

VOICE BASED EMAIL SYSTEM FOR VISUALLY IMPAIRED

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

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the supervision of

Mr. Yashdeep Singh
Assistant Professor (Grade-1)

By

Saurav Mishra (141318)
Ashwani Panwar (141294)



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT, SOLAN – 173234
HIMACHAL PRADESH, INDIA

CONTENTS

- i. Candidate declaration
- ii. Acknowledgement
- iii. List of figure
- iv. Abstract

| S.No. | Topic | Page No |
|--------------|--|----------------|
| 1. | Introduction | 1 |
| 1.1 | Project Objectives | 6 |
| 1.2 | IVR(Interactive Voice Response) | 8 |
| 1.3 | Speech Recognition | 10 |
| 1.4 | Speech to text | 12 |
| 1.5 | Text to speech | 14 |
| 2. | Literature Review | 16 |
| 2.1 | Voice Based System in Desktop and Mobile Devices for Blind People. | 16 |
| 2.2 | Voice Based Search Engine and Web page Reader | 17 |
| 3. | System Development | 19 |
| 3.1 | Proposed Model | 19 |
| 3.2 | Existing System | 20 |
| 3.3 | Design | 22 |
| 4. | Test Plan And Performance | 38 |
| 4.1 | Tools Used | 39 |
| 4.2 | Software/hardware Requirements | 39 |
| 5. | Conclusion | 40 |
| 5.1 | Application | 40 |
| 5.2 | Future Scope | 40 |
| 5.3 | References | 41 |

List Of Figures

- 1.Speech to text
- 2.Speech recognition
- 3.System block diagram(speech to text)
- 4.Text to speech diagram
- 5.Pyttsx code demo
- 6.gTTS code demo
- 7.Voice recognition diagram from research paper
- 8.Flow chart of the proposed model
- 9.smptplib code demo
- 10.imaplib code demo

Candidate's Declaration

I hereby declare that the work presented in this report entitled "**Voice Based Email System For Visually Impaired**" submitted at **Jaypee University of Information Technology, Wagnaghat India**, is a genuine record of my work carried out under the guidance of **Mr. Yashdeep Singh (Assistant Professor, Computer Science and Engineering)**. I have not submitted this work anywhere else for any other degree.

Saurav Mishra
(141318)

Ashwani Panwar
(141294)

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

Mr. Yashdeep Singh
Assistant Professor (Grade-1)
Computer Science and Engineering
Jaypee University of Information Technology, Wagnaghat, India
Dated: 27/11/2017

ACKNOWLEDGEMENT

There were lots of individual who were certainly very helpful, kind, cooperative and generous along the development of this project. I express my appreciation to those concerned. This project and report would have been not possible if it was not for the kind support, tremendous support and continuous supervision of my learned project guide. The deep sense of sincere appreciation that I owe to my learned guide and supervisor “**Mr.Yashdeep Singh**” cannot be expressed in words. His impossible help, constant support, invaluable supervision, kind nature are among the very few consent that he bestowed upon me time to time from the beginning of this project.

Abstract

Internet is one of the basic luxury for daily living. Every person is using the facts and information on internet. On the other hand, blind people face difficulty in accessing the text resources. The development in computer based handy systems has opened up numerous oppurtunities for the visually disabled across.

Audio response based virtual environment, the screen readers are helps blind people a lot to use internet applications.

This project introduces the Voicemail system structural design that can be used by a blind person to access E-Mails easily. The involvement of research is helping blind individual to send and receive voice based mails messages in their inhabitant language with the help of a computer.

Chapter 1: INTRODUCTION

We have seen that the introduction of Internet has revolutionized many fields. Internet has made life of people so easy that people today have access to any information they want easily. Communication is one of the main fields highly changed by Internet.

E-mails are the most dependable way of communication over Internet, for sending and receiving some important information. But there is a certain norm for humans to access the Internet and the norm is you must be able to see.

But there are also differently abled people in our society who are not gifted with what you have. There are some visually impaired people or blind people who can't see things and thus can't see the computer screen or keyboard.

A survey has shown that there are more than 240 million visually impaired people around the globe. That is, around 240 million people are unaware of how to use Internet or E-mail. The only way by which a visually challenged person can send an E-mail is, they have to speak the entire content of the mail to another person(not visually challenged) and then that third person will compose the mail and send on the behalf of the visually challenged person. But this is not a right way to deal with the problem. It is very unlikely that every time a visually impaired person can find someone for help.

1.1: Project Objectives

This project proposes a python based application, designed specifically for visually impaired people. This application provide a voice based mailing service where they could read and send mail on their own, without any guidance through their g-mail accounts. Here, the users have to use certain keywords which will perform certain actions for e.g. Read, Send, Compose Mail etc. The VMAIL system can be used by a blind person to access mails easily and adeptly. Hence dependence of visually challenged on other individual for their activities associated to mail can be condensed.

The application will be a python-based application for visually challenged persons using IVR- Interactive voice response, thus sanctioning everyone to control their mail accounts using their voice only and to be able to read, send, and perform all the other useful tasks. The system will ask the user with voice commands to perform certain action and the user will respond to it. **The main advantage of this system is that use of keyboard is completely eliminated , the user will have to respond through voice only.**

1.2: Interactive Voice Response(IVR)

Interactive voice response (IVR) is a technology that allows a computer to interact with humans through the use of voice and DTMF tones input via a keypad. In telecommunications, IVR allows customers to interact with a company's host system via a telephone keypad or by speech recognition, after which services can be inquired about through the IVR dialogue. IVR systems can respond with pre-recorded or dynamically generated audio to further direct users on how to proceed. IVR systems deployed in the network are sized to handle large call volumes and also used for outbound calling, as IVR systems are more intelligent than many predictive dialer systems.

IVR systems can be used for mobile purchases, banking payments and services, retail orders, utilities, travel information and weather conditions. A common misconception refers to an automated attendant as an IVR. The terms are distinct and mean different things to traditional telecommunications professionals—the purpose of an IVR is to take input, process it, and return a result, whereas that of an automated attendant is to route calls. The term **voice response unit (VRU)** is sometimes used as well. DTMF decoding and speech recognition are used to interpret the caller's response to voice prompts. DTMF tones are entered via the telephone keypad.

Other technologies include using text-to-speech (TTS) to speak complex and dynamic information, such as e-mails, news reports or weather information. IVR technology is also being introduced into automobile systems for hands-free operation. TTS is computer generated synthesized speech that is no longer the robotic voice traditionally associated with computers. Real voices create the speech in fragments that are spliced together (concatenated) and smoothed before being played to the caller.

Another technology which can be used is using text to speech to talk advanced and dynamic data, such as e-mails, reports and news and data about weather. IVR used in automobile systems for easy operations too. Text To Speech is system originated synthesized speech that's not the robotic voice historically related to computer. Original voices produce the speech in portions that are joined together and rounded before played to the caller.

1.3:Speech Recognition

Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT). It incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields. Some speech recognition systems require "training" (also called "enrollment") where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called "speaker independent" systems. Systems that use training are called "speaker dependent".

Speech recognition applications include voice user interfaces such as voice dialing (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), domestic appliance control, search (e.g. find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g. a radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed Direct Voice Input).

The term *voice recognition* or *speaker identification* refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice or it can be used to authenticate or verify the identity of a speaker as part of a security process.

From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data. The advances are evidenced not only by the surge of academic papers published in the field, but more importantly by the worldwide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems

Speech recognition works using algorithms through acoustic and language modeling. Acoustic modeling represents the relationship between linguistic units of speech and audio signals; language modeling matches sounds with word sequences to help distinguish between words that sound similar. Often, hidden Markov models are used as well to recognize temporal patterns in speech to improve accuracy within the system. The most frequent applications of speech recognition within the enterprise include call routing, speech-to-text processing, voice dialing and voice search.

While convenient, speech recognition technology still has a few issues to work through, as it is continuously developed. The pros of speech recognition software are it is easy to use and readily available. Speech recognition software is now frequently installed in computers and mobile devices, allowing for easy access. The downside of speech recognition includes its inability to capture words due to variations of pronunciation, its lack of support for most languages outside of English and its inability to sort through background noise. These factors can lead to inaccuracies.

