GymBro

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 27.04.2020. | 1.0 | Inicijalna verzija | Mladen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 Registracija 8

6.2.2 Pregled osnovnih podataka 8

6.2.3 Dodavanje novog događaja 8

6.2.4 Prikaz podataka o kreiranom događaju 8

6.2.5 Prikaz podataka o konkretnom događaju 8

6.2.6 Brisanje podataka o završenom događaju 8

6.2.7 Otkazivanje događaja 8

6.2.8 Prijavljivanje na dogadjaj 8

6.2.9 Brisanje postojećih korisnika 8

6.2.10 Unos podataka o korisniku 9

6.2.11 Ažuriranje podataka o korisniku 9

6.2.12 Prikaz podataka o korisniku 9

6.2.13 Unos podataka o sportskim objektima 9

6.2.14 Ažuriranje podataka o sportskim objektima 9

6.2.15 Prikaz podataka o sportskim objektima 9

6.2.16 Mapirani prikaz sportskih objekata i događaja 9

6.2.17 Sistem ocenjivanja sportskih objekata 9

6.2.18 Sistem ocenjivanja drugih korisnika 9

6.2.19 Prijavljivanje na sistem 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 HTML 10

7.1.5 PHP 11

7.1.6 MySQL 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 PHP-CGI 11

8.1.4 MySQL Server 12

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 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 16

12. Kvalitet 16

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture GymBro portala.

# Opseg dokumenta

Dokument se odnosi na GymBro portal koji će biti razvijen od strane Bilingvalaca. Omogućiti socijalizaciju I lakše povezivanje ljudi koji žele zdrav život, fizičku aktivnost i sticanje novih znanja u oblasti fitnesa

# Reference

Spisak korišćene literature:

1. GymBro – Predlog projekta, V1.0, Bilingvalci.
2. GymBro – Planirani raspored aktivnosti na projektu, V1.0, 2020, Bilingvalci.
3. GymBro – Plan realizacije projekta, V1.0, 2020, Bilingvalci.
4. GymBro – Vizija sistema, V1.0, 2020, Bilingvalci.
5. GymBro – Specifikacija zahteva, V1.0, 2020, Bilingvalci.

# 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. GymBro portal će biti implementiran kao Web aplikacija zasnovana na .NET entity framework i MySQL bazi podataka.
2. Klijentski deo GymBro portala će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, kao i Firefox (Mozilla), Google Chrome.
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].

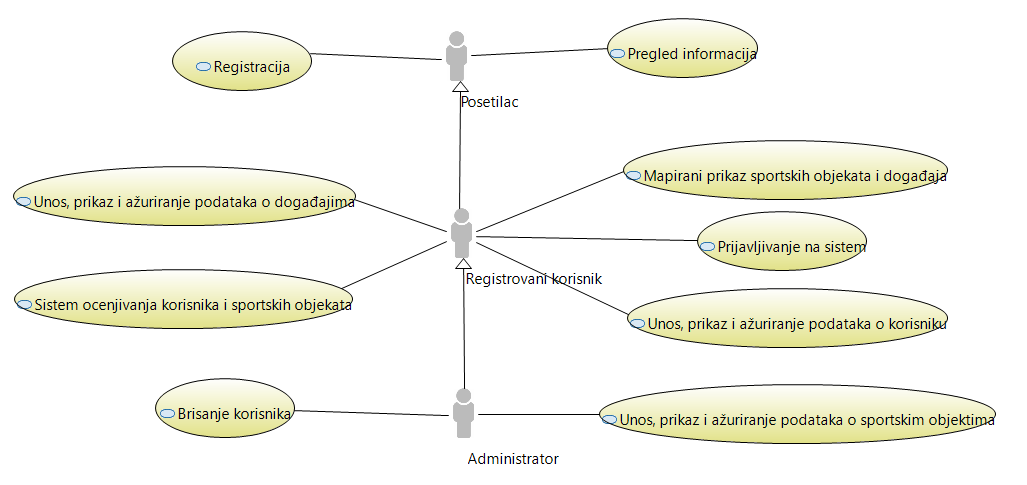
Slučajevi korišćenja PeNcIL portala su:

* Registracija
* Pregled informacija
* *Unos, prikaz i ažuriranje podataka o događajima*
  + Dodavanje novog događaja
    - Unos podataka o događajima
  + *Prikaz liste kreiranih dogadjaja*
    - Prikaz podataka o konkretnom dogadjaju
      * Vrsta dogadjaja
      * Broj učesnika
      * Vreme i mesto održavanja
      * Prijavljivanje na dogadjaj
  + *Brisanje podataka po završetku događaja*
  + *Otkazivanje događaja*
* Sistem ocenjivanja korisnika i sportskih objekata
* Brisanje korisnika
* *Unos, prikaz i ažuriranje podataka o korisniku*
  + Unos podataka
  + Prikaz podataka
  + Ažuriranje podataka
* Mapirani prikaz sportskih objekata
* *Unos, prikaz i ažuriranje podataka o sportskim objektima*
  + Unos podataka
  + Prikaz podataka
  + Ažuriranje podataka
* Prijavljivanje na sistem

Ove slučajevi korišćenja mogu da iniciraju posetilac portala, član laboratorije, šef laboratorije, vođa projekta ili administrator.

## Dijagrami slučajeva korišćenja

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

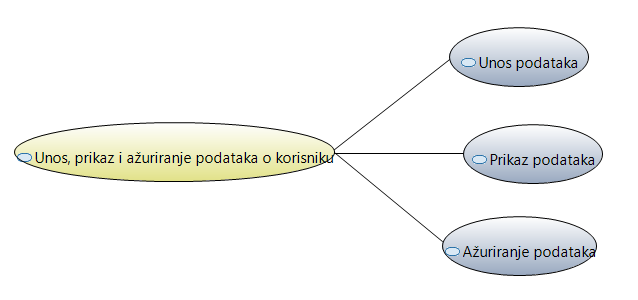


Slučajevi korišćenja *unos, prikaz i ažuriranje podataka o događajima/korisnicima/sportskim objektima* obuhvataju složenije radnje koje se mogu dalje razložiti na pojedinačne slučajeve korišćenja.

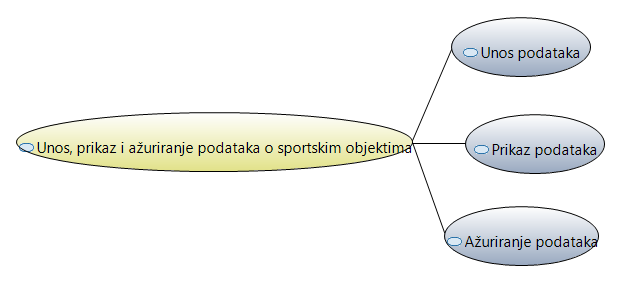
Detaljni UML dijagram za slučaj korišćenja *unos, prikaz i ažuriranje podataka o događajima* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *unos, prikaz i ažuriranje podataka o korisnicima* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *unos, prikaz i ažuriranje podataka sportskim objektima* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Registracija

Kratak opis: Registracija posetioca portala na sajt..

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Administrator

### Pregled osnovnih informacija

Kratak opis: Prikaz stranice portala sa osnovnim podacima o sajtu.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Član laboratorije, Šef laboratorije, Vođa projekta, Administrator.

### Dodavanje novog događaja

Kratak opis: Registrovani korisnik kreira novi događaj.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator

### Prikaz podataka o kreiranim događajima

Kratak opis: Prikaz stranice sa listom kreiranih događaja i njihovih osnovnih podataka.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator

### Prikaz podataka o konkretnom događaju

Kratak opis: Prikaz stranice sa osnovnim podacima o konkretnom događaju.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Adminitstrator

### Brisanje podataka o završenom događaju

Kratak opis: Brisanje svih podataka o završenom događaju.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Otkazivanje događaja

Kratak opis: Korisnik koji je kreirao događaj otkazuje taj događaj.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Prijavljivanje na događaj

Kratak opis: Korisnik se prijavljuje na događaj iz liste kreiranih događaja.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik

### Brisanje postojećih korisnika

Kratak opis: Administrator briše postojećeg korisnika iz baze.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Unos podataka o korisniku

Kratak opis: Registrovani korisnik unosi detaljnije podatke o sebi (slika, biografija, interesovanja...).

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Ažuriranje podataka o korisniku

Kratak opis: Ažuriranje podataka o sebi od strane registrovanog korisnika.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Prikazivanje podataka o korisniku

Kratak opis: Otvara se stranica sa prikazanim podacima o registrovanom korisniku.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Unos podataka o sportskim objektima

Kratak opis: Administrator unosi podatke o sportskom objektu.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje podataka o sportskim objektima

Kratak opis: Izmena osnovnih podataka o izabranom sportskom objektu..

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Prikaz podataka o sportskim objektima

Kratak opis: Otvara se stranica sa podacima o izabranom sportskom objektu.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Registrovani korisnik, Administrator.

### Mapirani prikaz sportskih objekata i dogadjaja

Kratak opis: Prikazivanje sportskih objekata na mapi.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Registrovani korisnik, Administrator.

### Sistem ocenjivanja sportskih objekata

Kratak opis: Registrovanim korisnicima se pruža mogućnost ocenjivanja sportskih objekata.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Sistem ocenjivanja drugih korisnika

Kratak opis: Registrovani korisnici imaju mogućnost ocenjivanja drugih korisnika.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Prijavljivanje na sistem

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

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, 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.

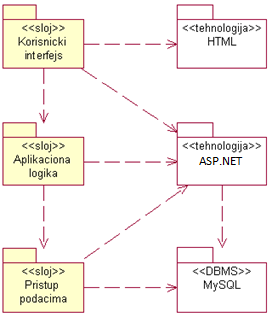
Logički pogled na GymBro portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži .NET framework zadužen za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži .NET framework koji predstavlja interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs portala. U njemu su sadržane sve HTML, multimedijalni sadržaji i ASP.NET aplikacije koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i ASP.NET.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi GymBro portala. Sadrži ASP.NET aplikacije koje 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 ASP.NET paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži ASP.NET aplikacije zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MySQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa ASP.NET i MySQL baza podataka.

### HTML

tehnologija

Tehnologija HTML-a definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### ASP.NET

tehnologija

Tehnologija ASP.NET-a obezbeđuje sadržaj od servera do računara klijenta preko Interneta. To su samostalne aplikacije koje objedinjuju razne aspekte Interneta kako bi obezbedile onlajn registraciju, pomoć, ažuriranje podataka ili druge usluge korisniku na Internetu.

### MySQL

DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju GymBro portala.

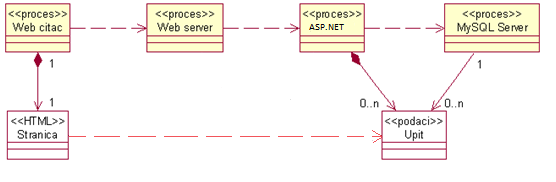
# Pogled na procese

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

Web aplikacije zasnovane na ASP.NET-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera.

## Procesi

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



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu HTML stranicu.

Web čitač zavisi od Web servera koji generiše i vraća odgovarajuću HTML stranicu na zahtev.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. U kontekstu ASP.NET MVC aplikacija, Web Server opslužuje HTTP zahteve sa čitača i na osnovu njih generiše HTML stranice iz .cshtml koda za odgovarajući pogled i po potrebi komunicira sa MySQL serverom da bi učitao podatke iz baze podataka koje potom prikazuje korisniku u sklopu prikazane HTML stranice na klijentskoj strani.

### ASP.NET MVC

ASP.NET MVC arhitektura omogućava sistem podeljen na 3 dela - **Model** koji modelira domen podataka i služi kao međusloj između aplikacije i baze podataka na MySQL serveru, **View** koji generiše HTML stranice koje se prikazuju korisniku u vreme izvršenjai **Controller** koji služi za povezivanje modela sa kljentskom stranom aplikacije i obezbeđuje funkcionalnosti aplikacije.

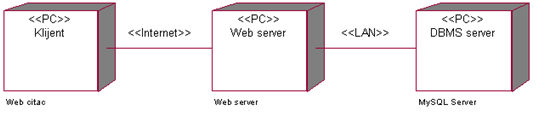
### MySQL Server

MySQL Server je proces koji izvršava funkcionalnost MySQL sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

# 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 GymBro portala.



## Klijent

Pristup GymBro portal 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, na ovom računaru koji koristi ASP.NET MVC arhitekturu aplikacije, mogu da se izvršavaju i procesi koji omogućuju prevođenje .cshtml koda u regularn HTML stranicu koja se prikazuje na klijentu. 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 MySQL 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).

