WellNi

**Web aplikacija za Wellness centar**

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 18.04.2023. | 1.0 | Inicijalna verzija | Katarina Lazarević, Tijana Ilić, Petra Petronijević |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 Pregled naslovne strane portala 7

6.2.2 Pregled zaposlenih 7

6.2.3 Pregled sadržaja centra 8

6.2.4 Pregled usluga 8

6.2.5 Pregled recenzija 8

6.2.6 Filtriranje na osnovu naziva tipa usluge 8

6.2.7 Filtriranje na osnovu vršilaca usluge 8

6.2.8 Sortiranje na osnovu cene usluge 8

6.2.9 Ocenjivanje usluga 8

6.2.10 Kreiranje naloga 8

6.2.11 Prijavljivanje 8

6.2.12 Ažuriranje podataka o uslugama 8

6.2.13 Ažuriranje osnovnih podataka o centru 8

6.2.14 Kreiranje specijalnih ponuda 8

6.2.15 Kreiranje novog zaposlenog 8

6.2.16 Brisanje postojećeg zaposlenog 8

6.2.17 Brisanje profila korisnika 9

7. Pogled na logičku arhitekturu sistema 9

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 9

7.1.1 Korisnički interfejs 9

7.1.2 Aplikaciona logika 9

7.1.3 Pristup podacima 10

7.1.4 HTML 10

7.1.5 .NET 10

7.1.6 CSS 10

7.1.7 Microsoft Azure SQL 10

8. Pogled na procese 10

8.1 Procesi 10

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 .NET 11

8.1.4 Microsoft Azure SQL Server 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 13

10.3.1 Komponente korisničkog interfejsa 13

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 ***WellNi*** Web portala.

# Opseg dokumenta

Dokument se odnosi na WellNi portal koji će biti razvijen od strane TreSoft tima. Namena sistema je efikasno i jednostavno zakazivanje usluga centra, pregled sadržaja i osoblja, ocenjivanje i komentarisanje usluga.

# Reference

Spisak korišćene literature:

1. WellNi – Predlog projekta, TreSoft-WellNi-01, V1.0, 2023, TreSoft.
2. WellNi – Vizija sistema, TreSoft-WellNi-02, V1.0, 2023, TreSoft.
3. WellNi – Plan realizacije projekta, TreSoft-WellNi-03, V1.0, 2023, TreSoft.
4. WellNi – Raspored aktivnosti, TreSoft-WellNi-04, V1.0, 2023, TreSoft.
5. WellNi – Specifikacija zahteva, TreSoft-WellNi-05,V1.0, 2007, TreSoft.

# 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. ***WellNi*** portal će biti implementiran kao Web aplikacija zasnovana na .NET, javascript, react i Microsoft Azure SQL Server bazi podataka [2].
2. Klijentski deo ***WellNi*** portala će biti optimizovan za sledeće Web čitače: Edge, Google Chrome, Firefox (Mozilla), Safari, Opera. [5].
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 *WellNi* Web portala su:

* *Pregled informacija*
  + Pregled naslovne strane portala
  + *Pregled zaposlenih*
    - Pregled trenera
    - Pregled fizioterapeuta
  + Pregled sadržaja centra
  + *Pregled usluga*
    - *Pregled dodatnih informacija o usluzi*
  + *Pregled recenzija*
  + *Filtriranje*
    - Filtriranje na osnovu naziva tipa usluge
    - Filtriranje na osnovu vršilaca usluge
* Sortiranje

