

# **“Text and Speech Recognition Mobile App”**

*A Societal Centric and Industry Related Project Report submitted in the partial fulfillment of the requirements for the award of the Degree of*

## **BACHELOR OF TECHNOLOGY**

In

## **INFORMATION TECHNOLOGY & COMPUTER APPLICATIONS**

**Submitted**

by

**PRAMOD KUMAR**  
(201FA07083)

**VIKASH KUMAR**  
(201FA07078)

**Under the guidance of**

**Prof. B. Premamayudu**



**VIGNAN'S**

Foundation for Science, Technology & Research

(Deemed to be University)

-Estd. u/s 3 of UGC Act 1956

**VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY  
AND RESEARCH**

**DEEMED TO BE UNIVERSITY VADLAMUDI, GUNTUR.**

**ANDHRA PRADESH, INDIA,  
PIN-522 213.**

**OCT-2023**



**VIGNAN'S**

Foundation for Science, Technology & Research

(Deemed to be University)

-Estd. u/s 3 of UGC Act 1956

**VIGNAN'S FOUNDATION FOR SCIENCE TECHNOLOGY  
AND RESEARCH**

**VADLAMUDI, GUNTUR DIST, ANDHRA PRADESH, INDIA, PIN-522 213**

**CERTIFICATE**

This is to certify that the Inter-Departmental Project-II entitled “**Text and Speech Recognition Mobile App**” is being submitted by **Vikash Kumar (201FA07078) & Pramod Kumar (201FA07083)** for partial fulfilment of Inter- Departmental Project-II is a bonafide work carried out under the supervision of “**Prof. B. Premamayudu (Professor)**” from Department of Information Technology and Computer Application's.

**Dr. B. Premamayudu**  
(Professor)

**Mr. M. Srikanth Yadav**  
Project Coordinator

**Dr. N. Veeranjanyulu**  
Professor & HOD,  
IT & CA

Internal Examiner

External Examiner



**VIGNAN'S**

Foundation for Science, Technology & Research

(Deemed to be University)

-Estd. u/s 3 of UGC Act 1956

**VIGNAN'S FOUNDATION FOR SCIENCE TECHNOLOGY  
AND RESEARCH**

**VADLAMUDI, GUNTUR DIST, ANDHRA PRADESH, INDIA, PIN-522 213**

**DECLARATION**

We hereby declare that the a “Societal Centric and Industry Related Project” entitled **“INFULENCE MAXIMIZATION USING HYBRID APPROACH”** that is being submitted by **PRAMOD KUMAR (201FA07083), VIKASH KUMAR (201FA07078)** which is being submitted by me for the partial fulfilment of the requirements for the award of Bachelor of Technology in the department of IT & CA, Vignan's Foundation For Science Technology and Research, deemed to be university Vadlamudi, Guntur District, Andhra Pradesh, and the result of investigations are carried out by us under the guidance of **Prof. B. Premamayudu (Professor)**.

**Submitted By:**

**Vikash Kumar (201FA07078)**

**Pramod Kumar (201FA07083)**

**Date:**

## **ACKNOWLEDGEMENTS**

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved vice chancellor **Dr. P. Nagabhushan** Who took keen interest in our every effort throughout this course.

We express our deep-felt gratitude to our head of the department **Dr. N. Veeranjaneyulu** and **Mr. M. Srikanth Yadav** coordinator of the project for extending their encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout the project work.

We wish to express our sincere deep sense of gratitude to our guide **Mr. K. Praveen Kumar** significant suggestions and help in every respect to accomplish the project work. His persisting encouragement, everlasting patience and keen interest in discussions have benefited us to be extent that cannot be spanned by words to our college management for providing excellent lab facilities for completion of project within our campus.

Finally, we extend our wholehearted gratitude to all our faculty members of Department of Information Technology who helped us in our academics throughout course.

With Sincere regards,

**Vikash Kumar (201FA07078)**  
**Pramod Kumar (201FA07083)**

# CONTENTS

S.NO.	DESCRIPTION	PAGE NO.
1.	ABSTRACT	05
2.	PROBLEM STATEMENT	06
3.	INTRODUCTION	07
4.	SYSTEM ENVIRONMENT	08
5.	REQUIREMENTS	09
6.	ARCHITECTURE	10
7.	DESIGN	12
8.	IMPLEMENTATION	13
9.	RESULT	27
10.	TESTING	29
11.	CONCLUSION AND FUTURE SCOPE	30

## **ABSTRACT**

Our mobile app is a powerful tool that allows users to take a picture of any text and converts it into a digital format that can be translated into any language. Additionally, the app features a voice-operated playback option that allows visually impaired individuals to navigate and use it with ease. However, our app goes beyond language translation. It features an innovative voice-operated playback option, specifically designed to cater to visually impaired individuals. With this inclusive feature, visually impaired users can easily navigate and utilize the app, breaking down accessibility barriers and providing an equal opportunity to engage with digital content. The impact of our app extends globally. It not only facilitates communication and understanding across diverse languages but also fosters a sense of unity and inclusivity among people from various cultural backgrounds. By democratizing access to information, we aim to bridge the gap and empower individuals worldwide. By empowering users to easily access and understand information in different languages, this app has the potential to benefit people around the world. The voice-operated feature ensures that visually impaired users can use the app without any barriers, making it a truly inclusive solution

