4.6.2021.



Milica Bogosavljević 2018200926

[*milica.bogosavljevic.18@singimail.rs*](mailto:milica.bogosavljevic.18@singimail.rs)

Miloš Jeknić 2018201579

[*milos.jeknic.18@singimail.rs*](mailto:milos.jeknic.18@singimail.rs)

Veb aplikacija za oglašavanje nekretnina

Predmet: Projektovanje softvera

Sadržaj

**1. Opis projekta3**

**2. Postupak razvoja softvera3**

2.1. Plan projekta4

**3. Analiza zahteva5**

3.1. Prikupljanje zahteva5

3.2. Modelovanje ponašanja6

3.3. Specifikacija i definicija zahteva7

**4. Projektovanje sistema8**

4.1. Projektovanje arhitekture8

4.2. Definisanje entiteta, atributa, relacija i ograničenja 8

4.3. Projektovanje modela baze podataka9

4.4. Ograničenja nad korisničkim akcijama 10

4.5. UML dijagram – dijagram klase 10

4.6. UML dijagram – dijagram aktivnosti11

4.7. Projektovanje programskog koda11

4.8. Projektovanje dizajna sistema12

**5. Implementacija softvera13**

**6. Testiranje sistema14**

6.1. Jedinično testiranje14

6.2. Integraciono testiranje15

**7. Isporuka i održavanje softvera16**

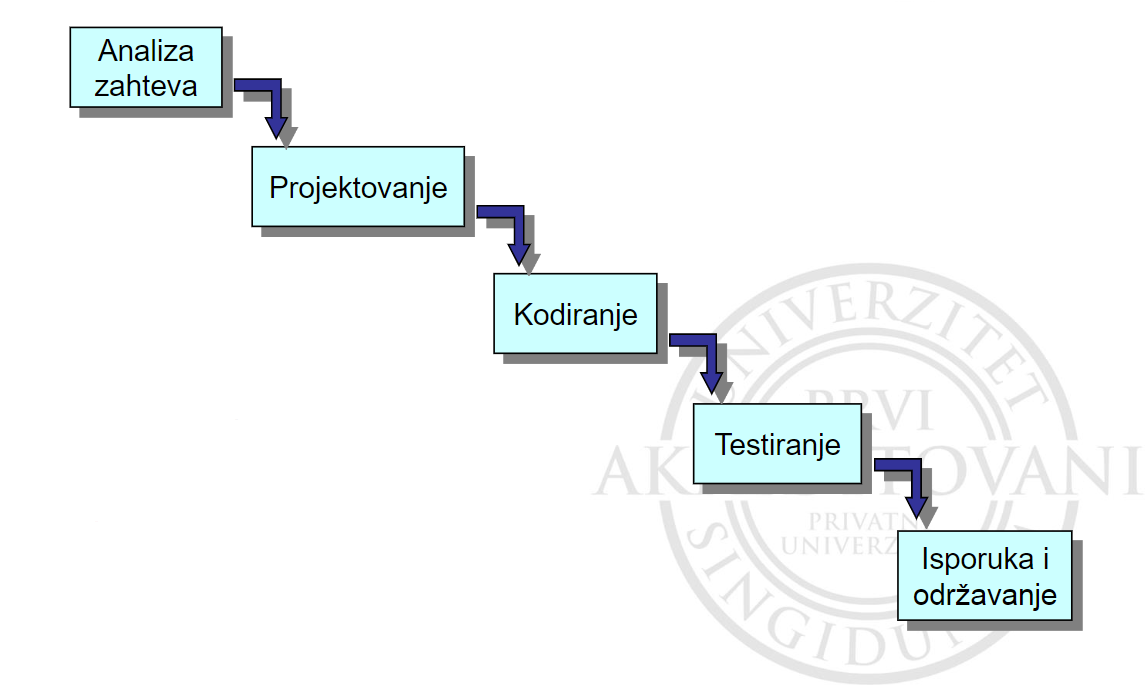
1. Opis projekta

Navedena dokumentacija se bavi projektovanjem i razvojem veb aplikacije za oglašavanje nekretnina. Primarna namena aplikacije je da omogući njenim korisnicima organizaciju različitih oglasa prodaje ili izdavanja nekretnina, pretragu istih, a u isto vreme da izvrši i pojednostavljivanje interakcija u ovoj grani, podsredstvom Interneta.

Potpunim razvojem ove veb aplikacije, njeni korisnici će moći bliže da se upoznaju sa principima trgovanja u pogledu nekretnina.

1. Postupak razvoja softvera

Prilikom izrade projekta korišćen je kaskadni model, koji omogućava da se za svaku fazu softvera definišu kritične tačke, kojima će se zadovoljiti prelazak na narednu fazu. Glavni nedostatak primene ovog modela je nepostojanje povratnih sprega. Kada razvojni tim, nakon revizije kritičnih tačaka, odluči da je moguće preći na sledeću fazu, nema povratka na prethodne.

