

ICP 13

Speech to Text and Text to Speech

ICP Group 4

Name: Sailaja Narra

Email: sntnn@umsystem.edu

ICP Report:

<https://drive.google.com/file/d/1BUFyllBabit5KijDovyjVWYVObfCM4V/view?usp=sharing>

ICP Video:

<https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3e7f87b4-ca6b-4b19-a5d1-adf40151a2db>

GitHub (Sourcecode):

<https://github.com/UMKC-APL-WebMobileProgramming/ICP13-sailajanarra/tree/main/Source>

My Partner

- Partner: Venkata Mahesh Mokkapati
- Email: vmzwn@umsystem.edu
- Repo: <https://github.com/UMKC-APL-WebMobileProgramming/ICP13-Mahesh68>

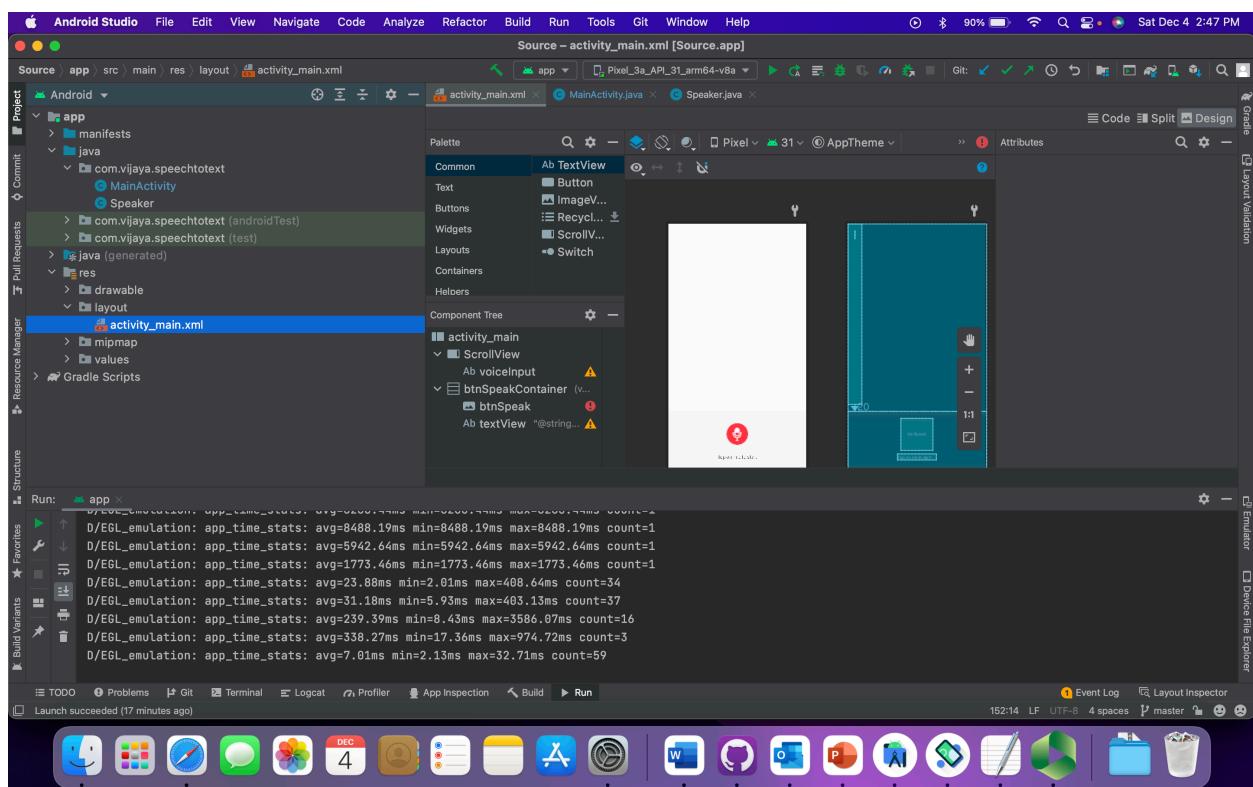
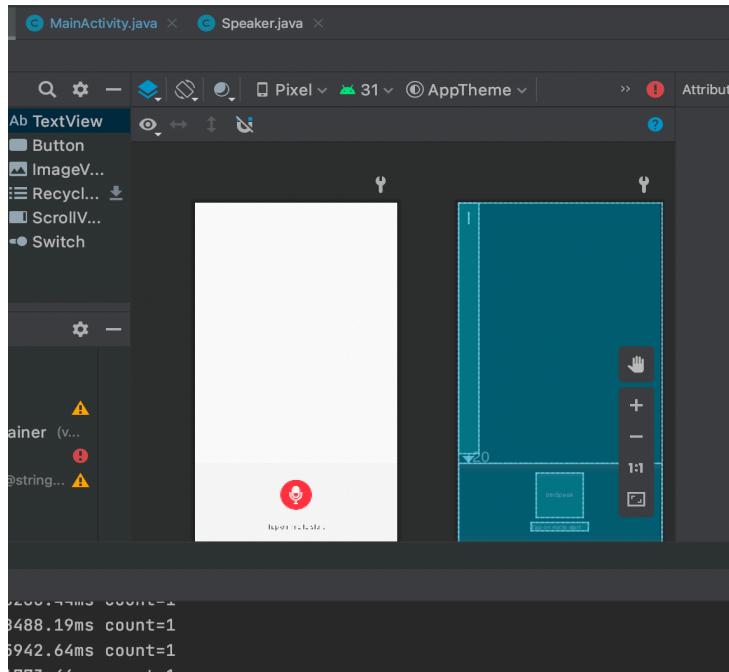
Task:

Creating a medical assistant for suggesting the user some medicines and make this app user friendly.

Initially some source code has been given and asked to make some step by step approach to convert speech to text and text to speech from the application.

1. Use the layout given in the second use case speech to text.

So for this layout is designed as following

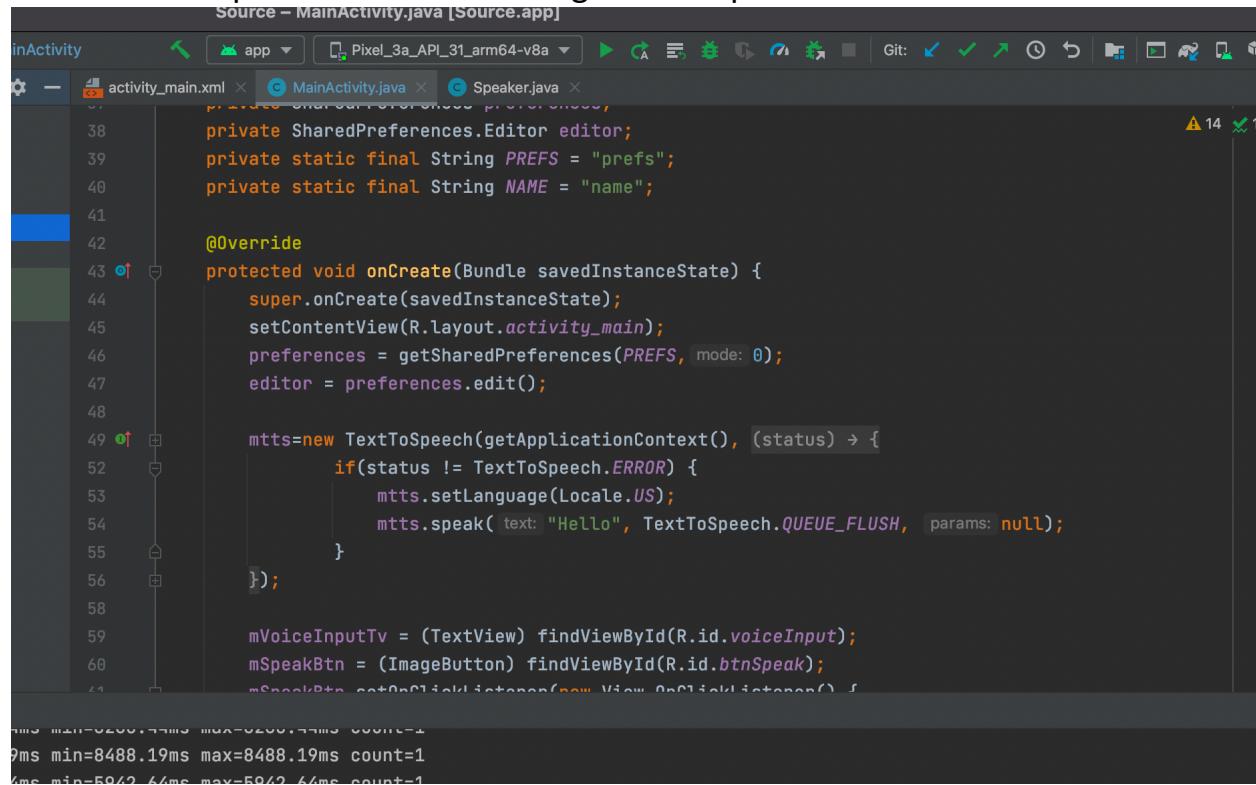


This is given in the source code and made some small improvements.

2. As soon as the app opens, it should say hello

For this we make use of Text to speech API and set the parameters to the desired text output.

So that it will speak the text which is given as input to this API.



The screenshot shows the Android Studio interface with the code editor open. The file is `MainActivity.java`. The code initializes a `TextToSpeech` object and speaks the word "Hello".

