Ball Ball

Web aplikacija za zakazivanje termina

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 16.04.2024. | 1.0 | Inicijalna verzija | Svetomir |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 8

6.2.1 Registacija 8

6.2.2 Prijavljivanje 8

6.2.3 Ažuriranje podataka o sebi 8

6.2.4 Ocenjivanje drugih igrača 8

6.2.5 Formiranje timova 8

6.2.6 Pregled slobodnih termina 8

6.2.7 Rezervacija termina 8

6.2.8 Kreiranje termina za sale 8

6.2.9 Rezervacija termina za sale 8

6.2.10 Brisanje korisnika 9

6.2.11 Pregled liste korisnika 9

6.2.12 Pregled terena 9

6.2.13 Kreiranje novog termina 9

6.2.14 Ažuriranje statusa termina 9

6.2.15 Lični opis igrača 9

6.2.16 Ocenjivanje igrača od strane drugih korisnika 9

6.2.17 Rangiranje timova na osnovu uspeha 9

6.2.18 Dodavanje igrača u tim 9

6.2.19 Komunikacija između timova putem chat-a 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 Pristup podacima 10

7.1.4 Angular 10

7.1.5 .NET Core 11

7.1.6 PostgreSQL 11

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 PostgreSQL baza podataka 11

9. Pogled na raspoređivanje sistema 11

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 12

10. Pogled na implementaciju sistema 12

10.1 Model domena 12

10.2 Šema baze podataka 13

10.3 Komponente sistema 14

10.3.1 Komponente korisničkog interfejsa 14

10.3.2 Komponente aplikacione logike 14

10.3.3 Komponente za pristup podacima 15

11. Performanse 15

12. Kvalitet 15

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Ball Ball web aplikacije.

# Opseg dokumenta

Dokument se odnosi na Ball Ball aplikaciju koja će biti razvijen od strane Bušido tima. Ball Ball će korisnicima omogućiti brzo i lako zakazivanje termina za sportske događaje.

# Reference

Spisak korišćene literature:

1. Ball Ball – Predlog projekta, SWE-BallBall-01, V1.0, 2024, Bušido.
2. Ball Ball – Planirani raspored aktivnosti na projektu, V1.0, 2024, Bušido.
3. Ball Ball – Plan realizacije projekta, V1.0, 2024, Bušido.
4. Ball Ball – Vizija sistema, V1.0, 2024, Bušido.
5. Ball Ball – Specifikacija zahteva, V1.0, 2024, Bušido.

# 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. Klijent-server model komunikacije koji podrazumeva postojanje centralizovanog servera preko kog se vrši razmena poruka između učesnika [4].
2. Klijentski deo Ball Ball aplikacije će biti optimizovan za sledeće Web čitače: Microsoft Edge, Google Chrome, Opera kao i Mozilla Firefox. [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].

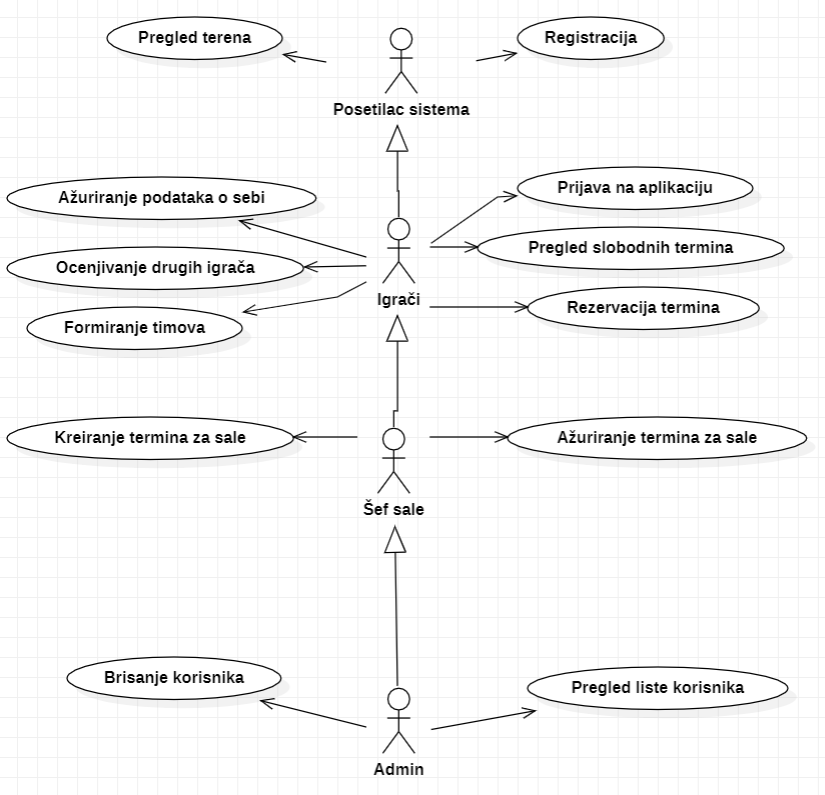
Slučajevi korišćenja Ball Ball aplikacije su:

