SVEUČILIŠTE U SPLITU

**PRIRODOSLOVNO MATEMATIČKI FAKULTET**

Dokumentacija projekta

**Aplikacija Ivanov Movie Data Base za ocjenjivanje filmova koristeći MongoDB bazu podataka**

Ivan Beran

Split, lipanj 2021.

**SADRŽAJ**

[1. Uvod 1](#_Toc75299250)

[2. MongoDB pojmovi 1](#_Toc75299251)

[2.1. Baza podataka 3](#_Toc75299252)

[3. Izgled aplikacije 3](#_Toc75299253)

[3.1. Početni zaslon 3](#_Toc75299254)

[3.2. Forma za dodavanje filmova 5](#_Toc75299255)

[3.3. Forma za ažuriranje filmova 7](#_Toc75299256)

[3.4. Forma za dodavanje redatelja 8](#_Toc75299257)

[3.5. Glavna forma za ispis podataka, pretragu i brisanje 8](#_Toc75299258)

[3.6. Forma za unos osvrta na film 12](#_Toc75299259)

[4. MongoDB 14](#_Toc75299260)

[4.1. Konekcija s bazom podataka 14](#_Toc75299261)

[4.2. Stvaranje kolekcije 15](#_Toc75299262)

[4.3. Pretraživanje kolekcije Artikl 16](#_Toc75299263)

[4.4. Ažuriranje podataka kolekcije Artikl 17](#_Toc75299264)

[4.5. Brisanje podataka kolekcije Artikl 17](#_Toc75299265)

[5. Zaključak 18](#_Toc75299266)

[6. Popis slika 19](#_Toc75299267)

# Uvod

U izradi ovog projekta korišten je programski jezik C# i baza podataka MongoDB. Kod je pisan u razvojnom okruženju Visual Studio. Osim toga bilo je potrebno instalirati MongoDb Community kako bi se mogla koristiti MongoDB baza podataka.

MongoDB baza podataka se pokreće u Windows Command pomoću naredbe *mongo.* Zbog jednostavnosti u ovome projektu je korišten MongoDBCompass sučelje koje olakšava pregled kreiranih baza podataka te upravljanje podatcima koji se nalaze u njoj.

Tema ovog projekta je bila izrada aplikacije koja omogućava ocjenjivanje filmova te pisanja osvrta za navedene filmove.

Funkcionalnost koju aplikacija treba imati su: dodavanje novih filmova u bazu podataka, dodavanje redatelja u bazu podataka, postavljanje ocjena i osvrta koji se također spremaju u bazu podataka, ažuriranje informacija o filmovima te brisanje pojedinih filmova, ispis svih filmova, ispis svih osvrta te prosjek svih ocjena po filmu i pretraživanje po nekim argumentima.

# MongoDB pojmovi

MongoDB je višeplatformska (radi na različitim OS) dokument orijentirana baza podataka koja pruža, visoke performanse, visoku dostupnost i laku skalabilnost. MongoDB radi na principu zbirke (kolekcije) i dokumenta.

Prednosti MongoDB-a nad RDBMS-om:

* Shema agnostic - MongoDB je baza dokumenata u kojoj jedna zbirka sadrži različite dokumente. Broj polja, sadržaj i veličina dokumenta iste zbirke mogu se razlikovati od jednog dokumenta do drugog.
* Struktura jednog objekta je jasna.
* Nema složenih pridruživanja.
* Mogućnost dubokog upita. MongoDB podržava dinamičke upite o dokumentima pomoću jezika upita utemeljenog na dokumentima koji je gotovo jednako moćan kao i SQL.
* Tuning.
* Jednostavnost skaliranja - MongoDB je lako mjeriti.
* Pretvorba / mapiranje aplikacijskih objekata u objekte baze podataka koji nisu potrebni.
* Koristi unutarnju memoriju za spremanje (prozora) radnog skupa, omogućujući brži pristup podacima.

Zašto koristiti MongoDB?

* Skladištenje usmjereno na dokument - podaci se pohranjuju u obliku JSON dokumenata.
* Indeksiranje po bilo kojem atributu
* Replikacija i visoka dostupnost
* Auto-sharding
* Bogati upiti
* Brza ažuriranja na mjestu
* Profesionalna podrška tvrtke MongoDB

Gdje koristiti MongoDB?

* Big data
* Upravljanje sadržajem i isporukom
* Mobilna i društvena infrastruktura
* Upravljanje korisničkim podacima
* Data Hub

Tipovi podataka

MongoDB podržava mnoge tipove podataka, a neki od njih su:

* **String** − This is the most commonly used datatype to store the data. String in MongoDB must be UTF-8 valid.
* **Integer** − This type is used to store a numerical value. Integer can be 32 bit or 64 bit depending upon your server.
* **Boolean** − This type is used to store a boolean (true/ false) value.
* **Double** − This type is used to store floating point values.
* **Min/ Max keys** − This type is used to compare a value against the lowest and highest BSON elements.
* **Arrays** − This type is used to store arrays or list or multiple values into one key.
* **Timestamp** − ctimestamp. This can be handy for recording when a document has been modified or added.
* **Object** − This datatype is used for embedded documents.
* **Null** − This type is used to store a Null value.
* **Symbol** − This datatype is used identically to a string; however, it's generally reserved for languages that use a specific symbol type.
* **Date**− This datatype is used to store the current date or time in UNIX time format. You can specify your own date time by creating object of Date and passing day, month, year into it.
* **Object ID** − This datatype is used to store the document’s ID.
* **Binary data** − This datatype is used to store binary data.
* **Code** − This datatype is used to store JavaScript code into the document.
* **Regular expression** − This datatype is used to store regular expression.

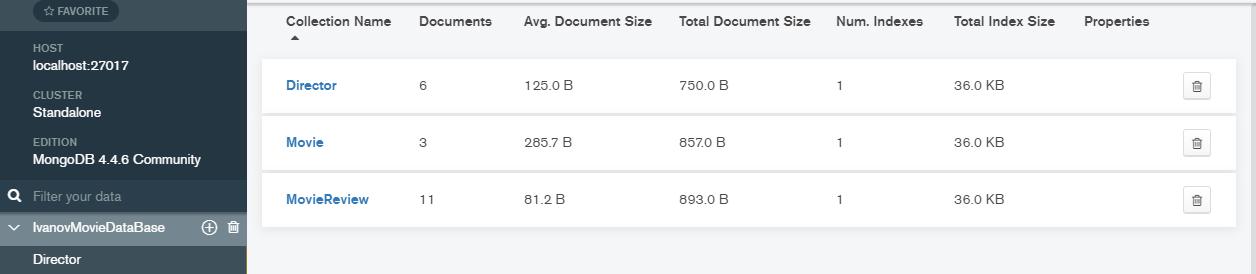
## Baza podataka

Baza podataka je fizički spremnik za zbirke. Svaka baza podataka dobiva vlastiti skup datoteka na datotečnom sustavu. Jedan MongoDB poslužitelj obično ima više baza podataka.

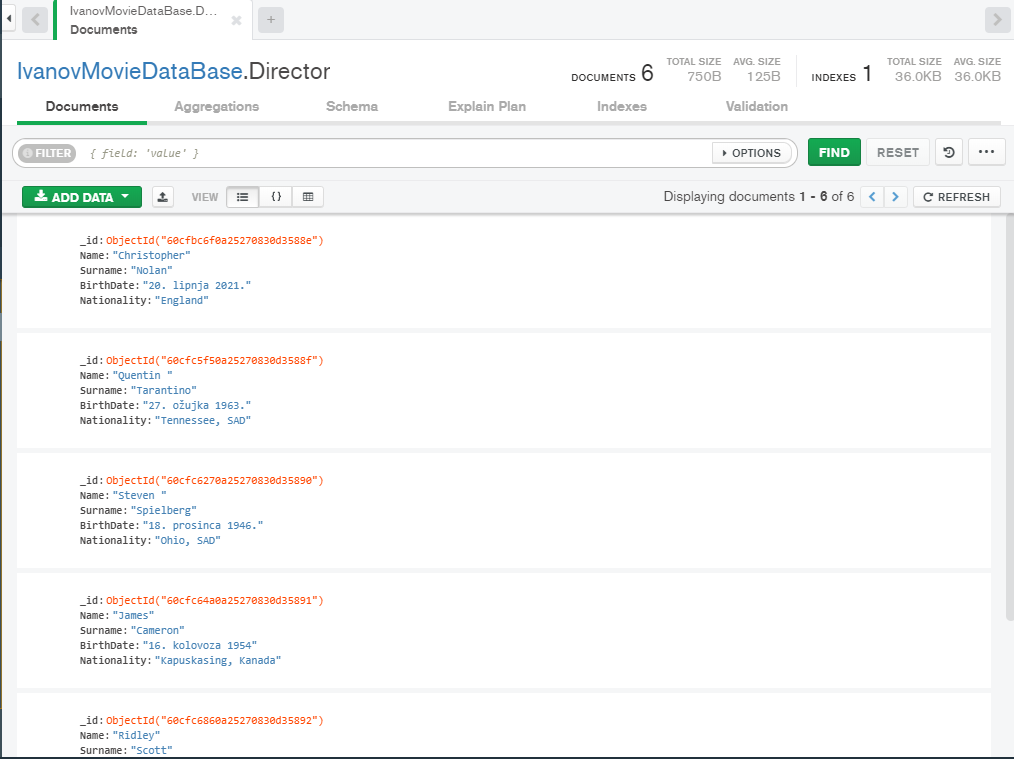
Svaka relacijska baza podataka ima tipičan dizajn sheme koji pokazuje broj tablica i odnos između tih tablica. MongoDB je nerelacijska baza podataka i stoga u njoj ne postoji koncept odnosa.

Baza podataka korištena u ovoj aplikaciji se sastoji od tri kolekcije: *Director*, *Movie* i *MovieReview*. Unutar tih kolekcija spremamo shodne dokumente.

