



Prolećni semestar, 2018/19

PREDMET: Razvoj mobilnih aplikacija

Projektni zadatak - MovieDB

Ime i prezime: **Marko Rajević**

Broj indeksa: **2720**

Datum izrade: **10.08.2019.**

Uvod

MovieDB predstavlja mobilnu aplikaciju pisanu u JAVI u okviru Android Studio okruženja. Kao dependency menadžer, korišćen je Gradle. Sama svrha aplikacije je orijentisana ka korisniku, gde korisnik jasno može imati uvid u filmove koje je pogledao, kao i ocene koje im je dao. Dakle, korisnik unosi listu filmova, kao i rejting za iste.

Tehnologija aplikacije

Kao što je već navedeno, aplikacija je razvijena u Android Studio okruženju. Za prikaz razvoja korišćen je Google Pixel 3 emulator uređaja.

Aplikacija koristil SQLite bazu podataka, za čuvanje informacija o korisnicima kao i samim filmovima. Kroz samu aplikaciju pokazani su primeri korišćenja CRUD operacija, objektno orijentisanog koda, Geokodiranja, korišćenja servisa telefona kao i manipulacije fotografijama. Napravljen je su entitetske klase koje smo koristili unutar DBHelper klase za manipulaciju podacima prema bazi podataka.

Takođe, postoji i pomoćna InputValidation klasa koja se koristi za validaciju podataka pre čuvanja u bazi podataka.

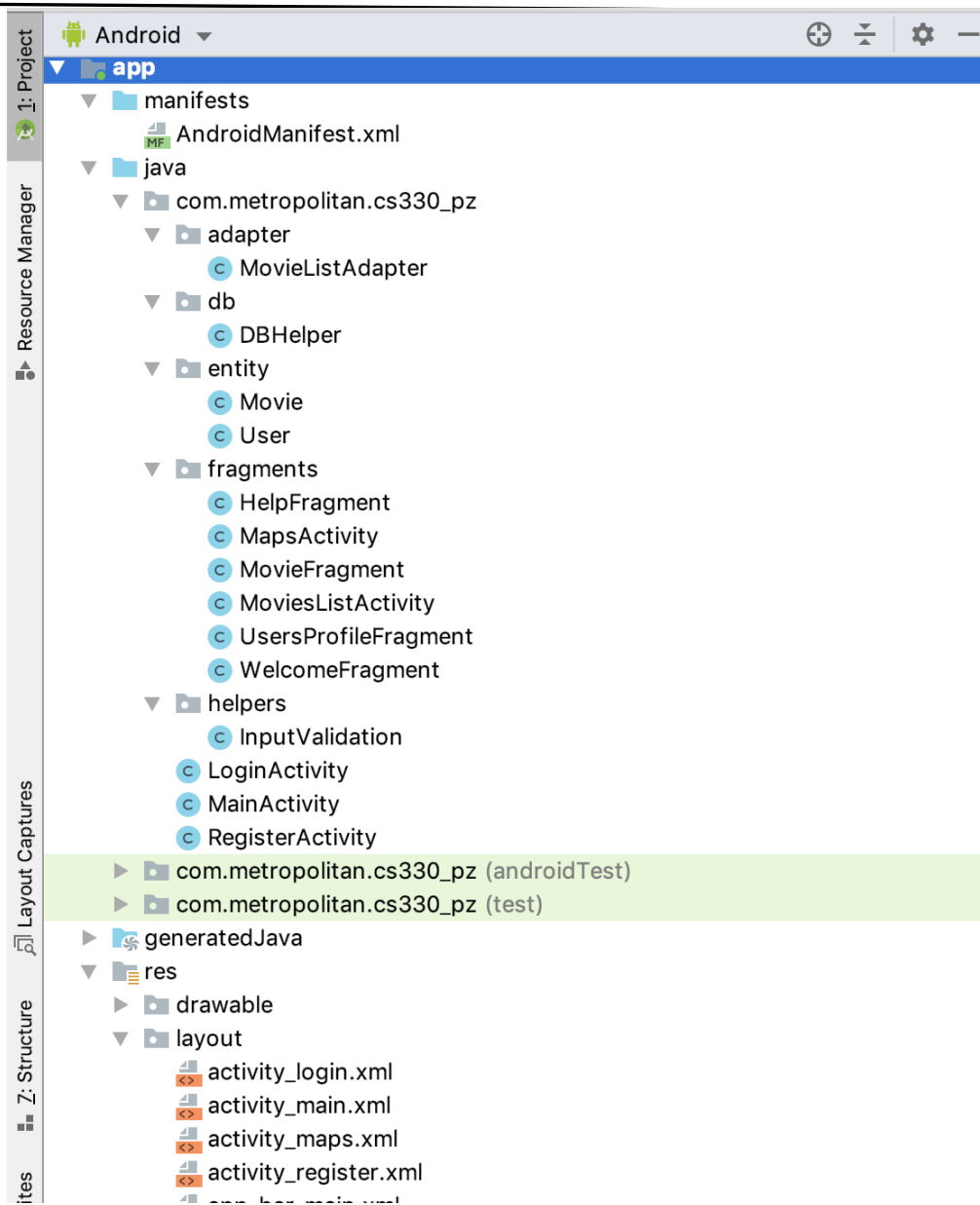
Takođe, korišćen je i Android Design Support.

