Eventopia

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autori** |
| 13.04.2024. | 1.0 | Inicijalna verzija | Luka Veličković,  Dušan Gligorijević,  Veljko Živković |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 5

2. Opseg dokumenta 5

3. Reference 5

4. Predstavljanje arhitekture 5

5. Ciljevi i ograničenja arhitekture 5

6. Pogled na slučajeve korišćenja 5

6.1 Dijagrami slučajeva korišćenja 6

6.2 Kratak opis slučajeva korišćenja 7

6.2.1 Izbor jezika 7

6.2.2 Registracija 7

6.2.3 Pregled liste preporučenih događaja 7

6.2.4 Filtriranje događaja 7

6.2.5 Ocenjivanje događaja 7

6.2.6 Unos osnovnih podataka o događaju 8

6.2.7 Rezervacija prostora 8

6.2.8 Preuređivanje plana prostora 8

6.2.9 Pretraživanje događaja 8

6.2.10 Pretraživanje dostupnih prostora 8

6.2.11 Pregled statistike o događaju 8

6.2.12 Pregled liste aktivnih događaja 8

6.2.13 Prijavljivanje 8

6.2.14 Pregled istorije posećenih događaja 8

6.2.15 Pregled detalja o događaju 8

6.2.16 Rezervacija mesta 8

6.2.17 Dodavanje plana prostora 8

6.2.18 Modifikacija plana prostora 8

6.2.19 Pregled statistike o prostoru 9

6.2.20 Uklanjanje događaja 9

6.2.21 Izdavanje zabrane pristupa korisniku 9

6.2.22 Pregled liste korisnika 9

7. Pogled na logičku arhitekturu sistema 9

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10

7.1.1 Korisnički interfejs 10

7.1.2 Aplikaciona logika 10

7.1.3 React.js 10

7.1.4 .NET Core 10

7.1.5 DBMS 10

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 .NET Core 11

8.1.4 SQL Server 11

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 12

10. Pogled na implementaciju sistema 13

10.1 Model domena 13

10.2 Šema baze podataka 14

10.3 Komponente sistema 15

10.3.1 Komponente korisničkog interfejsa 15

10.3.2 Komponente aplikacione logike 17

10.3.3 Komponente za pristup podacima 19

11. Performanse 20

12. Kvalitet 20

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Eventopia platforme.

# Opseg dokumenta

Dokument se odnosi na Eventopia platformu koja će biti razvijena od strane Average Debugging Enjoyers-a. Eventopia predstavlja platformu za otkrivanje i oglašavanje događaja. Namena sistema je jednostavno pronalaženje, zakazivanje događaja i izdavanje prostora.

# Reference

Spisak korišćene literature:

1. Eventopia – Predlog projekta, ADE-Eventopia-01, V1.0, 2024, Average Debugging Enjoyers.
2. Eventopia – Planirani raspored aktivnosti na projektu, V1.0, 2024, Average Debugging Enjoyers.
3. Eventopia – Plan realizacije projekta, V1.0, 2024, Average Debugging Enjoyers.
4. Eventopia – Vizija sistema, V1.0, 2024, Average Debugging Enjoyers.
5. Eventopia - Specifikacija zahteva, V1.0. 2024. Average Debugging Enjoyers.

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

# Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. Eventopia portal će biti implementiran kao Web aplikacija zasnovana na React biblioteci, ASP.NET framework-u i SQL Server bazi podataka [4].
2. Klijentski deo Eventopia portala će biti optimizovan za sledeće Web čitače: Microsoft Edge, Opera, Firefox (Mozilla), kao i Chrome (Google). [4]
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

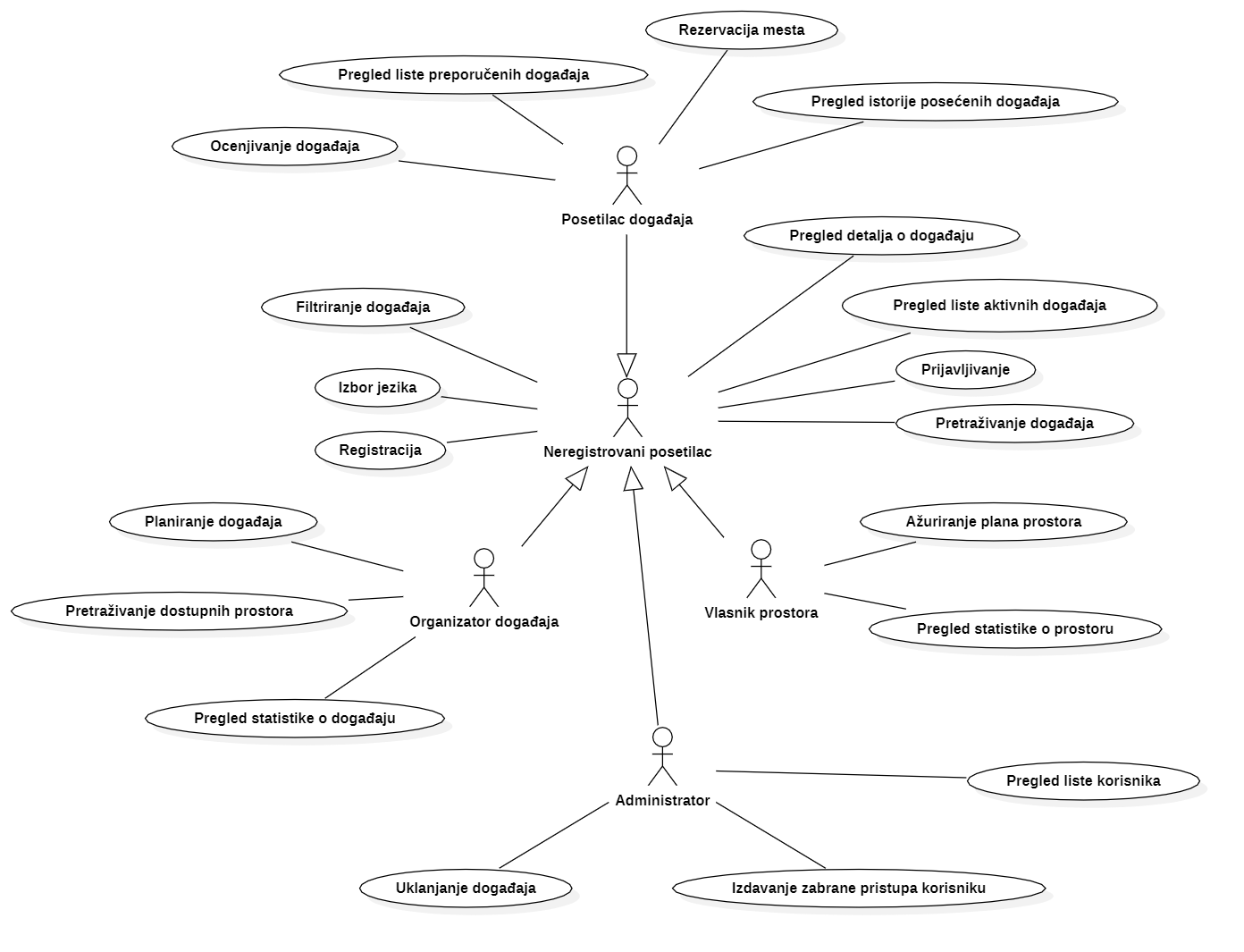
Slučajevi korišćenja Eventopia platforme su:

* Izbor jezika
* Registracija korsnika
* Pregled liste preporučenih događaja
* Filtriranje događaja
* Ocenjivanje događaja
* Unos osnovnih podataka o događaju
* Rezervacija prostora
* Preuređenje plana prostora
* Pretraživanje događaja
* Pretraživanje dostupnih prostora
* Pregled statistike o događaju
* Pregled liste aktivnih događaja
* Prijavljivanje
* Pregled istorije posećenih događaja
* Pregled detalja o događaju
* Rezervacija mesta
* Dodavanje plana prostora
* Modifikacija plana prostora
* Pregled statistike o prostoru
* Uklanjanje događaja
* Izdavanje zabrane pristupa korisniku
* Pregled liste korisnika

Ove slučajevi korišćenja mogu da iniciraju neregistrovani korisnik, posetilac događaja, organizator događaja, vlasnik prostora ili administrator.

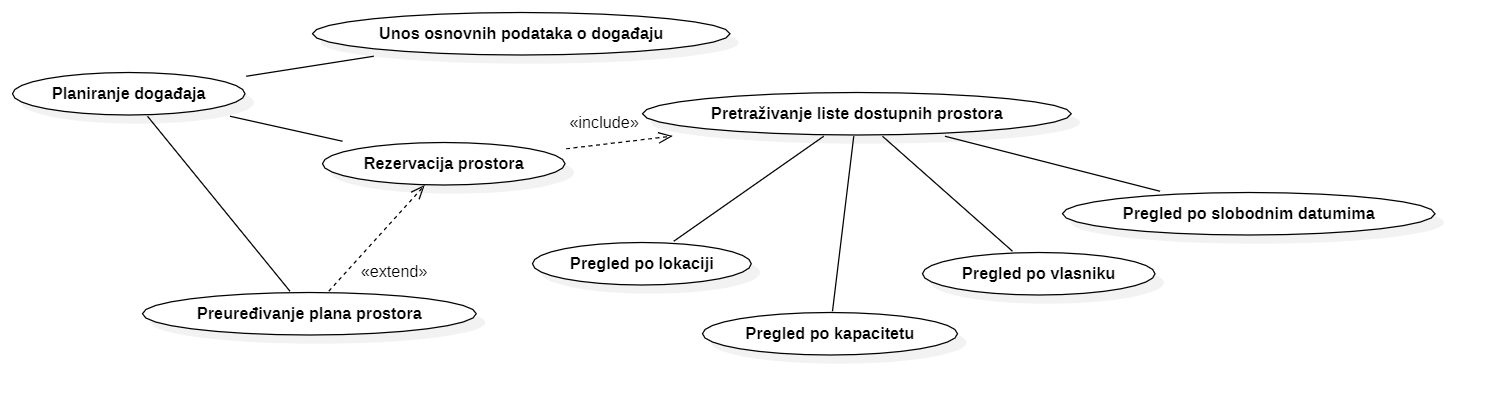
## Dijagrami slučajeva korišćenja

UML dijagram koji prikazuje korisnike i slučajeve korišćenja Eventopia portala prikazan je na sledećoj slici:

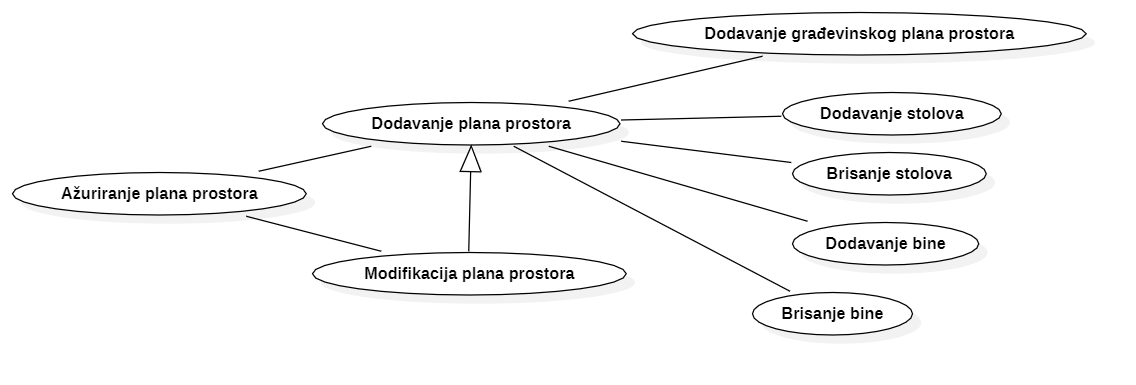


Slučajevi korišćenja *planiranje događaja,* *ažuriranje plana prostora* i *registracija* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

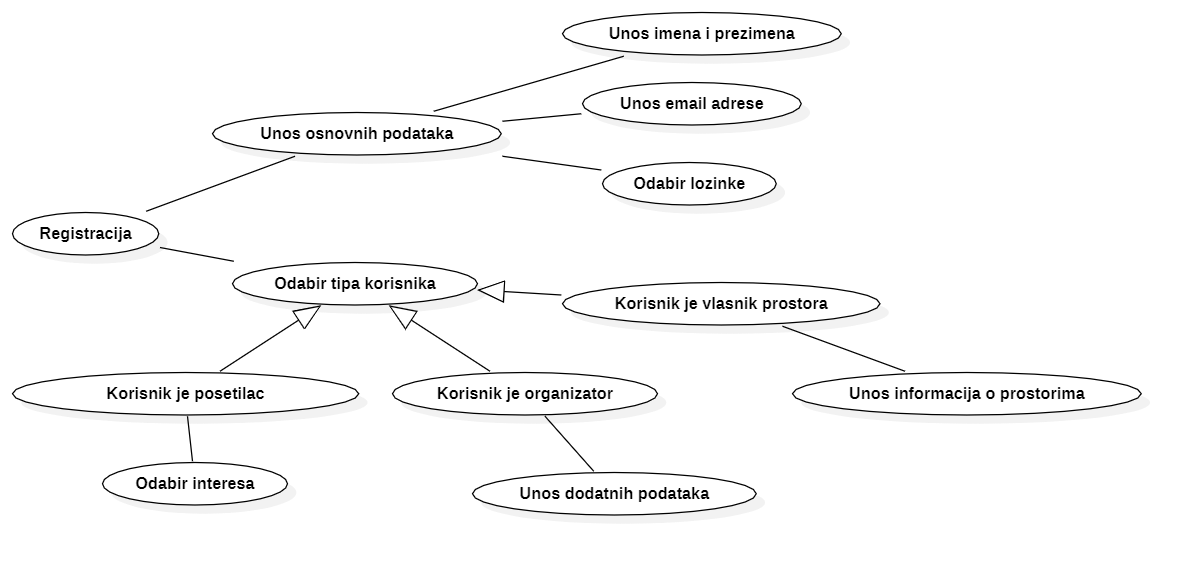
Detaljni UML dijagram za slučaj korišćenja *planiranje događaja* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *ažuriranje plana prostora* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *registracija* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Izbor jezika

Kratak opis: Izbor između srpskog i engleskog jezika za prikaz informacija na portalu.

Akteri koji iniciraju slučaj korišćenja: Neregistrovani posetilac, posetilac događaja, organizator događaja. vlasnik prostora i administrator.

### Registracija

Kratak opis: Registracija novog korisnika.

Akteri koji iniciraju slučaj korišćenja: Neregistrovani posetilac.

### Pregled liste preporučenih događaja

Kratak opis: Prikaz personalizovane liste preporučenih događaja i osnovnim informacijama o njima.

Akteri koji iniciraju slučaj korišćenja: Posetilac događaja.

### Filtriranje događaja

Kratak opis: Prilikom prikaza liste događaja, korisnik ima opciju da stavi filter na tu listu.

Akteri koji iniciraju slučaj korišćenja: Neregistrovani posetilac, posetilac događaja, organizator događaja. vlasnik prostora i administrator.

### Ocenjivanje događaja

Kratak opis: Ukoliko je posetilac događaja rezervisao i prisustvovao događaju koji je završen, korisnik ima pravo da oceni događaj i ostavi komentar.

Akteri koji iniciraju slučaj korišćenja: Posetilac događaja.

### Unos osnovnih podataka o događaju

Kratak opis: Organizator događaja dodaje osnovne podatke o događaju koji organizuje.

Akteri koji iniciraju slučaj korišćenja: Organizator događaja.

### Rezervacija prostora

Kratak opis: Odabir prostora i konkretnog datuma kada će prostor biti korišćen.

Akteri koji iniciraju slučaj korišćenja: Organizator događaja.

### Preuređivanje plana prostora

Kratak opis: Ažuriranje plana prostora od strane organizatora događaja koji rezerviše prostor.

Akteri koji iniciraju slučaj korišćenja: Organizator događaja.

### Pretraživanje događaja

Kratak opis: Pretraživanje događaja uključuje pretraživanje događaja na osnovu njihovih imena.

Akteri koji iniciraju slučaj korišćenja: Neregistrovani posetilac, posetilac događaja, organizator događaja. vlasnik prostora i administrator.

### Pretraživanje dostupnih prostora

Kratak opis: Organizator ima opciju da pretraži prostore unosom željenog kapaciteta i drugih osobina kao što su tip prostorije.

Akteri koji iniciraju slučaj korišćenja: Organizator događaja.

### Pregled statistike o događaju

Kratak opis: Organizator ima opciju da pristupi statističkim podacima, o broju rezervisanih mesta, slobodnih mesta, koliko je posetilaca na kraju prisustvovalo, za događaje koje je on organizovao

Akteri koji iniciraju slučaj korišćenja: Organizator događaja.

### Pregled liste aktivnih događaja

Kratak opis: Prikaz stranice portala sa spiskom svih događaja koji predstoje (aktivnih događaja).

Akteri koji iniciraju slučaj korišćenja: Neregistrovani posetilac, posetilac događaja, organizator događaja. vlasnik prostora i administrator.

### Prijavljivanje

Kratak opis: Prijavljivanje korisnika na portal u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

Akteri koji iniciraju slučaj korišćenja: Neregistrovani posetilac, posetilac događaja, organizator događaja. vlasnik prostora i administrator.

