



UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI FAKULTET
DEPARTMAN ZA MATEMATIKU I
INFORMATIKU



Android aplikacija za preporuku filmova

Mozzak

Seminarski rad C

Razvoj mobilnih aplikacija

Ime i prezime: Andrej Budinčević

Broj indeksa: 186m/16

Sadržaj

1. Opis funkcionalnosti aplikacije	3
2. Arhitektura aplikacije.....	3
3. Prikaz rada aplikacije	6
4. Prilog	12

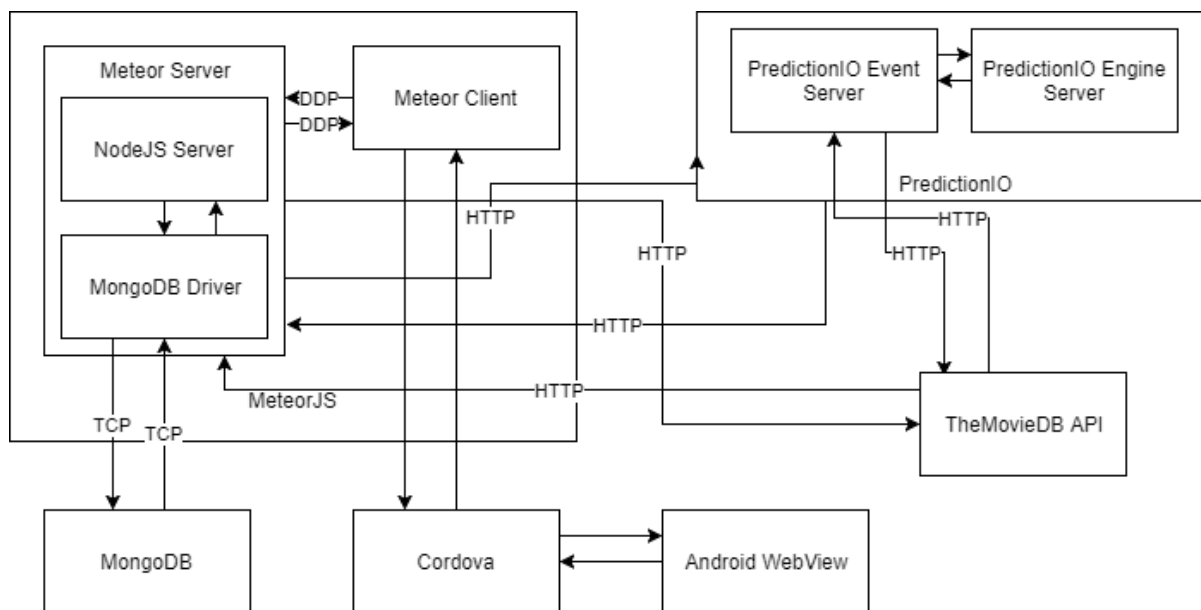
1. Opis funkcionalnosti aplikacije

Mozzak je aplikacija za preporuku filmova bazirana na mašinskom učenju, na recommender sistemu. Na početku interakcije sa korisnikom nudi nasumično odabrane filmove i daje korisniku mogućnost da oceni ponuđene filmove ocenama od 1 do 5. Nakon dovoljno ocenjenih filmova, recommender sistem počinje da preporučuje filmove i polako ih uključuje u listu rezultata. Kako skuplja više podataka, tako su preporuke sve bolje i bolje.

2. Arhitektura aplikacije

Aplikacija je implementirana koristeći javascript framework MeteorJS i Cordovu koji omogućavaju pravljenje nativnih Android aplikacija pomoću web tehnologija.

