Gde je moj ljubimac?

Web aplikacija za pronalazak kućnih ljubimaca

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 8.5.2021. | 1.0 | Inicijalna verzija | Voja, Petra, Milica |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[**Cilj dokumenta**](#_heading=h.gjdgxs) **5**

[**Opseg dokumenta**](#_heading=h.30j0zll) **5**

[**Reference**](#_heading=h.1fob9te) **5**

[**Predstavljanje arhitekture**](#_heading=h.2et92p0) **5**

[**Ciljevi i ograničenja arhitekture**](#_heading=h.tyjcwt) **5**

[**Pogled na slučajeve korišćenja**](#_heading=h.3dy6vkm) **5**

[Dijagrami slučajeva korišćenja](#_heading=h.2bn6wsx) 6

[Kratak opis slučajeva korišćenja](#_heading=h.qsh70q) 7

[Pregled osnovnih informacija o nameni aplikacije](#_heading=h.1pxezwc) 7

[Pregled oglasa po tipu](#_heading=h.49x2ik5) 7

[Pregled oglasa po specijalnom zahtevu](#_heading=h.2p2csry) 7

[Pregled podataka o određenom oglasu](#_heading=h.147n2zr) 7

[Pregled podataka o određenom registrovanom korisniku](#_heading=h.3o7alnk) 7

[Pregled oglasa po autoru](#_heading=h.23ckvvd) 7

[Prijavljivanje](#_heading=h.ihv636) 7

[Ažuriranje podataka o korisniku](#_heading=h.32hioqz) 8

[Dodavanje novog oglasa](#_heading=h.1hmsyys) 8

[Brisanje postojećeg oglasa](#_heading=h.41mghml) 8

[Ažuriranje postojećeg oglasa](#_heading=h.2grqrue) 8

[Kreiranje novog korisnika](#_heading=h.vx1227) 8

[Brisanje postojećeg člana](#_heading=h.3fwokq0) 8

[**Pogled na logičku arhitekturu sistema**](#_heading=h.28h4qwu) **8**

[Pregled arhitekture – organizacija paketa i podsistema u slojeve](#_heading=h.nmf14n) 9

[Korisnički interfejs](#_heading=h.37m2jsg) 9

[Aplikaciona logika](#_heading=h.1mrcu09) 9

[Pristup podacima](#_heading=h.46r0co2) 9

[React](#_heading=h.2lwamvv) 9

[C#](#_heading=h.111kx3o) 9

[SQLite](#_heading=h.3l18frh) 10

[**Pogled na procese**](#_heading=h.206ipza) **10**

[Procesi](#_heading=h.4k668n3) 10

[Web čitač](#_heading=h.2zbgiuw) 10

[Web server](#_heading=h.1egqt2p) 10

[ASP-CGI](#_heading=h.3ygebqi) 10

[SQLite Server](#_heading=h.2dlolyb) 10

[**Pogled na raspoređivanje sistema**](#_heading=h.sqyw64) **11**

[Klijent](#_heading=h.3cqmetx) 11

[Web server](#_heading=h.1rvwp1q) 11

[DBMS server](#_heading=h.4bvk7pj) 11

[**Pogled na implementaciju sistema**](#_heading=h.2r0uhxc) **11**

[Model domena](#_heading=h.1664s55) 11

[Šema baze podataka](#_heading=h.3q5sasy) 12

[Komponente sistema](#_heading=h.25b2l0r) 12

[Komponente korisničkog interfejsa](#_heading=h.kgcv8k) 12

[Komponente aplikacione logike](#_heading=h.34g0dwd) 13

[Komponente za pristup podacima](#_heading=h.1jlao46) 13

[**Performanse**](#_heading=h.43ky6rz) **14**

[**Kvalitet**](#_heading=h.2iq8gzs) **14**

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Gde je moj ljubimac? aplikacije.

# Opseg dokumenta

Dokument se odnosi na Gde je moj ljubimac? portal koji će biti razvijen od strane Vinopije.net-a. Gde je moj ljubimac? je web aplikacija za pronalazak kućnih ljubimaca. Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja vezanih za rad jedne aplikacije na principu društvene mreže.

