**VOICE BASED CONTROL SYSTEM**

**AND ASSISTANT**

**A Project Report**

***Submitted By:***

**ARPIT KUMAR SINGH : 1180432020**

**AYUSH AWASTHI : 1170432028**

**ABBAS ALI : 1180432001**

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

****at

**BABU BANARASI DAS UNIVERSITY**

**SECTOR II, DR AKHILESH DAS NAGAR, FAIZABAD ROAD, LUCKNOW (UP) – INDIA, 226028**

**JUNE, 2022**

**DECLARATION**

We hereby declare that the project entitled “**VOICE BASED CONTROL SYSTEM AND ASSISTANT**” submitted for the award of the degree of Bachelor of Technology (in Computer Science and Engineering) is our original work and the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar titles.

**Signature of the Project member(s):**

Arpit Kumar Singh.

Ayush Awasthi.

Abbas Ali.

**Place:** Lucknow

**Date:** June, 2022.

**CERTIFICATE**

This is to certify that the project titled **“VOICE BASED CONTROL SYSTEM AND ASSISTANT”** is the bona fide work carried out by ***Arpit Kumar Singh, Ayush Awasthi, Abbas Ali***, the students of Bachelor of Technology (in Computer Science and Engineering) of Babu Banarasi Das University, Lucknow, Uttar Pradesh, during the academic year 2021-22, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

**Signature of the Guide:**

**Place:** Lucknow.

**Date:** June, 2022.

**ACKNOWLEDGEMENT**

The completion of this project gives us much Pleasure. We wish to express our heartfelt gratitude to the all the people who have played a crucial role in the research for this project, without their active cooperation, the preparation of this project could not have been completed within the specified time limit.

In completing this project titled **VOICE BASED CONTROL SYSTEM & ASSISTANT**, we were guided and assisted by certain respected people, who deserve our greatest gratitude.

We are thankful to our respected Director and Dean(of school of engineering), ***Dr. Apurva Anand*** and our respected Head of Department (CSE), ***Dr. Praveen Shukla*** for motivating us to complete this project with complete focus and attention.

We would like to show our gratitude to our project guide and respected lecturer ***Mr.Sarvesh Kumar*** for giving us a good guideline and support throughout this project with utmost cooperation and patience for the completion of this Project.

We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in completing this project and project report.

Many people, especially our respected professors and friends, have made valuable comments and suggestions during the project which gave us inspiration to improve our project. Here, we thank all the people for their help directly and indirectly to complete this project report.

**ABSTRACT**

The project aims to develop a voice based control system and personal-assistant for Windows-based systems.

Our software application is a digital life monitor, administrator and assistant that is much more than just a digital virtual assistant. As a personal assistant, it assists the end-user with day-to-day activities like general human conversation, searching queries in web-browser, searching for videos, images, live weather conditions, word meanings, searching for medicine details, health recommendations based on symptoms and set reminder and also reminding the user about the scheduled events and tasks.

It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands. Users can interact with the assistant through voice commands and certain functionalities can also be accessed using mouse\keyboard input.

It can perform all basic tasks on a desktop machine such as: launch applications, play/switch music or videos, set and play reminder, tell date and time, take screenshot, send email, etc. Complex tasks that it can execute include: using system camera (if available) to capture image, record video, perform image face-detection, perform real-time face detection, and execute interactive games to play.

**TABLE OF CONTENT**

1. **Declaration………………………………………….……… 02**
2. **Certificate………………………………………….……… 03**
3. **Acknowledgement………………………………………… 04**
4. **Abstract…………………………………………………… 05**
5. **Introduction………………………………………………. 09**
   1. Background……………………………………………... 10
   2. Problem Statement……………………………………… 12
   3. Project Overview……………………………………….. 13
      1. Purpose………………………………...…… ……. 13
      2. Objectives………………………………………… 13
      3. Scope……………………………………………… 15
      4. Applicability……………………………………… 16
   4. Hardware Specifications………………………………… 17
   5. Software Specifications………………………………… 18
6. **Literature Survey………………………………………… 19**
   1. Existing System………………………………………… 19
   2. Proposed System……………………………………… 23
   3. Feasibility Study………………………………………… 25
   4. System Development…………………………………… 29
   5. Survey Of Technology………………………………… 32
   6. Technologies to be used………………………………… 33
7. **System Analysis and Design……………………………… 36**
   1. Use-case Diagram……………………………………… 36
   2. Class Diagram………………………………………….. 37
   3. E-R Diagram…………………………………………… 38
   4. Component Diagram……………………………………. 39
   5. Sequence Diagram……………………………………… 40
      1. Sequence diagram for query response…………… 40
      2. Sequence diagram for task execution…………… 41
   6. Data Flow Diagram……………………………………. 42
      1. DFD Level 0 (Context Level Diagram…………… 42
      2. DFD Level 1……………………………………. 43
      3. DFD Level 2……………………………………. 43
      4. Deployment Diagram…………………………… 45
      5. Activity Diagram………………………………… 46
8. **Test Case Design…………………………………………… 47**
   1. Test Case 1…………………………………………… 47
   2. Test Case 2…………………………………………… 48
   3. Test Case 3…………………………………………… 49
9. **Results……………………………………………………… 50**
   1. Test Case 1…………………………………………… 50
   2. Test Case 2…………………………………………… 51
   3. Test Case 3…………………………………………… 52
10. **Conclusion And Recommendations……………………… 54**
11. **References………………………………………………… 56**
12. **Appendices………………………………………………… 58**
    1. List of Figures. ………………………………………… 58
    2. About technologies used……………………………… 59
13. **INTRODUCTION**

There is no doubt that human intervention cannot be completely eliminated from computer-based systems because human intelligence is required at every level of any system development life cycle. Our purpose in this project is to develop such a software application that can be a simple automation system with integrated digital assistant that can make day-to-day tasks of the user a seamless experience and also provide certain added features for utility and entertainment purposes also (like games, etc.).

Voice searches have dominated over text search. Web searches conducted via mobile devices have only just overtaken those carried out using a computer and the analysts are already predicting that 50% of searches will be via voice by 2023. Digital assistants are turning out to be smarter than ever. Allow your intelligent assistant to make email work for you. Detect intent, pick out important information, automate processes, and deliver personalized responses.

This system is designed to be used efficiently on desktops. Personal assistant software improves user productivity by managing routine tasks of the user and by providing information from online sources to the user.

This project was started on the premise that there is sufficient amount of openly available data and information on the web that can be utilized to build a digital assistant that has access to making intelligent decisions for routine user activities.

* 1. **BACKGROUND**

Intelligent voice-recognition based personal assistants are software that have been developed and designed to assist user with basic tasks, usually providing information using natural language. Most of the voice assistants use online resources to answer a user's questions about the weather, sport scores, to provide driving directions and to answer similar information-based queries and also provide services, such as calendar and meeting reminders while many offer essential services, like health monitoring and alerts via special applications. Typically, an intelligent personal assistants will answer queries and perform actions via voice commands using a natural language user interface.

There are a variety of terms that refer to agents that can perform tasks or services for an individual, and they are almost interchangeable — but not quite. They differ mainly based on how we interact with the technology, the app, or a combination of both. Here are some basic definitions, similarities, and differences:

1. **Intelligent Personal Assistants (IPA)**: This type of software can assist users with some basic tasks, usually using natural language. Intelligent personal assistants are also so smart that they go online and search for an answers to a user’s question. It may text or voice either of them trigger an action.
2. **Automated Personal Assistant:** automated means the task is performed by itself. The personal assistants are using AI and deep learning according to the user’s experience and behaviour towards the IPA they are able to do some automated task.
3. **Smart Assistant:** It is usually refers to the types of physical devices (pertaining to IoT devices and technology) that can provide various advance features and services by using smart speakers that listen for a wake up word to become active and can perform certain tasks. Amazon Echo, Google Home, and Apple HomePod are examples of smart assistant’s devices.
4. **Chatbot:** Its function is similar to its name it uses text as medium to communicate and provide information and do task for user. Chatbots can imitate a conversation with a human user.
5. **Voice Assistant:** The input key here is our voice. It is a digital assistant that uses voice recognition, speech synthesis, and natural language processing (NLP) and also AI to provide an amazing service through an application exam Siri, Ok Google. Cortana, etc.
   1. **PROBLEM STATEMENT**

* There are several automation softwares that can automate pc; but generally are limited to specific functionality of their own, like automating only the aspects of single application in the system.
* The current trend in research and study of automation programs is on a rise, and various tech. companies are either developing automation algorithms or the one’s which have developed are working to optimize and enhance their algorithms that can integrate automation with integrated technologies and more new features in order to achieve higher productivity and efficiency.
* Some popular optimized Voice Assistants have been developed by tech. (giant) companies such as Google’s google assistant, Apple’s Siri, Amazon’s Alexa, etc. and many other leading tech. companies have also developed their own version of voice assistants.

