# 

# **MY SERIES**

# Sustav za praćenje serija

Grupa: Nova nada

Lovro Filipović

Janko Kovačević

Matija Maljić

Filip Štivičić

# Design story

MySeries je aplikacija kojoj je namjena olakšati korisniku praćenje televizijskih serija. Broj serija je velik te kako se svaka sastoji od više sezona, a svaka sezona od više epizoda praćenje istih predstavlja određeni problem. Aplikacija pruža korisniku odabir vlastite liste serija tako da se pretplaćuje na serije koje su mu zanimljive, koje je pogledao ili tek planira pogledati.

Korisnik se prilikom početka rada mora autorizirati vlastitim korisničkim imenom i lozinkom. Korisniku je ponuđena lista svih serija iz koje može dodavati iste na svoju osobnu listu, a tako može i maknuti seriju sa vlastite liste. Epizode svih serija dodanih na osobnu listu su prikazane u posebnom pregledu gdje su epizode sortirane po datumu izlaska i prikladno označene s obzirom na to jesu li pogledane. U tom pregledu korisnik može vidjeti koje epizode kada izlaze i vidljiv mu je prikaz što je pogledao, a što tek namjerava pogledati. Odabirom pojedine serije sa nekog od pregleda se mogu vidjeti podaci o svakoj pojedinoj seriji što uključuje naziv, kratak sadržaj, ocjenu serije, popis epizoda te popis glumaca. Za svakog glumca se također mogu vidjeti serije u kojima je glumio, ali i podaci o njemu samom. Odabirom epizode se mogu vidjeti podaci o epizodi, kratak sadržaj te se epizoda može označiti kao pogledana ili se ta oznaka može maknuti. Također za svaku epizodu korisnik može ostaviti komentar ili neki naputak o epizodi koji je vidljiv samo njemu, u komentaru može na primjer staviti kad planira pogledati epizodu ili dojmove o epizodu nakon šta ju pogleda.

Mane aplikacije su nepotpuna baza serija za što bi se bilo potrebno spajati na neku od stranica kao što je IMDB.

Aplikacija bi trebala raditi na web-u, na Windows OS-u te na mobilnoj platformi Android.

# Use case

Use case 1 – Prijava na sustav

Use case 2 – Pregled epizoda pretplaćenih serija

Use case 3 – Pregled svih serija

Use case 4 – Pregled serije (o seriji, glumci i epizode)

Use case 5 – Dodaj pretplatu na seriju

Use case 6 – Makni pretplatu na seriju

Use case 7 – Označi epizodu sa pogledan

Use case 8 – Odznači epizodu

Use case 9 – Dodaje komentar na epizodu

Use case 10 – Pregled svih glumaca

Use case 11 – Pregled glumca

Use case 12 – Odjava iz sustava

## Use case 1 – Prijava na sustav

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi koristiti funkcionalnosti aplikacije.

**Preconditions:** Računalo na kojem se pokreće aplikacija je ispravno.

**Postconditions:** Korisnik je uspješno prijavljen na sustav.

**Glavni uspješni scenariji:**

1. Korisnik pokreće aplikaciju.
2. Aplikacija je pokrenuta i traži od korisnika korisničko ime i lozinku.
3. Korisnik unosi korisničko ime i lozinku.
4. Aplikacija provjerava uneseno korisničko ime i lozinku.
5. Aplikacija prijavljuje korisnika u sustav i omogućava njene funkcionalnosti.

**Alternativni scenariji:**

4a. Pogrešno uneseno korisničko ime i/ili lozinka.

4a1. Aplikacija omogućuje korisniku da ponovno unese korisničko ime i lozinku.

## Use case 2 – Pregled epizoda pretplaćenih serija

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi lako doći informacije o epizodama serija na koje je pretplaćen.

**Preconditions:** Korisnik je uspješno prijavljen na sustav.

**Postconditions:** Aplikacija je ispisala popis epizoda serija na koje je korisnik pretplaćen.

**Glavni uspješni scenariji:**

1. Korisnik zatraži popis epizoda.
2. Aplikacija ispisuje popis epizoda.

**Alternativni scenariji:**

2a. Ne postoji niti jedan zapis epizode u bazi.

2a1. Aplikacija javlja korisniku da u bazi nema zapis o epizodama.

2b. Nije odabrana niti jedna serija za pretplatu.

2b1. Aplikacija javlja korisniku da nije odabrao niti jednu seriju za pretplatu.

## Use case 3 – Pregled svih serija

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi lako doći informacije o serijama za koje je omogućena pretplata.

**Preconditions:** Korisnik je uspješno prijavljen na sustav.

**Postconditions:** Aplikacija je ispisala popis serija u dva bloka, pretplaćene zatim nepretplaćene.

**Glavni uspješni scenariji:**

1. Korisnik zatraži popis serija.
2. Aplikacija ispisuje popis serija.

**Alternativni scenariji:**

2a. Ne postoji niti jedan zapis serije u bazi.

2a1. Aplikacija javlja korisniku da u bazi nema zapisa serija.

## Use case 4 – Pregled serije (o seriji, glumci i epizode)

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi imati pristup informacijama o pojedinoj seriji.

**Preconditions:** Korisnik je uspješno prijavljen na sustav. Mora postojati barem jedan zapis serije u bazi.

**Postconditions:** Aplikacija je ispisala informacije o seriji.

**Glavni uspješni scenariji:**

1. Korisnik zatraži popis serija.
2. Aplikacija ispisuje popis serija.
3. Korisnik odabire seriju.
4. Aplikacija ispisuje informacije o odabranoj seriji: o seriji, glumci, epizode.

**Alternativni scenariji:**

2a. Ne postoji niti jedan zapis serije u bazi.

2a1. Aplikacija javlja korisniku da u bazi nema zapisa serija.

## Use case 5 – Dodaj pretplatu na seriju

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi se pretplatiti za praćenje serije.

**Preconditions:** Korisnik je prijavljen na sustav.

**Postconditions:** Korisnik se uspješno pretplatio za praćenje serije.

**Glavni uspješni scenariji:**

1. Korisnik zatraži stranicu sa popisom serija.
2. Aplikacija ispisuje popis dostupnih serija za pretplatu.
3. Korisnik odabire seriju na koju se želi pretplatiti.
4. Aplikacija pohranjuje pretplatu na seriju korisnika.

**Alternativni scenariji:**

1a. Ne postoji niti jedna serija spremljena u bazi.

2a. Korisnik je već pretplaćen na odabranu seriju.

2a1. Aplikacija javlja korisniku da je već pretplaćen za seriju.

2a2. Aplikacija ispisuje popis dostupnih serija.

## Use case 6 – Makni pretplatu na seriju

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi prestati pratiti seriju.

**Preconditions:** Korisnik je prijavljen na sustav. Korisnik je pretplaćen na barem jednu seriju.

**Postconditions:** Korisnik je maknuo pretplatu na seriju.

**Glavni uspješni scenariji:**

1. Korisnik zatraži stranicu sa popisom serija.
2. Aplikacija ispisuje popis pretplaćenih serija korisnika.
3. Korisnik odabire seriju za koju želi maknuti pretplatu.
4. Aplikacije pohranjuje izmjene o pretplati serije.

**Alternativni scenariji:**

Nema.

## Use case 7 – Označi epizodu sa pogledano

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi moći pratiti koje epizode serije je pogledao.

**Preconditions:** Korisnik je prijavljen na sustav.

**Postconditions:** Epizoda je označena kao pregledana.

**Glavni uspješni scenariji:**

1. Korisnik zatraži stranicu sa pregledom epizoda.
2. Aplikacija ispisuje popis epizoda.
3. Korisnik odabire epizodu koju želi označiti sa pogledano.
4. Aplikacija pohranjuje izmjene i označuje epizodu kao pogledanu.

**Alternativni scenariji:**

1a. Korisnik zatraži stranicu sa podatcima o seriji.

1a1. Aplikacija ispisuje podatke o seriji i popis epizoda serije.

1a2. Povratak na 3. korak glavnog scenarija.

2a. Korisnik nema pretplaćenih serija.

2a1. Aplikacija javlja korisniku da nema pretplatu na niti jednu seriju.

3a. Korisnik odabire epizodu i zatraži informacije o epizodi.

3a1. Aplikacija prikazuje podatke o epizodi.

3a2. Korisnik označuje epizodu kao pogledanu.

## Use case 8 – Odznači epizodu

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi moći pratiti koje epizode serije je pogledao.

**Preconditions:** Korisnik je prijavljen na sustav. Korisnik ima pretplatu na barem jednu seriju. Korisnik ima barem jednu epizodu označenu sa pogledano.

**Postconditions:** Epizoda nije više označena kao pogledana.

**Glavni uspješni scenariji:**

1. Korisnik zatraži stranicu sa pregledom epizoda.
2. Aplikacija ispisuje popis epizoda.
3. Korisnik odabire epizodu koju želi odznačiti.
4. Aplikacija pohranjuje izmjene i uklanja oznaku epizode kao pogledanu.

**Alternativni scenariji:**

1a. Korisnik zatraži stranicu sa podatcima o seriji.

1a1. Aplikacija ispisuje podatke o seriji i popis epizoda serije.

1a2. Povratak na 3. korak glavnog scenarija.

3a. Korisnik odabire epizodu i zatraži informacije o epizodi.

3a1. Aplikacija prikazuje podatke o epizodi.

3a2. Korisnik odznači epizodu kao pogledanu.

## Use case 9 – Dodaje komentar na epizodu

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi moći prisjetiti se stvari iz epizode serije.

**Preconditions:** Korisnik je prijavljen na sustav. Korisnik ima pretplatu na barem jednu seriju. Korisnik ima barem jednu epizodu označenu sa pogledano.

**Postconditions:** Uspješno je dodan komentar na epizodu.

**Glavni uspješni scenariji:**

1. Korisnik zatraži stranicu sa pregledom epizoda.
2. Aplikacija ispisuje popis epizoda.
3. Korisnik odabire epizodu i zatraži informacije o epizodi.
4. Aplikacija otvara stranicu epizode sa podatcima o epizodi.
5. Korisnik upisuje komentar.
6. Aplikacija pohranjuje komentar.

**Alternativni scenariji:**

1a. Korisnik zatraži stranicu sa podatcima o seriji.

1a1. Aplikacija ispisuje podatke o seriji i popis epizoda serije.

1a2. Korisnik odabire epizodu i zatraži informacije o epizodi.

## Use case 10 – Pregled svih glumaca

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi lako doći informacije o glumcima koji se pojavljuju u serijama.

**Preconditions:** Korisnik je uspješno prijavljen na sustav.

**Postconditions:** Aplikacija je ispisala popis glumaca.

**Glavni uspješni scenariji:**

1. Korisnik zatraži popis glumaca.
2. Aplikacija ispisuje popis glumaca.

**Alternativni scenariji:**

2a. Ne postoji niti jedan zapis glumca u bazi.

2a1. Aplikacija javlja korisniku da u bazi nema zapis o glumcima.

## Use case 11 – Pregled glumca

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – želi imati pristup informacijama o pojedinom glumcu.

**Preconditions:** Korisnik je uspješno prijavljen na sustav. Mora postojati barem jedan zapis glumaca u bazi.

**Postconditions:** Aplikacija je ispisala informacije o glumcu.

**Glavni uspješni scenariji:**

1. Korisnik zatraži popis glumaca.
2. Aplikacija ispisuje popis glumaca.
3. Korisnik odabire glumca.
4. Aplikacija ispisuje informacije o odabranom glumcu.

**Alternativni scenariji:**

1a. Korisnik pregledava informacije o seriji.

1a1. U informacijama odabire glumca.

1a2. Aplikacija ispisuje informacije o glumcu.

## Use case 12 – Odjava iz sustava

**Primary actor:** Korisnik

**Stakeholders and interests:** Korisnik – ne želi više koristit aplikaciju.

**Preconditions:** Korisnik je uspješno prijavljen na sustav.

**Postconditions:** Korisnik je odjavljen sa sustava.

**Glavni uspješni scenariji:**

1. Korisnik zatraži odjavu sa sustava.
2. Aplikacija odjavljuje korisnika iz sustava.

**Alternativni scenariji:** Nema

# Opis objektnog modela



Class diagram modela domene

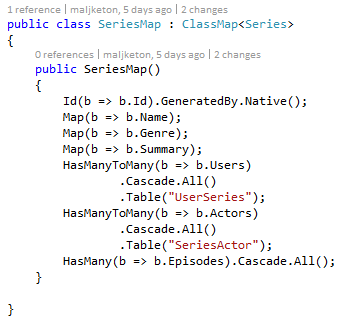




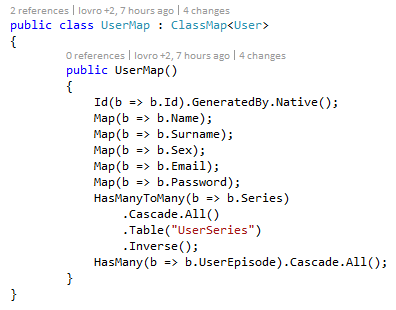
Class diagram repozitorija i njihovih sučelja

# Opis implementacije perzistencije

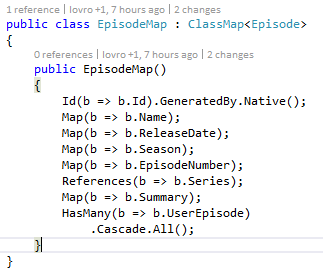
Perzistencija podataka u sustavu napravljena je O/R mapiranjem modela iz domena koristeći fluent Nhibernate framework. Za pohranu podataka koristi se MySQL open source sustav za upravljanjem bazom podataka. Kako bi NHibernate mogao izvršiti mapiranje objektnih modela u SQL naredbe, potrebno mu je dati podatke o mapiranju. Podatci o mapiranju govore Nhibernateu gdje se nalaze podatci za klasu i opisuju odnose među klasama.



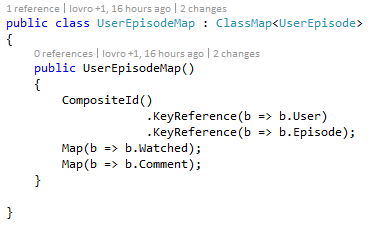
Mapiranje klase Series



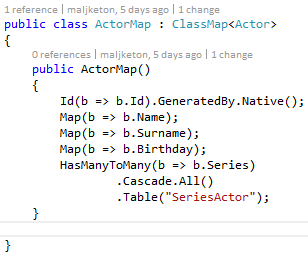
Mapiranje klase User



Mapiranje klase Episode



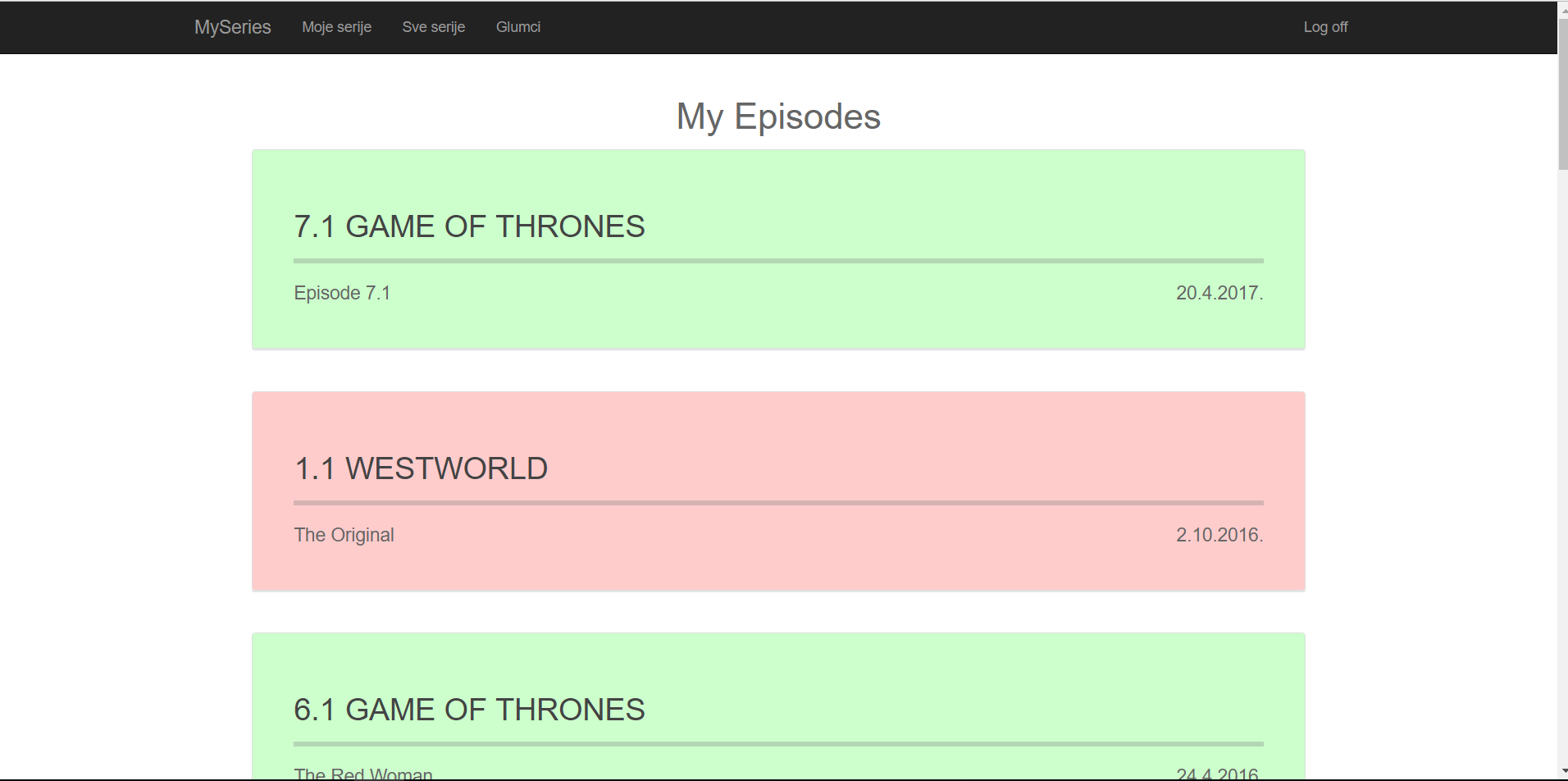
Mapiranje klase UserEpisode



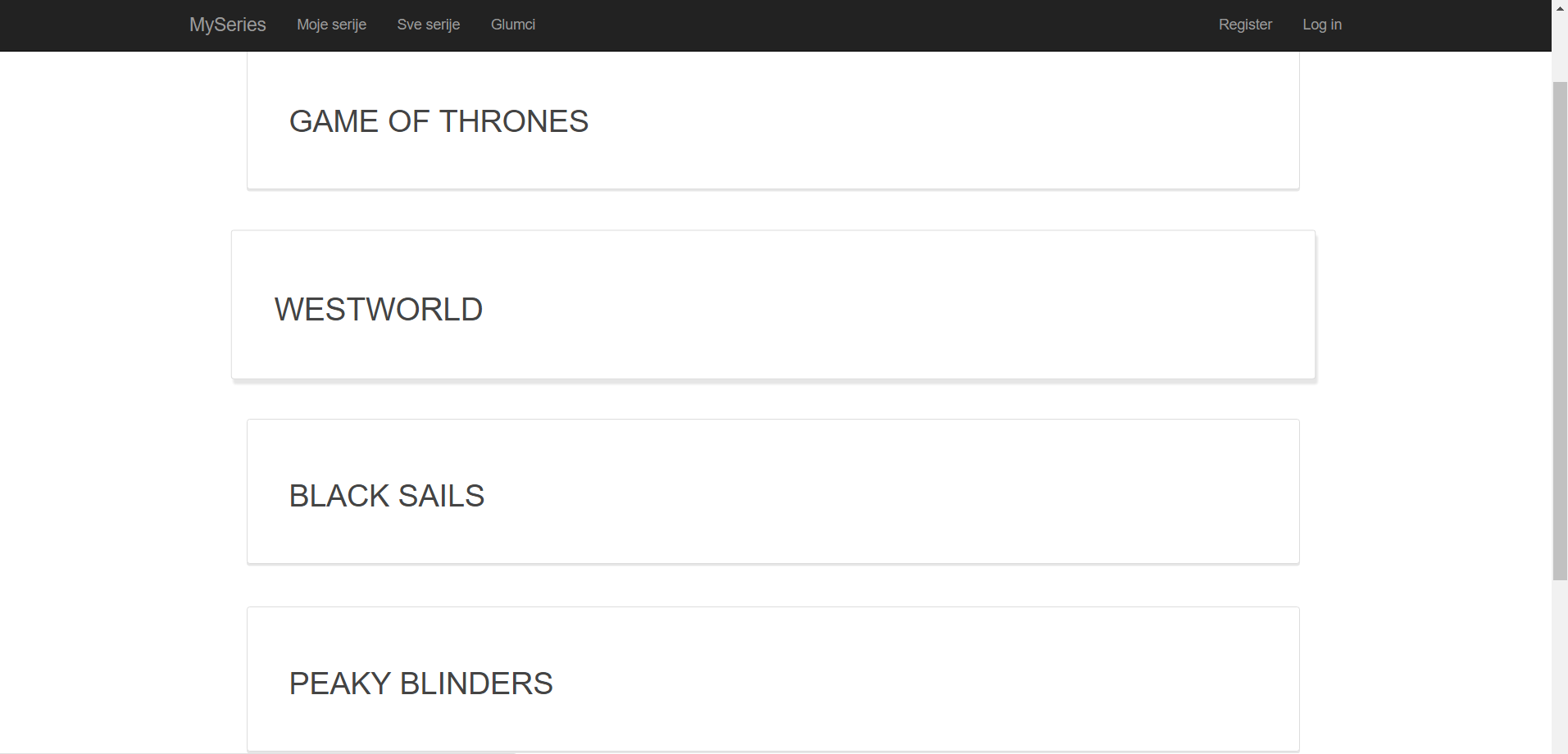
Mapiranje klase Actor

Transakcije pokreće kontroler unutar pojedine sesije. Kontroler priprema podatke i poziva odgovarajuće metode repozitorija te potvrđuje transakcije. Transakcije se pokreću samo prilikom dodavanja i izmjena podataka iz baze podataka.

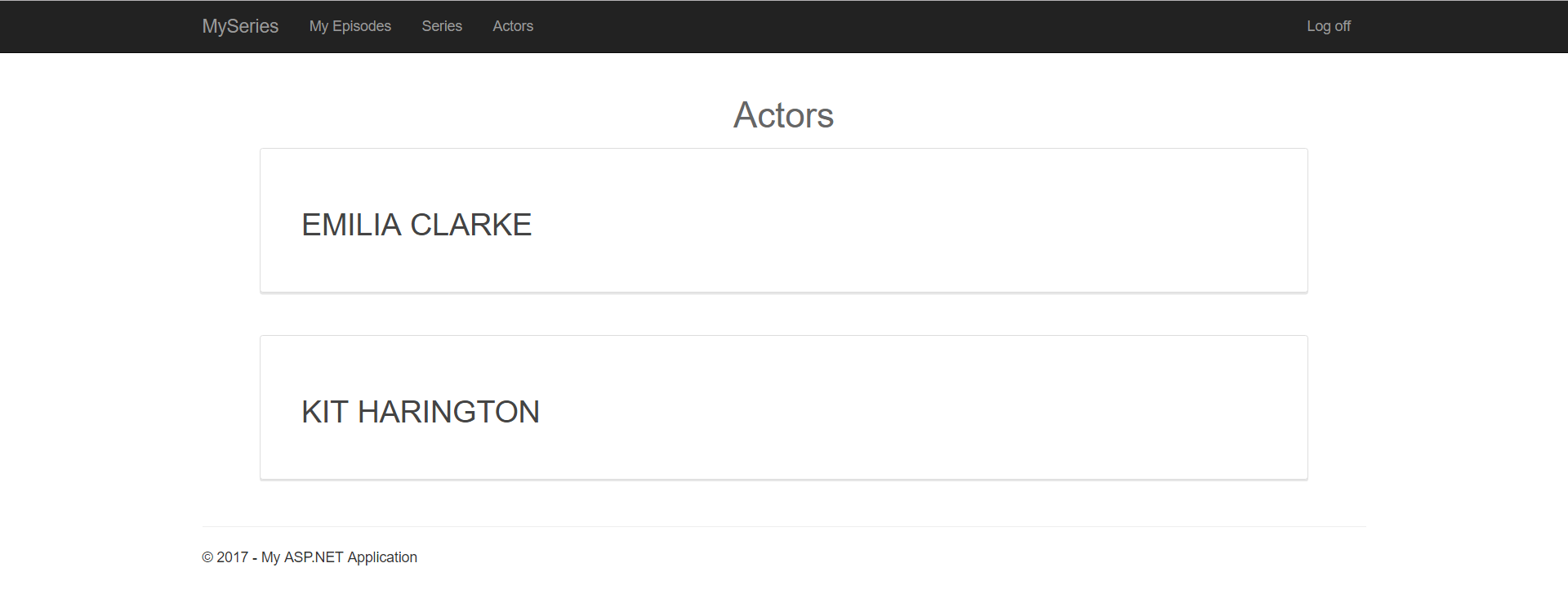
# Opis izgrađene web-aplikacije



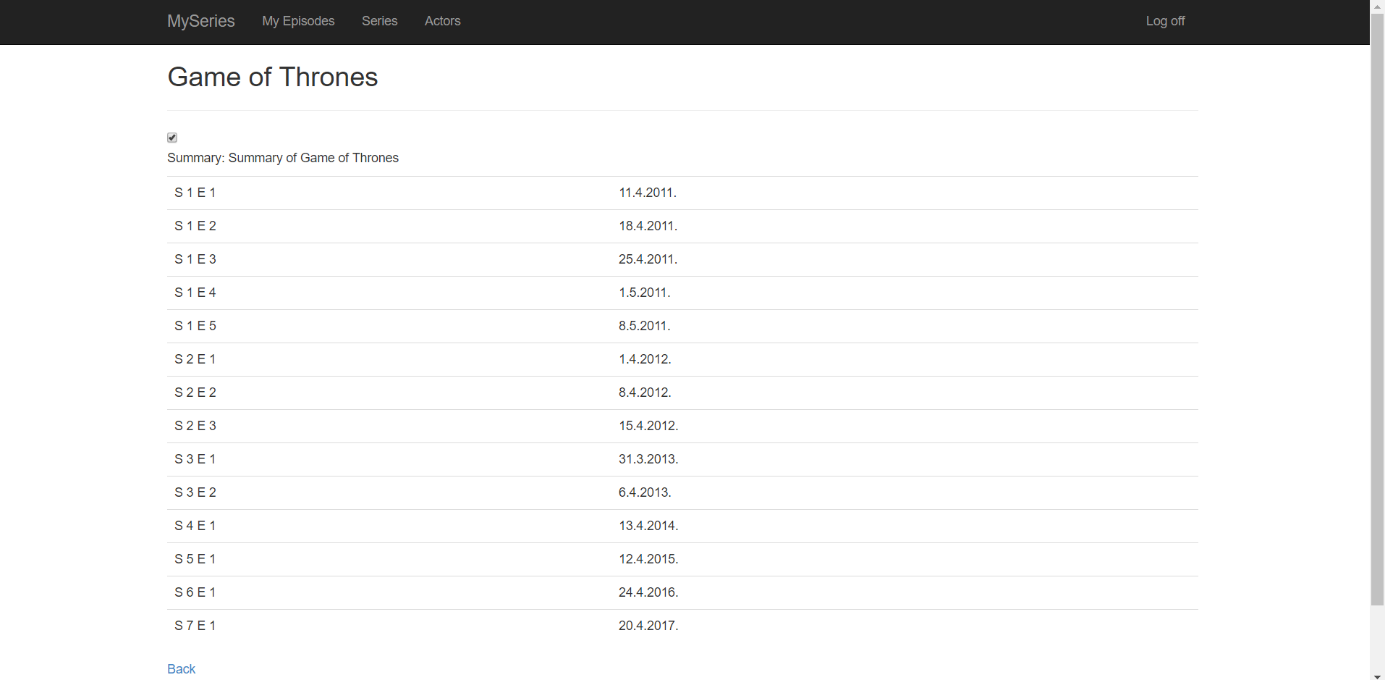
1. Prikaz stranice na kojoj je lista epizoda pretplaćenih serija sa datumom izlaska i oznakom je li serija pogledana.



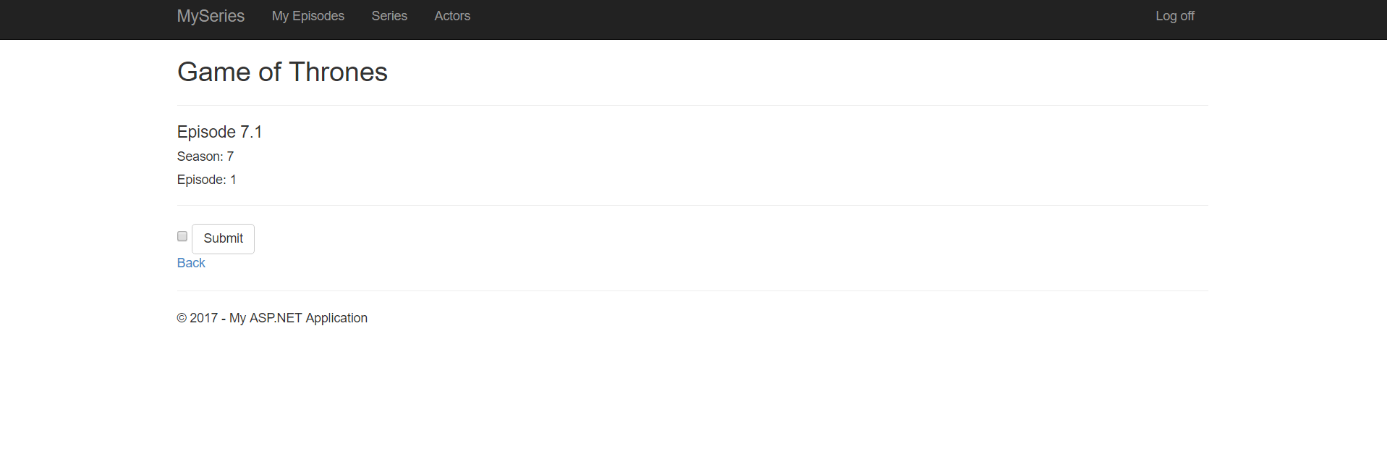
1. Prikaz stranice svih serija, a zatim i sve ostale sa mogućnošću odabira serije.



