

Dokumentacija projekta iz predmeta: Razvoj višeslojnih aplikacija u elektroenergetskim sistemima

Web aplikacija za rukovanje ateljeima

Autor: Aleksandar Stamenković PR34-2018

Mentor: Željko Čavić

Profesor: Darko Čapko

Opis aplikacije:

Web aplikacija za rukovanje ateljeima (Atelje Management). U okviru aplikacije korisnik ima mogućnost za rukovanje (dodavanje, brisanje, izmenu i prikaz) ateljea, umetničkih dela i autora tih dela. Primarni podaci u aplikaciji su ateljei, i nad akcijama koje se sprovode nad ateljeima je moguće izvršavati undo/redo. Ukoliko je korisnik administrator, ima dodatnu mogućnost da registruje korisnike. Pored osnovnih funkcionalnosti, dostupni su log podaci za svakog korisnika, kao i log podaci celog sistema.

Odjava

Registruj korisnika

Ateljei

Umetnička dela

Autori

Adresa:	PIB:	MBR:	
Nikole Pasica 3	222222	455423	
Izmeni	Izbrisi	Detalji	Dupliraj
Ive Andrica 1	123123	123123	
Izmeni	Izbrisi	Detalji	Dupliraj
Zlatne Grede 10	555555	123456	
Izmeni	Izbrisi	Detalji	Dupliraj
Bulevar Oslobođenja 1	456456	654654	
Izmeni	Izbrisi	Detalji	Dupliraj
Nikole Pasica 15	111111	111111	
Izmeni	Izbrisi	Detalji	Dupliraj

adresa:Nikole Pasica 3

mmb:222222

pib:455423

Adresa

Nikole Pasica 3

Mmbr

222222

Pib

455423

Sacuvaj

Udajg emnteta

Dodaj atelje

Dodaj umetnicko delo

Dodaj autora

Umetnicko delo:

Naziv:

Naziv dela:

Pravac:

Pejzaz

Stil

Barok

Autor:

Pera Peric

Atelje:

Nikole Pasica 3

Dodaj

Korišćena tehnologija:

Frontend:

Za frontend aplikacije korišćen je [ReactJS](#), u okviru koga su korišćene biblioteke [js-sha256](#) (za enkodovanje), [Redux](#) (za stanje podataka u aplikaciji).

Backend:

Za backend aplikacije korišćen je [ASP.NET Core](#) 5.0 (šablon za razvoj aplikacija u Visual Studio ASP.NET Core with React).

Povezivanjem backend-a i frontend-a dobija se SPA (Single-Page Application) RESTful (Representational State Transfer) API (Application Programming Interface).

Kontakt sa bazom:

Za kontakt sa bazom korišćen je [Entity Framework](#) (Code First pristup, generisanje klasa za rukovanje entitetima u bazi na osnovu baze podataka koja je napravljena u sistemu za rukovanje bazom podataka (DBMS)).

DBMS (Database Management System):

Korišćen je [Microsoft SQL Server](#) (uz [Microsoft SQL Server Management Studio](#)) za kreiranje, pregled i rukovanje konkretnom bazom podataka.

Modelovanje projekta:

Korišćen je [Enterprise Architect](#) (klasni dijagram).

Serijalizacija (Deserijalizacija) podataka:

Backend i frontend razmenjuju JSON objekte, za tu svrhu se koristi [Newtonsoft.Json](#) biblioteka.

Korišćeni obrasci za razvoj softvera:

1. Singleton: primenjen nad klasom koja predstavlja kontekst baze podataka (DbContext, konkretno u projektu AteljeDB) radi uniformnog pristupa, uz korišćenje lock blokova gde je potrebno zbog osiguravanja konzistentnosti podataka koja može biti ugrožena zbog multi-thread razvoja web aplikacije.

```
48 references | Aleksandar Stamenković, 9 days ago | 1 author, 2 changes
public partial class AteljeDB : DbContext
{
    1 reference | Aleksandar Stamenković, 9 days ago | 1 author, 1 change
    public AteljeDB()
        : base("name=AteljeDB")
    {
    }

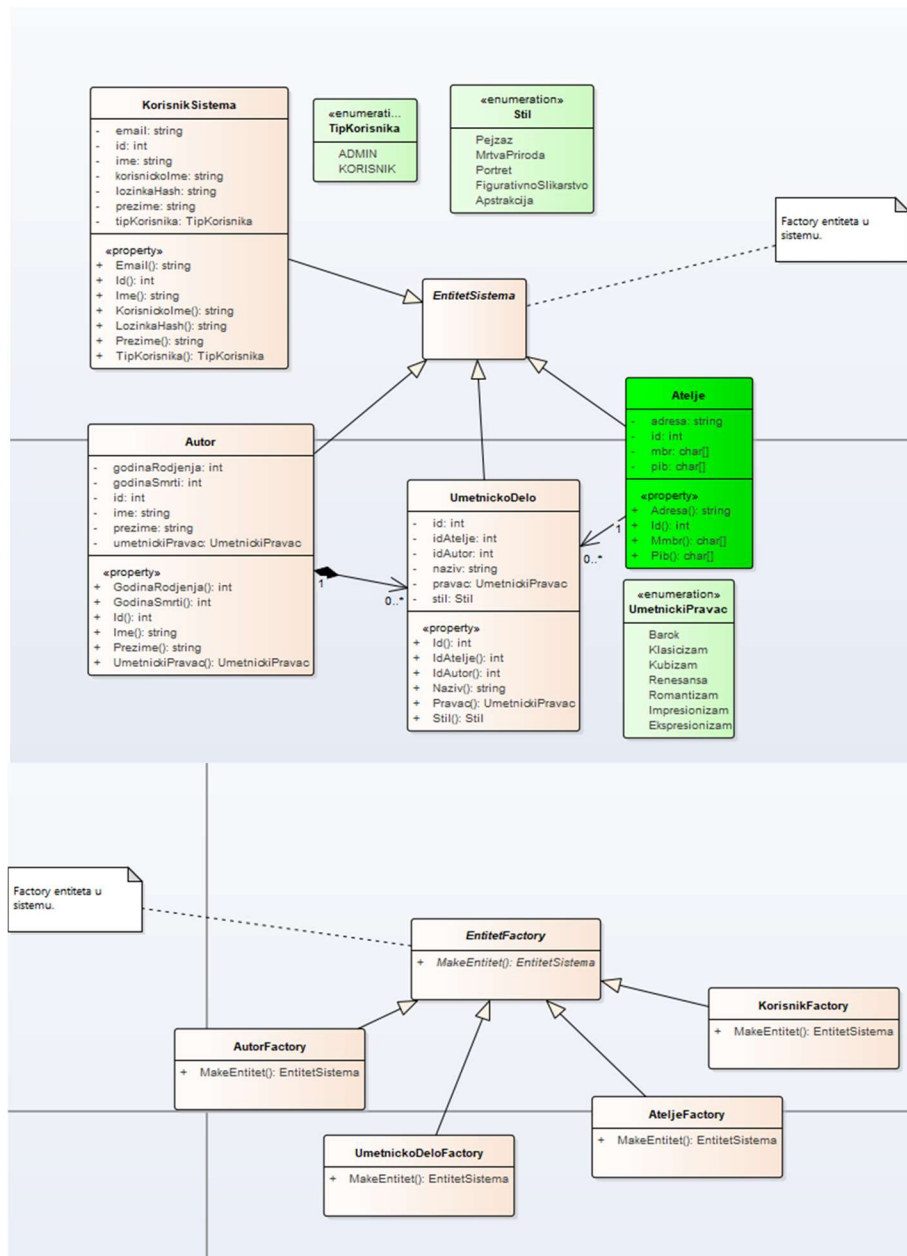
    private static AteljeDB instance;

    22 references | Aleksandar Stamenković, 9 days ago | 1 author, 1 change
    public static AteljeDB Instance()
    {
        if (instance == null)
            instance = new AteljeDB();

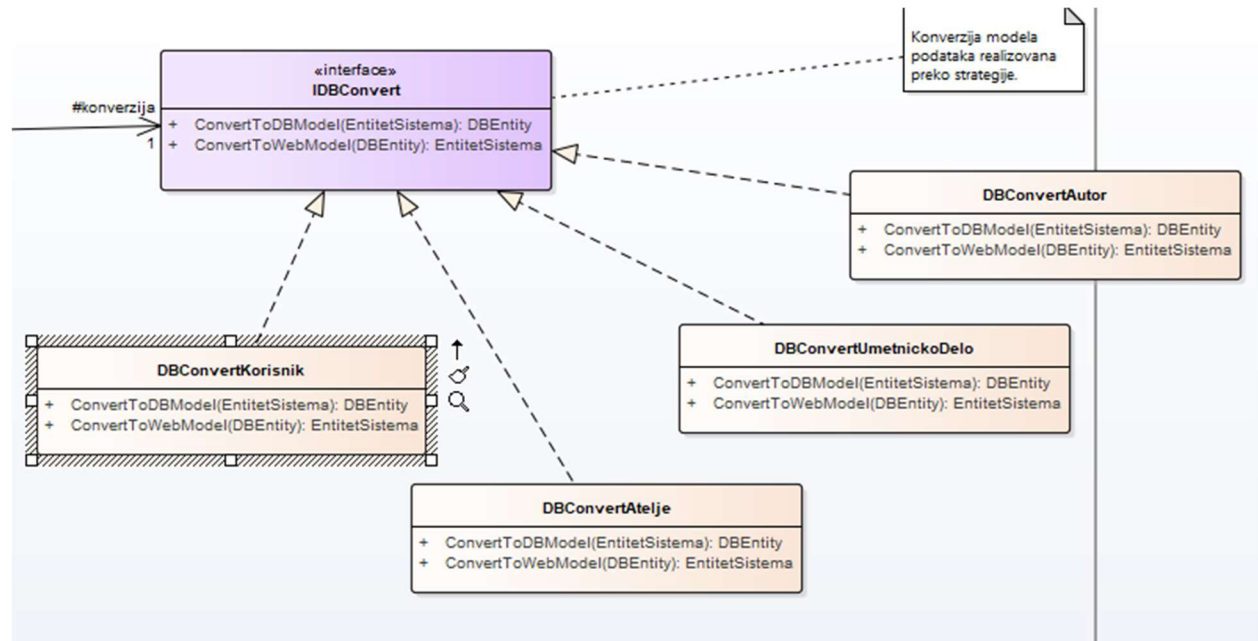
        return instance;
    }

    12 references | Aleksandar Stamenković, 9 days ago | 1 author, 1 change
    public virtual DbSet<Atelje> Ateljes { get; set; }
    9 references | Aleksandar Stamenković, 9 days ago | 1 author, 1 change
```

