VOICE CALCULATOR

STUDENT NAME : PAVITHRAN K

REGISTER NUMBER : 2018202036

COURSE NAME : MOBILE APPLICATION DEVELOPMENT

COURSE CODE : CA7412

DEPARTMENT NAME : IST

COLLEGE : ANNA UNIVERSITY, CEG CAMPUS.

CONTENTS

I. Introduction

A calculator is a device which allows students to do math operations much easily when compared to manual. For example, most calculators will add, subtract, multiply, and divide. Some also do such scientific operations like square roots, and scientific calculators calculates such complex operations with calculus and draw function graphs. A smartphone or other computer has a calculator. Some calculators, like the abacus, will work without batteries. Others, like the electronic calculator, require batteries. A mobile apps is oriented designed to run on a mobile device such as a mobile phone, watches and tablet. Android application are originally developed for people's assistance such as email, calculator, and contact databases, but the public demand for application areas such as mobile games, automations , GPS and location based applications, ticket booking, and order summary, so that there are now plenty of mobile applications were available. Android apps are generally downloaded from application distribution platforms(like ware house) which are operated by the founder of the operating system developers, such as the Google Play Store. Some apps are free of cost, and others have some cost , with the profit being distributed between the creator of the application and the operating system developer. Mobile applications stand in vice versa to the desktop applications which are designed to run on desktop computers, and web applications which run in mobile web browsers rather than directly on the mobile device. In this mobile application, a voice calculator is used to perform the basic mathematical operation through voice input as well as keyboard user interface. There are two main module in this application. Standard keyboard input system and voice input system. Voice recognition plays a vital role among smartphone package these days, and a corresponding part is the delay while you wait for Siri, Alexa or Google to return your query, either correctly interpreted or horribly mangled. Google's latest update on speech recognition works offline as well as and eliminating that delay altogether though of course mangling is still an option. The delay occurs because your voice breaking, or some data derived from it anyway, has to travel from your phone to the servers of whoever operates the service, where it is analysed and sent back a short time later. This can take anywhere from a handful of milliseconds to multiple entire seconds, or longer if your packets get lost in the ether. Why not just do the voice recognition on the device? There's nothing these companies would like more, but turning voice into text on the order of milliseconds takes quite a bit of computing power. It's really not just about hearing what people said and writing in a word format , its like understanding what someone is saying word by word involves a whole lot of context about language and intention.

II. Requirements

a) Hardware Requirements
   i. Android Mobile phone.
   ii. Laptop (Min 4Gb Ram).
   iii. USB cable.

b) Software Requirements
   i. Android Studio.
   ii. Java Library.

III. Code

a) Opening Splash Screen.

This code displays the logo for few seconds and after that it will automatically redirect into Home Screen.

Game_Splash.java

```
package com.example.firstone;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.WindowManager;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class Game_Splash extends AppCompatActivity {
    private static int SPLASH_SCREEN_TIME_OUT=2000;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.game_splash);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);
        new Handler().postDelayed(() → {
                Intent i=new Intent(getApplicationContext(),
                        MainActivity.class);
                startActivity(i);
                finish();
        }, SPLASH_SCREEN_TIME_OUT);
    }
}
```

b) Home Screen.

Home screen contains tab activity which contains two main activities are voice input and keyboard input controls.

MainActivity.java

```java
package com.example.firstone;

import ...

public class MainActivity extends AppCompatActivity
        implements BottomNavigationView.OnNavigationItemReselectedListener{

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

        BottomNavigationView bt =findViewById(R.id.nav_view);
        bt.setOnNavigationItemReselectedListener(this);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
       getSupportFragmentManager().beginTransaction().replace(R.id.fr1,new Calc()).commit();

    }

    public void onNavigationItemReselected(@NonNull MenuItem menuItem) {

            switch(menuItem.getItemId())
            {
                case R.id.Calc:
                    getSupportFragmentManager().beginTransaction().replace(R.id.fr1,new Calc()).commit();
                    break;
                case R.id.voice:
                    getSupportFragmentManager().beginTransaction().replace(R.id.fr1,new VoiceCalc()).commit();
                    break;
            }

    }
}
```

c) Key Board Type.