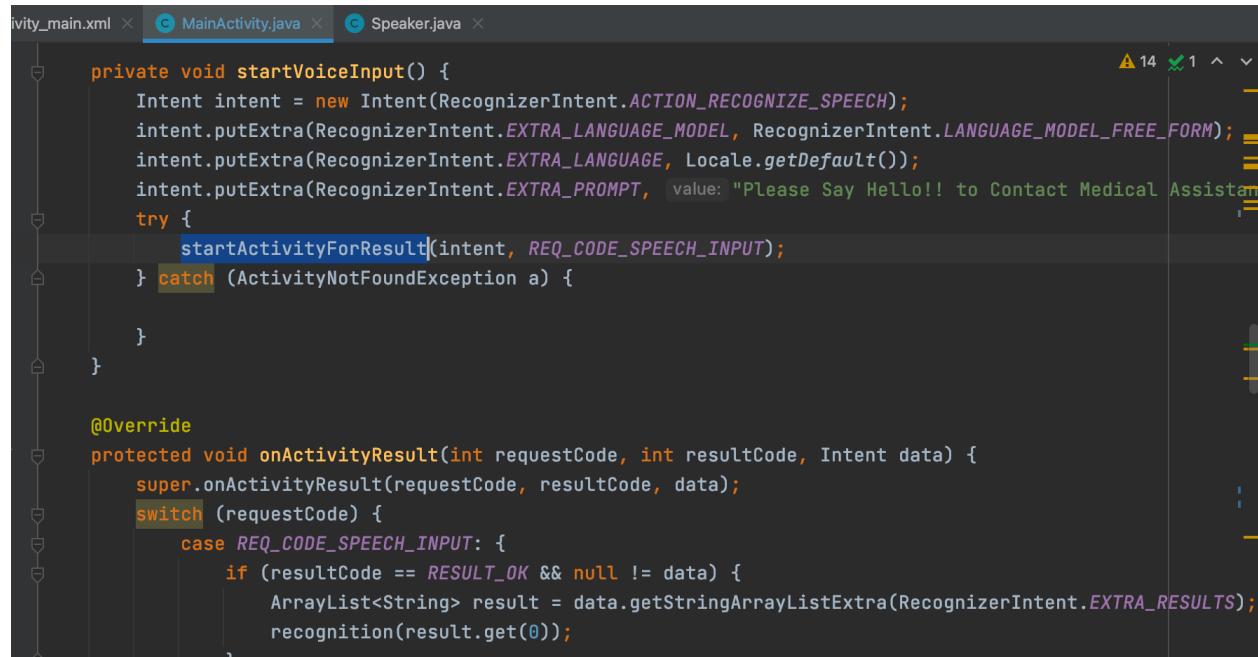
```
Source - MainActivity.java [Source.app]
activity_main.xml MainActivity.java Speaker.java
1  package com.example.source;
2
3  import android.os.Bundle;
4  import android.speech.tts.TextToSpeech;
5  import android.util.Log;
6  import android.view.View;
7  import android.widget.EditText;
8  import android.widget.ImageButton;
9
10 import androidx.appcompat.app.AppCompatActivity;
11
12 public class MainActivity extends AppCompatActivity {
13
14     private EditText mVoiceInputTv;
15     private ImageButton mSpeakBtn;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21         preferences = getSharedPreferences(PREFS, mode: 0);
22         editor = preferences.edit();
23
24         mtts=new TextToSpeech(getApplicationContext(), (status) > {
25             if(status != TextToSpeech.ERROR) {
26                 mtts.setLanguage(Locale.US);
27                 mtts.speak( text: "Hello", TextToSpeech.QUEUE_FLUSH, params: null);
28             }
29         });
30
31         mVoiceInputTv = (EditText) findViewById(R.id.voiceInput);
32         mSpeakBtn = (ImageButton) findViewById(R.id.btnSpeak);
33         mSpeakBtn.setOnClickListener(new View.OnClickListener() {
34             @Override
35             public void onClick(View v) {
36                 String voiceInput=mVoiceInputTv.getText().toString();
37                 Log.d("MSPEAK", "min=8200.7ms max=8200.7ms count=1");
38                 Log.d("MSPEAK", "min=8488.19ms max=8488.19ms count=1");
39                 Log.d("MSPEAK", "min=5942.64ms max=5942.64ms count=1");
40             }
41         });
42     }
43 }
```

3. Then the user clicks on the mic button to say hello

```
mVoiceInputTv = (TextView) findViewById(R.id.voiceInput);
mSpeakBtn = (ImageButton) findViewById(R.id.btnSpeak);
mSpeakBtn.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        startVoiceInput();
    }
});
```

So when the user clicks on the mic button this on click listener is called and voiceinput method will be called.



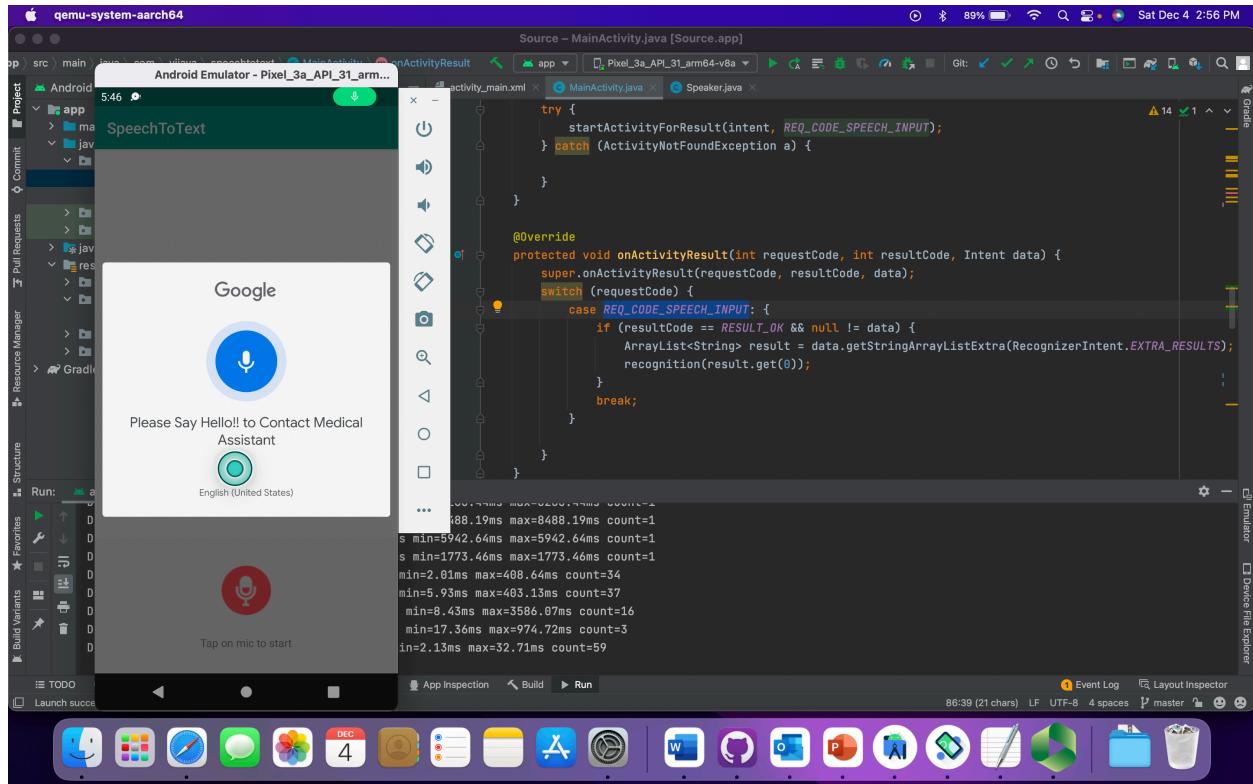
The screenshot shows the Android Studio interface with three tabs: activity_main.xml, MainActivity.java (selected), and Speaker.java. The code in MainActivity.java is as follows:

```
private void startVoiceInput() {
    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, "Please Say Hello!! to Contact Medical Assistant");
    try {
        startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
    } catch (ActivityNotFoundException a) {
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
        case REQ_CODE_SPEECH_INPUT: {
            if (resultCode == RESULT_OK && null != data) {
                ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                recognition(result.get(0));
            }
        }
    }
}
```

So when the user clicks on mic button some text will be shown to the user so that user interaction will happen.

So here we are displaying “Please say hello to medical assistant”



So from here manual interaction with the application takes place and it is as following:

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
        case REQ_CODE_SPEECH_INPUT: {
            if (resultCode == RESULT_OK && null != data) {
                ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                recognition(result.get(0));
            }
            break;
        }
    }
}

```

User interaction has these following words then respective speech will be given from the robo:

```

private void recognition(String text){
    //creating an array which contains the words of the answer
    String name;
    String[] speech = text.split( regex: " " );

    if(text.contains("hello")) {
        mtts.speak( text: "What is your name", TextToSpeech.QUEUE_FLUSH, params: null );
        mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:red;\">User : " +text+"</p>" ));
        mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:green;\">Medical Assistant : What is " ));
    }
    if(text.contains("my name is"))
    {
        name = speech[speech.length-1];
        editor.putString(NAME,name).apply();
        mtts.speak( text: "Your name is "+preferences.getString(NAME, s1: null), TextToSpeech.QUEUE_FLUSH,
        mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:black;\">.... Changing User's name as " ));
        mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:red;\">"+name+": " +text+"</p>" ));
        mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:green;\">Medical Assistant : Your na " ));

    }
    if(text.contains("not feeling well"))
    {
        mtts.speak( text: "I can understand. Please tell your symptoms in short.", TextToSpeech.QUEUE_FLUSH );
        mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:red;\">"+preferences.getString(NAME, s1: null)+" " ));
        mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:green;\">Medical Assistant : I can u " ));
    }
}

```

4.Once the user said hello,the app should speak,"what is your name?"

```

if(text.contains("hello")) {
    mtts.speak( text: "What is your name", TextToSpeech.QUEUE_FLUSH, params: null );
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:red;\">User : " +text+"</p>" ));
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:green;\">Medical Assistant : What is your name ?</p>" ));
}

```

With this code the audio that is extracted from the audio is stored in an array and that is passed to the recognition method.

Here if that text contains any of the hint words then we are letting the API to speak the given text to the user.

5.Then the user clicks on the mic button to say, "My name is"+<your name>

```
if(text.contains("my name is"))
{
    name = speech[speech.length-1];
    editor.putString(NAME,name).apply();
    mtts.speak( text: "Your name is "+preferences.getString(NAME, s1: null), TextToSpeech.QUEUE_FLUSH, params: null);
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:black;\">.... Changing User's name as "+name+"</p>"));
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:red;\">"+name+ ": " +text+"</p>"));
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:green;\">Medical Assistant : Your name is "+preferences.getString(NAME, s1: null)));
}
```

6.Extract the name of the user and then save it as editor level.

```
if(text.contains("my name is"))
{
    name = speech[speech.length-1];
    editor.putString(NAME,name).apply();
```

7.Also,show the name on the screen

```
if(text.contains("my name is"))
{
    name = speech[speech.length-1];
    editor.putString(NAME,name).apply();
    mtts.speak( text: "Your name is "+preferences.getString(NAME, s1: null), TextToSpeech.QUEUE_FLUSH, params: null);
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:black;\">.... Changing User's name as "+name+"</p>"));
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:blue;\">"+name+ ": " +text+"</p>"));
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:yellow;\">Medical Assistant : Your name is "+preferences.getString(NAME, s1: null)));
```

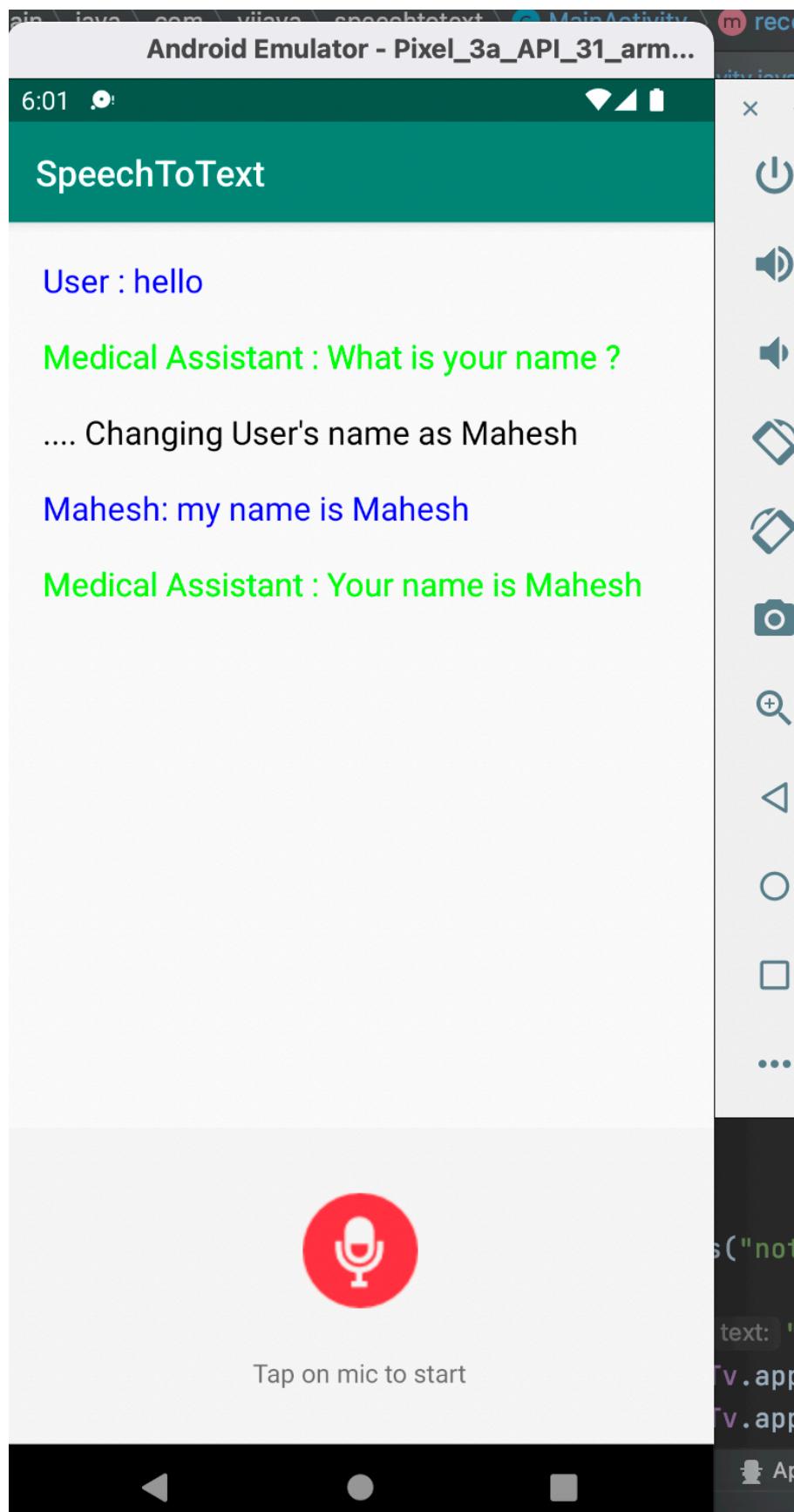
8. When the user asks the following questions, appropriate answers should be given:

```
if(text.contains("not feeling well"))
{
    mtts.speak( text: "I can understand. Please tell your symptoms in short.", TextToSpeech.QUEUE_FLUSH, params: null);
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:blue;\">" + preferences.getString(NAME, s1: null) +
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:yellow;\">Medical Assistant : I can understand. P
}

if(text.contains("time"))
{
    SimpleDateFormat sdfDate = new SimpleDateFormat( pattern: "HH:mm"); //dd/MM/yyyy
    Date now = new Date();
    String[] strDate = sdfDate.format(now).split( regex: ":" );
    if(strDate[1].contains("00")) {
        strDate[1] = "o'clock";
    }
    mtts.speak( text: "The time is " + sdfDate.format(now), TextToSpeech.QUEUE_FLUSH, params: null);
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:red;\">" + preferences.getString(NAME, s1: null) +
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:green;\">Medical Assistant : The time is " + sdfD
}

if(text.contains("medicine"))
{
    mtts.speak( text: "I think you have fever. Please take this medicine.", TextToSpeech.QUEUE_FLUSH, params: null);
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:red;\">" + preferences.getString(NAME, s1: null) +
    mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:green;\">Medical Assistant : I think you have fev
}

if(text.contains("thank you"))
{
    if (text.contains("medical assistant")) {
        mtts.speak( text: "Thank you too " + preferences.getString(NAME, s1: null) + "Take care", TextToSpeech.QUEUE_FLUSH);
