

RVA Projekat

Dokumentacija

Andrija Ristić PR6-2019

1. Tehnologije

Serverska aplikacija je izgradjena korišćenjem ASP.NET web tehnologije i troslojne web arhitekture. Upotrebljena je .NET 6 verzija. Korišćeni NuGet paketi su:

- **Microsoft.EntityFrameworkCore v6.0.7, Microsoft.EntityFrameworkCore.Tools v6.0.7, Microsoft.EntityFrameworkCore.SqlServer v6.0.7 :**
Korišćeni za rad sa bazom podataka. Korišćena je Microsoft SQL Server baza podataka.
- **Microsoft.AspNetCore.Authentication.JwtBearer v6.0.7:** Korišćen za izdavanje i verifikaciju tokena za autorizaciju i autentifikaciju korisnika.
- **Bcrypt.Net-Next v4.0.3:** Korišćen za enkripciju korisničkih lozinki.
- **AutoMapper v11.0.1:** Korišćen za mapiranje objekata.
- **Serilog v2.12:** Korišćen za logovanje korisničkih akcija.

Klijentska aplikacija je izgradjena korišćenjem **React.js v18.2.0** biblioteke. Za rutiranje je korišćena **React Router v5.3.3** biblioteka.

2. Model podataka i relacije u bazi podataka

Baza podataka se sastoji iz 6 tabela.

Tabela **Users** čuva informaciju o registrovanim korisnicima sistema. Svaki entitet ima Id (generisan pri dodavanju novog entita u tabelu), Username, Password (sačuvan je u enkriptovanom formatu), Name, Lastname i UserType (STUDENT ili ADMIN).

	Id	Username	Password	Name	Lastname	UserType
	1	admin	\$2a\$11\$.Zjj9K4...	Andrija	Ristic	0
	2	string	\$2a\$11\$kHEOg...	Marko	Markovic	1
	4	guest	\$2a\$11\$u07Ircb...	Ana	Antic	1
	6	guest1	\$2a\$11\$F2aUM...	Tanja	Tasic	1

Slika 1. Tabela Users

Tabela **Students** čuva informaciju o svim registrovanim korisnicima koji su studenti (UserType Usera je STUDENT). Svaki entitet ima Id (generisan pri dodavanju novog entita u tabelu), Name, LastName i UserUsername (predstavlja korisničko ime Usera na koji se odnosi student).

	Id	Name	LastName	UserUsername
	1	Marko	Nikolic	string
	2	Ana	Antic	guest
	4	Tanja	Tasic	guest1
	9	Nikola	Matic	userFront

Slika 2. Tabela Students

Tabela **Exams** čuva informaciju o svim postojećim ispitima. Svaki entitet ima Id (generisan pri dodavanju novog entita u tabelu), ExamName, SubjectId (strani ključ koji pokazuje na entitet iz tabele Subjects) i ExamDate

	Id	ExamName	SubjectId	ExamDate
	8	RVA - Januarski rok	3	19-Aug-22 17:32:31
	13	Ispit #2	9	19-Aug-22 17:32:31
	14	Ispit #2	10	19-Aug-22 17:32:31
	15	RVA - Oktobar #1	5	30-Sep-22 00:00:00
	16	Analiza #1	2	29-Aug-22 19:40:00

Slika 3. Tabela Exams

Tabela **Subject** čuva informaciju o svim postojećim predmetima. Svaki entitet ima Id (generisan pri dodavanju novog entita u tabelu) i SubjectName.

	Id	SubjectName
	2	Analiza
	3	Razvoj viseslojnog softvera
	4	Cloud Computing
	5	Razvoj viseslojnih aplikacija

Slika 4. Tabela Subject

Tabela **StudentResult** čuva informaciju o svim postojećim prijavama na ispit od strane studenta. Svaki entitet ima Id (generisan pri dodavanju novog entita u tabelu), ExamId (strani ključ koji pokazuje na entitet iz tabele **Exams**), StudentId (strani ključ koji pokazuje na entitet iz tabele **Students**), Result i isTouched (polje koje pokazuje da li je ocenjen taj student).

	Id	ExamId	StudentId	Result	isTouched
	6	8	1	False	True
	8	8	2	True	True
	57	16	18	True	True

Slika 5. Tabela StudentResult

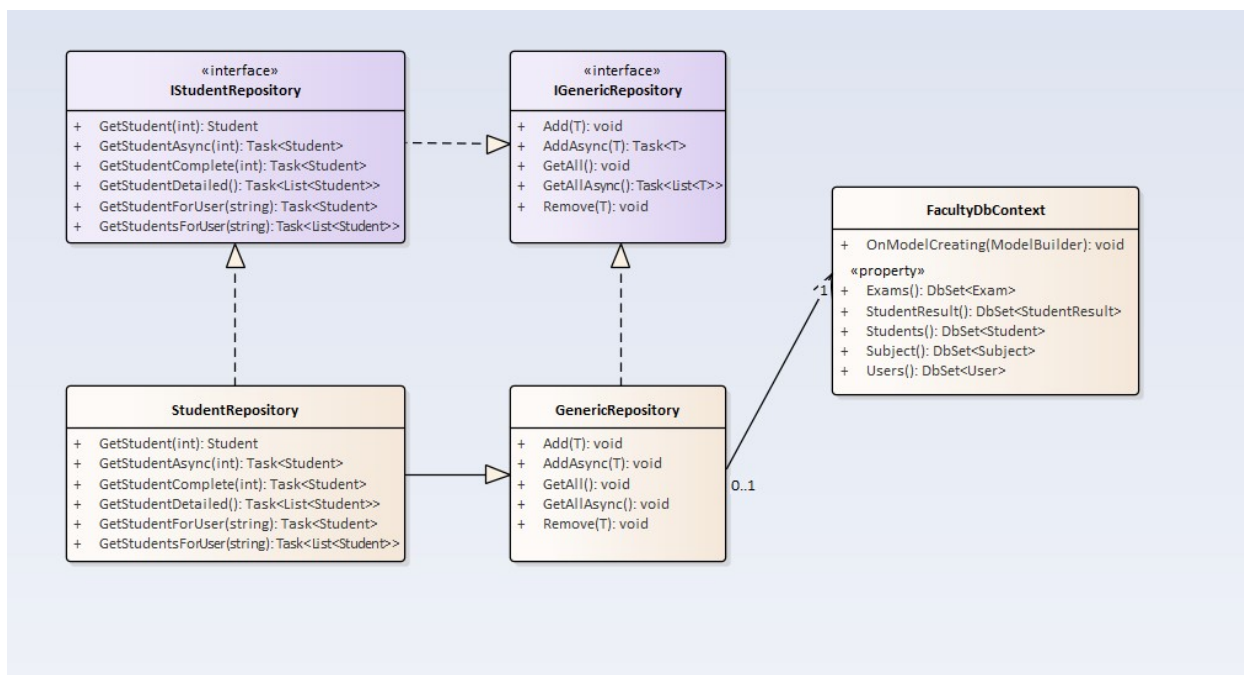
Tabela **ExamStudent** čuva informacije o vezama izmedju entita koji pripadaju tabelama **Exams** i **Students** (na ovaj nacin se modeluje veza više na više). Svaki entitet ima ExamsId (strani ključ koji pokazuje na entitet iz tabele **Exams**) i StudentsId (strani ključ koji pokazuje na entitet iz tabele **Students**).

	ExamsId	StudentsId
	8	1
	8	2
	13	11
	14	12
	16	18

Slika 6. Tabela ExamStudent

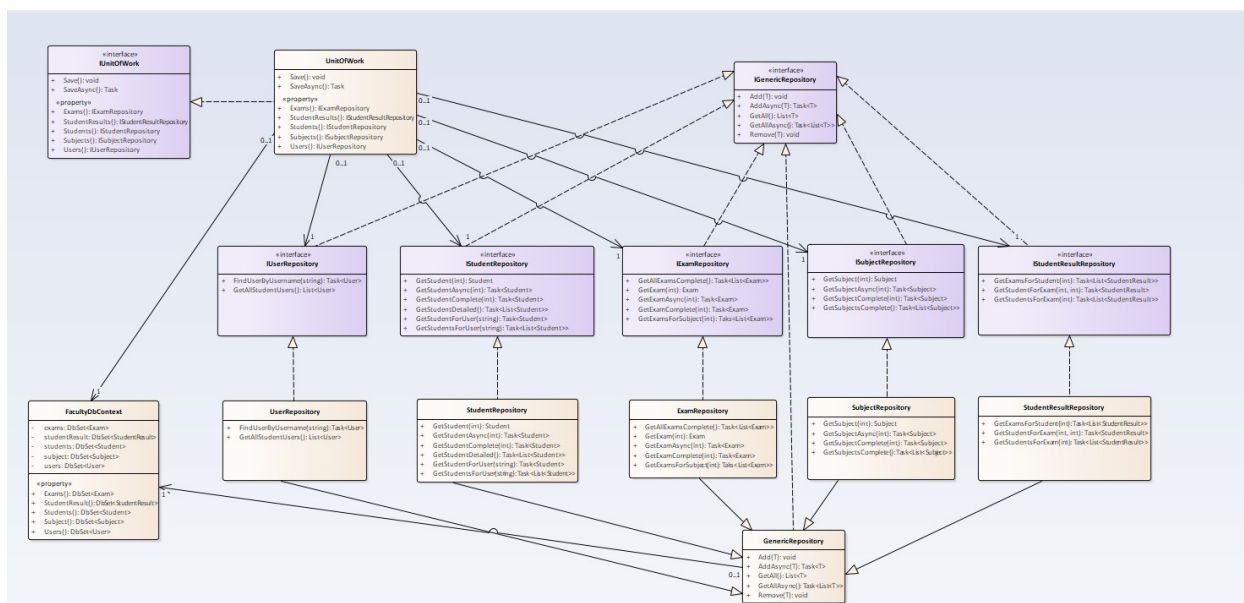
3. Primenjeni programski obrasci

Repository obrazac je iskorišćen kako bi se napravio sloj apstrakcije izmedju servisnog sloja i sloja podataka.



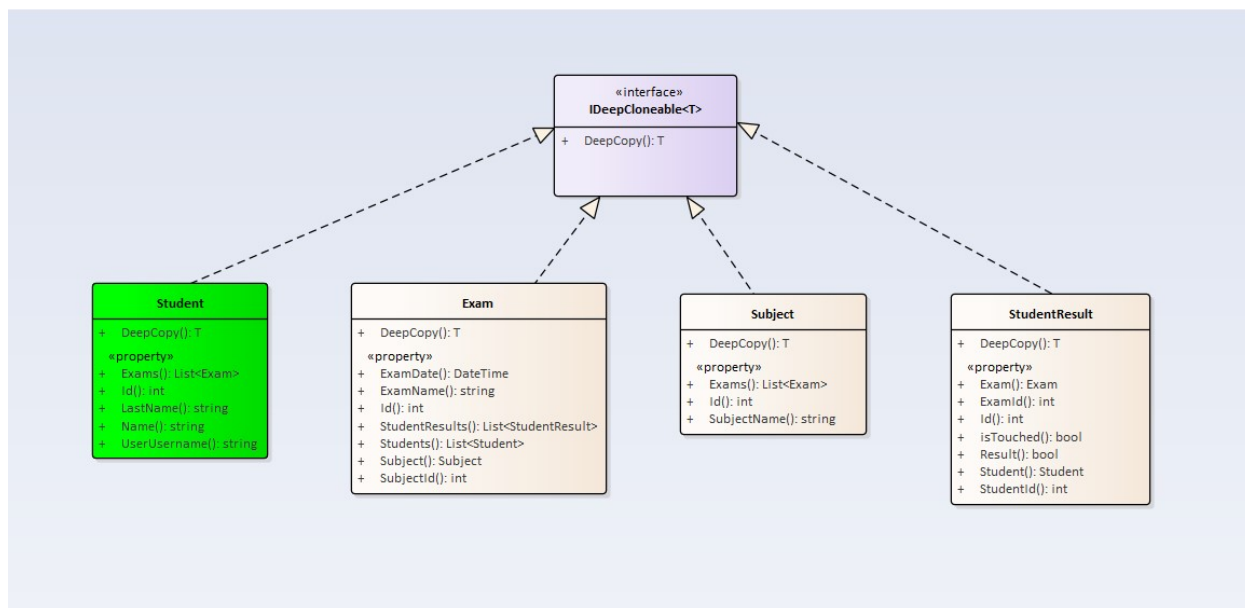
Slika 7. Repository obrazac

Unit of Work obrazac je iskorišćen kako bi svi repozitorijumi bili objedinjeni i kako bi bilo osigurano da svaki ima referencu ka istom DbContext-u.



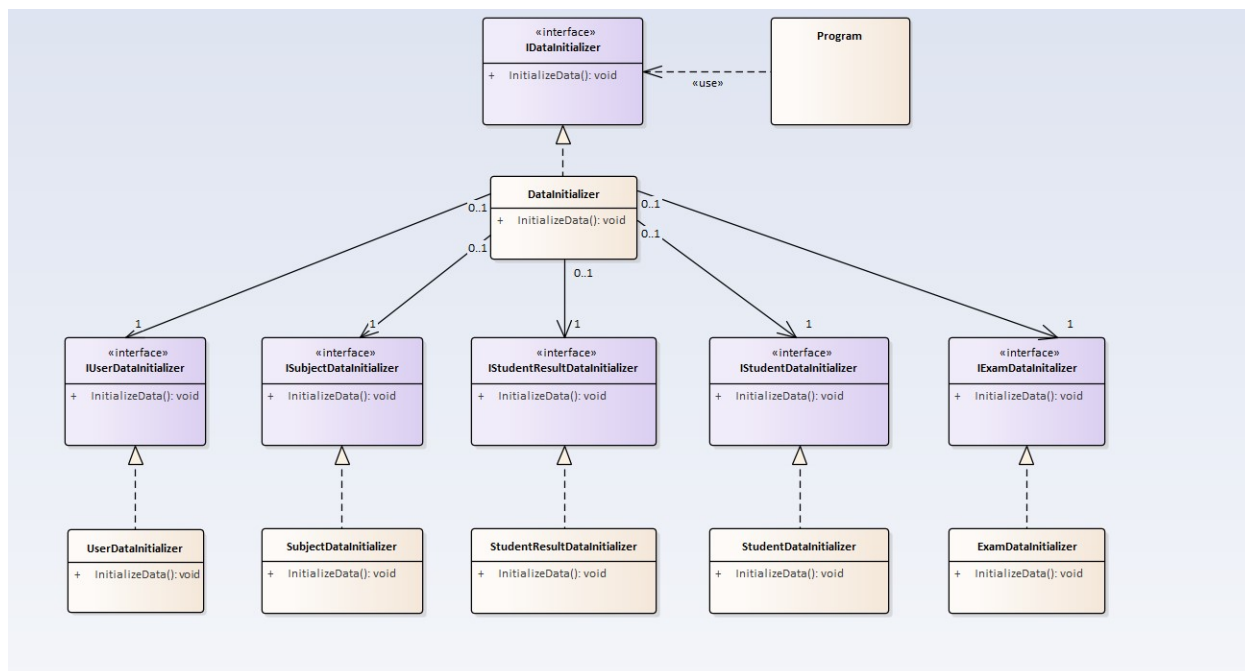
Slika 8. Unit of Work obrazac

Prototype obrazac je iskorišćen radi pravljenja duboke kopije (**Deep copy**) prilikom dupliranja primarnih podataka.



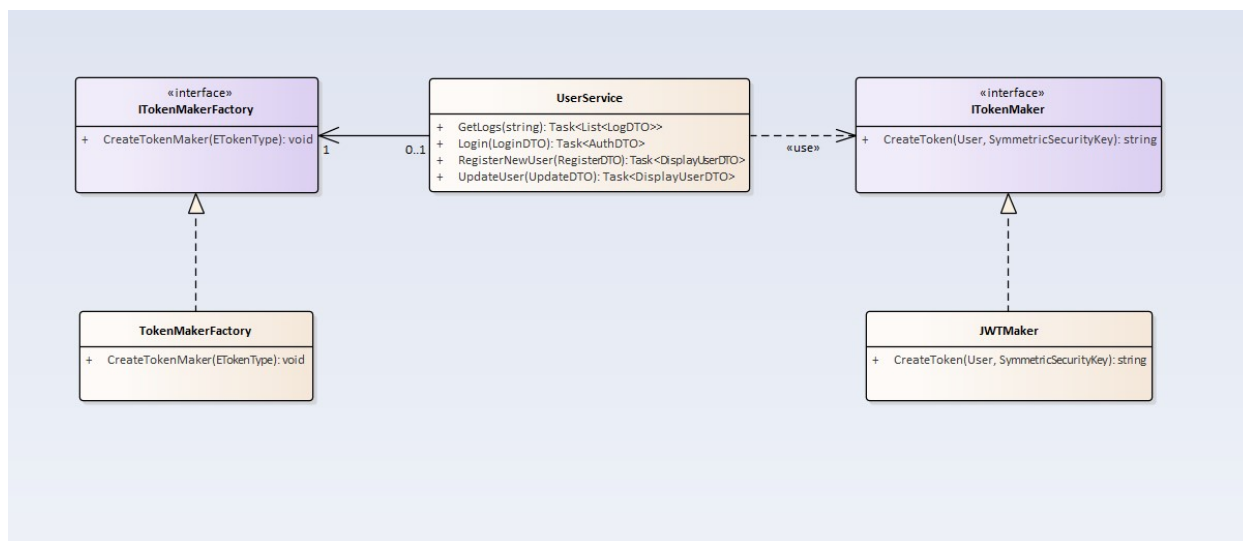
Slika 9. Prototype obrazac

Facade obrazac je iskorišćen za inicijalizaciju podataka. Koristi se tako da prilikom inicijalizacije samo pozivamo metodu InitializeData(), umesto posebno svaku metodu za inicijalizaciju podataka odgovarajuće tabele.



Slika 10. Facade obrazac

Factory Method obrazac je iskorišćen kako bi pravili objekat koji služi za kreiranje tokena. Trenutno, samo postoji implementacija za JWTMaker, ali ovakav pristup omogućava proširenje za bilo koji drugi tip tokena.



Slika 11. Factory Method obrazac

4. Pokretanje aplikacije

Prvo je potrebno pokrenuti serversku aplikaciju, za čije pokretanje je potrebno imati instaliran Visual Studio 2022 i sve potrebne NuGet pakete. Nakon pokretanja serverske aplikacije potrebno je pokrenuti klijentsku aplikaciju. Za to je potrebno imati instaliran Node.js v16.16.0 (LTS), zatim pozicionirati se u direktorijum gde se nalazi klijentska aplikacija i izvršiti komande **“npm install”** i **“npm start”** u tom redosledu.