جامعة الإسكندرية
ALEXANDRIA UNIVERSITY
كلية الهندسة
Faculty of Engineering

Final Project Report

Machine Learning On Mobile Devices

Text To Speech Android Application

Names:

- **Youssef hossam eldine aboelwafa** **20012263**
- **Saad El Dine Ahmed Saad Shehata** **7370**
- **Asser Tamer Mohamed** **7363**

Repo link: https://github.com/SaadElDine/Text-To-Speech-Android-App.git

Supervised by:

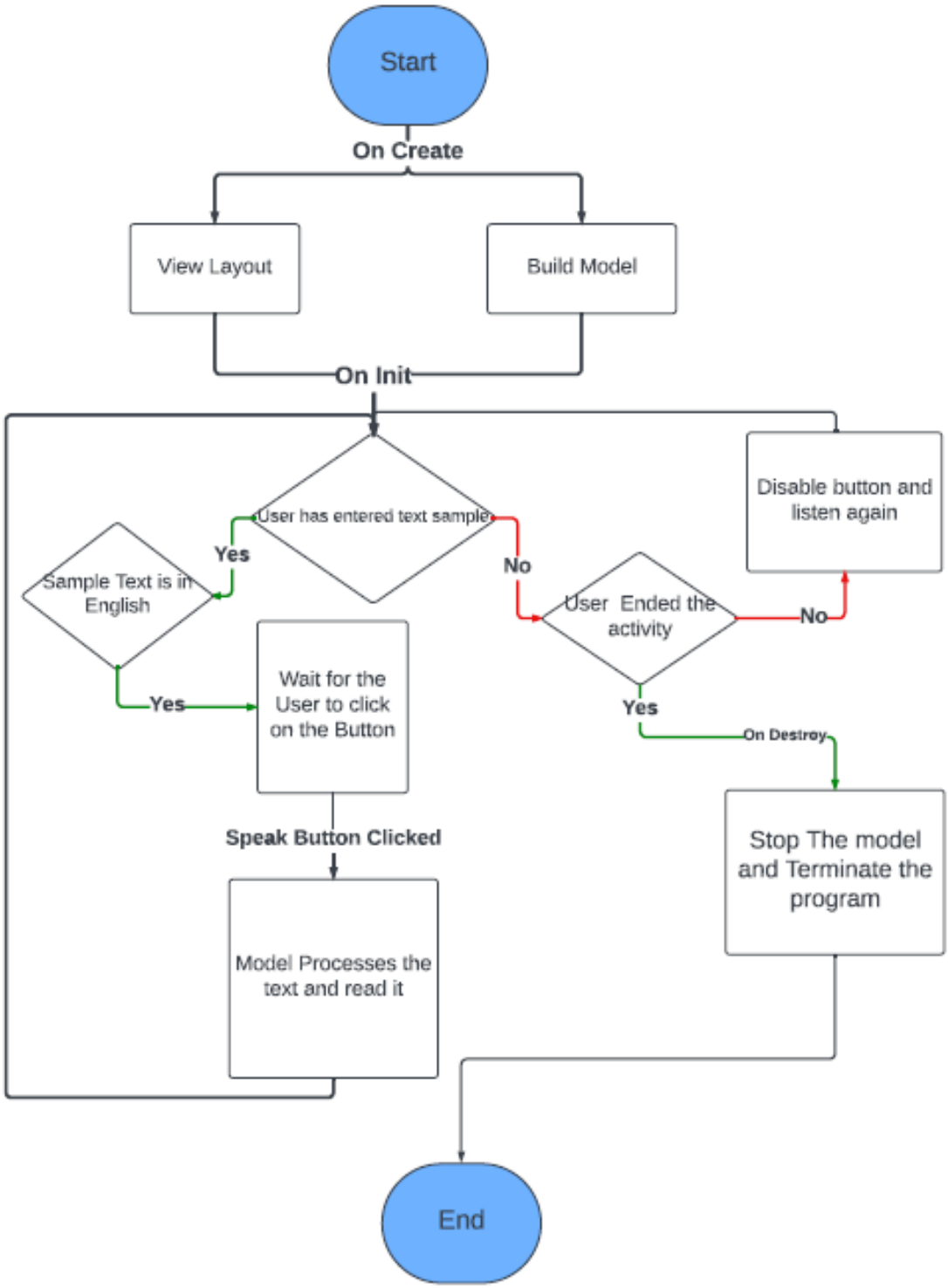
- **Dr. Salah Selim**
- **Eng. Ahmed Waheed**

Description

Text to speech application that take the user input as English text and read it with aid of android text to speech API

Main Functions Used

- **onCreate**
- **onInit**
- **Speak**
- **onDestroy**



Code:

Main Activity:

MainActivity.java

```java
package com.example.text_to_speech;

import ...


public class MainActivity extends AppCompatActivity {

    private TextToSpeech TTS;

    private EditText Text;

    private Button Btn;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Btn = findViewById(R.id.button_speak);

        TTS = new TextToSpeech(this, new TextToSpeech.OnInitListener() {

            @Override
            public void onInit(int status) {
                if (status == TextToSpeech.SUCCESS) {
                    int result = TTS.setLanguage(Locale.ENGLISH);
                    if (result == TextToSpeech.LANG_MISSING_DATA
                            || result == TextToSpeech.LANG_NOT_SUPPORTED) {
                        Toast.makeText(MainActivity.this, "Language not supported", Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText( context: MainActivity.this,  text: "Initialization failed", Toast.LENGTH_SHORT).show();
                }
            }
        });

        Text = findViewById(R.id.edit_text);

        Btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { speak(); }
        });
    }

    1 usage
    private void speak() {
        String text = Text.getText().toString();
        TTS.speak(text, TextToSpeech.QUEUE_FLUSH,  params: null);
    }

    2 usages
    @Override
    protected void onDestroy() {
        if (TTS != null) {
            TTS.stop();
            TTS.shutdown();
        }
        super.onDestroy();
    }
}
```

Functions Explanation:

1. **onCreate(Bundle savedInstanceState)**: This is the method called when the activity is first created. It sets up the initial state of the activity, initializes the user interface, and prepares the TextToSpeech engine. The layout is inflated from the XML layout file "activity_main".
2. **onInit(int status)**: This is the callback method that's called when the TextToSpeech engine initialization is complete. It checks whether the initialization was successful and whether the selected language (English in this case) is supported. If the language is supported, it enables the "Speak" button; otherwise, it shows a Toast message indicating that the language is not supported.
3. **onClick(View v)**: This method is triggered when the "Speak" button is clicked. It calls the **speak()** method to read out the text entered in the EditText.
4. **speak()**: This method retrieves the text entered in the EditText, and then uses the TextToSpeech engine (TTS) to speak out the text using the **speak()** function. The **QUEUE_FLUSH** parameter ensures that any ongoing speech is interrupted and replaced with the new text.
5. **onDestroy()**: This method is called when the activity is being destroyed. It's used to stop and shut down the TextToSpeech engine to release resources and prevent memory leaks.

Sample Run: