

Bojana Mitrović 2018200421

Stojanka Tešanović 2018200736

Veb aplikacija za online prodaju auto delova

Predmet:Projektovanje softvera

Sadržaj

**1. Opis projekta2**

**2. Postupak razvoja softvera2**

2.1. Plan projekta3

2.2. Planirani rezultati za kritične tačke6

**3. Analiza zahteva6**

3.1. Prikupljanje zahteva6

3.2. Modelovanje ponašanja8

3.3. Specifikacija i definicija zahteva8

**4. Projektovanje sistema10**

4.1. Prikaz proizvoda10

4.2. Projektovanje arhitekture10

4.3. Definisanje entiteta, atributa, relacija i ograničenja 11

4.4. Projektovanje baze podataka12

4.5. Ograničenja nad korisničkim akcijama 13

4.6. UML dijagram – dijagram klase 14

4.7. UML dijagram – dijagram aktivnosti15

4.8. Projektovanje programskog koda15

4.9. Projektovanje dizajna sistema16

**5. Implementacija programskog koda18**

**6. Testiranje softvera19**

6.1. Jedinično testiranje20

6.2. Integraciono testiranje20

**7. Isporuka i održavanje softvera21**

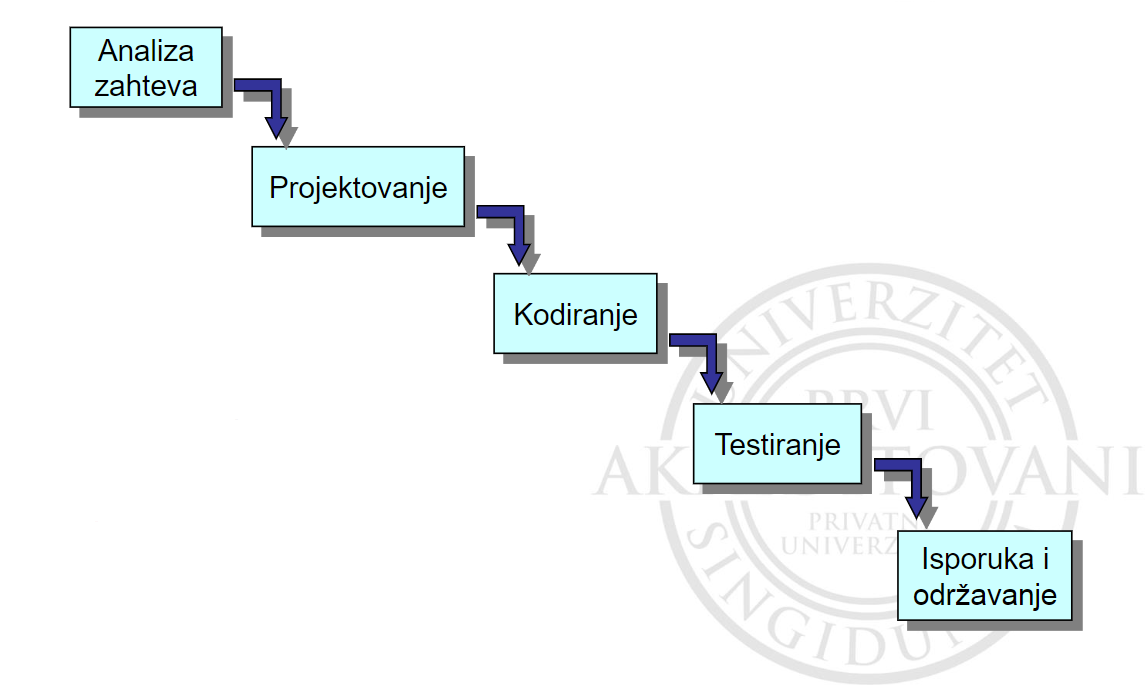
1. **Opis projekta**

Dokumentacija se bavi razvojem veb aplikacije za online prodaju auto delova.Namena aplikacije je da omogući korisnicima lakše pretraživanje I pregledanje auto delova,kao i poručivanje istih posredstvom Interneta.

Kompletnim razvojem veb aplikacije,korisnici će moći da se upoznaju sa principima online prodaje auto delova.

1. **Postupak razvoja softvera**

Prilikom izrade projekta korišćen je kaskadni model.On omogućava da se za svaku fazu razvoja softvera definišu kritične tačke kojima će se zadovoljiti prelazak u narednu fazu. Glavni nedostatak primene ovog modela je nepostojanje povratnih sprega.



