**PROJECT REPORT**

**ON**

**“Messager-Chatting Application”**

Submitted in partial fulfilment of the requirements for the award of degree of

**MASTER’S OF COMPUTER APPLICATION**

**To**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**UTTRANCHAL SCHOOL OF COMPUTING SCIENCES**

**UTTRANCHAL UNIVERSITY, DEHRADUN**



**(Session: 2022-2023)**

**Under the guidance of Submitted by**

**Mr. Monisha Awasthi Raman Daksh**

**Assistant Professor (MCA-3-B)**

**ACKNOWLEDGEMENT**

There are many people who helped directly or indirectly on the successful completion of my mini project. I would like to thank Dean of USCS “**Dr. Sonal Sharma**” for providing us with all the necessary resources required to complete the project. We profusely are thankful to the program coordinator of the department “**Mr. Sameer Sharma**” for their valuable guidance, inspiring guide and committed caretaker for her unflinching devotion. The encouragement and support by her, especially in carrying out this project motivated me to complete this project. I would like to express deep sense of gratitude to our staff members of Dept. of Computer Applications for their co-operation, which has given in the congruency to build up this project. I would like to thank all our friends for their help and constructive criticism during my project period.

Finally, I am very much indebted to my parents for moral support and encouragement to achieve goals. I have no words to express my gratitude and still I am very thankful to my parents who have shown me this world and for

every support they gave me.

**RAMAN DAKSHI**

**MCA SEMESTER(III)**

**DECLARATION**

I hereby declare that the project report entitled “**Messager”** submitted by **Raman Daksh** to Uttaranchal Institute of Computing Sciences**.** I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this university or any other university or institute.

**RAMAN DAKSHI**

**MCA SEMESTER(III)**

**CERTIFICATE**

This is to certify that the project entitled “**Messager”** submitted by **Raman Daksh** and has been submitted in the partial fulfilment of the requirements for the award of the degree of MCA from Uttaranchal University, Dehradun. The results embodied in this project have not been submitted to any other University or Institution for the record of any degree.

**Under the guidance of:**

Dr. Monisha Awasthi

(Assistant Professor)

**CONTENTS**

[1.INTRODUCTION 6](#_Toc133266572)

[2.OBJECTIVE 7](#_Toc133266573)

[2.1PURPOSE, SCOPE AND APPILCABILITY 8](#_Toc133266574)

[3.SYSTEM ANALYSIS 9](#_Toc133266575)

[3.1 PROBLEM DEFINITION 9](#_Toc133266576)

[3.2 FEASIBILITY STUDY 10](#_Toc133266577)

[3.3 HARDWARE AND SOFTWARE REQUIREMENTS 11](#_Toc133266578)

[4.SYSTEM DESIGN 11](#_Toc133266579)

[4.1 PROJECT PLANNING 11](#_Toc133266580)

[4.2 Methodology of working Virtual Assistant Using Python 12](#_Toc133266581)

[4.3 REQUIREMENT SPECIFICATION(SRS) 14](#_Toc133266582)

[4.4 PROJECT SCHEDULING 15](#_Toc133266583)

[4.4.1 GANTT CHART 15](#_Toc133266584)

[4.4.2 PERT CHART 16](#_Toc133266585)

[5.DATA MODELS 17](#_Toc133266586)

[5.1](#_Toc133266587) [ACTIVITY DIAGRAM 17](#_Toc133266588)

[5.3 USE CASE DIAGRAM 18](#_Toc133266589)

[5.4 SEQUENCE DIAGRAM 19](#_Toc133266590)

[5.5](#_Toc133266591) [COMPONENT DIAGRAM 20](#_Toc133266592)

[6.Testing: Testing techniques and Testing strategies 21](#_Toc133266593)

[6.1 TEST CASE DESIGN](#_Toc133266594) 22

[7. COST ESTIMATION 23](#_Toc133266595)

[7.1 Project Estimation](#_Toc133266596) 24

[7.2 Project Estimation Methods](#_Toc133266597) 25

[7.3 Future Scope Of This Project 26](#_Toc133266598)

[8. APPENDICES](#_Toc133266599) 27

[8.1 Source Code:](#_Toc133266600) 28

[9. CONCLUSION 32](#_Toc133266601)

[10.REFERENCE & BIBLIOGRAPHY](#_Toc133266602) 33

# INTRODUCTION

Messager is a sleek and intuitive Flutter application that allows users to chat seamlessly with their friends and loved ones. Inspired by the iOS design principles. Messager provides a familiar and elegant user experience.

# 2.OBJECTIVE

# 

Messager have the following main objectives: -

1. **Sleek Design:** Messager's design is influenced by the clean and modern aesthetics of iOS, using the Cupertino package to create a native iOS-like feel.
2. **Easy Login Authentication:** Users can create accounts and log in securely using their Google Account.
3. **Real-time Messaging:** Enjoy real-time chat with your contacts. Messages are delivered instantly, creating a seamless conversation experience.
4. **Media Sharing:** Users can Share photos within their chats.
5. **Emoji and Stickers:** Express yourself with a wide range of emojis and stickers that enhance your conversations.
6. **Read Receipts:** Know when your messages are read with read receipts that show when your contacts have seen your messages.
7. **Push Notifications:** Stay connected even when the app is not open, thanks to push notifications that alert you of new messages.
8. **Privacy:** Users data privacy is maintained as the application uses Google Firebase as its storage for storing messages & media and Google authentication for login is a secure method by which only the email, name and google profile picture is used by the application.

## 2.1PURPOSE, SCOPE AND APPILCABILITY

**Purpose**

Purpose of Messager is to provide all the features a messaging app can do along with maintaining the privacy of its users.