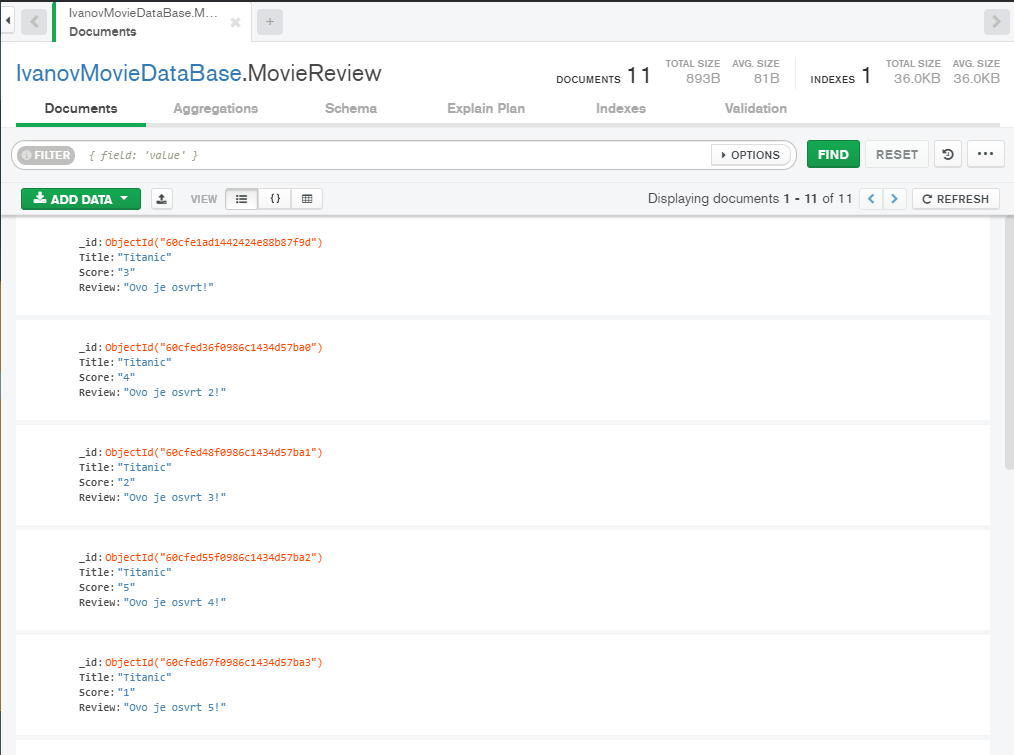
Kolekcija *Director* služi za spremanje svih redatelja koje možemo dodavati unutar aplikacije te ih pridjeljivati filmovima. Kolekcija *Movie* služi za spremanje filmova, a kolekcija *MovieReview* za spremanje osvrta na pojedine filmove to ocjene tih filmova.



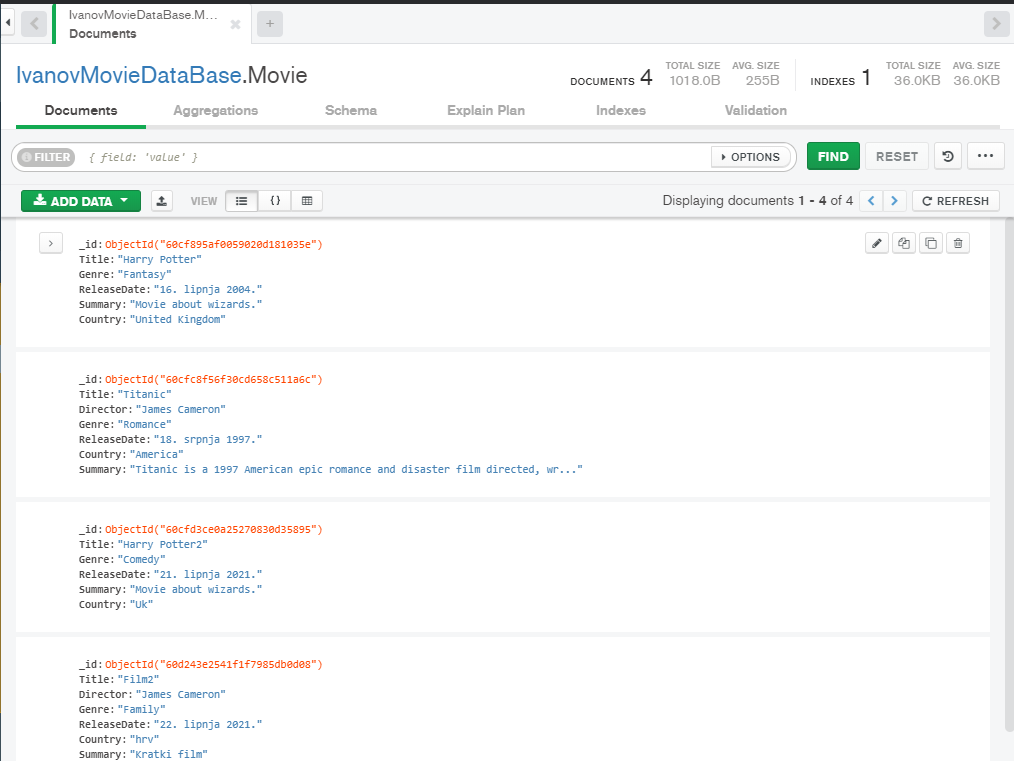
Slika 1. MongoDb Compass Community sučelje i prikaz storenih kolekcija