Implementacija

```
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 28
5      buildToolsVersion "28.0.3"
6      defaultConfig {
7          applicationId "com.metropolitan.cs330_pz"
8          minSdkVersion 15
9          targetSdkVersion 28
10         versionCode 1
11         versionName "1.0"
12         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
13     }
14     buildTypes {
15         release {
16             minifyEnabled false
17             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
18         }
19     }
20 }
21
22 dependencies {
23     implementation fileTree(dir: 'libs', include: ['*.jar'])
24     implementation 'androidx.appcompat:appcompat:1.0.2'
25     implementation 'com.android.support.design:28.0.0'
26     implementation 'de.hdodenhof:circleimageview:3.0.0'
27     implementation 'com.theartofdev.edmodo:android-image-cropper:2.6.+'
28     implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
29     testImplementation 'junit:junit:4.12'
30     androidTestImplementation 'androidx.test.runner:1.2.0'
31     androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
32     implementation 'com.google.android.gms:play-services-maps:17.0.0'
33 }
34
```

Slika 1 - Gradle build fajl

Kao što je već napomenuto u prvom delu dokumenta koristi se nekoliko biblioteka za dodatne funkcionalnosti aplikacije, poput recimo biblioteke za manipulaciju slika,.



Slika 2 – Struktura fajlova

Kao osnovni OOP koncept prvenstveno možemo reći za Movie i User klasu, koje ne predstavljaju ništa drugo do Entitete za film i korisnika. Poseduju sve osnovne funkcije poput gettera i settera i naravno konstruktor za inicijalizaciju istih.

Klasa DBHelper vrši kreiranje SQ Lite baze podataka ukoliko ona već ne postoji. Bitno je napomenuti da se ta baza čuva u memoriji telefona, stoga baza živi dok aplikacija živi.

