PROJECT REPORT
On
VOICE BASED EMAIL FOR VISUALLY CHALLENGED

By

AKSHAY ASHOK (RA1931241020009)

SUNIL S (RA1931241020016)

TAMIL SELVAN T (RA1931241020037)

Submitted to the

DEPARTMENT OF COMPUTER APPLICATIONS (BCA)

Under the guidance of

Mrs. Dr. T.S.SUGANYA, MCA., Ph.D.,

Assistant Professor

Department of Computer Applications

Submitted in partial fulfilment of the requirement

For the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Ramapuram, Chennai

May 2022

COLLEGE OF SCIENCE & HUMANITIES

Ramapuram, Chennai

BONAFIDE CERTIFICATE

Certified that this project report titled **"VOICE BASED EMAIL FOR VISUALLY CHALLENGED"** is the bonafide work of **AKSHAY ASHOK (RA1931241020009), SUNIL S (RA1931241020016) and TAMIL SELVAN T (RA1931241020037)** who carried out the Main project work done under my supervision.

Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Signature of Guide **Signature of Head of the Department**

Signature of External Examiner

ABSTRACT

The Internet has become one of the most common things that people use on a day-to-day basis. Also with the growth of the internet, communication has also started to play a vital role for sending and receiving important information. Though there have been many advancements in internet communication, No normal or naive visually challenged person will be confident enough to use the generic email system on their own. This is mainly due to the fact that using them visual perception(ability to see).

This project aims at developing an Email System that will help even the visually challenged to use the Email system without any previous training or additional assistance.The system will not let the user make use of any physical inputs and instead will solely use Interactive Voice Response along with Text-to-Speech and Speech-to-Text Conversion. This System will also be helpful for not just visually Challenge but other differently abled people and people who cannot read.

Keyword: Visually Challenged, Interactive Voice Response, Text-to-Speech, Speech-to-Text.

ACKNOWLEDGEMENT

I extend my sincere gratitude to the Chancellor Dr. T.R.PACHAMUTHU and to Chairman Dr. R. SHIVAKUMAR of SRM Institute of Science and Technology, Ramapuram and Trichy campuses for providing me the opportunity to pursue the BCA degree at this University.

I express my sincere gratitude to Dr.C.SUNDAR, Principal, College of Science Humanities, SRM IST, Ramapuram for his support and encouragement for the successful completion of the project.

I am thankful to Dr. J. DHILLIPAN, Vice Principal-Admin, College of Science Humanities, SRM IST, Ramapuram, for his support and encouragement for the successful completion of the project.

I am grateful to Dr. N. ASOKAN, Vice Principal- Academic, College of Science Humanities, SRM IST, Ramapuram, for his provision and support for the successful completion of the project.

I record my sincere thanks to Dr. AGUSTHIYAR.R, Professor and Head, Department of Computer Applications (BCA), SRM IST, Ramapuram for his continuous support and keen interest to make this project a successful one.

I find no word to express profound gratitude to my guide Mrs.DIVYA MCA.,ME.,(Ph.D), ASSISTANT PROFESSOR (OG), Department of Computer Applications (BCA), SRM IST Ramapuram.

I thank the almighty who has made this possible. Finally I thank my beloved family members and friends for their motivation, encouragement and cooperation in all aspects which led me to the completion of this project.

Name of the Students
AKSHAY ASHOK (RA1931241020009)
SUNIL S (RA1931241020016)
TAMIL SELVAN T (RA1931241020037)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1 PROJECT INTRODUCTION

Today the world is running on the basis of the Internet. The Internet plays a vital role in the world of Communication. Without the Internet no work can be easily done Today. e-mail is one of the main fields with the use of the Internet. Among them Electronic mail (E-mail) is one of the most dependable and reliable means of communication. But there are people who are visually challenged who can’t see things and can't use the generic email system.

Among 7.9 Billion people as of now (2022) there are around 2.2 Million people are Visually Challenged. The only way which will help them to use the email system is by Speech Recognition and screen readers, So we came up with our Project called **Voice Based Email System for Visually Challenged.** In this system the usage of keyboard and mouse is completely avoided. All the Interactions between the system and the user is done through IVR (Interactive Voice Response). Our architecture will help them to operate the system easily and efficiently. Our System can also be used by the Handicapped People.

Additionally, It is estimated that there are approximately 1.8 million individuals in the United States who are blind with no residual vision, 980,000 individuals in the United Kingdom with significant sight loss, 284 million individuals worldwide who are visually impaired, and 39 million worldwide who are fully blind. The World Health Organisation identifies four levels of visual function: normal vision, moderate visual impairment, severe visual impairment, and blindness.

When one considers the unemployment statistics of between 70-75% for working-age blind individuals in the United States and 75% for blind and visually impaired individuals in the United Kingdom, the usability of email becomes a major concern due to its intersection with many vocational responsibilities, such as collaboration with co-workers.

Studies have shown that email frustrations waste the time of all users. For blind users, the increased costs of frustration due to accessibility and usability problems may present a barrier to effective participation in workplace communication.

1.2 LITERATURE SURVEY

1.2.1 Usability Issues for Blind Users

Assistive technology tools such as screen readers are necessary for blind individuals to use most software. A screen reader (such as JAWS, System Access, VoiceOver, or Window-Eyes) is software that will audibly read the visual content that is being displayed on a computer screen to a blind user. HearSay, a non-visual web browser developed by Stony Brook University, and WebAnywhere, a portable screen reader developed by the University of Washington, are examples of alternative approaches to screen readers. Another method that blind people utilise to access software is Braille and Braille-supported devices. The challenge with Braille devices is that they are often cost-prohibitive, and the rate of Braille literacy among blind users is very low (an estimated 10-20% in the United States).

It is known that blind users are more likely to avoid content when they are aware in advance that it will cause them accessibility problems (such as those presented by dynamic web content), and blind users are also often forced to discover some sort of workaround to complete a particular task. Computer frustrations that impact one's ability to complete a work task can affect the mood of blind users. Examples of the challenges that are faced by blind users are well illustrated in the Lazar, et al. study on the frustrations that screen reader users experience on the Web. The study identified poorly labelled links and forms, missing or confusing alternate text for graphics, and problems with PDF files as being some of the challenges commonly faced by blind users.

On the legal front, in the United States, Section 508 of the Rehabilitation Act has served as the guide for government websites and technology, yet the laws that apply to private companies, such as the US Americans with Disabilities Act, do not provide specific technical guidelines. Section 508 is currently being revised, and the Justice Department in the United States is planning to expand the technical guidelines of the Americans with Disabilities Act to more specifically address websites. In the United Kingdom, the Disability Discrimination Act specifies accessibility standards for web sites, but the focus for this has also been government web sites. However, this was updated by the Equality Act in October 2010, which strengthens the impact of the Disability Discrimination Act, including restrictions on pre-employment, disability-related inquiries.

1.2.2 Potential Email Concerns for Blind Users

Accessibility refers to users with impairments being able to technically access technology, while usability is a broader topic, relating to true ease of use. This study is focused on usability, in order to fully understand how users who are blind use email applications, and to identify which features or applications they are not able to fully use.

Unsolicited email (spam) is a particular concern for blind users. While sighted users can visually scan and skip over offensive or non-relevant email in their inbox, blind users must listen to the email in their inbox one email at a time. Spam can also present a security threat since it is one of the most common carriers of electronic viruses and worms [26]. The obvious primary solution to managing spam is through aggressive spam filtering software. The major tradeoff with a spam filter is that by its very nature (filtering email) the likelihood of false positive and negative identification of spam email is always possible. It is perceived that blind users tend to use high levels of spam filtering, which may filter out legitimate incoming emails that are sent using the BCC feature (blind carbon copy).

Studies have shown that an email inbox full of messages is something that most users struggle with. Some of the most common methods for managing email revolve around archiving and storing messages in folders as well as the common practice of “inbox message visibility,” which involves

visually scanning the inbox for messages. It is important to determine how blind users handle email organisation as well as other extended features such as calendars and contacts in order to develop suggestions for improvement in design.

A focus group held in May 2008 at the National Federation of the Blind in Baltimore, Maryland, USA, identified some possible usage barriers of email that may impact blind users:

- Spam, which was described as both frustrating and at times embarrassing, due to the potentially objectionable content of some spam messages
- Methods of searching for and organising email
- Cluttered and hard-to-navigate web-based email interfaces
- The use of extended features (such as contacts and calendaring)
- Visual CAPTCHAs (distorted text used to verify that a user is human and not an automated security threat)

The results of this focus group prompted the creation of an adaptive, web-based survey to further explore email usability for blind users.

1.3 OVERVIEW - VOICE BASED EMAIL

In this proposed system, the user can send and receive email and also listen to either of these processes with the help of IVR (Interactive Voice Response) and Voice commands. In this Email System, the application makes use of the SMTP protocol for sending emails and POP3 protocol for receiving emails. SMTP (Simple Mail Transfer Protocol) is the reliable protocol to send emails and it works in a very simple way that the SMTP server passes on the email messages quickly. POP3 (Post Office Protocol) is used to receive emails. The POP3 server stores the email and on request the emails are displayed. The same is implemented in our application, that on the request by the user the emails are downloaded.

