AI DESKTOP ASSISTANT

A PROJECT REPORT

Submitted By

Aditya Agarwal
(University Roll No- 2100290140008)

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Mr. Ankit Verma
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(FEB 2023)

CERTIFICATE

Certified that **Aditya Agarwal (University Roll. No-2100290140008),** has carried out the project work having "**Title of Report AI DESKTOP ASSISTANT**" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself /herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Aditya Agarwal (University Roll No- 2100290140008)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Mr. Ankit Verma

Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Arun Kumar Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

In this modern era, day-to-day life became smarter and interlinked with technology. We already know some voice assistants like google and Siri. etc. Now in our voice assistance system, it can act as a basic weather forecast, daily schedule reminder, note writer, calculator, and search tool. This can also provide the functionality of chatbot and Face Mask Detection. This project works on voice input and gives output through voice and displays the text on the screen. The main agenda of our voice assistance makes people smart and give instant and computed results. The voice assistant takes the voice input through our microphone (Bluetooth and wired microphone) and it converts our voice into computer-understandable language and gives the required solutions and answers which are asked by the user. This assistance connects with the world wide web to provide results that the user has questioned. Natural Language Processing algorithm helps computer machines to engage in communication using natural human language in many forms.

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, **Mr. Ankit Verma Sir** for his guidance, help, and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi Sir** Professor and Head of the Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kinds of help. Without their support, the completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

ADITYA AGARWAL

LIST OF CHAPTERS

CHAPTER 1

INTRODUCTION

1.1 PROJECT DESCRIPTION

Today the development of artificial intelligence (AI) systems that can organize a natural human-machine interaction (through voice, communication, gestures, facial expressions, etc.) is gaining in popularity. One of the most studied and popular was the direction of interaction,based on the understanding of the machine by the machine of the natural human language. It is no longer a human who learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, and behavior and trying to becomehis personalized assistant.

Virtual assistants are software programs that help you ease your day-to-day tasks, such as showing weather reports, creating reminders, making shopping lists, etc. They can take commands via text (online chatbots) or by voice. Voice-based intelligent assistants need aninvoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple's Siri, Amazon's Alex, and Microsoft's Cortana. This assistant is designed to be used efficiently on desktops. Personal assistants' software improves user productivity by managing routine tasks of the user and by providing information from an online source to the user. This project was started on the premise that there is enough openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

Keywords: Virtual Assistant Using Python, AI, Digital assistance, Python.

1.2 PROJECT PURPOSE

The purpose of virtual assistant is to be 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 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 screeninteraction.

1.3 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 focuson 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.

1.4 OVERALL OBJECTIVE

The main objective of building personal assistant software (an ai desktop assistant) is using semantic data sources available on the web, user-generated content, and providing knowledge from knowledge databases.

The main purpose of an intelligent desktop assistant is to answer questions that users may have.This may be done in a business environment, for example, on a business website, with a chat interface. On the mobile platform, the intelligent virtual assistant is available as a call-button- operated service where a voice asks the user "What can I do for you?" and then responds to verbal input. Desktop assistants can tremendously save you time. We spend hours in online research and then making the report in our terms of understanding. An assistant can do that for you. Provide a topic for research and continue with your tasks while the assistant does the research. Another difficult task is to remember test dates, birthdates, or anniversaries. It comes with a surprise when you enter the class and realize it is a class test today. Just tell the assistant in advance about your tests and she remind you well in advance so you can prepare for the test.One of the main advantages of voice searches is their rapidity. In fact, voice is reputed to be four times faster than a written search: whereas we can write about 40 words per minute, we can speak around 150 during the same period of time15. In this respect, the ability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

1.5 HARDWARE & SOFTWARE SPECIFICATIONS

1.5.1 HARDWARE SPECIFICATIONS

- 500GB HDD and above
- 4GB RAM and above
- Camera and Microphone
- Processor core i3 and above

1.5.2 SOFTWARE SPECIFICATIONS

- Operating System: Window 7,8,10
- Python and Machine Learning
- Anaconda Software
- Jupyter Notebook

CHAPTER 2

LITERATURE REVIEW

2.1 ABSTRACT

The main goal of the smart desktop assistant is to decrease human efforts as nowadays everyone usually is in hurry and wants to be smart. A smart desktop assistant is a virtual assistant that will help the userto perform many tasks like searching on the internet, opening and closing tabs, moving the window to the left or right, and many more tasks that generally a user does while using a machine. The smart desktop assistant also decreases the gap between humans and technology as handicapped people or peoples that are physically challenged are not able to use the desktop machines properly. It will help them by just asking for the command that they want to do, and the desktop assistant will itself do the task for them.

There are many smart devices available in the market like Alexa, Siri, etc. but those all are applicable either for the mobiles or for home automation. Desktop assistant is a type of program that help the people who cannot afford expensive devices or are not able to use the complex devices. Desktop Assistant will be a game changer as it is the step towards the modernization as it includes AI and can be synced with the IOT also. It can be said as a step towards making the desktop smarter which has oldwindows. The informed searching technique In desktop assistants help them to do the task at a faster rate with the best result as the output.

2.2 INTRODUCTION

The AI Desktop Assistant is a computer software that is capable to do the general or daily work task without any physical interaction with the machine. The desktop assistant will be ablet to the task like telling the date and the time, temperature, opening and closing of windows, searching on the internet, moving the window to the left or to the right.

Major tasks that are supported by the assistant is that the user can schedule a WhatsApp message to anyone even after any period. The person can be either in the contact list or you just simply have to dictate his name or mobile number and the assistant will schedule a message for that person at the time you have allotted to it. Another major task which the assistant can do is taking of screenshot from the voice command and the taken screenshot will be saved at a specific position and the user can also name the screenshot according to its preference for better remembrance at the time oftaking the screenshot.

The desktop Assistant will help the user in many ways either in the term of reducing time or by notinvolving any physical interaction with the machine. It can be very impactful towards the capabilityand usage of any machine.

2.3 LITERATURE REVIEW

A smart assistant is made for the desktop or mobiles that have connectivity to the internet. The functionof the smart assistant is to perform the task without allowing the user to do any physical touch. The smart assistant takes the voice as the input and perform the desired tasks. We describe an intelligent personal assistant that has been developed to aid a busy knowledge worker in managing time commitments and performing tasks [1]. Desktop assistants are made to decrease the gap between the human and the technology and allow the differentially able people to interact with the technology easily.

Emerging technologies like virtual reality, augmented reality and voice interaction are reshaping the way people engage with the world and transforming digital experiences. Voice control is the next evolution of human-machine interaction, thanks to advances in cloud computing, Artificial Intelligence(AI) and the Internet of Things (IoT). In the last years, the heavy use of smartphones led to the appearance of voice assistants such as Apple’s Siri, Google’s Assistant, Microsoft’s Cortana and Amazon’s Alexa. Voice assistants use technologies like voice recognition, speech synthesis, and Natural Language Processing (NLP) to provide services to the users.[2]

2.4 FUNCTIONALITIES

The capabilities of the voice assistant cannot be understood as weak. Some of the major task categoriesof the desktop assistant are: -

- Answer to questions asked by users.
- Play music from streaming music services.
- Set timers or alarms.
- Automate home appliances.
- Make calls or send messages.
- Set Reminders
- Provide information about the weather.
- Control other smart devices (lights, locks, thermostats, vacuum cleaners, switches).

Nowadays voice assistant has become a common thing as it is available widely and comes integratedalong with the smart phones for the user, but still, many of the users are unable to use all the functionality of the assistant properly. The old age people can be considered as the group which still faces the difficulties however the younger ones can perform many tasks in comparison of the old age group peoples. The old age people are only able to do task like play songs or call someone whose contact are saved in their device. They should be telling the real power and all the capabilities of the assistant by which they will not feel lonely and can do many tasks which will provide them comfort.

2.5 WORKING

Previous studies have investigated the diversity of user requirements for various age groups. Arfi et al.provided two categories of consumers: IoT natives (Gen Z, born in the new century and considered IoTnatives) and IoT immigrants (the other generations). The perceived risk has no impact on IoT natives; however, IoT immigrants are more likely to consider the perceived risk. A study on the user acceptance of smart bracelets by younger and older adults showed that the requirements of older adults were strongly linked to their health status. Older adults are less concerned about privacy issues and more likely to share with their family members than their younger counterparts [3].

The below diagram will show you the functionality of the known smart assistant Alexa that how tospeech is recognized and how tha data is processed.

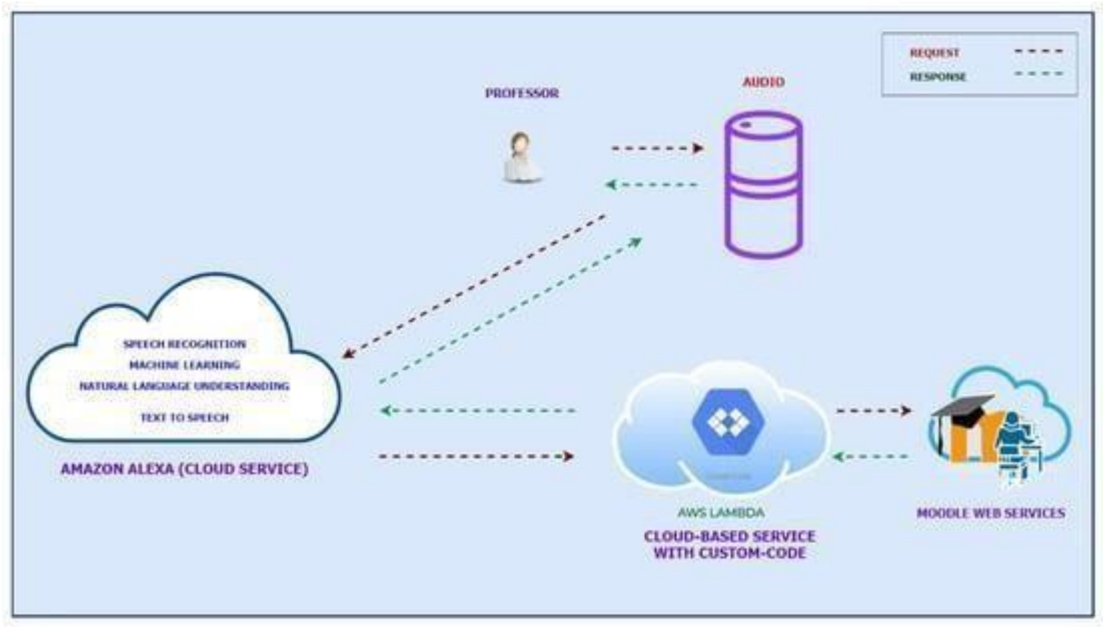


Fig 1. Diagram of the operation of the "ALEXA" application

2.6 CAPABILITIES

In this paper, we aim to answer these questions: 1) How do older adults who do not regularly use a computing device perceive intelligent voice assistants embodied in smart speakers? 2) What do theyuse these devices for? 3) What challenges arise from the use of these systems and how can those challenges be addressed? To answer these questions, we conducted a study by deploying Amazon Echo Dot devices – a smart speaker with the Alexa voice assistant – in seven households with older adults who used digital technology infrequently. We studied participants' usage over a period of threeweeks using a range of data sources, including usage logs from the paired Amazon Alexa app, self- reported data from semi-structured interviews, and daily diary entries.[4].

2.7 BENEFITS

Three main benefits that arose were efficiency, impacts on independence, and an ability to replace a range of other technologies. Toward the theme of efficiency, seven participants mentioned that the device had enabled them to perform tasks faster than before, such as online shopping, checking the weather, listening to news, playing music, and setting timers. For example, P11 said that compared to using a traditional browser, Alexa is "able to accomplish [making a purchase] in seconds versus a few. minutes." Four participants also referred to the IPAs as enabling them to multitask in new ways. P15, for example, felt that the voice interface was easier than using a smartphone to set a timer while cooking because it was hands-free: "I think as a blind person, you tend to get your hands messier thanperhaps some sighted people do." [5].

The desktop assistant making is not a hard task integrating a set of code and combining them all together built a working assistant, but the main task is to automate it and design its interface because all of the things will be depending upon these two factors. The different assistants have different capabilities and one of the ways the companies are differentiating is in the interaction techniques theirassistants that are controlled with, most uses voice, and some uses text [6].

2.8 SCOPE

The quickly adapting of the desktop assistants are showing the path towards the technology adaptationand how people are becoming smarter. The popularity of the voice assistants has increased day by day and there are matching the level of performance that the user has expected. The most quickly adoptedtechnology is the virtual assistant. From smartwatches to smart speakers, all products now have a virtual voice recognition model built in. The future has numerous potentials for voice technology to advance. However, there are still adjustments and advancements to be made in this subject. The existing system's understanding has to be greatly improved.

Virtual assistants using Artificial Intelligence, such as Machine Learning, Neural Networks, and IoT, will be the future of these helpers. By adopting this technology, we will be able to achieve new heights. Virtual assistants have the potential to achieve far more than we have so far [7].

2.9 CONCLUSION

The conclusion which can be calculated from the above review is that adapting and using of assistants will help an individual in many ways. The most quickly adopted technology is the virtual assistant.

From smartwatches to smart speakers, all products now have a virtual voice recognition model built in. The future has numerous potentials for voice technology to advance. However, there are still adjustments and advancements to be made in this subject. The existing system's understanding must be greatly improved. Virtual assistants using Artificial Intelligence, such as Machine Learning, Neural Networks, and IoT, will be the future of these helpers. By adopting this technology, we will be able to achieve new heights. Virtual assistants have the potential to achieve far more than we have so far [8].

2.10 REFERENCES

[1] Karen Myers1 Pauline Berry1 Jim Blythe2 Ken Conley1 Melinda Gervasio1 Deborah McGuinness3 David Morley1 Avi Pfeffer4 Martha Pollack5 Milind Tambe " An Intelligent PersonalAssistant for Task and Time Management".

[2] George TERZOPOULOS, Maya SATRATZEMI " Voice Assistants and Smart Speakers inEveryday Life and in Education".

[3] Runting Zhong, Mengyao Ma, Yutong Zhou, Qingxia Lin, Leiling Li & Nengjing Zhang "Useracceptance of smart home voice assistant: a comparison among younger, middle-aged, and older adults".

[4] ALISHA PRADHAN, AMANDA LAZAR, LEAH FINDLATER, "Use of Intelligent VoiceAssistants by Older Adults with Low Technology Use".

[5] Dr. Amanda Lazar, Assistant Professor, " EXPLORING THE ACCESSIBILITY OF HOME-BASED, VOICE-CONTROLLED INTELLIGENT PERSONAL ASSISTANTS".

[6] Dr. Amanda Lazar, Assistant Professor, " EXPLORING THE ACCESSIBILITY OF HOME-BASED, VOICE-CONTROLLED INTELLIGENT PERSONAL ASSISTANTS".

[7] Smita Srivastava, Dr. Devesh Katiyar , Mr. Gaurav Goel " Desktop Virtual Assistant".

[8] Deepak Shende, Ria Umahiya, Monika Raghorte, Aishwarya Bhisikar,AnupBhange "AI BasedVoice Assistant Using Python".

CHAPTER 3

FEASIBILITY STUDY

A feasibility analysis is used to determine the viability of an idea, such as ensuring a project islegally and technically feasible as well as economically justifiable. It tells us whether a projectis worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable. A well-designed study should offer a historical background of the business or project, such as adescription of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

3.1 TECHNICAL FEASIBILITY

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team can convert the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. It includes finding out technologies for the project, both hardware and software. For a virtual assistant, the user must have a microphone to convey their message and a speaker to listen when the system speaks. These are very cheap nowadays and everyone generally possesses them. Besides, the system needs an internet connection. While using the assistant, make sure you have a steady internet connection. It is also not an issue in this era where almost every home oroffice has Wi-Fi.

3.2 OPERATIONAL FEASIBILITY

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development. It is the ease and simplicity of operation of the proposed system. The system does not require any special skill set for users to operate it. This shows the management and organizational structure of the project. This project is not built by a team. The management tasks are all to be carried out by a single person. That won't create any management issues and will increase the feasibility of the project.

3.3 ECONOMICAL FEASIBILITY

In the Economic Feasibility study cost and benefit of the project are analyzed. This means under this feasibility study a detailed analysis is carried out will be the cost of the project for development which includes all required costs final development hardware and software resources required, design and development costs and operational costs, and so on.

After that, it is analyzed whether the project will be beneficial in terms of finance for the organization or not. we find the total cost and benefit of the proposed system over the current system. For this project, the main cost is the documentation cost. Users also would have to pay for the microphone and speakers. Again, they are cheap and available. As far as maintenance is concerned, Assistant won't cost too much.

3.4 BEHAVIOURAL FEASIBILITY

It evaluates and estimates the user attitude or behavior toward the development of the new system. It helps in determining if the system requires special effort to educate, retrain, transfer, and change an employee's job status on new ways of conducting business. Establishing the cost-effectiveness of the proposed system i.e., if the benefits do not outweigh the costs, then it is not worth going ahead. In the fast-paced world today there is a great need for online social networking facilities. Thus, the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

CHAPTER 4

DATABASE DESIGN

A properly designed database provides you with access to up-to-date, accurate information. Because a correct design is essential to achieving your goals in working with a database, investing the time required to learn the principles of good design makes sense. Inthe end, you are much more likely to end up with a database that meets your needs and can easily accommodate change.

This article provides guidelines for planning a desktop database. You will learn how to decide what information you need, how to divide that information into the appropriate tables and columns, and how those tables relate to each other. You should read this article before you create your first desktop database.

Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of an enterprise data management system. Designing a proper database reduces the maintenance cost therebyimproving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept for designing a database. The designer should follow the constraints and decide how the elements correlate and what kind ofdata must be stored.

The main objectives behind database designing are to produce physical and logical design models of the proposed database system. To elaborate on this, the logical model is primarily concentrated on the requirements of data and the considerations must be made in terms of monolithic considerations hence the stored physical data must be stored independent of the physical conditions. On the other hand, the physical database design model includes a translation of the logical design model of the database by keeping control of physical media using hardware resources and software systems such as Database Management System (DBMS).

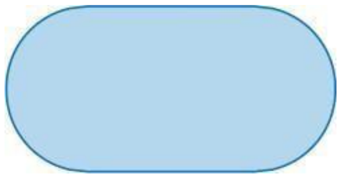
4.1 FLOW CHART

A flowchart is a graphical representation of an algorithm. Programmers often use it as aprogram-planning tool to solve a problem. It makes use of symbols that are connected among them to indicate the flow of information and processing.

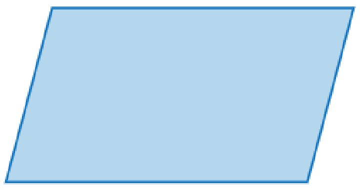
The process of drawing a flowchart for an algorithm is known as “flowcharting”.

4.1.1 Basic Symbols used in Flowchart Designs

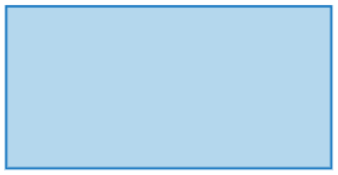
- **Terminal:** The oval symbol indicates Start, Stop, and Halt in a program’s logic flow. A pause/halt is generally used in a program logic under some error conditions. The terminal is the first and last symbol in the flowchart.



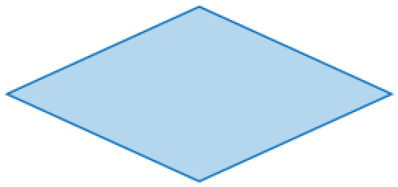
- **Input/Output:** A parallelogram denotes any function of input/output type. Program instructions that take input from input devices and display output on output devices areindicated with parallelograms in a flowchart.



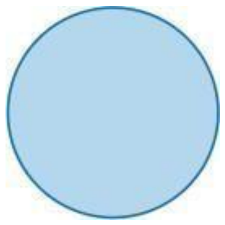
- **Processing:** A box represents arithmetic instructions. All arithmetic processes such as adding, subtracting, multiplication, and division are indicated by action or process symbol.



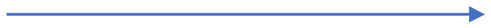
- **Decision** Diamond symbol represents a decision point. Decision-based operations such as yes/no questions or true/false are indicated by diamonds in the flowchart.



- **Connectors:** Whenever the flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusions. It is represented by a circle.



- **Flow lines:** Flow lines indicate the exact sequence in which instructions are executed. Arrows represent the direction of the flow of control and the relationship among different symbols of the flowchart.



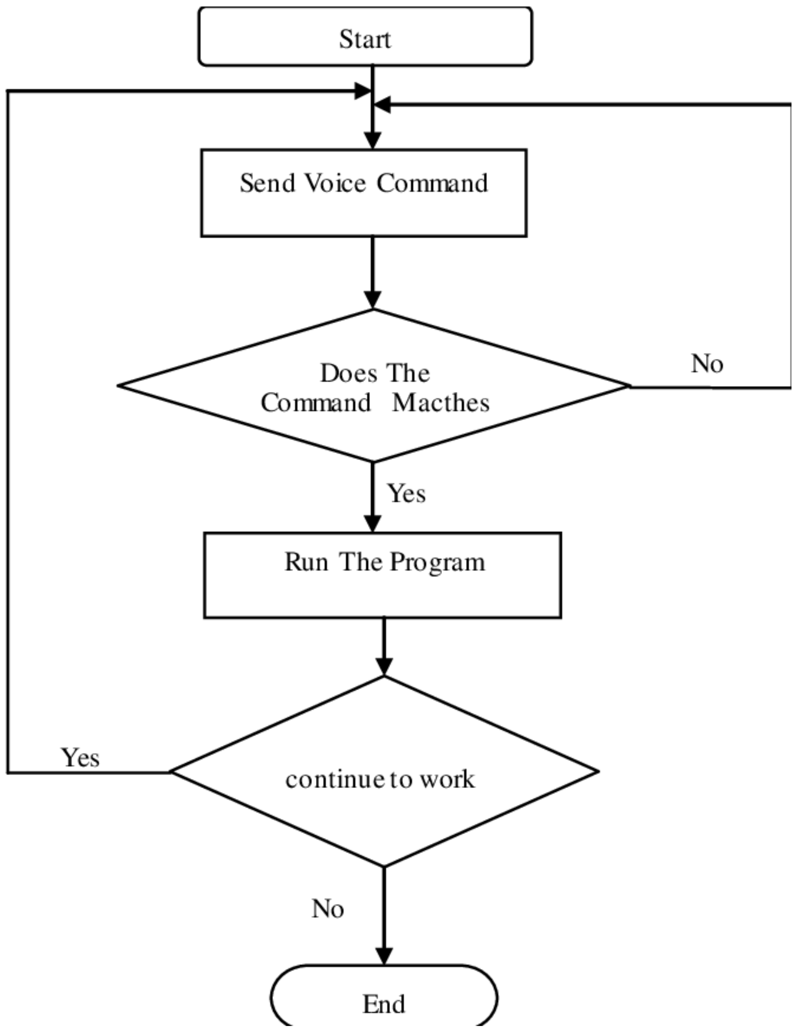


Fig 4.1: Flow Chart for Desktop Assistant

4.2 USE CASE DIAGRAM

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

The purpose of a use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as the other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will investigate some specific purpose, which will distinguish it from the other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a systemis analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modeled to present the outside view.

4.2.1 In brief, the purposes of use case diagrams can be said to be as follows −

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements actors.

4.2.2 Use case diagram components.

To answer the question, "What is a use case diagram?" you need to first understand itsbuilding blocks. Common components include:

- **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

- **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

- **Goals:** The result of most use cases. A successful diagram should describe theactivities and variants used to reach the goal.

4.2.3 Use case diagram symbols and notation.

The notation for a use case diagram is straight forward and doesn't involve as manytypes of symbols as other UML diagrams.

4.2.4 Use cases

Horizontally shaped ovals that represent the different uses that a user might have.

- **Actors:** Stick figures that represent the people employing the use cases.
- **Associations:** A line between actors and use cases. In complex diagrams, it is important toknow which actors are associated with which use cases.
- **System boundary boxes:** A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, PsychoKiller is outside the scope of occupations in the chainsaw example found below.
- **Packages:** A UML shape that allows you to put different elements into groups. Just as with component diagrams, these groupings are represented as file folders.

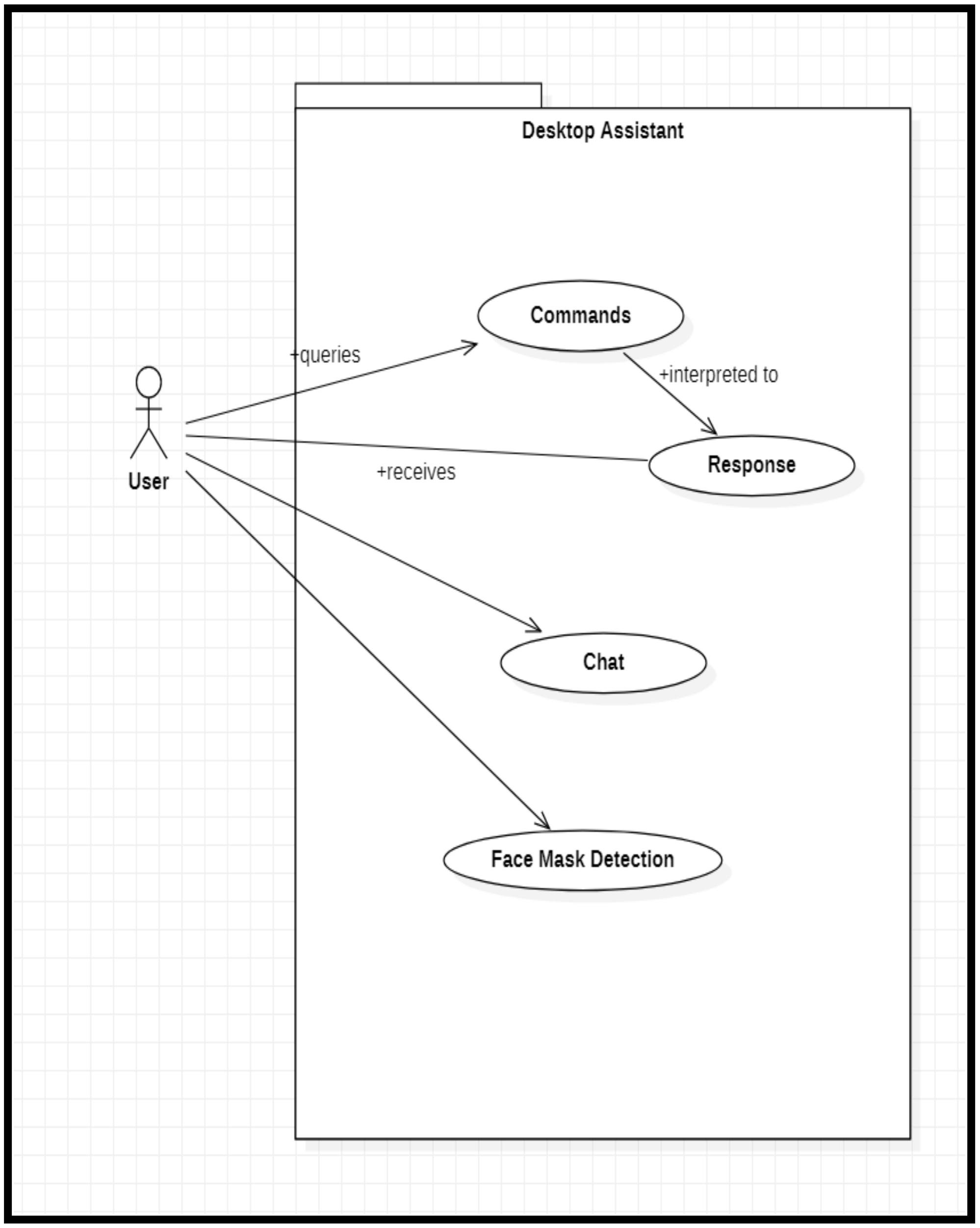


Fig 4.2: Use Case Diagram of Desktop Assistant

4.3 SEQUENCE DIAGRAM

A **sequence diagram** or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

For a particular scenario of a use case, the diagrams show the events that external actors generate, their order, and possible inter-system events. All systems are treated as a black box; the diagram places emphasis on events that cross the system boundary from actors to systems. A system sequence diagram should be done for the main success scenario of the use case, and frequent or complex alternative scenarios.

- To model high-level interaction among active objects within a system.
- To model interaction among objects inside realizing a use case.
- It either models' generic interactions or some certain instances.

4.3.1 Sequence Diagram Notations –

- **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.
- **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.
- **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages usingarrows. Lifelines and messages form the core of a sequence diagram.
- **Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guardsplay an important role

in letting software developers know the constraints attached to a system or a particular process.

4.3.2 Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation, orprocedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future. systems.
- Visualize how messages and tasks move between objects or components in asystem.

4.3.3 For Task Execution:

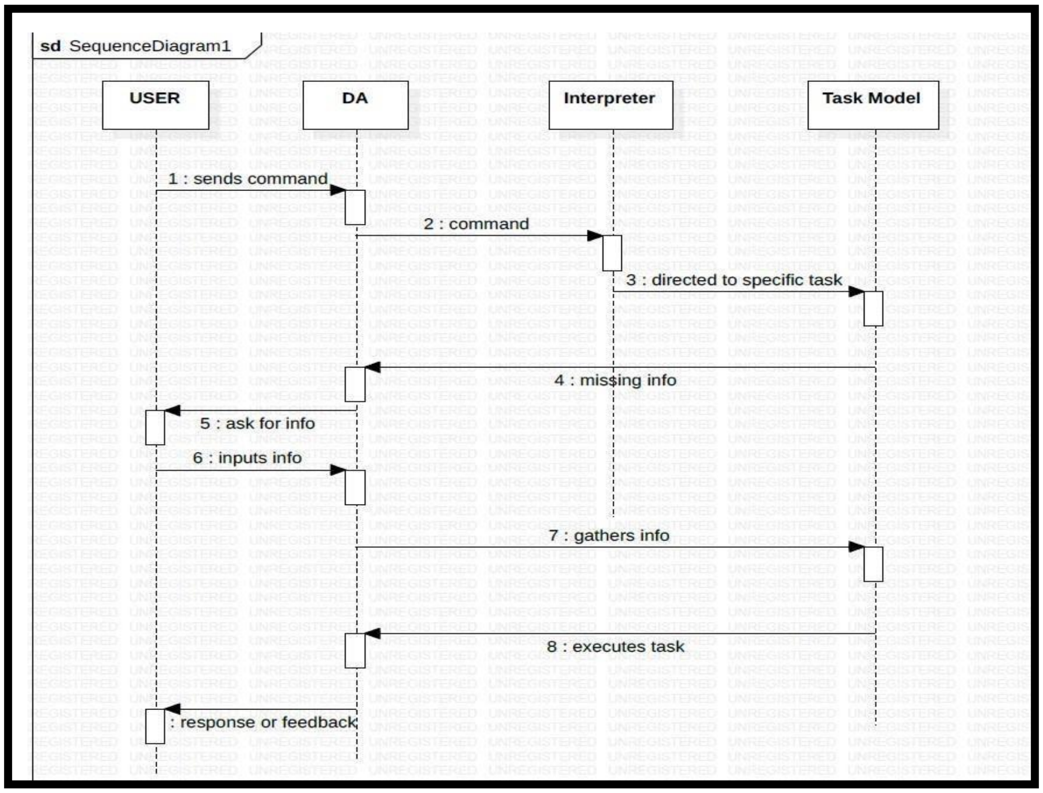


Fig 4.3.3.1: Sequence Diagram for Task Execution

4.3.4 For Query Response:

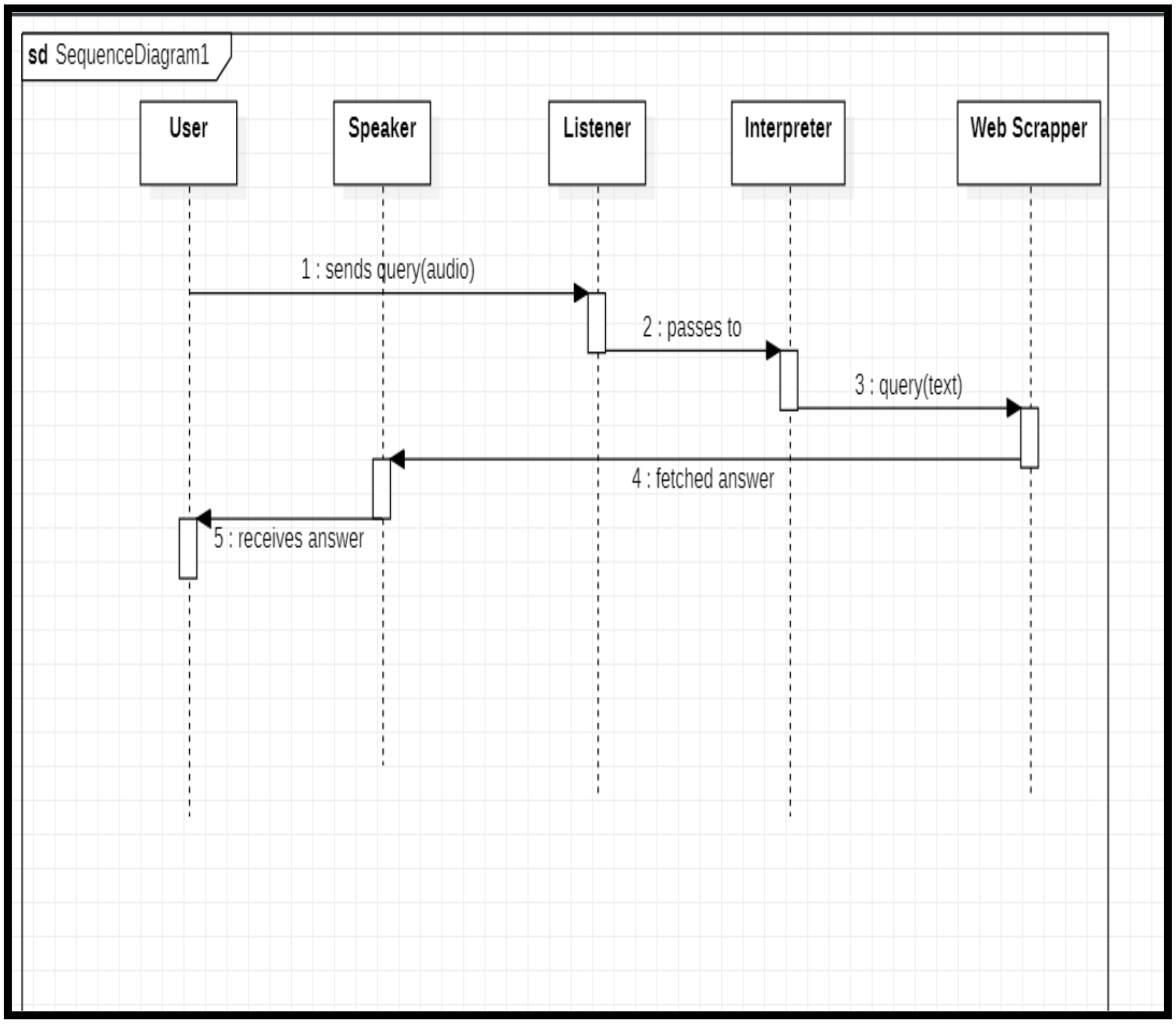


Fig 4.3.4: Sequence Diagram for Query Response

4.3.5 For Chatbot:

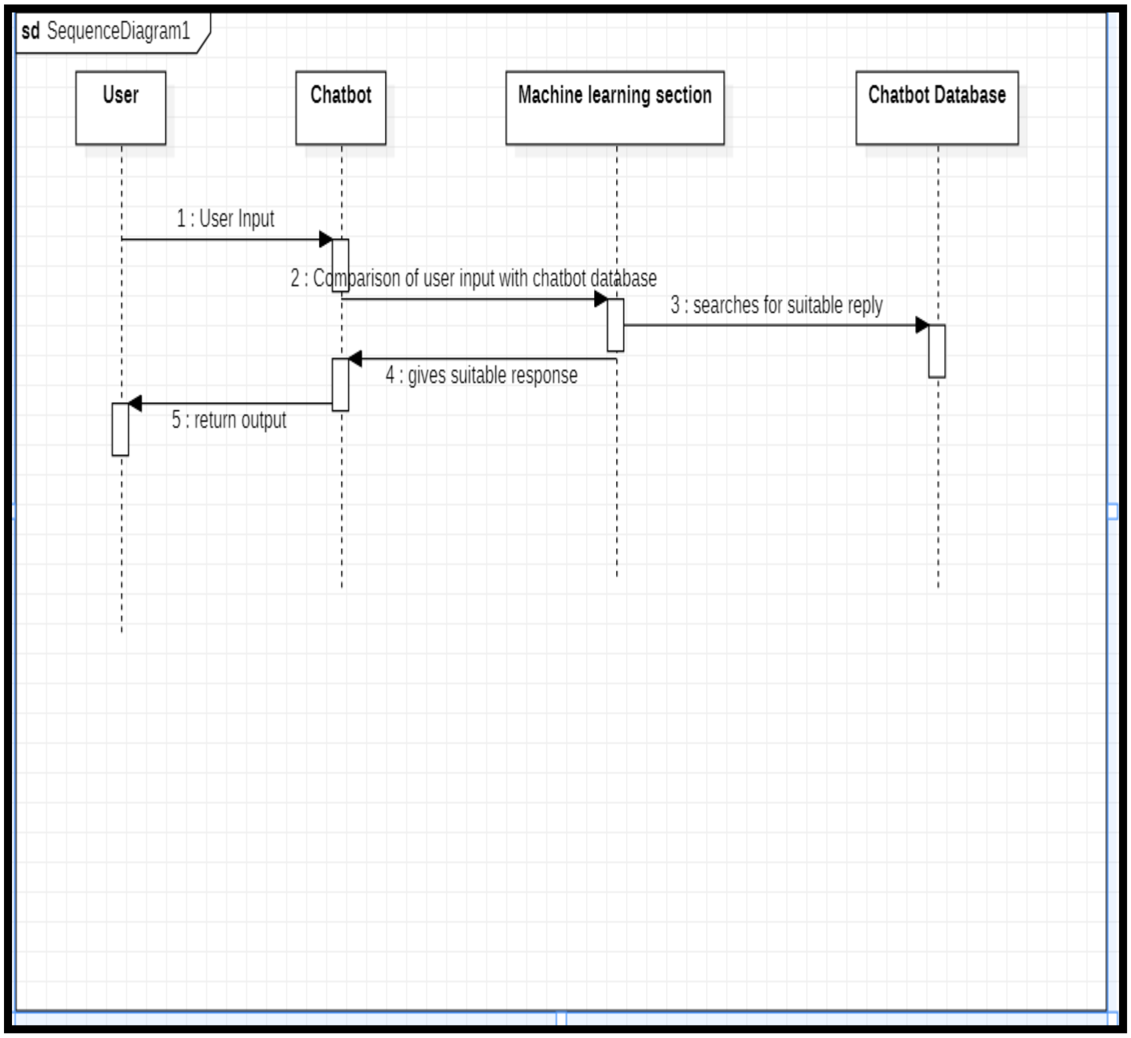

sd SequenceDiagram1
User
Chatbot
Machine learning section
Chatbot Database
1 : User Input
2 : Comparison of user input with chatbot database
3 : searches for suitable reply
4 : gives suitable response
5 : return output

Fig 4.3.5: Sequence Diagram for chatbot

4.3.6 For Face Mask Detection:

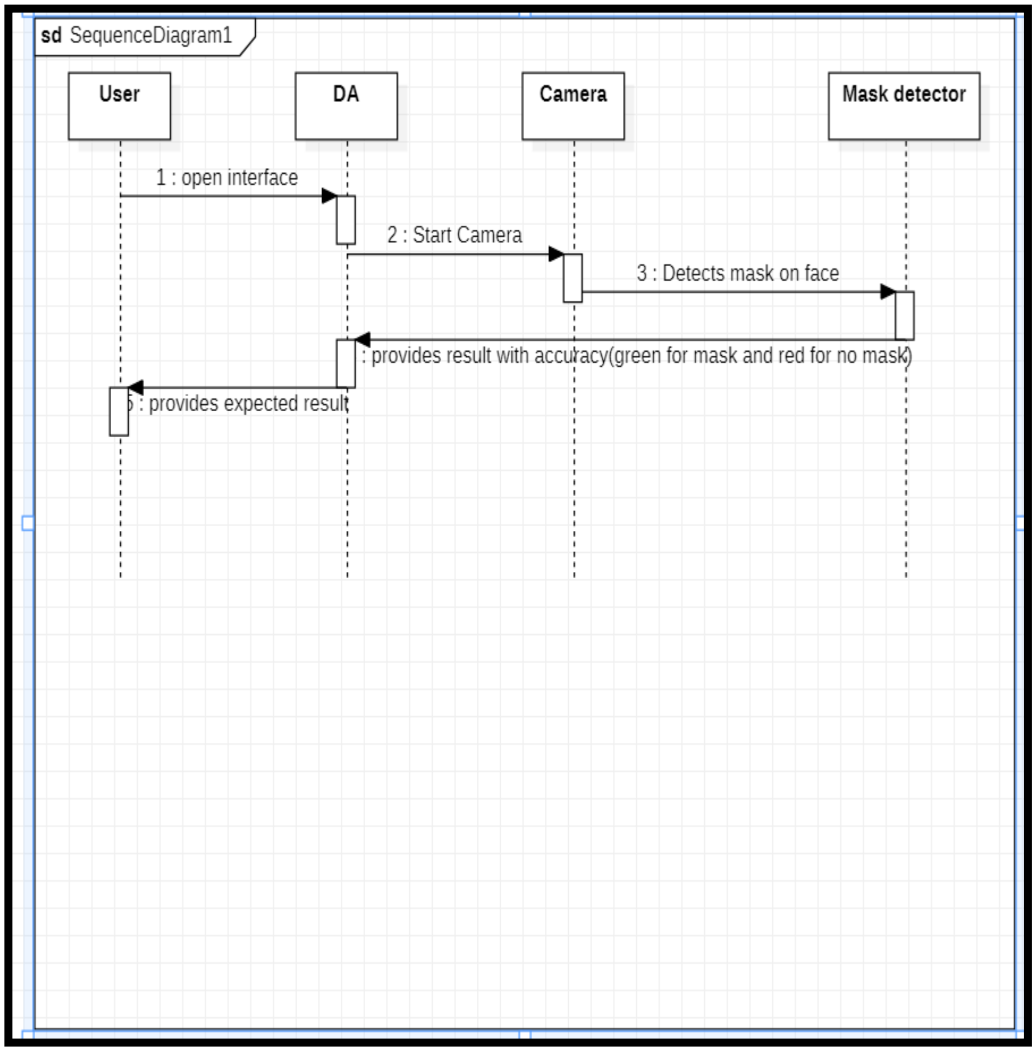


Fig 4.3.4: Sequence Diagram for Face Mask Detection

4.4 ACTIVITY DIAGRAM

The activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.

The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc.

It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

4.4.1 Components of an Activity Diagram

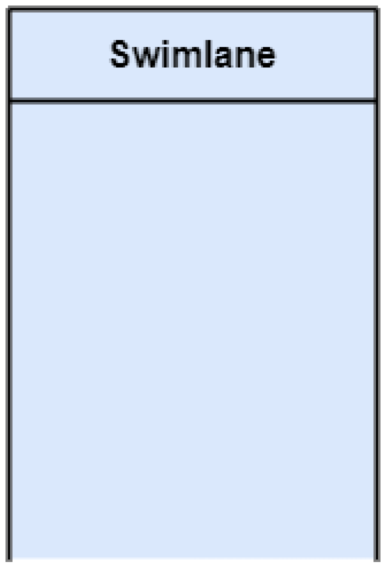
Following are the component of an activity diagram:

4.4.1 Activities

The categorization of behavior into one or more actions is termed as an activity. In other words, it can be said that an activity is a network of nodes that are connected by edges. The edges depict the flow of execution. It may contain action nodes, control nodes, or object nodes. The control flow of activity is represented by control nodes and object nodes that illustrates the objects used within an activity. The activities are initiated at the initial node and are terminated atthe final node.

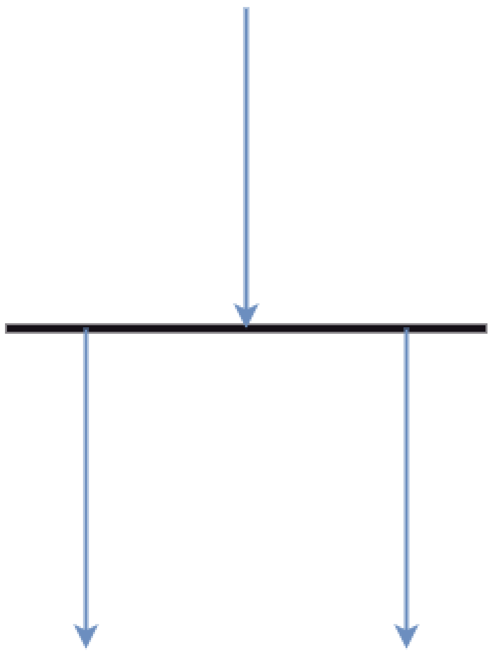
4.4.2 Activity partition /swim lane

The swim lane is used to cluster all the related activities in one column or one row. It can be either vertical or horizontal. It used to add modularity to the activity diagram. It is not necessary to incorporate swim lane in the activity diagram. But it is used to add more transparency to the activitydiagram.



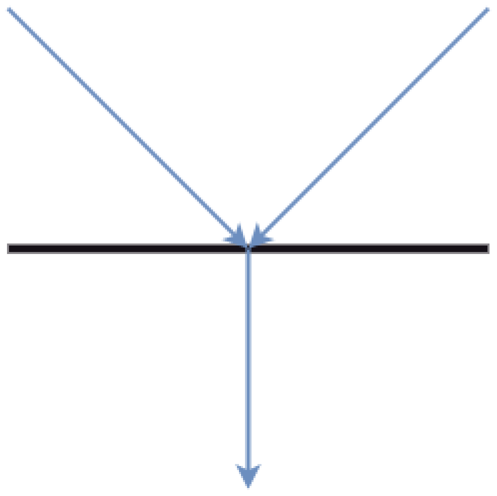
4.4.3 Forks

Forks and join nodes generate the concurrent flow inside the activity. A fork node consists of one inward edge and several outward edges. It is the same as that of various decision parameters. Whenever a data is received at an inward edge, it gets copied and split crossways various outward edges. It split a single inward flow into multiple parallel flows.



4.4.4 Join Nodes

Join nodes are the opposite of fork nodes. A Logical AND operation is performed on all of the inward edges as it synchronizes the flow of input across one single output (outward) edge.



4.4.5 Pins

It is a small rectangle, which is attached to the action rectangle. It clears out all the messy and complicated thing to manage the execution flow of activities. It is an object node that precisely represents one input to or output from the action.

4.4.6 Notation of an Activity diagram

- **Initial State:** It depicts the initial stage or beginning of the set of actions.
- **Final State:** It is the stage where all the control flows and object flows end.
- **Decision Box:** It makes sure that the control flow or object flow will follow. only one path.
- **Action Box:** It represents the set of actions that are to be performed.

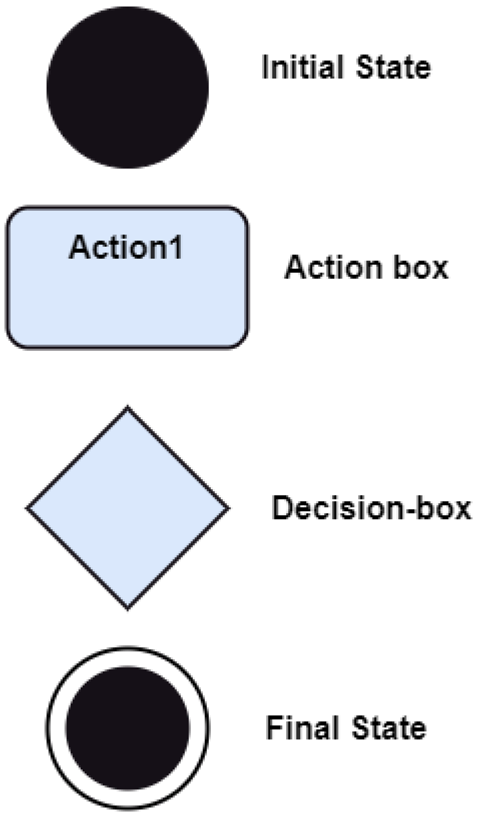


Fig 4.4.6.1 Activity Diagram

An event is created as an activity diagram encompassing a group of nodes associated with edges. To model the behavior of activities, they can be attached to any modeling element. It can model use cases, classes, interfaces, components, and collaborations.

It mainly models processes and workflows. It envisions the dynamic behavior of the system as well as constructs a runnable system that incorporates forward and reverse engineering. It does not include the message part, which means message flow is not representedin an activity diagram.

It is the same as that of a flowchart but not exactly a flowchart itself. It is used to depictthe flow between several activities.

4.4.7 How to draw an Activity Diagram?

An activity diagram is a flowchart of activities, as it represents the workflow among various activities. They are identical, but they themselves are not exactly theflowchart. In other words, it can be said that an activity diagram is an enhancement of the flowchart, which encompasses several unique skills.

Since it incorporates swim lanes, branching, parallel flows, join nodes, control nodes, and forks, it supports exception handling. A system must be explored before drawing an activity diagram to provide a clearer view of the user. All of the activities are explored after they are properly analyzed for finding out the constraints applied to the activities. Each activity, condition, and association must be recognized. After gathering all the essential information, an abstract or a prototype is built, which is then transformed into the actual diagram.

4.4.8 Following are the rules that are to be followed for drawing an activity. diagram:

- A meaningful name should be given to each activity.
- Identify all the constraints.
- Acknowledge the activity associations.

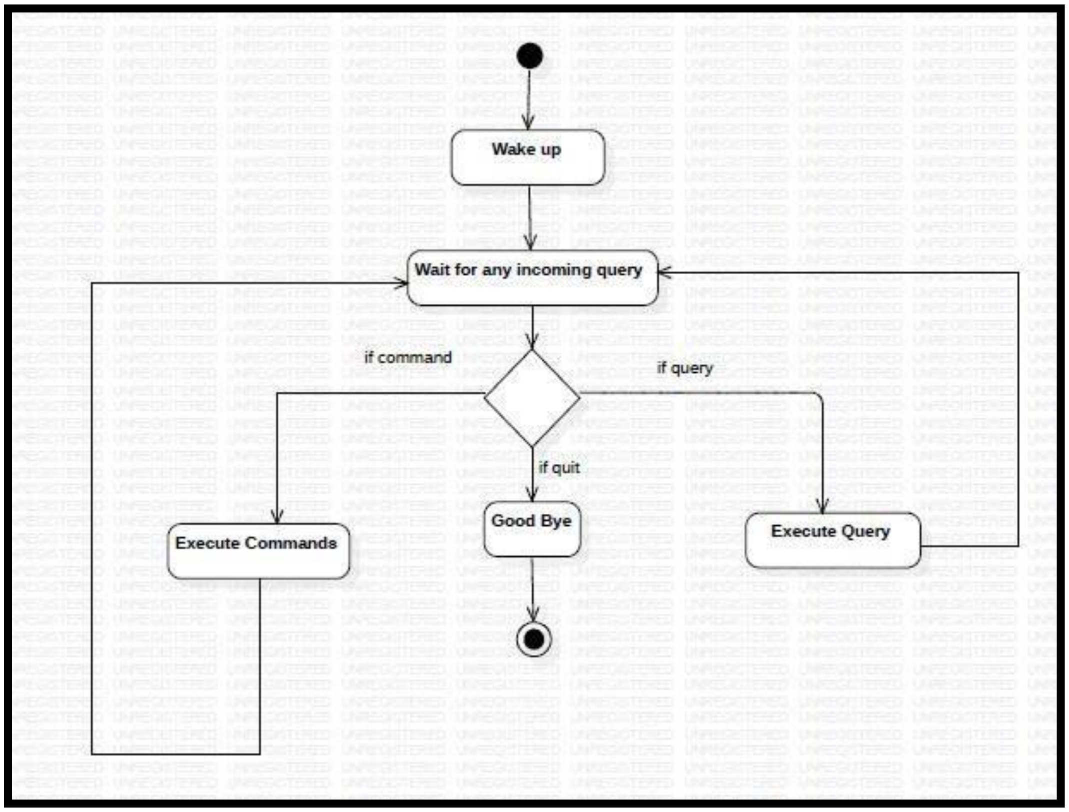


Fig 4.4.8.1: Activity Diagram of Desktop Assistant

4.5 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 it may inherit from other classes. A class diagram is used to visualize, describe, document various aspects of the system, and 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 helpsin software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

4.5.1 Purpose of Class Diagrams

The main purpose of class diagrams is to build a static view of an application. It is the only diagram that is widely used for construction, and it can be mapped with object- oriented languages. It is one of the most popular UML diagrams. Following are the purpose of class diagrams given below:

- It analyses and designs a static view of an application.
- It describes the major responsibilities of a system.
- It is a base for component and deployment diagrams.
- It incorporates forward and reverse engineering.

4.5.2 Benefits of Class Diagrams

- It can represent the object model for complex systems.
- It reduces the maintenance time.
- It provides a general schematic of an application.
- It represents a detailed chart by highlighting the desired code.
- It is helpful for the stakeholders and the developers.

4.5.3 How to draw a Class Diagram?

The class diagram is used most widely to construct software applications. It not only represents a static view of the system but also all the major aspects of an application. A collection of class diagrams represents a system.

Some key points that are needed to keep in mind while drawing a class diagram are given below:

- To describe a complete aspect of the system, it is suggested to give a meaningful name to theclass diagram.
- The objects and their relationships should be acknowledged in advance.
- The attributes and methods (responsibilities) of each class must be known.
- A minimum number of desired properties should be specified as a greater number of the unwanted property will lead to a complex diagram.
- Notes can be used as and when required by the developer to describe the aspects of a diagram.
- The diagrams should be redrawn and reworked as many times as possible to make it correct beforeproducing its final version.

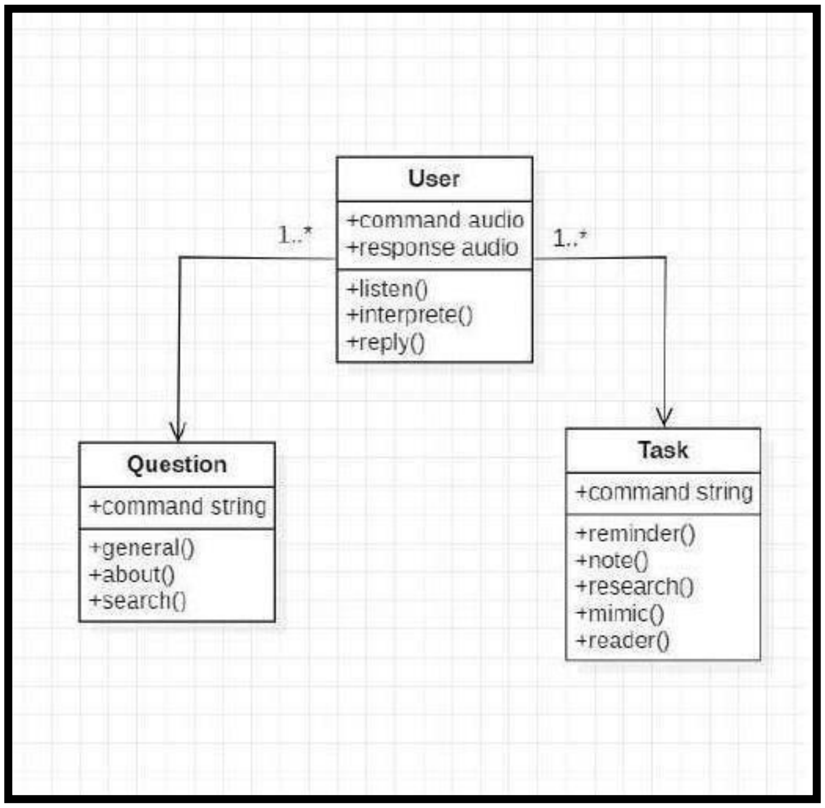


Fig 4.5.3.1: Class Diagram of Desktop Assistant

4.6 STATE DIAGRAM

The state machine diagram is also called the State chart or State Transition diagram, which shows the order of states underwent by an object within the system. It captures the software system's behavior. It models the behavior of a class, a subsystem, a package, and a complete system.

It tends out to be an efficient way of modeling the interactions and collaborations in the external entities and the system. It models event-based systems to handle the state of an object. It also defines several distinct states of a component within the system. Each object/component has a specific state.

4.6.1 Notation of a State Machine Diagram

Following are the notations of a state machine diagram enlisted below:

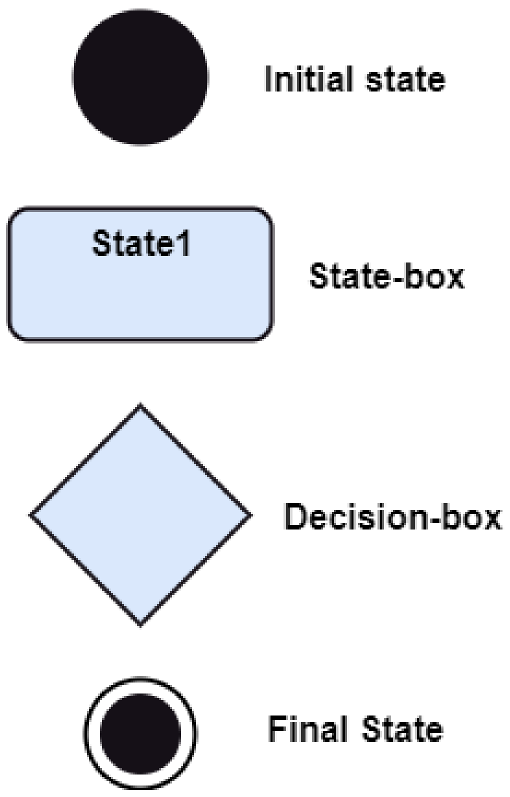


Fig 4.6.1.1 State Machine Diagram

- **Initial state:** It defines the initial state (beginning) of a system, and it is represented by ablack filled circle.
- **Final state:** It represents the final state (end) of a system. It is denoted.by filled circlepresent within a circle.
- **Decision box:** It is of diamond shape that represents the decisions.to be made based on an evaluated guard.
- **Transition:** A change of control from one state to another due to the occurrence of some event is termed as a transition. It is represented by an arrow labeled with an event due to which the change has ensued.
- **State box:** It depicts the conditions or circumstances of a particular object of a class at a specific point of time. A rectangle with round corners is used to represent the state box.

4.6.2 How to Draw a State Machine Diagram?

The state machine diagram is used to portray various states underwent by an object. The change in one state to another is due to the occurrence of some event. All of the possible states of a particular component must be identified before drawing a state machine diagram.

The primary focus of the state machine diagram is to depict the states of a system. These states areessential while drawing a state transition diagram. The objects, states, and events due to which the state transition occurs must be acknowledged before the implementation of a state machine diagram.

4.6.3 Following are the steps that are to be incorporated while drawing a state machine diagram:

- A unique and understandable name should be assigned to the state transition that describesthe behavior of the system.
- Out of multiple objects, only the essential objects are implemented.
- A proper name should be given to the events and the transitions.

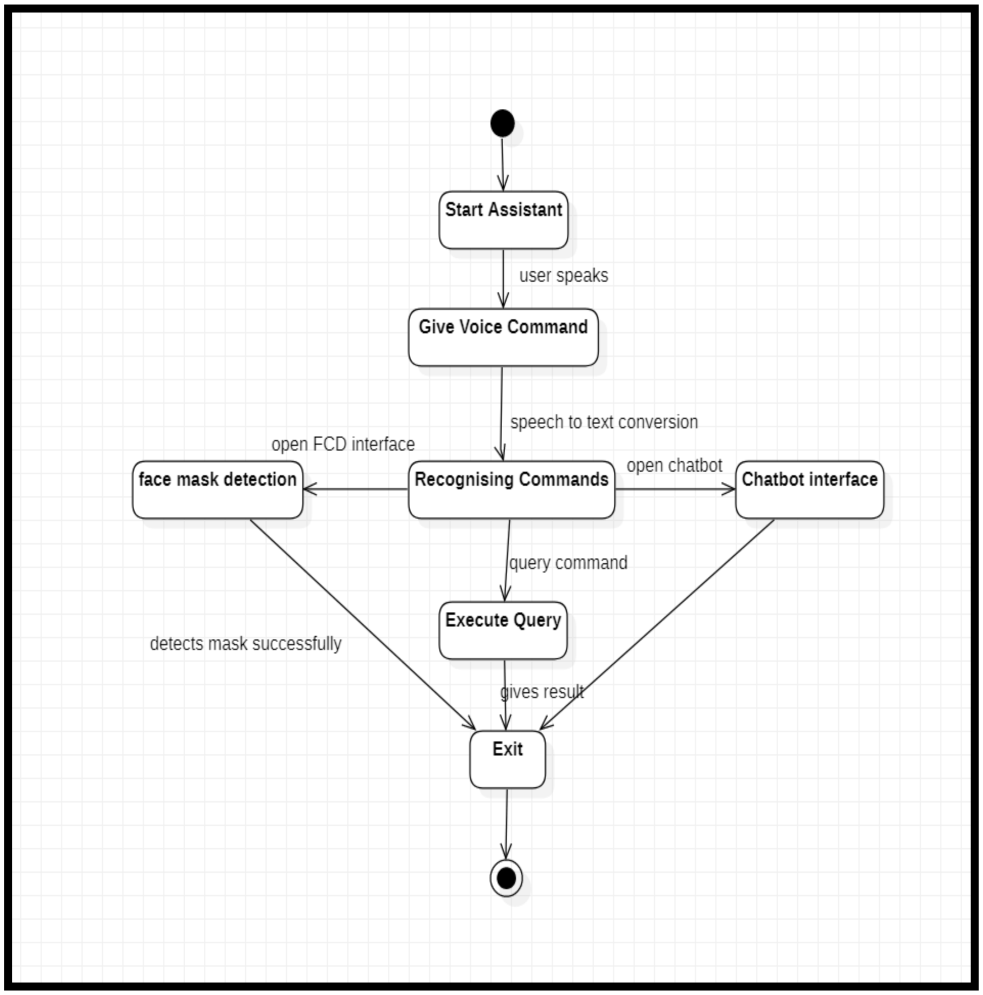


Fig 4.6.3.1: State Chart Diagram of Desktop Assistant

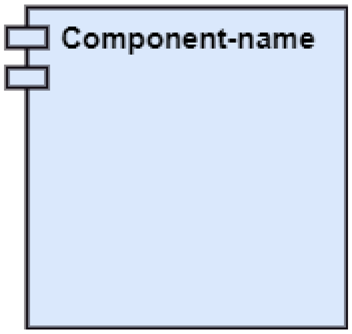
4.7 COMPONENT DIAGRAM

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.

