Voice Based Email System for the Visually Challenged

Satya Prakash[1*], Kartikey Agrawal[2], Siddharth Dosaj[3], Shatakshi Singh[4]

[1,2,3,4]*Student, Department of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University, Noida, India*

***Abstract*: In today's digital world, email is a crucial form of communication, but it can be difficult for those who are blind or visually impaired to access and use it. The idea of a voice-based email system created exclusively for those with visual impairments is presented in this abstract. The objective is to provide a welcoming atmosphere that enables people who are blind or visually impaired to use computers on their own to send and receive emails. The suggested solution has cutting-edge components designed with visually impaired users in mind. By integrating a screen reader, users may convert text into voice and interact with the email interface aloud. Users may create emails using their voice thanks to reliable speech recognition technology that properly converts spoken words into written text. To continually develop the system and adapt it to unique tastes and needs, user input and customization options are included.**

***Keywords*: Voice Recognition, Text-to-Speech (TTS), Speech-to-Text (STT), Visually challenged people.**

1. Introduction

Email has become a vital tool for communication and information sharing in today's digitally linked society. Individuals with vision impairments, on the other hand, confront distinct obstacles in accessing and utilising this critical technology. The creation of a voice-based email system particularly built for visually impaired people attempts to bridge this accessibility gap and enable them to send and receive emails on their own computers.

To access digital material, the visually impaired population has long relied on assistive technology such as screen readers and text-to-speech synthesis. While these technologies have been quite helpful, they frequently require external assistance or other software to fully interface with email systems. We can give visually impaired people with a more inclusive and efficient method of communication by developing a dedicated voice-based email system.

The voice-based email system's major goal is to enable visually impaired users to engage with email interfaces via voice commands and speech-to-text conversion. The system accurately transcribes spoken words into written text by using powerful voice recognition technology, allowing users to create, edit, and manage emails using their natural voice.

Furthermore, the system uses natural language processing to decode spoken commands, allowing for smooth navigation across the email interface. Visually impaired users may easily execute operations such as checking their inbox, creating new messages, replying to emails, and organising folders by comprehending and reacting to audio instructions.

Overall, the voice-based email system marks a substantial leap in accessibility technology, allowing visually impaired people to access and use email independently. This system provides new channels for communication and productivity for the visually impaired community by using speech recognition, natural language processing, and other cutting-edge technology.

A. Overview

The creation of this system intends to promote a sense of community among people who are blind or visually impaired by giving them access to inclusive and accessible technologies. It encourages independence and fair participation in the digital world by making it simple for people of all ages to utilise the email system.

This article's main goal is to provide a voice-based emailing system that caters to the demands of blind and illiterate people and enables them to use commonplace technology like email and internet access. The suggested solution eliminates the requirement for conventional text-based input techniques by allowing blind people to log in using voice instructions.

B. Project's Scope

The goal of the speech-based email system project is to create a complete system that especially addresses the needs of people who are blind or visually impaired and enables them to send and receive emails using their voice. To provide a welcoming and accessible environment for users who are visually impaired, the project considers several important factors. The project's primary objective is to create a voice-based email system. The technology used for this project may, however, be expanded in the future to support other services like texting and the use of voice commands to control other programmes. The ultimate objective is to provide a complete and usable solution that improves the productivity and independence of visually impaired people while communicating via email.

C. Objective

The objective of the study is to offer users a simple, effective method of using voice commands to create and send emails. Users can dictate email content instead of manually typing it

[*]Corresponding author: satya.prakash.exe@gmail.com

thanks to an application that converts spoken words into text using speech recognition technology. The goal is to improve productivity and accessibility by enabling hands-free email composition. By including functions like message confirmation, login authentication and error handling the application also strives to offer a smooth user experience. This study compares and assesses various speech-to-text tools in various real-world settings, including the built-in tool in Android Studio, Google Cloud speech to text, and Microsoft Azure speech to text. The project aims to enhance user experience by addressing challenges like noise, accents, multilingualism, and accuracy.

2. Literature Survey

The solution presented in the study [3] is based on a voice command mechanism, in contrast to the current email system. In essence, text to number conversion is the foundation of the entire system. The system will prompt the user for voice commands once it is started in order to access the required services. It is crucial to note that the user must want to access the relevant services for this command to work. This programme makes use of IMAP (Internet Message Access Protocol). When sending emails from mail servers over TCP/IP, email clients frequently use this Internet protocol. The principal activity type, the screen, will be the first one to appear at the beginning of the year. The user only needs to tap one button on this screen for the device to start responding to voice instructions. There is just one full-size button, and you may tap it anywhere on the screen. The user can then send an email and use voice commands to read it.

Saurabh Sawant et al. offer a solution in their study [21] for those who are visually impaired or illiterate to increase their involvement with email systems. Screen readers and Braille keyboards used in IVR systems are no longer required thanks to this solution. There, both speech-to-text and text-to-speech conversions have been used. Other operations also make use of voice commands. To register, enter your email address and password. Use PHP mailer, a PHP feature, for the functionality. It is a library that enables email sending. so that the user's mail can be retrieved from the server of IMAP. For searching mail in inboxes, the Knuth-Morris-Pratt algorithm is utilised in this instance. As a conclusion, the entire system environment is voice-driven, and each level receives the proper system response. The drawback of this approach is that we can't use other email providers, such as Yahoo, etc., because it requires Gmail as a host server.

The [6] approach offers a stronger emphasis on user friendliness for all kinds of users, including regular users who are blind or illiterate, as compared to the current system, which lays more emphasis on user friendliness for regular users. The core of the entire system is IVR, or interactive voice response. When using this system, the computer will request the user to complete specific actions in order to access the associated services, and the user must complete those actions in order to access the related services. One of the biggest advantages of this method is that the keyboard is not required. All actions will be based on mouse click events. How blind individuals will be able to find the mouse pointer is now a question that needs to be answered.

The creation of a search engine is the aim of this paper [10] which only permits voice-based man and machines interaction. This was the debut of a ground-breaking page reader and search engine that is driven entirely by voice. It enables end-users to control and surf the web using user speech to navigate. In response to user text requests, current search engines secure relevant records from the server and display them as text. The paper's authors [7] suggested a user-friendly email system for individuals who are blind or visually handicapped. TTS (text-to-speech) module, STT (speech-to-text) module, as well as Module for mail composition activities (which includes composing, inbox, and sending) module makes up the system design. This system uses speech-to-text functionality. using an API to implement artificial intelligence utilising Google-provided neural network models Speech-to-text in the cloud for developers.

[8] employ artificial intelligence to help the blind utilise cutting-edge technology for their development and advancement. The suggested system is a desktop programme that uses artificial intelligence to reduce costs and make maintenance simple. The voice detection and conversion are utilised by the suggested system. Since it is entirely voice based and uses neither the mouse nor keyboard, it eliminates the drawbacks of the previous system. Because it uses voices, it offers an intuitive, interactive, and user-friendly GUI that even blind users who are not computer literate can use.

[19] talks about a straightforward programme with text-to-speech capabilities. The application is split into two main modules: the main application module, which contains the fundamental GUI elements and manages the application's fundamental activities, such as parameter input for conversion through file, direct keyboard input, or web. Both DJNativeSwing and the open source SWT API would be used in this. The major conversion engine of the second module, which is integrated into the main module, is responsible for accepting data and converting it. This would put into practise the free TTS API.

A study that was conducted resulted in the creation of a programme that might assist users in sending and receiving mail in English language. It was found during this investigation that the proposed architecture outperformed the current architecture. In order to make it simple for blind people to access information, text-to-speech and speech-to-text conversion techniques were employed. [11]

A. Advantage of the Surveyed Techniques

Many articles demonstrate how the text-to-speech and the speech-to-text conversion processes make it easier and even participatory for people who are blind or visually impaired. People with disabilities feel like regular users thanks to this system. Voice-based technology is also helpful for those who are illiterate or disabled. Automatic speech recognition is one of the main benefits. We can observe a decrease in the mental effort required by visually handicapped people to remember and write characters on a keyboard. A user-friendly system is the speech driven email system.

B. Limitations of the Surveyed Techniques

It is evident that mouse clicks are used for several activities in almost all the articles. It becomes more challenging for those who are blind. Additionally, because there are many languages spoken there that speech recognition software cannot grasp, the subcontinents of India do not benefit from this. English is the preferred language in its entirety.

The proposed system for those who are blind is described in the section that follows.

3. Methodology

There are numerous crucial elements in the development process for an Android voice-based email system.

First, in-depth requirements collecting is done in order to comprehend the unique requirements and difficulties experienced by visually impaired people using Android email. In order to learn more about their preferences, ideal features, and pain spots, user research techniques such as interviews and questionnaires are used.

