Visvesvaraya Technological University
Belgaum, Karnataka- 590014

A Project Report On

"Voice Based Email for Visually Impaired"

Submitted in the partial fulfilment of the requirements for the award of the Degree of
BACHELOR OF ENGINEERING
In
INFORMATION SCIENCE AND ENGINEERING

ACCREDITED BY NBA

Submitted by

Rakshitha A (1DS17IS076)
Rachana N Shenoy (1DS17IS119)
Viram Jain (1DS17IS113)
Krithika A K (1DS17IS127)

Under the Guidance of
Dr. Suma V
Professor, Dept. of ISE

2020-2021

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
DAYANANDA SAGAR COLLEGE OF ENGINEERING
SHAVIGE MALLESHWARA HILLS, KUMARASWAMY LAYOUT, BANGALORE-78

DAYANANDA SAGAR COLLEGE OF ENGINEERING

Shavige Malleshwara Hills, Kumaraswamy Layout
Bangalore-560078

Department of Information Science and Engineering

ACCREDITED BY NBA

2020-2021

Certificate

This is to certify that the Project Work entitled **—"Voice Based Email for Visually Impaired"** is a bonafide work carried out by **Rakshitha A** (1DS17IS076), **Viram Jain** (1DS17IS113), **Rachana N Shenoy**(1DS17IS119), and **Krithika A K** (1DS17IS127) in partial fulfilment for the 8th semester of Bachelor of Engineering in Information Science & Engineering of the Visvesvaraya Technological University, Belgaum during the year 2020-2021. The Project Report has been approved as it satisfies the academics prescribed for the Bachelor of Engineering degree.

Signature of Guide	Signature of HOD	Signature of Principal
[Dr. Suma V]	[Dr. K.N. Rama Mohan Babu]	[Dr. C.P.S Prakash]

Name of the Examiners	Signature with Date
1.	
2.	

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project.

We take this opportunity to express our sincere gratitude to **Dayananda Sagar College of Engineering** for having provided us with a great opportunity to pursue our Bachelor Degree in this institution.

In particular we would like to thank **Dr. C. P. S Prakash,** Principal, Dayananda Sagar College of Engineering for his constant encouragement and advice.

Special thanks to **Dr. K.N. Rama Mohan Babu,** HOD, Department of Information Science & Engineering, Dayananda Sagar College of Engineering for his motivation and invaluable support through the development of this project.

We are highly indebted to our internal guide **Dr. Suma V,** Professor, Department of Information Science & Engineering, Dayananda Sagar College of Engineering for her constant support and guidance. She has been a great source of inspiration throughout the course of this project.

Finally, we gratefully acknowledge the support of our families during the completion of the project.

Rakshitha A (1DS17IS076)
Viram Jain (1DS17IS113)
Rachana N Shenoy(1DS17IS119)
Krithika A K (1DS17IS127)

ABSTRACT

In today's world, the internet plays an important role. Everyone is widely accessing knowledge and information through the internet on a day-to-day basis. The most significant aspect of communication is email. However, some people are blind or unable to read. As a result, surviving in the internet age becomes increasingly challenging for them. Various technologies, such as screen readers, are now available. But they are inefficient in their application. As using technology necessitates visual perception, visually impaired persons have a tough time gaining access to it. This project describes a voice-based mail system that the visually handicapped can use to efficiently manage their mail and will allow visually impaired people to send and receive messages effortlessly. It also does away with the traditional methods of employing screen readers, which necessitates users remembering keyboard shortcuts, adding to the cognitive stress on visually challenged people.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1: INTRODUCTION

1.1 Overview

With the emergence of the Internet, several sectors were changed. From the convenience of our own homes, we can obtain any information required. Communication is one of the primary areas which has changed the Internet. The first thing that springs to mind regarding communication through the Internet is e-mail. E-mail is seen as a more formal method of sending and receiving messages.

However, the most important criterion for gaining access to the internet is visual perception. According to surveys, India is home to over 60% of the world's visually impaired population. Consider the predicament of visually impaired people who must rely on a third party to perform any of these tasks. The advancements in computer technology have now created platforms for visually challenged persons all over the world.

The goal of the project is to build a python-based web application for visually impaired individuals that uses speech to text and text to speech technology that allows the visually impaired to control their email accounts and carry out different activities using only their voice.

1.2 Problem Statement

As a large part of the population, deprived of their sight, have a tough time accessing basic Internet facilities such as email because it necessitates visual perception. The proposed system is to build a voice based email application which will be useful to society, especially the blind. The main benefit of this system will be that the use of the keyboard is completely eliminated, the system will respond to voice commands and mouse clicks only.

1.3 Objectives

- Create a voice based web application to access email systems of recognised domains.
- Allows users to login, compose, read, send and delete mails using only voice commands (IVR).
- Allow users to start its functionalities based on single left-clicks anywhere on the screen, regardless of where the cursor is placed on the screen.

1.4 Motivation

It is estimated that nearly 285 million people in the world are visually impaired and the idea is to facilitate a suitable communication system for them. This reason was the driving force behind developing the given system. One of the major disadvantages of the existing system is that all operations are based on mouse click events. Operations depend completely on types of clicks specified by idea. Also sometimes remembering keyboard shortcuts is difficult. The extent of the existing system is limited for blind and visually impaired people. There is a high need of developing a proper system.which curbs all the above drawbacks and turn into a simple system. Idea focuses on providing basic functionalities like compose, send, receive Email and delete mails which goes to the trash.

2: LITERATURE SURVEY

A literature survey or a literature review in a project report shows the various analyses and research made in the field of interest and the results already published, taking into account the various parameters of the project and the extent of the project.

Literature survey describes the existing work on the given project. It deals with the problems associated with the existing system and also gives users a clear knowledge on how to deal with the existing problems and how to provide solutions to the different existing problems.

OBJECTIVES OF LITERATURE SURVEY

- Concentrate on your own field of expertise - Even if another field uses the same words, they usually mean completely.
- It improves the quality of the literature survey to exclude sidetracks. Remember to explicate what is excluded.
- Learning the definitions of the concepts.
- Access to latest approaches, methods and theories.
- Discovering research topics based on the existing research.

[1] Topic: Voice Based System in Desktop and Mobile Devices for Blind People.

Authors: Jagtap Nilesh, Pavan Alai, Chavhan Swapnil, Bendre M.R

Year: 2018

Introduction: Today in the information age the computer has become an integral part of everybody's life. The information access and computer handling has to be done with the mouse and keyboard and by reading all the things present on the screen and then deciding what to do making it a visual process which means we need eyesight to handle the information on the computer. The blind people cannot read the information and cannot view the mouse cursor to give commands to the computer. They cannot access their mail and cannot send mail. Keeping in view all of these, the goal of the project is to build a system that allows the blind to convert speech to text, the system directly sends recorded voice messages to recipients' mail address as an attachment.

Methodology: In user authentication module user has to give login information such as his/her username, password through voice command. In the send mail module the compose window will open; the user has the option of either to record a voice message or to type text. In the inbox module, the blind user can check the voicemail received in the mailbox. The GUI operation is accessed by using voice command and mouse operation performed by the user instead of searching the short-cut key from the keyboard. Proposed system reads messages on mobile. As well as E-mail, other multimedia functions like (audio, text).

Conclusion: Voicemail architecture helps blind people to access email and other multimedia functions of operating systems (songs, text).Also in mobile applications SMS can be read by the system itself.

[2] Topic: Voice Based Email System for Blinds

Authors: Pranjal Ingle, Harshada Kanade, Arti Lanke
Year: 2017
Introduction: The Internet plays a vital role in today's world of communication.No work can be done without use of the internet. Electronic mail i.e. email is the most important part in day to day life. But some of the people in today's world don't know how to make use of internet, some are blind or some are illiterate.Therefore we came up with the project as voice based email system for blinds which will help a lot to visually impaired peoples and also illiterate peoples for sending their mails.The users of this system don't need to remember any basic information about keyboard shortcuts as well as location of the keys. Simple mouse click operations are needed for functions making the system easy to use for users of any age group.
Methodology: In this system mainly three types of technologies are used namely: STT (Speech-to-text), here whatever we speak is converted to text.TTS (text-to-speech) this method is the full opposite of STT. In this method, which converts the text format of the emails to synthesized speech.IVR (Interactive voice response): IVR is an advanced technology describes the interaction between the user and the system in the way of responding by using keyboard for the respective voice message.Audio voice to further assist users on how to proceed.The audio that would be pre-recorded and the system need to have large volumes.
Conclusion: This e-mail system can be used by any user of any age group with ease of access. It has features of speech to text as well as text to speech with a speech reader which makes a designed system to be handled by visually impaired people as well as blind people.