**Scope**

**Applicability**

# 3.SYSTEM ANALYSIS

System Analysis is about complete understanding of existing systems and finding where the existing system fails. The solution is determined to resolve issues in the proposed system. It defines the system. The system is divided into smaller parts. Their functions and inter relation of these modules are studied in system analysis. The complete analysis is followed below.

## 3.1 PROBLEM DEFINITION

## 3.2 FEASIBILITY STUDY

Feasibility study can help you determine whether or not you should proceed with your project. It is essential to evaluate cost and benefit. It is essential to evaluate cost and benefit of the proposed system. Five types of feasibility study are taken into consideration.

1. **Technical feasibility:** It includes finding out technologies for the project, both hardware and software. For Messager, user must have internet connection. While using A, make sure you have a steady internet connection. It is also not an issue in this era where almost every home or office has Wi-Fi.2.
2. **Operational feasibility:**
3. **Economic feasibility:**

1. **Organizational feasibility:**
2. **Cultural feasibility:**

## 3.3 HARDWARE AND SOFTWARE REQUIREMENTS

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

**Hardware:**

Any Device which can Run Android OS.

Internet Connectivity. (Network Card)

**Software:**

Android 6 (Min.)

# 4.SYSTEM DESIGN

## 4.1 PROJECT PLANNING

The work started with analysing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer’s internal files, etc. This is an empirical qualitative study, based on reading above mentioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Voice Assistant.

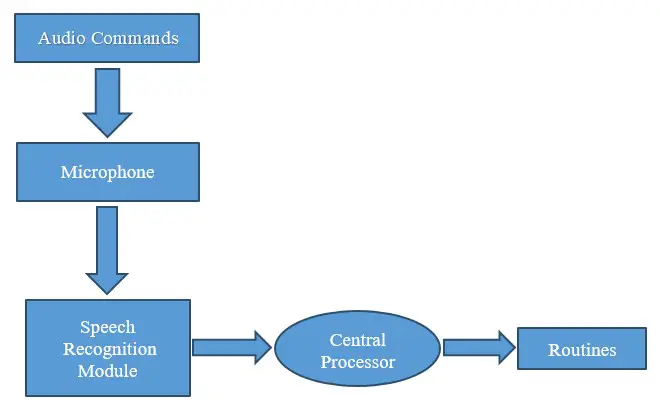


Fig. shows the workflow of the basic process of the voice assistant. Speech recognition is used to convert the speech input to text. This text is then fed to the central processor which determines the nature of the command and calls the relevant script for execution.

## 4.2 Methodology of working Virtual Assistant Using Python

**Speech Recognition module**

The system uses Google’s online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information centre from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

**Python Backend:**

The python backend gets the output from the speech recognition module and then identifies whether the command or the speech output is an API Call and Context Extraction. The output is then sent back to the python backend to give the required output to the user.

**API calls**

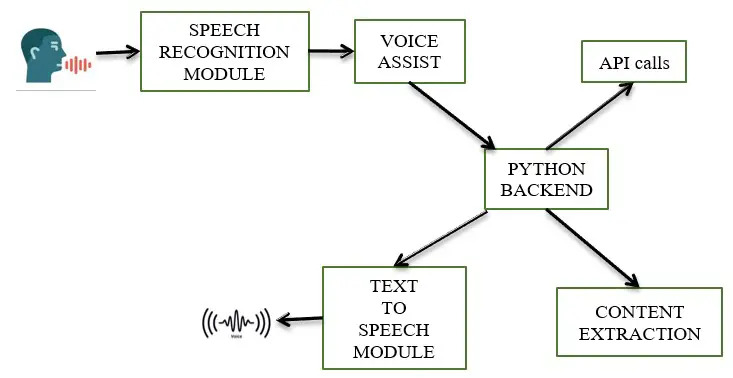
API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is a messenger that delivers your request to the provider that you’re requesting it from and then delivers the response back to you.

**Content Extraction**

Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most cases, this activity concerns processing human language texts using natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as context extraction TEST RESULTS.

**Text-to-speech module**

Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.



## 4.3 REQUIREMENT SPECIFICATION(SRS)

Personal assistant software is required to act as an interface into the digital world by understanding user requests or commands and then translating into actions or recommendations based on agent’s understanding of the world.

JIA focuses on relieving the user of entering text input and using voice as primary means of user input. Agent then applies voice recognition algorithms to this input and records the input. It then use this input to call one of the personal information managementapplications such as task list or calendar to record a new entry or to search about it on searchengines like Google, Bing or Yahoo etc. Focus is on capturing the user input through voice,recognizing the input and then executing the tasks if the agent understands the task. Softwaretakes this input in natural language, and so makes it easier for the user to input what he or shedesires to be done.

Voice recognition software enables hands free use of the applications, lets users toquery or command the agent through voice interface. This helps users to have access to theagent while performing other tasks and thus enhances value of the system itself. JIA also haveubiquitous connectivity through Wi-Fi or LAN connection, enabling distributed applicationsthat can leverage other APIs exposed on the web without a need to store them locally.

