Faramarz Zareian
4716747

Speech Recognition Android App

Speech recognition

Speech recognition found in 1950’s which had a focus on numbers not words.

In 1960 IBM introduced a system called “shoebox”, this system could recognize the 16 words in English.

IN 1980’S growth of speech recognition from recognize the hundreds of words improved to recognize thousand of words with “HMM” Hidden Markov Model. Which beside of recognizing the words based on analyzing the sound and voice , it estimated the probability of the unknown sounds actually being words.

Speech recognition

- Speech recognition was used in department of defense of USA at first with a lot of limits , but now it is used in core of android , iOS mobile phones for different uses : security , voice search , voice commands for smart home apps .

Basic Methods

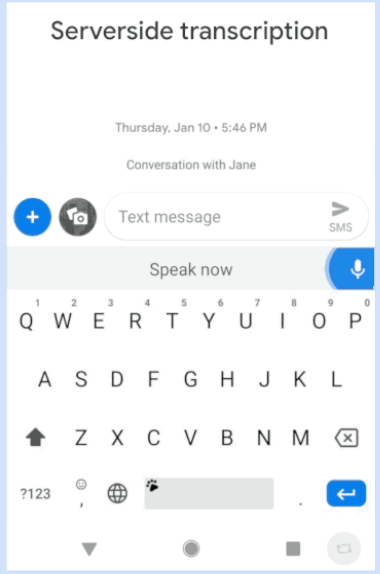
- Two method for speech recognition used in android devices:
- On device :

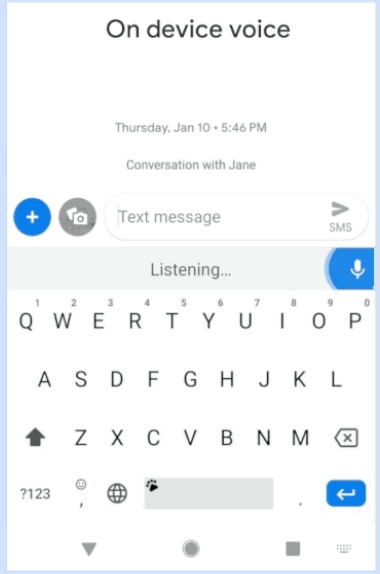
This method uses the offline data which is downloaded before such as dictionaries for specific language user want to use and algorithms which is necessary for recognition.

- On server:

In this method recorded sound/voice compressed and send to the server of google and most of speed is depend on ping and internet speed between user phone and google server .

When we want to compare the speed of recognition between “on device” and “on server” method , we can see the “on device” method is slightly faster because “on server” method data should send to server and process there and results should send back to the user phone .





Streaming End-to-end Speech Recognition For Mobile Devices

- End-to-end (E2E) models, which directly predict output character sequences given input speech, are good candidates for on-device speech recognition. E2E models, however, present numerous challenges: In order to be truly useful, such models must decode speech utterances in a streaming fashion, in real time; they must be robust to the long tail of use cases; they must be able to leverage user-specific context (e.g., contact lists); and above all, they must be extremely accurate. In this work, we describe our efforts at building an E2E speech recognizer using a recurrent neural network transducer. In experimental evaluations, we find that the proposed approach can outperform a conventional CTC-based model in terms of both latency and accuracy in a number of evaluation categories.