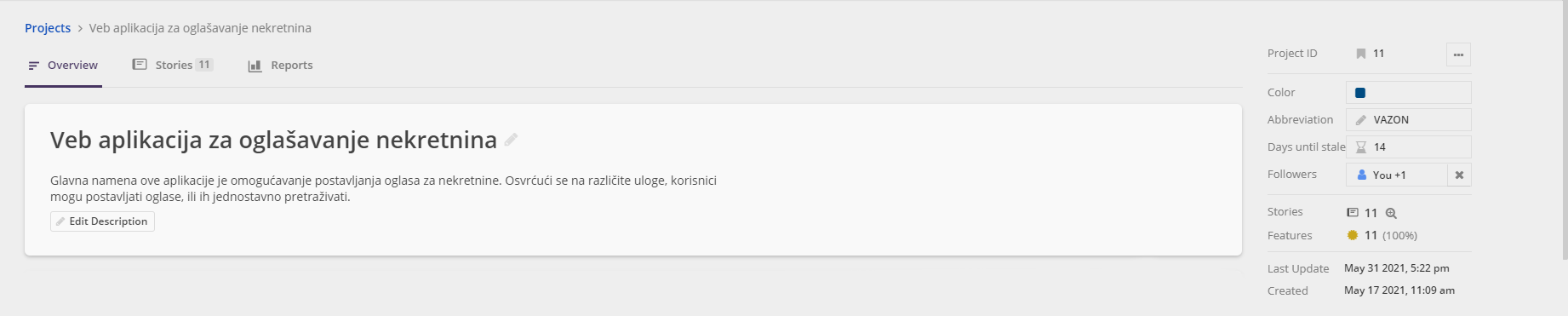


Ilustracija 1: Vizuelni primer kaskadnog modela

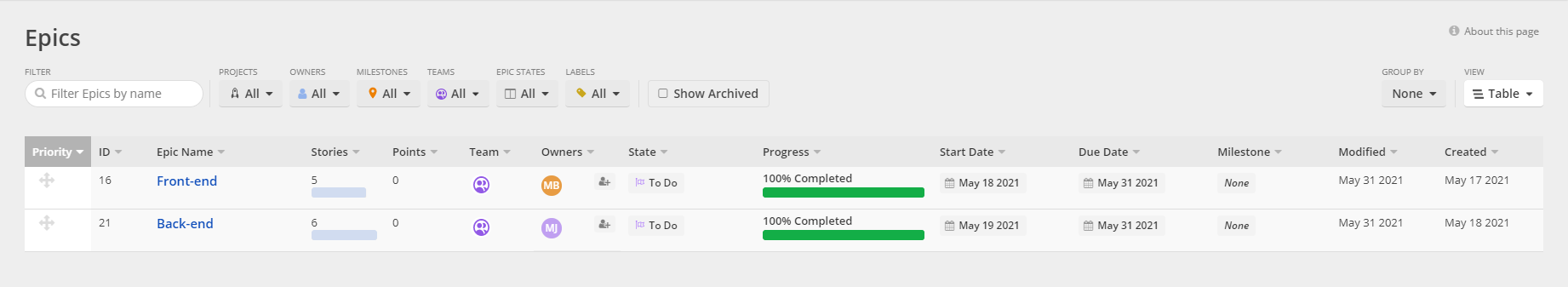
* 1. Plan projekta

Svaki projekat bi trebalo da poseduje kritične tačke, u odnosu na koje se članovi razvojnog tima orijentišu, po pogledu izvršavanja koraka u procesu projektovanja. Naredna slika ilustruje podelu poslova u okviru razvojnog tima.

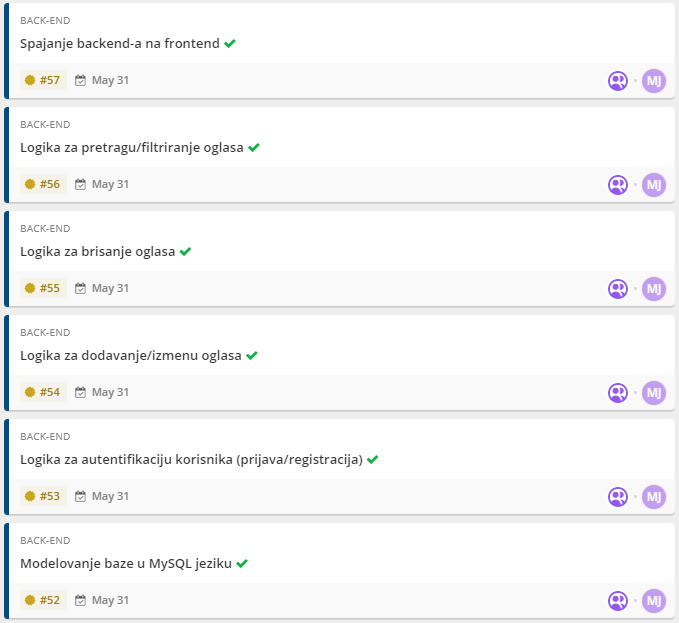
Član tima zadužen za razvoj logičkog dela koji podatke povezuje sa korisničkim interfejsom je Miloš Jeknić, dok se izradom samog korisničkog interfejsa bavila Milica Bogosavljević.



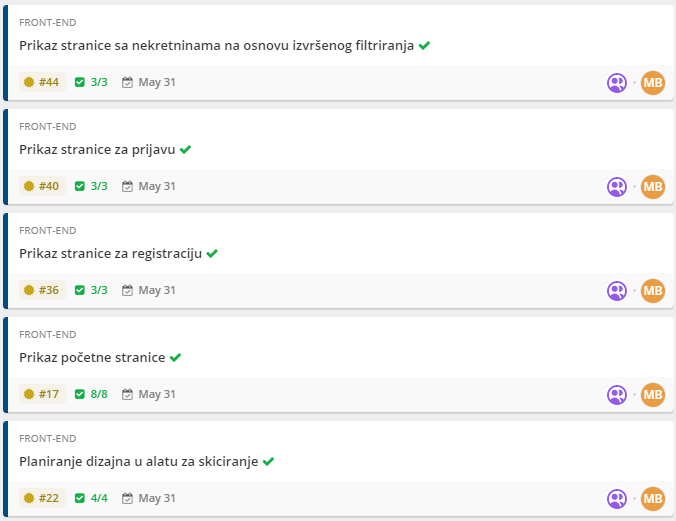
Ilustracija : Metapodaci o projektu



Ilustracija : Raspodela posla sa datumima početka i završetka izrade



Ilustracija 4: Kritične tačke na razvoju backend-a



Ilustracija 5: Kritične tačke na razvoju frontend-a

1. Analiza zahteva

Skup zahteva je rezultat intenzivne saradnje sa naručiocima u cilju razumevanja njihovih osnovnih problema i potreba.

* 1. Prikupljanje zahteva

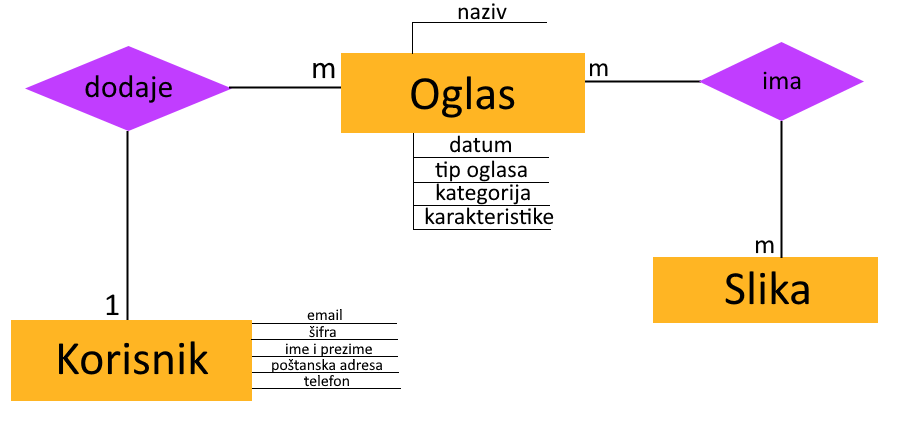
Ovaj deo prve faze omogućava da se od naručioca traženog softvera kroz stavke razgovora, čitanja dokumentacije, ustanovljavanja korisničkih potreba i neophodnih funkcija razmatranog sistema, utvrde njegove potrebe. Koristeći navedene stavke, došlo se do sledećih zahteva:

* Aplikacija treba da omogući korisniku da može da postavi oglas za izdavanje ili prodaju nekretnine.
* Korisnik mora da se registruje i da bude prijavljen da bi dodavao oglas za nekretninu na portal. Za svaku nekretninu treba omogućiti da korisnik obeleži najosnovnija svojstva nekretnine, kao što su cena, lokacija nekretnine, naslov, kratak opis ograničene dužine, naslovnu fotografiju koju upload-je, kao i karakteristike same nekretnine, kao što su sprat u zgradi od ukupne spratnosti, grejna i ukupna površina, broj terasa, broj soba, vrsta grejanja, postojanje određenih pogodnosti, kao što su lift u zgradi, priključan za kablovsku televiziju, Internet, telefon, interfon na ulazu, garaža ili parking mesto koje pripada vlasniku stana, kao i određena legalna svojstva nekretnine, kao što je informacija o tome da li je nekretnina uknjižena, ako je na placu, da li postoji dozvola za gradnju, itd.
* Sve ostale osobine, karakteristike i svojstva oglasa ili nekretnine koja mogu da budu podešena su prednost u smislu realizacije projekta.
* Neprijavljeni korisnici mogu da vrše pretragu po filterima, npr. opseg cena, opseg površine nekretnine, posedovanje određenih pogodnosti (parking, lift, interfon…), lokacija na kojoj je nekretnina itd.
* Rezultati pretrage treba da prikazuju liste nekretnina u istom obliku kao i listanje po kategoriji (stan, kuća itd.) ili po lokaciji (mestima u kojima je nekretnine).
* Kada se otvori stranica određene nekretnine, svi podaci treba da budu prikazani na pregledan način i da budu jasno uočljivi i dostupni bez dodatnih akcija od strane korisnika.

Zahteve koji su definisani dalje možemo podeliti prema prioritetu izvršenja. U slučaju datog projekta primeri mogu biti:

1. suštinski – korisniku se mora omogućiti pregledanje oglasa za nekretnine.
2. poželjni – nekretnine prilikom pretrage mogu da se filtriraju po kvadraturi (fiksna vrednost ili u opsegu).
3. opcioni – cena nekretnine je ispisana podebljanim, krupnim slovima, za jasnije isticanje.
   1. Modelovanje ponašanja

U ovom delu faze, omogućava se bolje razumevanje zahteva, kao i otkrivanje njihovih nedostataka ili nedoslednosti. Jedan od primera predstavljaju ER dijagrami. Naredni dijagram predstavlja povezanost najvažnijih entiteta u sistemu.



Ilustracija : ER dijagram povezanosti najvažnijih entiteta

* 1. Specifikacija i definicija zahteva

Zahtevi se obično formulišu pripremom dokumentacije koja se deli na dve osnovne grupe: definiciju i specifikaciju zahteva.

Definicija zahteva:

1. Korisniku se onemogućava dodavanje oglasa bez prethodne autentifikacije.
2. Korisniku se onemogućava prijavljivanje bez prethodne registracije.
3. Korisniku se onemogućava izmena i brisanje oglasa koji nisu njegovi.
4. Korisniku se omogućava filtriranje nekretnina po atributima.
5. Korisnik može u svakom trenutku da promeni svoju šifru.

Specifikacija zahteva:

* 1. Aplikacija treba da omogući korisniku da može da postavi oglas za izdavanje ili prodaju nekretnine. Namena oglasa, prema kojoj se on razvrstava, biće definisana u posebnom atributu odgovarajućeg entiteta.
  2. Korisnik mora da se registruje, unošenjem svoje email adrese, proizvoljne šifre, svog imena i prezimena, poštanske adrese i broja telefona, kao i da bude prijavljen sa svojom email adresom i šifrom, sa kojom je prethodno registrovan, da bi dodavao oglas za nekretninu na portal, pomoću specifičnog interfejsa za unos atributa o oglasu. Za svaku nekretninu treba omogućiti da korisnik unese najosnovnija svojstva nekretnine: cena, lokacija nekretnine, naslov, kratak opis ograničene dužine, naslovnu fotografiju koju upload-je, pritom i karakteristike same nekretnine: sprat u zgradi od ukupne spratnosti, grejna i ukupna površina, broj terasa, broj soba, vrsta grejanja, postojanje određenih pogodnosti: lift u zgradi, priključan za kablovsku televiziju, Internet, telefon, interfon na ulazu, garaža ili parking mesto koje pripada vlasniku stana, kao i određena legalna svojstva nekretnine: informacija o tome da li je nekretnina uknjižena, ako je na placu, da li postoji dozvola za gradnju.
  3. Neprijavljeni korisnici mogu da vrše pretragu na početnoj stranici po sledećim filterima: kategoriji nekretnine (kuća ili stan), nameni/tipu oglasa (prodaja ili izdavanje), cene sa krajnjom vrednošću, površinom/kvadraturom sa početnom vrednošću i lokaciji na kojoj je nekretnina.
  4. Oglas zbog programske logike mora da poseduje polja koja nisu vidljiva krajnjem korisniku, to su: identifikacioni broj oglasa i email korisnika koji je dati oglas kreirao, kako bi bio u mogućnosti da menja ili briše samo sopstvene oglase.
  5. Kada se otvori stranica određene nekretnine, svi podaci treba da budu prikazani na sledeći način: naslovna slika oglasa na njegovoj levoj ivici, na desnoj ivici su ispisane važne informacije o: nazivu, ceni u evrima, kvadraturi u kvadratnim metrima, spratnosti (broj sprata od ukupnog broja spratova), i to crnim, podebljanim slovima, kako bi bile istaknutije, a ispod njih nalaze se sekundarne informacije, koje sadrže: kategoriju nekretnine (kuća ili stan), namenu/tip oglasa (prodaja ili izdavanje), lokaciju nekretnine, kratak opis, uknjiženost nekretnine, da li je nekretnina na placu, da li poseduje građevinsku dozvolu, datum kreiranja oglasa, broj soba, tip grejanja, posedovanje lifta, terase, parkinga, kablovske televizije, Interneta, telefona i interfona.

1. Projektovanje sistema

Navedena faza predstavlja kreativni i iterativni proces prevođenja problema u njegovo rešenje.

* 1. Projektovanje arhitekture

Prilikom projektovanja sistema, korišćen je stil klijent-server arhitekture. Navedeni stil omogućava da klijent potražuje podatke o oglasima, a da mu server pruža podatke o istima u što kraćem vremenskom roku.

Na sledećoj ilustraciji ukratko je predstavljen glavni princip ovog stila.

