

SERENGETI TECH d.o.o.

PRAKSA

CINEMARK – Web Aplikacija

Bjelovar, 2023.

SERENGETI TECH d.o.o.

PRAKSA

Studenti:

Erik Baukovac

Karlo Feltrin

Filip Pavliš

Marko Žuna

TEHNIČKA DOKUMENTACIJA

Mentor:

Nenad Kovačević

Scrum Master:

Goran Barunić

Bjelovar, svibanj 2023.

Sadržaj

1.	Tehnička dokumentacija.....	1
2.	Korištene tehnologije i alati.....	2
3.	ERA model	3
4.	Dijagram slučajeva korištenja.....	5
5.	Glavne Funkcionalnosti Web Aplikacije	7
5.1.	Search Film	7
5.2.	Film Rating.....	9
5.3.	Commenting	11
5.4.	Bookmarking.....	14
5.5.	Recommended Films	16
5.6	Login and Registration.....	19
5.7.	Adding Movies.....	21
5.8.	Sort By	23
5.9.	Filtering.....	27
6.	Popis slika	30

1. Tehnička dokumentacija

U ovome dokumentu opisat će se tehnička dokumentacije web aplikacije CineMark. CineMark je movie review web site koji omogućuje korisnicima pregled raznih filmova. Web aplikacija nudi mnogo različitih funkcionalnosti koje će se u sljedećim poglavljima prikazati i detaljnije objasniti.

Najprije će se objasniti i navesti sve tehnologije i alati korišteni pri izradi ovog projekta. Nakon toga će se prikazati ERA model web aplikacije, tj. model baze podataka. Bit će prikazana grafička slika modela baze, te će se objasniti tablice i povezanosti odnosno relacije među njima. Poslije toga će se prikazati dijagram slučajeva korištenja. Web aplikacija nudi 3 različite korisničke uloge: obični korisnik, admin i super-admin. U tom poglavlju dat će se objašnjenje za svaku pojedinu ulogu. Osim toga, kao što je gore već spomenuto, opisat će se glavne funkcionalnosti web aplikacije. Svaka funkcionalnost bit će prikazana dijagramom aktivnosti, te će također biti potkrijepljena daljnjim objašnjenjem procesa same funkcionalnosti.

2. Korištene tehnologije i alati

Za izradu web aplikacije CineMark korišteno je više alata. Baza podataka je izrađena u Microsoft SQL Serveru. Ona je relacijska baza podataka, te joj je primarni jezik za upite Transact SQL, što znači da osim klasičnih i standardnih SQL upita dozvoljava neke složenije radnje poput mijenjanja programskog toka i drugo.

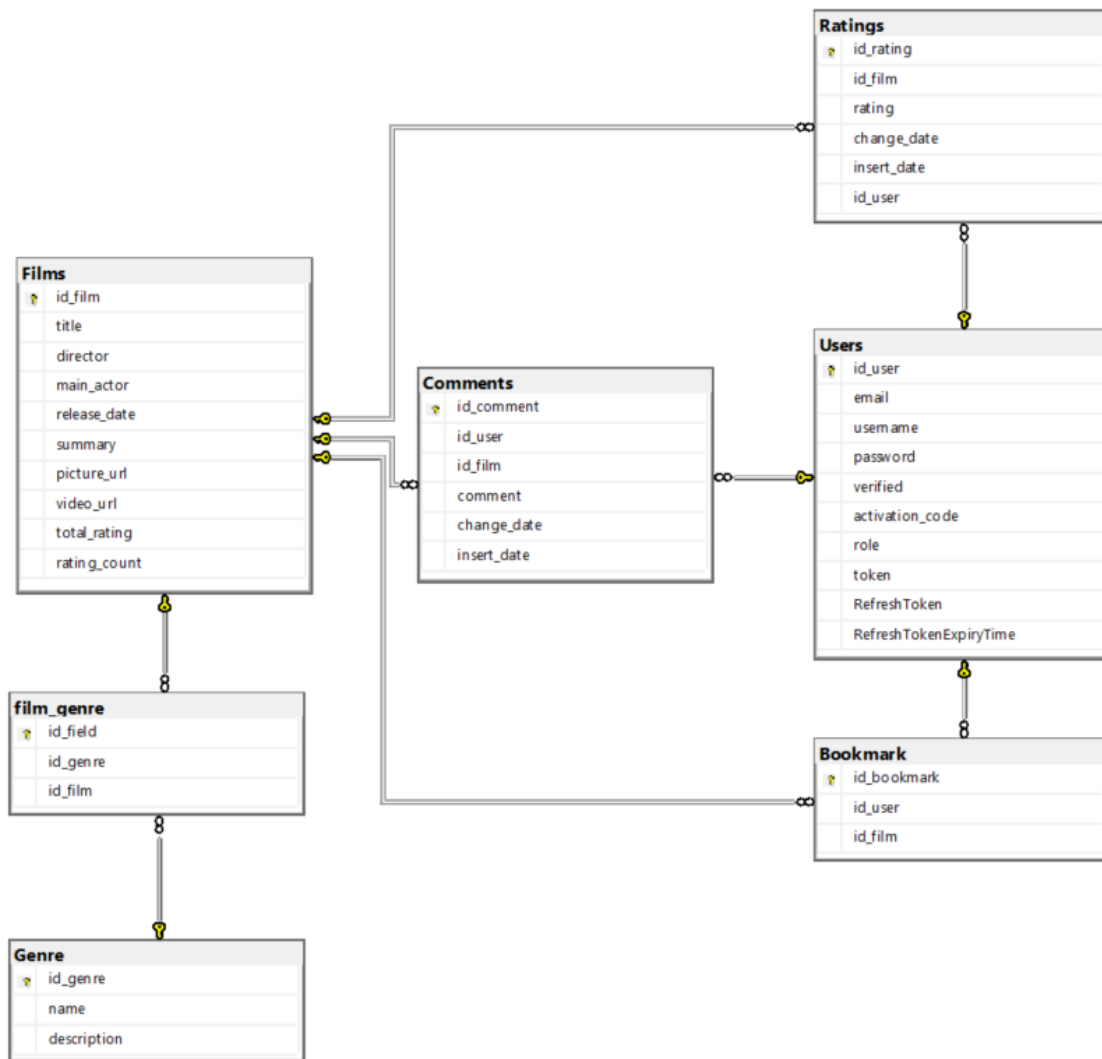
Za backend dio projekta korišten je Visual Studio 2022 Community Edition. U njemu je napravljen Web API projekt (ASP.NET Core 7.0 verzija) koji je pisan C# programskim jezikom.

Front end dio web aplikacije izrađen je pomoću platforme Angular. Verzija koja se koristila je sljedeća: Angular CLI-15.2.6, Angular-15.2.8. Angular je besplatni okvir web-aplikacije otvorenog koda. Korišteni jezik prilikom izrade frontend dijela je TypeScript. HTML i CSS su također korišteni radi izrade potrebnog dizajna web stranice.

Pošto je projekt rađen u timu od četiri studenata, kako bi se ujedinio kod i kako bi se lakše navigiralo postupnom izradom aplikacije korišten je alat Sourcetree. To je Git GUI koji pruža vizualnu reprezentaciju odabranih repozitorija. Za održavanje dva repozitorija: frontenda i backenda, korišten je GitLab (gitlab.com).

Pomoć pri izradi ove dokumentacije pružio je alat Visual Paradigm – Online (online.visual-paradigm.com), pomoću kojega su izrađeni svi dijagrami prikazani u sljedećim poglavljima. Za dobivanje screenshotova ekrana poslužio je alat Lightshot (app.prntscr.com).

3. ERA model



Slika 1. ERA Model

ERA dijagram nam služi kako bi detaljno prikazali kako će izgledati struktura baze podataka. Naša baza podataka sastoji se od šest glavnih entiteta i jednog pomoćnog. Glavni entiteti su: Users, Films, Comments, Ratings, Bookmark i Genre, a pomoćni entitet je film_genre.

Films entitet je entitet oko kojeg se vrti cijela aplikacija. Atributi entiteta Films su primarni ključ id_film tipa int, title tipa varchar 255, director tipa varchar 255, main_actor tipa varchar 255, release_date tipa date, summary tipa varchar 255, picture_url tipa varchar 255, video_url tipa varchar 250, total_rating tipa float i rating_count tipa int.

Users je entitet koji koristimo za kontrolu pristupa stranici i lakše pratimo komentare i ocijene filmova. Users se sastoji od primarnog ključa id_user tipa int, email tipa varchar 255, username tipa varchar 255, password tipa varchar 255, verified tipa bit, activation_code tipa nvarchar 50, role tipa varchar 255, token tipa varchar max, RefreshToken tipa varchar max i RefreshTokenExpiryTime tipa datetime.

Rating je entitet koji se koristi za praćenje ocijena filma. Rating ima vanjske ključeve na entitete Films i Users jer preko njih znamo koji je korisnik dao kojem filmu ocjenu i tako ga ujedno i ograničimo da ne ocijeni isti film više puta. Rating ima attribute id_rating koji je primarni ključ tipa int, vanjski ključ id_film tipa int, rating tipa float, change_date tipa date, insert_date tipa date i vanjski ključ id_user tipa int.