### Pregled istorije posećenih događaja

Kratak opis: Prikaz stranice sa listom svih prethodno posećenih događaja određenog korisnika.

Akteri koji iniciraju slučaj korišćenja: Posetilac događaja.

### Pregled detalja o događaju

Kratak opis: Prikaz stranice sa detaljima o konkretnom događaju.

Akteri koji iniciraju slučaj korišćenja: Neregistrovani posetilac, posetilac događaja, organizator događaja. vlasnik prostora i administrator.

### Rezervacija mesta

Kratak opis: Rezervacija mesta na određenom događaju od strane korisnika.

Akteri koji iniciraju slučaj korišćenja: Posetilac događaja.

### Dodavanje plana prostora

Kratak opis: Dodavanje novog plana prostora od strane vlasnika prostora.

Akteri koji iniciraju slučaj korišćenja: Vlasnik prostora.

### Modifikacija plana prostora

Kratak opis: Ažuriranje plana prostora od strane vlasnika tog prostora.

Akteri koji iniciraju slučaj korišćenja: Vlasnik prostora.

### Pregled statistike o prostoru

Kratak opis: Prikaz stranice sa statistikom o konkretnom prostoru.

Akteri koji iniciraju slučaj korišćenja: Vlasnik prostora.

### Uklanjanje događaja

Kratak opis: Administrator ima mogućnost uklanjanja događaja iz sistema Eventopia.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Izdavanje zabrane pristupa korisniku

Kratak opis: Administrator ima mogućnost zabrane pristupa bilo koje ostale vrste korisnika koji su registrovani u sistemu.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Pregled liste korisnika

Kratak opis: Administrator događaja ima mogućnost pregleda liste svih registrovanih korisnika u sistemu Eventopia.

Akteri koji iniciraju slučaj korišćenja: Administrator.

# Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

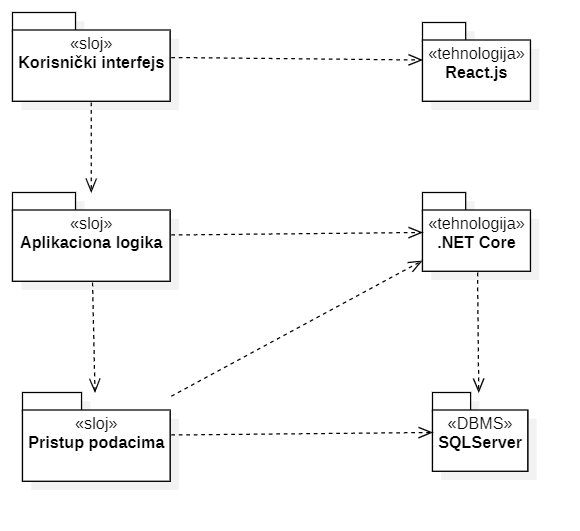
Logička arhitektura Eventopia portala obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika i Pristup podacima.

Paket *Korisnički interfejs* sadrži React komponente, osnovnu HTML stranicu, TS skripte i multimedijalni sadržaj kojim se realizuje grafički dizajn i forme preko kojih korisnici interaguju sa sistemom.

Paket *Aplikaciona logika* čini srednji sloj sistema koji sadrži C# .NET fajlove koji implementiraju poslovnu logiku domena primene i preko koga se vrši komunikacija sa bazom podataka.

Paket *Pristup podacima* sadrži C# .NET fajlove kojima se implementira model domena primene, uključujući i odgovarajuća EF mapiranja.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs portala. U njemu su sadržane svi HTML i CSS fajlovi, multimedijalni sadržaji i React.js komponente koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i biblioteke React.js.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Eventopia portala. Sadrži .NET Core kontrolere koji realizuju funkcionalnost karakterističnu za domen primene portala i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i .NET Core framework-a.

### React.js

tehnologija

React.js tehnologija omogućava kreiranje skripti koje se izvršavaju na klijentskoj strani, omogućujući dinamično generisanje HTML elemenata i upravljanje događajima tih elemenata.

### .NET Core

tehnologija

Tehnologija .NET Core se izvršava na serverskoj strani, obezbeđuje mehanizam za pisanje modela i kontrolera. Model je veza između baze podataka i aplikacione logike, a kontroleri predstavljaju samu aplikacionu logiku.

### DBMS

tehnologija

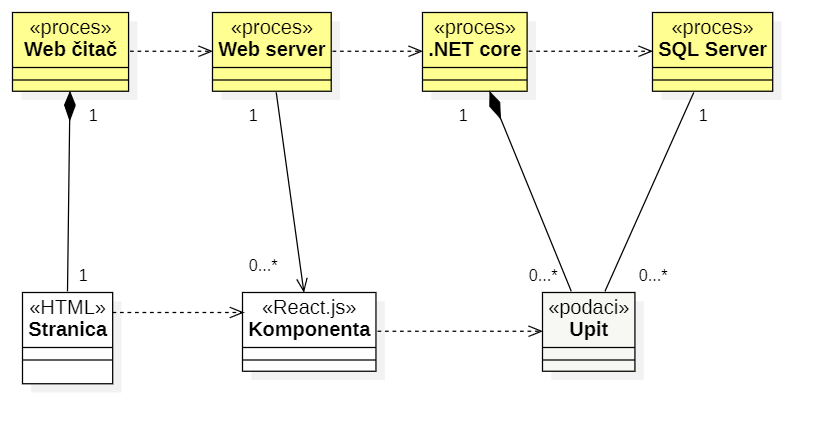
SQL Server je sistem za upravljanje bazama podataka koji će biti korišćen za implementaciju Eventopia web aplikacije.

# Pogled na procese

U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe su dati dijagrami koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Eventopia sistema.



### Web čitač

Web čitač je proces koji je zadužen za prikaz HTML stranice dobijene od nekog Web servera. U najopštijem slučaju Web pretraživač može da prikazuje samo jednu HTML stranicu u jednom pokrenutom procesu. U slučaju SPA aplikacije, veći broj “stranica” se dostavlja Web čitaču, ali se kolekcija komponenata koje se prikazuju određuje dinamički.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja klijentskih zahteva. U slučaju SPA aplikacija pisanih u React.js, Web server kroz prvi zahtev vraća HTML stranicu, kao i bundle svih pratećih JS i CSS fajlova, dok svi zahtevi koji proisteknu kao posledica interakcije korisnika sa Web stranicom se prosleđuju .NET Core kontrolerima.

### .NET core

.NET core je proces koji radi u sadejstvu sa Web serverom. Web server prosleđuje HTTP zahtev .NET core procesu, koji te zahteve obrađuje i vraća podatke klijentu u JSON formatu.

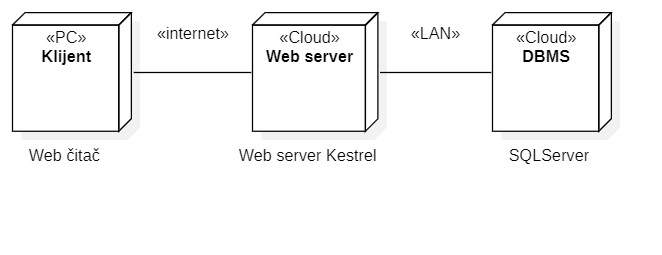
### SQL Server

SQL Server je proces koji pruža funkcionalnost sistema za upravljanje relacionim bazama podataka. Ovaj proces je sposoban da konkurentno prihvati određeni broj upita, izvrši ih nad bazom podataka i vrati rezultate korisnicima koji su postavili te upite.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja Eventopia portala.



## Klijent

Pristup Eventopia portalu se obavlja preko klijentskih računara na kojima se izvršava Web čitač. Za povezivanje između klijenata i Web servera se koristi Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

* 1. **Web server**

Web server opslužuje više klijenata koji pristupaju preko Interneta, i izvršava se na Cloud-u (računaru u oblaku). Pored osnovnog procesa koji realizuje funkcionalnost Web servera (Kestrel), na ovom računaru se izvršavaju i ASP.NET Core kontroleri koji vrše obradu pristiglih zahteva.

* 1. **DBMS server**

DBMS server se izvršava na Cloud-u (računaru u oblaku) i to je proces za upravljanje bazama podataka. Radi sigurnosti podataka, pristup bazi je ograničen samo na računare unutar lokalne mreže (LAN), što osigurava zaštitu od neovlašćenog pristupa.

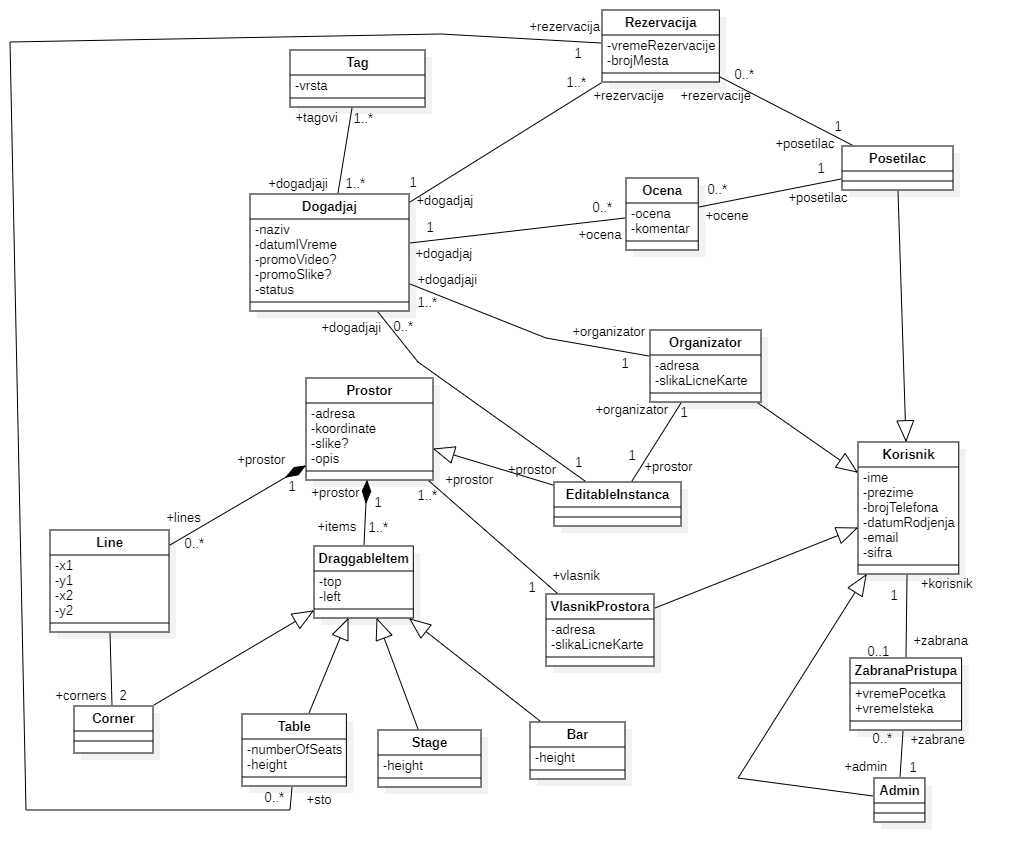
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju Eventopia portala ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

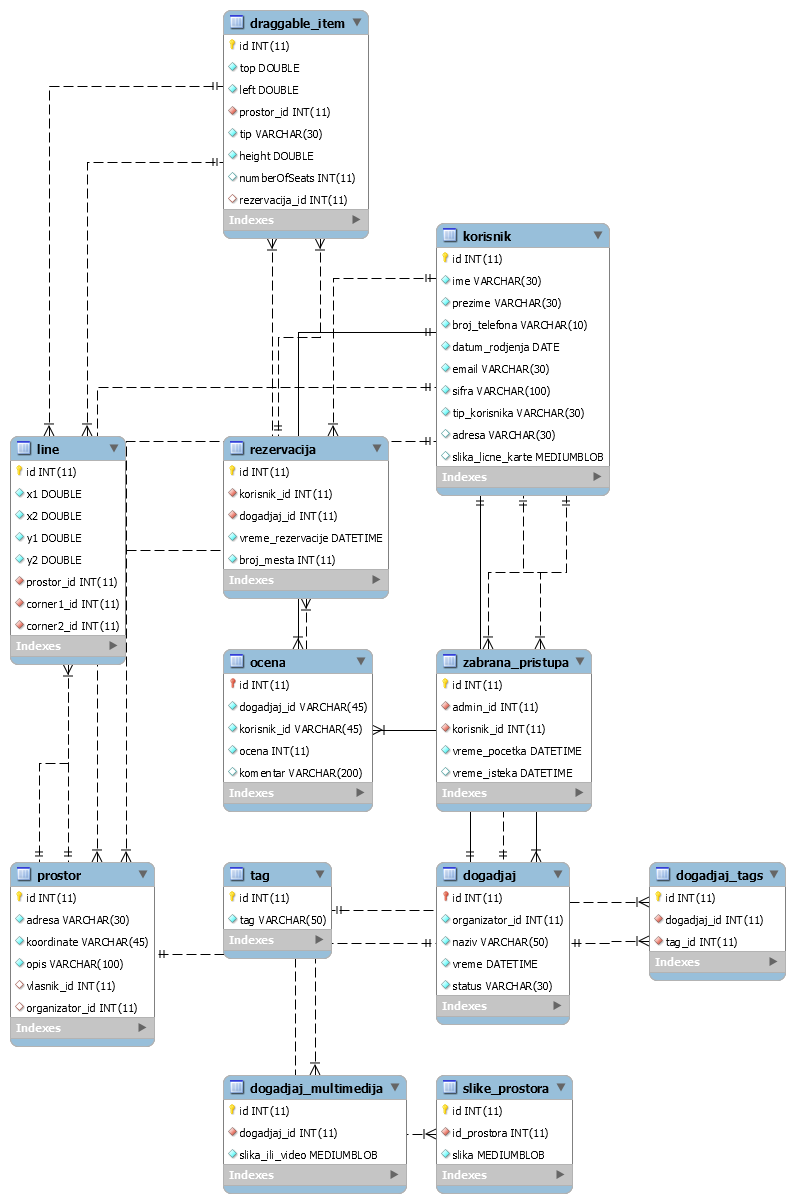
Model domena za koji se Eventopia portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (.NET kontrolera) koje će biti implementirane.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *MySQL Workbench-a*.

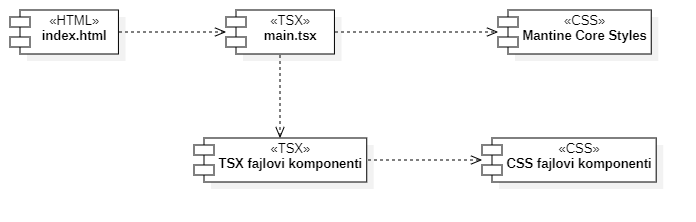


## Komponente sistema

Komponente sistema Eventopia platforme su React.js skripte i .NET fajlovi čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti i dijagrami klasa.

### Komponente korisničkog interfejsa

Globalno uređenje komponenti korisničkog interfejsa Eventopia platforme predstavljeno je sledećim komponentama:



Komponenta **index.html** je osnovna HTML stranica koju Web server dostavlja korisniku prilikom pristupa platformi. Ona učitava bundle pratećih CSS i JS fajlova.

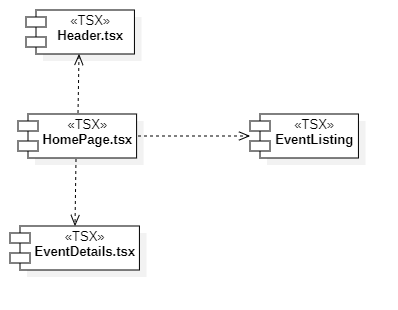
Komponenta **main.tsx** predstavlja ulaznu tačku za JS fajlove, koja takođe pruža Provider usluge svim ugnježdenim komponentama, i koja učitava sve globalne CSS stilove.

**Mantine Core Styles** predstavlja kolekciju globalnih CSS stilova i dodatnih React.js komponenti, koje mogu biti korišćene u korisničkim komponentama.

**TSX fajl** predstavlja implementaciju konkretne React komponente.

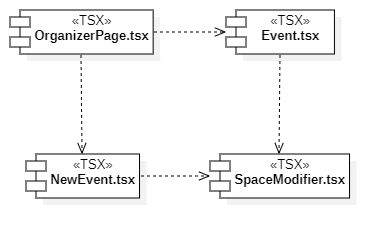
**CSS fajl** predstavlja prateći stil za svaku od React komponenti.

Dizajn korisničkog interfejsa za *posetioca* Eventopia platforme je obuhvaćen sledećim komponentama



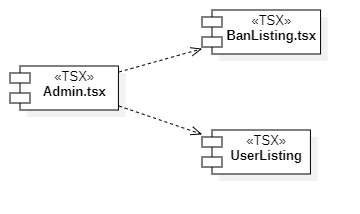
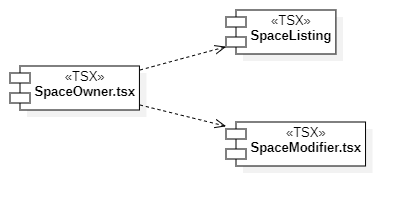
* Komponenta **HomePage.tsx** predstavlja osnovnu stranicu, koja se prikazuje bilo kom tipu korisnika kada pristupi Eventopia platformi. Preko ove komponente korisnik se može prijaviti na platformu, pregledati promo materijal ili vršiti osnovnu pretragu događaja.
* Komponenta **Header.tsx** predstavlja navigacioni panel koji pruža mogućnosti prijavljivanja i registrovanja na platformu i pristup drugim sekcijama sajta.
* Komponenta **EventListing.tsx** je komponenta koja prikazuje rezultate filtriranja.
* Komponenta **EventDetails.tsx** neregistrovanom korisniku prikazuje detalje o određenom događaju, dok registrovanom posetiocu nudi mogućnost pregleda dostupnih mesta i rezervacije.