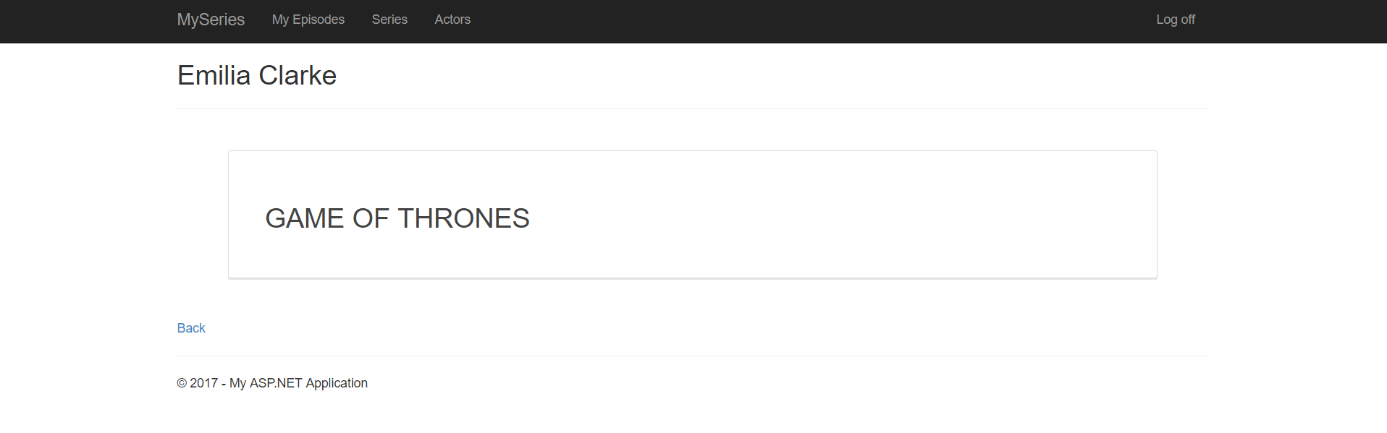
1. Lista svih glumaca sa mogućnošću odabira glumca.



1. Prikaz stranice sa podacima o pojedinoj seriji te popisom epizoda koje pripadaju seriji.

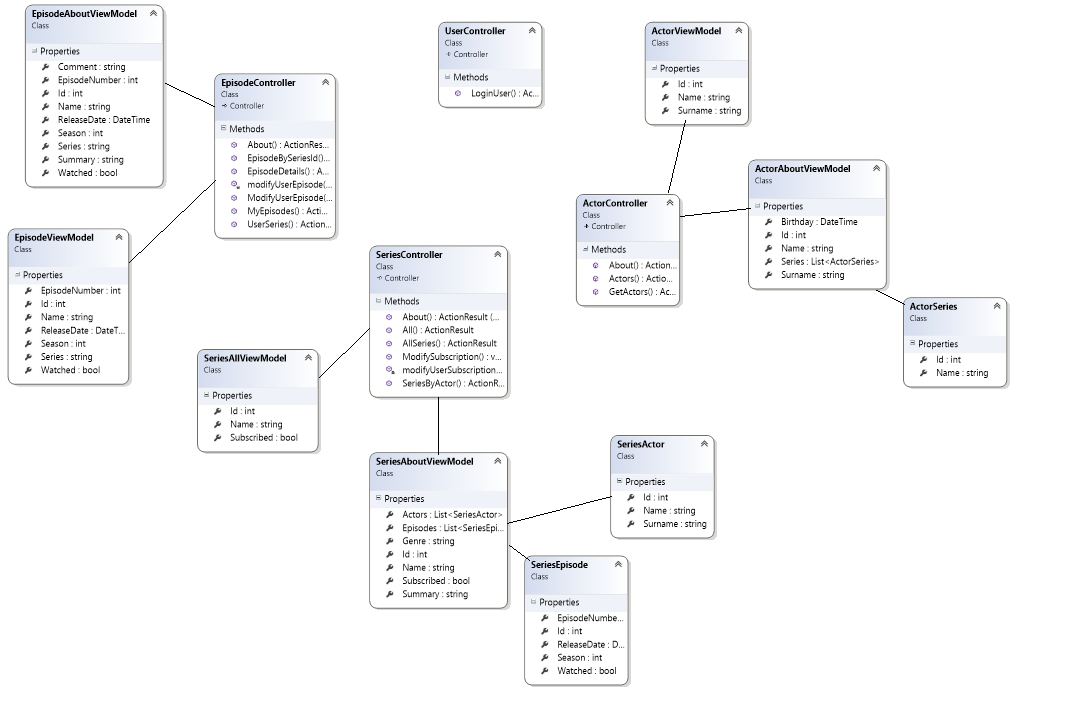


1. Prikaz stranice sa podacima o pojedinoj epizodi sa informacijom o pogledanosti i komentarom na pogledanu epizodu.



1. Prikaz stranice sa podacima o glumcu sa listom serija u kojima je glumio.

Web aplikacija je pisana u .NET MVC-u te posjeduje standardne komponente model-view-contoller aplikacija. Kontroleri primaju i obrađuju zahtjeve od korisnika (pozivi iz Viewa odgovarajućih Action metoda), dohvaćaju liste serija, epizoda, glumaca pozivanjem odgovarajućih metode iz repozitorija te pripremaju View modele za odgovarajuće stranice. View modeli su odvojeni od modela domene i sadrže podatke koji su potrebni za taj pogled. U kontrolerima se obavlja pokretanje transakcija te spremanje ili izmjena podataka o pretplati na serije, pogledanosti epizode i komentaru na epizodu. Kontroleri su SeriesController, EpisodeController i ActorController. Sva tri kontrolera imaju po dva pogleda, za prikaz liste podataka (epizode, serije) te svakog podatka posebno. Tu je i Account kontroler koji je promijenjen kako bi bila omogućena prijava sa podacima iz baze podataka. Na pogledu sa epizodama serija zeleno su označene one koje još nisu pogledane te je svaku moguće odabrati i označiti ju kao pogledanu te nakon toga i dodati komentar. Kontroleri obrade podatke na način da pripreme nove objekte i pozivaju metode repozitorija koji spremaju objekte u bazu. U nastavku su povezane klase web aplikacije gdje kontroleri imaju reference na odgovarajuće view modele. Da bi graf bio potpun fale mu repozitoriji te view-ovi koji također imaju referencu na view modele koje im šalje kontroler prilikom prikazivanja view-a.

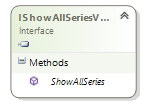


1 Class diagram web- aplikacije

# Opis izrađene Desktop aplikacije

Za izradu dektop aplikacije korišten je MVP obrazac. Aplikacija koristi programske knjižice MySeries.Model i MySeries.DAL. Unutar MySeries.DAL se nalaze repozitorij klase iz kojih se dohvaćaju i pohranjuju iz baze podataka podatci potrebni za rad aplikacije. Dekstop aplikacija je ostvarena kroz tri projekta: MySeries.BaseLib, MySeries.Controllers i MySeries.Desktop.