U pitanju je klasična klijent – server arhitektura gde server u pozadini obrađuje zahteve, a klijentski deo služi za prikaz informacija. U našem slučaju, server je MeteorJS server koji je hostovan na eksternom servisu (Scalewayⁱ), a klijentski deo je sama Android aplikacija koja se izvršava unutar Android WebView-a preko Cordove. Ovakva arhitektura ima jednu veliku prednost u odnosu na klasične Android aplikacije, a to je podrška za hot code pushⁱⁱ. Naime, zbog eksternog servera moguće je dostaviti novu verziju aplikacije korisnicima bez ponovnog pakovanja aplikacije u APK.



Ilustracija 1 - Grafički prikaz arhitekture aplikacije
*(*DDP je data definition protocol koji MeteorJS nativno koristi za komunikaciju između klijenta i servera, omogućava nativnu reaktivnost)*

Za pribavljanje podataka o filmovima korišćen je TheMovieDB APIⁱⁱⁱ koji sadrži veliku kolekciju raznoraznih filmova. Ti podaci su korišćeni za trening recommender sistema koristeći open-source projekat PredictionIO^{iv}. PredictionIO se osnovno sastoji od dve komponente, Engine i Event servera. Event server se koristi za prikupljanje događaja (rate, etc) koji se kasnije koriste za trening modela. Engine server se koristi za davanje preporuka iz istreniranog modela. U našem slučaju korišćena je samo rate akcija sa numeričkom vrednošću rejtinga u intervalu od 1 do 5. Za generisanje preporuka korišćen je Universal Recommender algoritam sa default parametrima. Ponovni trening mreže se vrši na svakih 15 minuta, zbog ograničenih hardverskih resursa na eksternom serveru.

Struktura direktorijuma MeteorJS aplikacije je poprilično jednostavna, klijentski deo ide u *client* folder, a serverski deo u *server*.

Na serverskom delu se nalaze metodi koji se pozivaju samo na serveru zbog potrebe za api ključevima.

- *callTMDBApi*: (*method*, *endpoint*, *data*) – Uzima podatke sa TMDB API-a, prima metod poziva (*GET*, *POST*, *etc*), endpoint API-a (*/discover*, *etc*) i dodatne podatke za API poziv.
- *importTMDB*: (*token*) – Prima samo token da bi se sprečilo neautorizovano pozivanje metoda, importuje podatke sa TMDB API-a u PredictionIO Event server, efektivno kreirajući dataset za kasniji trening mreže. Ne treba ga pozivati, jer je već odradio svoj posao.
- *recordAction*: (*movieId*, *event*, *properties*) – Čuva akciju na PredictionIO Event serveru. Prima identifikator filma, tip akcije (*rate*) i objekat sa dodatnim podacima. Generiše glavne podatke za trening mreže.
- *getRecommendation*: () – Uzima preporuke sa PredictionIO Engine servera.
- *saveSettings*: (*infinite*, *movies*) – Čuva podešavanja korisnika, prima boolean vrednost *infinite* koja označava da li je infinite scroll uključen ili ne i numeričku vrednost *movies* koja označava koliko se filmova prikazuje na početnoj stranici.
- *user1* (test): () – Pravi novog korisnika na PredictionIO Event serveru, samo za testiranje.

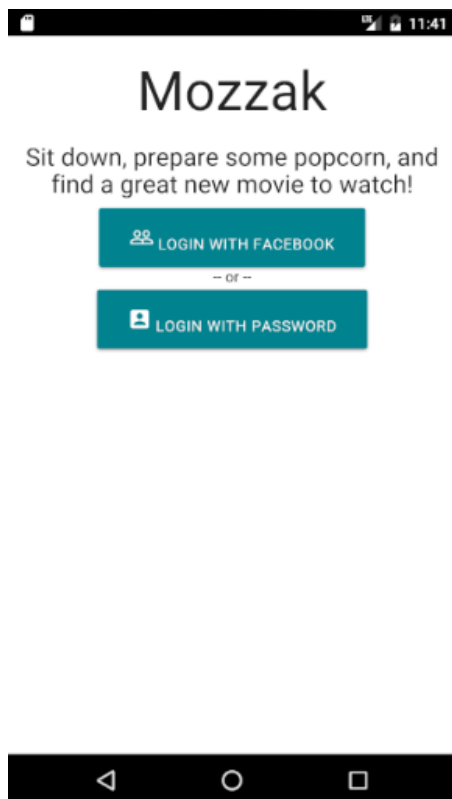
Pored toga, na serveru se podešavaju Facebook login parametri i logika kreiranja korisnika.