Dizajn korisničkog interfejsa koji je specifičan za *organizatora događaja* je obuhvaćen sledećim komponentama:



* Komponenta **OrganizerPage.tsx** predstavlja komponentu koja je prikazana organizatoru događaja prilikom prijavljivanja na platformu. Obuhvata osnovne informacije koje se prikazuju.
* Komponenta **NewEvent.tsx** sačinjena je od niza formi koje organizator događaja popunjava prilikom organizovanja novog događaja.
* Komponenta **Event.tsx** pruža organizatoru mogućnost pregleda svih događaja koje je organizovao/organizuje, kao i osnovne statističke detalje.
* Komponenta **SpaceModifier.tsx** je prost grafički editor koji organizatoru pruža mogućnost preuređivanja izabranog prostora prema svojim potrebama, uz određenja ograničenja.

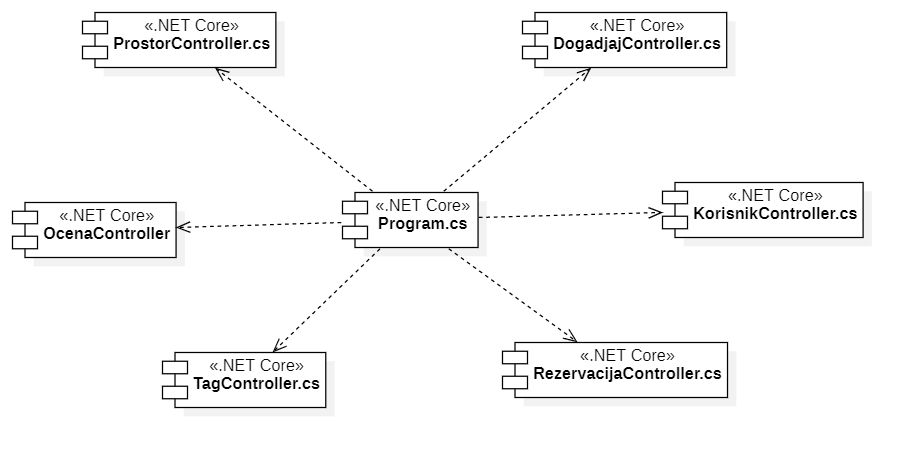
Dizajn korisničkog interfejsa koji je specifičan za *vlasnika prostora* i *administratora* je obuhvaćen sledećim komponentama.

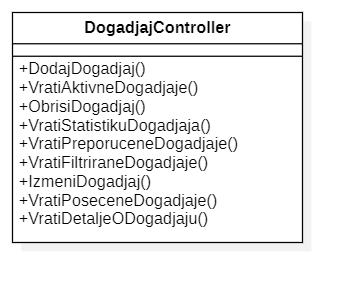


* Komponenta **SpaceOwner.tsx** predstavlja komponentu koja je prikazana vlasniku prostora prilikom prijavljivanja na platformu. Obuhvata osnovne informacije koje se prikazuju.
* Komponenta **SpaceListing.tsx** je komponenta koja prikazuje listu svih prostora koje je vlasnik dodao kao moguće za iznajmljivanje, i prikazuje osnovnu statistiku.
* Komponenta **SpaceModifier.tsx** je prost grafički editor koji organizatoru pruža mogućnost preuređivanja prostora, dodavanja i uklanjanja objekata.
* Komponenta **Admin.tsx** predstavlja komponentu koja je prikazana administratoru prostora prilikom prijavljivanja na platformu. Obuhvata osnovne informacije koje se prikazuju.
* Komponenta **BanListings.tsx** predstavlja komponentu koja obuhvata listu svih izdatih zabrana pristupa korisnicima, uz mogućnost opozivanja zabrane pristupa.
* Komponenta **UserListing.tsx** administratoru pruža uvid u listu svih korisnika

### Komponente aplikacione logike

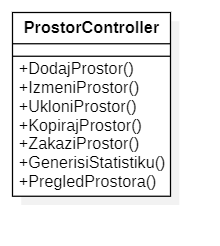
Komponente koje realizuju domen problema predstavljaju Controller klase u .NET Core framework-u. Uključuju se u aplikaciju preko Program.cs fajla. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:





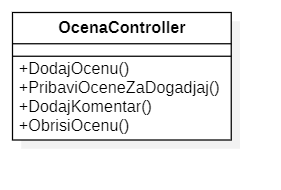
Navedene metode iz kontrolera obavljaju sledeće zadatke:

* DodajDogadjaj - dodaje novi događaj
* VratiAktivneDogadjaje - vraća listu svih događaja koji se dešavaju u budućnosti
* ObrisiDogadjaj - brise odgovarajući događaj
* VratiStatistikuDogadjaja - generiše osnovnu statistiku o događaju, uključujući broj rezervisanih mesta, prosečnu ocenu i sl.
* VratiPreporuceneDogadjaje - metoda kontrolera koja za svakog korisnika vraća događaje za koje postoji velika verovatnoća da će interesovati korisnika, koristeći state of the art recommendation engine
* VratiFiltriraneDogadjaje - vraća listu događaja koji zadovoljavaju kriterijum filtiranja
* IzmeniDogadjaj - ažurira informacije o događaju
* VratiPoseceneDogadjaje - za odgovarajućeg korisnika vraća spisak događaja koje je posetio
* VratiDetaljeODogadjaju - vraća informacije o jednom događaju