# Reference

Spisak korišćene literature:

1. Gde je moj ljubimac? – Predlog projekta, V1.0, 2021, Vinopije.net
2. Gde je moj ljubimac?– Planirani raspored aktivnosti na projektu, V1.0, 2021, Vinopije.net.
3. Gde je moj ljubimac?– Plan realizacije projekta, V1.0, 2021, Vinopije.net.
4. Gde je moj ljubimac?– Vizija sistema, V1.0, 2021, Vinopije.net.
5. Gde je moj ljubimac?– Specifikacija zahteva, V1.0, 2021, Vinopije.net.

# 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. Gde je moj ljubimac? aplikacija će biti implementirana kao Web aplikacija zasnovana na ASP.NET jeziku i SQLite - promeniti u prošlim dokumentima bazi podataka [4].
2. Klijentski deo Gde je moj ljubimac? aplikacije 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.

# Pogled na slučajeve korišćenja

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

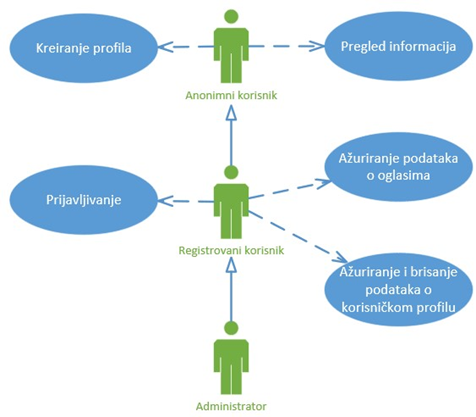
Slučajevi korišćenja Gde je moj ljubimac? aplikacije su:

* *Kreiranje profila*
* *Pregled informacija*
  + *Pregled osnovnih podataka o portalu*
  + *Pregled podataka o registrovanom korisniku*
    - Pregled oglasa po autoru
  + *Pregled podataka o oglasima*
    - Pregled oglasa po tipu
    - Pregled oglasa po specijalnom zahtevu
* *Prijavljivanje*
* *Ažuriranje i brisanje podataka o korisničkom profilu*
  + Ažuriranje registrovanog korisnika
  + Brisanje postojeće registrovanog korisnika
* *Ažuriranje podataka o oglasima*
  + Dodavanje novog oglasa
  + Ažuriranje oglasa
  + Brisanje postojećeg oglasa

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

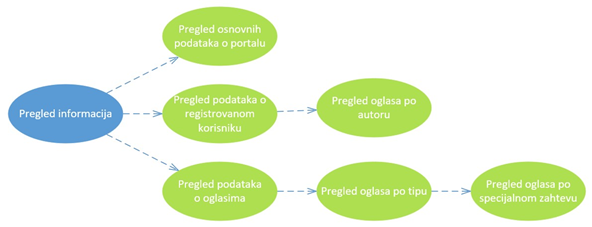
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Gde je moj ljubimac? aplikacije prikazan je na sledećoj slici:

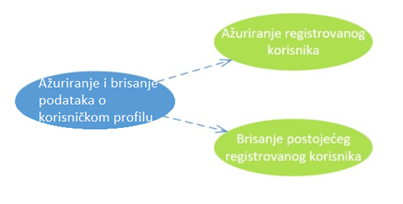


Slučajevi korišćenja *pregled informacija*, *ažuriranje i brisanje podataka o korisničkom profilu* i *ažuriranje podataka o oglasima* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

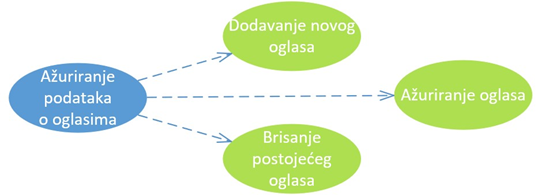
Detaljni UML dijagram za slučaj korišćenja *pregled informacija* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *ažuriranje i brisanje podataka o korisničkom profilu* je prikazan na sledećoj slici:



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



## Kratak opis slučajeva korišćenja

### Pregled osnovnih informacija o nameni aplikacije

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

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

### Pregled oglasa po tipu

Kratak opis: Prikaz stranice sa oglasima razvrstanim po tipu.

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

### Pregled oglasa po specijalnom zahtevu

Kratak opis: Prikaz stranice sa oglasima razvrstanim po specijalnom zahtevu.

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

### Pregled podataka o određenom oglasu

Kratak opis: Prikaz stranice sa informacijama o određenom oglasu.

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

### Pregled podataka o određenom registrovanom korisniku

Kratak opis: Prikaz stranice sa informacijama o određenom registrovanom korisniku.

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

### Pregled oglasa po autoru

Kratak opis: Prikaz stranice sa spiskom oglasa čiji je autor određeni registrovani korisnik.

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

### Prijavljivanje

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

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Ažuriranje podataka o korisniku

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

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Dodavanje novog oglasa

Kratak opis: Dodavanje novog oglasa od strane autora koji je registrovani korisnik.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Brisanje postojećeg oglasa

Kratak opis: Biranje postojećeg oglasa od strane registrovanog korisnika koji je taj oglas uneo ili administratora.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Ažuriranje postojećeg oglasa

Kratak opis: Izmena podataka o objavljenom oglasu.

Akteri koji iniciraju slučaj korišćenja: Registrovani korisnik, Administrator.

### Kreiranje novog korisnika

Kratak opis: Kreiranje korisničkog naloga za novog registrovanog korisnika.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Administrator.

### Brisanje postojećeg člana

Kratak opis: Brisanje korisničkog naloga i oglasa postojećeg registrovanog korisnika.

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.

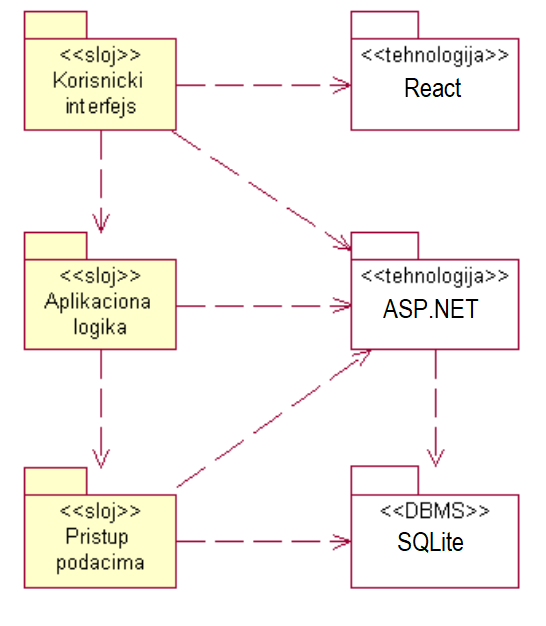
Logički pogled na Gde je moj ljubimac? portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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 C# skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži C# 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 portala. U njemu su sadržane sve React, multimedijalni sadržaji i C# skripte koje generišu React stranice preko kojih korisnici komuniciraju sa sistemom.

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

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Gde je moj ljubimac? portala. Sadrži C# 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 C# paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži C# skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u SQLite bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa C# i SQLite baza podataka.

### React

tehnologija

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

### C#

tehnologija

Tehnologija C# 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.

### SQLite

DBMS

SQLite predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Gde je moj ljubimac? portala.

# Pogled na procese

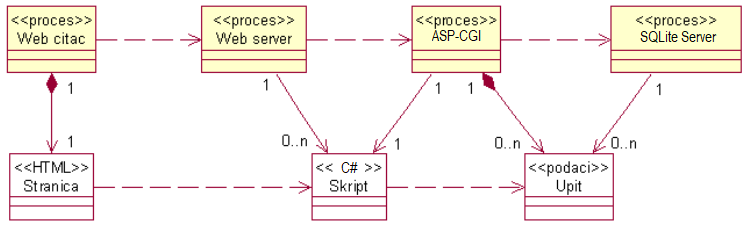
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na C#-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta C# 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 Gde je moj ljubimac? portala kao Web aplikacije.

## Procesi

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

### ASP-CGI

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

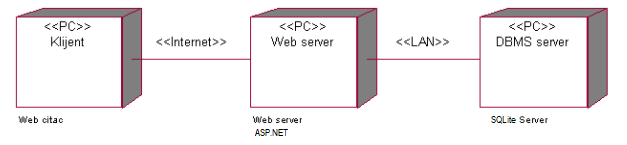
### SQLite Server

SQLite Server je proces koji izvršava funkcionalnost SQLite 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 Gde je moj ljubimac? portala.



## Klijent

Pristup Gde je moj ljubimac? 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 ASP.NET-a koji vrše obradu zadatih C# 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 SQLite 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).

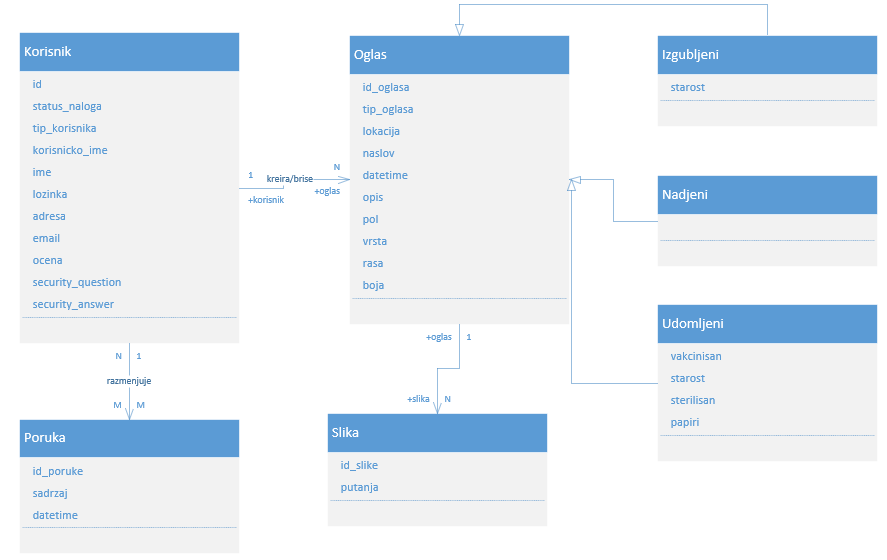
# Pogled na implementaciju sistema

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

## Model domena

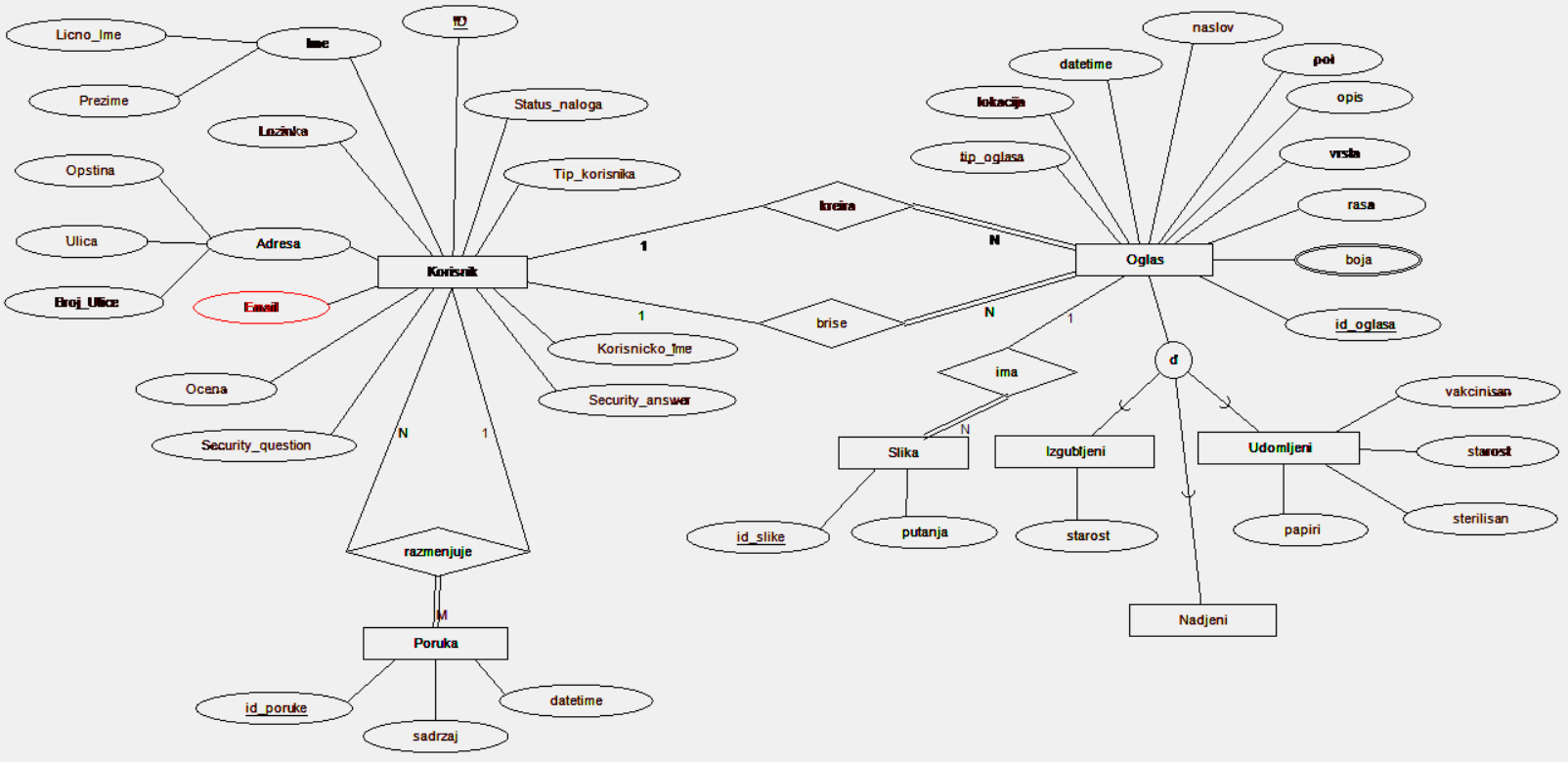
Model domena za koji se Gde je moj ljubimac? portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka je kreirana korišćenjem *SQLite-a*, a dijagram korišćenjem #EER alata.



## Komponente sistema

Komponente sistema Gde je moj ljubimac? portala su C# 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 C# 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:

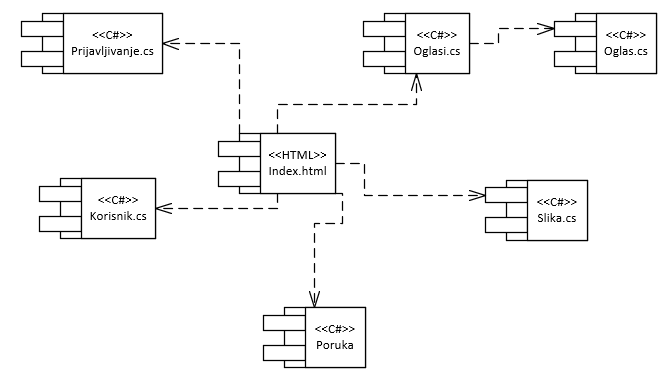


Komponenta **index.html** je implementira stranicu portala čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

Komponenta **main.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.html** 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:

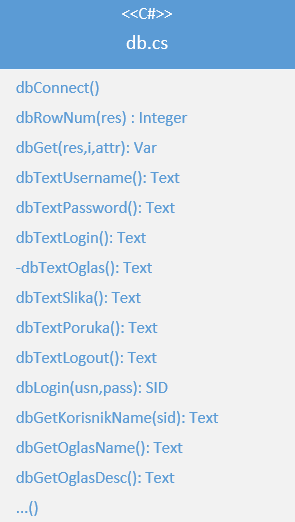


### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru C# skripta **db.cs**. Pomenuti skript se (uključuje na početku index.html-a,) povezuje sa klijentskim delom tako da su sve funkcije za pristup podacima dostupne svim komponentama.



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



Navedene funkcije obavljaju sledeće zadatke:

* dbConnect – povezivanje na bazu podataka, poziva se u samom skriptu
* dbRowNum – vraća broj slogova sadržanih u rezultatu upita
* dbGet – vraća vrednost polja iz rezultata upita
* dbText... – vraća vrednost pojedinih konstanti
* dbLogin – loguje korisnika i vraća SID
* dbGetKorisnikName – vraća ime člana po jeziku i SID-u
* dbGetOglasName – vraća naslov oglasa
* dbGetOglasDesc – vraća opis oglasa
* ... – ostale funkcije

# 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. Gde je moj ljubimac? 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.