[3] Topic:Voice Based Email Application for Visually Impaired

Authors:Milan Badigar,Nikita Dias ,Jemima Dias,Mario Pinto

Year:2018

Introduction: Email is considered as one of the most pervasive forms of communication. However, all these technologies can be of no use to the people who are visually impaired as all activities that can be performed on the computer are based on visual perception. The application will not let the user make use of the keyboard, instead will work on text to speech and vice versa to facilitate sending, reading, forwarding and replying to emails using an android smartphone.
Methodology: The application is totally voice-based allowing blind people to send and receive emails on the go. It converts the user's spoken voice into text and performs the action accordingly. It consists of voice confirmation. Since users tend to forget their passwords or simply use weak passwords that allow an adversary to break into their email accounts, the application makes use of fingerprints , the user will have to use certain keywords which will perform certain actions.
Conclusion: This project proposes an android application, designed specifically for visually challenged people. This application provides a voice based mailing service where they could

read and send mail on their own, without any guidance. Here the users have to use certain keywords which will perform certain actions.

[4] Topic: Voice mail application for visually impaired persons

Authors: Taslima Binte Hossaina, Yeasmin Ara Akterb, Md. Ataur Rahmanc

Year: 2020

Introduction: Nowadays, the internet has become an inevitable part of our day to day life since most of the mobile applications run through the internet. One of the major applications is Email. Although, a lot of people can't use this application because of their visual impairment. To solve their problem, we have implemented a voice mail application which is accessible by voice and this application doesn't need any vision to be executed. The system processes user's voice inputs to use a mail system e.g. Gmail account. Several voice mail applications have already been made by scholars.

Methodology: The system is a voice mail android application rather than desktop application by which visionless people can easily transfer email through their voice. To make the application, several processes have been employed to transfer the Voice Signals to the gmail server. Google Speech API- This API is used to convert the audio into text by applying powerful neural network models,Google Text to Speech API- It takes a text stream and sends a request to Google Translate API Web servers to return an audio file with text converted into speech,Gmail API- This API gives the advantage of accessing Gmail mailboxes and send mail. Most of the programmers give highest priority to the Gmail API for authorized access to a user's Gmail data.

Conclusion: The voicemail application can be used easily and efficiently by a blind person to access mails. Thus reliance of visually impaired persons on other people for their activities related to mail can be reduced.

[5] Topic: An Application of Voice Mail: Email Services for the Visually Challenged Individual

Authors: Rastogi, Anshika Rajput, Archana, Komal

Year: 2019

Introduction: Internet is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic,wireless,and optical networking technologies. Email is defined as a way of exchanging information among people using electronic devices such as computers, mobile phones, tablets etc. Any technology can be considered beneficial if it can be accessed by all types of people. This will be an Android application that will be completely based on "voice or"speech".

Methodology: The user will be able to give commands to the system, which the system will follow. Moreover, the system will prompt the user to perform specific actions to avail respective services. The application will be accessible on all Android based devices and will be voice controlled. First the user needs to either register if he's a new user or login using his user id. After logging in, the user will be directed to the dashboard containing options for: Inbox, Compose new mail,Sent mail, User info. To compose a mail, the user will just have to give the input through his voice. No keyboard shortcuts or typed input will be required. When the user

speaks "Inbox", this screen will be displayed. The system will prompt the user about all the new mails received and read out the sender's name one by one. Sent Mails maintains a record of the emails sent by a particular user.

Conclusion: This application is user friendly, efficient and an economical system, which allows a visually challenged individual to interact with an Android application easily. This application can help in overcoming some of the drawbacks of the existing email systems. In this system, the use of a keyboard has been eliminated completely and thus reduces the cognitive load of remembering keyboard shortcuts as well as the position of the keys on a keyboard.

[6] Topic: Voice based email system for blinds

Authors: T.Shabana, A.Anam, A.Rafiya, K.Aisha

Year: 2015

Introduction: The Internet is considered as a major storehouse of information in today's world.And out of all methods available email is one of the most common forms of communication especially in the business world.However not all people can use the internet. This is because in order to access the internet you would need to know what is written on the screen. Therefore we have come up with this project in which users of this system would not need to have any basic information regarding keyboard shortcuts or where the keys are located. All functions are based on simple mouse click operations making it very easy for any type of user to use this system.

Methodology: The system is based on IVR- interactive voice response. Using this system the computer will be prompting the user to perform specific operations to avail respective services and if the user needs to access the respective services then he/she needs to perform that operation. Any user who wishes to use the system should first register to obtain username and password,Once the registration is done the user can login to the system. This module will ask the user to provide the username and password, the user is redirected to this page once login done successfully. The options available are: Inbox, Compose,Sent mail ,Trash**.** Prompt will provide the mouse click operation that needs to be performed for the required service.composing mail would only be done on voice input and mouse operations. No typed input will be required. Inbox helps the user view all the mails that have been received to his/her account.Sent mail will keep a track of all the mails sent by the user.

Conclusion: In this system ,the user only needs to follow the instructions given by the IVR and use mouse clicks accordingly to get the respective services offered. Other than this the user might need to feed in information through voice inputs when specified.

3: REQUIREMENTS

3.1 Functional Requirements

- This project allows the visually impaired to carry out different functionalities of an email system based on mouse clicks and voice commands(IVR) only.
- Each functionality is initiated with a single left-click anywhere on the screen, regardless of where the cursor is placed on the screen.
- Followed by text to speech and speech to text interactive voice response to carry out various tasks such as login,compose,read,receive,send and delete mails.
- This system completely eliminates the use of a keyboard.

3.2 Non-Functional Requirements

Usability

The client acknowledged being typical nearly the buyer interfaces and committed to ask for ambush pressure in relocating to a unique framework with another condition.

Reliability

The progressions made by the Programmer ought to be obvious both to the Project pioneer and in addition the Test design.

Security

Counting bugs following the framework must give important security and must secure the entire procedure from slamming.

Performance

The framework will be facilitated on a solitary web server with a solitary database server out of sight, consequently execution turns into a noteworthy concern.

Portability

This is required when the web server, which is facilitating the framework, stalls out because of a few issues, which requires their framework to be taken to another framework.

Reusability

The framework ought to be separated into such modules that it could be utilized as a piece of another framework without requiring a lot of work.

3.3 Hardware Requirements

- System : Intel i3 2.1 GHZ
- Memory : 4GB.
- Hard Disk : 10 GB
- Earphones with mic

3.4 Software Requirements

- Programming Languages : Python, HTML, CSS, JavaScript
- Framework : Django
- IDE : Visual Studio Code
- Operating System : Ubuntu 20.04

3.5 Software Description

3.5.1 Ubuntu

Ubuntu is a Linux-based operating system. It is designed for computers, smartphones, and network servers. The system is developed by a UK based company called Canonical Ltd. All the principles used to develop the Ubuntu software are based on the principles of Open Source software development. Following are some of the significant features of Ubuntu are

- The desktop version of Ubuntu supports all the normal software on Windows such as Firefox, Chrome, VLC, etc.
- It supports the office suite called LibreOffice.
- Ubuntu has an in-built email software called Thunderbird, which gives the user access to email such as Exchange, Gmail, Hotmail, etc.
- There are a host of free applications for users to view and edit photos.
- There are also applications to manage videos and it also allows the users to share video.
- It is easy to find content on Ubuntu with the smart searching facility.
- The best feature is, it is a free operating system and is backed by a huge open source community.
- Hardware auto-configuration :Ubuntu comes with the drivers for most hardware built-in.
- Software repositories: Being able to install thousands of applications from Ubuntu's repositories in a few clicks is a huge plus. Besides the fact that the software is free and more secure than .exe packages downloaded from random websites, it's much more convenient to install programs from a centralized location.
- Ssh client: Having an ssh client built into the operating system has major advantages.There are many ssh clients available for Windows, like Putty, but none are

available in Windows out-of-the-box, and even the best of them isn't as functional as the trusty gnome-terminal.

- Secure:With a built-in firewall and virus protection software, Ubuntu is one of the most secure operating systems around. And the long-term support releases give you five years of security patches and updates.
- Support and management:Ubuntu Advantage is the professional support package from the experts at Canonical. Get 24x7 support with access to engineers with first-hand experience of your issues. It includes Landscape, the systems management tool, for monitoring, managing, patching, and compliance reporting on all your Ubuntu desktops.

3.5.2 Python

Python is a high-end programming language because of dynamic features that functions based on code readability.Python is a high level, interpreted and general purpose dynamic programming language that focuses on code readability.It has fewer steps when compared to Java and C.It was founded in 1991 by developer Guido Van Rossum.It is used in many organizations as it supports multiple programming paradigms.It also performs automatic memory management. Python gives a complete room to increase productivity because of integrity and convenience. The robust control capabilities provide a way to increase the pace of applications quickly.