        mVoiceInputTv.append(Html.fromHtml( source: "<p style=\"color:red;\">" + preferences.getString(NAME, s1: null) +
    }
}
```



6:02



SpeechToText

User : hello

Medical Assistant : What is your name ?

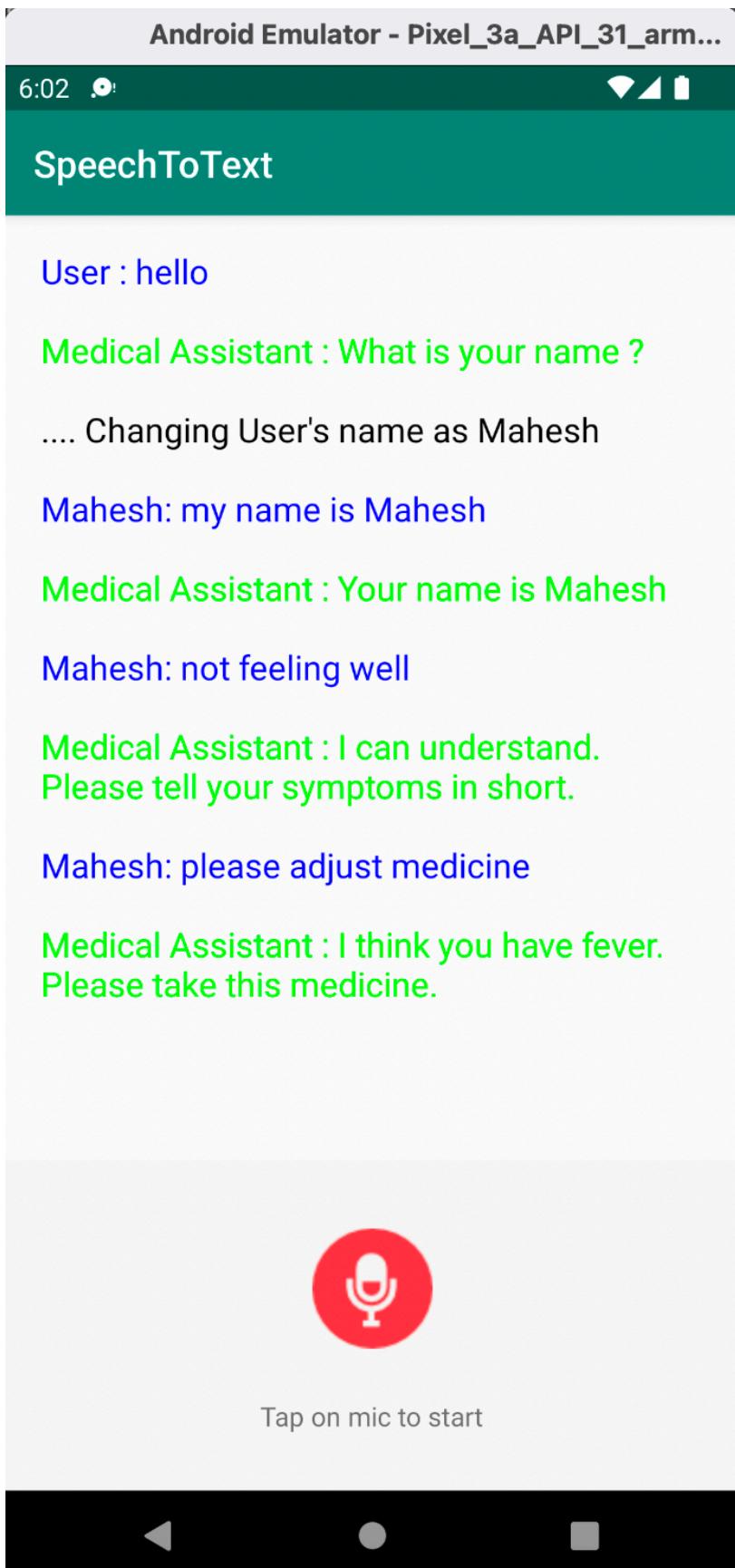
.... Changing User's name as Mahesh

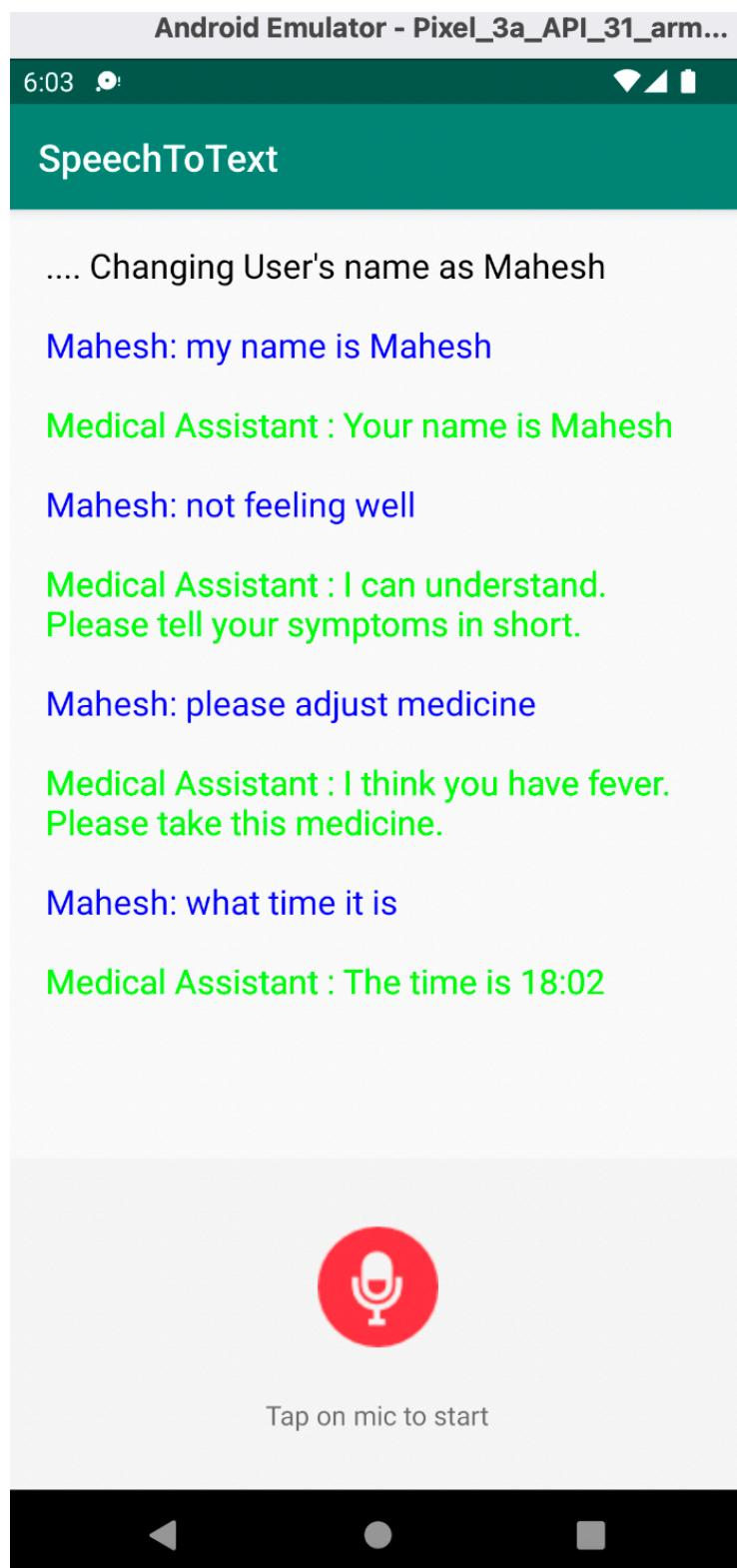
Mahesh: my name is Mahesh

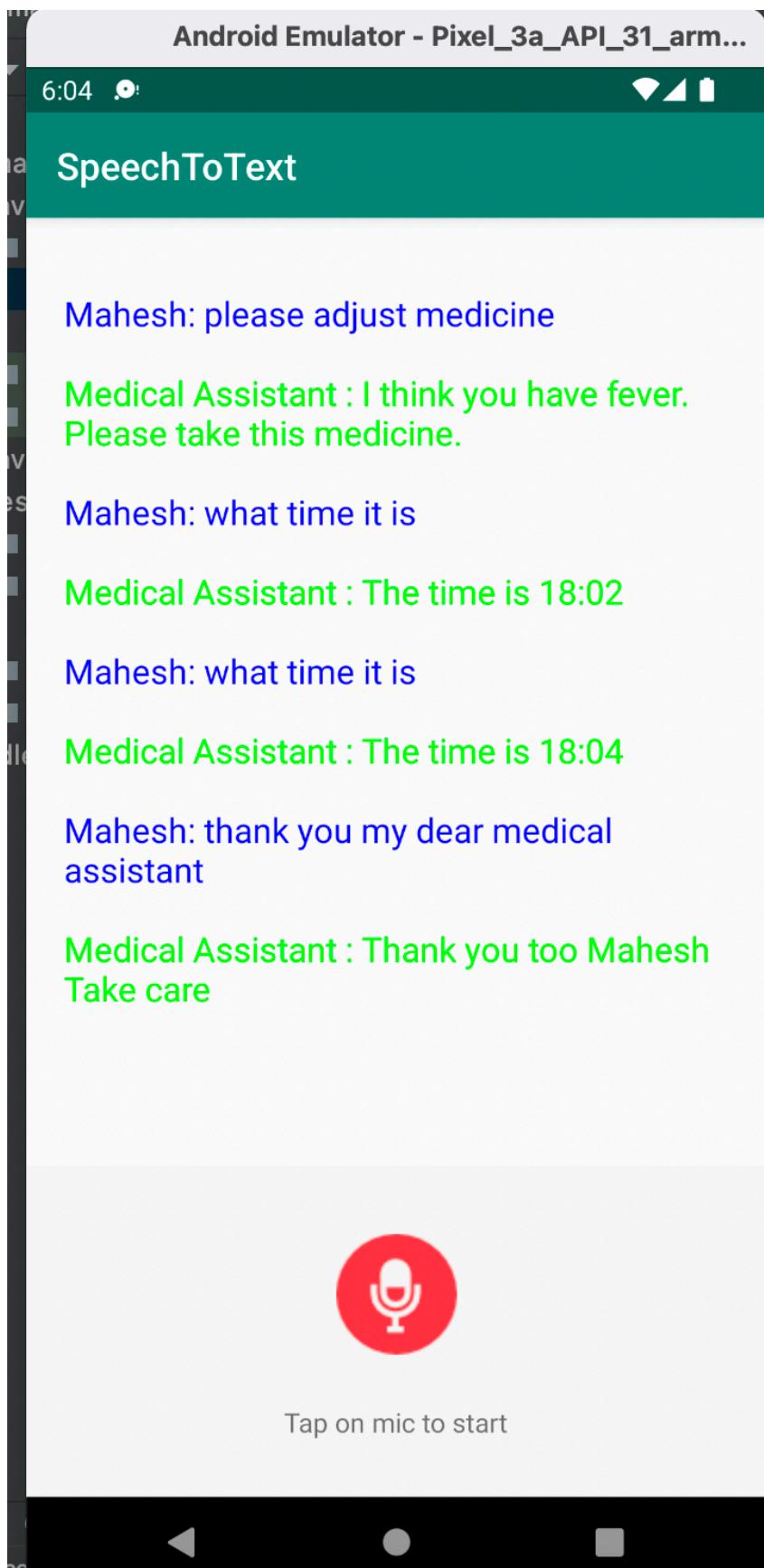
Medical Assistant : Your name is Mahesh

Mahesh: not feeling well

Medical Assistant : I can understand.
Please tell your symptoms in short.







With this tasks are successfully completed for this ICP.

Summary:

We learnt about some of the google voice to text and voice to text APIs and made use of these API in the tasks.

Thank you.