Manarat International University

Department of Computer Science & Engineering

Software Development (CSE312)

Team Information:

Team Name - Aid

- Sumaia Siddika-1846CSE00661
- Khawla Kamal-1846CSE00675

Project Name - Personal Assistant Mithu (Android App)

GitHub Repository - https://github.com/SumaiaSiddika/Personal-Assistant-Mithu.git

Promo Video -
https://github.com/SumaiaSiddika/Personal-Assistant-Mithu/blob/master/Personal%20Assistant%20Mithu.mp4

Objective

It shows a very basic logic in which a file is opened by the App automatically whenever a user says to open it. The App uses TextToSpeech API to speak to the user for the inputs it requires. SpeehRecognizer API is used to capture the inputs spoken by the user.

Introduction

Personal assistant is an application program that understands natural language voice commands and completes tasks for the user.

Features Description

This project is designed and constructed as an application that is an android personal assistant, which can control mobile through voice command like checking Gmail, Facebook, YouTube, predict current time & weather, take a photo, check current news and also can read the news, can detect voice for both male & female, also able to custom google search and it can navigate basic function of mobile phone. It will also send a warning message if any app doesn't install yet!
When installing this app, it may require app permission from phone storage.
The functionality is cleared by the methods name.

1. READ_EXTERNAL_STORAGE
2. RECORD_AUDIO
3. WRITE_EXTERNAL_STORAGE
4. textToSpeech
5. speechRecognizer
6. textView
7. intent

Technology

Language - Java
Feature - Speech recognition, Chat assistant, Content control.
Platform - Android Studio
Libraries - Java Libraries

Source Code

1. Code from java (.java file)
2. Code from layout (.xml file)
3. Code from manifests (.xml file)

Code from java (.java file) –

```
package com.example.personalassistantmithu;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.speech.RecognitionListener;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.TextView;

import java.io.File;
import java.io.FileWriter;
import java.util.ArrayList;

import static android.Manifest.permission.READ_EXTERNAL_STORAGE;
import static android.Manifest.permission.RECORD_AUDIO;
import static android.Manifest.permission.WRITE_EXTERNAL_STORAGE;

public class MainActivity extends AppCompatActivity
{
    private SpeechRecognizer speechRecognizer;
    private TextToSpeech textToSpeech;

    private TextView textView;
    private Intent intent;

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

        ActivityCompat.requestPermissions(this, new String[]{RECORD_AUDIO,
WRITE_EXTERNAL_STORAGE, READ_EXTERNAL_STORAGE}, PackageManager.PERMISSION_GRANTED);
```

```
        textView = findViewById(R.id.textView);

        textToSpeech = new TextToSpeech(this, new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {

            }
        });

        intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);

        speechRecognizer = SpeechRecognizer.createSpeechRecognizer(this);
        speechRecognizer.setRecognitionListener(new RecognitionListener() {
            @Override
            public void onReadyForSpeech(Bundle params) {
            }

            @Override
            public void onBeginningOfSpeech() {
            }

            @Override
            public void onRmsChanged(float rmsdB) {
            }

            @Override
            public void onBufferReceived(byte[] buffer) {
            }

            @Override
            public void onEndOfSpeech() {
            }

            @Override
            public void onError(int error) {
            }

            @Override
            public void onResults(Bundle results) {
                ArrayList<String> matches =
results.getStringArrayList(speechRecognizer.RESULTS_RECOGNITION);
                String string = "";
                textView.setText("");    //what is (the text is) appearing in the app
opening
                if (matches != null) {
                    string = matches.get(0);
                    textView.setText(string);

                    if (string.equals("Please open this file")) {
                        createMethod();
                    }
                }
            }

            @Override
            public void onPartialResults(Bundle partialResults) {
```

```
            }

            @Override
            public void onEvent(int eventType, Bundle params) {
            }
        });

    }

    public void startButton(View view) {

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            textToSpeech.speak("Hello, how can I help you?",
TextToSpeech.QUEUE_FLUSH, null, null);
        }

        try {
            Thread.sleep(3000);    //catch the line that I am saying
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }

        speechRecognizer.startListening(intent);
    }

    private void createMethod () {
        File file = new File(Environment.getExternalStorageDirectory() +
File.separator + "PersonalAssistantMithu.txt");

        try {
            if (!file.exists()) {
                file.createNewFile();
            }
            FileWriter fileWriter = new FileWriter(file);
            fileWriter.append("What are you looking for!!!");
            fileWriter.flush();
            fileWriter.close();
        }

        catch (Exception e) {
            e.printStackTrace();
            return;
        }

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            textToSpeech.speak("The text file has been created. Thank you for using
Mithu's service.", TextToSpeech.QUEUE_FLUSH, null, null);
        }
    }
}
```

