

# Programsko inženjerstvo

Ak. god. 2022./2023.

## CjenikSvega

Dokumentacija, Rev. 2

Grupa: *Životinjsko Carstvo*

Voditelj: *Joško Vrsalović*

Datum predaje: *13. 1. 2023.*

Nastavnik: *Igor Stančin*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>5</b>
<b>3 Specifikacija programske potpore</b>	<b>9</b>
3.1 Funkcionalni zahtjevi . . . . .	9
3.1.1 Obrasci uporabe . . . . .	11
3.1.2 Sekvencijski dijagrami . . . . .	21
3.2 Ostali zahtjevi . . . . .	24
<b>4 Arhitektura i dizajn sustava</b>	<b>25</b>
4.1 Baza podataka . . . . .	26
4.1.1 Opis tablica . . . . .	27
4.1.2 Dijagram baze podataka . . . . .	32
4.2 Dijagram razreda . . . . .	33
4.3 Dijagram stanja . . . . .	35
4.4 Dijagram aktivnosti . . . . .	36
4.5 Dijagram komponenti . . . . .	37
<b>5 Implementacija i korisničko sučelje</b>	<b>38</b>
5.1 Korištene tehnologije i alati . . . . .	38
5.2 Ispitivanje programskog rješenja . . . . .	40
5.2.1 Ispitivanje komponenti . . . . .	40
5.2.2 Ispitivanje sustava . . . . .	41
5.3 Dijagram razmještaja . . . . .	46
5.4 Upute za puštanje u pogon . . . . .	47
<b>6 Zaključak i budući rad</b>	<b>53</b>
<b>Popis literature</b>	<b>54</b>
<b>Indeks slika i dijagrama</b>	<b>57</b>

**Dodatak: Prikaz aktivnosti grupe**

**58**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak. Napisan opis zadatka.	Vlado Perković	29.10.2022.
0.2	Dodani obrasci uporabe.	Lovro Vrsalović	3.11.2022.
0.3	Dodana 3 sekvencijska dijagrama, dodane dvije slike u opis zadatka.	Vlado Perković	10.11.2022.
0.4	Dodani dijagrami obrazaca uporabe, dodan UC15	Lovro Vrsalović	16.11.2022.
0.5	Dodano poglavlje 3.2 Ostali zahtjevi	Lovro Vrsalović	16.11.2022.
0.6	ažurirana tablica aktivnosti i dodan dnevnik sastajanja	Vlado Perković	16.11.2022.
0.7	Dodan opis arhitekture i dizajna	Luka Zekan	16.11.2022.
0.8	Dijagram razreda	Petar Novak	17.11.2022.
0.9	Opis baze podataka, tablice entiteta i E-R dijagram	Anton Macan	17.11.2022.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Vlado Perković	17.11.2022.
1.1	Dodan dijagram komponenti i napisane korištene tehnologije	Vlado Perković	4.1.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.2	Dijagram razmještaja	Vlado Perković	5.1.2023.
1.3	dodane upute za puštanje u pogon	Vlado Perković	7.1.2023.
1.4	izmjena stvari prvog ciklusa	Lovro Vrsalović	9.1.2023.
1.5	dodan dijagram stanja i aktivnosti	Lovro Vrsalović	11.1.2023.
1.6	dodan zaključak	Vlado Perković	11.1.2023.
1.7	dodano ispitivanje	Vlado Perković	13.1.2023.
2.0	Konačni tekst dokumentacije	Vlado Perković	13.1.2023.

## 2. Opis projektnog zadatka

Cilj projekta je izgraditi web-aplikaciju koja će omogućiti korisnicima praćenje cijena proizvoda u trgovinama u kojoj cijene ažuriraju registrirani korisnici uz prik- ladne nagrade za aktivnost. Zamislimo sljedeću situaciju: Mario, student Fakulteta elektrotehnike i računarstva, želi minimizirati potrošnju novca na svakodnevne namirnice koje nabavlja u obližnjim trgovinama. Kako bi to učinio, Mario prati kataloge koje pronade na web-stranicama istih trgovina te nakon pronalaska (ili ne pronalaska) određenih proizvoda razmisli isplati li mu se otići u najbližu trgo- vinu ili je bolje posjetiti više njih. Ovdje se javljaju očiti problemi da nije sve na jednom mjestu te nerijetko ti katalogi uopće nisu pretraživi. Drugi će se korisnici možda odlučiti na fizičke kataloge koji su loša opcija i za okoliš i za pretraživanje. Još jedna mana kataloga jest da ne prikazuju cijene svih proizvoda, nego samo onih sniženih.

Kao rješenje na navedene probleme nastaje aplikacija CjenikSvega. Svi pro- izvodi s cijenama u stvarnom vremenu na jednom mjestu. Aplikacija će biti od najveće koristi osobi koja živi u blizini nekoliko različitih trgovina, na primjer stu- dent u domu ili odrasla osoba u gradskoj četvrti.

Ta osoba kao **neregistrirani** korisnik može pregledavati i pretraživati sadržaj aplikacije. Odluči li se **registrirati**, dodatno će moći unositi cijene proizvoda i dodavati oznake (engl. tag) proizvodima. Pri registraciji osoba unosi svoje ime, prezime, nadimak i e-mail adresu. Korisnik svoje podatke može uređivati, može odabrati što će od podataka biti javno te može obrisati korisnički račun. Jedan od primjera korištenja aplikacije jest da registrirani korisnik opazi razliku u cijenama proizvoda u trgovini i na aplikaciji te uslika taj proizvod s cijenom. Sliku će preko sučelja postaviti na web odakle će administrator odobriti izmjenu cijene. Valja napomenuti da korisnik šalje sliku jedino kada postoji odstupanje u cijeni. Ako administrator potvrdi promjenu cijene, šalje se obavijest i korisniku i trgovini. Po- hranjuju se sve promjene koje su nastale intervencijom korisnika te se to bilježi na stranici trgovine kao krivi unos.

**Administrator** ima sve mogućnosti kao i registrirani korisnik. Uz to može zabraniti pristup stranici registriranim korisnicima ili trgovinama i napisati komentar o svakoj pojedinoj trgovini koji je onda istaknut na stranicama trgovine. Dakle, pronađu li se neki korisnici koji pokušaju prevariti sistem lažiranim fotografijama proizvoda i cijena, administrator će ih blokirati i zabraniti pristup aplikaciji. Primjer jednog komentara kojeg bi administrator mogao ostaviti trgovini jest: "Trgovina xy u posljednjih mjesec dana nije imala nijedan pogrešan unos cijena."

**Trgovine** se registriraju na aplikaciju kako bi postavljale cijene **vlastitih** proizvoda. Prilikom registracije trgovina mora unijeti popis svih svojih proizvoda i njihove standardne cijene. Trgovina svaki dan ažurira cijene koje odstupaju od standardnih tako što postavi datoteku na računalo koju aplikacija automatski dohvaća. U slučaju da nema promjena, aplikacija pretpostavlja da za taj dan vrijede sve standardne cijene.

Svaki proizvod može imati do pet oznaka (engl. tag) vrste proizvoda. Primjeri oznaka su pekarski proizvod, mliječni proizvod, hrana, piće i slično. Konkretno za čips oznake bi mogle biti hrana, slane grickalice. Registrirani korisnici mogu predložiti do 5 oznaka po proizvodu. Prikaz oznaka određuje se većinskim glasanjem, odnosno prikazuju se onih pet za koje se najviše ljudi složilo da dobro opisuju navedeni proizvod. Svaki proizvod ima vlastitu stranicu na kojoj su prikazani detalji i slika proizvoda, sve trgovine u kojima se proizvod može kupiti te kako su se cijene mijenjale u zadnjih sedam dana u svakoj pojedinoj trgovini.

Rješenje problema pretraživanja jest tražilica u aplikaciji koja proizvode pretražuje po nazivu i/ili po oznakama. Osim proizvoda, mogu se pretraživati i trgovine. Korisnik može filtrirati rezultate tražilice po cijeni, popularnosti...

Rješenje koje gleda na sličan problem iz druge perspektive koje je dostupno za preuzimanje jest mobilna aplikacija Listonic, slika 2.1. Glavna misao vodilja aplikacije jest da se namirnice na popisima generalno ponavljaju te umjesto da se ponovno piše popis, nudi digitalno rješenje. Razlikuje se od našeg rješenja tako što nije povezano s trgovinama koje postavljaju cijene nego korisnici sami zadaju cijene. Pogledamo li malo šire, na primjer aplikacija Basket, slika 2.2, mobilna apli-

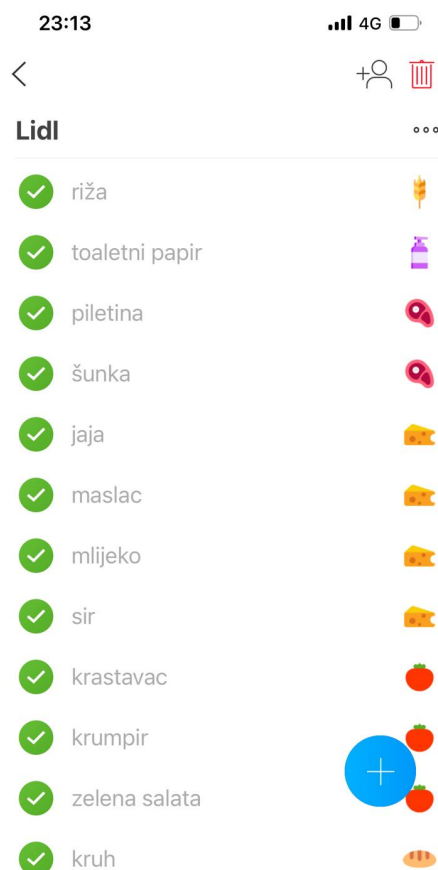
kacija nedostupna korisnicima na ovim prostorima koja pruža uslugu dostavljanja proizvoda. Funkcionira na sličan način kao naše rješenje, trgovine postavljaju cijene koje se ažuriraju u aplikaciji te korisnici vrlo lako pronađu proizvode koji ih zanimaju. Razlike su u tome što korisnici ne ispravljaju cijene i u tome što aplikacija nudi dostavu.

Rješenje je moguće prilagoditi kao mobilnu aplikaciju, što bi korisnicima olakšalo samo korištenje i postavljanja fotografija. Kako bi učinili cijeli projekt profitabilnim, svakako moguća prilagodba bi bila dodati reklame i premium članstva. Provele li se prilagodba na mobilnu aplikaciju, moguće je naplaćivati samu aplikaciju nakon probnog perioda.

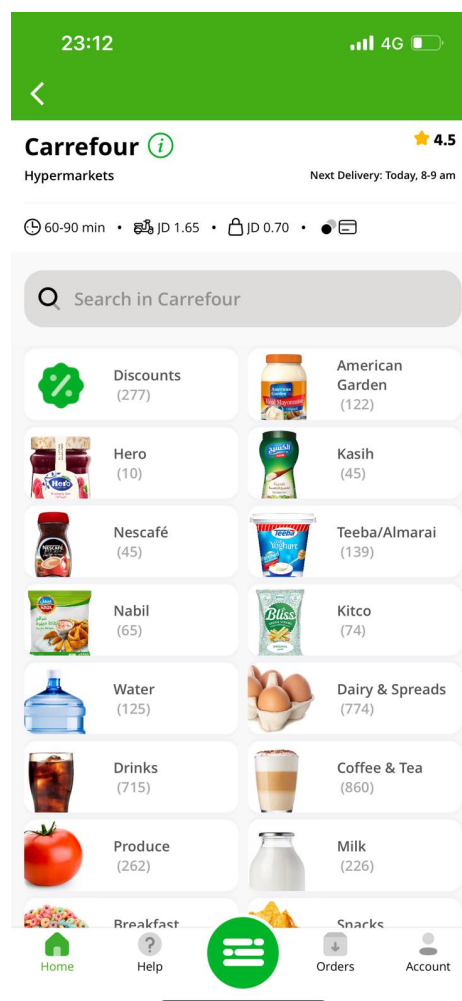
Opseg projekta je razviti web-aplikaciju s bazom podataka uz tehničku dokumentaciju. Početak je kod detaljnog definiranja zadatka u tehničkoj dokumentaciji što će dalje biti temelj projekta. Projekt se radi u dvije revizije, u prvoj reviziji isporučit će se tehnička dokumentacija sa obrascima uporabe i prikladnim dijagramima te web-aplikacija s generičkim funkcijama, dakle registracija, login i homepage. U drugoj reviziji naglasak je na implementaciji te tu web-aplikacija dobija potpunu opisanu funkcionalnost. Važan dio projekta je rad i organizacija u timu od sedam članova.

Moguće nadogradnje su sustav za planiranje rute, dijeljenje popisa proizvoda s drugim korisnicima, dodavanje proizvoda na popis skeniranjem, prikaz proizvoda gdje se nalazi unutar trgovine. Sustav za planiranje rute bi optimizirao gdje se koji proizvod kupuje tako što uzima u obzir cijenu, udaljenost između trgovina i isplativost dodatnog putovanja. Dijeljenje popisa proizvoda s drugim korisnicima je vrlo korisno u slučaju da osoba svojoj bližnjoj osobi sastavi popis i jednostavno podijeli.





Slika 2.1: Listonic



Slika 2.2: Basket

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Neregistrirani korisnik
2. Registrirani korisnik
3. Administrator
4. Trgovina
5. Razvojni tim
6. Vlasnik aplikacije
7. Asistent

#### Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
  - (a) pregledavati sadržaj aplikacije
  - (b) se registrirati u sustav, stvoriti novi korisnički račun za koji su mu potrebni korisničko ime, prezime, nadimak i e-mail
2. Registrirani korisnik (sudionik) može:
  - (a) birati što od navedenog će biti javno, a što privatno
  - (b) pregledavati sadržaj aplikacije
  - (c) unositi cijene i dodavati oznake proizvodima
  - (d) predložiti oznaku za proizvod
3. Administrator (sudionik) može:
  - (a) birati što od navedenog će biti javno, a što privatno
  - (b) pregledavati sadržaj aplikacije
  - (c) unositi cijene i dodavati oznake proizvodima

- (d) zabraniti pristup stranici registriranim korisnicima ili trgovinama
- (e) napisati komentar o svakoj pojedinoj trgovini koji je istaknut na stranici trgovine

4. Trgovina (sudionik) može:

- (a) se registrirati u sustav, te prilikom registracije mora unijeti popis svih svojih proizvoda i njihove standardne cijene
- (b) svaki dan na svoje računalo postaviti datoteku u kojoj se nalaze cijene onih proizvoda koje odstupaju od standardnih

### 3.1.1 Obrasci uporabe

#### UC1 - Pretraživanje sadržaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati proizvode i ponude
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Prilikom učitavanja aplikacije prikazuje se tražilica
  2. Korisnik upisuje željeni proizvod
  3. Aplikacija prikazuje dostupnost željenog proizvoda
- **Opis mogućih odstupanja:**
  - 2.a Pretraživanje nepostojećeg proizvoda
    - (a) Sustav obavještava korisnika kako u sustavu nema traženog proizvoda
    - (b) Korisnik mijenja potrebne podatke te završava unos ili odustaje od pretraživanja

#### UC2 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju
  2. Korisnik unosi potrebne korisničke podatke

3. Korisnik prima obavijest o uspješnoj registraciji

- **Opis mogućih odstupanja:**

2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnog e-maila

(a) Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju

(b) Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

### UC3 - Prijava u sustav

- **Glavni sudionik:** Korisnik

- **Cilj:** Dobiti pristup korisničkom sučelju

- **Sudionici:** Baza podataka

- **Preduvjet:** Registracija

- **Opis osnovnog tijeka:**

1. Unos korisničkog imena i lozinke

2. Potvrda o ispravnosti unesenih podataka

3. Pristup korisničkim funkcijama

- **Opis mogućih odstupanja:**

2.a Neispravno korisničko ime/lozinka

(a) Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu

### UC4 - Pregled osobnih podataka

- **Glavni sudionik:** Korisnik

- **Cilj:** Pregledati osobne podatke

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Osobni podatci"
  2. Aplikacija prikazuje osobne podatke korisnika