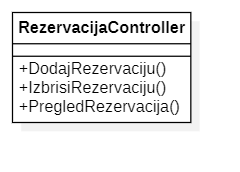
Navedene metode iz kontrolera obavljaju sledeće zadatke:

* DodajProstor - dodaje novi prostor u bazu podataka
* IzmeniProstor - menja podatke o prostoru i planu prostora u bazi podataka
* UkloniProstor - briše prostor iz baze podataka
* KopirajProstor - kada organizator događaja vrši izmenu plana prostora, ta instanca prostora se pamti nezavisno u odnosu na originalni plan prostora koji pripada vlasniku prostora
* ZakaziProstor - organizator događaja vrši rezervaciju prostora
* GenerisiStatistiku - vraća statistiku o prostoru
* PregledProstora - vraća podatke koje su potrebni za generisanje plana prostora na klijentskoj strani aplikacije, kao i vremena kada je prostor dostupan



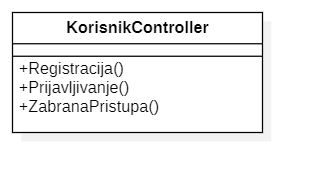
Navedene metode iz kontrolera obavljaju sledeće zadatke:

* DodajOcenu - dodaje novu ocenu za događaj koji je završen
* PribaviOceneZaDogadjaj - vraća spisak svih ocena i komentara vezanih za određeni dogadjaj, kao i prosečnu ocenu
* DodajKomentar - dodaje komentar uz odgovarajuću ocenu
* ObrisiOcenu - briše se ocena vezana za odgovarajući događaj od strane korisnika, ili nasilno od strane administratora



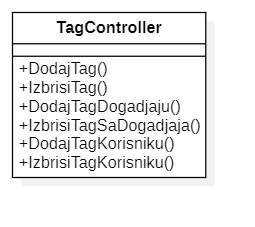
Navedene metode iz kontrolera obavljaju sledeće zadatke:

* DodajRezervaciju - dodaje novu rezervaciju u bazu podataka
* IzbrisiRezervaciju - briše rezervaciju iz baze podataka
* PregledRezervacija - vraća sve rezervacije za određeni događaj koji se na klijentskoj strani prikazuju organizatoru događaja



Navedene metode iz kontrolera obavljaju sledeće zadatke:

* Registracija - vrši odgovarajuću validaciju i registruje novog korisnika određenog tipa
* Prijavljivanje - proverava unete kredencijale i prijavljuje korisnika na platformu
* ZabranaPristupa - dodeljuje novu zabranu pristupa korisniku, čime mu se zabranjuje pristup platformi

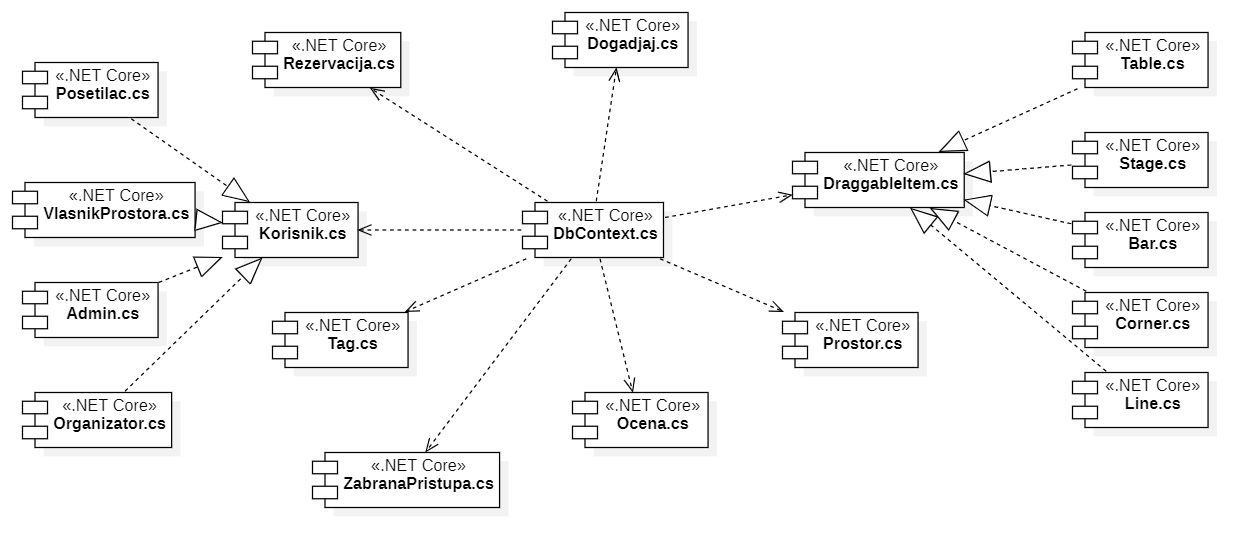


Navedene metode iz kontrolera obavljaju sledeće zadatke:

* DodajTag - dodaje novi tag u bazu podataka
* IzbrisiTag - briše tag iz baze podataka
* DodajTagDogadjaju - dodeljuje određeni tag navedenom događaju
* IzbrisiTagSaDogadjaja - briše tag sa događaja
* DodajTagKorisniku - dodeljuje određeni tag korisničkom profilu
* IzbrisiTagKorisniku - briše tag sa korisnikovog profila

### Komponente za pristup podacima

Pristup bazi podataka ostvaruje se korišćenjem objektno-relacionog mapera Entity Framework, što znači da nije potrebno eksplicitno kreirati funkcije za pristup i pretraživanje podataka, već se sam pristup podacima ostvaruje kroz kontrolere. Na sledećem dijagramu predstavljene su komponente koje čine model, a za čije preslikavanje na bazu podataka je zadužen Entity Framework.



# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu specificirane u zahtevima u pogledu performansi [5]:

1. Sistem će da podrži do 2000 simultanih pristupa korisnika portalu.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme [5].

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [5]:

1. Eventopia porotal će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 30%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 96 sati.