Objektno orijentirani razvoj programa

Ak. god. 2015./2016.

WEB APLIKACIJA ZA PRODAJU

AUTO – DIJELOVA

Dokumentacija

Studenti: *Josip Kovaček (1191212746),*

*Matej Jemrić (0246045018)*

Datum predaje: *travanj, 2016.*

Nastavnik: Dr.sc. Marko Horvat, pred.

Sadržaj

[Sadržaj 2](#_Toc449352103)

[1. Dnevnik promjena dokumentacije 3](#_Toc449352104)

[2. Opis projektnog zadatka 4](#_Toc449352105)

[3. Pojmovnik 5](#_Toc449352106)

[4. Funkcionalni zahtjevi 6](#_Toc449352107)

[Sekvencijalni dijagrami: 12](#_Toc449352108)

[5. Ostali zahtjevi 15](#_Toc449352109)

[6. Arhitektura i dizajn sustava 16](#_Toc449352110)

[6.1. Svrha, opći prioriteti i skica sustava 16](#_Toc449352111)

[6.2. Dijagram razreda s opisom 18](#_Toc449352112)

[6.3. Dijagram komponenti 21](#_Toc449352113)

[7. Implementacija i korisničko sučelje 22](#_Toc449352114)

[7.1. Dijagram razmještaja 22](#_Toc449352115)

[7.2. Korištene tehnologije i alati 23](#_Toc449352116)

[7.3. Upute za instalaciju 24](#_Toc449352117)

[7.4. Korisničke upute 25](#_Toc449352118)

[8. Reference 29](#_Toc449352119)

# Dnevnik promjena dokumentacije

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev.** | **Opis promjene/dodatka** | **Autor(i)** | **Datum** |
| 0.1 | Napravljen predložak, dodan opis projektnog zadatka | Jemrić, Kovaček | 19.03.2016. |
| 0.2 | Popunjavanje 2.,3.,4.,5. poglavlja | Kovaček, Jemrić | 26.03.2016. |
| 0.3 | Izrada dijagrama obrazaca uporabe | Kovaček, Jemrić | 20.4.2016. |
| 0.4 | Izrada dijagrama razreda | Kovaček, Jemrić | 20.4.2016. |
| 0.5 | Dorada i završetak 2. i 3. poglavlja | Kovaček, Jemrić | 20.4.2016. |
| 0.6 | Dorada i završetak 4. i 5. poglavlja | Kovaček, Jemrić | 21.4.2016. |
| 0.7 | Dorada i završetak 6. poglavlja | Kovaček, Jemrić | 22.4.2016. |
| 0.8 | Dorada i završetak 7. poglavlja | Kovaček, Jemrić | 23.4.2016. |
| 0.9 | Manje izmjene, ispravci, dopune | Kovaček, Jemrić | 24.4.2016. |
| **1.0** | **Konačna verzija dokumentacije** | **Kovaček, Jemrić** | **25.4.2016.** |

# Opis projektnog zadatka

Glavni cilj projekta je stvaranje web - shop aplikacije koja omogućuje korisnicima brzu, jednostavnu pretragu i kupovinu auto - dijelova. Takav sustav korisnicima omogućuje ugodnije iskustvo kod nabave željenog proizvoda za svoj automobil, a ujedno pruža administratorima bolji uvid u što korisnici pretražuju i koje su potrebe tržišta te prednost nad konkurencijom.

Iako postoje razne web - aplikacije za prodaju auto – dijelova najčešće ne nude odabir proizvoda na jednostavan način. Cilj je razviti sustav u kojem korisnik u što manje koraka dođe do željenog proizvoda te ima pregled stanja odabranih proizvoda u bilo kojem trenutku kupnje.

Izazovi koje aplikacija treba riješiti su:

- filtriranje proizvoda

- aktualno stanje proizvoda ( dostupnost proizvoda )

- privatnost i sigurnost kupovine

- jednostavnost upotrebe

- generiranje izvještaja o najčešće biranim proizvodima

Kroz ovo rješenje želi se omogućiti korisnicima prednost nad ostalim web – shop aplikacijama te konkurentnost na tržištu auto - dijelova.

Mogućnosti za daljnju nadogradnju predviđene su kroz izradu dodatnih servisa za dohvaćanje informacija o novim, aktualnim proizvodima direktno od proizvođača ili nabavljača. Na taj način omogućuje se korisnicima direktan pristup novitetima.

Predviđeno rješenje ne obuhvaća proces plaćanja putem servisa sigurne kupovine koje pružaju za to specijalizirane firme i banke. (*eng. Internet Payment Gateway*).

# Pojmovnik

Web - shop – vrsta trgovine u realnom vremenu u kojoj kupci kupuju proizvode putem Interneta.

Administrator – korisnička uloga koja upravlja aplikacijom.

Web servis – Resurs preko kojeg aplikacija komunicira s drugim aplikacijama ili korisnikom preko web preglednika.

IPG (Internet Payment Gateway) – pružatelj usluge koja autorizira plaćanja putem interneta

# Funkcionalni zahtjevi

Procesi sustava su:

* ažuriranje i dodavanje proizvoda od strane administracije
* registracija i prijava korisnika od strane korisnika
* pregledavanje detalja proizvoda i kupovina od strane korisnika

Koraci korištenja aplikacije (engl. Use Case) od strane korisnika:

* Anonimni korisnici (gosti) vide samo ekran za registraciju te ne mogu obaviti kupovinu bez registracije i prijave na web shop
* Anonimni korisnici mogu izvršiti registraciju korisnika
* Anonimni korisnici mogu izvršiti prijavu korisnika
* Kupac ima mogućnost odabira i pregleda detalja odabranog proizvoda
* Kupac ima mogućnost dodavanja proizvoda u košaricu
* Kupac ima mogućnost pregleda košarice
* Kupac ima mogućnost završiti kupovinu
* Sustav može primijeniti popust na kupovinu

Koraci korištenja aplikacije (engl. Use Case) od strane administracije:

* Administrator može dodati novi proizvod u bazu podataka
* Administrator može obrisati postojeći proizvod iz baze podataka
* Administrator može urediti postojeći proizvod u bazi podataka

**Dionici:**

* Gost
* Kupac
* Administrator

**Aktori i njihovi funkcionalni zahtjevi:**

* Kupac, inicijator:
  + registrira se i prijavljuje u sustav
  + dodaje odabrane proizvode u košaricu
  + odjavljuje se iz sustava
  + pretražuje proizvode
  + pregledava odabrane proizvode (košaricu)
  + briše odabrani proizvod iz košarice
  + završava kupovinu
* Administrator, inicijator:
  + pregled proizvoda
  + uređivanje postojećih proizvoda
  + dodavanje novih proizvoda
  + brisanje postojećih proizvoda
* Baza podataka, sudionik:
  + sadrži podatke o korisnicima, korisničkim ulogama,proizvodima i narudžbama
* Gost, inicijator:
  + mogućnost prijave
  + mogućnost registracije
* Poslužitelj, sudionik:
  + pristupa bazi podataka na zahtjev korisnika sustava

**Opis obrazaca uporabe:**

* UC1 – *Registracija korisnika*
  + **Glavni sudionik**: gost
  + **Cilj:** Registrirati korisnika te mu omogućiti prava te pristup web-shop aplikaciji, unijeti podatke u bazu podataka
  + **Sudionici:** Baza podataka
  + **Preduvjet:** Pristup računalu i Internetu
  + **Rezultat**: Korisnik je registriran u sustav
  + **Željeni scenarij:**

- korisnik preko računala i web preglednika pristupa aplikaciji

- kod pristupa web - shop aplikaciji unosi potrebne podatke u formu za registraciju

* **Mogući scenarij:**

- korisnik popunjava formu za registraciju s neispravnim podacima te

mu nije omogućena registracija ukoliko ne unese ispravne podatke

* UC2 – *Prijava korisnika*
  + **Glavni sudionik**: gost
  + **Cilj:** Autorizacija korisnika u svrhu dobivanja pristupa odabiru i rezervaciji proizvoda (autodijelovi)
  + **Sudionici:** Baza podataka
  + **Preduvjet:** Korisnik registriran u sustav, pristup Internetu i računalu

(mobitel, tablet)

* + **Rezultat**: Korisnik je prijavljen u sustav
  + **Željeni scenarij:**

- korisnik preko računala i web preglednika pristupa web-shop aplikaciji - kod pristupa stranici unosi potrebne podatke u formu za prijavu

koristeći postojeći e-mail i lozinku.

- korisnik je uspješno prijavljen u sustav te ostvaruje prava rezervacije i

kupovine autodijelova

* **Mogući scenarij:**

- korisnik popunjava formu za prijavu s neispravnim podacima te se ne

može prijaviti na sustav

* UC3 – *Ažuriranje baze podataka*
  + **Glavni sudionik**: administrator
  + **Cilj:** Dodati novi proizvod, promijeniti postojeći ili obrisati postojeći proizvod
  + **Sudionici:** Baza podataka
  + **Preduvjet:** Administrator s potrebnim ovlastima prijavljen u sustav
  + **Rezultat**: Ažurirana baza podataka
  + **Željeni scenarij:**

- administrator nakon prijavljivanja na stranicu ima mogućnost odabira

dodavanja brisanja i promjenu postojećih proizvoda

- administrator unosi novi proizvod, ažurira se baza podataka tako da

kupac može vidjeti proizvod

- administrator briše postojeći proizvod, ažurira se baza podataka

- administrator mijenja vrijednosti postojećeg proizvoda, ažurira se baza

podataka tako da kupac vidi promijenu napravljenu nad proizvodom

* **Mogući scenarij:**

- administrator obriše krivi proizvod

- administrator unese krive podatke o postojećem proizvodu

* UC4 – *Dodati proizvod u košaricu*
  + **Glavni sudionik**: kupac
  + **Cilj:** Dodati odabrani proizvod u košaricu
  + **Sudionici:** Baza podataka
  + **Preduvjet:** Korisnik prijavljen u sustav s odgovarajućom ulogom (uloga kupac), postojanje traženog proizvoda
  + **Rezultat**: Dodan odabrani proizvod u košaricu
  + **Željeni scenarij:**

- kupac nakon prijavljivanja na stranicu vidi tražilicu pomoću koje

pretražuje proizvode

- kupac upisuje naziv ili dio naziva proizvoda u tražilicu, potom mu se

ispisuju svi mogući proizvodi

- kupac nakon što pronađe željeni proizvod, dodaje proizvod u košaricu

* **Mogući scenarij:**

- kupac unosi neispravan naziv proizvoda

- kupac unosi naziv proizvoda kojeg nema u ponudi

* UC5 – *Pregledati košaricu*
  + **Glavni sudionik**: kupac
  + **Cilj:** Provjeriti stanje košarice i kad je kupac zadovoljan odabirom može završiti kupovinu
  + **Sudionici:** Baza podataka
  + **Preduvjet:** Korisnik prijavljen u sustav s odgovarajućom ulogom (uloga kupac), imati minimalno jedan artikl dodan u košaricu
  + **Rezultat**: Provjereno stanje košarice i završena kupnja
  + **Željeni scenarij:**

- kupac nakon odabranog proizvoda može provjeriti stanje košarice, ukoliko je zadovoljan odabranim može također završiti kupovinu ili nastaviti s odabirom proizvoda

* **Mogući scenarij:**

- kupac odustaje od kupovine, kupac uklanja odabrani proizvod

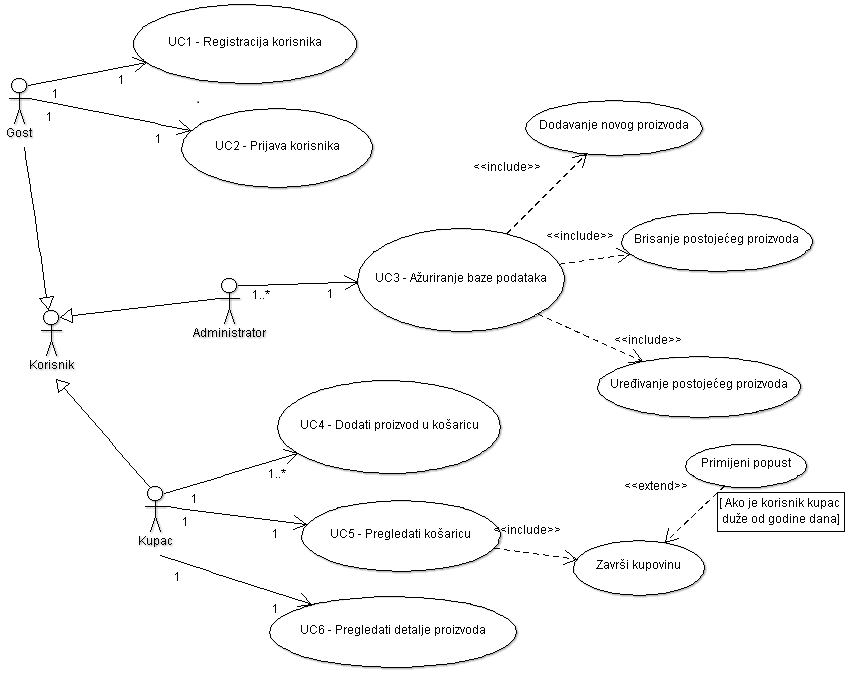
* UC6 – *Pregledati detalje proizvoda*
  + **Glavni sudionik**: kupac
  + **Cilj:** Prikazati detaljne informacije o proizvodu
  + **Sudionici:** Baza podataka
  + **Preduvjet:** Korisnik prijavljen u sustav s odgovarajućom ulogom (uloga kupac), odabrani proizvod za koji želimo dodatne informacije
  + **Rezultat**: Prikazane detaljne informacije o proizvodu
  + **Željeni scenarij:**

- kupac kad pronađe željeni proizvod pomoću tražilice, odabire željeni proizvod

- kad korisnik pritisne na gumb „Detalji proizvoda“ prikazani su mu detalji proizvoda

* **Mogući scenarij:**

- nema

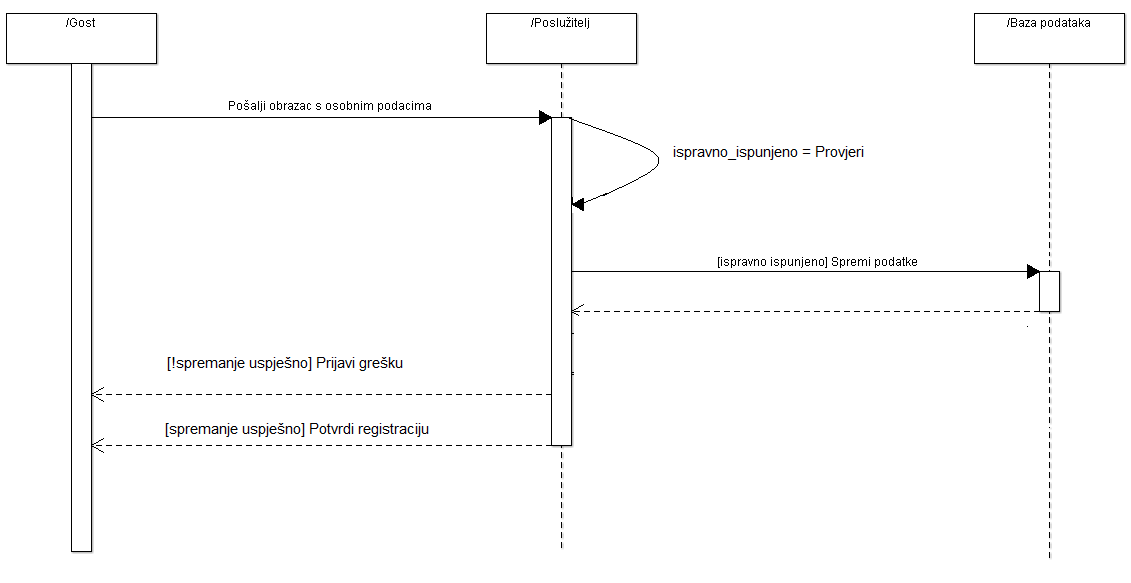


Slika : Obrazac uporabe web-shop aplikacije

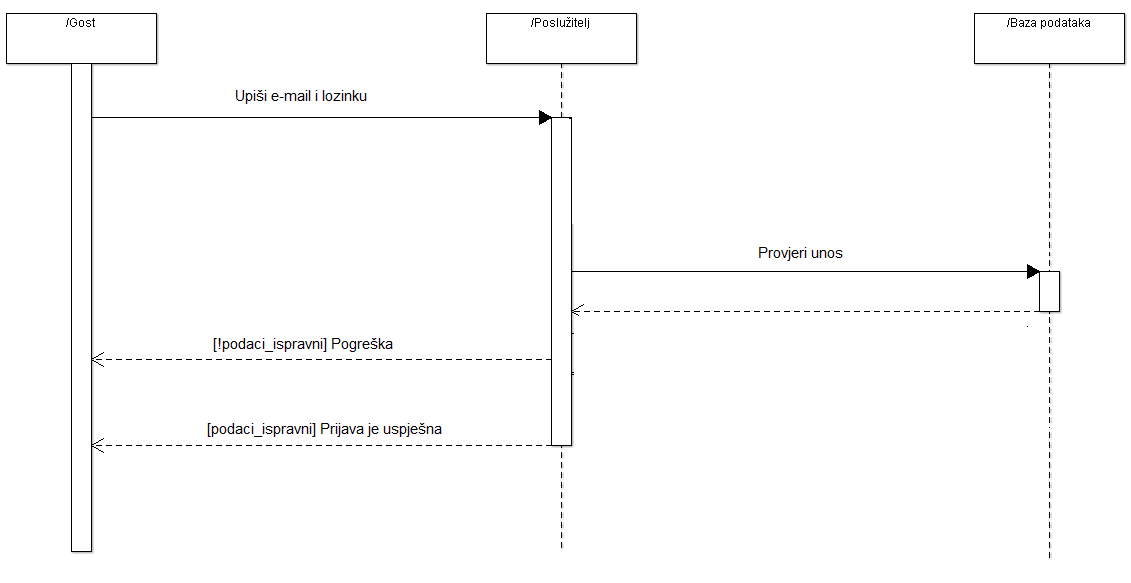
1: Obrazac uporabe web-shop aplikacije

Baza podataka je sudionik povezan sa svim obrascima uporabe (UC1-UC6) vezom asocijacije.

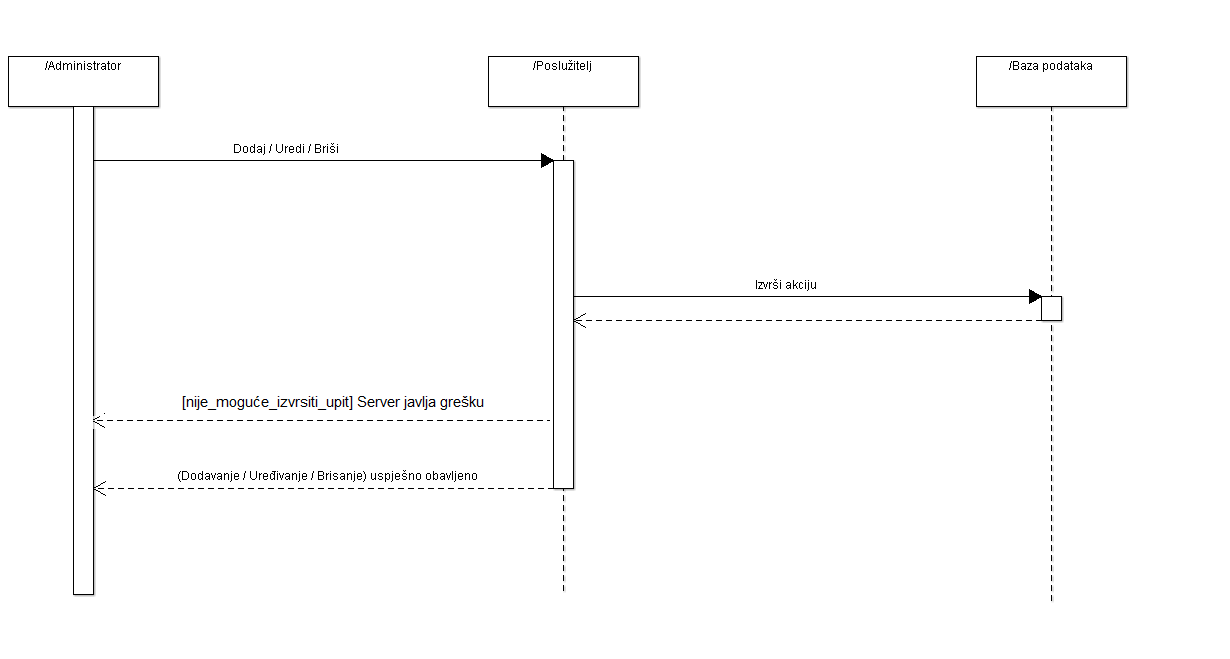
## Sekvencijalni dijagrami:



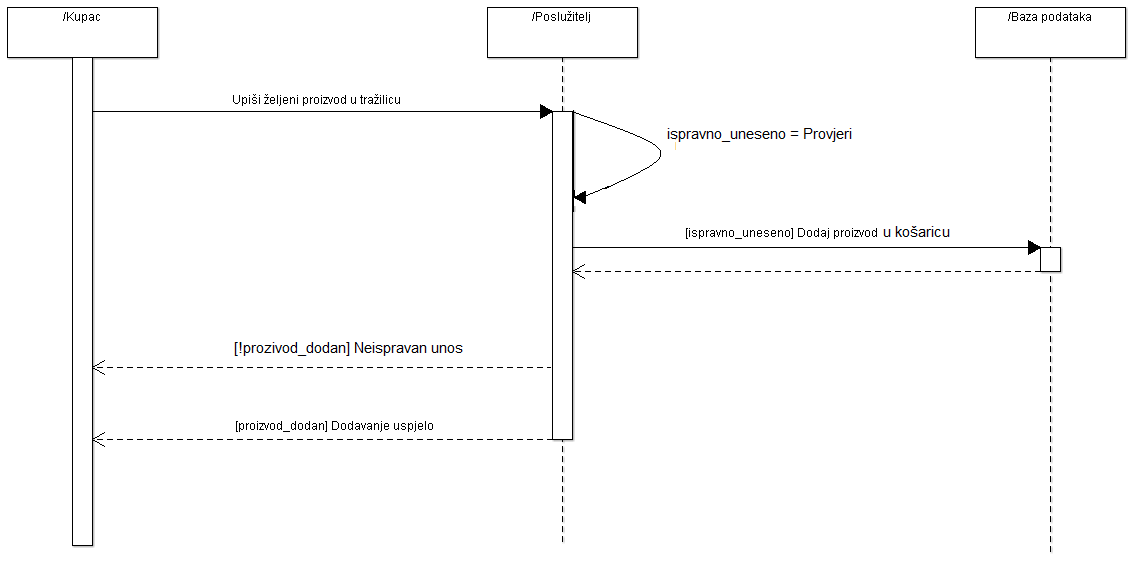
Slika :UC1 - Sekvencijalni dijagram – registracija korisnika



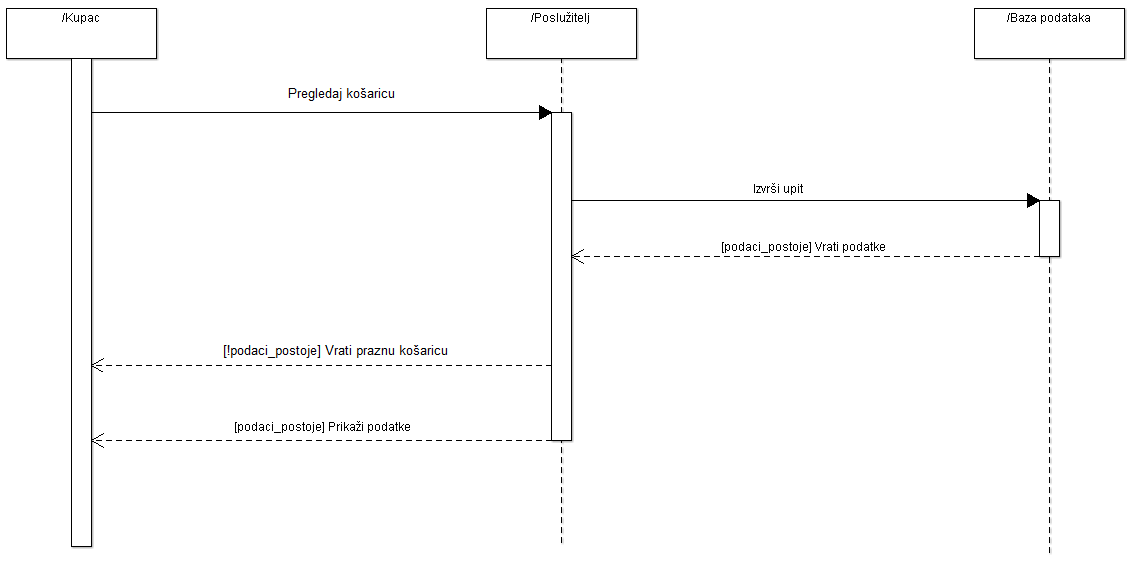
Slika : UC2 - Sekvencijalni dijagram – prijava korisnika



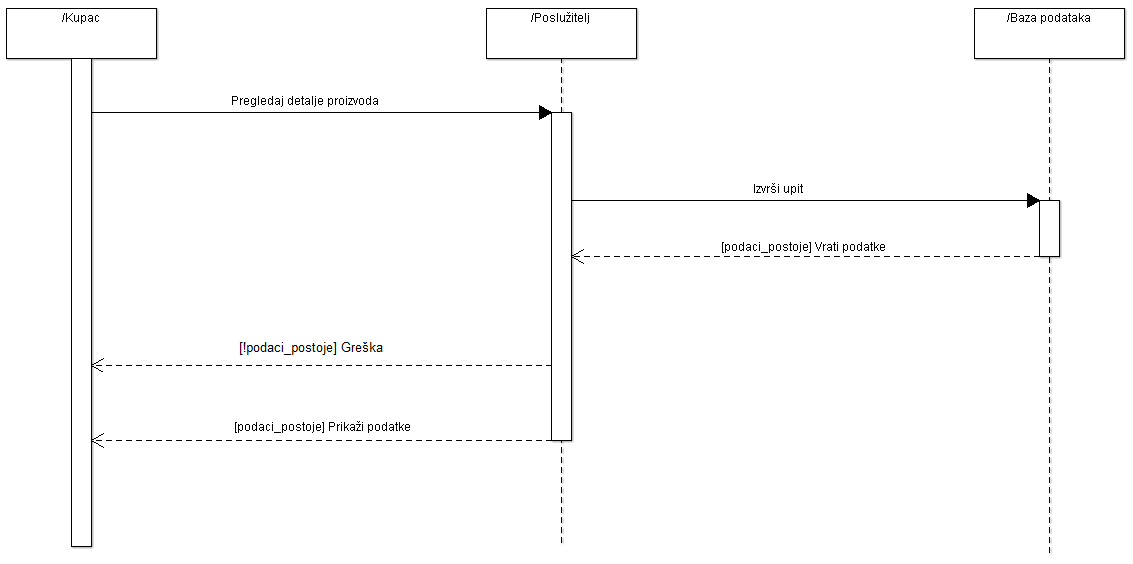
Slika : UC3 - Sekvencijalni dijagram – ažuriranje baze



Slika : UC4 - Sekvencijalni dijagram – pretraga proizvoda



Slika : UC5 - Sekvencijalni dijagram – pregled narudžbe



Slika : UC6 - Sekvencijalni dijagram – pregled detalja proizvoda

# Ostali zahtjevi

**Zahtjevi kvalitete:**

* Automatsko skaliranje sustava ovisno o broju korisnika koji su pristupili web shopu kako korisnici ne bi došli u situaciju da ne mogu pristupiti kupovini zbog preopterećenosti ili usporenja sustava.
* Sve akcije od strane korisnika se trebaju obaviti u razumnom vremenu tako da korisnik ne mora čekati na tražene rezultate. Zahtijevano vrijeme trajanja za sve akcije je da se obave unutar jedne do dvije sekunde.
* Responzivni dizajn stranice tako da korisnik može pristupiti i obaviti kupovinu na bilo kojem uređaju ( mobitel, tablet, laptop, računalo i drugi).
* Sva komunikacija korisnika i sustava mora biti zaštićena kako bi korisnici i njihovi podaci bili sigurni i privatni.
* Web - shop mora biti jednostavan za uporabu te jasno shvatljiv ne - tehničkom osoblju.
* Otpornost na greške, npr. neispravan unos od strane korisnika ne smije imati utjecaj na stanje sustava

**Ograničenja:**

* Da bi se sustav nadogradio potrebno je imati pristup izvornom kodu i poznavati programski jezik u kojem je aplikacija izrađena.
* Za funkcionalnost sustava korisnik mora imati pristup Internetu

# Arhitektura i dizajn sustava

## Svrha, opći prioriteti i skica sustava

Sustav je baziran na klijent – poslužitelj arhitekturi. To je model u kojem su dijelovi aplikacije podijeljeni između strane koja pruža neki resurs (poslužitelj) i strane koja traži taj resurs (klijent). Arhitektura sustava je objektno orijentirana. Razlog odabira objektno orijentirane arhitekture je njena stabilnost, pouzdanost te lakše održavanje sustava. Sustav je podijeljen na slojeve što ga čini lakšim za održavanje pri čemu se dobiva modularnost gdje bilo koji od slojeva u nekom trenutku (npr. dolazak nove tehnologije) može biti zamijenjen s nekom drugom tehnologijom bez da to posebno utječe na rad ostalih slojeva. Svaki sloj ima svoju zasebnu zadaću o kojoj brine. Korištenjem modernih programskih okvira olakšava se razvoj i smanjuje mogućnost pogreške kod implementacije, oslobađa se programere od pisanja nepotrebnog koda i ubrzava se razvoj.

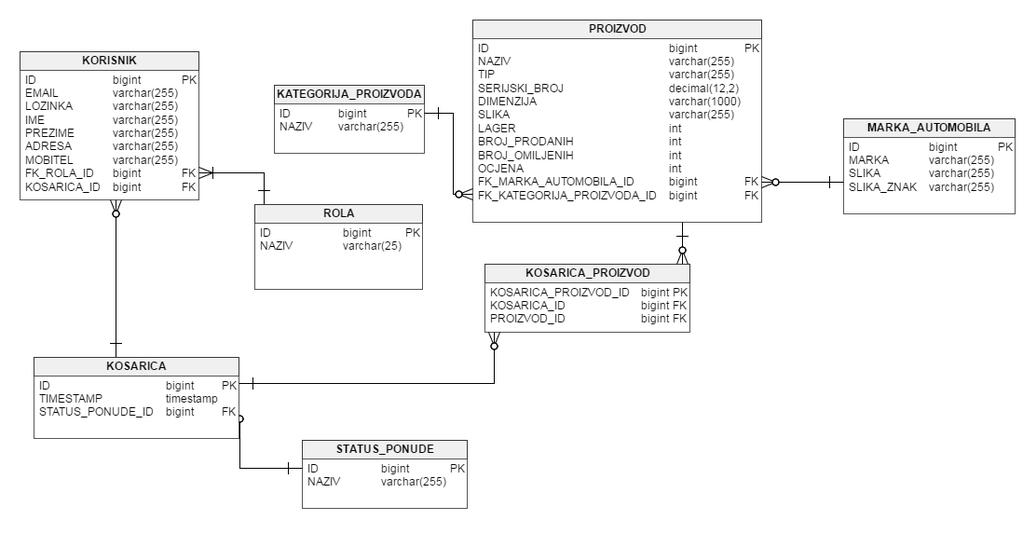
Aplikacija je podijeljena na 3 sloja: prezentacijski, aplikacijski i sloj perzistencije. Pod prezentacijom se smatra ono što korisnik vidi, dakle prikaz u pregledniku. Aplikacijski sloj podrazumijeva tehnologije koje procesuiraju podatke i omogućuju pregledniku da ih prikaže korisniku dok perzistenciju čine tehnologije koje serviraju aplikacijski sloj s

podacima ili pohranjuju rezultate obrade koje su nastale u aplikacijskom sloju.

Prezentacijski sloj ove aplikacije čine HTML predlošci i AngularJS javascript framework. Kod inicijalizacije stranice AngularJS povezuje traženu HTML datoteku s odgovarajućim AngularJS kontrolerom koji preuzima odgovornost za daljnju komunikaciju s aplikacijskim serverom te iz HTML predloška kreira stvarnu stranicu koju korisnik vidi.

Aplikacijski sloj je interno podijeljen na Spring kontrolere, modele (entitete) i sloj za pristup podacima (repozitoriji). Svaki entitet ima svoj repozitorij u sloju za pristup podacima. Repozitoriji sadrže metode za dohvaćanje i spremanje entiteta ili metode s posebnim upitima na bazu koje dohvaćaju samo određene podatke.

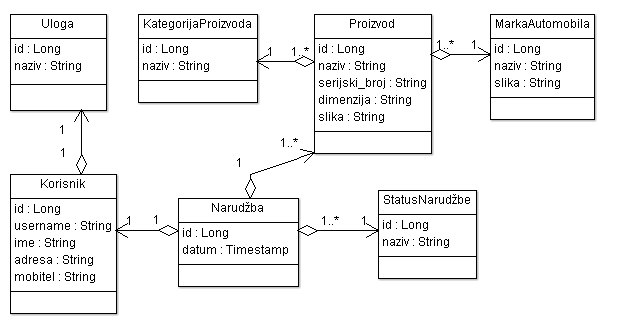
Za perzistenciju podataka koristi se Hibernate programski okvir. Hibernate povezuje modele (entitete) s tablicama u relacijskoj bazi podataka te se brine za dohvaćanje i ažuriranje podataka.



Slika :Model baze podataka

## Dijagram razreda s opisom

**Sloj modela:**



Slika 9: Dijagram razreda modela

**Uloga**

Predstavlja ulogu koju korisnik ima u aplikaciji. Ovo je šifrarnik sa mogućim vrijednostima „Administrator, Kupac, Gost“

**Korisnik**

Predstavlja entitet korisnika u aplikaciji. Definiran je svojim imenom, korisničkim imenom, adresom, mobitelom i ulogom.

**Kategorija proizvoda**

Šifrarnik mogućih kategorija proizvoda.

**Marka automobila**

Šifrarnik mogućih marki automobila.

**Proizvod**

Entitet koji predstavlja dio automobila koji korisnik može pretraživati, pregledati detalje, odabrati i naručiti. Definiran je svojim nazivom, serijskim brojem i dimenzijama.

**Status narudžbe**

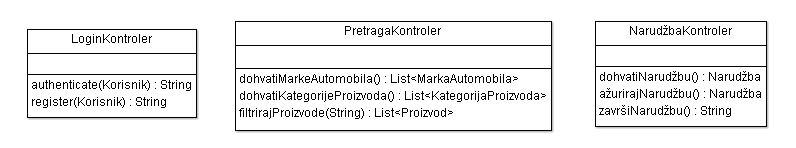
Šifrarnik koji sadrži moguća stanja narudžbe. Moguće vrijednosti su „Odabrano, Naručeno, Plaćeno“

**Narudžba**

Predstavlja ponudu proizvoda koje korisnik odabire i naručuje. Definirana je skupom odabranih proizvoda, statusom i datumom.

**Sloj kontrolera:**

Ovaj sloj zaprima korisničke zahtjeve, validira ih, obrađuje i odgovara na njih s podacima. Sastoji se od sljedećih razreda:



Slika 10: Dijagram razreda kontrolera

**LoginKontroler**

Služi za prijavu i registraciju korisnika. Kod prijave provjerava da li podaci korisnika odgovaraju onima u bazi. Kod registracije provjerava da li su svi podaci ispravno popunjeni te sprema novog korisnika. U slučaju greške vraća statusnu poruku sa detaljima greške.

**PretragaKontroler**

Služi za dohvat i filtriranje podataka koje korisnik pretražuje. Prosljeđuje upit na bazu te vraća dobiveni rezultat korisniku. Također dohvaća podatke iz šifrarnika (marke automobila i kategorije proizvoda)

**NarudžbaKontroler**

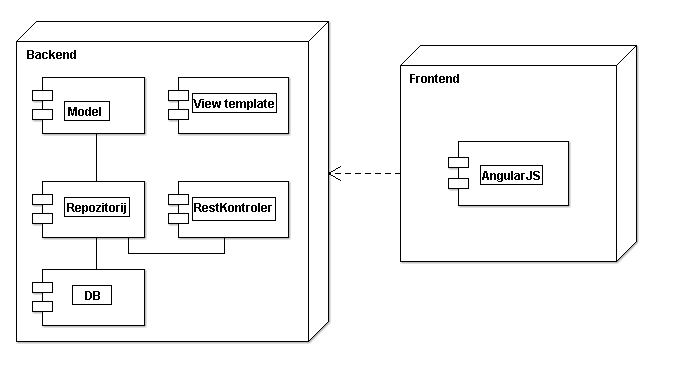
Služi za dohvat i ažuriranje narudžbe korisnika, odnosno proizvoda koje korisnik dodaje u košaricu.

## Dijagram komponenti

Komponente sustava su podijeljene na backend i frontend dio.

Backend dio čine model, kontroleri, repozitoriji, baza i HTML predlošci.

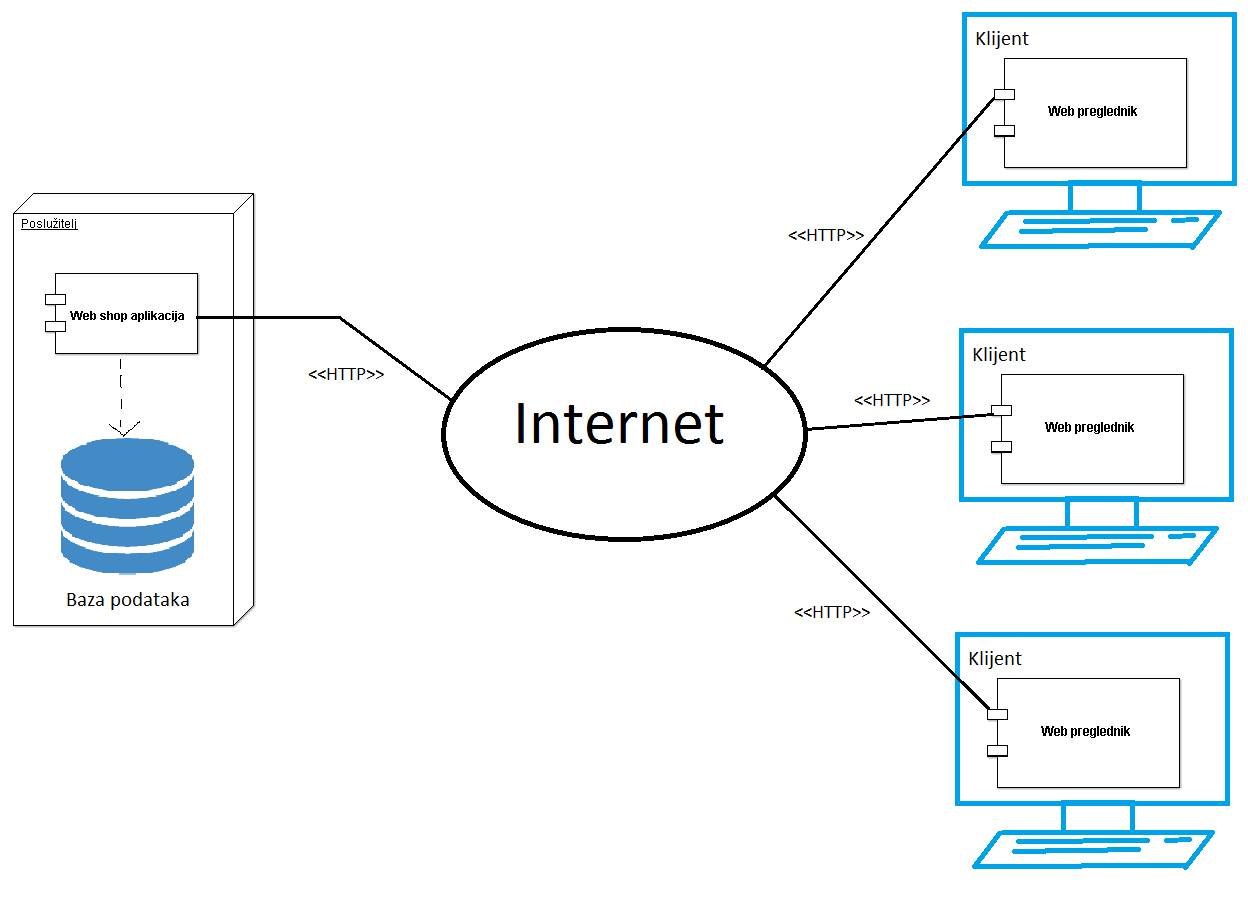
Frontend dio čini AngularJS i HTML stranice kreirane iz HTML predložaka



Slika : Prikaz dijagrama komponenti

# Implementacija i korisničko sučelje

## Dijagram razmještaja



Slika : Slika prikazuje razmještaj resursa potrebnih za funkcioniranje sustava

## Korištene tehnologije i alati

**Korištene tehnologije:**

* Java – objektno orijentirani programski jezik i okvir za izradu aplikacija opće namjene. Omogućuje izvršavanje jednom napisanog koda na različitim platformama zahvaljujući virtualnom stroju (JVM) koji izvršava kod.
* Spring – programski okvir otvorenog koda za brzu izradu java aplikacija
* Spring Boot - „convention over configuration“ programski okvir za brzi razvoj web aplikacija baziranih na Spring programskom okviru
* AngularJS – javascript programski okvir otvorenog koda za izradu web aplikacija
* Apache Tomcat - web aplikacijski server otvorenog koda
* H2 – java SQL baza podataka koja je vrlo mala po veličini i korištenju memorije te je idealna kao ugrađena (embedded) baza.
* Liquibase – biblioteka koja omogućava olakšano migriranje modela i podataka među različitim bazama
* OpenShift – platform as a service usluga (PAAS) koja omogućuje besplatno postavljanje i pokretanje web aplikacija u Internet cloudu

**Korišteni Alati:**

* Eclipse IDE – beplatan alat otvorenog koda za pisanje java aplikacija
* SublimeText – editor koda
* ArgoUML – alat otvorenog koda za UML modeliranje

## Upute za instalaciju

Aplikacija se izvodi unutar Java virtualne mašine (JVM) pa je aplikaciju moguće instalirati na bilo koji operacijski sustav koji ima podršku za JVM. Prije toga potrebno je instalirati Java radno okruženje (JRE) verzije 8.

Kako se radi o web aplikaciji, potreban je odgovarajući web aplikacijski server koji će pokretati aplikaciju i preko kojeg aplikacija komunicira sa vanjskom okolinom. Informacija o web serveru je ugrađena u samu aplikaciju te se kod prve inicijalizacije dohvaćaju izvršne datoteke web servera.

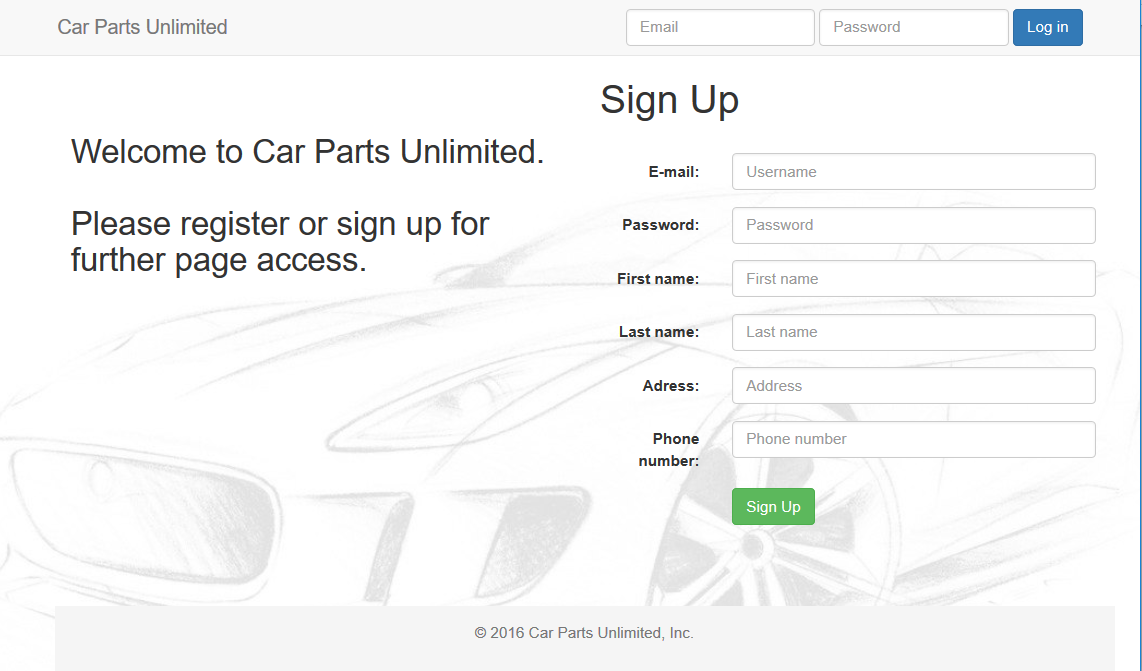
Aplikacijski web server koji se koristi je Apache Tomcat verzije 8.

Informacija o bazi ugrađena je u samu aplikaciju te se kod prve inicijalizacije dohvaćaju izvršne datoteke baze i raspakiravaju na disk računala te je baza spremna za daljnje korištenje.

Koristi se H2 baza verzije 1.4.

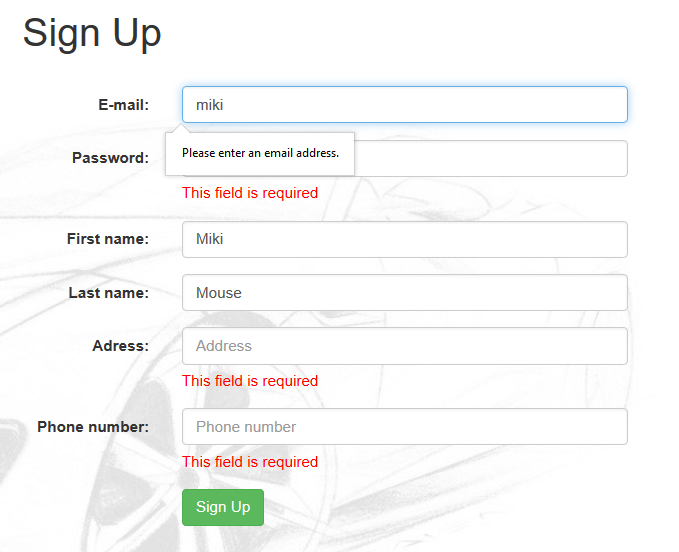
## Korisničke upute

Prilikom dolaska na stranicu korisnik je neprijavljen, anoniman, ne može pregledavati proizvode te se treba registrirati i prijaviti kako bi mogao započeti koristiti sustav (Slika 13).



Slika : Početni ekran web-aplikacije

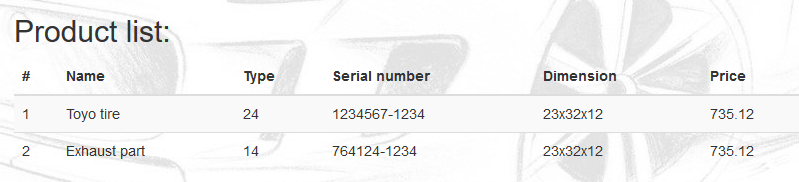
Kod registracije korisnik ispunjava formu za registraciju na kojoj postoje validacijska pravila te ukoliko korisnik unese ispravno formatirane podatke sustav ga registrira ili u suprotnom dobije poruku o neispravnom unosu. (Slika 14).



Slika : Izgled validacije forme za registraciju

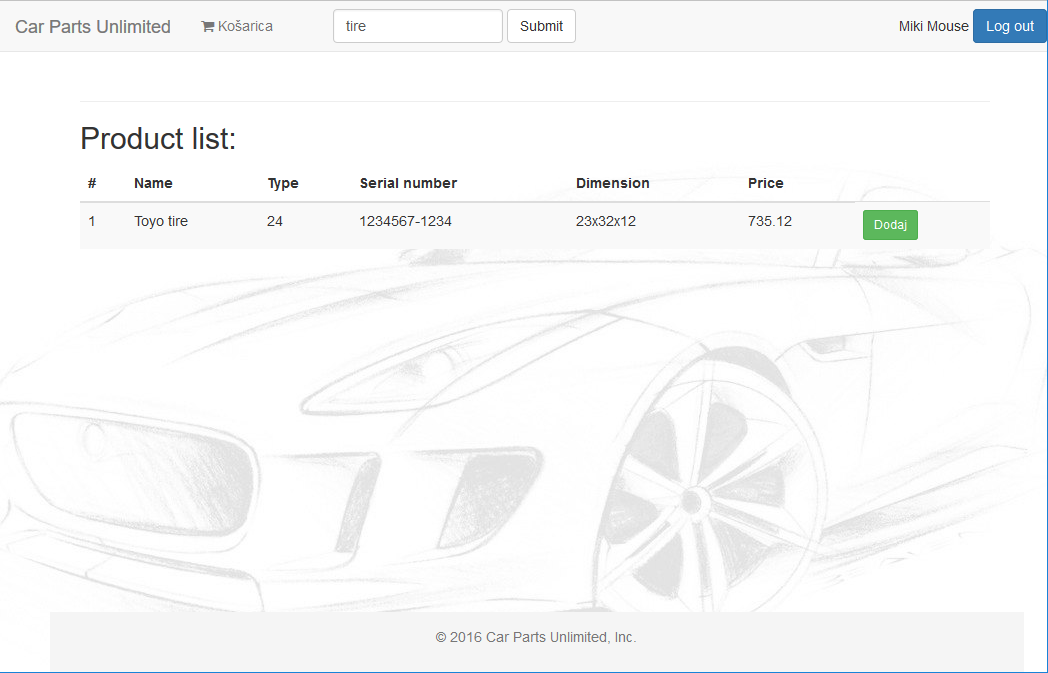
Nakon što se korisnik uspješno prijavi u web - shop prikazuje mu se stranica gdje može upisati u tražilicu naziv željenog proizvoda i košarica koja je vidljiva cijelo vrijeme u navigaciji.

Korisnik može odabrati detalje proizvoda tako da klikne na željeni proizvod (Slika 15). Na ekranu sa detaljima vidi kratki opis proizvoda, cijenu i dimenzije. Također vidi opciju dodavanja tog proizvoda u košaricu te opciju povratka na glavni izbornik.



Slika : Ispis proizvoda

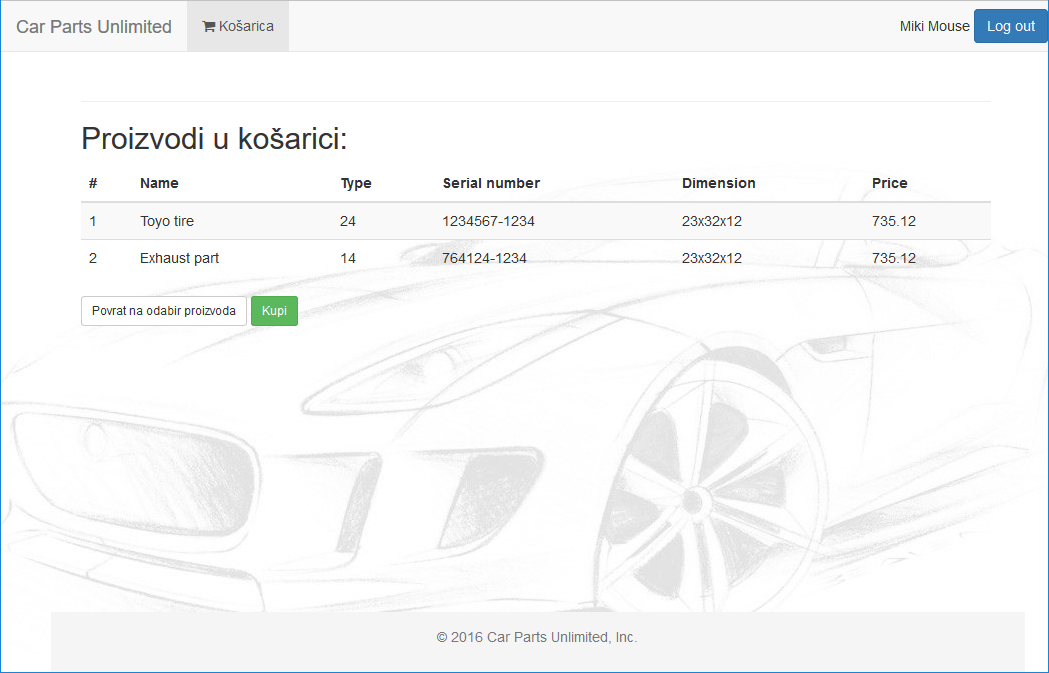
Nakon što je korisnik zadovoljan s odabranim proizvodom, može ga dodati u košaricu. (Slika 16).



Slika : Ispis traženog proizvoda pomoću tražilice

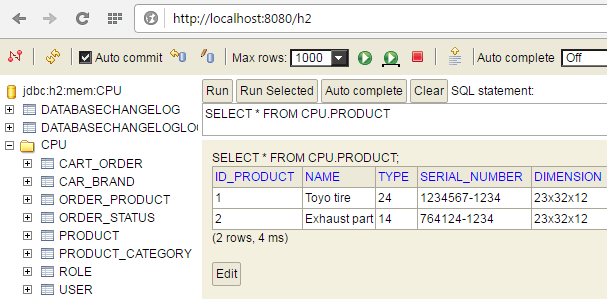
Nakon što korisnik želi završiti kupnju, odabire gumb "Košarica". Na ekranu „Košarica“, korisnik ima opciju „Kupi“ ili „Povratak na odabir proizvoda“. (Slika 17).

Klikom na gumb "Kupi" korisnika se preusmjerava na ekran sa zahvalom i potvrdom da je njegova kupovina uspješno provedena. Također resetira se košarica na inicijalno stanje. Dodatno se izračuna popust ovisno o tome kako dugo je korisnik vjeran kupac.



Slika : Pregled košarice i mogućnost završetka kupnje ili povratka na odabir proizvoda

Administrator ima mogućnost ažuriranja stanja proizvoda, dodavanja novih proizvoda i brisanja postojećih (Slika 18).



Slika : Web sučelje za pristup administraciji baze

# Reference

1. M. Horvat: Objektno orijentirani razvoj programa, TVZ, predavanja 2015./2016. <https://moj.tvz.hr/studijspecinf1/predmet/Objektno+orijentirani+razvoj+programa>
2. A. Jović, M. Horvat, I. Grudenić, „UML-dijagrami: Zbirka primjera i riješenih zadataka“, 2014.
3. ArgoUML<http://argouml.tigris.org/>