**A**

**Mini Project Report**

**On**

**VOICE ASSISTANT**

**Submitted By:**

**M. ABHISHEK SINGH (19M91A0532)**

**MIRZA DANISH BAIG (19M91A0533)**

**T. AKHILESHWAR (19M91A0538)**

**T. VIJAY KUMAR (20M95A0502)**

**HARSHAVARDHAN GOUD (19U61A0556)**



**DEPARTMENT OF**

**COMPUTER SCIENCE AND ENGINEERING**

**AURORA’S SCIENTIFIC AND TECHNOLOGICAL INSTITUTE**

**Approved by AICTE, affiliated JNTU Hyderabad GHATKESAR -501301, TELANGANA**

**[2022-2023]**



**AURORA’S SCIENTIFIC AND TECHNOLOGICAL INSTITUTE**

**Approved by AICTE, affiliated JNTU Hyderabad GHATKESAR -501301, TELANGANA**

**[2022-2023]**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

Certified that Mini Project work entitled **“Voice Assistant”** is a bonafide work carried out in the IV/I semester by **“M. Abhishek Singh 19M91A0532, Mirza Danish Baig 19M91A0533, T. Akhileshwar 19M91A0538, T. Vijay Kumar 20M95A0502 & Harshavardhan Goud 19U61A0556”** in partial fulfillment for the award of Bachelor of Technology in Computer science and Engineering from Jawaharlal Nehru Technological University Hyderabad.

**Mini Project Coordinators Head of Department**

**R. KAVYA DR. M. SRIDHAR**

**B. LAXMIPATHI**

**External Examiner**

**ACKNOWLEDGEMENT**

The completion of this Mini Project work gives me an opportunity to convey my gratitude to all those who helped me to complete the Technical Seminar successfully.

First, I gratefully acknowledge my deep sense of gratitude to Almighty for spiritual Guidance blessings shown to complete the Mini Project. I thank my parents for unconditional support to improve myself throughout my life.

My sincere thanks to **Management**, of Aurora’s Scientific and Technological Institute, for providing this opportunity to carry out the Mini Project in the institution.

I owe my respectable thanks to **Dr. R. Mahesh Prabhu, (Principal)** of Aurora’s Scientific and Technological Institute, for providing all necessary facilities and encouraging words for completion of this Mini Project.

I gratefully acknowledge **Dr. M. Sridhar, (Head of Department)** of Computer Science and Engineering, for her encouragement and advice during this Mini Project.

My sincere thanks to **R. KAVYA, B. LAXMIPATHI,** **(Mini project Coordinators)** of major project Aurora’s Scientific and Technological Institute, for continuous support for doing this Mini Project.

I would like to express my thanks to all the faculty members of Department of

Computer Science and Engineering, and Non-teaching staff of Aurora’s Scientific and Technological Institute who have rendered valuable help in making this Mini Project successful.

**M. ABHISHEK SINGH (19M91A0532)**

**MIRZA DANISH BAIG (19M91A0526)**

**T. AKHILESHWAR (19M91A0538)**

**T. VIJAY KUMAR(20M95A0502)**

**HARSHAVARDHAN GOUD (19U61A0556)**

**INDEX**

**TABLE OF CONTENT PAGE NO**

**1. INTRODUCTION 7**

1.1 PROJECT SCOPE 8

1.2 PROJECT PURPOSE 8

1.3 PROJECT FEATURES 9

**2. SYSTEM ANALYSIS 9**

2.1 PROBLEM DEFINITION 10

2.2 EXISTING SYSTEM 10

2.2.1 LIMITATIONS OF THE EXISTING 11

SYSTEM

2.3 PROPOSED SYSTEM 12

2.3.1 ADVANTAGES OF PROPOSED SYSTEM 12

2.4 FEASIBILITY STUDY 13

2.4.1 ECONOMIC FEASIBILITY 13

2.4.2 TECHINICAL FEASIBILITY 13

2.4.3 BEHAVIORAL FEASIBILITY 13

2.5 HARDWARE & SOFTWARE REQUIREMENTS 14

2.5.1 HARDWARE REQUIREMENTS 14

2.5.2 SOFTWARE REQUIREMENTS 14

**3. ARCHITECTURE 14**

3.1 PROJECT ARCHITECTURE 14

3.2 DESCRIPTION 15

3.3 USECASE DIAGRAM 15

3.4 SEQUENCE DIAGRAM 16

1. **THEORY 21**

4.1 VOICE ASSISTANT 21

**5. IMPLEMENTATION 19**

5.1 INPUT 19

5.2 OUTPUT 20

**6. SCREENSHOTS 19**

**7. TESTING 22**

7.1 INTRODUCTION TO TESTING 22

7.2 TYPES OF TESTING 23

7.3 TEST CASES 26

**8. CONCLUSION & FUTURE SCOPE 31**

8.1 CONCLUSION 31

8.2 FUTURE SCOPE 32

**9. BIBLIOGRAPHY 33**

9.1 REFERENCES 33

9.2 WEBSITES 33

**LIST OF FIGURES**

**FIGURESNO FIGURENAME PAGENO**

1. 1. Project Architechture 16
2. 2. Use Case Diagram 18
3. 3. Sequential Diagram 19
4. 4. Activity diagram 20

**LIST OF SCREENSHOTS**

**SCREENSHOTNO SCREENSHOTNAME PAGENO**

Screenshot 1: CODE IMPLEMENTATION 19

Screenshot 2: TAKING INPUT FROM THE USER 19

Screenshot 3: RECOGNISING THE COMMAND FROM THE USER 20

Screenshot4: EXECUTED THE TASK SUCCESSFULLY 20

**ABSTRACT**

Voice assistants are software agents that can interpret human speech and respond via synthesized voices. Voice assistants are devices/apps that use voice recognition technology, natural language processing, and AI **to respond to humans**. Using the technology, the device synthesises the user's message, breaks it down, evaluates it, and offers a meaningful response in return. With the use of voice assistants, we can be able to control our digital devices without actually touching them.

Voice assistants are software agents that can interpret human speech

and respond via synthesized voices. Apple’s Siri, Amazon’s Alexa, Microsoft’s Cortana, and Google’s Assistant are the most popular voice assistants and are embedded in smartphones or dedicated home speakers.

Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other

basic tasks such as email, to-do lists, and calendars with verbal commands. This column will explore the basic workings and common features of

today’s voice assistants. It will also discuss some of the privacy and security issues inherent to voice assistants and some potential future users for these devices.

1. **INTRODUCTION**

A voice assistant is a digital assistant that uses voice recognition, language processing algorithms, and voice synthesis to listen to specific voice commands And return relevant information or perform specific functions as requested by the user Based on specific commands, sometimes called intents, spoken by the user, voice assistants can return relevant information by listening for specific keywords and filtering out the ambient noise. While voice assistants can be completely software based and able to integrate into most devices, some assistants are designed specifically for single device applications, such as the Amazon Alexa Wall Clock.

Today, voice assistants are integrated into many of the devices we use on a daily basis, such as cell phones, computers, and smart speakers. Because of their wide array of integrations, There are several voice assistants who offer a very specific feature set, while some choose to be open ended to help with almost any situation at hand.

**Voice recognition** works by taking an analog signal from a users voice and turning it into a digital signal. After doing this, the computer takes the digital signal and attempts to match it up to words and phrases to recognize the users intent. To do this, the computer requires a database of pre-existing words and syllables in a given language to be able to closely match the digital signal with. Checking the input signal with this database is known as **pattern recognition**, and is the primary force behind voice recognition.

Artificial intelligence is using machines to simulate and replicate human intelligence.

In 1950, Alan Turing (The namesake of our company) published his paper “Computing Machinery and

Intelligence” that first asked the question, can machines think? Alan Turing then went on to develop the Turing Test, a method of evaluating a computer to test its capability of thinking like a human. There were four approaches later developed that defined AI, Thinking humanly/rationally, and acting humanly/rationally. While the first two deal with reasoning, the second two deal with actual behavior. Modern AI is typically seen as a computer system designed to accomplish tasks that typically require human interaction. These systems can improve upon themselves using a process known as machine learning.

1.1 PROJECT SCOPE

Voice assistants will continue to offer more individualized experiences as they get better at differentiating between voices. However, it’s not just developers that need to address the complexity of developing for voice as brands also need to understand the capabilities of each device and integration and if it makes sense for their specific brand. They will also need to focus on maintaining a user experience that is consistent within the coming years as complexity becomes more of a concern. This is because the visual interface with voice assistants is missing. Users simply cannot see or touch a voice interface. Technological advances are making voice assistants more capable, particularly in AI, natural language processing (NLP), and machine learning. To build a robust speech recognition experience, the artificial intelligence behind it must become better at handling challenges such as accents and background noise.

1.2 PROJECT PURPOSE

Purpose of virtual assistant is to being capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Virtual assistants enable users to speak natural language voice commands in order to operate the device and its apps. There is an increased overall awareness and a higher level of comfort demonstrated specifically by millennial consumers. In this ever-evolving digital world where speed, efficiency, and convenience are constantly being optimized, it’s clear that we are moving towards less screen interaction.

1.3 PROJECT FEATURES

A voice assistant is a digital assistant that uses voice recognition, language processing algorithms, and voice synthesis to listen to specific voice commands and return relevant information or perform specific functions as requested by the user. These features work in different ways, one works as a

“personal assistant”, performing full tasks for you, i.e. send a text message, check stuff online, or tell you the weather. The other one allows you to select and open apps, dictate text in text fields, send, tap, scroll, and other tasks.

1. **SYSTEM ANALYSIS**

System Analysis is about complete understanding of existing systems and finding where the existing system fails. The solution is determined to resolve issues in the proposed system. It defines the system. The system is divided into smaller parts. Their functions and inter relation of these modules are studied in system analysis. You’ll need Python 3.6. We’ll be using the pyttsx3 package which is a text-to-speech library for Python. The basic reason why we use this is because it works offline. Another basic requirement of this project will be Python’s Speech Recognition library. There are other requirements for the project which are listed below; we’ll understand them as we go ahead. Inappropriate college description is also conveyed as all terms and conditions of college are not known to students.

Speech recognition, also known as automatic speech recognition (ASR), computer speech recognition, or speechto-text, is a capability which enables a program to process human speech into a written format. While it’s commonly confused with voice recognition, speech recognition focuses on the translation of speech from a verbal format to a text one whereas voice recognition just seeks to identify an individual user’s voiceSpeech recognition processing Speech recognition is the process of converting human sound signals into words or instructions. Speech recognition is based on speech. It is an important research direction of speech signal processing and a branch of pattern recognition. The research of speech recognition involves many subject areas such as computer technology, artificial intelligence, digital signal processing, pattern recognition, acoustics, linguistics, and cognitive science. It is a multidisciplinary comprehensive research field. Different research areas have emerged based on research tasks under different constraints. According to the requirements of the speaker's way of speaking, these areas can be divided into isolated words, connected words, and continuous speech recognition systems. According to the degree of dependence on the speaker, these areas can be divided into speech recognition systems for the specific person and nonspecific person. According to the size of vocabulary, they can be divided into small vocabulary, medium vocabulary, large vocabulary, and infinite vocabulary speech recognition systems.

**2.1 PROBLEM DEFINITION**

Usually, user needs to manually manage multiple sets of applications to complete one task. For example, a user trying to make a travel plan needs to check for airport codes for nearby airports and then check travel sites for tickets between combinations of airports to reach the destination. There is need of a system that can manage tasks effortlessly. We already have multiple virtual assistants. But we hardly use it. There are number of people who have issues in voice recognition. These systems can understand English phrases but they fail to recognize in our accent. Our way of pronunciation is way distinct from theirs. Also, they are easy to use on mobile devices than desktop systems. There is need of a virtual assistant that can understand English in Indian accent and work on desktop system. When a virtual assistant is not able to answer questions accurately, it’s because it lacks the proper context or doesn’t understand the intent of the question. Its ability to answer questions relevantly only happens with rigorous optimization, involving both humans and machine learning. Continuously ensuring solid quality control strategies will also help manage the risk of the virtual assistant learning undesired bad behaviors. They require large amount of information to be fed in order for it to work efficiently. Virtual assistant should be able to model complex task dependencies and use these models to recommend optimized plans for the user. It needs to be tested for finding optimum paths when a task has multiple sub-tasks and each sub-task can have its own sub-tasks. In such a case there can be multiple solutions to paths, and the it should be able to consider user preferences, other active tasks, priorities in order to recommend a particular plan.

2.2 EXISTING SYSTEM

The term virtual assistant, or virtual personal assistant, is also commonly used to describe contract workers who work from home doing administrative tasks typically performed by executive assistants or secretaries. Virtual assistants are typically cloud-based programs that require internet-connected devices and/or applications to work. Three such applications are Siri on Apple devices, Cortana on Microsoft Devices and Google Assistant on Android devices. There are also devices dedicated to providing virtual assistance. The most popular ones are available from Amazon, Google and Microsoft. To use the Amazon Echo virtual assistant, called Alexa, users call out the wake word, "Alexa." A light on the device signals to the user it is ready to receive a command, which typically involves simple language requests, such as

"what is the weather today," or "play pop music." Those requests are processed and stored in Amazon's cloud. The technologies that power virtual assistants require massive amounts of data, which feeds artificial intelligence (AI) platforms, including machine learning, natural language processing and speech recognition platforms. As the end user interacts with a virtual assistant, the AI programming uses sophisticated algorithms to learn from data input and become better at predicting the end user's

needs.

2.2.1 LIMITATIONS OF EXISTING SYSTEMS

Virtual assistants have a variety of privacy concerns associated with them. Features such as activation by voice pose a threat; as such features require the device to always be listening. However, such features are important to make devices accessible for people who may otherwise have trouble. Modes of privacy such as the virtual security button have been proposed to create a multilayer authentication for virtual assistants. Cortana, for example, works best by using data from a user's device, including emails and other communications,

a user's contacts, location data, search history, and data from other Microsoft services and skills

thirdparty applications that users choose to connect with. Users can choose not to sign in and share this data with Cortana, and adjust permissions to prevent certain data from being collected, though these actions limit the virtual assistant's usefulness. These virtual assistants require large amounts

of personal data and are always "listening" in order to respond to voice commands. Virtual assistants then retain voice interactions and personal information to improve the user experience. Virtual assistant

providers also maintain privacy policies, which define how each

company uses and shares personal information. In most cases, companies do not share customer- identifiable information without a customer's consent.

**2.3 PROPOSED SYSTEM**

To design a device that acts as a digital organizer to provide variety of services to its master. It will look at examples of intelligent programs with natural language processing that are currently available, with different categories of support, and examine the potential

usefulness of one specific piece of software as a VPA. It continues to expand its digital abilities in organizing events, ordering food, playing music, guiding services for travelling, game prediction etc. It is suggested that new technologies may soon make the idea of virtual personal assistants a reality.

Experiments conducted on this system, combined with user testing, have provided evidence that a basic program with machine learning algorithms in the form of a digital personal assistant. Using machine learning algorithms to iteratively learn user’s preference for each

theme based on quality feedback given by the user. The concept of a virtual assistant which is a digital service looking after a range of our needs is fast becoming a reality. As artificial intelligence and machine learning progress at pace, digital assistants are set to become our gateway to the internet and know more about us than we do ourselves. Siri and Google now are just the beginning. The device accepts voice input processes it through various machine learning algorithms to provide desired output to user.

1. User Interface: User interface module will be used by both Professor and students. User interface module will help to interact with Students and Professors vice versa. This module will contain all designing of web applications containing all views of activities like schedule exam, Complete training etc. In this module we are going to use asp.net MVC with HTML and CSS.

1. Speech Recognition Module: As the name suggests this module will help in recognizing the voice of the user. We will use windows speech recognition library to recognize speech. Speech recognition libraries will recognize commands given by use and will convert them into text. As per voice we will also recommend various commands.

1. Pattern recognition: It is the process of recognizing patterns by using machine learning algorithms. Pattern recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation.

One of the important aspects of the pattern recognition is its application potential.

1. Training Module: Training module will contain sets of commands for various activities. The training module will contain SQL databases from which application will get the best recommendation for users. For every voice command there will be a training set. As commands increase, the training module will expand further and will help users to do certain activities easily. Training module will directly be connected to our application.

1. Database Module: Database module will contain a database of applications alongside training databases. Database will be in SQL server. Application will fetch data for various activities. Training module will also get training sets from the training database. Depending on the domain and data characteristics, different types of combinations might produce dissimilar outputs. The following list describes several hybridization techniques that come into consideration to merge CF and CBF recommenders.

**2.4 FEASIBILITY STUDY**

Feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed system.

Five types of feasibility study are taken into consideration.

2.4.1 ECONOMIC FEASIBILITY

Here, we find the total cost and benefit of the proposed system over current system. For this project, the main cost is documentation cost. User also would have to pay for microphone and speakers. Again, they are cheap and available. As far as maintenance is concerned, JIA won’t cost too much.

2.4.2 TECHNICAL FEASIBILITY

It includes finding out technologies for the project, both hardware and software. For virtual assistant, user must have microphone to convey their message and a speaker to listen when system speaks. These are very cheap now a days and everyone generally possess them. Besides, system needs internet connection. While using JIA, make sure you have a steady internet connection. It is also not an issue in this era where almost every home or office has Wi-Fi.

**2.5 HARDWARE AND SOFTWARE REQUIREMENTS**

The software is designed to be light-weighted so that it doesn’t be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility.

Here are the minimum hardware and software requirement for virtual assistant.

2.5.1 HARDWARE -

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

2.5.2 SOFTWARE –

* Windows 7(32-bit) or above.
* Python 2.7 or later
* Chrome Driver
* Selenium Web Automation
* SQLite

1. **ARCHITECTURE**

The total design consists of these phases:

1. Collection of data which is in speech format.
2. Analyse the voice and convert it to text.
3. storing the data and processing it.
4. Speech generation from the text output that is processed