Slika 2: Kolekcija Director



Slika 3: Kolekcija MovieReview

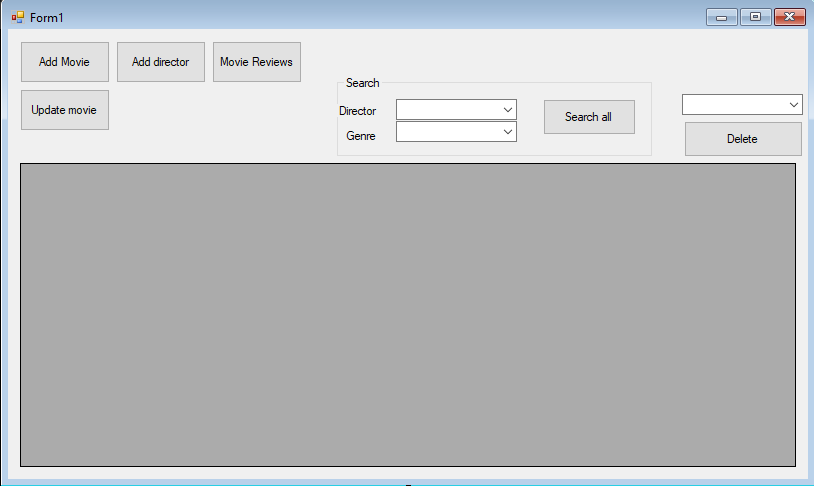


Slika 4: Kolekcia Movie

# Izgled aplikacije

## Početni zaslon

Grafičko sučelje ove aplikacije je napravljeno pomoću Windows formi u Visual Studio-u. Aplikacija se sastoji od pet formi: *MainForm, AddMovieForm, AddDirectorForm, MovieReviewForm i UpdateMovieForm*. Početna forma koja se pojavi pokretanjem aplikacije je *MainForm*.

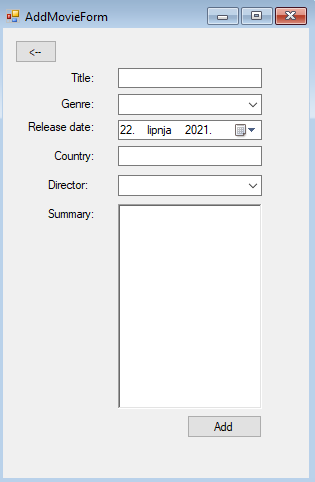


Slika 5: Izgled početne stranice

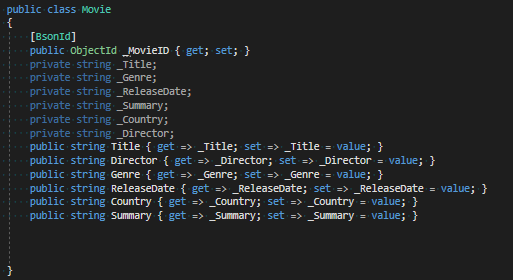
U gorenjem lijevom kutu se nalaze tipke za navigaciju u aplikaciji. Te tipke vode na svoje dotične forme.

## Forma za dodavanje filmova

U slici koja se nalazi ispod je prikazana forma za dodavanje novog filma u bazu podataka. Podatci koje treba unijeti su idući: *title, genre, release date, country, director i summary*. Za spremanje podataka u bazu korištenjem objekata napravljena je istoimena klasa.

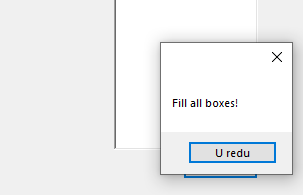


Slika 6: Prikaz forme za dodavanje filma

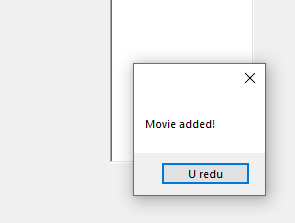


Slika 7: Klasa - Movie

U slučaju da neki od ovih podataka nije unesen javit će se greška prikazana na slici *Slika 5: Greška prilikom unosa*. Kada korisnik uspješno unese film u bazu podataka, ispisuje se poruka vidljiva na slici *Slika 6: Uspješan unos*.



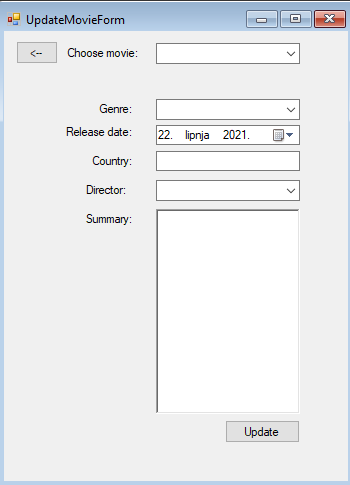
Slika 8: Greška prilikom unosa



Slika 9: Uspješan unos

## Forma za ažuriranje filmova

Sljedeća opcija koju korisnik ima je ažuriranje podataka o filmu. Klikom na botun *Update movie* dolazi se na formu za ažuriranje.



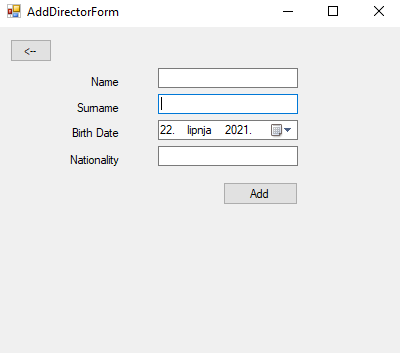
Slika 10: Forma za ažuriranje podataka

Ova forma funkcionira tako da korisnik prvo odabere koji film želi ažurirati. To radi tako da koristi padajući izbornik. Podatci u padajućem izborniku se dinamički osvježavaju, odnosno podatci se dohvaćaju iz baze podataka i prikazuju u padajućem izborniku što omogućava da su vidljivi svi filmovi koje je korisnik unio u drugoj formi.

Nakon odabira filma korisnik unosi podatke koje želi promijeniti. Polja za unos koja korisnik ostavi prazna se ne ažuriraju. Za potvrdu ažuriranja korisnik mora pritisnuti tipku *Update*, nakon čega dobiva poruku o uspješnom ažuriranju.