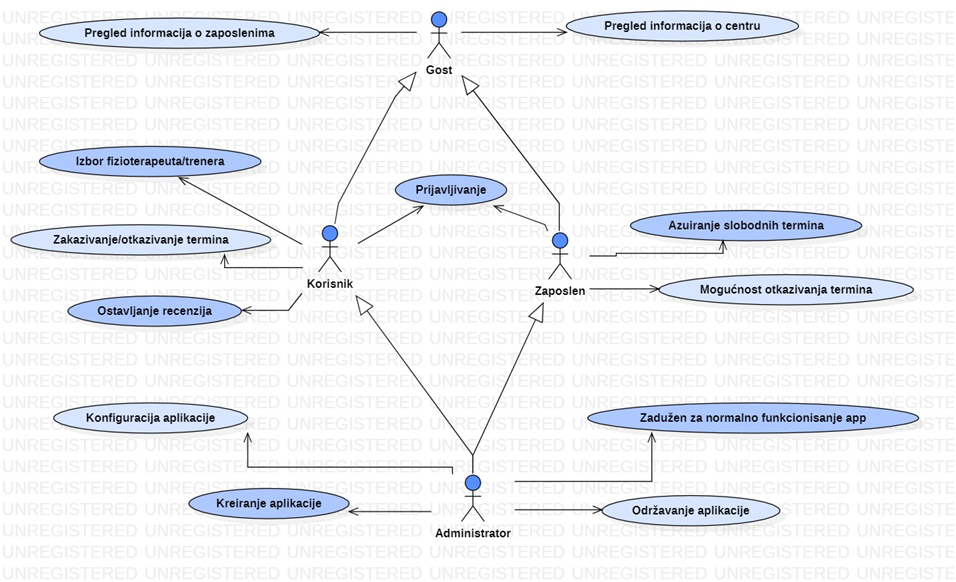
## Sortiranje na osnovu cene usluge

* Prijavljivanje
  + Kreiranje naloga
* Ocenjivanje usluga
  + Ostavljanje ocena
  + Ostavljenje komentara
* *Ažuriranje podataka o uslugama*
  + Dodavanje nove usluge
  + Brisanje postojeće usluge
  + Izmena postojeće usluge
* Ažuriranje osnovnih podataka o centra
* Kreiranje specijalnih ponuda
* *Kreiranje i brisanje zaposlenih*
  + Kreiranje novog zaposlenog
  + Brisanje postojećeg zaposlenog
* Brisanje profila korisnika
* Održavanje sistema

Ove slučajeve korišćenja mogu da iniciraju posetilac portala, korisnik ili administrator.

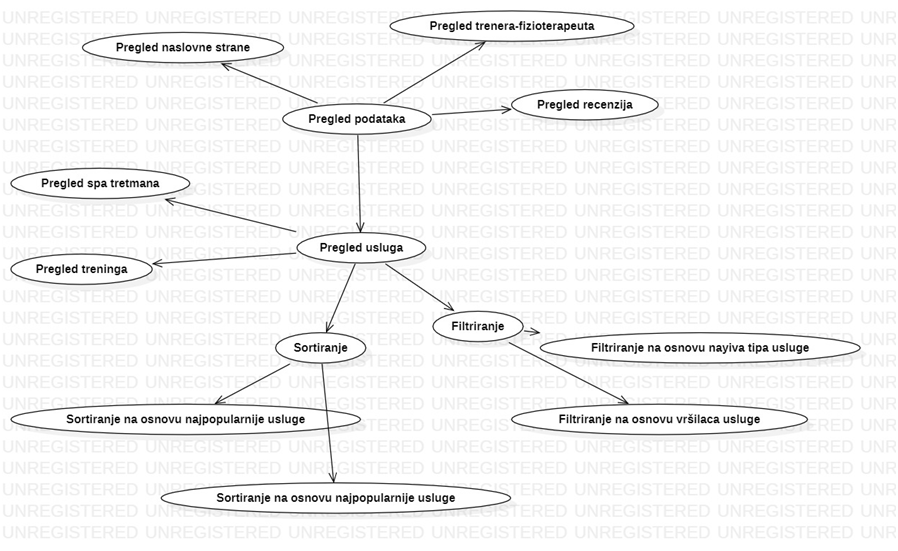
## Dijagrami slučajeva korišćenja

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

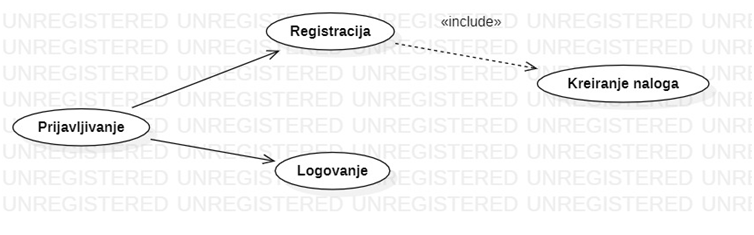


Slučajevi korišćenja *pregled informacija* i *kreiranje, korisnika obu*hvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

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



Detaljni UML dijagram za slučaj korišćenja *kreiranje naloga korisnika* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Pregled naslovne strane portala

Kratak opis: Prikaz naslovne strane portala.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Prijavljeni korisik, Zaposleni i Administrator.

### Pregled zaposlenih

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

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Prijavljeni korisik, Zaposleni i Administrator.

### Pregled sadržaja centra

Kratak opis: Prikaz stranice portala sa sadržajem i slikama centra.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Prijavljeni korisik, Zaposleni i Administrator.

### Pregled usluga

Kratak opis: Prikaz stranice sa informacijama o uslugama centra.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Prijavljeni korisik, Zaposleni i Administrator.

### Pregled recenzija

Kratak opis: Prikaz stranice sa recenzijama zaposlenih i usluga centra.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Prijavljeni korisik, Zaposleni i Administrator.

### Filtriranje na osnovu naziva tipa usluge

Kratak opis: Na stranici se prikazuju usluge koje ispunjavaju određeni kriterijum.

Akteri koji iniciraju slučaj korišćenja: portala, Prijavljeni korisik, Zaposleni i Administrator.

### Filtriranje na osnovu vršilaca usluge

Kratak opis: Na stranici se prikazuju usluge koje ispunjavaju određeni kriterijum.

Akteri koji iniciraju slučaj korišćenja: portala, Prijavljeni korisik, Zaposleni i Administrator.

### Sortiranje na osnovu cene usluga

Kratak opis: Na stranici se prikazuju usluge koje ispunjavaju određeni kriterijum.

Akteri koji iniciraju slučaj korišćenja: portala, Prijavljeni korisik, Zaposleni i Administrator.

### Ocenjivanje usluga

Kratak opis: Korisnik dodaje recenziju.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisik.

### Kreiranje naloga

Kratak opis: Registrovanje korisnika.

Akteri koji iniciraju slučaj korišćenja: Gost.

### Prijavljivanje

Kratak opis: Prijavljivanje korisnika na portal sa postojećim nalogom.

Akteri koji iniciraju slučaj korišćenja:Prijavljeni korisnik.

### Ažuriranje podataka o uslugama

Kratak opis: Ažuriranje podataka ouslugama centra.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Ažuriranje osnovnih podataka o centru

Kratak opis: Izmena osnovnih podataka o centru, dodavanje i brisanje informacija.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Kreiranje specijalnih ponuda

Kratak opis:Administrator objavljuje specijalne ponude centra koje su vidljive registrovanim korisnicima.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Kreiranje novog zaposlenog

Kratak opis: Kreiranje korisničkog naloga za novog člana centra.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje postojećeg zaposlenog

Kratak opis: Brisanje korisničkog naloga i podataka za postojećeg člana centra.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje profila korisnika

Kratak opis: Brisanje korisničkog naloga i podataka za postojećeg korisnika.

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.

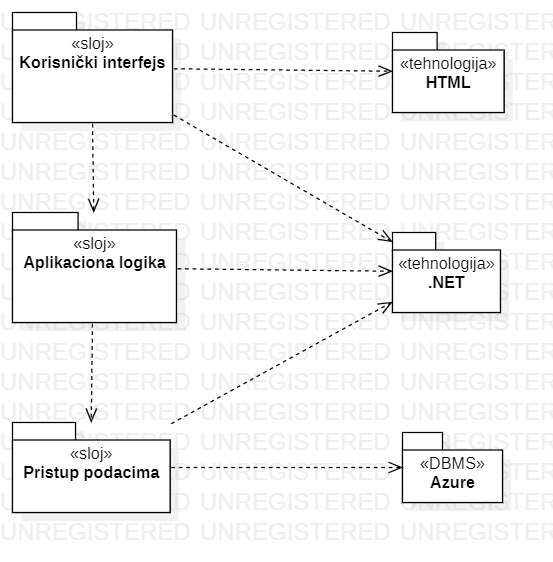
Logički pogled na WellNi Web portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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

Paket *Pristup podacima* sadrži .NET skripte koje 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 ***WellNI*** Web portala. U njemu su sadržane sve HTML stranice, CSS stilove i React skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

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

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi ***WellNi***  Web portala. Sadrži .NET skripte koje realizuju funkcionalnost karakterističnu za domen primene aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i .NET paketa.

### Pristup podacima

sloj

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

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa .NET i Microsoft Azure SQL 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.

### .NET

tehnologija

Tehnologija Microsoft .NET Framework-a za razvoj dinamičkih web sajtova, interaktivnih veb aplikacija i veb servisa sa korišćenjem baza podataka za PC i mobilne uređaje.

ASP.NET stranice se izvršavaju na serverskoj strani i generišu HTML, WML ili XML koji se šalje desktop ili mobil pretraživačima.ASP.NET koristi „event-driven“ model programiranja koji poboljšava performanse i omogućava sepraciju korisničkog interfejsa od logike aplikacije.ASP.NET radi na vrhu HTTP protokola koristeći HTTP komande i pravila kako bi omogućio obostranu komunikaciju izmedju klijenta i servera.Kod je moguće pisati u C#, VisualBasic, Jscript ili J# jezicima.

ASP.NET omogućava 3 metode razvoja :

• Web Forms (tradicionalni ASP.NET)

• Web Pages

• MVC (Model View Controller)

### CSS

tehnologija

Dok HTML definiše gradivne elemente stranica, CSS je jezik formatiranja pomoću kog se definiše izgled elemenata Web-stranice.

### Microsoft Azure SQL

DBMS

Microsoft Azure SQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju WellNi Web portala.

# Pogled na procese

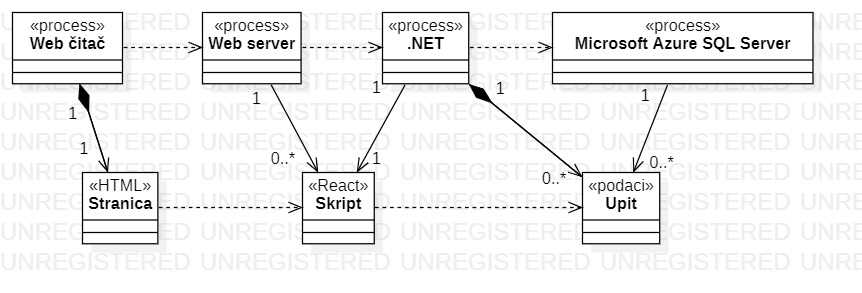
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na Reactu imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta React 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 WellNi portala kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju WellNi portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na .NET-u i Azure 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 React skript, Web server inicira izvršenje .NET procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj .NET procesa.

### .NET

.NET omogućava komunikaciju između web servera baze podataka (Microsoft Azure SQL Servera). Posredstvom .NET funkcija web server može dobiti tražene podatke ili poslati podatke.

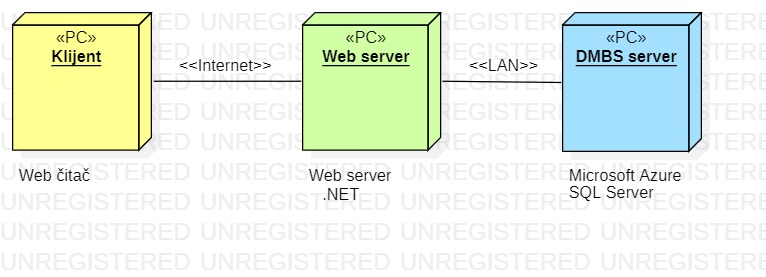
### Microsoft Azure SQL Server

Microsoft Azure SQL Server je proces koji izvršava funkcionalnost Azure 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 WellNi stranice.



## Klijent

Pristup WellNi stranici 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 .NET procesi koji vrše obradu zadatih React 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 Microsoft Azure SQL 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).

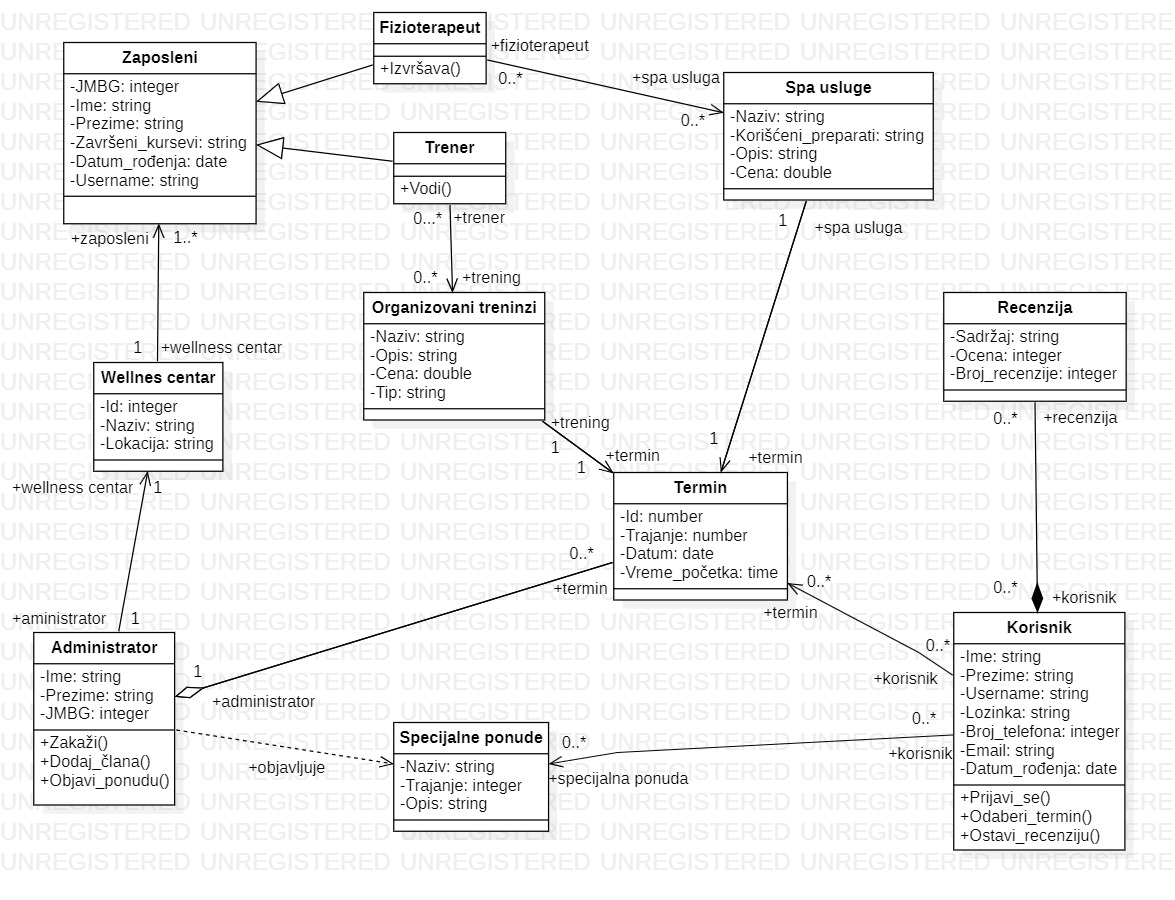
# Pogled na implementaciju sistema

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

## Model domena

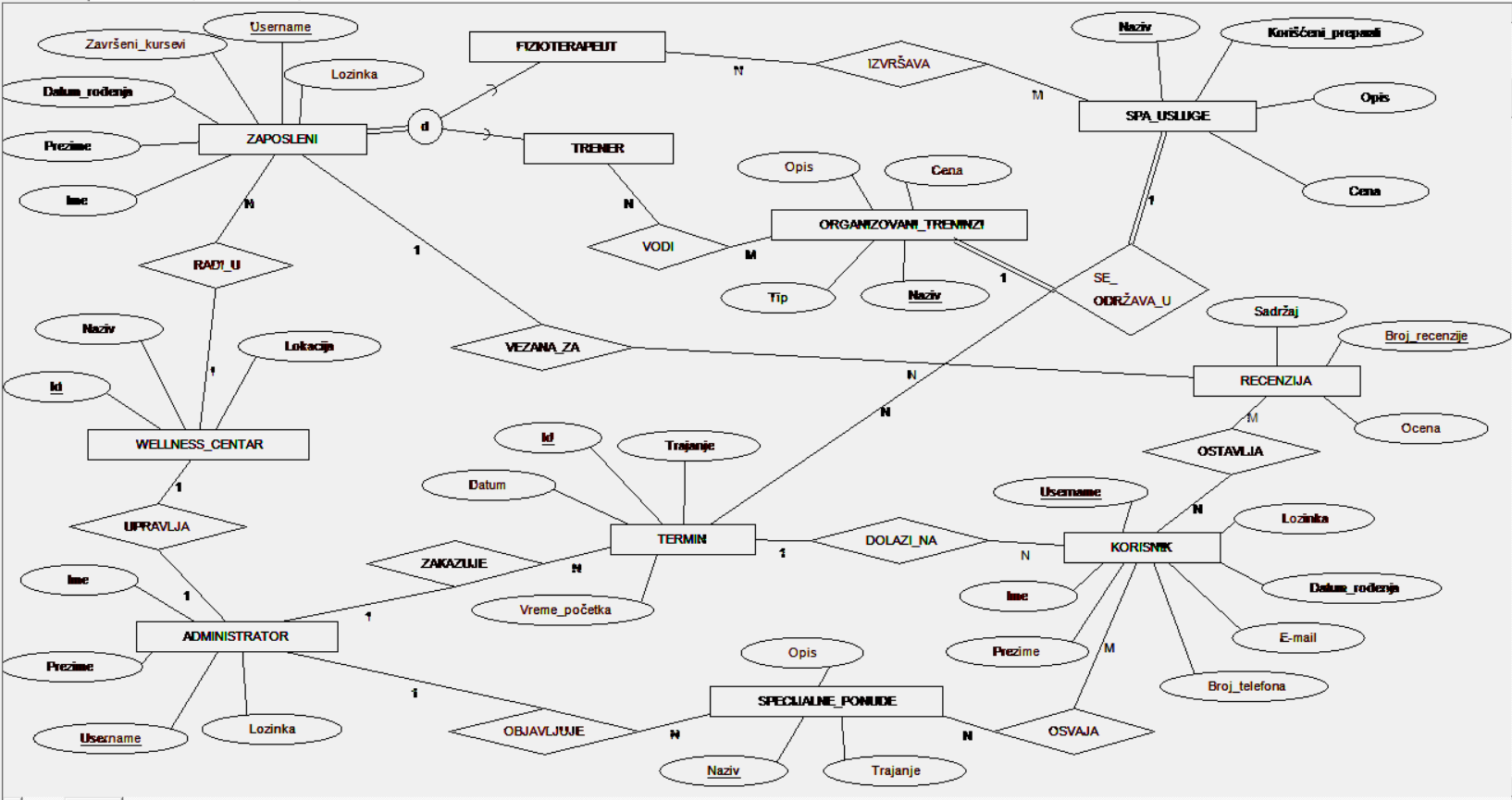
Model domena za koji se WellNi portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Šema baze podataka je prikazana na EER dijagramu ispod. Dijagram je kreiran korišćenjem SharpEER softvera.