* Registacija
* Prijavljivanje
* Ažuriranje podataka o sebi
* Ocenjivanje drugih igrača
* Formiranje timova
* Pregled slobodnih termina
* Rezervacija termina
* Kreiranje termina za sale
* Rezervacija termina za sale
* Brisanje korisnika
* Pregled liste korisnika
* Pregled terena
* Kreiranje novog termina
* Ažuriranje statusa termina
* Lični opis igrača
* Ocenjivanje igrača od strane drugih korisnika
* Rangiranje timova na osnovu uspeha
* Dodavanje igrača u tim
* Komunikacija između timova putem chat-a

Ove slučajevi korišćenja mogu da iniciraju posetilac sistema, igrači, šef sale ili administrator.

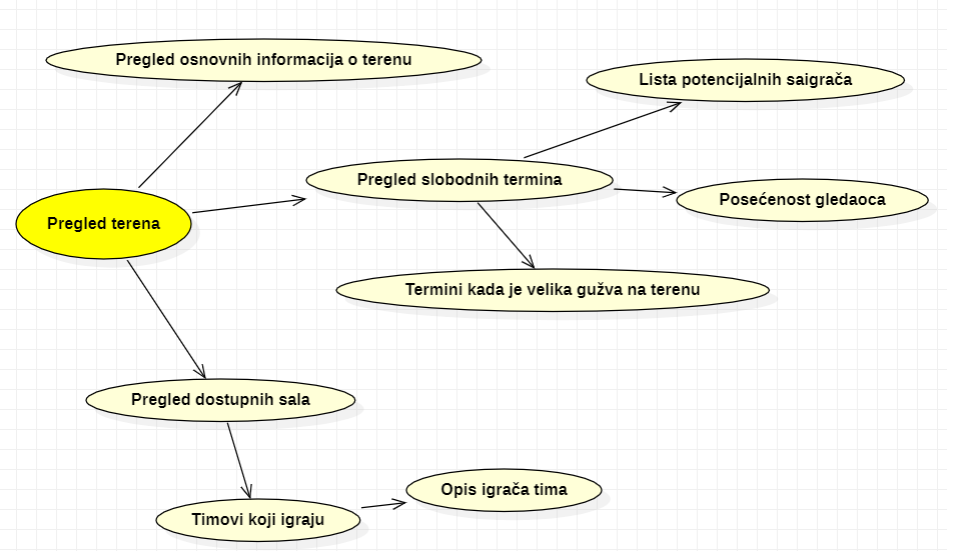
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Ball Ball aplikacije prikazan je na sledećoj slici:

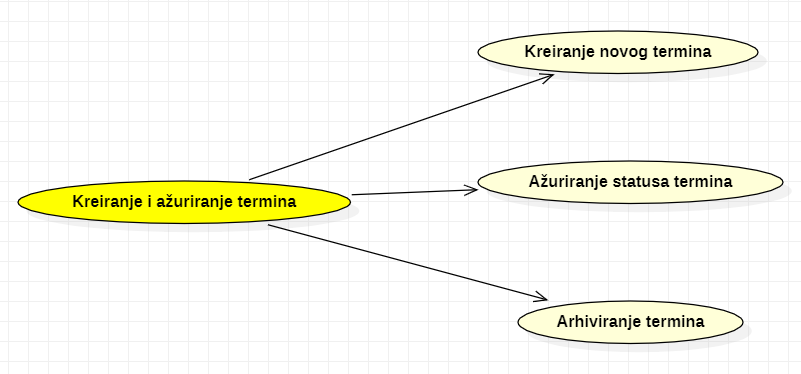


Slučajevi korišćenja *pregled terena* i *kreiranje i ažuriranje termina* i *ažuriranje podataka o igračima* i *pregled timova* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

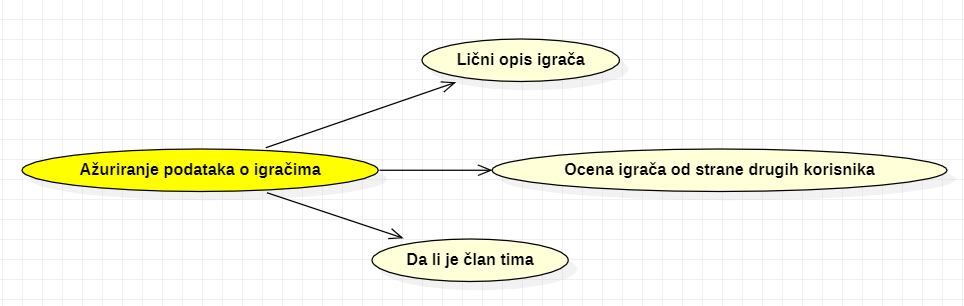
Detaljni UML dijagram za slučaj korišćenja *pregled terena* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *kreiranje i ažuriranje termina* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *ažuriranje podataka o igračima* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Registracija

Kratak opis: Dodavanje novog korisnika u bazu aplikacije.

Akteri koji iniciraju slučaj korišćenja: Posetilac aplikacije.

### Prijavljivanje

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

Akteri koji iniciraju slučaj korišćenja: Igrači, menadžeri sale, adminsitrator.

### Ažuriranje podataka o sebi

Kratak opis: Ažuriranje podataka o sebi od strane prijavljenog korisnika

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Ocenjivanje drugih igrača

Kratak opis: Mogućnost ostavljanja ličnog komentara o igraču.

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Formiranje timova

Kratak opis: Spajanje igrača u timove da bi se takmičili s drugim timovima.

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Pregled slobodnih termina

Kratak opis: Korisnik može izabrati njemu odgovarajući termin za teren.

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Rezervacija termina

Kratak opis: Korisnik rezerviše svoj termin na terenu.

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Kreiranje termina za sale

Kratak opis: Menadžer sale kreira termin u kome igrači mogu da iznajme salu.

Akteri koji iniciraju slučaj korišćenja: Menadžer sale.

### Rezervacija termina za sale

Kratak opis: Korisnici šalju zahteve menadžeru koje će menadžer odobriti/odbiti.

Akteri koji iniciraju slučaj korišćenja: Igrači, menadžeri sale

### Brisanje korisnika

Kratak opis: Administator aplikacije može po potrebi obrisati korisnika iz baze.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Pregled liste korisnika

Kratak opis: Prikaz svih korisnika aplikacije.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Pregled terena

Kratak opis: Prikaz informacija o terminima i terenima.

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Kreiranje novog termina

Kratak opis: Dodaje se novi slobodan termin u listu slobodnih termina.

Akteri koji iniciraju slučaj korišćenja: Menadžer sale.

### Ažuriranje statusa termina

Kratak opis: Menadžer može promeniti informacije o terminima.

Akteri koji iniciraju slučaj korišćenja: Menadžer sale.

### Lični opis igrača

Kratak opis: Korisnik može dodati par osnovnih informacija o sebi.

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Ocena igrača od strane drugih korisnika

Kratak opis: Igrači se mogu međusobno ocenjivati na osnovu odigranih partija.

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Rangiranje timova na osnovu uspeha

Kratak opis: Formira se rang lista timova.

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Dodavanje igrača u tim

Kratak opis: Igrač se može dodati u tim.

Akteri koji iniciraju slučaj korišćenja: Igrači.

### Komunikacija između timova putem chat-a

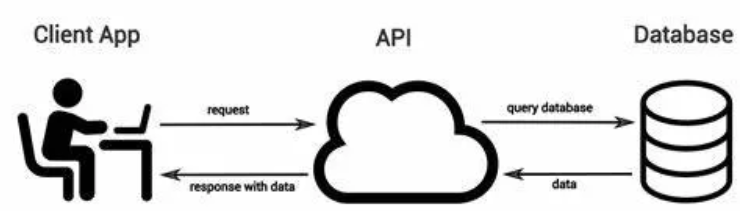
Kratak opis: Jedinstveni chat za dogovor izmedju timova, uživo komunikacija.