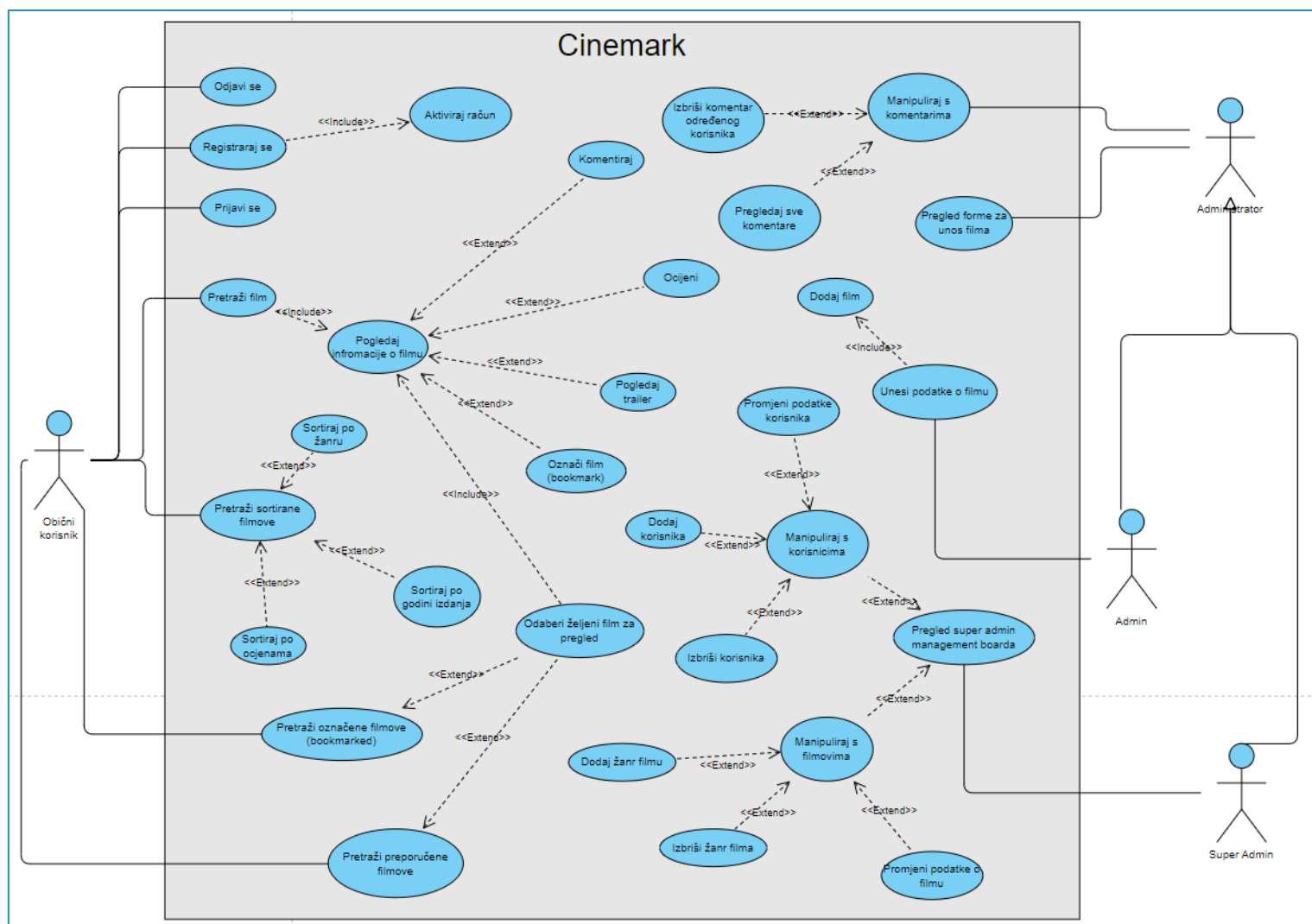
Entitet Bookmark koristi se za praćenje označenih filmova korisnika. Atributi ovog entiteta su primarni ključ id_bookmark tipa int i dva vanjska ključa tipa int id_film i id_user.

Comments entitet služi za spremanje komentara korisnika za određeni film. Atributi ovog entiteta su primarni ključ tipa int id_comment, vanjski ključevi tipa int id_film i id_user, comment tipa nvarchar max, change_date tipa date i insert_date tipa date.

Genre entitet je entitet za dodavanje žanrova u bazu podataka on sadrži attribute tipa int id_genre koji je ujedno i primarni ključ, name tipa varchar 255 i description tipa varchar 255.

Jedini pomoćni entitet film_genre stvoren je zbog veze više na više između entiteta Films i Genre. Atributi ovog entiteta su primarni ključ id_field tipa int i dva vanjska ključa tipa int id_genre i id_film.

4. Dijagram slučaja korištenja



Slika 2. Dijagram Slučajeva Korištenja

Web aplikacija Cinemark ima više uloga s kojima se može pristupiti aplikaciji. Svaka uloga ima svoja prava i mogućnosti korištenja. Postoje 3 različite uloge: **obični korisnik**, te **administrator** koji se dijeli na **admina** i **super-admina**. Obični korisnici su uglavnom svi krajnji korisnici aplikacije. Kada netko stvori novi račun, automatski postaje krajnji korisnik, odnosno obični korisnik. Administrator ima veći stupanj mogućnosti upravljanja aplikacijom i ima dozvolu pristupa mogućnostima koje krajnji korisnik ni ne vidi. U sljedećem poglavlju bit će opisane sve mogućnosti koje određena uloga nudi. Prvo će se opisati obični korisnik.

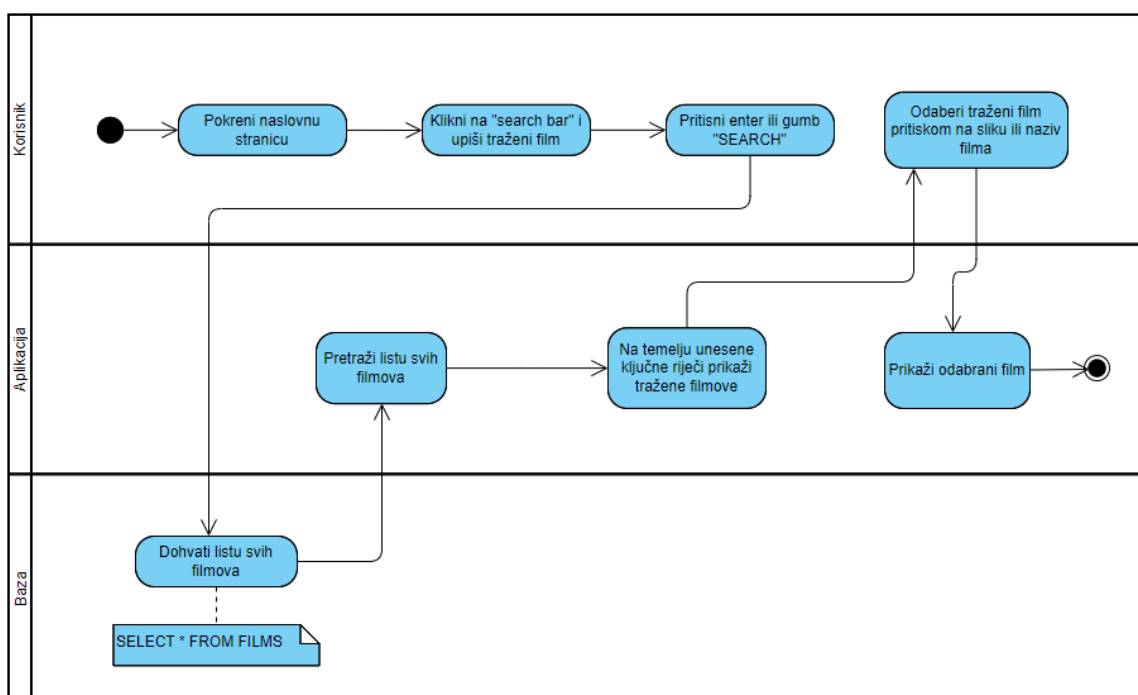
Obični korisnik prvo se mora registrirati i aktivirati račun kako bi se uspješno mogao prijaviti. Nakon prijave korisnik može koristiti neke funkcionalnosti koje ne bi mogao ako nije prijavljen kao što je ocijeniti neki određen film, pisati komentare, označavati filmove itd... Bez prijave, znači svatko tko pristupi web aplikaciji, može pretraživati filmove pomoću search bara, pogledati informacije o određenom filmu, pogledati trailer filma, te sortirati filmove po žanru, ocjenama i godini izdanja. Obični prijavljeni korisnik, na drugu ruku, može i prilikom pregleda određenog filma dati ocjenu filmu od jedan do pet. Tu ocjenu može promijeniti kada god hoće. Svakom filmu može dati samo jednu ocjenu. Osim toga, može iznositi svoja mišljenja o filmu pišući različite komentare. Ako korisnik odluči da mu se sviđa film i želi ga staviti u svoj privatni popis, može film i označiti (bookmark funkcionalnost). Kasnije kada pogleda taj film, može ga iz popisa obrisati. Ispod popisa označenih filmova (bookmark tablice), nalazi se sekcija u kojoj obični korisnik može pregledavati i istražiti njemu preporučene filmove (4 preporučena filma svaki put kad se osvježi stranica).

Administrator ima dvije razine: **admin i super-admin**. Svaki user koji ima ulogu administratora (admin i super-admin) ima pristup formi za unos filma, te može pregledavati i brisati komentare. **Admini** samo mogu unositi podatke o filmu i dodati novi film u bazu, dok super-admin ima veću razinu mogućnosti. **Super-admin** ima pristup super-admin management boardu. Na njemu može manipulirati s korisnicima: pregled svih korisnika, promjena njihovih podataka, brisanje korisnika, te mijenjanje uloge korisnika. Osim toga, on ima veću razinu upravljanja filmovima. Naime, super-admin može i mijenjati podatke o filmu, brisati krivo unesene žanrove, te unositi nove žanrove određenom filmu.