U okviru helpers paketa vidimo InputValidation klasu koja ne predstavlja ništa drugo do pomoćne klase za validaciju podataka.

```

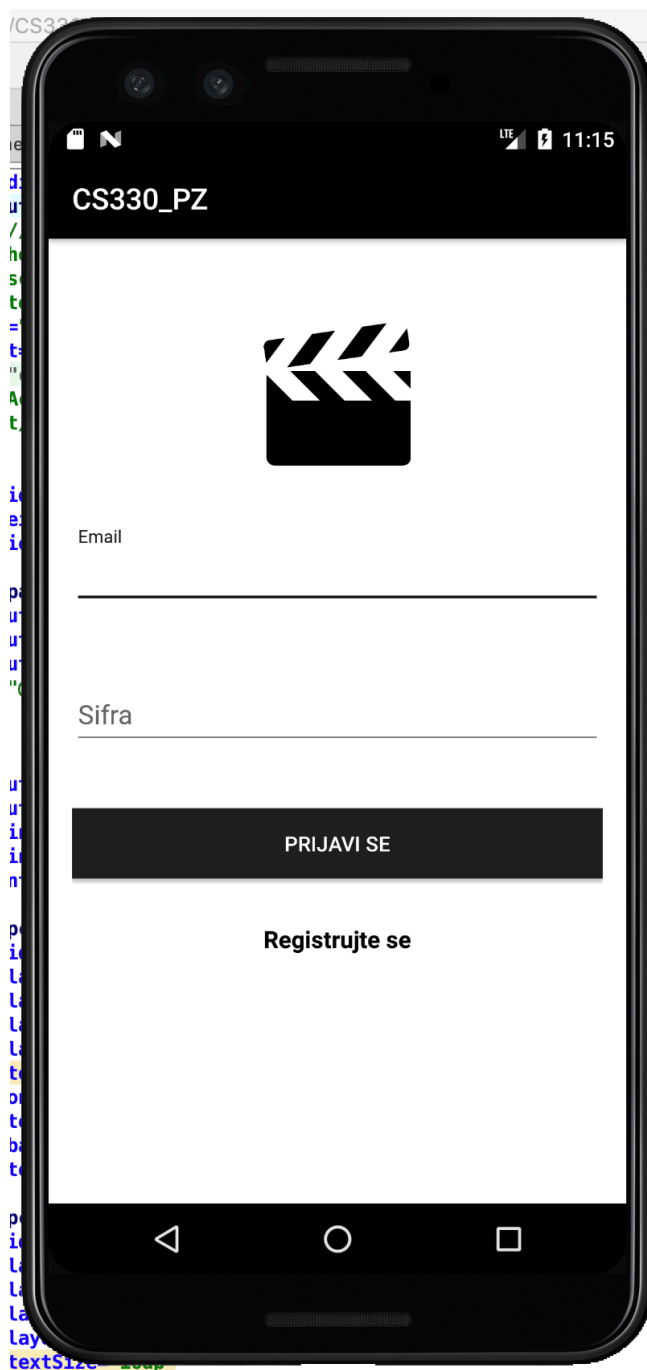
1 package com.metropolitan.cs330_pz.helpers;
2
3 import android.app.Activity;
4 import android.content.Context;
5 import android.util.Patterns;
6 import android.view.View;
7 import android.view.WindowManager;
8 import android.view.inputmethod.InputMethodManager;
9
10 import com.google.android.material.textfield.TextInputEditText;
11 import com.google.android.material.textfield.TextInputLayout;
12
13
14 public class InputValidation {
15
16     private Context context;
17
18     public InputValidation(Context context) { this.context = context; }
19
20
21
22     public boolean isInputEditTextFilled(TextInputEditText textInputEditText, TextInputLayout textInputLayout, String message){
23         String value = textInputEditText.getText().toString().trim();
24         if(value.isEmpty()){
25             textInputLayout.setError(message);
26             hideKeyboardForm(textInputEditText);
27             return false;
28         } else{
29             textInputLayout.setErrorEnabled(false);
30         }
31         return true;
32     }
33
34     public boolean isInputEditTextEmail(TextInputEditText textInputEditText, TextInputLayout textInputLayout, String message){
35         String value = textInputEditText.getText().toString().trim();
36         System.out.println("Patterns.EMAIL_ADDRESS.matcher(value).matches()" + value + Patterns.EMAIL_ADDRESS.matcher(value).matches());
37         if(value.isEmpty() || !Patterns.EMAIL_ADDRESS.matcher(value).matches()){
38             textInputLayout.setError(message);
39             hideKeyboardForm(textInputEditText);
40             return false;
41         } else {
42             System.out.println("Nema errora");
43             textInputLayout.setErrorEnabled(false);
44         }
45         return true;
46     }
47
48     public boolean isInputEditTextMatches(TextInputEditText textInputEditText1, TextInputEditText textInputEditText2, TextInputLayout textInputLayout){
49         String value1 = textInputEditText1.getText().toString().trim();
50         String value2 = textInputEditText2.getText().toString().trim();
51         if(!value1.contentEquals(value2)){
52             textInputLayout.setError(message);
53             hideKeyboardForm(textInputEditText2);
54             return false;
55         }
56     }
57 }