#### **UC5 - Promjena osobnih podataka**

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za promjenu podataka
  2. Korisnik mijenja svoje osobne podatke
  3. Korisnik sprema promjene
  4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 2.a Korisnik mijenja svoje podatke, ali ne odabire opciju "Spremi promjenu"
    - (a) Sustav obavještava korisnika o neuspjeloj promjeni podataka prije izlaska iz prozora

#### **UC6 - Brisanje korisničkog računa**

- **Glavni sudionik:** Korisnik
- **Cilj:** Izbrisati svoj korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**

1. Korisnik pregledava osobne podatke
2. Otvara se stranica s osobnim podacima korisnika
3. Korisnik briše račun
4. Baza podataka se ažurira tj. korisnički račun se izbriše iz baze podataka
5. Otvara se stranica za registraciju

#### **UC7 - Unos cijena proizvoda**

- **Glavni sudionik:** Korisnik
- **Cilj:** Unijeti cijene proizvoda koje odstupaju od onih na aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire trgovinu u kojoj je zapazio drugačiju cijenu
  2. Korisnik unosi novu cijenu
  3. Korisnik prilaže sliku na kojoj je vidljiva cijena u navedenoj trgovini
  4. Zahtjev se šalje administratoru na provjeru

#### **UC8 - Provjera promjene cijene**

- **Glavni sudionik:** Administrator
- **Cilj:** Potvrditi promjenu cijene
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju za pregled zahtjeva
  2. Administrator odabire zahtjev
  3. Administrator provjerava zahtjev i odobrava ga

4. Šalje obavijest o promijeni cijene korisniku i trgovini

- **Opis mogućih odstupanja:**

3.a Korisnik unosi cijenu koja je jednaka cijeni na aplikaciji

(a) Administrator odbija zahtjev

(b) Sustav šalje obavijest korisniku o neuspjeloj promijeni cijene

### UC9 - Promjena prava pristupa

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti razinu pristupa korisnika ili trgovine
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju za promjenu prava pristupa
  2. Administrator pronalazi željenog korisnika ili trgovinu
  3. Administrator mijenja razinu pristupa željenom korisniku

### UC10 - Dodavanje komentara trgovinama

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati komentar određenoj trgovini
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju za dodavanje komentara trgovini
  2. Administrator pronalazi željenu trgovinu
  3. Administrator dodaje komentar



**UC11 - Dodavanje oznaka proizvodima**

- **Glavni sudionik:** Korisnik
- **Cilj:** Dodati oznaku određenom proizvodu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire proizvod
  2. Korisnik bira oznaku za odabrani proizvod

**UC12 - Registracija trgovine**

- **Glavni sudionik:** Trgovina
- **Cilj:** Stvoriti račun za trgovinu kako bi pristupila sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Trgovina odabire opciju za registraciju trgovine
  2. Trgovina unosi potrebne podatke za registraciju trgovine
  3. Trgovina unosi popis svih proizvoda i njihove standardne cijene
  4. Trgovina prima obavijest o uspješnoj registraciji trgovine
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzetog imena trgovine i/ili e-maila, unos podatka u nedozvoljenom formatu
    - (a) Sustav obavještava trgovinu o neuspjelom upisu i vraća ga na stranicu za registraciju trgovine
    - (b) Trgovina mijenja potrebne podatke te završava unos ili odustaje od registracije

**UC13 - Prijava trgovine u sustav**

- **Glavni sudionik:** Trgovina
- **Cilj:** Trgovini omogućiti pristup sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija trgovine
- **Opis osnovnog tijeka:**
  1. Unos imena trgovine i lozinke
  2. Potvrda o ispravnosti unesenih podataka
  3. Pristup funkcijama trgovina
- **Opis mogućih odstupanja:**
  - 2.a Neispravno ime trgovine/lozinka
    - (a) Sustav obavještava trgovinu o neuspjelom upisu i vraća ju na stranicu za prijavu trgovine

**UC14 - Postavljanje datoteke s novim cijenama**

- **Glavni sudionik:** Trgovina
- **Cilj:** Postaviti nove cijene proizvoda
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija trgovine
- **Opis osnovnog tijeka:**
  1. Trgovina postavlja datoteku na računalo
  2. Web-aplikacija dohvaća postavljenu datoteku i ažurira bazu podataka
- **Opis mogućih odstupanja:**
  - 2.a Trgovina nije priložila datoteku
    - (a) Sustav pretpostavlja da nema promjena u cijenama

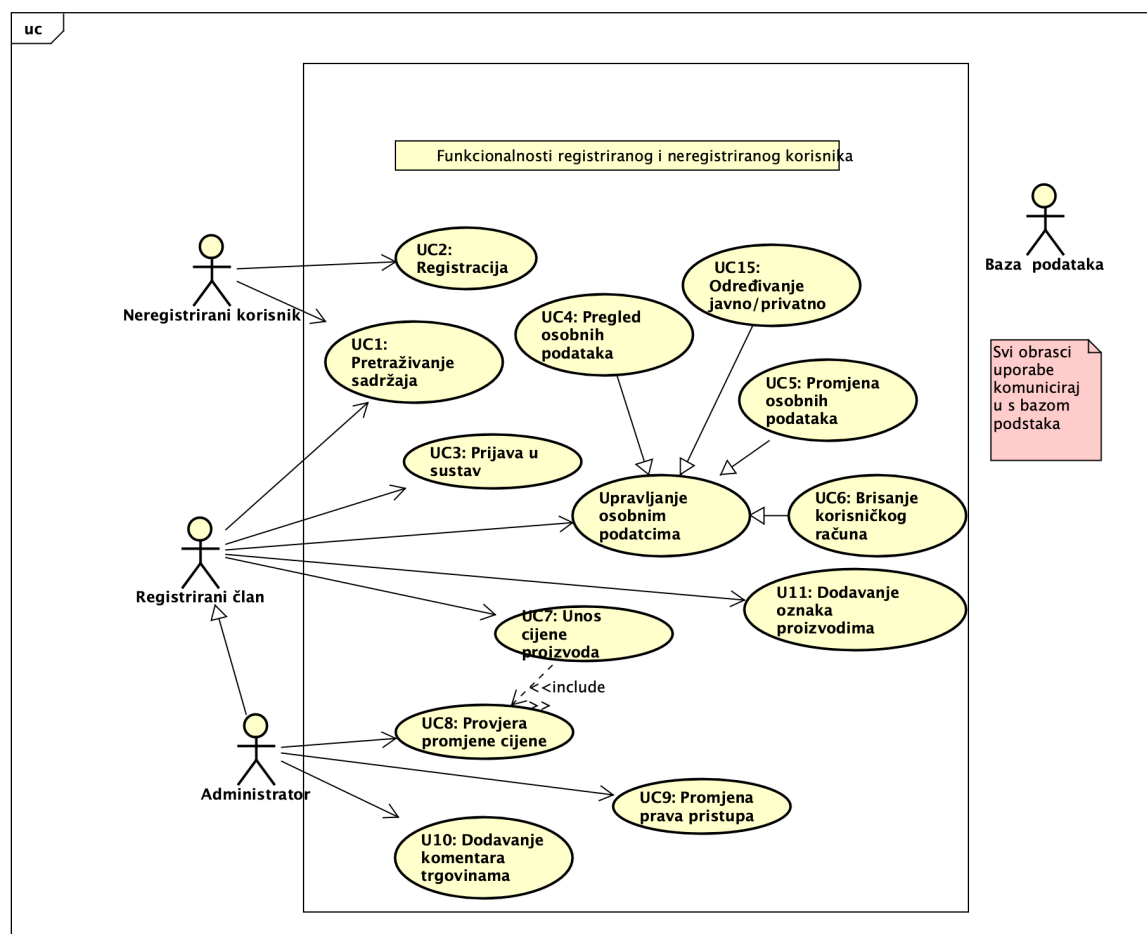
2.b Trgovina je priložila neispravnu datoteku

(a) Sustav šalje poruku trgovini o neispravnoj datoteci

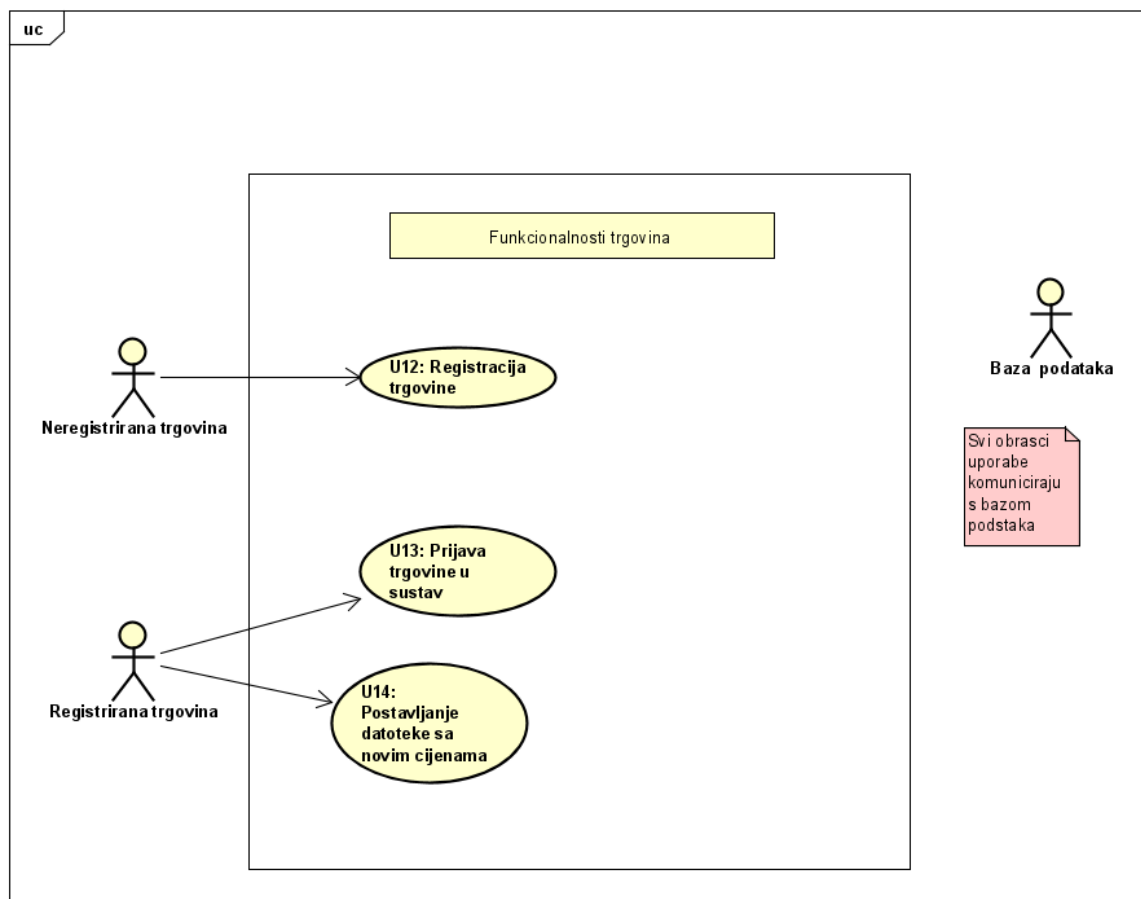
#### **UC15 - Određivanje javno/privatno**

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijeniti vidljivost svojih podataka (javno/privatno)
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za uređivanje osobnog profila
  2. Korisnik odlučuje što će biti javno. a što dostupno privatno

## Dijagrami obrazaca uporabe



Slika 3.1: Use case diagram - Korisnik

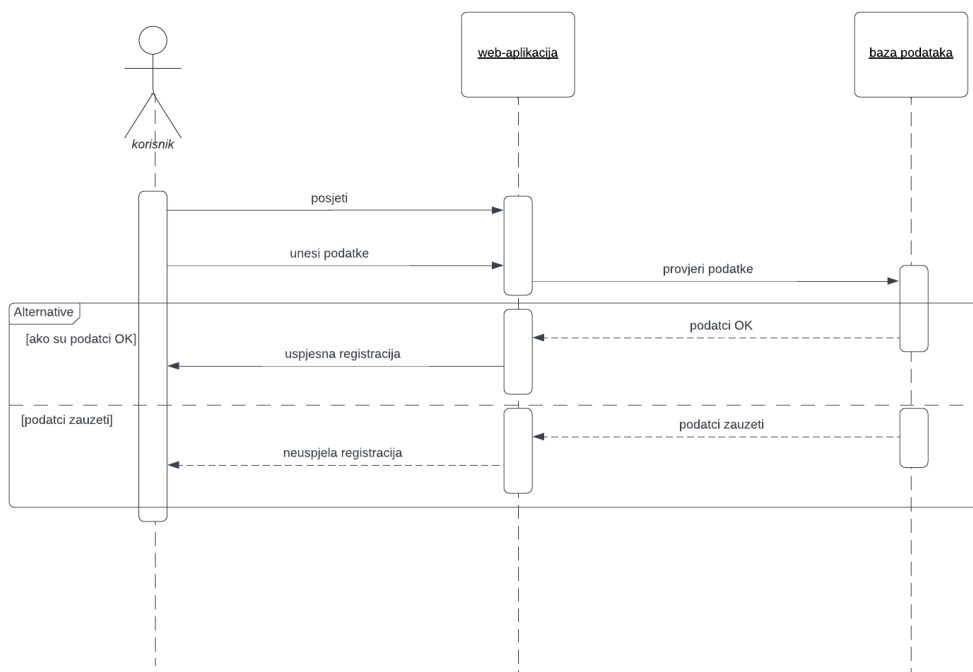


Slika 3.2: Use case dijagram - Trgovina

### 3.1.2 Sekvencijski dijagrami

#### UC2 - Registracija

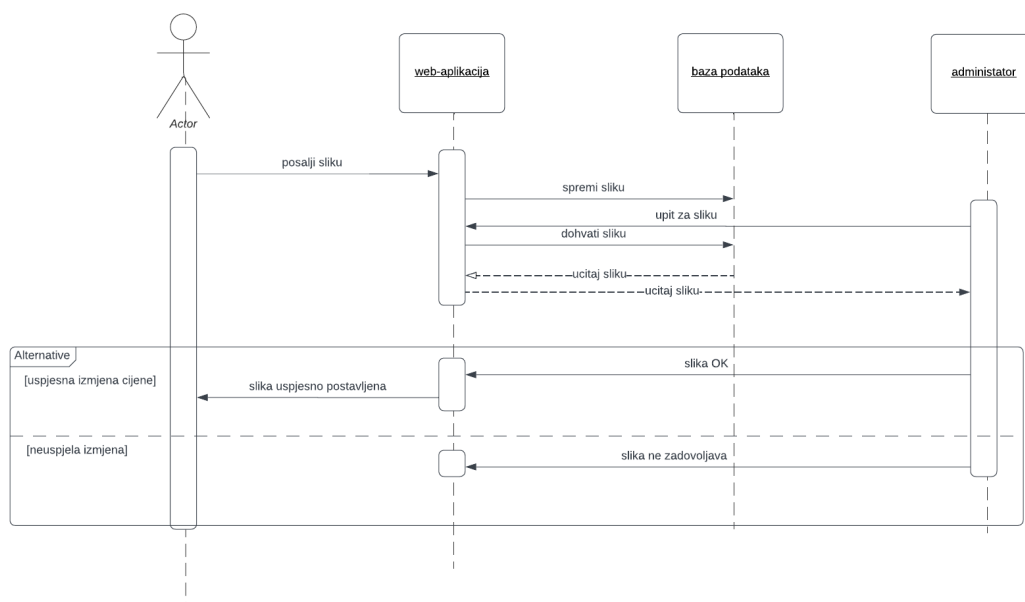
Neregistrirani korisnik želi se registrirati kako bi dobio pogodnosti registriranog korisnika. Kako bi to učinio prvu odabire opciju za registraciju nakon koje mu se pokazuje register page. Unosi korisničke podatke u predviđena polja (e-mail, korisničko ime, ime, prezime i lozinku.) Aplikacija zatim provjerava s bazom postoje li profili s tom e-mail adresom ili tim korisničkim imenom. Ako su e-mail i/ili korisničko ime zauzeti, aplikacija dojavljuje pogrešku, vraća ga na stranicu za registraciju te prikazuje obavijest da izabere neku drugu e-mail adresu ili korisničko ime. Korisnik može isprobavati slobodne opcije sve dok jedna kombinacija ne bude jedinstvena ili može odustati od registracije. Kada korisnik unese jedinstvenu kombinaciju e-maila i korisničkog imena, uspješno se registrira i sustav mu šalje obavijest o uspješnoj registraciji.



