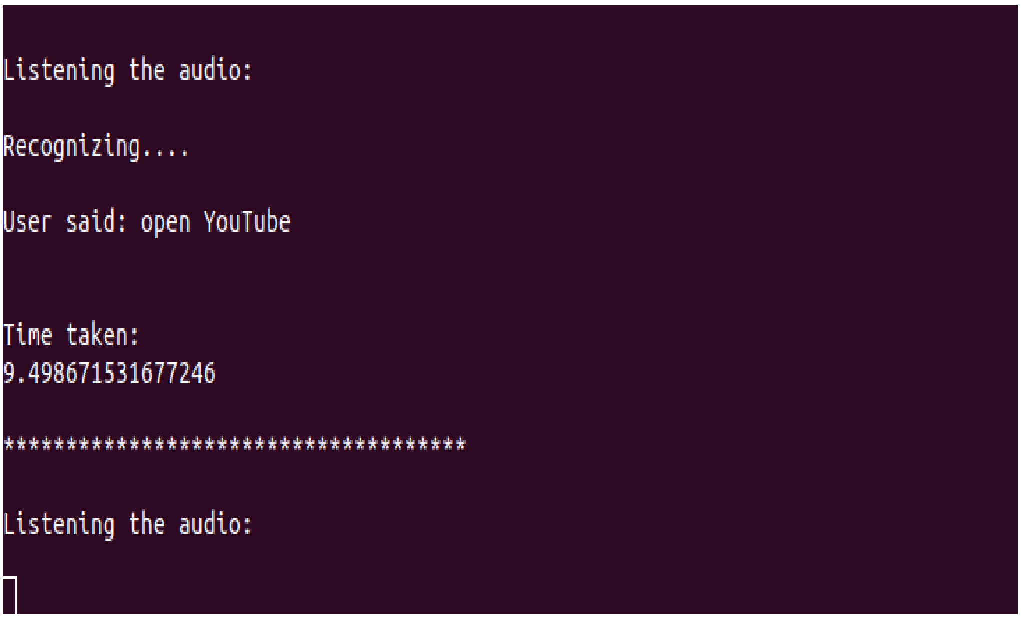


Fri 10:01
Open ▾
Fri_Apr_2_10|01|25_2021_note.txt
~/Documents/Mini_Project_Offical

```
the class representative of our class is not behaving properly with the students
```

Test Case 8.6.1: Saved in Text File

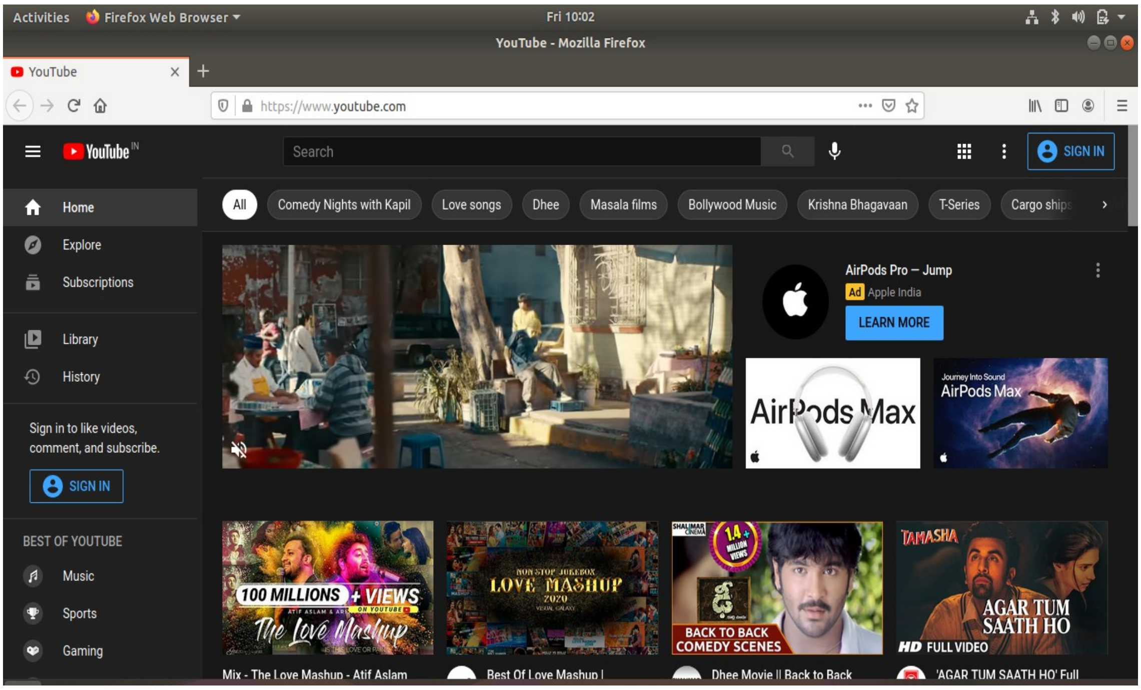
Test Case 8.7: Open YouTube: If the user wants to go to youtube just he or she as to say it “open youtube” then it will take you to the youtube with the default browser.



```
Listening the audio:

Recognizing....

User said: open YouTube


Time taken:
9.498671531677246

*************************************

Listening the audio:

```

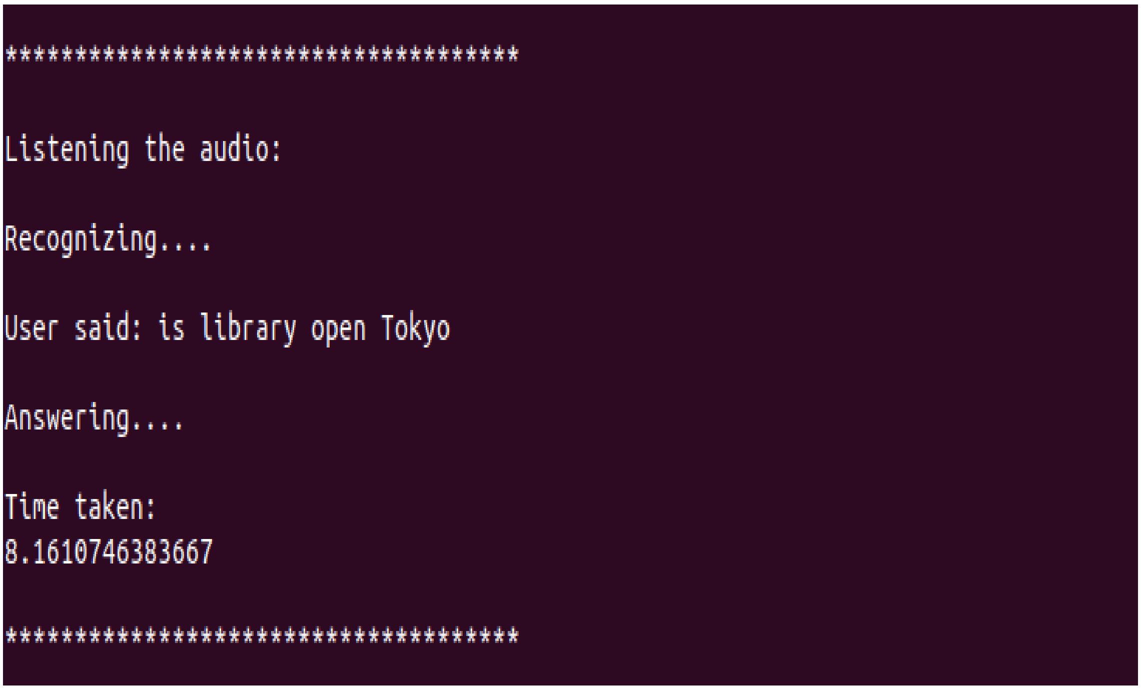
Test Case 8.7: Open YouTube

Test Case 8.7.1 : Youtube.com: It shows the user “YouTube” as per the given input and it will show up.



Test Case 8.7.1 : Youtube.com:

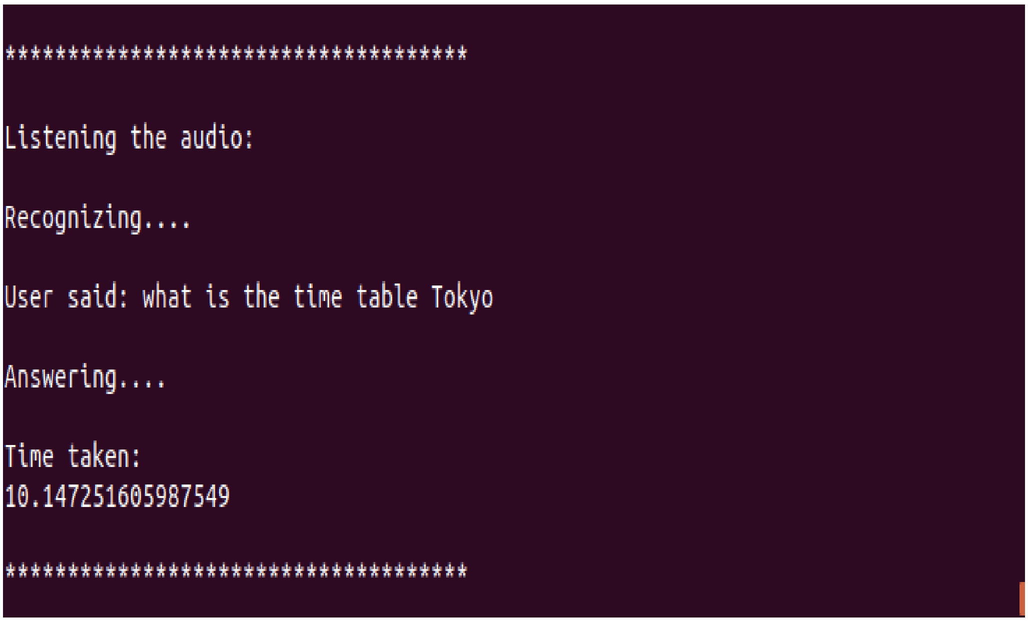
Test Case 8.8: Library Information: The timings of the library will be given to the program and it is pre-defined and will tells the user if it is open or close by the local time of the user.



```
*************************************

Listening the audio:

Recognizing....

User said: is library open Tokyo

Answering....

Time taken:
8.1610746383667

*************************************
```

Test Case 8.8: Library Information

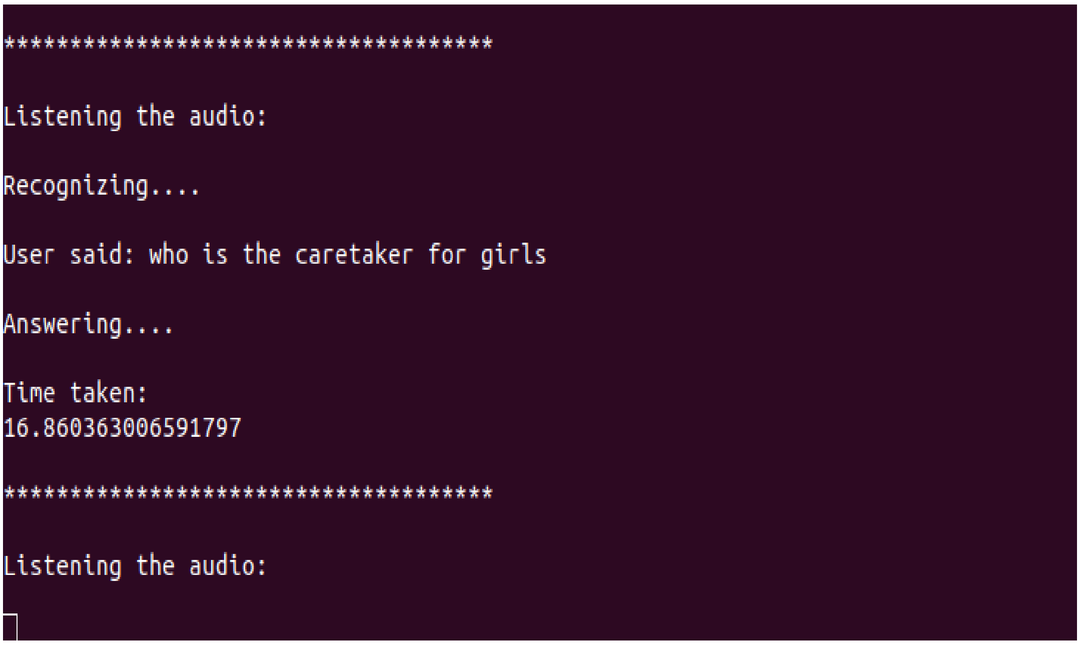
Test Case 8.9: Time Table Information: The time table data base is being stored and updated by the user when its changed and the timetable of the particular day will be speak out.



```
*************************************

Listening the audio:

Recognizing....

User said: what is the time table Tokyo

Answering....

Time taken:
10.147251605987549

*************************************
```