Speech recognition performance is measured by accuracy and speed. Accuracy is measured with word error rate. WER works at the word level and identifies inaccuracies in transcription, although it cannot identify how the error occurred. Speed is measured with the real-time factor. A variety of factors can affect computer speech recognition performance, including pronunciation, accent, pitch, volume and background noise. It is important to note the terms *speech recognition* and *voice recognition* are sometimes used interchangeably. However, the two terms mean different things. Speech recognition is used to identify words in spoken language. Voice recognition is a biometric technology used to identify a particular individual's voice or for speaker identification.

1.4:Speech Recognition In Python

The improvement and accesibility alone in the field of speech recognition are worth considerable. It allows the physically and the elderly and visually challenged people to collaborate with state of the art products and services quickly and naturally no graphical user interface is needed.

If you want to use speech recognition or simply convert speech to text in your python it is very easy to use. Let's see how:-

- Working of speech recognition.
- Packages available in PyPI.
- How to install and how to use speech recognition package using python library.

A handful of packages for speech recognition exist on PyPI. A few of them include:

- Google-cloud-speech
- Watson-developer-cloud
- Pocketsphinx
- Wit
- Apiai
- SpeechRecognition

SpeechRecognition is a library that acts as a wrapper for many popular speech APIs and is thus very flexible to use. One of these is the Google Web Speech API which supports a default API key that is hard coded into the SpeechRecognition library.

The elasticity and easy to use features of the SpeechRecognition package in python make it a very good choice for developers who are working on any python project. It does not guarantee to support every feature that is wrapped with this API. You will have to dispense some time searching for the easily available options to find out if SpeechRecognition is going work in your particular case.

1.4.1:Required Installations

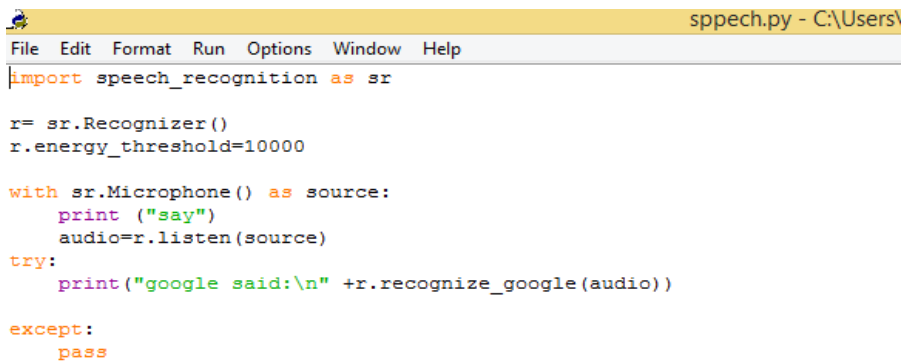
SpeechRecognition is the library which is compatible with Python 2.6, 2.7 and 3.3+, but it will require some additional installation steps for Python v2.0. For our project we have used Python v3.0+.

1.>shell-\$ pip install SpeechRecognition.

2.>shell-\$ pip install python3-pyaudio.

SpeechRecognition will work very good if you need to work with existing audio files. The pyaudio package comes in play when you need to capture microphone input.

The main class which is used in this package is Recognizer class. The use of recognizer instance is obviously to recognize the speech. Every instance of this class comes with various settings and functionality for recognizing speech from the speaker.



```
File Edit Format Run Options Window Help
import speech_recognition as sr

r= sr.Recognizer()
r.energy_threshold=10000

with sr.Microphone() as source:
    print ("say")
    audio=r.listen(source)
try:
    print("google said:\n" +r.recognize_google(audio))
except:
    pass
```

Figure-1

The Microphone class used in this python program will let the user use the default microphone of their system instead of using some audio files as a source.

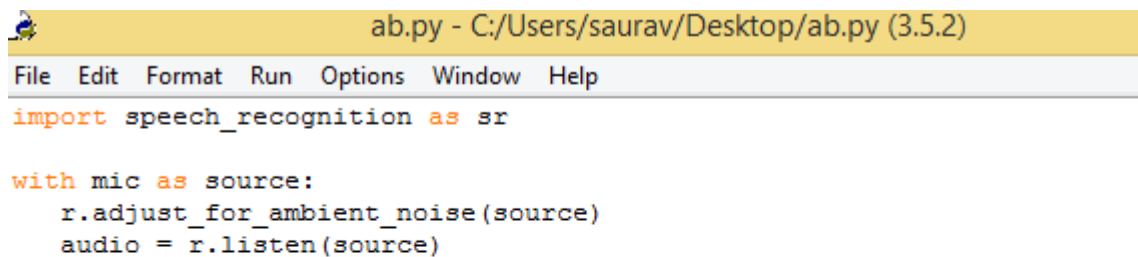
If the system of the user doesn't have the default microphone or in case they want to use some other microphone then they will need to specify which one to use by giving a device index. The list can be seen by calling list_microphone_names() which is static method of Microphone class.

Every instance of Recognizer class has seven methods for recognizing speech from speaker source using various APIs:-

- recognize_bing(): Used in **“Microsoft Bing Speech”**
- recognize_google(): Used in **“Google Web Speech API”**
- recognize_google_cloud():Used in **“Google Cloud Speech”** - requires installation of the google-cloud-speech package
- recognize_houndify(): Used in **“Houndify by SoundHound”**
- recognize_ibm(): Used in **“IBM Speech to Text”**
- recognize_sphinx():Used in CMU Sphinx - requires installing **PocketSphinx**
- recognize_wit(): Used in **“Wit.ai”**

listen():- It is another function used for capturing microphone input. It works just like the AudioFile class while Microhpone is a context manager. Input can be captured from microphone using listen() method of Recognizer class.The first argument taken by this method is an audio source and it will keep on detecting the audio input until the silence is detected by it.

The audio input is generally mixed with ambient noises which can be handled by using the in-built method of recognizer class adjust_for_ambient_noise().

A screenshot of a Python IDE window titled "ab.py - C:/Users/saurav/Desktop/ab.py (3.5.2)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code in the editor is as follows:

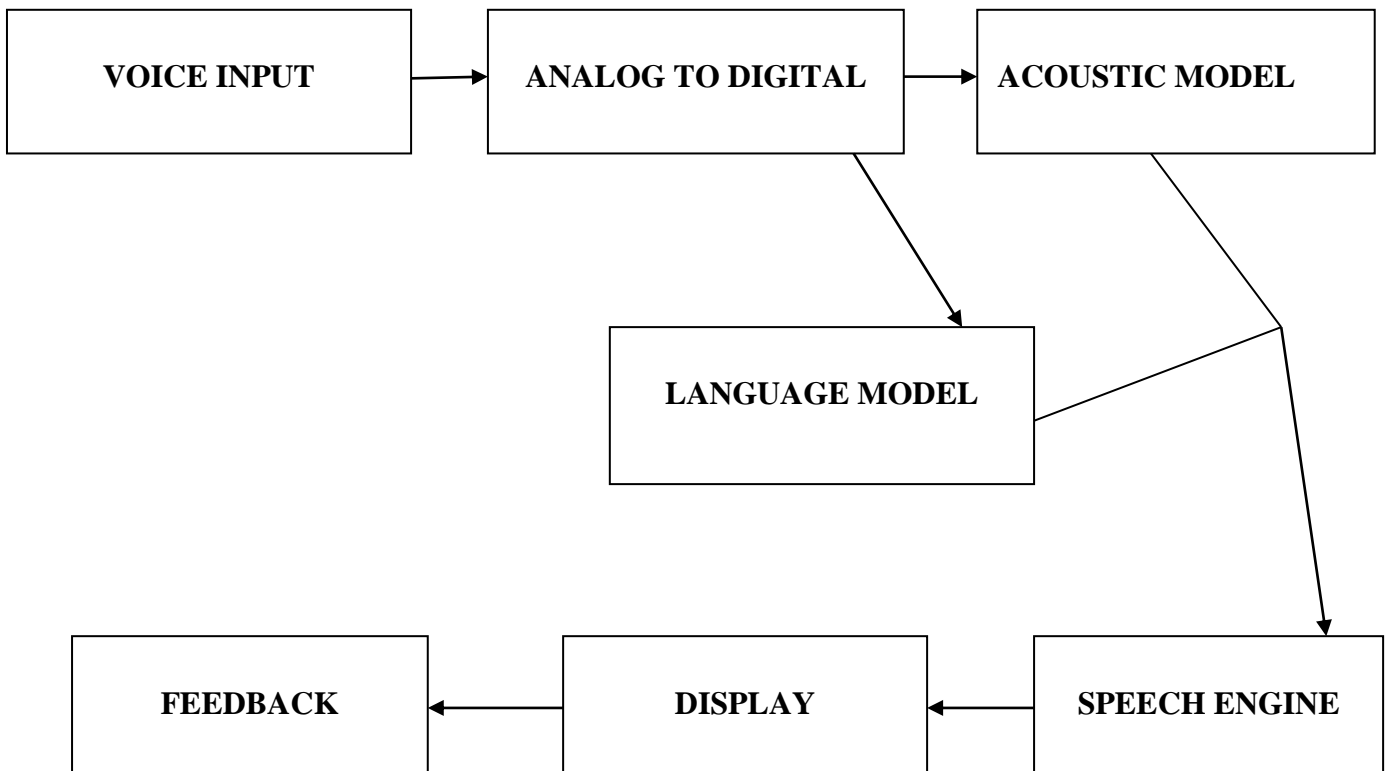
```
import speech_recognition as sr

with mic as source:
    r.adjust_for_ambient_noise(source)
    audio = r.listen(source)
```