Easy integration is one of the top benefits of using Python because of absolute requirements. Python can easily connect with C, C++, Java, and so on. It also processes markup languages like XML because it helps modern applications to work efficiently.

Almost every Linux distribution comes with a version of Python included in the default system packages. If not found, python are be installed using the following steps;
Apt, or Advanced Package Tool is the default package manager that you will find on Ubuntu From which you can download the Python package from the official Ubuntu repository.

1. Open up your terminal by pressing Ctrl + Alt + T.
2. Update your local system's repository list by entering the following command:
3. **sudo apt-get update**
4. Download the latest version of Python:
5. **sudo apt-get install python**
6. Apt will automatically find the package and install it on your computer.

To check if Python is installed on your system or not, open up your terminal by pressing **Ctrl + Alt + T**. Type in "python" and press **Enter.**

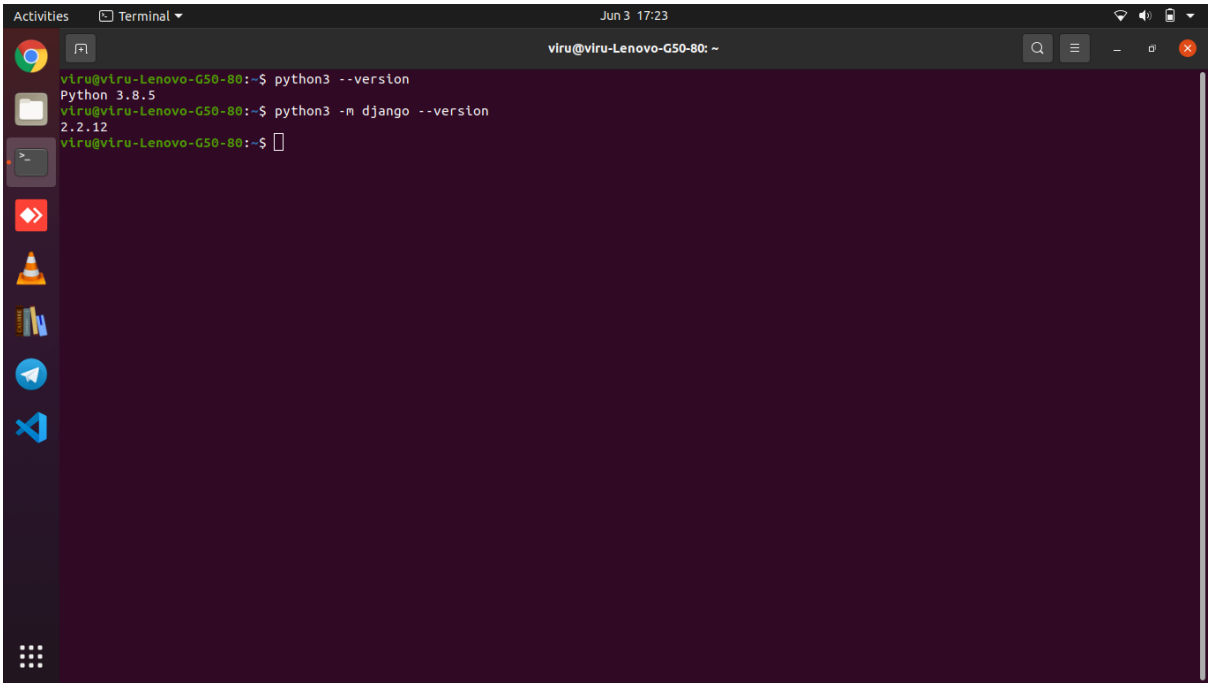


Fig 1 - Software Description

Installing text to speech and speech to text libraries for interactive voice response.

The required modules/ libraries are Gtts and Pyaudio.

Gtts: gTTS (Google Text-to-Speech), a Python library and CLI tool to interface with Google Translate's text-to-speech API.

Features:It is customizable speech-specific sentence tokenizer that allows for unlimited lengths of text to be read, all while keeping proper intonation, abbreviations, decimals and more.

It has customizable text pre-processors which can, for example, provide pronunciation corrections.

To install gtts on ubuntu,

sudo apt-get install -y

python-gtts

Pyaudio:PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms. PyAudio is inspired by:

1.pyPortAudio/fastaudio: Python bindings for PortAudio v18 API.

2.tkSnack: cross-platform sound toolkit for Tcl/Tk and Python.

To install pyaudio on ubuntu,

sudo apt install python3-pyaudio

3.5.3 Django Framework

Django is an MVT web framework that is used to build web applications. Django comes with common libraries which are essential to build common functionalities like URL routing, authentication, an object-relational mapper (ORM), a templating system and db-schema migrations. Django has an in-built administration interface which lets you handle your models, user/ group permissions and to manage users. With model interface in place, there is no need for a separate database administration program for all but advanced database functions.

Django is based on MVC design patterns. It means that all the entities like database, back-end and front-end code are individual entities. Django allows us to separate code from the static media including pictures, files, CSS and JavaScript that make up your site.

Django supports a full list of third-party libraries for web servers, caching, performance management, clustering and balancing. One of the advantages Django provides is the support for major email and messaging applications.

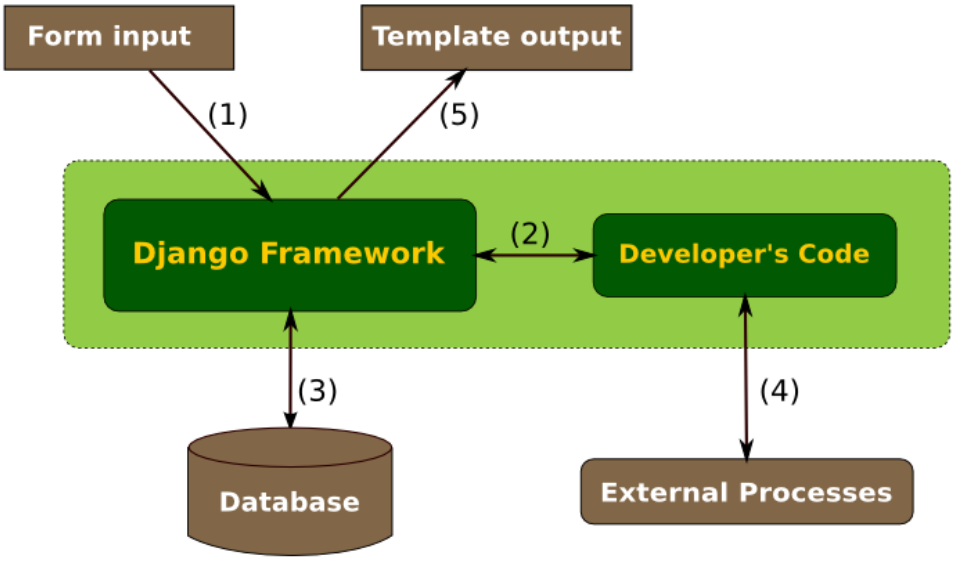


Fig 2 - Django Framework

The architecture works in the following way:

First, a user requests for a resource. Django considers the request as a URL and matches it with the existing URL paths in the urls.py file. This process of matching the user-requested URL to the one in urls.py is known as URL mapping. Once the URL matches, Django carries out the further process.

Once the URL is mapped, Django jumps to the views.py folder and calls a view. The triggered view looks for models and templates to interact with, and then it returns the response back to the user. Here, the model deals with the data associated with the user request. On the other hand, the template deals with the HTML and the static files, such as CSS files, JS files, images, etc., that are required to complete the view.

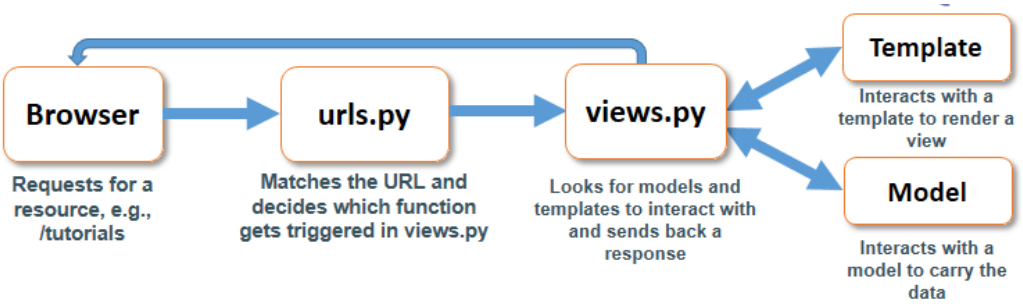


Fig 3 - Django Architecture

3.5.4 HTML, CSS and JavaScript

HTML is the language in which most websites are written. It is used to create pages and make them functional. HTML stands for Hyper Text Markup Language.

- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Cascading Style Sheets is a simple design language intended to simplify the process of making web pages presentable. It is one of the most widely used style languages over the web. Some of the advantages are:

- CSS saves time - You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- Pages load faster - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- Easy maintenance - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Superior styles to HTML - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Multiple Device Compatibility - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complementary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

- Javascript is the most popular programming language in the world and that makes it a programmer's great choice. Once you learn Javascript, it helps you develop great front-end as well as back-end softwares using different Javascript based frameworks like jQuery, Node.JS etc.
- Javascript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience.
- JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as a Javascript Programmer.
- Great thing about Javascript is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.

3.5.5 Visual Studio Code

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new tool sets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Azure DevOps client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge. The slogan for Visual Studio Community edition is "Free, fully-featured IDE for students, open-source and individual developers".

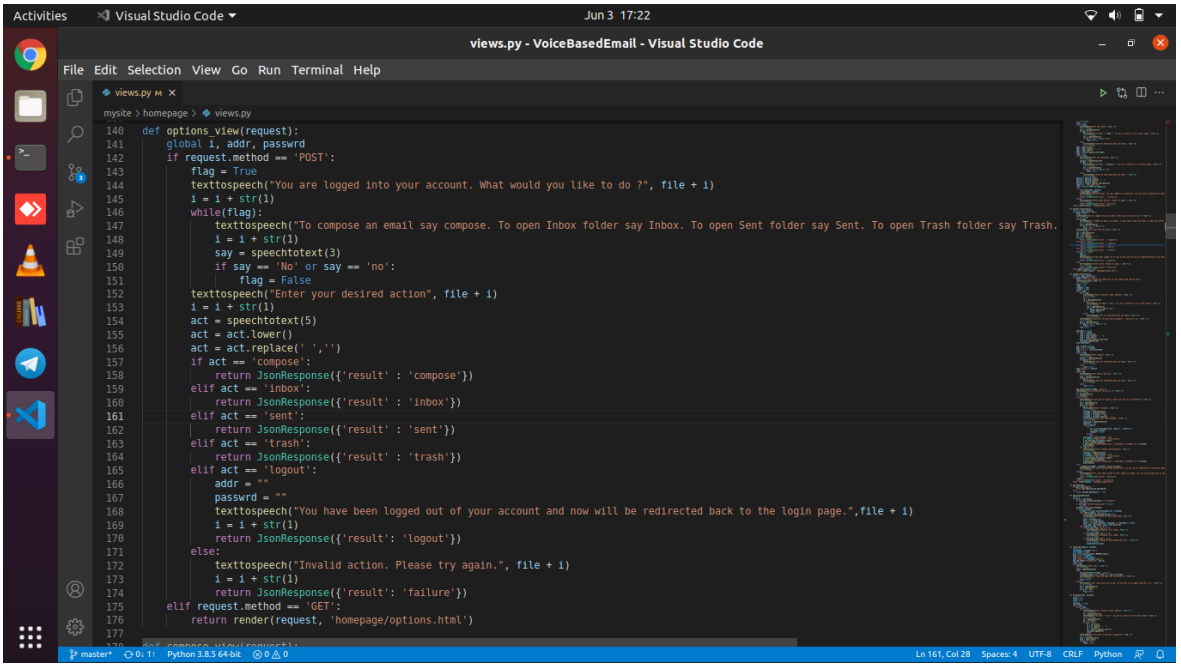

Activities
Visual Studio Code
Jun 3 17:22
views.py - VoiceBasedEmail - Visual Studio Code
File Edit Selection View Go Run Terminal Help
views.py M
mysite > homepage > views.py
def options_view(request):
global i, addr, passwrd
if request.method == 'POST':
flag = True
texttospeech("You are logged into your account. What would you like to do ?", file + i)
i = i + str(1)
while(flag):
texttospeech("To compose an email say compose. To open Inbox folder say Inbox. To open Sent folder say Sent. To open Trash folder say Trash.
i = i + str(1)
say = speechtotext(3)
if say == 'No' or say == 'no':
flag = False
texttospeech("Enter your desired action", file + i)
i = i + str(1)
act = speechtotext(5)
act = act.lower()
act = act.replace(' ','')
if act == 'compose':
return JsonResponse({'result' : 'compose'})
elif act == 'inbox':
return JsonResponse({'result' : 'inbox'})
elif act == 'sent':
return JsonResponse({'result' : 'sent'})
elif act == 'trash':
return JsonResponse({'result' : 'trash'})
elif act == 'logout':
addr = ""
passwrd = ""
texttospeech("You have been logged out of your account and now will be redirected back to the login page.",file + i)
i = i + str(1)
return JsonResponse({'result': 'logout'})
else:
texttospeech("Invalid action. Please try again.", file + i)
i = i + str(1)
return JsonResponse({'result': 'failure'})
elif request.method == 'GET':
return render(request, 'homepage/options.html')
master*
Python 3.8.5 64-bit
Ln 161, Col 28
Spaces: 4
UTF-8
CRLF
Python

Fig 4 - VS Code

4: SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

Here, we build a web based application following the django framework which is a high level python web framework, which enables the visually blind to access mails from recognised domains only with mouse clicks and voice commands. The voice recognition functionality is added with the help of google to text speech library that acts as an interface to the google Translate's text-to-speech API and also pyaudio module.

The visual layout of the application is coded in html,css and javascript with less preference as the functionality of the application is more prioritized than the appearance. The voice is recorded using an earphone consisting of a mic.

4.2 Existing System

The mail services that are available today are of no use to the people who are visually impaired. This is because these systems are not helpful to them in any way as it cannot provide any audio feedback to readout the contents for them. As they are unable to visualize things that are present on the screen, they find it difficult to perform operations such as performing mouse clicks specifically. The systems available nowadays use screen readers which read information displayed on desktop or it prints information on a Braille printer. ASR (Automatic speech recognizer) and TTS (text to speech) get used for converting speech to text and vice versa. These systems have the following drawbacks:

- screen readers basically read out content from the screen and in order to respond to it, they need to provide input through a keyboard. So, in order to accomplish this, the user needs to be aware of the positions of the keys on the keyboard. Hence, a person who has never made use of a computer will never be able to use such a system.
- In case of noisy environments, performance of ASR degrades.
- With the help of screen readers it is difficult for blind people to access E-mail systems and computers operating easily, because they have noisy audio interfaces.

4.3 Proposed System/Solution

Keeping in view all of these, the goal of the project is to reduce limitations and problems mentioned above, The goal of the project is to build a python-based web application for visually impaired individuals that uses speech to text and text to speech technology that allow visually impaired people to control their email accounts using only their voice and to read, send, delete, and conduct all of the necessary functions as needed.The system also directly sends recorded voice messages to recipients' mail address as an attachment and operates multimedia functions of the computer such as audio, text.The web-based application employs interactive voice response (IVR), allowing the visually handicapped to control their environment.

The fundamental advantage of this system is that it eliminates the usage of a keyboard; the user will only rely on voice commands and a single mouse click.

- The system will start its functionality based on single left-clicks anywhere on the screen, regardless of where the cursor is placed on the screen.
- It also does away with the traditional methods of employing screen readers, which necessitates users remembering keyboard shortcuts, adding to the cognitive stress on visually challenged people.

5: SYSTEM DESIGN

5.1 Introduction

The framework configuration develops a general structure building outline. Programming diagrams incorporate addressing the item system in a shape that might be changed into at least one anticipated. The essential demonstrated by the end customer must be placed in a systematic way. Diagram is a creative system; an extraordinary design is the best approach to reasonable structure. The structure "Layout" is portrayed as "The methodology of applying distinctive frameworks and guidelines with the ultimate objective of describing a strategy or a system in sufficient purpose important to permit its physical affirmation". Diverse design segments are taken after to add to the system. The design detail depicts the segments of the system, the sections or segments of the structure and their appearance to end-customers.

5.2 Design Consideration

The explanation behind the plan is to orchestrate the course of action of the issue dictated by the necessities report. This stage is the underlying stage in moving from issue to the game plan space. All things considered, start with what is obliged; diagram takes us to work towards how to satisfy those necessities. The design of the system is perhaps the most essential segment affecting the way of the item and note-worthily affects the later stages, particularly testing and upkeep. System diagram delineates all the huge data structure, report game plan, yield and genuine modules in the system and their Specification is picked.

5.3 System Architecture

The architectural configuration procedure is concerned with building up a fundamental basic system for a framework. It includes recognizing the real parts of the framework and interchanges between these segments.

The beginning configuration procedure of recognizing these subsystems and building up a structure for subsystem control and correspondence is called construction modeling outline and the yield of this outline procedure is a portrayal of the product structural planning. The proposed architecture for this system is given below.