Slika 3.3: UC2 - sekvencijski dijagram

### UC7 - Unos cijene proizvoda, UC8 - Provjera promjene cijene

Ako postoji razlika u cijenama u trgovini i na web-stranici, korisnik odabire trgovinu u kojoj je zapazio drugačiju cijenu, unosi novu cijenu, šalje sliku proizvoda i stvarne cijene u aplikaciju. Aplikacija tu sliku sprema u bazu podataka odakle administrator odabire opciju za pregled zahtjeva, odabire zahtjev te preuzima sliku na pregled. Ako su slike u redu, administrator šalje potvrdu, cijena se promijeni te se pošalje poruka potvrde korisniku i trgovini. Ako se cijene ne podudaraju ili je zahtjev neispravan, administrator odbija zahtjev i sustav šalje obavijest korisniku o neuspjeloj promjeni cijene.

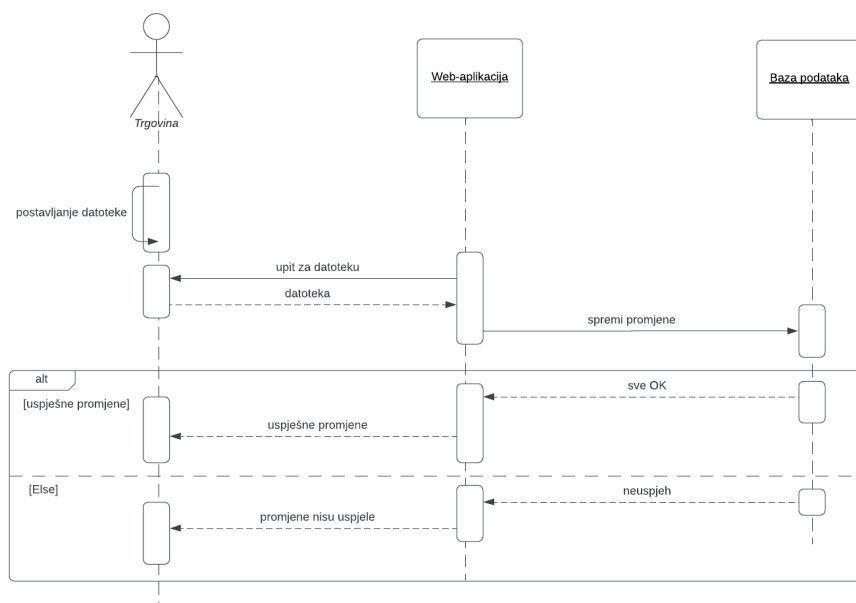


Slika 3.4: UC7, UC8 - sekvencijski dijagram

### UC14 - Postavljanje datoteke sa novim cijenama

Početkom svakog radnog dana trgovina unosi datoteku s promjenama cijena na svoje računalo. Web-aplikacija automatski preuzima tu datoteku i ažurira cijene u bazi podataka. Ako trgovina nije postavila datoteku, podrazumijeva se da nema promjena. Ako trgovina neispravno postavi datoteku, sustav obavještava trgovinu

o neuspjeloj predaji.



Slika 3.5: UC14 - sekvencijski dijagram

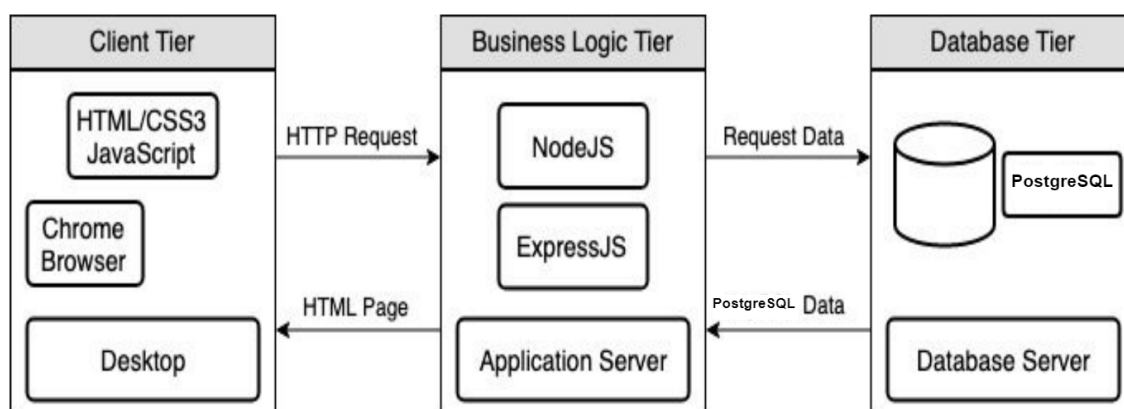


## 3.2 Ostali zahtjevi

- Omogućen istovremeni rad više korisnika
- Postupak pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Funkcionalnost i rad sustava ne smiju biti narušeni neispravnim korištenjem korisničkog sučelja.
- Sustav treba biti jednostavan i intuitivan za korištenje
- Postojeće funkcionalnosti sustava trebaju biti održane pri nadogradnji sustava
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Pristup serveru mora biti omogućen iz javne mreže pomoću HTTPS
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Sustav kao valutu koristi EUR

## 4. Arhitektura i dizajn sustava

Stil arhitekture koji smo odabrali je arhitektura zasnovana na događajima gdje se događaji javno objavljuju te se pozivaju registrirane procedure, dok komponente koje objavljuju događaj nemaju informaciju koje će sve komponente reagirati i kako. Za razliku od objektno usmjerenog stila, komponente se ne pozivaju eksplicitno, već generiraju signale, tj. događaje. Ova arhitektura je odabrana jer je najefikasnija za obradu korisničkih zahtjeva, laka je za održavanje i reciklabilna za potrebe budućih projekata ili nadogradnje ovog. Što se tiče spremišta podataka, svi podaci će biti spremeni te dohvaćani iz univerzalne baze podataka. Mrežni protokoli koji će se pozivati tokom komunikacije klijenta s poslužiteljem su: TCP, IP, HTTP. Sustav je organiziran na sljedeći način:



Slika 4.1: grafički prikaz arhitekture

Korisnik putem odabranog web preglednika šalje HTTP zahtjev za web aplikacijom i njenim komponentama zadanom web poslužitelju. Aplikacija komunicira s bazom podataka u kojoj su sadržani svi podatci i iz nje izvlači sve potrebne podatke za klijentov zahtjev. Poslužitelj, kada aplikacija dohvati sve potrebne podatke, odgovara na klijentov zahtjev sa statusom 200 OK (ako je sve u redu) i šalje mu HTML dokument koji se prikazuje u zadanom web pregledniku. Programski jezik koji smo koristili za izradu web aplikacije je Javascript. Arhitektura sustava se temelji na MVC konceptu, stilističkoj varijaciji arhitekture zasnovanoj na događajima.

Odabrali smo taj koncept jer odvaja korisničko sučelje od ostatka sustava, što čini razvoj i nadogradnju komponenata jednostavnijima. Sastoji se od modela, pogleda i upravitelja.

- Model – dojavljuje sebi pridruženim pogledima i upravitelju kada je došlo do promjene u njegovom stanju. Ove dojave omogućuju pogledu da prikaže obnovljeno stanje modela, a upravitelju promjenu dostupnog skupa naredbi
- View (Pogled) - od modela dobija informacije koje su mu potrebne za prikaz korisniku
- Controller (Upravitelj) – šalje naloge modelu koji ažurira svoje stanje i naredbe pogledima kojima mijenja prikaz modela

## 4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo relacijsku bazu podataka koja svojom strukturom nam olakšava da zorno modeliramo stvarni svijet. Baza podataka se sastoji od tablica definiranih imenom i atributima. Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvat podataka za daljnju obradu. Baza podataka sastoji se od ovih entiteta:

- Korisnik
- Proizvod
- Oznake
- Komentar
- Pretinac
- Privatnost
- Obavijest
- ProizvodTrgovina
- PromjenaCijenaKorisnik
- PromjenaCijenaTrgovina

- Trgovina
- session

#### 4.1.1 Opis tablica

Entitet **Korisnik** sadržava sve važne informacije o registriranim korisnicima aplikacije. Sadrži attribute: ID, Ime, Prezime, Email, Nadimak, Lozinka, RazinaPristupa i ZabranjenPristup. Ovaj entitet u vezi je One-to-Many s Oznake preko KorisnikID, u vezi je Many-to-Many s Komentar preko KorisnikID, u vezi je Many-to-Many s Pretinac preko KorisnikID, u vezi je One-to-One s Privatnost preko KorisnikID i u vezi je Many-to-Many s PromjenaCijenaKorisnik preko KorisnikID.

Korisnik		
ID	INT	Unikatni identifikator svakog prijavljenog korisnika
Ime	VARCHAR	Ime korisnika
Prezime	VARCHAR	Prezime korisnika
Email	VARCHAR	Email korisnika
Nadimak	VARCHAR	Unikatni nadimak korisnika
Lozinka	VARCHAR	Korisnikova lozinka za pristup računu
RazinaPristupa	SMALLINT	Ako je 0 onda je registrirani korisnik, ako je 1 onda je trgovina i ako je 2 onda je administrator
ZabranjenPristup	BOOLEAN	Ako je true onda je administrator zabranio pristup ovom profilu

Entitet **Trgovina** sadržava informacije vezane za određenu trgovinu. Sadrži attribute: ID i Naziv. Ovaj entitet u vezi je Many-to-Many s entitetom PromjenaCijenaTrgovina preko atributa TrgovinaID, u vezi je Many-to-Many PromjenaCijenaKorisnik preko atributa TrgovinaID, u vezi je Many-to-Many ProizvodTrgovina preko atributa TrgovinaID i u vezi je Many-to-One s entitetom Komentar preko atributa TrgovinaID.

Trgovina		
ID	INT	Unikatni identifikator svake trgovine
Naziv	VARCHAR	Naziv trgovine

Entitet **Proizvod** sadržava osnovne informacije o proizvodima. Sadrži attribute: Barkod i Naziv. Ovaj entitet u vezi je Many-to-Many s entitetom Oznake preko atributa Barkod, u vezi je One-to-Many s ProizvodTrgovina preko atributa Barkod, u vezi je One-to-Many s PromjenaCijenaKorisnik preko atributa Barkod i u vezi je One-to-Many s PromjenaCijenaTrgovina preko atributa Barkod.

Proizvod		
Barkod	VARCHAR	Unikatni identifikator proizvoda
Naziv	VARCHAR	Naziv proizvoda

Entitet **Komentar** sadržava zapise o komentarima napisanim od administratora za određenu trgovinu. Sadrži attribute: KorisnikID, TrgovinaID i OpisKomentara. Ovaj entitet u vezi je One-to-Many s entitetom Trgovina preko atributa TrgovinaID i u vezi Many-to-Many s Korisnik preko KorisnikID.

Komentar		
KorisnikID	INT	Identifikator korisnika koji je napisao komentar
TrgovinaID	INT	Identifikator trgovine kod koje je komentirani proizvod
OpisKomentara	VARCHAR	Upisani komentar

Entitet **Obavijest** sadržava informacije vezane za određenu obavijest. Sadrži attribute: ID, DatumVrijeme, Opis i Procitano. Ovaj entitet u vezi je Many-to-One s entitetom Pretinac preko atributa ObavijestID.

Obavijest		
ID	INT	Unikatni identifikator obavijesti
DatumVrijeme	TIMESTAMP	Vrijeme kada je dostavljena obavijest
Opis	VARCHAR	Opis obavijesti
Procitano	BOOLEAN	Ako je True onda je korisnik pročitao obavijest

Entitet **Oznake** sadržava zapise o proizvodima s oznakama stavljenim od korisnika. Sadrži attribute: Barkod, KorisnikID i Oznaka. Ovaj entitet u vezi je Many-to-Many s entitetom Proizvod preko atributa Barkod i u vezi Many-to-One s Korisnik preko KorisnikID.

Oznake		
Barkod	VARCHAR	Identifikator proizvoda pod kojim je oznaka
KorisnikID	INT	Identifikator korisnika koji je napisao oznaku
Oznake	VARCHAR	Naziv oznake

Entitet **Pretinac** sadržava zapise o obavijestima kod korisnika. Sadrži attribute: KorisnikID i ObavijestID. Ovaj entitet u vezi je One-to-Many s entitetom Obavijest preko atributa ObavijestID i u vezi Many-to-Many s Korisnik preko KorisnikID.

Pretinac		
KorisnikID	INT	Identifikator korisnika koji je dobio obavijest
ObavijestID	INT	Identifikator obavijesti

Entitet **Privatnost** sadržava zapise o kojim informacijama se prikazuju na profilu korisnika. Sadrži attribute: KorisnikID, Ime, Prezime, Email i Nadimak. Ovaj entitet je u vezi One-to-One s Korisnik preko KorisnikID.

Privatnost		
KorisnikID	INT	Identifikator korisnika
Ime	BOOLEAN	Ako je true onda se ime prikazuje na profilu, ako je false onda se ne prikazuje
Prezime	BOOLEAN	Ako je true onda se prezime prikazuje na profilu, ako je false onda se ne prikazuje
Email	BOOLEAN	Ako je true onda se email prikazuje na profilu, ako je false onda se ne prikazuje
Nadimak	BOOLEAN	Ako je true onda se nadimak prikazuje na profilu, ako je false onda se ne prikazuje

Entitet **ProizvodTrgovina** sadržava zapise o proizvodima u trgovinama. Sadrži attribute: Barkod, TrgovinaID i Cijena. Ovaj entitet u vezi je Many-to-Many s entitetom Trgovina preko atributa TrgovinaID i u vezi Many-to-One s Proizvod preko Barkod.

ProizvodTrgovina		
Barkod	VARCHAR	Identifikator proizvoda
TrgovinaID	INT	Identifikator trgovine u kojoj je proizvod
Cijena	MONEY	Cijena proizvoda u određenoj trgovini

Entitet **PromjenaCijeneKorisnik** sadržava zapise o promjenama cijena koje je upisao korisnik. Sadrži attribute: KorisnikID, Barkod, TrgovinaID, DatumVrijeme, NovaCijena, Slika i Status. Ovaj entitet u vezi je Many-to-Many s entitetom Korisnik preko atributa KorisnikID, u vezi Many-to-One s Proizvod preko Barkod i u vezi Many-to-Many s Trgovina preko atributa TrgovinaID.

PromjenaCijeneKorisnik		
Barkod	VARCHAR	Identifikator proizvoda

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

PromjenaCijeneKorisnik		
KorisnikID	INT	Identifikator korisnika koji je upisao promjenu cijene
TrgovinaID	INT	Identifikator trgovine gdje je pogrešna cijena
DatumVrijeme	TIMESTAMP	Vrijeme unosa nove cijene
NovaCijena	MONEY	Nova cijena proizvoda u određenoj trgovini
Slika	VARBIT	Slika kao dokaz nove cijene
Status	VARCHAR	Kakav je status promjene cijene

Entitet **PromjenaCijeneTrgovina** sadržava zapise o promjenama cijena koje je upisala trgovina. Sadrži attribute: Barkod, TrgovinaID, DatumVrijeme, NovaCijena, Slika i Status. Ovaj entitet u vezi je Many-to-Many s entitetom Trgovina preko atributa TrgovinaID i u vezi Many-to-One s Proizvod preko Barkod.

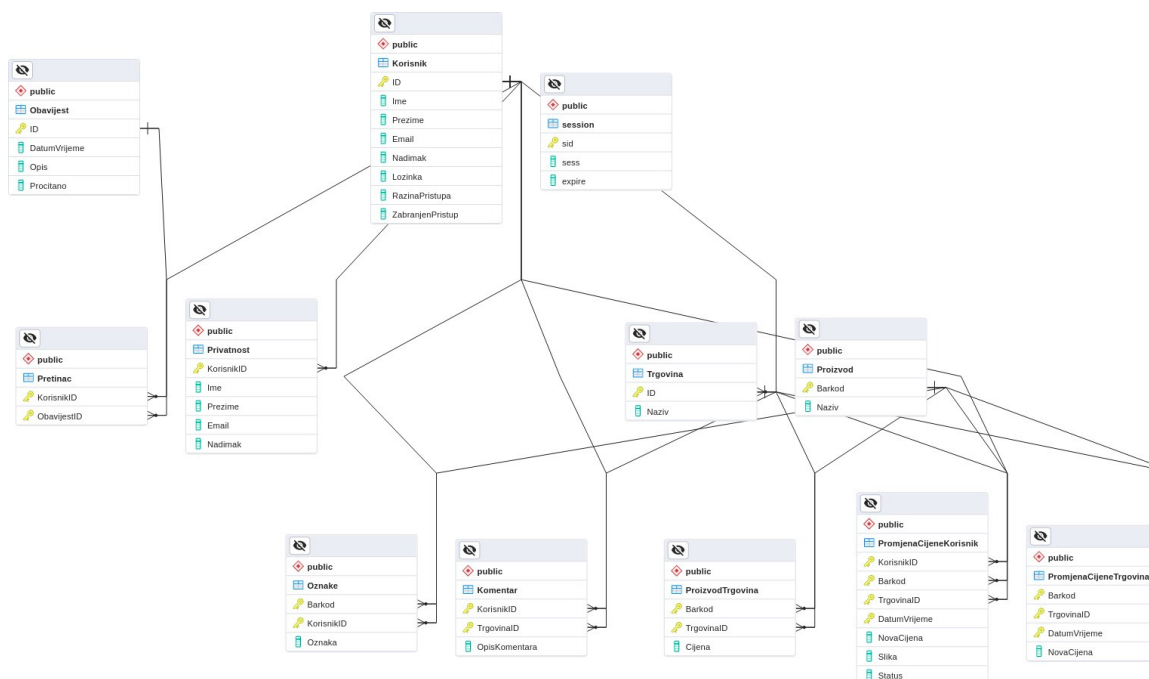
PromjenaCijeneTrgovina		
Barkod	VARCHAR	Identifikator proizvoda
TrgovinaID	INT	Identifikator trgovine gdje je promjena cijene
DatumVrijeme	TIMESTAMP	Vrijeme unosa nove cijene
NovaCijena	MONEY	Nova cijena proizvoda u određenoj trgovini

Entitet **session** sadržava zapise o sjednicama. Sadrži attribute: sid, sess i expire.

session		
sid	VARCHAR	Identifikator sjednice
sess	JSON	Json zapis sjednice
expire	TIMESTAMP	Datum i vrijeme isteka sjednice



## 4.1.2 Dijagram baze podataka



Slika 4.2: E-R Dijagram baze podataka

## 4.2 Dijagram razreda

**UserModel** služi za spremanje podataka o korisniku te za autentifikaciju i autorizaciju korisnika. **UserModel** generalizira **Admin** i **StoreModel**.

**PrivacyModel** pohranjuje podatke o tome koji će podaci o korisniku biti javni, a koji privatni.

**Admin** sadrži metode koje može izvršiti isključivo korisnik s admin pravom pristupa(odobravanje promjena cijena, zabrana pristupa drugim korisnicima...)

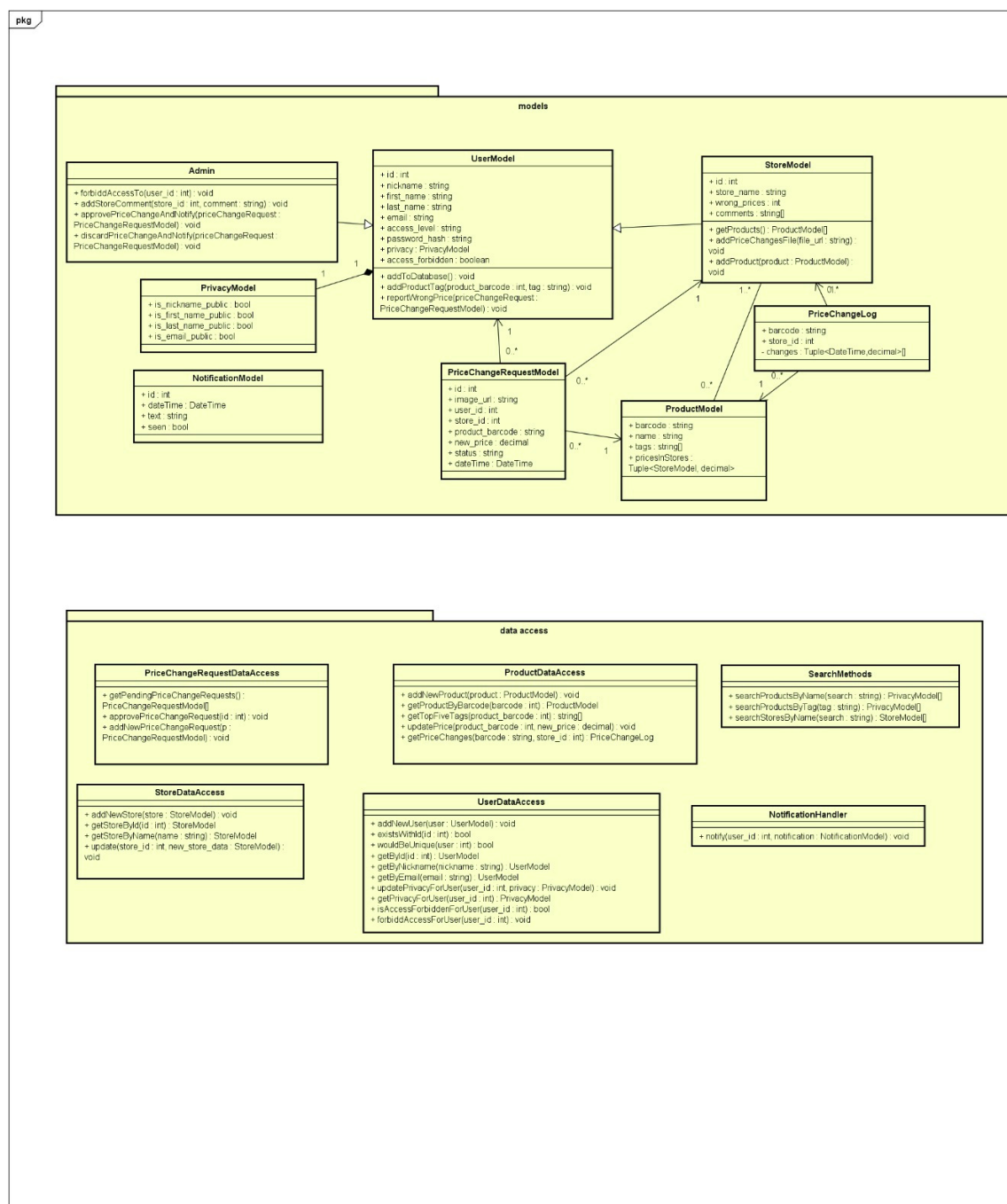
**StoreModel** modelira trgovinu.

**ProductModel** modelira proizvod. Proizvod je jednoznačno određen barkodom kako bi više trgovina moglo imati isti proizvod u ponudi.

**PriceChangeLog** služi za opisivanje promjena cijena proizvoda u nekoj trgovini u nekom vremenskom razdoblju.

**PriceChangeRequestModel** modelira zahtjev za promjenom cijene koje šalje korisnik za proizvod u nekoj trgovini u kojoj se stvarna cijena razlikuje od cijene navedene u aplikaciji.

**NotificationModel** modelira obavijesti koje admin može slati korisniku ili trgovini. Klase iz data access paketa sadržavaju isključivo statičke metode koje služe za komunikaciju s bazom podataka.



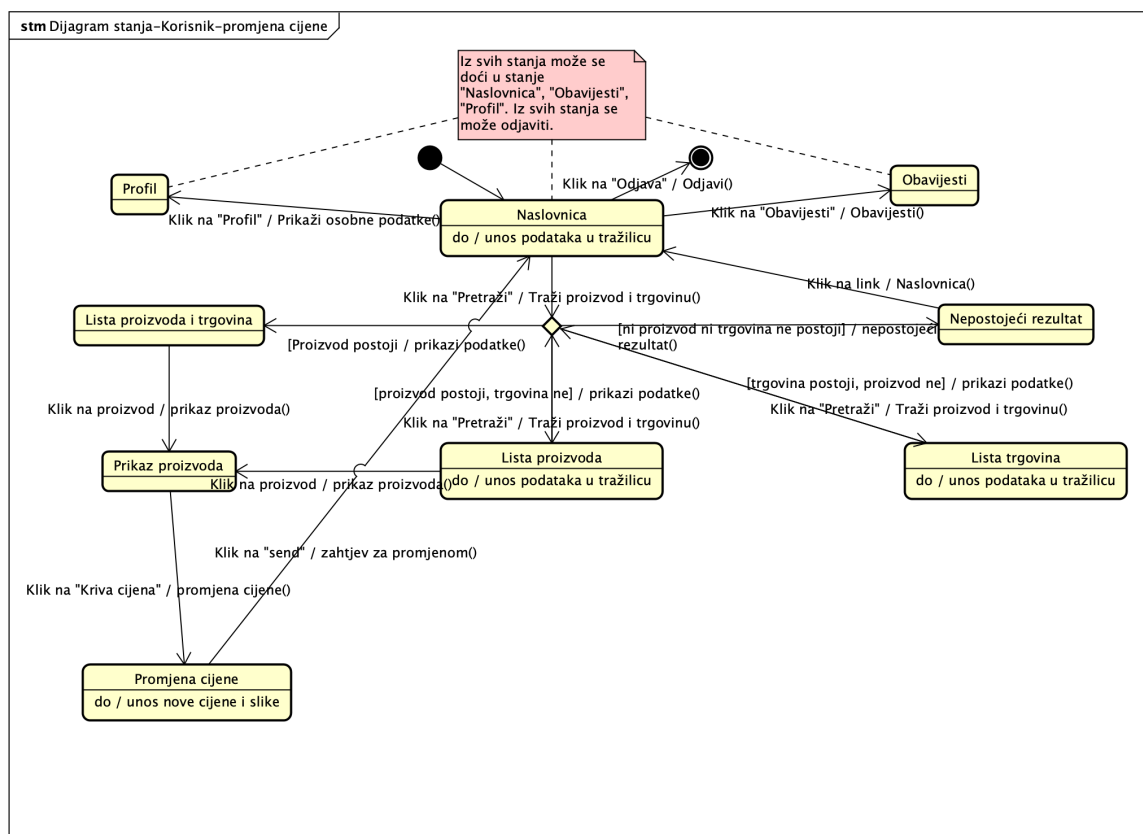
Slika 4.3: dijagram razreda

**dio 2. revizije**

*Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije*

### 4.3 Dijagram stanja

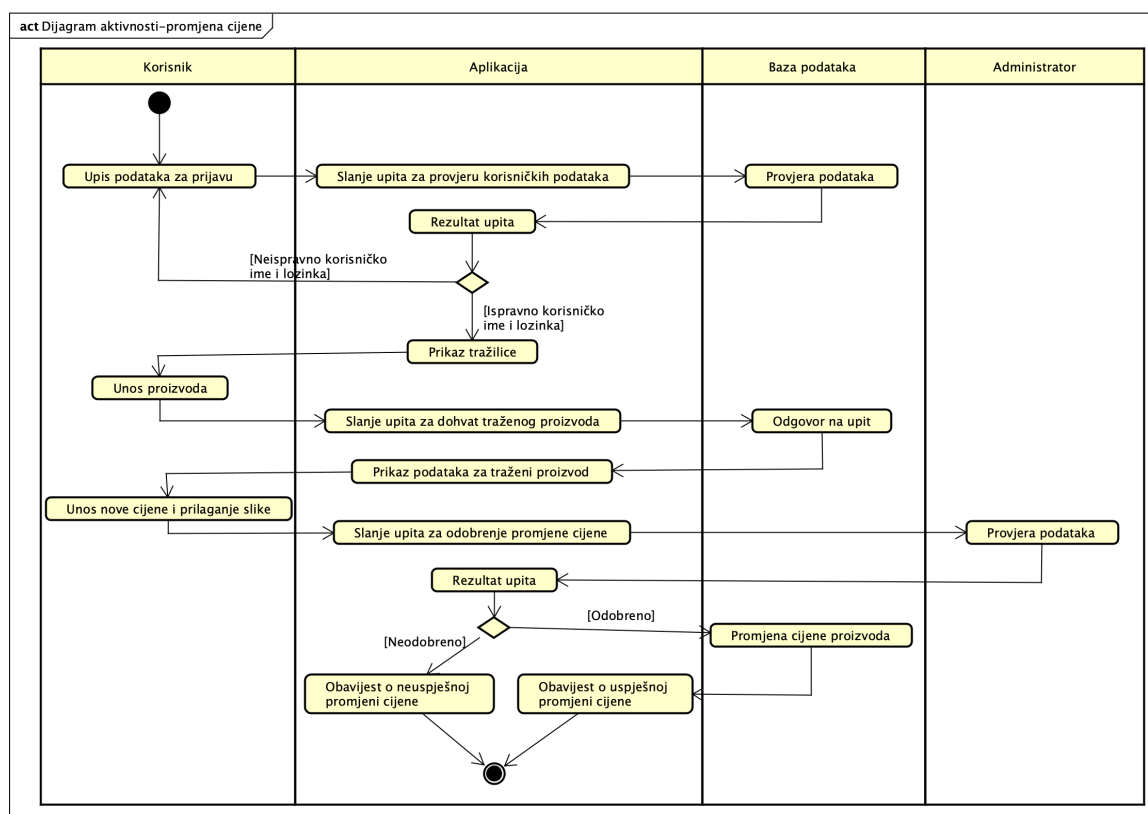
Dijagram stanja prikazuje stanje objekta te prijelaze iz jednog stanja u drugo temeljeno na događajima. Na slici 4.4 prikazan je dijagram stanja za registriranog korisnika. Nakon prijave, korisniku se prikazuje početna stranica na kojoj se mogu pretraživati željene proizvode. Za odabrani proizvod korisnik ima opciju predložiti do pet oznaka. Korisnik također za svaki proizvod može unijeti novu cijenu ako je uočio da se cijena u aplikaciji razlikuje od stvarne cijene. Prilikom unosa nove cijene prilaže se slika na kojoj je vidljiva cijena u navedenoj trgovini. Zatim se slika šalje administratoru na odobrenje. Ako administrator potvrdi promjenu cijene, obavijest o promjeni cijene dobivaju i korisnik i trgovina koju možete vidjeti klikom na "Obavijesti". Klikom na "Profil" korisniku se prikazuju njegovi podaci koje je moguće mijenjati. Klikom na "Odjava" korisnik se odjavljuje iz sustava."



Slika 4.4: dijagram stanja

## 4.4 Dijagram aktivnosti

Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. Ne upotrebljava se za modeliranje događajima poticanog ponašanja. U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na dijagramu aktivnosti 4.5 prikazan je proces unosa nove cijene proizvoda. Korisnik se prijavi u sustav, pretraži željeni proizvod. Ukoliko se cijena proizvoda u aplikaciji razlikuje od stvarne cijene u trgovini unosi novu cijenu proizvoda te prilaže sliku na kojoj se vidi cijena u navedenoj trgovini. Slika se šalje administratoru na odobrenje te ukoliko administrator potvrdi promjenu cijene cijena se mijenja, a korisnik i trgovina dobivaju obavijest o promjeni cijene.

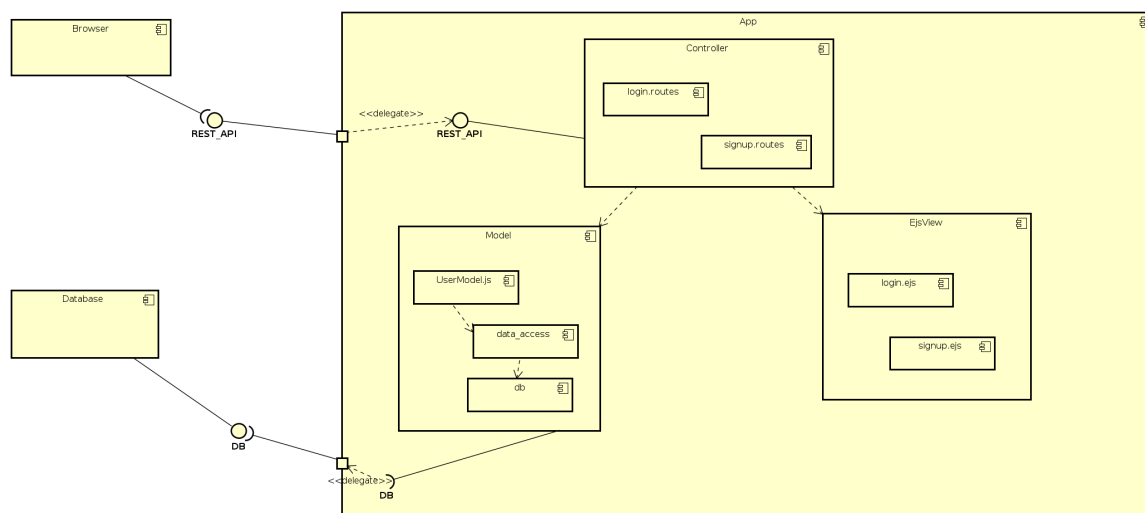


Slika 4.5: dijagram aktivnosti

## 4.5 Dijagram komponenti

### dio 2. revizije

Na prikazanom dijagramu komponenti jasno vidimo MVC model koji se koristi u izradi aplikacije. Klijent komunicira preko web-preglednika (eng. *Browser*) sa aplikacijom. U stvarnosti web-preglednik komunicira preko REST API-ja sa kontrolerom (eng. *Controller*). Kontroler se koristi modelom da dohvati podatke te vraća prikaz putem pogleda (eng. *View*). Na nižoj razini apstrakcije vidimo da se kontroler, model i pogled sastoji od mnoštva manjih komponenti pa tako u kontrolere spadaju svi usmjerivači (eng. *routers*) od kojih su samo dva prikazana, u modele spadaju razredi i funkcije kojima se pristupa bazi podataka te ostali razredi koji modeliraju podatke. U poglede spadaju sve datoteke nastavka ".ejs", tj. ono što se prikazuje klijentu. Radi izbjegavanja gomilanja komponenti na dijagramu izostavljena je većina komponenti s niže razine apstrakcije.



Slika 4.6: dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Za izradu dokumentacije korišten je alat **Texmaker**<sup>1</sup> koji je razvojna okolina za jezik **LaTeX**<sup>2</sup>. *LaTeX* je programski jezik za pisanje strukturiranih tekstova. Najčešće se koristi u oblikovanju znanstvenih radova.

Dijagrami unutar dokumentacije izrađeni su pomoću alata **Astah UML**<sup>3</sup> i **lucid.app**<sup>4</sup>. *Astah UML* aplikacija je za modeliranje UML-a. *Lucid.app* je web-stranica koja nudi isto to, no pristupačnija jer se ne mora preuzimati. Obje aplikacije se za studente ne naplaćuju.

Komunikacija tima odvijala se preko aplikacije **WhatsApp**<sup>5</sup>. Za potrebe ovog projekta bila je sasvim dovoljna, no za neke ozbiljnije projekte bilo bi potrebno prijeći na bolji alat.

Kao sustav za upravljanje izvornim kodom korišten je **Git**<sup>6</sup>. Udaljeni repozitorij projekta je dostupan na web platformi **GitLab**<sup>7</sup>.

Aplikacija je napisana u **Express-u**<sup>8</sup>, framework za izgradnju web-aplikacija na platformi **Node.js**<sup>9</sup>. Sve to je ostvareno programskim jezikom **Javascript**<sup>10</sup>. *Node.js* je back-end JavaScript runtime okruženje, radi na V8 JavaScript Engineu i izvršava JavaScript kod izvan web preglednika.

Sam kod aplikacije pisan je u proizvoljnom *code editoru*, svi smo odabrali **Visual Studio Code**<sup>11</sup>.

Baza podataka je **PostgreSQL**<sup>12</sup> u oblaku na **Renderu**<sup>13</sup>. Za lokalno korištenje

---

<sup>1</sup><https://www.xmlmath.net/texmaker/>

<sup>2</sup><https://www.latex-project.org/>

<sup>3</sup><https://astah.net/>

<sup>4</sup><https://lucid.app/>

<sup>5</sup><https://www.whatsapp.com/>

<sup>6</sup><https://git-scm.com/>

<sup>7</sup><https://gitlab.com/>

<sup>8</sup><https://expressjs.com/>

<sup>9</sup><https://nodejs.org/en/>

<sup>10</sup><https://www.javascript.com/>

<sup>11</sup><https://code.visualstudio.com/>

<sup>12</sup><https://www.postgresql.org/>

<sup>13</sup><https://render.com/>

baze korišten je alat **pgAdmin**<sup>14</sup>.

---

<sup>14</sup><https://www.pgadmin.org/>



## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

Ispitivanje komponenti proveli smo unutar Node.js tehnologije. U prva tri testa ispitivali smo funkcionalnosti trgovine. Testira se dohvaćanje imena trgovine preko ID-a, postojanje komentara za neku trgovinu i samo postojanje trgovine s nekim ID-jem.

```
test > JS TrgovinaDataAccess.test.js > ...
1  const TrgovinaDataAccess = require('../data_access/TrgovinaDataAccess')
2
3  test('The function should return a Trgovina object with Naziv parameter \'Konzum\' wiht ID 36', async () =>{
4    let res = await TrgovinaDataAccess.getTrgovina(36);
5    expect(res.naziv).toBe('Konzum');
6  })
7
8  test('The function should return false which means that a from admin with ID 100 to a shop with ID 15 doesn\'t exist', async () =>{
9    let res = await TrgovinaDataAccess.commentExists(100,15);
10   expect(res).toBe(false);
11 })
12
13 test('The function should return undefined which means that a user with ID 100 doesn\'t exist', async () =>{
14   let res = await TrgovinaDataAccess.getTrgovina(100);
15   expect(res).toBe(undefined);
16 })
```

Slika 5.1: testovi 1, 2 i 3

U druga tri testa ispitivale su se funkcionalnosti korisnika. Testiralo se postojanje korisnika s nekim ID-jem, dohvaćanje korisnika preko *username*-a i nepostojanje korisnika s nevalidnim ID-jem.

```
test > JS UserDataAccess.test.js > ...
1  const UserDataAccess = require('../data_access/UserDataAccess')
2
3  test('The function should return true which means that a user with ID 15 exists', async ()=>{
4    let res = await UserDataAccess.existsWithId(15);
5    expect(res).toBe(true);
6  });
7
8  test('The function should return an User object which ID should be 15', async () => {
9    let res = await UserDataAccess.getByNickname('trgovinal');
10   expect(res.id).toBe(15);
11 })
12
13 test('The function should return false which means that a user with ID 100 doesn\'t exist', async () =>{
14   let res = await UserDataAccess.existsWithId(100);
15   expect(res).toBe(false);
16 })
```

Slika 5.2: testovi 4, 5 i 6

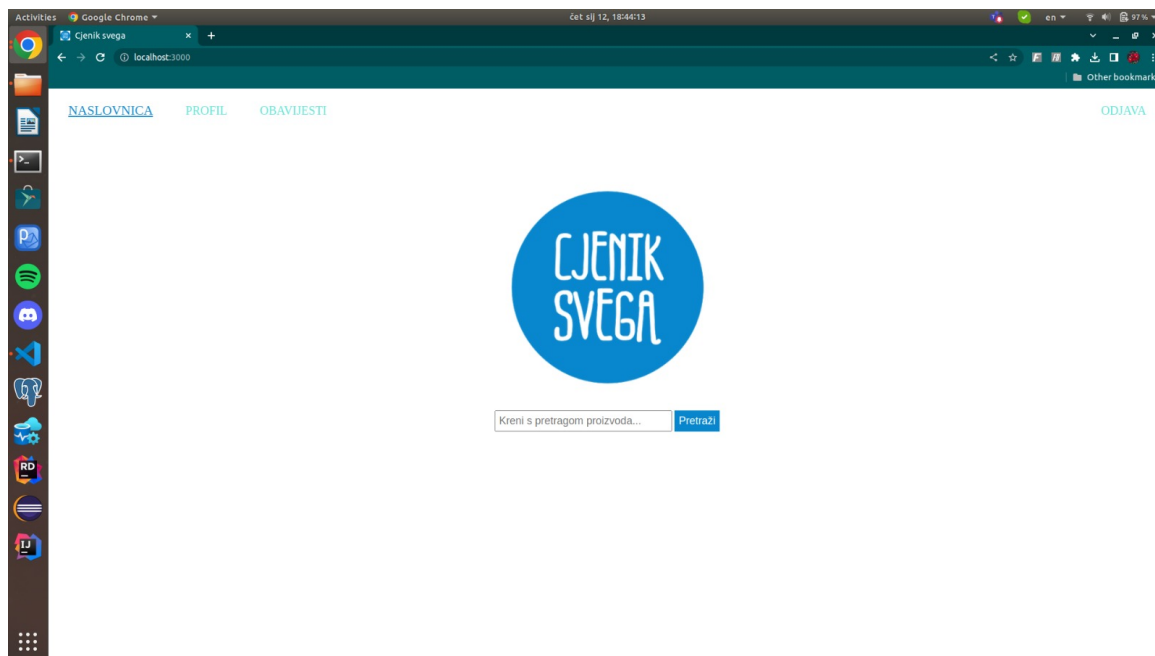
Svi testovi su prolazni i očekivano ponašanje aplikacije je postignuto.

```
Test Suites: 2 passed, 2 total
Tests: 6 passed, 6 total
Snapshots: 0 total
Time: 0.949 s, estimated 1 s
Ran all test suites.
Jest did not exit one second after the test run has completed.
```

Slika 5.3: Rezultati testova

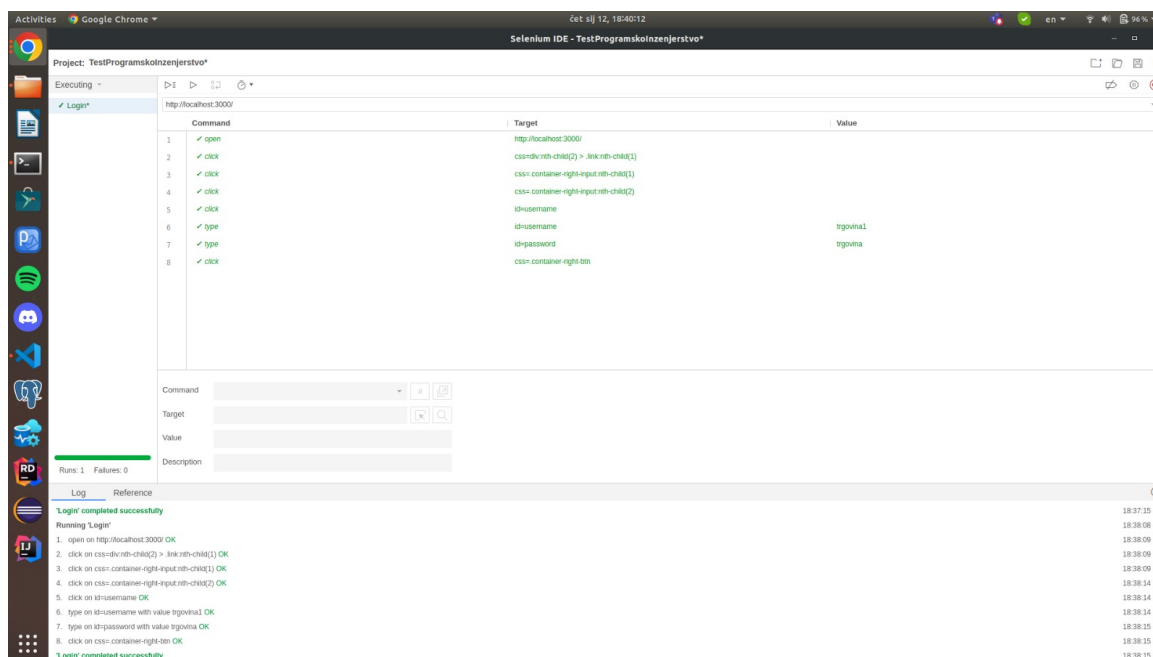
## 5.2.2 Ispitivanje sustava

Prvi *Selenium* test provodio se za funkcionalnosti **login**.



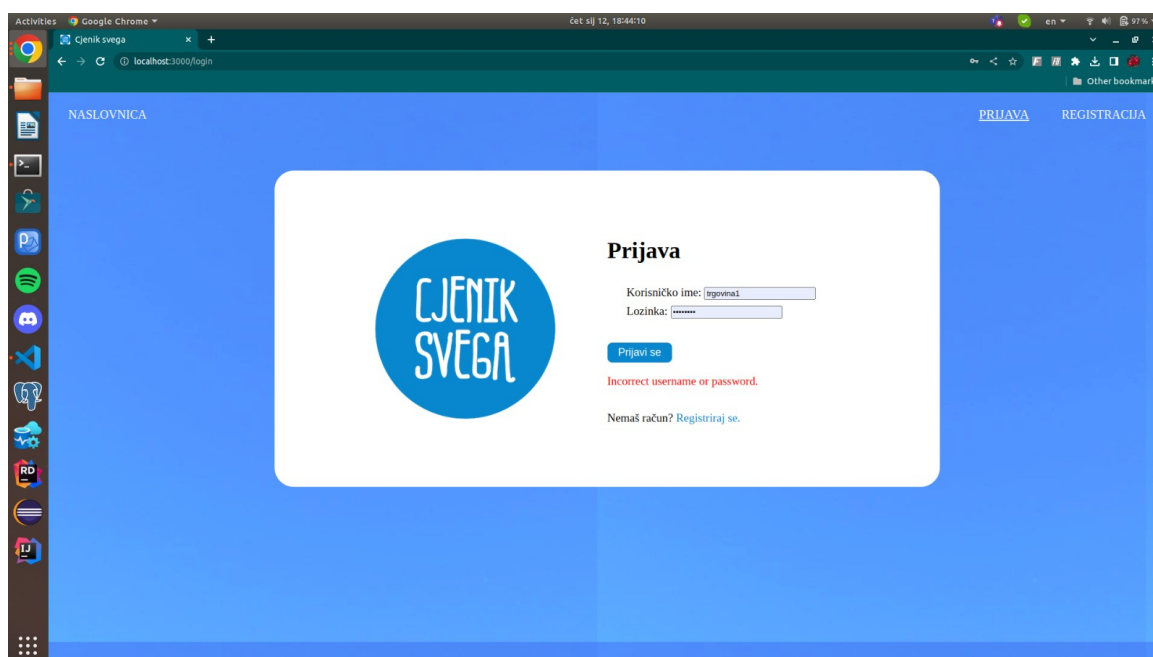
Slika 5.4: Selenium test 1, izgled sučelja

Test je uspješan.