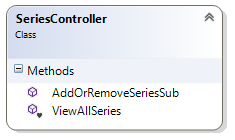
MySeries.BaseLib projekt sadrži skup sučelja (interface) kojim su opisane metode koje prihvaćaju i prikazuju podatke u prezentacijskom sloju. Prezentacijski sloj predstavlja projekt MySeries.Desktop. Postoji sučelje za svaku od formi koje sadrži metode za ispis i obradu podataka, kao i sučelje za factory Windows Formi.



1 Primjer interface unutar MySeries.BaseLib projekta

Na slici 1. možemo vidjeti primjer interface objekta koji služi za pristupanje metodi koja generira u prezentacijskom sloju formu koja prikazuje u ovom slučaju sve Serije koje su korisniku na raspologanju.

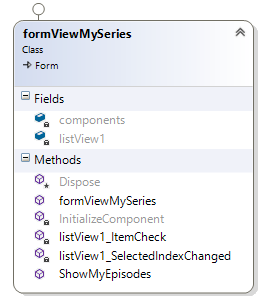
MySeries.Controller projekt sadrži skup controller klasa koje su zadužene za prosljeđivanje podataka između MySeries.DAL projekta i prezentacijskog projekta MySeries.Desktop. Uz controller klase tu je i statična klasa Common zadužena za pohranu podataka o korisniku koji trenutno koristi aplikaciju.



2 Primjer controller-a unutar MySeries.Controller projekta

Na slici dva možemo vidjeti primjer controller objekta koji ima dvije metode implementirane. Metoda AddOrRemoveSeriesSub omogućuje uklanjanje ili dodavanje pretplate na seriju koja se poziva ukoliko korisnik svojim unosom u formu promjeni stanje pretplate. Ova metoda poziva Repository metode MySeries.DAL sloja i prosljeđuje im novo stanje pretplate koje se zatim sprema u bazu podataka. Metoda ViewAllSeries omogućava dohvat podataka o svim serijama koje se dohvaćaju na zahtjev korisnika za otvaranje forme Sve serije. Podatci se zatim proslijede formi Sve serije koja ih poveže s objektima za prikaz.

MySeries.Desktop projekt sadrži prezentacijske elemente odnosno WindowsForm-e, klasu za pokretanje aplikacije i factory za WindowsForm-e. Forme sadrže metode za prikazivanje definirane ranije navedenim sučeljima. Metode za prikazivanje povezuju podatke koje dohvaća MySeries.Controller projekt na same prezentacijske objekte poput ListView-a i TextBox-ova. Također forme sadrže metode za obradu ulaza od strane korisnika aplikacije. Ulazni podatci i promjene koje korisnik napravi prenose se MySeries.Controller projektu koji ih obrađuje i radi željene izmjene. Factory unutar ovog projekta je dodan zbog dodatne strukturiranosti programskog koda odnosno preglednosti.

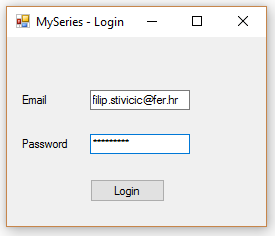


3 Primjer forme unutar MySeries.Desktop projekta

Na slici 3. možemo vidjeti primjer form objekta u prezentacijskom sloju aplikacije. Objek sadrži listView na kojem se prikazuju podatci o serijama koje formi prosljedi controller objekt iz prethodnog odlomka. Metode unutar forme osim povezivanja podataka rade i obradu unosa korisnika odnosno događaja. Novim događajem, odnosno korisnikovim unosom, pozivaju se controller objekti i prenose im se parametri koji ti objekti dalje obrađuju.

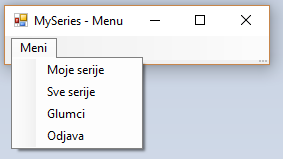
## Primjeri formi Desktop projekt

U nastavku promatramo koje objekte sadrži klijentsko sučelje, koja je svrha nekih od objekata te što se izvršava tokom interakcije klijenta s desktop aplikacijom.



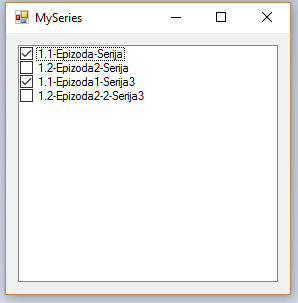
1. Login

Unutar ove forme postoje je polja za unos i gumb za potvrdu unosa. Korisnik mora unijeti valjanje podatke, u protivnom dobiva obavijest da to nije slučaj. Uspješnom prijavom korisnički podatci se statično pohranjuju do kraja korištenja aplikacije, a zatim se prikazuje sljedeća forma Menu.



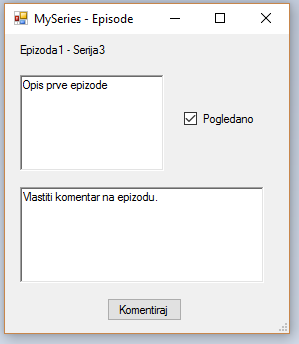
1. Meni

Ova forma sadrži navigacijsku traku koja sadrži poveznice na forme Moje Serije, Sve serije, Glumci te poziv metode za kraj korištenja aplikacije Odjava. Odabirom neke od prve tri stavke otvara se novi prozor s odgovarajućim sadržajem dok nam Menu forma osta je i dalje dostupna dok ne odaberemo Odjava.



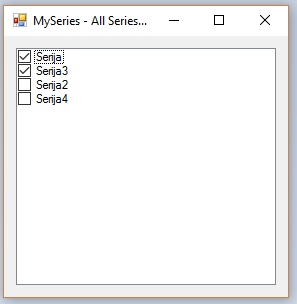
1. MySeries

Prikazana je lista epizoda serije na koje je korisnik pretplaćen. Omogućeno je označiti epizodu kao pogledanu odabirom odgovarajućeg check box elementa liste epizoda. Ukoliko želimo pogledati detalje o episodi dovoljno je odabrati željenu epizodu kako bi otvorili pregled Episode.



1. Episode

U formi Episode prikazani su detalji epizode. Prikazuju se ime epizode zajedno sa imenom serije kojoj epizoda pripada, opis epizode, komentar korisnika na epizodu, check box za označavanje da li je korisnik pogledao epizodu i gumb za unos izmjena komentara. Ukoliko korisnik označi ili odznači navedeni check box forma poziva metodu controllera zaduženog za odnos epizode i korisnika te prosljeđuje parametre novog stanja. Isto se odvija pritiskom gumba komentiraj, gdje se prenosi novo stanje komentara korisnika na otvorenu epizodu.