The planning step for the design and interface starts once the requirements have been acquired. In order to ensure ease of use for visually impaired people, a user-friendly and accessible interface is built, taking into mind high contrast colours, big buttons, and intuitive navigation. The Android application's speech recognition system will be integrated as the following phase. To reliably translate spoken words into written text, this calls for the incorporation of a trustworthy voice recognition library or API, such as Google Cloud voice-to-Text or Pocket Sphinx. The Android application incorporates accessibility features that follow Android accessibility rules. This includes features that improve usability and accessibility for those who are blind or visually impaired, such as high contrast mode, haptic feedback, keyboard navigation, and gesture-based controls. To enable communication with email servers, integration with an email API or protocols like IMAP or SMTP is essential. As a result, users who are blind may use the programme to send, receive, and manage emails.

User data is protected by security and privacy safeguards. To prevent unauthorised access to email accounts, encryption technologies are included for data transmission and storage. Voice-based user authentication is also available. In order to discover and fix any usability or accessibility concerns, the produced system goes through extensive testing, including usability testing with users who are visually impaired. The design and functionality are iterated upon in response to user and accessibility expert feedback, enhancing the overall user experience.

Android's speech-to-text conversion: To receive user voice, Android supports a range of interfaces for speech recognition listeners. The recognizer intent comes first. Before returning to the previous activity, it first accepts the user's speech as input. The ten available languages on the platform can be selected by the Android voice recognizer when speaking input. The verbal response must then be captured while performing another or ongoing task. The response is converted into text by the code, which is then either displayed or transmitted once more as input to the text-to-speech converter.

Android's text-to-speech conversion: This is a crucial component of the program. It examines the text, creates an audio version of it, and then plays it through the user's microphone. This text to speech capability was created by Android and is supported, especially for persons who are blind or visually handicapped. To obtain the text, a class object is constructed. Many languages are supported by Android TT's engine. The function that converts text to speech receives the text as an argument. In the listener function, the transformed text is transmitted to the user as an internal voice.

Mail Programming Module: Email appears to be one of the most useful services now offered online. Many internet-based services use the SMTP protocol to transfer mail from one user to another. SMTP is a sending protocol used to send mail, whereas POP (post office protocol) or IMAP (internet message access protocol) are used to receive emails at the recipient's end.

Sending email: A distributed email will contain components, such as a header and a separate body. The customer and server line up a sequence of responses to the client's request before sending the email. The header is different from the body in that it will come to an end when there are no more lines. The message body by reception contains the specific information. After a null line, each data point in the body is taken.

Password organisation could be risky. Keeping a password straight might be risky, so make sure to teach them how to establish a database table while also keeping the password straightforward. The server will be contacted when the user requests to log in so it may check the live load and save the username and password.

A. System Architecture

The architectural diagram displays the system design for the voice-based email system for visually challenged users. The architecture consists of numerous parts and how they interact to enable users to send emails using voice commands. Let us talk about the architecture in more detail. The User component represents the visually impaired user who interacts with the system. The Speech Recognition component captures the user's spoken speech instructions. With voice recognition technology, which converts spoken words into text, the system can process the user's requests.

The Email Server component is in charge of handling the email-related functions. By connecting to the email service provider's server, it enables email sending and receiving. The Email Server component checks the user's login details with the Authentication component before beginning email operations. The authentication component establishes a safe and regulated environment by verifying that the user has authorization to access their email account.

When a user tries to log in, the system prompts them for their login details. The authentication component authenticates the credentials by comparing the provided credentials to the user data that has been saved. If the credentials are legitimate, the system moves on to the next phase. Even if the user's login credentials are entered incorrectly, the system will prompt them to do so again until they are properly authorised. After the user has been validated, they can proceed to the "Compose Email" component. At this stage, the user follows audio directions to input the recipient's email address, message body, and topic. The NLP component processes the user's input before extracting the relevant data.

An accessible user experience is produced via the system's Text-to-Speech (TTS) Converter component. This section converts the system's output into voice format, including email content or confirmation messages. The TTS Converter enables the system to interact with the visually impaired user vocally, allowing them to effectively receive and understand the system's replies. Arrows on the architectural diagram depict the information flow and interactions among the parts. The arrow pointing from the User component to the Speech Recognition component, for example, depicts the flow of the user's spoken input. The arrows between the components depict how information and control are passed between them in a manner like this.