## PROBLEM STATEMENT

The limited access to information in different languages poses significant challenges for individuals who do not speak or understand certain languages. They may encounter difficulties in accessing valuable information that is not available in their native language, limiting their ability to fully engage and participate in various contexts. Additionally, the lack of efficient tools to convert physical text into a digital format further exacerbates this issue. Many people face challenges in digitizing and translating physical text, hindering their access to crucial information. Moreover, visually impaired individuals often encounter obstacles when using mobile apps that are not designed with their specific needs in mind. The inaccessibility of such apps creates barriers to communication and limits their ability to access and benefit from the information provided. The lack of inclusive design and features, such as voice-operated playback or text-to-speech functionality, hampers their independence and inhibits their ability to navigate and utilize mobile apps effectively. These barriers not only restrict individuals' access to information but also impede their participation in various domains of life, including education, employment, and social interactions. By addressing these limitations and ensuring that mobile apps are designed with accessibility and inclusivity in mind, we can bridge the gap and empower individuals of diverse linguistic backgrounds and visual abilities to access and utilize information more effectively, promoting equal opportunities and fostering a more inclusive society.

# INTRODUCTION

The Text and Speech Recognition Mobile App project is a groundbreaking endeavor that aims to develop a mobile application capable of real-time recognition and transcription of text and speech. In today's world, where mobile devices have become ubiquitous and effective communication is essential, such an application holds immense potential for individuals, businesses, and organizations. The primary goal of the application is to provide users with a seamless and efficient way to capture spoken or written text using their device's microphone or camera. With just a few taps, users will be able to convert their spoken words or physical text into accurate and instant transcriptions. This functionality alone will greatly enhance productivity, as users can effortlessly capture important information or record conversations without the need for manual transcription. Additionally, the application will offer advanced features to further elevate the user experience. For instance, it will include translation capabilities, allowing users to instantly translate transcriptions into different languages, facilitating communication across language barriers. The app will also incorporate editing tools, enabling users to refine and polish their transcriptions for greater accuracy or readability. Furthermore, users will have the option to share their transcriptions seamlessly, making collaboration and information sharing more efficient. The impact of the Text and Speech Recognition Mobile App project goes beyond individual convenience. Businesses and organizations can benefit greatly from this technology, as it streamlines their communication processes, enhances customer service, and facilitates multilingual interactions. From transcription services and language learning platforms to customer support centers and conference organizers, the potential applications of this app are vast and diverse. By revolutionizing the way we communicate and interact with mobile devices, this project has the power to transform our daily lives. It will bring convenience, accuracy, and efficiency to various communication tasks, saving time and effort for users across different domains. As the project continues to evolve and refine, it holds immense promise for empowering individuals, optimizing workflows, and fostering seamless global communication.



# SYSTEM ENVIRONMENT

## Existing System

1. **Limited functionality:** Current systems for text and speech recognition are often limited in their capabilities. They may lack real-time recognition or provide inaccurate transcriptions, hindering effective communication.
2. **Complex user interface:** Many existing systems have complex user interfaces, making them challenging to navigate and utilize. This complexity can result in user frustration and slower adoption rates.
3. **Inconsistent language support:** Existing systems may have limited language support, restricting users' ability to transcribe and translate text and speech in diverse languages. This can be a significant barrier for global communication.
4. **Limited accessibility features:** In terms of accessibility, existing systems may not cater to the needs of visually impaired individuals, lacking features like voice-operated playback or text-to-speech functionality.
5. **Lack of real-time capabilities:** Real-time recognition is often lacking in current systems, resulting in delays between capturing text or speech and receiving transcriptions. This delay can hinder efficient communication and time-sensitive tasks.

## Proposed System

1. **Real-time recognition:** Our proposed app offers real-time recognition, enabling users to receive instant transcriptions, facilitating efficient and timely communication.
2. **User-friendly interface:** The app features a user-friendly interface, making it easy to navigate and utilize, resulting in a smoother user experience and higher adoption rates.
4. **Extensive language support:** The proposed app supports a wide range of languages, breaking down language barriers and promoting global communication and inclusivity.
9. **Real-time capabilities:** The proposed app excels in real-time recognition, allowing users to capture text and speech instantly, enhancing communication efficiency in time-sensitive situations.
5. **High accuracy:** With advanced algorithms and technologies, our app ensures high accuracy in transcriptions, minimizing misunderstandings and errors in communication.

# REQUIREMENTS

## Used Technologies and SDKs

**1.Firebase ML Kit:** It provides a range of features for implementing machine learning in mobile apps, including image labelling, text recognition, face detection, barcode scanning, and more. These features are powered by Google's machine learning technologies, such as TensorFlow Lite and the Google Cloud Vision API, and can be easily integrated into mobile applications without requiring extensive knowledge of machine learning

**2.Android Speech API:** It is a framework that allows developers to add voice-based interaction to their applications, enabling users to control the app with their voice and receive audio feedback from the app.

