



Predmet :
Objektno-orijentisano programiranje

Seminarski rad
Tema : “Mobilna aplikacija za prevođenje govora u tekst “

Profesor:
Saša Salapura

Student :
Elvir Lović ,217-18/RITP-s

Banja Luka, 10. 2019.

Sadržaj

Uvod.....	1
1. Razvojno okruženje	2
2. Opis razvoja aplikacije	3
2.1 Fajlovi	3
2.2 Korisnički interfejs	4
2.3 Kod aplikacije	5
2.3.1 Kod korisničkog interfejsa aplikacije	5
2.3.2 Kod logike aplikacije	8
2.4 Prikaz korištenja aplikacije.....	11
3.Zaključak	14
4. Reference	15
5.Spisak slika	15

Uvod

Ovaj rad obuhvata postupak izrade Android aplikacije. Koristeći aplikaciju, korisnik snima svoj glas koji se konvertuje u tekst.

Poznato je da kada se rješava softverski problem, hardverski ili problem kombinacije softvera i hardvera, uvijek treba nastojati da se problem riješi što jednostavnije. Jedan od razloga tome je što će i drugi razvojni programeri koristiti to rješenje, biblioteke implementirati u svoj projekat, itd. Rješenje problema treba biti što efektivnije. Isto tako, kada se riješi srž problema, potrebno je analizirati rješenje i nastojati primijeniti rješenje i na druge platforme na što jednostavniji i efektivniji način.

Tako danas imamo tehnologiju kreiranu od strane Google kompanije, nazvanu „Flutter“ koja služi za kreiranje aplikacija za Android i iOS operativne sisteme. Prva stabilna verzija ove tehnologije je izbačena u decembru 2018. godine što je čini relativno novom. Tehnologija se polako proširuje i na web aplikacije a jedna od glavnih karakteristika ove tehnologije je korištenje aplikacije na ranije navedenim operativnim sistemima iz samo jednog baznog koda (engl. „single code base“).

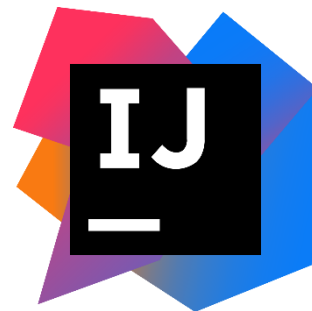
Više nije potrebno kreirati različite aplikacije za oba operativna sistema. Međutim, aplikacija koja je obrađena u ovom radu kreirana je u programskom jeziku Java koja razvojnim programerima pruža odličnu mogućnost korištenja svih usluga koje su potrebne za uspješan rad aplikacije.

1. Razvojno okruženje

Integrisano razvojno okruženje koje je korišteno za izradu aplikacije je Android Studio. Ovo razvojno okruženje je standardno okruženje za razvoj aplikacija za Android operativni sistem. Izgrađeno je od strane JetBrains kompanije za razvoj softvera čiji su alati usmjereni ka razvojnim programerima softvera i projekt menadžerima. Android Studio je izgrađen na bazi IntelliJ IDEA što je integrisano razvojno okruženje programskog jezika Java, i namijenjeno za razvoj kompjuterskog softvera. Android Studio okruženje je dostupno za sljedeće operativne sisteme: Windows, macOS i Linux.



Slika 1 Android Studio Logo



Slika 2 IntelliJ IDEA logo

Android studio također podržava korištenje Fluttera za razvoj aplikacija. Kako biste koristili Flutter, potrebno je instalirati dodatak (engl. plugin). Nakon verzije 3.0 ovog okruženja, programerima je pružena podrška Kotlin programskog jezika. Verzija je puštena u upotrebu u oktobru 2017. godine. Kao što ste do sad mogli primijetiti, ovaj alat je veoma moćan a također je i vrlo jednostavan za upotrebu kada koristite isti duže vrijeme i kada shvatite koncept rada istog.

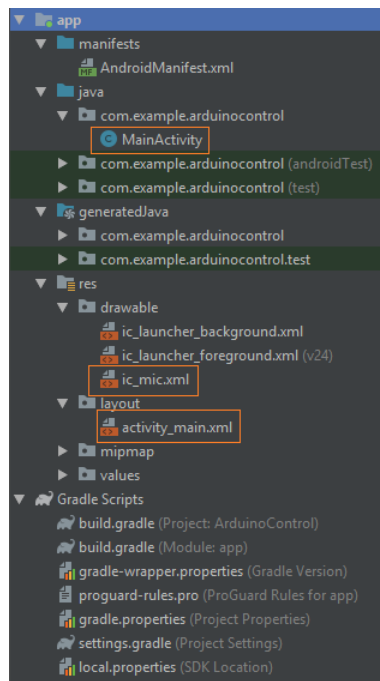
2. Opis razvoja aplikacije

U ovom dijelu rada će biti opisani sljedeći elementi: fajlovi projekta, korisnički interfejs, kod, prikaz korištenja aplikacije, itd.

2.1 Fajlovi

Aplikacija je bazirana za pretvaranja glasa u tekst i ispis na ekranu

Fajlovi će biti prikazani i objašnjeni za prvi dio aplikacije.



Slika 3 Struktura fajlova aplikacije

Na slici iznad je prikazana struktura svih fajlova aplikacije. Međutim, fajlovi koji su modifikovani od strane programera su označeni narandžastim pravougaonikom.

Ti fajlovi su sljedeći:

1. MainActivity (klasa),
2. ic_mic.xml,
3. activity_main.xml.

Fajl pod brojem jedan predstavlja klasu glavne aktivnosti ovog dijela aplikacije. Klasa će biti naknadno objašnjena. Drugi fajl sadrži ikonu koja se sastoji od vektora i prikazuje mikrofona.

Klasa će biti naknadno objašnjena. Drugi fajl sadrži ikonu koja se sastoji od vektora i prikazuje mikrofona.

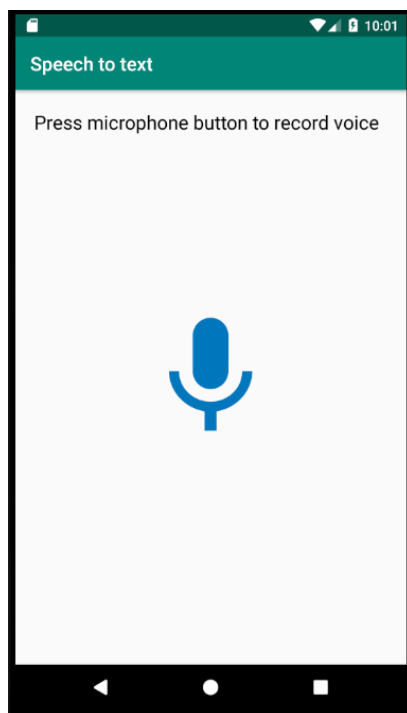


Slika 4 Ikona mikrofona

Posljednji fajl u listi iznad sadrži izgled korisničkog interfejsa prikazan u „xml“ formatu. Isti fajl će biti u nastavku objašnjen.

2.2 Korisnički interfejs

Korisnički interfejs ovog dijela aplikacije je poprilično jednostavan. Sadrži samo jednu labelu sa početnim prikazanim tekstom i ikonu mikrofona koja služi kao dugme kako bi se pokrenuo proces snimanja glasa. Korisnički interfejs je prikazan na sljedećom slici.



Slika 5 Korisnički interfejs

2.3 Kod aplikacije

Kao što je već navedeno, implementacija aplikacije je izvršena u programskom jeziku Java. U nastavku će biti objašnjen kod sljedećih dijelova aplikacije:

- Korisnički interfejs i
- Logički dio koda aplikacije.

2.3.1 Kod korisničkog interfejsa aplikacije

Korisnički interfejs ovog dijela aplikacije je jednostavan. Prikaz interfejsa se može vidjeti na slici 5. Na sljedećoj slici je prikazan prvi dio koda korisničkog interfejsa.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="20dp"
    tools:context=".MainActivity">

```

Slika 6 Prvi dio koda Korisničkog interfejsa

Najčešće se koriste dva načina kreiranja korisničkog interfejsa:

- Koristeći gotove „drag&drop“ alate koji prevlačenjem na formu aplikacije iz palete generišu kod u pozadini. Razvojni programeri Android studija su također kreirali grafički korisnički interfejs kojeg programeri mogu koristiti kako bi modifikovali postavke zadanih elemenata.
- Pisajući kod u „xml“ formatu. Ovaj način je zastario jer postoje gotovi alati koji generišu kod kao što je navedeno ranije. Naravno, po potrebi programer može ručno editovati generisani kod.

Kod na slici 5 prikazuje postavke izgleda aplikacije kao što su: verzija xml fajla, enkodiranje, koji izgled koristi interfejs (trenutno je u pitanju „RelativeLayout“), alati, širina, dužina, itd.

```

<TextView
    android:id="@+id/textTv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Press microphone button to record voice"
    android:textSize="20sp"
    android:textColor="#000"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

Slika 7 Drugi dio koda korisničkog interfejsa

Na slici iznad je prikazan element korisničkog interfejsa „TextView“. Ovaj element ustvari ima svojstvo labele što se može vidjeti u gornjem dijelu slike korisničkog interfejsa (slika 4). Neka od svojstava ovog elementa su sljedeća:

- *id* – predstavlja identifikaciju specifičnog elementa. Služi da bi se mogao element koristiti prilikom pisanja logike aplikacije.

- *text* – predstavlja niz karaktera koji će se nalaziti u elementu. Dozvoljeno je da niz bude prazan. Međutim, bitno je napomenuti sljedeće. Na slici se vidi da je linije koda pod ovom postavkom obavijena žućkastom bojom. Razlog tome je upozoravanje da je taj niz karaktera (u nastavku „string“) pravilnije koristiti iz resursa stringova nego pisati direktno u postavki (engl. hardcoded). Resurs stringova je ništa više nego drugi fajl koji sadrži nazive elemenata (njihove id) i string koji predstavlja tekst tog elementa.
- *veličina teksta*,
- *boja teksta*, itd.

```

    <ImageButton
        android:id="@+id/voiceBtn"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:background="@null"
        android:src="@drawable/ic_mic"
        android:scaleType="fitCenter"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"/>
</RelativeLayout>

```

Slika 8 Treći dio koda korisničkog interfejsa

Slika iznad predstavlja kod posljednjeg elementa ovog dijela aplikacije a to je „*ImageButton*“. Ovaj element predstavlja dugme koje ustvari nije standardnog oblika i postavki nego koristi sliku i njene dimenzije kako bi bila uspostavljena postojanost, izgled, veličina, itd. Neka od svojstava ovog elementa su sljedeća:

- *pozadina* (engl. *background*) – postavka služi da bi se odredila vrijednost pozadine elementa. U ovom slučaju, to je svojstvo „*null*“ koje označava da vrijednost u tom polju ne postoji.
- *izvor* (engl. *source*, prikazano kao „*src*“ skraćenica) – ova postavka služi da bi se odredio izvor slike koju koristi ovaj element. Vrijednost ove postavke dolazi iz resursa „*drawable*“ foldera koji sadrži ikonu mikrofona.
- *tip skaliranja* (engl. *scale type*) – Služi kako bi se odredilo skaliranje i/ili centriranje slike u elementu.

2.3.2 Kod logike aplikacije

Ovaj dio rada je posebno zanimljiv jer prikazuje šta se skriva „ispod haube“ aplikacije i kako ista funkcioniše.

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

import android.content.Intent;
import android.media.Image;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Locale;
```

Slika 9 Prvi dio koda logike aplikacije

Na slici 9 se može vidjeti uključivanje raznih resursa, biblioteka u fajl. Neke od resursa su već poznati kao na primjer: *ImageButton*, *TextView*, *View*, itd. Zanimljivo je da se tu nalaze još i resursi za rad sa listama, resursi za ispis teksta na ekran, itd.

```
public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_CODE_ON_SPEECH_INPUT = 1000;
    //activity views
    TextView mTextTv;
    ImageButton mVoiceBtn;
```

Slika 10 Drugi dio koda logike aplikacije

Na slici 10 je prikazan veoma kratak dio koda koji je jednostavan a svodi se na kreiranje naziva klase pod imenom „*MainActivity*“ koja nasljeđuje klasu „*AppCompactActivity*“ i klasa je isto tako javna. U nastavku se deklarišu varijable i instanciraju pogledi aktivnosti što su ustvari ranije objašnjeni elementi korisničkog interfejsa. Razlog ovome je da se može programabilno, putem logike koda, pristupiti svojstvima elementa i promijeniti ih u odgovarajućem momentu.

```

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

    mTextTv = findViewById(R.id.textTv);
    mVoiceBtn = findViewById(R.id.voiceBtn);

    // show speech to text dialog
    mVoiceBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            speak();
        }
    });
}

```

Slika 11 Treći dio koda logike aplikacije

Na slici 11 je kreirana metoda pod nazivom „*onCreate*“ koja će se pozvati kada korisnik pokrene aplikaciju. Nekoliko od glavnih zadataka metode jeste da sadržaj korisničkog interfejsa postavi na sadržaj fajla koji je prethodno objašnjen. Zatim pronalazi elemente labela i dugmeta putem njihovog *id* svojstva. Posljednje, postaviće proces osluškivanja na element dugmeta čiji je zadatak da pozove metodu „*speak*“ kada se klikne na dugme. Ova metoda je objašnjena u nastavku.

```

private void speak() {
    // intent to show speech to text dialog

    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 speak something");

    //start intent
    try {
        startActivityForResult(intent, REQUEST_CODE_ON_SPEECH_INPUT);
    }
    catch (Exception e){
        Toast.makeText(context: this, text: ""+e.getMessage(), Toast.LENGTH_SHORT).show();
    }
}

```

Slika 12 Četvrti dio koda logike aplikacije

Slika 12 prikazuje logiku metode speak. Metoda je zadužena da instancira namjeru izvršenja akcije i pokrene tu akciju. Svojstva ove akcije su da prikupi govorne podatke korisnika. Jezik govora kojeg će prepoznavati je zadani jezik uređaja. Prilikom pokretanja akcije, korisniku će također biti prikazan tekst sa ciljem da korisnik nešto kaže. U slučaju da se dogodi greška, ista će biti ispisana korisniku u kratkom vremenskom trajanju.

```
// receive input and handle it
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data){
    super.onActivityResult(requestCode, resultCode, data);

    switch(requestCode){
        case REQUEST_CODE_ON_SPEECH_INPUT:{
            if (resultCode == RESULT_OK && null!=data){
                // text array from voice intent
                ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

                //set to textView
                mTextView.setText(result.get(0));
            }
            break;
        }
    }
}
```

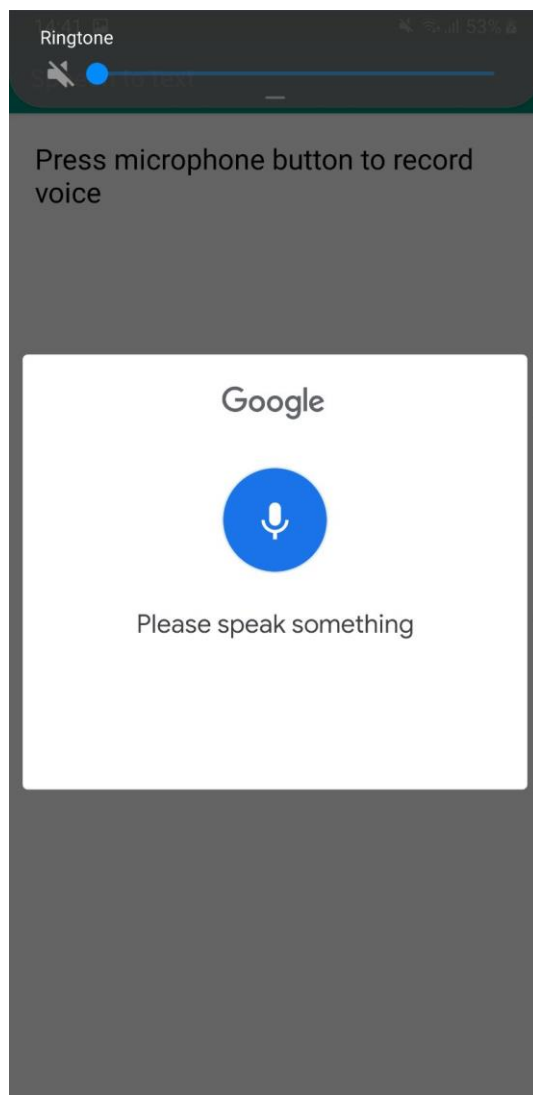
Slika 13 Posljednji dio koda logike aplikacije

Na slici 13 je prikazan dio koda čiji je zadatak da primi korisnički unos, što je u ovom slučaju govor, obradi podatke i prikaže na ekran ako je sve uspješno urađeno. Metoda se izvršava tako što će provjeriti koju je brojčanu vrijednost koda dobila prilikom korisničkog unosa. Ako je kod rezultata bio jednak kodu koji označava da je nešto uredno obavljeno i ako su proslijeđeni neki podaci, isti podaci će se podijeliti u niz a zatim prikazati na mjestu elementa labela korisničkog interfejsa.

2.4 Prikaz korištenja aplikacije

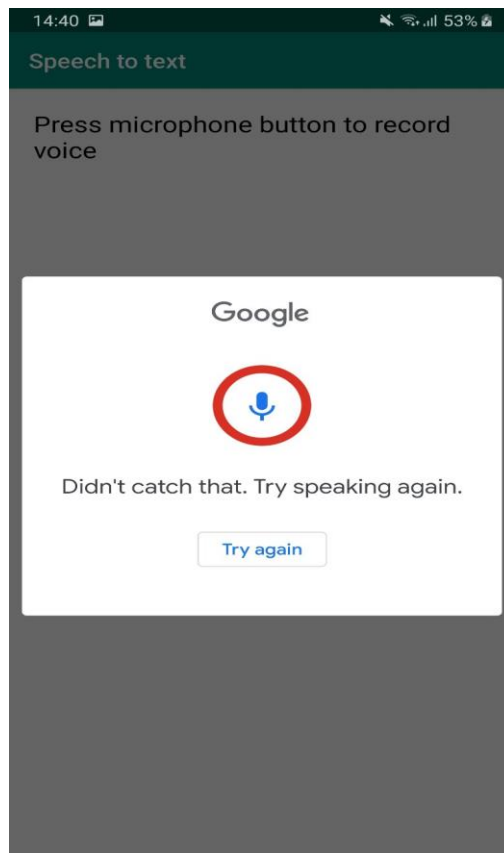
U ovom dijelu rada korisnik će biti upućen u to kako da koristi ovaj dio aplikacije.

Nakon što pokrene aplikaciju, korisniku će biti prikazan interfejs kao na slici 5. Kada korisnik bude imao namjeru da koristi ovaj dio aplikacije, potrebno je da pritisne dugme mikrofona nakon čega će mu se otvoriti prozor kao na sljedećoj slici.



Slika 14 Prvi dio korištenja aplikacije

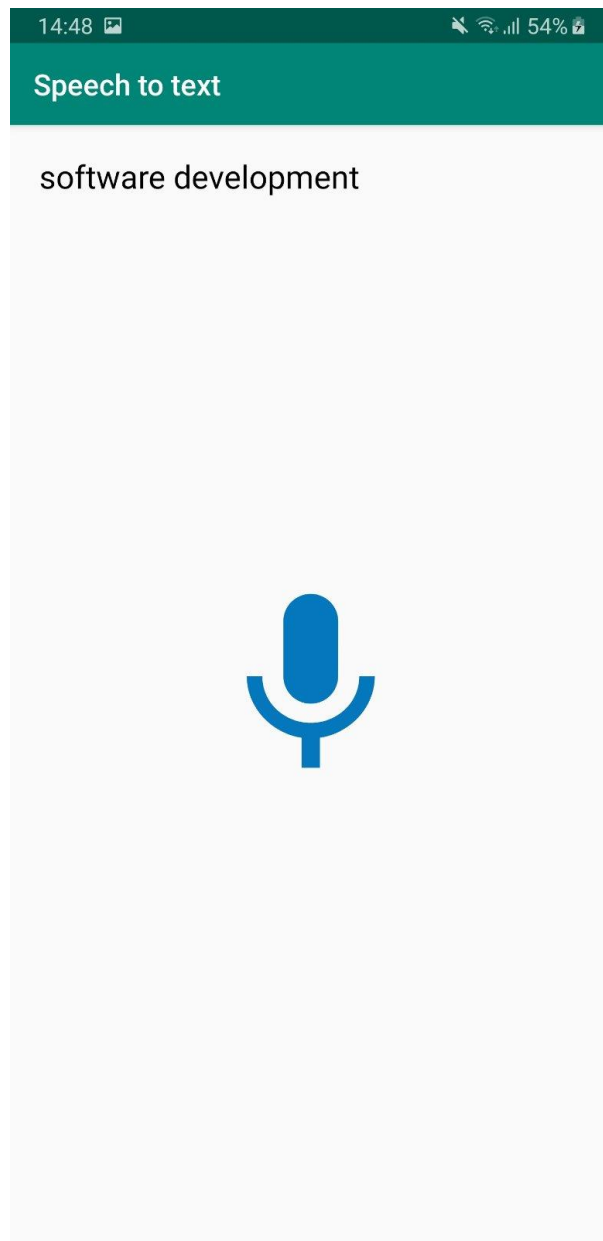
Zamislite da koristite aplikaciju, međutim, kada pokrenete process slušanja vašeg glasa, ustvari ne kažete ništa, tj. ne proslijedite aplikaciji nikakve podatke. U tom slučaju će se pojaviti prikaz kao na sljedećoj slici. Crveni krug će biti prikazan oko mikrofona, korisniku će biti prikazan tekst shodan tome i biće punuđeno dugme koje služi da nakon klika korisnik može pokušati ponovno skeniranje glasa. Klikom bilo gdje na zatamljeni dio aplikacije (izvan trenutnog prozora) ili klikom na dugme povratka, korisnik izlazi iz procesa snimanja glasa.



Slika 15 Prikaz greške snimanja glasa

U slučaju da je korisnik proslijedio podatke aplikaciji, podaci će biti ispisani na trenutnom prozoru u aplikaciji. Međutim, ako korisnik proslijedi podake aplikaciji koji nisu validni za zadani jezik, neće biti prepoznati od strane aplikacije i ništa neće biti prikazano na ekranu.

Nakon što se uspješno izvrši registrovanje i obrada podataka proslijeđena od strane korisnika, obrađen tekst će biti prikazan u polju labela kao što je prikazano na sljedećoj slici.



Slika 16 Prikaz registrovanih podataka u labeli

3. Zaključak

Prvi dio aplikacije je jedan od dva ključna dijela i bavi se registrowanjem, obradom i prikazom registrovanih podataka na korisnički interfejs. Može se primijetiti da prvi dio aplikacije nije kompleksan i da se rješenje problema može implementirati za veoma kratak vremenski period. S obzirom da je i sam programski jezik Java jezik višeg nivoa, implementacija rješenja nije bila kompleksna. Naravno, u zavisnosti od toga koji je opseg projekta i koja je kompleksnost istog, nemoguće je uvijek rješenje svesti na potpuno jednostavno. Međutim, princip se uvijek svodi na to da se nastoji implementirati što jednostavnije rješenje a da je i dalje ispravno i u potpunosti funkcionalno.

4. Reference

1. https://en.wikipedia.org/wiki/Android_Studio
2. <https://developer.android.com/studio>
3. <https://developer.android.com/reference/org/w3c/dom/Element>
4. <https://developer.android.com/guide/topics/appwidgets>
5. <https://developer.android.com/reference/android/content/Intent>

https://www.androidpolice.com/wp-content/uploads/2017/05/nexus2cee_Android-Studio-3.0-hero_thumb-728x409.png

5. Spisak slika

Slika 1 Android Studio Logo	Slika 2 IntelliJ IDEA logo.....	2
Slika 3 Struktura fajlova aplikacije		3
Slika 4 Ikona mikrofona		4
Slika 5 Korisnički interfejs		5
Slika 6 Prvi dio koda Korisničkog interfejsa		6
Slika 7 Drugi dio koda korisničkog interfejsa		6
Slika 8 Treći dio koda korisničkog interfejsa		7
Slika 9 Prvi dio koda logike aplikacije.....		8
Slika 10 Drugi dio koda logike aplikacije		8
Slika 11 Treći dio koda logike aplikacije		9
Slika 12 Četvrti dio koda logike aplikacije		9
Slika 13 Posljednji dio koda logike aplikacije.....		10
Slika 14 Prvi dio korištenja aplikacije		11
Slika 15 Prikaz greške snimanja glasa		12
Slika 16 Prikaz registrovanog glasa u obliku teksta	Error! Bookmark not defined.	
Slika 17 Prikaz registrovanih podataka u labeli		13