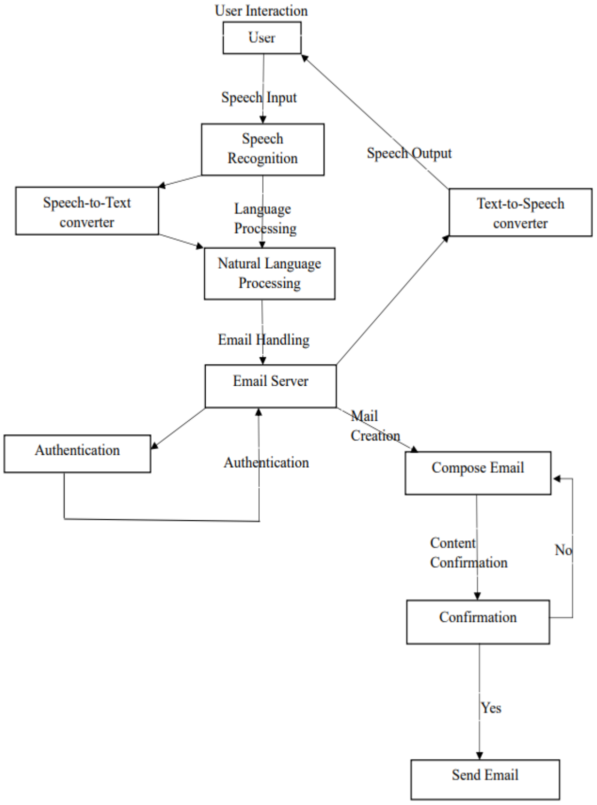

Fig. 1. System architecture of the proposed system

B. Design

1) User Interface Design

The user interface for the voice-based email system has been painstakingly created using the tools in Android Studio, resulting in a seamless and straightforward operation. The interface is made up of three independent displays: the logo screen, the login page, and the message page. Every screen has a distinct purpose that makes it simpler for users to interact with and utilise the programme.

2) Logo Screen

When users initially launch the programme, they are welcomed by the Logo Screen, which serves as the system's initial visual depiction of the voice-based email system. This page presents the user interface and displays the application's amiable and distinctive logo or symbol.

3) Login Page

After viewing the Logo Screen, users are sent to the Login Page, which acts as the doorway to their email account. On the login page, the User ID and Password sections are both very important.

a) *User ID Field:* Users can fill out this field with their unique user ID for their Gmail account. Users can enter text using a virtual keyboard or a voice command utilising the text entry function it provides.

b) *Password Field:* The Password field's safe input area allows users to enter their account password. The password entering is covered up in order to safeguard privacy and prevent unauthorised access.

4) Message Page

After successfully authenticating into the system, the user is then sent to the Message Page, where they may create and send emails. There are many text areas on the Message Page where users may type the necessary email details:

a) *Recipient Email Address Field:* Users can use this field to input the recipient's email address. There is a text entry form available that is like the User ID field on the Login Page.

b) *Message Subject Field:* This allows users to provide a brief yet informative subject line for their email. Users may speak the subject into a text entry box while utilising voice recognition.

c) *Message Text Field:* The Message Text box allows users to enter email content in a larger text input area. Users may enter the message's text, along with any attachments or additional information, using the virtual keyboard or voice input.

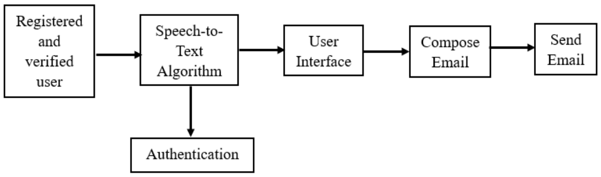

Fig. 2. System design of the proposed system

C. Implementation

- *Login*: A Boolean that denotes a user's ability to log in. It verifies the user's credentials and returns a Boolean value indicating if the login process was successful or unsuccessful.
- *Compose Email*: A void function allows the user to create an email by passing parameters for the subject, message content, and recipient's email address.
- *Confirm Content*: A Boolean value is used to verify the email's content. It provides a Boolean result that indicates if the user has approved or disapproved of

the material.

- *Send Email*: The process of sending the composed email is started by the void function sendEmail().

Converting spoken words into text is possible using the public function convertSpeechToText(): string of the SpeechRecognition class. This function retrieves the text that corresponds to the user's spoken input.

Three public methods for identifying specific information in the text are provided by the NLP class. extractRecipient(text: string), extractSubject(text: string), and extractMessage(text: string) are some of these methods. Every method extracts the recipient email address, subject, or message content from the input text and then returns it.

The Email Server class represents the component responsible for handling email-related operations. It has two public methods:

The component in charge of dealing with email-related processes is represented by the Email Server class. There are two public methods:

- *Authenticate User:* The user's email address and password are verified using the authenticateUser (email: string, password: string) function, which returns a Boolean value. It requests the user's password and email as parameters and then returns a Boolean value indicating whether the authentication was successful.
- Send Email: The sending of an email is started by this function. It sends the email using the recipient's email address, the message's subject, and its content.

The component in charge of verifying the email's content is represented by the Confirmation class. It has a single public method called confirmContent(email: string, subject: string, message: string): Boolean, which accepts the email address, subject, and message content as parameters and returns a Boolean value indicating whether the user has confirmed that the content is correct.

Text to speech conversion is handled by the TTSConverter class. It has a single public method, convertTextToSpeech(text: string): void, that turns the input text into audible speech.

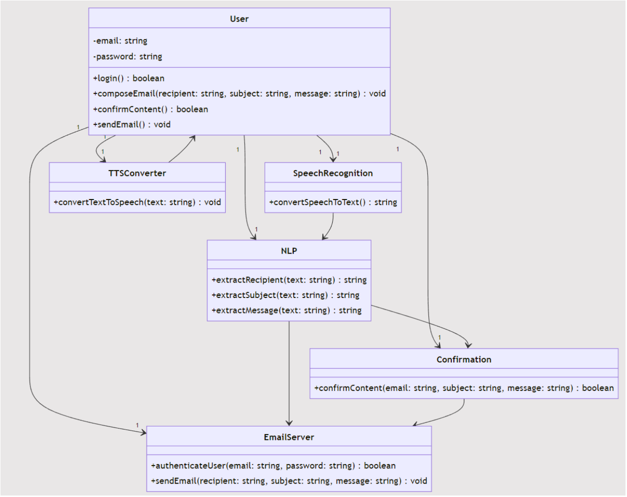
User
-email: string
-password: string
+login() : boolean
+composeEmail(recipient: string, subject: string, message: string) : void
+confirmContent() : boolean
+sendEmail() : void
TTSConverter
+convertTextToSpeech(text: string) : void
SpeechRecognition
+convertSpeechToText() : string
NLP
+extractRecipient(text: string) : string
+extractSubject(text: string) : string
+extractMessage(text: string) : string
Confirmation
+confirmContent(email: string, subject: string, message: string) : boolean
EmailServer
+authenticateUser(email: string, password: string) : boolean
+sendEmail(recipient: string, subject: string, message: string) : void
1
1
1
1
1
1
1
1
1
1

Fig. 3. The main classes involved in the voice-based email system, their attributes, and their relationships

4. Results and Discussions

A. Analysis of Speech to Text Tool Used

The inbuilt speech-to-text tool provided by the android studio was utilised in this project. Six test cases were employed in this experiment to determine how accurately the speech-to-text system performed. The spoken sentence is represented by the original text, while the speech-to-text system's output is represented by the transcribed text. In order to evaluate how accurately the machine transcribes spoken text, the accuracy percentage is calculated.

1) Word error rate formula

Word error rate often referred to as WER is a way to measure the performance of an automatic speech recognition (ASR) system. It is tricky to measure because the "ASR result" can have a different length than the "Voice input."

Here is a simple way to understand how WER is calculated:

$$\mathbf{WER} = \frac{\mathbf{S + D + I}}{\mathbf{N}} \tag{1}$$

Where,
S = number of substitutions
D = number of deletions
I = number of insertions
N = number of words in reference
To help clarify further, here are some definitions:

a) *Deletion by ASR system*:

- Voice input: I surf small waves
  ASR result: I surf waves

b) *Insertion by ASR system*:

- Voice input: I surf waves
  ASR result: I surf small waves

c) *Substitution by ASR system*:

- Voice input: I surf small waves
  ASR result: I surf all waves

The accuracy percentage is calculated by comparing the transcribed text with the original text and calculating the percentage of correctly transcribed words.

Test cases 1 and 2 both display 100% accuracy, demonstrating that the speech-to-text system accurately and error-free transcribed the uttered sentences. The transcription accuracy in test instances 3 and 4 is only 75% due to some transcription errors. Test case 5 shows a lower accuracy of 50% due to inaccurate word transcriptions in multiple instances. Test case 6 displays an 87.5% accuracy rate with some transcription issues.