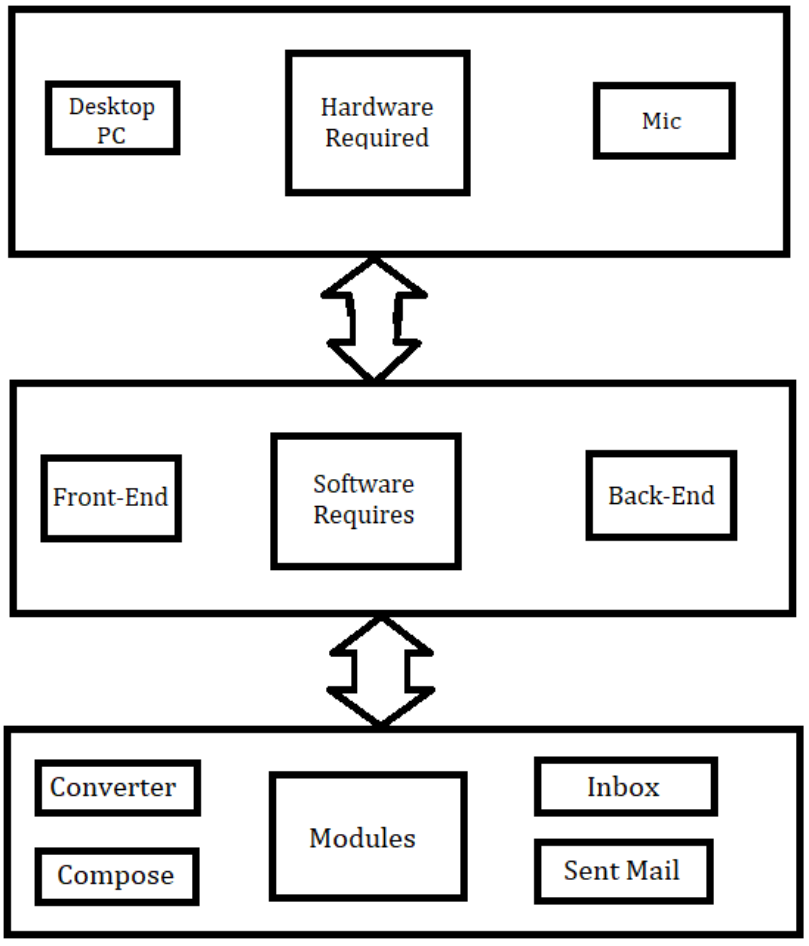


Figure 1.1 - Overview of Proposed System

CHAPTER 2

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

On studying many researches and going through many papers on the above existing technology the visually challenged for using the Generic Email System, In the paper “VOICE BASED EMAIL SYSTEM FOR VISUALLY IMPAIRED” by Saurav Mishra, Ashwani Panwar deals with “Voice Based System in Desktop and Mobile Devices for Blind People''. Voicemail architecture helps blind people to access e-mail and other multimedia functions of operating systems (songs, text). Also in mobile applications SMS can be read by the system itself. Nowadays the advancement made in computer technology opened platforms for visually impaired people across the world. It has been observed that nearly 60% of the total blind population across the world is present in INDIA. In this paper, we describe the voicemail architecture used by blind people to access E-mail and multimedia functions of operating systems easily and efficiently. This architecture will also reduce cognitive load taken by blind to remember and type characters using a keyboard.

There is a 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 analysing 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 the 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, it still did not meet the required standards of commercial use. Considering the Indian scenario, ShrutiDrishti and WebBrowser for Blind are the two web browser frameworks 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 DISADVANTAGES OF THE EXISTING SYSTEM

- There is no voice command or sound system,
- Still the visually impaired person has to use the keyboard,
- It needs a lot of practice for them to use,
- Third person is sending mails behalf of them,
- There is no STT (speech to text), TTS (text to speech), or IVR (Interactive voice Response).

2.3 PROPOSED SYSTEM

We now know and have established that even with the latest advancements in the field of communications, even now visually challenged people find it difficult to utilise these technologies because using them requires you to be able to see and figure out things visually. However we easily tend to forget that there are still people who are not as fortunate as most of us and are visually impaired. They can not easily figure out what's on the screen. This makes the whole concept of the internet a completely inert and useless technology for visually challenged and even illiterate people in some cases.

In our proposed system, we mainly use three types of technology, they are:

STT(Speech-to-text): Here whatever we speak is converted to text. Based on a certain user command, his/her speech will be converted to text format, which the people could see and read also.

TTS(text-to-speech): This method is the complete opposite of Speech-to-text. Here, the text format of the emails to synthesized speech.

IVR(Interactive voice response): IVR is an advanced technology that describes the interaction between the user and the system in the way of responding by using physical input for the respective voice messages. IVR allows users to interact with an email host system via a system commands, after that the users can easily service their own queries by listening to the IVR prompt. IVR systems generally respond with pre-recorded audio messages to further assist users on how to proceed.

2.4 BENEFITS OF PROPOSED SYSTEM

- The disabilities of visually impaired people are Thrashed.
- This system makes the disabled people feel like a normal user.
- They can hear the recently received mails to the inbox,as well as the IVR technology proves very effective for them in terms of guidance.
- The visually impaired people do not “necessarily” require any assistance technically.
- This presumably has far more implications than just email systems in the future.

The main benefit of this system is that the use of the keyboard is completely eliminated, the user will have to respond through voice commands only.

Therefore, the user does not need to worry about the location of the mouse pointer nor the physical keyboard keys at all. This proposed system will be easily accessible to nearly all types of users as it is just based on interactive voice recognition and speech conversion, there is absolutely no need to locate the mouse pointer on the screen. Also because of this, the people who are unable to read and write need not worry as they can hear the prompting done by the system and perform respective actions. Also, the idea focuses on providing basic functionalities like compose, send, receive, check inbox and along with advanced features like Voice based operation, Search Mail, provision for voice as well as text. Every basic functionality will be provided.

CHAPTER 3

3. WORKING ENVIRONMENT

3.1 SCOPE OF THE PROJECT

For the visually impaired people, it is a big deal to access the common email systems. Therefore, the proposed system will be of great application for the visually impaired people as the system is designed with text to speech and speech to text that enables them to access their account, compose and send mails without having to be restricted to a keyboard or any physical input.

3.2 HARDWARE REQUIREMENTS

- SYSTEM WITH RAM - 1 GB (MIN.) RAM
- HARD DRIVE - 5 GB HARD DISK SPACE (MIN.)
- PROCESSOR - 1.2 GHz PROCESSOR

3.3 SOFTWARE REQUIREMENTS

- OS - WINDOWS
- TOOLKIT - PYTHON/PYCHARM

3.4 METHODOLOGIES

3.4.1 USER REGISTRATION:

Here, we get the details such as name, contact details, dob and security questions from the user.

3.4.2 SETTING UP CREDENTIALS:

Like in the process of User Registration, here we get the USER ID and PASSWORD.

3.4.3 SERVICE SELECTION:

We select the process and service that the user wants to perform.

3.4.4 OUTPUT PROCESS:

Based on the task or process chosen, the respective functions are performed and the required result of the process is received.

3.4.5 LOGOUT OR SIGNOUT:

Finally, after the user has completed his/her desired task, the user is prompted or suggested to Logout/Sign-out.

The entire system is dependent on voice prompts and mouse clicks. When utilising this system, the computer will prompt the user to execute specified activities in order to access particular services, and the user must complete those activities in order to access those services. One of the most significant advantages of this system is that the user will rarely need to use a keyboard.All actions will be triggered by mouse clicks. The challenge now is how blind users will determine where the mouse pointer is located. Because the blind user cannot track a specific place, he or she must move the mouse across the screen from top to bottom and then left to right. Because it is only a basic system, it will be fully accessible to all types of users.

CHAPTER 4

4. SYSTEM TESTING

4.1 UNIT TESTING

Unit testing is a Software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinised for proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff. The main objective of unit testing is to isolate written code to test and determine if it works as intended.

Unit testing is an important step in the development process, because if done correctly, it can help detect early flaws in code which may be more difficult to find in later testing stages.

Unit testing is a component of test-driven development(TTD), a pragmatic methodology that takes a meticulous approach to building a product by means of continual testing and revision. This testing method is also the first level of software testing, which is performed before other testing methods such as integration testing. Unit tests are typically isolated to ensure a unit does not rely on any external code or functions. Testing can be done manually but is often automated.

4.2 INTEGRATION TESTING

Integration testing is a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. Integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as 'I & T' (Integration and Testing), String Testing and sometimes Thread Testing.

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

Once all the components or modules are working independently, then we need to check the data flow between the dependent modules is known as integration testing.

4.3 FUNCTIONAL TESTING

Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations. The testing is done by providing sample inputs, capturing resulting outputs, and verifying that actual outputs are the same as expected outputs.

Unlike non-functional testing, functional testing isn't concerned with investigating the quality, security, or performance of the application's underlying source code. It doesn't gauge speed, scalability, and reliability.

Rather, functional testing focuses on the results of processing and not the mechanics of the processing, and determines whether the application satisfies the basic minimum user expectations.

In this sense, the functional testing definition is near synonymous with black-box testing, whereas white-box testing on the other hand is more commonly a characteristic of non-functional tests. For a deeper dive into functional and non-functional testing.