Virtual assistants must provide a wide variety of services. These include:

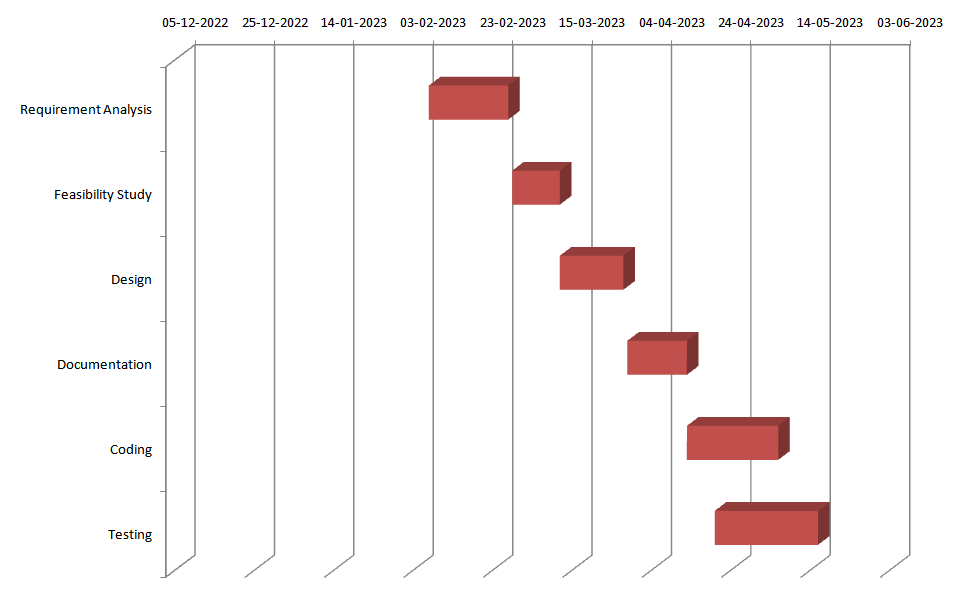
* Providing information such as weather, facts from e.g. Wikipedia etc
* Set an alarm or make to-do lists and shopping lists.
* Remind you of birthdays and meetings.
* Play music from streaming services such as Saavn and Gaana.
* Play videos, TV shows or movies on televisions, streaming from e.g. Netflix orHotstar.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SDLC Phase** | **Project Activity** | **Duration** | **Start** | **Finish** |
| **1** | Requirement Analysis | 20 days | 06-02-2023 | 25-02-2023 |
| **2** | Feasibility Study | 12 days | 25-02-2023 | 08-03-2023 |
| **3** | Design | 16 days | 09-03-2023 | 25-03-2023 |
| **4** | Documentation | 15 days | 26-03-2023 | 09-04-2023 |
| **5** | Coding | 23 days | 10-04-2023 | 03-05-2023 |
| **6** | Testing | 26 days | 11-04-2023 | 15-05-2023 |