Figure-2

You need to wait for a second or two to adjust_for_ambient() to perform its task and then try speaking “Whatever you want” in your microphone and wait for sometime before returning it to recognize the speech again. It only recognizes the speech for one second and it also give you the option to set the duration for wait time.

1.4.2 SYSTEM BLOCK DIAGRAM



1.5: Speech to text Converter

The process of converting spoken speech or audio into text is called speech to text converter. The process is usually called speech recognition. The Speech recognition is used to characterize the broader operation of deriving content from speech which is known as speech understanding. We often associate the process of identifying a person from their voice, that is voice recognition or speaker recognition so it is wrong to use this term for it.

As shown in the above block diagram speech to text converters depends mostly on two models 1.**Acoustic model** and 2.**Language model**. Systems generally use the pronunciation model. It is really imperative to learn that there is nothing like a universal speech recognizer. If you want to get the best quality of transcription, you can specialize the above models for the any given language communication channel.

Likewise another pattern recognition technology, speech recognition can also not be without error. Accuracy of speech transcript deeply relies on the voice of the speaker , the characteristic of speech and the environmental conditions. Speech recognition is a tougher method than what folks unremarkably assume, for a personality's being. Humans are born for understanding speech, not to transcribing it, and solely speech that's well developed will be transcribed unequivocally. From the user's purpose of read, a speech to text system will be categorised based in its use.

1.5.1: Speech Synthesis(TTS)

Speech synthesis is the synthetic production of speech. A automatic data handing out system used for this purpose is called as **speech synthesizer**, and may be enforced in software package and hardware product. A **text-to-speech (TTS)** system converts language text into speech, alternative systems render symbolic linguistic representations.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly. An intelligible text to speech program permits individual with ocular wreckage or reading disabilities to concentrate to written words on a computing device. Several computer operational systems have enclosed speech synthesizers since the first nineteen nineties years.

The text to speech system is consist of 2 parts:-front-end and a back-end. The front-end consist of 2 major tasks. Firstly, it disciple unprocessed text containing symbols like numbers and abstraction into the equivalent of written out words. This method is commonly known as text, standardization, or processing. Front end then assigns spoken transcriptions to every word, and divides and marks the text into speech units, like phrases, clauses, and sentences.

The process of assigning phonetic transcriptions to words is called *text-to-phoneme* or *grapheme-to-phoneme* conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the *synthesizer*—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the *target prosody* (pitch contour, phoneme durations), which is then imposed on the output speech.

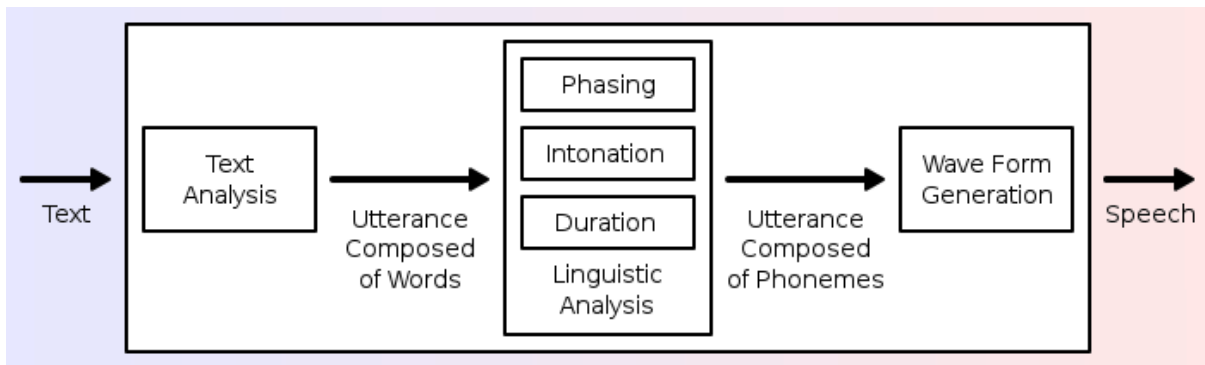


Figure-4

Text-to-speech (TTS) is a type of speech synthesis application that is used to create a spoken sound version of the text in a computer document, such as a help file or a Web page. TTS can enable the reading of computer display information for the visually challenged person, or may simply be used to augment the reading of a text message. Current TTS applications include voice-enabled e-mail and spoken prompts in voice response systems. TTS is often used with voice recognition programs. There are numerous TTS products available, including Read Please 2000, Proverbe Speech Unit, and Next Up Technology's TextAloud. Lucent, Elan, and AT&T each have products called "Text-to-Speech".

In addition to TTS software, a number of vendors offer products involving hardware, including the Quick Link Pen from WizCom Technologies, a pen-shaped device that can scan and read words; the Road Runner from Ostrich Software, a handheld device that reads ASCII text; and DecTalk TTS from Digital Equipment, an external hardware device that substitutes for a sound card and which includes an internal software device that works in conjunction with the PC's own sound card.

1.5.3 TEXT TO SPEECH IN PYTHON

So we have seen above what is a speech to text converter and the theory behind it but the question arises “**how to implement this converter in python**” so here we go.

1.5.3.1 Pyttsx

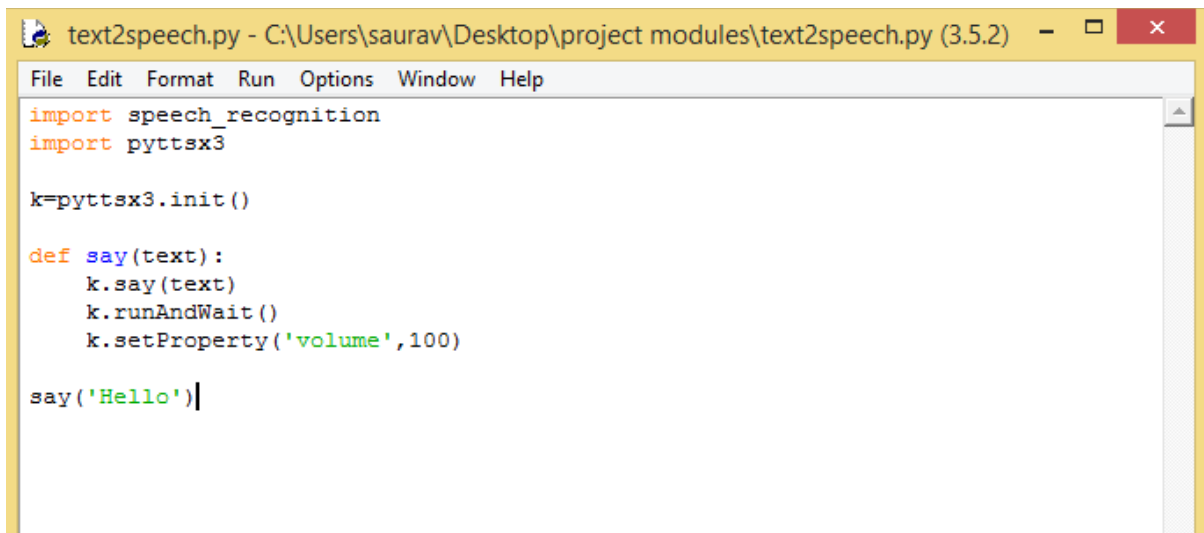
Pyttsx is platform independent that is it is compatible with Windows, Linux, and MacOS speech library. This offers a great set of functionality and features.

