

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6714

**BAZA PODATAKA I WEB-APLIKACIJA ZA  
RECENZIJU FILMOVA**

Robert Pavliš

Zagreb, lipanj 2020.



# Sadržaj

Uvod.....	5
1. Aplikacija za recenziju filmova.....	6
1.1. Opis zadatka .....	6
1.2. Analiza zahtjeva .....	6
1.2.1. Analiza zahtjeva korisničkog sučelja .....	7
1.3. Opis slučajeva uporabe.....	8
1.4. Baza podataka .....	11
1.4.1. ER model .....	13
1.4.2. ER dijagram baze podataka .....	13
1.4.3. Opis ER modela .....	14
1.4.4. Popis entiteta i atributa .....	15
1.4.5. Popis veza.....	15
1.4.6. Preslikavanje ER modela u relacijski model .....	15
1.4.7. Popis relacija .....	16
1.4.8. Stvaranje tablica i okidača.....	23
2. Implementacija web aplikacije .....	28
2.1. Arhitektura aplikacije .....	28
2.2. Izvod arhitekture aplikacije .....	29
2.2.1. Repozitoriji .....	29
2.2.2. Servisi .....	30
2.2.3. Upravljači .....	31
2.2.4. Sigurnost aplikacije .....	34
2.2.5. Sloj korisničkog sučelja .....	35
3. Rad s aplikacijom .....	37
3.1. Registracija korisnika .....	37

3.2. Autentifikacija korisnika.....	38
3.3. Dodavanje filmova.....	39
3.4. Sortiranje filmova .....	39
3.5. Pretraživanje filmova .....	41
3.6. Pretraga statistike filmova .....	42
3.7. Pregled pojedinačnog filma .....	43
3.8. Ažuriranje i brisanje filma .....	43
Zaključak .....	45
Literatura .....	46
Sažetak .....	47
Summary .....	48

## Uvod

Prosječna osoba kroz tjedan pogleda nekoliko filmova. Pomnoži li se taj broj sa brojem tjedana u godini pa sa nekoliko godina dobije se nemali ukupan broj filmova. Osoba, naravno, teško da može pratiti svaki film koji je pogledala pa onda neki filmovi neizbježno ostaju zaboravljeni. Kako se tako nešto ne bi dogodilo te kako bi korisnik mogao imati strukturirani pregled filmova koje je pogledao, može koristiti web aplikaciju koja će mu u tome pomoći.

Zadatak ovog završnog rada je izraditi bazu podataka i web aplikaciju za praćenje i recenziju pogledanih filmova. Prvo je potrebno kreirati bazu podataka u koju će biti spremljeni svi filmovi te popratni podaci vezani uz njih. Nakon toga treba osmisliti web aplikaciju koja će na jednostavan način omogućiti pregled filmova te recenziju istih.

Aplikacija je dizajnirana tako da korisnici neće sami upisivati osnovne podatke o filmovima, jer bi to bilo zamorno, nego se podaci o filmovima uzimaju s vanjske lokacije (u ovom slučaju stranice IMDb).

U ovom će se radu koristiti sustav za upravljanjem bazom podataka H2 te programski jezici Java i JavaScript te razvojni okvir Spring boot i biblioteka React. Također su korištene i tehnologije HTML5 i CSS.

U nastavku će biti opisana sama aplikacija što se svodi na opisivanje baze podataka (zajedno sa ER modelom i relacijskim modelom) te implementacije aplikacije i korisničkog sučelja te rada s istim.

# **1. Aplikacija za recenziju filmova**

## **1.1. Opis zadatka**

Aplikacija treba omogućiti registriranim korisnicima da unesu poveznicu na IMDb stranicu pogledanog filma, nakon čega će se stvoriti instanca filma koja sadrži osnovne podatke o filmu kao što su: ime redatelja, glumaca, kategorija, itd. Korisnik nakon toga samo mora upisati ocjenu filma, datum gledanja i recenziju (ako želi). Aplikacija također treba omogućiti jednostavan način brisanja filmova i izmjene podataka. Korisnik u svakom trenutku može na naslovnoj stranici vidjeti koje je filmove dodao te ih sortirati po raznim kategorijama poput: naslova, datuma gledanja, godini izlaska, itd. Korisnik također može pretraživati filmove po: naslovu, redatelju, glumcu, itd. Korisnik također može gledati statistiku nad svim filmovima u obliku najboljeg/najgoreg glumca/redatelja/scenarista po IMDb/osobnim ocjenama. Postoje dva tipa korisnika: osnovni korisnik koji unosi i ocjenjuje filmove te administrator koji može uklanjati podatke iz baze podataka. Aplikacija treba omogućiti korisniku pregled vlastitih dodanih filmova, ali ne i filmove dodane od strane drugih korisnika (čak i ako se radi o istim filmovima). Novi posjetitelj se treba moći registrirati i ulogirati te tako postati korisnik.

## **1.2. Analiza zahtjeva**

Svaki korisnik pri registraciji postavlja korisničko ime, e-mail adresu i lozinku te mu se dodjeljuje šifra koja se sprema u bazi. Korisnik se prijavljuje sa korisničkim imenom i lozinkom, nakon čega mu se dodjeljuje status korisnika.

Za svakog korisnika u bazi podataka postavljaju se:

- šifra korisnika (generira se automatski)

- korisničko ime, lozinka (tj. sažetak engl. *hash* lozinke) i e-mail adresa (postavlja korisnik)
- status (interna struktura koja ovisi o korisničkom imenu, tj. ako se korisničko ime razlikuje od „admin“, onda se radi o normalnom korisniku)

Web aplikacija sadrži i drugu vrstu korisnika – administratora. Pri radu s aplikacijom, administrator ima iste ovlasti kao i normalan korisnik, međutim administrator može rukovati direktno s bazom podataka, tj. može mijenjati i brisati podatke iz baze.

Za administratora se u bazi podataka postavlja:

- šifra korisnika (generira se automatski)
- korisničko ime (koje mora glasiti „admin“), lozinka (tj. sažetak engl. *hash* lozinke) te e-mail adresa
- statusi (status korisnika i status administratora)

### **1.2.1. Analiza zahtjeva korisničkog sučelja**

Korisničko sučelje služi kako bi korisnik brzo, efikasno i intuitivno mogao raditi sa aplikacijom, bez da zna išta o tehnologijama kojima je web aplikacija izvedena. Temeljne zadaće sučelja su:

- unos filmova
- pregled, sortiranje i traženje filmova
- izmjena i brisanje filmova

Korisničko sučelje dizajnirano je tako da korisniku omogućuje lako snalaženje te jednostavnu interakciju s aplikacijom. Tako je, na primjer, za dodavanje filma (što je

osnovna operacija u aplikaciji), nakon autentifikacije potrebno samo zalijepiti poveznicu filma sa IMDb stranice i pritisnuti jedan gumb.

### **1.3. Opis slučajeva uporabe**

#### ➤ Registracija novog korisnika

- Glavni sudionik: neregistrirani korisnik
- Cilj: registrirati novog korisnika u aplikaciju
- Ostali sudionici: baza podataka
- Preduvjeti: dostupnost poslužitelja
- Željeni scenarij: korisnik se registrira u aplikaciju
- Alternativni scenariji: registracija ne uspijeva iz nekog od sljedećih razloga: korisnik već postoji u aplikaciji, e-mail koji je korisnik dao se već koristi, korisničko ime je prekratko/predugo, lozinka je prekratka/preduga, e-mail je prekratak/predug, e-mail nema oblik e-mail adrese

#### ➤ Autentifikacija korisnika

- Glavni sudionik: registrirani korisnik
- Cilj: dopustiti ulazak korisnika u aplikaciju
- Ostali sudionici: baza podataka
- Preduvjeti: dostupnost poslužitelja
- Željeni scenarij: korisnik ulazi u aplikaciju
- Alternativni scenariji: korisnik krivo upisuje podatke te mu se ne dozvoljava ulaz u aplikaciju

#### ➤ Dodavanje filma

- Glavni sudionik: registrirani korisnik



- Cilj: dodavanje filma
- Ostali sudionici: baza podataka, stranica IMDb
- Preduvjeti: dostupnost poslužitelja, poveznica filma s IMDb stranice, dostupnost IMDb stranice
- Željeni scenarij: korisnik dodaje film
- Alternativni scenariji: film već postoji te se ne dodaje, poveznica ne pripada IMDb filmu

#### ➤ Sortiranje filmova

- Glavni sudionik: registrirani korisnik
- Cilj: sortirati filmove po nekom kriteriju
- Ostali sudionici: baza podataka
- Preduvjeti: dostupnost poslužitelja, korisnik ima barem jedan film u bazi
- Željeni scenarij: korisnik sortira filmove
- Alternativni scenariji: korisnik nema filmova u bazi te se nema što sortirati

#### ➤ Pretraživanje filmova

- Glavni sudionik: registrirani korisnik
- Cilj: pretražiti filmove po nekom parametru
- Ostali sudionici: baza podataka
- Preduvjeti: dostupnost poslužitelja, korisnik ima barem jedan film u bazi
- Željeni scenarij: korisnik pronalazi filmove
- Alternativni scenariji: korisnik nema filmova u bazi te se nema što pretražiti

➤ Pretraga statistike filmova

- Glavni sudionik: registrirani korisnik
- Cilj: dobiti statistiku nad svim filmovima po odabranom kriteriju
- Ostali sudionici: baza podataka
- Preduvjeti: dostupnost poslužitelja, korisnik ima barem jedan film u bazi
- Željeni scenarij: korisnik dobiva statistiku nad traženim parametrima o filmovima
- Alternativni scenariji: korisnik nema filmova u bazi te nema nad čime tražiti

➤ Pregled pojedinačnog filma

- Glavni sudionik: registrirani korisnik
- Cilj: pregledati podatke o filmu
- Ostali sudionici: baza podataka
- Preduvjeti: dostupnost poslužitelja, korisnik ima barem jedan film u bazi
- Željeni scenarij: korisnik sortira filmove
- Alternativni scenariji: korisnik nema filmova u bazi te se nema što pregledati, korisnik nema pristup filmu kojeg pokušava pregledati te ga neće moći pregledati

➤ Ažuriranje filma

- Glavni sudionik: registrirani korisnik
- Cilj: ažurirati podatke o filmu
- Ostali sudionici: baza podataka
- Preduvjeti: dostupnost poslužitelja, korisnik ima barem jedan film u bazi
- Željeni scenarij: korisnik ažurira podatke o filmu

- Alternativni scenariji: ažuriranje ne uspijeva iz nekog od sljedećih razloga: naslov filma je predugačak, godina je izvan raspona, IMDb ocjena jer izvan raspona, URL slike je predugačak, popis redatelja je predugačak, popis scenarista je predugačak, popis glumaca je predugačak, popis kategorija je predugačak, popis jezika je predugačak, popis država je predugačak, korisnikova ocjena je izvan raspona, korisnik nije dao ocjenu, korisnik nije postavio datum gledanja, trajanje je izvan raspona, polje recenzija je predugačko

#### ➤ Brisanje filma

- Glavni sudionik: registrirani korisnik
- Cilj: obrisati film
- Ostali sudionici: baza podataka
- Preuvjeti: dostupnost poslužitelja
- Željeni scenarij: korisnik briše film
- Alternativni scenariji: brisanje ne uspijeva iz nekog od sljedećih razloga: odabrani film je već obrisani, odabrani film ne postoji, korisnik nema pristup filmu kojeg pokušava obrisati

## 1.4. Baza podataka

Baza podataka je skup međusobno povezanih podataka, pohranjenih zajedno, uz isključenje bespotrebne zalihosti (redundancije), koji mogu zadovoljiti različite primjene. Podaci su pohranjeni na način neovisan o programima koji ih koriste.

Prilikom dodavanja novih podataka, mijenjanja i pretraživanja postojećih podataka primjenjuje se zajednički i kontrolirani pristup. Podaci su strukturirani tako da služe kao osnova za razvoj budućih primjena. [1]

Računalni program korišten za upravljanje i ispitivanje baze podataka zove se sustav upravljanja bazom podataka (SUBP).

SUBP se sastoji od:

- logičke razine
- koncepcijske razine
- fizičke razine

Ova web aplikacija koristi sustav za upravljanjem bazom podataka H2. H2 je sustav upravljanja relacijskim bazama podataka napisan na Javi. Može se ugraditi u Java aplikacije ili pokrenuti u načinu klijent-poslužitelj. Jako je brz i siguran. [2]

Podatke u bazi logički su strukturirani modelom podataka. To je skup pravila koja određuju kako sve može izgledati logička struktura baze podataka. Model čini osnovu za projektiranje i implementiranje baze. Podaci u bazi moraju biti logički organizirani u skladu s modelom koji podržava odabrani SUBP. [3]

Neki od modela podataka su:

- Relacijski model – podaci i veze među podacima prikazuju se tablicama koje se sastoje od redaka i stupaca
- ER model – detaljan logički prikaz podataka preko skupa entiteta, njihovih atributa i međusobnih veza
- Mrežni model – baza je predložena mrežom koja se sastoji od čvorova i usmjerenih lukova
- Hijerarhijski model – baza je predložena jednim stablom (hijerarhijom) ili skupom stabala
- Objektni model – baza je predložena kao skup trajno pohranjenih objekata koji se sastoje od svojih internih „atributa“ (podataka) i „metoda“ (operacija)

Danas su najčešće korišteni modeli relacijski model i ER model.

#### 1.4.1. ER model

ER model je model između entiteta u sustavu. Kratica ER dolazi od engleskih riječi *Entity Relationship*. ER shema sama po sebi prilično je jednostavna i služi kako bi se prikazalo koji su entiteti u bazi i njihove međusobne veze.