SERVER

Dobavlja listu oglasa, filtriranu na osnovu primljenih parametara

KLIJENT

Inicira pretragu prema postavljenim parametrima

Ilustracija : Prikaz klijent-server stila

* 1. Definisanje entiteta, atributa, relacija i ograničenja

Prijavljeni korisnik

Prijava

Dodavanje nekretnine

Izmena nekretnine

Uklanjanje nekretnine

Uređivanje naloga

Promena lozinke

Registracija naloga

Posetilac sajta

Pretraga nekretnina filtriranjem

Pretraga nekretnina po lokaciji

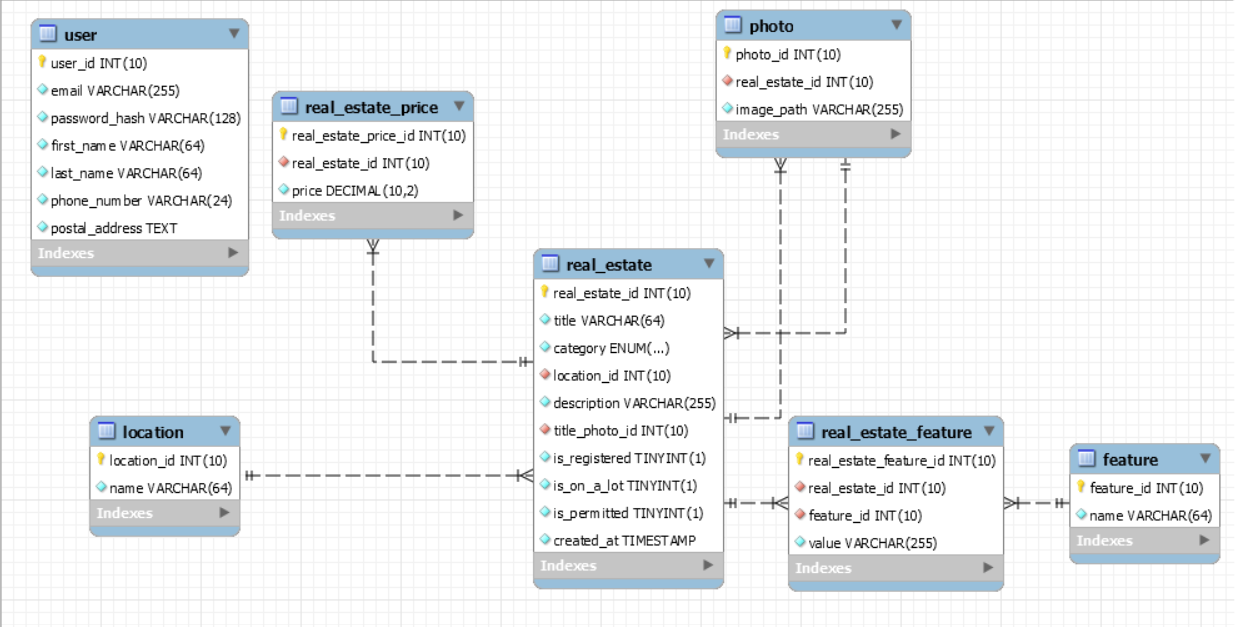
Pretraga nekretnina po kategorijama

Pregled nekretnina po kategorijama

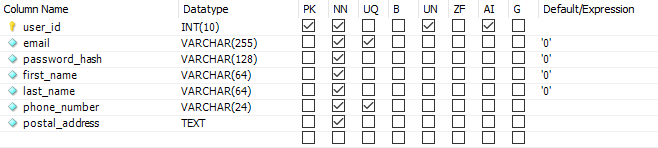
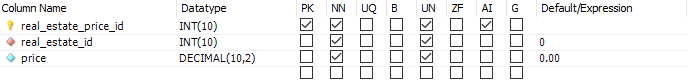
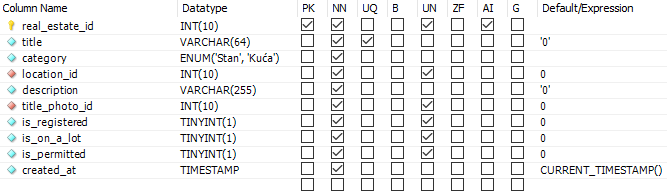
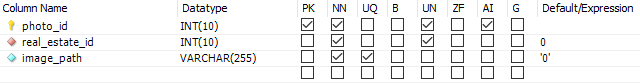
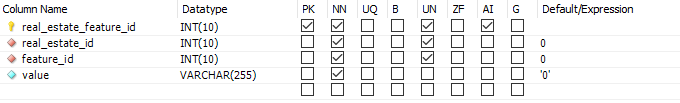
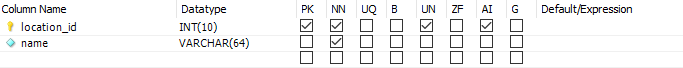
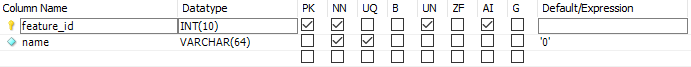
Ilustracija : Use case dijagrami za uloge u softveru

* 1. Projektovanje modela baze podataka

Korišćenjem MySQL jezika, modelovana je baza podataka na sledeći način:



Ilustracija 9: Grafički prikaz entiteta, atributa i njihovih relacija u bazi podataka



* 1. Ograničenja nad korisničkim akcijama
* Posetiocima sajta dozvoljeno je **samo** pregledanje nekretnina kao i pretraga nekretnina po filterima.
* Dodavanje novih oglasa kao i njihova izmena i brisanje dozvoljena je **isključivo** registrovanim i ulogovanim korisnicima.
* Registrovanim i ulogovanim korisnicima moguće je dodavanje kratkog opisa od **najviše** 255 karaktera.
* Svim posetiocima sajta omogućena je pretraga nekretnina čija je lokacija **samo** na teritoriji grada Beograda.
  1. UML dijagram – dijagram klase

dodajOglas

izmeniOglas

obrisiOglas

Ilustracija :Primer dijagrama klase za entitet oglasa

IdOglasa:Number[1]

Kategorija:Enum[1]

TipOglasa:Enum[1]

NazivOglasa:String[1]

Lokacija:String[1]

Opis:String[1]

PutanjaDoSlike:String[1]

Registrovan:Boolean[1]

NaPlacu:Boolean[1]

ImaDozvoluZaGradnju:Boolean[1]

DatumKreiranja:Date[1]

Cena:Number[1]

Oglas

Kvadratura:Number[1]

BrojSoba:Number[1]

Grejanje:String[1]

Spratnost:String[1]