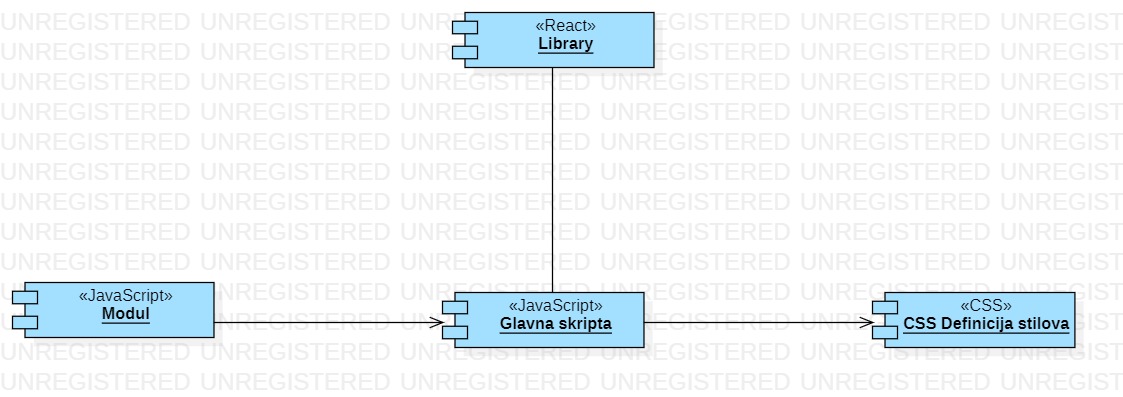


## Komponente sistema

Komponente sistema WellNiWeb aplikacije su JavaScript, CSS i React fajlovi 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 prikazan kao klasa, atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripti.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa sastoji se od glavne skripte JavaScripti modula od kojih je sačinjen, više css dokumenata i Reactbiblioteke.



Komponenta **Glavna skripa** predstavlja centar sistema i koristi se za generisanje dinamičkih HTML stranica i prosleđivanje već kreiranih statičkih front end datoteka.

Komponente **Modul** predstavljaju neke celine koje mogu da se ponavljaju u aplikaciji.