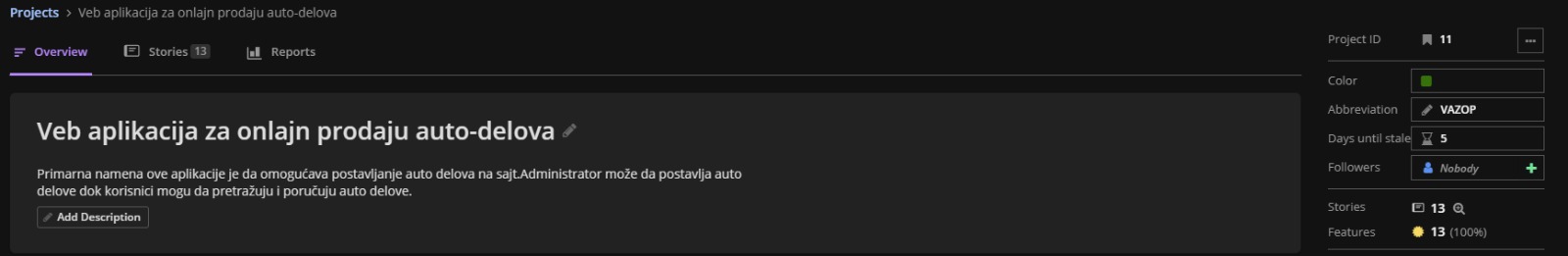
*Slika 1. Kaskadni model*

* 1. **Plan projekta**

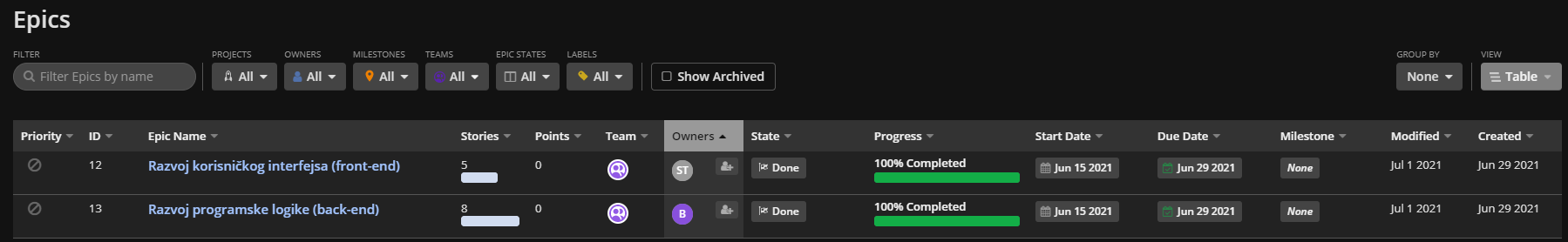
U nastavku slede kritične tačke za koje će svako od članova tima biti zadužen za vreme razvoja projekta.

Član tima zadužen za logički deo aplikacije koji se povezuje sa korisničkim interfejsom je Bojana Mitrović, a za izradu korisničkog interfejsa je Stojanka Tešanović.

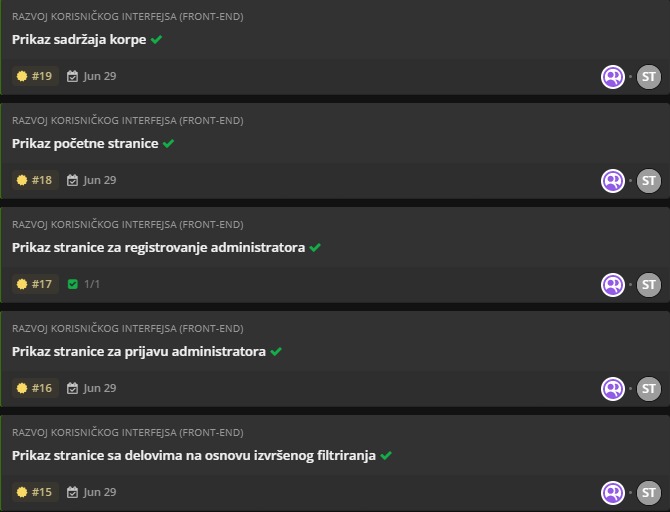
Naredne slike ilustruju podelu uloga u okviru razvojnog tima.



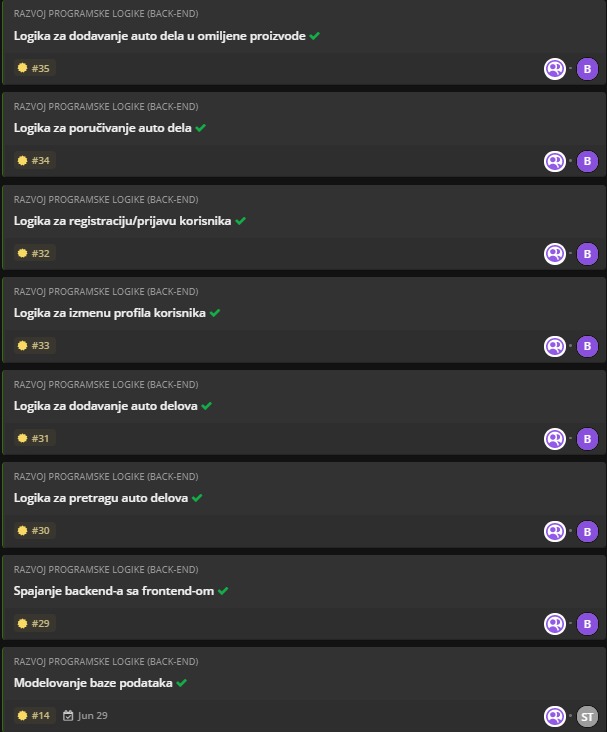
*Slika 2. Metapodaci o projektu*



*Slika 3. Raspodela posla sa datumima*



*Slika 4.Kritične tačke za frontend*

1. 

*Slika 5.Kritične tačke za backend*

* 1. **Planirani rezultati za kritične tačke**
* **Modelovanje baze podataka:** Prvi korak pri izradi aplikacije jeste modelovanje relacione baze podataka odakle kreću sve funkcionalnosti aplikacije. Prvo što radimo jeste identifikacija administratora i sve aktivnosti koje on treba da obavlja, u ovom slučaju dodavanje novih delova. Takođe određuju se ograničenja za obične korisnike, i korisnike koji žele da poruče proizvod, kao i identifikacija međusbnih veza između ostalih tabela u bazi.
* **Izrada baze podataka:** U bazi podataka postoji tabela koja omogućava administratorima da dodaju nove proizvode u online prodavnicu. Obični korisnici takođe sve proizvode mogu pretraživati po ključnoj reči, ali da bi korisnik naručio proizvod mora da ostavi podatke o sebi koji se čuvaju u posebnoj tabeli korisnik. U tabeli korpa se čuvaju proizvodi koje korisnik želi da naruči.
* **Backend programiranje:** Što se tiče pozadinske logike, pre svega je neophodno da se omogući povezivanje sa bazom podataka. Treba omogućiti pristup različitim podacima iz različitih tabela radi uspešnog izvršavanja upita
* **Frontend programiranje:** S’obzirom da je aplikacija namenjena korisnicima, fokus je stavljen na korišćenje intuitivnih imena i ikonica radi lakšeg snalaženja i korišćenja. Za predstavljanje podataka korisniku i dizajn stranica korišćen je HTML I CSS, a za validaciju unetih podataka korišćen je JavaScript.