**3.SpeechRecognizer:** Its component provides speech recognition capabilities, allowing users to speak commands or input text into an application by speaking. It is capable of recognizing spoken words and converting them into text.

**4.TextToSpeech:** Its component provides text-to speech capabilities, allowing the application to generate spoken output based on text input. It can synthesize speech in multiple languages and voices, and can be customized to adjust the pitch, speed, and volume of the generated speech.

**5.Android Studio :** Android Studio is an integrated development environment (IDE) specifically designed for Android app development. It provides a comprehensive set of tools and features to streamline the app development process. With its user-friendly interface and extensive functionality, Android Studio is a popular choice among developers.

# ARCHITECTURE

Our mobile app is built upon a robust and scalable architecture to ensure efficient and seamless functionality. The architecture comprises several key components working together to deliver a powerful and inclusive user experience.

**1. User Interface (UI):** The UI component focuses on providing an intuitive and visually appealing interface for users. It incorporates elements such as text input fields, camera integration, language selection options, and navigation controls, making it easy for users to interact with the app.

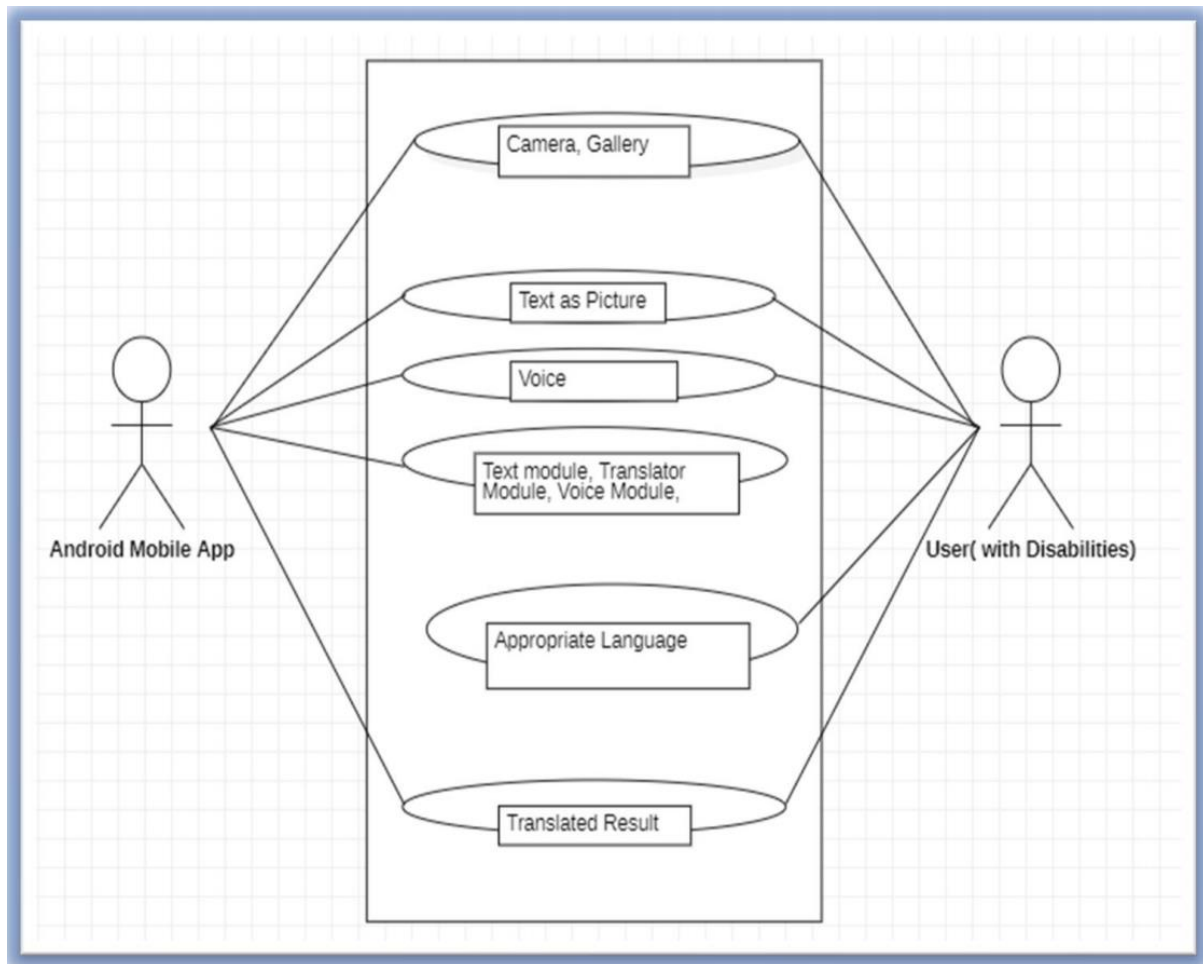
**2. Text and Image Recognition Module:** This module leverages advanced optical character recognition (OCR) technology to analyze and extract text from images captured by the device's camera. It utilizes machine learning algorithms to accurately identify and convert the text into a digital format.