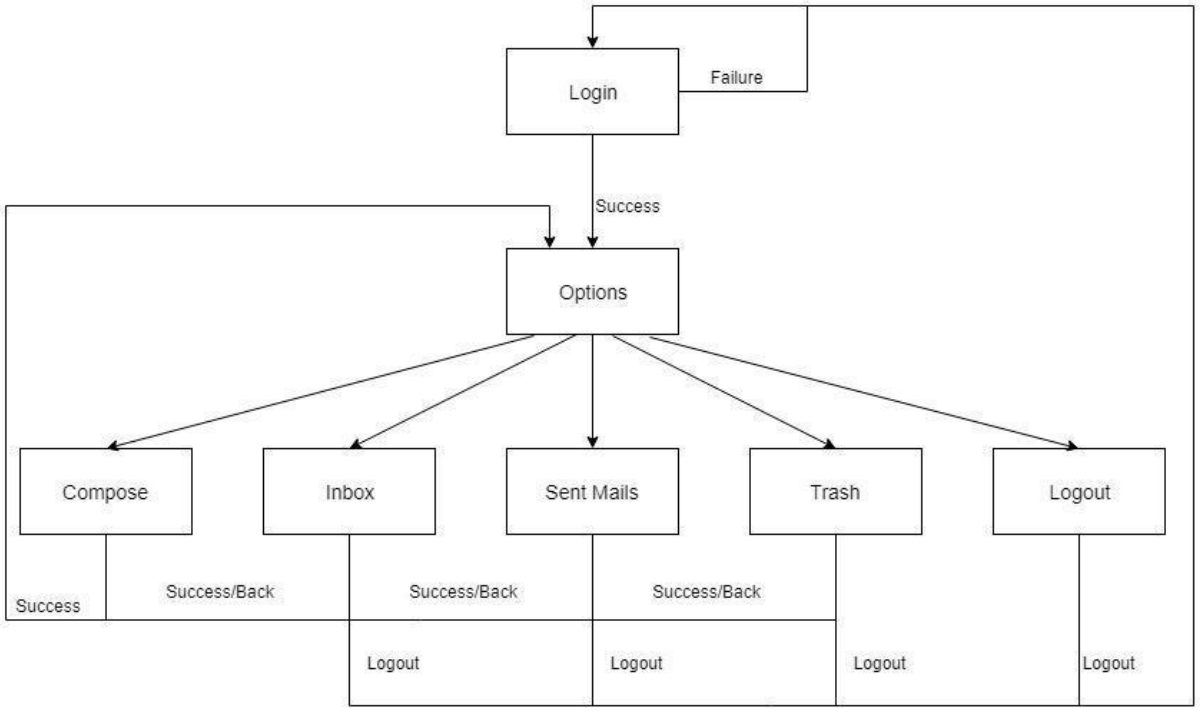


Fig 5 - System Architecture

Data Flow Diagram

The DFD is straightforward graphical formalism that can be utilized to speak to a framework as far as the info information to the framework, different preparation did on this information and the yield information created by the framework. A DFD model uses an exceptionally predetermined number of primitive images to speak to the capacities performed by a framework and the information stream among the capacities.

The principal motivation behind why the DFD method is so famous is most likely in light of the way that DFD is an exceptionally basic formalism- It is easy to comprehend and utilization. Beginning with the arrangement of abnormal state works that a framework performs, a DFD display progressively speaks to different sub capacities. Actually, any various leveled models are easy to get.

The human personality is such that it can without much of a stretch see any progressive model of a framework in light of the fact that in a various leveled model, beginning with an extremely straightforward and unique model of framework, distinctive points of interest of a framework are gradually presented through the diverse orders. A data-flow diagram (DFD) is a graphical representation of the "stream" of information through a data framework. DFDs can likewise be utilized for the perception of information handling.

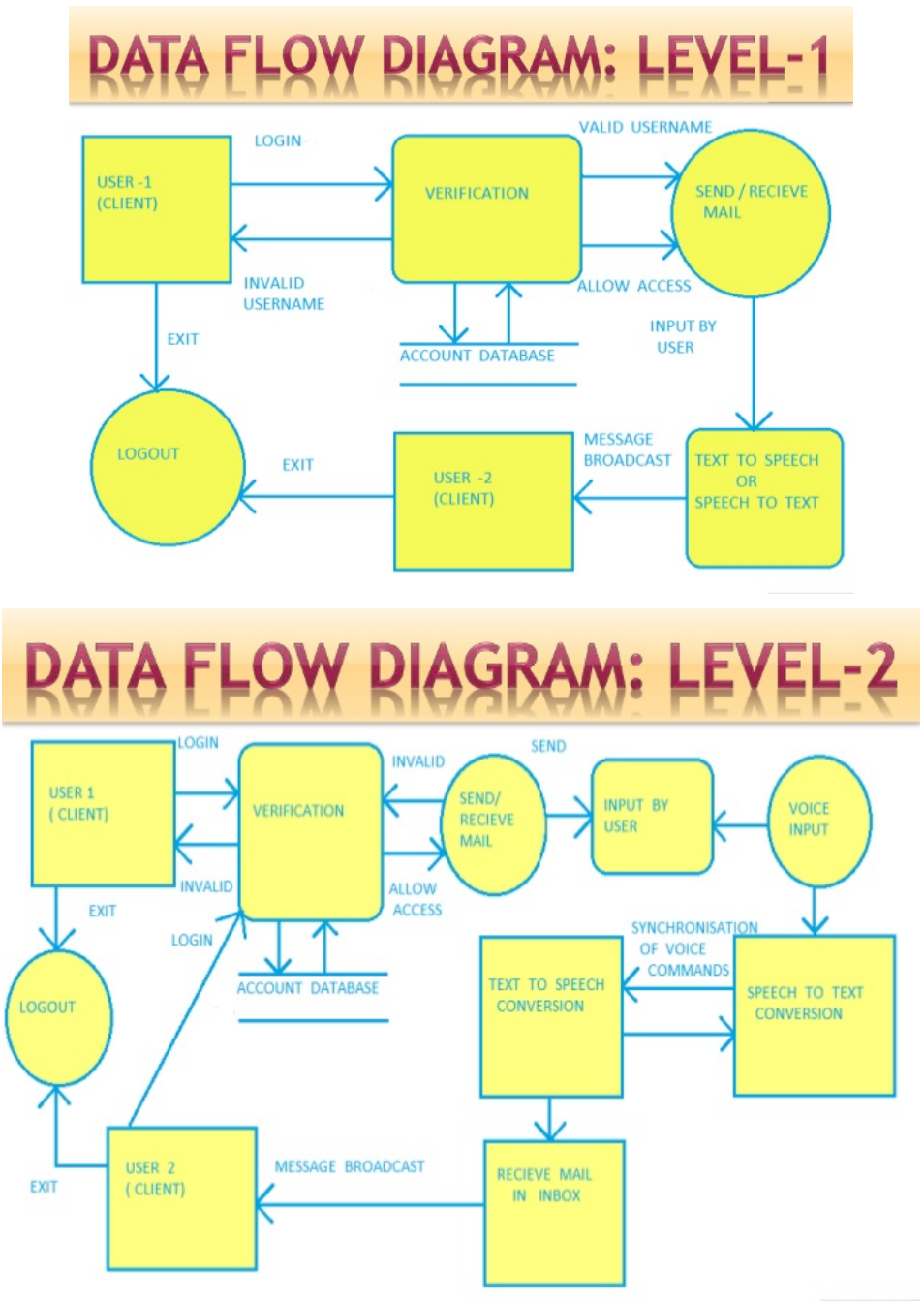


Fig 6 - Data Flow Diagrams

Use Case Diagram

A use case chart is a kind of behavioral graph made from a Use-case examination. Its object is to present a graphical diagram of the usefulness given by a framework regarding performers, their objectives (spoken to as utilization cases), and any conditions between those utilization cases. Use case charts give us the data about how clients and utilization cases are connected with the framework. Use cases are used amid prerequisites elicitation and examination to speak to the usefulness of the framework. Use cases concentrate on the conduct of the framework from an outside perspective.

A use case depicts a capacity given by a framework that yields an obvious result for a performer. A performing artist portrays any element that collaborates with the system. The performers are outside the limit of the framework, while the use cases are inside the limit of the framework. On-screen characters are spoken to with stick figures, use cases with ovals, and the limit of the framework with a container encasing the use cases.

