SC Active

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 19.04.2022. | 1.0 | Inicijalna verzija | Dušan Lukić, Dušan Kostadinović, Ivan, Marko Kostić |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 5](#_Toc101882703)

[2. Opseg dokumenta 5](#_Toc101882704)

[3. Reference 5](#_Toc101882705)

[4. Predstavljanje arhitekture 5](#_Toc101882706)

[5. Ciljevi i ograničenja arhitekture 5](#_Toc101882707)

[6. Pogled na slučajeve korišćenja 5](#_Toc101882708)

[6.1 Dijagrami slučajeva korišćenja 6](#_Toc101882709)

[6.2 Kratak opis slučajeva korišćenja 6](#_Toc101882710)

[6.2.1 Brisanje članova 6](#_Toc101882711)

[6.2.2 Ažuriranje tima 6](#_Toc101882712)

[6.2.3 Izaberi sport 6](#_Toc101882713)

[6.2.4 Pregled proizvoda po kategoriji 6](#_Toc101882714)

[6.2.5 Pregled liste trenera 7](#_Toc101882715)

[6.2.6 Pregled liste sportova 7](#_Toc101882716)

[6.2.7 Izaberi trenera 7](#_Toc101882717)

[6.2.8 Brisanje tima 7](#_Toc101882718)

[6.2.9 Kreiranje tima 7](#_Toc101882719)

[6.2.10 Ažuriraj progres klijenata 7](#_Toc101882720)

[6.2.11 Pregled podataka 7](#_Toc101882721)

[6.2.12 Kreiranje plana ishrane 7](#_Toc101882722)

[6.2.13 Ažuriranje podataka 7](#_Toc101882723)

[6.2.14 Prijavljivanje 7](#_Toc101882724)

[6.2.15 Dodavanje članova u listi klijenata 7](#_Toc101882725)

[6.2.16 Provera i validacija podataka 7](#_Toc101882726)

[6.2.17 Pregled slobodnih termina 7](#_Toc101882727)

[7. Pogled na logičku arhitekturu sistema 7](#_Toc101882728)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 8](#_Toc101882729)

[7.1.1 Korisnički interfejs 8](#_Toc101882730)

[7.1.2 Aplikaciona logika 8](#_Toc101882731)

[7.1.3 Pristup podacima 8](#_Toc101882732)

[7.1.4 HTML 9](#_Toc101882733)

[7.1.5 CSS 9](#_Toc101882734)

[7.1.6 JavaScript 9](#_Toc101882735)

[7.1.7 WEB API 9](#_Toc101882736)

[7.1.8 Azure Data Studio 9](#_Toc101882737)

[8. Pogled na procese 9](#_Toc101882738)

[8.1 Procesi 9](#_Toc101882739)

[8.1.1 Web čitač 10](#_Toc101882740)

[8.1.2 Web server 10](#_Toc101882741)

[8.1.3 JavaScript-CGI 10](#_Toc101882742)

[8.1.4 Azure Data Studio Server 10](#_Toc101882743)

[9. Pogled na raspoređivanje sistema 10](#_Toc101882744)

[9.1 Klijent 10](#_Toc101882745)

[9.2 Web server 10](#_Toc101882746)

[9.3 DBMS server 10](#_Toc101882747)

[10. Pogled na implementaciju sistema 11](#_Toc101882748)

[10.1 Model domena 11](#_Toc101882749)

[10.2 Šema baze podataka 11](#_Toc101882750)

[10.3 Komponente sistema 12](#_Toc101882751)

[10.3.1 Komponente korisničkog interfejsa 12](#_Toc101882752)

[10.3.2 Komponente aplikacione logike 13](#_Toc101882753)

[10.3.3 Komponente za pristup podacima 13](#_Toc101882754)

[11. Performanse 13](#_Toc101882755)

[12. Kvalitet 13](#_Toc101882756)

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture SC Active.

# Opseg dokumenta

Dokument se odnosi na SC Active portal koji će biti razvijen od strane IMD2. SC predstavlja skraćenicu za sportski centar. Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja vezanih za rad jednog sportskog centra

# Reference

Spisak korišćene literature:

1. D01\_Predlog\_Projekta.docx
2. D02\_Vizija\_Sistema.docx
3. D03\_Plan\_Realizacije.docx
4. D03\_Raspored\_Aktivnosti.pod
5. D04\_Spec\_Zahteva.docx

# 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. SC Active portal će biti implementiran kao Web aplikacija zasnovana na Java Script skripting jeziku i Azure Data Studion bazi podataka [4].
2. Klijentski deo SC Active portala će biti optimizovan za sve Web čitače:[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].