```

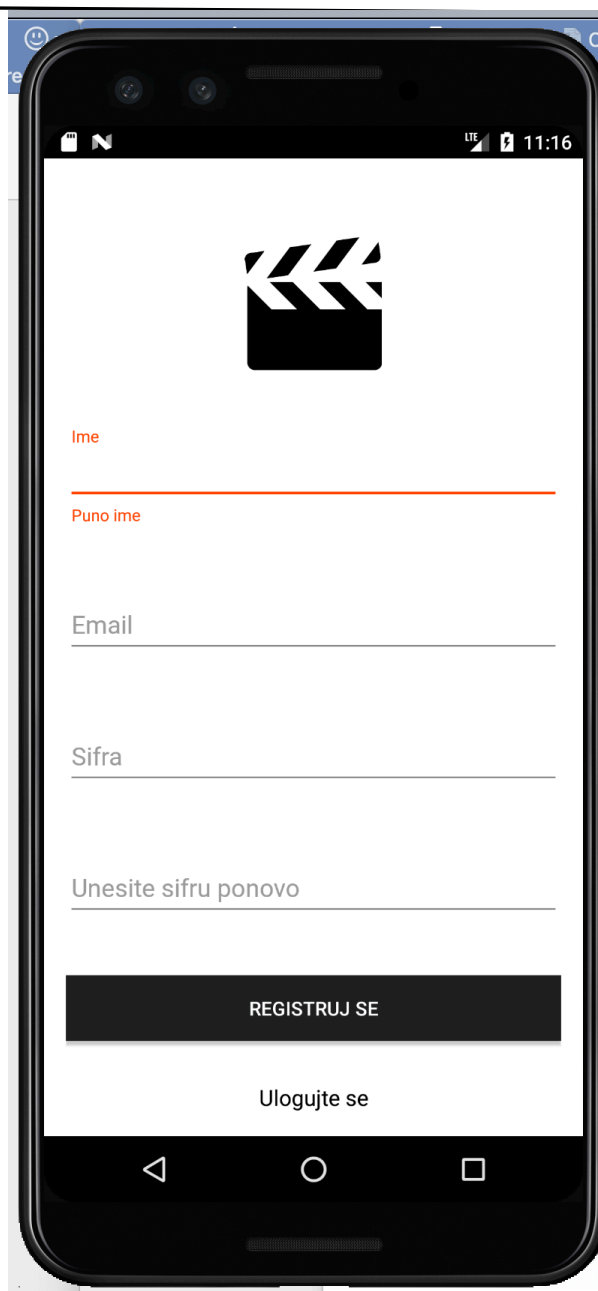
Slika 3 – InputValidation

U okviru gore pomenute klase nalaze se metode za proveru da li je podataka unet, validacija email adrese, kao i da li se dve šifre matchuju.