The user can set their voice metadata that is information about the data such as gender male or female, pitch of the voice, age, name and language. It supports large set of voices.

So to install it in windows platform depending upon which version of python you are using.

For example if you are using python3 so you need to install pyttsx3.

>>>shell> pip install pyttsx3.



```
text2speech.py - C:\Users\saurav\Desktop\project modules\text2speech.py (3.5.2)
File Edit Format Run Options Window Help
import speech_recognition
import pyttsx3

k=pyttsx3.init()

def say(text):
    k.say(text)
    k.runAndWait()
    k.setProperty('volume',100)

say('Hello')
```

Figure-5

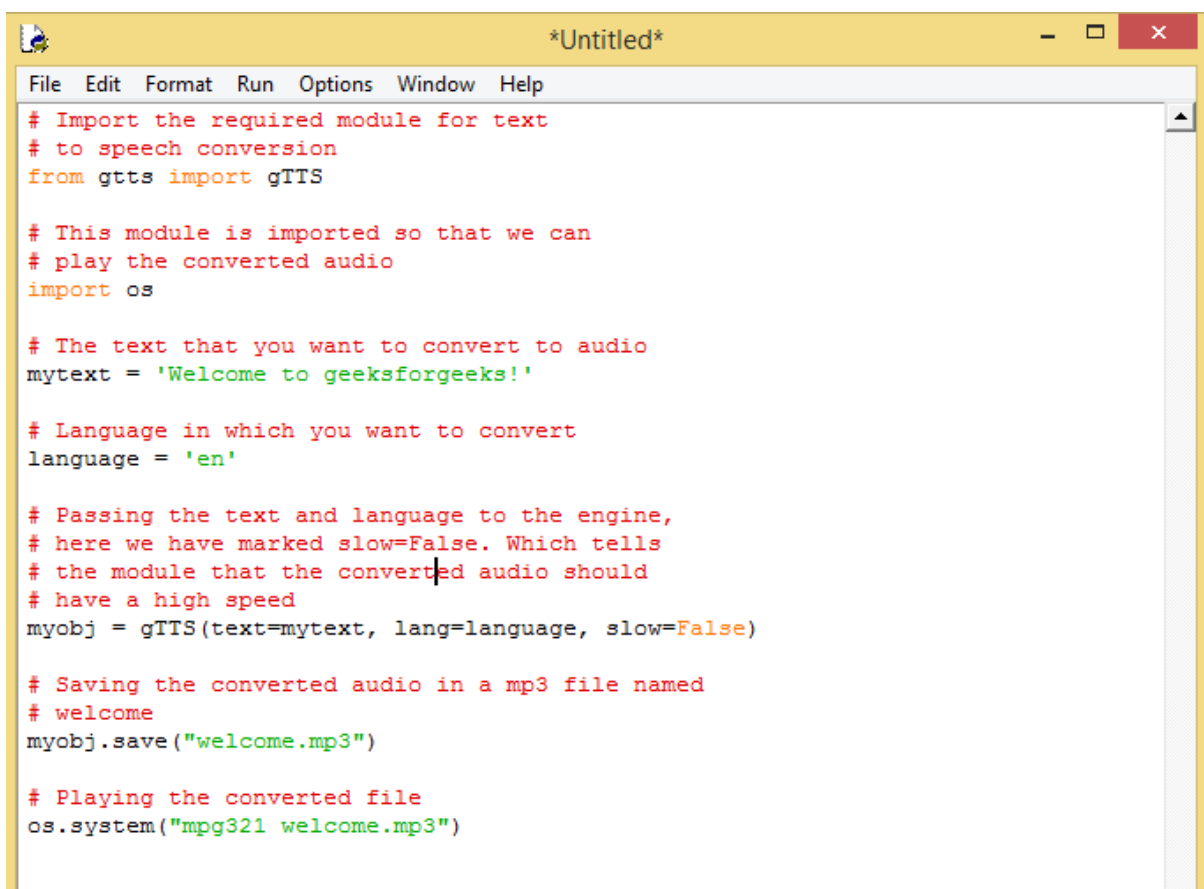
Another module which can be used in python for conversion is:-

1.5.3.2gTTS

This module is Google text to speech library in python.

To install this API in windows platform

```
>>>shell>pip install gTTS
```



```
File Edit Format Run Options Window Help
# Import the required module for text
# to speech conversion
from gtts import gTTS

# This module is imported so that we can
# play the converted audio
import os

# The text that you want to convert to audio
mytext = 'Welcome to geeksforgeeks!'

# Language in which you want to convert
language = 'en'

# Passing the text and language to the engine,
# here we have marked slow=False. Which tells
# the module that the converted audio should
# have a high speed
myobj = gTTS(text=mytext, lang=language, slow=False)

# Saving the converted audio in a mp3 file named
# welcome
myobj.save("welcome.mp3")

# Playing the converted file
os.system("mpg321 welcome.mp3")
```

Figure-6

Chapter 2: Literature Review

2.1 “Voice Based System in Desktop and Mobile Devices for Blind People”. In International Journal of Emerging Technology and Advanced Engineering (IJETAE), 2014

This paper deals with **“Voice Based System in Desktop and Mobile Devices for Blind People”**. Voice mail architecture helps blind people to access e-mail and other multimedia functions of operating system (songs, text). Also in mobile application SMS can be read by system itself. Now a days the advancement made in computer technology opened platforms for visually impaired people across the world. It has been observed that nearly about 60% of total blind population across the world is present in INDIA. In this paper, we describe the voice mail architecture used by blind people to access E-mail and multimedia functions of operating system easily and efficiently. This architecture will also reduce cognitive load taken by blind to remember and type characters using keyboard.

There is bulk of information available on technological advances for visually impaired people. This includes development of text to Braille systems, screen magnifiers and screen readers. Recently, attempts have been made in order to develop tools and technologies to help Blind people to access internet technologies. Among the early attempts, voice input and input for surfing was adopted for the Blind people. In IBM’s Home page the web page is an easy-to-use interface and converts the text-to-speech having different gender voices for reading texts and links. However, the disadvantage of this is that the developer has to design a complex new interface for the complex graphical web pages to be browsed and for the screen reader to recognize.

Simple browsing solution, which divides a web page into two dimensions. This greatly simplifies a web page’s structure and makes it easier to browse. Another web browser generated a tree structure from the HTML document through analyzing links. As it attempted to structure the pages that are linked together to enhance navigability, it did not prove very efficient for surfing. After, it did not handle needs regarding navigability and usability of current page itself.

Another browser developed for the visually handicapped people was eGuideDog which had an integrated TTS engine.

This system applies some advanced text extraction algorithm to represent the page in a user-friendly manner. However, still it did not meet the required standards of commercial use.

Considering Indian scenario, ShrutiDrishti and WebBrowser for Blind are the two web browser framework that are used by Blind people to access the internet including the emails. Both the systems are integrated with Indian language ASR and TTS systems. But the available systems are not portable for small devices like mobile phones.