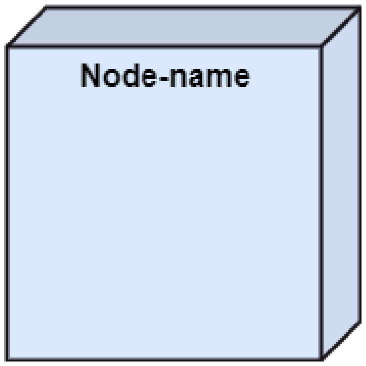
It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.

Notation of a Component Diagram

- A component



- A node



4.7.1 How to Draw a Component Diagram?

The component diagram is helpful in representing the physical aspects of a system, which are files, executables, libraries, etc. The main purpose of a component diagram is different from that of otherdiagrams. It is utilized in the implementation phase of any application.

Once the system is designed employing different UML diagrams, and the artifacts are prepared, the component diagram is used to get an idea of implementation. It plays an essential role in implementing applications efficiently.

4.7.2 Following are some artifacts that are needed to be identified before drawing a component diagram:

- What files are used inside the system?
- What is the application of relevant libraries and artifacts?
- What is the relationship between the artifacts?

4.7.3 Following are some points that are needed to be kept in mind after the artifacts are identified:

- Using a meaningful name to ascertain the component for which the diagram is about to bedrawn.
- Before producing the required tools, a mental layout is to be made.
- To clarify the important points, notes can be incorporated.

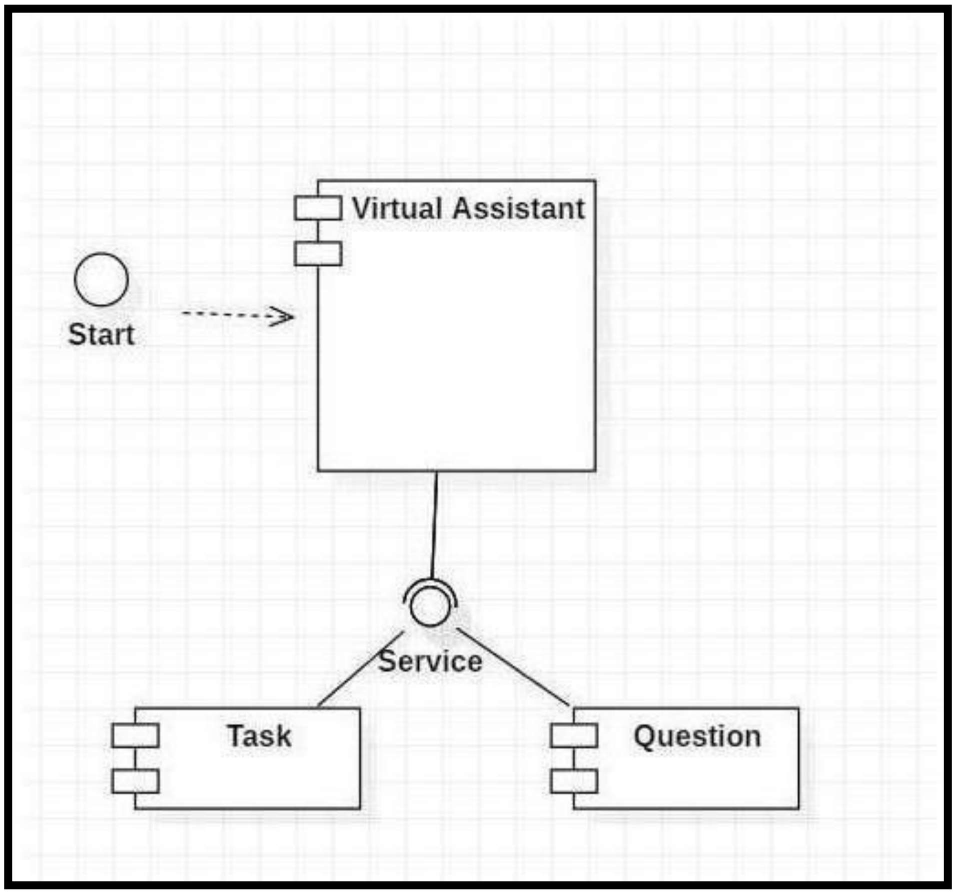


Fig 4.7.3.1: Component Diagram of Desktop Assistant

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages. Python provides a huge list of benefitsto all.

The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For desktop assistant, libraries used are speech recognition to recognize voice, Pyttsx for text to speech, selenium for web automation etc. Pythonis reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize)than writing everything in a low-level language.

5.2 Artificial Intelligence

In today's world, technology is growing very fast, and we are getting in touch with different new technologies day by day.

Here, one of the booming technologies of computer science is Artificial Intelligence which is ready to create a new revolution in the world by making intelligent machines. The Artificial Intelligence is now all around us. It is currently working with a variety of subfields, ranging from general to specific, such as self-driving cars, playing chess, playing music, Painting, etc.

AI is one of the fascinating and universal fields of Computer science which has a great scope in future. AI holds a tendency to cause a machine to work as a human. Artificial Intelligence is composed of two words **Artificial** and **Intelligence**, where Artificial defines *"man-made,"* and intelligence defines *"thinking power"*, hence AI means *"a man-made thinking power.*

5.3 Machine Learning

Machine Learning (ML) is an area of computer science that "gives computers the ability to learn without being explicitly programmed". The parameter of the formulas is calculated from the data, rather than defined by the programmer. Two most common usage of ML is Classification and Regression. Classification means to categorize different types of data, while Regression means to find a way to describe the data. Basic ML program will have two stages, fitting and predicting. In the fitting stage, the program will be givena large set (at least thousands) of data. The program will try to adjust its parameter based on some statistical models, in order to make it "fit" the input data best. In the predicting stage, the program will givea prediction for a new input based on the parameters it just calculated out. For example, the famous Iris flower dataset contains the measurement of several features of three different species of flowers, such asthe length of sepals and petals. A well-defined ML program can learn the pattern behind this feature and give prediction accordingly.

5.4 Modules used in Virtual Assistant

5.4.1 pyttsx3

Pyttsx stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to- speech synthesis. It is a Python package supporting common text-to-speech engines on MacOS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

5.4.2 Speech Recognition

This is a library for performing speech recognition, with support for several engines and APIs,online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc. Speech recognition helps us to save time by speakinginstead of typing. It gives the power to communicate devices without writing one line of code.

5.4.3 Playsound

The playsound module is a cross platform module that can play audio files. This doesn't have any dependencies, simply install with pip in your victualing and run! Implementation is different on platforms. It uses Windell. win on Windows, AppKit.NSSound on Apple OS X and GStreamer on Linux.

5.4.4 pynput

This library allows you to control and monitor input devices. Currently, mouse and keyboardinput and monitoring are supported.

5.4.5 wordtodigits

To convert spoken words into digits.

5.4.6 Pydictionary

PyDictionary is a Dictionary Module for Python 2/3 to get meanings, translations, synonyms, and Antonyms of words. It uses WordNet for getting meanings, Google for translations, and synonym.com for getting synonyms and antonyms.

5.4.7 keyboard

Take full control of your keyboard with this small Python library. Hook global events, registerhotkeys, simulate key presses aIt helps to enter keys, record the keyboard activities and block the keys until a key is entered and simulate the keys. specified.

5.4.8 pyautogui

PyAutoGUI is a cross-platform GUI automation Python module for human beings. Used toprogrammatically control the mouse & keyboard.

5.5 Chatbot GUI

5.5.1 Tokenization

In Python tokenization basically refers to splitting up a larger body of text into smaller lines, words or even creating words for a non-English language. Tokenization is the process by which a large quantity of text is divided into smaller parts called tokens. These tokens are veryuseful for finding patterns and are considered as a base step for stemming and lemmatization. for example: - sentence data = The First sentence is about Python. The Second: about Django. You can learn Python, Django, and Data Analysis here.

5.5.2 Stemming

Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. For example: -

Words = ["program","programs","programmer","programing",]

[program, program, program, program, program]

5.5.3 Bag_of_Word

A Natural Language Processing technique of text modeling known as Bag of Words model. Whenever we apply any algorithm in NLP, it works on numbers. We cannot directly feed our text into that algorithm. Hence, Bag of Words model is used to preprocess the text by converting it into a bag of words, which keeps a count of the total occurrences of most frequently used words.

5.5.4 Modules used.

5.5.4.1 Random

It can be used to perform some action randomly such as to get a random number, selecting a random element from a list, shuffle elements randomly.

5.5.4.2 JSON

It will convert strings to Python datatypes, normally the JSON functions are used to readand write directly from JSON file.

5.5.4.3 PyTorch

It will convert strings to Python datatypes, normally the JSON functions are used to readand write directly from JSON file.

CHAPTER 6

TESTING

6.1 UNIT TESTING

Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validateunit components with its performance. A unit is a single testable part of a software system and tested during the development phaseof the application software.

The purpose of unit testing is to test the correctness of isolated code. A unit component is anindividual function or code of the application. White box testing approach used for unit testing and usually done by the developers. Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as **Unit testing** or **components testing**.

Techniques of Unit Testing: -

6.1.1 White Box Testing

White box testing techniques analyze internal structures the used data structures, internaldesign, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

6.1.2 Black Box Testing

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer.

6.1.3 Grey Box

Grey box testing is a software testing method to test the software application with partial knowledge of the internal working structure. It is a **combination of black box and white box testing** because it involves access to internal coding to design test cases as white boxtesting and testing practices are done at functionality level as black box testing.

6.2 INTEGRATION TESTING

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integratedcomponents or units.

Unit Testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with several software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

6.3 Techniques of Integrating Testing:

6.3.1 Incremental Approach

In the Incremental Approach, modules are added in ascending order one by one or according to need. The selected modules must be logically related. Generally, two or morethan two modules are added and tested to determine the correctness of functions. The process continues until the successful testing of all the modules.

6.3.2 Top-down Approach

The top-down testing strategy deals with the process in which higher level modules are tested with lower-level modules until the successful completion of testing of all the modules. Major design flaws can be detected and fixed early because critical modules tested first. In this type of method, we will add the modules incrementally or one by oneand check the data flow in the same order.

6.3.3 Bottom – Up Approach

The bottom to up testing strategy deals with the process in which lower-level modules are tested with higher level modules until the successful completion of testing of all the modules. Top level critical modules are tested at last, so it may cause a defect. Or we can say that we will be adding the modules from **bottom to the top** and check the data flow inthe same order.

6.3.4 Big Bang Approach

In this approach, testing is done via the integration of all modules at once. It is convenient for small software systems, if used for large software systems identification of defects is difficult.

Since this testing can be done after completion of all modules due to that testing team has less time for execution of this process so that internally linked interfaces and high-risk critical modules can be missed easily.

6.2 SYSTEM TESTING

System Testing includes testing of a fully integrated software system. Generally, a computer system is made with the integration of software (any software is only a single element of a computer system). The software is developed in units and then interfaced with other software and hardware to create a complete computer system. In other words, a computer system consists of a group of software to perform the various tasks, but only software cannot perform the task; for that software must be interfaced with compatible hardware. System testing is a series of different type of tests with the purpose to exercise and examine the full working of an integrated software computer system against requirements.

To check the end-to-end flow of an application or the software as a user is known as **System** testing. In this, we navigate (go through) all the necessary modules of an application and check if the end features or the end business works fine and test the product as a whole system. It is **end-to-end testing** where the testing environment is like the production environment.

6.3 Types of System Testing:

6.3.1 Performance Testing:

Performance Testing is a type of software testing that is carried out to test the speed,scalability, stability and reliability of the software product or application.

6.3.2 Load Testing:

Load Testing is a type of software Testing which is carried out to determine the behavior of a system or software product under extreme load.

6.3.3 Stress Testing:

Stress Testing is a type of software testing performed to check the robustness of thesystem under the varying loads.

6.1.3 Scalability Testing:

Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale upor scale down the number of user request load.

6.5 ACCEPTANCE TESTING

Acceptance testing is formal testing based on user requirements and function processing. It determines whether the software is conforming specified requirements and user requirements or not. It is conducted as a kind of Black Box testing where the number of required users involved testing the acceptance level of the system. It is the fourth and last level of software testing.

User acceptance testing (UAT) is a type of testing, which is done by the customer before accepting the final product. Generally, UAT is done by the customer (domain expert) for their satisfaction, and check whether the application is working according to given business scenarios, real-time scenarios. In this, we concentrate only on those features and scenarios which are regularly used by the customer or mostly user scenarios for the business or those scenarios which are used daily by the end-user or the customer.

6.2 SOFTWARE VERIFICATION AND VALIDATION

6.2.3 Software Verification

Verification testing includes different activities such as business requirements, system requirements, design review, and code walkthrough while developing a product.

It is also known as static testing, where we are ensuring that "**we are developing the right product or not**". And it also checks that the developed application fulfilling all the requirements given by the client. Verification is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. Itverifies whether the developed product fulfills the requirements that we have.

Activities involved in verification:

- Inspections
- Reviews
- Walkthroughs
- Desk-checking

6.2.4 Software Validation

Validation testing is testing where tester performed functional and non-functional testing. Here **functional testing** includes Unit Testing (UT), Integration Testing (IT) and System Testing (ST), and **non-functional** testing includes User acceptance testing (UAT).

Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e., it checks what we are developing is the right product. it is validation of actual and expected product.

Activities involved in validation:

- Black box testing
- White box testing
- Unit testing
- Integration testing

6.3 TEST PROCEDURES

A test procedure is a formal specification of test cases to be applied to one or more target program modules. Test procedures are executable. A process called the VERIFIER appliesa test procedure to its target modules and produces an exception report indicating which test cases, if any, failed.

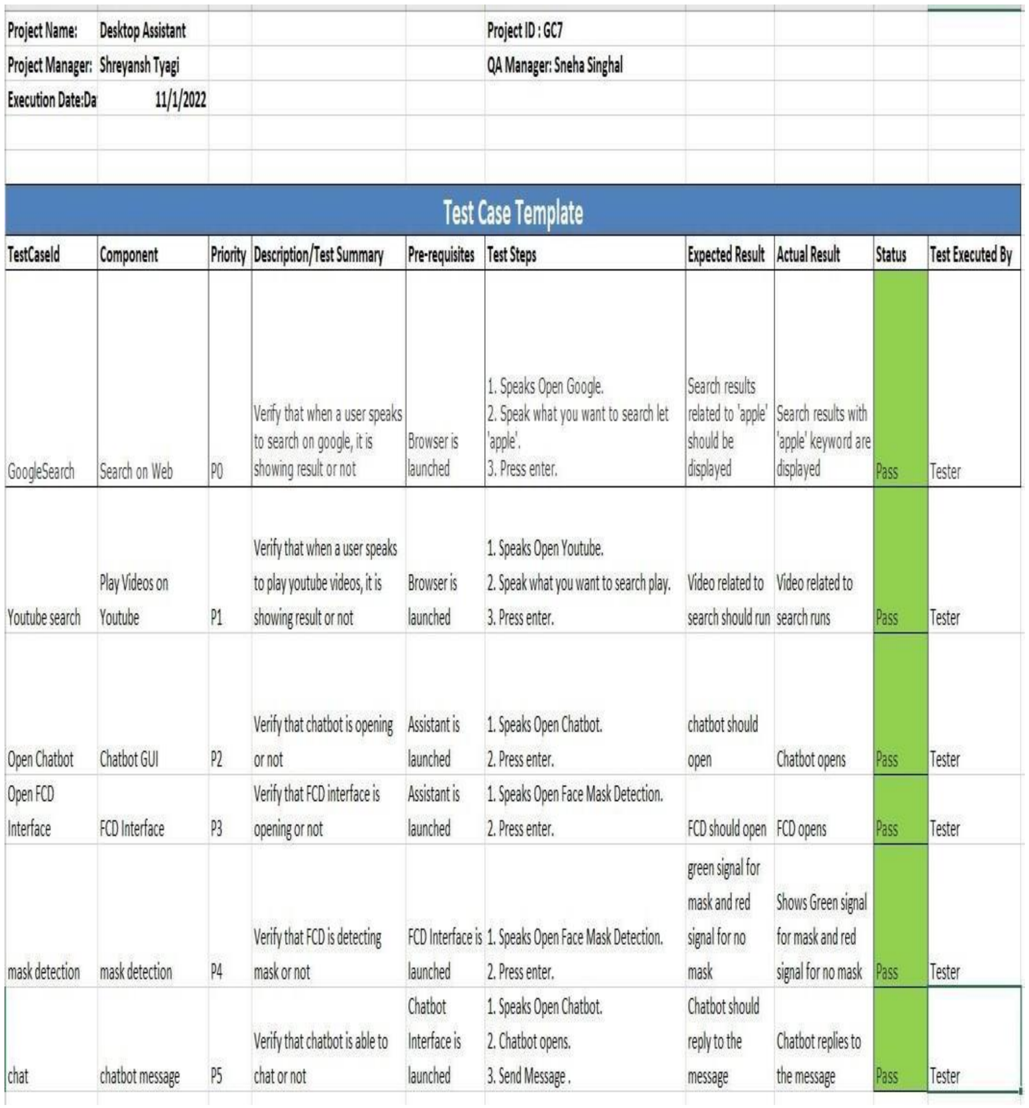
Test procedures facilitate thorough software testing by allowing individual modules or arbitrary groups of modules to be thoroughly tested outside the environment in which theywill eventually reside. Test procedures are complete, self-contained, self-validating and execute automatically. Test procedures are a deliverable product of the software development process and are used for both initial checkout and subsequent regression testing of target program modifications.

Test procedures are coded in a new language called TPL (Test Procedure Language). Thepaper analyzes current testing practices, describes the structure and design of test procedures, and introduces the Fortran Test Procedure Language.

6.4 TEST CASES

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario to verify compliance againsta specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point oralso known as execution postcondition.



Project Name:	Desktop Assistant	Project ID : GC7	
Project Manager:	Shreyansh Tyagi	QA Manager: Sneha Singhal	
Execution Date:Da	11/1/2022		

Test Case Template

TestCaseId	Component	Priority	Description/Test Summary	Pre-requisites	Test Steps	Expected Result	Actual Result	Status	Test Executed By
GoogleSearch	Search on Web	P0	Verify that when a user speaks to search on google, it is showing result or not	Browser is launched	1. Speaks Open Google. 2. Speak what you want to search let 'apple'. 3. Press enter.	Search results related to 'apple' should be displayed	Search results with 'apple' keyword are displayed	Pass	Tester
Youtube search	Play Videos on Youtube	P1	Verify that when a user speaks to play youtube videos, it is showing result or not	Browser is launched	1. Speaks Open Youtube. 2. Speak what you want to search play. 3. Press enter.	Video related to search should run	Video related to search runs	Pass	Tester
Open Chatbot	Chatbot GUI	P2	Verify that chatbot is opening or not	Assistant is launched	1. Speaks Open Chatbot. 2. Press enter.	chatbot should open	Chatbot opens	Pass	Tester
Open FCD Interface	FCD Interface	P3	Verify that FCD interface is opening or not	Assistant is launched	1. Speaks Open Face Mask Detection. 2. Press enter.	FCD should open	FCD opens	Pass	Tester
mask detection	mask detection	P4	Verify that FCD is detecting mask or not	FCD Interface is launched	1. Speaks Open Face Mask Detection. 2. Press enter.	green signal for mask and red signal for no mask	Shows Green signal for mask and red signal for no mask	Pass	Tester
chat	chatbot message	P5	Verify that chatbot is able to chat or not	Chatbot Interface is launched	1. Speaks Open Chatbot. 2. Chatbot opens. 3. Send Message .	Chatbot should reply to the message	Chatbot replies to the message	Pass	Tester

Fig 6.4.1 Test Cases for Desktop Assistant

CHAPTER 7

FORM DESIGN

7.1 Desktop Assistant Interface

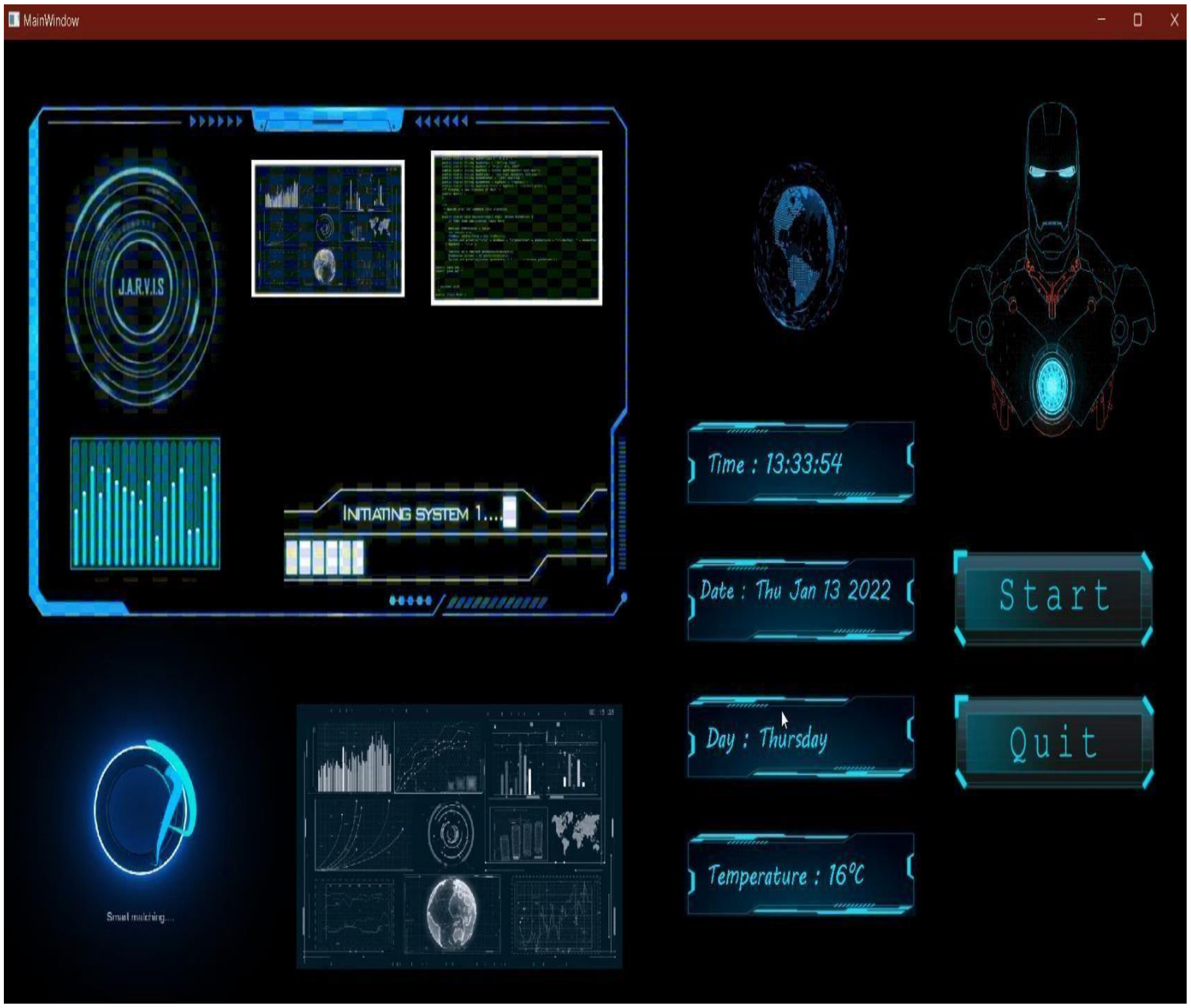
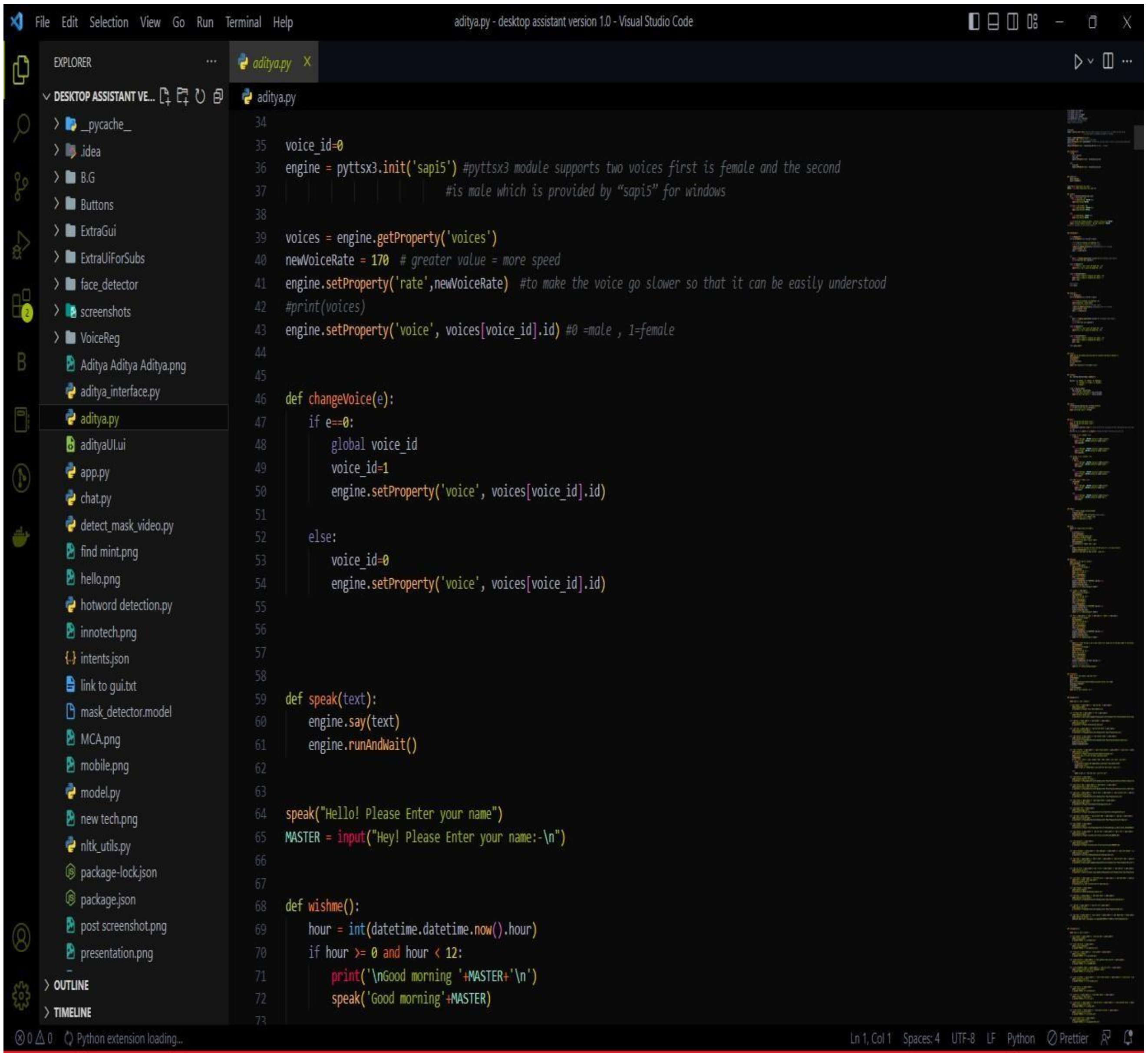


Fig 7.1 Desktop Assistant Interface

7.2 Backend Interface

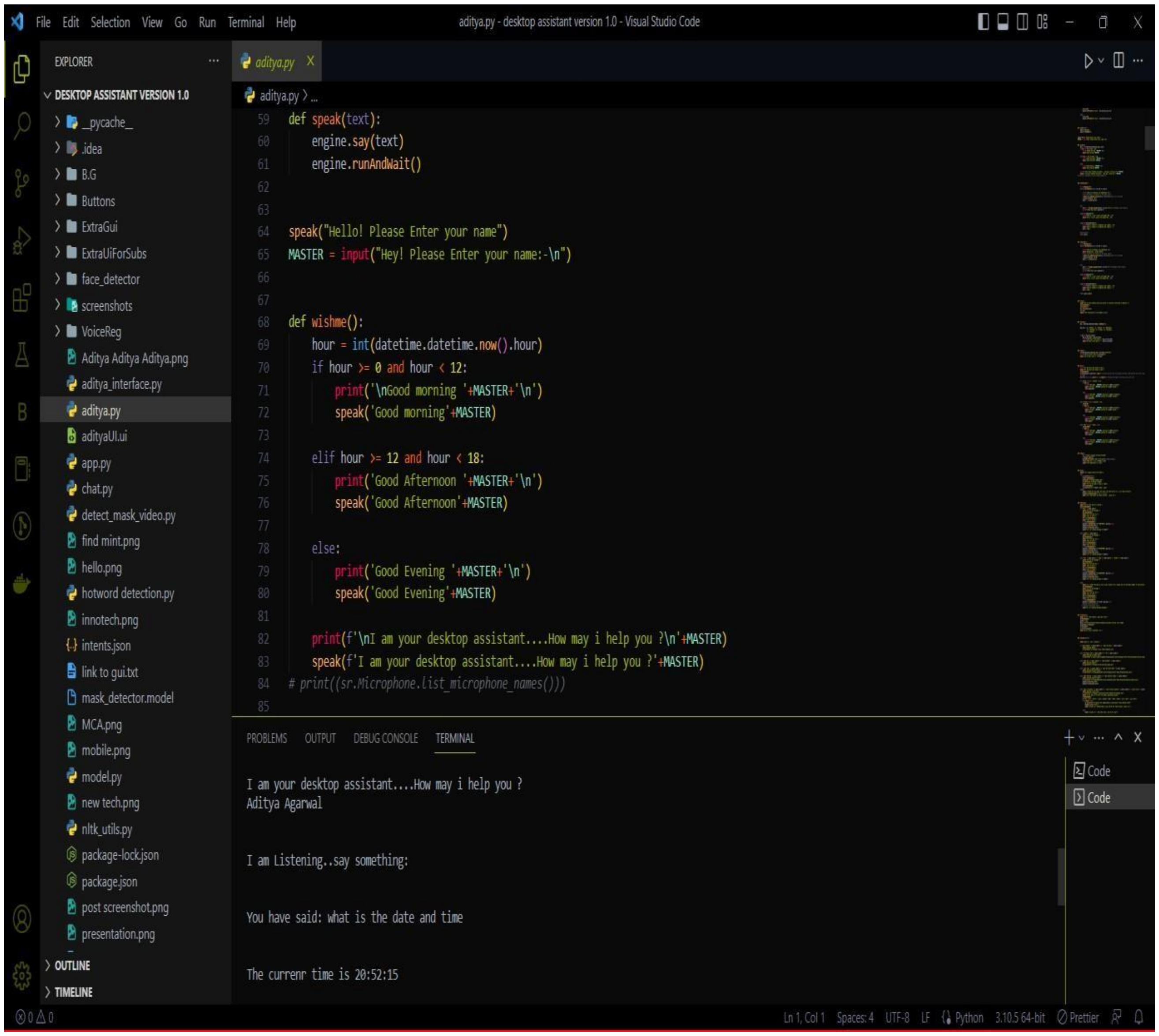


```
voice_id=0
engine = pyttsx3.init('sapi5') #pyttsx3 module supports two voices first is female and the second
                               #is male which is provided by “sapi5” for windows

voices = engine.getProperty('voices')
newVoiceRate = 170  # greater value = more speed
engine.setProperty('rate',newVoiceRate)  #to make the voice go slower so that it can be easily understood
#print(voices)
engine.setProperty('voice', voices[voice_id].id) #0 =male , 1=female


def changeVoice(e):
    if e==0:
        global voice_id
        voice_id=1
        engine.setProperty('voice', voices[voice_id].id)

    else:
        voice_id=0
        engine.setProperty('voice', voices[voice_id].id)




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


speak("Hello! Please Enter your name")
MASTER = input("Hey! Please Enter your name:-\n")


def wishme():
    hour = int(datetime.datetime.now().hour)
    if hour >= 0 and hour < 12:
        print('\nGood morning '+MASTER+'\n')
        speak('Good morning'+MASTER)
```