**3. Translation Engine:** The translation engine component plays a crucial role in our app's functionality. It integrates with language translation APIs or libraries to perform real-time translation of the extracted text into the desired language. This enables users to easily comprehend and communicate in different languages.

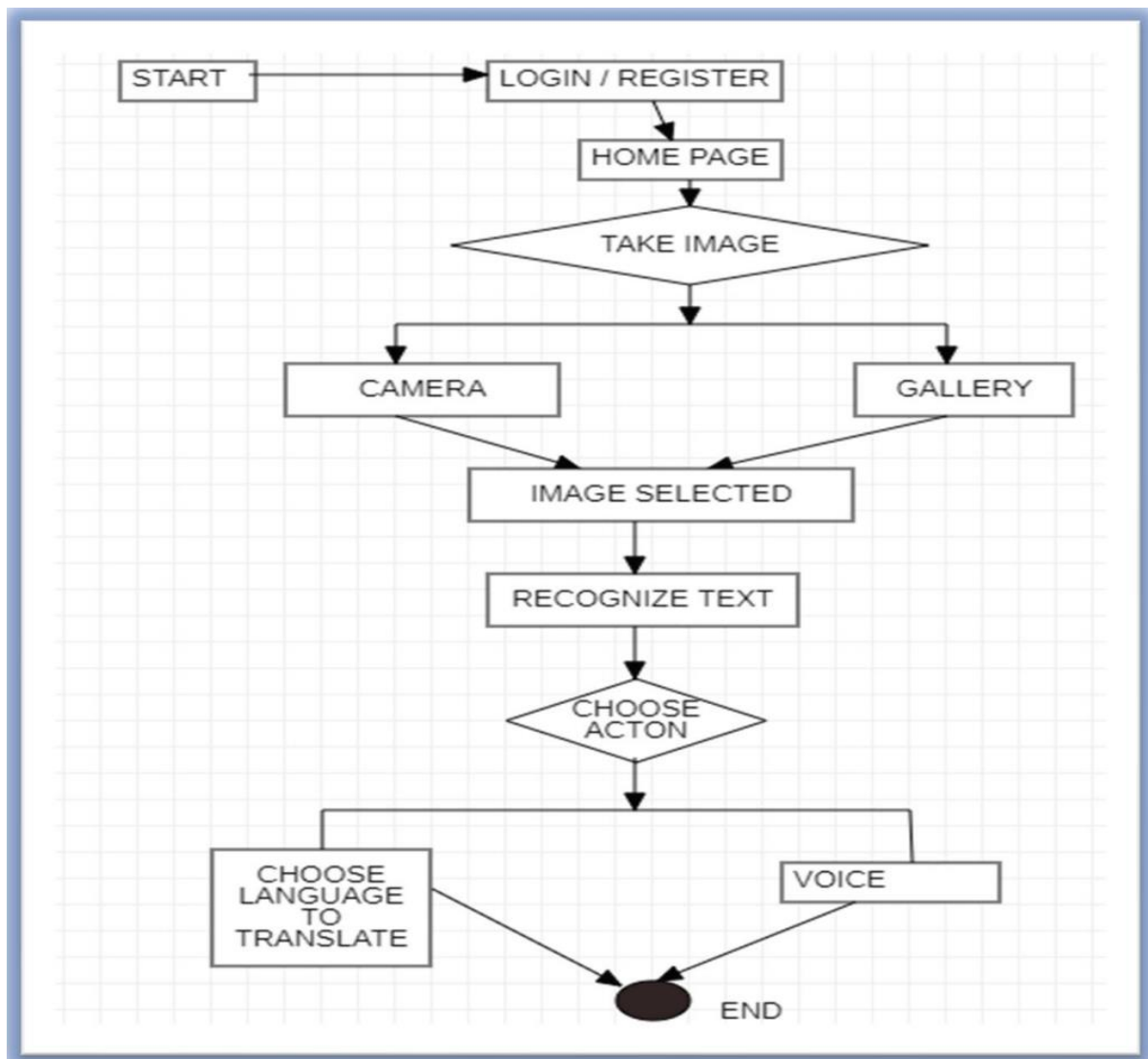
**4. Voice-Operated Playback Module:** The voice-operated playback feature caters specifically to visually impaired users. It utilizes speech recognition technology to convert voice commands into actionable tasks within the app. This module enables visually impaired individuals to navigate through menus, select options, and interact with the app using voice commands, ensuring an inclusive user experience.