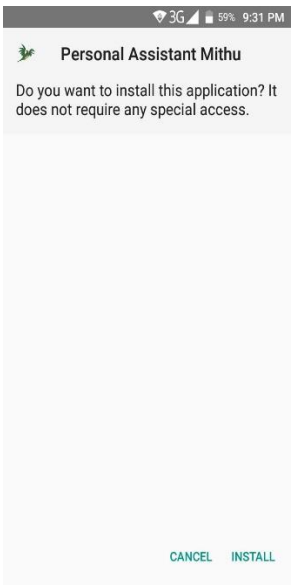
Code from layout (.xml file) –

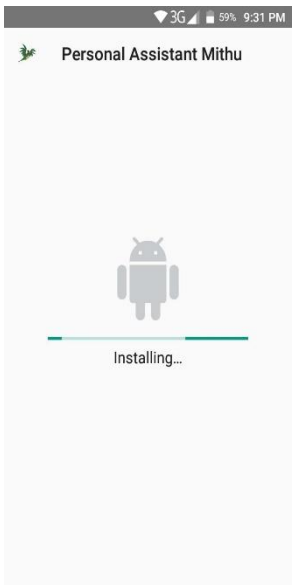
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="149dp"
        android:layout_marginTop="241dp"
        android:layout_marginEnd="149dp"
        android:layout_marginBottom="10dp"
        android:text="@string/welcome_to_mithu"
        android:textColor="#CD2027"
        android:textSize="18sp"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="278dp"
        android:onClick="startButton"
        android:text="@string/start"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

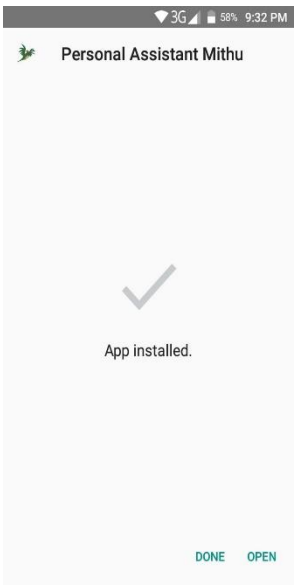
Code from manifests (.xml file) –

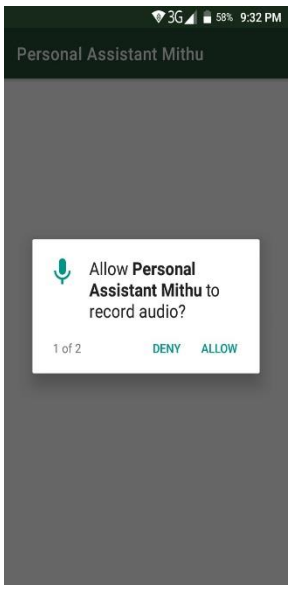
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.personalassistantmithu">

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/mithu_foreground"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/mithu_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.PersonalAssistantMithu">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

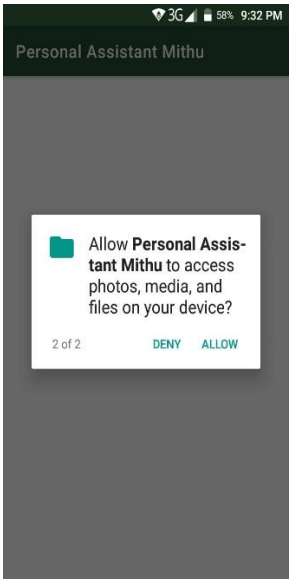
Installation & testing of the App (APK file) in a real phone (screenshots)











Future Plan

Weather forecasting, open websites, take photo, predict current time & weather and almost everything that user ask for.

Conclusion

Voice recognition is a technology which is capable of receiving the worlds that is spoken by user with the help of microphone. These sounds are acknowledged by voice recognizer. The procedure of voice recognition has different steps that will be observed in consequence of one by one. A voice recognition engine is recognized all words pronounced by human but successfully the performance of voice recognition engine gets on number procedure. The major procedures are calculated by different user's noisy environment.