FIGURE 2. USE CASE DIAGRAM



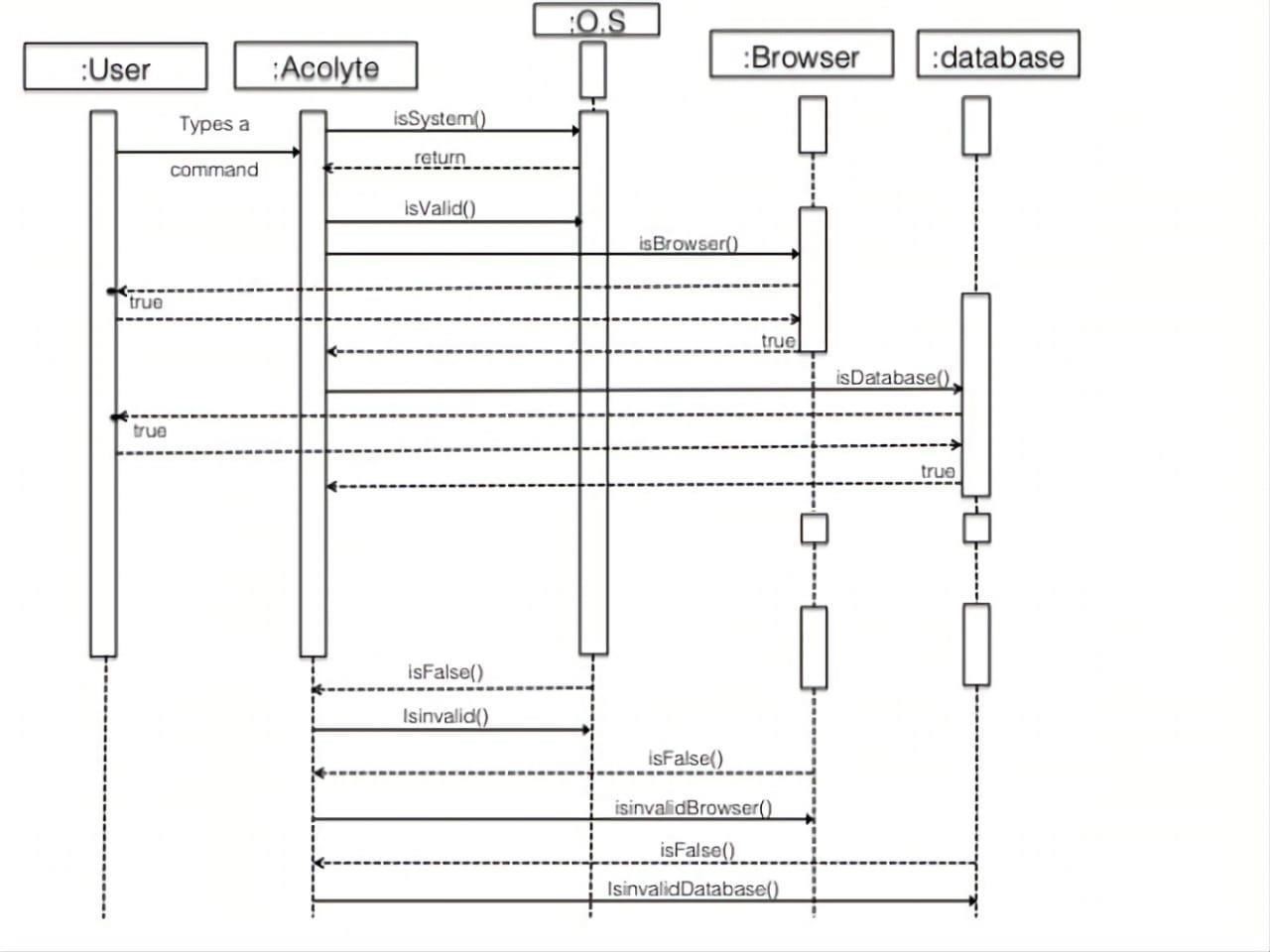


FIGURE 2. SEQUENTIAL DIAGRAM

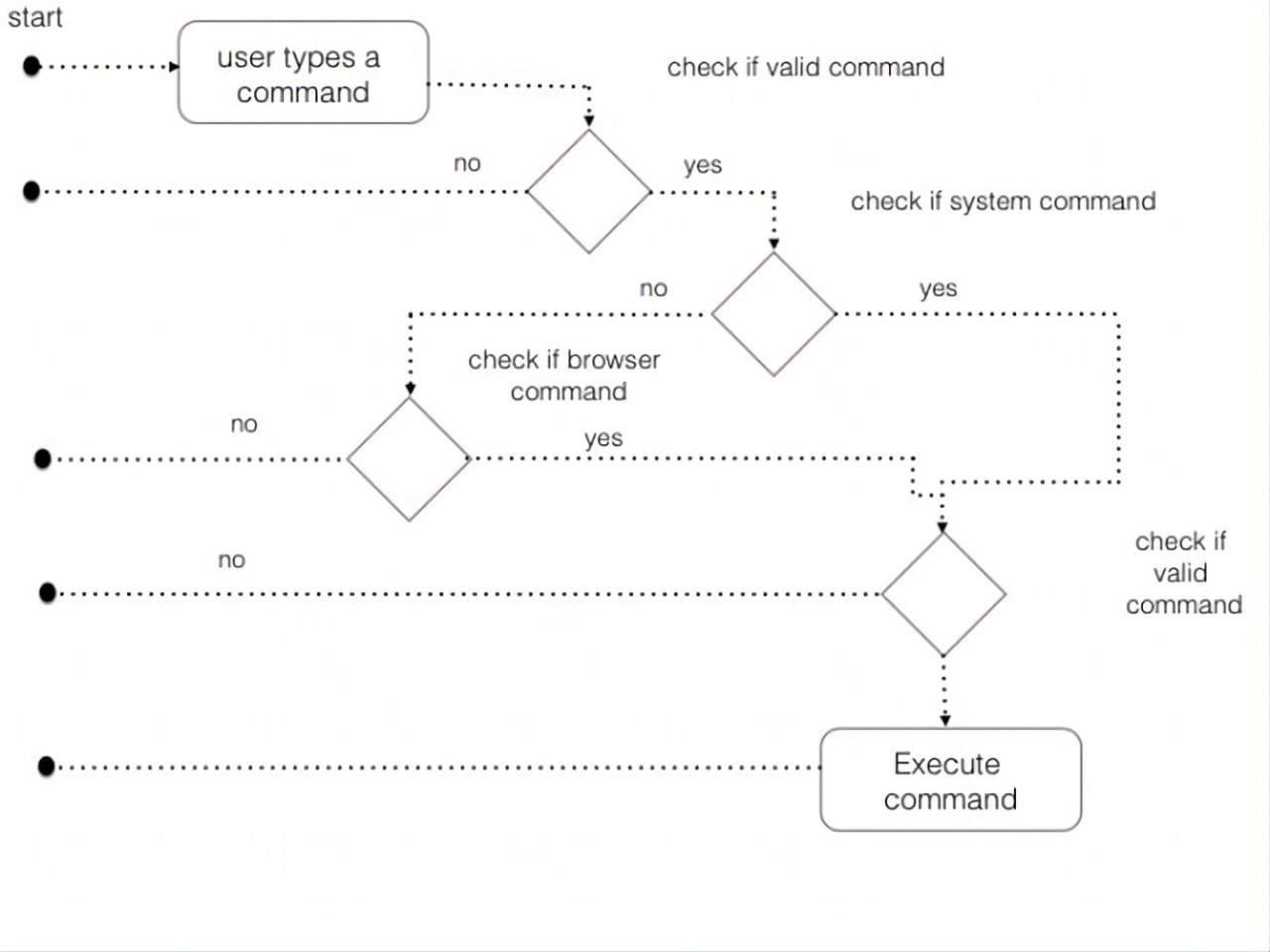
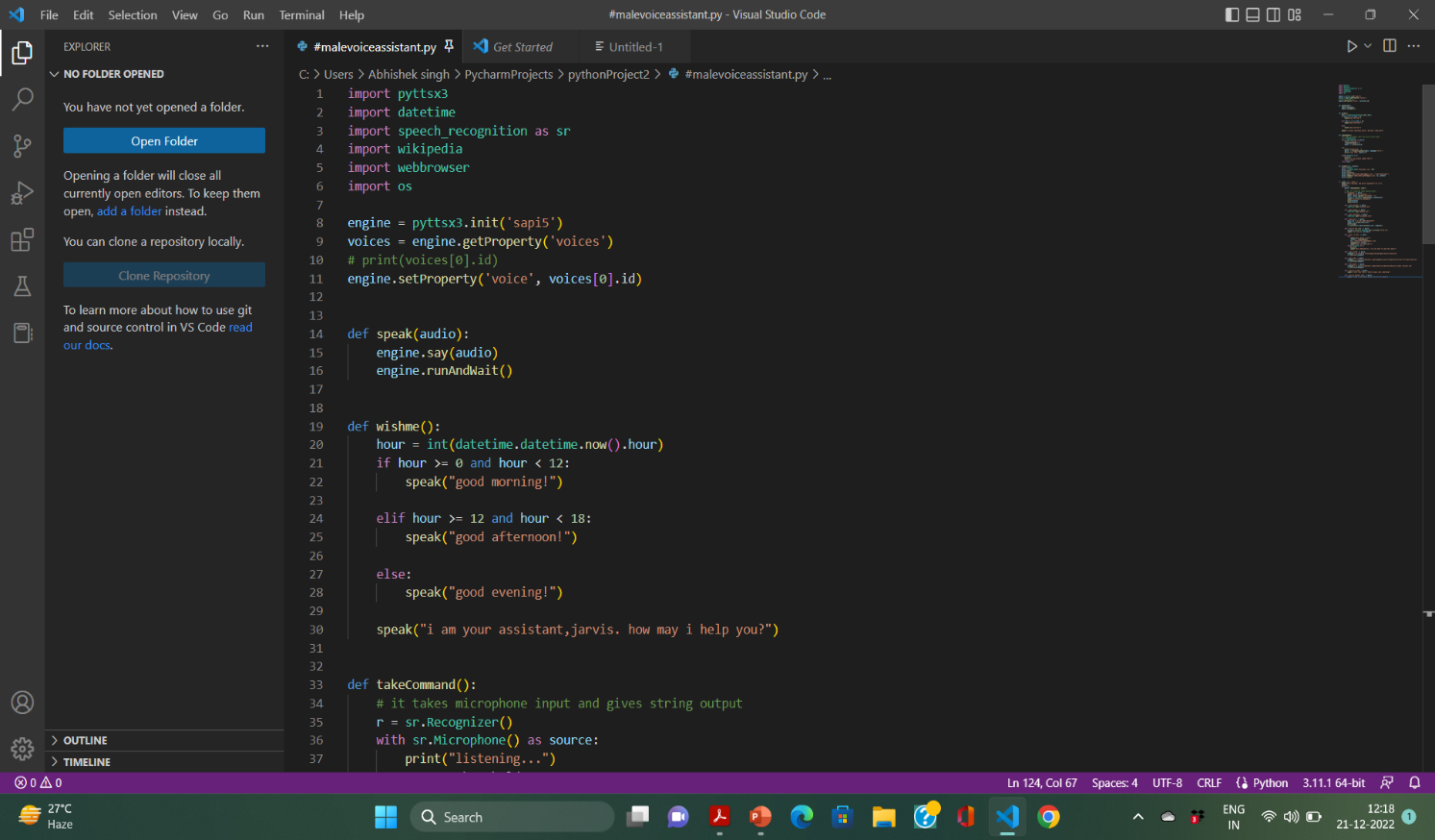