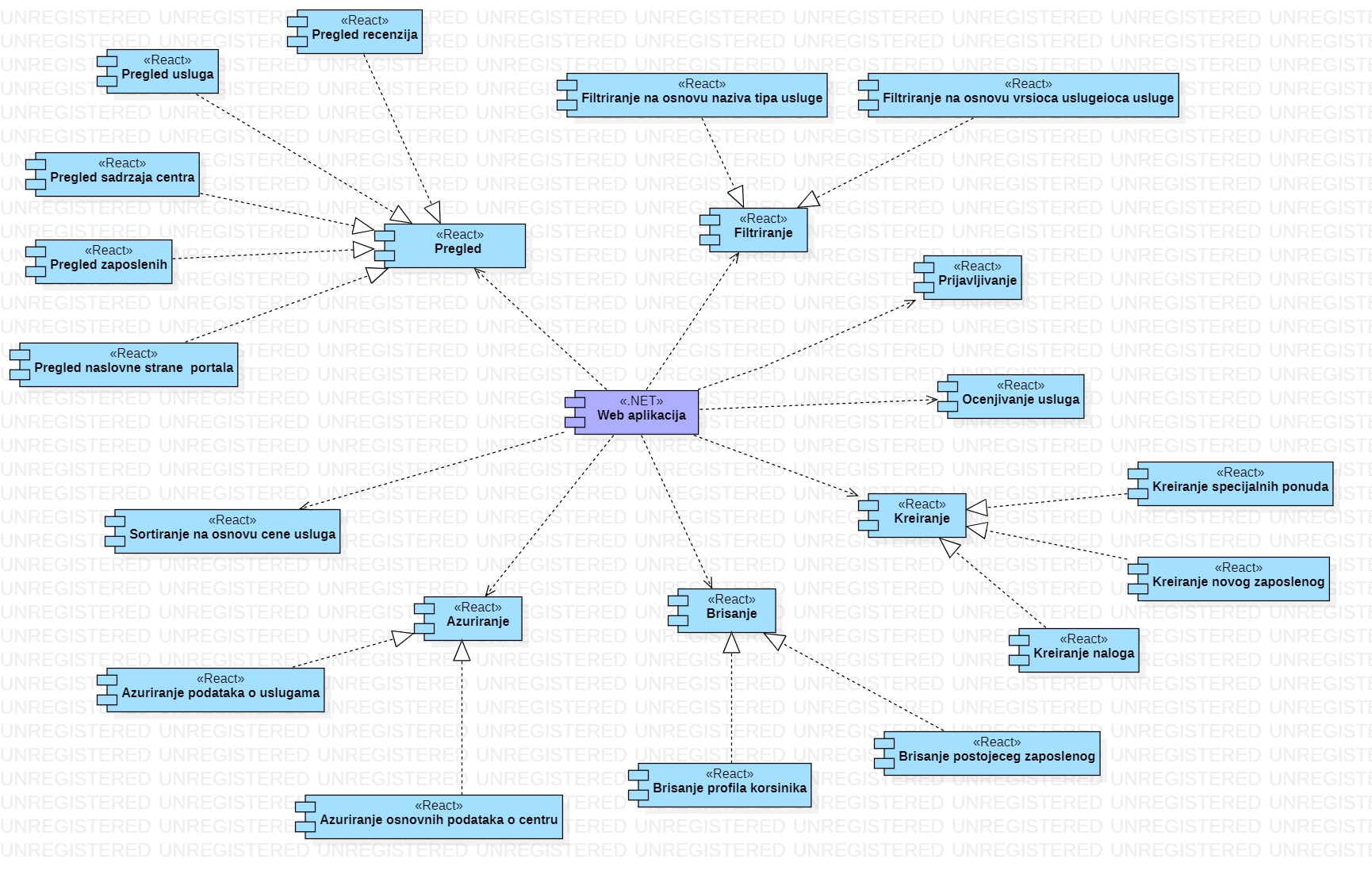
Komponente **Css definicije stilova** predstavlja generalizaciju CSS fajlova koji se koriste za stilizovanje pojedinačnih dinamičkih HTML modula.

Komponenta **React** predstavlja eksternu biblioteku za stilizovanje HTML elemenata i koristi se za kreiranje fleksibilnog korisničkog interfejsa.

Za celokupan dizajn Web Aplikacije se koristi JavaScript

### Komponente aplikacione logike

Na sledećem dijagramu su prikazane komponente aplikacione logike i njihove međusobne zavisnosti:



Pregled funkcija svake komponente:

-Pregled naslovne strane portala: Prikaz naslovne strane portala.

-Pregled zaposlenih: Prikaz stranice portala sa osnovnim podacima o zaposlenima.