## 4.4 PROJECT SCHEDULING

### 4.4.1 GANTT CHART

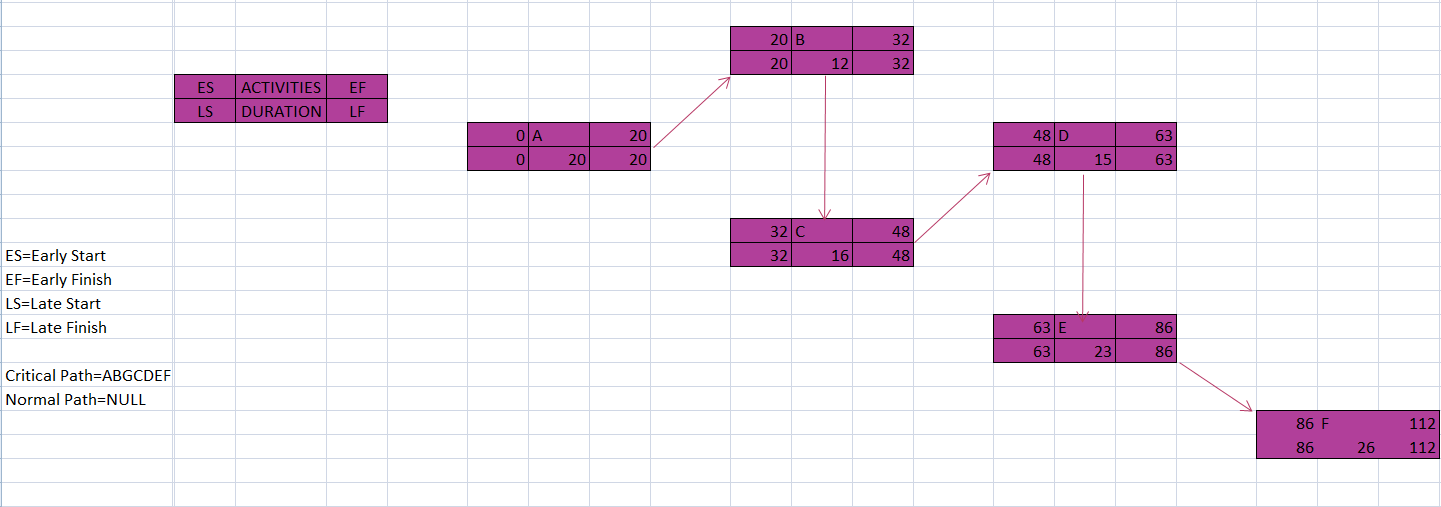
**TABLE 1: GANTT CHART**

****

### 4.4.2 PERT CHART

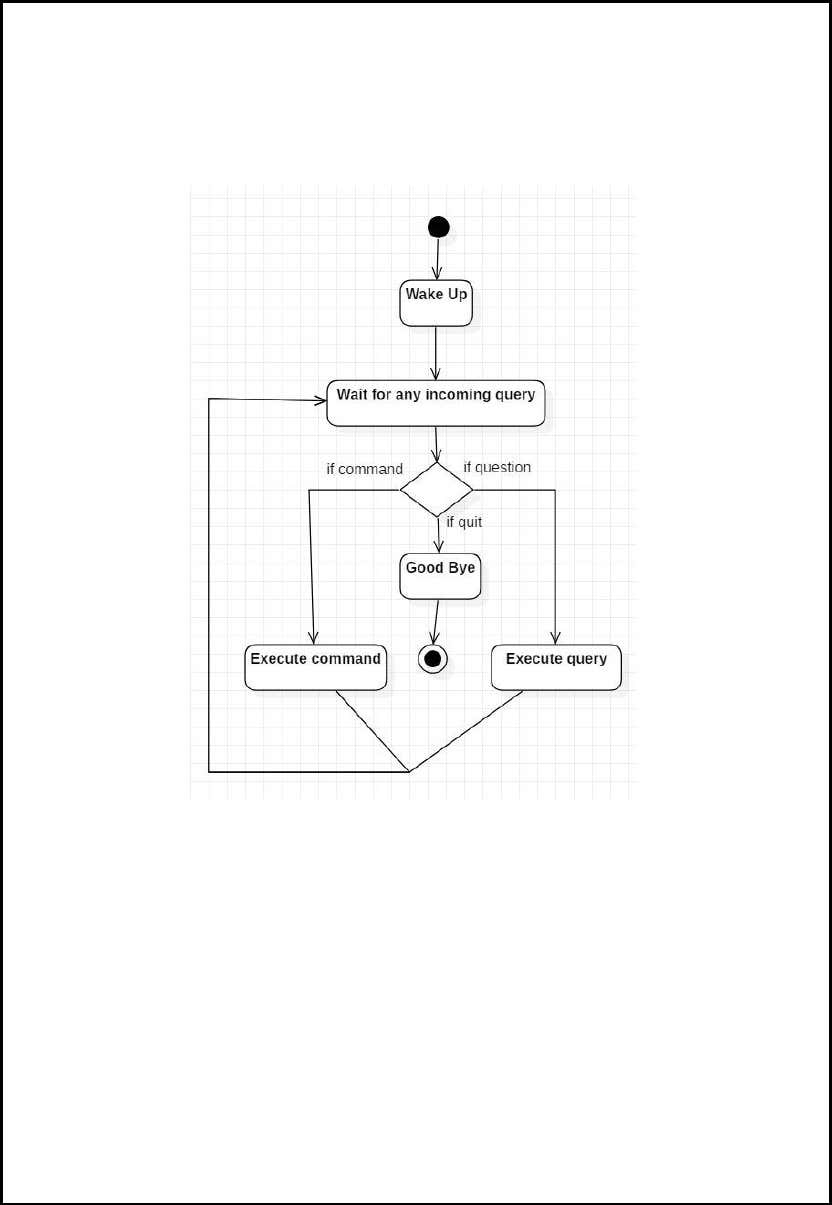
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SDLC Phase** | **Project Activity** | **Duration** | **Start** | **Finish** |
| **1** | Requirement Analysis | 20 days | 06-02-2023 | 25-02-2023 |
| **2** | Feasibility Study | 12 days | 25-02-2023 | 08-03-2023 |
| **3** | Design | 16 days | 09-03-2023 | 25-03-2023 |
| **4** | Documentation | 15 days | 26-03-2023 | 09-04-2023 |
| **5** | Coding | 23 days | 10-04-2023 | 03-05-2023 |
| **6** | Testing | 26 days | 11-04-2023 | 15-05-2023 |