ImaLift:Boolean[1]

ImaTerasu:Boolean[1]

ImaParking:Boolean[1]

ImaKablovsku:Boolean[1]

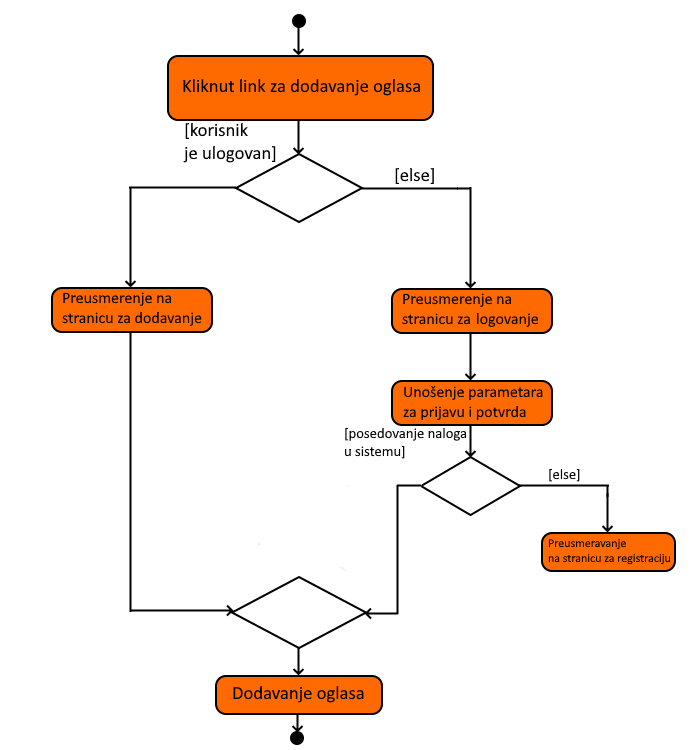
ImaInternet:Boolean[1]

ImaTelefon:Boolean[1]

ImaInterfon:Boolean[1]

KreatorOglasa:User[1]

* 1. UML dijagram – dijagram aktivnosti



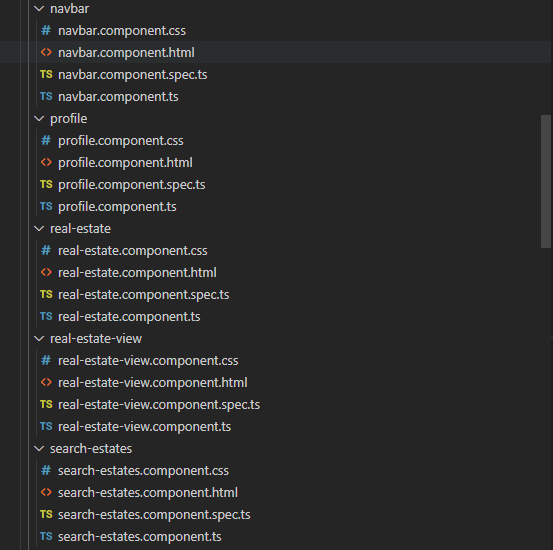
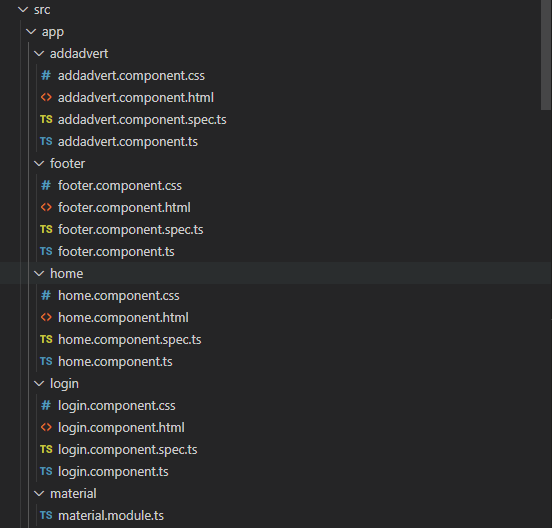
Ilustracija :Primer dijagrama aktivnosti za dodavanje novog oglasa

* 1. Projektovanje programskog koda

U dogovoru sa projektantom sistema, razvojni tim je odlučio da će se za implementaciju programskog koda koristiti Angular Framework, koji je zbog tehničkih i vremenskih ograničenja poslužio za sve aspekte prilikom razvoja projekta. U te aspekte su uključeni dizajn korisničkog interfejsa (frontend), simulacija logičkog pristupa (backend) kao i uključivanje uzoraka predefinisanih podataka, koji vrše simulaciju strukture baze podataka.

Prikaz interfejsa na strani klijenta omogućen je primenom HTML-a i CSS-a, dok je programska logika odrađena u programskom jeziku TypeScript.

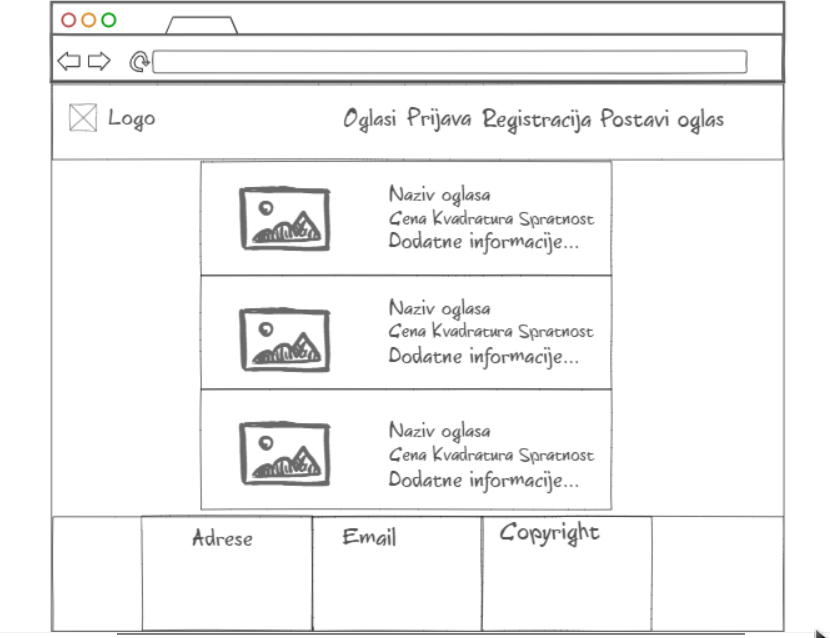
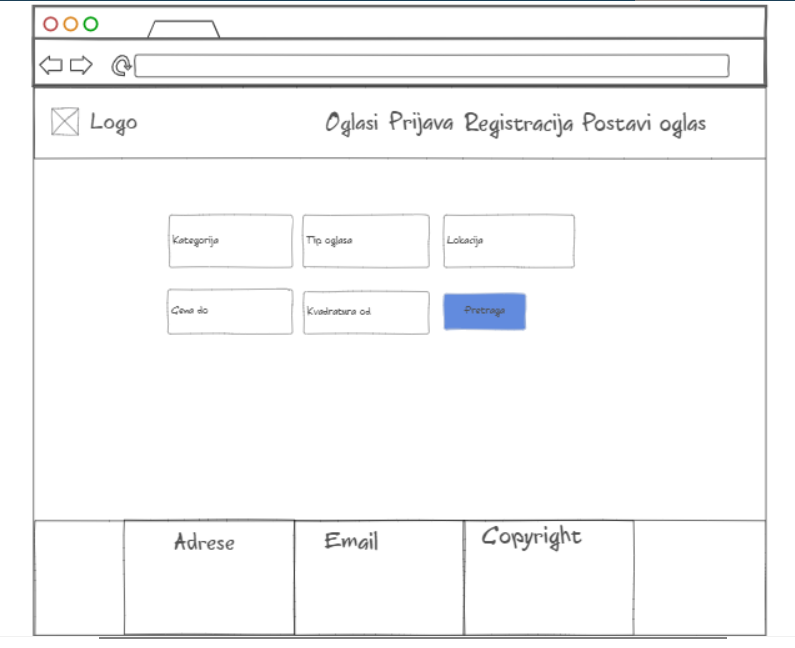
Projekat će biti implentiran na mašinama koje pokreću Windows OS.