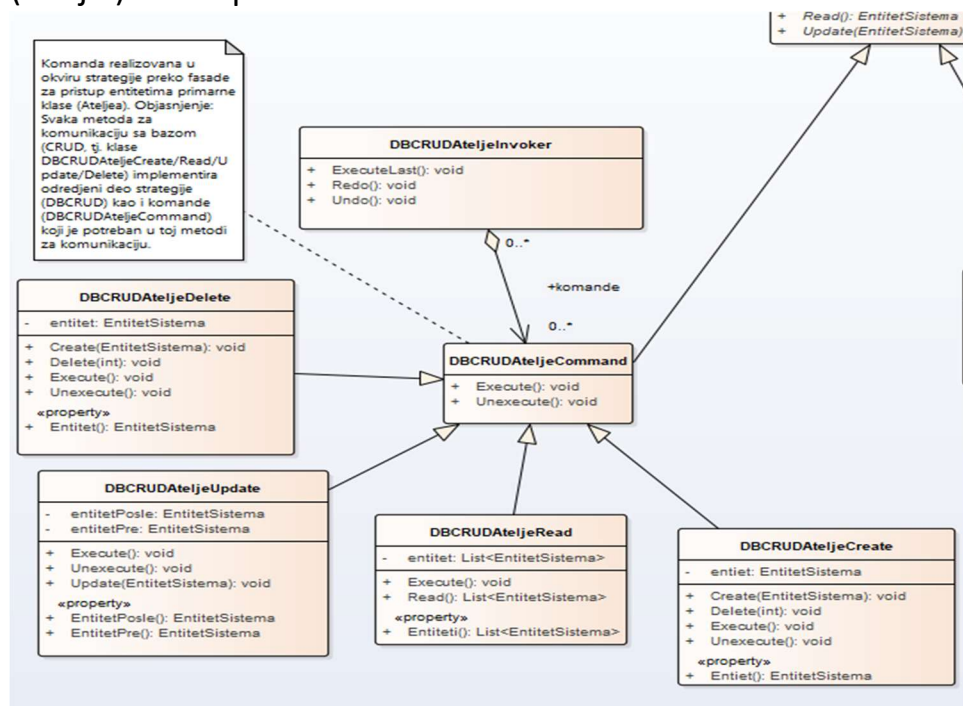
2. Factory: primenjena za kreiranje entiteta u sistemu (ateljei, umetnička dela, autori, korisnici), omogućava uniformni tip podataka za korišćenje u aplikaciji.