# DESIGN

## USECASE DIAGRAM



## FLOWCHART DIAGRAM



# IMPLEMENTATION

## login.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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:id="@+id/SplashScreen"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="#151411"

    android:gravity="center"

    android:orientation="vertical"

    tools:context=".LoginActivity">

    <EditText

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:gravity="center"

        android:ems="50"

        android:text="Welcome to TextOn"

        android:textColor="@color/white"

        android:textSize="30dp"

        android:textStyle="bold" />

    <com.airbnb.lottie.LottieAnimationView

        android:id="@+id/lotieeee"

        android:layout_width="match_parent"

        android:layout_height="500dp"
```

```
        android:backgroundTint="#9E7D28"

        app:lottie_autoPlay="true"

        app:lottie_loop="true"

        app:lottie_rawRes="@raw/lotli"

    />

</LinearLayout>
```

## main.xml

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"
    android:orientation="vertical"
    android:background="@color/black_shade_1"
    android:padding="10dp"
    tools:context=".MainActivity">

    <ScrollView

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:layout_marginTop="40dp">

        <LinearLayout

            android:layout_width="match_parent"

            android:layout_height="match_parent"

            android:orientation="vertical">
```

```
<!-- show image take from gallery or camera -->
```

```
<com.google.android.material.imageview.ShapeableImageView
```

```
    android:id="@+id/imageIv"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:adjustViewBounds="true"
```

```
    android:src="@drawable/baseline_image_24"
```

```
    app:strokeWidth="2dp" />
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:orientation="horizontal">
```

```
    <com.google.android.material.button.MaterialButton
```

```
        android:id="@+id/inputImageBtn"
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
```

```
        android:layout_marginEnd="5dp"
```

```
        android:layout_weight="1"
```

```
        android:text="Take image"
```

```
        app:cornerRadius="5dp"
```

```
        app:icon="@drawable/baseline_add_a_photo_24" />
```

```
<!--      <com.google.android.material.button.MaterialButton-->
```

```
<!--      android:id="@+id/identify"-->
```

```
<!--      android:text="Identify"-->
```

```
<!--      app:cornerRadius="5dp"-->
```

```
<!--      android:layout_weight="1"-->
```



```

<!--      android:layout_marginEnd="5dp"-->
<!--      android:layout_width="match_parent"-->
<!--      android:layout_height="match_parent" />-->

    <com.google.android.material.button.MaterialButton
        android:id="@+id/recognizeTextBtn"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginEnd="5dp"
        android:layout_weight="1"
        android:text="Recognize Text"
        app:cornerRadius="5dp"
        app:icon="@drawable/baseline_document_scanner_24" />

</LinearLayout>

<!-- to show the rechognize text -->

<!--      <EditText-->
<!--      android:id="@+id/recognizeTextEt"-->
<!--      android:layout_width="match_parent"-->
<!--      android:layout_height="wrap_content"-->
<!--      android:textSize="12dp" />-->

</LinearLayout>

</ScrollView>
</LinearLayout>

```

## **MainActivity.java**

// MainActivity.java

package com.amankr.textrecognition;

import androidx.activity.result.ActivityResult;

import androidx.activity.result.ActivityResultCallback;

import androidx.activity.result.ActivityResultLauncher;

import androidx.activity.result.contract.ActivityResultContracts;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

import android.Manifest;

import android.app.Activity;

import android.app.ProgressDialog;

import android.content.ContentValues;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.net.Uri;

import android.os.Bundle;

import android.provider.MediaStore;

import android.speech.RecognizerIntent;

import android.speech.tts.TextToSpeech;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

```
import android.widget.EditText;

import android.widget.PopupMenu;

import android.widget.Toast;


import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.button.MaterialButton;
import com.google.android.material.imageview.ShapeableImageView;
import com.google.mlkit.vision.common.InputImage;
import com.google.mlkit.vision.text.Text;
import com.google.mlkit.vision.text.TextRecognition;
import com.google.mlkit.vision.text.TextRecognizer;
import com.google.mlkit.vision.text.latin.TextRecognizerOptions;


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


public class MainActivity extends AppCompatActivity {

    private MaterialButton inputImageBtn;
    private MaterialButton recognizeTextBtn;
    private ShapeableImageView imageIv;
    private EditText recognizedTextEt;


    TextToSpeech tts;

    // Tag
    String recognizedText;


    private static final String TAG="MAIN_TAG";
```

```

private Uri imageUri=null;

private static final int CAMERA_REQUEST_CODE=100;
private static final int STORAGE_REQUEST_CODE=101;

private String[] cameraPermissions;
private String[] storagePermissions;

// progress dialog
private ProgressDialog progressDialog;
boolean f=false;

private TextRecognizer textRecognizer;
private static final int REQUEST_PERMISSION_CODE = 1;
SpeakBro obj;
@Override
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    inputImageBtn = findViewById(R.id.inputImageBtn);
    recognizeTextBtn = findViewById(R.id.recognizeTextBtn);
//    identifybtn = findViewById(R.id.identify);
    imageIv = findViewById(R.id.imageIv);

    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setIcon(R.drawable.baseline_home_24);
    getSupportActionBar().setTitle(" HOME");

```

```

// EditText t=findViewById(R.id.recognizeTextEt);

// Button b=findViewById(R.id.voicebtn);


// b.setOnClickListener(new View.OnClickListener() {
//     @Override
//     public void onClick(View view) {
//         tts = new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
//             @Override
//             public void onInit(int i) {
//                 if(i==TextToSpeech.SUCCESS){
//                     tts.setLanguage(Locale.US);
//                     tts.setSpeechRate(1);
//                     tts.speak(t.getText().toString(),TextToSpeech.QUEUE_ADD,null);
//                 }
//             }
//         });
//     }
// });

```

```
obj=new SpeakBro();
```

```
obj.speakGivenText("To take image say open camera",MainActivity.this);
```

```
Thread thread = new Thread(){
```

```
    public void run(){
```

```
        try {
```

```
            sleep(4000);
```

```
        }
```

```
        catch (Exception e){
```

```
            e.printStackTrace();
```

```

    }

    finally {

        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, "Listenning...");

        try {

            startActivityForResult(intent, REQUEST_PERMISSION_CODE);

        } catch (Exception e) {

            e.printStackTrace();

            Toast.makeText(MainActivity.this, "" + e.getMessage(), Toast.LENGTH_SHORT).show();

        }

    }

}

};

thread.start();

// permission for camera,galery
cameraPermissions=new
String[]{Manifest.permission.CAMERA,Manifest.permission.WRITE_EXTERNAL_STORAGE};

storagePermissions=new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE};

progressDialog = new ProgressDialog(this);

progressDialog.setTitle("Please wait");

progressDialog.setCanceledOnTouchOutside(false);

textRecognizer = TextRecognition.getClient(TextRecognizerOptions.DEFAULT_OPTIONS);

inputImageBtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

```

```
        showInputDialogDialog();
    }
});

public boolean onCreateOptionsMenu(Menu menu)
{
    getMenuInflater().inflate(R.menu.main,menu);
    return super.onCreateOptionsMenu(menu);
}

public boolean onOptionsItemSelected(@NonNull MenuItem item)
{
    switch(item.getItemId())
    {
        case R.id.feed:
            Intent i = new Intent(getApplicationContext(),FeedbackActivity.class);
            startActivity(i);
            break;
        case R.id.info:
            Intent intent = new Intent(getApplicationContext(),InfoActivity.class);
            startActivity(intent);
            break;
    }
    return super.onOptionsItemSelected(item);
}

private void recognizeTextFromImage() {
    Log.d(TAG,"recognizeTextFromImage: ");
}
```

```

progressDialog.setMessage("Preparing image.....");

progressDialog.show();

try {

    InputImage inputImage=InputImage.fromFilePath(this,imageUri);

    progressDialog.setMessage("Recognizing text....");

    Task<Text> textTaskResult= textRecognizer.process(inputImage).addOnSuccessListener(new
    OnSuccessListener<Text>() {

        @Override

        public void onSuccess(Text text) {

            progressDialog.dismiss();

            recognizedText=text.getText();

            Log.d(TAG,"onSuccess: recognizedText: "+recognizedText);

            // Toast.makeText(MainActivity.this, recognizedText, Toast.LENGTH_LONG).show();
            // obj.speakGivenText(recognizedText,getContext());

            Intent intent = new Intent(getApplicationContext(),ResultActivity.class);

            intent.putExtra("text",recognizedText);

            startActivity(intent);

        }

    })

    .addOnFailureListener(new OnFailureListener() {

        @Override

        public void onFailure(@NonNull Exception e) {

            progressDialog.dismiss();

            Log.d(TAG,"onFailure: ",e);

            Toast.makeText(MainActivity.this, "Failed recognizing text due to "+e.getMessage(),
            Toast.LENGTH_SHORT).show();

```



```

        }

        });

        requestStoragePermission();

    }

}

return true;

}

});

}

private void pickImageGallery(){
    Log.d(TAG,"pickImageGallery: ");
    Intent intent=new Intent(Intent.ACTION_PICK);
    intent.setType("image/*");
    galleryActivityResultLauncher.launch(intent);

}

private  ActivityResultLauncher<Intent>   galleryActivityResultLauncher=registerForActivityResult(new
ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {

    @Override

    public void onActivityResult(ActivityResult result) {

        if(result.getResultCode() == Activity.RESULT_OK){

            Intent data=result.getData();

            imageUri=result.getData().getData();

            Log.d(TAG,"onActivityResult: imageUri "+imageUri);

            imageIv.setImageURI(imageUri);

```

```

    }else {

        Log.d(TAG,"onActivityResult: Cancelled ");

        Toast.makeText(MainActivity.this, "Cancelled..", Toast.LENGTH_SHORT).show();

    }

}

});

private void pickImageCamera(){

    Log.d(TAG,"pickImageCamera: ");

    ContentValues values=new ContentValues();

    values.put(MediaStore.Images.Media.TITLE,"Sample Title");

    values.put(MediaStore.Images.Media.DESRIPTION,"Sample DESCRIPTION");

    imageUri=getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI,values);

    Intent intent=new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    intent.putExtra(MediaStore.EXTRA_OUTPUT,imageUri);

    cameraActivityResultLauncher.launch(intent);

}

private   ActivityResultLauncher<Intent>   cameraActivityResultLauncher=registerForActivityResult(new
ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {

    @Override

    public void onActivityResult(ActivityResult result) {

        if(result.getResultCode() == Activity.RESULT_OK){

            Log.d(TAG,"onActivityResult: imageUri "+imageUri);

            imageIv.setImageURI(imageUri);

            recognizeTextFromImage();

        }else{