4.4 SYSTEM TESTING

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behaviour of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements.System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartially. It has both functional and non-functional testing.

4.5 WHITE BOX TESTING

The box testing approach of software testing consists of black box testing and white box testing. We are discussing here white box testing which is also known as glass box testing, structural testing, clear box testing, open box testing and transparent box testing. It tests internal coding and infrastructure of a software focused on checking predefined inputs against expected and desired outputs. It is based on the inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is

to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.

Developers do white box testing. In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform the Black box testing and verify the application along with the requirements and identify the bugs and send it to the developer.

The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular feature is working fine on the application.

4.6 BLACK BOX TESTING

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure. Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues. Black box testing is a powerful testing technique because it exercises a system end-to-end. Just like end-users “don’t care” how a system is coded or architected, and expect to receive an appropriate response to their requests, a tester can simulate user activity and see if the system delivers on its promises. Along the way, a black box test evaluates all relevant subsystems, including UI/UX, web server or application server, database, dependencies, and integrated systems. An example of a security technology that tests products in staging or production and provides feedback on compliance and security issues.

4.7 ACCEPTANCE TESTING

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

User acceptance testing (UAT) is a type of testing, which is done by the customer before accepting the final product. Generally, UAT is done by the customer (domain expert) for their satisfaction, and checks whether the application is working according to given business scenarios, real-time scenarios.

In this, we concentrate only on those features and scenarios which are regularly used by the customer or mostly user scenarios for the business or those scenarios which are used daily by the end-user or the customer.

However, the software has passed through three testing levels (Unit Testing, Integration Testing, System Testing) But still there are some minor errors which can be identified when the system is used by the end user in the actual scenario.

CHAPTER 5

5.SYSTEM STUDY

5.1 PYTHON FOR PROGRAMMING

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasises readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

5.2 WHY PYTHON?

According to the TIOBE index, which measures the popularity of programming languages, Python is the third most popular programming language in the world, behind only Java and C. There are many reasons for the ubiquity of Python, including:

- **Its ease of use.** For those who are new to coding and programming, Python can be an excellent first step. It's relatively easy to learn, making it a great way to start building your programming knowledge.
- **It's simple syntax.** Python is relatively easy to read and understand, as its syntax is more like English. Its straightforward layout means that you can work out what each line of code is doing.
- **Its thriving community.** As it's an open-source language, anyone can use Python to code. What's more, there is a community that supports and develops the ecosystem, adding their own contributions and libraries.
- **Its versatility.** As we'll explore in more detail, there are many uses for Python. Whether you're interested in data visualisation, artificial intelligence or web development, you can find a use for the language.

5.3 INTERACTIVE VOICE RESPONSE

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 synthesised 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 to advanced and dynamic data, such as e-mails, reports and news and data about weather. IVR is used in automobile systems for easy operations too. Text To Speech is a system originated synthesised speech that's not the robotic voice

historically related to computers. Original voices produce the speech in portions that are joined together and rounded before played to the caller.

5.4 SPEECH RECOGNITION

Speech recognition is the interdisciplinary 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 analyses 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 dialling (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), domotic 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 modelling. Acoustic modelling represents the relationship between linguistic units of speech and audio signals; language modelling 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 dialling 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.

5.5 SPEECH RECOGNITION IN PYTHON

The improvement and accessibility 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 packages 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 to work in your particular case.

5.6 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 well 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 the 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.

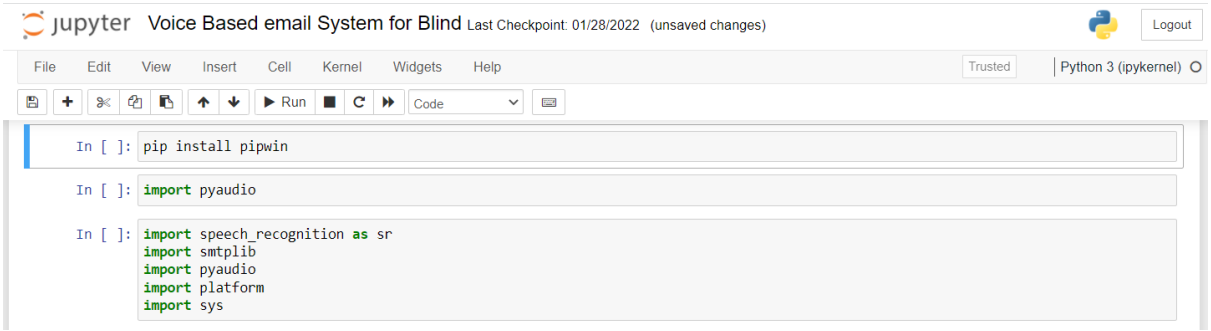


Figure 5.1 - Required Installation

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 a static method of the 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 Microphone is a context manager. Input can be captured from the microphone using the listen() method of the 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().

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 gives you the option to set the duration for wait time.

CHAPTER 6

6. PROJECT DESCRIPTION

MODULES EXPLANATION

6.1 CONVERTOR

In our Proposed system for Speech recognition we are using Two Modules such as :

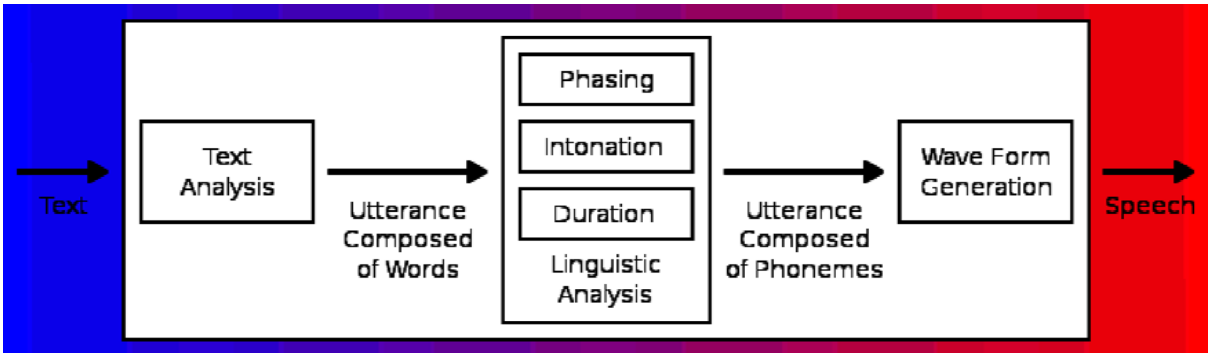
- Speech - to - Text (STT)
- Text - to - Speech (TTS)

SPEECH - TO - TEXT (STT)

In this STT process the speech is given as the input using the microphone. The Output will be in the form of a Text. In this process to convert a Speech to a text there are various Libraries in python such as Google Cloud speech, WIT, pyaudio, Speech Recognition. Here the user gives the speech as an input, which the system recognizes and converts it into a text. If everything is done from the user side then the user can send the mail to the selected participant by declaring the appropriate command.

TEXT - TO - SPEECH (TTS)

In this TTS Process the Text is given/taken as an input and the Speech will be the output. In our project the system will read the messages for visually challenged people, although it is a difficult task this can be implemented by using the specific commands. This mechanism will be much more helpful for the visually challenged people for communications.



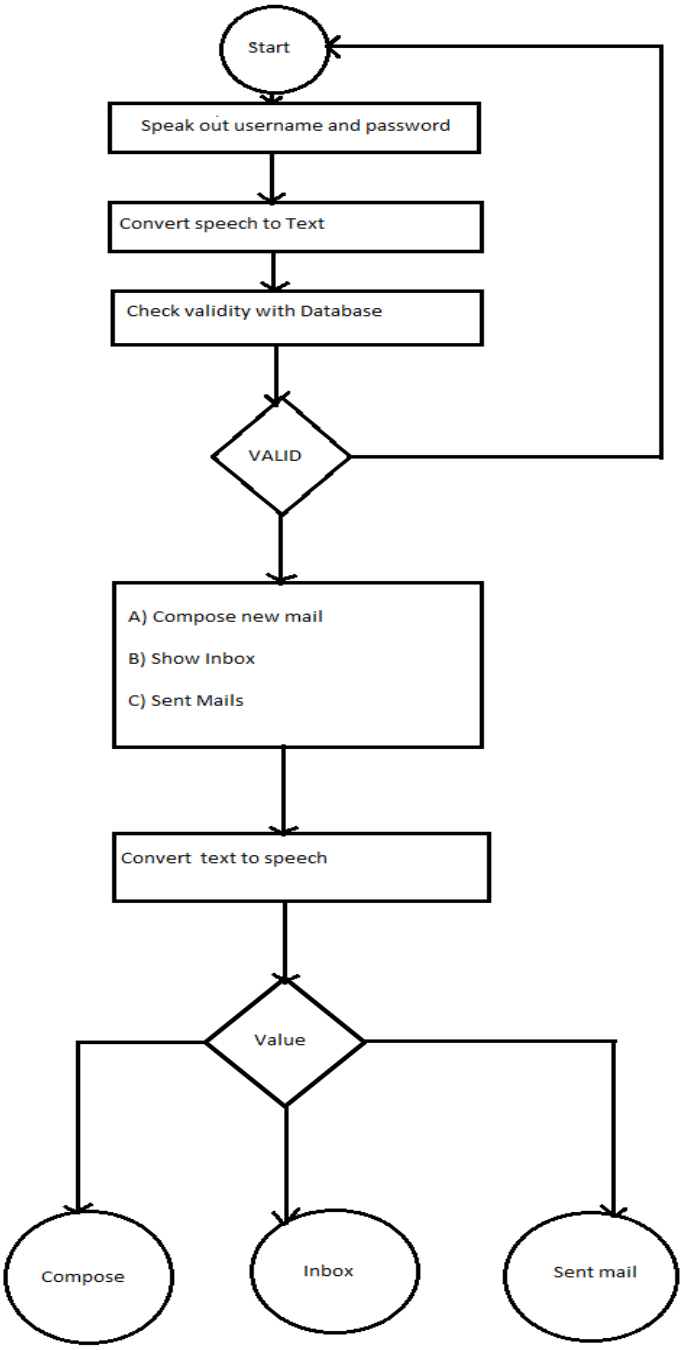


Figure 6.1 - Converter (TTS/STT)