Na klijentskom delu se nalaze svi ekrani aplikacije u folderu *templates* koji su u stvari obične HTML stranice. U *layouts* folderu se nalazi glavni layout aplikacije. Helper-i^v na stranicama i događaji se obrađuju u fajlovima *helpers.js* i *events.js* respektivno. *Main.js* sadrži osnovnu klijentsku logiku aplikacije i metode za pribavljanje podataka sa servera. *Router.js* je zadužen za rutiranje stranica unutar aplikacije.

Unutar *public* foldera se nalaze podaci koji su javno dostupni na adresi (*slike*, *fontovi*).

3. Prikaz rada aplikacije

Početni ekran (za neulogovane korisnike)



Na ovom ekranu korisnik ima izbor kako želi da se uloguje u aplikaciju i počne da je koristi. Moguće je ulogovati se putem Facebook naloga ili putem mejl adrese. Login preko Facebook-a pokreće oauth2 flow i loguje korisnika u aplikaciju nakon uspešne verifikacije.

Login with password ekran

register here.' At the bottom of the screen is a black Android navigation bar with back, home, and recent apps icons." data-bbox="177 105 426 415"/>

Na ovom ekranu korisnik može da se uloguje sa već registrovanim nalogom. Klikom na Login dugme odlazi na glavnu početnu stranu za ulogovane korisnike. Ukoliko korisnik još nema nalog, može otići na ekran za registraciju klikom na odgovarajući link.

Register ekran

Register

Name
Andrej Budincevic

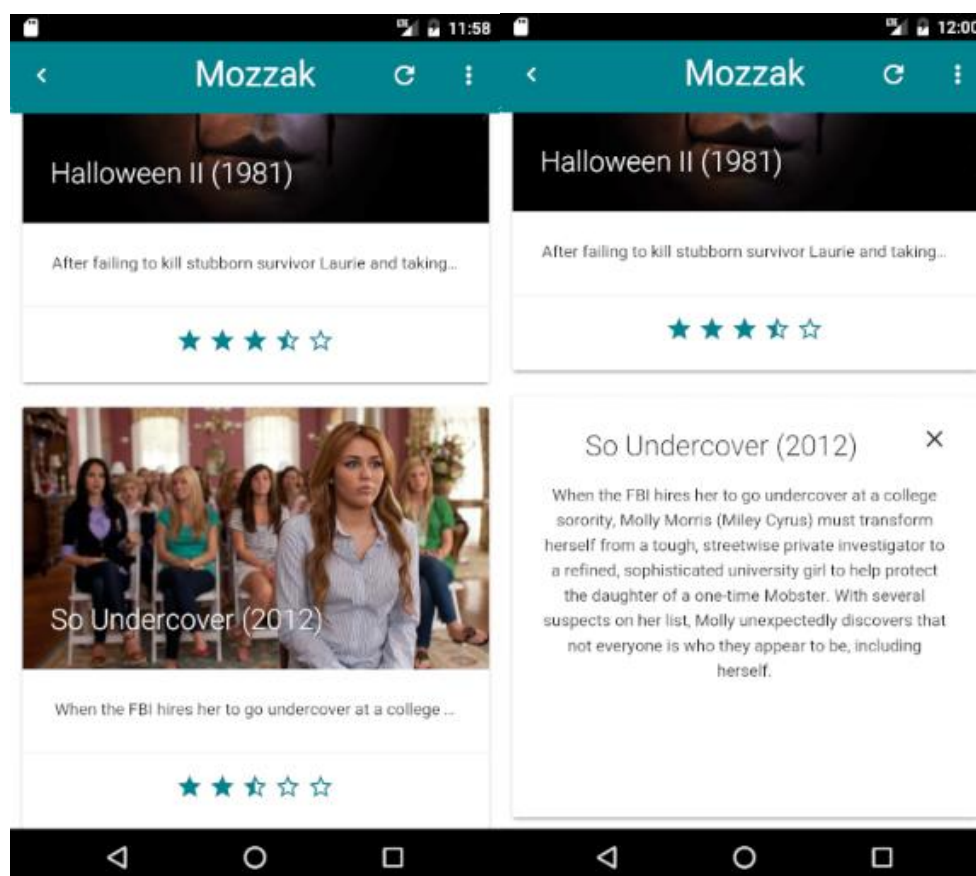
Email
andrew@hotmail.rs

Password

REGISTER >

Na ovom ekranu korisnik može da se registruje za korišćenje aplikacije. Za registraciju je potrebno ime, mejl adresa i lozinka. Trenutno nema ograničenje na dužinu lozinke, ali postoji provera da li je mejl adresa validna (u validnom formatu) i da nije disposable mejl adresa. Potvrda mejl adrese takođe nije obavezna. Klikom na dugme Register, korisnik se automatski loguje na aplikaciju i može da nastavi sa korišćenjem.

Glavni ekran (za ulogovane korisnike)



Na glavnom ekranu aplikacije se prikazuju preporučeni filmovi. Na početku, tu su samo nasumični filmovi koje korisnik može da ocenjuje da bi trenirao model za preporuku filmova. Kako korisnik ocenjuje filmove, tako se polako lista nasumičnih filmova menja listom preporučenih filmova. Naravno, što više filmova korisnik oceni, to su preporuke bolje.

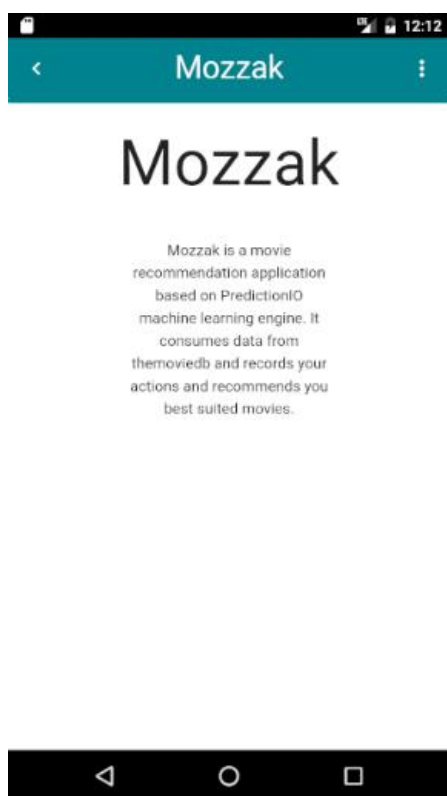
Prikaz na glavnom ekranu je podeljen u kartice, gde je svaka kartica jedan film. Klikom na bilo koji deo kartice otvara duži opis filma. Klikom na zvezdice dole (koje na početku prikazuju prosečnu ocenu filma) korisnik ocenjuje film, zvezdice menjaju boju i model se trenira sa novim podacima. Za prikaz podataka je umesto klasične paginacije korišćen sve popularniji infinite scroll mehanizam, gde se lista filmova dopunjava kako korisnik prolazi kroz postojeću listu. Ukoliko korisnik ne želi takvo ponašanje, može isključiti infinite scroll u podešavanjima. Tada, ukoliko želi da obnovi listu filmova, mora da klikne na refresh ikonicu u gornjem baru.

