Lako do stana

Portal za oglašavanje stanova i traženje cimera

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 26.04.2020. | 1.0 | Inicijalna verzija | Nenad, Nikola |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 4](#_Toc39095225)

[2. Opseg dokumenta 4](#_Toc39095226)

[3. Reference 4](#_Toc39095227)

[4. Predstavljanje arhitekture 4](#_Toc39095228)

[5. Ciljevi i ograničenja arhitekture 4](#_Toc39095229)

[6. Pogled na slučajeve korišćenja 4](#_Toc39095230)

[6.1 Dijagrami slučajeva korišćenja 5](#_Toc39095231)

[6.2 Kratak opis slučajeva korišćenja 7](#_Toc39095232)

[6.2.1 Izbor teme 7](#_Toc39095233)

[6.2.2 Pregled osnovnih informacija o funkcionalnostima koje pruža web sajt 7](#_Toc39095234)

[6.2.3 Pregled spiska korisnika 7](#_Toc39095235)

[6.2.4 Pregled podataka o određenom korisniku 7](#_Toc39095236)

[6.2.5 Pregled oglasa po nazivu oglašivača 7](#_Toc39095237)

[6.2.6 Pregled oglasa po tipu 7](#_Toc39095238)

[6.2.7 Pregled oglasa po datumu objavljivanja 7](#_Toc39095239)

[6.2.8 Pregled interakcije sa drugim korisnicima 7](#_Toc39095240)

[6.2.9 Prijavljivanje 7](#_Toc39095241)

[6.2.10 Ažuriranje podataka o sebi 7](#_Toc39095242)

[6.2.11 Registracija 7](#_Toc39095243)

[6.2.12 Interakcija sa drugim korisnicima 7](#_Toc39095244)

[6.2.13 Dodavanje novog oglasa 7](#_Toc39095245)

[6.2.14 Brisanje postojećeg oglasa 8](#_Toc39095246)

[6.2.15 Izmena postojećeg člana 8](#_Toc39095247)

[6.2.16 Pregled spiska korisnika 8](#_Toc39095248)

[6.2.17 Kreiranje novog korisnika 8](#_Toc39095249)

[6.2.18 Brisanje postojećeg korisnika 8](#_Toc39095250)

[7. Pogled na logičku arhitekturu sistema 8](#_Toc39095251)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 9](#_Toc39095252)

[7.1.1 Korisnički interfejs 9](#_Toc39095253)

[7.1.2 Aplikaciona logika 9](#_Toc39095254)

[7.1.3 Pristup podacima 9](#_Toc39095255)

[7.1.4 HTML 9](#_Toc39095256)

[7.1.5 ASP.NET 9](#_Toc39095257)

[7.1.6 SQL Server 9](#_Toc39095258)

[8. Pogled na procese 10](#_Toc39095259)

[8.1 Procesi 10](#_Toc39095260)

[8.1.1 Web čitač 10](#_Toc39095261)

[8.1.2 Web server 10](#_Toc39095262)

[8.1.3 SQL Server 10](#_Toc39095263)

[9. Pogled na raspoređivanje sistema 10](#_Toc39095264)

[9.1 Klijent 11](#_Toc39095265)

[9.2 Web server 11](#_Toc39095266)

[9.3 DBMS server 11](#_Toc39095267)

[10. Pogled na implementaciju sistema 11](#_Toc39095268)

[10.1 Model domena 11](#_Toc39095269)

[10.2 Šema baze podataka 12](#_Toc39095270)

[10.3 Komponente sistema 13](#_Toc39095271)

[10.3.1 Komponente korisničkog interfejsa 13](#_Toc39095272)

[10.3.2 Komponente aplikacione logike 14](#_Toc39095273)

[10.3.3 Komponente za pristup podacima 15](#_Toc39095274)

[11. Performanse 15](#_Toc39095275)

[12. Kvalitet 15](#_Toc39095276)

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Lako do stana portala.

# Opseg dokumenta

Dokument se odnosi na Lako do stana portal koji će biti razvijen od strane tima Podstanari. Namena sistema je efikasno i lako postavljanje oglasa za izdaju stanova i kuća, postavljanje oglasa za pronalazak cimera, brza i jednostavna pretraga postavljenih oglasa.

# Reference

Spisak korišćene literature:

1. Lako do stana – Predlog projekta, SWE-Lako od stana-01, V1.0, 2020, Podstanari.
2. Lako do stana – Planirani raspored aktivnosti na projektu, V1.0, 2020, Podstanari.
3. Lako do stana – Plan realizacije projekta, V1.0, 2020, Podstanari.
4. Lako do stana – Vizija sistema, V1.0, 2020, Podstanari.
5. Lako do stana – Specifikacija zahteva, V1.0, 2020, Podstanari.

# 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. Lako do stana portal će biti implementiran kao Web aplikacija zasnovana na ASP.NET jeziku i SQL Server bazi podataka [4].
2. Klijentski deo Lako do stana portala će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Google Chrome, Microsoft Edge,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].

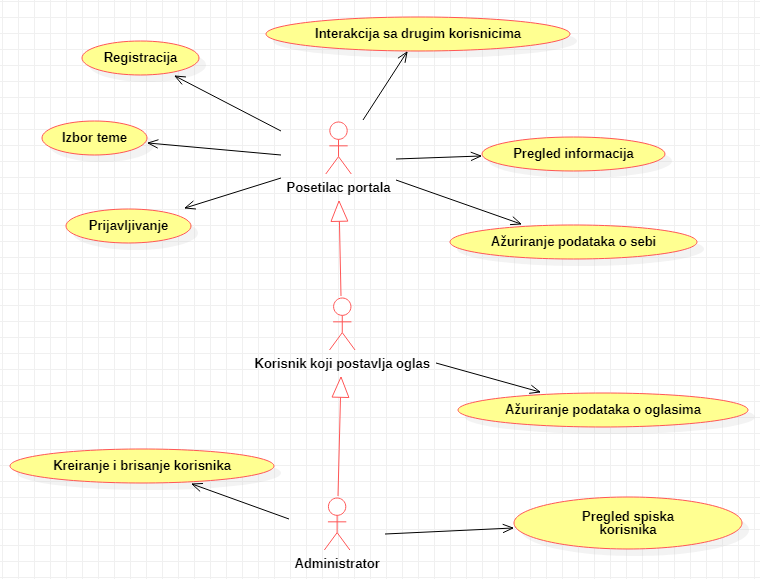
Slučajevi korišćenja Lako do stana portala su:

* Izbor teme
* *Pregled informacija*
  + Pregled osnovnih informacija o funkcionalnostima koje pruža web sajt
  + *Pregled podataka o korisnicima*
    - Pregled podataka o određenom korisniku
  + *Pregled podataka o oglasima*
    - Pregled oglasa po nazivu oglašivača
    - Pregled oglasa po datumu objavljivanja
    - Pregled pglasa po tipu
  + Pregled interakcije sa drugim korisnicima
* Prijavljivanje
* Ažuriranje podataka o sebi
* Registracija
* Interakcija sa drugim korisnicima
* *Ažuriranje podataka o oglasima*
  + Dodavanje novog oglasa
  + Brisanje postojećeg oglasa
  + Izmena postojećeg oglasa
* *Kreiranje i brisanje korisnika*
  + Kreiranje novog korisnika
  + Brisanje postojećeg korisnika
* Pregled spiska korisnika

Ove slučajevi korišćenja mogu da iniciraju posetilac portala, korisnik koji postavlja oglas ili administrator.

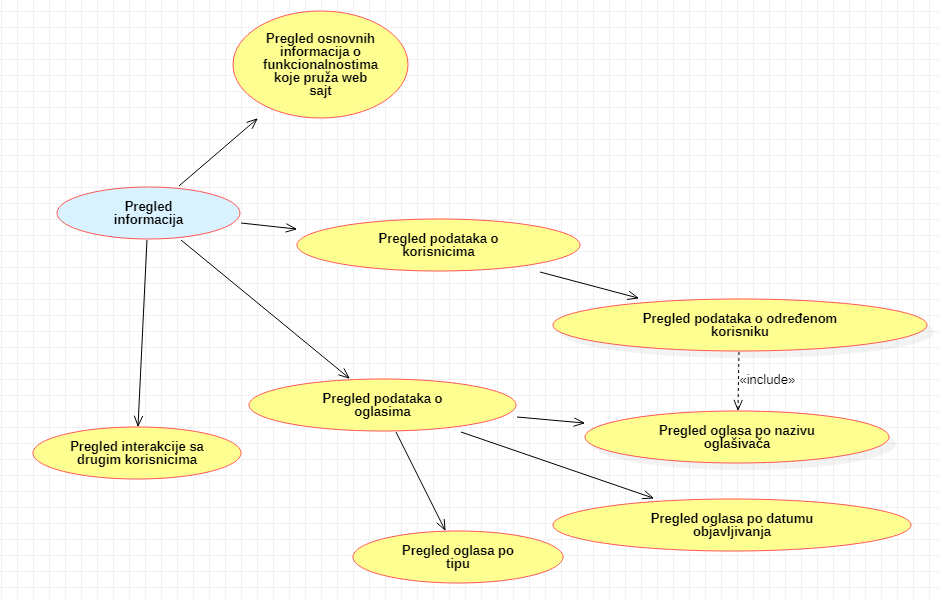
## Dijagrami slučajeva korišćenja

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

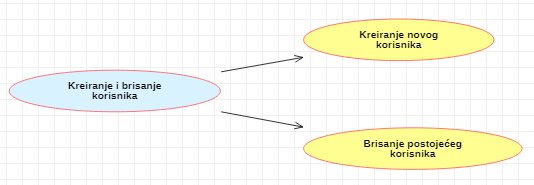


Slučajevi korišćenja *pregled informacija*, *kreiranje i brisanje članova* i *ažuriranje podataka o oglasima* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

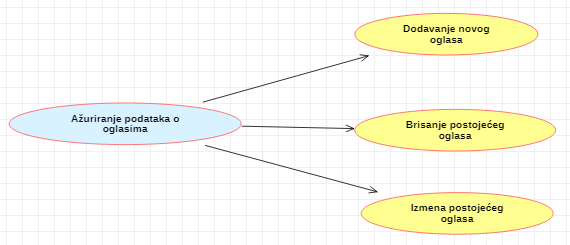
Detaljni UML dijagram za slučaj korišćenja *pregled informacija* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *kreiranje i brisanje članova* je prikazan na sledećoj slici:



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



## Kratak opis slučajeva korišćenja

### Izbor teme

Kratak opis: Izbor između svetle i tamne teme za prikaz informacija na portalu.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Pregled osnovnih informacija o funkcionalnostima koje pruža web sajt

Kratak opis: Prikaz stranice portala sa osnovnim podacima o portalu i tipovima korisnićkih naloga.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Pregled spiska korisnika

Kratak opis: Prikaz stranice portala sa spiskom registrovanih korisnika.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Pregled podataka o određenom korisniku

Kratak opis: Prikaz stranice sa informacijama o određenom korisniku.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Pregled oglasa po nazivu oglašivača

Kratak opis: Prikaz stranice sa spiskom oglasa čiji je vlasnik uneti oglašivač.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Pregled oglasa po tipu

Kratak opis: Prikaz stranice sa oglasima razvrstanim po tipu.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Pregled oglasa po datumu objavljivanja

Kratak opis Prikaz stranice sa oglasima razvrstanim po određenom datumu.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Pregled interakcije sa drugim korisnicima

Kratak opis: Prikaz stranice za interakciju sa određenim korisnikom.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Prijavljivanje

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

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Ažuriranje podataka o sebi

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

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Registracija

Kratak opis: Registracija korisnika na portal u cilju pristupa specifičnim funkcijama koje zahtevaju registraciju.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas.

### Interakcija sa drugim korisnicima

Kratak opis: Prijavljeni korisnik interaguje sa drugim korisnikom kroz prosleđivanje poruka.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, Korisnik koji postavlja oglas, Administrator.

### Dodavanje novog oglasa

Kratak opis: Dodavanje novog oglasa od strane korisnika koji postavlja oglase.

Akteri koji iniciraju slučaj korišćenja: Korisnik koji postavlja oglas, Administrator.

### Brisanje postojećeg oglasa

Kratak opis: Brisanje postojećeg oglasa od strane korisnika koji je taj oglas postavio.

Akteri koji iniciraju slučaj korišćenja: Korisnik koji postavlja oglas, Administrator.

### Izmena postojećeg člana

Kratak opis: Izmena postojećeg oglasa od strane korisnika koji je taj oglas postavio.

Akteri koji iniciraju slučaj korišćenja: Korisnik koji postavlja oglas, Administrator.

### Pregled spiska korisnika

Kratak opis: Prikaz stranice portala sa spiskom registrovanih korisnika.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Kreiranje novog korisnika

Kratak opis: Kreiranje korisničkog naloga za novog korisnika web sajta.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje postojećeg 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.

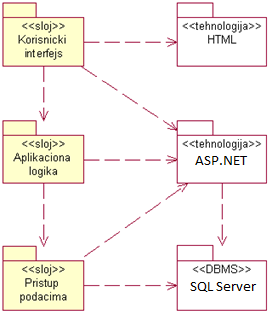
Logički pogled na Lako do stana portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Razor Web stranice, JavaScript 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# kod zadužen za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži C# kod 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 stranice preko kojih korisnici komuniciraju sa sistemom, multimedijalni sadržaji i JavaScript skripte.

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

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Lako do stana portala. Sadrži C# izvorne fajlove koji 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.

### Pristup podacima

sloj

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

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od C# koda i SQL Server 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.

### ASP.NET

tehnologija

Tehnologija ASP.NET-a obezbeđuje mehanizam za pisanje C# koda koji omogućava kreiranje modela, realizaciju funkcionalnosti i pristup bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka i HTML-a koji realizuje korisnički interfejs.

### SQL Server

DBMS

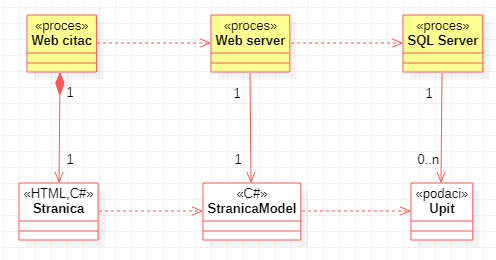
SQL Server predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Lako do stana portala.

# Pogled na procese

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

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Lako do stana portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na ASP.NET-u i SQL Server 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. Takođe je zadužen za obavljanje funkcionalnosti koje su definisane u modelima Razor stranica i za generisanje stranica koje treba vratiti web čitaču kao odgovor.

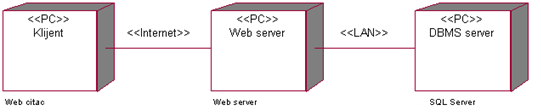
### SQL Server

SQL Server je proces koji izvršava funkcionalnost 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 Lako do stana portala.



## Klijent

Pristup Lako do stana portalu 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. Na ovom računaru se izvršava osnovni proces koji realizuje funkcionalnost Web servera. 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 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).

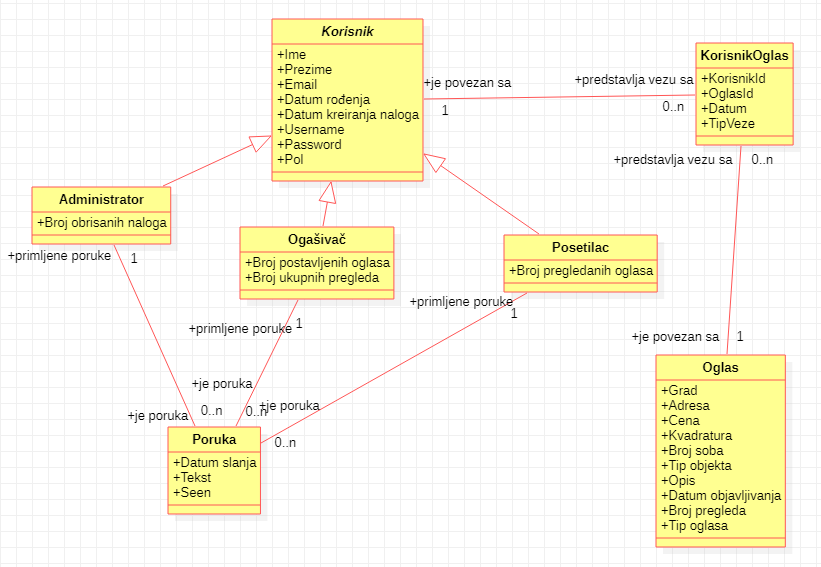
# Pogled na implementaciju sistema

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

## Model domena

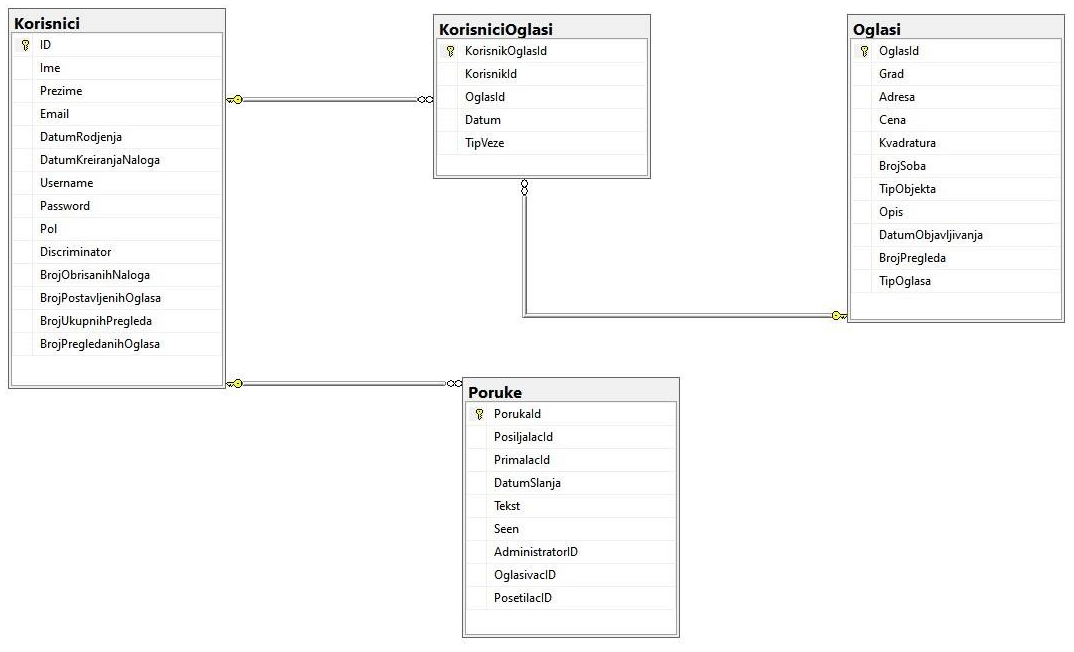
Model domena za koji se Lako do stana portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Kreiranje baze podataka i migracija na SQL Server su obavljeni pomoću odgovarajućih naredbi u konzoli razvojnog okruženja na osnovu skupa klasa koje čine model, dok je dijagram kreiran korišćenjem *Microsoft SQL Server Menagement Studio*-a.



## Komponente sistema

Komponente sistema Lako do stana portala su Razor stranice čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je Razor stranica prikazana kao klasa atributi predstavljaju podatke koje ta stranica čuva, dok metodi predstavljaju funkcije definisane u okviru fajla koji predstavlja C# deo Razor stranice (ImeStranice.cshtml.cs).

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

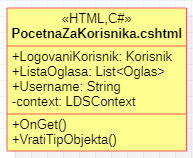


Komponenta **PocetnaZaKorisnika.cshtml** implementira stranicu portala čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

Komponenta **Bootstrap** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama. Sve stranice na portalu za izgled i raspored elemenata na stranici koriste bootstrap tehnologiju.

Sve stranice sem stranica za registraciju, logovanje i početne imaju isti osnovni izgled po uzoru na stranicu **PocetnaZaKorisnika.cshtml** a razlikuju se u svom sadržaju i funkcionalnostima koje omogućavaju.

Parametri koji su potrebni za pravilan rad stranice ilustrovani su sledećim dijagramom klasa:

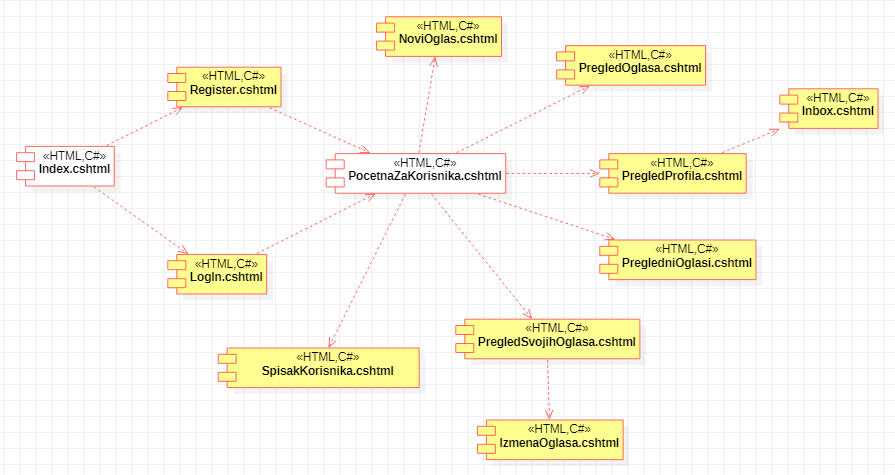


Značenje atributa je sledeće:

* LogovaniKorisnik – objekat koji čuva logovanog korisnika
* ListaOglasa – lista koja čuva sve oglase koji se pribavljaju pri učitavanju stranice
* Username – atribut koji čuva username logovanog korisnika koji se prosleđuje stranici kada se ona otvara
* context – objekat koji služi za rad sa bazom podataka

### Komponente aplikacione logike

Komponente koje realizuju domen problema predstavljaju Razor stranice od kojih svaka pruža potrebne informacije korisniku i implementira i obavlja određene funkcionalnosti koje omogućavaju pravilan rad te stranice. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



**Index.cshtml** – početna stranica koja pruža osnovne informacije o portalu i nudi mogućnost registrovanja novih korisnika i logovanja postojećih na portal.

**Register.cshtml** – stranica za registraciju novih korisnika koji se pružanjem osnovnih podataka i odabirom tipa naloga registruju na portal.

**LogIn.cshtml –** stranica za logovanje postojećih korisnika.

**PocetnaZaKorisnika.cshtml** – početna stranica koje se otvara nakon registracije ili logovanja i omogućava pregled postavlenih oglasa i neke osnovne funkcionalnosti koje variraju u zavisnosti od tipa naloga korisnika.

**SpisakKorisnika.csht**ml – stranica sa spiskom registrovanih korisnika kojoj mogu da pristupe samo administratori. Omogućava pregled i brisanje korisničkih naloga.

**NoviOglas.cshtml** – stranica za dodavanje novih oglasa na portal. Ne mogu da joj pristupe korisnici sa tipom naloga Posetilac.

**PregledOglasa.cshtml** – stranica koja prikazuje osnovne informacije vezane za postavljeni oglas, informacije o korisniku koji je taj oglas postavio i takođe pruža mogućnost kontaktiranja oglašivača.

**PregledProfila.cshtml** – stranica koja prikazuje informacije o nalogu korisnika i omogućava mu promenu istih.

**Inbox.cshtml** – stranica koja korisniku prikazuje sve poruke koje su mu poslali drugi korisnici.

**PregledaniOglasi.cshtml** – stranica koja prikazuje listu oglasa koje je pregledao korisnik sa korisničkim nalogom tipa Posetilac.

**PregledSvojihOglasa.cshtml** – stranica koja korisnicima koji imaju nalog tipa Oglašivač prikazuje listu svih oglasa koji su oni postavili na portal i pruža mogućnost izmene oglasa i brisanje oglasa sa portala.

**IzmenaOglasa.cshtml** – stranica za izmenu podataka o željenom oglasu.

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru C# dela Razor stranica (**ImeStranice.cshtml.cs**) korišćenjem LINQ-a za pisanje upita i celokupni rad sa bazom. Za bilo kakav rad sa bazom korišćen je objekat klase LDSContext koja nasleđuje DbContext.

# 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 1500 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. Lako do stana 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 110 sati.