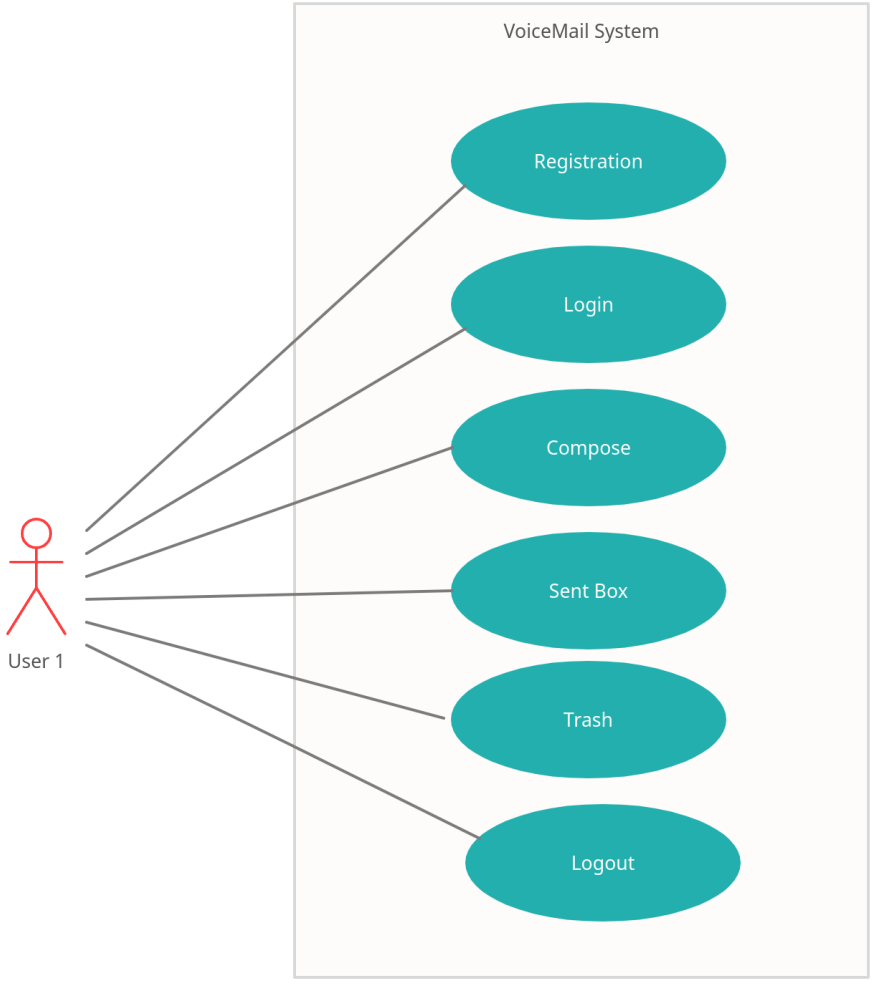


Fig 7 - Use case Diagram

Flow Chart Diagram

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

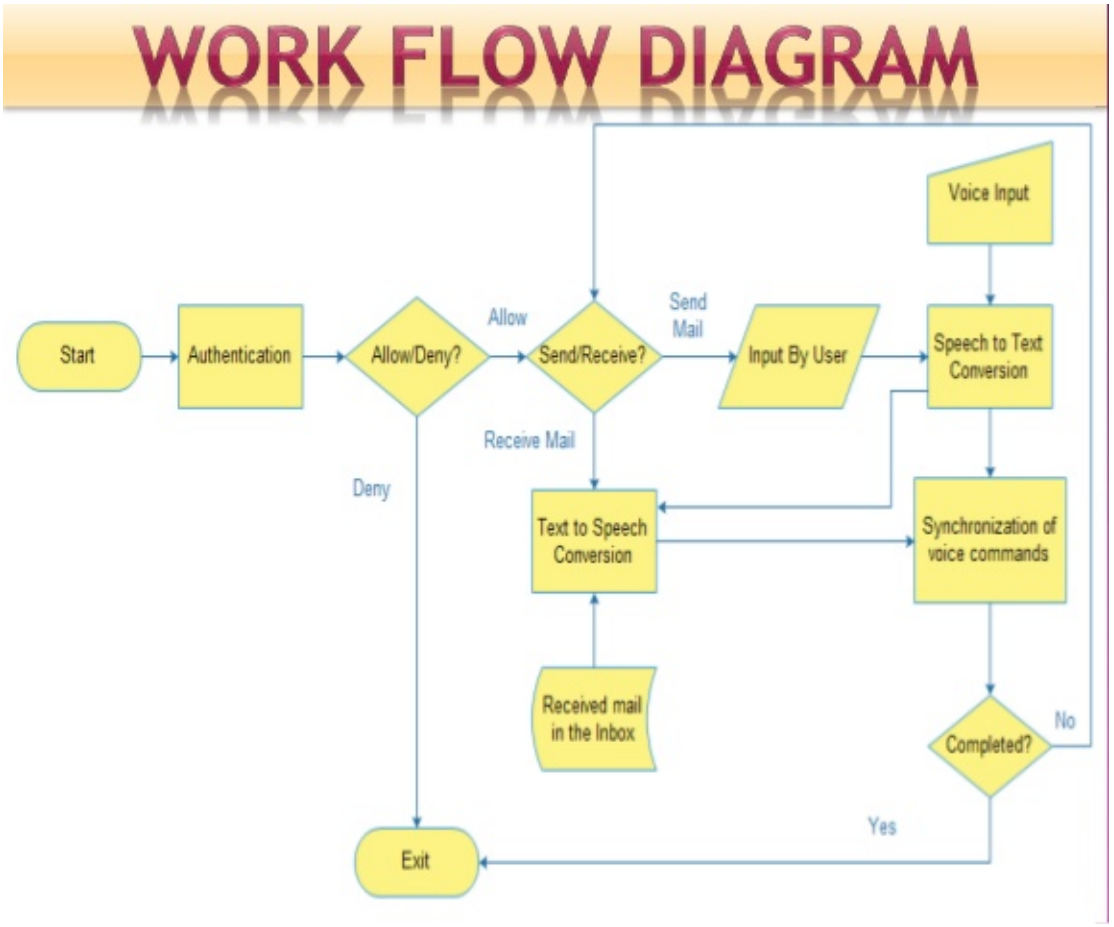


Fig 8 - Work Flow Diagram

Sequence Diagram

A sequence diagram is a system is an interaction diagram that shows how processes operate with one and other and in what order. It's a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are sometimes called event diagrams or event scenarios.

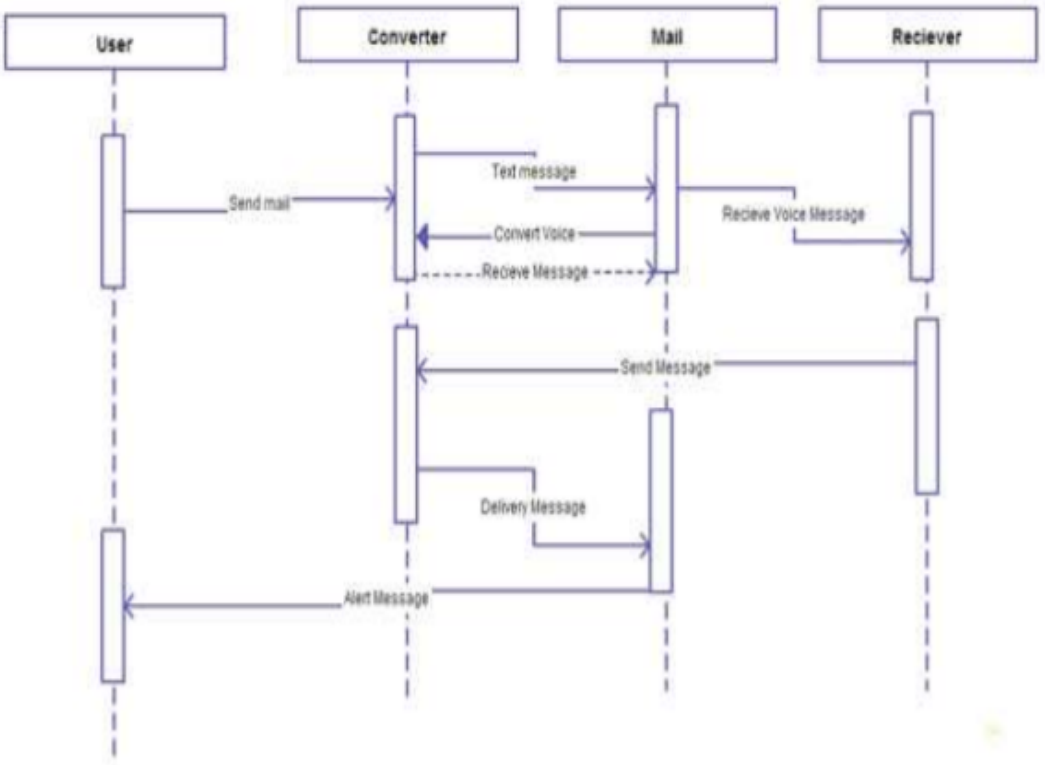


Fig 9 - Sequence Diagram

6: IMPLEMENTATION

6.1 Introduction

This stage is the underlying stage in moving from issue to the course of action space. Accordingly, starting with what is obliged; the diagram takes us to work towards how to fully fill those requirements. System plot portrays all the critical data structure, record course of action, yield and genuine modules in the structure and their Specification is picked.
The program will function in the following way:

1. When the application is launched, the user can log in.
2. The user is prompted to enter their username and password on the login screen.
3. After a successful login, the menu page opens, prompting the user for input instructions such as Inbox, Sent, Compose, Trash, and Logout. The user must say one of the orders, after which he will be led depending on the voice commands.
4. On clicking Compose, the user will be sent to the Compose Page, where they will be asked to enter the recipient's ID, topic, and email message. The email is successfully delivered to the recipient's ID.
5. The user will be led to the received mails page after tapping on the screen, and will be asked if they want to view the first, second, or third message.

6.2 Module Description

1. **LOGIN:** This module will prompt the user for their account and password. In discourse, this will be acceptable. The speech will be converted to text, and the user will be prompted to authenticate whether or not the information is correct. The dashboard will be displayed to the user after a successful login.
2. **COMPOSE:** To create a new email, select this option. The user will just have to utilise his voice to provide input. There will be no need for keyboard shortcuts or textual input. The system will prompt the user to tap anywhere on the screen and voice out the recipient's mail address, topic, and mail body. After the input of all the content, the system will read it aloud so the user can verify that it is right. Following completion of all needed inputs, the system will request the user to say "yes" to confirm and send the email.
3. **INBOX:** This inbox will be displayed when the user says "Inbox." The system will notify the user of every new message received and will read them out one by one.
4. **SENT MAIL:** This component keeps track of the emails sent by a certain user. If the user wishes to examine the emails he has sent thus far, he may do so by selecting the “Sent Mails” option from the dashboard.
5. **TRASH:** This option will retain track of all the emails that the user has deleted.
6. **LOGOUT:** Allows the user to log out of the app and returns them to the login screen.

7: PSEUDO CODE

Pseudo Code uses the structural conventions of a normal programming language but the intention is for human reading rather than machine reading. Omission of details that are essential for machine understanding of the algorithm such as variable declarations, specific system code and some sub-routines is pseudocode. Augmentation of programming language is done with natural language description details where convenient or with compact mathematical notation. The purpose of using pseudocode is that it is easier for people to understand than conventional programming language code and that the key principles of an algorithm are efficiently and the environment independently described. Pseudocode is commonly used in textbooks and scientific publications that are documenting various algorithms and also in the plan of computer program development, for sketching the structure of the program before the actual coding takes place.

```
def login_view(request):
    global i, addr, passwrd
    if request.method == 'POST':
        text1 = "Welcome to our Voice Based Email Application. Login
with your email account to continue. "
        texttospeech(text1, file + i)
        i = i + str(1)
        flag = True
        while (flag):
            texttospeech("Enter your Email", file + i)
            i = i + str(1)
            addr = speechtotext(10)
            if addr != 'N':
                texttospeech("You meant " + addr + " say yes to
confirm or no to enter again", file + i)
                i = i + str(1)
                say = speechtotext(3)
                if say == 'yes' or say == 'Yes':
                    flag = False
            else:
                texttospeech("could not understand what you meant:",
file + i)
                i = i + str(1)
        addr = addr.strip()
        addr = addr.replace(' ', '')
        addr = addr.lower()
        addr = convert_special_char(addr)
        flag = True
        while (flag):
            texttospeech("Enter your password", file + i)
```

```
            i = i + str(1)
            passwrd = speechtotext(10)
            if addr != 'N':
                texttospeech("You meant " + passwrd + " say yes to
confirm or no to enter again", file + i)
                i = i + str(1)
                say = speechtotext(3)
                if say == 'yes' or say == 'Yes':
                    flag = False
            else:
                texttospeech("could not understand what you meant:",
file + i)
                i = i + str(1)
        passwrd = passwrd.strip()
        passwrd = passwrd.replace(' ', '')
        passwrd = passwrd.lower()
        passwrd = convert_special_char(passwrd)
        imap_url = 'imap.gmail.com'
        conn = imaplib.IMAP4_SSL(imap_url)
        try:
            conn.login(addr, passwrd)
            s.login(addr, passwrd)
            texttospeech("Congratulations. You have logged in
successfully. You will now be redirected to options page.", file + i)
            i = i + str(1)
            return JsonResponse({'result' : 'success'})
        except:
            texttospeech("Invalid Login Details. Please try again.",
file + i)
            i = i + str(1)
            return JsonResponse({'result': 'failure'})
    return render(request, 'homepage/home.html')

def options_view(request):
    global i, addr, passwrd
    if request.method == 'POST':
        flag = True
        texttospeech("You are logged into your account. What would you
like to do ?", file + i)
        i = i + str(1)
        while(flag):
            texttospeech("To compose an email say compose. To open
Inbox folder say Inbox. To open Sent folder say Sent. To open Trash
folder say Trash. To Logout say Logout. Do you want me to repeat?",
file + i)
            i = i + str(1)
```

```
            say = speechtotext(3)
            if say == 'No' or say == 'no':
                flag = False
        texttospeech("Enter your desired action", file + i)
        i = i + str(1)
        act = speechtotext(5)
        act = act.lower()
        act = act.replace(' ','')
        if act == 'compose':
            return JsonResponse({'result' : 'compose'})
        elif act == 'inbox':
            return JsonResponse({'result' : 'inbox'})
        elif act == 'sent':
            return JsonResponse({'result' : 'sent'})
        elif act == 'trash':
            return JsonResponse({'result' : 'trash'})
        elif act == 'logout':
            addr = ""
            passwrd = ""
            texttospeech("You have been logged out of your account and
now will be redirected back to the login page.",file + i)
            i = i + str(1)
            return JsonResponse({'result': 'logout'})
        else:
            texttospeech("Invalid action. Please try again.", file +
i)
            i = i + str(1)
            return JsonResponse({'result': 'failure'})
    elif request.method == 'GET':
        return render(request, 'homepage/options.html')
```

8: TESTING

Testing of any product consists of giving the product an arrangement of test information and watching if the product carries on, not surprisingly, if the product neglects to carry on obviously, then the conditions under which disappointment happens are noted for investigation and amendment. At last the framework in general is tried to guarantee that blunders in past countenances are revealed and the venture acts as determined.

8.1 Basics of Software Testing

8.1.1 Black Box Testing

Black box testing is done to find the following

- Incorrect or missing functions
- Interface errors
- Errors on external database access
- Performance error
- Initialization and termination error

8.1.2 White Box Testing

This allows the tests to

- Check whether all independent paths within a module have been exercised at least once
- Exercise all logical decisions on their false sides
- Execute all loops and their boundaries and within their boundaries
- Exercise the internal data structure to ensure their validity
- Ensure whether all possible validity checks and validity lookups have been provided to validate data entry.

8.2 Types of Testing

Following are the different types of testing

- Unit Testing
- Integration Testing
- System Testing
- Performance Testing
- Validation Testing
- Acceptance Testing

Let us consider each testing and discuss it in detail. Firstly we move to the first testing and give its detailed description.

8.2.1 Unit Testing

Singular parts are tried to guarantee that they work accurately. Every part is tried freely, without another framework segment. This framework was tried with the arrangement of legitimate test information for every module and the outcomes were checked with the normal yield. Unit testing centers around confirmation exertion on the littlest unit of the product outline module. This is otherwise called MODULE TESTING. This testing is done amid stages, every module is observed to work agreeable as respects to the normal yield from the module.

8.2.2 Integration Testing

Mix testing is another part of testing that is for the most part done keeping in mind the end goal to reveal mistakes related to the stream of information crosswise over interfaces. The unit-tried modules are assembled together and tried in little sections, which make it less demanding to seclude and revise mistakes. This approach proceeded with unit I have coordinated all modules to frame the framework all in all.

8.2.3 System Testing

Framework testing is really a progression of various tests whose basic role is to completely practice the PC based framework. Framework testing guarantees that the whole incorporated programming framework meets prerequisites. It tests a design to guarantee known and unsurprising outcomes. A case of framework testing is the setup arranged framework mix testing. Framework testing depends on process depiction and streams, underscoring pre-driver process and incorporation focuses.

8.2.4 Performance Testing

The execution testing guarantees that the yield being delivered inside as far as possible and time taken for the framework aggregating, offering reaction to the clients and demand being sent to the framework so as to recover the outcomes.

8.2.5 Validation Testing

The approval testing can be characterized from multiple points of view, however a straightforward definition is that. Approval succeeds when the product capacities in a way that can be sensibly expected by the end client.

8.2.6 Acceptance Testing

This is the last phase of testing procedure before the framework is acknowledged for operational utilization. The framework is tried inside the information provided from the framework procurer instead of recreated information.