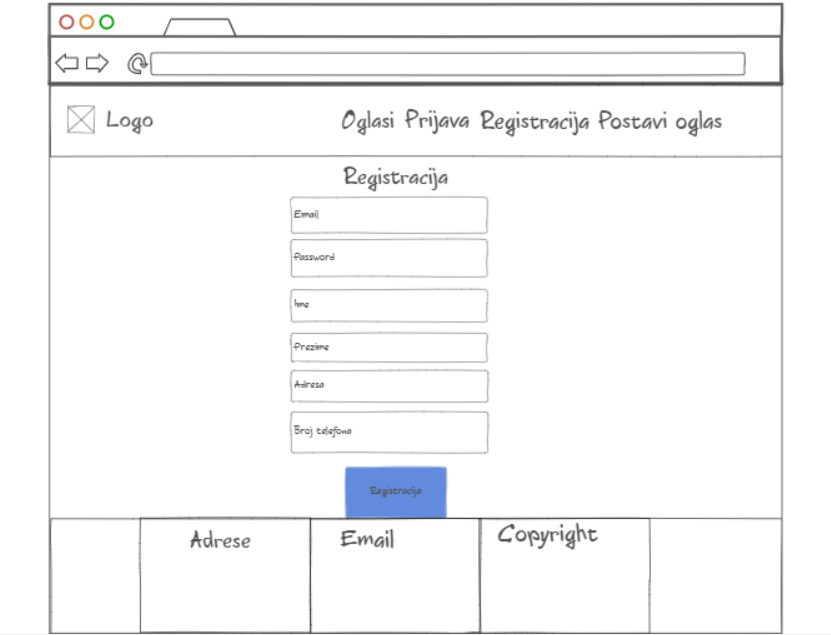
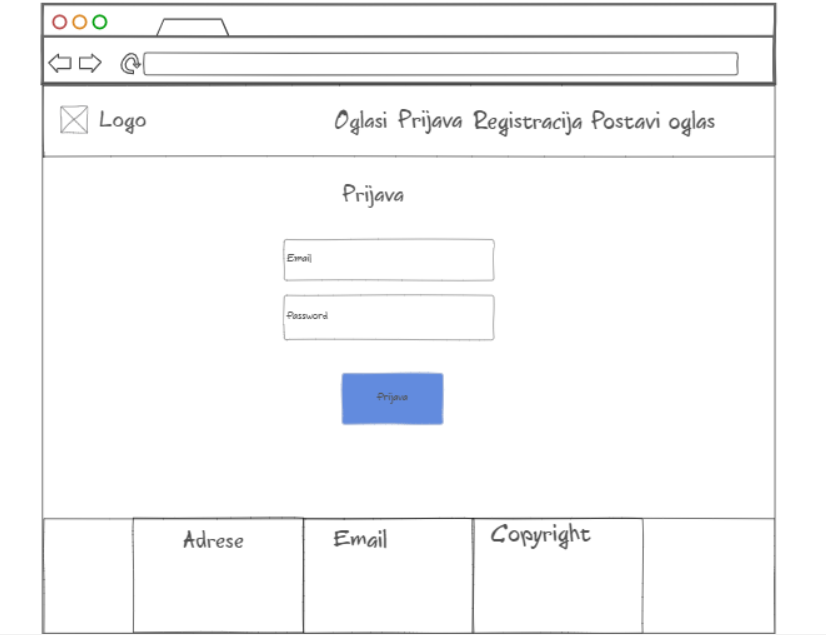
Ilustracija : Deo strukture programskih modula i komponenti

4.8. Projektovanje dizajna sistema

U ovom poglavlju razrađuje se inicijalni koncept izgleda korisničkog interfejsa date veb aplikacije. Program koji se koristio prilikom realizacije jeste NinjaMock.



