Kućni poslovi

Web aplikacija za pronalaženje i ponude honorarnih poslova vezanih za pomoć u kući

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 29.04.2020. | 1.0 | Inicijalna verzija | Petar, Malbaša, Filip |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 Kreiranje naloga 7

6.2.2 Pregled početne stranice 7

6.2.3 Prijavljivanje na Web aplikaciju 7

6.2.4 Ažuriranje svog naloga 7

6.2.5 Slanje zahteva 7

6.2.6 Potvrda zahteva 8

6.2.7 Filtriranje zahteva 8

6.2.8 Ocenjivanje korisnika 8

6.2.9 Kreiranje oglasa 8

6.2.10 Ažuriranje oglasa 8

6.2.11 Brisanje oglasa 8

6.2.12 Prikaz prethodnih oglasa 8

6.2.13 Prikaz prethodnih saradnji 8

6.2.14 Pretraživanje pružalaca usluga 8

6.2.15 Pretraživanje oglasa 8

6.2.16 Filtriranje oglasa 8

7. Pogled na logičku arhitekturu sistema 8

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 9

7.1.4 HTML 9

7.1.5 Blazor Server-side 10

7.1.6 MySQL 10

8. Pogled na procese 10

8.1 Procesi 10

8.1.1 Web browser 11

8.1.2 Web server 11

8.1.3 MySQL Server 11

9. Pogled na raspoređivanje sistema 11

9.1 Klijent 11

9.2 Web server 11

9.3 DBMS server 11

10. Pogled na implementaciju sistema 12

10.1 Model domena 12

10.2 Šema baze podataka 12

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 14

11. Performanse 15

12. Kvalitet 15

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Web aplikacije Kućni poslovi.

# Opseg dokumenta

Dokument se odnosi na Web aplikaciju Kućni poslovi koja će biti razvijena od strane NWTeam-a. Namena sistema je efikasno brzo i lako povezivanje pružalaca i korisnika usluga u oblasti pomoći u kući.

# Reference

Spisak korišćene literature:

1. Kućni poslovi – Predlog projekta, NWTeam-Kućni poslovi-01, V1.0, 2020, NWTeam.
2. Kućni poslovi – Vizija sistema, NWTeam-Kućni poslovi-02, V1.0, 2020, NWTeam
3. Kućni poslovi – Raspored aktivnosti, NWTeam\_03\_Raspored\_aktivnosti, V1.0, 2020, NWTeam
4. Kućni poslovi – Plan realizacije projekta, NWTeam\_03\_Plan\_realizacije\_projekta-03, V1.0, 2020, NWTeam
5. Kućni poslovi – Specifikacija zahteva, NWTeam\_04\_Specifikacija\_zahteva, V1.0, 2020, NWTeam

# 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. Aplikacija Kućni poslovi će biti implementirana kao Web aplikacija zasnovana na Blazor Server-side framework-u i MySQL bazi podataka [4].
2. Klijentski deo Web aplikacije Kućni poslovi će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, kao i Firefox (Mozilla) [4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.
4. Klasifikaciju publikacija treba standardizovati sa postojećom tipologijom propisanom od strane Ministarstva za nauku.