Fig 7.2 Backend Interface

7.3 Task Performed

7.3.1 Time & Date

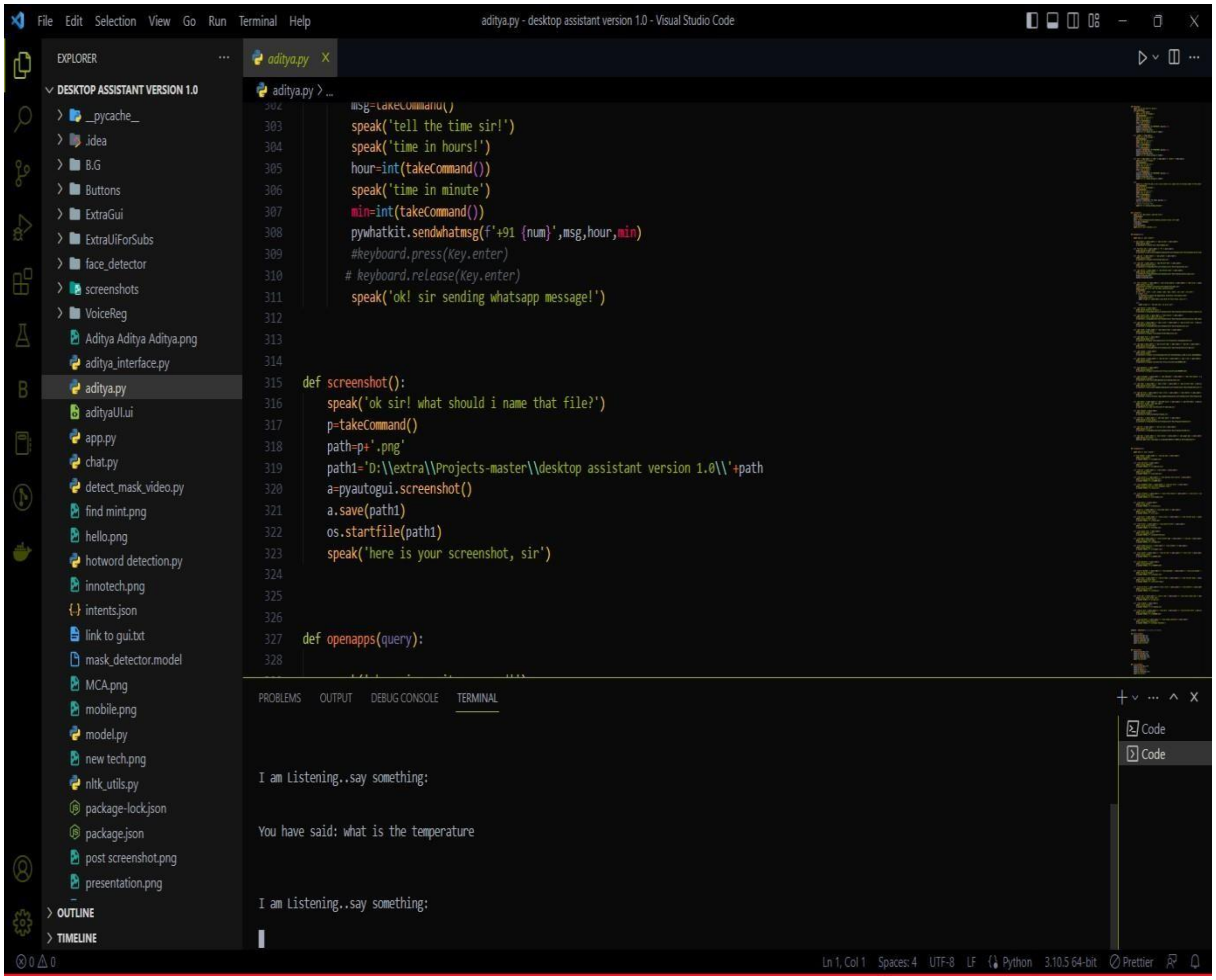


```python
def speak(text):
    engine.say(text)
    engine.runAndWait()


speak("Hello! Please Enter your name")
MASTER = input("Hey! Please Enter your name:-\n")


def wishme():
    hour = int(datetime.datetime.now().hour)
    if hour >= 0 and hour < 12:
        print('\nGood morning '+MASTER+'\n')
        speak('Good morning'+MASTER)

    elif hour >= 12 and hour < 18:
        print('Good Afternoon '+MASTER+'\n')
        speak('Good Afternoon'+MASTER)

    else:
        print('Good Evening '+MASTER+'\n')
        speak('Good Evening'+MASTER)

    print(f'\nI am your desktop assistant....How may i help you ?\n'+MASTER)
    speak(f'I am your desktop assistant....How may i help you ?'+MASTER)
# print((sr.Microphone.list_microphone_names()))
```

```
I am your desktop assistant....How may i help you ?
Aditya Agarwal


I am Listening..say something:


You have said: what is the date and time


The currenr time is 20:52:15
```

Fig 7.3.1 Time and Date

7.3.2 Temperature

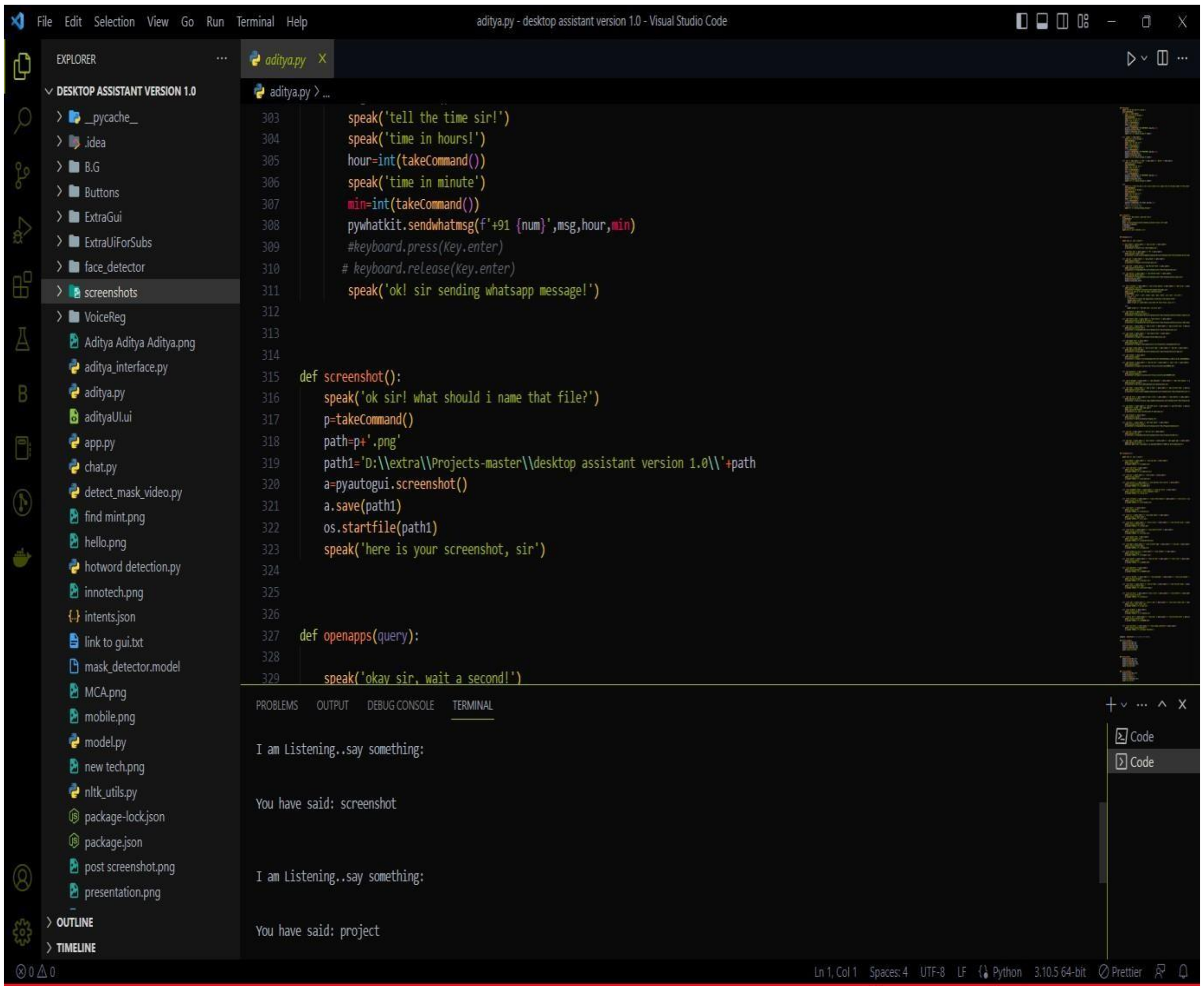


```
            msg=takeCommand()
            speak('tell the time sir!')
            speak('time in hours!')
            hour=int(takeCommand())
            speak('time in minute')
            min=int(takeCommand())
            pywhatkit.sendwhatmsg(f'+91 {num}',msg,hour,min)
            #keyboard.press(Key.enter)
           # keyboard.release(Key.enter)
            speak('ok! sir sending whatsapp message!')



def screenshot():
    speak('ok sir! what should i name that file?')
    p=takeCommand()
    path=p+'.png'
    path1='D:\\extra\\Projects-master\\desktop assistant version 1.0\\'+path
    a=pyautogui.screenshot()
    a.save(path1)
    os.startfile(path1)
    speak('here is your screenshot, sir')



def openapps(query):

```

```
I am Listening..say something:

You have said: what is the temperature

I am Listening..say something:
```

Fig 7.3.2 Temperature

7.3.3 Screenshot

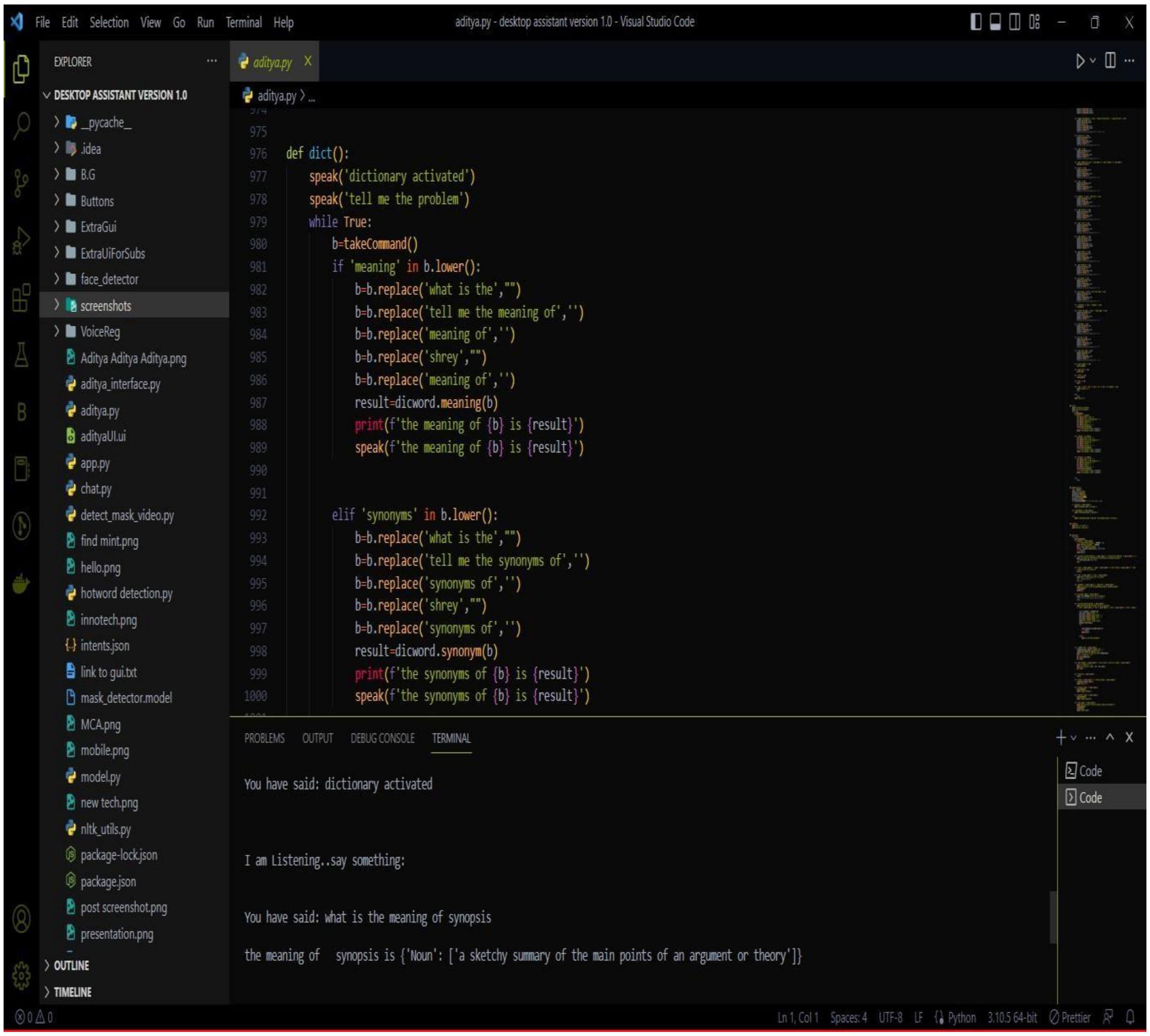


```
        speak('tell the time sir!')
        speak('time in hours!')
        hour=int(takeCommand())
        speak('time in minute')
        min=int(takeCommand())
        pywhatkit.sendwhatmsg(f'+91 {num}',msg,hour,min)
        #keyboard.press(Key.enter)
       # keyboard.release(Key.enter)
        speak('ok! sir sending whatsapp message!')



def screenshot():
    speak('ok sir! what should i name that file?')
    p=takeCommand()
    path=p+'.png'
    path1='D:\\extra\\Projects-master\\desktop assistant version 1.0\\'+path
    a=pyautogui.screenshot()
    a.save(path1)
    os.startfile(path1)
    speak('here is your screenshot, sir')



def openapps(query):

    speak('okay sir, wait a second!')
```

```
I am Listening..say something:

You have said: screenshot


I am Listening..say something:

You have said: project
```

