Uttarakhand Residential University, Almora

A Project Report on

Voice Based Email System for blind people

Under the Guidance of

Mis. *[ ]*

Submitted by

Mr: Deepak kandpal

Enrollment Number: *20181119202*

Roll Number: *BSCCS015*

Course name & Semester: (Cyber security *and 6th sem)*

Uttarakhand Residential University, Almora

A Minor Project Synopsis Report Submitted in Partial Fulfilment of the Requirement for the Award of Degree of

BACHELOR OF CYBER SECURITY

To

Uttarakhand Residential University, Almora

ACKNOWLEDGEMENT

It is great pleasure to present this report on the project name[illegible] undertaken by me as part of my bachleor of cyber security(BSC CS) curriculum.

I would like to give my sincere thanks to my Project Guide **Amir hussian** for providing us a vision about the system. I am grateful to them for their guidance, encouragement, understanding and insightful support in the development process.

Last but not the least I would like to mention here that I greatly indebted to each and everybody who has been associated with my project at any stage but whose name does not find a place in this acknowledgement.

Thanking you.

Abstract

Internet has become one of the basic amenities for day-to-day living. Every human being is widely accessing the knowledge and information through internet. However, blind people face difficulties in accessing these text materials, also in using any service provided through internet. The advancement in computer based accessible systems has opened up many avenues for the visually impaired across the globe in a wide way. Audio feedback based virtual environment like, the screen readers have helped Blind people to access internet applications immensely. We describe the Voicemail system architecture that can be used by a Blind person to access e-Mails easily and efficiently. The contribution made by this research has enabled the Blind people to send and receive voice based e-Mail messages in their native language with the help of a computer.

TABLE OF CONTENT

1.Introduction

Internet plays a vital role in today's world of communication. Today the world is running on the basis of internet. No work can be done without use of 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. So it goes very difficult to them when to live in this world of internet. Nowadays there are various technologies available in this world like screen readers, ASR, TTS, STT, etc. but these are not that much efficient for them. Around 39 million people are blind and 246 people have low vision and also 82 of people living with blindness are 50 aged and above. We have to make some internet facilities to them so they can use internet. Therefore we came up with our 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 system easy to use for user of any age group. Our system provides location of where user is prompting through voice so that user doesn't have to worry about remembering which mouse click operation he/she wants to achieve.

• Objective

Communication is one of the most important aspects of human life. The world has been turned into a global village after the integration of communication technologies with the Internet. And of all the different communication technologies available, email is considered as one of the most pervasive forms of communication. Emails have become a norm of communicating with each other replacing letters completely. Even after other technologies such as social networking are also taking hold over the Internet world, email remains the most ubiquitous form of business communication.

This project aims to develop a voice based email system that would help visually impaired people to access email in a hassle free manner. Along with providing usage of mail services easily and efficiently, the system will also reduce the cognitive workload which has to be normally taken by the visually impaired to remember and type characters using the traditional Braille keyboards available to them. The GUI of this system has been evaluated against the interface of the traditionally available mail system. Along with the blind, those who are illiterate can also benefit from this system.

The most crucial aspect that will be considered while developing this system is that the users of this system need not have any basic information about the

keyboard shortcuts used or where the keys are located. All functions to be used in this system will be based on simple mouse click operations making the system very user friendly.

- # Application:

This project is proposed for the betterment of society. This project aims to help the visually impaired people to be a part of growing digital India by using internet and also aims to make life of such people quite easy. Also, the success of this project will also encourage developers to build something more useful for visually impaired or illiterate people, who also deserves an equal standard in society.

This application proposes an android application, designed specifically for visually impaired 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 for e.g. Read, Send, Compose Mail, Address Book etc. The VMAIL system can be used by a blind person to access mails easily and efficiently. Thus reliance of visually impaired on other people for their activities related to mail can be reduced.

2.System Analysis

2.1 Existing System:

In previous work, blind people does not send email using the system. The multitude of email types along with the ability setting enables their use in nomadic daily contexts. But these emails are not useful in all types of people such as blind people they can't send the email. Audio based email are only preferable for blind peoples. They can easily respond to the audio instructions. In this system is very rare. So there is less chance to available this audio based email to the blind people.

2.2 Proposed System:

We describe the voicemail system architecture that can be used by a blind person to access e-mails easily and efficiently. The contribution made by this research has enabled the blind people to send and receive voice-based e-mail messages in their native language with the help of a computer or a mobile device. Our proposed system GUI has been evaluated against the GUI of a traditional mail server. We found that our proposed architecture performs much better than that of the existing GUIS.

In this system mainly three types of technologies are used namely:

SPEECH_ TO_ TEXT Converter The system acquires speech at run time through a microphone and processes the sampled speech to recognize the uttered

text. The recognized text can be stored in a file. We are developing this on Android platform using Eclipse workbench. Our speechto-text system directly acquires and converts speech to text. It can supplement other larger systems, giving users a different choice for data entry. A speech-to-text system can also improve system accessibility by providing data entry options for blind, deaf, or physically handicapped users. Speech recognition system can be divided into several blocks: feature extraction, acoustic models database which is built based on the training data, dictionary, language model and the speech recognition algorithm. Analog speech signal must first be sampled at time and amplitude axes, or

usually 20 ms because the signal in this interval is considered stationary. Speech feature extraction involves the formation of equally spaced discrete vectors of speech characteristics. Feature vectors from training database are used to estimate the parameters of acoustic models. The acoustic model describes properties of the basic elements that can be recognized. The basic element can be a phoneme for continuous speech or word for isolated words recognition.

TEXT_ TO_ SPEECH Converter Converting text to voice output using speech synthesis techniques. Although initially used by the blind to listen to written material, it is now used extensively to convey financial data, e-mail messages, and other information via telephone for everyone. Text-to-speech is also used on handheld devices such as portable GPS units to announce street names when giving directions. Our Text-to-Speech Converter‖ accepts a string of 50 characters of text (alphabets and/or numbers) as input. In this, we have interfaced the keyboard with the controller and defined all the alphabets as well as digits keys on it. The speech processor has an unlimited dictionary and can speak out almost

any text provided at the input most of the times. Hence, it has an accuracy of above 90%. It is a microcontroller based hardware coded in Embedded C language. Further research is to be done to optimize various methods of inputting the text i.e. Reading the text using optical sensor and converting it to speech so that almost all sorts of physical challenges faced by the people while communicating are overcome.

WORD RECOGNITION Voice recognition software (also known as speech to text software) allows an individual to use their voice instead of typing on a keyboard. Voice recognition may be used to dictate text into the computer or to give commands to the computer.Voice recognition software allows for a quick method of writing onto a computer. It is also useful for people with disabilities who find it difficult to use the keyboard. This software can also assist those who have difficulty with transferring ideas onto paper as it helps take the focus out of the mechanics of writing. Word recognition is measured as a matter of speed, such that a word with a high level of recognition is read faster than a novel one. This manner of testing suggests that comprehension of the meaning of the words being read is not required, but rather the ability to recognize them in a way that allows proper pronunciation.

Therefore, context is unimportant, and word recognition is often assessed with words presented in isolation in formats such as flash cards Nevertheless, ease in word recognition, as in fluency, enables proficiency that fosters comprehension of the text being read.

2.3 Diagram:

2.3.1 UML Diagrams

The Unified Modeling Language (UML) is a general "purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system".

- Use-Case Diagram:

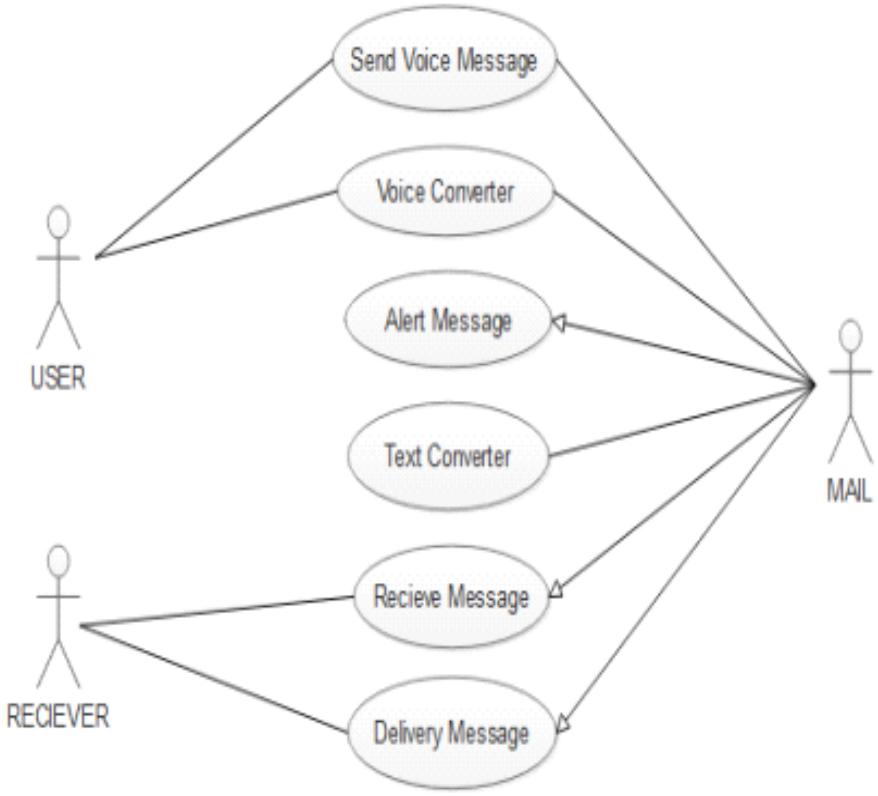


Fig No: 1 Use Case Diagram for user

- Class Diagrams:

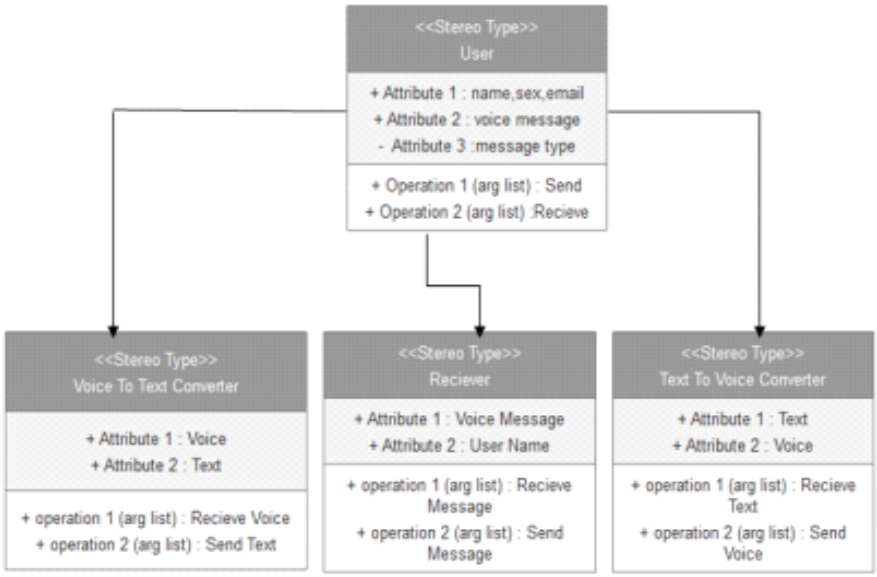


Fig No. : 2 Class diagram for user

- Sequence Diagram:

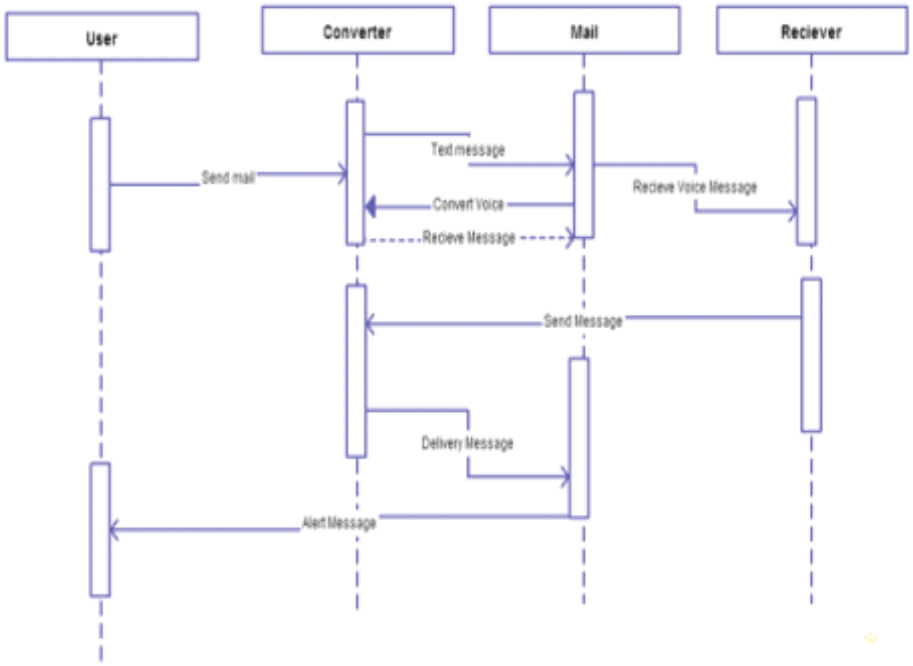


Fig No: 3 Sequence Diagram

2.3.2 DFD(Data Flow Diagram):

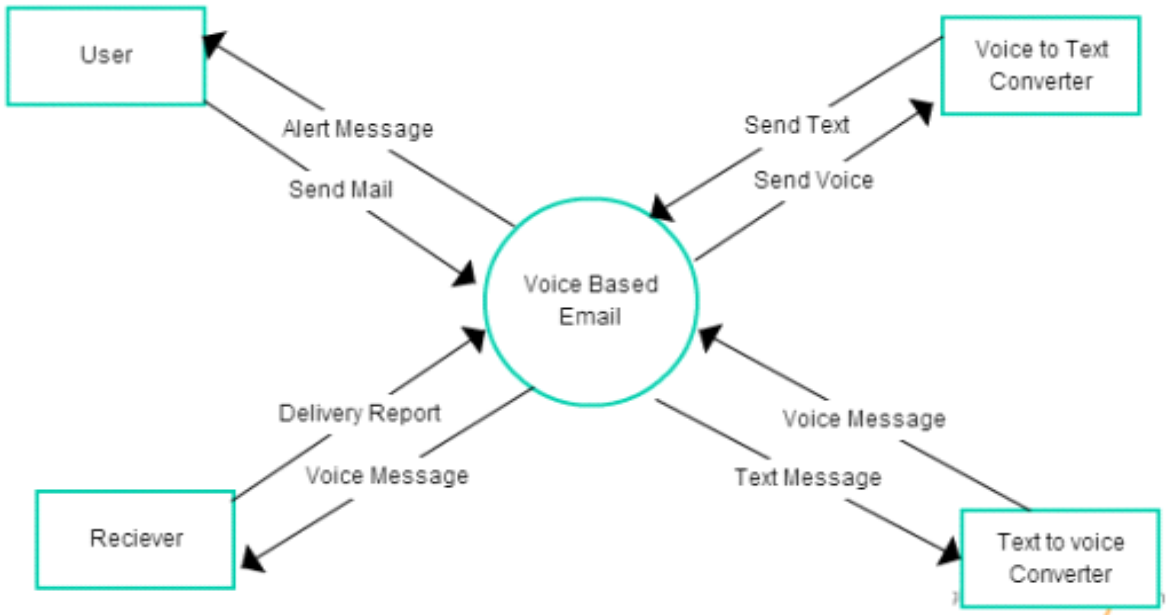


Fig No: 4 User Details

3.Implementation

3.1 Flow Diagram of Proposed Project

The above flow diagram explains the complete working of the project and the project can be easily implemented using the above flow diagram and ER diagram. However, the project is currently under development so we are providing details of few modules which are already developed by us:

- **Login:** This is the very first page and will ask user to enter login credentials. It will prompt user with voice command to enter user name. After receiving user name it will prompt again for password. After receiving all of the details from user, it will encrypt and check the validity of the details entered by user. If valid, then user will be redirected to dashboard else will be sent back to login page.
- **Dashboard:** After successful login, user will be redirected to this page and this is the main page from where user can perform all the activities like, compose a new mail, check inbox, save to draft etc.

  Below steps specifies the operation that will be executed based on a specific click of mouse button. As the user is supposed to be blind, so it is allowed to click blindly anywhere on the screen:
  - Left Click to Compose a new Mail.
  - Right Click to Go to the Sent Mails.
  - Double Left Click to Go to the Inbox View.
  - Scroll Button Click to go to Trash Messages.
  - Double Right Click to Log out of the Session.