5. Glavne Funkcionalnosti Web Aplikacije

U ovome poglavlju bit će opisati glavne funkcionalne Cinemark web aplikacije. Svaka funkcionalnost će biti prvo prikazana dijagramom aktivnosti. Nakon toga slijedi opis te funkcionalnosti uz detaljnije informacije i jasna objašnjenja.

5.1. Search Film



Slika 3. Activity Dijagram Search Funkcionalnosti

Kako bi korisnici mogli pretraživati određene filmove, na naslovnoj stranici (home page) napravljen je search bar koji služi za pretraživanje raznih filmova koji su dostupni u bazi web aplikacije. Search funkcionalnost započinje kada korisnik napiše traženu riječ pomoću koje želi pretraživati, te klikne na gumb „search“ ili jednostavno klikne enter na tipkovnici. Najprije se iz baze vraća lista svih dostupnih filmova, te se onda poziva funkcija koja služi za dohvaćanje pretraženih filmova. Naziv te funkcije je „getSearchedFilms“.

```

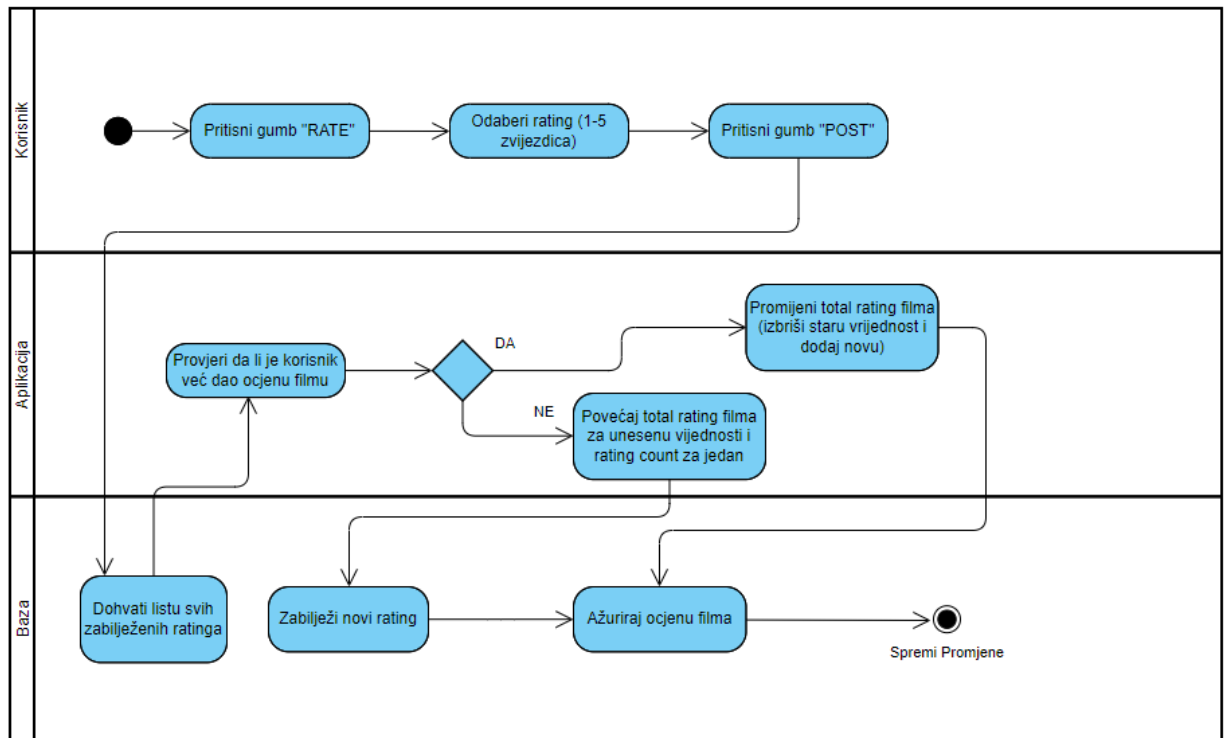
getSearchedFilms(searchInput:string){
    this.searchFilms = [];
    for(var item of this.list){
        if(searchInput.length == 1){
            if(searchInput[0].toLowerCase() == item.title[0].toLowerCase()){
                this.searchFilms.push(item);
            }
        }
        if(searchInput.length==2){
            if(searchInput[0].toLowerCase() == item.title[0].toLowerCase() && searchInput[1].toLowerCase() == item.title[1].toLowerCase()){
                this.searchFilms.push(item);
            }
        }
        if(searchInput.length>2 && item.title.toLowerCase().includes(searchInput.toLowerCase())){
            if(this.searchFilms.includes(item) != true){this.searchFilms.push(item)};
        }
    }
}

```

Slika 4. Funkcija „getSearchedFilms“

U glavi funkcije, kao argument, proslijeđen je unos u obliku stringa koji je korisnik utipkao u search bar. Na početku ove funkcije se inicijalizira prazna lista odnosno polje u koju će se kasnije pohraniti rezultati pretraživanja i koja će se nakon toga prikazati na ekranu. Kako bi se pronašli svi filmovi po ključnoj riječi korisnika, potrebno je iterirati svaki film iz liste svih filmova. Ovaj search radi na metodi prvih slova u imenu filma. Primjerice, ako korisnik upiše slovo „a“, ispisati će mu se svi filmovi koji počinju na to slovo. Nadalje, ako bi korisnik upisao samo prva dva slova, ispisat će mu se svi filmovi koji počinju na ta prva dva slova. Ta dva slučaja rade na principu uspoređivanja indeksa stringa naziva filma i korisnikovog unosa. Jedino što je tu bitno je da se svaki unos prebaci u lower case, te da se onda uspoređuje. Kada se uvjet zadovolji, novi film se stavlja u listu pretraženih filmova. Ako korisničkog unos ima više od 3 slova, pokreće se nova funkcija pretraživanja koja se naziva „includes“. Ta funkcija jednostavno traži u imenu filma postoji li podudaranost između unosa i naziva filma. Ako postoji, film se stavlja u listu pretraženih filmova. Tu funkcija završava i korisniku se prikazuju rezultati pretraživanja, odnosno prikazuje mu se lista pretraženih filmova.

5.2. Film Rating



Slika 5. Activity Dijagram Rating Funkcionalnosti

Funkcionalnost ocjenjivanja filmova započinje kada korisnik pritisne na gumb „RATE“ na određenom filmu. Tada mu se nudi izbornik između ocjene od 1-5 zvijezdica. Nakon što korisnik odabere svoj rating, može kliknuti na gumb „POST“ gdje se pokreće funkcija koja će izvesti funkcionalnost ratinga određenog filma. Ovisno o tome da li je korisnik već ocjenio film ili ne, događaju se dvije različite radnje. Prije toga, mora se znati da li korisnik mijenja svoju ocjenu ili unosi novu ocjenu. To će se odlučiti funkcijom nazvanom „checkIfVoted“. Na sljedećoj slici može se vidjeti kako ta funkcija izgleda:

```

checkIfVoted(){
  const film=this.storageService.getFilm();
  const ratings = this.storageService.getRatings();
  this.check = false;

  for(var rating of ratings){
    if(film.id_film == rating.id_film && parseInt(this.id_user$) == rating.id_user){
      this.check = true;
      this.changeRating = rating.rating;
    }
  }
}

```

Slika 6. Funkcija „checkIfVoted“

Ako je user već votao, varijabla check će se postaviti na true. Pomoću te varijable sada glavna funkcija „postNewRating“ koji blok radnji treba izvesti:

```

postNewRating(newRating:number){
  const film= this.storageService.getFilm();
  this.selectedFilm=film;
  this.ratingService.getRatings();
  this.checkIfVoted();

  if(this.check == false) {
    this.increaseRating = this.selectedFilm.total_rating + newRating;
    this.increaseCount = this.selectedFilm.rating_count + 1;

    this.selectedFilm.total_rating = this.increaseRating;
    this.selectedFilm.rating_count = this.increaseCount;

    this.storageService.saveFilm(this.selectedFilm);

    this.formData = this.selectedFilm;

    this.http.put(`${this.baseUrl}/${this.formData.id_film}`,this.formData).subscribe();

    this.ratingService.postRating(this.formData, newRating);

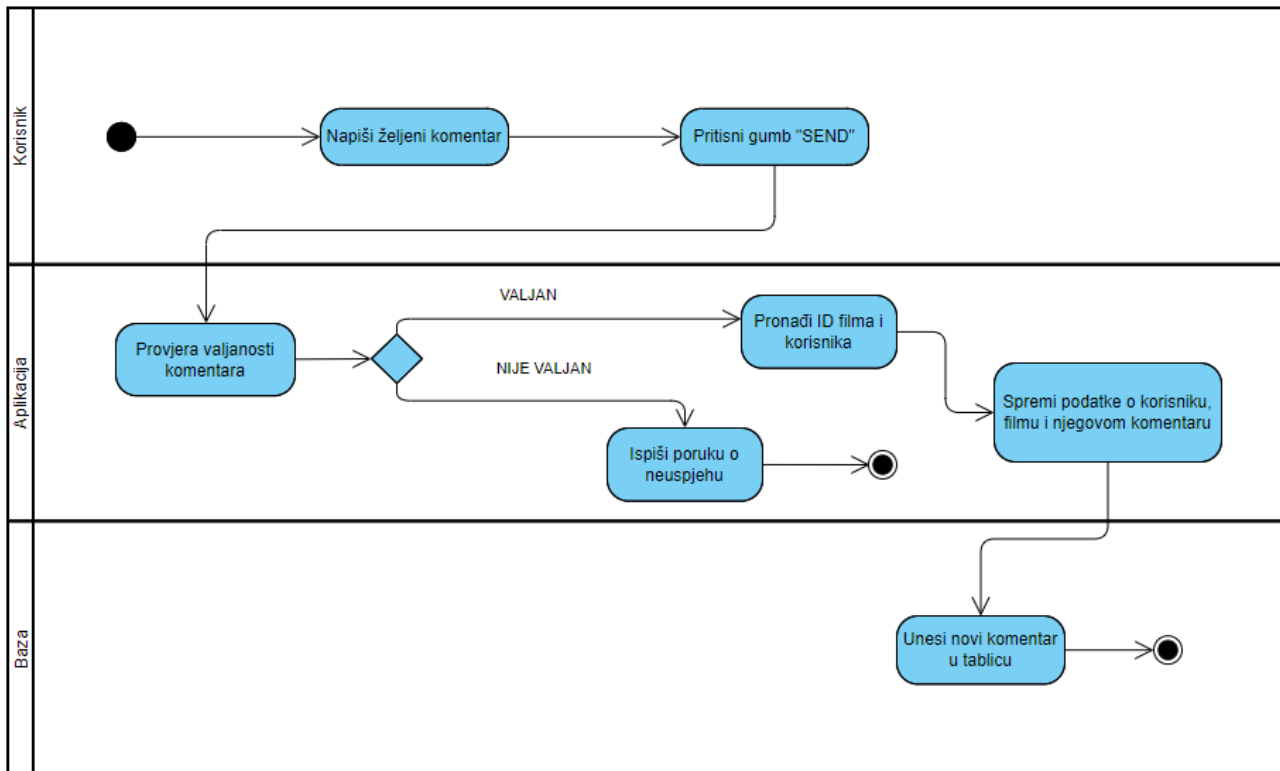
    this.updateRating();
    this.toastr.success('Ratings saved.','Thank you!')
  }
  else{
    this.increaseRating = this.selectedFilm.total_rating + newRating - this.changeRating;
    this.selectedFilm.total_rating = this.increaseRating;
    this.formData = this.selectedFilm;
    this.storageService.saveFilm(this.selectedFilm);
    this.http.put(`${this.baseUrl}/${this.formData.id_film}`,this.formData).subscribe();
    this.ratingService.postRating(this.formData, newRating);
    this.updateRating();
    this.toastr.success('Rating changed!','Thank you for voting.');
```

Slika 7. Funkcija „postNewRating“

Ako korisnik prvi put ocjenjuje neki film, total rating tog filma će se povećati za njegov odabir. Nakon toga će se rating count povećati za još jednog korisnika, dakle za jedan. Te sve promjene će se ažurirati u bazi u tablici tog određenog filma pomoću metode `http.put()`. Nakon toga će se također zabilježiti i novi rating tog korisnika metodom „`postRating`“ u kojoj se nalazi jedna `http.post()` metoda koja sprema novi rating. Kada se sve uspješno provede, korisnik dobiva poruku: „Ratings saved. Thank you!“

U slučaju da je korisnik već ocjenio film, total rating tog filma će se umanjiti za staru vrijednost i povećati za novu vrijednost. Rating count se neće promijeniti jer se u ovom slučaju samo mijenja rating, ne dodaje se novi rating. Ta promjena se također ažurira u bazi pomoću `http.put()` metode, te se nakon toga zabilježi taj novi rating promijenjen od strane korisnika pomoću `http.post()` metode. Na kraju se korisniku ispuše poruka: „Rating changed! Thank you for voting.“ Tu završava rating funkcionalnost.

5.3. Commenting



Slika 8. Aktivirij Dijagram Add Comment Funkcije

Svaki korisnik može pisati željene komentare za određeni film. Kada korisnik napiše željeni komentar treba pritisnuti gumb „SEND“ kako bi se komentar objavio. Tada se pokreće funkcija „saveComment()“.

```
async saveComment() {  
    const val = document.getElementById("commentInput") as HTMLInputElement;  
    var userComment = val.value.trim();  
  
    if (userComment != null && userComment != "" && userComment != " ") {  
        this.commentService.postComment(this.selectedFilm, userComment);  
        val.value = "";  
  
        await this.UpdateLists();  
    }  
    else  
        this.toastr.error("Comment must not be empty");  
}
```

Slika 9. Funkcija „saveComment“

U njoj se prvo pokreće provjera valjanosti komentara. Ta provjera se pokreće radi toga da ne bi neki korisnik slučajno pritisnuo gumb „SEND“ bez da je što komentirao. Kada ne bi bilo te provjere korisnik bi mogao slati prazne komentare, zato se u toj provjeri provjerava da li je komentar uistinu prazan ili nije. Ako je prazan, ispisuje se poruka: „Comment must not be empty.“ U slučaju da komentar nije prazan, pokreće se funkcija postComment koja za argumente uzima taj određeni film na kojem korisnik komentira i korisnikov uneseni komentar.

```

postComment(film:Films, comment:string){
  const currentDate = new Date();
  if(this.storageService.isLoggedIn()==true){
    this.formData.change_date = currentDate;
    this.formData.insert_date = currentDate;
    this.formData.id_film = film.id_film;
    this.formData.comment = comment;
    this.formData.id_user = +this.id_user$;

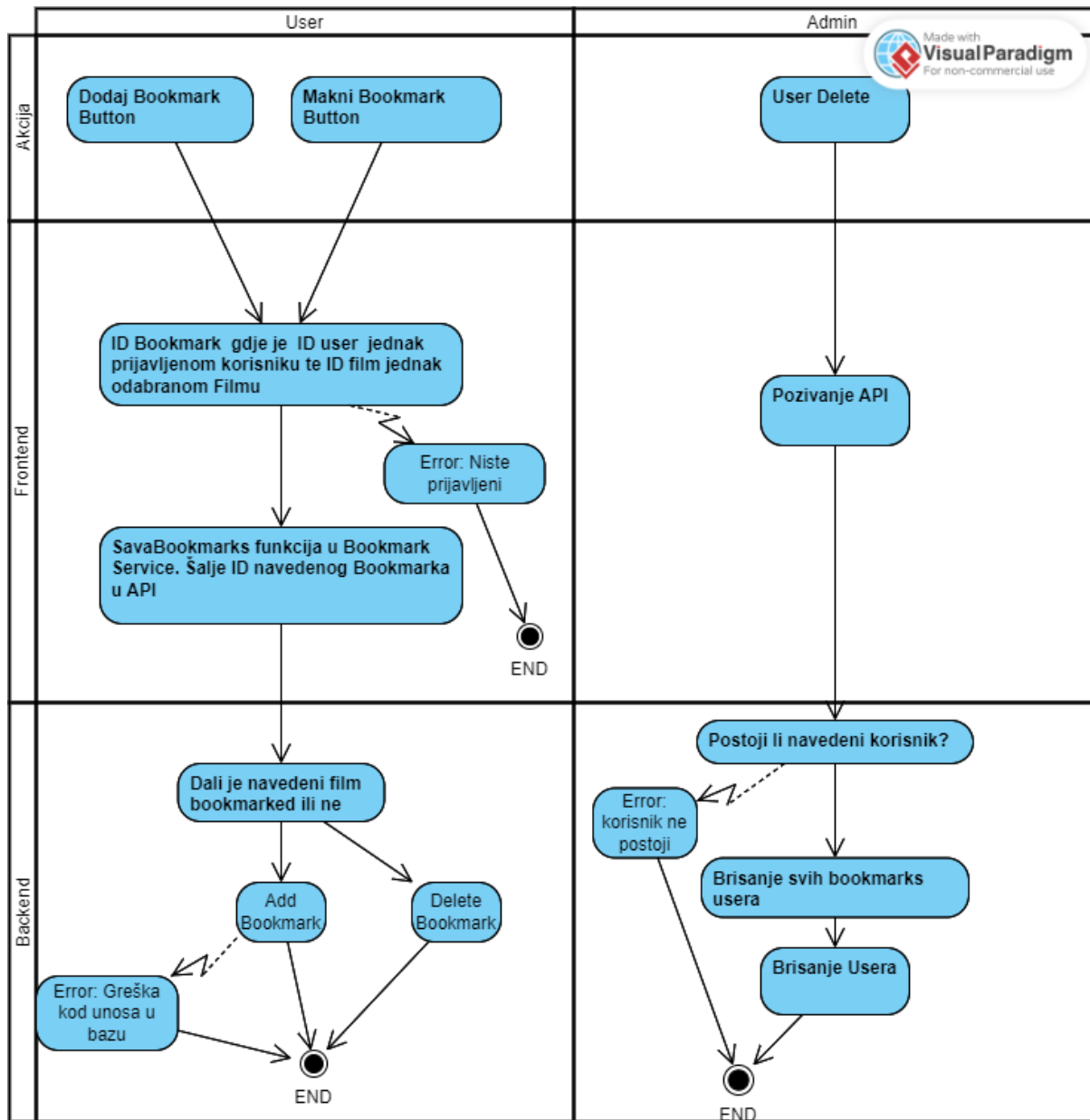
    this.http.post(this.baseURL,this.formData).subscribe();
    this.toastr.success("Comment posted.", "Success!")
  }
  else{
    this.toastr.error("You Can't comment if you are not logged in");
    console.error("You Can't comment if you are not logged in");
  }
}

```

Slika 10. Funkcija „postComment“

Funkcija postComment prvo sprema sve podatke u varijablu tipa comment nazvanu formData. Prije nego što to dogodi, provjerava se da li je korisnik uopće prijavljen. Ako nije, ispisati će se error poruka: „You can't comment if you are not logged in.“ Ako je, svi potrebni podaci o novom komentaru i sam komentar će se spremiti u bazu http.post metodom te će se ispisati poruka o uspjehu: „Success! Comment posted.“ Tu završava funkcionalnost komentiranja.

5.4. Bookmarking



Slika 11. Activity Dijagram Bookmark Funkcionalnosti

Bilo koji korisnik koji je registriran te ima aktiviran account; bio User , Admin ili Superadmin. Ima mogućnost spremanja omiljenih filmova u formi Bookmark, Može spremiti neograničen broj filmova u bookmarks te ih potom i maknuti.

Ako Superadmin izbriše Usera potom se brišu Bookmarks istog usera, Admin i Superadmin ne mogu u samoj aplikaciji brisati korisniku Bookmarks ili mu davati nove filmove u Bookmarks.

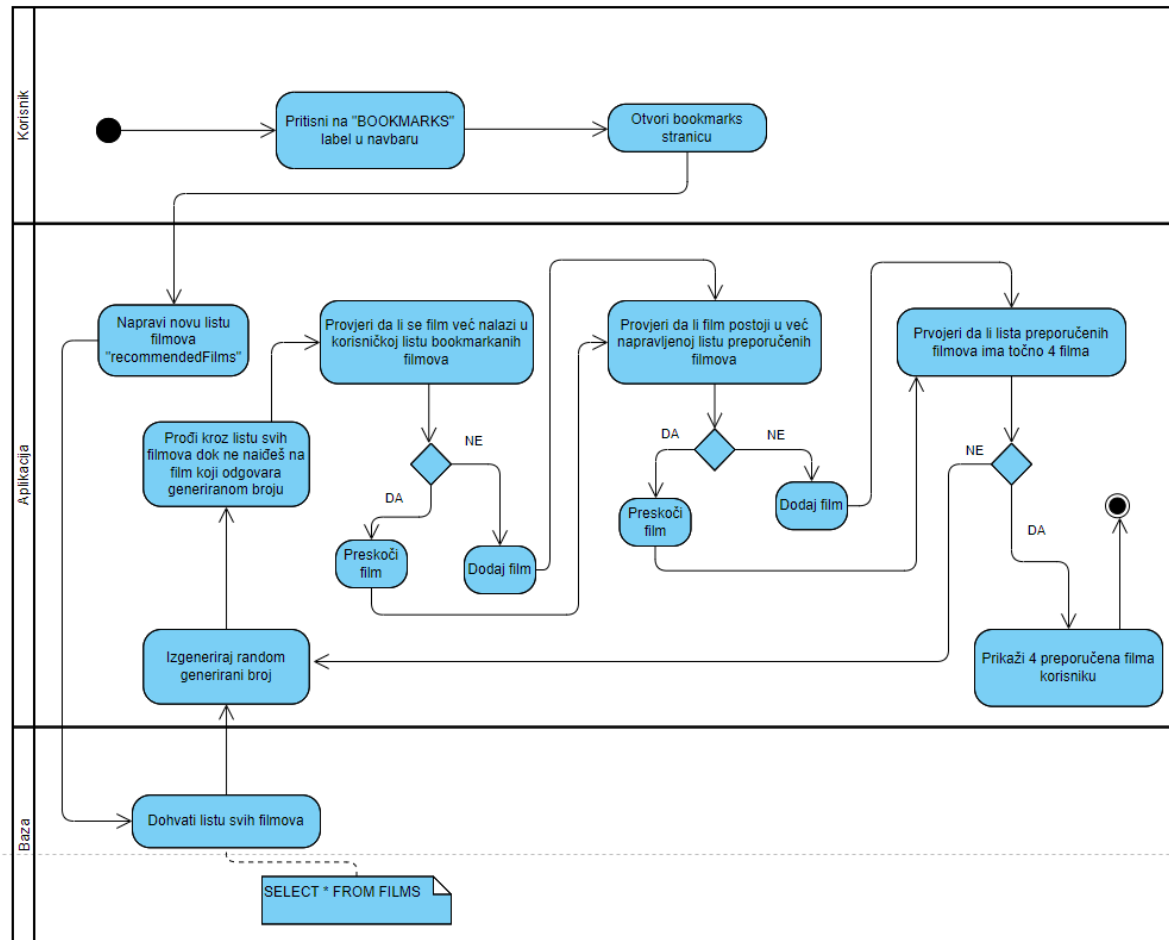
Kada korisnik pritisne button za Bookmark koji ima dva stanja "Bookmark" i "UnBookmark" uzima se ID_User te ID_Film Usera koji je prijavljen te filma koji je odabran te se šalju u API api/Bookmark/Find te potom api/Bookmark/Post. Ovdje dolazi do mogućeg EXCEPTION gdje korisnik nije prijavljen te se akcija terminira i ispisuje se poruka da se korisnik prvo mora prijaviti. Ako je korisnik prijavljene nastavlja se gore navedeno.

Pokreće se Find (authBookmark) API koji provjeri li postoji Bookmark s navedenim ID_User i ID_Film ako postoji onda ga briše, javlja se konsol log s potvrdom te se button na stranici promjeni u praznu verziju sebe koja indicira da film više nije bookmarked.

U protivnom ako Bookmark s nevedenim ID_User i ID_Film ne postoji onda se pokreće API Post (postBookmarks) koji kreira novi Bookmark s ID_Bookmark, ID_User te ID_Film vrijednostima te ih sprema u bazu, ovdje je moguć još jedan EXCEPTION u slučaju da dođe do greške unosa u bazu. Ovdje Akcija završava.

Druga moguća akcija s Bookmarks je kod Superadmin koji ima mogućnost brisanja User te se također izbrišu njegovi Bookmarks. u Backend se poziva API api/Bookmark/Delete Delete, kojem se šalje ID_User. Provjerava se dali postoji User s navedenim ID_User ako ne javlja se EXCEPTION te se terminira Akcija, u slučaju da postoji prvo se brišu svi Bookmarks Gdje je ID_User isti navedenom ID_User, isto se radi i za njegove komentare te se potom briše User. Ovdje 2. Akcija završava.

5.5. Recommended Films



Slika 12. Activity Dijagram Recommended Films

Sekcija „preporučeni filmovi“ je jedna od dodataka na web stranici. Na bookmark stranici, ispod korisničkove tablice označenih filmova, nalaze se 4 preporučena filma. Svaki put kada korisnik refresha page, prikaže mu se randomly odabrana nova 4 preporučena filma. Dakle, funkcija „getRecommendedMovies“ pokreće se dolaskom na bookmarks stranicu pritiskom na labelu „bookmarks“ u navigation baru. Ta funkcija stavlja u listu „recommendedFilms“ četiri nasumično odabrana nova filma s time da poštuje neke uvjete definirane u njoj. Na sljedećoj slici može se vidjeti cijela funkcija:

```

getRecommendedMovies() {
  this.recommendedFilms = [];
  const films = this.storageService.getFilms();
  const mybookmarks = this.storageService.getMyBookmarks()
  do {
    //stavi checker na false u svakoj iteraciji
    this.check = false;

    //generiranje random broja od 0-20
    let x = Math.round(Math.random() * 250);

    //prolaz kroz listu svih filmova
    for (var film of films) {

      //ako je id filma isti random generiranom broju
      if (film.id_film == x) {

        //provjera da li se ponavlja isti film u recommended listi
        for (var reclist of this.recommendedFilms) {
          if (reclist.id_film == film.id_film) {
            this.check = true;
          }
        }

        //provjera da li korisnik ima bookmarkan film (ako da onda neće biti u recommended listi)
        for (var booklist of mybookmarks) {
          if (booklist.id_film == film.id_film) {
            this.check = true;
          }
        }

        //ako je su sve provjere uspješno prošle dodaj recommended film u listu
        if (this.check == false) {
          this.recommendedFilms.push(film);
        }
      }
    }
    //ponavljaj sve dok nemamo 4 filma
  } while (this.recommendedFilms.length < 4);
}

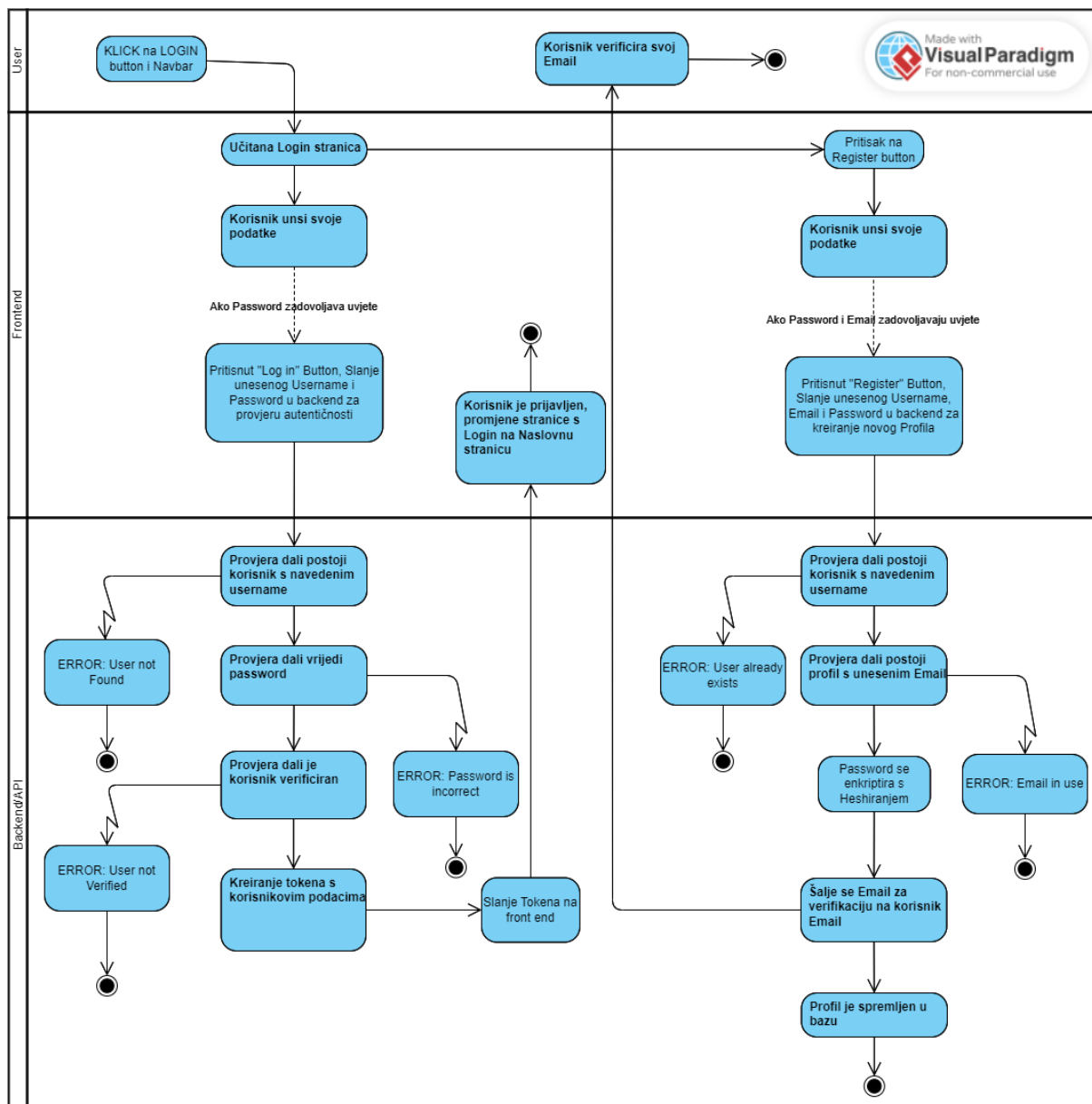
```