In key board input system, we give operators and operands through keyboard.

Calc.java

```java
        n1 = view.findViewById(R.id.one);
        n2 = view.findViewById(R.id.two);
        n3 = view.findViewById(R.id.three);
        n4 = view.findViewById(R.id.four);
        n5 = view.findViewById(R.id.five);
        n6 = view.findViewById(R.id.six);
        n7 = view.findViewById(R.id.seven);
        n8 = view.findViewById(R.id.eight);
        n9 = view.findViewById(R.id.nine);
        n0 = view.findViewById(R.id.z1);
        n00 = view.findViewById(R.id.z2);
        add = view.findViewById(R.id.add);
        sub = view.findViewById(R.id.sub);
        mul = view.findViewById(R.id.mul);
        div = view.findViewById(R.id.div);
        eql = view.findViewById(R.id.eq);
        per = view.findViewById(R.id.per);
        ac = view.findViewById(R.id.ac);
        t1=view.findViewById(R.id.txt_input);
        t2=view.findViewById(R.id.txt_result);
        dot = view.findViewById(R.id.dot);
        clear = view.findViewById(R.id.clear);
        ac = view.findViewById(R.id.ac);
        n1.setOnClickListener((v) -> { t2.setText(t2.getText()+"1"); });
        n2.setOnClickListener((v) -> { t2.setText(t2.getText()+"2"); });
        n3.setOnClickListener((v) -> { t2.setText(t2.getText()+"3"); });
        n4.setOnClickListener((v) -> { t2.setText(t2.getText()+"4"); });
        n5.setOnClickListener((v) -> { t2.setText(t2.getText()+"5"); });
        n6.setOnClickListener((v) -> { t2.setText(t2.getText()+"6"); });
        n7.setOnClickListener((v) -> { t2.setText(t2.getText()+"7"); });
        n8.setOnClickListener((v) -> { t2.setText(t2.getText()+"8"); });
        n9.setOnClickListener((v) -> { t2.setText(t2.getText()+"9"); });
        n0.setOnClickListener((v) -> { t2.setText(t2.getText()+"0"); });
```

d) Voice Input Type.

In voice input, we have to speech through the microphone, the operators and operands through voice to mobile.

VoiceCalc.java

```
package com.example.firstone;
import ...
public class VoiceCalc extends Fragment {
    public ImageView mic; public TextView f1,f2;
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {...}

    private void speech() {
        Intent i=new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        i.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        i.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
        i.putExtra(RecognizerIntent.EXTRA_PROMPT, value: "Say Something");
        try
        {
            startActivityForResult(i, requestCode: 2000);
        }
        catch (Exception ex)
        {
            Toast.makeText(getActivity(), text: "Speech Not Available",Toast.LENGTH_LONG).show();
        }
    }
    @Override
    public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode==2000 && resultCode==RESULT_OK && data!=null)
        {
            ArrayList<String> result=data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
            f1.setText(result.get(0));
            Log.e( tag: "ARRAY LIST",  msg: "onActivityResult: "+result.toString() );
        }
    }
}

```

e) Processing.