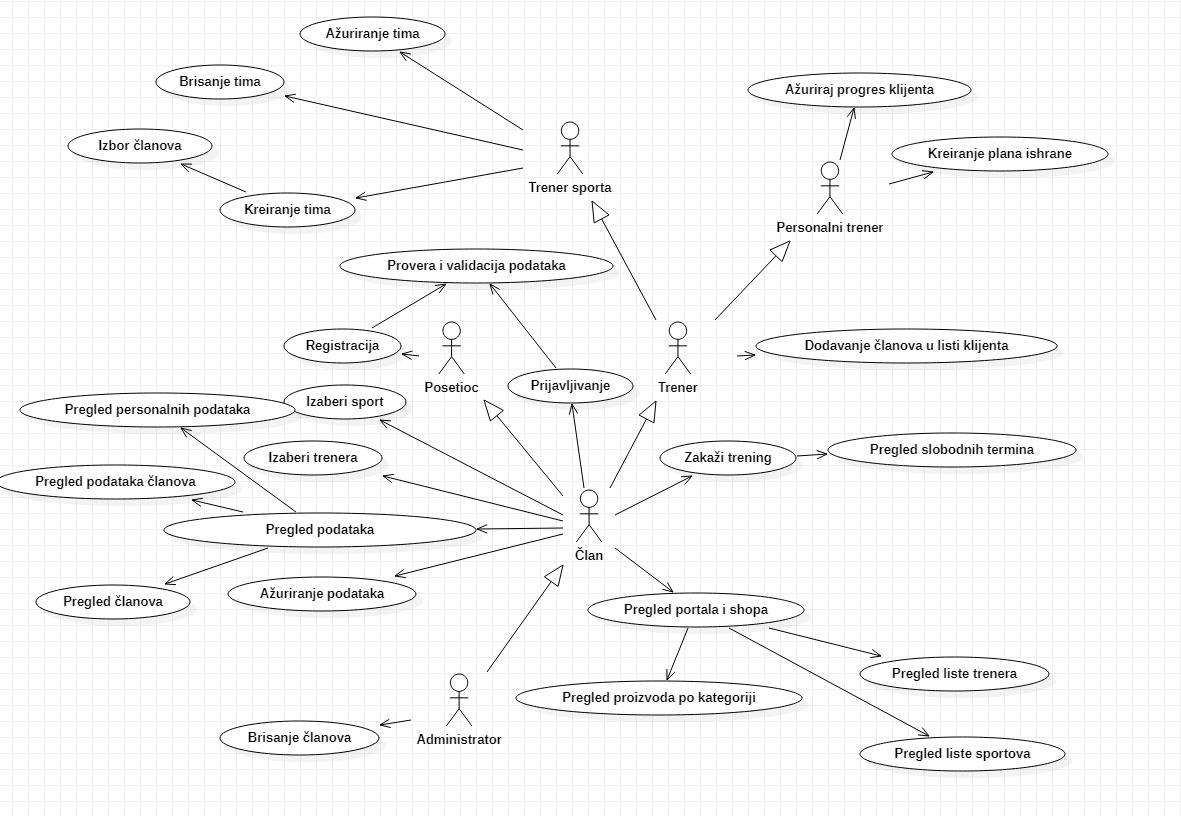
Slučajevi korišćenja SC Active portala su:

* Registracija
  + Provera i validacija podataka
* Prijavljivanje
  + Provera i validacija podataka
* Kreiranje tima
  + Izbor članova
* Brisanje tima
* Ažuriranje tima
* Ažuriraj progres klijenata
* Kreiranje plana ishrane
* Dodavanje članova u listi klijenata
* Zakaži trening
  + Pregled slobodnih termina
* Izaberi sport
* Izaberi trenera
* Pregled podataka
  + Pregled personalnih podataka
  + Pregled podataka članova
  + Pregled članova
* Ažuriranje podataka
* Pregled portala i shop
  + Pregled proizvoda po kategoriji
  + Pregled liste sportova
  + Pregled liste trenera
* Brisanje članova

Ove slučajevi korišćenja mogu da iniciraju Psetilac,Trener sports, Personalni trener,Trener,Član,Administrator.