Although several automation programs and various chatbots with speech-to-text feature have been developed; but the idea of a single automation software that can combine different automating functions locally was fascinating.

* 1. **PROJECT OVERVIEW**

**(1.3.1) PURPOSE :**

According to the overall description in the context of introduction, the purpose of the project is to develop a desktop application that provides an intelligent voice assistant with the functionalities as mail exchange, alarm, event handler, local services, music play service, video play service, checking weather, searching engine (Google, Wikipedia), camera, web search result translator, help menu and extra features.

Hence our application aims at developing a voice-recognition based control system cum digital assistant for windows based personal computer systems. The main purpose of the software is to perform the tasks of the user at certain commands, provided in speech. It will ease daily routine of the work of the user as a complete task can be done on a single command.

**(1.3.2) OBJECTIVES :**

Main objective of building an automating system integrated with digital personal assistant software, is using semantic data sources available on the web, user generated content and providing knowledge from knowledge databases. The main purpose of an apt digital assistant in our system is to answer questions that users may have. This may be done in a business environment, for example, on the business website, you want to search about certain terms or phrases, if you ask the assistant to search it will do so by opening them in new tabs without disturbing your work.

Digital assistants can tremendously save you time. We spend hours in online research and then making the report in our terms of understanding. Our voice based control system can do that for you. Provide a topic for research and continue with your tasks while our application does the research.

Another difficult task is to remember scheduled event dates, birthdates or anniversaries. It comes with a surprise when you enter the class and realize it is class test today. Just tell our system digital assistant in advance about your tests and it reminds 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 are capable of speaking around 150 during the same period of time. In this respect, the ability of personal assistants to accurately recognize spoken words is a prerequisite for them to be adopted by consumers.

Objective motto:

***Fully voice controlled and interactive desktop application.***

**(1.3.3) SCOPE :**

Defining scope was an overwhelming exercise as it involved collecting use cases where a smart agent would be useful for a person. Initially, list of use cases where a smart agent would come in handy as a personal assistant to manage or automate the tasks were identified and documented.

This turned out to be a wish list for the agent and so specific boundaries were defined based on the availability of data sources, technologies and concepts that could be validated for these use cases. The initial list of use cases was then categorized based on user-agent interactions, and based on type of inputs and outputs.

Voice assistant based systems will continue to offer more individualized experiences as they get better at differentiating between voices and evolve with improvement in algorithm.

Presently, our Voice Based Control System is being developed as a simple PC automation tool and digital assistant. Among the Various roles played by it are:

1. Search Engine with voice interactions
2. Search Wikipedia.
3. Open applications.
4. Reminder and To-Do facility.
5. Audio/Video song play/switch.
6. Stream music.
7. Sending and checking emails.
8. Image and video capture using system camera.
9. Performing face-detection.
10. And much more.

**(1.3.4) APPLICABILITY :**

The large-scale adoption of artificial intelligence in users’ everyday lives is also fueling the shift towards voice. The number of IoT devices such as smart electronic home equipments and speakers are giving voice assistants more utility in a connected user’s life. Smart speakers are the number one way we are seeing voice being used. Many industry experts even predict that nearly every application will integrate voice technology in some way in the next 7 years. The use of virtual assistants also enhances the system of IoT (Internet of Things).

Modern virtual assistants are capable of a lot of things: they recognize 97% of speech, quickly find information, and analyze it effectively. Besides, they’re easy to integrate into the application. And this is just the beginning!

Digital virtual assistants have expanding scope in a variety of fields. As in every field the first applicability that comes into force is its task handling easing ability. Hence in following sector one can see its wide and expanding scope:

1. Business Solutions
2. Financial work solutions
3. Hospital management
4. Personal app integration
5. Vehicle integration
   1. **HARDWARE SPECIFICATIONS**

Hardware Requirements: (Minimum):

* CPU : core i3 – 1st generation (or equivalent)
* GPU : nvidia GeForce 8 series (or equivalent)
* RAM : 4 GB DDR3
* Hard Disk : SATA 160 GB

Hardware Requirements: (Recommended):

* CPU : core i5 – 5th generation (or equivalent or above)
* GPU : nvidia geforce GTX 700 series (or equivalent or above)
* RAM : 8 GB DDR3
* Hard Disk : SATA 500 GB
  1. **SOFTWARE SPECIFICAIONS**

Software requirements (absolutely needed):

* Operating System : Windows 7 (64-bit or above)
* Python : 3.7 or later
* Microsoft .Net framework 4.8 or later.

Software requirements (supplementary):

* MS Visual C++ (2005 to 2015) all x86 versions.
* MS Visual C++ (2005 to 2015) all x64 versions.
* Python libraries :
  + pyttsx3
  + speech\_recognitiondatetime
  + wikipedia
  + smtplib
  + webbrowser
  + pyscreenshot
  + psutil
  + pyjokes
  + requests
  + jsonlib
  + wolframalpha
  + opencv-python

1. **LITERATURE SURVEY**

This field of digital assistants having speech recognition has seen some major advancements or innovations. This is mainly because of its demand in devices like smartwatches or fitness bands, speakers, bluetooth earphones, mobile phones, laptop or desktop, television, etc. Almost all the digital devices which are coming nowadays are coming with voice assistants which help to control the device with speech recognition only. A new set of techniques is being developed constantly to improve the performance of voice automated search.

**(2.1) EXISTING SYSTEM :**

There already exists a number of desktop automating digital assistants. A few examples of popular digital assistants available in market are discussed in this section along with the tasks they can provide and their drawbacks.

**SIRI (from Apple):**

SIRI is personal assistant software that interfaces with the user through voice interface, recognizes commands and acts on them. It learns to adapt to user’s speech and thus improves voice recognition over time. It also tries to converse with the user when it does not identify the user request.

It integrates with calendar, contacts and music library applications on the device and also integrates with GPS and camera on the device. It uses location, temporal, social and task based contexts, to personalize the agent behavior specifically to the user at a given point of time.

***Supported Tasks:***

• Call someone from my contacts list

• Launch an application on my iPhone

• Send a text message to someone

• Set up a meeting on my calendar

• Set an alarm

• Play a specific song in my iTunes library

• Enter a new note

***Drawback:***

• SIRI does not maintain a knowledge database of its own and its understanding comes from the information captured in domain models and data models.

**ReQall (from reQall, Inc.) :**

ReQall is personal assistant software that runs on smartphones running iOS or Android operating system. It helps user to recall notes as well as tasks within a location and time context. It records user inputs and converts them into commands, and monitors current stack of user tasks to proactively suggest actions while considering any changes in the environment. It also presents information based on the context of the user, as well as filter information to the user based on its learned understanding of the priority of that information.

***Supported Tasks:***

• Reminders

• Email

• Calendar, Google Calendar

• Outlook

• Evernote

• Facebook, LinkedIn

• News Feeds

***Drawbacks:***

• Will take some time to recognize certain sensor’s data for certain automating functionality which causes a notable delay.

• Spoken note-making generally takes too long to process.

**Google Assistant (from Google):**

Google Assistant is a virtual assistant software that interfaces with the user through voice interface, recognizes commands through Google’s API and acts on them. It learns to adapt to user’s speech and thus improves voice recognition over time. It also tries to converse with the user when it does not identify the user request.

It integrates best with google applications and then with calendar, contacts and music library applications on the device and also integrates with GPS and camera on the device. It uses location, temporal, social and task based contexts, to personalize its behavior specifically to the user at a given point of time.

***Supported Tasks:***

• Call or send message to someone from contacts list

• Launch an application

• Set up a meeting or alarm on calendar

• Create custom commands

• Play a specific song or video online.

• Can perform Google image search.

• Multilingual support (continuously improving)

***Drawback:***

• It cannot work offline.

• It is a little heavy on resources.

**(2.2) PROPOSED SYSTEM :**

A light-weight simple application is the main purpose of our proposed system that is light on system resources and works seamlessly.