Slika 13. Funkcija „getRecommendedMovies“

Na početku funkcije se inicijalizira prazna lista „recommendedFilms“ koja će kasnije biti popunjena s četiri filma. Cijeli postupak stavljanja filma u listu nalazi se u do...while petlji koja ima uvjet da se ponavlja sve dok prijašnje inicijalizirana prazna lista ne bude imala četiri elemenata u njoj, odnosno četiri filma. Dakle, kada to polje napokon bude imalo četiri elemenata, do...while petlja se prekida. Na početku petlje se varijabla check postavlja na false. Ona nam pomaže da se u uvjetu odluči hoće li film biti stavljen u listu ili ne. Osim što sve u svakoj iteraciji varijabla check postavlja na false, generira se i random broj koji će nam služiti da se odabere film po nekoj slučajnoj odabranoj vrijednosti. Nakon toga se počne

iterirati lista svih filmova. Kada napokon id filma bude isti random generiranom broju, slijede dvije provjere. Prva provjera provjerava da li se ponavlja isti film već u recommended listi jer bi bio glupo da se isti film prikaže dva puta u sekciji preporučenih filmova. Druga provjera provjerava da li se film nalazi u korisničkoj listi bookmarkanih filmova. Ne bi bilo primjereno da se korisniku prikaže film koji je on već označio, pa da se to spriječi koristi se ta provjera. U slučaju da su sve provjere uspješno prošle (varijabla check nije primila vrijednost true), film se dodaje u listu preporučenih filmova. Taj isti proces se ponavlja sve dok lista preporučenih filmova nema četiri elementa. Kada napokon dođe do četvrtog elemenata, do...while petlja završava te se korisniku na ekranu u sekciji preporučenih filmova prikazuju ti filmovi. Tu funkcionalnost recommended filmova završava.

5.6 Login and Registration



Slika 14. Activity Dijagram Login i Registracije

Prilikom pritiska tipke "LOGIN" na navigacijskoj traci, učitava se Login stranica koja omogućuje prijavu i registraciju korisnika. Korisnik zatim odabire dali se želi prijaviti ili registrirati pritiskom na "Log In" ili "Register" tipke pri vrhu forme.

Kod odabira Log in, korisnik unosi svoj "username" i "password" gdje password mora biti dulji od 4 znamenke, tipka Login sada se omogućiti te ju korisnik može pritisnuti, proces

prijave je sada pokrenut, podaci Username i Password spremljeni su u form te poslani u API `"/Users/authenticate"`. Prvo se napravi provjera dali uopće postoji korisnik s navedenim Username, ako ne proces se terminira te se javlja greška `"User not Found"`. Ako postoji username dalje se provjerava dali vrijedi password, u slučaju da password ne vrijedi javlja se greška `"Password is incorrect"` te se terminira proces. Kada su oboje uredi tada se provjerava dali je korisnik verificiran drugim riječima dali mu je aktiviran account, ako ne javlja se greška `"User not Verified"` te se terminira proces.

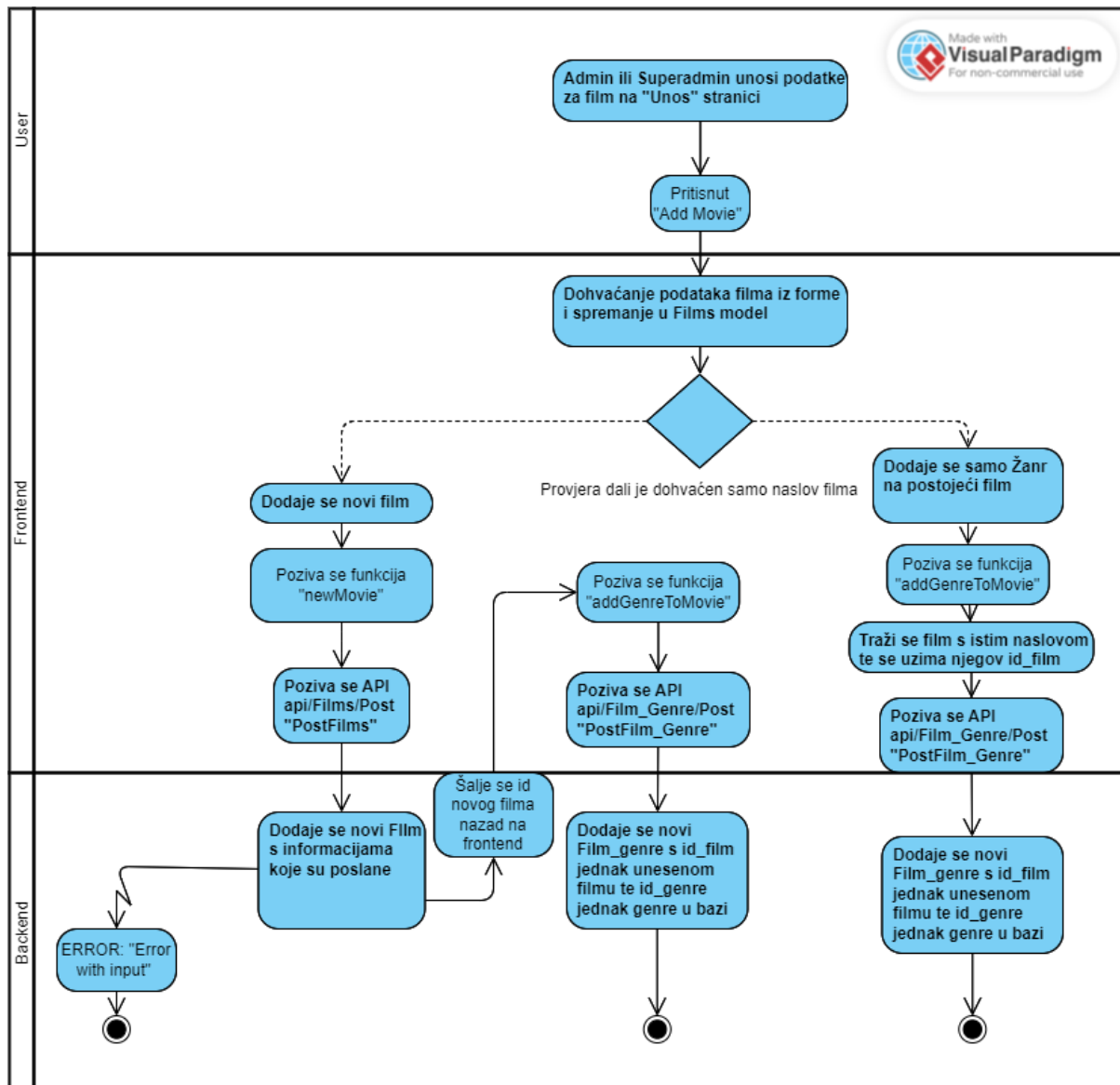
Kada su svi podaci uredi kreira se Token s ID od usera (`id_user`); Username, Email te Role i šalje se na front end, token se sprema u Session Storage te se korisnika šalje na Naslovnu stranicu te je proces prijave završen.

Kod odabira Register, korisnik unosi svoj `"username"`, `"email"` i `"password"`. Email mora biti važeći te password mora biti dulji od 4 znamenke. Klikom tipke Register uneseni podaci spremaju se u form te se šalju u API `"/Users/Register"`. Provjerava se dali postoji navedeni Username ako da javlja se greška `"User already exists"` te se terminira proces, ako ne postoji postoji proces nastavlja dalje, sljedeće se provjerava Email, ako se koristi javlja se greška `"Email in use"` te se terminira proces.