```

```

        Log.d(TAG,"onActivityResult: Cancelled ");

        Toast.makeText(MainActivity.this, "camera & storage permissions are required",
        Toast.LENGTH_SHORT).show();

    }

}

else {

    Toast.makeText(MainActivity.this, "Cancelled..", Toast.LENGTH_SHORT).show();

}

}

break;

case STORAGE_REQUEST_CODE:{

    if(grantResults.length>0)

    {

        boolean storageAccepted=grantResults[0]==PackageManager.PERMISSION_GRANTED;

        if(storageAccepted){

            pickImageGallery();

        }else {

            Toast.makeText(MainActivity.this, "Storage permissions is required",
            Toast.LENGTH_SHORT).show();

        }

    }

}

}

break;

}

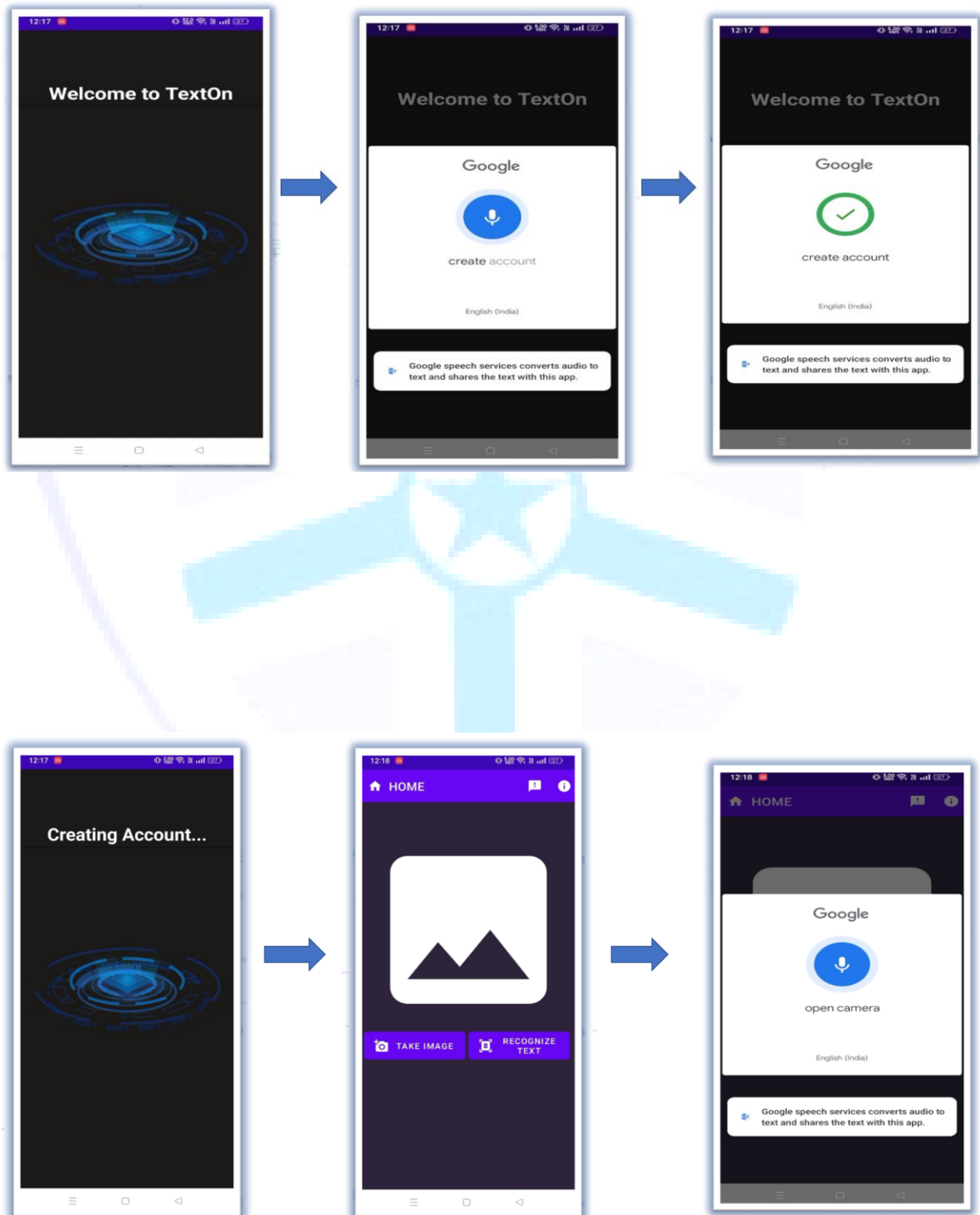
}

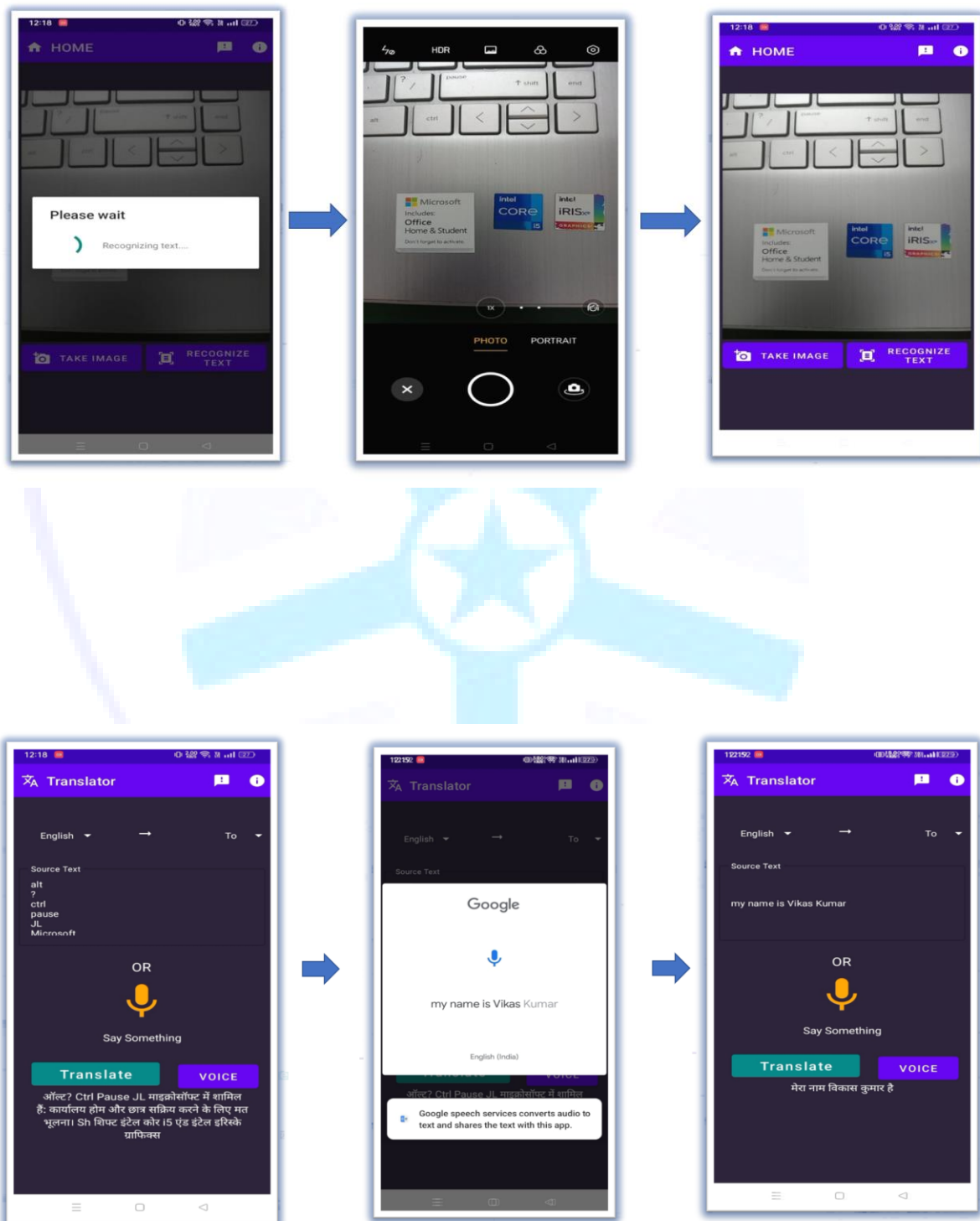
}