B. Analysis of Comparison

In this study, the performance of three speech to text tools - Android Studio speech to text, Google Cloud speech to text, and Microsoft Azure speech to text - was compared on six parameters: accented speech, multilingual speech, noisy environment, challenging vocabulary, speed and pacing, and homophones and ambiguities. Test cases were designed for each parameter to evaluate the accuracy of transcriptions

Table 1
Speech-To-Text accuracy experiment results

Test Case	Original Text	Transcribed Text	Accuracy (%)
1	"Hello, how are you today?"	"Hello, how are you today?"	100
2	"I have a meeting at 2 PM."	"I have a meeting at 2 PM."	100
3	"Can you please call me back?"	"Can you please call me bank?"	75
4	"The weather is sunny and warm."	"The weather is money and warm."	75
5	"What time is the train arriving?"	"What time is the train a driving?"	50
6	"I need to buy some groceries."	"I need to pie some groceries."	87.5

Table 2
Input sample and its transcription in noisy environment

Test Sample	Google Cloud	Android Studio	Microsoft Azure
Intermittent noise	"Can you call me back later?"	"Can you call me back later?"	"Can you call me black cater?"
Ambient noise	"I'd like a cappuccino, please."	"I'd like a cappuccino, please."	"I'd like a cup of chino, please."

Table 3
Input sample transcription with challenging vocabulary

Test Sample	Google Cloud	Android Studio	Microsoft Azure
Medical terminology	"The patient has a myocardial infarction."	"The patient has a myocardial infarction"	"The patient has a myocardial infection"
Scientific terms	"The experiment yielded significant results"	"The experiment yielded significant results”	"The experiment shielded significant results"
Legal jargon	"The defendant pleaded not guilty"	"The defendant pleaded not guilty"	"The defendant pleaded not guilty"

generated by the tools. The results provide insights into the tools' performance and their suitability for different speech recognition scenarios.

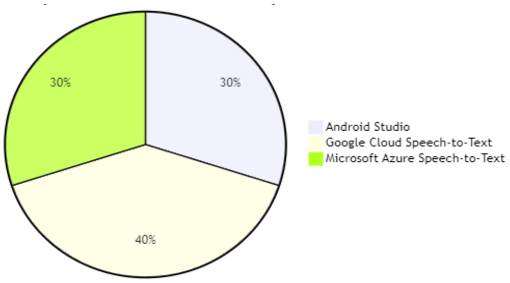

Fig. 4. Performance of speech to text tools in noisy environment

This pie chart illustrates the distribution of performance among the three tools in a noisy environment. It shows that Google Cloud Speech-to-Text performed the best with 40% accuracy, followed by Android Studio with 30% accuracy, and Microsoft Azure Speech-to-Text also with 30% accuracy.

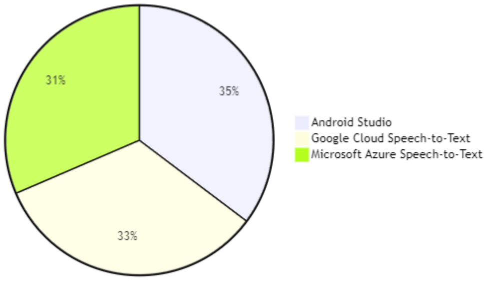

Fig. 5. Performance of speech to text tools with challenging vocabulary

The bar graph visually represents the performance of three speech to text tools - Android Studio speech to text, Google Cloud speech to text, and Microsoft Azure speech to text - across different parameters. The graph consists of vertical bars that indicate the average accuracy rate of each tool for each parameter.

The x-axis represents the six different parameters, namely accented speech, multilingual speech, noisy environment, challenging vocabulary, speed and pacing, and homophones and ambiguities. Each parameter is labelled on the x-axis.

The y-axis represents the accuracy rate in percentage, ranging from 0% to 100%. The scale on the y-axis allows for easy comparison of the accuracy rates between the tools.

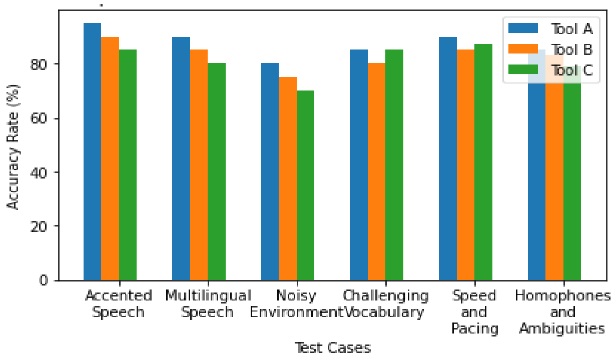

Fig. 6. Grouped bar graph showing speech to text tool performance in different scenarios

Tool A is Google Cloud speech to text whereas tool B is Android Studio inbuilt speech to text while tool C is Microsoft Azure text to speech. We chose to use the built-in speech-to-text tool in Android Studio for our project even though the text-to-speech tool from Google Cloud performed better in our evaluation. A number of factors went into this choice. The built-in tool, for starters, provided easy interaction with our Android application development environment, enabling us to use its features without further setup or configuration. Our development process was expedited by this integration, which also made testing and debugging more effective.

Furthermore, our program could run without an active internet connection thanks to the built-in speech-to-text tool's offline functionality. In situations where users might not have continuous or dependable internet access, this was useful because it guaranteed uninterrupted voice recognition functionality.

Table 4
The comparison between traditional system and progressive system of emailing

S. No.	Traditional System	Progressive System
1.	Compared to the progressive system, it is unstable.	It is more trustworthy because of its high level of security.
2.	This system demands the use of the keyboard.	A keyboard is not necessary since the system relies on IVR or Interactive Voice Response.
3.	lengthy procedure (Slow execution).	It is quicker and more effective than a normal system.
4.	The system can only be used by those who are disability-free.	Users of the system can be both able-bodied and impaired.
5.	People with visual impairments cannot communicate with others or use online email services.	The mobile application-based email system will enable interaction for blind persons (using voice commands).

Cost-effectiveness also played a role in our decision. Although the text-to-speech tool from Google Cloud may have provided higher performance, it frequently has additional expenses based on usage and API calls. We were able to eliminate these extra costs by making use of the built-in tool, making our application more user-friendly and economical.

Finally, we made decisions based in large part on privacy and data security concerns. We were able to prevent sensitive user voice data from leaving the device and application by using the built-in tool, potentially lowering privacy issues related to transferring speech data to external services or APIs.

Overall, the advantages of seamless integration, offline functionality, cost effectiveness, and improved privacy and data security offered by Android Studio's built-in speech-to-text tool outweighed the potential performance gains. This is true even though Google Cloud's text-to-speech tool may have offered better performance.

The graph below (figure 7) depicts the comparison between progressive emailing system and traditional system. The parameters are placed on the x-axis with different parameters being Security, Dependability on keyboard for any sort of input, processing of input, accessibility to different emailing components and interaction with internet. The y-axis contains the score to show which performs better. Reviewed by general public and visually handicapped people via google forms which had all the parameters and were asked to score based on these 5 parameters. The score for both the systems has been displayed below. Their parameters are compared on a scale of 1 to 15.

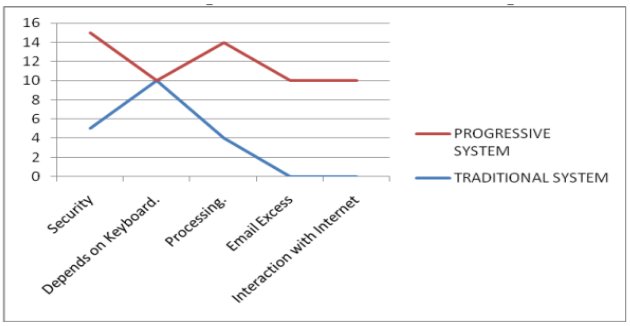

Fig. 7. Results with comparison study

5. Conclusion

In conclusion, an Android voice-based email system has the potential to completely change how people with visual impairments access and manage their emails. This system offers an inclusive and accessible environment where users may communicate with their email accounts using voice commands by utilising speech recognition technology. The creation of such a system solves the challenges that users of conventional email interfaces, which mainly rely on visual interactions, encounter. Users may create, send, receive, and manage emails using spoken instructions by incorporating speech recognition, doing away with the requirement for visual clues or outside help.

Android requires the use of technologies like the JavaMail API and speech recognition libraries in order to develop a voice-based email system. To guarantee a seamless and safe user experience, the system architecture includes elements like user interface design, voice recognition, email integration, accessibility features, and security measures. Beyond email communication, voice-based email systems have several advantages. The system may be developed to include further functions like text messaging and voice-activated control of other programmes, increasing the freedom and productivity of users who are blind or visually impaired.

In conclusion, a voice-based email system for Android allows people who are blind or visually impaired to access and use email services on their own, closing the accessibility gap and enabling them to fully engage in the digital world.

References

[1] Rahul Kumar, Vaishali Singh, Nikhat Akhtar, Versha Verma, Shivam Srivastava, “Voice Based Email System for People with Visual Impairment”. International Journal of Advances in Engineering and Management, Volume 4, Issue 3, pp. 476-484, Mar. 2022.

[2] Harivans Pratap Singh, Aman Pratap Kushwaha, Aayushmaan, Harendra Singh, “Voice-Based Email System”, International Journal of Innovative Science and Research Technology, Volume 6, Issue 7, July 2021.