Fig 7.3.3 Screenshot

7.3.4 Word Meaning

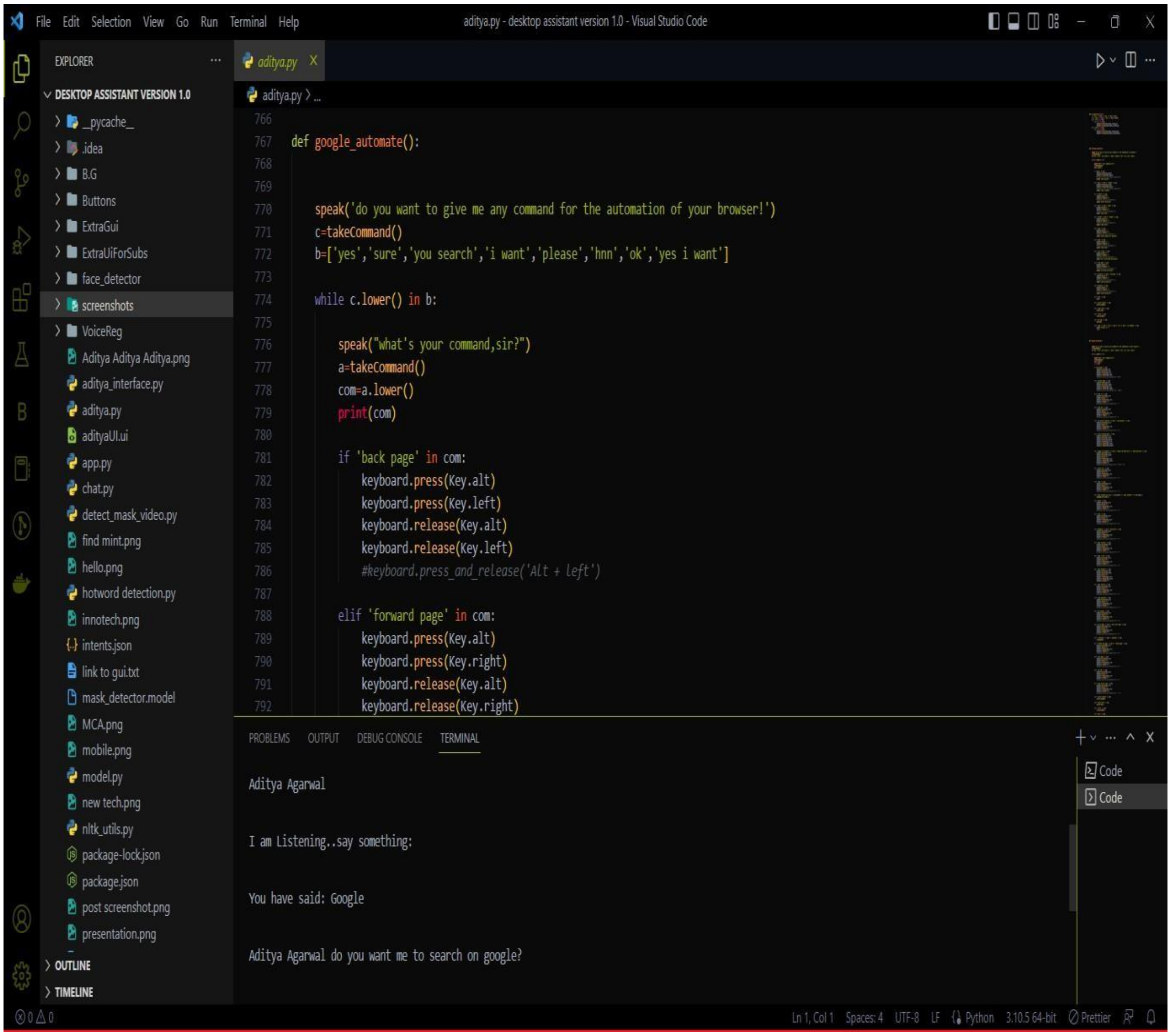

aditya.py - desktop assistant version 1.0 - Visual Studio Code
EXPLORER
DESKTOP ASSISTANT VERSION 1.0
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```python
def dict():
    speak('dictionary activated')
    speak('tell me the problem')
    while True:
        b=takeCommand()
        if 'meaning' in b.lower():
            b=b.replace('what is the',"")
            b=b.replace('tell me the meaning of','')
            b=b.replace('meaning of','')
            b=b.replace('shrey',"")
            b=b.replace('meaning of','')
            result=dicword.meaning(b)
            print(f'the meaning of {b} is {result}')
            speak(f'the meaning of {b} is {result}')


        elif 'synonyms' in b.lower():
            b=b.replace('what is the',"")
            b=b.replace('tell me the synonyms of','')
            b=b.replace('synonyms of','')
            b=b.replace('shrey',"")
            b=b.replace('synonyms of','')
            result=dicword.synonym(b)
            print(f'the synonyms of {b} is {result}')
            speak(f'the synonyms of {b} is {result}')
```

```
You have said: dictionary activated


I am Listening..say something:


You have said: what is the meaning of synopsis

the meaning of   synopsis is {'Noun': ['a sketchy summary of the main points of an argument or theory']}
```

Fig 7.3.4 Word Meaning

7.3.5 Google Search

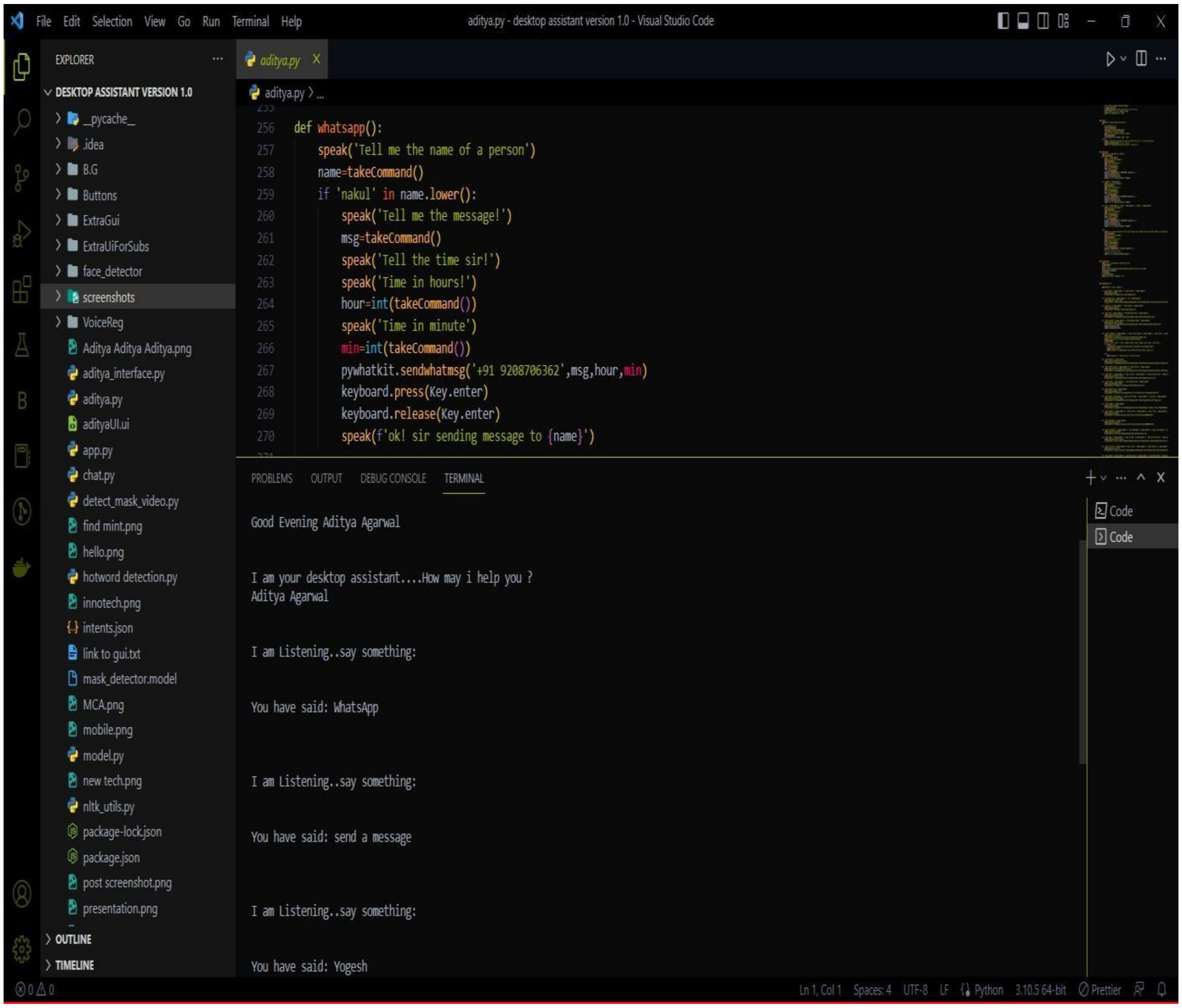


```
def google_automate():


    speak('do you want to give me any command for the automation of your browser!')
    c=takeCommand()
    b=['yes','sure','you search','i want','please','hnn','ok','yes i want']

    while c.lower() in b:

        speak("what's your command,sir?")
        a=takeCommand()
        com=a.lower()
        print(com)

        if 'back page' in com:
            keyboard.press(Key.alt)
            keyboard.press(Key.left)
            keyboard.release(Key.alt)
            keyboard.release(Key.left)
            #keyboard.press_and_release('Alt + left')

        elif 'forward page' in com:
            keyboard.press(Key.alt)
            keyboard.press(Key.right)
            keyboard.release(Key.alt)
            keyboard.release(Key.right)
```

```
Aditya Agarwal

I am Listening..say something:

You have said: Google

Aditya Agarwal do you want me to search on google?
```

Fig 7.3.5 Google Search

7.3.6 WhatsApp Chat Scheduling

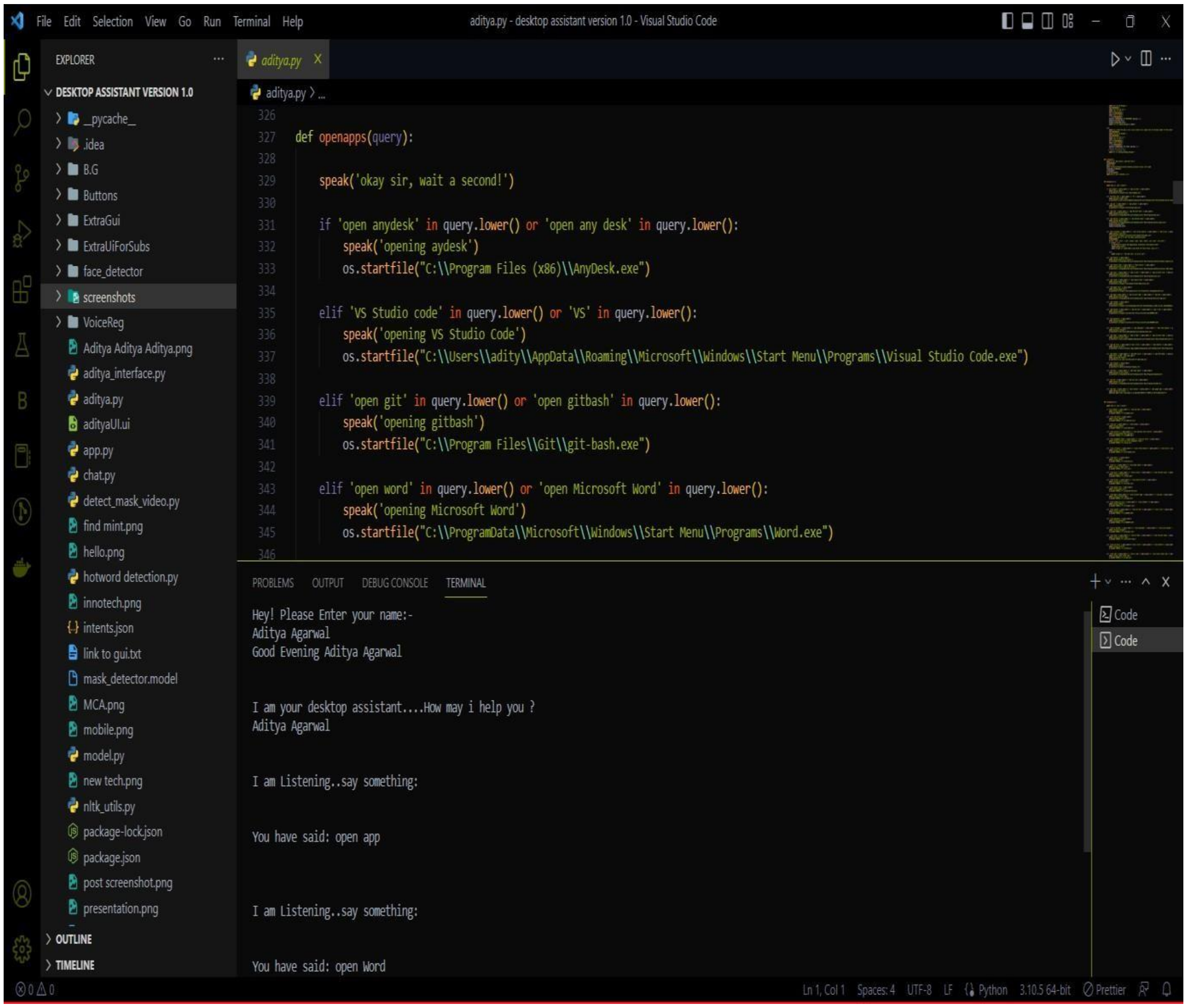


```
def whatsapp():
    speak('Tell me the name of a person')
    name=takeCommand()
    if 'nakul' in name.lower():
        speak('Tell me the message!')
        msg=takeCommand()
        speak('Tell the time sir!')
        speak('Time in hours!')
        hour=int(takeCommand())
        speak('Time in minute')
        min=int(takeCommand())
        pywhatkit.sendwhatmsg('+91 9208706362',msg,hour,min)
        keyboard.press(Key.enter)
        keyboard.release(Key.enter)
        speak(f'ok! sir sending message to {name}')
```

```
Good Evening Aditya Agarwal


I am your desktop assistant....How may i help you ?
Aditya Agarwal


I am Listening..say something:


You have said: WhatsApp



I am Listening..say something:


You have said: send a message



I am Listening..say something:


You have said: Yogesh
```

Fig 7.3.6 WhatsApp Scheduling

7.3.7 Opening New Apps



```python
def openapps(query):

    speak('okay sir, wait a second!')

    if 'open anydesk' in query.lower() or 'open any desk' in query.lower():
        speak('opening aydesk')
        os.startfile("C:\\Program Files (x86)\\AnyDesk.exe")

    elif 'VS Studio code' in query.lower() or 'VS' in query.lower():
        speak('opening VS Studio Code')
        os.startfile("C:\\Users\\adity\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Visual Studio Code.exe")

    elif 'open git' in query.lower() or 'open gitbash' in query.lower():
        speak('opening gitbash')
        os.startfile("C:\\Program Files\\Git\\git-bash.exe")

    elif 'open word' in query.lower() or 'open Microsoft Word' in query.lower():
        speak('opening Microsoft Word')
        os.startfile("C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Word.exe")
```

```
Hey! Please Enter your name:-
Aditya Agarwal
Good Evening Aditya Agarwal


I am your desktop assistant....How may i help you ?
Aditya Agarwal


I am Listening..say something:


You have said: open app



I am Listening..say something:


You have said: open Word
```

Fig 7.3.7 Opening New App

CHAPTER 8

ADVANTAGES

- Help you save time.
- Users can talk on the phone without putting it near to the ears.
- Improve your Speech Recognition Skills
- Can get things done fast.
- Can control multiple products efficiently.
- Help With Your Distress
- Decreases the gap between humans and technology.
- Helps to perform tasks without physical interaction.
- Helps Physically Challenged people to get engage with technology.
- Make people smarter.
- Allow people not to be dependent on others.

CHAPTER 9

CONCLUSION

In this project “Desktop Assistant” we discussed the design and implementation of Digital Assistance. The project is built using open-source software modules with Jupyter Notebook backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easier to add additional features without disturbing current systemfunctionalities.

It not only works on human commands but also give responses to the user based on the querybeing asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way the user feels more comfortable and feels free to interact with the voice assistant. The application should also eliminate any kind of unnecessary manual work required in the user life of performing every task. The entire system works on the verbal inputrather than the next one.

CHAPTER 10

BIBLIOGRAPHY

Online Sites

- https://www.tutorialspoint.com
- https://www.javatpoint.com
- https://www.w3schools.com
- https://stackoverflow.com
- https://academia.com

YouTube Channels

- Code with harry.
- Free Code Camp
- Telusko
- Simply Learn
- Tech with Tim