## Forma za dodavanje redatelja

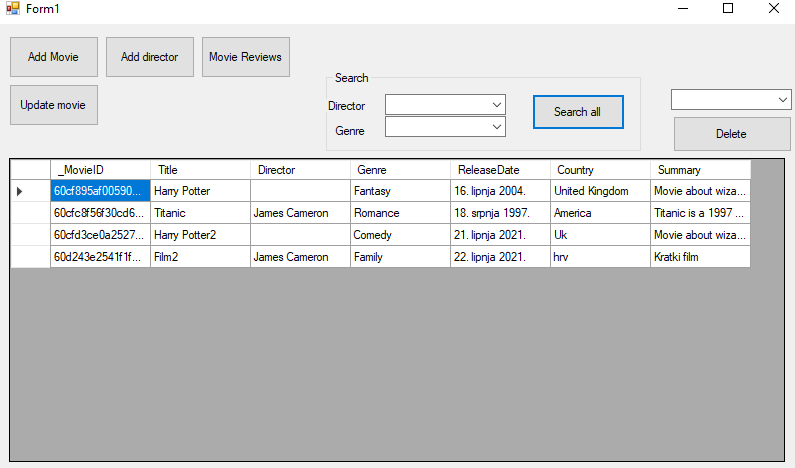
Sljedeća opcija koju korisnik ima je dodavanje redatelja filmova. Klikom na botun *Add director* dolazi se na formu za dodavanje redatelja. Forma je slična kao i forma za dodavanje filma. Od korisnika se isto traži da unese podatke u sva polja prije samog spremanja u bazu podataka.



Slika 11: Forma za dodavanje redatelja

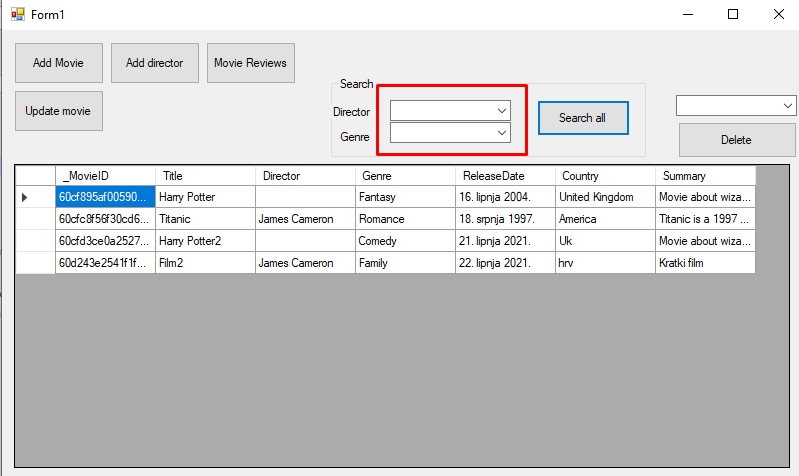
## Glavna forma za ispis podataka, pretragu i brisanje

Za ispise podataka je korišten *DataGridView* objekt. U glavnoj formi se ispisuju podatci o filmovima.



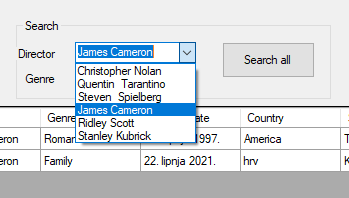
Slika 12: Ispis filmova iz baze

Korisnik filmove može pretražiti na tri načina. Prvi način je odabirom botuna *Search all*. Klikom na taj botun, na ekran u *DataGridView* se ispisuju svi filmovi koji su spremljeni u bazi podataka. Druga dva načina su pretraživanje po nekim vrijednostima. U ovom slučaju korisnik može pretražiti filmove po vrijednosti redatelja i žanra. Kako bi se olakšala pretraga postavljena su dva padajuća izbornika.

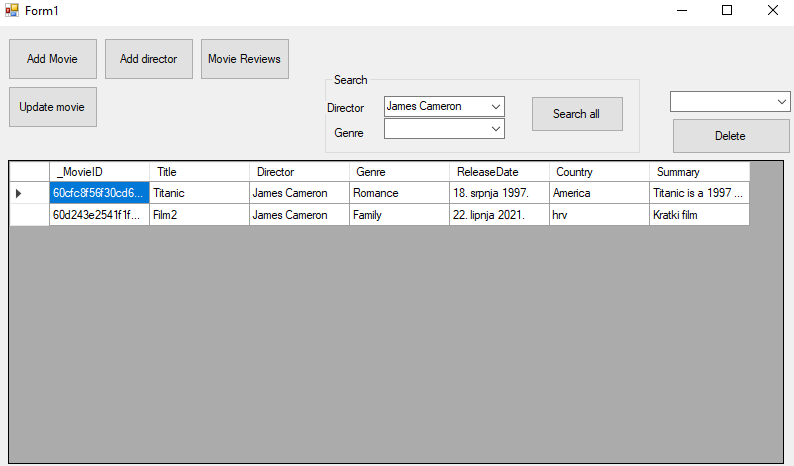


Slika 13: Pretraga filmova pomoću argumenta

U prvom padajućem izborniku se dinamički dohvaćaju vrijednosti svih redatelja koji su spremljeni u bazi podataka, te odabirom jednog se automatski ažuriraju podatci u DataGridViewu i ispisuju filmovi samo tog redatelja.

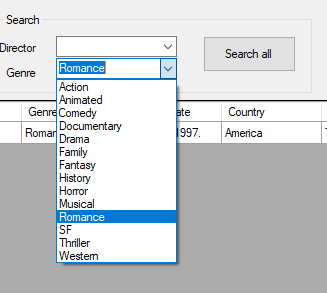


Slika 14: Odabir redatelja za pretragu



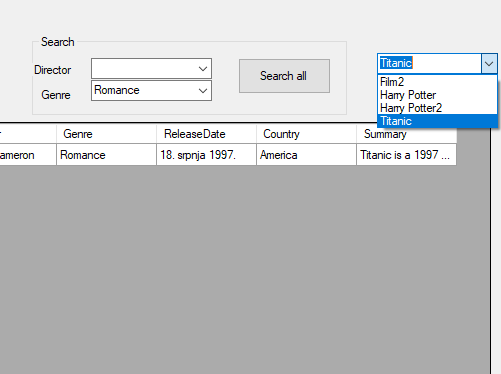
Slika 15: Ispis nakon odabira redatelja

Na isti način funkcionira i drugi padajući izbornik, gdje se odabirom nekog žanra ispisuju svi filmovi koji se nalaze u bazi podataka i izabranog su žanra.



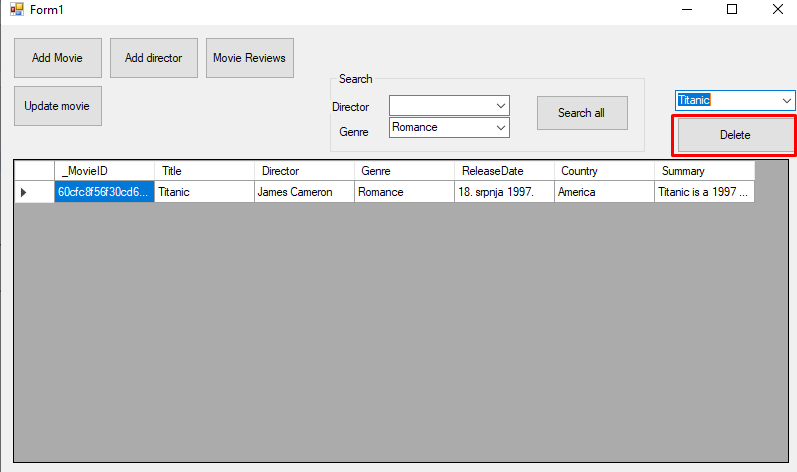
Slika 16: Pretraga po žanrovima

Još jedna opcija koja se nalazi na ovoj formi je opcija brisanja. Brisanje filmova iz baze podataka se odvija u dva koraka. Prvo se odabere film koji se želi izbrisati iz padajućeg izbornika.



Slika 17: Odabir filma za brisanje

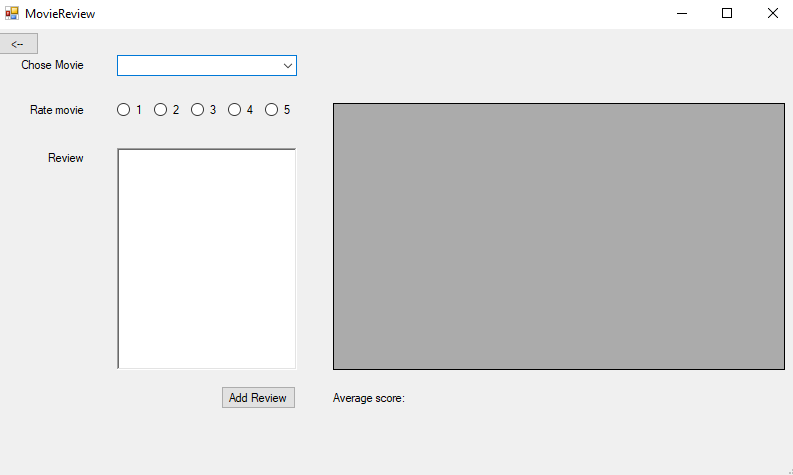
Nakon toga se pritiskom na botun Delete briše odabrani film iz baze podataka.



Slika 18: Brisanje filma

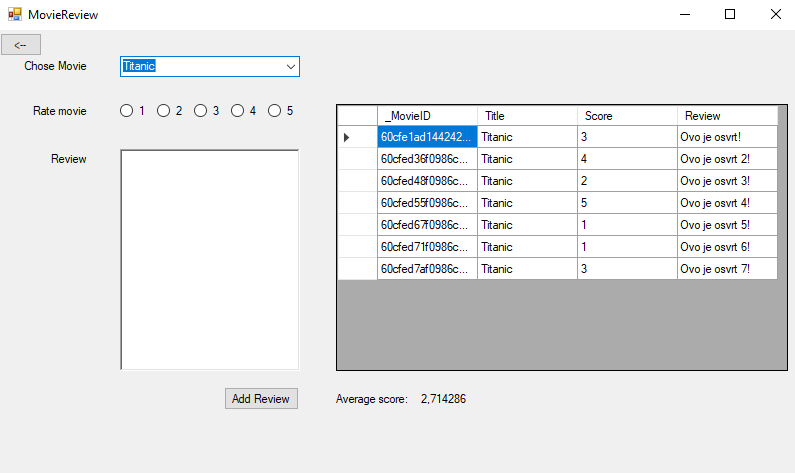
## Forma za unos osvrta na film

Posljednja forma koja je preostala je forma za unos osvrta na film.