**TABLE 2: PERT CHART**

****

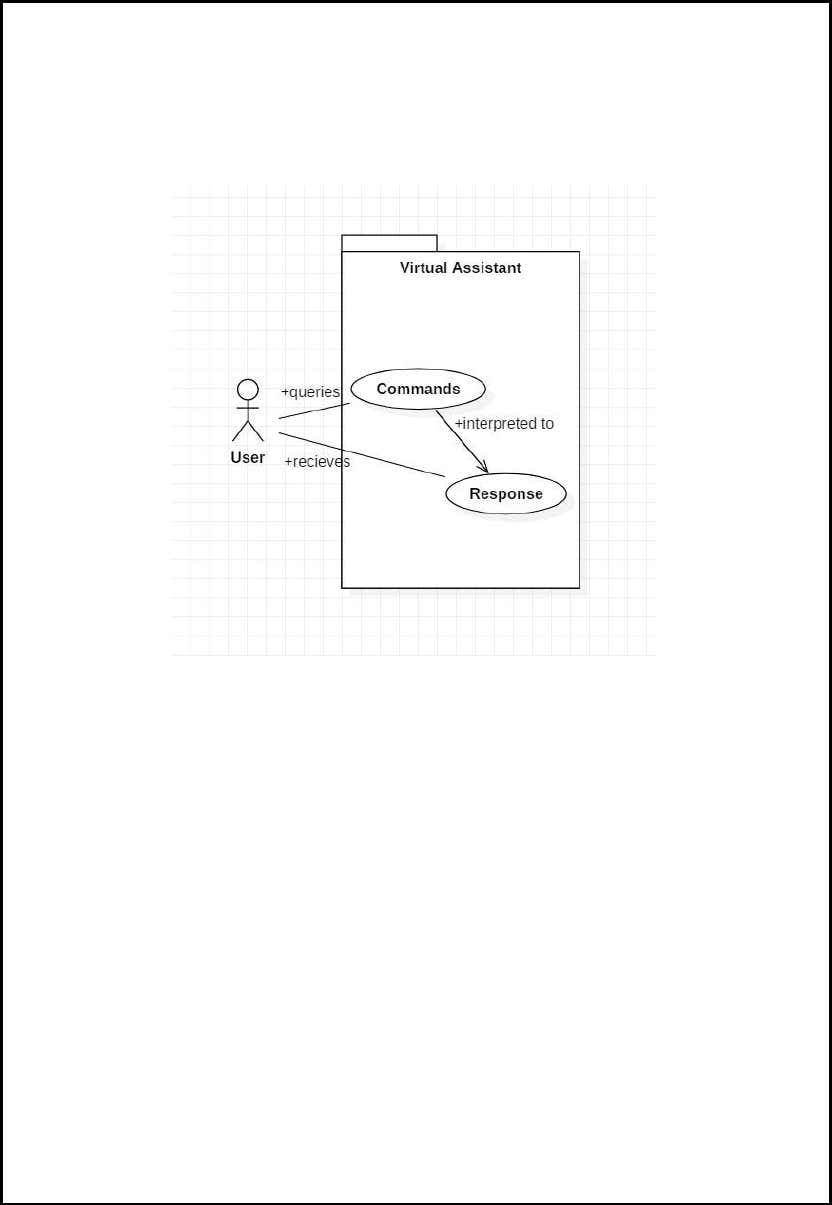
# 5.DATA MODELS

## 5.1 ACTIVITY DIAGRAM



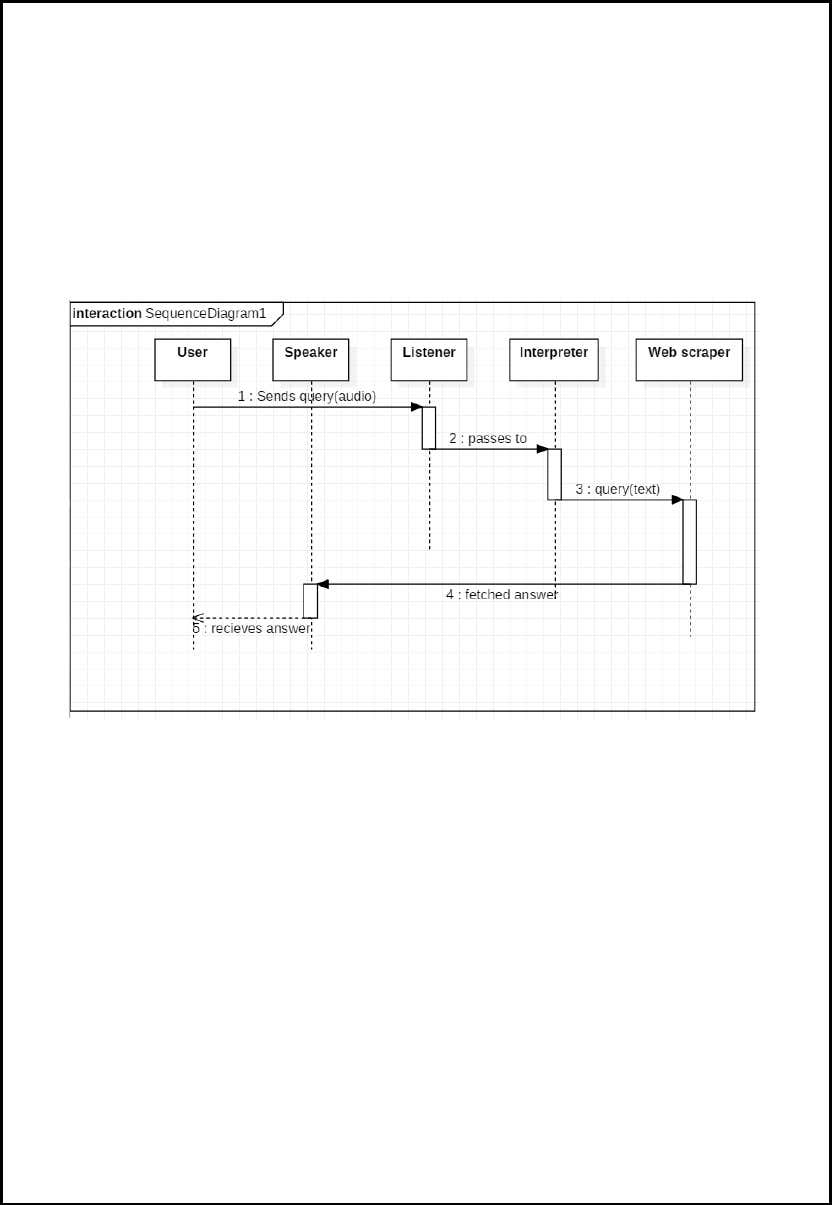
Initially, the system is in idle mode. As it receives any wake up cal it begins execution.The received command is identified whether it is a questionnaire or a task to be performed.Specific action is taken accordingly. After the Question is being answered or the task is being performed, the system waits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