1. **Analiza zahteva**

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

* 1. **Prikupljanje zahteva**

Prikupljanje zahteva predstavlja veoma važnu aktivnost u procesu definisanja zahteva jer se tokom nje utvrđuje šta naručioci I korisnici stvarno žele.

Aplikacija treba da ispunjava sve zahteve koji su dobijeni u fazi prikupljanja zahteva kao što su:

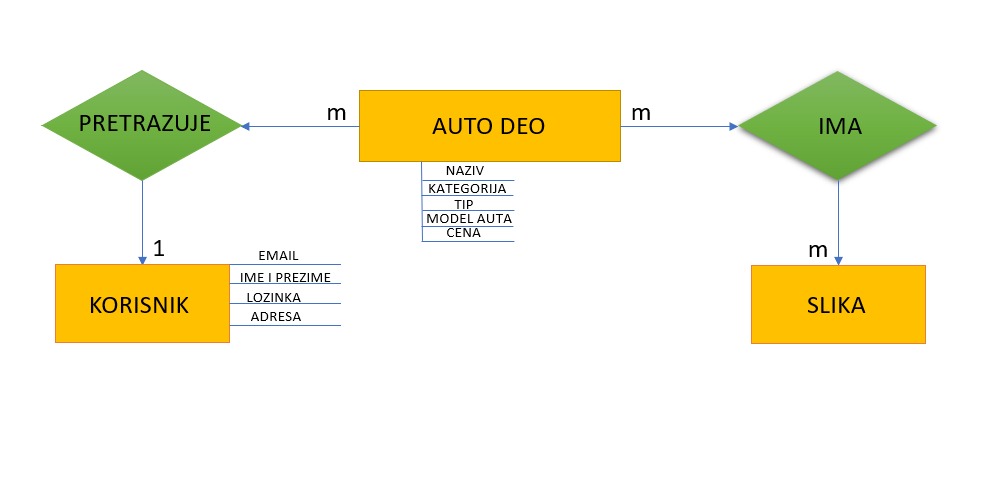
* Aplikacija treba da omogući administratorima da dodaju nove auto delove u bazu podataka aplikacije za Internet prodaju.
* Samo prijavljeni korisnici, koji se na portal prijave sa ispravnih parametrima naloga administratora sadržaja mogu da pristupe administrativnom panelu portala.
* U ovom panelu mogu da dodaju nove kategorije/vrste delova, za koje proizvođače i modele automobila su, kao i koje je raspoloživo stanje pojedinog dela. Podaci o delovima se sastoje od naziva, slike, detaljnog opisa, cene, vrste/kategorije, za koje sve modele automobila mogu da se koriste, da li su fabrički/originalni ili zamenski, da li se izdaje garancija i ako da, na koliko mesec itd.
* Sa korisničke strane treba omogućiti prikaz svih delova poređanih po ceni za svaku od kategorija/vrsta.
* Kada korisnik otvori stranicu nekog dela treba da vidi sve detalje o njemu, kao što su naziv, slika, opis, cena i druge karakteristike i da dobije mogućnost da izvrši kupovinu, tj. dodavanje u korpu.
* Izabrani deo treba čuvati u korpi dokle god korisnik ne odabere opciju za kraj kupovine.
* Na kraju kupovine, prikazati korisniku listu artikala koje je dodao u korpu i ponuditi da neki obriše, da se vrati nazad na kupovinu ili da potvrdi kupovinu.
* Ako korisnik potvrdi kupovinu, uzeti podatke o korisniku, među kojima su ime, prezime, poštanska adresa i adresa elektronske pošte, a korisniku prikazati izgled uplatnice, sa pozivom na broj koji je identifikacioni broj porudžbine, njegove podatke i podatke primaoca, sa prikazom tekućeg računa i svrhom uplate.

U zavisnosti od prioriteta,zahteve koji su gore definisani možemo klasifikovati u 3 kategorije:

1. **Suštinski**-korisniku se mora omogućiti pregledanje I poručivanje auto delova
2. **Poželjni**-auto delovi se mogu pretraživati po ceni
3. **Opcioni**-cena auto dela je ispisana podebljanim brojevima
   1. **Modelovanje ponašanja**

Modelovanje ponašanja sistema na osnovu prikupljenih zahteva doprinosi boljem razumevanju zahteva,lakšem uočavanju nedostataka u smislu nepoznatog ponašanja u pojedinim situacijama i lakšem otkrivanju nedoslednosti zahteva(isti ulazni signal daje različite vrednosti na izlazu).

Sledeći ER dijagram predstavlja model aplikacije za poručivanje auto delova.



*Slika 6. ER dijagram za online prodavnicu auto delova*

* 1. **Specifikacija i definicija zahteva**

Dokument sa zahtevima je jedinstven za sve učesnike na projektu I često se izrađuju dva dokumenta: **definicija zahteva I specifikacija zahteva**

**Definicija zahteva:**

1. Korisnik ne može da dodaje nove auto delove.
2. Dodavanje novih auto delova vrši isključivo administrator.
3. Korisnik ne može da se prijavi ako se nije registrovao.
4. Korisnik može da pretražuje auto delove po atributima.
5. Korisnik može da poručuje odabrane auto delove.
6. Korisnik može da dodaje izabrane auto delove u omiljenje proizvode.

**Specifikacija zahteva:**

* Aplikacija treba da omogući administratoru da doda novi auto deo u online prodavnicu.Za svaki auto deo koji se dodaje,treba da unese:naziv auto dela,opis,cenu,kategoriju kojoj pripada,model auta,tip proizvoda(da li je originalan ili zamenski deo),da li ima garanciju I ako ima na koliko meseci,stanje(dostupan/nedostupan deo) I vreme dostave.
* Korisnik mora da se registruje unošenjem svoje email adrese,lozinke,imena I prezimena I adrese stanovanja.Korisnik nakon registracije treba da se prijavi kako bi mogao da naručuje auto delove.Prijavljuje se svojom email adresom I šifrom kojom je prethodno registrovan.
* Neprijavljeni korisnici mogu samo da pregledaju I pretražuju auto delove po sledećim filterima:kategoriji auto delova(karoserija,sijalice,elektrika,akumulator,trap,pogon točkova,motor),nazivu auto dela,ceni,po modelu auta za koji je namenjen auto deo,po tipu auto dela(originalan/zamenski),garanciji,opisu I vremenu dostave.
* Kada se otvori početna stranica sa auto delovima,svaki auto deo treba da se prikazuje na sledeći način:slika datog auto dela treba da se prikaže sa leve strane I ona predstavlja uopštenu sliku za datu kategoriju,dok sa desne strane treba da se nadju bitne informacije o auto delu kao što su:Kategorija,naziv,cena,model auta,tip dela,garancija,opis,vreme dostave I stanje tj.dostupnost nekog dela.Na dnu prikaza svakog auto dela se nalazi ocena korisnika,ukupan broj korisnika koji su ocenili proizvod tj. auto deo kao I dugme za dodavanje u korpu I dugme za dodavanje u omiljenje proizvode.

1. **Projektovanje sistema**

Projektovanje sistema je proces prevodjenja problema u njegovo rešenje.

* 1. **Prikaz proizvoda**
* **Prijava i odjava administratora**

Administrator se na sistem prijavljuje sa ispravnim korisničkim imenom i lozinkom. Ako su parametri uneti kako treba, pristupa delu za dodavanje prozvoda, a ako nisu mora da ponovi postupak logovanja.

* **Dodavanje novog administratora**

Novi asministrator se dodaje kada se popuni forma za registraciju na ispravan način. Administrator takođe može da menja svoje lične podatke.

* **Pregled svih proizvoda**

Svi korisnici koji pristupe online prodavnici mogu da vrše pregled svih proizvoda na sajtu i mogu videte njihov status, odnosno da li su dostupni za poručivanje ili nisu.

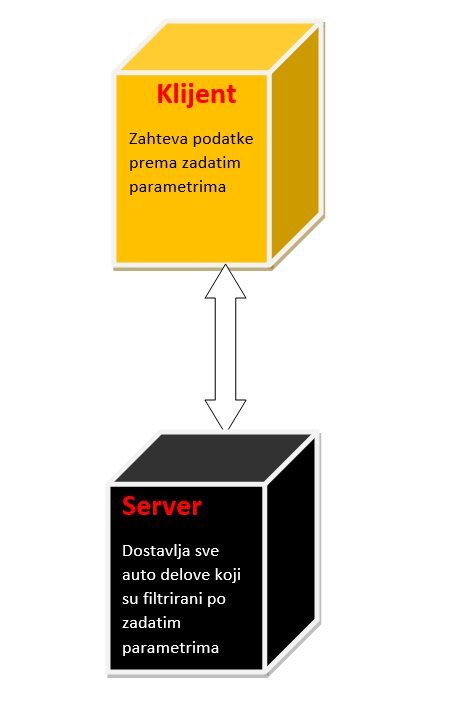
* **Pretraga proizvoda**

Svi korisnici mogu da vrše pretragu proizvoda po ključnoj reči.

* 1. **Projektovanje arhitekture**

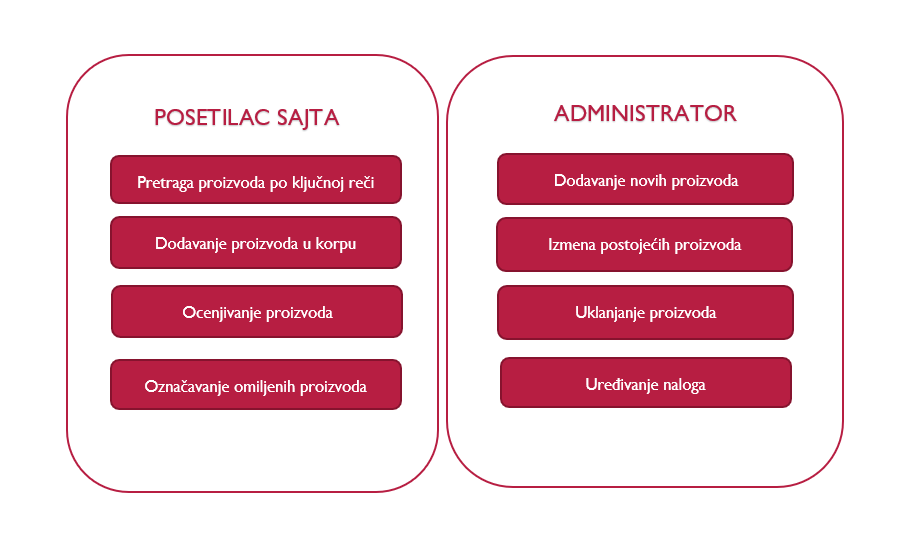
Korišćena je klijent-server arhitektura gde serverska komponenta pruža uluge klijentskim komponentama dok klijentske komponente zahtevaju podatke od servera.

Na slici 7. Je prikazana klijent-server arhitektura.



*Slika 7. Klijent-server arhitektura*

* 1. **Definisanje entiteta,atributa,relacija i ograničenja**



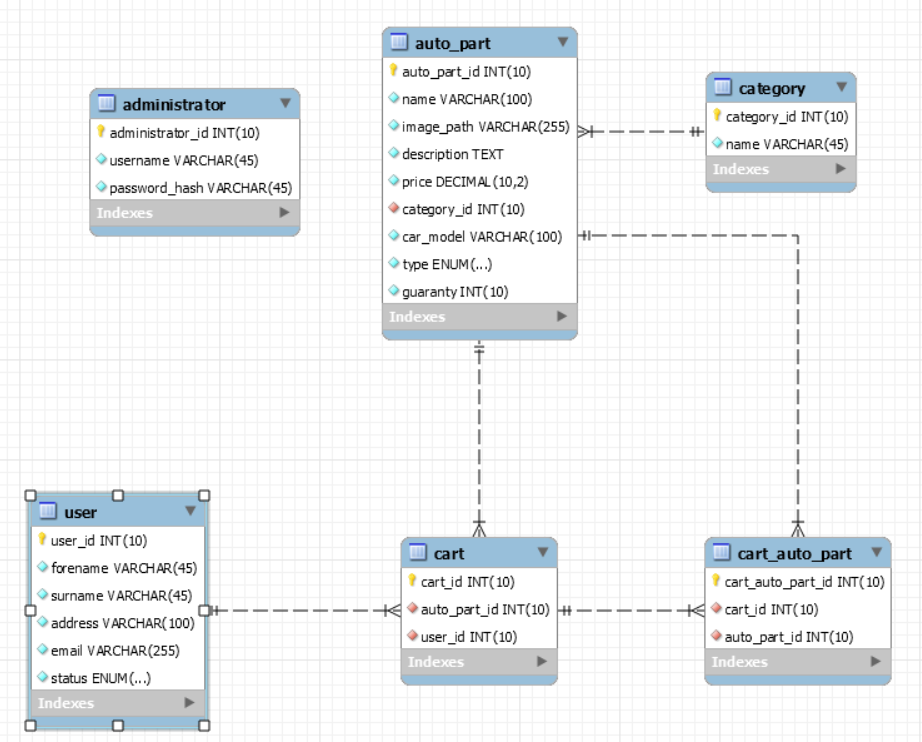
*Slika 8. Use case dijagram*

* 1. **Projektovanje baze podataka**

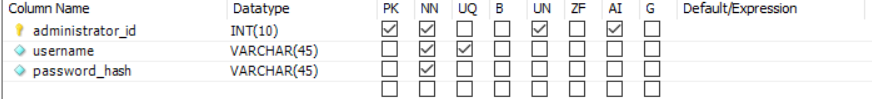
Za modelovanje baze podataka je korišćen MySql jezik.

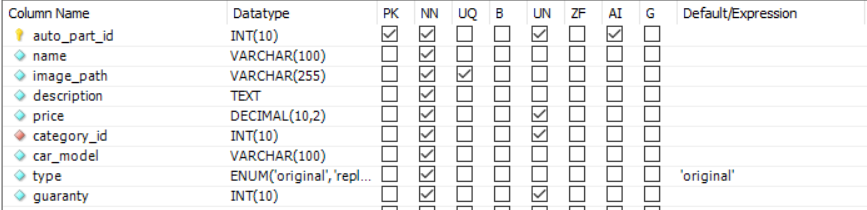
Tabele u bazi podataka su:

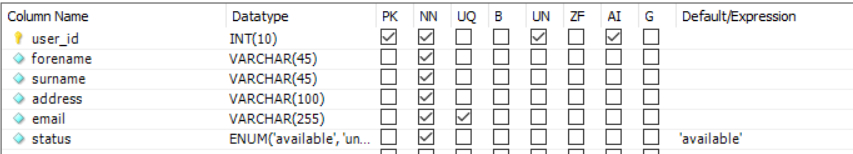
* Administrator
* Auto\_part
* Cart
* Cart\_auto\_part
* Category
* User

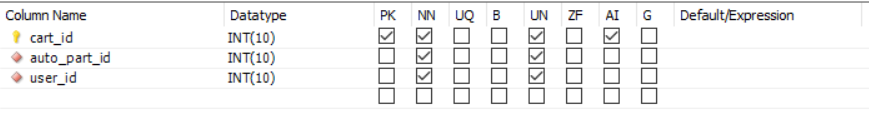


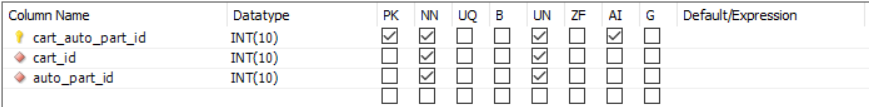
*Slika 9. Model baze podataka za online prodavnicu auto delova*













*Slika 10. Grafički prikaz entiteta I atributa*

* 1. **Ograničenja nad korisničkim akcijama**
* Svi posetioci sajta mogu da vrše pretragu proizvoda
* Dodavanje novih proizvoda mogu da vrše **samo** registrovani administratori
* Da bi korisnik naručio proizvod **mora** da ostavi lične podatke
* Korisnik može da pretražuje i porudžbine koje je ranije poručio
  1. **UML dijagram - dijagram klase**

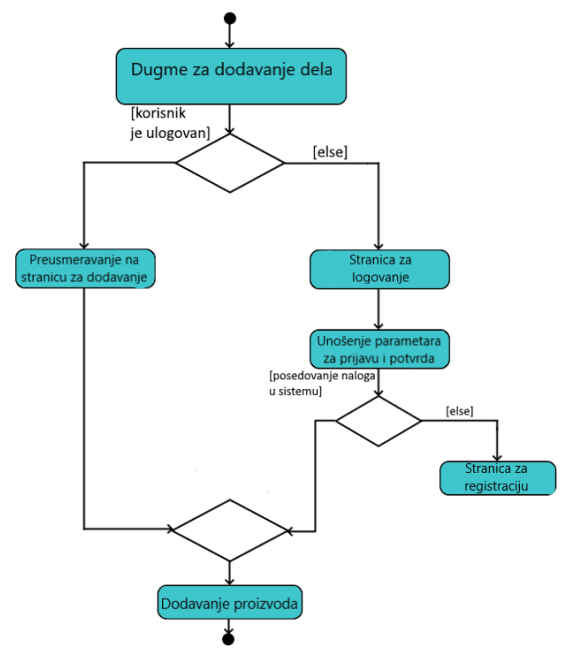
Dijagrami klase opisuju tipove entiteta u sistemu,veze koje postoje između entiteta I ograničenja u načinu njihovog povezivanja.



*Slika 11. Dijagram klase*

* 1. **UML dijagram – dijagram aktivnosti**

Dijagram aktivnosti prikazuje tok posla koji treba obaviti da bi administrator dodao novi proizvod tj. auto deo.

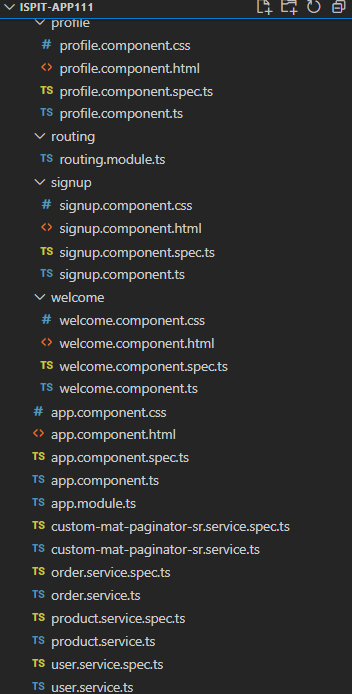
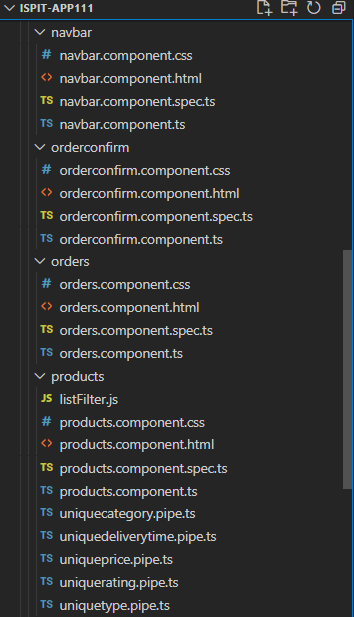
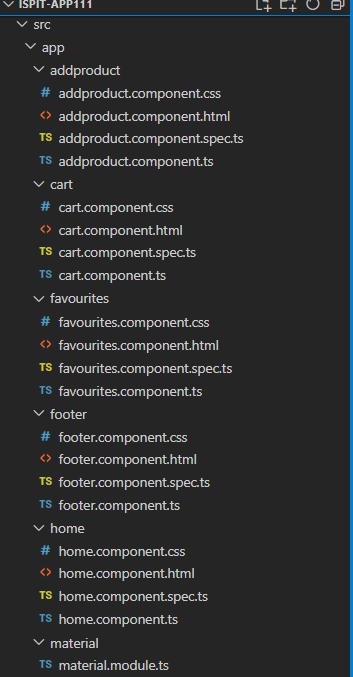


*Slika 12. Dijagram aktivnosti*

* 1. **Projektovanje programskog koda**

Za implementaciju programskog koda korišćen je Angular Framework koji je poslužio za sve aspekte prilikom razvoja aplikacije.U te aspekte spadaju razvoj interfejsa (frontend), simulacija logičkog pristupa (backend) kao i povezivanje baze podataka.

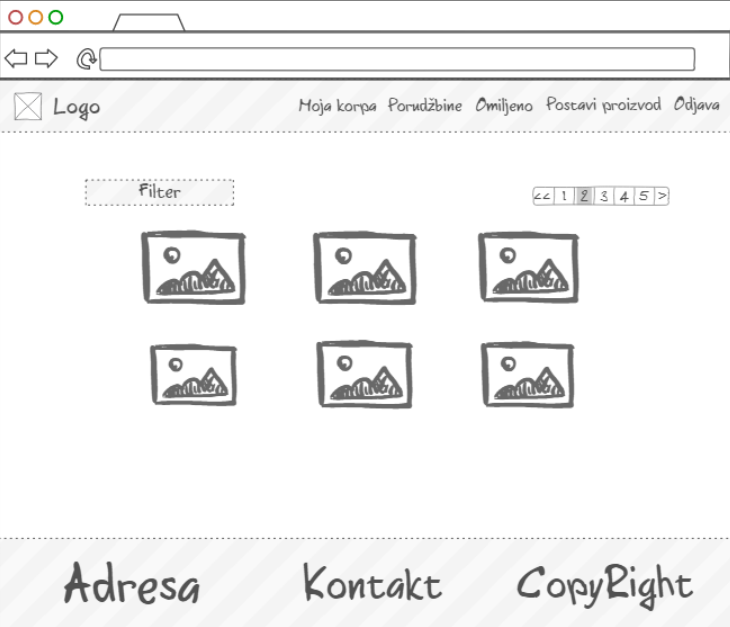
Prikaz interfejsa na strani klijeta omogućen je primenom HTML-a i CSS-a, dok je programska logika odrađena u TypeScript-u.



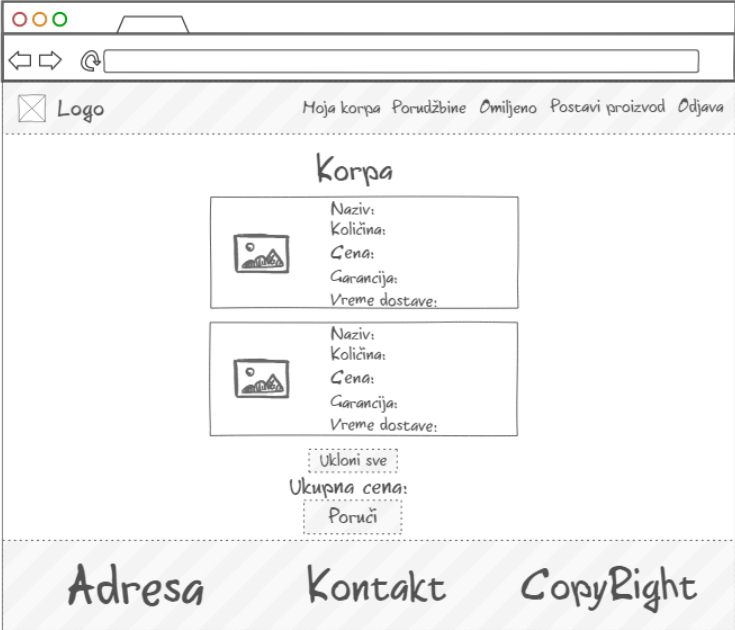
*Slika 13. Struktura programskih modula I komponenti*

* 1. **Projektovanje dizajna sistema**

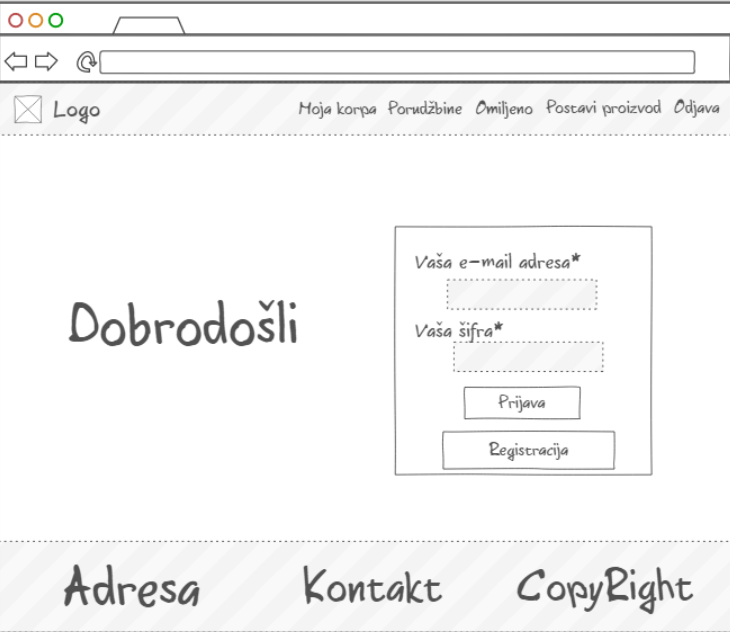
U ovom poglavlju osmišljavamo koncept veb aplikacije. Program koji smo koristili za izradu je NinjaMock.



*Slika 14. Prototip početne stranice*



*Slika 15. Prototip stranice sa svim dodatim proizvodima u korpu*

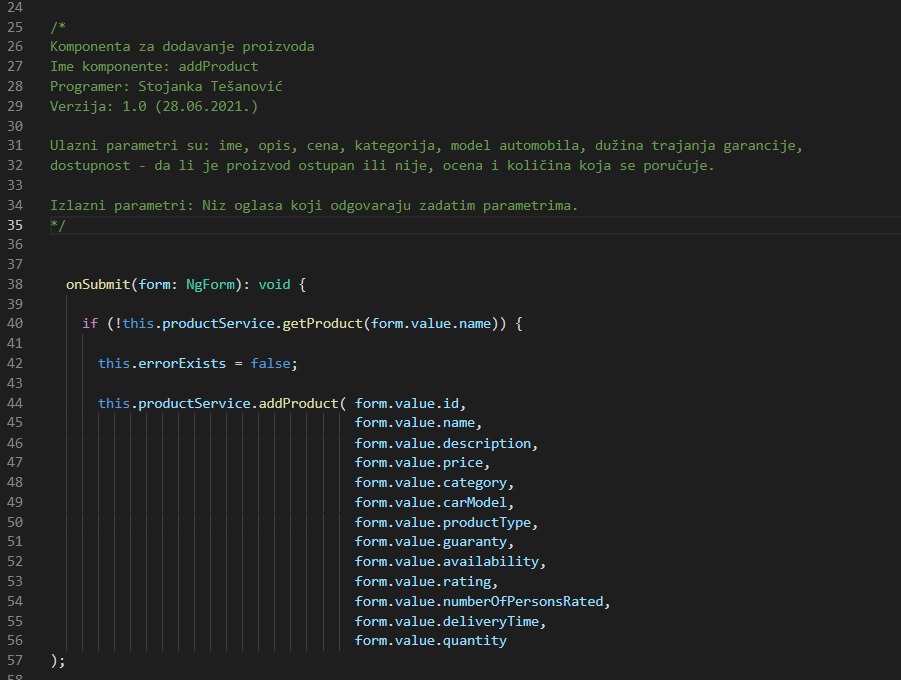
*Slika 16. Prototip stranica za prijavu I registrovanje korisnika*

1. **Implementacija programskog koda**

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:

* **addProduct()** – funkcija koja omogućava dodavanje novog proizvoda među postojeće. Kao ulazne parametre uzima već unapred predefinisane podatke koji su prikazani u obliku forme, a izlazni parametar je sam proizvod.
* **isPasswordCorrect()** – funkcija koja vrši proveru lozinke unete u korisničkom interfejsu. Kao ulazne parametre uzima e-mail i lozinku, a kao izlazni parametar vraća da li je lozinka dobro unešena ili nije.
* **loggedInCheck()** – funkcija koja izvršava proveru da li je administrator ulogovan u aktivnoj sesiji. Namena ove funkcije je da spreči neautorizovano dodavanje proizvoda. Ukoliko administrator nije ulogovan, oglašava se obaveštenje da administrator nije ulogovan i da mora da se uloguje i preusmerava ga na stranicu za prijavu.

Na slici 17. Je prikazana unutrašnja dokumentacija koja treba da pomogne da se kod lakše shvati I protumači.



*Slika 17. Grafički prikaz programskog koda sa unutrašnjom dokumentacijom*

1. **Testiranje softvera**

Za potrebe testiranja softvera, Angular omogućava korišćenje sopstvenih test klasa. Pomoću komande „ng test“ pravi se izdvojeno okruženje koje radi na drugom portu nezavisno od same aplikacije. Struktura testiranja je u Karma i Jasmine okruženjima.

* 1. **Jedinično testiranje**
* Prilikom pokušaja da se izbegne unošenje neodgovarajućih parametara dolazi do izbacivanja poruke “Polje ne sme da bude prazno”
* Test primer za potrebe testiranja komponente za prijavu:

-Koristimo e-mail [test1@gmail.com](mailto:test1@gmail.com), a za lozinku ostavljamo prazno polje

Pošto ništa nije uneto u prazno polje, dolazi do greške i izbacuje se poruka koja glasi “Polje ne sme biti prazno”

Ovde takođe postoji ograničenje što se tiče dužine, odnosno lozinka mora da bude veća od određenog broja karaktera (dužina lozinke > 8).Ako je dužina manja od datog broja, korisnik dobija obaveštenje sadržine “Lozinka mora da ima više od 8 karaktera”.

* Za potrebe testiranja unosimo ispravnu e-mail adresu i pogrešno korisničko ime. Korisnik dobija obaveštenje koje glasi “Pogrešna lozinka”

Ulazni podatak za e-mail je [test123@gmail.com](mailto:test123@gmail.com), a za lozinku je test1234

Za datu e-mail adresu lozinka je: test123456 i zato dolazi do izbacivanja obaveštenja koje glasi “Pogrešna šifra”

* 1. **Integraciono testiranje**
* Za potrebe provere saradnje između između komponenata za prijavu i registraciju, kao test primer unosili smo u polja za prijavu parameter koji ne postoje u evidenciji registrovanih administratora.

U evidenciji registrovanih korisnika postoje:

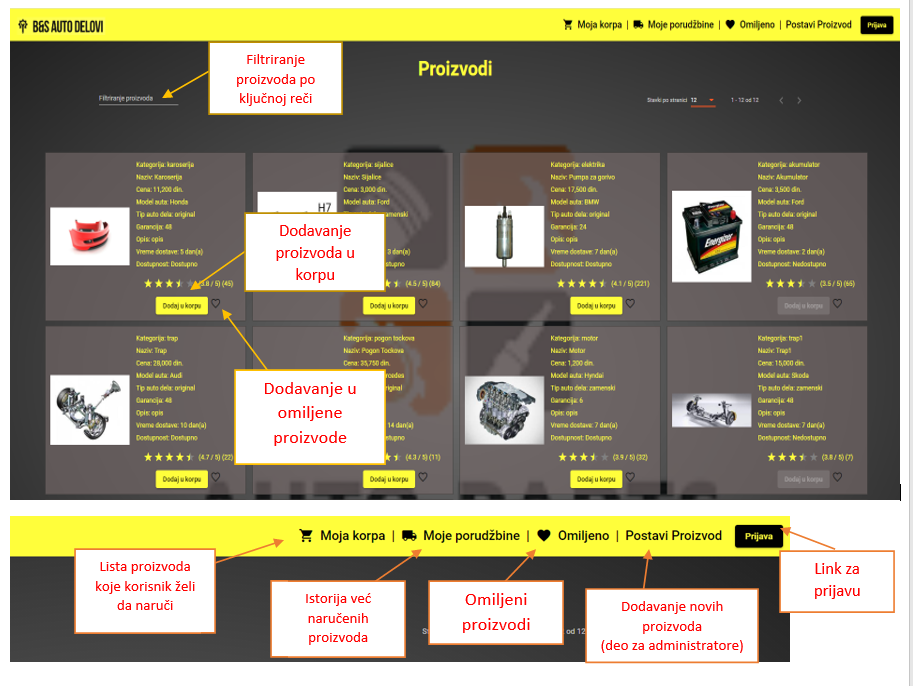
{ id: 1, “test1@gmail.com”, lozinka: “test123456”, name: “Bojana”, surname: “Mitrović”, address: “Svetozara Šemića 16, Pančevo”}

{ id: 2, “test1@gmail.com”, lozinka: “test654321”, name: “Stojanka”, surname: “Tešanović”, address:”Skadarska 78, Pančevo”}

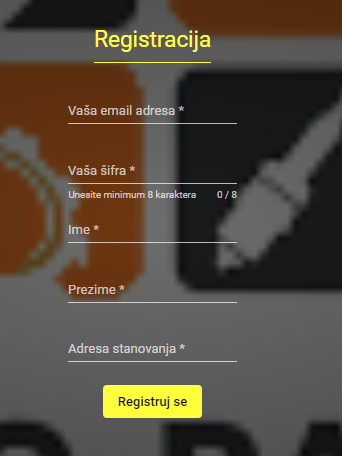
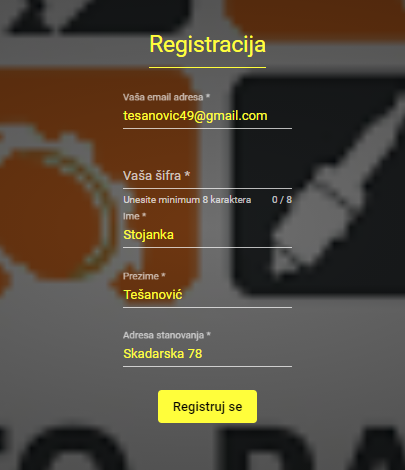
* Kao primer smo uzeli e-mail prvog administratora i lozinku drugog administratora. Odnosno e-mail: [test1@gmail.com](mailto:test1@gmail.com) i lozinku: “test654321”. S’obzirom da administrator sa takvim parametrima ne postoji, izlazi poruka “Nema registrovanog korisnika sa takvim parametrima”.

1. **Isporuka I održavanje softvera**

U ovom poglavlju biće objašnjen način upotrebe softvera kako bi budući korisnici mogli lakše da se snađu.



*Slika 18. Prikaz opcija za filtriranje,dodavanje u korpu I u omiljenje proizvode I prikaz opcija iz navigacionog menija*

*Slika 19. Prikaz nepopunjene i popunjene forme za registraciju*