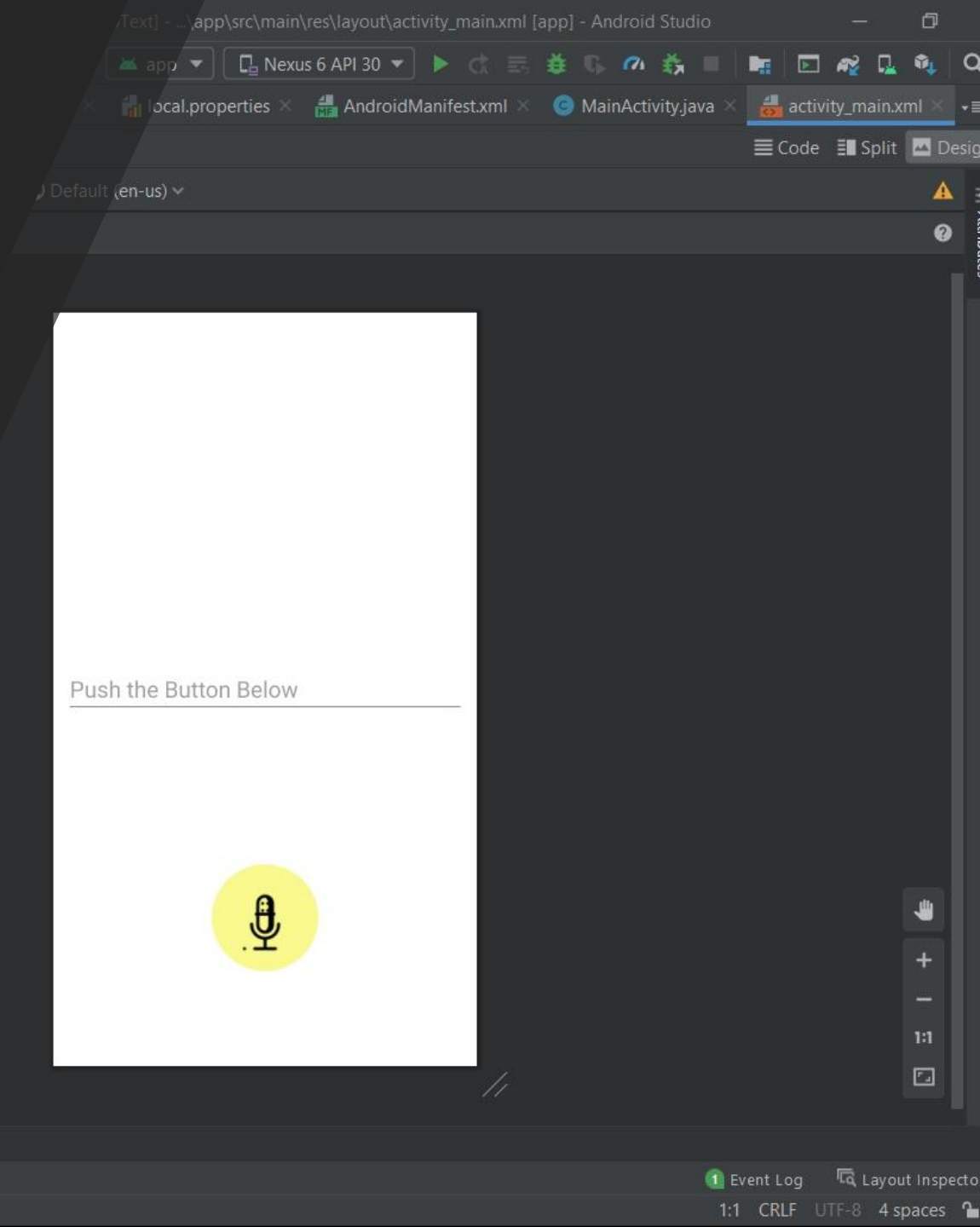
Streaming End-to-end Speech Recognition For Mobile Devices