## 5.2 USE CASE DIAGRAM



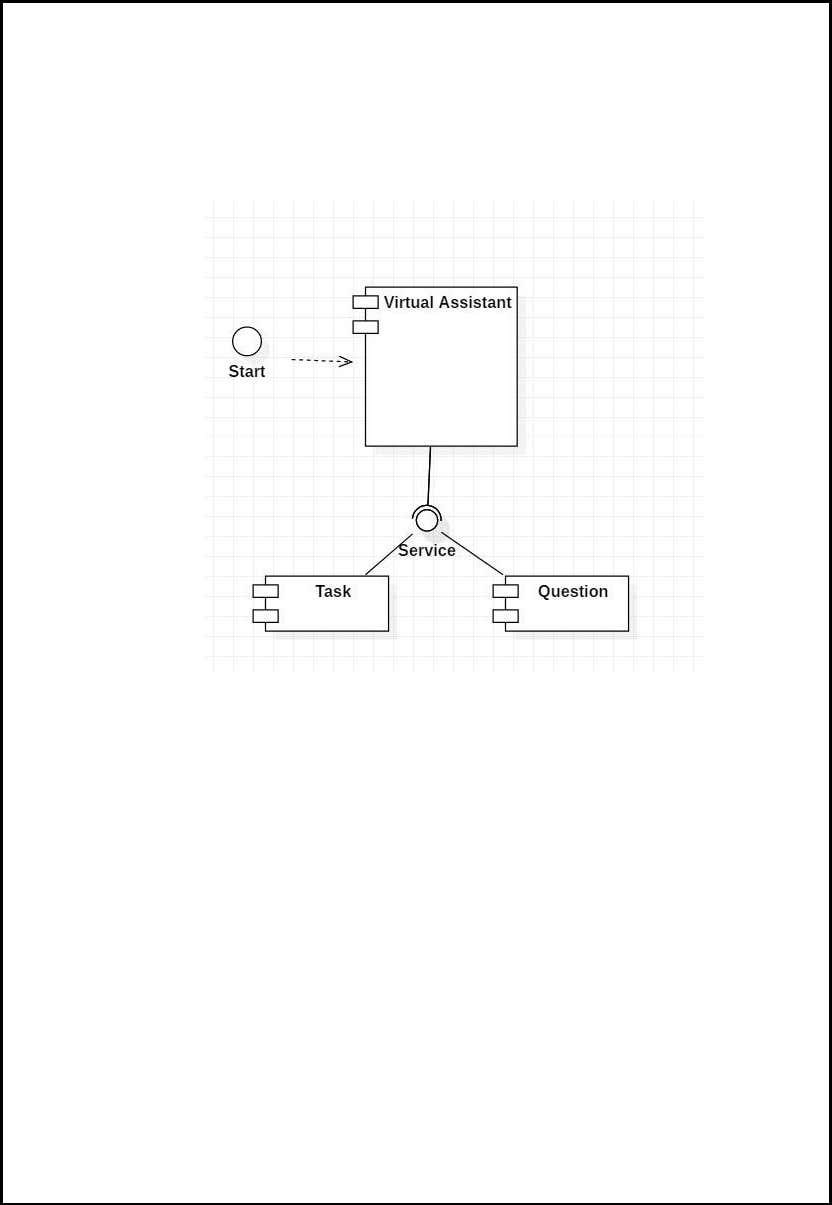
In this project there is only one user. The user queries command to the system. Systemthen interprets it and fetches answer. The response is sent back to the user.

## 5.3 SEQUENCE DIAGRAM



The above sequence diagram shows how an answer asked by the user is being fetchedfrom internet. The audio query is interpreted and sent to Web scraper. The web scrapersearches and finds the answer. It is then sent back to speaker, where it speaks the answer touser.

## 5.4 COMPONENT DIAGRAM



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

# 6.TESTING: TESTING TECHNIQUES AND TESTING STRATEGIES

**Unit Testing:**   
Unit Testing is the type of [software testing](https://www.geeksforgeeks.org/software-testing-basics/) level in which each individual component of the software is tested. Unit Testing is generally performed by the developer. Unit Testing can’t be used for those systems which have a lot of interdependence between different modules. It does not allow for parallel testing.

**System Testing:**   
System testing is done to check whether the software or product meets the specified requirements or not. It is done by both testers and developers. It contains System testing and Integration testing. It is done through more positive and negative test cases.

Testing Messagers to iterate and improve the NLU model is a critical step in the virtual assistant life cycle. Before you launch your virtual assistant to production, before folks are actually using it, you want to make sure that it works. It's pretty simple, but testing is so important.

There are a variety of ways in which you can go about your testing because ultimately when you do get to production you want to deliver an experience that people love and really enjoy using.

## 6.1 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 arespeaking them. The system must also reply in a moment. User must get instant response of thequery 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 utmostimportance in a virtual assistant system.

**Test Case 3**

Test Title:Approximation

Test ID:t3

Test priority:Moderate

Test Objective:To check approximate answers about calculations.

Description:There are times when mathematical calculation requires approximate value. Forexample, if someone asks for value of PI the system must respond with approximate value andnot 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 changewith the final software development.

# 7. COST ESTIMATION

## 7.1 Project Estimation

There are several common IT project cost estimation models, also called methods. There is no one-size-fits-all solution; rather, select a method by identifying your project’s needs using IT heuristics, such as project size, complexity, necessary resources, and preparation time.

Some models base the estimate on the smallest building block, such as a line of code, while others compare the average size of previous projects. Use the following heuristics as you consider which model best serves your project:

**Complexity:** Estimate the project’s complexity. Measure both subjective (based on your past experience) and objective (such as limited resources) factors.

**Method Reliability:** While there is not necessarily a preferred method, some methods are more reliable than others in their estimated accuracy.

**Necessary Resources:** Some projects have a wealth of historical data that is easily accessible through public databases. Others may have inconsistent or incomplete data, which will impact your ability to conduct thorough research. You may also run into personnel ability issues, such as working with a new or inexperienced team.

**Preparation Time:** Some methods are more time-consuming to prepare. Your project schedule, size, and complexity influence operational time constraints within your allotted schedule.

**Project Size:** Measure the project size using one of several common methods, such as function points (a unit of measurement to approximate the functional business value of a product), T-shirt sizing (an estimation tool defining the relative size of work), or lines of code.

## 7.2 Project Estimation Methods

There are two categories of IT project estimation methods: algorithmic and non-algorithmic. Algorithmic methods involve an equation to quantify the effort. Non-algorithmic methods use data and analysis. Use at least two methods in your cost estimation to increase accuracy.