- **Common Rule:**

  Left Click = Next Step

  Right Click = Back

  Scroll Button Click = Dashboard
- **Compose a Mail:** This module is used to compose a new mail. Below are the steps followed by this module to compose a new mail:
  - Left Click from dashboard to Compose a new Mail.
  - Give Voice Data about the Recipient, and CC, BCC, the Subject and then the body
  - If satisfied with current input Left Click to go to next Stage
  - In next Stage your voice will be recognized,Left Click to proceed
  - In this stage your voice and input will be verified if any problem is found you are redirected to that or Left Click to proceed to Send the Mail or Right Click to Double Left Click to save as Draft
- **Inbox:** This page will store all of the mails received by user. Below steps explains how to access a mail from inbox:
  - All the received Mails will be listed sorted in order of date
  - Double left Click to give voice input to filter Mail,when Satisfied Left click to proceed
  - In this Stage your mail will be read out , Double Left Click to start/pause
- **Trash:** This folder will store all of mails deleted by the user. Below steps provide detailed explanation about this module:
  - All the deleted Mails will be listed sorted in order of date
  - Double left Click to give voice input to filter Mail, when Satisfied Left click to proceed to Read Section.

    - In this Stage your mail will be read out.
    - Double Left Click to start/pause
    - Left Click to proceed to Delete the Mail or Right Click to back
    - If in Delete Section Left Click to Delete the Mail
- **Sent Mail:** This folder will store all of the mails sent from the user. Below steps explains the working of this module:
    - All the sent Mails will be listed sorted in order of date

    - Double left Click to give voice input to filter Mail, when Satisfied Left click to proceed to Read Section.
    - In this Stage your mail will be read out, Double Left Click to start/pause
    - Left Click to proceed to Delete the Mail or Right Click to back
    - If in Delete Section Left Click to Delete the Mail

3.2 Code

```
package com.example.pallavi.voicemail;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.speech.RecognizerIntent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {
    TextView outputText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        outputText=(TextView) findViewById(R.id.txt_output);
    }

    public void btnSpeech(View view) {

        Intent intent =new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
                RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,
Locale.getDefault());
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT,"Speak
Your Mail");
        try {
            startActivityForResult(intent, 1);
        }
        catch(ActivityNotFoundException e )
        {
            Toast.makeText(this, e.getMessage(),
```

```
Toast.LENGTH_SHORT).show();
        }

    }

    @Override
    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode) {
            case 1:
                if (resultCode == RESULT_OK && null != data) {
                    ArrayList<String> result =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    outputText.setText(result.get(0));

                }


                break;
        }
    }
    }

package com.example.pallavi.voicemail;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.speech.RecognizerIntent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Locale;

public class Main2Activity extends AppCompatActivity {
    TextView outputText;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        outputText=(TextView) findViewById(R.id.txt_output);
    }

    public void btnSpeech(View view) {
```



```
Intent intent =new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
                RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,
Locale.getDefault());
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT,"Send
To");
        try {
            startActivityForResult(intent, 1);
        }
        catch(ActivityNotFoundException e )
        {
            Toast.makeText(this, e.getMessage(),
Toast.LENGTH_SHORT).show();
        }

    }

    @Override
    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode) {
            case 1:
                if (resultCode == RESULT_OK && null != data) {
                    ArrayList<String> result =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
```

```
                    outputText.setText(result.get(0));

                }


                break;
            }
        }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.pallavi.voicemail.MainActivity">
```



```
<TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="22sp"
        android:textAlignment="center"
        android:layout_marginTop="30dp"
        android:padding="20dp"
        android:id="@+id/txt_output"
        />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:text="Tap To Open Mic"
        android:textSize="20sp"
        android:textColor="#ffffff"
        android:layout_alignParentBottom="true"
        android:layout_margin="20dp"
```

```
        android:onClick="btnSpeech"
        />

</RelativeLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/andro
id"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#36a707"

tools:context="com.example.pallavi.voicemail.splashscre
en">

    <ImageView
        android:id="@+id/logo"
```



```
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:src="@drawable/textspeech" />
```

```
</RelativeLayout>
```

```
package com.example.pallavi.voicemail;

import android.content.Intent;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class splashscreen extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState)
{
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splashscreen);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent i =new
Intent(splashscreen.this,MainActivity.class);
                startActivity(i);
                finish();
            }
        }, 6000);
```

```
    }
```



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/andro
id"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

tools:context="com.example.pallavi.voicemail.MainActivi
ty">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="22sp"
        android:textAlignment="center"
        android:layout_marginTop="30dp"
        android:padding="20dp"
        android:hint="Enter Email"
        android:id="@+id/txt1_output"
        />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:text="suject"
        android:id="@+id/editText"
        />

    <Button
        android:id="@+id/btn1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
                            17
        android:layout_alignParentStart="true"
        android:layout_below="@+id/txt_output"
        android:layout_marginTop="116dp"
        android:background="@color/colorPrimary"
        android:text="Send "
        android:textColor="#36a707"
        android:textSize="20sp"