6.2 INBOX

In this module the user is able to view (Listen) all the mails that have been received to his/her mail. The user can listen to the mails he/she wants to hear with the help of certain special commands. If the user needs to navigate through different mails, the system will specify which operations want to be performed. When a particular mail is selected the system will prompt the user about the details of who the sender is and what is the subject of the particular mail. Accordingly the user can decide whether the mail needs to be read out by the system or not.

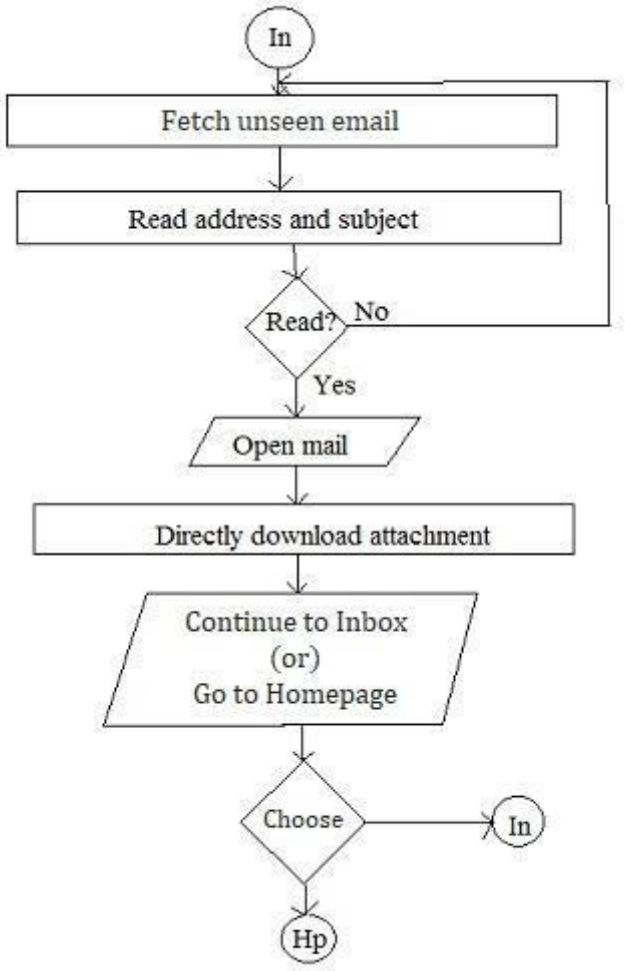


Figure 6.2 - Inbox

6.3 COMPOSE MAIL

This is the most important option provided by the mail services. The functionality of the Compose mail option in our email system Is completely different from the existing mail system. Since our system is for visually challenged people therefore the keyboard operations and mouse operations are completely avoided, the composing mail would be done through the Voice Input method (Speech Recognition). The user can hear the system and compose the mail they want to send, once the mail is composed the mail will be ready to send for a selected participant.

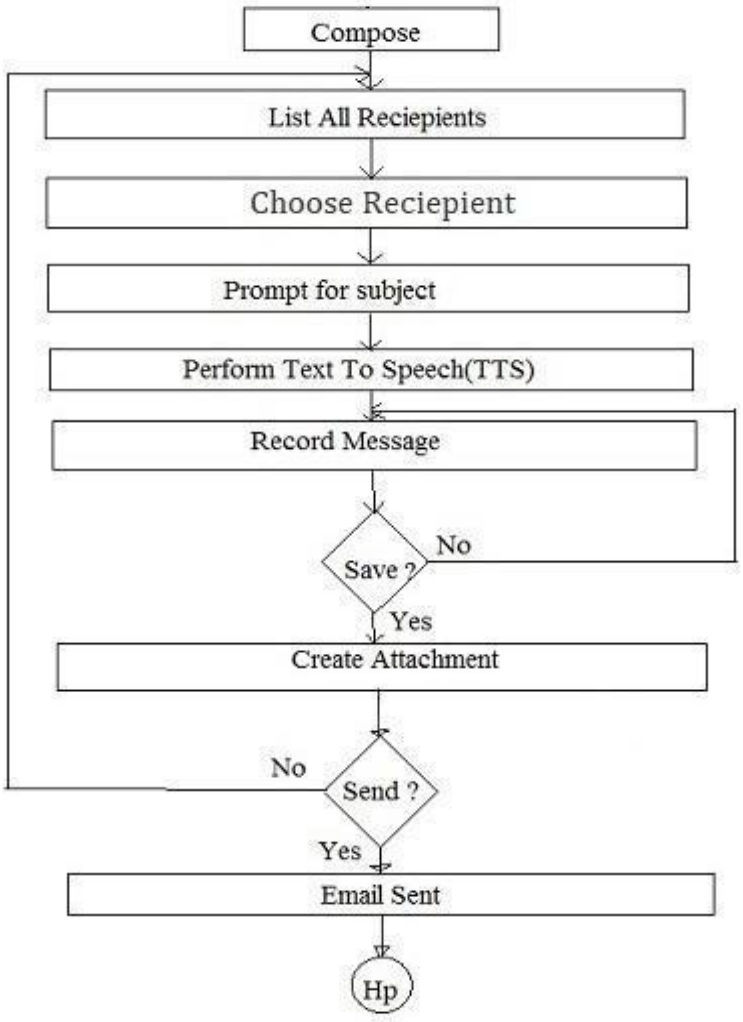


FIgure 6.3 - Compose Mail

6.4 SENT MAIL

The use of the sent mail option is to keep track of all the mails which had been sent by the user. If the user wants to access these mails this option will help the user by providing the details the user needs. In order to access the sent mails the user needs to perform certain actions provided by the system. If the user accessed the particular mail the system will provide the information as who the receiver was and what is the subject of the mail. This will help them in efficiently understanding the process and extracting the required mail.