**Algorithmic Methods**

Algorithmic estimation methods are formalized estimation models. These methods use a mechanical process, most often a formula created from patterns in historic data, and are the most standardized.

**Constructive Cost Model (COCOMO):** This method estimates effort, cost, and schedule using a three-tiered process: basic, intermediate, and detailed. It uses complex algorithmic formulas derived from historic project data to construct. This parametric method is one of the most accurate techniques for estimating project costs and often used for incremental software development.

**Function Point (FP):** This is the industry standard for sizing software. FPs are a measurement unit to show how functional an information system is to the user in the business. It uses past projects to calculate the hourly or monetary cost.

**Enhanced Version of Intermediate COCOMO (REVIC)**:Revised This method modifies the COCOMO process by including testing, integration, and maintenance in the publicly available copyrighted program. It is primarily used for military projects.

**Source Lines of Code (SLOC):** Also called lines of code (LOC), this method counts the number of all source code command lines as the base unit metric of the project size. Many complex software methods use this method as the base.

**Weighted Micro Function Points (WMFP):** This method calculates the size of the project by parsing the program code into micro functions to create volume metrics and code complexity. Teams calculate a final effort score with the WMFPs. This method is compatible with Waterfall software development lifecycle, Agile, and Six Sigma. The estimate results are more accurate than traditional sizing methods.

**Non-Algorithmic Methods**

Non-algorithmic estimations analyze assembled historic data to infer the best estimate.

**Activity-Based Costing**: This method considers the activities of the people and equipment necessary to deliver the product. This style evaluates the indirect costs in proportion to the type of work, as well as the resource demand of each activity.

**Analogy:** This method uses previous projects to compare and estimate the cost, size, complexity, or schedule. The method takes into account the similarities, differences, and actual results. You do not need an expert for this method, but must have fully accurate historic data.

**Bottom-Up:** This technique starts the estimate from the low-level details of the work breakdown structure (WBS) and builds up to a higher level. This method provides a high level of accuracy due to its attention to detail.

**Delphi:** This is a structured and systematic Expert Judgment method subset to forecast with an expert panel. The panel weighs in with opinions and debates the factors that influence the estimate. The method assumes that the accuracy from structured group interactions is more reliable than from unstructured group interactions. Other subset adaptations of Delphi are available, like the mini-Delphi or the estimate-talk-estimate (ETE). This method removes politics and bias from the estimate, but it may be time-consuming.

**Expert Judgment (EJ):** In this method, you consult experts who have extensive experience and have completed similar projects. This method can be highly subjective, inconsistent, and unstructured, which poses risks to accuracy.

Planning Poker: In this gamified method, teams use a card deck to build consensus. This helps to avoid anchoring, which is a bias toward the first estimate. This method is a variation of the Wideband Delphi, most often used in Scrum and XP in Agile development.

**Process Group:** This is another Wideband Delphi variant. Here, the estimation team collectively creates a WBS and details any assumptions. From this meeting, individual team members estimate effort, and use these projections in a second meeting, where they reach a consensus.

**Relative Sizing (T-Shirt Sizing and Fibonacci Sequence):** Here, the team breaks down work tasks and sizes them in relative values to each other. These methods are popular in Agile software development, especially when there is a significant amount of ambiguity. These methods use an iterative process to quickly assemble historic data from the team’s past projects to improve estimate accuracy over time. The most common methods are T-Shirt sizing and Fibonacci sequence sizing, which estimate the relative sizes of T-shirts (XS, S, M, L, XL) or the Fibonacci sequence (1, 2, 3, 5, 8, 13). This method provides a quick estimate, which allows the team to get started more quickly. However, this method requires a greater degree of management trust up front, since accuracy only improves by continually using this method over time.

Teams break down their work into tasks, or story points, and then place them on an empty chart or graph based on the amount of effort and risk in relation to the other tasks. At this stage, only the facilitator knows the specific sizing numbers to keep the team focused on the relative size to each other. Once all story points are placed on the chart, the facilitator reveals the sizing underneath. The team then organizes the assigned points and estimates how long the project will take (or how much work can be completed in a given cycle, or sprint) based on the team’s capacity.

## 7.3 Future Scope Of This Project

Virtual assistants are making a huge splash, predominantly in the consumer and social spaces, as people become reliant on them to perform tedious tasks quickly and efficiently. As of mid-2017, there were an estimated 1 billion users of virtual assistants worldwide and that number is rapidly increasing. In addition, Skills, the applications that extend the power of the virtual assistants, are being developed at a fast pace by third-parties--Amazon’s Alexa already has over 50,000. Opportunities for new use cases across industries continue to reveal themselves as the virtual assistant user base grows and new Skills hit the market.

**Virtual Assistants, Today and Tomorrow**

Virtual assistant technology really came into its own in 2017, and as we journey into 2018, more promise is on the horizon. So, what can we expect from virtual assistants in the coming years?

**SMARTER VIRTUAL ASSISTANTS**

Much of what virtual assistants do now are basic skills, such as retrieving data and basic computation. As natural language processing (NLP) continues to mature, virtual assistants will improve their comprehension and response capabilities, allowing for their use to become more widespread and complex. Also, as machine learning progresses,, we may see virtual assistants become smarter and begin to learn and predict customer needs.

**PARTNERSHIPS**

Currently, there are several different companies developing virtual assistant technology, each targeting their own devices and hardware. For example, Microsoft’s Cortana works best with Windows 10 devices, Amazon’s Alexa works best with Amazon Devices, and Google Home uses its own platform. Consumers haven’t be able to transition between multiple platforms easily. In 2017, the first unexpected partnership developed between Microsoft and Amazon to integrate Cortana and Alexa. Alexa leads the household market, while Cortana leads the business market. This partnership may be the first step to seamless, integrated experience for customers as they move through different environments. Imagine relaxing on your couch at the end of a long day when you realize you need to reschedule a work meeting. No problem! “Alexa, ask Cortana to move my 8AM meeting to 9AM”. Done.