ER model sastoji se od entiteta (označenih pravokutnicima) i veza među njima (označenih rombovima). Odnos između dva entiteta može biti:

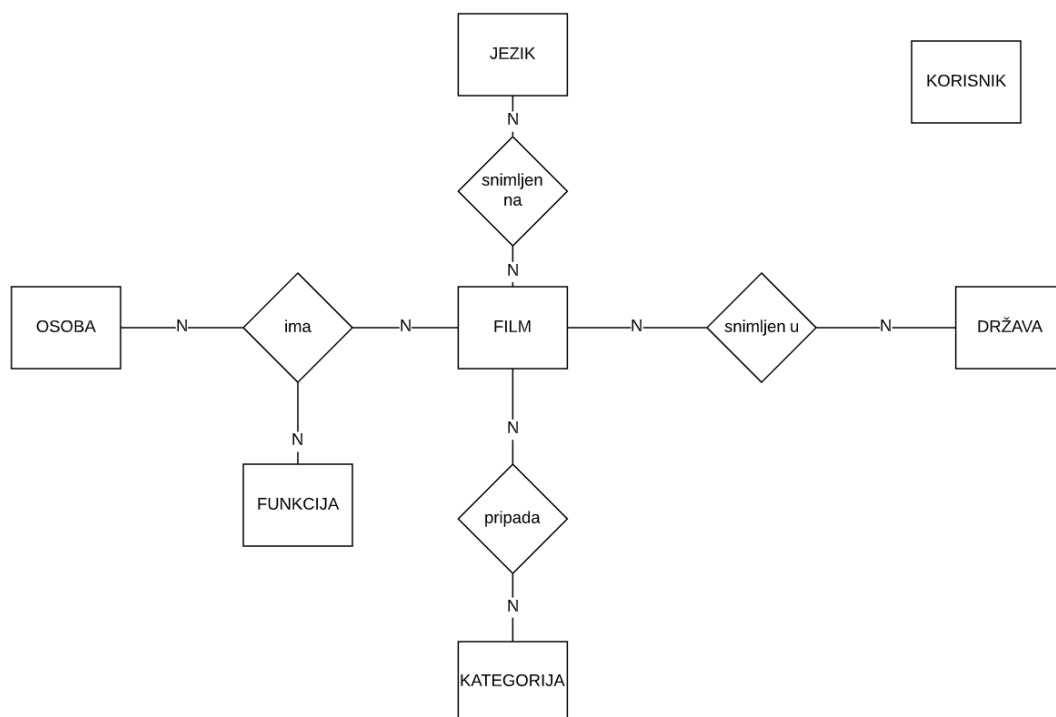
- 1:1 – jedan zapis entiteta vezan je za jedan zapis drugog entiteta
- 1:N – jedan zapis entiteta vezan je za N zapisa drugog entiteta
- N:1 – N zapisa entiteta vezano je za jedan zapis drugog entiteta
- N:N – N zapisa entiteta vezano je za N zapisa drugog entiteta

U ER modelu ove web aplikacije nalazi se i ternarna veza. To je veza između tri entiteta. Varijacije za ternarne veze su iste kao i za veze između dva entiteta. U ovom slučaju imamo vezu N:N:N (N zapisa entiteta vezano je za N zapisa drugog entiteta te za N zapisa trećeg entiteta).

Za prikaz baze podataka ove web aplikacije korišteni su ER model i relacijski model.

#### 1.4.2. ER dijagram baze podataka

ER dijagram baze podataka prikazan je na slici 1.1.



*Slika 1.1: ER model baze podataka*

### 1.4.3. Opis ER modela

Domena podataka su filmovi. Svaki film spada pod neku kategoriju (ili više njih), kao što su: akcijski, komedija, horror, triler, itd. Osobe koje rade na filmu imaju funkcije unutar filma: pisac (tj. scenarist), redatelj ili glumac. Osoba ima ime i prezime te može biti prisutna u više filmova i imati više uloga u istom filmu.

Podatke korisnika, koji unosi datum gledanja (formata dd.mm.yyyy), recenziju (ako želi) i ocjenu (broj između 0 i 10) za film, također se spremaju u bazu. Podaci korisnika uključuju korisničko ime, e-mail i lozinku. Korisnik može po želji uređivati sve podatke o filmu.

Podaci o filmu, uz najvažnije članove filmske ekipe i već spomenutu ocjenu, recenziju i datum gledanja, uključuju poveznicu s kojeg su preuzeti podaci o filmu, poveznicu gdje se nalazi poster (tj. naslovna slika filma), godinu izlaska filma, trajanje filma te ocjenu filma sa stranice IMDb (broj između 0 i 10). Svaki film ima i jezik, kojih može biti više, te može biti snimljen u jednoj ili više država.

#### 1.4.4. Popis entiteta i atributa

**OSOBA** – idOsoba, imeOsoba

**FILM** – idFilm, imeFilm, posterURL, poveznica, IMDbRating, godina, recenzija, datumGledanja, ocjenaKorisnika, trajanje

**KORISNIK** – idKorisnik, imeKorisnik, e-mail, lozinka

**FUNKCIJA** - idFunkcija, imeFunkcija

**JEZIK** – idJezik, imeJezik

**DRŽAVA** – idDržava, imeDržava

Napomena: Primarni ključ tablice je podvučen i masno otisnut engl. *bolded*

#### 1.4.5. Popis veza

**ima** (OSOBA - FILM - FUNKCIJA) – idOsoba, idFilm, idFunkcija (N:N:N)

**pripada** (KATEGORIJA - FILM) – idKategorija, idFilm (N:N)

**snimljen u** (DRŽAVA - FILM) – idDržava, idFilm (N:N)

**snimljen na** (JEZIK - FILM) – idJezik, idFilm (N:N)

Napomena: Primarni ključ tablice je podvučen i masno otisnut engl. *bolded*

#### 1.4.6. Preslikavanje ER modela u relacijski model

Preslikavanje u relacijski model odvit će se tako da sve N:N veze, kao i ternarna N:N:N veza, tvoriti novu relaciju.

### 1.4.7. Popis relacija

U nastavku su dane sve tablice koje se nalaze u bazi s pripadnim atributima, opisima atributa, tipom podataka i ograničenjima.

Tablica 1.1: Tablica Film

Atribut	Opis	Tip podatka	Ograničenja
idFilm	Identifikator filma	long	Primarni ključ
imeFilm	Ime filma	String	Maksimalna duljina = 1000
posterURL	Poveznica do naslovne slike filma	String	Maksimalna duljina = 2000
poveznica	Poveznica do IMDb stranice filma	String	Maksimalna duljina = 2000
IMDbOcjena	Službena ocjena sa stranice IMDb	double	Min = 0, Max = 10
godina	Godina snimanja filma	int	Min = 1900, Max = 2100
recenzija	Recenzija filma od strane korisnika web aplikacije	String	Maksimalna duljina = 50000



datumGledanja	Datum kad je film pogledan od strane korisnika aplikacije	Date	
ocjenaKorisnika	Ocjena koju je korisnik aplikacije dao filmu	int	Min = 0, Max = 10
trajanje	Trajanje filma (u minutama)	int	

*Tablica 1.2: Tablica Jezik*

Atribut	Opis	Tip podatka	Ograničenja
idJezik	Identifikator jezika	long	Primarni ključ
imeJezik	Ime jezika	String	Jedinstvena vrijednost (unique), Maksimalna duljina = 50

Tablica 1.3: Tablica Država

Atribut	Opis	Tip podatka	Ograničenja
idDržava	Identifikator države	long	Primarni ključ
imeDržava	Ime države	String	Jedinstvena vrijednost (unique), Maksimalna duljina = 50

Tablica 1.4: Tablica Kategorija

Atribut	Opis	Tip podatka	Ograničenja
idKategorija	Identifikator kategorije	long	Primarni ključ
imeKategorija	Ime države	String	Jedinstvena vrijednost (unique), Maksimalna duljina = 50

Tablica 1.5: Tablica SnimljenoU

Atribut	Opis	Tip podatka	Ograničenja
idDržava	Identifikator države	long	Dio primarnog ključa
idFilm	Identifikator filma	long	Dio primarnog ključa

Tablica 1.6: Tablica SnimljenoNa

Atribut	Opis	Tip podatka	Ograničenja
idJezik	Identifikator jezika	long	Dio primarnog ključa
idFilm	Identifikator filma	long	Dio primarnog ključa

Tablica 1.7: Tablica ImaKategoriju

Atribut	Opis	Tip podatka	Ograničenja
idKategorija	Identifikator kategorije	long	Dio primarnog ključa
idFilm	Identifikator filma	long	Dio primarnog ključa

Tablica 1.8: Tablica Funkcija

Atribut	Opis	Tip podatka	Ograničenja
idFunkcija	Identifikator funkcije	long	Primarni ključ
imeFunkcija	Ime funkcije	String	Jedinstvena vrijednost (unique), Maksimalna duljina = 50

Tablica 1.9: Tablica Osoba

Atribut	Opis	Tip podatka	Ograničenja
idOsoba	Identifikator osobe	long	Primarni ključ
imeFunkcija	Ime osobe	String	Jedinstvena vrijednost (unique), Maksimalna duljina = 150

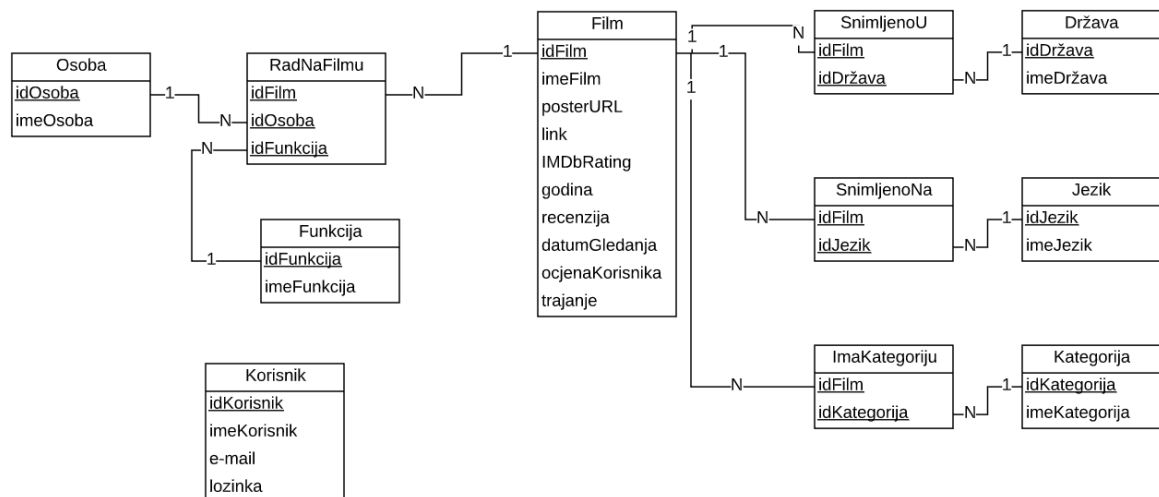
Tablica 1.10: Tablica RadNaFilmu

Atribut	Opis	Tip podatka	Ograničenja
idOsoba	Identifikator osobe	long	Dio primarnog ključa
idFilm	Identifikator filma	long	Dio primarnog ključa
idFunkcija	Identifikator funkcije	long	Dio primarnog ključa

Tablica 1.11: Tablica Korisnik

Atribut	Opis	Tip podatka	Ograničenja
idKorisnik	Identifikator korisnika	long	Primarni ključ
e-mail	e-mail adresa korisnika	String	Jedinstvena vrijednost (unique), Minimalna duljina = 3, Maksimalna duljina = 50
imeKorisnik	Ime korisnika	String	Jedinstvena vrijednost (unique), Minimalna duljina = 4, Maksimalna duljina = 30
lozinka	Lozinka korisnika	String	Jedinstvena vrijednost (unique), Minimalna duljina = 4, Maksimalna duljina = 30

Relacijski model baze podataka prikazan je na slici 1.2.



Slika 1.2: Relacijski model baze podataka

#### 1.4.8. Stvaranje tablica i okidača

Uz Spring Boot nije potrebno pisati SQL definicije za pojedine tablice, već je dovoljno napisati razred koji će nositi ime tablice te mu dodati anotaciju „@Entity“ te će Spring Boot sam obaviti ostatak tako da atributi u razredu postanu atributi u tablici. Također se u samom razredu mogu pisati ograničenja na attribute te će se ta ograničenja preslikati na bazu.

Povezivanje tablica isto je vrlo jednostavno te je samo potrebno navesti ime druge tablice i atribut po kojim ih želimo povezati.

Spring Boot također omogućava da se ne trebaju pisati nekakve velike naredbe kako bi se stvorili okidači, nego se to obavlja samo pomoću jedne naredbe, što uvelike skraćuje vrijeme pisanja i ne zahtijeva nikakvo znanje SQL sintakse.

U nastavku su prikazani programski odsječci za pojedine razrede iz kojih su onda stvorene SQL tablice.

Tablica 1.12: Programski odsječak stvaranja tablice Film i njenih okidača

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre> @Entity public class Movie {     @Id     @GeneratedValue     private Long id;     @Size(max = 1000)     private String name;     @Size(max = 2000)     private String poster_url;     @Size(max = 2000)     private String link;     @Min(0)     @Max(10)     private String IMDb_rating;     @Min(1900)     @Max(2100)     private String year;     @Size(max = 50000)     private String review;     private Date dateViewed;     @Min(0)     @Max(10)     private int myRating;     private int duration;      @OneToMany(cascade = CascadeType.ALL) ,     mappedBy = "movie"     Set&lt;WorksOnMovie&gt; worksOnMovie;      @OneToMany(cascade = CascadeType.ALL) ,     mappedBy = "movie"     Set&lt;FilmedInCountry&gt; filmedInCountry;      @OneToMany(cascade = CascadeType.ALL) ,     mappedBy = "movie"     Set&lt;FilmedOnLanguage&gt; filmedOnLanguage;      @OneToMany(cascade = CascadeType.ALL) ,     mappedBy = "movie"     Set&lt;HasCategory&gt; hasCategory; } </pre>	<p>Stvaranje tablice Film</p> <p>Popis atributa tablice i ograničenja tih atributa</p> <p>Povezivanje s drugim tablicama i stvaranje okidača</p>



Tablica 1.13: Programski odsječak stvaranja tablice Država

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre> @Entity public class Country {     @Id     @GeneratedValue     private Long id;      @Column(unique = true)     @Size(max = 50)     private String countryName;      @OneToMany(cascade = CascadeType.ALL),     mappedBy = "country"     Set&lt;FilmedInCountry&gt; filmedInCountry; } </pre>	<p>Stvaranje tablice Država</p> <p>Stupac imeDržave mora biti „unique“ jer je sporedni ključ</p> <p>Povezivanje s drugim tablicama i stvaranje okidača</p>

Tablica 1.14: Programski odsječak stvaranja tablice ImaKategoriju

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre> @Entity public class HasCategory {     @EmbeddedId     private HasCat id;      @ManyToOne     @MapsId ("movieIdCat")     private Movie movie;      @ManyToOne     @MapsId ("categoryIdCat")     private Category category; } </pre>	<p>Stvaranje tablice ImaKategoriju</p> <p>Povezivanje s tablicom Film</p> <p>Povezivanje s tablicom Kategorija</p>

Tablica 1.15: Programski odsječak stvaranja tablice RadNaFilmu

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre> @Entity public class WorksOnMovie {     @EmbeddedId     private WorksOn id;      @ManyToOne     @MapsId ("movieIdWorks")     private Movie movie;      @ManyToOne     @MapsId ("personIdWorks")     private Person person;      @ManyToOne     @MapsId ("roleIdWorks")     private Role role; } </pre>	<p>Stvaranje tablice RadNaFilmu</p> <p>Povezivanje s tablicom Film</p> <p>Povezivanje s tablicom Osoba</p> <p>Povezivanje s tablicom Funkcija</p>

Tablica 1.16: Programski odsječak stvaranja tablice Korisnik

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre> @Entity public class RegisteredUser {     @Id     @GeneratedValue     private Long id;      @Column(unique = true)     @Size(min = 3, max = 50)     private String email;      @Column(unique = true)     @Size(min = 4, max = 30) </pre>	<p>Stvaranje tablice Korisnik</p> <p>Stupac email mora biti „unique“ jer je sporedni ključ</p> <p>Stupac korisničko ime mora biti</p>

<pre>private String username;  private String password;  }</pre>	<p>„unique“ jer je sporedni ključ</p>
--	---

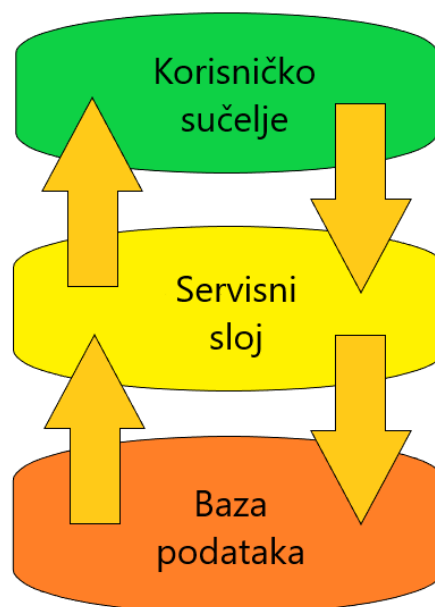
## 2. Implementacija web aplikacije

### 2.1. Arhitektura aplikacije

Sustav za recenziju filmova ima 3 sloja:

- Korisničko sučelje – zaduženo za komunikaciju aplikacije s vanjskim svijetom
- Servisni sloj – zadužen za obradu podataka koji dolaze iz korisničkog sučelja ili baze i prosljeđivanje obrađenih podataka na ostale slojeve
- Baza podataka – u nju se spremaju podaci koje koristi aplikacije

Slojevi sustava za recenziju filmova prikazan je na slici 2.1.



*Slika 2.1: Slojevi sustava za recenziju filmova*

Za izradu servisnog sloja korišten je razvojni okvir Spring Boot. Spring Boot se koristi za stvaranje mikro servisa. Mikro servisi su tip arhitekture koja omogućava razvoj

servisa neovisno jedni o drugima. Drugim riječima, svaki servis ima svoj vlastiti proces koji se vrti neovisno o procesima drugih servisa. [4]

Mikro servisi nude niz raznih prednosti u razvoju aplikacije, kao što su:

- Jednostavna skalabilnost
- Kompatibilnost s kontejnerima
- Minimalno konfiguriranje
- Kraće vrijeme razvoja

## **2.2. Izvod arhitekture aplikacije**

Servisni sloj sastoji se od 3 dijela:

- Repozitoriji
- Servisi
- Upravljači

### **2.2.1. Repozitoriji**

Služe za komunikaciju s bazom bez eksplicitne potrebe pisanja SQL naredba kako bi se došlo do zapisa u bazi.

Tablica 2.1: Programski odsječak repozitorija za tablicu Film

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre> public interface MovieRepository extends JpaRepository&lt;Movie, Long&gt; {      int countByName (String name);      int countById (long id);      int countByLink (String link);  } </pre>	<p>Stvaranje repozitorija</p> <p>Stvaranje metoda koje će se vršiti kao SQL naredbe nad bazom podataka</p>

Još jedna posebnost Spring Boota je ta što je dovoljno implementirati sučelje sa imenima metoda koje se zovu kao operacije koje bi vršili u SQL naredbama u bazi podataka. Nema nikakve potrebe za implementacijom metoda sučelja.

### 2.2.2. Servisi

Primaju podatke od upravljača, provode validaciju i poslovnu logiku te pozivaju repozitorije za manipulaciju podacima.



Tablica 2.3: Programski odsječak upravljača za stvaranje filma uz pomoć poveznice poslane od strane korisnika

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre> public class MovieController {      @PostMapping("/movies")      @Secured("ROLE_USER")      public String createMovieTemplate(@RequestBody String link, @AuthenticationPrincipal User u) {          return movieService.createMovieTemplate(link, u.getUsername());      } } </pre>	<p>Stvaranje upravljača</p> <p>Postavljanje putanje na kojoj će biti poslana poveznica</p> <p>Samo registrirani korisnik smije poslati poveznicu</p> <p>U tijelu poruke nalazi se poveznica te ime korisnika koji ju je poslao</p> <p>Pozivanje metode servisa zaduženog za stvaranje filma</p>



Tablica 2.4: Programski odsječak upravljača za stvaranje popisa filmova koji će biti sortirani prema naslovu i poslani korisniku