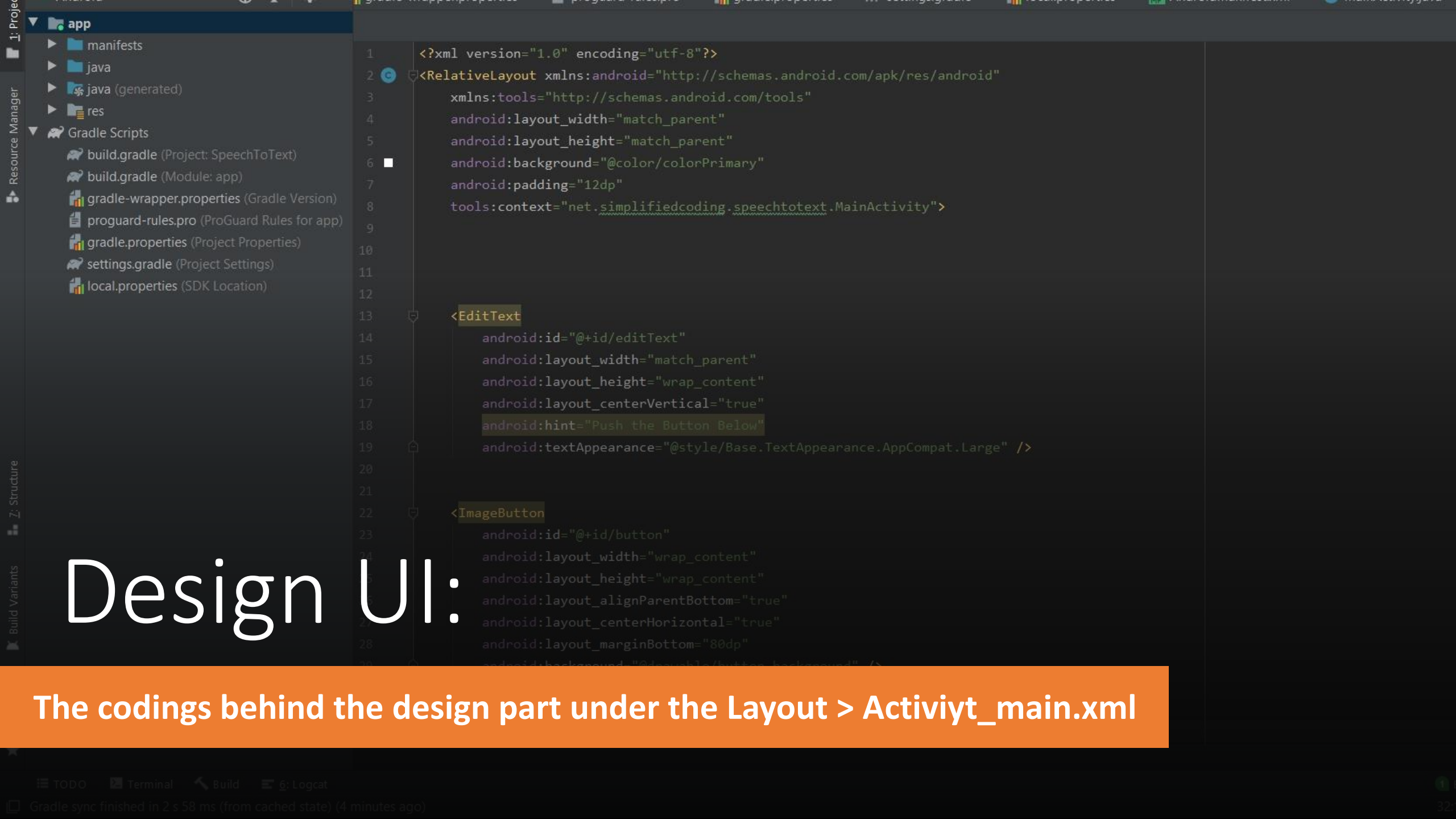
IN THIS MODEL OF SPEECH RECOGNITION USE, WE NEED A VERY GOOD SECURITY AND PRIVACY PROTECTION BECAUSE THE DEVICES ARE LISTENING MOST OF THE TIME AND PERMISSION TO HAVE ACCESS TO STORAGE OR CAMERA OR CONTACTS ARE GRANTED AND THIS MAKE A DEVICE INTO A VERY GOOD TARGET FOR HACKERS OR OTHER ELECTRONIC SABOTEURS.

AND IN THE CODING PART WE MUST HAVE VERY GOOD RATE OF ACCURACY TO PREVENT DO THINGS BY MISTAKE , FOR EXAMPLE SENDING A MESSAGE BY VOICE TO A WRONG PERSON OR PERSON WITH SIMILAR CONTACT NAME WILL CAUSE A SERIOUS PROBLEMS.

Project

- Design UI:
- For this part I used a minimal design which is based on a png button and defined background when there is no touch on the button #not_active mode is yellow
- And when touch is active on the button area #active mode make the background of button change to light color.
- For the text area I used a hint to guide the user.