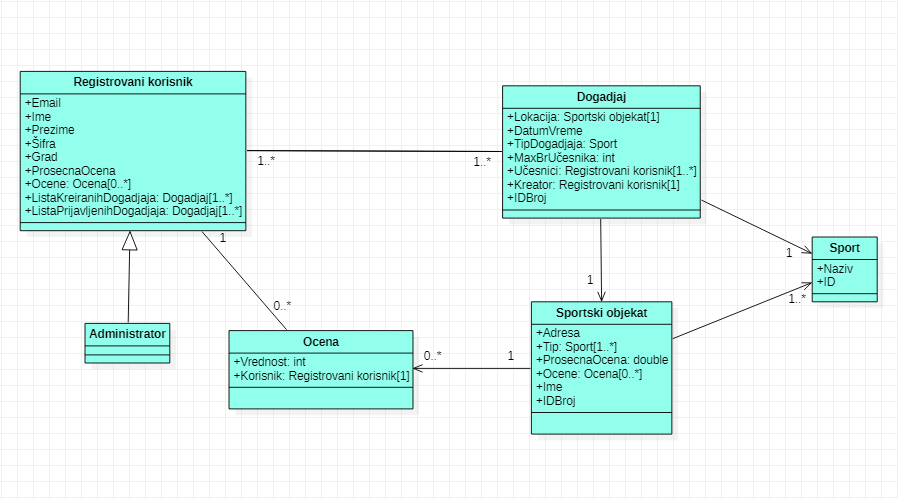
# Pogled na implementaciju sistema

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

## Model domena

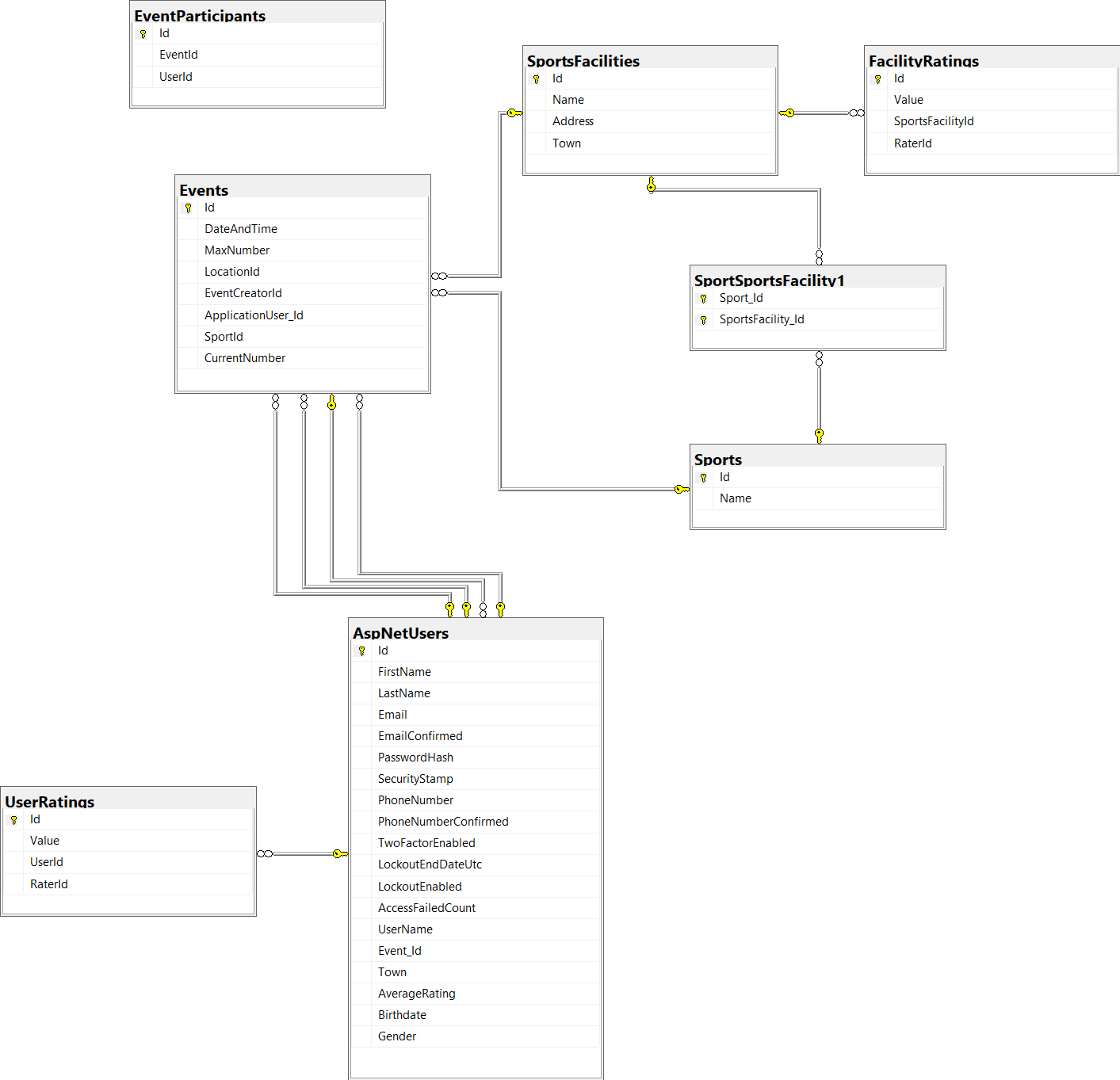
Model domena za koji se GymBro portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *Microsoft SQL Servera*.