2.2“Voice Based Search Engine and Web page Reader”. In International Journal of Computational Engineering Research (IJCER)

This paper aims to develop a search engine which supports Man-Machine interaction purely in the form of voice. A novel Voice based Search Engine and Web-page Reader which allows the users to command and control the web browser through their voice, is introduced. The existing Search Engines get request from the user in the form of text and respond by retrieving the relevant documents from the server and displays in the form of text

Even though the existing web browsers are capable of playing audios and videos, the user has to request by typing some text in the search text box and then the user can play the interested audio/video with the help of Graphical User Interfaces (GUI). The proposed Voice based Search Engine aspires to serve the users especially the blind in browsing the Internet. The user can speak with the computer and the computer will respond to the user in the form of voice. The computer will assist the user in reading the documents as well.

Voice-enabled interface with addition support for gesture based input and output approaches are for the “Social Robot Maggie” converting it into an aloud reader . This voice recognition and synthesis can be affected by number of reasons such as the voice pitch, its speed, its volume etc. It is based on the Loquendo ETTS (Emotional Text-To-Speech) software.

Robot also expresses its mood through gesture that is based on gestuary..Speech recognition accuracy can be improved by removal of noise. In A Bayesian scheme is applied in a wavelet domain to separate the speech and noise components in a proposed iterative speech enhancement algorithm. This proposed method is developed in the wavelet domain to exploit the selected features in the time frequency space representation.

It involves two stages: a noise estimate stage and a signal separation stage. In the Principle Component Analysis (PCA) based HMM for the visual modality of audio-visual recordings is used. PCA (Principle Component Analysis) and PDF (Probabilistic Density Analysis).

Presents an approach to speech recognition using fuzzy modelling and decision making that ignores noise instead of its detection and removal. In the speech spectrogram is converted into a fuzzy linguistic description and this description is used instead of precise acoustic features. In Voice recognition technique combined with facial feature interaction to assist virtual artist with upper limb disabilities to create visual cut in a digital medium, preserve the individuality and authenticity of the art work. Techniques to recover phenomena such as Sentence Boundaries, Filler words and Disfluencies referred to as structural Metadata are discussed in and describe the approach that automatically adds information about the location of sentence boundaries and speech disfluencies in order to enrich speech recognition output.

Clarissa a voice enabled procedure browser that is deployed on the international space station (ISS). The main components of the Clarissa system are speech recognition module a classifier for executing the open microphone accepts/reject decision, a semantic analysis and a dialog manager. Mainly focuses on expressions.

To build a prosody model for each expressive state, an end pitch and a delta pitch for each syllable are predicted from a set of features gathered from the text. The expression-tagged units are then pooled with the neutral data, In a TTS system, such paralinguistic events efficiently provide cues as to the state of a transaction, and Markup specifying these events is a convenient way for a developer to achieve these types of events in the audio coming from the TTS engine.

Main features of are smooth and natural sounding speech can be synthesized, the voice characteristics can be changed, it is “trainable. Limitations of the basic system is that synthesized speech is “buzz” since it is based on a vocoding technique, it has been overcome by high quality vocoder and hidden semi-Markov model based acoustic modelling. Speech synthesis consists of three categories: Concatenation Synthesis, Articulation Synthesis, and Formant Synthesis.

Mainly focuses on formant synthesis, array of phoneme of syllable with formants frequency is given as input, frequency of given input is processed, on collaborated with Thai-Tonal-Accent Rules convert given formants frequency format to wave format, so that audio output via soundcard.

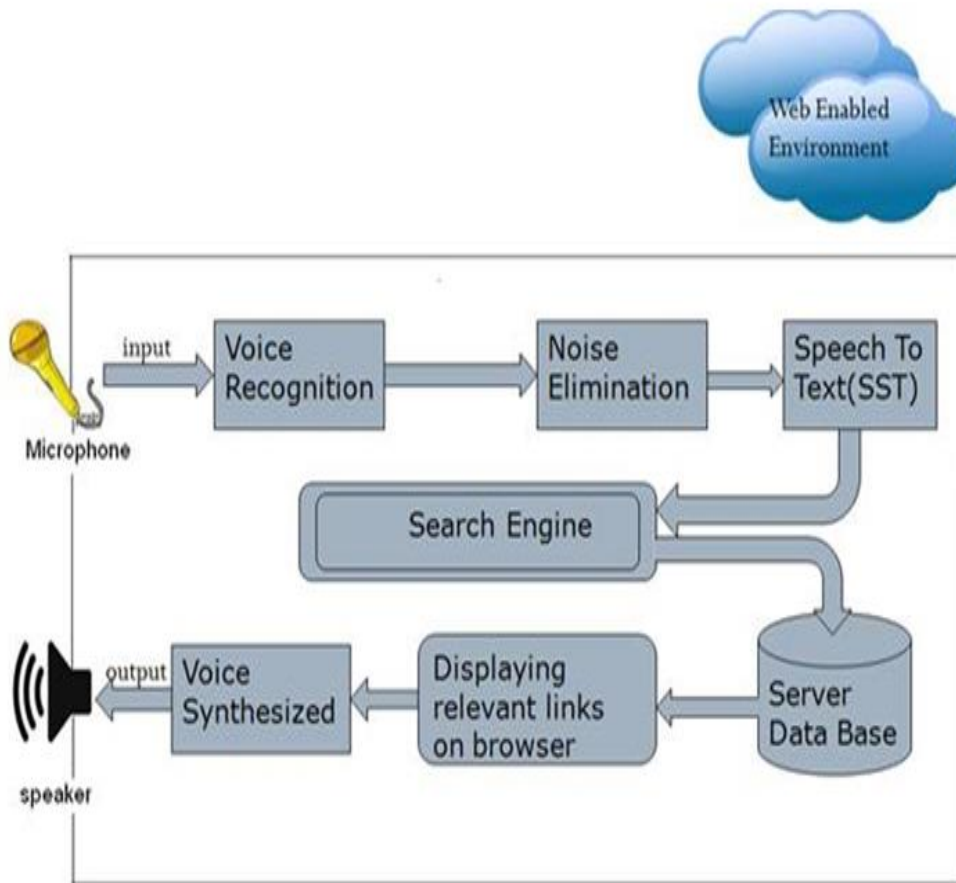


Figure-7

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Proposed model

The planned system is relies on a very fresh plan and obscurity just like the accessible mail systems. The foremost necessary facet that erstwhile unbroken in brain whereas developing the planned system's accessibility.

The present systems don't give this much convenience. So the systems present have a tendency to area unit developing is totally dissent from this system. In contrast to present system which emphasize more on user easiness of naive users, this system focus more on user easiness of all kind of folks including naive folks visually disabled people as well as uneducated people.

The entire structure is based on IVR- interactive voice response. When using this system the computer will prompt the client to perform precise operations to gain relevant services and if the client needs to way in the relevant services then they need to perform that particular operation.

One of the most important recompense of this system is that user will not need to use the keyboard. All operations will be based on voice proceedings.

3.2 Existing System

There are a total number of 4.1 billion email accounts created until 2014 and an there will be estimated 5.2 billion accounts by end of 2018.[4] this makes emails the most used form of communication.

The most common mail services that we use in our day to day life cannot be used by visually challenged people. This is because they do not provide any facility so that the person in front can hear out the content of the screen. As they cannot visualize what is already present on screen they cannot make out where to click in order to perform the required operations.[3] For a visually challenged person using a computer for the first time is not that convenient as it is for a normal user even though it is user friendly.

Although there are many screen readers available then also these people face some minor difficulties. Screen readers read out whatever content is there on the screen and to perform those actions the person will have to use keyboard shortcuts as mouse location cannot be traced by the screen readers. This means two things; one that the user cannot make use of mouse pointer as it is completely inconvenient if the pointer location cannot be traced and second that user should be well versed with the keyboard as to where each and every key is located. A user is new to computer can therefore not use this service as they are not aware of the key locations.

Another drawback that sets in is that screen readers read out the content in sequential manner and therefore user can make out the contents of the screen only if they are in basic HTML format. Thus the new advanced web pages which do not follow this paradigm in order to make the website more user-friendly only create extra hassles for these people. All these are some drawbacks of the current system which we will overcome in the system we are developing.

3.3 Design

A. Phase-1:

The tasks that can be performed using the program developed will be prompted using the voice prompt. In background python module pyttsx3 is used for text to speech conversion.

User will be asked to provide input for the following tasks written below.

The input is expected in the form of speech by the user which will be converted to text by the Google speech application interface in python and accordingly tasks will be performed.

- **Login to their Gmail account.**
- **Send e-mail through Gmail.**
- **Read e-mail through Gmail.**

B. Phase-2:

In phase-2 of our program the user will give speech input to the system.

This speech input will be handled by **speech_recognition** module. It is a python library which is used to handle the voice requests and it converts speech into text.

Now after receiving input from the user speech to text converter will save the response in respective variables used in the script and based on their value it will further enter into respective modules.

C. Phase-3:

In this phase our program will handle the requests by the user. Based on the speech input given by the user it will launch the modules.

- **Login to G-mail account:-** This module will handle the request by user to login in their g-mail account. This module will make the connection with the user's gmail account based on the credentials provided through voice input. This module's script designed as such it will prompt user to enter their g-mail username and password and then it will use selenium web-driver to automate the task for the user and as a result connection will be made.

- **Send E-mail through G-mail:-** This module will handle the request by user to send email through their g-mail account. The python script for this module will prompt the user to enter their credentials and then it will make connection with their account.

After the connection has been done it will further prompt the user to enter the receiver's account e- mail id and it will then allow the user to speak their message and it will repeat it for them and by saying ok it will send the mail.

SMTP library in python is used for the above task.

- **Read E-mail through G-mail:-** This module will handle the request by user to read email through their g-mail account. The python script for this module will prompt the user to enter their credentials and then it will make connection with their account.

After the connection has been done it will start fetching the unread mails for the user and will speak it for them with the help of pyttsx3 library in python for text to speech conversion.

FLOW CHART OF PROPOSED MODEL

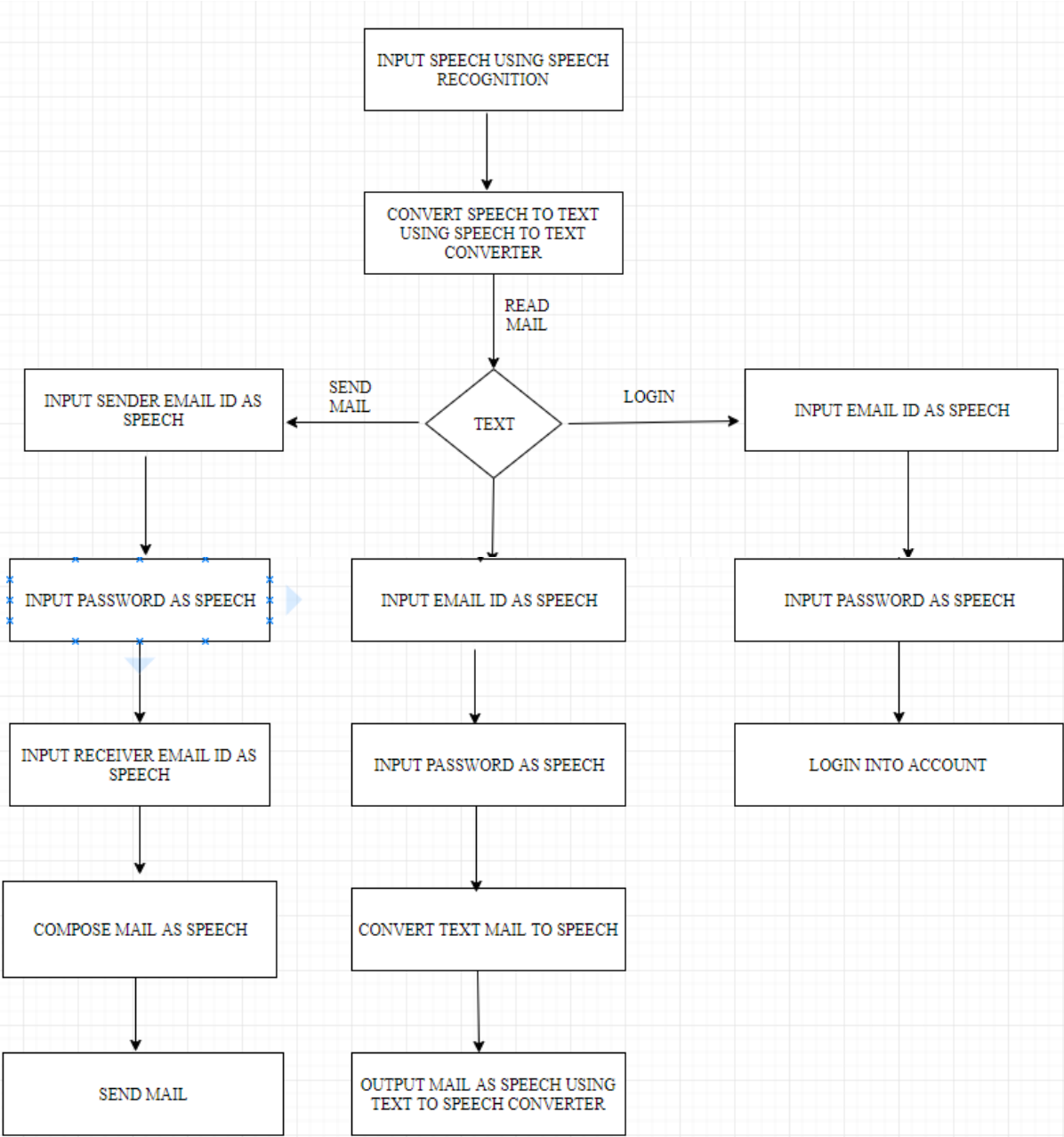


Figure-8

3.3.1 SMTP (Simple Mail Transfer Protocol)

Email is rising because the one among the foremost valuable service in net nowadays. Most of the web systems use SMTP as a technique to transmit mail from one client to different.

SMTP may be a thrust set of rules and is employed to send the mail whereas POP (post workplace protocol) or IMAP (internet message access protocol) square measure accustomed retrieve those mails at the receiver's aspect.

SMTP is Associated with the application layer protocol of OSI model of network.

The user who desires to launch the mail open a TCP (Transmission Control Protocol) connection to the SMTP server and then sends the mail to the other connection. The SMTP server is mostly on listening mode. No sooner the server listens for a TCP connection from any user, the SMTP procedure initiate a connection usually on port number 25.

When the successful establishment of TCP connection has been done, the client can send the mail.

The two processes that is sender process and the receiver process carry out a simple request response dialogue, outlined by the SMTP protocol within which the client process transmits the mail address of the mastermind and the recipient for a message. Once the server method accept these mail addresses, the consumer method broadcast the e-mail instant message. The message should include a message header and message text ("body") formatted in accord with RFC 822.

The following example illustrates a message in the RFC 822 message format:

| |
|--|
| <p>From: ashwanikumar@example.com To: sauravmishra@example.com Subject: An RFC 822 formatted message</p> <p>This is a simple text body of the message. The blank line separates the header and body of the message.</p> |
|--|

The **SMTP** model is of two types :-

- 1.End-to- end method**
- 2.Store-and- forward method**

The SMTP model chains both end-to-end no intermediate message transfer agents and store-and-forward mail delivery methods. The end-to-end method of SMTP is used between organization, and the store-and forward method is chosen for sending mails within organizations which have TCP/IP and SMTP-based networks.

End-To-End

In this method , a SMTP client will speak to the destination host's SMTP server directly to transport the mail. It will keep the mail item from being transmitted until it has been successfully copied to the recipient's SMTP.

Store-and-Forward

In this method a mail can be sent through a number of intermediary hosts, before reaching to the final destination.

A successful transmission from a hosts signify only that the mails has been sent to the next host, and then the mail will be sent to next host.