```

# RESULT

## SCREENSHOTS





# TESTING

## UNIT TESTING :-

Unit testing is undertaken when a module has been created and successfully reviewed. In order to test a single module we need to provide a complete environment i.e. besides the module we would require

- The procedures belonging to other modules that the module under test calls
- Non local data structures that module accesses
- A procedure to call the functions of the module under test with appropriate parameters

## INTEGRATION TESTING :-

In this type of testing we test various integration of the project module by providing the input.

The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module.

## CONCLUSION & FUTURE SCOPE

In conclusion, our mobile app is a powerful and versatile tool that revolutionizes the way users interact with text and language. Its ability to capture and convert any text into a digital format, while offering seamless translation into multiple languages, empowers individuals to access and understand information effortlessly. Furthermore, the voice-operated playback feature ensures inclusivity for visually impaired users, making the app a truly inclusive solution. Looking ahead, we are committed to enhancing the app's functionality and providing even more value to our users. Future updates will include advanced features such as image labeling, object detection, barcode scanning, and landmark detection. These additions will expand the app's capabilities, enabling users to interact with their environment in innovative ways. Additionally, we will continue to prioritize accessibility and inclusivity by improving the voice-operated feature and incorporating smart reply functionality.

By continuously evolving and updating our mobile app, we aim to meet the diverse needs of our users and remain at the forefront of technology. We believe that through these future updates, our app will continue to positively impact the lives of people around the world, fostering greater understanding and accessibility for all.