Ilustracija : Prototip dizajna početne i stranice sa listom oglasa

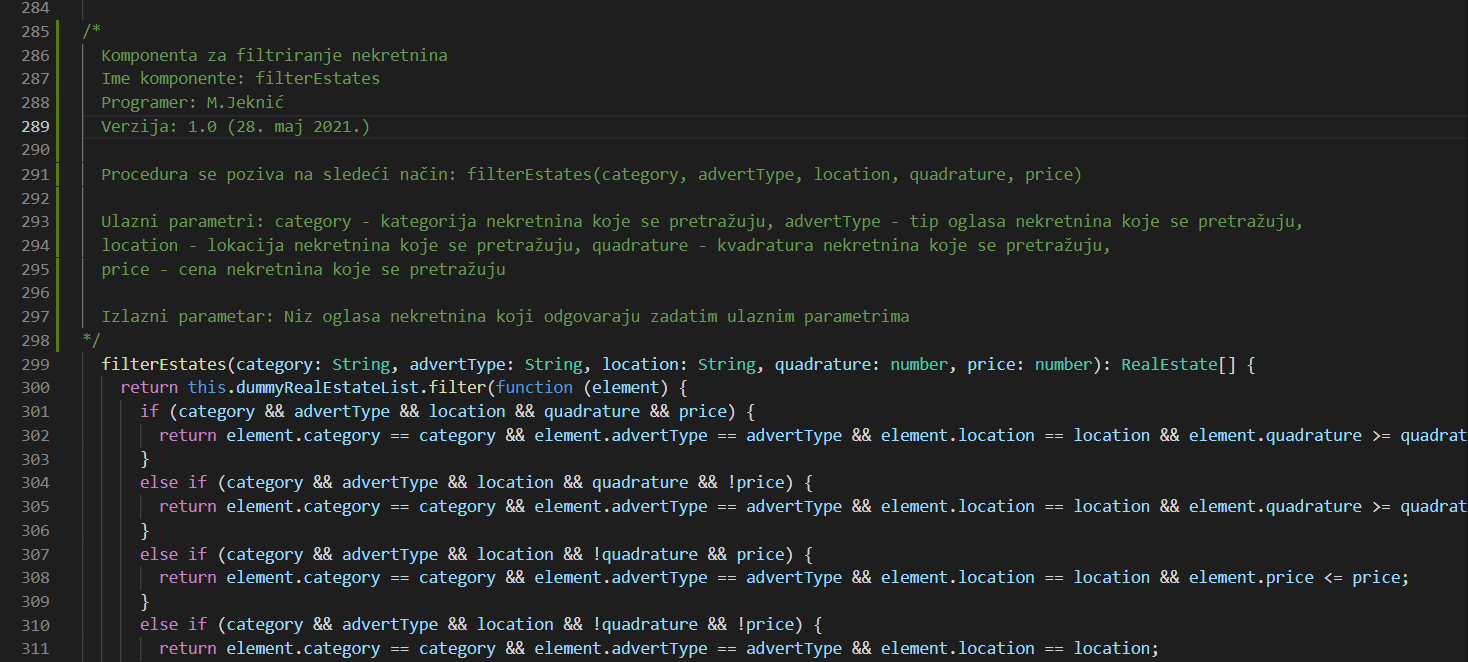


Ilustracija : Prototip dizajna stranica za prijavu i registraciju

1. Implementacija softvera

Ova faza predstavlja softversku realizaciju rešenja problema. Njen konačni rezultat predstavlja skup programa koji ispunjavaju zadate funkcionalne zahteve. Neke od funkcija koje su korišćene pri implementaciji su sledeće:

* addAdvert() – funkcija koja izvršava dodavanje novog oglasa u listu postojećih. Kao ulazni parametar se postavljaju atributi koji su definisani unapred pripremljenom korisničkom formom, a izlazni parametar je primerak klase samog oglasa.
* isPasswordCorrect() – funkcija vrši proveru podudaranja šifre unete u korisničkom interfejsu, sa šifrom koja postoji u sistemu korisnika. Ulazni parametri su email i šifra iz interfejsa, a kao izlaznu vrednost dobijamo tačan ili netačan iskaz (boolean)
* loggedInCheck() – funkcija izvršava proveru da li je korisnik ulogovan u aktivnoj sesiji. Namena funkcije je da spreči neautorizovano dodavanje oglasa. Ukoliko korisnik nije ulogovan (provera posebne promenljive isSignedIn), oglašava se obaveštenje da korisnik mora izvršiti prijavu, a zatim se pregledač upućuje na stranicu prijave.



Ilustracija : Primer dobro napisane unutrašnje dokumentacije

1. Testiranje softvera

Za potrebe testiranja softvera, Angular omogućava korišćenje sopstvenih test klasa. Pokretanjem komande ng test pravi se izdvojeno okruženje, koje radi na drugom portu nezavisno od same aplikacije. Struktura testiranja je potpomognuta Karma + Jasmine okruženjima. Primenjeni metod testiranja bio je metod „crne kutije“, gde se ne zna previše o unutrašnjim funkcijama programa.

* 1. Jedinično testiranje

1) Za potrebe testiranja komponente za prijavu, u polja za unos lozinke za ispravnu mejl adresu, uneli smo pogrešnu lozinku, što je dovelo do izbacivanja teksta sa odgovarajućom greškom „Pogrešan unos šifre!“.

Test primer za potrbe testiranja komponente za prijavu:

-ulazni podatak za email je: [test1@gmail.com](mailto:test1@gmail.com) a za password je: 123456

S obzirom na to da je za zadatu email adresu password: test123456 dolazi do izbacivanja greške: „Pogrešan unos šifre!“. Prema tome, možemo izdvojiti dve klase ekvivalencije: jednu legalnu klasu (šifra = test123456) i jednu nelegalnu (šifra ≠ test123456).

2) Prilikom pokušaja da se izbegne unošenje bilo kojih neophodnih parametara dolazi do izbacivanja poruke „Polje ne bi smelo da bude prazno“.

Test primer za potrebe testiranja komponente za prijavu:

-ulazni podatak za email je: [test1@gmail.com](mailto:test1@gmail.com) a za password ostavljamo prazno polje.

Pošto ništa nije uneto u polje za unos šifre, dolazi do izbacivanja greške: “Polje ne bi smelo da bude prazno!“.

Ovde se izdvajaju takođe dve klase ekvivalencije: jednu legalnu (dužina šifre > 0) i jednu nelegalnu (dužina šifre ≤ 0).

3) Za potrebe testiranja komponente za filtriranje, kako bi se sprečilo unošenje neodgovarajućih tipova parametara za kategoriju, tip oglasa i lokaciju, korisniku je na raspolaganju odabir izmedju unapred definisanih vrednosti.

Legalne klase za tip oglasa su: prodaja i izdavanje, za kategoriju nekretnine su kuća i stan a za lokacije su isključivo krajevi grada Beograda.

* 1. Integraciono testiranje

1) Za potrebe provere saradnje između komponenata za prijavu i registraciju, kao test primer unosili smo u polja za prijavu parametre koji ne postoje u evidenciji registrovanih korisnika.

U evidenciji registrovanih korisnika postoje:

{ id: 1, email: "test1@test.com", password: "test123456", name: "Pera", surname: "Perić", address: "Mirka Mirkovića 17, Beograd", telephone: "+381641234567" },

{ id: 2, email: "test2@test.com", password: "test654321", name: "Mika",surname: "Mikić", address: "Ružice Sokić 54, Beograd",telephone: "+381637654321 }

Kao test primer uneli smo email: [test1@test.com](mailto:test1@test.com), a kao password uneli smo test654321. S obzirom na to da u evidenciji registrovanih korisnika ne postoji onaj sa zadatom kombinacijom, dolazi do greške: „Nema registrovanog korisnika sa zadatim parametrima!“.