Prikaz aplikacije



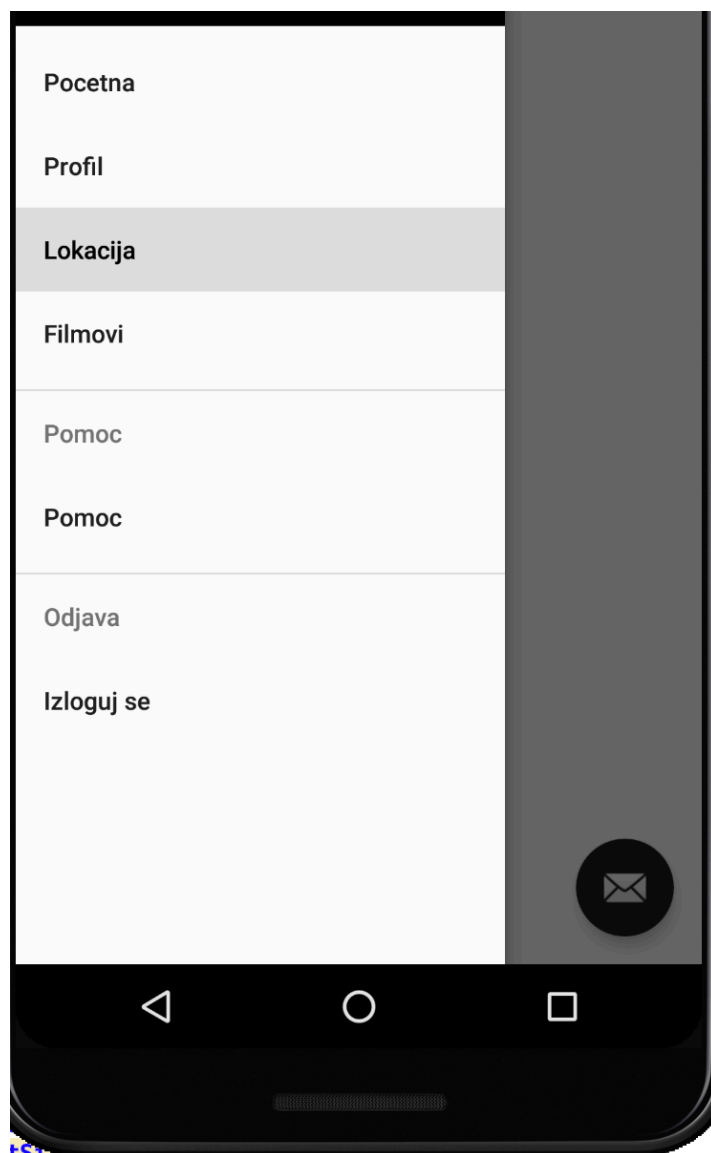
Slika 4 - Početni prikaz aplikacije



Slika 5 - Registraciona strana aplikacije

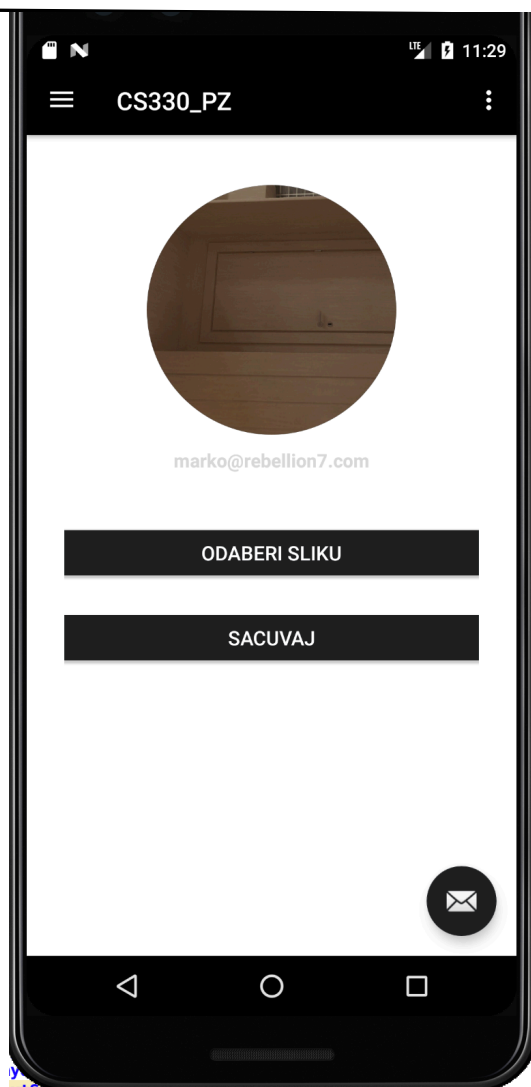
Kao što možemo videti, postoji validacija unetih podataka. Po unetim detaljima nakon klika na dugme, prikazuje se obaveštenje da je korisnik ulogovan te da može pristupiti samoj aplikaciji kroz login screen.