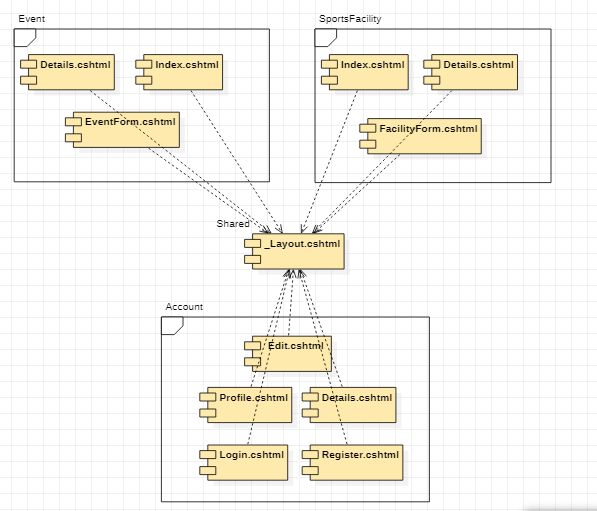


## Komponente sistema

Komponente sistema GymBro portala čine MVC arhitekturu podeljenu na 3 dela, **Models** koji modeluju entitete aplikacije i služe za generisanje baza podataka i čuvanje podataka o entitetima; **Views** koje uključuju Razor (.cshtml) stranice iz kojih se generišu HTML stranice koje se učitavaju na klijentovom Web čitaču; **Controllers** obezbeđuju funkcije aplikacije i služe kao međusloj za komunikaciju stranica na klijentskoj strani(Views) sa bazom podataka na serveru.

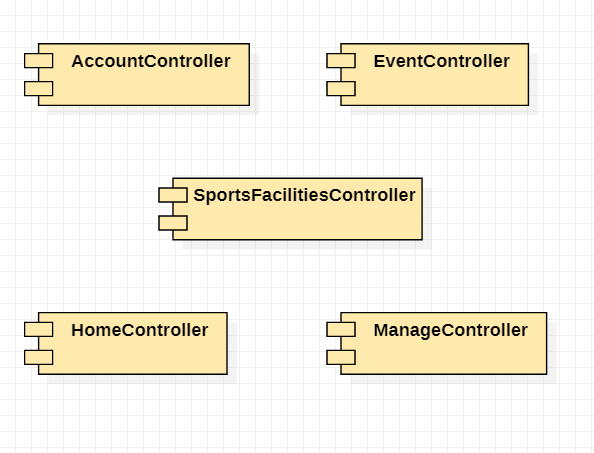
### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen svim .cshtml Razor stranicama. Razor stranice su smeštene u odgovarajuće direktorijume koji su imenovani po modelima.



### Komponente aplikacione logike

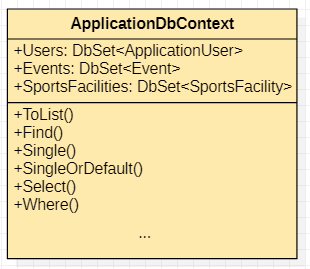
Komponente aplikacione logike čine kontroleri. Za svaki model postoji jedan kontroler koji čine sve funkcije odnosno akcije modeliranog entiteta. Akcije najčešće kao povratnu vrednost funkcije imaju određeni pogled (view) i u zavisnosti od logike kontrolera, bira se pogled odnosno stranica koja će biti prikazana na klijentu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru klase ApplicationDbContext koja u sebi sadrži svojstva (DbSet) koja predstavljaju tabele u bazi podataka. Svaki DbSet predstavlja jedan model aplikacije.

Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



# 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. GymBro portal će biti dostupan 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.