The working of our application is described in detail below:

* To first understand properly the user’s query we made use of conditional statements, and also made sure if user enters any negative statement then the program understands not to execute the instruction.
* In case of any negative statement by user the program shall reply with apt answer telling user about the improper statement.
* Then adding certain conditions, so that program can know and provide assistance if user is requesting assistance else provide service if user is instructing certain task.
* If requested service or task by user requires executing certain other application on the system then it must be installed and added to environment path or else the program may not be able to find it.
* In such case the program will tell the user of its inability to find the path or absence of required application from the system.
* Certain applications require parameters to be given for further execution of it and our program will ask all such parameters from user one by one.
* In case user doesn’t respond with parameters then the program shall use some default parameters to launch the instructed application.
* For playing audio and video user needs to place audio and video files in (D:\Music) or (D:\Videos) drive as for now this functionality is restricted to a pre-decided drive and later it could be upgraded to be played from user-desired directory.
* For capturing image and video there must be a functioning camera (either inbuilt or webcam).
* For face-detection in image, image must be placed in the indicated directory, to be able processed by our application.

**(2.3) FEASIBILITY STUDY :**

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

Five types of feasibility study are taken into consideration.

**1. Technical feasibility:**

It includes finding out technologies for the project, both hardware and software. For digital assistant, user must have microphone to convey their message and a speaker to listen when system speaks. There are many types of affordable equipment now a days and everyone generally possess them. Besides, system needs internet connection. While using our application, 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 and with digitalization campaigns across many countries, internet has never been so much accessible and affordable.

**2. Operational feasibility:**

It is the ease and simplicity of operation of proposed system. System does not require any special skill set for users to operate it. In fact, it is designed to be used by almost everyone. Just speak your requirements or commands to our application’s digital personal assistant and it will do the best in its capability.

And furthermore any tech/coding savvy individuals or students will find our user manual easy to go through and our provided source code to be able to add his desired functionality or feature to the application with ease, since we used pure python to code our application.

**3. 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, available and affordable.

As far as maintenance is concerned, our application won’t cost a dime as it will be open-source and furthermore the use of python language to develop it has made it easy for even basic python-familiar individual to atleast add any basic new functionalities desired, by himself.

**4. Organizational feasibility:**

This shows the management and organizational structure of the project. This project might be built by a team, but it is must to be taken under consideration that the team was of only 5 individuals and all were totally new to the language (python) as well as the concepts used to build this project.

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. Hence, excellent organizational feasibility is accomplished.

**5. Cultural feasibility:**

It deals with compatibility of the project with cultural environment. Application integrated digital assistant is built in accordance with the general culture. This project is technically feasible with no (negligible) external hardware requirements. Also it is simple in operation and does not cost training or repairs.

**Conclusion of feasibility study:**

This feasibility study examined the possibility of using an independent voice recognition based control system as the input device during the normal system work to enhance and ease user operations. The intent was to determine whether the voice recognition system could be incorporated into a voice based control system designed to increase productivity and easiness of daily routine tasks of the user.

This study showed how the voice recognition system worked in an integrated voice based delivery system for the purpose of delivering instructions and services. An added importance of the study was that the voice system was an independent speech recognition system. At the time this study was conducted, there did exist a lot of different speech recognition systems that interfaced with both graphics and authoring software which allows any user to speak to the system without training the system to recognize the individual user’s voice. This feature increased the usefulness and flexibility of the system.

However, our proposed system was always intended to be simple, light-weight, light on system resources, easy to use, easy to debug and easy to modify.

Since, overall feasibility study of the project reveals that the goals of the proposed system are achievable.

Hence, decision was taken to proceed with the project.

**(2.4) SYSTEM DEVELOPMENT :**

We decided to divide our project into 5 major modules and an additional (maintenance) module (for adding/improving new features and functionalities even after base project completion).

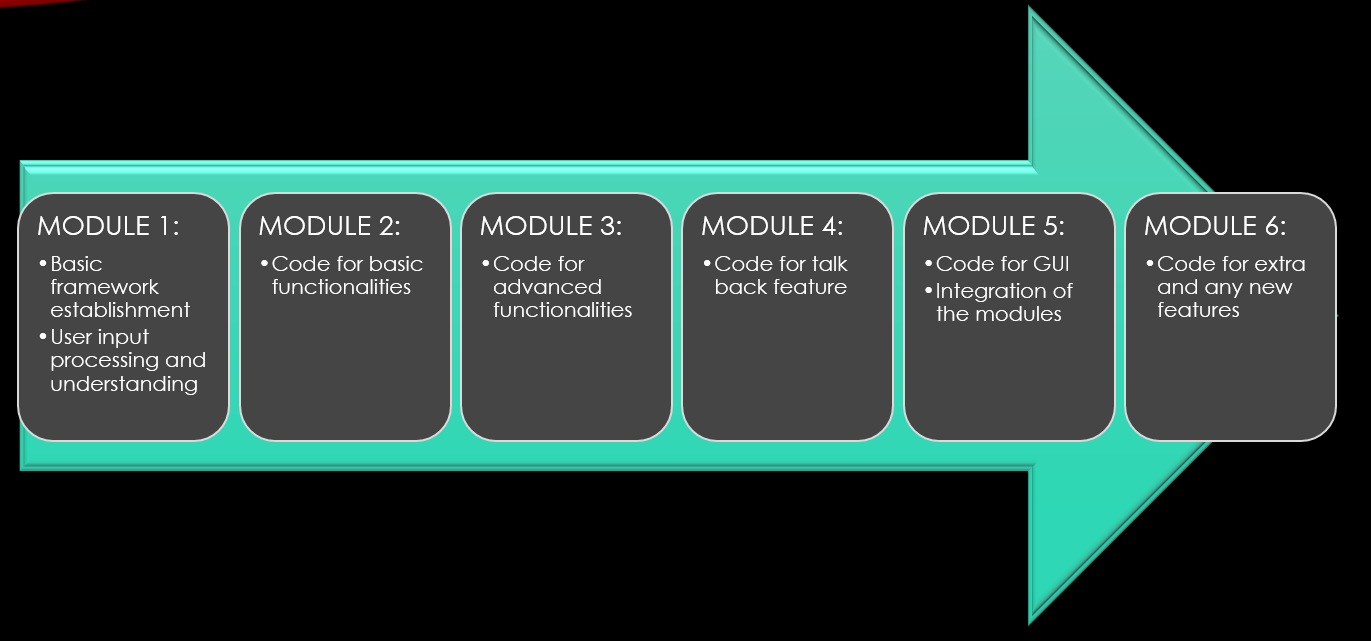


Figure 1

* **MODULE 1:** 
  + Basic framework establishment.
  + Work on prompting and taking user input.
  + Use of speech-recognition for user input through voice.
* **MODULE 2:** 
  + User input statement processing and understanding.
  + Analyzing input.
  + Determining next apparent action.
  + Replying to user with apt statements if user query can be processed or not.
  + And if it can be processed then taking further input for detailed actions if any.
* **MODULE 3:** 
  + Designing code for simple tasks such as application execution like web browser, launch an application, media player, and more commonly used application softwares.
  + Web search by user input.
  + Wikipedia search functionality.
  + Reminder and to-do notes functionality.
  + Sending email facility.
  + Open whatsapp, youtube, stream music, etc.
  + Play or switch audio/video file.
  + Also designing parametric execution of any application if feasible (like google search a topic, etc.).
* **MODULE 4:** 
  + Designing code for complex tasks such as accessing system camera.
  + Using system camera for image capture.
  + Using system camera to record and save video.
  + Performing face-detection on provided image.
  + Performing real-time Face-detection.
  + Simple playable games (made purely in python), etc.
* **MODULE 5:** 
  + Designing code for talk-balk type of feature i.e. replying user’s statements/queries not related to instructing or requesting any task with apt statements.
  + GUI creation.
* **MODULE 6:** 
  + Designing and adding code for any new functionality or feature to the application pertaining to simple o complex functionality after base project completion.
  + Includes designing and integrating code for adding any new feasible feature or function.

**(2.5) SURVEY OF TECHNOLOGY :**

**Python:**

Python is an OOP (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "***batteries included***" language due to its comprehensive standard library.

Python was created in the late 1980s, and first released in 1991, by ***Guido van Rossum*** as a successor to the ABC programming language.