Test Case 8.9: Time Table Information

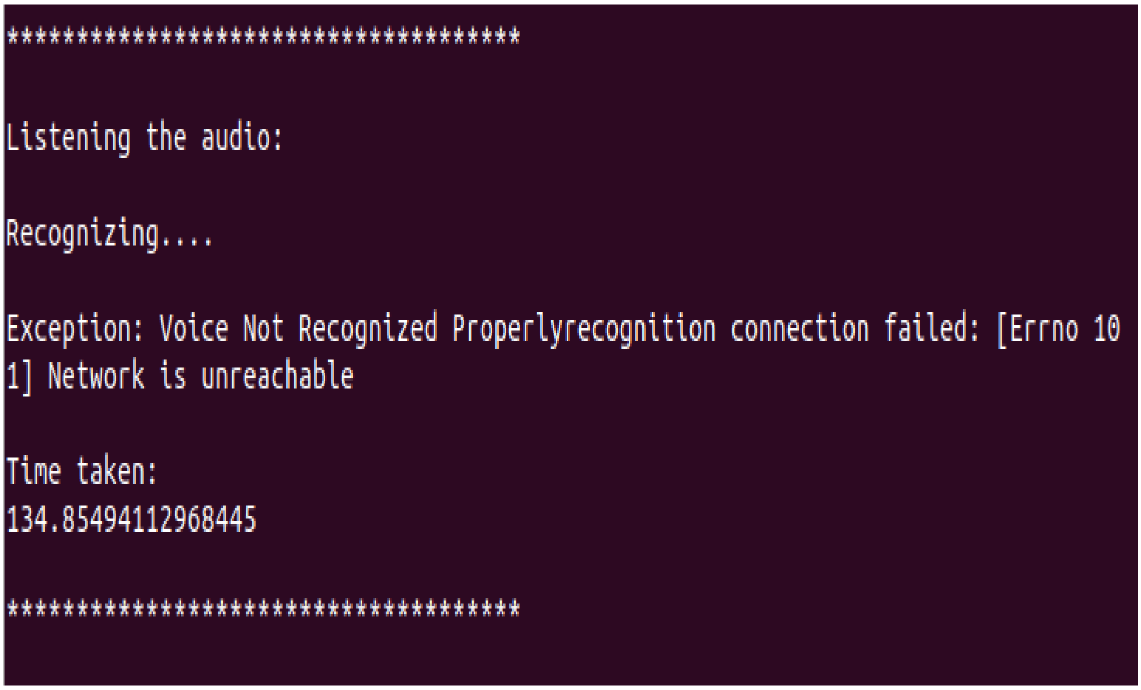
Test Case 8.10: Care Taker Information : It gives the information of the respective warden information and it will say the name of the sir or madam and speak out the contact number.



```
************************************

Listening the audio:

Recognizing....

User said: who is the caretaker for girls

Answering....

Time taken:
16.860363006591797

************************************

Listening the audio:
```

Test Case 8.10: Care Taker Information

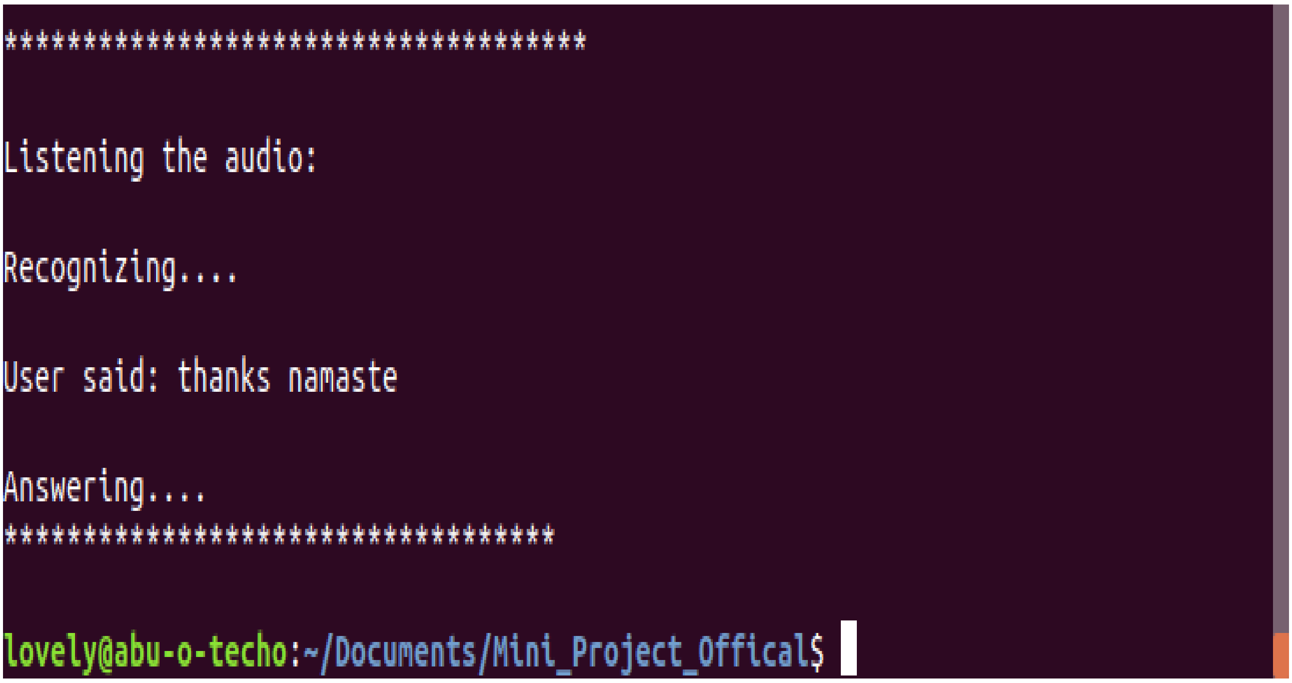
Test Case 8.11: Voice Input Exception: If the Voice of the user is not recognised properly or there is background noise then it will generate the Voice Input Exception.



```
************************************

Listening the audio:

Recognizing....

Exception: Voice Not Recognized Properlyrecognition connection failed: [Errno 10
1] Network is unreachable

Time taken:
134.85494112968445

************************************
```

Test Case 8.11: Voice Input Exception

Test Case 8.12: Thanking Assistant: When the user said “thanks namaste” then it will terminate the program by showing gratitude to the user for using it.



```
*************************************

Listening the audio:

Recognizing....

User said: thanks namaste

Answering....
***********************************

lovely@abu-o-techo:~/Documents/Mini_Project_Offical$
```

Test Case 8.12: Thanking Assistant

8. FUTURE SCOPE :

[9]The virtual assistants which are currently available are fast and responsive but we still have to go a long way. The understanding and reliability of the current systems need to be improved a lot. The assistants available nowadays are still not reliable in critical scenarios.

And the main thing is to be connected to the internet and should be online if running the program.

The future of these assistants will have the virtual assistants incorporated with Artificial Intelligence which includes Machine Learning, Neural Networks, etc. and IoT[x]. With the incorporation of these technologies, we will be able to achieve new heights. What the virtual assistants can achieve is much beyond what we have achieved till now.

Most of us have seen Jarvis, that is a virtual assistant developed by iron man which is although fictional but this has set new standards of what we can achieve using voice-activated virtual assistants.

9. CONCLUSION:

In this Project, we have discussed a **Voice Personal Assistant developed** using python. This assistant currently works online and performs basic tasks like takes the complaint, takes the attendance, opens the youtube, stream music, search wikipedia, open desktop applications,care taker information, library information, provides the timings of classes etc.

Through this voice assistant, we have automated various services using a voice input. It is designed for give responses to the user on the basis of **query** being asked or the words spoken by the user such as **opening tasks and operations**.

The functionality of the current system is limited to working online only. The upcoming updates of this assistant will have machine learning incorporated in the system which will result in better suggestions with IOT[x] to control the nearby devices similar to what Amazon's Alexa does. This Intelligent Voice Assistant has an enormous and limitless scope inthe future. Like **Siri**, **Google Now** and **Cortana** most popular personal voice assistants.

[2]The project will easily able to integrate with devices near future for a Connected Home using Internet of Things, voice command system and computer vision. The concept of voice recognition can be applied in different industries as in many situations it will be more convenient, and save a lot of time and helpful especially for those who have difficulty in working with manual operations. The primary objective of the program is to provide services using the voice, and it enables more people who can enjoy this program.

10. REFERENCES:

[1] https://www.youtube.com/playlist?list=PLzMcBGfZo4-mBungzp4GO4fswxO8wTEFx

[2] https://www.geeksforgeeks.org/voice-assistant-using-python

[3] https://pypi.org/project/face-recognition/

[4] https://pypi.org/project/numpy/

[5] https://pypi.org/project/opencv-python/

[6] https://pypi.org/project/pycopy-webbrowser/

[7] https://pypi.org/project/pytest-shutil/

[8] https://www.w3schools.com/python/

[9] https://www.techwithtim.net/tutorials/voice-assistant/

11. BIBLIOGRAPHY:

[i]	Package Installer for Python3.
[ii]	Google Text-To-Speech.
[iii]	Application programming Interface.
[iv]	Moving Picture Experts Group -1 Audio Layer III.
[v]	User Input.
[vi]	Operating System.
[vii]	Uniform Resource Locater.
[viii]	Directory Libraries.
[ix]	Unified Modeling Language.
[x]	Internet of Things.