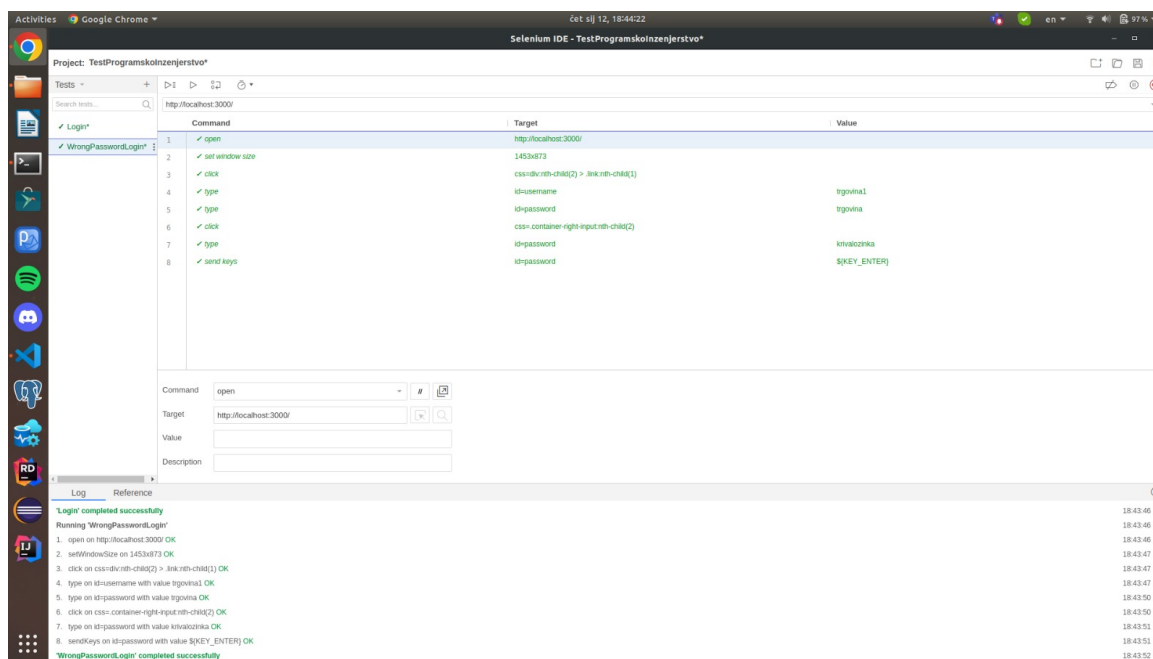


Slika 5.5: Selenium test 1, popis koraka

Drugi *Selenium* test provodio se za funkcionalnosti logina s krivom lozinkom.

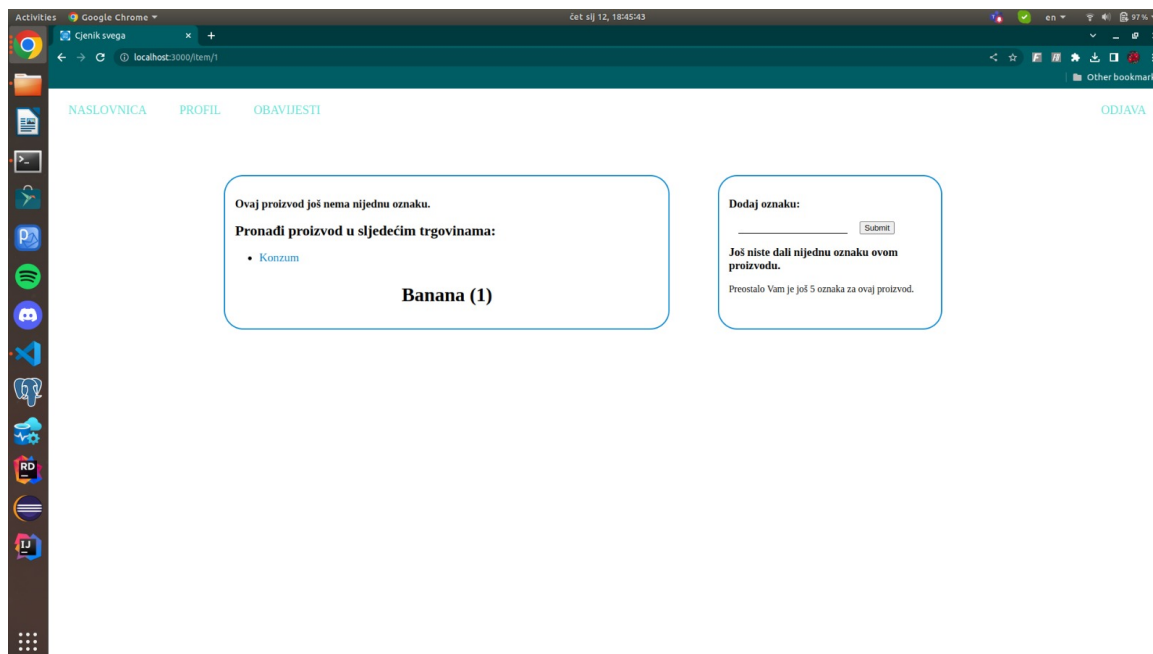


Slika 5.6: Selenium test 2, izgled sučelja  
Dobili smo očekivani rezultat.



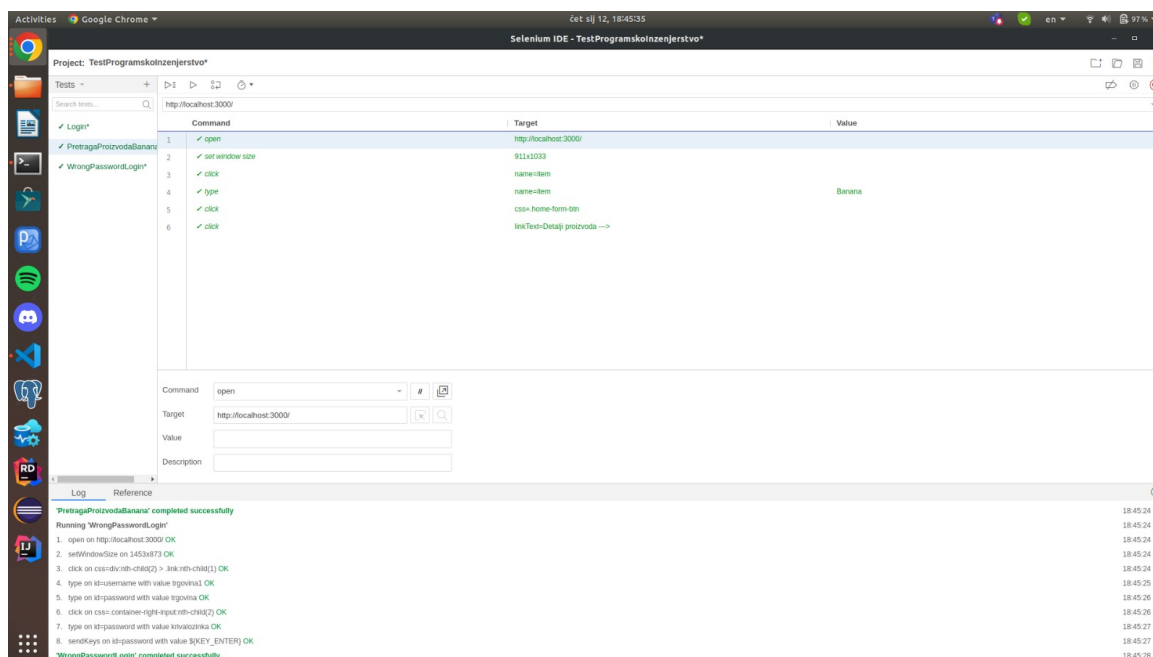
Slika 5.7: Selenium test 2, popis koraka

Treći *Selenium* test provodio se za funkcionalnosti pretraživanja proizvoda. Konkretno u testu pretraživali smo proizvod "banana".



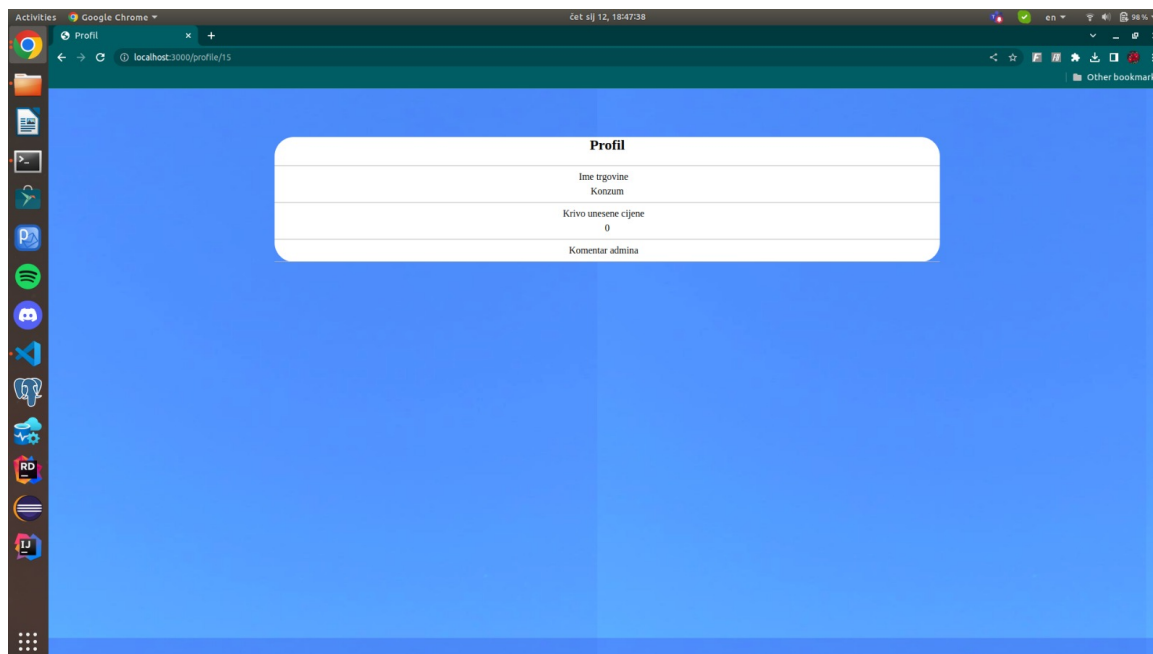
Slika 5.8: Selenium test 3, izgled sučelja

Test je uspješan.

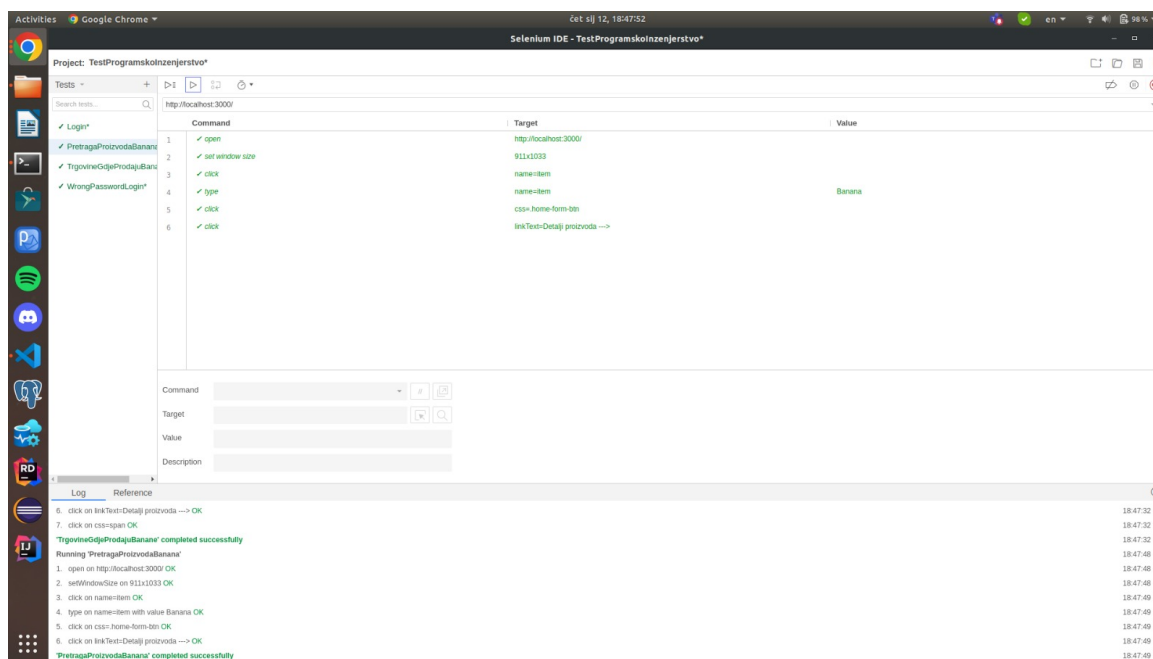


Slika 5.9: Selenium test 3, popis koraka

Četvrti *Selenium* test provodio se za funkcionalnosti pretrage trgovina. Konkretno testirali smo pretraživanje trgovina koje prodaju banane.



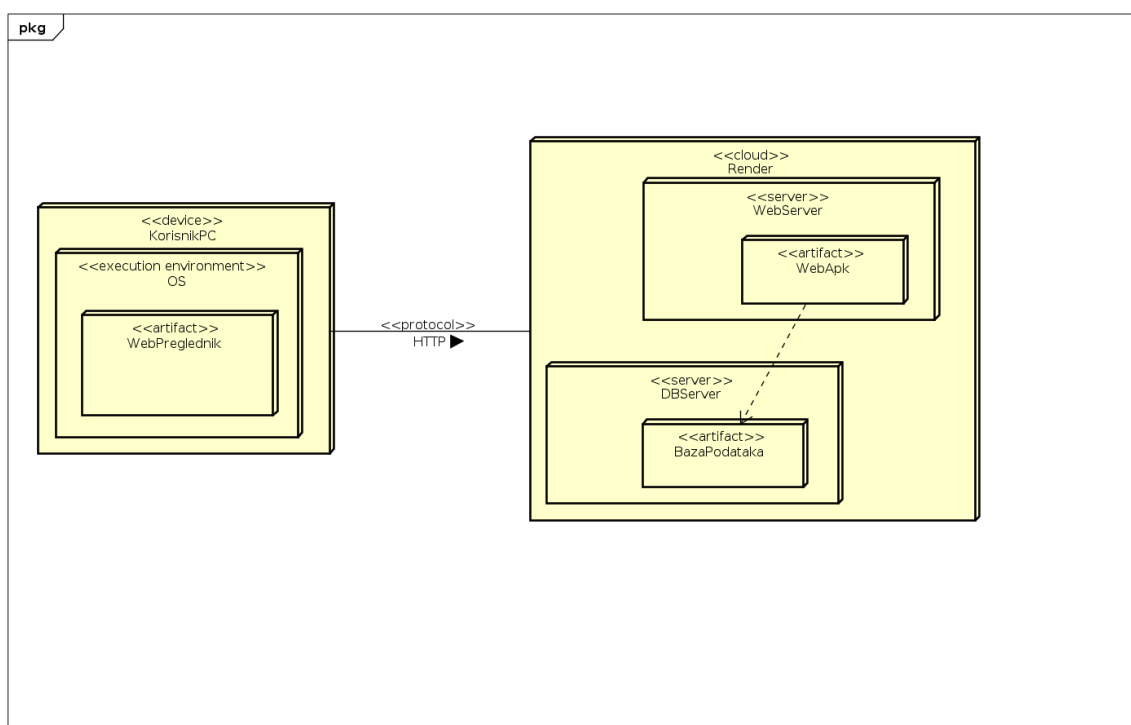
Slika 5.10: Selenium test 4, izgled sučelja  
Također, i ovaj test je uspješan.



Slika 5.11: Selenium test 4, popis koraka

## 5.3 Dijagram razmještaja

Na oblaku (eng. *cloud*) pokrenuti su poslužitelji (eng. *servers*) web-stranice i baze podataka koji međusobno komuniciraju, odnosno web-aplikacija sprema podatke u bazu. Korisnik sa svog računala preko web-preglednika pomoću HTTP protokola pristupa aplikaciji.