Navigacija



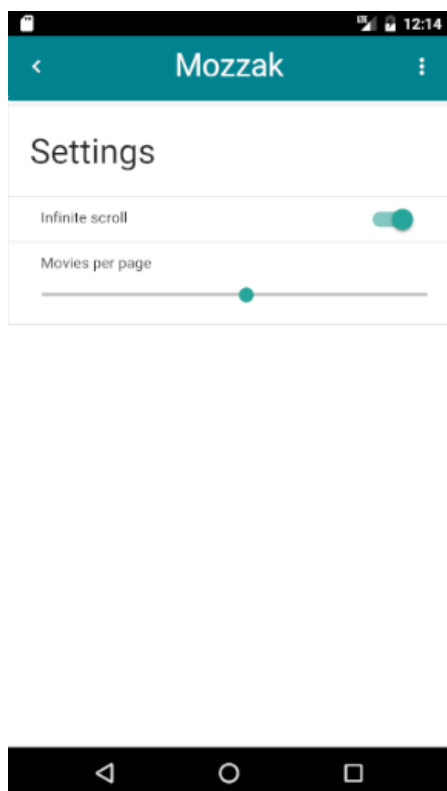
Za potrebe navigacije kroz aplikaciju korišćen je dropdown koji se pojavljuje nakon klika na meni dugme u gornjem baru. U prvom polju se nalazi ime korisnika. Nakon toga ima opciju da se izloguje, da namesti podešavanja aplikacije, da pogleda About stranicu ili da jednostavno napusti aplikaciju.

About ekran



Na About ekranu se nalaze osnovni podaci o aplikaciji.

Settings ekran



Na Settings ekranu korisnik može da promeni podešavanja vezana za infinite scroll i broj filmova koji se učitava na glavnu stranu.

Zbog responsivnog dizajna, aplikacija korektno radi i na portrait i na landscape modu, i radi korektno na različitim veličinama ekrana, od tableta, preko phableta do ekrana od 3".

4. Prilog

Uz seminarski rad priložen je programski kod opisane aplikacije (*Code*) i unapred zapakovana potpisana APK aplikacija (*App/mozzak.apk*). Eksterni server je u trenutku predavanja rada pokrenut, te je za testiranje dovoljno samo pokrenuti već zapakovanu potpisanu aplikaciju u APK format-u.

PredictionIO Event server radi na: <http://pio.zx.rs:7070> (*Scaleway VC1L*)

PredictionIO Engine server radi na: <http://pio.zx.rs:8000> (*Scaleway VC1L*)

MeteorJS Backend server radi na: <http://mozzak.zx.rs/> (ovde je takođe moguće pristupiti aplikaciji koristeći običan browser) (*Scaleway VC1S instance*)

Ukoliko postoji želja za testiranjem izvornog koda, potrebno je instalirati MeteorJS.^{vi} Pokretanje MeteorJS servera se vrši sa:

```
meteor
```

komandom što omogućava testiranje aplikacije u browser-u, na adresi <http://localhost:3000/>. Za pokretanje aplikacije na već pokrenutom Android emulatoru se koristi:

```
meteor run android
```

Za pokretanje na fizičkom uređaju povezanom preko ADB-a koristi se:

```
meteor run android-device.vii
```

Za pakovanje aplikacije u APK format koristi se:

```
meteor build ../folder --server mozzak.zx.rs
```

što generiše nepotpisani APK. Potpisivanje se vrši *keytool*-om i *jarsigner*-om.^{viii}

-
- i <https://www.scaleway.com/>
 - ii <https://blog.meteor.com/meteor-hot-code-push-with-great-power-comes-great-responsibility-7e9e8f7312d5>
 - iii <https://www.themoviedb.org/documentation/api>
 - iv <https://predictionio.incubator.apache.org/>
 - v <http://blazejs.org/api/spacebars.html>
 - vi <https://www.meteor.com/install>
 - vii <https://guide.meteor.com/mobile.html>
 - viii <https://stackoverflow.com/questions/10930331/how-to-sign-an-already-compiled-apk>