Akteri koji iniciraju slučaj korišćenja: Igrači

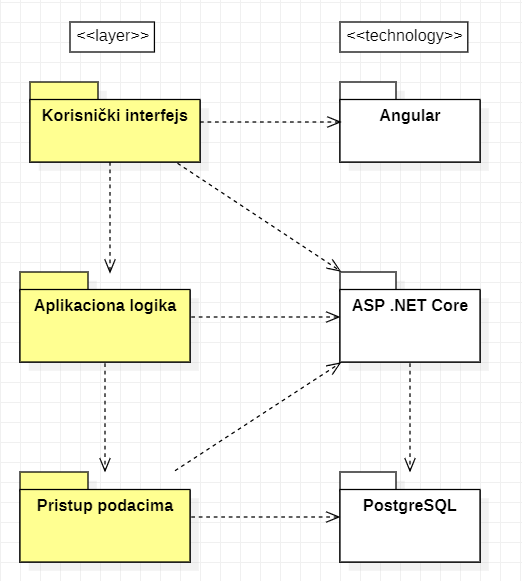
# Pogled na logičku arhitekturu sistema

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.

Logički pogled na Ball Ball aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.



## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

Ovaj sloj realizuje korisnički interfejs aplikacije. Koristeći Angular tehnologije pravimo single-page application (SPA) koja učitava samo jedan web dokument, na kome se dinamički generiše sadržaj stranice.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa Angular i .NET Core.

### Aplikaciona logika

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Ball Ball aplikacije. Koristi .NET Framework koji realizuju funkcionalnost karakterističnu za domen aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Angular u kombinaciji sa .NET-om pruža dobru osnovu za izradu modernih i povezanih aplikacija.

### Pristup podacima

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži moćan objektno-orijentisan model zadužen za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u PostgreSQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od .NET Core i PostgreSQL baza podataka.

### Angular

Angular je zasnovan na TypeScript-u, open-source framework za razvoj web aplikacija. Razvijen od strane Google-a i služi za izradu dinamičkih, jednostraničnih aplikacija (SPA) i progresivnih aplikacija (PWA).

### .NET Core

Tehnologija koja se koristi za razvoj različitih vrsta aplikacija kao što su: web aplikacije, mikroservici, API-ji, desktop aplikacije i mobilne aplikacije. Odlukuje ga podrška za različite platforme kao i modularnost.

### PostgreSQL

Podržava objektno-relacioni model, pruža transakcijsku podršku, višekorisničku podršku, proširivost i može obraditi velike količine podataka sa visokom brzinom, i kao takav popularan je izbor za projekte koji zahtevaju pouzdanu, skalabilnu i prilagodljivu bazu podataka.

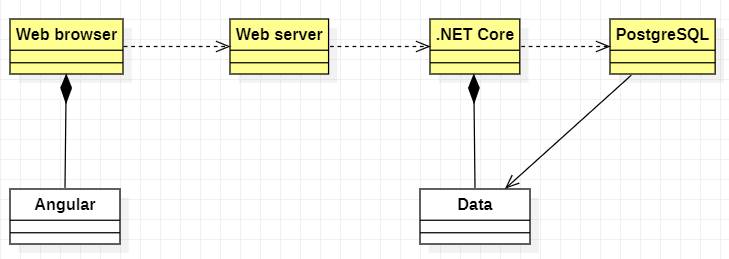
# Pogled na procese

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Ball Ball aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na .NET-u i PostgreSQL bazi podataka.

### Web čitač

Korisnik interaguje sa web aplikacijom putem web čitača, pokrećući HTTP zahteve ka serveru.



### Web server

Web server prima HTTP zahteve od web čitača. Zatim obrađuje dolazne zahteve i prosleđuje ih odgovarajućim kontrolerima u .NET Core backend-u.

### .NET Core

Backend komponente obrađuju HTTP zahteve koje primi web server.

To uključuje izvršavanje poslovne logike aplikacije, pristupanje bazi podataka, autentifikaciju, autorizaciju i slično.

Nakon obrade zahteva, backend generiše odgovor koji se šalje nazad web serveru.

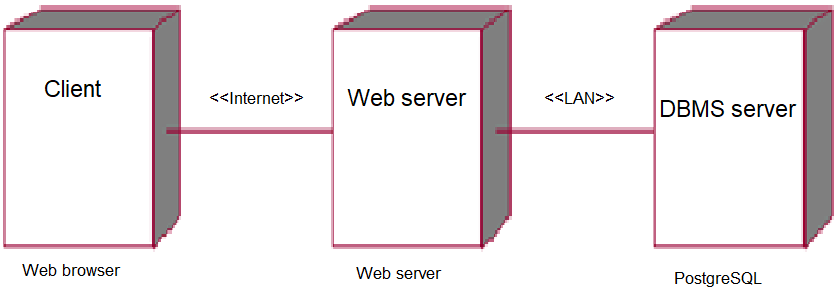
### PostgreSQL baza podataka

.NET Core backend komponente zavise od baze podataka kako bi izvršile operacije čitanja i upisa podataka.

# 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 Ball Ball aplikacije.



## Klijent

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

## Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, prima HTTP zahteve od web čitača, prosleđuje ih backend-u na obradu, a zatim šalje odgovore nazad web čitaču. Ovo omogućava jasno definisanu komunikaciju između klijentskog (frontend) i serverskog (backend) dela vaše web aplikacije. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

## DBMS server

DBMS server je računar na kome se izvršava PostgreSQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