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre> public class MovieController {      @GetMapping("/movies/byTitle")      @Secured("ROLE_USER")      public List&lt;Movie&gt; listMoviesByTitle(@AuthenticationPrincipal User u) {      return movieService.listAllByTitle (u.getUsername());  } } </pre>	<p>Stvaranje upravljača</p> <p>Postavljanje putanje na koju će biti poslan popis filmova</p> <p>Samo registrirani korisnik smije poslati poveznicu</p> <p>U tijelu poruke nalazi se ime korisnika koji je poslao zahtjev za popisom</p> <p>Pozivanje metode servisa zaduženog za stvaranje popisa sortiranog popisa filmova</p>

## 2.2.4. Sigurnost aplikacije

Spring Boot ima nekoliko mehanizama zaštite aplikacije. Ova web aplikacija koristi jedan od osnovnih, ali učinkovitih mehanizama. Radi sigurnosti, u bazi podataka se ne čuva lozinka korisnika koju je dao pri registraciji, već sažetak (engl. *hash*) te lozinke od 60 znakova. Proces autentifikacije korisnika vrši se tako da se sažetak lozinke dan pri autentifikaciji uspoređuje sa sažetkom lozinke za ponuđeno korisničko ime u bazi. Uz to, korisnicama se daju uloge pri ulazu u aplikaciju. Korisniku koji ima ime „admin“ daje se uloga administratora, ali i uloga korisnika, dok se svim ostalim korisnicima daje uloga korisnika. Određene operacije mogu se provoditi u ovisnosti o ulozi koju korisnik ima.

Tablica 2.5: Programski odsječak autentifikacije korisnika

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre>public class MovieUserDetailsService implements UserDetailsService {      @Override     public UserDetails loadUserByUsername(String username) {          if ("admin".equals(username)) return new User (username, adminPasswordHash, commaSeperatedStringToAuthorityList ("ROLE_ADMIN", "ROLE_USER"))          for (RegisteredUser r : registeredUserRepo. findAll())</pre>	<p>Stvaranje razreda za autentifikaciju korisnika</p> <p>Provjera radi li se o administratoru te pridavanje odgovarajućih uloga</p> <p>Prolazak kroz sve korisnike i pregled odgovara</p>

<pre> {     if (r.getUsername().equals(username)) return new User (username, r.getPassword(), commaSeperatedStringToAuthorityList ("ROLE_USER")); }  return null; } </pre>	<p>li autentifikacija nekome od korisnika te pridavanje uloge korisnika</p> <p>Povratak u slučaju da traženi korisnika ne postoji</p>
--	---

Aplikacija treba vršiti validaciju podataka kako bi bila sigurna da su podaci ispravno navedeni i točni. To se odvija na svakom od tri sloja kako bi se minimizirala mogućnost pogreške.

### 2.2.5. Sloj korisničkog sučelja

Korisničko sučelje razvijeno je pomoću JavaScripta te biblioteke React, a izgled sučelja je ostvaren pomoću tehnologija HTML5 i CSS. React je JavaScript biblioteka otvorenog zadužena za izgradnju korisničkih sučelja. Svaka stranica aplikacije prima podatke iz aplikacije ili šalje u aplikaciju. Za to se koriste neke od sljedećih metoda:

- GET – preuzimanje podataka od aplikacije
- POST – ubacivanje zapisa u aplikaciju
- PUT – mijenjanje zapisa u aplikaciji
- DELETE – brisanje zapisa u aplikaciji

JavaScript poziva te metode pri određenim akcijama korisnika na sučelju.

Tablica 2.6: Programski odsječak korisničkog sučelja za registriranje korisnika

PROGRAMSKI ODSJEČAK	KOMENTAR
<pre> var currentLocation = window.location; await fetch(currentLocation, {   method: "POST",   headers: {     "Accept": "application/json",     "Content-Type": "application/json",   },   body: JSON.stringify ({     "id": "0",     "username": document.getElementById("username").value,     "password": document.getElementById("password").value,     "email": document.getElementById("email").value,    }) }).then(response =&gt; response.text())   .then (body=&gt; {   if(body === "User registered")   {     Window.location.replace("http://localhost:3000");   }   else{     alert (body);   } })  } </pre>	<p>Pozivanje metode POST na trenutnoj adresi</p> <p>Tijelo metode koje ima podatke o novom registriranom korisniku</p> <p>Ako je korisnik registriran, bit će odveden na naslovnu stranicu aplikacije, ako ne onda će doći upozorenje s porukom koja ovisi o grešci</p>

### **3. Rad s aplikacijom**

Tijekom rada s aplikacijom, korisnik će se susresti s nekoliko procesa, a to su:

- Registracija korisnika
- Autentifikacija korisnika
- Dodavanje filmova
- Sortiranje filmova
- Pretraživanje filmova
- Pretraga statistike filmova
- Pregled pojedinačnog filma
- Ažuriranje filma
- Brisanje filma

#### **3.1. Registracija korisnika**

Registracija korisnika provodi se tako da se, u za to predviđena polja, unesu podaci o novom korisniku te će se, ako registracija biti uspješna, korisnik proslijediti na početnu stranicu aplikacije. U slučaju problema pri registraciji, korisniku će se ispisati odgovarajuća poruka.

Na slici 3.1 prikazana je stranica za registraciju.

The image shows a registration form titled "Registracija" centered on a dark background with a repeating diamond pattern. The form is a white rectangle containing the following elements:

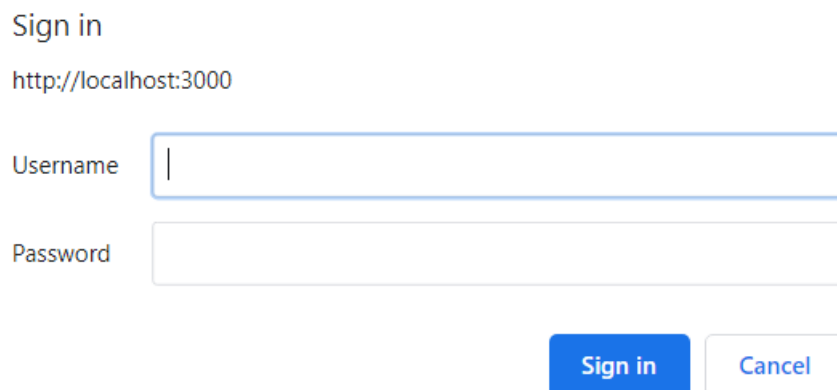
- Title: **Registracija**
- Field 1: "Korisničko ime:" followed by a text input box.
- Field 2: "e-mail adresa:" followed by a text input box.
- Field 3: "Lozinka:" followed by a text input box.
- Field 4: "Potvrdi lozinku:" followed by a text input box.
- Button: A small button labeled "Registracija" located below the input fields.

*Slika 3.1: Stranica za registraciju*

### **3.2. Autentifikacija korisnika**

Autentifikacija korisnika provodi se na početnoj stranici aplikacije, što znači da nema posebne stranice za autentifikaciju korisnika. To postupak čini jednostavnijim i omogućuje korisniku da pristupi početnoj stranici bez da ga se prosljeđuje. Korisnik će, nakon upisivanja točnog korisničkog imena i lozinke, odmah moći vidjeti svoju početnu stranicu aplikacije.