Po samom logovanju, korisnik ima nekoliko opcija iz samog menija da odabere:



Slika 6 - Opcije za izbor

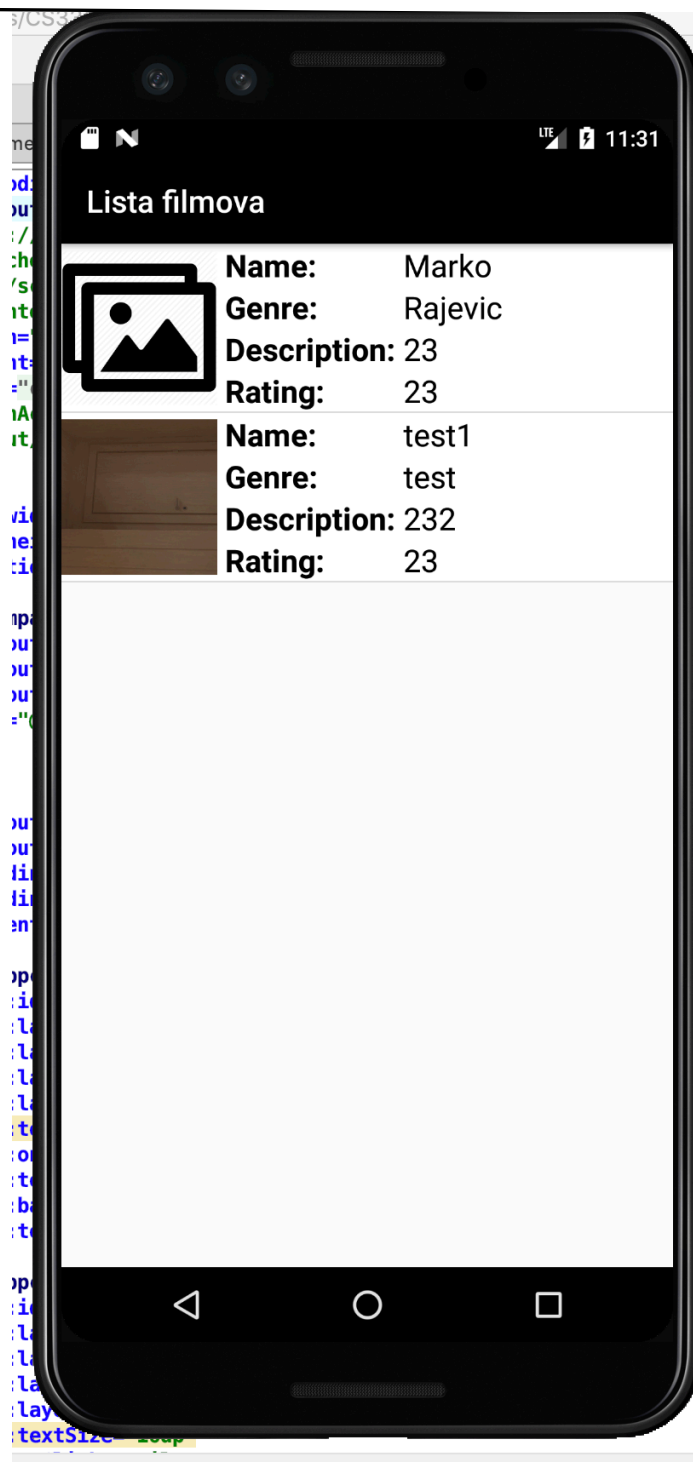
Odabirom opcije profil, korisniku se prikazuje ekran za manipulaciju licnim podacima kao i za dodavanje avatara.



