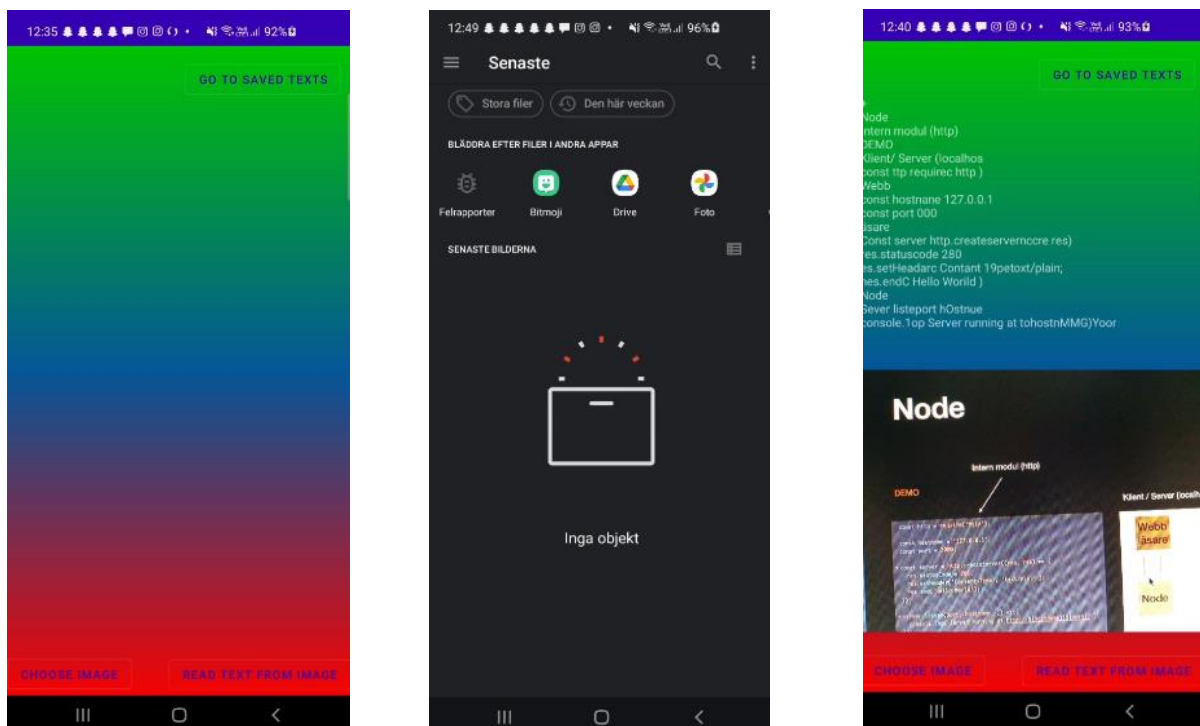


Bild till text till röst applikation – Joacim Ståhl 2021-06-30

Som slutprojekt har jag valt uppgift nummer 3, bild till text till röst:

Bild till röst: Google's Text-to-Speech med Text igenkänning från ML Kit för att läsa upp text från en bild. Ska kunna spara lokal ett viss antal tidigare texter, som kan spelas upp igen.

Startsidan i appen, när appen startas får man en förfrågan om att ge tillåtelse till ens bilder i mobilen. Högst upp till höger kan man klicka in till en annan vy där tidigare sparade texter finns. Knappen "Choose image" gör det den säger, man klickar på den och får då upp alla mappar med bilder man har i mobilen och kan då välja 1 bild. "Read text from image" säger upp texten om det fanns någon text i bilden. Texten som dyker upp från bilden samt bilden är båda två i varsin scrollview, ifall både texten och bild är för stor.



id	Saved_text
1	17:04 tittas... Y 46.1 86%a<linköptlistaChorizoGräddPommes1/10
2	+NodeIntern modul (http)DEMOKlient/ Server (localhostconst ttp requirec http /Webbconst hostname 127.0.0.1const port 000äsaareConst server http.createServer
3	17:04 tittas... Y 46.1 86%a<linköptlistaChorizoGräddPommes1/10
4	Man kan tro att det är snd, men hagar heladagen
5	Man kan tro att det är snd, men hagar heladagen

Databasen som syns ovan finns väl inte så mycket att säga om, sparar både id och texten som tillhör id.

Databas vyn ser ut som nedanför där man kan spela upp sparade texter genom att klicka på play ikonen (pil huvudet bredvid texten). Denna vyn har också en scrollview så man ska kunna se allt innehåll som finns sparad i databasen. Känner man behovet att rensa upp sparade texter så kan man klicka på " Delete Data " som då rensar databasen från all text. "Back" knappen leder tillbaka till första sidan.



kunna motivera och förklara hur ni har anpassat er användning av services efter de rekommendationer som dokumentationerna nämner.

Har använt mig av google ml kit TextRecognizer för att få fram texten ur bilderna, har gått efter googles egen dokumentation som är mycket välgjord.

(URL: <https://developers.google.com/android/reference/com/google/mlkit/vision/text/TextRecognizer>)

```
textRecognizer =
TextRecognition.getClient(TextRecognizerOptions.DEFAULT_OPTIONS);
inputImage = InputImage.fromFilePath(getApplicationContext(), imageUrl);
```

```
Task<Text> result = textRecognizer.process(image);
task.addOnSuccessListener(...).addOnFailureListener(...);
```

Ovanstående kodsnuitt är en kortfattad sammanfattning av hur jag använde koden i projektet utefter dokumentationen.

Har även använt mig av TextToSpeech för att få fram texten i tal, vilken även det har följts utefter dokumentationen gjort av android. Dokumentation för TTS kräver följande:

(URL: <https://developer.android.com/reference/android/speech/tts/TextToSpeech>)

1. Lägga in TTS_Service i manifestet.xml för android 11+
2. .shutdown() och .stop() när man är färdig med tjänsten
3. onInitListener för att bekräfta initialiseringen genomförts.

```
<intent>
```

```
    <action android:name="android.intent.action.TTS_SERVICE" />
```

```
</intent>
```

```
textToSpeech.speak(toRead, TextToSpeech.QUEUE_FLUSH, null);
```

Den tar emot en sträng (toRead), sedan tar man bort tidigare lagda uppspelningar i kön så den nya spelas. Den sista som anges som null är en hashmap parameter.

När programmet startas så aktiveras TTS i onStart lifecycle metoden och när programmet tabbas ur / stängs av så stänger TTS ned sig självt genom onStop metoden.

Jag valde att slänga in initialiseringen an TTS i en egen metod då jag kände att det skulle vara snyggare och mindre "kludd" i lifecycle metoderna.

```
@Override
protected void onStart() {
    super.onStart();
    dbHelper = new DBHelper(context, MainActivity.this);
    startTTS();
}

private void startTTS() {
    textToSpeech = new TextToSpeech(context, this, new TextToSpeech.OnInitListener() {
        @Override
        public void onInit(int status) {
            try {
                textToSpeech.setLanguage(Locale.GERMAN);
                Log.d(log, "tryTts", msg "onInit: " + status);
                // textToSpeech.setLanguage(Locale.ENGLISH);
            } catch (Exception e) {
                e.printStackTrace();
                Log.d(log, "error", msg "onInit: TextToSpeech.ERROR" + TextToSpeech.ERROR);
            }
            Log.d(log, "success", msg "onInit: language is available." + textToSpeech.isLanguageAvailable(Locale.getDefault()));
        }
    });
}
```

```
@Override
protected void onStop() {
    super.onStop();
    Log.d(log, "TAG", msg: "onStop: ");
    textToSpeech.stop();
    textToSpeech.shutdown();
}
```

Lite egna tankar:

Efter att ha suttit med detta projektet så kändes det verkligen som man blev utmanad, speciellt med det helt nya mlkit/ textRecognizer som tog en stund att sätta sig in i. Tidigare övningar i kursen har hjälpt enormt med detta projektet, och hade implementerat sharedPreferences i mainActivity för att spara både texten och bilden om tid hade funnits till detta, just nu rensas detta om man klickar på Back knappen i appen men inte om man backar genom mobilens bakåt funktion, hade nog varit kul att sätta sig in i den biten, om tiden då hade funnits. Även en enskild radera funktion för varje sparad text hade blivit implementerad samt att sidan refreshas automatiskt vid delete så man ej behöver ladda om den manuellt (byta sida fram och tillbaka).

Alla 5 milstolpar är färdiga, det grafiska är väl okej då det är min största svaghet i utveckling av appar i nuläget.