# Pogled na slučajeve korišćenja

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

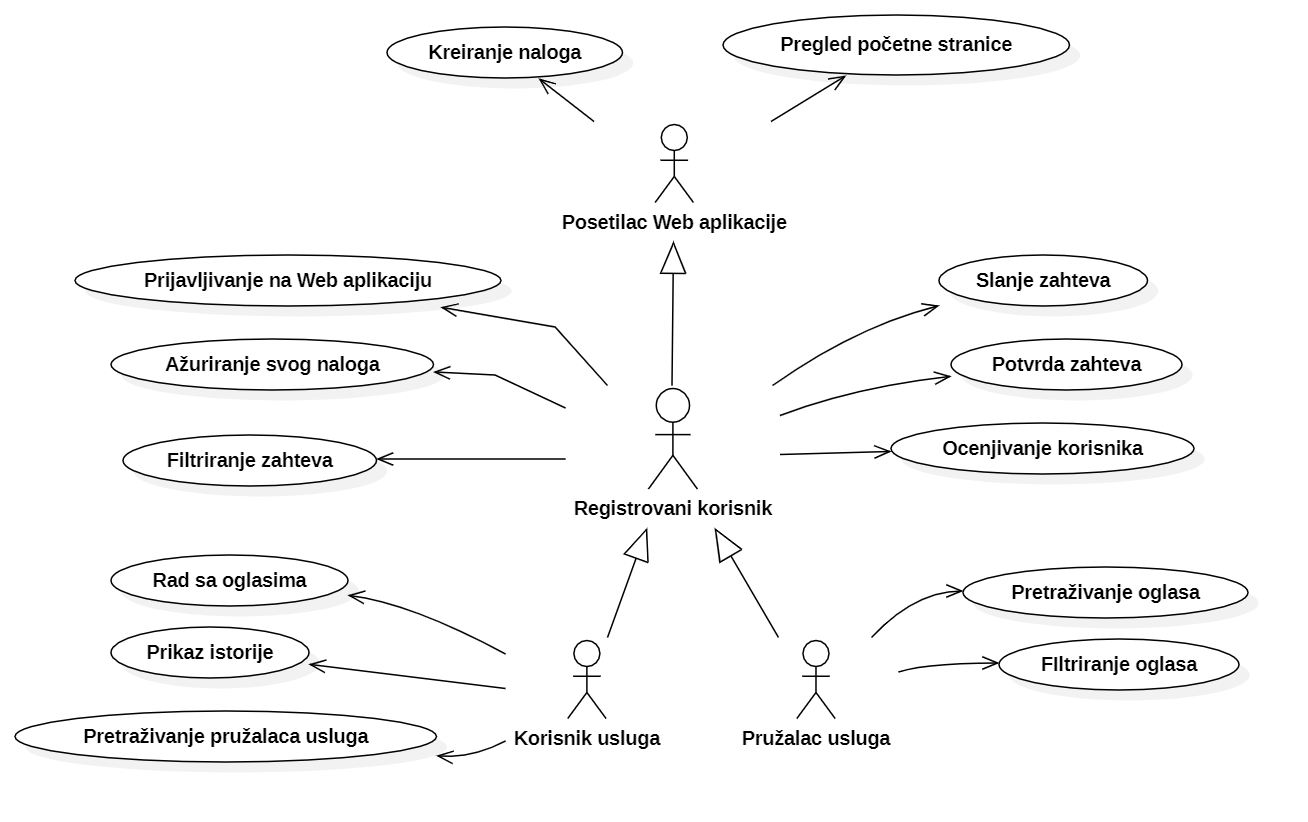
Slučajevi korišćenja Web aplikacije Kućni poslovi su:

* Kreiranje naloga
* Pregled početne stranice
* Prijavljivanje na Web aplikaciju
* Ažuriranje svog naloga
* Slanje zahteva
* Potvrda zahteva
* Filtriranje zahteva
* Ocenjivanje korisnika
* Rad sa oglasima
  + Kreiranje oglasa
  + Ažuriranje oglasa
  + Brisanje oglasa
* Prikaz istorije
  + Prikaz prethodnih oglasa
    - Prikaz detalja oglasa
  + Prikaz prethodnih saradnji
* Pretraživanje pružalaca usluga
* Pretraživanje oglasa
* Filtriranje oglasa

Ove slučajevi korišćenja mogu da iniciraju registrovani korisnici (korisnici usluga i pružaoci usluga).

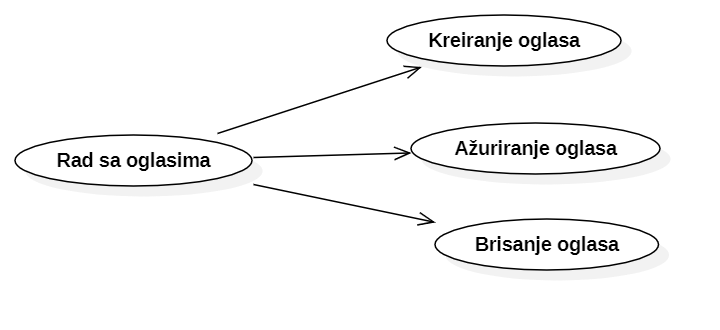
## Dijagrami slučajeva korišćenja

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



Slučajevi korišćenja *rad sa oglasima* i *prikaz istorije* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *rad sa oglasima* je prikazan na sledećoj slici:



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

## Prikaz istorije - Use Case

## Kratak opis slučajeva korišćenja

### Kreiranje naloga

Kratak opis:Kreiranje naloga korisnika radi korišćenja Web aplikacije.