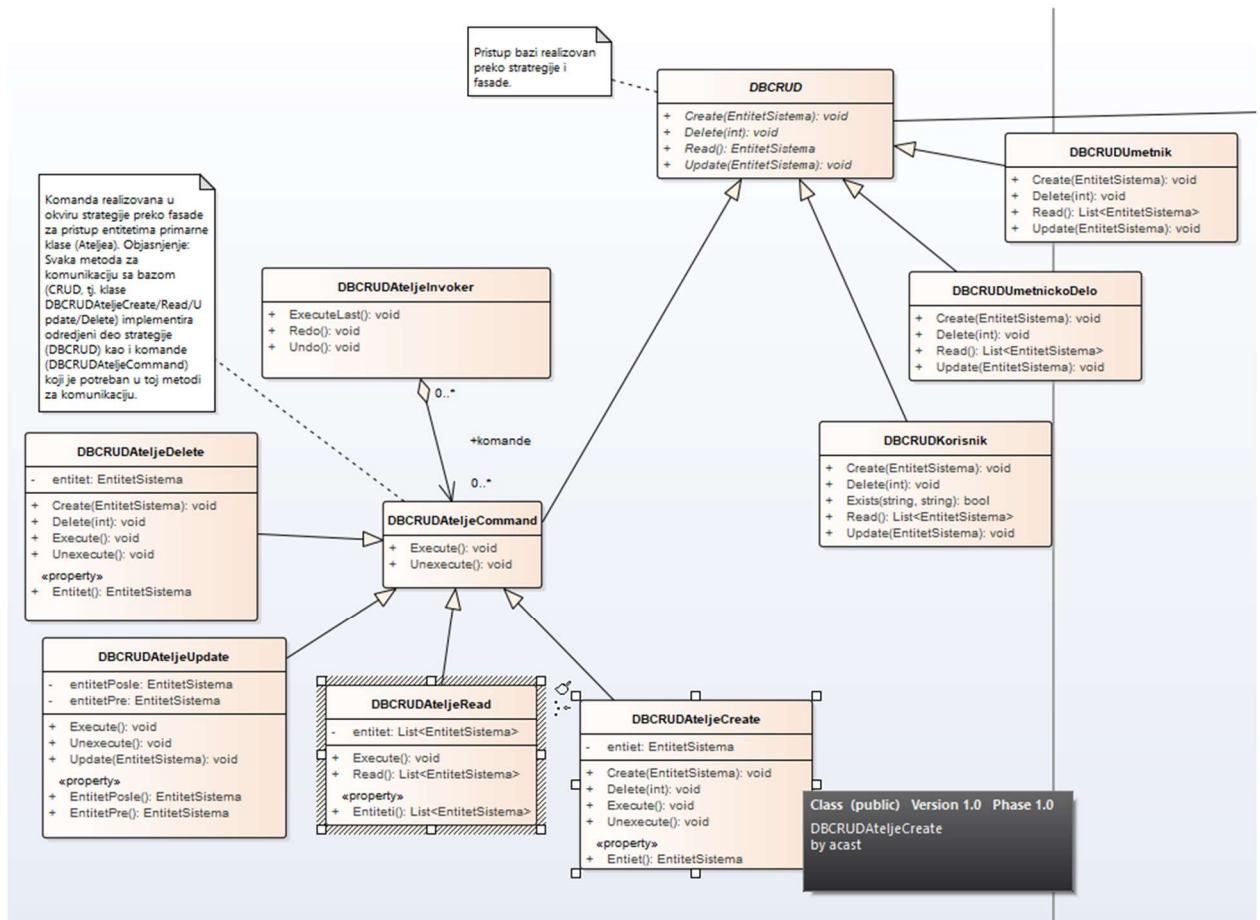
3. Strategy: za konverziju modela podataka Web aplikacije u model podataka baze podataka(koji je izgenerisao Entity Framework) i obrnuto.