8.3 Unit Testing Performed

Table 1 : Unit Test Case 1

Test Case	UTC-*1
Name of Test	Login with correct credentials
Expected Result	Successful login and redirect
Actual Result	Same as expected
Remarks	Successful

Table 2 : Unit Test Case 2

Test Case	UTC-*2
Name of Test	Login with incorrect credentials
Expected Result	Error message and prompt to input again
Actual Result	Same as expected
Remarks	Successful

Table 3 : Unit Test Case 3

Test Case	UTC-*3
Name of Test	Compose mail
Expected Result	Compose mail and send to one/many recipients
Actual Result	Same as expected
Remarks	Successful

Table 4 : Unit Test Case 4

Test Case	UTC-*4
Name of Test	Open unread mails
Expected Result	Unread mails are opened and narrated
Actual Result	Same as expected
Remarks	Successful

Table 5 : Unit Test Case 5

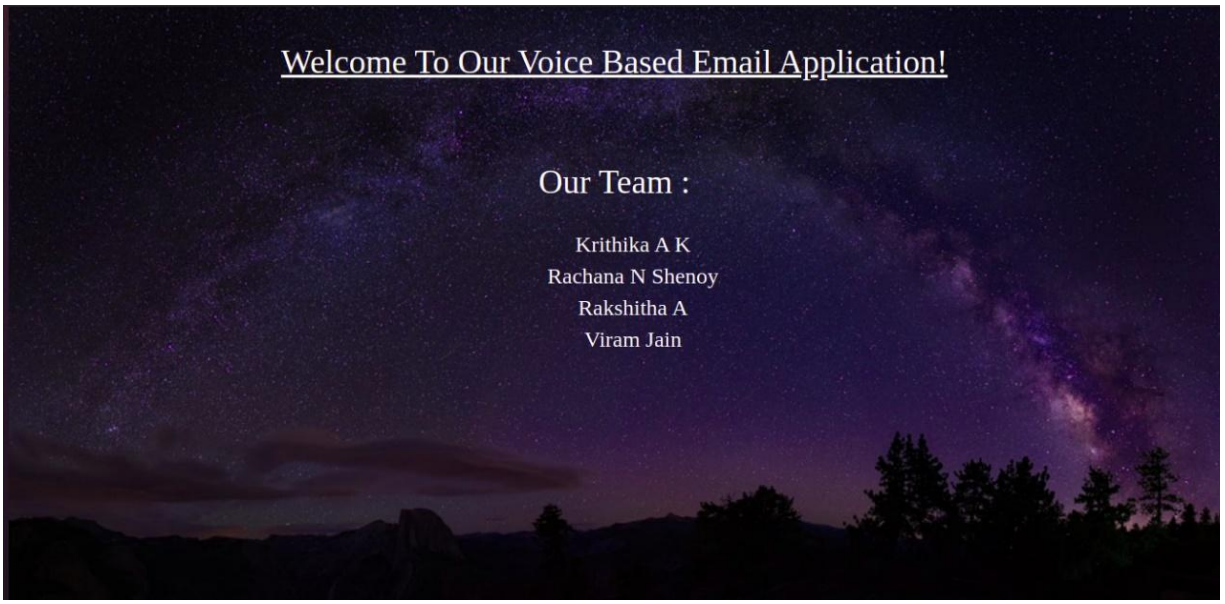
Test Case	UTC-*5
Name of Test	Logout
Expected Result	Successful logout and redirect to home page
Actual Result	Same as expected
Remarks	Successful

9: RESULTS

9.1 Expected Outcomes

- Successful/unsuccessful logins with appropriate messages.
- Options to compose, read messages, open drafts, open trash or delete mails.

9.2 Output Screenshots



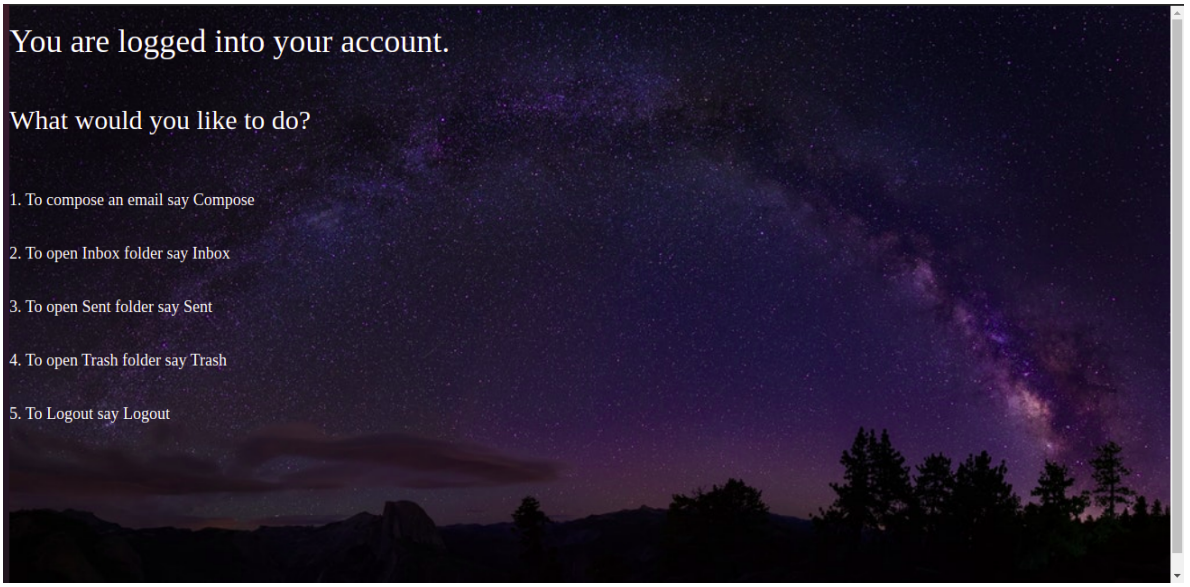


Fig 10 - Output Screenshots

10: CONCLUSION AND FUTURE SCOPE

10.1 Conclusion

This method will aid in overcoming the difficulties that visually challenged persons previously had in accessing e-mail. The usage of keyboard shortcuts with screen readers has been eliminated, which helps lessen the cognitive burden of knowledgeable keyboard shortcuts. This e-mail system is simple to use and may be utilised by users of all ages. The system has the feature of speech to text as well as text to speech which makes the designed system to be handled by the visually impaired person as well as illiterate people. Also, the success of this project might also encourage other developers to build a system more useful for visually impaired or illiterate people, who also deserve an equal stand in society.

10.2 Future Scope

There might be a variety of enhancements in the future, such as the ability to include attachments such as photographs, documents, audio, and video files. The username and password that are supplied during login can be protected using encryption and decryption algorithms. More operations, such as search, flag important, delete, archive, go back, report spam, and forward, can be added. It's also possible to add automated email replies. The programme may be customised in a variety of native languages, allowing it to be used by a wide range of individuals.

11: REFERENCES

[1] J. Nilesh, P. Alai, C. Swapnil, and Bendre M.R., “Voice Based System in Desktop and Mobile Devices for Blind People”. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.638.762&rep=rep1&type=pdf.

[2] K. Jayachandran and P. Anbumani, “Voice Based Email for Blind People”. [Online]. Available: https://www.ijariit.com/manuscripts/v3i3/V3I3-1462.pdf.

[3] M. Badigar, N. Dias, J. Dias, and M. Pinto, “Voice Based Email Application for Visually Impaired”. [Online]. Available: http://www.ijste.org/articles/IJSTEV4I12068.pdf.

[4] P. Ingle, H. Kanade, and A. Lanke, “Voice based e-mail System for Blinds”. [Online]. Available: https://www.arcjournals.org/pdfs/ijrscse/v3-i1/5.pdf.

[5] T.Shabana, A.Anam, A.Rafiya, and K.Aisha, “Voice based email system for blinds”. [Online]. Available: https://ijarcce.com/wp-content/uploads/2015/02/IJARCCE5C.pdf.

[6] T. Hossaina, Y. Akterb, and A. Rahmanc, “Voice mail application for visually impaired persons”. [Online]. Available: https://www.researchgate.net/publication/342835969_Voice_mail_application_for_visually_impaired_persons.

[7] R. Rastogi, A. Rajput, Archana, and Komal, “An Application of Voice Mail: Email Services for the Visually Challenged Individual”. [Online]. Available: https://zenodo.org/record/2683677#.YDFUyfszbEy.

[8] C. Belinda Mary, Rupavathy.N, and M. R N, “A voice-based text mail system for visually impaired”. [Online]. Available: https://www.researchgate.net/publication/324526106_A_voice_based_text_mail_system_for_visually_impaired.

[9] A. Verma, “Voice based Electronic Mail System for Visually Challenged Individuals”. [Online]. Available: https://www.ijitee.org/wp-content/uploads/papers/v8i12S/L103010812S19.pdf.