Akteri:Posetilac Web aplikacije, Registrovani korisnik, Korisnik usluga, Pružalac usluga.

### Pregled početne stranice

Kratak opis:Prikaz glavne stranice sa osnovnim podacima o aplikaciji.

Akteri:Posetilac Web aplikacije, Registrovani korisnik, Korisnik usluga, Pružalac usluga.

### Prijavljivanje na Web aplikaciju

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

Akteri:Registrovani korisnik, Korisnik usluga, Pružalac usluga.

### Ažuriranje svog naloga

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

Akteri:Registrovani korisnik, Korisnik usluga, Pružalac usluga.

### Slanje zahteva

Kratak opis:Korisnik šalje zahtev drugom korisniku u vezi konkretne usluge.

Akteri:Registrovani korisnik, Korisnik usluga, Pružalac usluga.

### Potvrda zahteva

Kratak opis:Potvrda aktivnih zahteva.

Akteri:Registrovani korisnik, Korisnik usluga, Pružalac usluga.

### Filtriranje zahteva

Kratak opis: Prikaz svih primljenih i poslatih zahteva i mogućnost filitriranja.

Akteri: Registrovani korisnik, Korisnik usluga, Pružalac usluga.

### Ocenjivanje korisnika

Kratak opis:Ocenjivanje korisnika nakon izvršene ili pružane usluge.

Akteri: Registrovani korisnik, Korisnik usluga, Pružalac usluga.

### Kreiranje oglasa

Kratak opis:Kreiranje novog oglasa radi dobijanja usluge.

Akteri: Korisnik usluga

### Ažuriranje oglasa

Kratak opis: Promena informacija na trenutno aktivnom oglasu.

Akteri: Korisnik usluga

### Brisanje oglasa

Kratak opis: Brisanje oglasa iz istorije oglasa.

Akteri: Korisnik usluga

### Prikaz prethodnih oglasa

Kratak opis: Prikaz kompletne istorije korisnikovih oglasa

Akteri: Korisnik usluga

### Prikaz prethodnih saradnji

Kratak opis: Prikaz istorije korisnika sa kojima je korisnik sarađivao.

Akteri: Korisnik usluga

### Pretraživanje pružalaca usluga

Kratak opis: Pretraživanje pružalaca usluga radi slanja zahteva

Akteri: Korisnik usluga

### Pretraživanje oglasa

Kratak opis: Pretraživanje oglasa i informacija o njima.

Akteri: Pružalac usluga

### Filtriranje oglasa

Kratak opis: Filtriranje oglasa po kritrijumima

Akteri:Pružalac usluga

# 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.

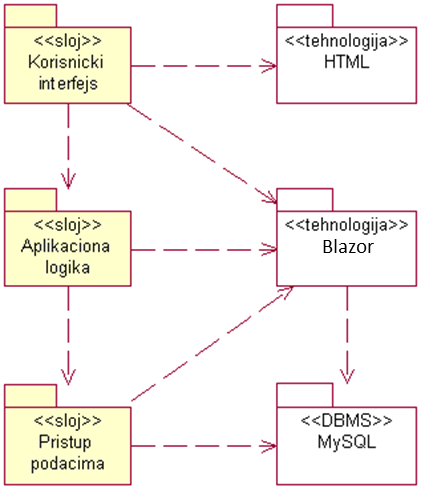
Logički pogled na Web aplikaciju Kućni poslovi obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice (HTML i JavaScript) 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 C# kod u Blazor Server-side framework-u zadužen za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži C# kod na serveru 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 sadržani svi multimedijalni sadržaji, HTML i JavaScript elementi koji se generišu na stranici.

Sloj korisničkog interfejsa preko SignalR veze komunicira sa serverom, na kome se UI ažurira, obrađuju svi događaji i svi JS pozivi.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Web aplikacije Kućni poslovi. Sadrži C# kod u Blazor Server-side framework-u, koji realizuje funkcionalnost karakterističnu za domen primene Web aplikacije i uspostavlja vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i ASP.NET Core biblioteka.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži C# kod u Blazor Server-side framework-u zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MySQL bazi podataka.

Ovaj sloj delimično zavisi od sloja Aplikacione logike, u kome su predefinisane osnovne metode za pristup podacima u bazi, kao i od ASP.NET Core biblioteka i MySQL baze 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.

### Blazor Server-side

tehnologija

Tehnologija Blazor omogućava kompletno izvršavanje aplikacije, napisane isključivo u C#-u, korišćenjem ASP.NET Core, na serveru preko SignalR konekcije sa klijentom.

### MySQL

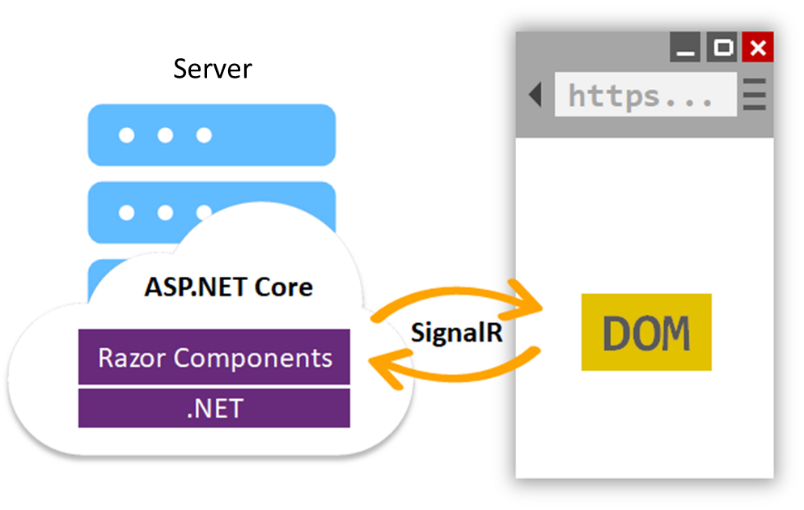
DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Web aplikacije Kućni poslovi.

# 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.

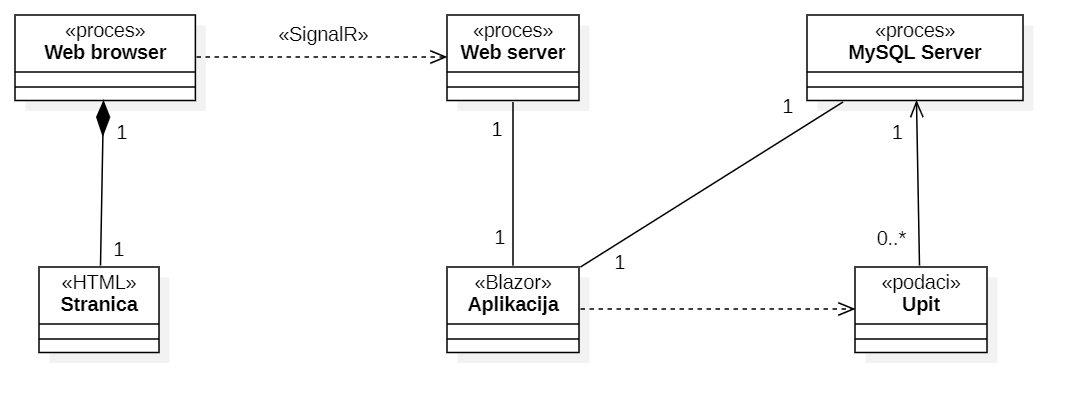
Web aplikacije zasnovane na Blazor-u imaju jednostavan procesni model koji je u potpunosti pod kontrolom Blazor servera. Jedan od uslova za korišćenje Web aplikacije je stalna internet konekcija. Prilikom otvaranja stranice na klijentskoj strani, kompletna logika aplikacije ostaje na serveru i nevidljiva je klijentu, bez pamćenja velike količine podataka na klijentu (kao što se može videti na slici ispod).



Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Web aplikacije Kućni poslovi.

## Procesi

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



### Web browser

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

Web browser zavisi od Web servera. Za svakog klijenta, koji se povezuje na Web server, kreira se posebna instanca SignalR konekcije, preko koje se vodi kompletna komunikacija klijenta i aplikacije na Web serveru.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web browser preko SignalR konekcija. Na Web serveru se pokreće i izvršava aplikacija i preko SignalR konekcija se obavlja komunikacija sa klijentima, koja rezultira otvaranjem odgovarajuće Web stranice u Web browseru.

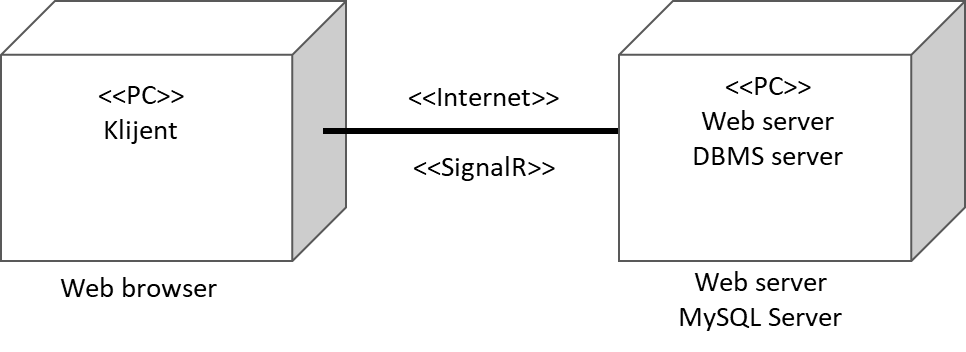
### MySQL Server

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 aplikaciji na Web serveru, koja ih postavlja.

# 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 Web aplikacije Kućni poslovi.



## Klijent

Pristup Web aplikaciji Kućni poslovi se obavlja preko klijentskih računara na kojima se izvršava Web browser. Za povezivanje između klijenta i Web servera koristi se Internet infrastruktura i SignalR konekcija, tako da nema ograničenja u pogledu lokacije klijenta.

## Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta, odnosno svoje SignalR konekcije. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru se izvršava Blazor aplikacija. U najopštijoj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

## DBMS server

Isti računar na kome se izvršava Web server je i DBMS server, na kome se izvršava MySQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka.

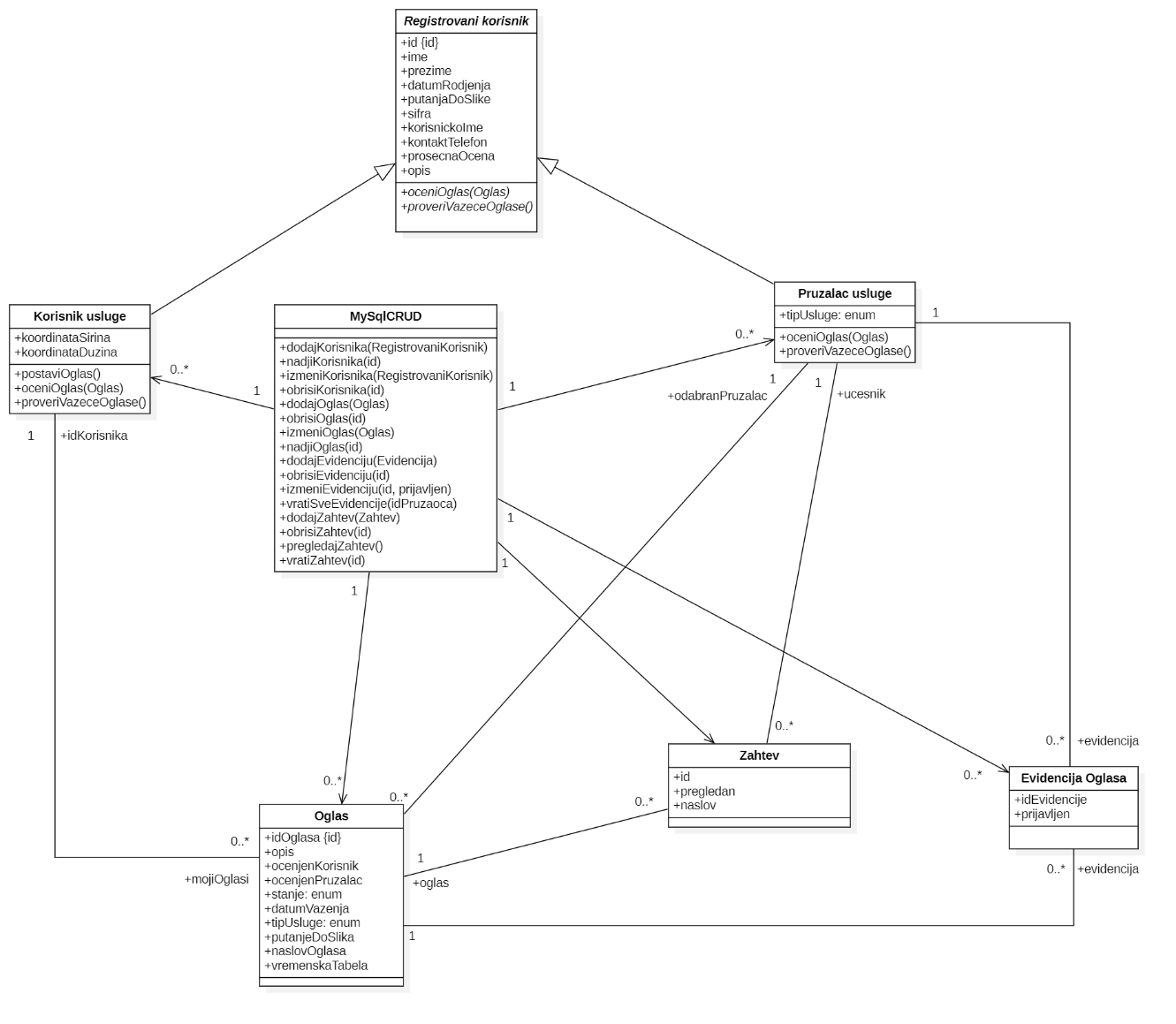
# Pogled na implementaciju sistema

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

## Model domena

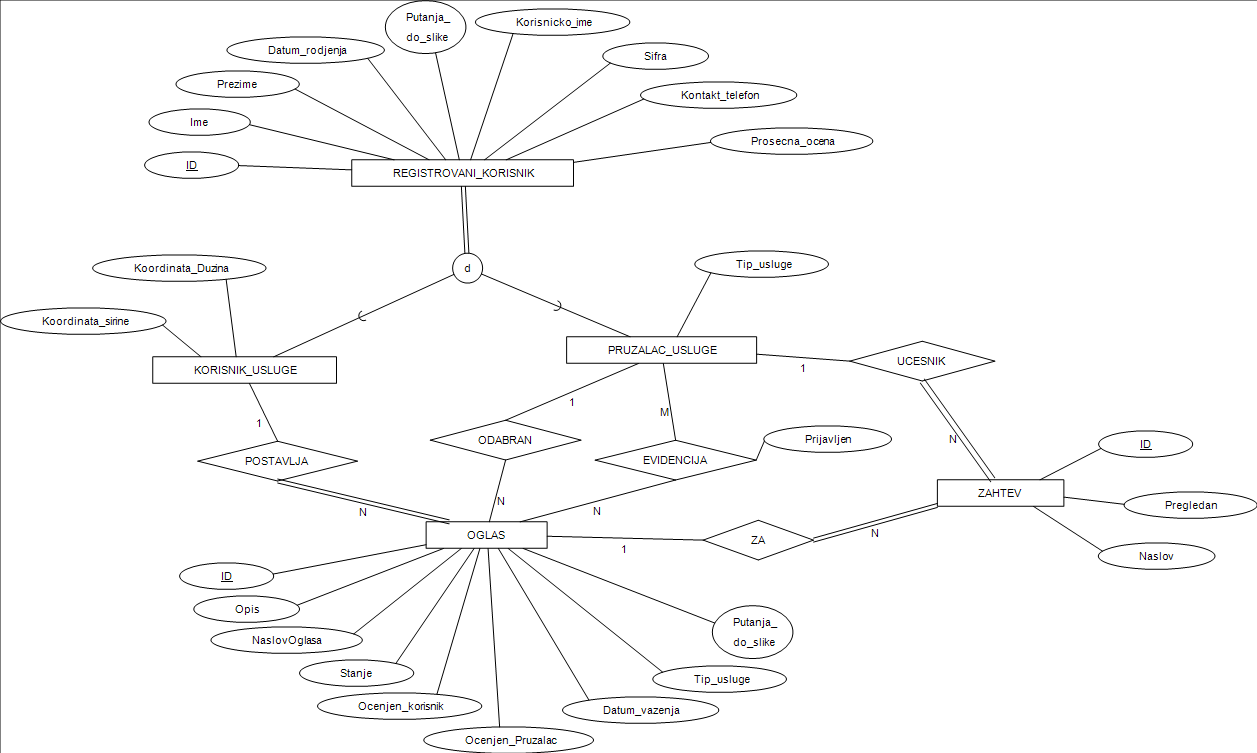
Model domena za koji se Web aplikacija Kućni poslovi projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



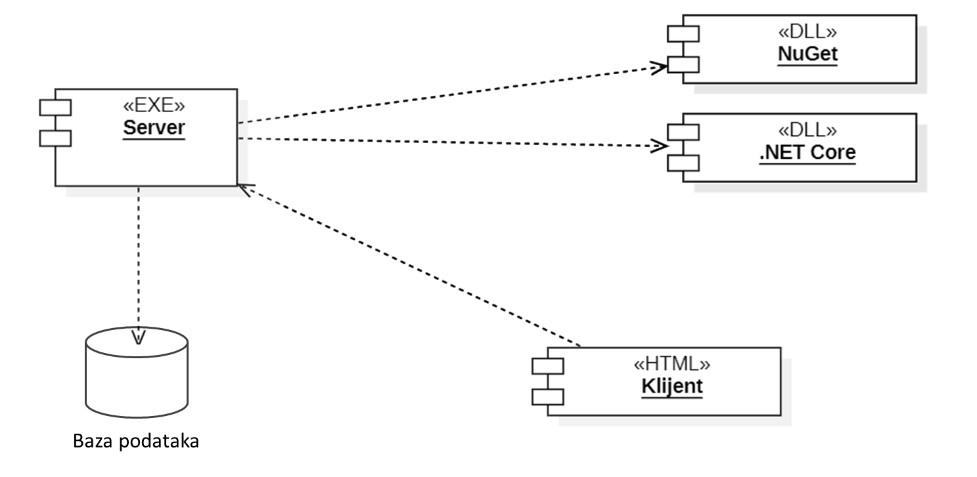
## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem EER dijagramu. Baza podataka i dijagram su kreirani korišćenjem fakultetskog softvera koji se koristi na predmetu Sistemi baza podataka *SharpEER*, dok je migracija na MySQL obavljena pomoću paketa *Microsoft.EntityFrameworkCore.SqlServer*.



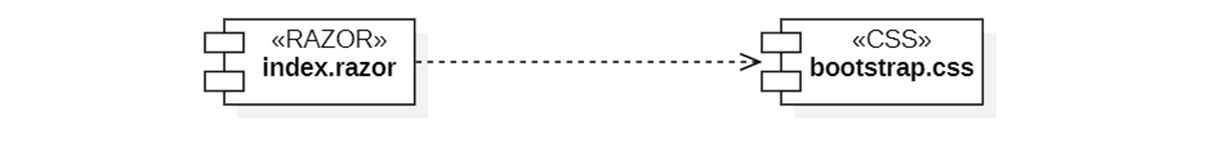
## Komponente sistema

Komponente sistema Web aplikacije Kućni poslovi su stranice (Blazor komponente - *.razor*) čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa.



### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

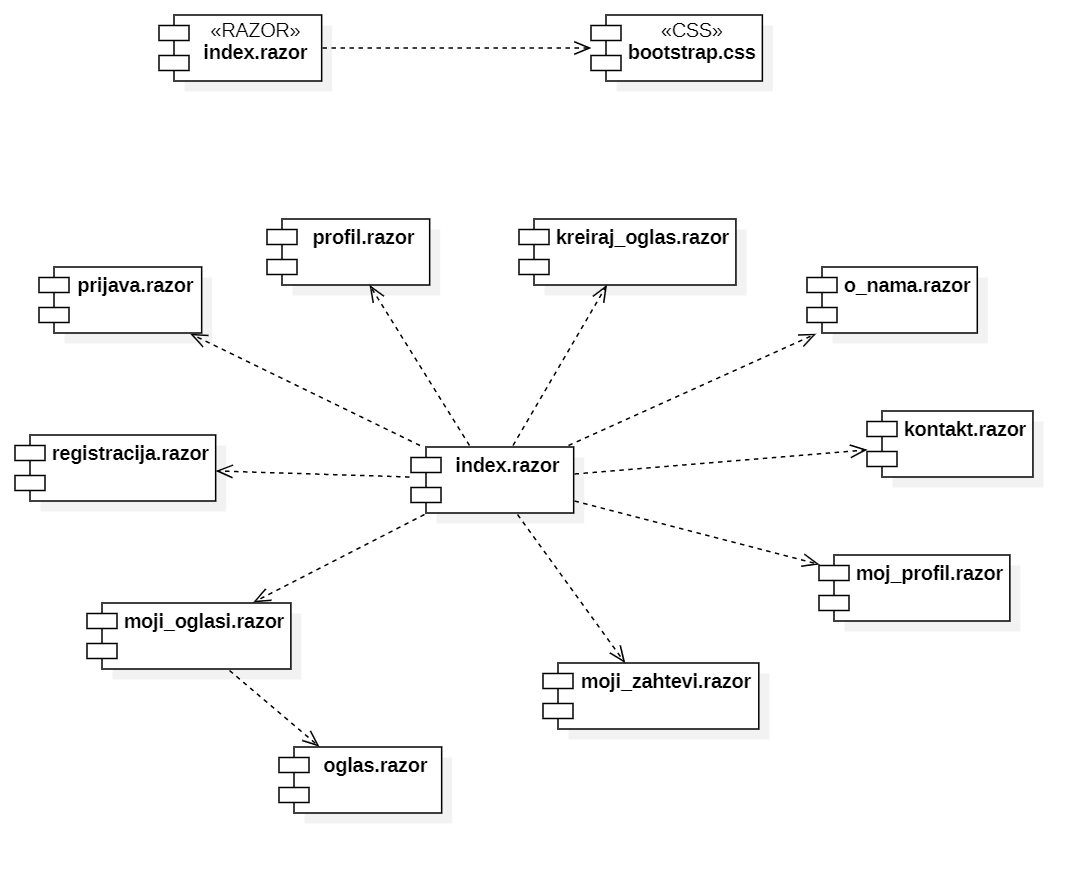


Komponenta **index.razor** implementira stranicu Web aplikacije čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu (u zavisnosti od toga da li je korisnik registrovan ili ne).

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

### Komponente aplikacione logike

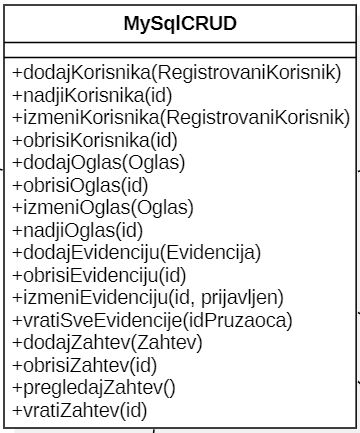
Komponente koje realizuju domen problema se uključuju isključivo preko **index.razor** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru klase **mySqlCRUD.cs**. Pomenutoj klasi pristupamo u aplikaciji preko odgovarajućih Blazor komponenti.

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



Navedene funkcije obavljaju sledeće zadatke:

* dodajKorisnika - kreira novog korisnika u bazi sa atributima prosleđenog objekta
* nadjiKorisnika - nalazi i vraća korisnika iz baze na osnovu prosleđenog ID-a
* izmeniKorisnika - nalazi i menja atribute korisnika iz baze
* obrisiKorisnika - nalazi i briše korisnika sa datim ID
* dodajOglas - kreira novi oglas u bazi sa atributima prosleđenog objekta
* nadjiOglas - nalazi i vraća oglas iz baze na osnovu prosleđenog ID-a
* izmeniOglas - nalazi i menja atribute oglasa iz baze
* obrisiOglas - nalazi i briše oglas sa datim ID-em
* dodajEvidenciju - kreira novu evidenciju u bazi sa atributima prosleđenog objekta
* vratiSveEvidencije - nalazi i vraća sve evidencije iz baze, koje imaju prosleđeni ID pružaoca usluge
* izmeniEvidenciju - nalazi evidenciju sa datim ID-em u bazi i menja atribut prijavljen (boolean)
* obrisiEvidenciju - nalazi i briše evidenciju sa datim ID-em
* dodajZahtev - kreira novi zahtev u bazi sa atributima prosleđenog objekta
* vratiZahtev - nalazi i vraća zahtev iz baze na osnovu prosleđenog ID-a
* pregledajZahtev - nalazi zahtev u bazi i postavlja vrednost atributa pregledan na *true*
* obrisiZahtev - nalazi i briše zahtev sa datim ID

# 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 5000 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. Web aplikacija Kućni poslovi 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.