

**Rent a bike agencija**

Specifikacija zahteva softvera

(Projektni zadatak)

**PREDMET: SE201 Uvod u softversko inženjerstvo**

Profesor: Student:

*Prof. dr Dragan Domazet*   *Ana Sajić Broj indeksa: 2773*

Asistent:

*MSc Lazar Mrkela*

# **Sadržaj**

[**1.** **Uvod** 2](#_Toc503793898)

[1.1 Opis i svrha dokumenta 2](#_Toc503793899)

[1.2 Definicije, akronimi i skraćenice 3](#_Toc503793900)

[1.3. Reference 3](#_Toc503793901)

[**2.** **Opšti opis** 3](#_Toc503793902)

[2.1. Funkcije softvera 4](#_Toc503793903)

[2.2 Korisnici sistema 4](#_Toc503793904)

[2.3 Perspektiva proizvoda 4](#_Toc503793905)

[2.3.1. Interfejs sistema: 4](#_Toc503793906)

[2.3.2. Korisnički interfejs: 4](#_Toc503793907)

[2.3.3. Hardverski interfejs: 5](#_Toc503793908)

[2.3.4. Softverski interfejs: 5](#_Toc503793909)

[2.3.5. Komunikacioni interfejs: 5](#_Toc503793910)

[2.3.6. Memorijski zahtevi: 5](#_Toc503793911)

[2.4 Zahtevi korisnika 5](#_Toc503793912)

[2.5 Generalna ograničenja 5](#_Toc503793913)

[2.6 Pretpostavke i ograničenja 5](#_Toc503793914)

[**3.** **Specifični zahtevi** 7](#_Toc503793915)

[3.1. Funkcionalni zahtevi korisnika 7](#_Toc503793916)

[3.1.1.Registracija i prijava 7](#_Toc503793917)

[3.1.2.Pretraga bicikala 7](#_Toc503793918)

[3.1.3.Kontakt 7](#_Toc503793919)

[3.1.4. Odjava 7](#_Toc503793920)

[3.2 Funkcionalni zahtevi administratora 7](#_Toc503793921)

[3.2.1. Prijava na sistem 8](#_Toc503793922)

[3.2.2. Unos bicikala 8](#_Toc503793923)

[3.2.3. Uklanjanje bicikala 8](#_Toc503793924)

[3.2.4. Pregled bickala u bazi 8](#_Toc503793925)

[3.2.5. Pregled korisnika u bazi 8](#_Toc503793926)

[3.2.6. Unos kategorija 8](#_Toc503793927)

[3.2.7. Odjava 8](#_Toc503793928)

[3.3 Nefunkcionalni zahtevi 9](#_Toc503793929)

[3.3.1. Prilagodljiv dizajn 9](#_Toc503793930)

[3.3.2. Informacije o agenciji 9](#_Toc503793931)

[3.3.3. Rad u realnom vremenu 9](#_Toc503793932)

[3.4 Slučajevi korišćenja 10](#_Toc503793933)

[3.4.1. Prijava i registracija 10](#_Toc503793934)

[3.4.2. Pregled bicikala 10](#_Toc503793935)

[3.4.3. Kontakt sa agencijom za rent a bike 10](#_Toc503793936)

[3.4.3. Prijava 11](#_Toc503793937)

[3.4.4. Dodavanje novih bicikala 12](#_Toc503793938)

[3.4.5. Izmena i brisanje bickala 12](#_Toc503793939)

[3.4.6. Unos kategorija 12](#_Toc503793940)

[3.5. Apstraktni dijagram sekvenci (sistem sekvencijalni dijagram) 13](#_Toc503793941)

[3.5.1.Dijagram sekvenci za registrovanje korisnika 13](#_Toc503793942)

[3.5.2. Dijagram sekvenci za prijavu na sistem – korisnik 14](#_Toc503793943)

[3.5.3. Pregled bicikala – korisnik 15](#_Toc503793944)

[3.5.4. Odjavljivanje sa sistema – korisnik 15](#_Toc503793945)

[3.5.5. Dijagram sekvenci za prijavu na sistem – Administrator 16](#_Toc503793946)

[3.5.6. Dijagram sekvenci za unos bicikala u bazu – Administrator 16](#_Toc503793947)

[3.5.7. Dijagram sekvenci za pregled svih bicikala – Administrator 17](#_Toc503793948)

[3.5.8. Dijagram sekvenci za pregled svih korisnika u sistemu – administrator 17](#_Toc503793949)

[3.5.9. Dijagram sekvenci za unos kategorija u bazu – administrator 18](#_Toc503793950)

[3.5.10. Odjava 18](#_Toc503793951)

[3.6. Klasni dijagrami 19](#_Toc503793952)

[3.7. Baza podataka 23](#_Toc503793953)

[**4.** **Arhitektura sistema** 24](#_Toc503793954)

[**4.** **Testiranje** 26](#_Toc503793955)

[**5.** **Prikaz korisničkog interfejsa** 32](#_Toc503793956)

[**6.** **Planovi za unapređenje aplikacije** 35](#_Toc503793957)

[**7.** **Zaključak** 36](#_Toc503793958)

# **Uvod**

Tema projektnog zadatka za ovaj predmet je Rent a bike agencija. Sistem za rent a bike razvijen kao veb aplikacija korišćenjem Angular 2 tehnologije. Za izradu dijagrama je korišćen PowerDesigner. Kroz ovaj projekat će biti dat prikaz svih funkcionalnih i nefukcionalnih zahteva koje sistem mora da zadovolji. Pored toga, biće prikazana i šema baze podataka, kao i najvažniji dijagrami kao što su sekvencijalni i klasni dijagramm, infrastrukturni i aplikacijski.

## 1.1 Opis i svrha dokumenta

Svrha dokumenta jeste prikaz i detaljan opis sistema koji će biti razvijen. U dokumentu je potrebno obuhvatiti sve zahteve koje sistem treba da zadovolji, zatim funkcionalnosti sistema, korisnike sistema, i tehnologiju koja će se koristiti za kasniju izradu sistema.

## 1.2 Definicije, akronimi i skraćenice

* Angular 2 - Angular 2 je MVC framework napravljen od strane Google-a , koji se koristi za kreiranje front-end aplikacija
* Selenium - Selenium je skup softverskih alata čija je namena da podrži i obezbedi automatizovano testiranje web aplikacija.

## 1.3. Reference

1. PacktPub Limited, Managing Software Development with Trac and Subversion, 2007. god.
2. Perišić Branko, Osnovi softverskog inzenjerstva, 2011. god.
3. Selenium, dostupno na : <http://www.seleniumhq.org/>
4. Mr Zoran Jeremić, Uvod u softevrsko inženjerstvo, VISER, 2011. god
5. UML dijagrami, dostupno na: <https://www.uml-diagrams.org/>

# **Opšti opis**

Tema projekta je kreiranje aplikacije za iznajmljivanje bicikala. Cilj aplikacije je da se ljudima koji žele da se voze a pri tom nemaju svoj bicikl, omogući iznjamljivanje na teritoriji grada Beograda. Kreiranjem ove web aplikacije potrebno je realizovati i obezbediti sve funkcionalnosti koje su potrebne zaposlenima u agenciji za rent a bike kao i potencijalnim klijentima agencije.

U zavisnosti od tipova funkcionalnosti koje treba da se obezbede putem ove aplikacije možemo razlikovati dve grupe korisnika. Prvu grupu bi predstavljali klijenti koji imaju potrebe za uslugama agencije rent a bike, dok administrator ima zadatak stavljanja novih informacija vezanih za iznajmljivanje, i vodi računa o tome da se svaka nova mušterija se upisuje u bazu podataka da bi se evidentirali iznajmljeni bicikli.

## 2.1. Funkcije softvera

Funkcije sistema su:

* Autentifikacija i autorizacija
* Sinhronizacija podataka sa bazom podataka
* Unos bicikala u bazu
* Unos korisnika u bazu
* Pregled svih bicikala koji se nalaze u bazi
* Pregleda svih korisnika u bazi

## 2.2 Korisnici sistema

Postoje dve vrste korisnika ovog softverskog sistema. To su:

* Korisnik(klijent)
* Administartor

Prvu grupu bi predstavljali klijenti koji imaju potrebe za uslugama agencije rent a bike, dok administrator ima zadatak stavljanja novih informacija vezanih za iznajmljivanje, i vodi računa o tome da se svaka nova mušterija se upisuje u bazu podataka da bi se evidentirali iznajmljeni bicikli.

## 2.3 Perspektiva proizvoda

### 2.3.1. Interfejs sistema:

Interfejs sistema je prilično intuitivan. Korisnik se može registrovati i prijaviti kroz formu koja je jednostavna. A pregled dostupnih bickala se može vršiti korišćenjem nekoliko filtera koji olakšavaju pretragu. Administratoru je za dodavanje i uklanjanje bicikala kreirana forma koja se na lak način popunjava.

### 2.3.2. Korisnički interfejs:

Korisniku su za pregled informacija omogućeni filteri, a slike bicikala su dodvoljno krupne i vidljive.

### 2.3.3. Hardverski interfejs:

Što se tiče hardverskog dela, sistem je relativno jednostavan i nije zahtevan, pa ne zahteva samim tim ni jaku hardversku podršku.

### 2.3.4. Softverski interfejs:

Za izradu je korišćen programski jezik PHP korišćenjem Angular 2 tehnologije(okvira).

### 2.3.5. Komunikacioni interfejs:

Korisnici pristupaju pomoću LAN ili Wireless mreže mrežnog interfejsa.

### 2.3.6. Memorijski zahtevi:

S obzirom na to da je aplikacija jednostavna i nije zahtevna, nisu joj potrebni memorijski resursi u većoj količini.

## 2.4 Zahtevi korisnika

Aplikacija je prilično jednostavna i interfejs je intuitivan, tako da je za korisnike dovoljno da poznaju osnovno znanje rada na računaru.

# 2.5 Generalna ograničenja

Sva polja u formi moraju biti popunjena u određenom formatu

Da bi se pristupilo aplikaciji, korisnik mora imati pristup internetu

Ukoliko nešto u formi nije dobro popunjeno, sistem mora dati povratnu poruku da je došlo do greške

## 2.6 Pretpostavke i ograničenja

S obzirom da korisnik koji koristi sistem, zna o kakvom sistemu se radi, za sve promene koje se dese u sistemu i načinu rada sistema, potrebno je obavestiti korisnike.

# **Specifični zahtevi**

U ovom delu ćemo opisati funkcionalne i nefunckionalne zahteve, kao i slučajeve korišćenja tj. interakciju između korisnika i sistema.

## 3.1. Funkcionalni zahtevi korisnika

### 3.1.1.Registracija i prijava

Klijenti bi pomoću aplikacije mogli da pronađu ogovarajući bicikl i evenutalno ga iznajme. Posetilac sajta može da bude bilo koji korisnik koji želi da iznajmi bicikl, a ukoliko se odluči može se i registrovati na sajt i primati obaveštenja vezana za agenciju.

### 3.1.2.Pretraga bicikala

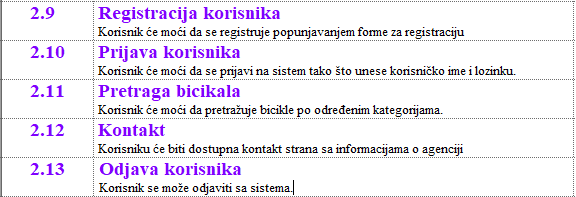
Klijent bi mogao da vrši pretragu tako što će bi birao bicikle po određenim kategorijama, kao što su proizvođač i vrsta kategorije(odrasli ili deca).

### 3.1.3.Kontakt

Klijentu će biti dostupna kontakt strana na kojoj može videti brojeve telefona, email i adresu agencije ukoliko želi da stupi u kontakt sa agencijom. Takođe će se na istoj strani pružiti mogućnost da preko email forme direktno sa sajta pošalje email agenciji.

### 3.1.4. Odjava

Korisnik ima mogućnost da se na kraju posete sajtu odjavi sa sistema



Slika 3.1.1. – Funkcionalni zahtevi korisnika

## Funkcionalni zahtevi administratora

### 3.2.1. Prijava na sistem

Administrator ima mogućnost da se prijavi na sisem

### 3.2.2. Unos bicikala

Pomoću fome administrator ima opciju unosa bicikala u bazu podataka

### 3.2.3. Uklanjanje bicikala

Administrator može ukloniti bicikle i ostale informacije ukoliko više nisu dostupni

### 3.2.4. Pregled bickala u bazi

Administrator ima mogućnost pregleda svih informacija za bicikle u bazi podataka

### 3.2.5. Pregled korisnika u bazi

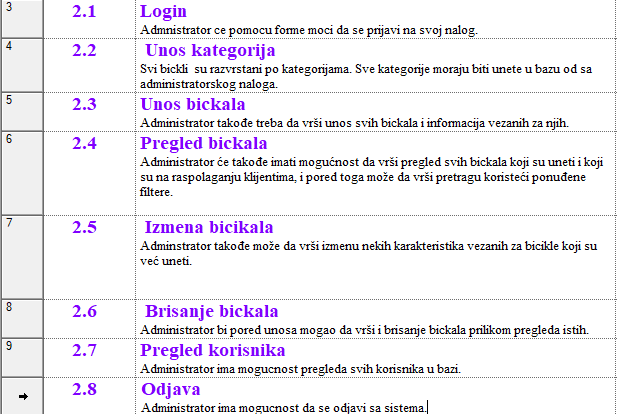
Administrator ima mogućnost pregleda svih korisnika u bazi podataka

### 3.2.6. Unos kategorija

Svi bicikli su razvrstani po kategorijama. Sve kategorije moraju biti unete u bazu od strane administratorskog naloga. Pod kategorijom se proizvođač, naziv i sl.

### 3.2.7. Odjava

Administrator ima mogućnost odjave sa sistema



Slika 3.2.1. – Funkcionalni zahtevi za administratora

## Nefunkcionalni zahtevi

### 3.3.1. Prilagodljiv dizajn

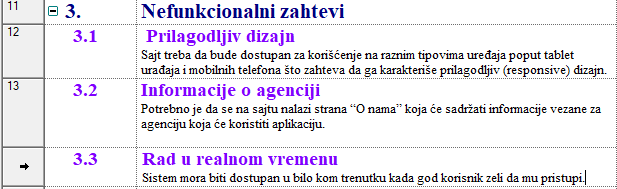
Sajt treba da bude dostupan za korišćenje na raznim tipovima uređaja poput tablet urađaja i mobilnih telefona što zahteva da ga karakteriše prilagodljiv (responsive) dizajn.

### 3.3.2. Informacije o agenciji

Potrebno je da se na sajtu nalazi strana “O nama” koja će sadržati informacije vezane za agenciju rent a bike.

### 3.3.3. Rad u realnom vremenu

Sam softver mora biti dostupan u svakom trenutku kada mu korisnik pristupi.



Slika 3.3.1. – Nefunkcionalni zahtevi

## Slučajevi korišćenja

1. Korisnik

### 3.4.1. Prijava i registracija

Preduslov:Da bi korisnik mogao da prima obaveštenja vezana za agneciju mora da se registruje

Način korišćenja**:** Kada pristupi sajtu onda u login formu unosi korisničko ime i šifru.

Greške:Greške koje korisnik može da napravi prilikom prijave su unos pogrešnog korisničkog imena ili lozinke nakon čega će biti obavešten porukom od strane sistema.

Rezultat:Očekivatni rezultat slučaja korišćenja je da korisniku pristupi svom nalogu

### 3.4.2. Pregled bicikala

Preduslov**:** Klijent mora da bude na web sajtu za rent a bike kako bi mogao da pregleda i izabere bicikl.

Način korišćenja**:** Kada je klijent na web strani aplikacije, može da vidi sve moguće bickle koje su trenutno dostupne. O svakoj bicikli se nalaze neophodne informacije.

Greške:Prilikom pregleda biciklala klijent samo pasivno sakuplja informacije, tako da ne može napraviti grešku. Greška sistema za ovu funkcionalnost može biti to da klijent može videti neki bickl koji trenutno nije dostupan. Kako bi se greška izbegla potrebno je redovno ažurirati nove bicikle i označiti one koji trenutno nisu dostupni.

Rezultat: Očekivani rezultat slučaja korišćenja je da samim pristupom sajtu korisnik može da pregleda sve moguće bicikle.

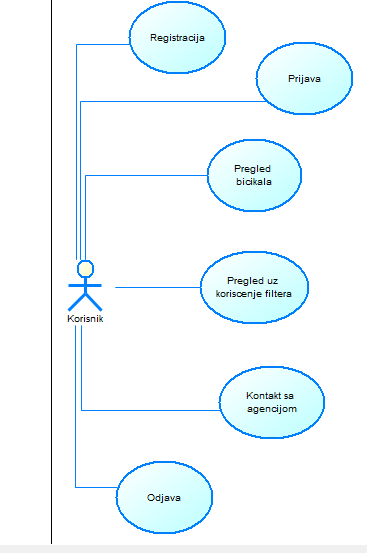
### 3.4.3. Kontakt sa agencijom za rent a bike

Preduslov:Klijent takođe mora biti na odgovarajućoj web strani gde iz početnog menija može odabrati stranu kontakt i na njoj videti sve kontakte sa agencijom kao i formu za direktno slanje emaila agenciji sa sajta.

Način korišćenja: Klijent mora popuniti email formu tako što će uneti validnu email adresu.

Greške:Kada su u pitanju greške u ovom slučaju korišćenja, klijent može uneti ne validnu email adresu nakon ćega će takođe biti obavešten da je napravio grešku.

Rezultat:Očekivani rezultat je da klijent uspešno kontaktira agenciju za rent a bike putem slanja email-a preko sajta.



Slika 3.4.1. – Use case dijagram - korisnik

1. Administrator

### 3.4.3. Prijava

Preduslov:Da bi administrator mogao da obavlja svoje obaveze potrebno je da se uloguje na web sajtu.

Način korišćenja**:** Kada pristupi sajtu onda u login formu unosi korisničko ime i šifru.

Greške:Greške koje administrator može da napravi prilikom prijave su unos pogrešnog korisničkog imena ili lozinke nakon čega će biti obavešten porukom od strane sistema.

Rezultat:Očekivatni rezultat slučaja korišćenja je da administrator pristupi svom nalogu nakon čega će obavljati funkcionalnosti za koje je zadužen.

### 3.4.4. Dodavanje novih bicikala

Preduslov:Da bi administrator mogao da doda bicikl neophodno je da se prijavi i pristupi administratorskom nalogu, nakon čega će moći da doda novo vozilo.

Način korišćenja: Prilikom unosa novog bickla, administrator popunjava polje za naziv i model bickla i ostale bitne informacije vezane za isti.

Greške: Greška koja se može desiti je da administrator unese bicikl koji već postoji u bazi što će prouzrokovati ispisivanje poruke za grešku.

Rezultat: Rezultat slučaja korišćenja je da se u bazu uspešno doda novi bicikl koji klijenti mogu videti kada pristupe sajtu.

### 3.4.5. Izmena i brisanje bickala

Preduslov: Da bi administrator mogao da obavi izmenu ili brisnje nekog bicikla neophodno je da bude ulogovan na svoj nalog, odabere određeni bicikl, i klikom na isti, otvori novu stranu izmeni informacije ili ga obriše.

Način korišćenja: Kada otvori stranu sa informacijama o biciklu administrator može izvršiti izmenu bilo kog polja, tj. promeniti sadržaj u formi i ponovo sačuvati isti sa promenjenim podacima ili klikom na dugme za brisanje ukloniti ga iz baze podataka.

Greške: Greške koje mogu nastati prilikom izmene su da administrator obriše vrednost iz nekog od obaveznih polja nakon čega se izmene neće primeniti i ispisaće se poruka za grešku.

Rezultat: Rezultat ova dva slučaja korišćenja je da se ažuriraju informacije o biciklima kao i da se obrišu isti ukoliko je to neophodno iz nekog razloga.

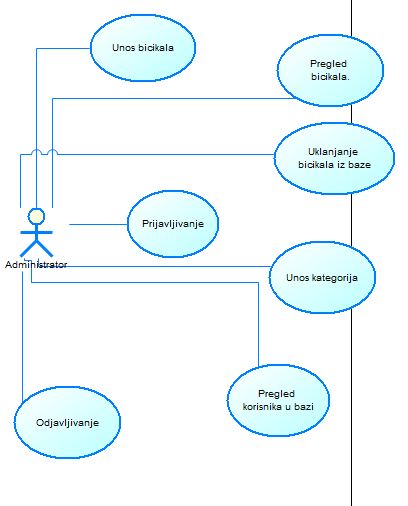
### 3.4.6. Unos kategorija

Preduslov:Da bi administrator mogao da doda kategoriju neophodno je da se prijavi i pristupi administratorskom nalogu, nakon čega će moći da doda novu kategoriju

Način korišćenja: Prilikom unosa nove kategorije, administrator popunjava polje za naziv kategorije

Greške: Greška koja se može desiti je da administrator unese kategoriju koji već postoji u bazi što će prouzrokovati ispisivanje poruke za grešku.

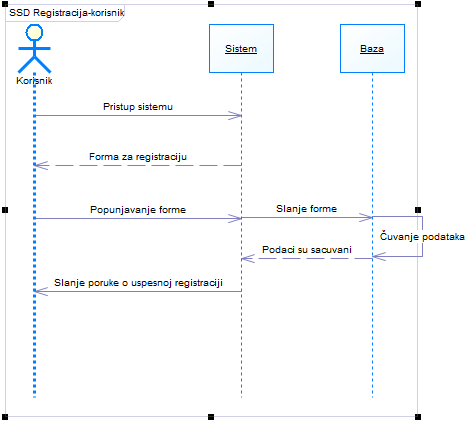
Rezultat: Rezultat slučaja korišćenja je da se u bazu uspešno doda novu kategoriju koji klijenti mogu videti kada pristupe sajtu.



Slika 3.4.2. – Use case dijagram - administrator

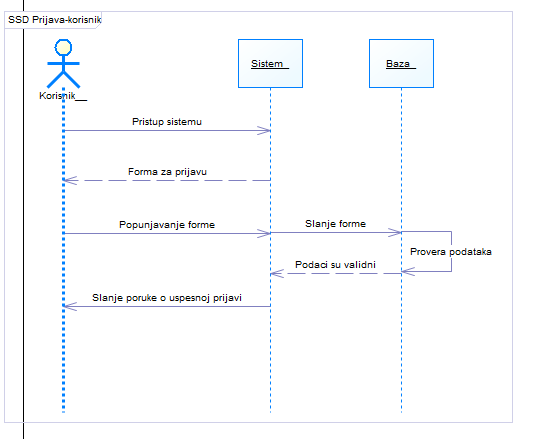
## Apstraktni dijagram sekvenci (sistem sekvencijalni dijagram)

### 3.5.1.Dijagram sekvenci za registrovanje korisnika



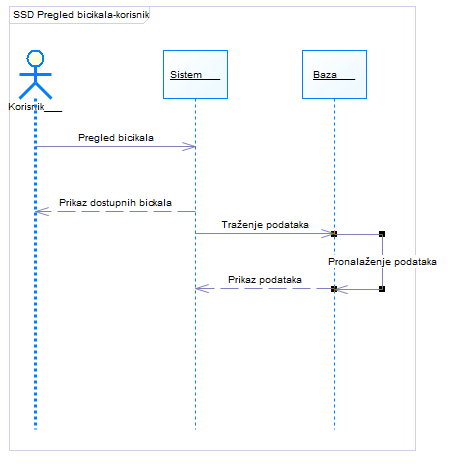
Slika 3.5.1.– Dijagram sekvenci za registrovanje korisnika

### Dijagram sekvenci za prijavu na sistem – korisnik



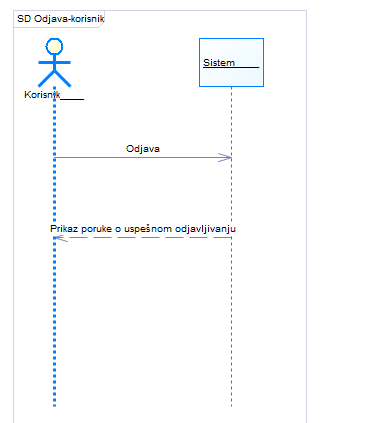
Slika 3.5.2. - Dijagram sekvenci za prijavu na sistem – korisnik

### Pregled bicikala – korisnik



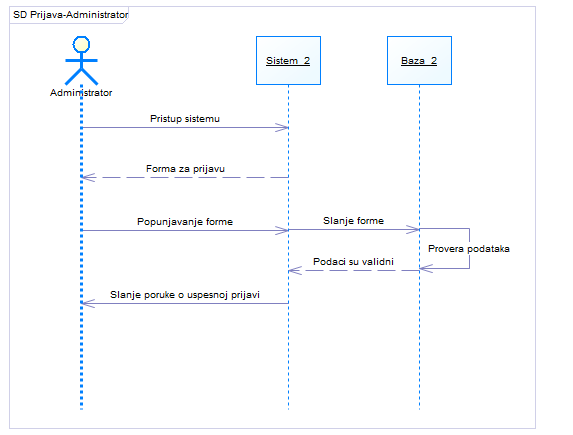
Slika 3.5.3.- Dijagram sekvenci za pregled dostupnih bicikala – korisnik

### Odjavljivanje sa sistema – korisnik



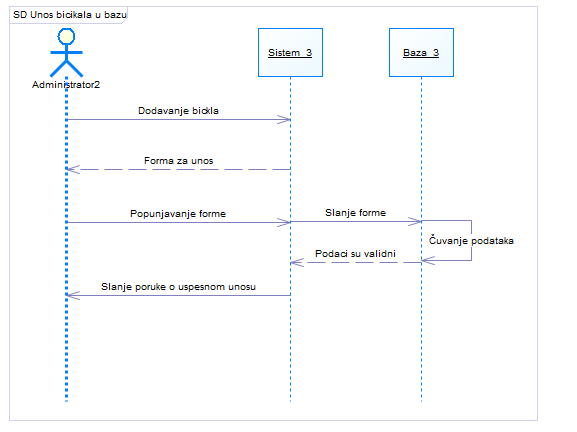
Slika 3.5.4. - Dijagram sekvenci za odjavljivanje – korisnik

### Dijagram sekvenci za prijavu na sistem – Administrator



Slika 3.5.5. - Dijagram sekvenci za prijavu na sistem – administrator

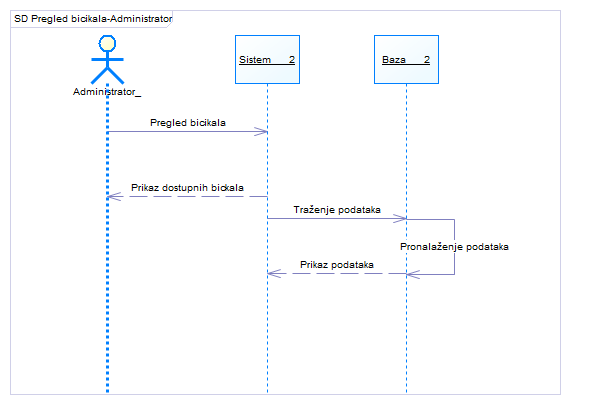
### Dijagram sekvenci za unos bicikala u bazu – Administrator



Slika 3.5.6. - Dijagram sekvenci za unos bicikala – administrator

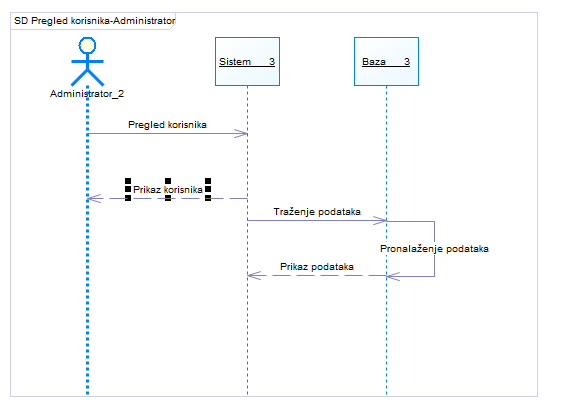
### 

### Dijagram sekvenci za pregled svih bicikala – Administrator



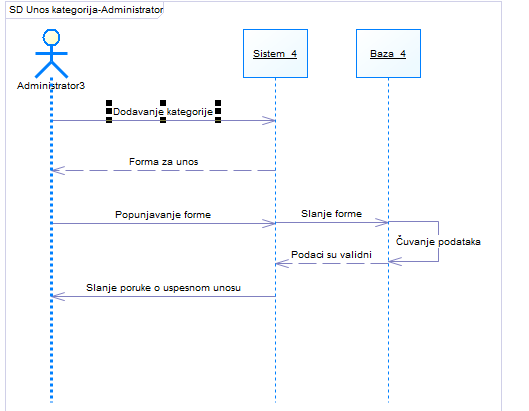
Slika 3.5.7. - Dijagram sekvenci za pregled bicikala – administrator

### Dijagram sekvenci za pregled svih korisnika u sistemu – administrator



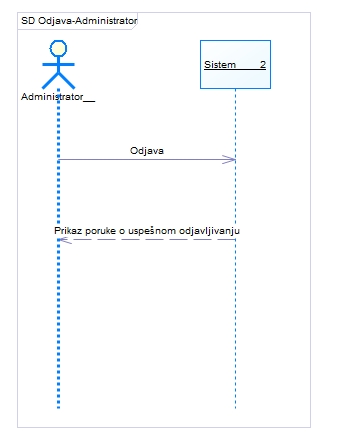
Slika 3.5.8. - Dijagram sekvenci za pregled korisnika – administrator

### Dijagram sekvenci za unos kategorija u bazu – administrator

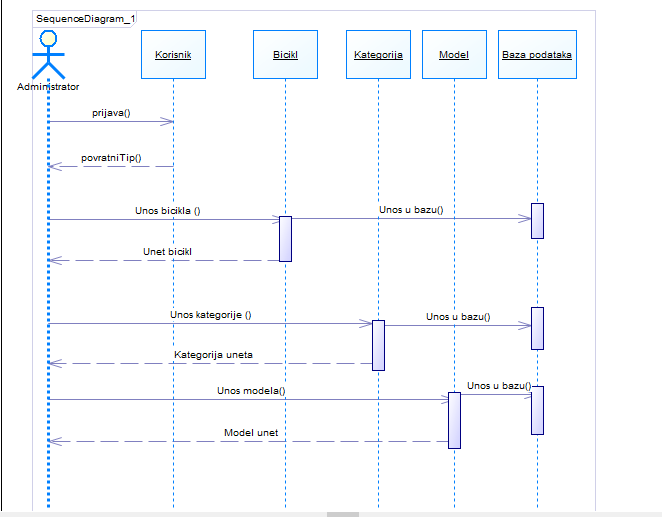


Slika 3.5.9. - Dijagram sekvenci za unos kategorija u bazu – administrator

### Odjava

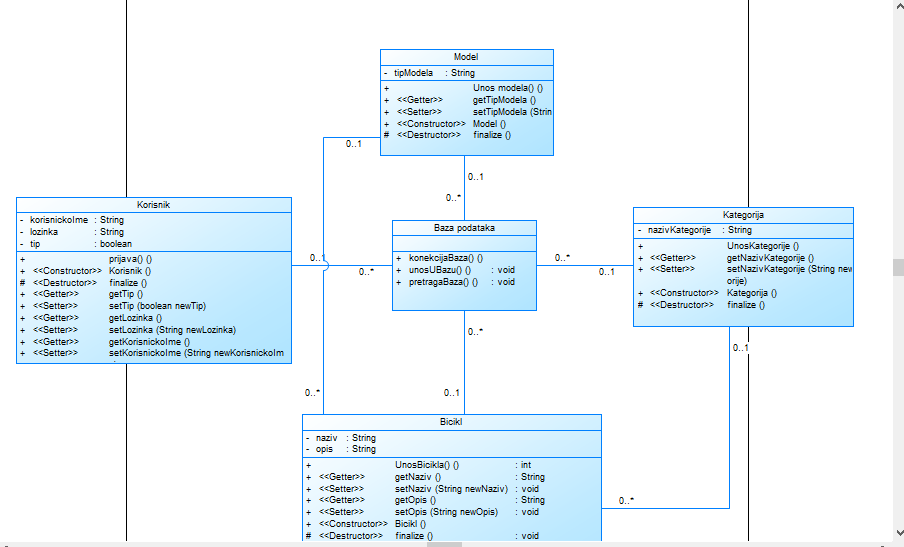


Slika 3.5.10.- Dijagram sekvenci za odjavu – administrator



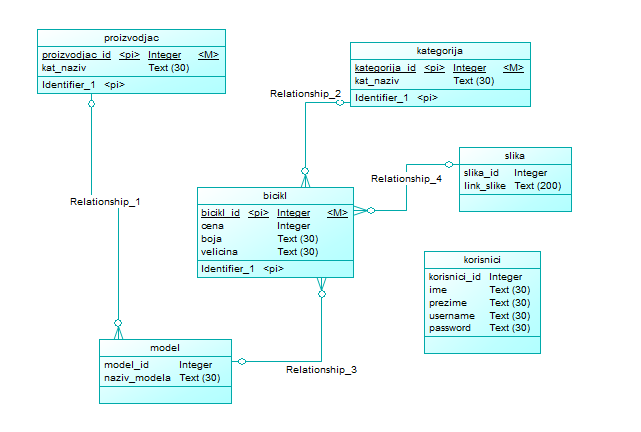
Slika 3.5.11. – SD dijagram(najviši nivo detaljnosti)

## Klasni dijagram

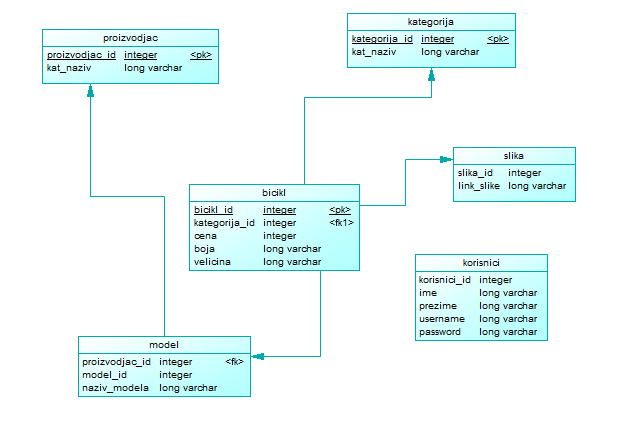


Slika 3.6.1. – Klasni dijagram

## Baza podataka



Slika 3.7.1. – Konceptualni model baze podataka



Slika 3.7.2. – Fizički model baze podataka

# **Arhitektura sistema**

MVC arhitektura – S obzirom na to da je aplikacija rađena i kreirana da bude web aplikacija, korišćenjem Angular 2 tehnologije, odlučujem se za MVC arhitekturu.

MVC deli sve ono što jedna aplikacija radi na tri dela. Ovakvo gledanje na funkcionalnost - takozvano razdvajanje zaduženja, smatra se i inače dobrom praksom.

***Model*** - sadrži podatke u obliku pogodnom za konkretnu primenu. Takođe sadrži i logiku aplikacije, definišući šta sve možemo da uradimo sa datim podacima. Mnoge aplikacije koriste mehanizme za trajno čuvanje podataka (bazu podataka, na primer). MVC ne definiše pristup podacima jer se smatra da je to “ispod” aplikacije ili enkapsulirano samim modelom.

***View***- prikazuje podatke iz modela u formatu pogodnom za interakciju, najčešće kao komponentu korisničkog interfejsa. U okviru jedne aplikacije može postojati više view-ova prilagođenih različitim situacijama koji prikazuju podatke iz istog modela.

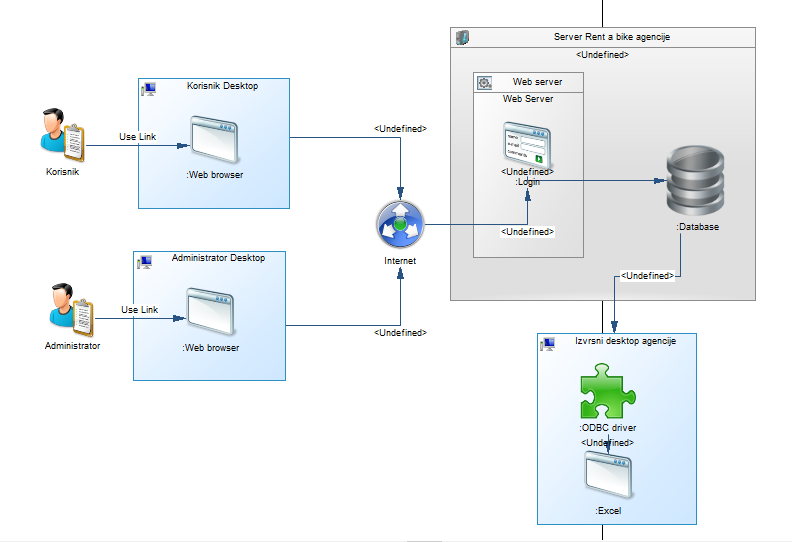
***Controller***- koordiniše modele i view-ove, uglavnom na osnovu korisnikovog unosa. Kada se desi neki događaj, na primer klik na neko dugme, obaveštava model o tome.

Prednosti:

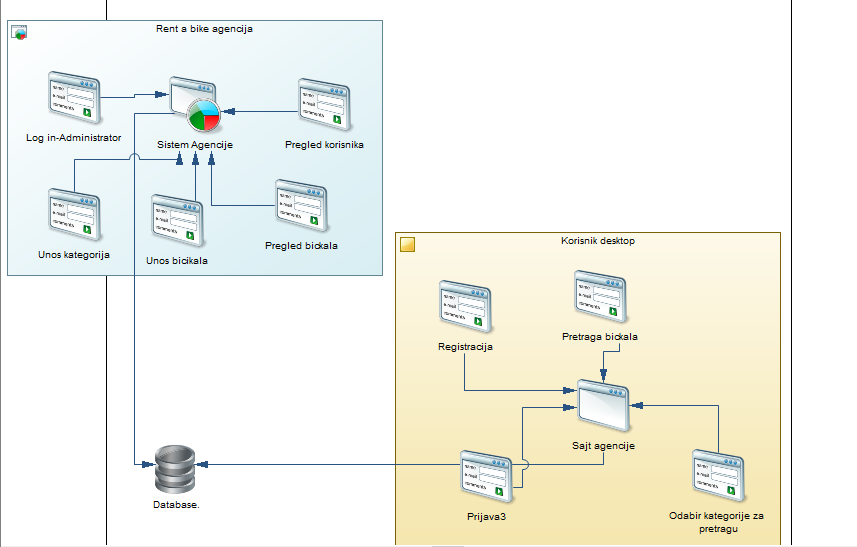
* omogućava da se model može prikazati na više načina
* lakše je dodati novi View (npr. novu web stranicu koja prikazuje postojeće podatke ili deo njih)
* omogućava da se interakcija sa korisnikom lako promeni
* omogućava da više programera rade paralelno na različitim slojevima (posebno važno u timovima specijalizovanim za pojedine slojeve npr. tim zadužen za implementaciju i održavanje korisničkog interfejsa, poslovne logike itd…)

Mane:

Ponekad se može desiti da usled čestih promena modela view bude preplavljen zahtevima za izmenu. Ukoliko on služi za prikazivanje sadržaja kojem je potrebno određeno vreme za renderovanje, česti zahtevi za izmenu mogu dovesti do kašnjenja.



Slika 4.1. – Infrastrukturni dijagram



Slika 4.2. – Aplikacijski dijagram

# **Testiranje**

Za testiranje moje aplikacije, odlučujem se za testiranje korišćenjem Seleniuma.

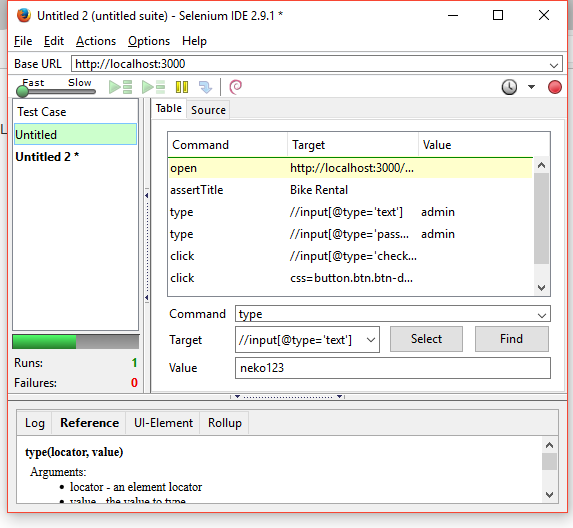
Selenium je alat za automatizovano testiranje korisničnog interfejsa Web aplikacija. Zahvaljujući svojoj univerzalnosti i tome što podržava testiranje u gotovo svim dostupnim pretraživačima, Selenium je trenutno najčešći izbor stručnjaka koji se bave testiranjem korisničkog interfejsa Web aplikacija.

Testiranje softvera je proces koji je usmeren ka procenjivanju karakteristika i sposobnosti softvera ili sistema i utvrđivanju da li taj softver ili sistem ispunjava određene uslove. Ovaj proces uključuje izvršavanje programa ili aplikacije sa namerom pronalaženja softverskih grešaka i svakog vida odstupanja sistema od originalnih zahteva korisnika, u toku izvršavanja programa.

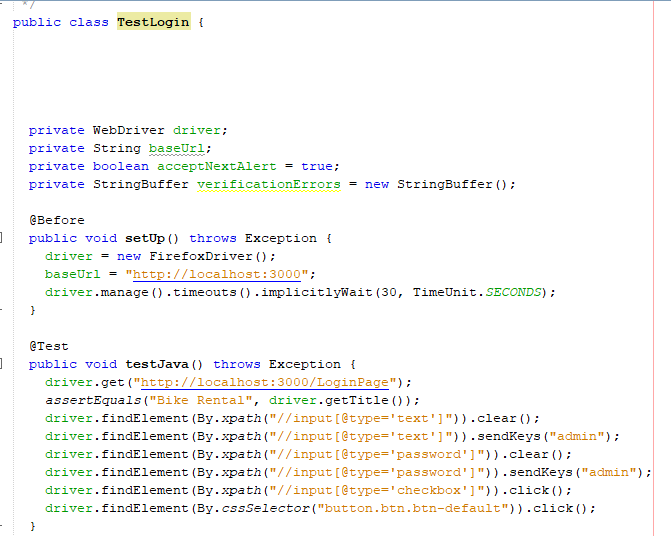
S obzirom na to da je testiranje celokupnog sistema prilično zahtevan posao, odlučujem se da testiram dve najvažnije funkcionalnosti u mojoj aplikaciji, a to su:

Prijava na sistem i Unos bicikala u bazu.

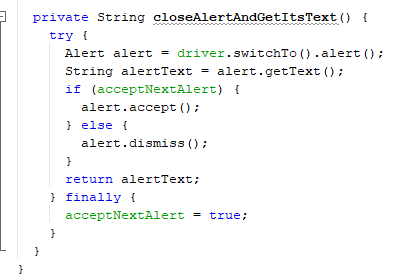
Prvi test se odnosi na prijavu. Najpre, otvaramo Selenium IDE, unosimo adresu kojoj ćemo pristupiti i pokrećemo snimanje. Zatim odlazimo na web adresu, unosimo podatke za prijavu(korisničko ime i lozinka) i nakon toga zaustavljamo snimanje. Pokrećemo test, i dobijamo rezultat da je ishod testa uspešan. Sledi prikaz ekrana i kod u java programskom jeziku.



Slika.4.1. – Prikaz uspešnosti prvog testa(Login)

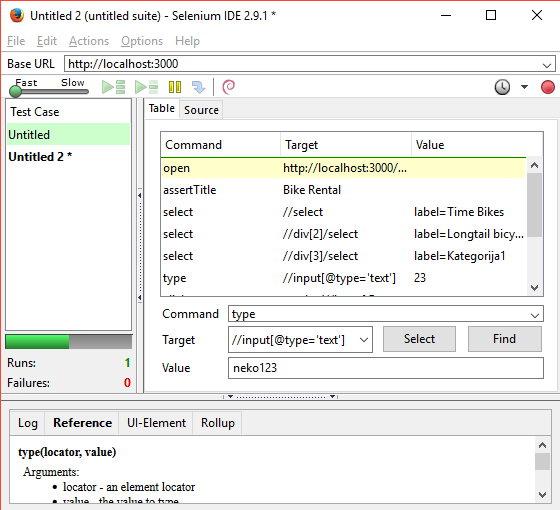






Slika 4.2. – Test za prijavu - kod

Isti postupak je i za unos bickala, s tim što se sada testira pravilan unos u polja za unos bickala. Sledi provera uspešnosti unosa bickla sa svim informacijama.

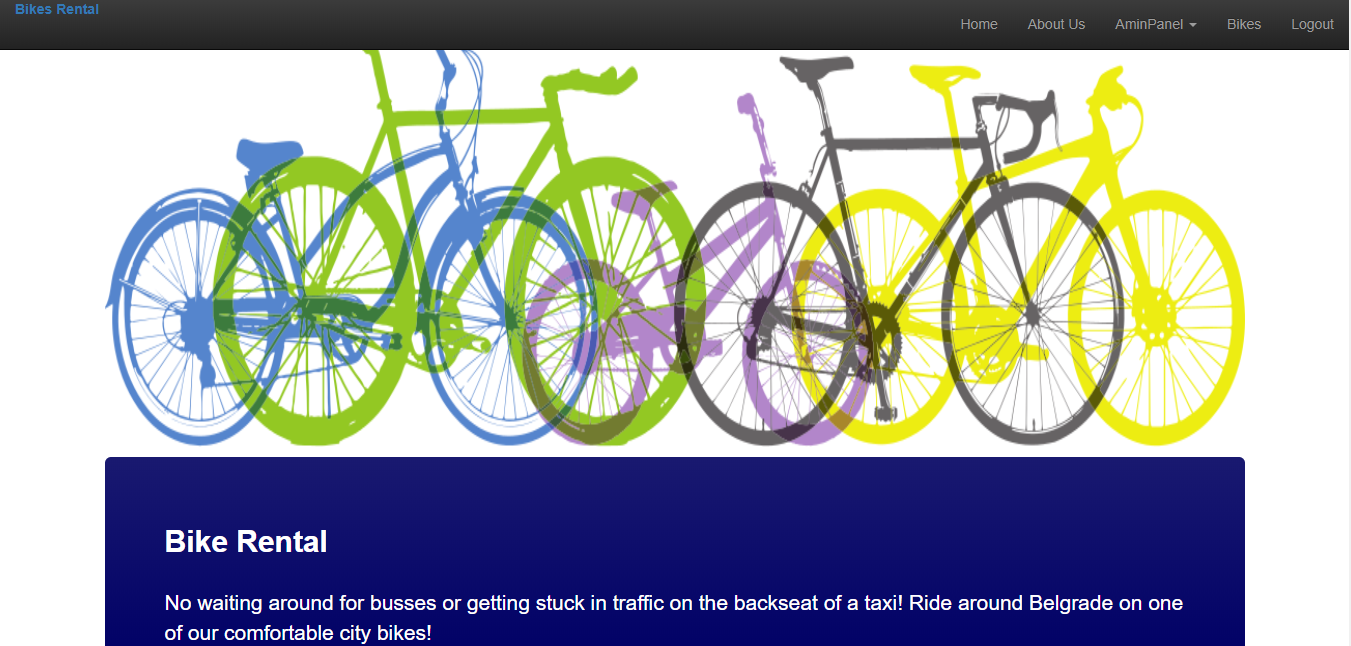


Slika 4.3. – Uspešan test za unos

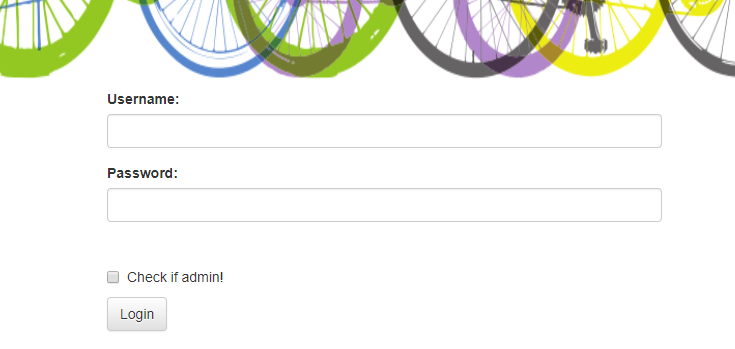


Slika 4.4. – Test za unos - kod

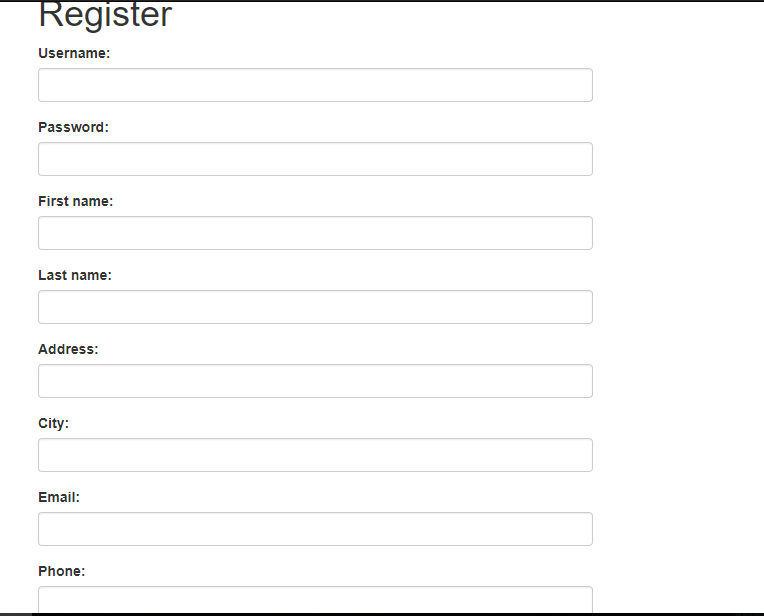
# **Prikaz korisničkog interfejsa**



Slika 5.1. – Početna strana



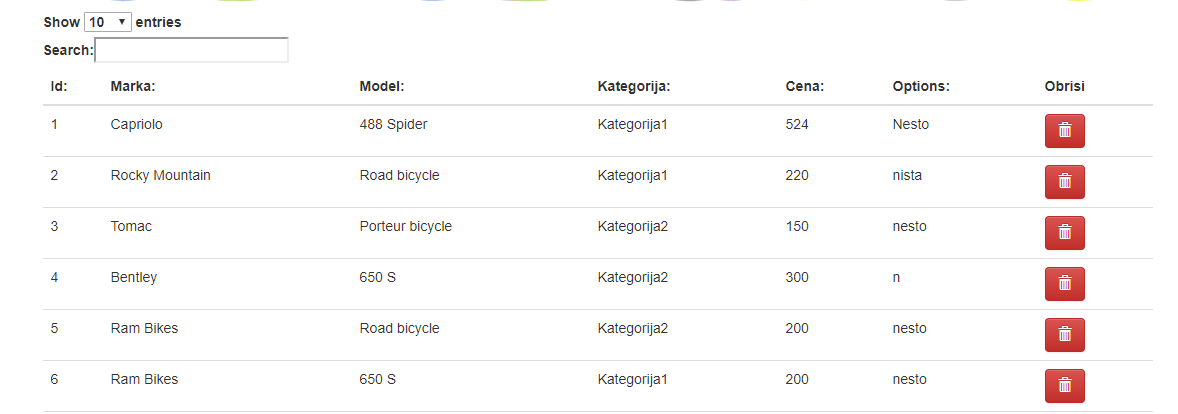
Slika 5.2. – Login strana



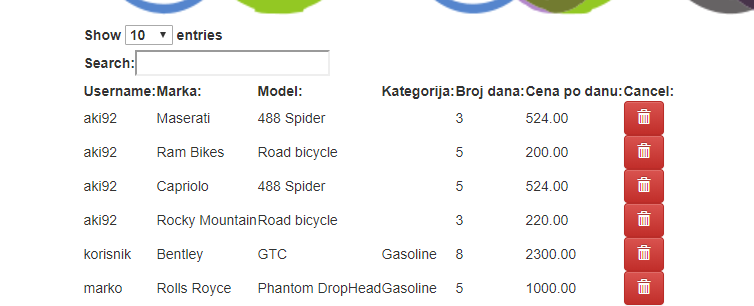
Slika 5.3. – Strana za registraciju



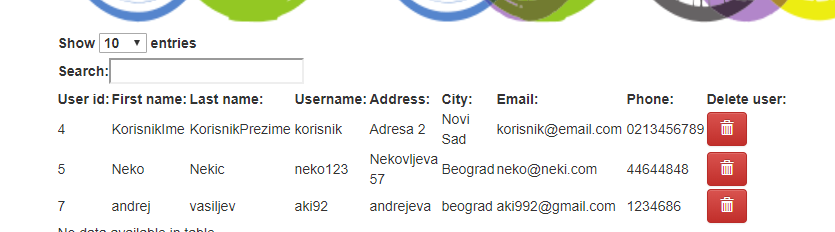
Slika 5.4. – Forma za dodavanje bickala



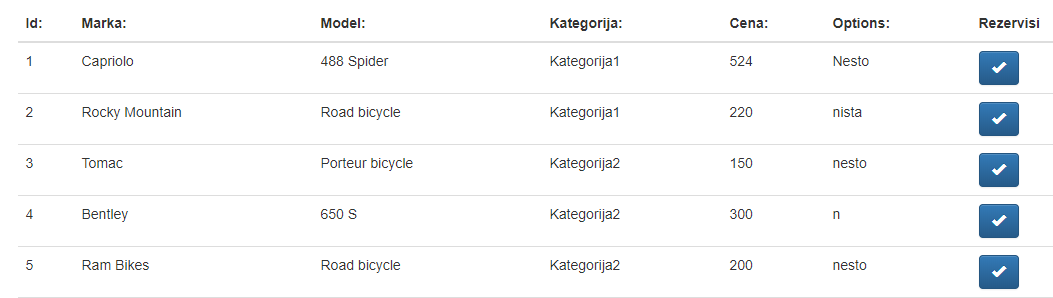
Slika 5.5. – Pregled bicikala



Slika 5.6. – Pregled rezervacija



Slika 5.7. – Pregled korisnika



Slika 5.8. – Pregled bickala za korisnike

# **Planovi za unapređenje aplikacije**

Što se tiče planova za unapređenje aplikacije, potrebno je:

Omogućiti da se rentiranje može vršiti i u drugim gradovima

Pretraga bi bila na taj način moguća i prema gradu

Dodati službenike tj. korisnike koji će biti zaduženi za unos informacija i pregled iznajmljivanja

Na taj način bi se smanjio posao administratoru

Dodati mapu grada sa informacijama gde se tačno nalazi agencija

# **Zaključak**

Tokom rada na ovom projektu, prošli smo kroz najbitnije faze razvoja sistema. Kreirali smo funkcionalne i nefunkcionalne zahteve, slučajeve korišćenja, klasne i sekvencijalne dijagrame, zatim aplikacijski i infrastrukturni dijagram. Takođe smo kreirali i bazu za sistem. Na kraju je izvšeno testiranje najvažnijih funkcionalnosti sistema. Vidimo da je potrebno mnogo vremena da bi se sve ove faze prošle, nakon čega tek možemo početi sa programiranjem. Takođe, zaključujemo da je programiranje znatno lakše, kada prođemo kroz sve faze i tačno znamo kako će funkcionistati naš sistem, pa u skladu sa tim se kasnije i razvija isti.