**(2.6) TECHNOLOGIES TO BE USED :**

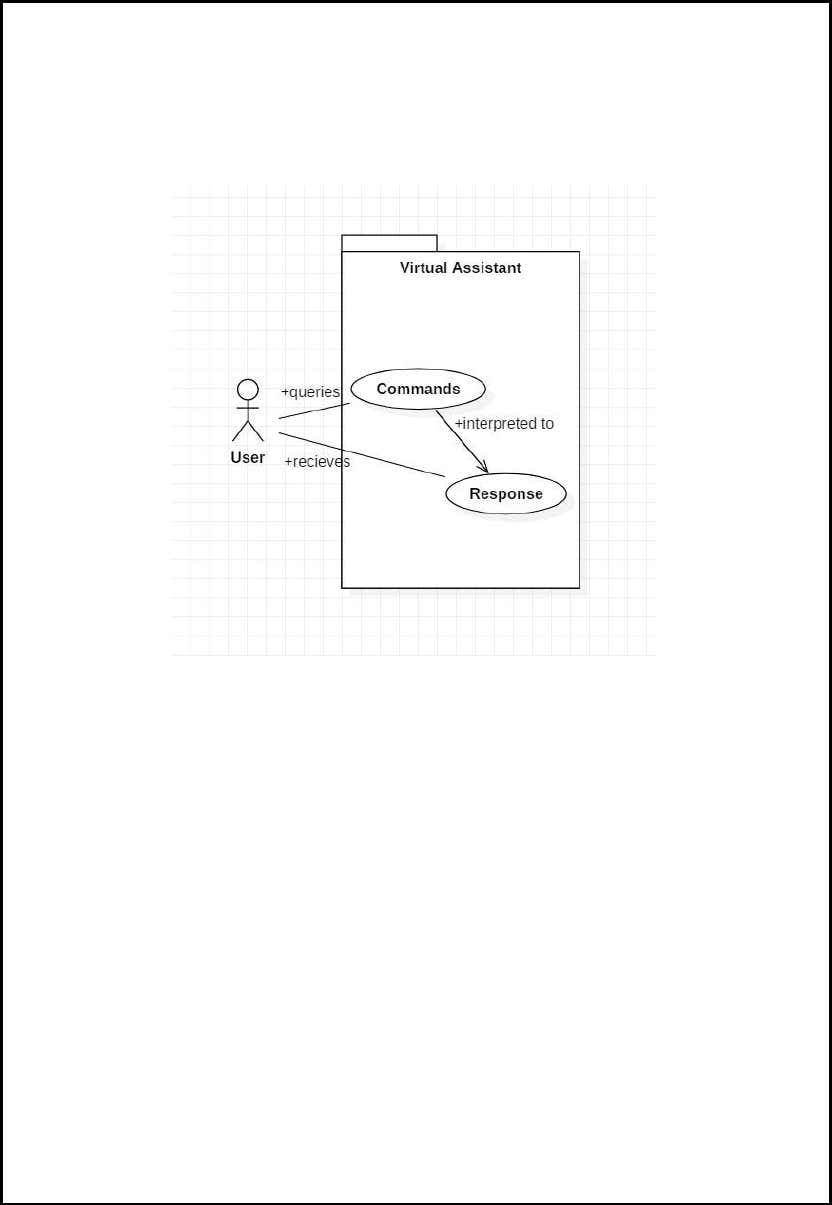
We will be making extensive use of python in developing our software hence technologies that we will be using for our software include Python and the different libraries of it.

And for any new technology that we will integrate with our software we will be only using its required functionalities.

**Packages required:**

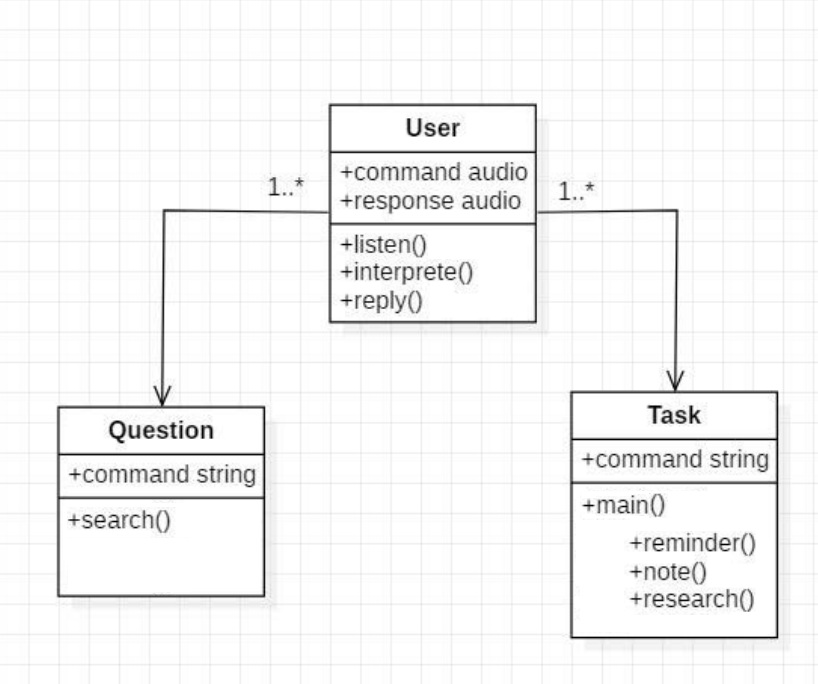
1. **Speech recognition** —The main function of this library is it converts speech to text by processing and understanding whatever the humans speak and converts the speech to text.
2. **pyttsx3** — pyttxs3 is a text to speech conversion library in python. This package supports text to speech engines on Mac os x, Windows and on Linux.
3. **wikipedia** — This package in python extracts information required from Wikipedia.
4. **ecapture** — This module is used to capture images from system camera.
5. **datetime** — This is an inbuilt module in python and it works on date and time.
6. **os** — This module is a standard library in python and it provides the function to interact with operating system and use system commands to operate various tasks.
7. **time**— The time module helps us to display time.
8. **web browser** — This is an in-built package in python. It extracts data from the web, as well provides functions to interact with system default web-browser.
9. **Subprocess** — This is a standard library used to process various system commands like to log off or to restart your PC and to create new processes to launch applications.
10. **Json**- The json module is used for storing and exchanging data.
11. **request**- The request module is used to send all types of HTTP request. Its accepts URL as parameters and gives access to the given URL’S.
12. **wolfram alpha** — Wolfram Alpha is an API which can compute expert-level answers using Wolfram’s algorithms, knowledge base and AI technology. It is made possible by the Wolfram Language.
13. **Pyscreenshot –** Pyscreenshot tries to allow to take screenshots without installing 3rd party libraries.
14. **Psutil** – psutil (process and system utilities) is a cross-platform library for retrieving information on **running processes** and **system utilization** (CPU, memory, disks, network, sensors) in Python.
15. **Smtplib –** This  module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.
16. **Turtle** – It is a pre-installed Python library that enables users to create pictures and shapes by providing them with a virtual canvas and design functions to even control them.
17. **Random** – The random module is a built-in module to generate the pseudo-random variables. It can be used perform some action randomly such as to get a random number, selecting a random elements from a list, shuffle elements randomly, etc.
18. **Pyjokes** – It is a python library that is used to create one-line jokes.
19. **Simple gui** – simplegui is a simplified GUI generator using Tkinter.
20. **Tkinter –** It is the standard GUI library in Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Creating a GUI application using Tkinter is an easy task.
21. **OpenCV** – by using it one can process images and videos to identify objects, faces, or even the handwriting of a human. When it integrated with various libraries, such as Numpuy, python is capable of processing the OpenCV array structure for analysis.
22. **Turtle** – Turtle is a Python library which used to create graphics, pictures, and games. It was developed by **Wally Feurzeig, Seymour Parpet** and **Cynthina Slolomon** in 1967. It was a part of the original Logo programming language.
23. **SYSTEM ANALYSIS AND DESIGN**

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. The complete analysis is followed below.

**(3.1) USE-CASE DIAGRAM :**

Figure

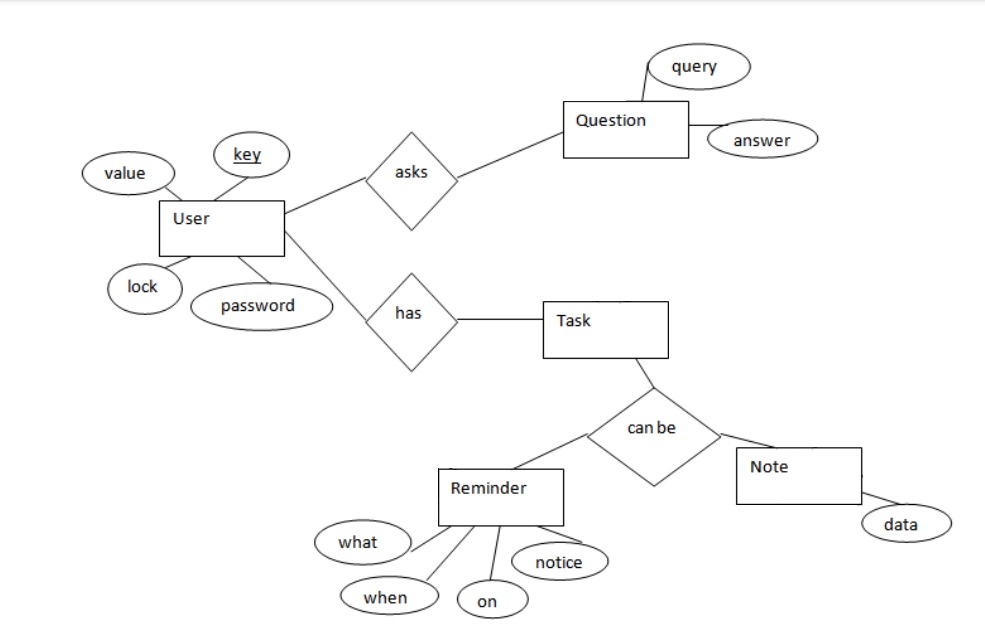
In this project there is only one user. The user queries command to the system. System then interprets it and fetches answer. The response is sent back to the user.