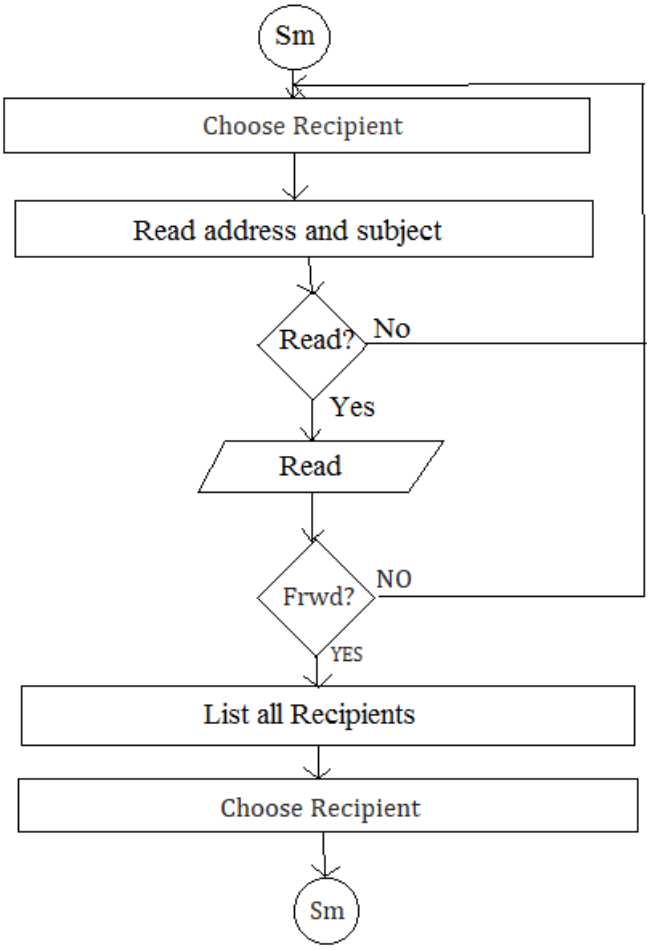


Figure 6.4 - Sent Mail

CHAPTER 7

7. SYSTEM DESIGN

7.1 SYSTEM ARCHITECTURE

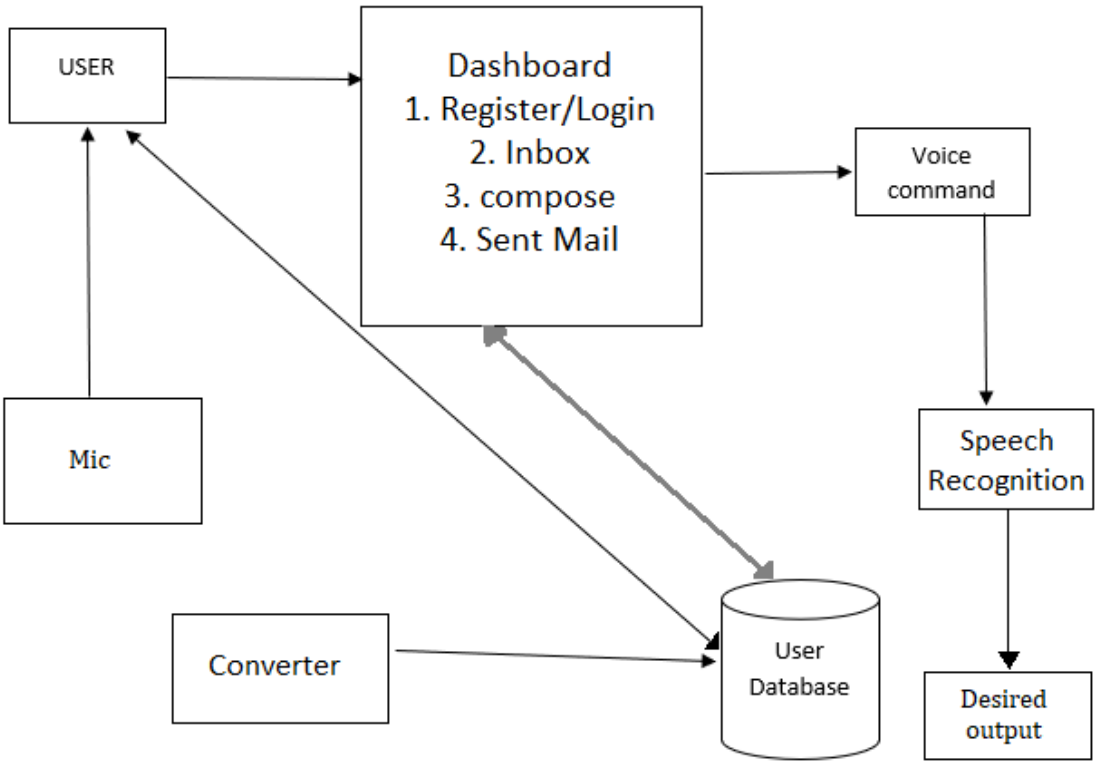


Figure 7.1 - System Architecture

In our system developed by us will first bring the user to the register/login page. The user will be prompted with the help of IVR throughout each and every checkpoint. The credentials gained in the above process will be recorded in the database. Then the user will be directed to the dashboard or the home page where the user will have to select from the menu options(Inbox, Compose, Sent mail). Based on the user's choice, the respective actions will be performed with the help of Guided Voice command and IVR. In Inbox, if the user selects any received mails the system will fetch it from the database and convert it using the Text-to-speech converter and prompt the user.

In compose mail, the user can give the voice input that will be converted by the speech to text converter and the mail can be sent by performing certain voice commands.

In sent mail the user will be prompted with the mail that has been sent by the user along with the name of the receiver and the subject of the mail and by performing certain voice command the user can read the sent mail.

7.2 DATA FLOW DIAGRAM

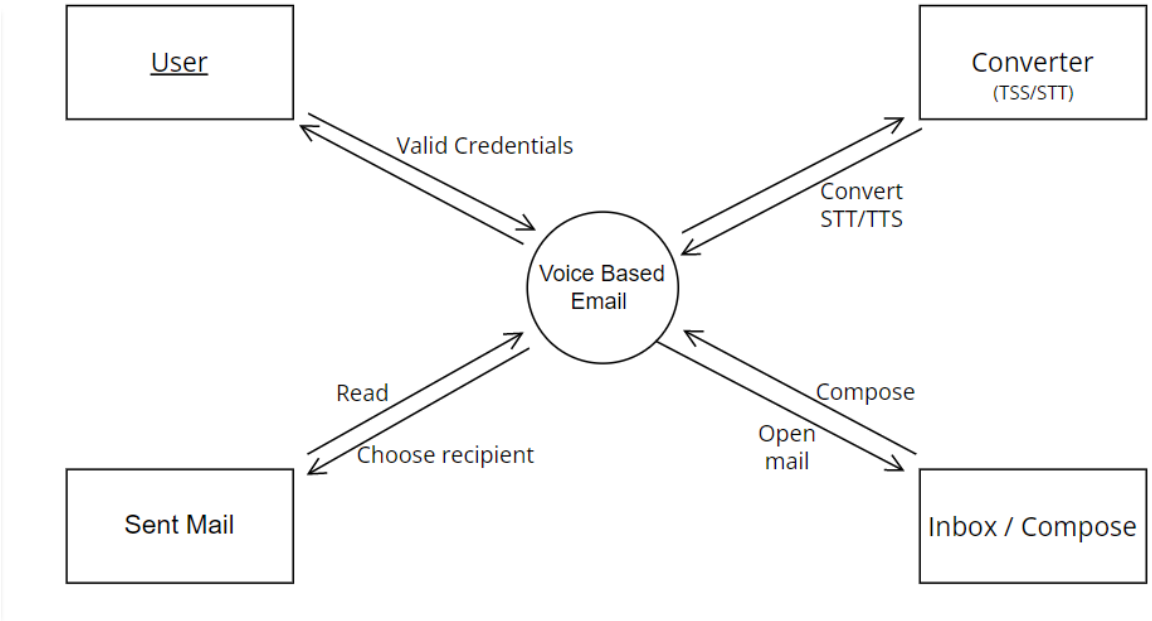


Figure 7.2 - Data Flow Diagram

DATABASE DESIGN

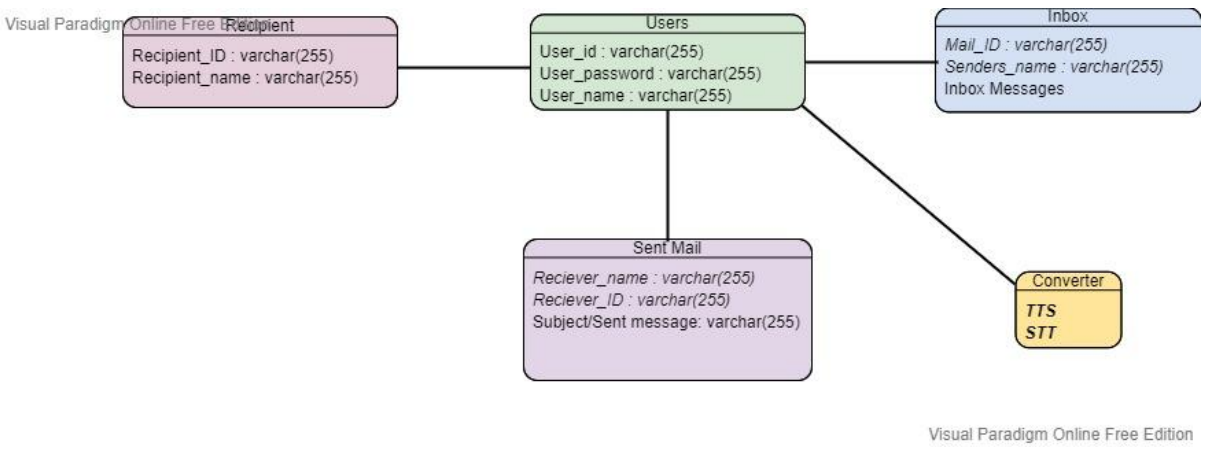


Figure 7.3 - Database Design

7.3 OPERATIONAL DESIGN

7.3.1 UML DIAGRAM:

UML stands for Unified Modelling Language. UML is a standardised general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML consists of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of an operation system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing object- oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects/operational system.

The Unified Modelling Language (UML) is a general purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualise the design of a system.

The creation of UML was originally motivated by the desire to standardise the disparate notational systems and approaches to software design. It was developed at Rational Software in 1994–1995, with further development led by them through 1996.

In 1997, UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organisation ever since. In 2005, UML was also published by the International Organisation for Standardisation (ISO) as an approved ISO standard. Since then the standard has been periodically revised to cover the latest revision of UML. The Object Management Group (OMG) has developed metamodeling architecture to define the UML, called the Meta-Object Facility. MOF is designed as a four-layered architecture, as shown in the image at right. It provides a meta-meta model at the top, called the M3 layer. This M3-model is the language used by Meta-Object Facility to build metamodels, called M2-models.

7.3.2 UML GOALS

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialisation mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modelling language.
5. Support higher level development concepts such as collaborations, frameworks.

7.3.3 USE CASE DIAGRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher- level view of the system. It has been said before that "Use case diagrams are the blueprints for your system".

Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand

how the system is going to be designed. Siau and Lee conducted research to determine if there was a valid situation for use case diagrams at all or if they were unnecessary. What was found was that the use case diagrams conveyed the intent of the system in a more simplified manner to stakeholders and that they were "interpreted more completely than class diagrams".

The purpose of a use case diagram is to capture the dynamic aspect of a system. Additional diagrams and documentation can be used to provide a complete functional and technical view of the system. They provide a simplified and graphical representation of what the system must actually do.

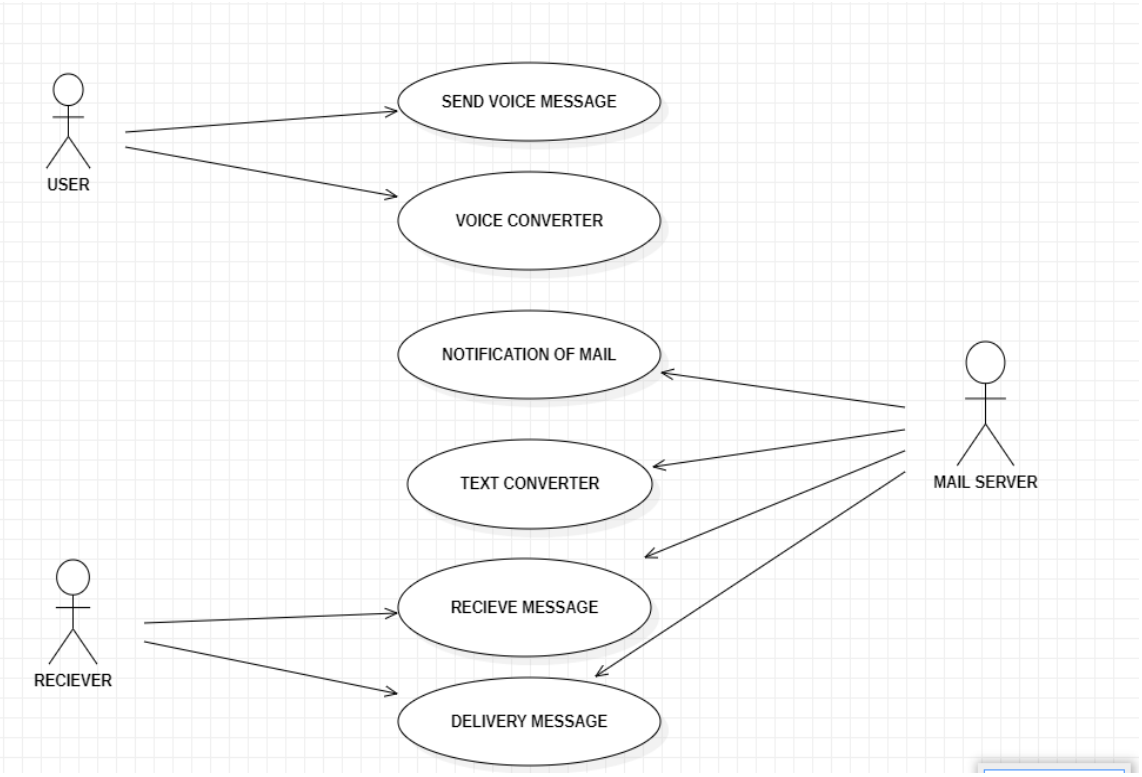


Figure 7.4 - Use Case Diagram

7.3.4 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. Messages, written with horizontal arrows with the message name written above them, display interaction. Solid arrowheads represent synchronous calls, open arrowheads represent asynchronous messages, and dashed lines represent reply messages.

If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine. If a caller sends an asynchronous message, it can continue processing and doesn't have to wait for a response. Asynchronous calls are present in multithreaded applications, event-driven applications and in message-oriented middleware. Activation boxes, or method-call boxes, are opaque rectangles drawn on top of lifelines to represent those processes that are being performed in response to the message (Execution Specifications in UML).

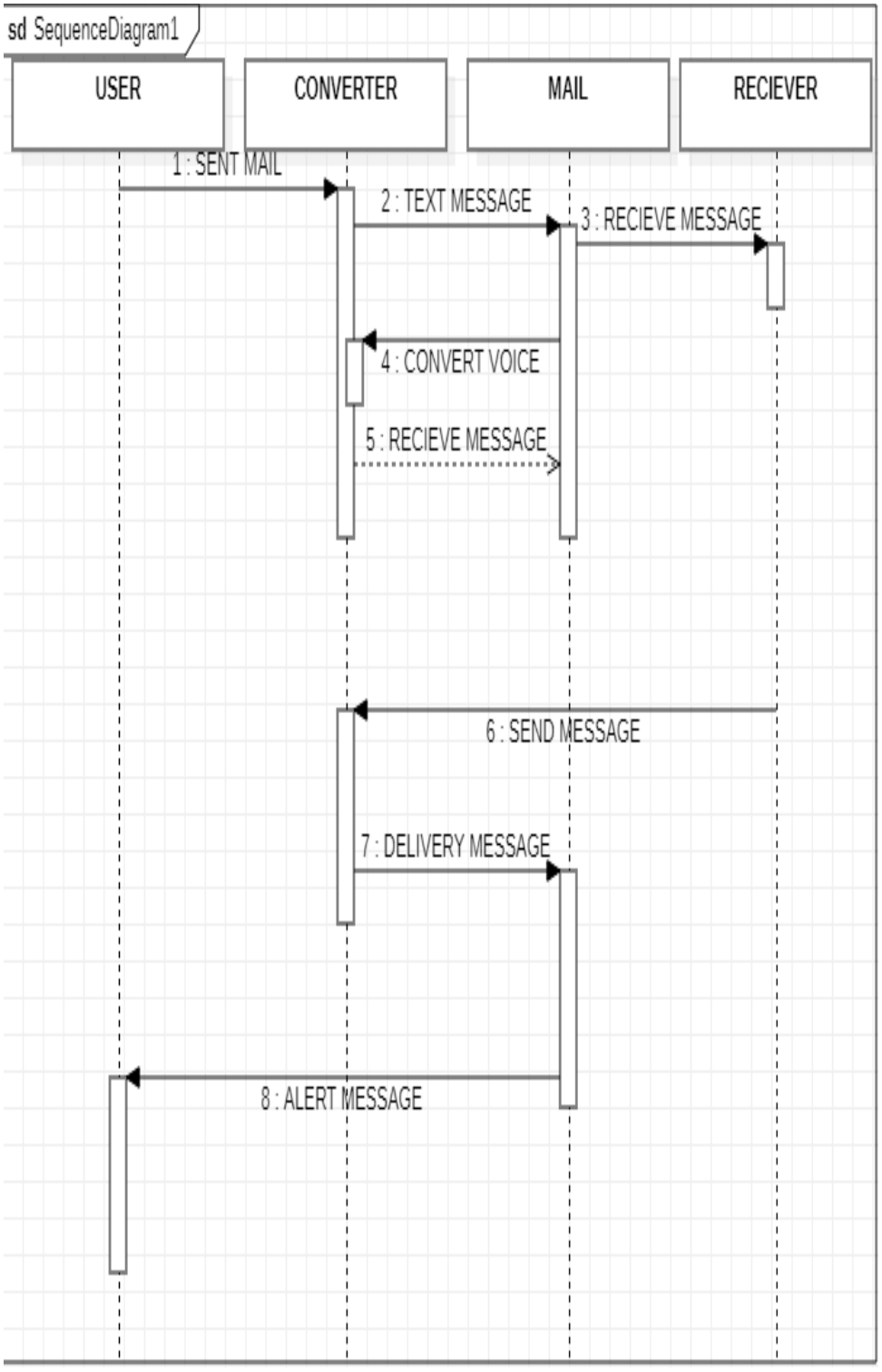


Figure 7.5 - Sequence Diagram

CHAPTER 8

8. CONCLUSION & FUTURE ENHANCEMENTS

8.1 CONCLUSION

On successful completion, This project will help not only the visually challenged but also other people who face problems accessing the email for communication. This will completely erase all the hurdles faced by the Visually impaired as it relies on IVR(Interactive Voice Response) and Voice commands. Here, we eliminate all sorts of physical input, as a result the user can send or receive email without any hassle and need not get the help of any third person.

Since the system uses only voice response, even a naive user or user without any previous experience can use this with ease. The system developed by us now works only on desktops. As use of mobile phones is emerging as a trend today, there is a scope to incorporate this facility as an application in mobile phones also. Also, security measures to be implemented during the login phase can be revised to make the system safer and reliable.

8.2 FUTURE ENHANCEMENTS

Many researches are undergoing regarding the usage of the internet by the visually challenged. In the total population of the world there are about 60% of visually challenged in India. The literacy rate of visually challenged is also getting high. Hence it is necessary to make developments in the visual perception of the internet to make them use the internet like a normal person. Currently available systems such as Screen readers, Text Notifiers, etc. also require some visual perception by disabled to create accounts and to use them. An interactive voice-based email system can help visually challenged to communicate easily in this modern world. To ensure privacy there are systems to identify human presence. Many sensors such as accelerometer, gyrometer or gyroscopes are getting embedded in many devices and smart phones as Micro Electro Mechanical Elements hence they are embedded. If such sensors are embedded in the devices and are specially designed for the visually challenged, then it will be very useful for them to use email and other applications.

This implementation (project) has a lot of potential in the future, with a lot of enhancements; perhaps it was a system that, in any language, had the functions of email access and spam emails. In addition, this system can be upgraded to send an attachment, and it is also very useful for people with poor eyesight. It can be made available to all the region's people, which is quite popular and will continue to be available in multiple languages; the system is simple and easy to access. Furthermore, the system employs sign language and can be integrated into it to make it more scalable, reliable and Robust.

The project will also be benefited with implementations like, associating or combining vibration technologies (vibrators) and haptics (haptic motors) that are embedded in devices from recent times with the application so that the user gets a seamless experience with actual physical feedback.

Also in future, we attempt to make the system autonomous. So, it's easy for the visually impaired people to access the services. The system developed now is working only on desktops. As use of mobile phones is emerging as a trend today, there is a scope to incorporate this facility as an application in mobile phones also. Also, security measures to be implemented during the login phase can be revised to make the system safer.

CHAPTER 9

9. APPENDIX

9.1 CODE

```
import speech_recognition as sr
import easyimap as e
import pyttsx3
import smtplib
from email.message import EmailMessage

        #Login credentials of our mail id

#unm = "email_id@gmail.com"
#pwd = "***Password***"

r = sr.Recognizer()

        #Defining an engine for text to speech conversion

engine = pyttsx3.init()
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
engine.setProperty('rate', 200)

def speak(str):
    print(str)
    engine.say(str)
    engine.runAndWait()

def listen():
    with sr.Microphone() as source:
        r.adjust_for_ambient_noise(source)
        str = "Speak Now:"
```

```
        speak(str)
        audio = r.listen(source)
        try:
            text = r.recognize_google(audio)
            return text
        except:
            str = "Sorry could not recognize what you said"
            speak(str)


def sendmail(reciever, subject, message):

        #This is a server, because when we send a message or
        email, this email goes to the server and then the server
        will send the mail to the specific person we want to send it
        to.

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

#we use gmail to access email and here we have to specify the
port no. which is 587

        server.starttls()

#TransportLayerSecurity=assuring or telling the server that you
can trust me

    server.login(unm, pwd)
    email = EmailMessage()
    email['From'] = unm
    email['To'] = reciever
    email['Subject'] = subject
    email.set_content(message)
```

```
    server.send_message(email)

email_list = {
    'Ashok': 'akshaysrandomemail@gmail.com',
    'Sat': 'satmail16@gmail.com',
    'Sunil': 'sunilsubramani23@gmail.com',
    'Tamil': 'tamilatiger22@gmail.com'
}

def get_email_info():
    speak('To Whom you want to send email')
    name = listen()

    if name in email_list:
        reciever = email_list[name]
        print(reciever)
        speak('What is the subject of your email?')
        subject = listen()
        str = "subject is"
        speak(str)
        speak(subject)
        speak('Speak the body for your email')
        message = listen()
        str= "Body of the mail is:"
        speak(str)
        speak(message)
        sendmail(reciever, subject, message)
        speak('Your Email has been sent successfully.')
        speak('Do you want to continue to the main menu? Please
Say YES or NO ')
        send_more = listen()
        if 'yes' in send_more:
            menu()
```

```
        else:
            str = "Thank you for using our service."
            speak(str)
            exit(1)

    else:
        str = "Sorry. Could not get it. Please try again."
        get_email_info()

def readmail():

    server = e.connect("imap.gmail.com", unm, pwd)
    server.listids()

     str = "Please say the Serial Number of the Email you wanna
read starting from the latest."
    speak(str)

    a = listen()
    if(a == "Tu"):
        a = "2"

    b = int(a) - 1

    email = server.mail(server.listids()[b])

    str = "The email is from: "
    speak(str)
    speak(email.from_addr)
    str = "The subject of the email is:"
    speak(str)
    speak(email.title)
    str = "The body of email is:"
```

```
    speak(str)
    speak(email.body)


speak("Before we start, Please mention your gmail ID")

#Login credentials of our mail id

unm = listen()
unm = unm.lower()
unm = unm.replace("at gmail.com","@gmail.com")
unm = unm.replace(" ","")
speak("The username is:")
speak(unm)
speak("Please Speak the password")
pwd = listen()
pwd = pwd.replace("at the rate","@")
pwd = pwd.replace("dot",".")
pwd = pwd.replace(" ","")
speak(pwd)

def menu():
    str = "Welcome to voice controlled email service"
    speak(str)

    while(1):

        str = "What do you want to do?"
        speak(str)

         str = "Speak SEND to Send Email. Speak READ to Read
Inbox. Speak EXIT to Exit"
        speak(str)
```

```
    ch = listen()

    if (ch == 'send'):
        str = "You have chosen to send an email"
        speak(str)
        get_email_info()
        sendmail()

    elif (ch == 'read'):
        str = "You have chosen to read email."
        speak(str)
        readmail()

    elif (ch == 'exit'):
        str = "You have chosen to exit, Bye Bye"
        speak(str)
        exit(1)

    else:
        str = "Invalid choice, you said:"
        speak(str)
        speak(ch)

menu()
```

9.2 OUTPUT

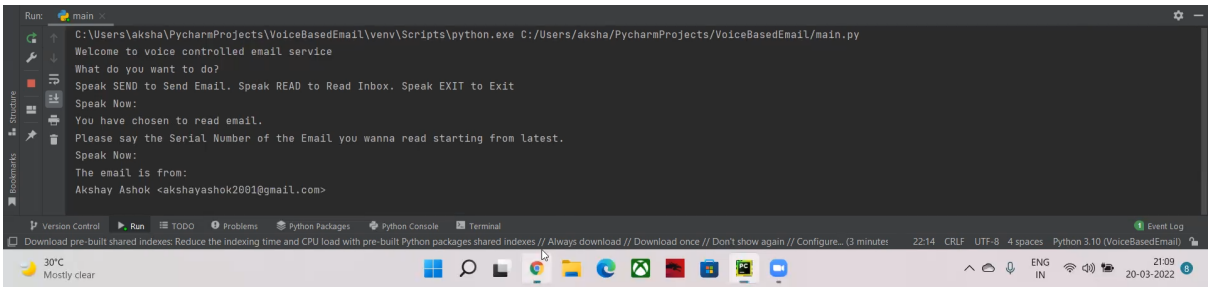


Figure 12 - Output 1

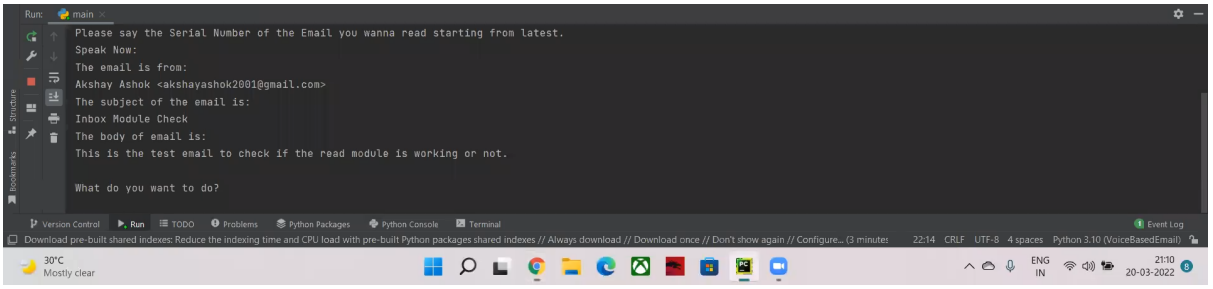


Figure 13 - Output 2

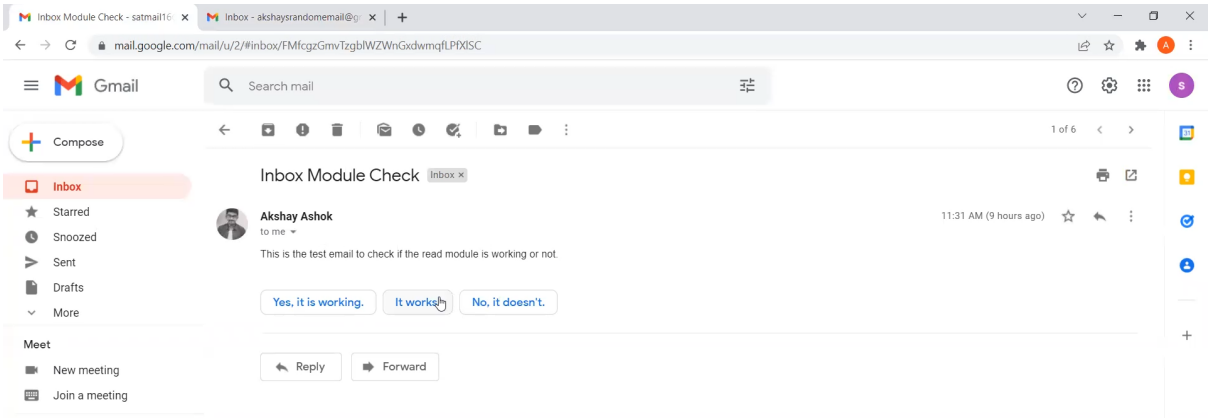


Figure 14 - Output 3

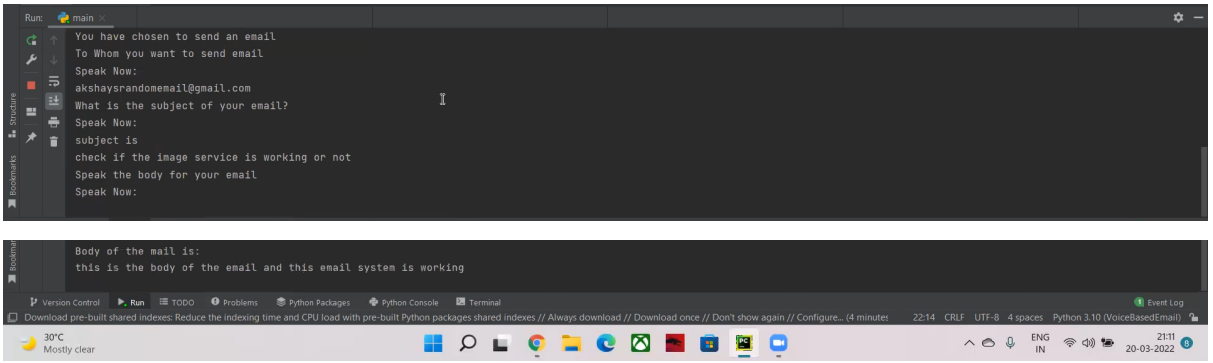


Figure 14 - Output 4

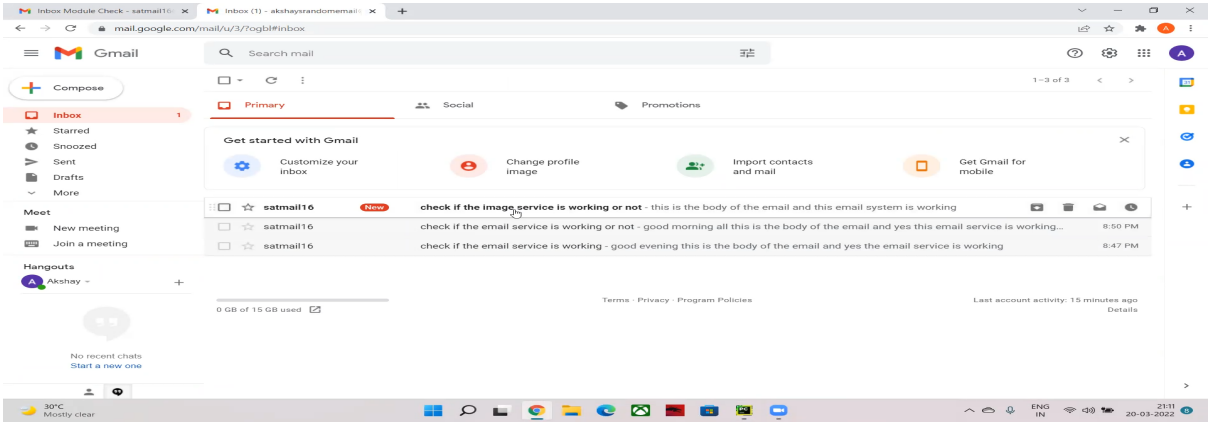


Figure 15 - Output 5

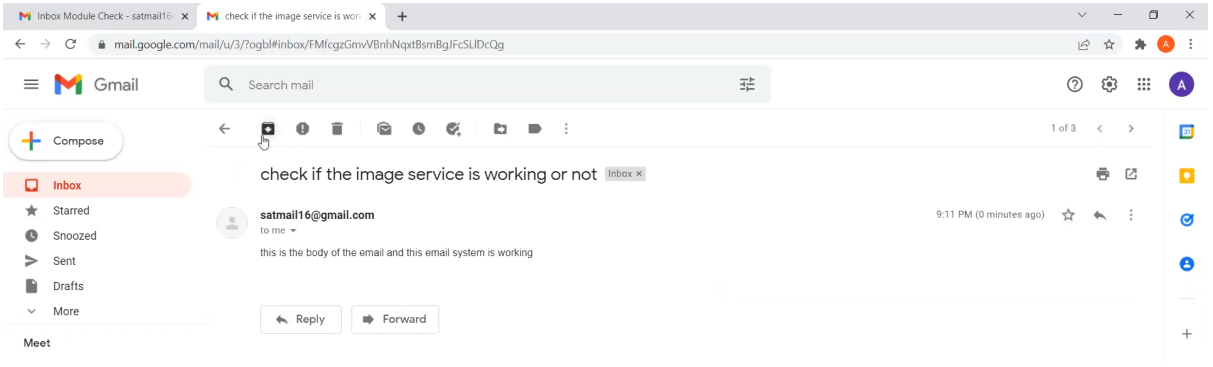


Figure 16 - Output 6

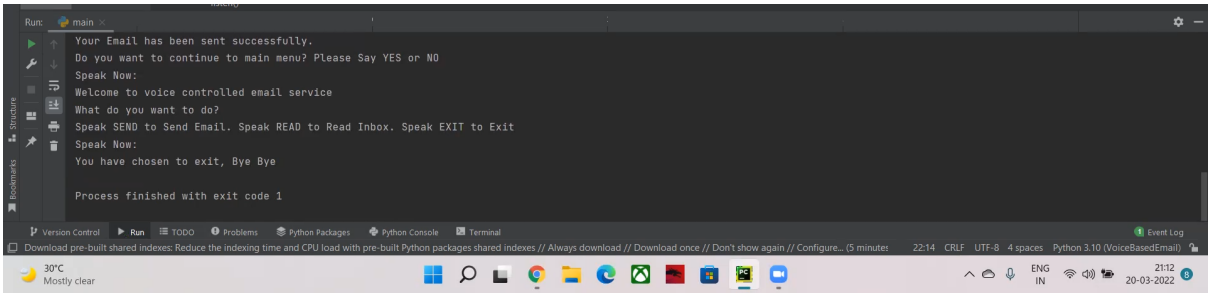


Figure 17 - Output 7

CHAPTER 10

10. REFERENCES

1. Pranjal Ingle, Harshada Kanade, Arti Lanke “Voice based Email System for Blinds” in International Journal of Research Studies in Computer Science and Engineering (IJRSCSE) Volume 3, Issue 1, 2016.
2. Maham Imtiaz, Samina Khalid, Saleema Khadam, Sumaira Arshad, Ali Raza, Tehmina Khalil “Blind Electronic Mail System” Published by Canadian Centre of Science and Education.
3. Rijwan Khan, Pawan Kumar Sharma, Sumit Raj, Sushil Kr. Verma, Sparsh Katiyar, “Voice Based EMail System using Artificial Intelligence”,in International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020.
4. Mullapudi Harshasri, Manyam Durga Bhavani, and Misra Ravikanth “Voice Based Email for Blind”, International Journal of Innovative Research in Computer Science & Technology (IJIRCST), ISSN: 2347-5552, Volume-9, Issue-4, July 2021, https://doi.org/10.21276/ijircst.2021.9.4.2, Article ID : IRP1185, Pages 10-13, www.ijircst.org
5. T.Shabana, A.Anam, A.Rafiya, K.Aisha “Voice based email system for blinds”, in International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 1, January 2015.
6. 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).
7. Ummuhani Sifa U, Nizar Banu P K , “Voice Based Search Engine and Web page Reader”. In International Journal of Computational Engineering Research (IJCER). Pages 1-5.
8. The WHO website. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/
9. Saurav Mishra, Ashwani Panwar “VOICE BASED EMAIL SYSTEM FOR VISUALLY IMPAIRED”, https://pdfcoffee.com/voice-based-email-system-for-visually-impaired-pdf-free.html

10. C. Kang, H. Jo and B. Kim, “A Machine-to-Machine based Intelligent Walking Assistance System for Visually Impaired Person”, The Journal of KICS, vol. 36, no. 3, (2011), pp. 195-304.
11. K. Jayachandran, P. Anbumani “Voice Based Email for Blind People” from International Journal of Advanced Research, ideas and Innovations in Technology. Available online at www.ijariit.com
12. Aishwarya Belekar, Shivani Sunka, Neha Bhawar, Sudhir Bagade, ”Voice based Email for the Visually Impaired” in International Journal of Computer Applications (0975 – 8887), Volume 175– No. 16, September 2020.
13. K. Venkatesh, P. Santhosh Kumar, A. Sivanesh Kumar “Voice Based EMail System For Visionless People And Object Detection Using Optimization Technique” in International Journal Of Scientific & Technology Research VOLUME 9, Issue 02, FEBRUARY 2020 ISSN 2277-8616.
14. Prof. Umesh A. Patil, Pranouti B. Patil, Teja P. Magdum, Shweta K. Goud and Latika R. Bhosale, "A Survey on Voice Based Mail System for Physically Impaired Peoples". International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE) - Volume 4, Issue 1, January 2016, pp. 1002-1006.
15. Bishal Kalita and Santosh Kumar Mahto, "Voice Based Email for Blind People". International Journal of Engineering Science and Computing (IJESC) - Volume 9, Issue 10, October-2019, pp. 23789-23799.
16. Saurabh Sawant, Amankumar Wani, Sangharsh Sagar, Rucha Vanjari and M R Dhage, "Speech Based Email System for Blind and Illiterate People''. International Research Journal of Engineering and Technology (IRJET) - Volume 05, Issue 04, April-2018, pp. 2398-2400.
17. 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.