## Dijagrami slučajeva korišćenja

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



## Kratak opis slučajeva korišćenja

### Brisanje članova

Kratak opis: Brisanje članova sportskog centra

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje tima

Kratak opis: Trener sporta vrši ažuriranje podataka o timu.

Akteri koji iniciraju slučaj korišćenja: Trener Sporta, Trener, Član, Administrator

### Izaberi sport

Kratak opis: Član bira između ponuđenih sportova iz liste.

Akteri koji iniciraju slučaj korišćenja: Član.

### Pregled proizvoda po kategoriji

Kratak opis: Pregled proizvoda u prodavnici sortiran po kategoriji

Akteri koji iniciraju slučaj korišćenja: Administrator, Član

### Pregled liste trenera

Kratak opis: Pregled svih trenera, personalnih i trenera sporta

Akteri koji iniciraju slučaj korišćenja: Administrator, Član

### Pregled liste sportova

Kratak opis: Pregled svih sportova

Akteri koji iniciraju slučaj korišćenja: Administrator, Član

### Izaberi trenera

Kratak opis: Član bira trenera iz liste ponuđenih trenera.

Akteri koji iniciraju slučaj korišćenja: Član.

### Brisanje tima

Kratak opis: Trener sporta vrši brisanje podataka o timu.

Akteri koji iniciraju slučaj korišćenja: Trener Sporta, Trener ,Član, Administrator

### Kreiranje tima

Kratak opis: Trener sports vrši kreiranje novog tima

Akteri koji iniciraju slučaj korišćenja: Trener Sporta,Trener,Član,Administrator

### Ažuriraj progres klijenata

Kratak opis: Personalni trener ažurira progres o izabranom klijentu i klijent mora da postoji da bi se njegov progres ažurirao.

Akteri koji iniciraju slučaj korišćenja: Trener Sporta,Trener,Član,Administrator

### Pregled podataka

Kratak opis: Pregled profila korisnika i osnovih informacija.

Akteri koji iniciraju slučaj korišćenja: Trener Sporta,Trener,Član,Administrator

### Kreiranje plana ishrane

Kratak opis: Personalni trener vrši kreiranje plana ishrane za svakog klijenta.

Akteri koji iniciraju slučaj korišćenja:Personalni trener, Trener, Član, Administrator

### Ažuriranje podataka

Kratak opis: Ažuriranje podataka na profilu korisnika.

Akteri koji iniciraju slučaj korišćenja: Trener Sporta, Trener, Član, Administrator

### Prijavljivanje

Kratak opis: Korisnik unosi korisničko ime i sifru za prijavljivanje.

Akteri koji iniciraju slučaj korišćenja: Posetilac, Član, Administrator

### Dodavanje članova u listi klijenata

Kratak opis: Kada treneru stigne zahtev od strane člana on ga dodaje u listi klijenata.

Akteri koji iniciraju slučaj korišćenja: Trener,Član,Administrator

### Provera i validacija podataka

Kratak opis: Provera podataka člana tokom registracije i validacija istih

Akteri koji iniciraju slučaj korišćenja: Administrator, Član, Posetioc

### Pregled slobodnih termina

Kratak opis: Omogućava članu da vidi koji termini za trening su slobodni

Akteri koji iniciraju slučaj korišćenja: Administrator, član, trener, personalni trener, trener sporta