**INTEGRATION WITH IOT DEVICES**

Car speakers, smart home devices, and wearables are just a few examples where the virtual assistant is departing from its original hardware and making its way to in-context devices. These integrations ensure that virtual assistants can always be near their human and ready to support any need. It is expected that these integrations will continue at an accelerated pace throughout 2018.

**Bring Your Virtual Assistant to Work – the future state**

As consumers become more reliant on their virtual assistants in their personal lives, it’s expected that demand will increase for virtual assistants in the workplace, with a delineation between the consumer market and enterprise. Virtual

assistants are already emerging in the workplace as taskmasters, completing repetitive tasks and freeing up time for workers to focus on more complex tasks.

**PERSONAL ASSISTANT**

Have you ever tried to arrange a meeting with three other people with busy calendars? It’s a time-consuming and frustrating job. Virtual assistant technology can extend to the personal assistant realm and handle these time consuming administrative tasks, leaving workers more time to focus on high-value tasks.

**DICTATION AND TRANSLATION**

Recording a brainstorming session or dictating a message is becoming possible thanks to natural language process (NLP) improvements. Even better, speak with a business associate in another country and a virtual assistant will translate the conversation in real-time.

**VOICE-CONTROL OF MACHINES**

In the manufacturing world, many machines are starting to be operated through voice-control. In the office setting, it’s possible that IT solutions could be run in the same fashion. Soon, we could be joining meetings with a voice command, instead of dialing in the long meeting ID and password.

**INTEGRATION WITH OTHER TECHNOLOGIES**

Integrating age and gender detection with other technologies such as augmented reality or virtual reality could also be an interesting area for future development. For example, age and gender detection could be used to create more personalized experiences in virtual reality games or augmented reality marketing campaigns.

**IMPROVED ACCURACY**

One of the primary goals for future development would be to improve the accuracy of age and gender detection algorithms. This could involve using deep learning techniques or incorporating more advanced features to better identify gender and age-related characteristics.

# 8. APPENDICES

## 8.1 Source Code:

import pyttsx3

import speech\_recognition as sr #pip install speechRecognition

import datetime

import wikidata #pip install wikidata

import webbrowser

import os

import smtplib

import pyaudio

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices') # print(voices[1].id)(There are two voices in System Male and Female)

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

def speak(audio):

engine.say(audio)

engine.runAndWait()

def wishMe():

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

if hour>=0 and hour<12:

speak("Good Morning!")

elif hour>=12 and hour<18:

speak("Good Afternoon!")

else:

speak("Good Evening!")

speak("I am Jarvis Sir. Please tell me how may I help you")

def takeCommand():

#It takes microphone input from the user and returns string output

r = sr.Recognizer()

with sr.Microphone() as source:

print("Listening...")

r.pause\_threshold = 1

audio = r.listen(source)

try:

print("Recognizing...")

query = r.recognize\_google(audio, language='en-in')

print(f"User said: {query}\n")

except Exception as e:

# print(e)

print("Say that again please...")

return "None"

return query

def sendEmail(to, content):

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

server.ehlo()

server.starttls()

server.login('tiwarianku2001@gmail.com', 'Tiwari2001')

server.sendmail('tiwarianku2001@gmail.com', to, content)

server.close()

if \_\_name\_\_ == "\_\_main\_\_":

wishMe()

while True:

# if 1:

query = takeCommand().lower()

# Logic for executing tasks based on query

if 'wikipedia' in query:

speak('Searching Wikipedia...')

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

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

speak("According to Wikipedia")

print(results)

speak(results)

elif 'open youtube' in query:

webbrowser.open("youtube.com")

elif 'open google' in query:

webbrowser.open("google.com")

elif 'open stackoverflow' in query:

webbrowser.open("stackoverflow.com")

elif 'play music' in query:

music\_dir = 'D:\\songs\\bhakti'

songs = os.listdir(music\_dir)

print(songs)

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

elif 'the time' in query:

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

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

elif 'open code' in query:

codePath = "C:\\Users\\ankit\\AppData\\Local\\Programs\\Microsoft VS Code\\Code.exe"

os.startfile(codePath)

elif 'email to Ankit' in query:

try:

speak("What should I say?")

content = takeCommand()

to = "tiwarianku2001@gmail.com"

sendEmail(to, content)

speak("Email has been sent!")

except Exception as e:

print(e)

speak("Sorry my friend . I am not able to send this email")

# 9. CONCLUSION

In this Project “ Messager Using Python” we discussed the design and implementation of Digital Assistance. The project is built using open source software modules with PyCharm community backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easy to add additional features without disturbing current system functionalities.

It not only works on human commands but also give responses to the user based on the query being 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 input rather than the next one.

# 10.REFERENCE & BIBLIOGRAPHY

**Websites referred:**

* www.stackoverflow.com
* www.pythonprogramming.net
* www.codecademy.com
* www.tutorialspoint.com
* www.google.co.in

**Books referred:**

* Python Programming - Kiran Gurbani
* Learning Python - Mark Lutz

**YouTube Channels referred:**

* CS Dojo
* edureka!

**Documents referred:**

* Designing Personal Assistant Software for Task Management using SemanticWeb Technologies and Knowledge Databases- Purushotham Botla
* Python code for Artificial Intelligence: Foundations of Computational Agents- David L. Poole and Alan K. Mackworth