**(3.2) CLASS DIAGRAM :**

Figure

The class user has 2 attributes command that it sends in audio and the response it receives which is also audio. It performs function to listen the user command. Interpret it and then reply or sends back response accordingly.

Question class has the command in string form as it is interpreted by interpret function. It sends it to general or about or search function based on its identification.

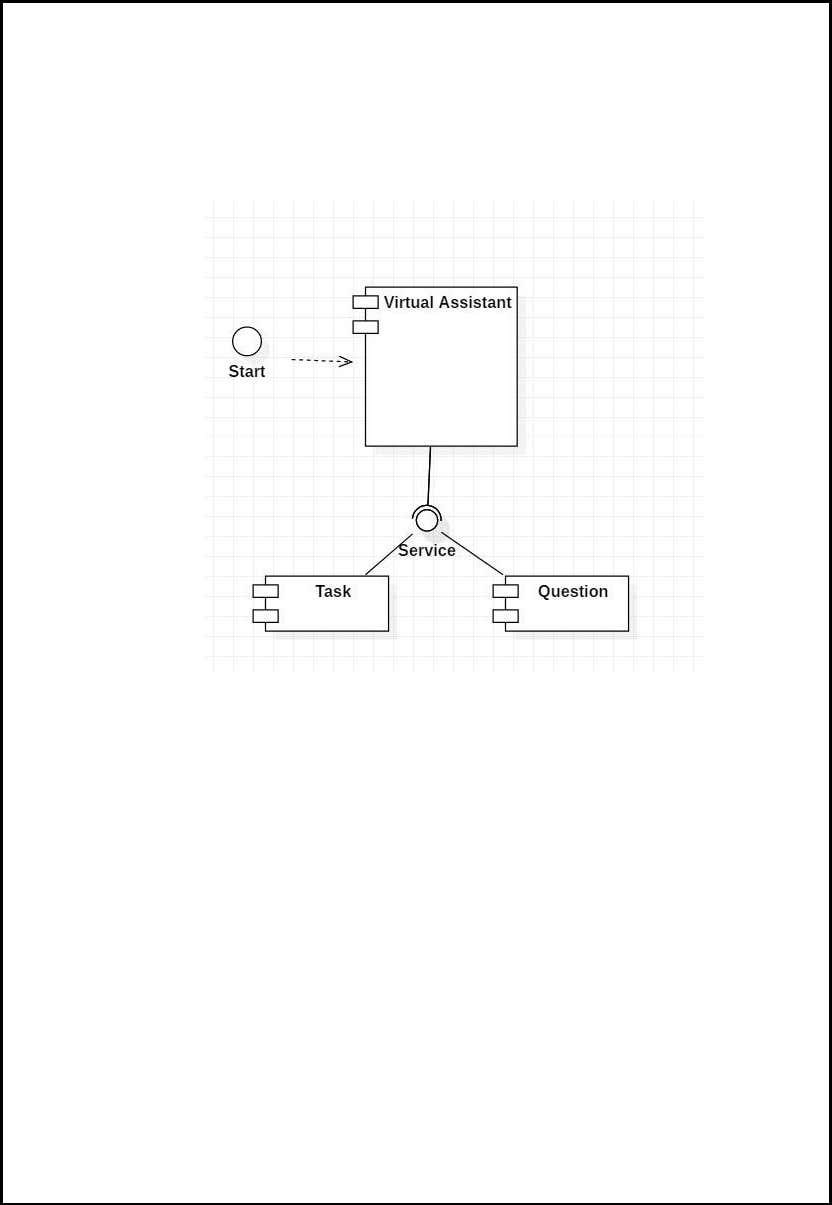
**(3.3) E-R DIAGRAM :**

Figure

The above diagram shows entities and their relationship for a digital assistant system.

We have a user of a system who can have their keys and values. It can be used to store any information about the user. Single user can ask multiple questions. Each question gets recognized in form of query and answer shall be fetched. User can also be having n number of tasks.

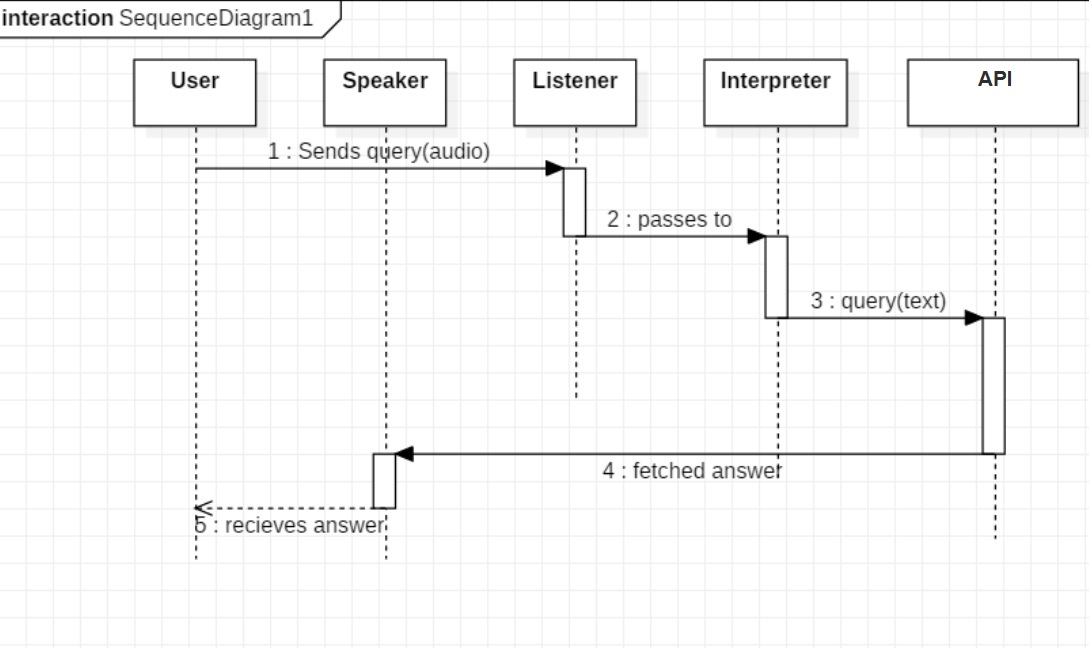
**(3.4) COMPONENT DIAGRAM :**

****

Figure

The main component here is the Virtual Assistant. It provides two specific service: executing Task or Answering your question.

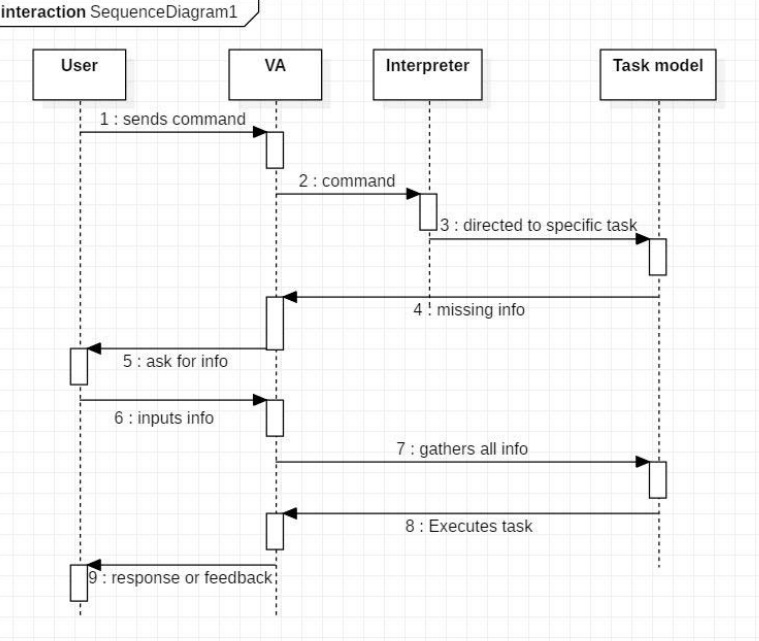
**(3.5) SEQUENCE DIAGRAM :**

***(3.4.1) Sequence diagram for query response***

Figure

The above sequence diagram shows how an answer asked by the user is being fetched from internet. The audio query is interpreted and sent to google API. The API parses speech and sends back the text. Query is then processed in the main function and appropriate action is taken.