# 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.

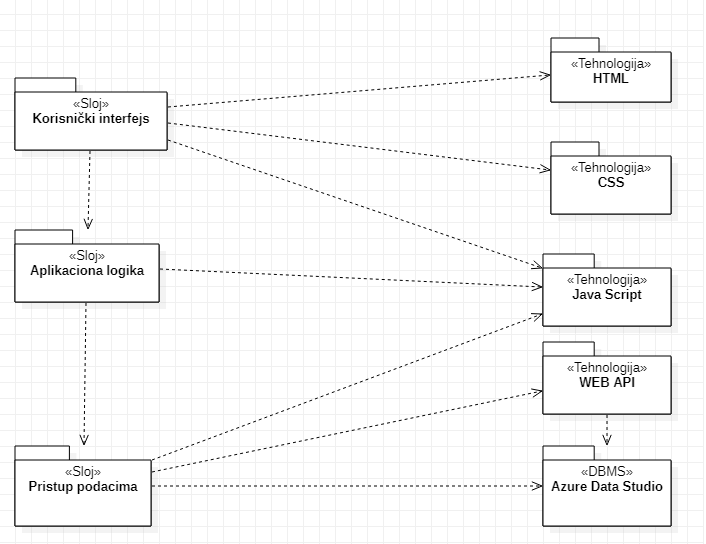
Logički pogled na SC Active portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, JavaScript skripte, HTML, CSS 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 JavaScript skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži WEB API, Azure Data Studio i JavaScript skripte koji predstavljaju 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, Css, multimedijalni sadržaji i JavaScript skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML, CSS i JavaScript.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi SC Active portala. Sadrži JavaScript skripte 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 JavaScript paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži JavaScript skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u Azure Data Studio bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa JavaScript i Azure Data Studio 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.

### CSS

tehnologija

Tehnologija CSS omogućuje uređivanje dizajna portala.

### JavaScript

tehnologija

Tehnologija JavaScript-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte mogu da generišu HTML kod koji realizuje korisnički interfejs i pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### WEB API

tehnologija

WEB API predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju SC Active portala.

### Azure Data Studio

DBMS

Azure Data Studio predstavlja sistem u kome će se nalaziti baza podataka portala..

# Pogled na procese

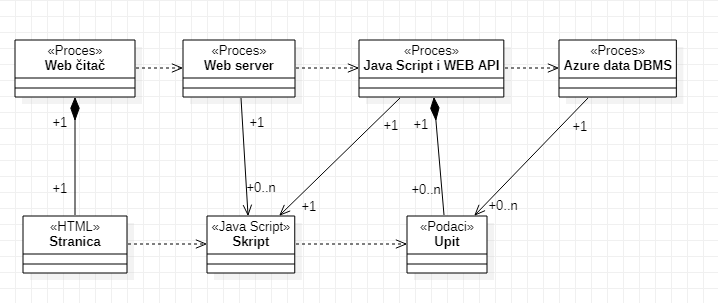
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na JavaScript-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta JavaScript Web aplikacije nije potrebno voditi računa o načinu rada Web servera i načinu izvršavanja skripti.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje SC Active portala kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju SC Active portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na JavaScript-u i Azure Data Studio 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. Ukoliko je zahtevana stranica JavaScript skript, Web server inicira izvršenje JavaScript-CGI procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj JavaScript-CGI procesa.

### JavaScript-CGI

JavaScript-CGI proces obavlja posao obrade zadatog JavaScript skripta i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web čitaču. Za izvršenje JavaScript skripta ovaj proces može da zahteva usluge Azure Data Studio servera-a. Komunikacija između JavaScript-CGI procesa i Azure Data Studio servera se obavlja preko prosleđivanja upita i vraćanja rezultat.

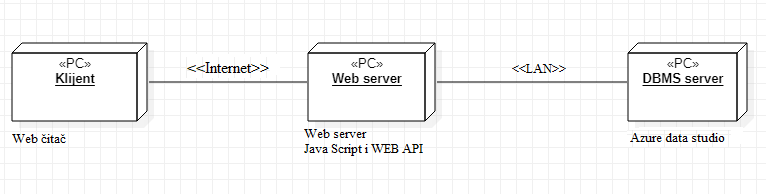
### Azure Data Studio Server

Azure Data Studio Server je proces koji izvršava funkcionalnost Azure Data Studio 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 SC Active portala.



## Klijent

Pristup SC Active 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 mogu da se izvršavaju i procesi JavaScript-WEB API koji vrše obradu zadatih JavaScript skripti. 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 Azure Data Studio 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).