Slika 19: Forma za unos osvrta i ocjene

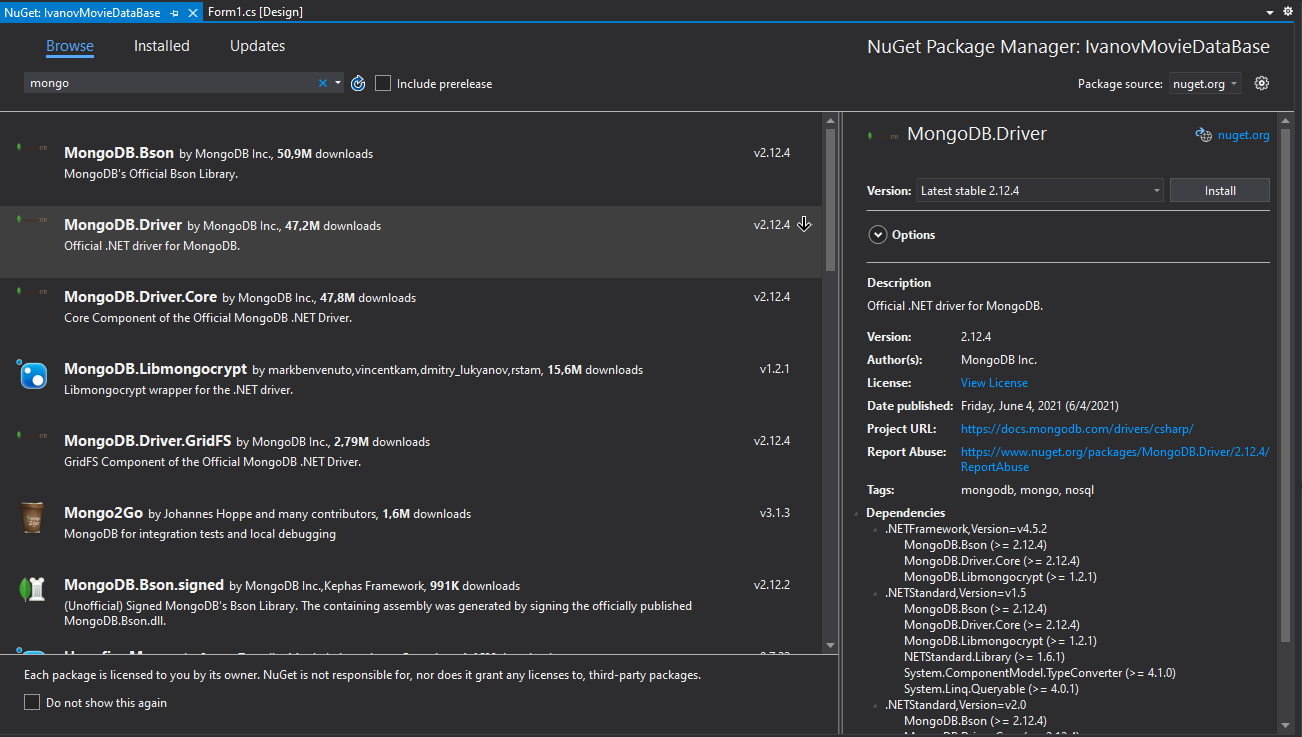
U padajućem izborniku se odabire koji film želite ocijeniti i napisati osvrt na njega. Nakon odabira filma, s desne strane ekrana u DataGridViewu se ispisuju svi trenutni osvrti za taj film koji se nalaze u bazi podataka i ispod njega se ispisuje prosjek ocjena za odabrani film.



Slika 20: Ispis osvrta i ocjena

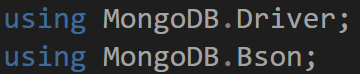
# MongoDB

Podaci koje aplikacija koristi spremaju se u nerelacijsku bazu podataka MongoDB. MongoDB se može preuzeti sa interneta na danom linku: <https://www.mongodb.com/try/download/community>. Kako bi našu C# aplikaciju povezali sa MongoDB potrebno je desnim klikom miša kliknuti na Solution te odabrati „Manage NuGet Packages“ te instalirati paket MongoDB.Driver.



Slika 21: Dodavanje MongoDB biblioteka u aplikaciju

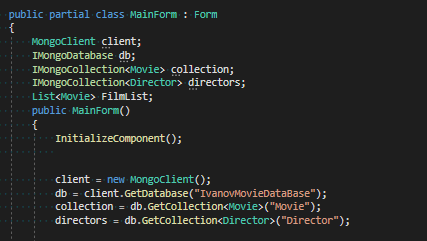
Nakon instalacije je u Visual Studiu potrebno uključiti MongoDB kako bi mogli koristiti reference neophodne za rad s MongoDB kao što je prikazano na slici ispod.



Slika 22-Uključivanje MongoDB u Visual Studiu

## Konekcija s bazom podataka

Nakon što smo u projekt uključili MongoDB potrebno je napraviti konekciju s bazom podataka.



Slika 23: Konekcija s bazom podataka

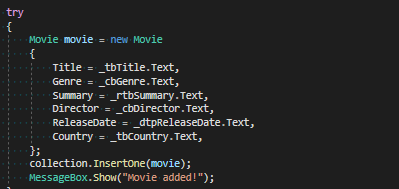
## Stvaranje kolekcije

Kako bi mogli raditi upite nad bazom podataka potrebno je dohvatiti pojedinu kolekciju iz baze podataka. Na slici ispod se vidi način na koji se dohvati kolekcija u kojoj su spremljeni podatci o redateljima.