Na slici 3.2 prikazan je obrazac za autentifikaciju na prvoj stranici aplikacije.



Sign in

http://localhost:3000

Username

Password

*Slika 3.2: Obrazac za autentifikaciju*

### 3.3. Dodavanje filmova

Filmovi se dodaju tako da se na vrh stranice zalijepi poveznica IMDb stranice na kojoj se nalazi film. Nakon toga korisnik treba pritisnuti gumb za dodavanje filma, nakon čega će se provjeriti validnost poveznice koju je zalijepio. Ako je poveznica validna i taj film se već ne nalazi u bazi korisnika, stranica će se osvježiti, nakon čega će se na naslovnici vidjeti slika dodanog filma.

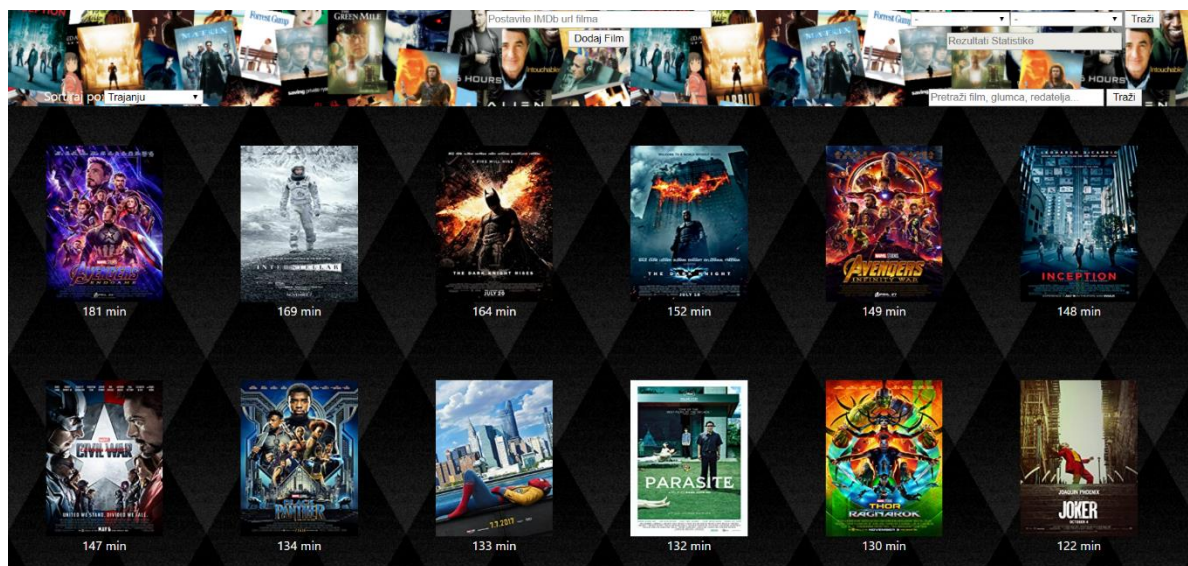
### 3.4. Sortiranje filmova

Filmovi se sortiraju tako da se odabere jedna od opcija u padajućem izborniku, nakon čega će se stranica osvježiti i filmovi će biti poredani po opciji koja je odabrana. Ispod slike filma pisat će parametar filma po kojem je bio sortiran. Mogući parametri su:

- Datum gledanja
- Naslov
- IMDb ocjena
- Korisnikova ocjena
- Godina
- Trajanje

Ako niti jedan parametar neće biti naveden, ispod filmova pisat će naslov filma.

Na slici 3.3 prikazana je početna stranica aplikacije na kojoj su filmovi sortirani prema trajanju.



*Slika 3.3: Početna stranica korisnika sa filmovima sortiranim prema trajanju*



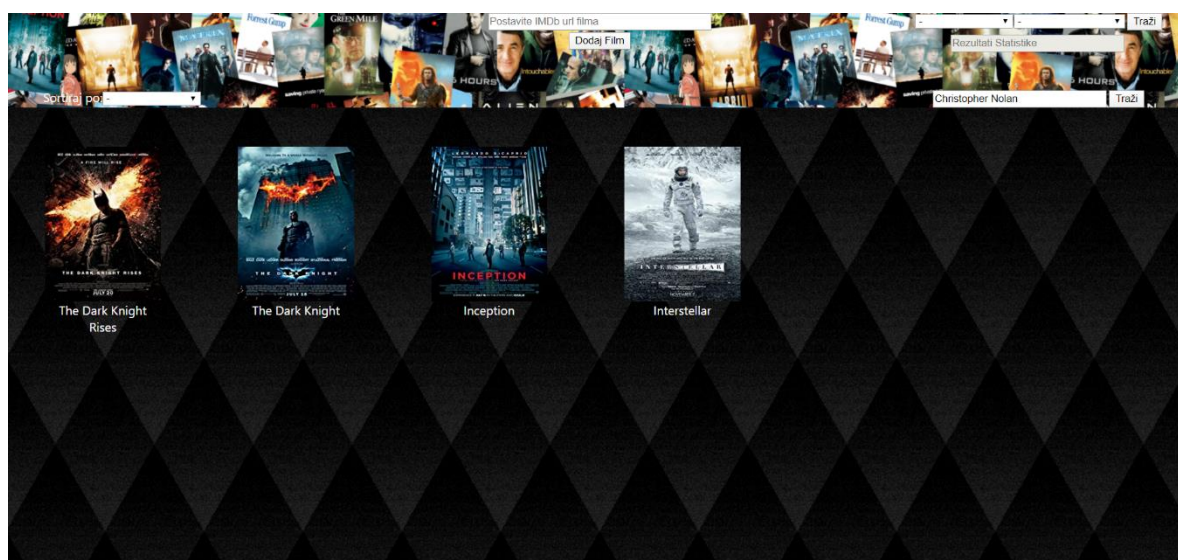
### 3.5. Pretraživanje filmova

Na naslovnoj stranici nalazi se traka za pretragu, gdje korisnik može upisati neki pojam te će se taj pojam pretražiti i filmovi koji sadrže pojam bit će navedeni.

Parametri po kojima je moguća pretraga su:

- Film
- Glumac
- Redatelj
- Scenarist
- Država
- Jezik

Na slici 3.4 prikazana je korisnikova početna stranica aplikacije nakon što se vršila pretraga nad filmovima na kojima je radio Christopher Nolan.



*Slika 3.4: Rezultat pretrage za filmovima na kojima je radio Christopher Nolan*

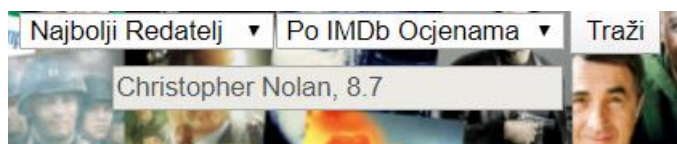
### 3.6. Pretraga statistike filmova

U gornjem desnom kutu početne stranice nalaze se dva padajuća izbornika kojima je moguće odabrati pregled statistike nad cijelim skupom korisnikovih filmova. Opcije koje korisnik može odabrati su:

- Najbolji/najgori glumac po IMDb/korisnikovoj ocjeni
- Najbolji/najgori redatelj po IMDb/korisnikovoj ocjeni
- Najbolji/najgori scenarist po IMDb/korisnikovoj ocjeni

Nakon odabira parametara i pritiska na gumb za pretragu, korisnik dobiva statističku informaciju koju je tražio.

Na slici 3.5 prikazan je rezultat pretrage statistike za najboljeg redatelja po IMDb ocjeni te prosječna ocjena tog redatelja.

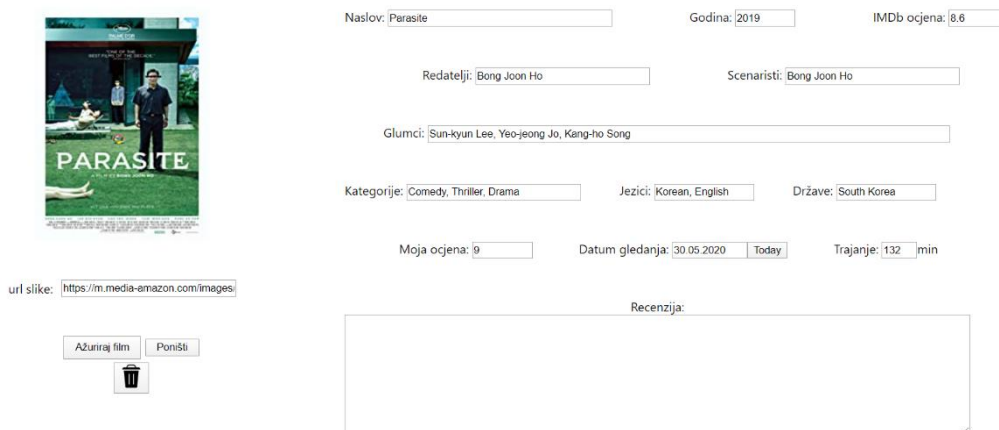


*Slika 3.5: Rezultati pretrage statistike za najboljim redateljem po IMDb ocjeni te prosječna ocjena tog redatelja*

### 3.7. Pregled pojedinačnog filma

Pritiskom na sliku filma na početnoj stranici, korisnik će biti proslijeđen na stranicu sa podacima o filmu.

Na slici 3.6 prikazana je stranica sa informacijama o filmu Parazit.



Naslov:  Godina:  IMDb ocjena:

Redatelj:  Scenaristi:

Glumci:

Kategorije:  Jezici:  Države:

Moja ocjena:  Datum gledanja:   Trajanje:  min

Recenzija:

url slike:

Slika 3.6: Prikaz stranice filma Parazit

### 3.8. Ažuriranje i brisanje filma

Podaci o filmu mogu se u bilo kojem trenutku promijeniti od strane korisnika. Korisnik treba unijeti ažurirane podatke o filmu te kliknuti na gumb za ažuriranje. Korisnik će

nakon toga biti prosljeđen na naslovnu stranicu filma. Parametri filma koje korisnik može promijeniti su:

- Naslov
- Godina
- IMDb ocjena
- Poveznica na kojoj se nalazi slika filma
- Redatelji
- Scenaristi
- Glumci
- Kategorije
- Jezici
- Države
- Korisnikova ocjena
- Datum gledanja
- Trajanje
- Recenzija

Korisnik može i odustati od ažuriranja filma pritiskom na gumb za poništavanje. Sve promjene nakon prošlog ažuriranja bit će u tom slučaju izgubljene.

Korisnik može obrisati film pritiskom na gumb sa slikom koša za smeće. Svi podaci o filmu bit će u tom slučaju izgubljeni, ali film se može u svakom trenutku ponovno dodati nakon čega će ponovno imati podatke o filmu sa IMDb stranice, bez korisnikovih prethodnih ažuriranja.

## **Zaključak**

Cilj ovog završnog rada bio je izraditi bazu podataka i web aplikaciju za recenziju filmova. Posebnost ove aplikacije bila je povlačenje podataka o filmovima sa IMDb web stranice i to samo preko poveznice na tu stranicu. Podaci koji se smatraju zanimljivim za korisnika su bili spremljeni u bazu podataka iz koje su se onda povlačili filmovi.

Korisničko sučelje napravljeno je tako da bude jednostavno i intuitivno, ali i da izgleda zanimljivo i privlačno za korištenje. Korisnik u aplikaciju može dodati koliko god filmova želi, ali može i upravljati postojećim filmovima ažuriranjem i brisanjem istih. Postupak upravljanja filmovima je brz i jednostavan. Filmovi se mogu ocjenjivati i komentirati. Aplikacija također osigurava da niti jedan korisnik ne može vidjeti podatke i filmove drugih korisnika.

## Literatura

[1] FER: Baze podataka, Predavanja, ožujak 2020.

[2] H2 sustav za upravljanje bazom podataka

Poveznica:

<http://www.h2database.com/html/features.html> (27.5.2020)

[3] Model podataka

Poveznica: [https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d310\\_polaznik.pdf](https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d310_polaznik.pdf) (27.5.2020)

[4] Spring Boot

Poveznica:

[https://www.tutorialspoint.com/spring\\_boot/spring\\_boot\\_introduction.htm](https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm) (28.5.2020)

## Sažetak

### Baza podataka i web aplikacija za recenziju filmova

U sklopu ovog završnog rada osmišljeni su baza podataka i web aplikacija za recenziju filmova. Podaci o filmovima se uzimaju sa IMDb stranice te se nakon toga spremaju u bazu. Napravljeno je intuitivno korisničko sučelje kojim je lako rukovati te koji omogućuje jednostavnu komunikaciju s ostatkom aplikacije. Aplikacija omogućava korisniku pregled, unos, brisanje i ažuriranje filmova. Ispravnost podataka provjerava se u svim slojevima aplikacije. Aplikacija koristi sustav za upravljanje bazom podataka H2. Aplikacija je realizirana korištenjem programskih jezika Java i JavaScript, razvojnim okvirom Spring Boot te bibliotekom React. Pri tome su korištene i tehnologije HTML5 i CSS za izradu sučelja.

**Ključne riječi:** recenzija filmova, aplikacija, Spring Boot, JavaScript, Java, H2, React, HTML5, CSS, korisničko sučelje

## Summary

### Database and Web Application for reviewing movies

In this paper, a database and web application for reviewing movies were designed. The movie data is taken from the movie page on the IMDb website and stored into the database. An intuitive, easy-to-use user interface was made, and it allows easy communication with the rest of the application. The application allows the user to view, insert, delete, and update movies. The validity of the information is checked on every application layer. The application uses the database management system H2. The application was made using the programming languages Java and JavaScript, the Spring Boot framework and library React. The HTML5 and CSS technologies were used for making the user interface.

**Keywords:** movie review, application, Spring Boot, JavaScript, Java, H2, React, HTML5, CSS, user interface