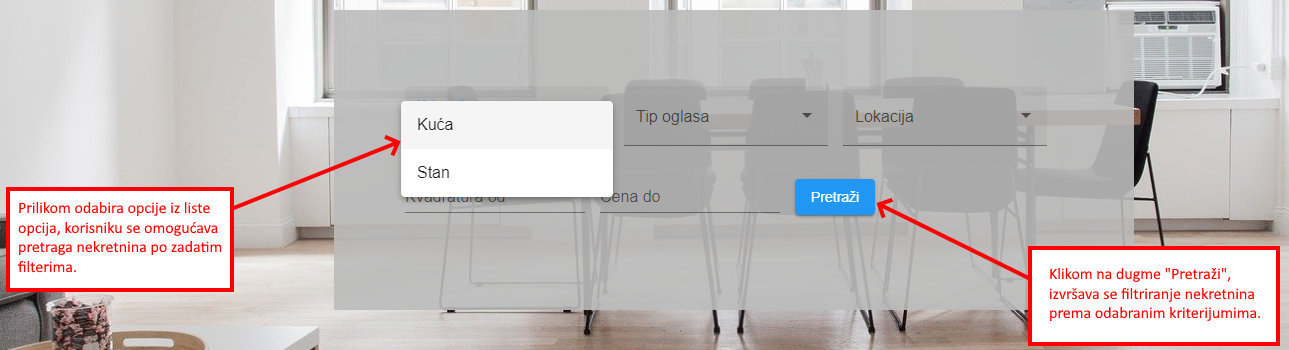
2) Izvršena je provera saradnje izmedju komponente za filtriranje i prikaz nekretnina tj. vraćanja odgovarajućih izlaznih parametara, za parametre zadate od strane korisnika, ručnom proverom podudaranja podataka koji postoje u interfejsu i sadrže tražene parametre sa listom nekretnina i parametrima koji su prikazani.

Za potrebe testiranja unete su sledece ulazne vrednosti: tip nekretnine: Kuća, tip oglasa: Izdavanje, lokacija:Vozdovac, cena do: 200.000€ , kvadratura od 50 m/2. Kao izlaz vraćen je izlazni niz: [{id: 6, category: "Kuća", advertType: "Izdavanje", title: "Kuca na izdavanje povoljno", location: "Vozdovac", description: "Kuca u blizini Dvanaeste beogradske gimnazije, okruzena zelenilom, prava idila!", imagePath: "../assets/kuca3.jpg", isRegistered: true, isOnALot: true, isPermitted: true, createdAt: new Date('2021-03-03').toDateString(), price: 150000, quadrature: 120,numOfRooms: 5,heating: "Centralno",numOfFloor: "1/1",hasElevator: false, hasTerrace: false, hasParking: true, hasCableTv: true, hasInternet: true, hasPhone: true, hasIntercom: false, createdBy: "test1@test.com" }]

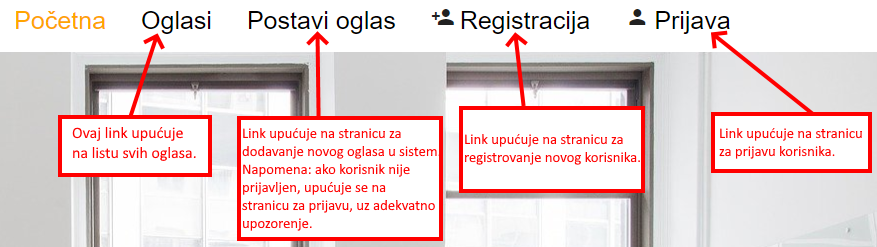
Ovim smo se uverili da je filtriranje odradjeno na adekvatan način.

7. Isporuka i održavanje softvera

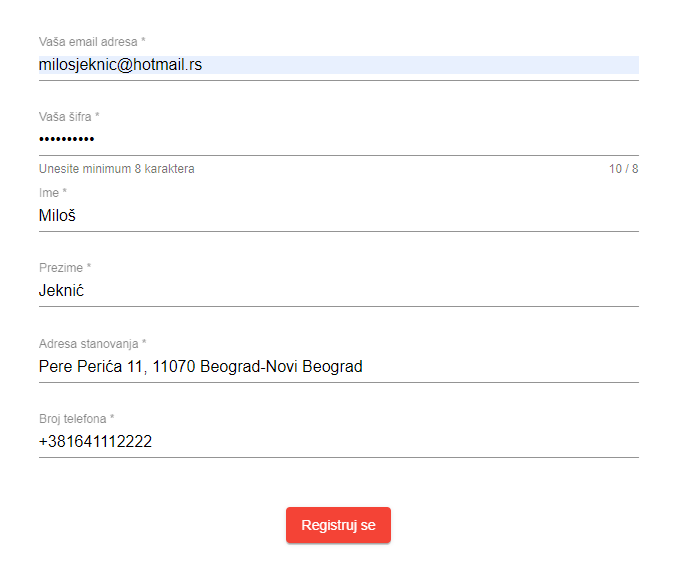
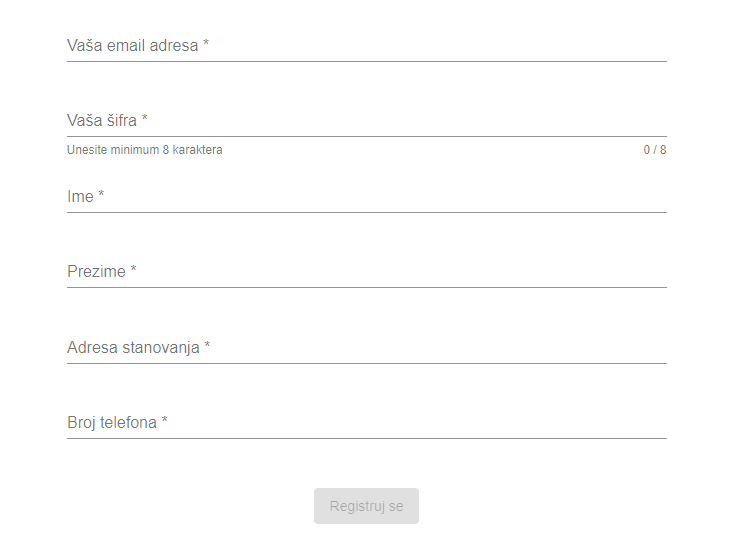
U ovom poglavlju biće konkretno opisan način upotrebe izrađenog softvera, kako bi se krajnjim korisnicima približio način njegovog funkcionisanja.



Ilustracija : Demonstracija popunjavanja forme za pretragu



Ilustracija : Pojedinačna objašnjenja opcija iz navigacionog menija



Ilustracija : Primer nepopunjene i popunjene forme za registraciju