3.3.2 SENDING EMAIL IN PYTHON USING SMTPLIB

Automation of sending mails using Python can be done by using the **smtplib** module of Python. **Smtplib** contains the class SMTP which is useful to connect with mail servers and can be used to send mails. It defines a SMTP client session object which is used to send mail to any internet connected machine that depends on SMTP format.

SMTP is normally used to connect to a mail server and transmit the messages.

The mail server host name and port can be passed to the constructor, or you can use **connect()** explicitly.

Once connected, just call **sendmail()** with the envelope arguments and body of the message.

The message text should be a completely created RFC 822-compliant message, since smtplib does not alter the contents of headers.

We have to add header and sender mail and receiver mail by ourselves.

```
S1 = smtplib.SMTP( h , p , l)
```

Where h=host name, p=port number,l=localhost name

Host – The argument is used to represent the host which provides you **SMTP** server. We can specify IP address of the host or a domain name like gmail.com or outlook.com. It is not a compulsory argument.

Port - If the host name is provided then we have to give a port number where SMTP server will listen the requests, normally this port number is 25.

Local hostname - If your SMTP server is running on your local machine, then you can give just localhost in this argument.

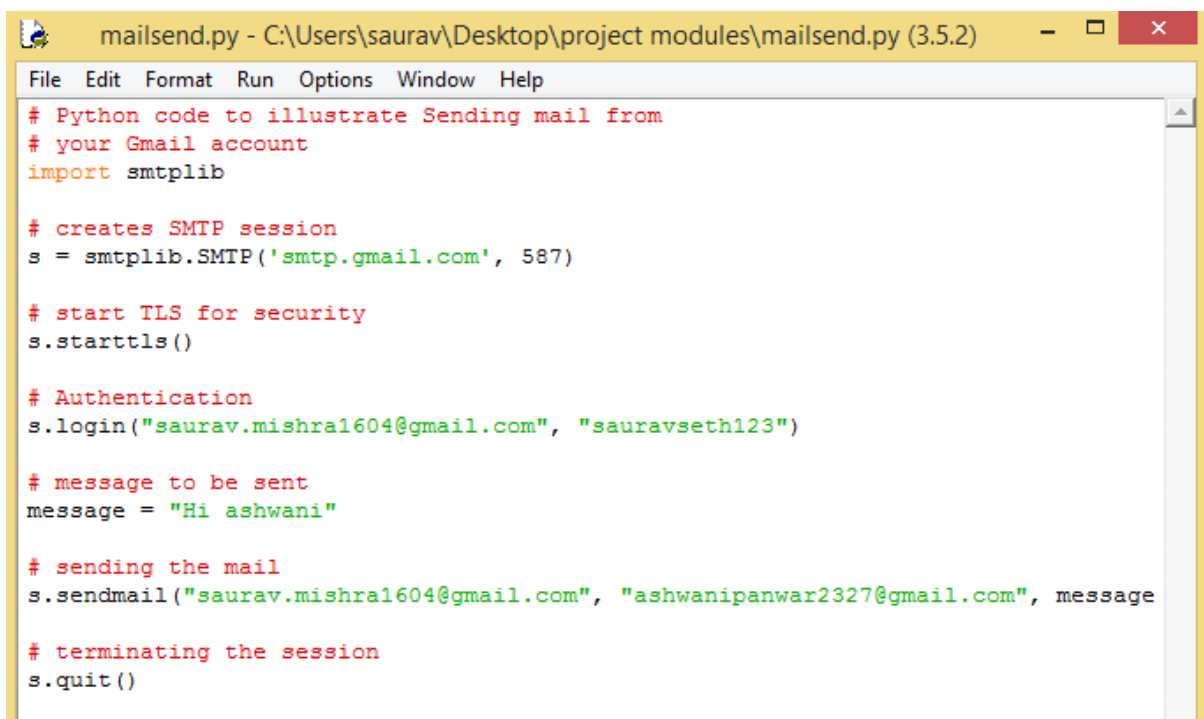
An SMTP object has a method called **sendmail**, which is usually used to send the mails. It takes following parameters -

```
s.sendmail("saurav.mishra1604@gmail.com", "ashwanipanwar2327@gmail.com", message)
```

The sender - Email-Id of sender.

The receivers - Email-Ids of receivers.

The message - A message arranged like RFC822



```
mailsend.py - C:\Users\saurav\Desktop\project modules\mailsend.py (3.5.2)
File Edit Format Run Options Window Help
# Python code to illustrate Sending mail from
# your Gmail account
import smtplib

# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)

# start TLS for security
s.starttls()

# Authentication
s.login("saurav.mishra1604@gmail.com", "sauravseth123")

# message to be sent
message = "Hi ashwani"

# sending the mail
s.sendmail("saurav.mishra1604@gmail.com", "ashwanipanwar2327@gmail.com", message)

# terminating the session
s.quit()
```

Figure-9

3.3.3 READING EMAIL FROM GMAIL USING PYTHON

This is very important part of our system but python as usual provides very high flexibility in providing this feature. Yes we can automate the process of reading the email from our **Gmail** account and this can be very useful for the people who can't see so they can use this system to read the email or we can say fetch the unread email from their **Gmail** account and can listen to it with the help of text to speech converter.

So to achieve our task we just need three modules or functionalities.

1. **A mail server and username and its password.**
2. **Login to Gmail account.(Already discussed how to login using python script) .**
3. **Servers such as imap.gmail.com and smtp.gmail.com.**
4. **Most important server needed would be imap.gmail.com.**

Imaplib-IMAP4 PROTOCOL

This is the main module which will be used in the process of reading email from your Gmail account using a python script.

Basically this module consists of three classes **1.IMAP4 2.IMAP4_SSL and 3.IMAP4_stream.**

These classes are contained in imaplib module whereas IMAP4 is the base class.

All IMAP4rev1 commands are represented by methods of the same name, either upper-case or lower-case. All arguments to commands are converted to strings, except for AUTHENTICATE, and the last argument to APPEND which is passed as an IMAP4 literal. If necessary (the string contains IMAP4 protocol-sensitive characters and isn't enclosed with either parentheses or double quotes) each string is quoted. However, the *password* argument to the LOGIN command is always quoted. If you want to avoid having an argument string quoted (eg: the *flags* argument to STORE) then enclose the string in parentheses)command. Each *data* is either a string, or a tuple. If a tuple, then the first part is the header of the response, and the second part contains the data.

IMAP4.check():-Checkpoint mailbox on server.

IMAP4.close():-Close currently selected mailbox. Deleted messages are removed from writable mailbox. This is the recommended command before LOGOUT.

IMAP4.copy(message_set, new_mailbox):-Copy message_set messages onto end of new_mailbox.

IMAP4.create(mailbox):-Create new mailbox named mailbox.

IMAP4.delete(mailbox):-Delete old mailbox named mailbox.

IMAP4.deleteacl(mailbox, who):-Delete the ACLs (remove any rights) set for who on mailbox.

IMAP4.enable(capability):-Enable capability Most capabilities do not need to be enabled. Currently only the UTF8=ACCEPT capability is supported.

IMAP4.expunge():-Permanently remove deleted items from selected mailbox. Generates an EXPUNGE response for each deleted message. Returned data contains a list of EXPUNGE message numbers in order received.

IMAP4.fetch(message_set, message_parts):-Fetch (parts of) messages. message_parts should be a string of message part names enclosed within parentheses,Returned data are tuples of message part envelope and data.

IMAP4.getacl(mailbox):-Get the ACLs for mailbox. The method is non-standard, but is supported by the Cyrus server.

IMAP4.getannotation(mailbox, entry, attribute):-Retrieve the specified ANNOTATIONS for mailbox. The method is non-standard, but is supported by the Cyrusserver.

IMAP4.getquota(root):-Get the quota root's resource usage and limits. This method is part of the IMAP4 QUOTA extension defined in rfc2087.

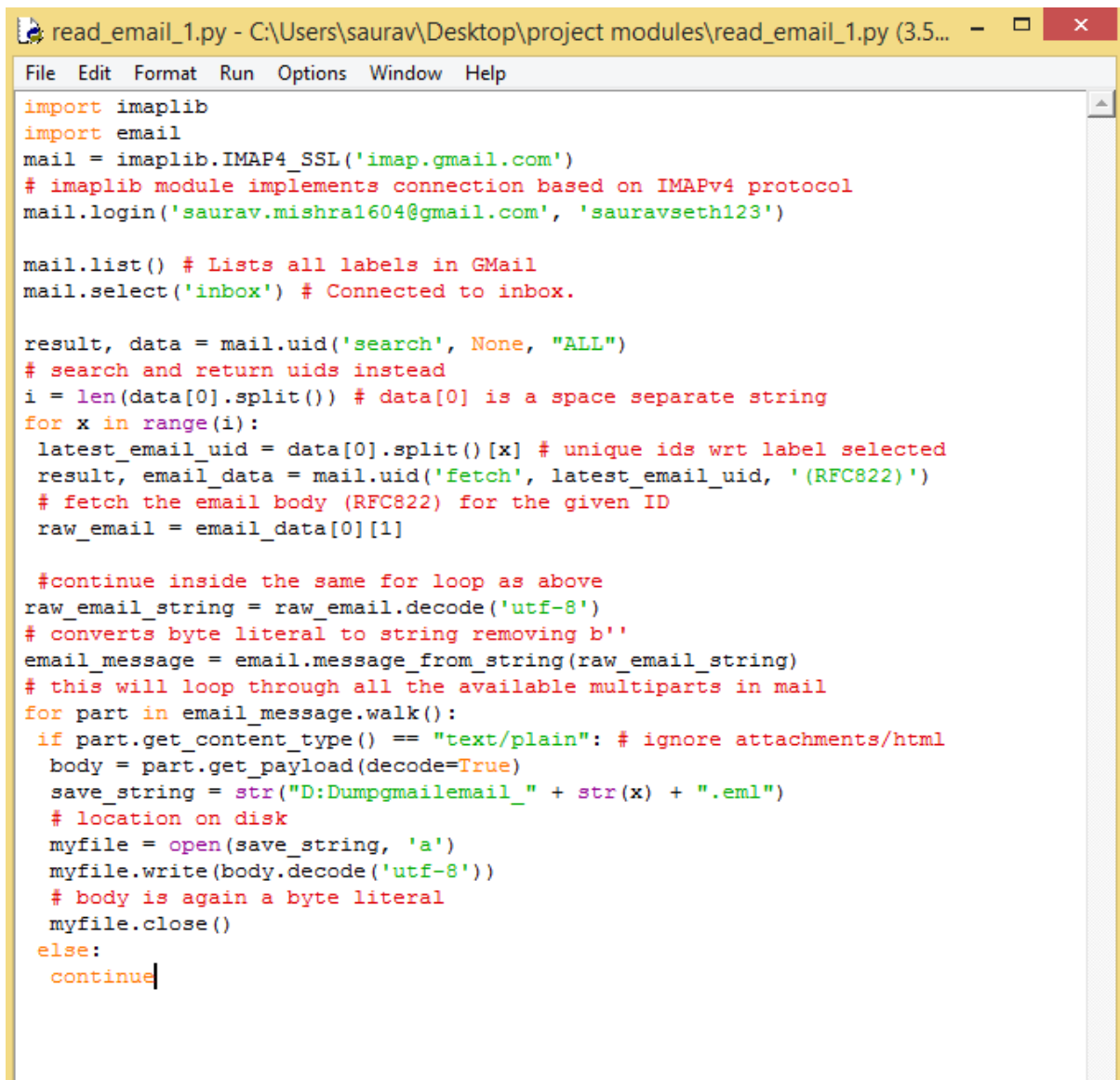
IMAP4.getquotaroot(mailbox):-Get the list of quota roots for the named mailbox. This method is part of the IMAP4 QUOTA extension defined in rfc2087.

IMAP4.login(user, password):-Identify the client using password. The password will be quoted.

IMAP4.login_cram_md5(user, password):-Force use of CRAM-MD5 authentication when identifying the client to protect the password. Will only work if the server CAPABILITY response includes the phrase AUTH=CRAM-MD5.

IMAP4.logout():-Shutdown connection to server. Returns server BYE response.

This code illustrates how to read email from **Gmail** using python.



```
read_email_1.py - C:\Users\saurav\Desktop\project modules\read_email_1.py (3.5... - □ ×
File Edit Format Run Options Window Help
import imaplib
import email
mail = imaplib.IMAP4_SSL('imap.gmail.com')
# imaplib module implements connection based on IMAPv4 protocol
mail.login('saurav.mishra1604@gmail.com', 'sauravseth123')

mail.list() # Lists all labels in Gmail
mail.select('inbox') # Connected to inbox.

result, data = mail.uid('search', None, "ALL")
# search and return uids instead
i = len(data[0].split()) # data[0] is a space separate string
for x in range(i):
    latest_email_uid = data[0].split()[x] # unique ids wrt label selected
    result, email_data = mail.uid('fetch', latest_email_uid, '(RFC822)')
    # fetch the email body (RFC822) for the given ID
    raw_email = email_data[0][1]

    #continue inside the same for loop as above
    raw_email_string = raw_email.decode('utf-8')
    # converts byte literal to string removing b'
    email_message = email.message_from_string(raw_email_string)
    # this will loop through all the available multiparts in mail
    for part in email_message.walk():
        if part.get_content_type() == "text/plain": # ignore attachments/html
            body = part.get_payload(decode=True)
            save_string = str("D:Dumpgmailemail_" + str(x) + ".eml")
            # location on disk
            myfile = open(save_string, 'a')
            myfile.write(body.decode('utf-8'))
            # body is again a byte literal
            myfile.close()
        else:
            continue
```

Figure-10

CHAPTER 4: PERFORMANCE ANALYSIS

4.1 Software Requirements:

Tools Used:

- Python IDLE.
- Interpreters for scripts.
- Selenium Web driver in python.
- Google Speech-to-text and text-to-speech Converters.
- Pyttsx text to speech api in python.

4.2 Hardware Requirements:

- Windows Desktop

CHAPTER 5: CONCLUSION

5.1 Conclusions

The project that we have projected is a system which will help the visually impaired people to access email services efficiently.

This system will help in overcoming some drawbacks that were earlier faced by the blind people in accessing emails. We have eliminated the concept of using keyboard shortcuts along with screen readers which will help reducing the cognitive load of remembering keyboard shortcuts.

Also any non-sophisticated user who does not know the position of keys on the keyboard need not bother as keyboard usage is eliminated. Instructions given by the IVR accordingly to get the respective services offered.

Other than this the user might need to feed in information through voice inputs when specified.

It is a observation that about 70% of total blind population across the world is present in INDIA. This project, describe the voice mail architecture used by blind people to access E-mail and multimedia functions of operating system easily and efficiently.

This architecture will also reduce cognitive load taken by blind to remember and type characters using keyboard.

It also helps handicapped and illiterate people.

5.1 Application

This project is proposed for the betterment of society. This project aims to help the visually impaired people to be a part of growing digital India by using internet and also aims to make life of such people quite easy.

Also, the success of this project will also encourage developers to build something more useful for visually impaired or illiterate people, who also deserves an equal standard in society.

5.2 Future Scope

Voice could be extended to image attachments and other options such as indentation, fonts etc., that are available with normal E-Mail.

5.2 References

- Jagtap Nilesh, Pawan Alai, Chavhan Swapnil and Bendre M.R.. “Voice Based System in Desktop and Mobile Devices for Blind People”. In International Journal of Emerging Technology and Advanced Engineering (IJETAE), 2014 on Pages 404-407 (Volume 4, issue 2).
- Ummuhanysifa U.,Nizar Banu P K , “Voice Based Search Engine and Web page Reader”. In International Journal of Computational Engineering Research (IJCER). Pages 1-5.
- G. Shoba, G. Anusha, V. Jeevitha, R. Shanmathi. “AN Interactive Email for Visually Impaired”. In International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 2014 on Pages 5089-5092.
- The Radicati website. [Online]. Available: <http://www.radicati.com/wp/wp-content/uploads/2014/01/EmailStatistics-Report-2014-2018-Executive-Summary.pdf>.