Design UI:

The codings behind the design part under the Layout > Activiyt_main.xml

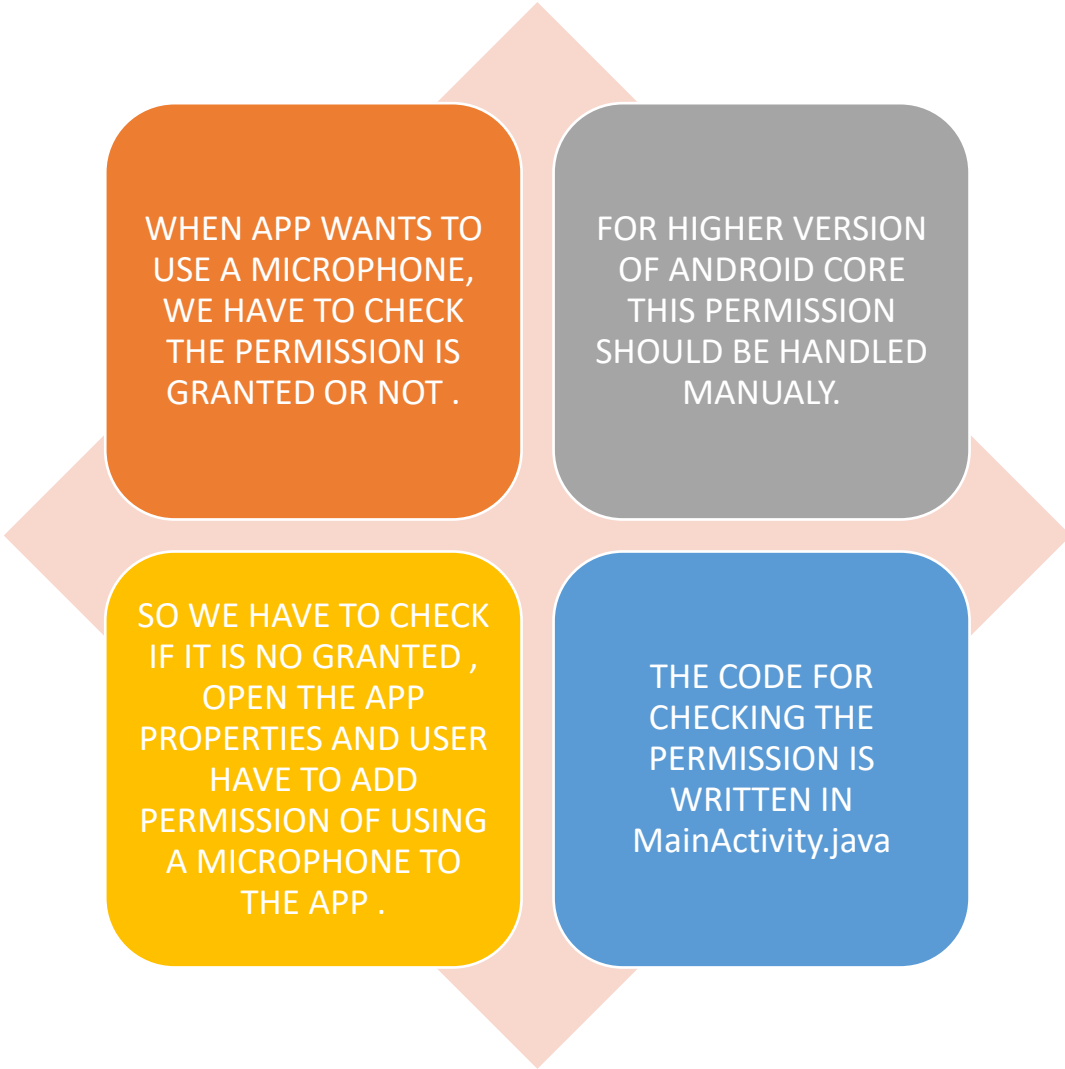
public class SpeechRecognizer.
android.speech.SpeechRecognizer

This class provides access to the speech recognition service. This service allows access to the speech recognizer. Do not instantiate this class directly, instead, call SpeechRecognizer#createSpeechRecognizer(Context). This class's methods must be invoked only from the main application thread.

Permission for Voice Record

- **We add the permission to have access through the button to record the voice , this permission should be written in the AndroidManifest.xml file under the manifests folder.**

CHECK THE PERMISSIONS



RECORDING THE SOUND/VOICE

- For this part we need to import the **RecognitionListener,**
- **Which allows the app to listen to inputs from the microphone and record the voice in buffer.**

OnTouchListener

THIS PART SET THE LISTENING BASED ON SITUATION OF BUTTON

WHEN THE BUTTON IS TOUCHED, AND SITUATION IS ACTIVE THE APP ALLOWED TO LISTEN AND RECORD THE INPUT AND CONTINUOUSLY CHECK THE BUTTON SITUATION.

WHEN THE BUTTON IS NOT_ACTIVE ANYMORE APP STOP TO LISTEN AND RECORD THE INPUTS.

WHEN RECORDING IS FINISHED SPEECH.RECOGNIZER CONVERT THE RECORDED DATA (ANALOG TO DIGITAL FROM MICROPHONE) TO ONSCREEN TEXTS.

Second project

- I enjoyed this kind of application but based on my previous experience using the MIT app inventor I tried to use a simple idea to create a speech recognition usage in android app .
- So I created a voice SMS writing app with MIT app inventor 2.
- It is Block-Based programming which is invented by google since 2013
- The apk files are available in the folder of codes.

smssending Screen1 ▾ Add Screen ... Remove Screen Designer Blocks

Blocks

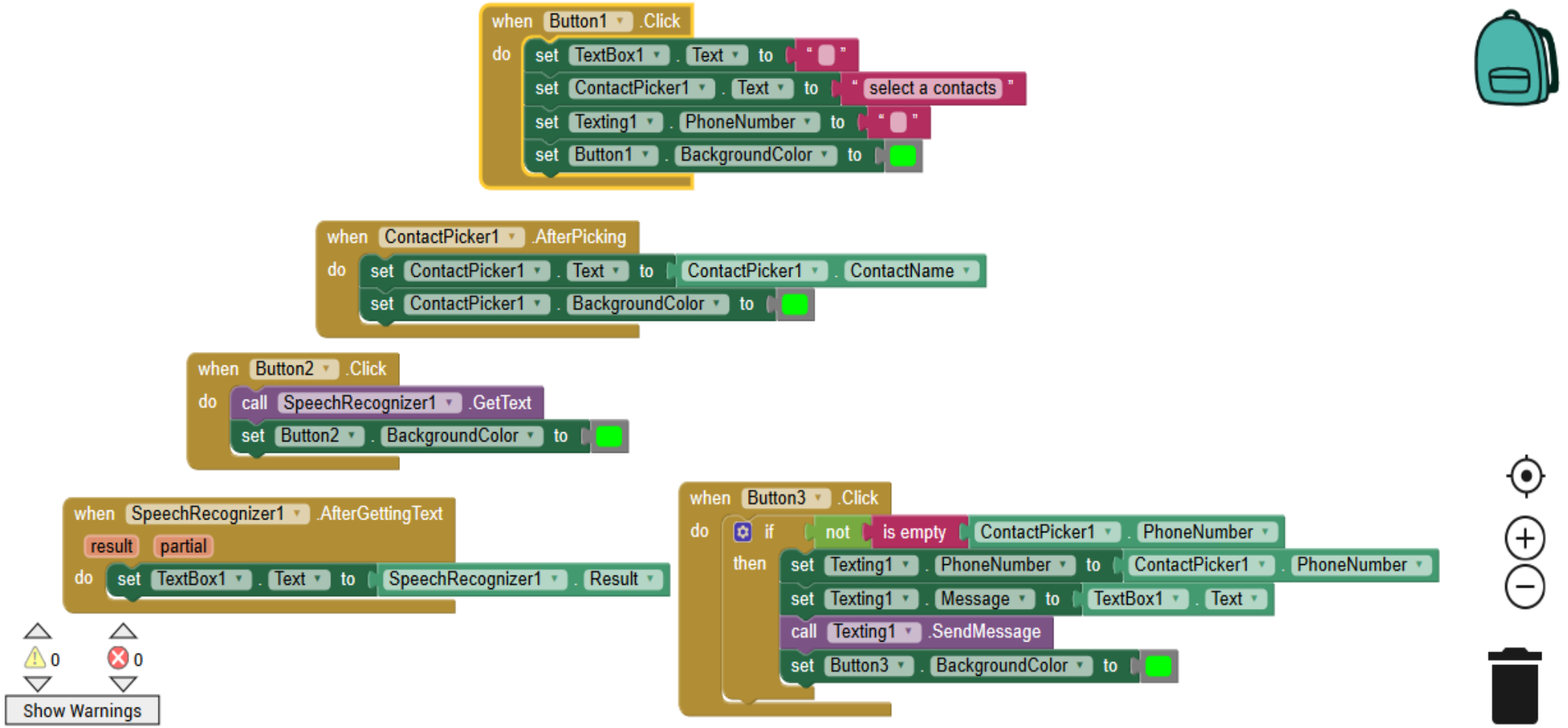
- Built-in
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Dictionaries
  - Colors
  - Variables
  - Procedures
- Screen1
  - TextBox1
  - Button1
  - ContactPicker1
  - Button2
  - Button3
  - SpeechRecognizer1
  - Texting1