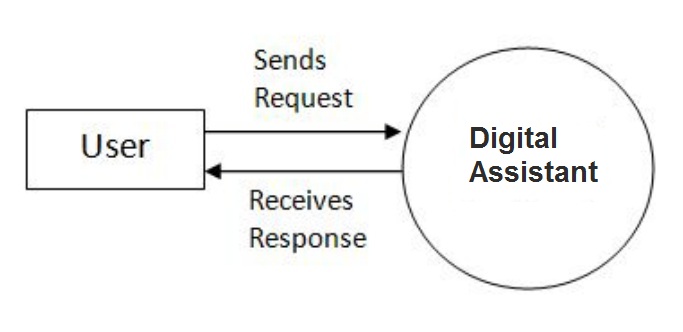
***(3.4.2) Sequence diagram for task execution***



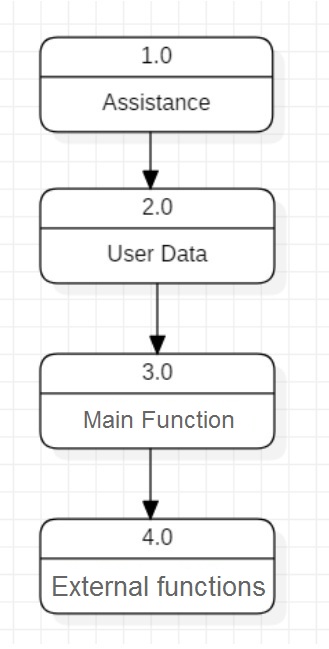
Figure

The user sends command to virtual assistant in audio form. The command is passed to the interpreter. It identifies what the user has asked and directs it to task executer. If the task is missing some info, the virtual assistant asks user back about it. The received information is sent back to task and it is accomplished. After execution feedback is sent back to user.

**(3.6) DATA FLOW DIAGRAM :**

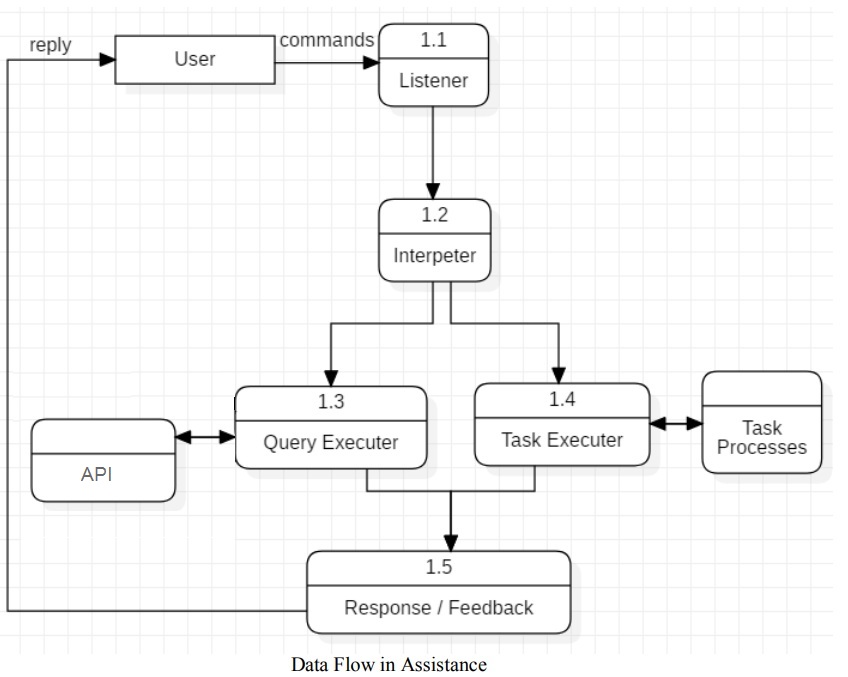
*(3.5.1) DFD Level 0 (Context Level Diagram)*

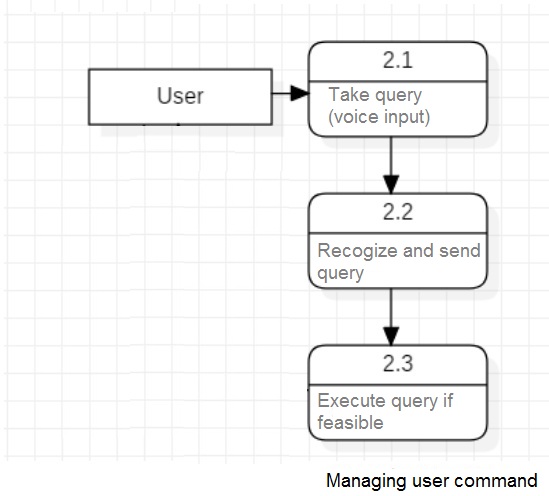
Figure

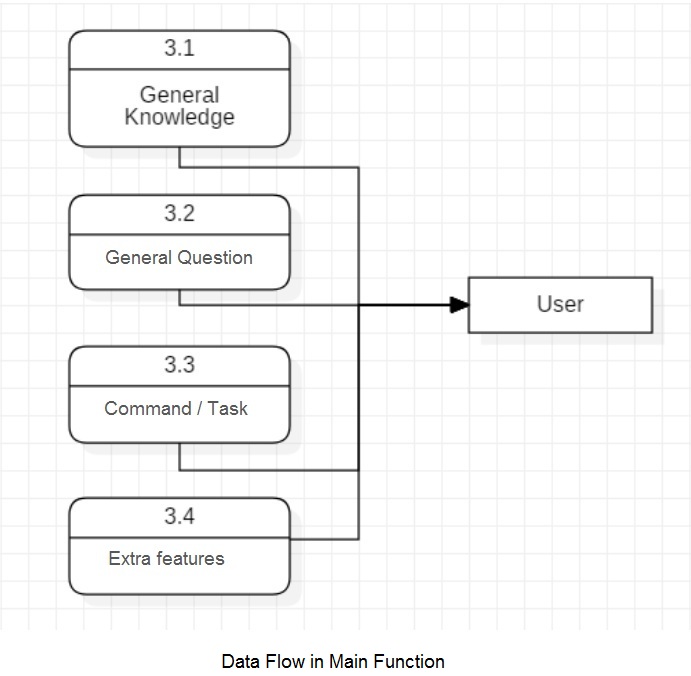
*(3.5.2) DFD Level 1*

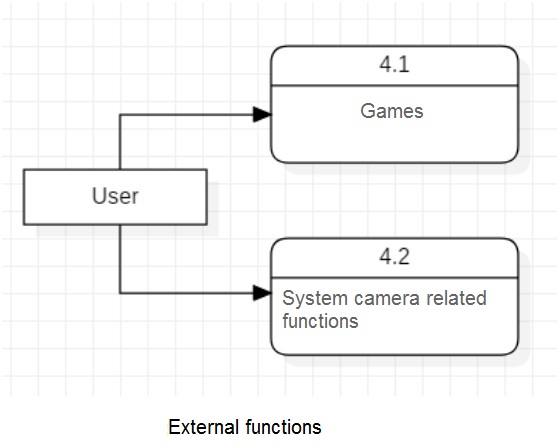
Figure

*(3.5.3) DFD Level 2*



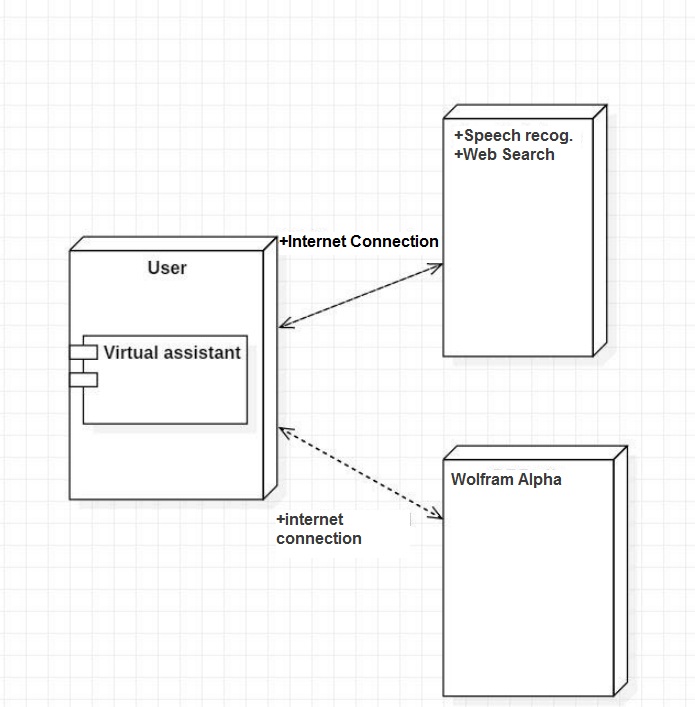






Figure

**(3.7) DEPLOYMENT DIAGRAM :**



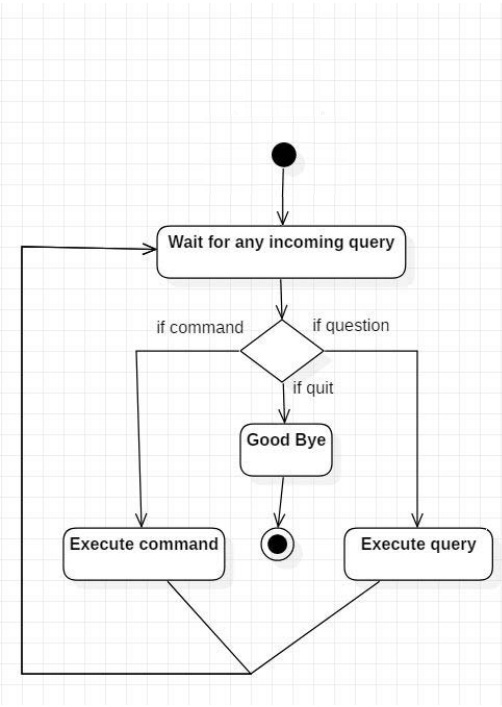
Figure

The user interacts with google API using a normal high speed internet connection.

The knowledge database Wolfram Alpha is used to compute expert level answers to general or random questions asked by the user.

**(3.8) ACTIVITY DIAGRAM :**

Figure



Initially, the system is in idle mode. As it receives any query it begins execution. The received command is identified whether it is a questionnaire or a task to be performed. Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits for another command. This loop continues unless it receives quit command. At that moment, it stops execution permanently.

1. **TEST CASE DESIGN**

**TEST CASE 1:**

**Test Title:** Response Time

**Test ID:** T1

**Test Priority:** High

**Test Objective:** To make sure that the system respond back time is efficient.

**Description:** Time is very critical in a voice based system. As we are not typing inputs, we are speaking them. The system must also reply in a moment. User must get instant response of the query made.

**TEST CASE 2:**

**Test Title:** Accuracy

**Test ID:** T2

**Test Priority:** High

**Test Objective:** To assure that answers retrieved by system are accurate as per gathered data.

**Description:** A virtual assistant system is mainly used to get precise answers to any question asked. Getting answer in a moment is of no use if the answer is not correct. Accuracy is of utmost importance in a virtual assistant system.

**TEST CASE 3:**

**Test Title:** Approximation

**Test ID:** T3

**Test priority:** Low

**Test Objective:** To check approximate answers about calculations.

**Description:** There are times when mathematical calculation requires approximate value. For example, if someone asks for value of PI the system must respond with approximate value and not the accurate value. Getting exact value in such cases is undesirable.

**Note:** There might include a few more test cases and these test cases are also subject to change with the final software development.

1. **RESULTS**

* **TEST CASE 1**

**Test Title:** Response Time

**Test ID:** T1.1

**Test Objective:** To make sure that the system respond back time is efficient for simple non-parametric query.

**Test Steps:**

1. Make query.
2. Record response feedback.

**Test Data:**

1. Search online for “bbd university”
2. Play me a song.

**Expected Test Results:**

1. Default web-browser should open with search results pertaining to “bbd university”.
2. Random song stored in the local directory should start playing in the system default media player.

**Actual Test Results:**

1. As expected.
2. As expected.

**Pass or Fail :** TEST PASSED.

**Test Title:** Response Time

**Test ID:** T1.2

**Test Objective:** To make sure that the system respond back time is efficient for simple parametric query.

**Test Steps:**

1. Make query.
2. Give further input.
3. Record response feedback.

**Test Data:**

1. Open chrome.
   1. bbd university

**Expected Test Results:**

1. Chrome application should open with search results pertaining to “bbd university”.

**Actual Test Results:**

1. As expected.

**Pass or Fail :** TEST PASSED

* **TEST CASE 2**

**Test Title:** Accuracy

**Test ID:** T2

**Test Objective:** To assure that answers retrieved by system are accurate as per gathered data.

**Description:** Getting answer in a moment is of no use if the answer is not correct. Accuracy is of utmost importance in a digital assistant system.

**Test Steps:**

1. Ask a general knowledge question.
2. Record response feedback

**Test Data:**

1. What is the capital of india

**Expected Test Results:**

1. Answer should contain “New Delhi”

**Actual Test Results:**

1. New Delhi, Delhi, India.

**Pass or Fail :** TEST PASSED

* **TEST CASE 3**

**Test Title:** Approximation

**Test ID:** T3

**Test priority:** Low.

**Test Objective:** To check approximate answers about calculations.

**Description:** Getting exact value in certain cases is undesirable.

**Test Steps:**

1. Ask a mathematical general knowledge question.
2. Record response feedback

**Test Data:**

1. What is the the value of pi

**Expected Test Results:**

1. Answer should contain approximate value of pi i.e. not more than 3 significant digits

**Actual Test Results:**

1. Near accurate value of Pi with significant digits exceeding 10 places.

**Pass or Fail :** TEST FAILED.

1. **CONCLUSION AND RECOMMENDATIONS**

Through this voice-controlled automation and control system, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, searching the Wikipedia, streaming music, playing and switching audio and video files on local directory, retrieving weather update details, vocabulary help, medical related queries, email sending automation, open whatsapp, tell random joke, take screenshot, make and retrieve note, make reminder, change desktop background, launch an installed application or system application and also converse with the user. It can also take image and record videos and save them, perform face-detection on image, provide simple interactive games, etc.

We aim to make this project a complete desktop assistant and make it smart enough to be more powerful than keyboard. The future plans include integrating our application to IoT devices and with other software applications for seamless automation and performance.

The digital voice assistant system presented in this project is very fundamental system with few features however the additional and advance feature may be introduced as future work of this project, In this project the design and implementation of a voice based control system digital assistant. The project is built using available open source software modules with python 3.x and its libraries backing which can accommodate any updates in future.

The modular approach used in this project makes it more flexible and easy to integrate additional modules and features without disturbing the current system functionaries. It not only works on human commands but also it is designed for give responses to the user on the basis of query being asked or the words spoken by the user such as opening tasks and operations. This software application has an enormous and limitless scope in the future, like Siri, Google Assistant and Cortana and other popular personal voice assistants. The project will easily able to integrate with devices near future for a Connected Home using Internet of Things, voice command system and computer vision.

1. **REFERENCES**

Documents referred:

1. “Automatic speech recognition technique for voice command”—“ Anshul Gupta; Nileshkumar Patel; Shabana Khan”—2014—<https://ieeexplore.ieee.org/document/7043641>
2. “ways in which new technology could be harnessed to create an intelligent Virtual Personal Assistant (VPA) with a focus on user-based information”—“ Peter Imrie, University of Portsmouth;Peter M. Bednar, Lund University”—“December 2013”<https://www.researchgate.net/publication/264001644_Virtual_Personal_Assistant>
3. “Natural Language Engineering 26(1):129-136
4. DOI:10.1017/S1351324919000640”—“Voice assistance in 2019” —“robert dale”—“jan,2010”<https://www.researchgate.net/publication/338204132_Voice_assistance_in_2019>
5. “Voice Assistant Made Easy”—“September 2020”—“International Journal for Modern Trends in Science and Technology 6(8S):102-107 DOI:10.46501/IJMTSTCIET19”—“Edara Nithin”<https://www.researchgate.net/publication/345931086_Voice_Assistant_Made_Easy>
6. “Conference: ItAIS 2013. The 10th Conference of the Italian Chapter of AISAt”—“Virtual Personal Assistant”—“December 2013—“Authors:Peter Imrie, University of Portsmouth; Peter M. Bednar, Lund University”<https://www.researchgate.net/publication/264001644_Virtual_Personal_Assistant>
7. “Evaluation methodology for Speech To Text Services similarity and
8. “speed characteristics focused on small size computers”—“ J.E. Aguilar-Chacon and D.A. Segura-Torres 2020 IOP Conf. Ser.: Mater. Sci. Eng. 844 012039”<https://iopscience.iop.org/article/10.1088/1757-899X/844/1/012039/pdf>
9. “Facial Recognition using OpenCV—March 2012”—“Authors: Shervin Emami, The University of Queensland; Valentin Petruț, Suciu”<https://www.researchgate.net/publication/267426877_Facial_Recognition_using_OpenCV>
10. “International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072© 2020, IRJET | Impact Factor value: 7.529 | ISO 9001:2008 Certified Journal | A REVIEW ON METHODS FOR SPEECH-TO-TEXT AND TEXT-TO-SPEECH CONVERSION—“Shivangi Nagdewani, Ashika Jain”<https://www.irjet.net/archives/V7/i5/IRJET-V7I5854.pdf>
11. “GENESIS THE DIGITAL ASSISTANT (PYTHON)”—“May 2020”—“DOI:10.33564/IJEAST.2020.v05i01.114”—“Authors: Tushar Bansal, Ritik Karnwal, Vishal Singh Hardik Bansal”<https://www.researchgate.net/publication/343543058_GENESIS-THE_DIGITAL_ASSISTANT_PYTHON>
12. “International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 06 Issue: 01 | Jan 2019 www.irjet.net p-ISSN: 2395-0072
13. © 2019, IRJET | Impact Factor value: 7.211 | ISO 9001:2008 Certified Journal | Page 1550 “—“AI-Smart Assistant”—“Authors: Tushar Gharge, Chintan Chitroda , Nishit Bhagat, Kathapriya Giri”<https://www.irjet.net/archives/V6/i1/IRJET-V6I1288.pdf>
14. “Engaging Students with Game Programming in Python, October 2009”—“Authors: Wang Hong”—<https://www.researchgate.net/publication/44260415_Engaging_Students_with_Game_Programming_in_Python>

Books referred:

* 1. Automate the Boring Stuff with Python, 2nd Edition: Practical Programming for Total Beginners Paperback – by Al Sweigart

1. **APPENDICES**

**(7.1) LIST OF FIGURES :**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | Figure | Description. | PageNo. |
| 1. | Figure 1 | Project development module division diagram | 27 |
| 2. | Figure 2 | Use case diagram | 34 |
| 3. | Figure 3 | Class diagram | 35 |
| 4. | Figure 4 | E-R diagram | 36 |
| 5. | Figure 5 | Component diagram | 37 |
| 6. | Figure 6 | Sequence diagram for query response | 38 |
| 7. | Figure 7 | Sequence diagram for task execution | 39 |
| 8. | Figure 8 | DFD Level 0 diagram | 40 |
| 9. | Figure 9 | DFD Level 1 diagram | 40 |
| 10. | Figure 10 | DFD Level 2 diagram | 41-42 |
| 11. | Figure 11 | Deployment diagram | 43 |
| 12 | Figure 12 | Activity Diagram | 44 |

**(7.2) ABOUT TEHNOLOGIES USED:**

1. **Speech recognition module of python :**

**Speech Recognition** is a library for performing speech recognition, with support for several engines and APIs, online and offline.

Google has a great Speech Recognition API. This API converts spoken text (microphone) into written text (Python strings), briefly Speech to Text. You can simply speak in a microphone and Google API will translate this into written text. The API has excellent results for English language. Google has also created the JavaScript Web Speech API, so you can recognize speech also in JavaScript

we worked with SpeechRecogntion library because of its low barrier to entry and it’s compatibility with much available speech recognition APIs. We can install *SpeechRecogntion* library by running the following line in our terminal window:

pip install SpeechRecognition

Recognizer Class :

SpeechRecognition library has many classes but we will be focusing on a class called Recognizer. This is the class that will help us to convert audio files into text. To access the Recognizer class, first, let’s import the library.

import speech\_recognition as sr

Now, let’s define a variable and assign an instance of recognizer class by calling it.

recognizer = sr.Recognizer()

Now, let’s set the energy threshold to 300. You can think of the energy threshold as the loudness of the audio files. The values below the threshold are considered silent, and the values above the threshold are considered speech. This will improve the recognition of the speech when working with the audio file.

recognizer.energy\_threshold = 300

SpeechRecognition’s documentation recommends 300 as a threshold value which works great with most of the audio files. Also, keep in mind that the energy threshold value will adjust automatically as the recognizer listens to audio files.

Speech Recognition Functions:

Speech Recognition has a built-in function to make it work with many of the APIs out there:

* recognize\_bing()
* recognize\_google()
* recognize\_google\_cloud()
* recognize\_wit()

**Bing Recognizer**function uses Microsoft’s cognitive services.

**Google Recognizer**function uses Google’s free web search API.

**Google Cloud Recognizer**function uses Google’s cloud speech API.

**Wit Recognizer**function uses the wit.ai platform.

We used the *Google Recognizer* function, which is recognize\_google(). It’s free and doesn’t require an API key to use. There is one drawback about this recognizer, it limits you when you want to work with longer audio files.

1. **Smtplib module of python:**

The [smtplib](https://docs.python.org/3/library/smtplib.html" \l "module-smtplib" \o "smtplib: SMTP protocol client (requires sockets).) module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

An [SMTP](https://docs.python.org/3/library/smtplib.html#smtplib.SMTP) instance encapsulates an SMTP connection. It has methods that support a full repertoire of SMTP and ESMTP operations. If the optional host and port parameters are given, the SMTP [connect()](https://docs.python.org/3/library/smtplib.html#smtplib.SMTP.connect) method is called with those parameters during initialization. If specified, *local\_hostname* is used as the FQDN of the local host in the HELO/EHLO command. Otherwise, the local hostname is found using [socket.getfqdn()](https://docs.python.org/3/library/socket.html" \l "socket.getfqdn" \o "socket.getfqdn).

If the [connect()](https://docs.python.org/3/library/smtplib.html#smtplib.SMTP.connect) call returns anything other than a success code, an [SMTPConnectError](https://docs.python.org/3/library/smtplib.html" \l "smtplib.SMTPConnectError" \o "smtplib.SMTPConnectError) is raised.

The optional *timeout* parameter specifies a timeout in seconds for blocking operations like the connection attempt (if not specified, the global default timeout setting will be used). If the timeout expires, [socket.timeout](https://docs.python.org/3/library/socket.html" \l "socket.timeout" \o "socket.timeout) is raised. The optional source\_address parameter allows binding to some specific source address in a machine with multiple network interfaces, and/or to some specific source TCP port. It takes a 2-tuple (host, port), for the socket to bind to as its source address before connecting. If omitted (or if host or port are '' and/or 0 respectively) the OS default behavior will be used.

For normal use, we only require the initialization/connect, [sendmail()](https://docs.python.org/3/library/smtplib.html" \l "smtplib.SMTP.sendmail" \o "smtplib.SMTP.sendmail), and [SMTP.quit()](https://docs.python.org/3/library/smtplib.html" \l "smtplib.SMTP.quit" \o "smtplib.SMTP.quit) methods.

1. **Smtplib module of python:**

The Wolfram|Alpha Webservice API provides a web-based API allowing the computational and presentation capabilities of Wolfram|Alpha to be integrated into web, mobile, desktop, and enterprise applications.

Wolfram Alpha is an API which can compute expert-level answers using Wolfram’s algorithms, knowledgebase and AI technology. It is made possible by the Wolfram Language.

Wolfram Alpha API is free (for non-commercial usage), but we still need to get API key (AppID) to perform queries against the API endpoints.

1. **OpenCV module of python:**

The OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

OpenCV was started at Intel in 1999 by Gary Bradsky, and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel's Russian software OpenCV team. Its active development continued under the support of Willow Garage with Gary Bradsky and Vadim Pisarevsky leading the project. OpenCV now supports a multitude of algorithms related to Computer Vision and Machine Learning and is expanding day by day.

OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc., and is available on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development.

OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language.

1. **Turtle module of python:**

Turtle is a Python library which used to create graphics, pictures, and games. It was developed by **Wally Feurzeig, Seymour Parpet** and **Cynthina Slolomon** in 1967. It was a part of the original Logo programming language.

The turtle module provides turtle graphics primitives, in both object-oriented and procedure-oriented ways. Because it uses [tkinter](https://docs.python.org/3/library/tkinter.html" \l "module-tkinter" \o "tkinter: Interface to Tcl/Tk for graphical user interfaces) for the underlying graphics, it needs a version of Python installed with Tk support.