</RelativeLayout>

package com.example.pallavi.voicemail;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.speech.RecognizerIntent;
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Locale;

public class Main2Activity extends AppCompatActivity {
    EditText outputText;
    EditText meditText;
    @Override
    protected void onCreate(Bundle savedInstanceState)
{
        super.onCreate(savedInstanceState);
```

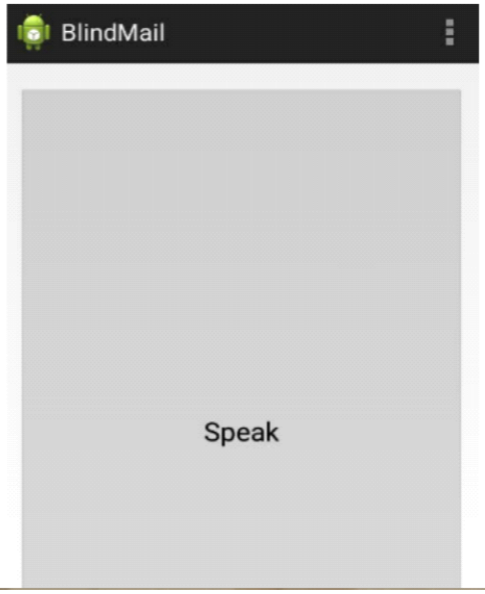


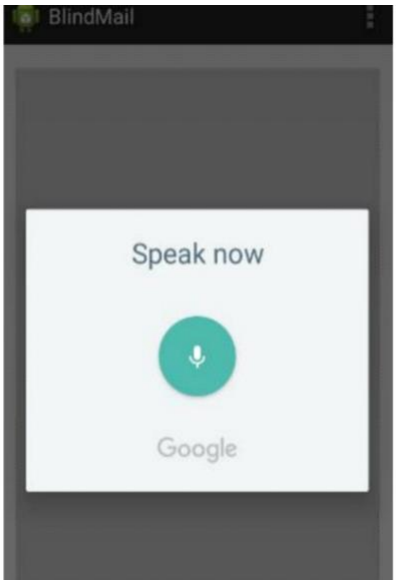
```
        outputText = (EditText)
findViewById(R.id.txt1_output);
        meditText=findViewById(R.id.editText);
        Button btnSend = findViewById(R.id.btn1);
        btnSend.setOnClickListener(new
View.OnClickListener() {


            @Override
            public void onClick(View view) {
```

```
                sendMail();

            }
        });
    }

                                        private void
sendMail() {
        String
recipientlist=outputText.getText().toString();
        String[] recipients=recipientlist.split(",");

        String subj=meditText.getText().toString();

        Intent intent=new Intent(Intent.ACTION_SEND);

        intent.putExtra(Intent.EXTRA_EMAIL,recipients);
        intent.putExtra(Intent.EXTRA_TEXT,subj);
        intent.setType("message/rfc822");

startActivity(intent.createChooser(intent,"choose an
email client"));
                                        }
```


Screen

- ***Welcome Page***





- ***Inbox Page***

3.3Software Requirement

- Android Studio
- Emulator
- Genymotion

3.4 Languages

- Java
- Kotlin
- CSS

- Html

3.5 API

- Google Api
- Speech-to-text Api

4. Conclusion And Future Scope

4.1 Conclusion

This e-mail system can be used by any user of any age group with ease of access. It has feature of speech to text as well as text to speech with speech reader which makes designed system to be handled by visually impaired person as well as blind person.

4.2 Advantages:

- The disabilities of visually impaired people are thrashed.
- This system makes the disabled people feel like a normal user.
- They can hear the recently received mails to the inbox, as well as the IVR technology proves very effective for them in the terms of guidance.
- The visually impaired people can advance from Desktop application to the web based application.

• 3 Future Scope

For people who can see, e-mailing is not a big deal, but for people who are not blessed with gift of vision it postures a key concern because of its intersection with many vocational responsibilities. This voice based email system has great application as it is used by blind people as they can understand where they are. E.g. whenever cursor moves to any icon on the website say Register it will sound like “Register Button”. There are many screen readers available. But people had to remember mouse clicks. Rather, this project will reduce this problem as mouse pointer would read out where he/she lies. This system focuses more on user friendliness of all types of persons including regular persons, visually compromised people as well as illiterate.