Rename Delete

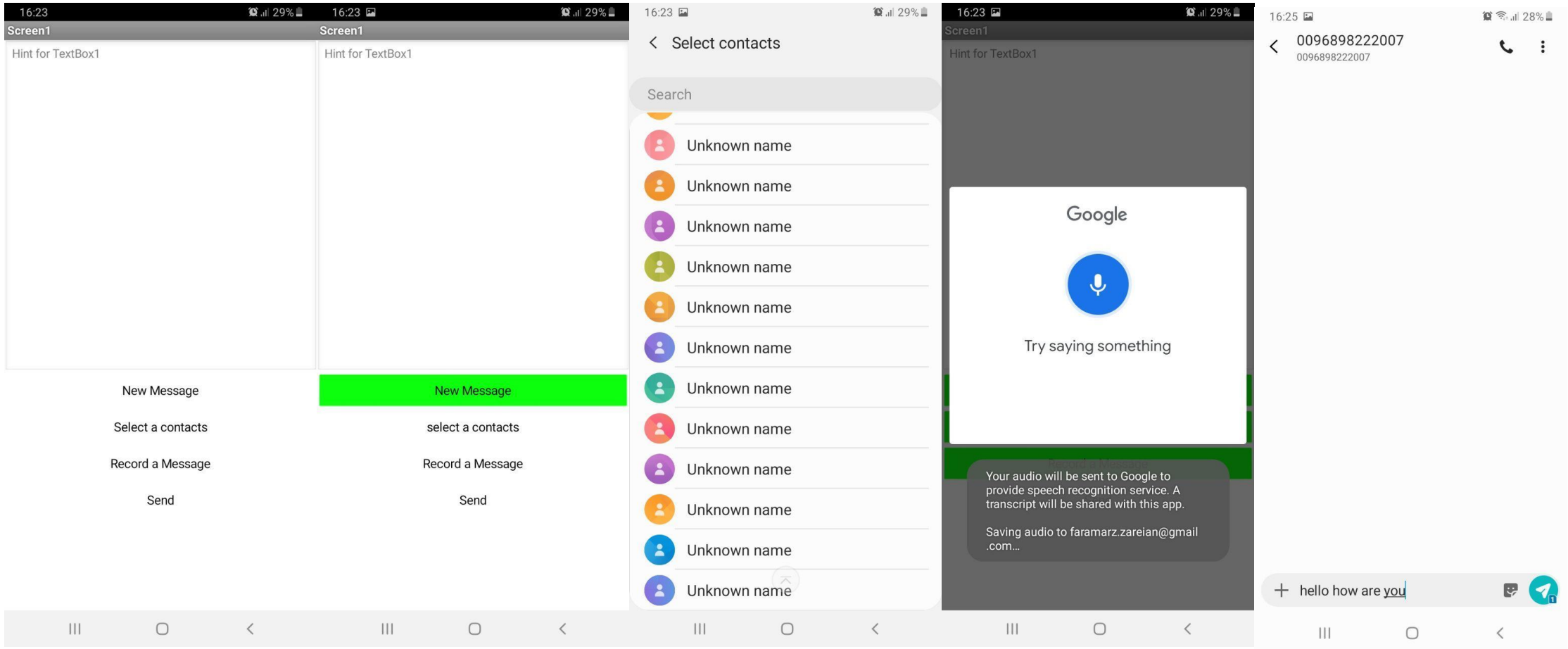
Media

Upload File ...

Viewer



Design of The App and how it works



But the main problem of MIT app inventor is that we cannot *update* the *permissions* in *manifest* file and without it google sees it as Violation of the *Permissions* policy and I had to use a normal send message which forward the text to the phone SMS sending app for sending the text.