Calc.java

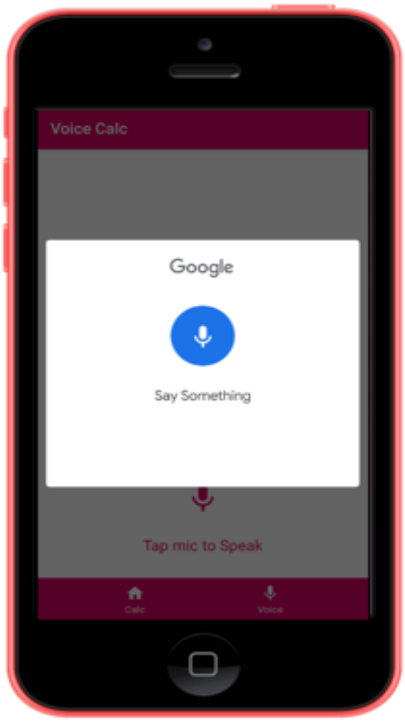
```
        eql.setOnClickListener((v) -> {
                if (a == 0) {
                    num2 = Double.parseDouble( s: t2.getText() + "");
                    t1.setText(t1.getText()+""+t2.getText());t2.setText("");
                    res = num1 + num2;t2.setText(res+ "");a = 1;
                }
                if (s == 0) {
                    num2 = Double.parseDouble( s: t2.getText() + "");
                    t1.setText(t1.getText()+""+t2.getText()); t2.setText("");
                    res = num1 + num2;t2.setText(res+ ""); s = 1;
                }
                if (m == 0) {
                    num2 = Double.parseDouble( s: t2.getText() + "");
                    t1.setText(t1.getText()+""+t2.getText());t2.setText("");
                    res = num1 * num2; t2.setText( res+"");m = 1;
                }
                if (d == 0) {
                    num2 = Double.parseDouble( s: t2.getText() + "");
                    t1.setText(t1.getText()+""+t2.getText()); t2.setText("");
                    t2.setText( res+""); d = 1;
                }
                if (p == 0) {
                    num2 = Double.parseDouble( s: t2.getText() + "")/100;
                    t1.setText(t1.getText()+""+t2.getText());t2.setText("");
                    t2.setText( res+""); p = 1;
                }
            });
            return view;
        }
    }
```

IV. Screen Shots

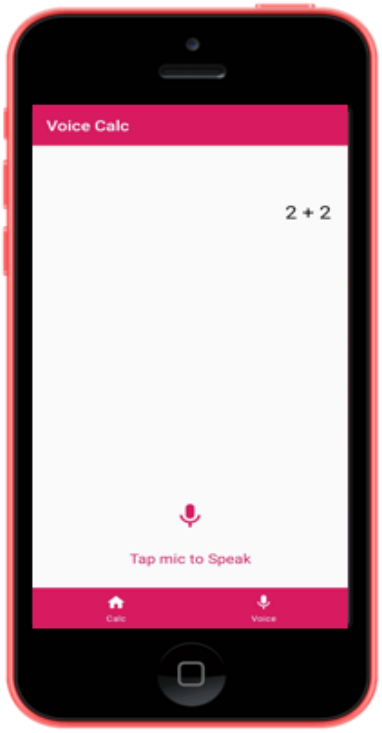
a) Front Screen.

b) Home Screen.

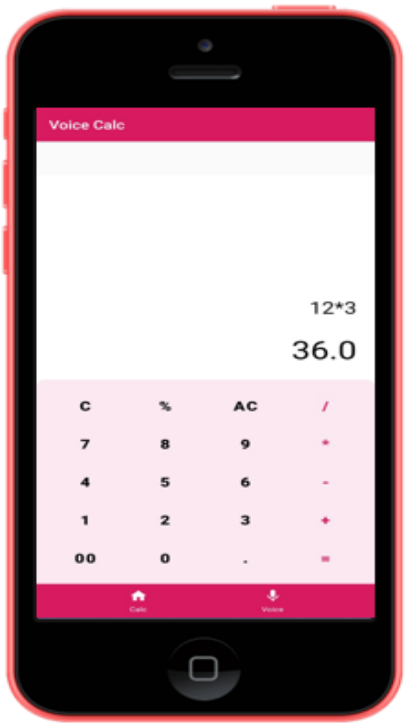
c) Voice Input System.



d) Voice Recognization.



e) Result.



V. Conclusion

In this work we have developed a calculator for exact real number computation and performed a theoretical analysis of the algorithms and experimented on their implementation. A simple user interface has been created to demonstrate the calculator. The interface addresses the problem of allowing the user to view a calculation as it proceeds but also giving an answer which can be used or checked, by outputting signed digits as they are computed and then converting the result into decimal when some specified precision is reached.

VI. Work Extension

In future this kind of calculator may adding features like, compute the days between two dates. For input the data be like, mind reading concepts. Whatever we think on our mind will be computed.