-Pregled sadržaja centra: Prikaz stranice aplikacije sa sadržajem i slikama centra.

-Pregled usluga: Prikaz stranice sa informacijama o uslugama centra.

-Pregled recenzija: Prikaz stranice sa recenzijama zaposlenih i usluga centra.

-Filtriranje na osnovu naziva tipa usluge: Na stranici se prikazuju usluge koje ispunjavaju određeni kriterijum.

-Filtriranje na osnovu vršioca usluge: Na stranici se prikazuju usluge koje ispunjavaju određeni kriterijum.

-Sortiranje na osnovu cene usluga: Na stranici se prikazuju usluge koje ispunjavaju određeni kriterijum.

-Ocenjivanje usluga: Korisnik dodaje recenziju.

-Kreiranje naloga: Registrovanje korisnika.

-Prijavljivanje: Prijavljivanje korisnika na portal sa postojećim nalogom.

-Ažuriranje podataka o uslugama: Ažuriranje podataka o uslugama centra.

-Ažuriranje osnovnih podataka o centru: Izmena osnovnih podataka o centru, dodavanje i brisanje informacija.

-Kreiranje specijalnih ponuda:Administrator objavljuje specijalne ponude centra koje su vidljive registrovanim korisnicima.

-Kreiranje novog zaposlenog: Kreiranje korisničkog naloga za novog člana centra.

-Brisanje postojećeg zaposlenog: Brisanje korisničkog naloga i podataka za postojećeg člana centra.

-Brisanje profila korisnika: Brisanje korisničkog naloga i podataka za postojećeg korisnika.

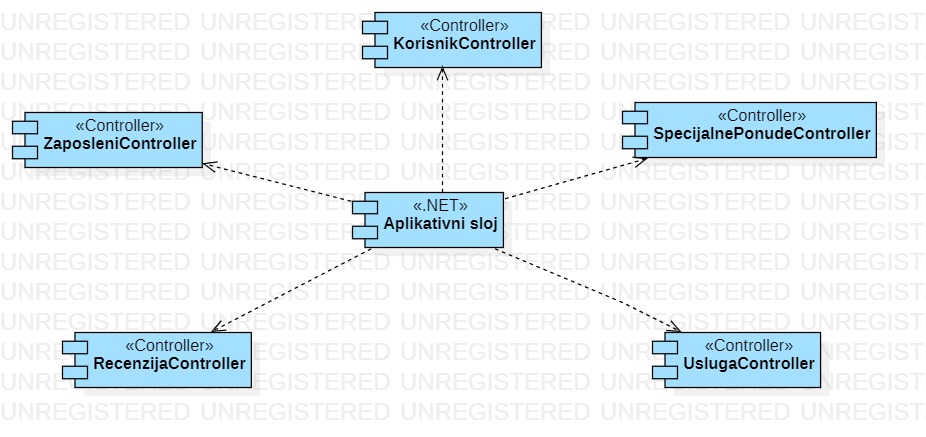
### Komponente za pristup podacima

Pristup bazi podataka omogućen je Web Aplikaciji preko Aplikativnog Sloja. Sva logika pristupa bazi podataka se nalazi na Aplikativnom Sloju. Web Aplikacija pristupa funkcijama Aplikativnog sloja(Kontrolerima i funkcijama kontrolera), onda Aplikativni sloj pristupa bazi podataka. Baza podataka(Microsoft Azure SQL Server) vraća odgovor aplikativnom sloju koji vraća odgovor Web Aplikaciji.



Kontroleri kojima Web Aplikacija pristupa radi pribavljanja podataka su sledeći:

1. KorisnikController - Sadrži funkcije za pristup podacima korisnika, kreiranje korisnika, prijavljivanje…
2. ZaposleniController - Sadrži funkcije za pristup podacima zaposlenih, kreiranje zaposlenih, prijavljivanje…
3. UslugaController - Sadrži funkcije za pristup podacima o uslugama, odabir usluga,…
4. RecenzijaController - Sadrži funkcije za pristup podacima o recenzijama (ukljucujuci i komentare), kreiranje recenzija, pregled recenzija o odredjenom treneru/fizioterapeutu, …
5. SpecijalnePonudeController - Sadrži funkcije za pristup podacima o specijalnim ponudama, kreiranje specijalnih ponuda,…



# 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. WellNi web aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupan ne sme da pređe 15%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.