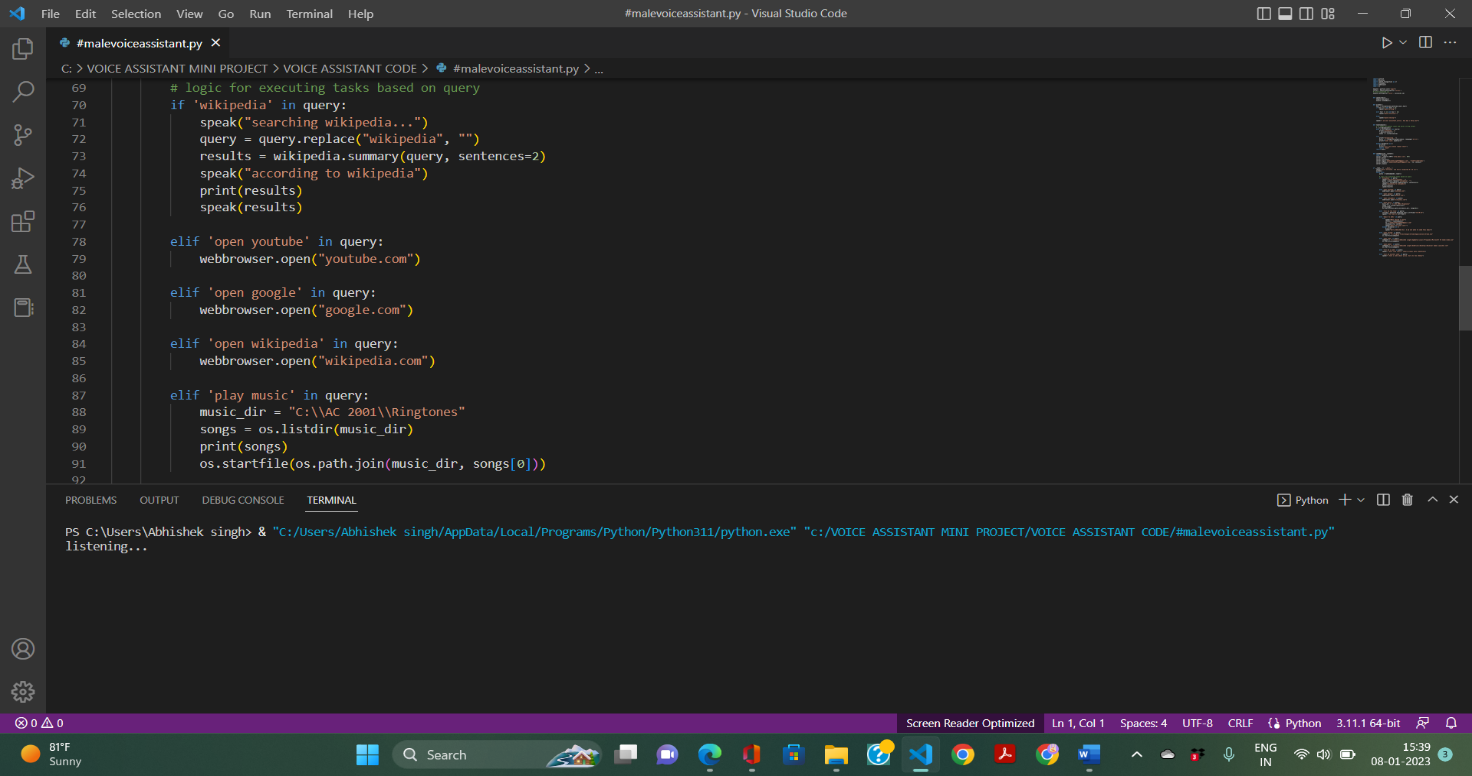
FIGURE 3. ACTIVITY DIAGRAM

LIST OF SCREENSHOTS

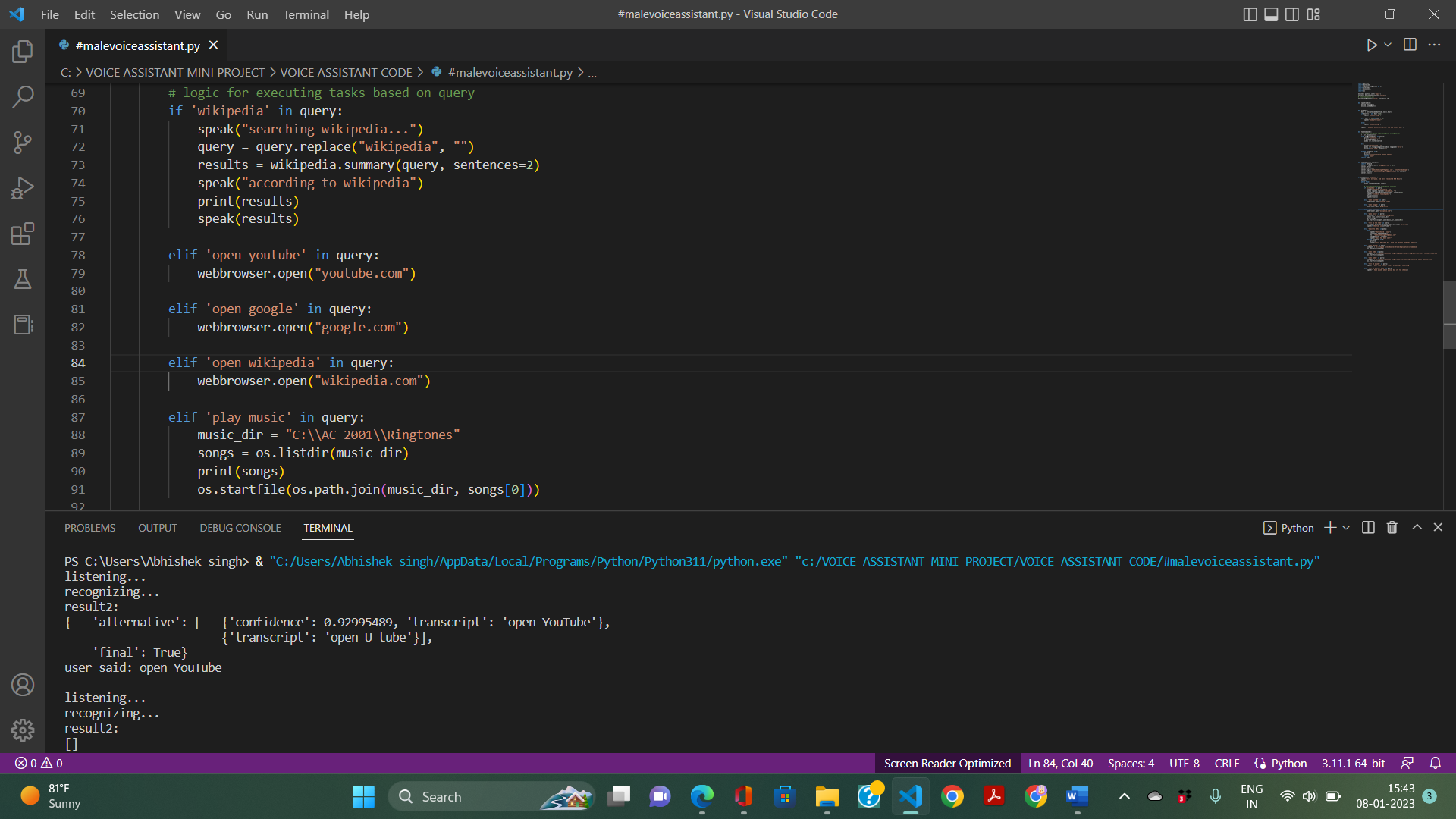
**5. IMPLEMENTATION**



Screenshot 1: CODE IMPLEMENTATION

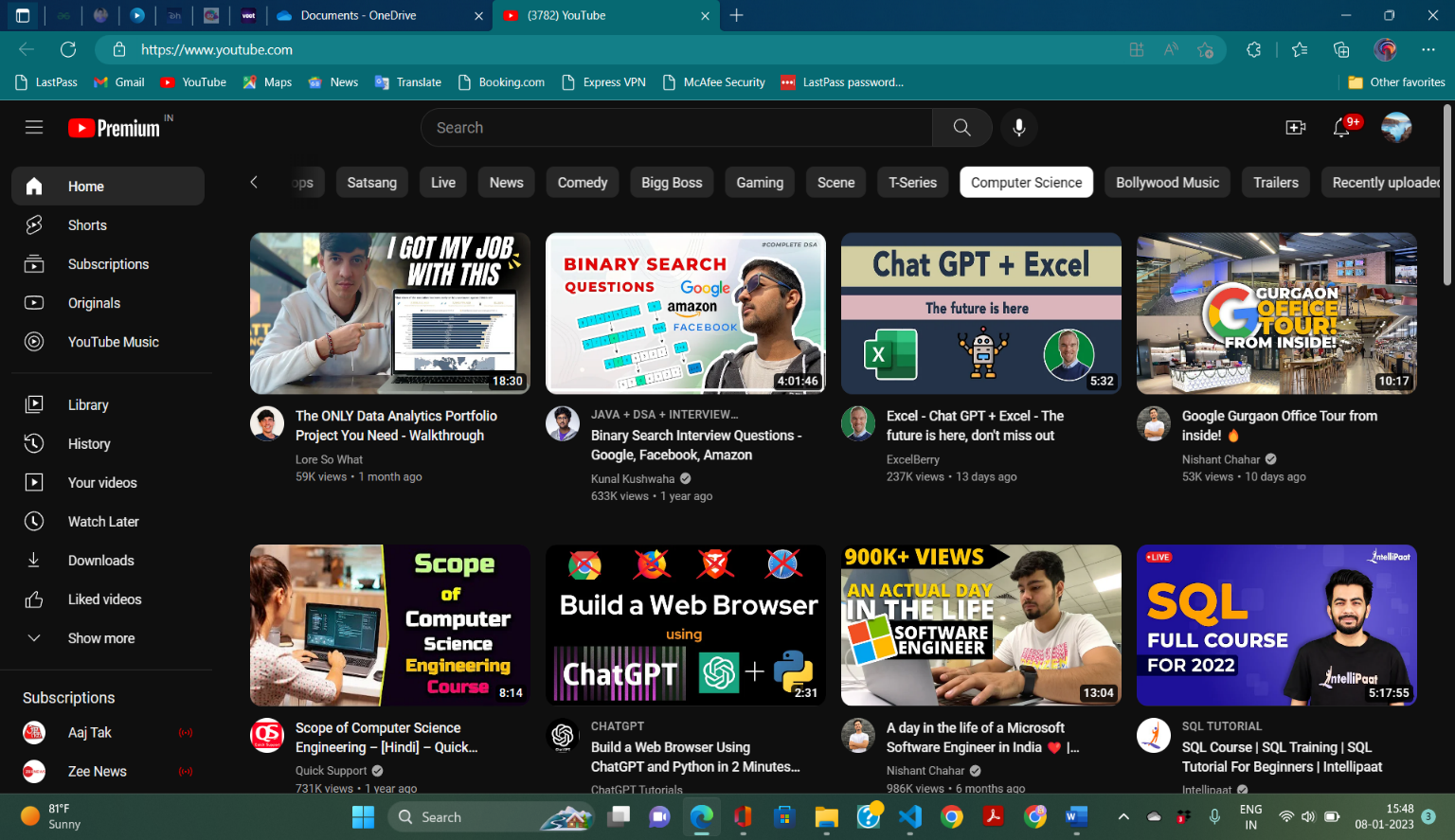


Screenshot 2: TAKING INPUT FROM THE USER



Screenshot 3: RECOGNISING THE COMMAND FROM THE USER

Screenshot 4: EXCECUTED THE TASK SUCCESSFULLY



4. THEORY

Personal Voice Assistant is developed as a desktop application with the help of Natural Language Processing which helps to send messages and use various built-in systems based and web based applications using voice commands. The Voice Assistant performs basic operations such as controlling computer tasks and operations, asking for temperature, humidity, date, time, and year. Adding, reading and deleting notes using voice commands and playing YouTube videos on demand. The above tasks can be performed using certain methodologies in which each technique has its own functionality and different operations to be performed. Each technique has different process logic to be executed.

You’ll need Python 3.6. We’ll be using the pyttsx3 package which is a text-to-speech library for Python. The basic reason why we use this is because it works offline. Another basic requirement of this project will be Python’s Speech Recognition library. There are other requirements for the project which are listed below; we’ll understand them as we go ahead. Inappropriate college description is also conveyed as all terms and conditions of college are not known to students.

Speech recognition, also known as automatic speech recognition (ASR), computer speech recognition, or speechto-text, is a capability which enables a program to process human speech into a written format. While it’s commonly confused with voice recognition, speech recognition focuses on the translation of speech from a verbal format to a text one whereas

voice recognition just seeks to identify an individual user’s voiceSpeech recognition processing Speech recognition is the process of converting human sound signals into words or instructions. Speech recognition is based on speech. It is an important research direction of speech signal processing and a branch of pattern recognition. The research of speech recognition involves many subject areas such as computer technology, artificial intelligence, digital signal processing, pattern recognition, acoustics, linguistics, and cognitive science. It is a multidisciplinary comprehensive research field. Different research areas have emerged based on research tasks under different constraints. According to the requirements of the speaker's way of speaking, these areas can be divided into isolated words, connected words, and continuous speech recognition systems. According to the degree of dependence on the speaker, these areas can be divided into speech recognition systems for the specific person and nonspecific person. According to the size of vocabulary, they can be divided into small vocabulary, medium vocabulary, large vocabulary, and infinite vocabulary speech recognition systems.

**7. TESTING**

7.1 INTRODUCTION TO TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

According to ANSI/IEEE 1059 standard, Testing can be defined as- A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

Who does Testing?

It depends on the process and the associated stake holders of the project(s). In the IT industry, large companies have a team with responsibilities to evaluate the developed software in context of the given requirements. Moreover, developers also conduct testing which is called Unit Testing. In most cases, the following professionals are involved in testing as system with in their respective capacities:

⚫ Software Tester

Software Developer

* Project Lead/Manager

End User

Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

Functional Testing

* Non-functional Testing

Functional Testing

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements

7.2 TYPES OF TESTING

1. Unit tests

Unit tests are very low level and close to the source of an application. They consist in testing individual methods and functions of the classes, components, or modules used by your software. Unit tests are generally quite cheap to automate and can run very quickly by a continuous integration server.

1. Integration tests

Integration tests verify that different modules or services used by your application work well together. For example, it can be testing the interaction with the database or making sure that microservices work together as expected. These types of tests are more expensive to run as they require multiple parts of the application to be up and running.

1. Functional tests

Functional tests focus on the business requirements of an application. They only verify the output of an action and do not check the intermediate states of the system when performing that action.

There is sometimes a confusion between integration tests and functional tests as they both require multiple components to interact with each other. The difference is that an integration test may simply verify that you can query the database while a functional test would expect to get a specific value from the database as defined by the product requirements.

1. End-to-end tests

End-to-end testing replicates a user behavior with the software in a complete application environment. It verifies that various user flows work as expected and can be as simple as loading a web page or logging in or much more complex scenarios verifying email notifications, online payments, etc...

End-to-end tests are very useful, but they're expensive to perform and can be hard to maintain when they're automated. It is recommended to have a few key end-to-end tests and rely more on lower level types of testing (unit and integration tests) to be able to quickly identify breaking changes.

1. Acceptance testing

Acceptance tests are formal tests that verify if a system satisfies business requirements. They require the entire application to be running while testing and focus on replicating user behaviors. But they can also go further and measure the performance of the system and reject changes if certain goals are not met.

1. Performance testing

Performance tests evaluate how a system performs under a particular workload. These tests help to measure the reliability, speed, scalability, and responsiveness of an application. For instance, a performance test can observe response times when executing a high number of requests, or determine how a system behaves with a significant amount of data. It can determine if an application meets performance requirements, locate bottlenecks, measure stability during peak traffic, and more.

1. Smoke testing

Smoke tests are basic tests that check the basic functionality of an application. They are meant to be quick to execute, and their goal is to give you the assurance that the major features of your system are working as expected.

Smoke tests can be useful right after a new build is made to decide whether or not you can run more expensive tests, or right after a deployment to make sure that they application is running properly in the newly deployed environment.

**SOFTWARE TESTING LIFE CYCLE**

The process of testing a software in a well-planned and systematic way is known as software testing lifecycle (STLC).

Different organizations have different phases in STLC however generic Software Test Life Cycle (STLC) for waterfall development model consists of the following phases.

1. Requirements Analysis

1. Test Planning

1. Test Analysis

1. Test Design

⚫ Requirements Analysis:

In this phase testers analyze the customer requirements and work with developers

during the design phase to see which requirements are testable and how they are going to test

those requirements. It is very important to start testing activities from the requirements phase itself because the cost of fixing defect is very less if it is found in re… [11:34 pm, 23/12/2022] Faisal Mirza: Test Design:

In this phase various black-box and white-box test design techniques are used to design the test cases for testing, testers start writing test cases by following those design techniques, if automation testing needs to be done then automation scripts also needs to written in this phase.

7.3 Test Cases -

The model is tested with test data.

Accuracy is calculated for algorithm.

voice assistant flows as part of your end to end testing can be achieved, in part, with an open-source

automation framework such as Appium.

There are four general steps to running automated tests for voice recognition software:

Screen navigation to voice assistant (supported by Appium).

Activate voice assistant (supported by Appium).

Say a voice command (requires advanced automation).

Validate screen/response (requires advanced automation).

While automated tests can handle the first two steps, the last two are challenging to perform as part of an automated script. Traditional libraries such as Appium do not support audio out of the box.

**Experimental Results** **Source Code:**

import pyttsx3

import datetime

import speech\_recognition as sr

import wikipedia

import webbrowser

import os

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

# print(voices[0].id)

engine.setProperty('voice', voices[0].id)

def speak(audio):

    engine.say(audio)

    engine.runAndWait()

def wishme():

    hour = int(datetime.datetime.now().hour)

    if hour >= 0 and hour < 12:

        speak("good morning!")

    elif hour >= 12 and hour < 18:

        speak("good afternoon!")

    else:

        speak("good evening!")

    speak("i am your assistant,jarvis. how may i help you?")

def takeCommand():

    # it takes microphone input and gives string output

    r = sr.Recognizer()

    with sr.Microphone() as source:

        print("listening...")

        r.pausethreshold = 1

        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("will you please repeat that?")

        return "none"

    return query

def sendEmail(to, content):

    import smtplib

    server = smtplib.SMTP('smtp.gmail.com', 587)

    server.ehlo()

    server.starttls()

    server.login('mabhisheksingh574@gmail.com', 'vtomfnljbubrnqmu')

    server.sendmail('mabhisheksingh574@gmail.com', to, content)

    server.close()

if \_\_name\_\_ == '\_\_main\_\_':

    speak("hello everyone. and hello respected H O D sir")

    wishme()

    while True:

        query = takeCommand().lower()

        # logic for executing tasks based on query

        if 'wikipedia' in query:

            speak("searching wikipedia...")

            query = query.replace("wikipedia", "")

            results = wikipedia.summary(query, sentences=2)

            speak("according to wikipedia")

            print(results)

            speak(results)

        elif 'open youtube' in query:

            webbrowser.open("youtube.com")

        elif 'open google' in query:

            webbrowser.open("google.com")

        elif 'open wikipedia' in query:

            webbrowser.open("wikipedia.com")

        elif 'play music' in query:

            music\_dir = "C:\\AC 2001\\Ringtones"

            songs = os.listdir(music\_dir)

            print(songs)

            os.startfile(os.path.join(music\_dir, songs[0]))

        elif 'tell me the time' in query:

            strTime = datetime.datetime.now().strftime("%H:%M:%S")

            speak(f"the time is {strTime}")

        elif 'email to abhi' in query:

            try:

                speak("What should I say?")

                content = takeCommand()

                to = "mabhisheksingh574@gmail.com"

                sendEmail(to, content)

                speak("Email has been sent!")

            except Exception as e:

                print(e)

                speak("Sorry Abhishek Sir. I am not able to send this email")

        elif 'open chrome' in query:

            codepath = "C:\Program Files\Google\Chrome\Application\chrome.exe"

            os.startfile(codepath)

        elif 'open code' in query:

            codepath = "C:\\Users\\Abhishek singh\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"

            os.startfile(codepath)

        elif 'open games' in query:

            codepath = "C:\\Users\\Abhishek singh\\OneDrive\\Desktop\\Rockstar Games Launcher.lnk"

            os.startfile(codepath)

        elif 'tell me a joke' in query:

            speak("i dont like stairs. theyre always upto something")

        elif 'tell me another joke' in query:

            speak("i have a joke about pizza. but its too cheesy")

**Below are the results when we run the code**

PS C:\Users\Abhishek singh> & "C:/Users/Abhishek singh/AppData/Local/Programs/Python/Python311/python.exe" "c:/VOICE ASSISTANT MINI PROJECT/VOICE ASSISTANT CODE/#malevoiceassistant.py"

listening...

recognizing...

result2:

{ 'alternative': [ {'confidence': 0.92995489, 'transcript': 'open YouTube'},

{'transcript': 'open U tube'}],

'final': True}

user said: open YouTube(opens youtube)

will you please repeat that?

listening...

recognizing...

result2:

{ 'alternative': [ {'confidence': 0.92995489, 'transcript': 'open Google'},

{'transcript': 'open gugal'},

{'transcript': 'open googol'},

{'transcript': 'open Guggal'},

{'transcript': 'Oppan Google'}],

'final': True}

user said: open Google(opens google)

listening...

**8. CONCLUSION AND FUTURE SCOPE**

71% of consumers already prefer voice search to manual typing since it’s much faster and also allows them to multitask. But as voice assistants become more powerful, easier to use, and able to understand context far better, more people will turn to voice search and virtual assistants for help with their everyday tasks.

In the near future, voice assistants are also expected to take a more proactive role. Rather than just waiting for user commands, assistants will collect context-specific information and then take the initiative by making helpful suggestions to the user. For example, people can interact with their in-car voice assistants to get information about fuel levels, diagnostics, and service needs or system settings that may need adjustment. So when fuel levels are low, the voice assistant may suggest going to the nearest gas station (with GPS directions if needed).

What’s more, an in-car voice assistant could be connected to intelligent home systems by integrating them with IoT devices or home automation systems. This would enable car owners to turn off the lights and set the alarm after they leave home or turn on the heating before they return.

Soon, voice assistants will also be able to authenticate purchases by recognizing a voice and matching it to a set credit card or bank account. Users will be able to pay for their orders simply by using voice commands - the voice assistant would only ask them to confirm the payment.

The option of paying through voice command is quickly growing in popularity. While only around 8% of the US adult population used voice payments in 2017, that number rose to 24% in 2021. Statista also predicts that over 30% of Americans will use voice payments by 2022 as a result of people increasingly looking for instant and contactless methods of payment.

Some companies are still hesitant to offer this payment method, fearing that it will open up new opportunities for fraudsters. However, using voice biometrics may prove to be a solution here. As each voiceprint is unique and nearly impossible to forge, voice assistants armed with voice biometrics technology shouldn’t have any problems differentiating real bank accounts or credit card owners from fraudsters. What’s more, it can work just as well at preventing any accidental purchases (made by children, for example) from going through by simply rejecting all payment orders that fail voice verification.

Conclusion : The future of voice search and assistants is looking bright. With the number of people already seeing how convenient those tools can be and the growing number of devices that use voice recognition. It's clear that the technology will soon be everywhere, and with 5G and improvements in machine learning, voice assistants might at some point become tools we can’t live without.

voice assistant is useful in many ways. we belive that voice assistants will become more advanced as the technology is advancing. voice assistants will be able to talk to us in a more natural way. they would be able to control the lights in our home, they would be able to warn us about the low fuel level in our cars, voice assistants will also be able to authenticate purchases by recognising a voice and matching it to a set credit card or bank account. users will be able to buy items online simply by using their voice commands etc.

**9. BIBLOGRAPHY**

9.1 REFERENCES:

[https://www.phonexia.com/blog/the-future-of-voicehttps://www.phonexia.com/blog/the-future-of-voiceassistants/ -](https://www.phonexia.com/blog/the-future-of-voice-assistants/#:~:text=Simple%20%2D%20they%20are%20incredibly%20helpful,traffic%20news%2C%20or%20ordering%20food)

[:~:text=Simple%20%2D%20they%20are%20incredibly%20helpful,traffic%20news%2C%20or%20orderi ng%20foodassistants/#:~:text=Simple%20%2D%20they%20are%20incredibly%20helpful,traffic%20ne ws%2C%20or%20ordering%20food.](https://www.phonexia.com/blog/the-future-of-voice-assistants/#:~:text=Simple%20%2D%20they%20are%20incredibly%20helpful,traffic%20news%2C%20or%20ordering%20food)

9.2 WEBSITE:

[www.researchgate.net](http://www.researchgate.net/)