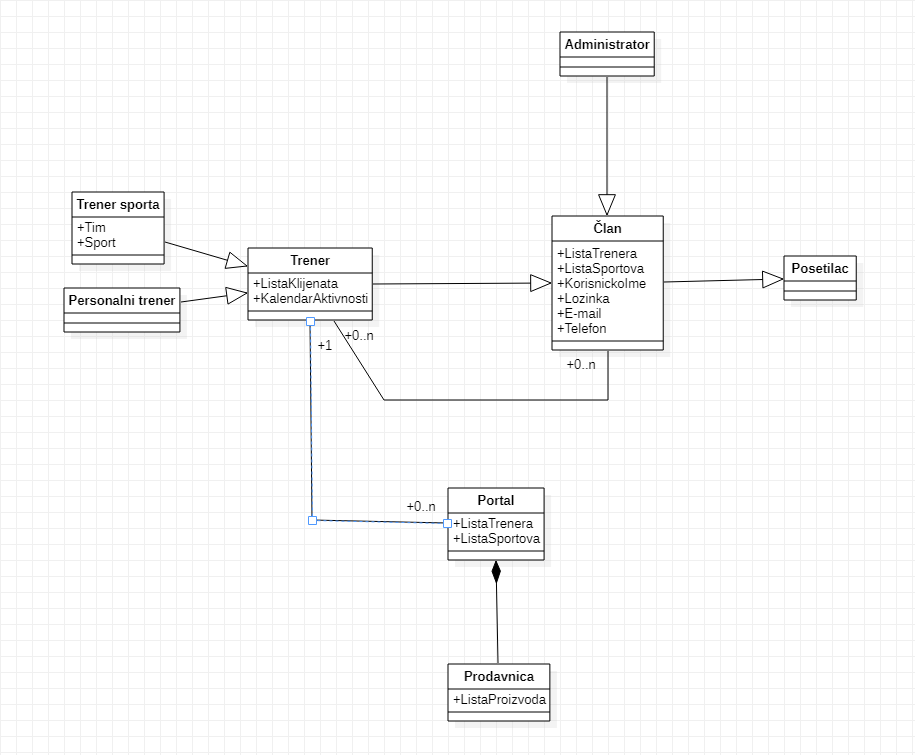
# Pogled na implementaciju sistema

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

## Model domena

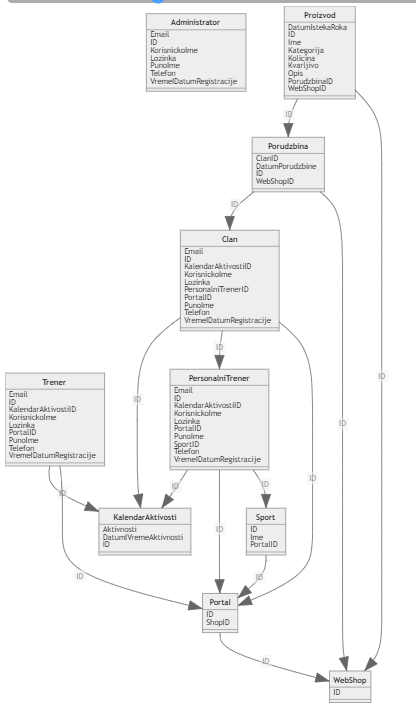
Model domena za koji se SC Active portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *MS Access*-a, dok je migracija na Azure Data Studio obavljena pomoću *Azure Data Studio Migration Toolkit*-a.



## Komponente sistema

*NAPOMENA: SC Active projekat nije kompletno implementiran, tj. implementiran je na nivou arhitekturnog prototipa. Zbog ovoga sledeću sekciju trebate shvatiti kao ilustraciju kako bi dokument trebao da izgleda. Opis komponenti sistema za vaše projekte mora biti kompletan! Procena složenosti vašeg rešenja će zavisiti od opisa sistema koji ovde date.*

Komponente sistema SC Active portala su JavaScript skripti čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je JavaScript skript prikazan kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

Komponenta **main.js** implementira stranicu portala čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

Komponenta **style.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

Komponenta **index.html** predstavlja strukturu samog portala.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **main.js** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu.

Backend aplikacije se radi uz pomoć .NET framework tehnologije.

Sami kontroleri i pristupaju bazi i izvršavaju određene zahteve i obradu podataka.  
Iz frontend-a se šalju api pozivi za određene kontrolere koji komuniciraju sa SQL serverom na kome se nalazi baza podataka. Modeli predstavlaju tabele u bazi a propertiji svakog modela kolone baze.

Sve se u frontend-u dodeljuje dinamički.

### Komponente za pristup podacima

Pristup bazi podataka izvršavaće odgovarajući kontroleri kroz DBContext klasu.

Server i baza su hostovani preko Azure-a.

# 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.

# Kvalitet

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

1. SC Active 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.