4. Command: za mogućnost undo/redo nad entitetima primarnih podataka (ateljei) u bazi podataka.



5. Strategy i facade: za celokupan pristup bazi.



6. Observer: primenjen implicitno u povezivanju frontenda i backenda, kroz osvežavanje prikaza potrebnih entiteta nakon njihove promene.

Diagram projekata u Visual Studio-u (reference):

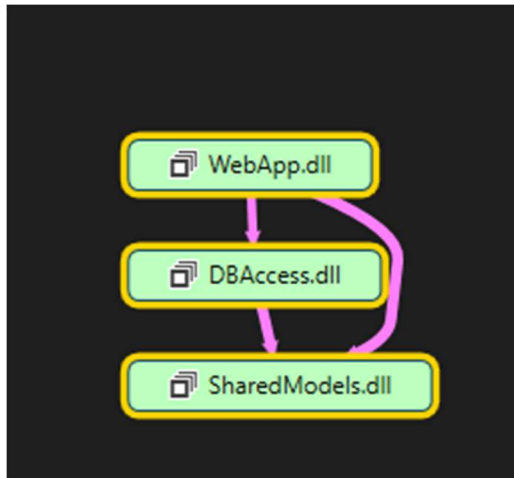
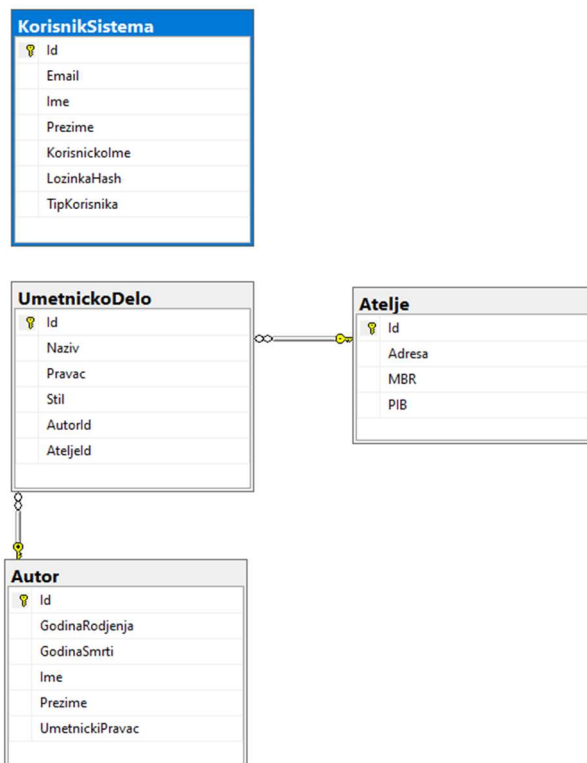


Diagram baze podataka:



Github repozitorijum:

[AteljeProjekat](#)

Potrebno je da autor (Aleksandar Stamenković) doda kao collaborator-a osobu koja želi pristup datom repozitorijumu (jer je privatan). Ukoliko je potrebno pristup se može promeniti na javni.