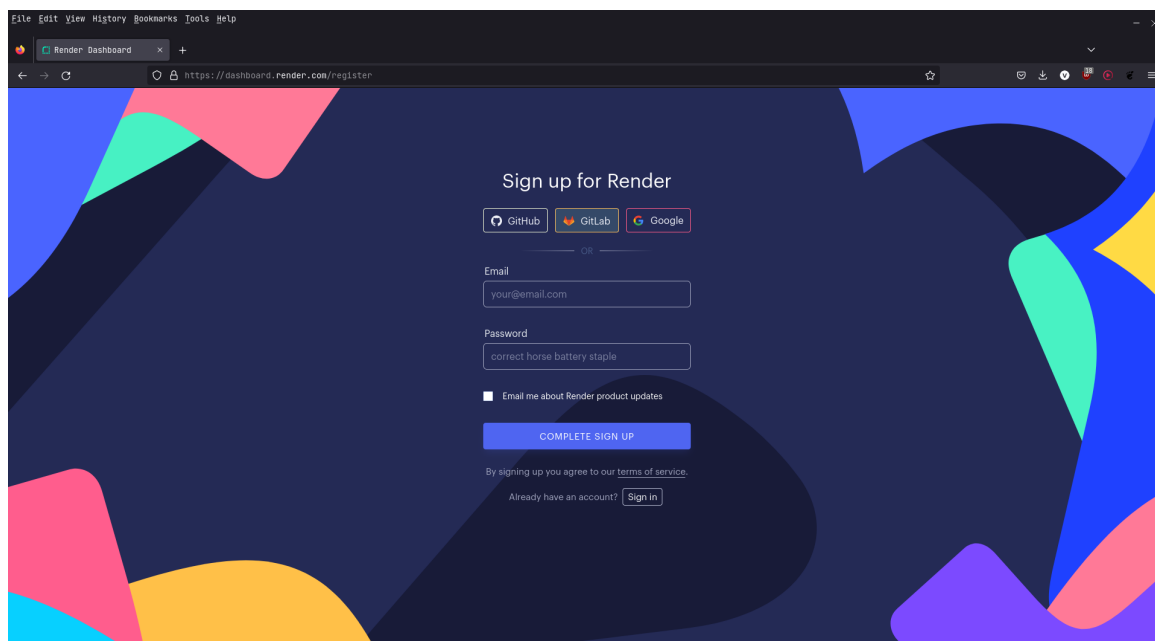


Slika 5.12: dijagram razmještaja

## 5.4 Upute za puštanje u pogon

### Prijava na sustav Render

Web-aplikaciju je potrebno ugostiti (eng. *to host*) na server kako bi bila dostupna svima. Kao servis za to odabran je **Render**. Za početak, potrebno je napraviti profil na stranici Render<sup>15</sup>.



Slika 5.13: forma za registraciju na Render

### Konfiguracija baze podataka

Kada smo se registrirali, potrebno je konfigurirati bazu podataka. Na *Dashboardu* odabrati opcije **New** pa **New PostgreSQL**. Formu ispunimo na proizvoljan način, verziju ostavimo na 15. U ovom koraku bitno je zapisati si nazive u poljima **Database** i **User**.

---

<sup>15</sup><https://render.com/>



**New PostgreSQL**

Name

Database

User

Region The region where your Database runs.

PostgreSQL Version

Datalog API Key

Please enter your payment information to select an instance type with higher limits.

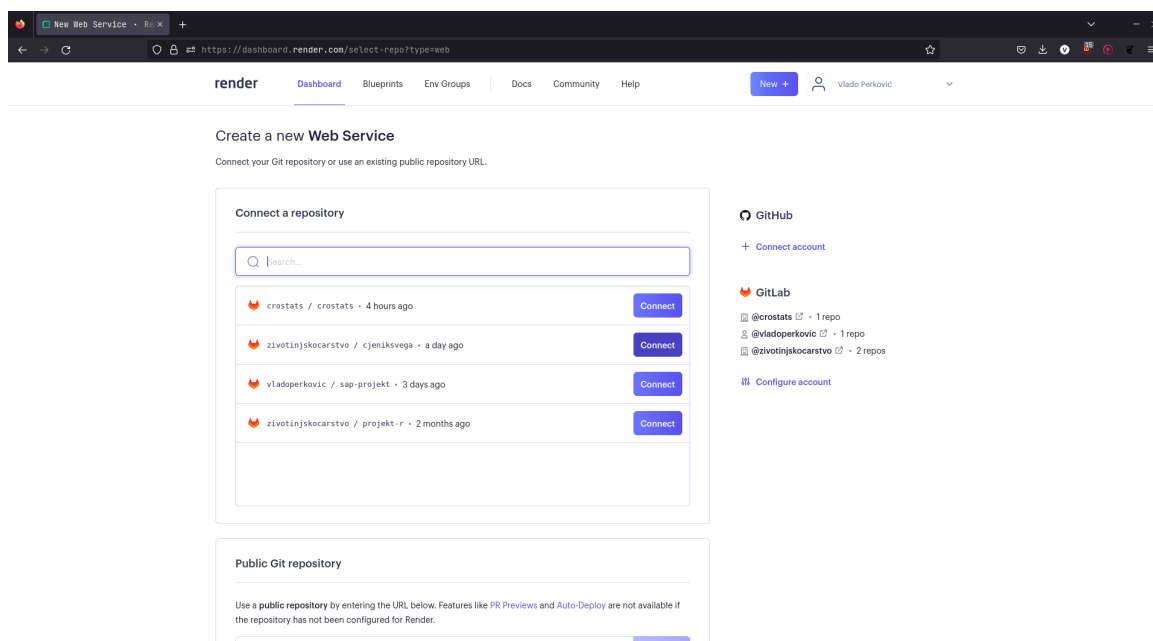
Instance Type	RAM	CPU	Storage	Price
<input checked="" type="radio"/> Free	256 MB	Shared	1 GB	\$0 / month
<input type="radio"/> Starter	256 MB	Shared	1 GB	\$7 / month
<input type="radio"/> Standard	1 GB	1 CPU	16 GB	\$20 / month
<input type="radio"/> Pro	4 GB	2 CPU	96 GB	\$95 / month
<input type="radio"/> Pro Plus	8 GB	4 CPU	256 GB	\$185 / month

Slika 5.14: forma za konfiguraciju baze podataka

Nastavimo dalje pritiskom na **Create database**. Pričekamo da se baza napravi te pronademo ju na *Dashboardu* ukoliko se nije automatski otvorilo. Zapišemo si sa strane vrijednosti iz polja **Hostname**, **Port** i **Password**.

### Konfiguracija web-servisa

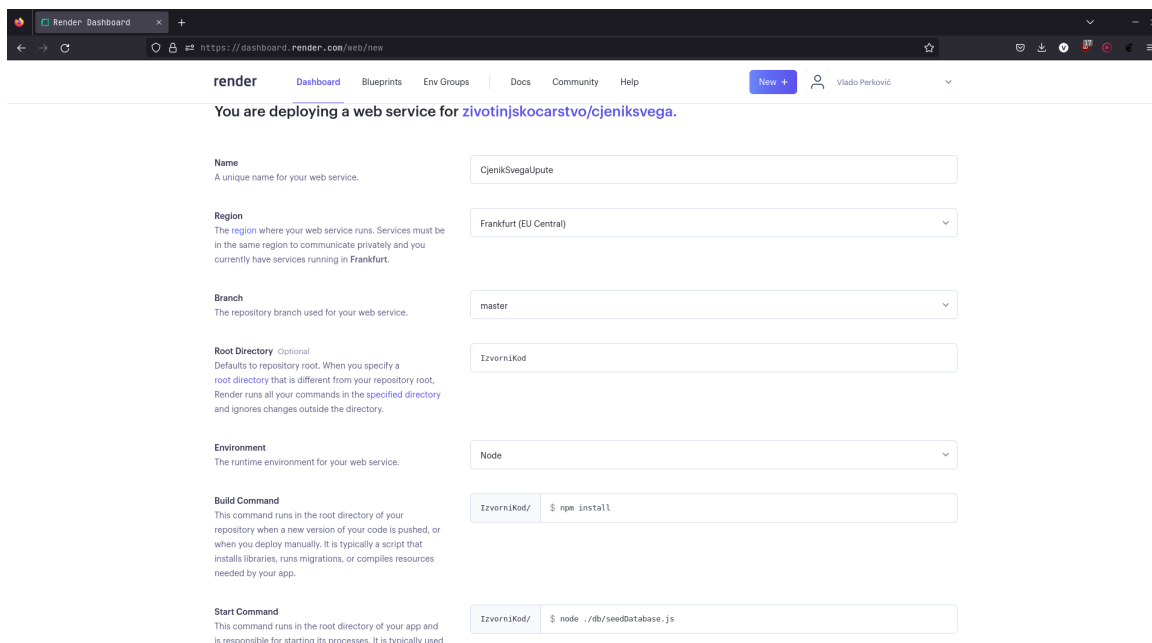
Nakon uspješne konfiguracije baze podataka, potrebno je konfigurirati web-servis. Klikom na **New** pa **Web Service** otvaramo novo sučelje. Ako je repozitorij javan, pri dnu stranice potrebno je upisati poveznicu na Gitlab repozitorij i pritisnuti **Continue**. U slučaju da je repozitorij privatan, potrebno je spojiti git profil koji ima pristup projektu, nakon čega će biti ponuđen repozitorij CjenikSvega te potrebno je pritisnuti na **Connect**.



Slika 5.15: spajanje na Gitlab repozitorij

U narednoj formi potrebno je navesti:

- Name: CjenikSvega (proizvoljno jedinstveno ime)
- Region: Frankfurt (preferira se najbliža)
- Branch: master
- Root Directory: IzvorniKod
- Environment: Node
- Build Command: `$ npm install`
- Start Command: `$ node ./db/seedDatabase.js`

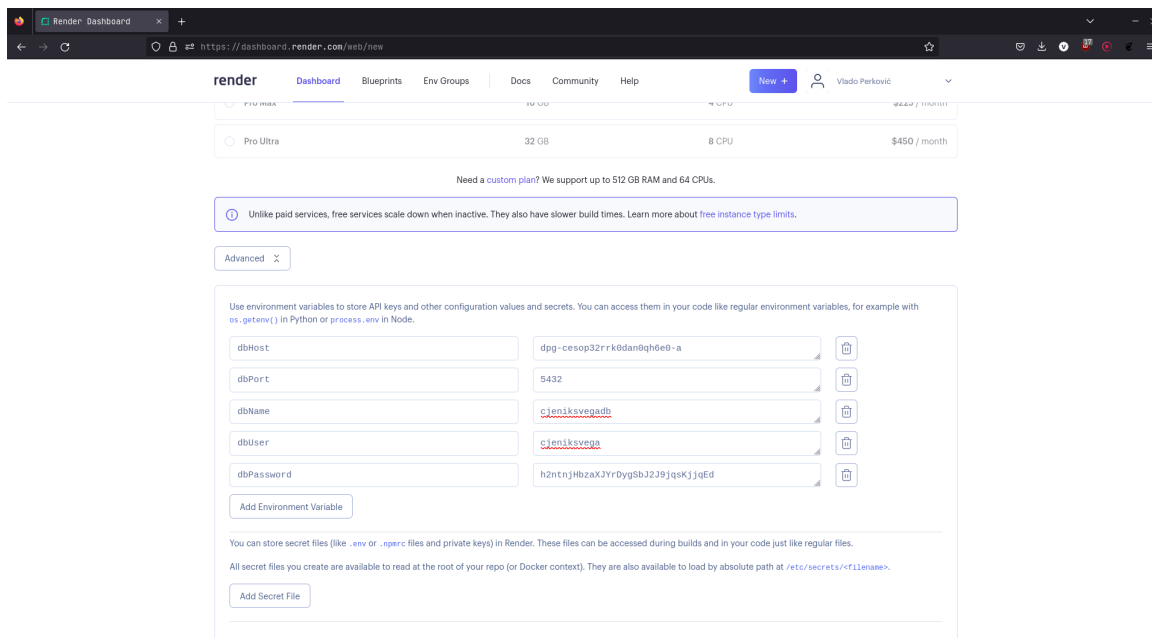


The screenshot shows the 'New' form in the Render Dashboard. The form is titled 'You are deploying a web service for zivotinjskocarstvo/cjeniksvega.' and contains several fields:

- Name:** CjenikSvegaUpute
- Region:** Frankfurt (EU Central)
- Branch:** master
- Root Directory:** IzvorniKod
- Environment:** Node
- Build Command:** IzvorniKod/ \$ npm install
- Start Command:** IzvorniKod/ \$ node ./db/seedDatabase.js

Slika 5.16: ispunjavanje forme za web-servis

Potrebno je pritisnuti na **Advanced** i dodati environment varijable pritiskom na **Add Environment Variable**: dbHost, dbPort, dbName, dbUser i dbPassword s vrijednostima koje smo prije zapisali.



The screenshot shows the 'Advanced' settings in the Render Dashboard. The 'Advanced' tab is selected, and the 'Environment Variables' section is expanded. The following variables are listed:

Variable	Value
dbHost	dpg-cesop32rrk8dan0qh6e0-a
dbPort	5432
dbName	cjeniksvegaadb
dbUser	cjeniksvega
dbPassword	h2ntnJHbzaX3YrDyGsb32J9jqSkJjQEd

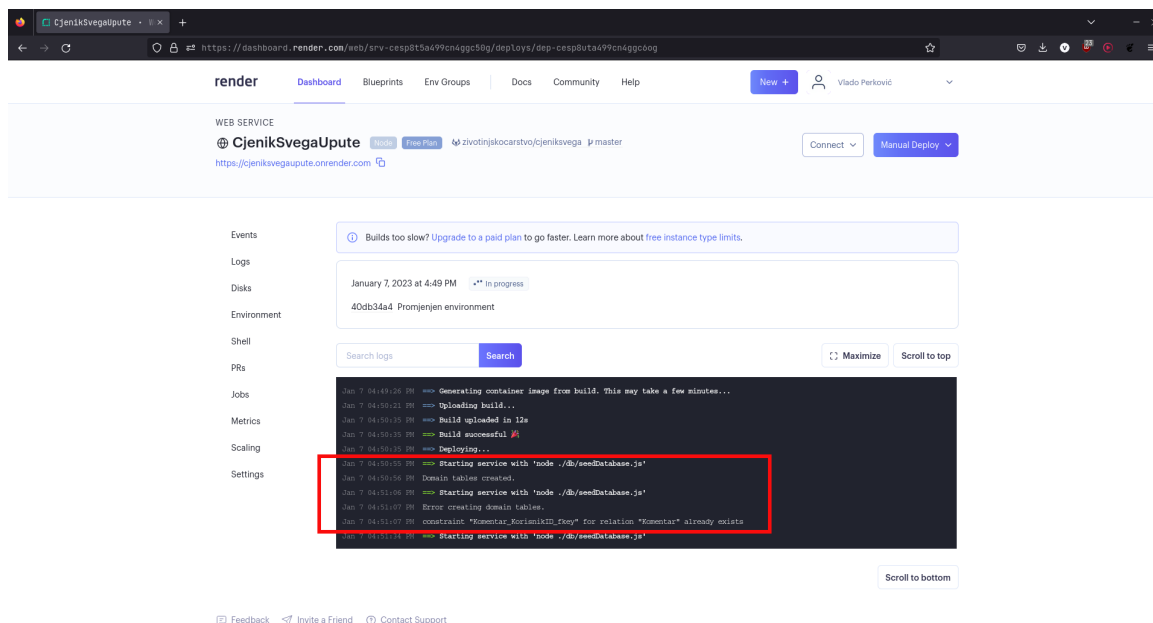
Below the variables, there is a section for 'Secret Files' with an 'Add Secret File' button.

Slika 5.17: Podešavanje *environment* varijabli

Na kraju pritisnemo **Create Web Service**.

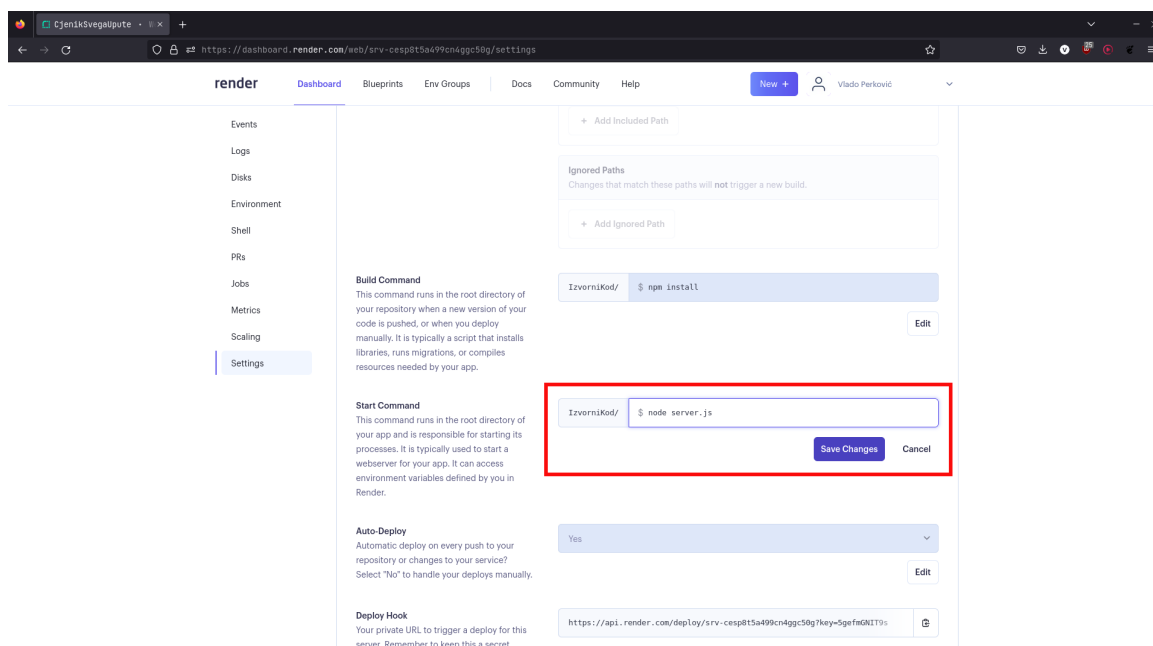
## Puštanje u pogon

Sada je potrebno pričekati da se baza kreira. Ako se nije automatski otvorio terminal, potrebno je pritisnuti na **Event** pa **Deploy** kako bismo mogli pratiti promjene. Kada uđe u *loop* znamo da je baza kreirana.

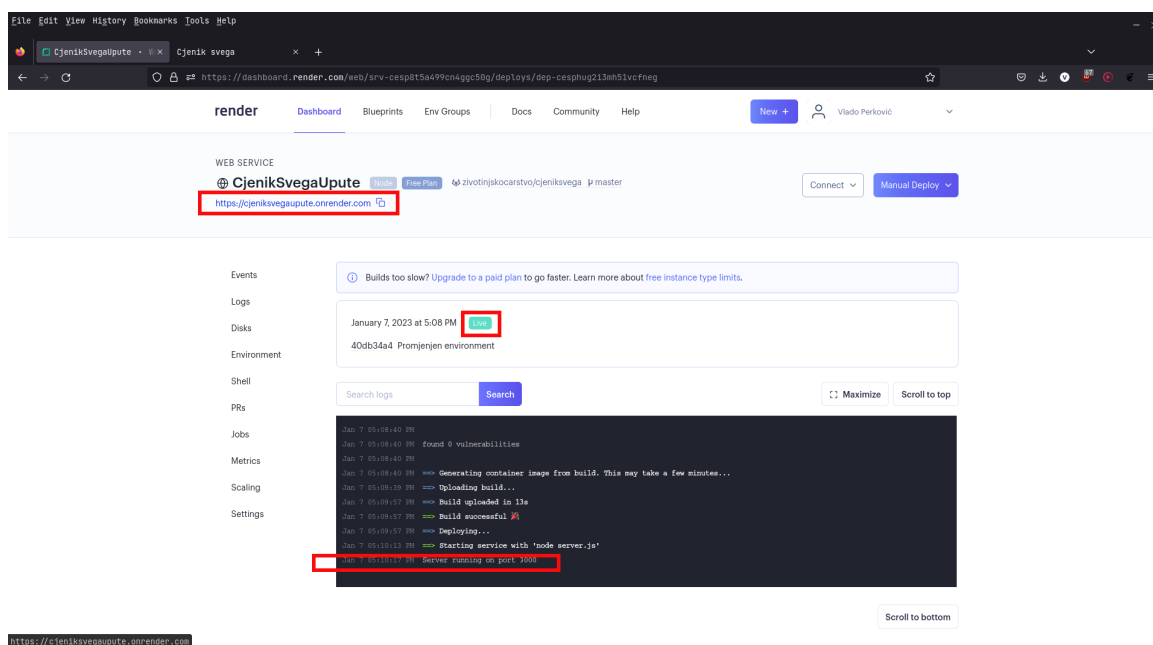


Slika 5.18: Kreiranje baze

Zatim je potrebno pritisnuti na **Settings** i promijeniti (eng. *edit*) **Start Command** u **\$ npm server.js**. Kada promijenimo potrebno je pohraniti promjene (eng. *Save Changes*) i vratiti se na **Event**.

Slika 5.19: Promjena *Start Command*

Nakon nekoliko minuta web-aplikacija bi trebala biti dostupna na linku u gornjem lijevom kutu.



Slika 5.20: Uspješno puštanje u pogon

## 6. Zaključak i budući rad

Zadatak projekta "CjenikSvega" je bio razviti web-aplikaciju koja omogućuje korisnicima da pronađu cijene proizvoda u različitim trgovinama te da slijede i ažuriranje cijena putem slanja slika. Okupljanje tima i dodjela projektnog zadatka omogućili su da se postavi jasna vizija projekta te da se svi tim članovi usaglasu u pogledu ciljeva i zahtjeva projekta. Dokumentiranje zahtjeva je također važno, jer je to osnova za daljnji rad na projektu te omogućava da se razumiju zahtjevi korisnika i kako se aplikacija treba ponašati. Tijekom dokumentiranja, tim se susreo sa izradom UML dijagrama obrazaca uporabe, sekvencijskih dijagrama, dijagrama razreda, dijagrama aktivnosti, komponenti, razmještaja i stanja.

U procesu rada na ovom projektu te ponajviše u drugom ciklusu, tim je naučio korištenje Express.js framework-a rame uz ramena sa drugim tehnologijama kao što su JavaScript, HTML i CSS te se susreo i riješio problem s radom s bazama podataka. Iako nam je tehnologija bila poznata od prije, implementacija nije došla sama od sebe pa smo tako morali uložiti podosta vremena na svladavanje tehnologije. Glavni izazov je bio upravo rad u timu, kako rasporediti posao tako da napredak ide nesmetano. Iskustvo igra veliku ulogu pa kako je ovo bio među prvim projektima članova ovog tima, sigurno je da bi idući projekt puno brže tekao.

Neke od nadogradnji na sustav bilo bi svakako izrada mobilne aplikacije da se olakša slanje slika. Korištenjem tehnologije OCR za automatizirano prepoznavanje cijena na slikama pomoglo bi se administratoru pri validaciji zahtjeva promjene cijene.

Zaključno, ova web-aplikacija ima potencijal da bude koristan alat za potrošače i vlasnike trgovina. Omogućujući korisnicima da jednostavno prijave netočne cijene putem kamere pametnog telefona, aplikacija može pomoći osigurati da su cijene u trgovinama točne i ažurne. Osim toga, mogućnost administratora trgovine da brzo ispravi sve netočnosti može pomoći u izgradnji povjerenja kod kupaca i održavanju poštenih cijena. Iako je ovaj projekt bio samo prototip, on pokazuje potencijal za sveobuhvatniju verziju koja bi se mogla implementirati u budućnosti.

# Popis literature

## *Kontinuirano osvježavanje*

*Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.*

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>
7. Lucid.app, <https://lucid.app/>
8. Pgadmin, <https://www.pgadmin.org/>
9. LaTeX, <https://www.latex-project.org/>
10. TexMaker, <https://www.xmlmath.net/texmaker/>
11. WhatsApp, <https://www.whatsapp.com/>
12. Git, <https://www.git-scm.com/>
13. GitLab, <https://www.gitlab.com/>
14. Express.js, <https://www.expressjs.com/>
15. Node.js, <https://www.nodejs.org/en/>
16. Javascript, <https://www.javascript.com/>
17. Visual Studio Code, <https://www.visualstudio.com/>

18. PostgreSQL, <https://www.postgresql.org/>

19. Render, <https://www.render.com/>



# Indeks slika i dijagrama

2.1	Listonic . . . . .	8
2.2	Basket . . . . .	8
3.1	Use case dijagram - Korisnik . . . . .	19
3.2	Use case dijagram - Trgovina . . . . .	20
3.3	UC2 - sekvencijski dijagram . . . . .	21
3.4	UC7, UC8 - sekvencijski dijagram . . . . .	22
3.5	UC14 - sekvencijski dijagram . . . . .	23
4.1	grafički prikaz arhitekture . . . . .	25
4.2	E-R Dijagram baze podataka . . . . .	32
4.3	dijagram razreda . . . . .	34
4.4	dijagram stanja . . . . .	35
4.5	dijagram aktivnosti . . . . .	36
4.6	dijagram komponenti . . . . .	37
5.1	testovi 1, 2 i 3 . . . . .	40
5.2	testovi 4, 5 i 6 . . . . .	40
5.3	Rezultati testova . . . . .	41
5.4	Selenium test 1, izgled sučelja . . . . .	41
5.5	Selenium test 1, popis koraka . . . . .	42
5.6	Selenium test 2, izgled sučelja . . . . .	42
5.7	Selenium test 2, popis koraka . . . . .	43
5.8	Selenium test 3, izgled sučelja . . . . .	43
5.9	Selenium test 3, popis koraka . . . . .	44
5.10	Selenium test 4, izgled sučelja . . . . .	44
5.11	Selenium test 4, popis koraka . . . . .	45
5.12	dijagram razmještaja . . . . .	46
5.13	forma za registraciju na Render . . . . .	47
5.14	forma za konfiguraciju baze podataka . . . . .	48
5.15	spajanje na Gitlab repozitorij . . . . .	49

5.16 ispunjavanje forme za web-servis . . . . .	50
5.17 Podešavanje <i>environment</i> varijabli . . . . .	50
5.18 Kreiranje baze . . . . .	51
5.19 Promjena <i>Start Command</i> . . . . .	52
5.20 Uspješno puštanje u pogon . . . . .	52
6.1 Prikaz aktivnosti na repozitoriju . . . . .	62

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: u ovom formatu: 20.10.2022.
- Prisustvovali: J.Vrsalović, V.Krušić, A.Macan, P.Novak, V.Perković, L.Vrsalović, L.Zekan
- Teme sastanka:
  - ideje o dodijeljenoj temi
  - raspodjela uloga u timu

### 2. sastanak

- Datum: u ovom formatu: 26.10.2022.
- Prisustvovali: L.Vrsalović, V.Perković
- Teme sastanka:
  - Podjela posla u dokumentaciji
  - Instalacija i pokretanje alata

### 3. sastanak

- Datum: u ovom formatu: 2.11.2022.
- Prisustvovali: J.Vrsalović, Petar Novak, Anton Macan
- Teme sastanka:
  - Back end
  - Koncept baze podataka

### 4. sastanak

- Datum: u ovom formatu: 3.11.2022.
- Prisustvovali: J.Vrsalović, V.Krušić, L.Zekan
- Teme sastanka:
  - Login forma
  - Homepage

### 5. sastanak

- Datum: u ovom formatu: 5.12.2022.

- Prisustvovali: A. Macan, P. Novak

- Teme sastanka:

- Usklađivanje baze podataka
- Planovi za dalje

#### 6. sastanak

- Datum: u ovom formatu: 9.12.2022.

- Prisustvovali: J.Vrsalović, V.Krušić, A.Macan, P.Novak, V.Perković, L.Vrsalović, L.Zekan

- Teme sastanka:

- Tražilica
- Plan za nastavak rada

#### 7. sastanak

- Datum: u ovom formatu: 15.12.2022.

- Prisustvovali: J.Vrsalović, L.Zekan

- Teme sastanka:

- Refaktoriranje frontenda
- razni bugovi

#### 8. sastanak

- Datum: u ovom formatu: 1.1.2023.

- Prisustvovali: J.Vrsalović, L.Zekan, V. Krušić

- Teme sastanka:

- Frontend
- Oznake na proizvodima

#### 9. sastanak

- Datum: u ovom formatu: 1.1.2023.

- Prisustvovali: L.Vrsalović, V.Perković

- Teme sastanka:

- Raspodjela posla dokumentacije

#### 10. sastanak

- Datum: u ovom formatu: 5.1.2023.

- Prisustvovali: J.Vrsalović, L.Zekan, P.Novak, A.Macan

- Teme sastanka:

- Funkcionalnost obavijesti
- Testiranje softvera

## Tablica aktivnosti

	Joško Vrsalović	Vida Krušić	Anton Macan	Petar Novak	Vlado Perković	Lovro Vrsalović	Luka Zekan
Upravljanje projektom	20						
Opis projektnog zadatka					4		
Funkcionalni zahtjevi						1	
Opis pojedinih obrazaca					1	4	
Dijagram obrazaca						3	
Sekvencijski dijagrami					2		
Opis ostalih zahtjeva						1	
Arhitektura i dizajn sustava							3
Baza podataka			4				
Dijagram razreda				5			
Dijagram stanja						4	
Dijagram aktivnosti						3	
Dijagram komponenti					4		
Korištene tehnologije i alati					2		
Ispitivanje programskog rješenja			3		1		
Dijagram razmještaja					2		
Upute za puštanje u pogon	1				3		
Dnevnik sastajanja					2		
Zaključak i budući rad					2		
Popis literature					1		

Nastavljeno na idućoj stranici

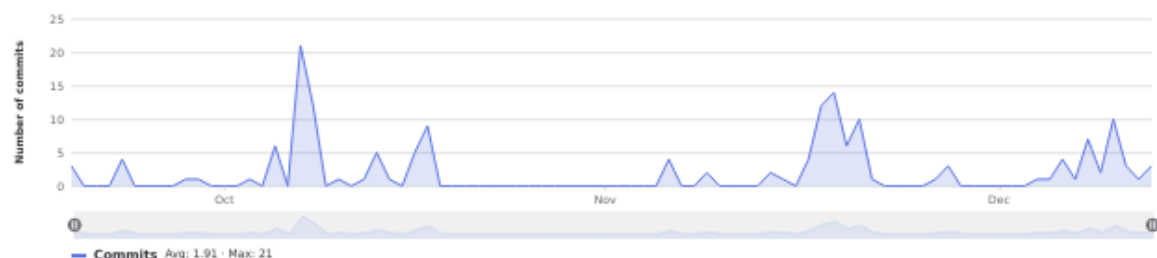
Nastavljeno od prethodne stranice

	Joško Vrsalović	Vida Krušić	Anton Macan	Petar Novak	Vlado Perković	Lovro Vrsalović	Luka Zekan
<i>homepage</i>		4					1
<i>forme za login i signup</i>		2					
<i>izrada baze podataka</i>			4				
<i>spajanje s bazom podataka</i>				4			
<i>back end</i>	2			5			
<i>deployment</i>	7						
<i>frontend</i>		4		1			22
<i>backend</i>	17		7	21			4

## Dijagrami pregleda promjena

### Commits to master

Excluding merge commits. Limited to 6,000 commits.



### Jay2999

49 commits (jasko.vrsalovic@fer.hr)



### Vlado Perkovic

36 commits (vlado.perkovic@fer.hr)



### Izekan

28 commits (iz53052@fer.hr)



### NovakPetar

22 commits (petar.novak@fer.hr)



### vida krusic

20 commits (vida.krusic@fer.hr)



### [Anton Macan]

6 commits (anti.macan88@gmail.com)



### Vrsa2936

2 commits (levro.vrsalovic@fer.hr)



### Anton Macan

1 commit (anton.macan@fer.hr)



Slika 6.1: Prikaz aktivnosti na repozitoriju