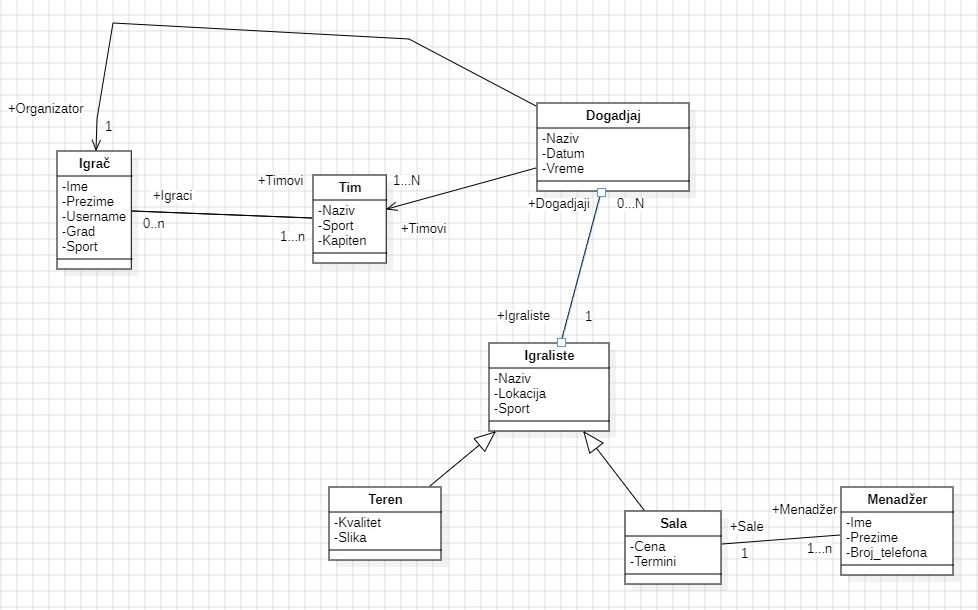
# Pogled na implementaciju sistema

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

## Model domena

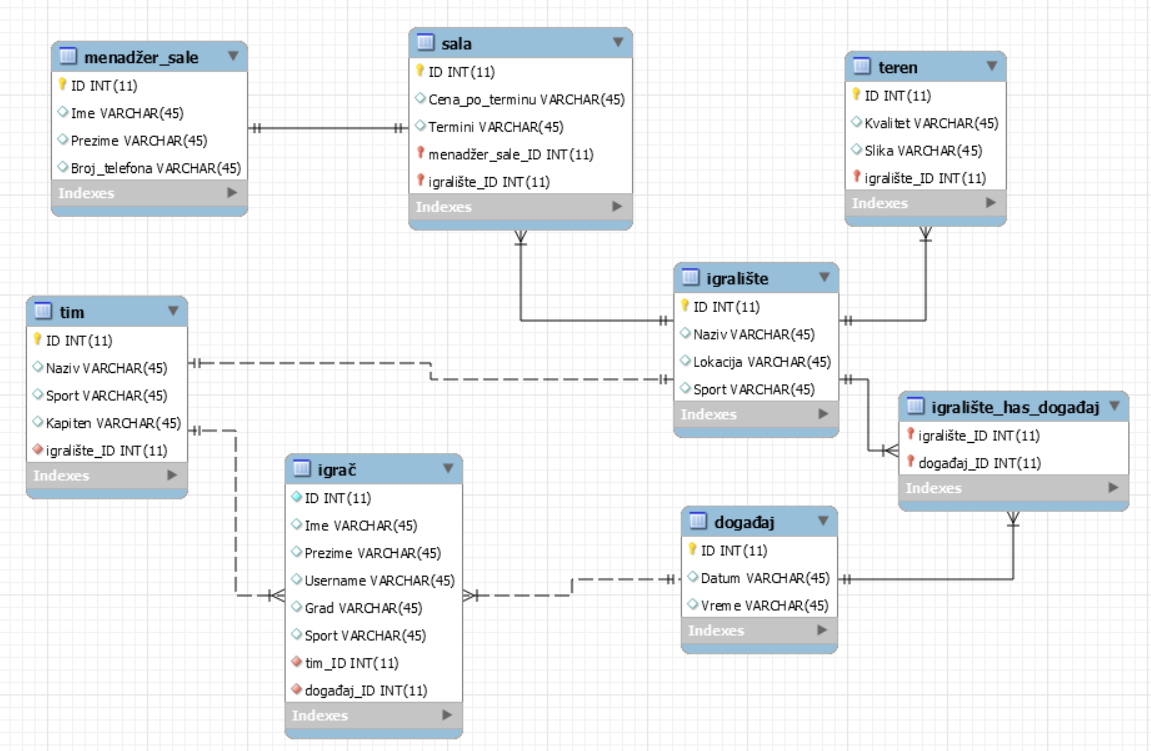
Model domena za koji se Ball Ball projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka.



## 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 korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

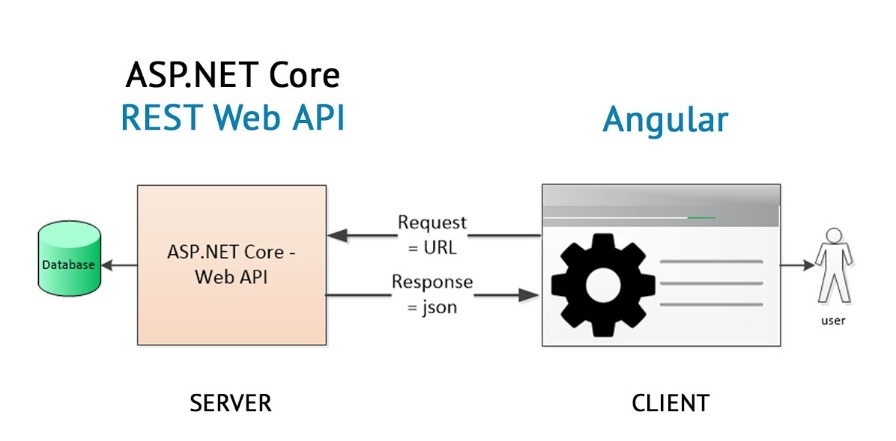


Komponenta **angular.ts** je implementira stranicu aplikacije čiji sadržaj može da se dinamički menja.

Komponenta **.NET Core** predstavlja skripte koje se koriste za povezivanje sa .NET Frameworkom za rad sa podacima u bazi.

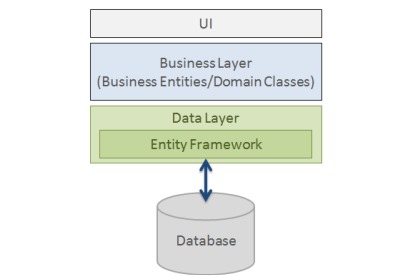
### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **angular.ts** komponente korisničkog interfejsa. Preko **.NET**-a se zahtev prosleđuje bazi i predstavlja vezu između krajnjeg korinsika i same baze podataka iz koje se podaci šalju korisniku.



### Komponente za pristup podacima

Pristup bazi podataka vrši se kreiranjem odgovarajućih upita koji se uz pomoć .NET-a šalju bazi. PostgreSQL baza podržava objektno-orijentisani model i brzo radi sa velikom količinom podataka.



# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi [5]:

1. Sistem će da podrži do 1000 simultanih pristupa korisnika portalu.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 sekundi.

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. Ball Ball aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.