[3] Payal Dudhbale J. S.Wankhade, P. S. Narawade ."Voice-Based System in Desktop and Mobile Devices for Blind People ". In International Journal of Scientific Research in Science and Technology, 2018.

[4] Aishwarya Belekar, Shivani Sunka, Neha Bhawar, Sudhir Bagade, “Voice based E-mail for the Visually Impaired”. International Journal of Computer Applications, Volume 175, No. 16, September 2020.

[5] Ritwika Ghose, Tirthankar Dasgupta, Anupam Basu, “Architecture of a Web Browser for Visually Handicapped People”. -Proceedings of the 2010 IEEE Students' Technology Symposium 3-4 April 2010, IIT Kharagpur

[6] T. Shabana, A. Anam, A. Rafiya, K. Aisha, “Voice based email system for blinds”. International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 1, January 2015.

[7] Khan, R., Sharma, P. K., Raj, S., Verma, S. K., & Katiyar S. Voice Based E-Mail System using Artificial Intelligence.

[8] Pacha Shobha Rani, Maddireddy Venkata Sai Likhitha, Borra Sneha Latha, “Voice Based E-Mail System for Blinds”. Journal of Emerging Technologies and Innovative Research, Volume 5, Issue 6, June 2018.

[9] Latha L, Babu B, Sowndharya S, “Voice Based Email with Security for Visually Challenged”. International Journal of Engineering and Advanced Technology, Volume 8, Issue 6S3, September 2019.

[10] Rohit Rastogi, Anshika Rajput, Archana, Komal, “An Application of Voice Mail: Email Services for the Visually Challenged Individual”. International Journal of Software and Computer Science Engineering, Volume 4, Issue 1.

[11] Joshua Hailpern, Loretta Guarino Reid, Richard Boardman, Srinivas Annam, “WEB 2.0: Blind to an Accessible New World”. WWW '09: Proceedings of the 18th international conference on World wide web, pp. 821–830, April 2009.

[12] K. Jayachandran, P. Anbumani, "Voice Based Email for Blind People". International Journal of Advance Research, Ideas and Innovations in Technology, Volume 3, Issue 3.

[13] Mullapudi Harshasri, Manyam Durga Bhavani, and Misra Ravikanth, "Voice Based Email for Blind". International Journal of Innovative Research in Computer Science & Technology, Volume 9, Issue 4, July 2021.

[14] Akshita Bhandari, Aayushi Shukla, Darshita Khanna, Garima Verma, Poorva Shinde, Asif Ali, "Voice Based Email System Using Python". EPRA International Journal of Research and Development, Volume 6, Issue 7, July 2021.

[15] Lee Jung-Il, Kim Soon-Cheol, Won Hui-Chul, "Voice-based messenger using NXT touch-sensor input unit and the Bluetooth wireless communication for the blind". Journal of the Korea Industrial Information Systems Research, Volume 13, Issue 5, pp. 78-86, 2008

[16] Jae Sung Cha, Dong Kyun Lim, Yong-Nyuo Shin, "Design and Implementation of a Voice Based Navigation for Visually Impaired Persons". International Journal of Bio-Science and Bio-Technology, Vol. 5, No. 3, June, 2013.

[17] M. Sohrawordi, M.M. Hossen, M.P. Uddin, A.M. Nitu, M.R. Islam "Android-Based Walking Assistant for Blind and Low-Vision People Suggesting the Shortest Path Using Floyd-Warshall Algorithm". Journal of Innovation & Development Strategy, Volume 9, Issue 2, August 2015.

[18] Rijwan Khan, Pawan Kumar Sharma, Sumit Raj, Sushil Kr. Verma, Sparsh Katiyar, "Voice Based E-Mail System using Artificial Intelligence". International Journal of Engineering and Advanced Technology, Volume 9, Issue 3, February 2020.

[19] Itunuoluwa Isewon, Jelili Oyelade, Olufunke Oladipupo, "Design and Implementation of Text to Speech Conversion for Visually Impaired People," International Journal of Applied Information Systems, Volume 7, No. 2, April 2014.

[20] Shakhovska, N., O. Basystiuk, and K. Shakhovska, "Development of the speech-to-text chatbot interface based on Google API." CEUR Workshop Proceedings, Vol. 2386, 2019.

[21] Saurabh Sawant, Amankumar Wani, Sangharsh Sagar, Rucha Vanjari and M. R. Dhage, "Speech Based E-mail System for Blind and Illiterate People." International Research Journal of Engineering and Technology, Volume 5, Issue 4, pp. 2398-2400, April 2018.