Ako su svi podaci uredni onda se enkriptira password Hashiranjem Rfc2898 metodom. Kreira se verifikacijski kod i šalje se korisniku na email, verifikacijski kod se također spremi u formu. Stavi se role User za korisnika i spremi u Form. Profil se spremi u bazu podataka te je proces gotov.

Kada korisnik verificira profil u bazi se samo promijeni stanje `"Verified"` polja iz `"False"` u `"True"`.

5.7. Adding Movies



Slika 15. Activity Dijagram Unosa Filma

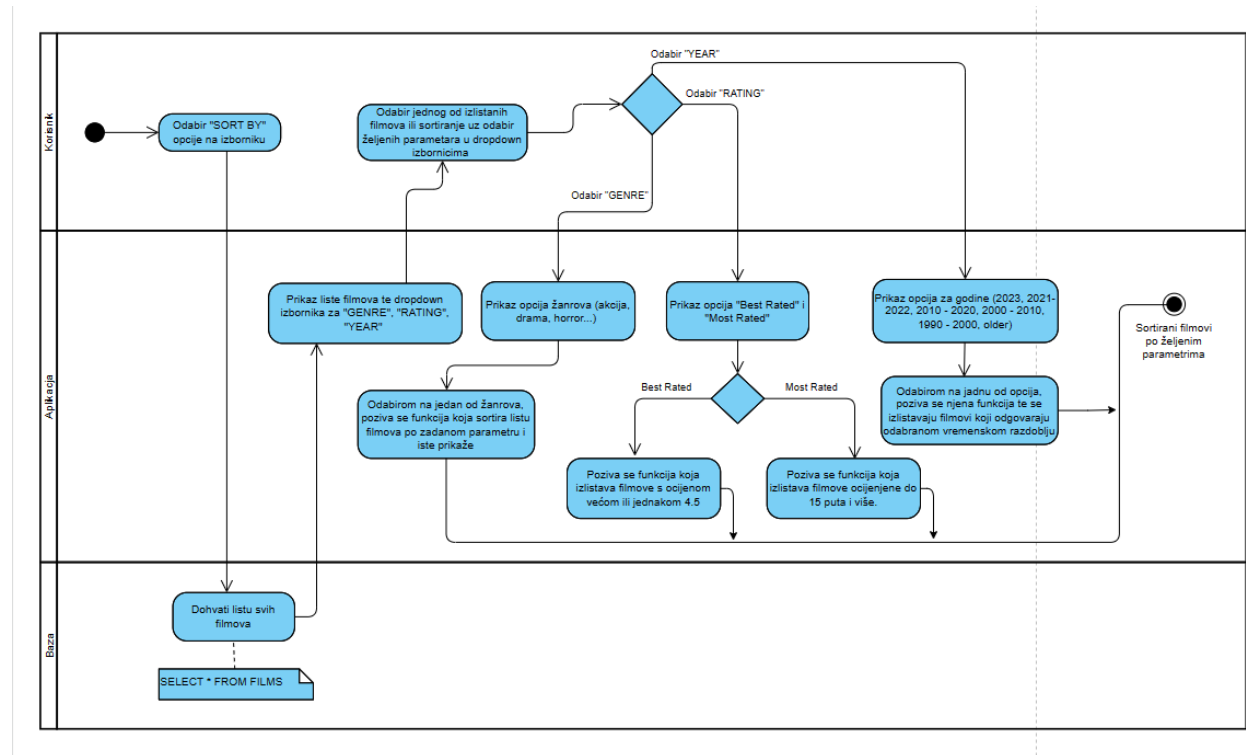
Admin i Superadmin mogu koristiti Unos stranicu, svrha je unos novih filmova u bazu podataka. Nakon što se unesu potrebne informacije kao što su: Naslov, Direktor, Glavni Glumac, Datum Izlaska, URL Slike, URL Trailera, Žanr i Kratak Opis. Korisnik pritisne "Add Movie" tipku te je akcija pokrenuta.

Nakon što su preuzeti podaci iz forme i spremljeni u Films model, gleda se dali je upisan samo Naslov filma ili svi podaci.

Ako su svi podaci poziva se newMovie funkcija koja poziva API `api/Films/Post "PostFilms"` te mu se šalju podaci ovog filma, automatski se kreira novi `id_Film` te se unose podaci, ako dođe do kreške kod unosa javlja se error "Error with input" moguće su nevažeći datum ili preveliki string za bazu, te se akcija terminira. Ako je uspješno unesen film onda se na Frontend vraća `id_Film` novo napravljenog filma te se sada poziva funkcija `addGenreToMovie` koja poziva API `api/Film_Genre/Post "PostFilm_Genre"`. te se dodaje novi `Film_Genre` koji sadrži `id_Genre` koji smo unijeli kao žanr novog filma te `id_Film` istog filma. Ovdje je akcija gotova te se terminira.

Ako je unesen samo Naslov filma, poziva se funkcija "addGenreToMovie", traži se `id_Film` s istim naslovom koji je unesen. Poziva se API `api/Film_Genre/Post "PostFilm_Genre"` te se kreira novi `Film_genre` s `id_genre` unesenog genre te `id_Film` unesenog Filma. Ovdje je akcija gotova te se terminira.

5.8. Sort By



Slika 16. Activity Dijagram Sort By Funkcionalnosti

Kako bi korisnici imali dodatne opcije i načine za pretraživanje filmova dostupnih u bazi web aplikacije, napravljena je „Sort By“ stranica. Odabirom te stranice, pred korisnikom se otvara stranica koja uz izlistane filmove iz baze također prikazuje tri dropdown izbornika i jedan checkbox.

Prvi dropdown izbornik je „Genre“ izbornik. Kada korisnik mišom hover-a na izbornik, pred njim se izlistaju svi dostupni parametri izbornika, u ovom slučaju svi žanrovi dostupni u web aplikaciji. Kada korisnik odabere žanr po svojoj želji, primjerice „Action“ žanr, iz baze se pozivaju svi filmovi koji odgovaraju tom žanru te se izlistavaju na stranici. Ta funkcionalnost postiže se pomoću „sortFilmByGenre“ funkcije.

```

sortFilmByGenre(gnre:Genres){
    var g = document.getElementById(gnre.name) as HTMLAnchorElement;
    var c = document.getElementById("checkbox") as HTMLInputElement;
    if(c.checked){g.style.color = '#00ffe58e';}
    this.filteredFilms = [];
    this.sortedFilms = [];
    const filmgenre=this.storageService.getFilmGenres()
    const films= this.storageService.getFilms()
    for(var item of filmgenre){
        if(gnre.id_genre == item.id_genre){
            for(var film of films){
                this.check = false;
                if(item.id_film == film.id_film){
                    this.sortedFilms.push(film);
                }
            }
        }
    }

    this.filterFilms();
    this.displayFilteredFilms();
}

```

Slika 17. Funkcija „sortFilmByGenre“

Unutar ove funkcije, na početku se nalazi inicijalizirana prazna lista „sortedFilms“ u koju će se na kraju spremati odgovarajući filmovi. For petlja prolazi sve dostupne predmete i gleda je li ID žanra iz modela Genres jednak ID-u žanra koji je pridodan ostalim karakteristikama filma. U slučaju da ID ne odgovara, filmovi se ignoriraju, a ako ID odgovara, onda se ti filmovi spremaju u „sortedFilms“ listu i izlistavaju se na web stranici.

Drugi dropdown izbornik je „Rating“ izbornik, gdje korisnik ima dvije opcije: „Best Rated“ i „Most Rated“. Odabirom „Best Rated“ opcije, korisnik brzo ima predodžbu koji su filmovi najbolje ocijenjeni na Cinemark web aplikaciji i može sa sigurnošću znati da se radi o kvalitetnim filmovima vrijednim njegova vremena. Odabirom „Most Rated“ opcije, pred korisnikom se izlistaju filmovi koji bilježe najveći broj ocjenjivanja, to jest, filmovi koje su korisnici web aplikacije najčešće ocjenjivali. Tako korisnik ima predodžbu koji su filmovi najpopularniji među korisnicima web aplikacije. Funkcionalnost ovih opcija postiže se korištenjem „sortFilmByBestRated“ i „sortFilmByRatingCount“ funkcijama.

```

sortFilmByBestRated(){
    var b = document.getElementById("BestRated") as HTMLAnchorElement;
    var c = document.getElementById("checkbox") as HTMLInputElement;

    if(c.checked){b.style.color = '#00ffe58e';}
    this.sortedFilms = [];
    this.filteredFilms = [];

    for(var item of this.service.list){
        if((item.total_rating)/(item.rating_count) >= 4.5){
            this.sortedFilms.push(item);
        }
    }

    this.filterFilms();
    this.displayFilteredFilms();
}

```

Slika 18. Funkcija „sortFilmByBestRated“

U „sortFilmByBestRated“ funkciji, provjerava se je li ocjena filma veća ili jednaka 4.5. Nakon što se prođe lista filmova i pronađu se filmovi gdje se ta tvrdnja pokaže istinitom, ti se filmovi spremaju u „sortedFilms“ listu koju smo inicijalizirali ranije u funkciji i filmovi se prikazuju pred korisnikom na web stranici. Kako bi dobili ocjenu filma od 1 do 5, potrebno je podijeliti broj ocjenjivanja („rating_count“) sa totalnim iznosom ocjenjivanja („total_rating“).

```

sortFilmByRatingCount(){
    var m = document.getElementById("MostRated") as HTMLAnchorElement;
    var c = document.getElementById("checkbox") as HTMLInputElement;

    if(c.checked){m.style.color = '#00ffe58e';}
    this.filteredFilms = [];
    this.sortedFilms = [];

    for(var item of this.service.list){
        if(item.rating_count >= 15){
            this.sortedFilms.push(item);
        }
    }

    this.filterFilms();
    this.displayFilteredFilms();
}

```

Slika 19. Funkcija „sortFilmByRatingCount“

Funkcija „sortFilmByRatingCount“ funkcioniра na identičan način kao i prethodna „sortFilmByBestRated“ funkcija, samo što se u ovom slučaju gleda „rating_count“ filma. Ako je „rating_count“ filma veći ili jednak 15, onda se taj film sprema u „sortedFilms“ i prikazuje se na web stranici.