Slika 24: Dohvaćanje kolekcije

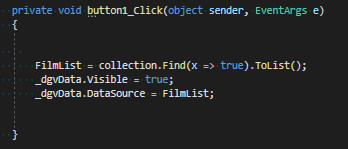
Kada želimo podatke spremiti u bazu moramo prvo dohvatiti kolekciju koju želimo. Nakon što to napravimo metodom *InserOne* dodajemo objekt, u ovom slučaju objekt *Movie*, u bazu podataka.



Slika 25: Dodavanje dokumenta u kolekciju pomoću metode InsertOne

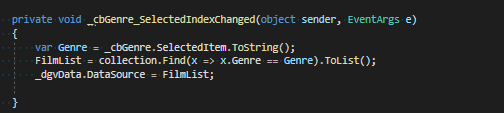
## Pretraživanje kolekcije Artikl

Kolekcije pretražujemo pomoću metode *Find.* U ovoj aplikaciji smo radili 2 vrste pretraga. Prva je bila pretraga koja nema argument pretrage.



Slika 26: Pretraga kolekcije pomoću metode Find

Druga vrsta je bila kada šaljemo neki argument metodi *Find*. U ovom primjeru metodi se šalje vrijednost odabranog žanra i to je argument po kojem se vrši pretraga baze podataka.



Slika 27: Pretraga kolekcije koristeći argument

## Ažuriranje podataka kolekcije Artikl

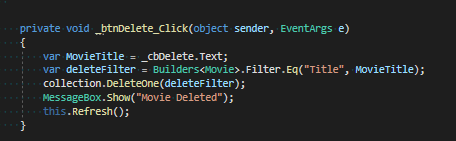
Ažuriranje podataka u kolekcijama vrši se pomoću metode *Update*.



Slika 28: Ažuriranje podataka kolekcije Movie

## Brisanje podataka kolekcije Artikl

Brisanje podataka iz baze podataka vrši se pomoću metoda *DeleteOne* ili *DeleteMany*. S obzirom na to da su se u ovoj aplikaciji brisali pojedini filmovi korištena je metoda *DeleteOne*.



Slika 29: Brisanje dokumenta iz kolekcije

# Zaključak

Izrada aplikacije koristeći nerelacijsku bazu podataka može biti malo izazovna jer se prvi put susrećemo s takvom bazom. S obzirom na to da ona funkcionira na drugačiji način od relacijskih baza moramo promijeni način na koji razmišljamo o podatcima te o dohvaćanju, dodavanju, brisanju i ažuriranju podataka u bazi. Nakon te neke početne faze navikavanja, korištenje nerelacijskih baza za osnovne naredbe postaje lagano. Izrada ove aplikacije je tekla bez većih problema zahvaljujući dostupnim materijalima profesora Dadića u kojima su objašnjene osnovne funkcionalnosti i naredbe. Dodatna pitanja koja sam imao vezano za razvoj aplikacije sam jednostavno i lako pronašao pretražujući Internet.

# Popis slika

[Slika 1. MongoDb Compass Community sučelje i prikaz storenih kolekcija 3](#_Toc75300453)

[Slika 2: Kolekcija Director 4](#_Toc75300454)

[Slika 3: Kolekcija MovieReview 4](#_Toc75300455)

[Slika 4: Kolekcia Movie 5](#_Toc75300456)

[Slika 5: Izgled početne stranice 6](#_Toc75300457)

[Slika 6: Prikaz forme za dodavanje filma 7](#_Toc75300458)

[Slika 7: Klasa - Movie 8](#_Toc75300459)

[Slika 8: Greška prilikom unosa 8](#_Toc75300460)

[Slika 9: Uspješan unos 8](#_Toc75300461)

[Slika 10: Forma za ažuriranje podataka 9](#_Toc75300462)

[Slika 11: Forma za dodavanje redatelja 10](#_Toc75300463)

[Slika 12: Ispis filmova iz baze 10](#_Toc75300464)

[Slika 13: Pretraga filmova pomoću argumenta 11](#_Toc75300465)

[Slika 14: Odabir redatelja za pretragu 11](#_Toc75300466)

[Slika 15: Ispis nakon odabira redatelja 12](#_Toc75300467)

[Slika 16: Pretraga po žanrovima 12](#_Toc75300468)

[Slika 17: Odabir filma za brisanje 13](#_Toc75300469)

[Slika 18: Brisanje filma 13](#_Toc75300470)

[Slika 19: Forma za unos osvrta i ocjene 14](#_Toc75300471)

[Slika 20: Ispis osvrta i ocjena 15](#_Toc75300472)

[Slika 21: Dodavanje MongoDB biblioteka u aplikaciju 16](#_Toc75300473)

[Slika 22-Uključivanje MongoDB u Visual Studiu 16](#_Toc75300474)

[Slika 23: Konekcija s bazom podataka 17](#_Toc75300475)

[Slika 24: Dohvaćanje kolekcije 17](#_Toc75300476)

[Slika 25: Dodavanje dokumenta u kolekciju pomoću metode InsertOne 17](#_Toc75300477)

[Slika 26: Pretraga kolekcije pomoću metode Find 18](#_Toc75300478)

[Slika 27: Pretraga kolekcije koristeći argument 18](#_Toc75300479)

[Slika 28: Ažuriranje podataka kolekcije Movie 19](#_Toc75300480)

[Slika 29: Brisanje dokumenta iz kolekcije 19](#_Toc75300481)