Slika 7 - Prikaz profila

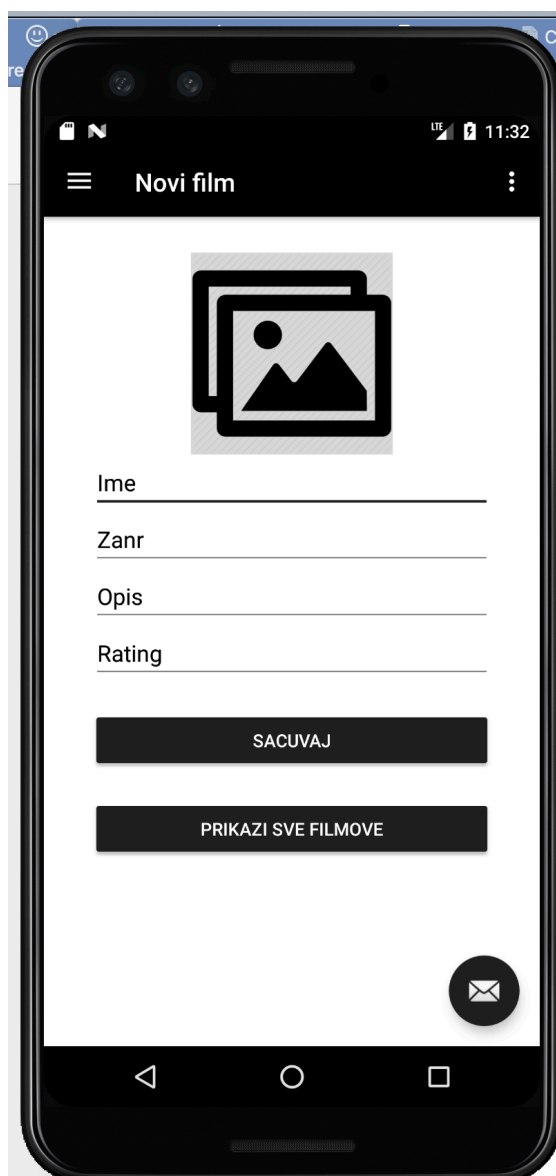
Odabirom opcije HELP korisniku se prikazuje HELP strana sa opcijama za Poziv, Mail kao i dugmetom za otvaranje lokacije koja koristi GoogleMaps servise. Takođe, korisniku je u svakom trenutku dostupno FAB dugme u donjem desnom uglu za pomoć.

Na sledećem ekranu prikazan je deo aplikacije za prikaz svih filmova:

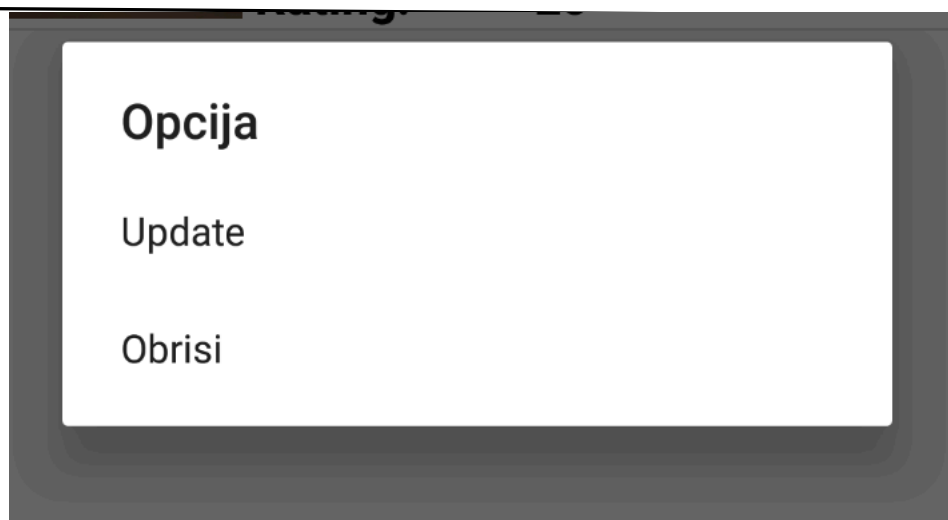


Slika 8 - Prikaz svih filmova

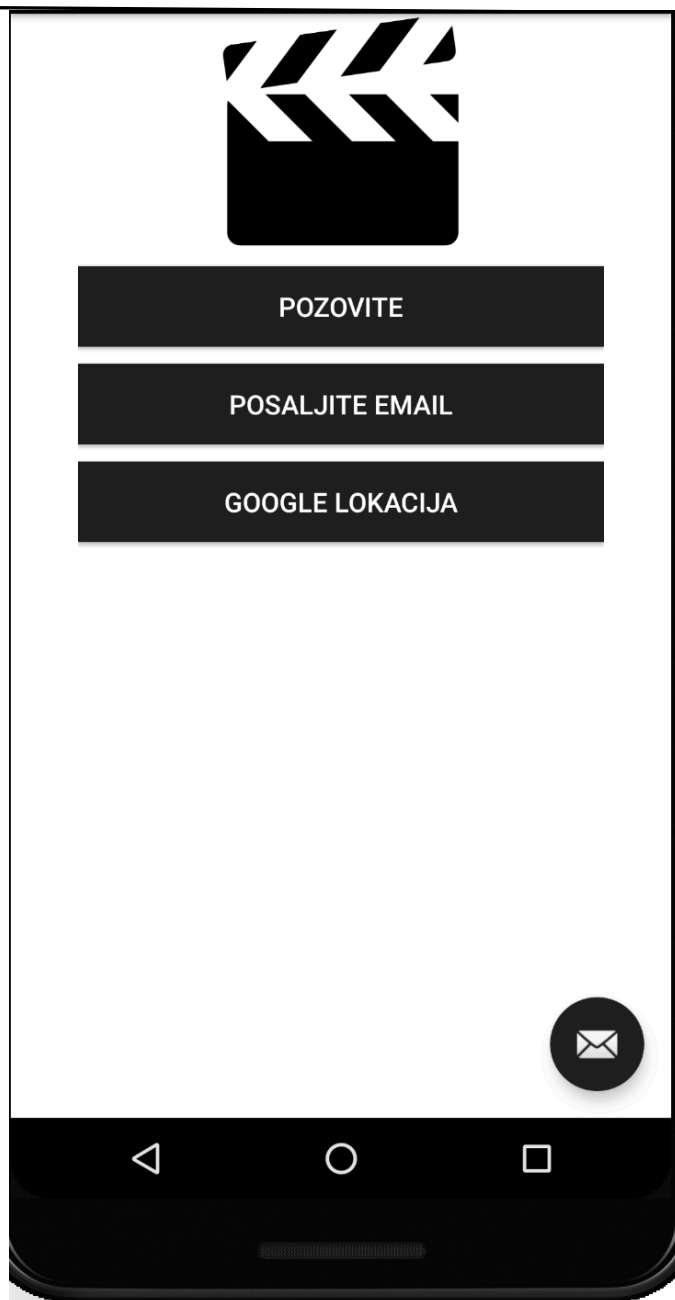
Na ekranu ispod možemo videti formu za unos filma:



Slika 9 - Unos filma



Slika 10 - Prikaz opcija nakon fokusa na list itemu



Slika 11 - Pomoćni ekran aplikacije

Zaključak

Kroz samu izradu aplikacije, usavršeno je znanje izrade istih kroz Java development. Kao osoba koja je do sada razvijala hibridne aplikacije, jasno vidim prednosti nativnih aplikacija razvijenih za specifičnu platformu. U toku izrade projekta korišćeni su primeri sa vežbi i predavanja što se tiče Fragmenta, Layouta i aktivnosti.

Literatura

1. <https://developer.android.com/jetpack/androidx/migrate>
2. Prof. dr. Vladimir Milićević , predavanja i vežbe CS330 – Univerzitet Metropolitan
3. Stackoverflow.com
4. <https://developer.android.com/>