Zadnji dropdown izbornik je „Year“ izbornik. U ovom izborniku kao opcije navedene su godine u kojima su filmovi izišli. Korisnik ovdje može sortirati odabirom na jednu od sljedećih opcija: „2023“, „2021 - 2022“, „2010 - 2020“, „2000 - 2010“, „1990 - 2000“ i „Older“. Opcija „Older“ izlistava sve filmove koji su izišli prije 1990 godine. Kada se odabere jedna od tih opcija, iz baze će se izlistati svi filmovi koji su izišli u odabranom periodu. Svaka od tih opcija ima svoju funkciju koja se poziva te je kao primjer dovoljno priložiti jednu funkciju, obzirom da sve rade na isti princip.

```
sortFilmByYear2000_2010(){
  var y2000_2010 = document.getElementById("2000 - 2010") as HTMLAnchorElement;
  var c = document.getElementById("checkbox") as HTMLInputElement;

  if(c.checked){y2000_2010.style.color = '#00ffe58e';}
  this.filteredFilms = [];
  this.sortedFilms = [];

  for(var item of this.service.list){
    if(item.release_date.substring(0,4) >= "2000" && item.release_date.substring(0,4) < "2010"){
      this.sortedFilms.push(item);
    }
  }

  this.filterFilms();

  this.displayFilteredFilms();
}
```

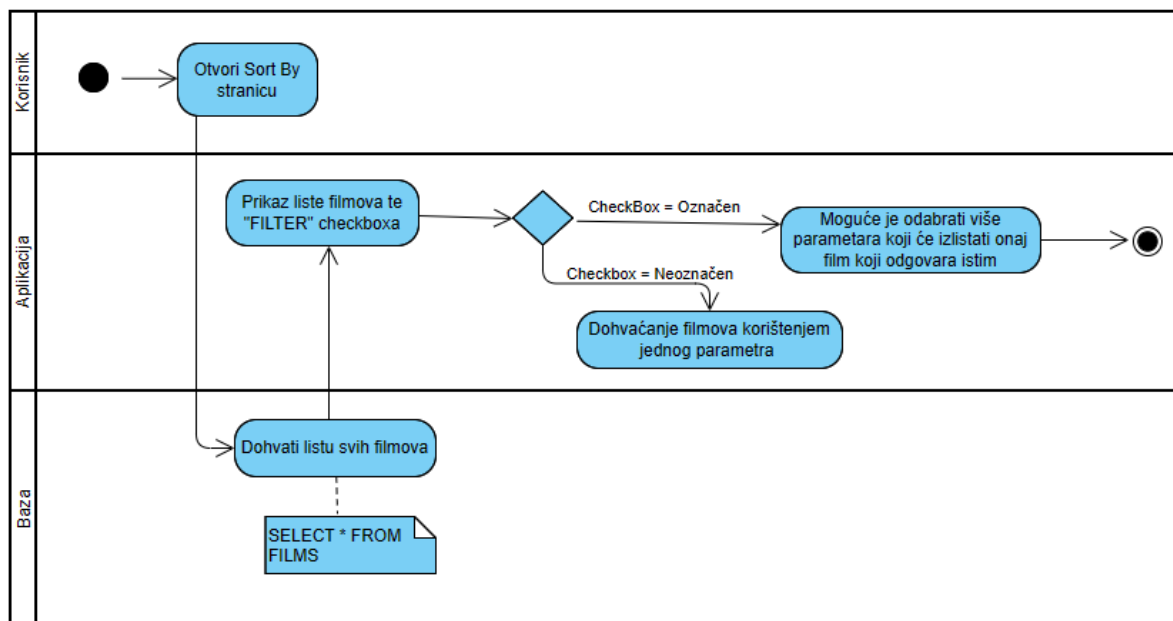
Slika 20. Funkcija „sortFilmByYear2000-2010“

U ovoj funkciji gleda se „release_date“ vrijedost svih filmova. U ovom slučaju, gdje je opcija „2000 - 2010“, potrebno je izlistati sve filmove koji su izašli u tome periodu, što se postiže pomoću if statementa u kojem je navedeno da se film spremi u „sortedFilms“ listu ako odgovara uvjetu da je njegov „release_date“ veći ili jednak „2000“ i manji od „2010“. Sve ostale funkcije osim „sortFilmByYear2023“ i „sortFilmByYearOlder“ rade na isti način. Kod „sortFilmByYear2023“ funkcije, bilo je dovoljno u if statement navesti da se dohvate

filmovi gdje je „release_date“ jednak 2023, a kod „sortFilmByYearOlder“ funkcije u if statement je navedeno da se dohvate filmovi gdje je „release_date“ manji od 1990.

To je to što se tiče dropdown izbornika, no to nije sve što se nalazi na „Sort By“ stranici. Ovdje također postoji i jedan checkbox. O tome više u nastavku.

5.9. Filtering



Slika 21. Activity Dijagram Filter Funkcije

Na „Sort By“ stranici korisnik također može vidjeti jedan checkbox uz kojega piše riječ „Filter“. Naime, kod prethodno spomenutih dropdown izbornika, korisnik bi pri odabiru neke od opcija svaki put osvježio listu s novim filmovima, što znači da korisnik cijelo vrijeme sortira koristeći se samo jednim parametrom. Kako bi se korisniku omogućilo sortiranje s mogućnosti više parametara, dodan je „Filter“ checkbox koji, kada se označi, poziva funkciju „filterFilms“ koja će pamtit i označiti sve odabrane parametre i izlistavati odgovarajuće filmove. Kada dođe do situacije gdje korisnik odabere kombinaciju parametara koja se ne podudara ni sa kojim filmom u listi, na stranici se ništa ne pokazuje i korisnik tada može iznova početi s odabirom željenih parametara. Ako korisnik želi ugasi mogućnost filtriranja, to jednostavno može učiniti tako što ponovno klikne na checkbox.

```

initializeFilteredFilmsList(){
    this.filteredFilmsHolder = [];
    this.displayedFilms = [];
    this.resetAllDropDowns();
}

filterFilms(){
    var c = document.getElementById("checkbox") as HTMLInputElement;

    if(c.checked){
        if(this.filteredFilmsHolder.length == 0){
            this.filteredFilmsHolder = this.sortedFilms;
            this.filteredFilms = this.sortedFilms;
        }
        else{
            for(var filteredFilm of this.filteredFilmsHolder){
                for(var sortedFilm of this.sortedFilms){
                    if(filteredFilm.id_film == sortedFilm.id_film){this.filteredFilms.push(sortedFilm)}
                }
            }
            this.filteredFilmsHolder = this.filteredFilms;
        }
    }
}

```

Slika 22. Funkcije „initializeFilteredFilms“ i „filterFilms“

Funkcije „initializeFilteredFilms“ i „filterFilms“ su ključne za ovu funkcionalnost. Unutar „initializeFilteredFilms“ funkcije inicijaliziraju se „filteredFilmsHolder“ i „displayedFilms“ liste, dok se u „filterFilms“ funkciji liste „filteredFilmsHolder“ i „filteredFilms“ pune filmovima iz „sortedFilms“. Odabrani filteri se spremaju u „displayedFilms“ listu koja se potom koristi u „displayFilteredFilms“ funkciji.

```

displayFilteredFilms(){
    var x = document.getElementById("checkbox") as HTMLInputElement;

    if(x.checked){
        this.displayedFilms = this.filteredFilms;
    }else{
        this.displayedFilms = this.sortedFilms;
    }
}

```

Slika 23. Funkcija „displayFilteredFilms“

Kako bi filtriranje radilo, funkcije „filterFilms“ i „displayFilteredFilms“ pozivaju se u svim funkcijama za sortiranje, kao što se može primjetiti na slikama iz prethodnih poglavlja.

6. Popis slika

Dijagrami prikazani u ovome radu izrađeni su pomoću visual paradigm online alata, a druge slike su snapshotovi codea uslikani pomoću LightShot alata.

Slika 1. ERA Model

Slika 2. Dijagram Slučajeva Korištenja

Slika 3. Activity Dijagram Search Funkcionalnosti

Slika 4. Funkcija „getSearchedFilms“

Slika 5. Activity Dijagram Rating Funkcionalnosti

Slika 6. Funkcija „checkIfVoted“

Slika 7. Funkcija „postNewRating“

Slika 8. Aktiviraj Dijagram Add Comment Funkcije

Slika 9. Funkcija „saveComment“

Slika 10. Funkcija „postComment“

Slika 11. Activity Dijagram Bookmark Funkcionalnosti

Slika 12. Activity Dijagram Recommended Films

Slika 13. Funkcija „getRecommendedMovies“

Slika 14. Activity Dijagram Login i Registracije

Slika 15. Activity Dijagram Unosa Filma

Slika 16. Activity Dijagram Sort By Funkcionalnosti

Slika 17. Funkcija „sortFilmByGenre“

Slika 18. Funkcija „sortFilmByBestRated“

Slika 19. Funkcija „sortFilmByRatingCount“

Slika 20. Funkcija „sortFilmByYear2000-2010“

Slika 21. Activity Dijagram Filter Funkcije

Slika 22. Funkcije „initializeFilteredFilms“ i „filterFilms“

Slika 23. Funkcija „displayFilteredFilms“