1. All Series

All Series prikaz sadrži jednostavnu listu svih serija koje postoje u bazi podataka aplikacije. Prvo su navedene serije na koje je korisnik već pretplaćen, odnosno koje korisnik prati. Odabirom odgovarajućeg check box-a u listi omogućeno je dodavanje odnosno uklanjanje pretplate korisnika. Odabirom same serije u listi otvara se novi pregled s detaljima o odabranoj seriji.

Opis izgrađene mobilne aplikacije

Mobilna aplikacija izvedena je na Android platformi. Programski jezik korišten za kreiranje iste je programski jezik Java. Integracija mobilne aplikacije s MySeries sustavom je pomoću RESTful web servisa. Protokol kojim se komunikacija izvodi jest HTTP protokol. Slika 1. prikazuje shemu komunikacije mobilne aplikacije s MySeries sustavom.

C:\Users\Janko\Downloads\myseries_json_restful.png

*Slika 1 Shema komunikacije između mobilne aplikacije i MySeries sustava.*

RESTful web servisi nalaze se u *Controller* dijelu MySeries-Web sustava. Pristupa im se preko *Uniform Resource Identifiera* (URI). Format koji se koristi za prijenos podataka između sustava i mobilne aplikacije je standardiziranim tekstualni format JSON.

U nastavku su opisane sve RESTful metode i njihovi parametri.

Sintaksa RESTful metode je: Naziv kontrolera / Naziv metode.

**Metoda User/LoginUser**

Metoda koja se koristi prilikom korisničke prijave u sustav. Nakon što korisnik unese email i lozinku na mobilnoj aplikaciji, podatci se prenose u sustav. Ako je korisnik pronađen u bazi podataka tada se sustav vraća odgovor koji sadrži model User.

Ako se korisnik ne može pronaći u bazi ili lozinka nije ispravna odgovor sadrži null vrijednost.

Tip metode: POST

Ulazni parametri:

email: adresa elektroničke pošte korisnika

password: lozinka korisnika

Povratni podatci:

User: JSON reprezentacija User modela ako su adresa elektronička pošte i lozinka ispravne, inače null.

**Metoda Episode/UserSeries**

Metoda koja se koristi kada korisnik zahtjeva pregled vlastitih serija na koje je pretplaćen. Na temelju korisničkog identifikatora metoda vraća sve serije na koje je korisnik pretplaćen. Za svaku seriju se dohvaća lista epizoda sa oznakom da li je korisnik pogledao epizodu.

Tip metode: GET

Ulazni parametri:

userId: identifikator korisnika

Povratni podatci:

List<Episode>: JSON reprezentacija liste Episode modela koje pripadaju serijama na koje je korisnik pretplaćen. Modeli episoda sadrže oznaku da li je epizoda pogledana.

**Metoda Series/AllSeries**

Metoda koja se koristi kada korisnik zahtjeva pregled svih serija. Za svaku seriju se oznaka da li je korisnik pretplaćen na seriju. Prilikom prikaza serija iste se rasčlanjuju na temelju korisničke pretplate.

Tip metode: GET

Ulazni parametri:

-

Povratni podatci:

List<Series>: JSON reprezentacija liste Series modela koje pripadaju serijama na koje je korisnik pretplaćen.

**Metoda Episode/EpisodeBySeriesId**

Metoda koja se koristi kada korisnik zahtjeva pregled detalja serije. Dohvaća se lista svih epizoda serije zajedno s oznakama da li je epizoda pogledana. Nadalje, dohvaćaju se podataci o seriji – naziv serije, opis serije te da li je korisnik pretplaćen na seriju.

Tip metode: GET

Ulazni parametri:

userId: identifikator korisnika

seriesId: identifikator serije

Povratni podatci:

List<Episode>: JSON reprezentacija liste Episode modela. Unutar liste Episode modela sadržane su informacije o seriji – naziv i opis te da li je korisnik pretplaćen na istu.

**Metoda Episode/EpisodeDetailsId**

Metoda koja se koristi kada korisnik zahtjeva pregled detalja epizode. Dohvaća se model epizode, zajedno sa nazivom, korisničkim komentarom te oznakom da li je korisnik pogledao seriju.

Tip metode: GET

Ulazni parametri:

userId: identifikator korisnika

episodeId: identifikator epizode

Povratni podatci:

Episode: JSON reprezentacija Episode modela. Model episode sadrži naziv epizode, opis iste, korisnički komentar na epizodu te oznaku da li je korisnik pogledao seriju.

**Metoda Actor/GetActors**

Metoda koja se koristi kada korisnik zahtjeva pregled svih glumaca. Dohvaća se lista modela glumaca. Za svakog glumca dohvaća se ime, prezime te datum rođenja.

Tip metode: GET

Ulazni parametri:

-

Povratni podatci:

List<Actor>: JSON reprezentacija liste Actor modela. Model actora sadrži ime I prezime glumca te datum rođenja.

**Metoda Series/SeriesByActor**

Metoda koja dohvaća sve serije u kojima je glumac glumio. Ova metoda se koristi prilikom prikaza detalja glumca. Model serije sadrži referencu na glumca iz koje se dohvaćaju podatci o glumcu.

Tip metode: GET

Ulazni parametri:

actorId: identifikator glumca

Povratni podatci:

List<Series>: JSON reprezentacija liste Series modela. Serije unutar modela predstavljaju sve serije u kojima je glumac glumio. Nadalje, model serije sadrži informacije o glumcu koji se koriste za prikaz detalja o glumcu.

**Metoda Series/ModifySubscription**

Metoda koja prijavljuje ili odjavljuje korisničku pretplatu sa serije.

Tip metode: POST

Ulazni parametri:

subscribed: da li se korisnik želi pretplatiti ili odjaviti pretplatu sa serije.

userId: identifikator korisnika

seriesId: identifikator serije

Povratni podatci:

-

**Metoda Episode/ModifyUserEpisode**

Metoda koja se pohranjuje u bazu podataka oznaku da li je korisnik pogledao epizodu te komentar korisnika na epizodu.

Tip metode: POST

Ulazni parametri:

watched: da li je korisnik pogledao epizodu.

comment: komentar korisnika na epizodu. Ako je vrijednost ovog parametra null onda se komentar ne ažurira.

userId: identifikator korisnika

episodeId: identifikator epizode.

Povratni podatci: