

Programsko inženjerstvo

Ak. god. 2020./2021.

SmartCart

Dokumentacija, Rev. 2

Grupa: *PreljevStoga*

Voditelj: *Tomislav Bjelčić*

Datum predaje: *14. siječnja 2021.*

Nastavnik: *dr. sc. Miljenko Krhen*

Sadržaj

1 Dnevnik promjena dokumentacije	2
2 Opis projektnog zadatka	3
2.1 Motivacija	3
2.2 Opis rada sustava	3
2.3 Mogućnosti proširenja	6
2.4 Slične aplikacije	7
2.4.1 SmartCart: Shopping list	7
2.4.2 nabava.net	7
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	12
3.1.2 Sekvencijski dijagrami	28
3.2 Ostali zahtjevi	31
4 Arhitektura i dizajn sustava	32
4.1 Baze podataka	33
4.1.1 Opis tablica na uređaju	34
4.1.2 Opis tablica na poslužitelju	35
4.1.3 Dijagrami baze podataka	39
4.2 Dijagrami razreda	40
4.3 Dijagram stanja	41
4.4 Dijagram aktivnosti	43
4.5 Dijagram komponenti	44
5 Implementacija i korisničko sučelje	46
5.1 Korištene tehnologije i alati	46
5.2 Ispitivanje programskog rješenja	48
5.2.1 Ispitivanje komponenti	48
5.2.2 Ispitivanje sustava	49

5.3	Dijagram razmještaja	52
5.4	Upute za puštanje u pogon	53
6	Zaključak i budući rad	55
	Popis literature	56
	Indeks slika i dijagrama	57
	Dodatak: Prikaz aktivnosti grupe	58

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Dodan opis zadatka i dio specifikacije.	Lakoš	27.10.2020.
0.2	Prilagođeno i prebačeno u \LaTeX	Grubelić	4.11.2020.
0.3	Dodani arhitektura i dodatak	Grubelić	8.11.2020.
0.4	Dodan dijagram razreda	Vladić, Pranjić	12.11.2020.
1.0	Dovršeno za predaju \LaTeX	Grubelić	12.11.2020.
1.1	Dodana struktura za 2. reviziju i ispitivanje programskog rješenja	Komljenović	14.1.2021.
1.2	Dodani dijagrami za 2. predaju	Lakoš	14.1.2021.
1.3	Dodan zaključak	Pranjić	14.1.2021.
2.0	Dovršena dokumentacija	Brdar, Grubelić, Lakoš	14.1.2021.

2. Opis projektnog zadatka

2.1 Motivacija

Cilj je razviti mobilnu aplikaciju koja će pomoći pri obavljanju kupovine u dućanima. Prije odlaska u trgovinu ljudi često pišu popis namirnica koje trebaju kupiti. Pri tom im je nerijetko bitan ukupan iznos kupovine, a posebno ima li željenog proizvoda u trgovini u koju namjeravaju ići. Aplikacija „SmartCart” će omogućiti kupcima da sastave popis namirnica za kupovinu i izračunati njihovu cijenu na temelju podataka o cijenama koje u aplikaciju mogu unijeti trgovci. Popis će imati mogućnost pretvaranja u košaricu tijekom kupnje kad se namirnice uzimaju s polica, tj. praćenja cijene uzetih namirnica, za one koji se ne drže popisa u potpunosti. To će također ubrzati proces kupovine namirnica pa će ljudi manje vremena provoditi u trgovini i bit će manje gužve. Osim toga, trgovcima će to biti odlična promocija jer će oni kupci kojima je svejedno u koju trgovinu idu, moći vidjeti onu s najmanjom cijenom namirnica u blizini.

2.2 Opis rada sustava

Budući da je srž aplikacije mogućnost stvaranja popisa, prikaz popisa se otvara i prilikom pokretanja aplikacije. Na popise se mogu dodavati artikli na četiri načina:

- Skeniranjem barkoda artikla - dodaje se točno određeni artikl zadan barkodom i na popis i u košaricu
- Pretragom i odabirom - dodaje se određeni artikl na popis pretragom imena (koja dopušta manje pogreške prilikom unosa)
- Zadavanjem filtra - na popisu se nalazi onaj artikl koji najbolje zadovoljava filter i redukcijsku funkciju (npr. najmanja cijena) u trenutku zadnjeg osvježavanja. Količina dostupnih filtra i redukcijskih funkcija će se mijenjati tijekom razvoja aplikacije. Primjeri funkcija filtra su "Cijena unutar intervala", "Vrsta proizvoda" i "Proizvedeno u", a primjeri redukcijskih funkcija su "Najmanja cijena" i "Najveća cijena po kilogramu"
- Dodavanjem iz popisa omiljenih artikala

Dodanim proizvodima moguće je mijenjati količinu, pregledati informacije ili ih dodati u popis omiljenih artikala, na popisu označenih zvjezdicom, koji se nalazi u bočnoj traci. Svaki se popis sastoji od artikala, njihovih količina i cijena (svaka cijena je povezana s nekom trgovinom). Artikle možemo dodati više puta na isti popis. Korisnik artikle na popisu može označiti dodanima u košaricu. Ukupna cijena košarice, kao i cijena cijelog popisa su prikazane uz popis. Moguće je imati više popisa. Moguća je potpuna manipulacija popisima – stvaranje, brisanje, kopiranje, zatvaranje i otvaranje. Popisi su pohranjeni lokalno na korisničkom uređaju.

Budući da svaki trgovac zadaje svoju cijenu za artikle na popisu, postoje tri načina izračuna cijena:

- Najjeftinija trgovina - korisnik prepušta sustavu da nađe najbližu trgovinu za koju je zbroj cijena svih artikala na popisu najmanji.
- Najjeftiniji artikli - korisnik prepušta sustavu da za svaki artikl nađe najbližu trgovinu u kojoj je cijena tog artikla najmanja
- Određena trgovina - korisnik pretražuje trgovine i odabire jednu te se cijene osvježuje u skladu s time. Ovo je pretpostavljeni način

Budući da se informacije o trgovinama i cijenama u njima često mijenjaju, korisnik može osvježavati svoje popise. Pritom bojanje oznake količine proizvoda crveno označava da proizvod nije više dostupan u trgovini s kojom je povezan na popisu. Bojanje imena proizvoda crveno označava da proizvod nije više dostupan niti u jednoj trgovini, a bojanje cijene proizvoda crveno ili zeleno označava da se ta cijena redom povećala ili smanjila.

Informacije o trgovini moguće je pronaći pritiskom na cijenu artikla u toj trgovini. Tu je moguće pronaći radno vrijeme, adresu, poveznicu za prikaz trgovine na *Google Maps* karti te udaljenost od trgovine ako je dozvoljen pristup lokaciji.

Informacije o artiklima i dostupnosti u trgovinama ne spremaju se na uređaju, pa je internetska veza s poslužiteljem nužna za cjelovitu funkcionalnost aplikacije. Aplikacija je u "online" načinu rada ako je povezana s poslužiteljem. Prekidom "online" načina rada aplikacija prelazi u "offline" način rada. Bez veze s poslužiteljem korisnik samo može micati artikle s popisa te stvarati, brisati, otvarati i zatvarati popise bez dodavanja novih artikala. Jedino što je spremljeno u bazi podataka na uređaju su podatci o artiklima na popisima te među omiljenim artik-

lima.

Sva ova funkcionalnost dostupna je neprijavljenom korisniku, tj. gostu. No, potrebno je i nabaviti informacije koje bi se prikazivale u aplikaciji. Neki će pisati korisne informacije, ali kako bi se zaustavilo one koji bi širili dezinformacije, za upis bilo kakvih informacija potrebno je imati korisnički račun koji administrator prema potrebi može onemogućiti. Neprijavljeni korisnici ne mogu promijeniti informacije o artiklu ni samo na uređaju, kako bi se potaknulo korisnike da podijele podatke koje steknu.

Dakle, postoji tri vrste korisnika s korisničkim računom:

- Kupac
- Trgovac
- Administrator

Kupac može, uz osnovnu funkcionalnost koju ima i gost, mijenjati informacije o artiklu, ali samo one koje nisu nužno vezane uz trgovinu. Dakle, opis artikla u pravilu nije vezan uz trgovinu. Samo cijena, popust i dostupnost. Kupac o svim informacijama o artiklu i/ili trgovini može glasati, a na temelju toga se izračunava ocjena pouzdanosti koju vide svi korisnici uz opis. Također, kako bi se povećala vjerojatnost da onaj tko piše informacije o artiklu piše točne podatke, prije spremanja izmijenjenih podataka mora očitati barkod artikla. Za registraciju su mu potrebni samo email i lozinka.

Trgovac, uz funkcionalnost kupca, ima i mogućnost dodavanja svojih trgovina te manipulacije njima. Za stvaranje trgovine su mu potrebni lokacija trgovine i radno vrijeme. Trgovac kroz svoju nadzornu ploču na kojoj ima pregled trgovina koje je dodao i proizvoda u njima dodaje ili mijenja artikle kao petorku (barkod, cijena, popust, dostupnost, email adresa željenog opisivača). Barkod je potreban da bi korisnici znali da je taj proizvod u toj trgovini, cijena i popust određuju ukupnu cijenu proizvoda koja će pisati na popis, a dostupnost označava ima li artikla u trgovini bez da ga se briše. Trgovac ne mora unositi opis artikla, ali ako postavi email adresu željenog opisivača i korisnik s tom email adresom je napisao opis o artiklu s tim barkodom, taj opis nadjača ostale kad se prikazuje cijena artikla u toj trgovini. Inače se informacije o artiklu u trgovini povezuju s najpouzdanijim opisom. Nove informacije o artiklima se mogu učitati i iz csv datoteke jednakog rasporeda

stupaca. Kako bi se smanjila vjerojatnost lažnog predstavljanja i neispravnih informacija o cijenama i dostupnosti artikala u trgovini, trgovac za registraciju mora upisati tajni broj koji je dobio (uživo) od administratora i samo on smije mijenjati informacije o trgovinama. Tu je ipak novac "u igri".

Administrator ima zadaću održavati red u sustavu. Može vidjeti sve podatke (poput broja glasova na opisima trgovaca i kupaca te njihovih mail adresa, lozinke su sakrivene SHA-256 algoritmom), a mijenjati može informacije o artiklima, kao i kupac. Može onemogućiti pristup bilo kojem korisničkom računu, uključujući trgovce i druge administratore (za micanje privilegija administratoru ipak treba najmanje pola glasova administratora u istom danu). Trgovcu, iako ne može mijenjati informacije o trgovini, može ukloniti trgovinu kako bi se ipak spriječilo zlonamjerno iskorištavanje sustava. Administrator također privilegije stiče registracijom uz tajni broj koji dobije (uživo) od drugih administratora.

2.3 Mogućnosti proširenja

Iako je aplikacija u ovom obliku potpuno funkcionalna, ako preraste u pravi proizvod postoji više mogućnosti proširenja i poboljšanja korisničkog iskustva:

- Proširenje mogućnosti korištenja u načinu rada bez veze s poslužiteljem
- Slike artikala
- API za trgovce i/ili kupce
- Spremanje popisa u bazu na poslužitelju
- Dijeljenje popisa
- Praćenje cijene artikala kroz vrijeme
- Tlocrt trgovina i promjena poretka artikala na popisu
- Podržavanje popusta vezanih uz kartice vjernosti
- Posredovanje u "online" kupovini
- Suradnja sa stranicom poput Coolinarike za ubacivanje sastojaka jela na korisnički popis
- Definiranje maksimalnog broja trgovina po kojima se podijele sastojci iz košarice za izračun minimalne cijene (trenutno postoje dvije krajnosti: 1 i broj artikala)

2.4 Slične aplikacije

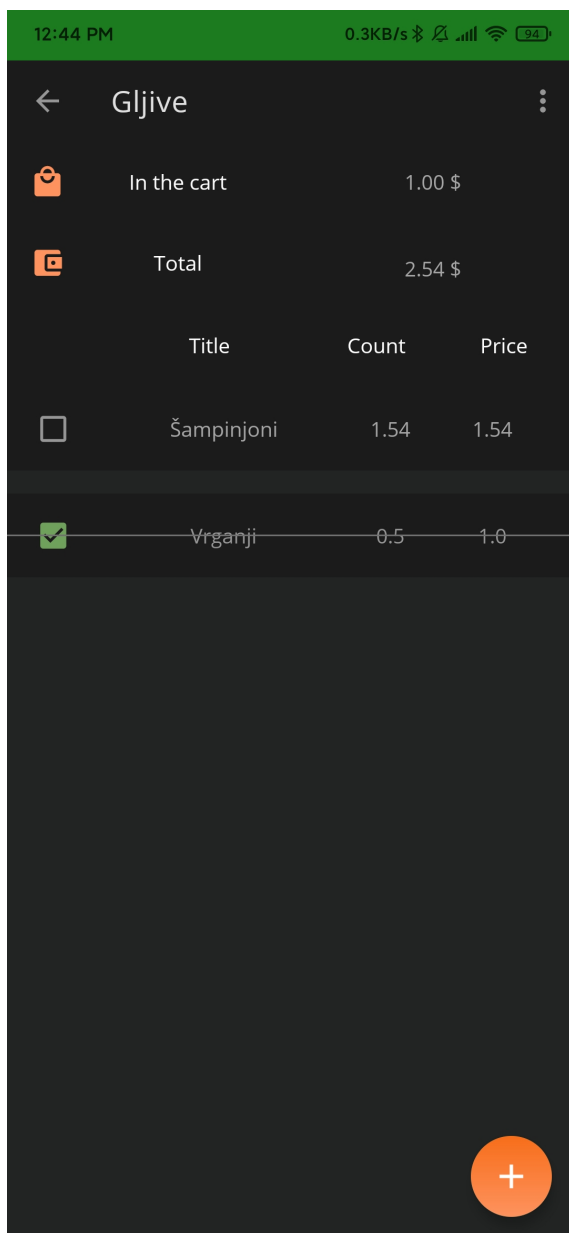
U nastavku su navedene dvije aplikacije čija je funkcionalnost slična onoj naše aplikacije. Ipak, valja naglasiti da nam one nisu ni inspiracija niti ih nastojimo imitirati. One su upravo to - aplikacije s nekim funkcionalnostima sličnima našoj.

2.4.1 SmartCart: Shopping list

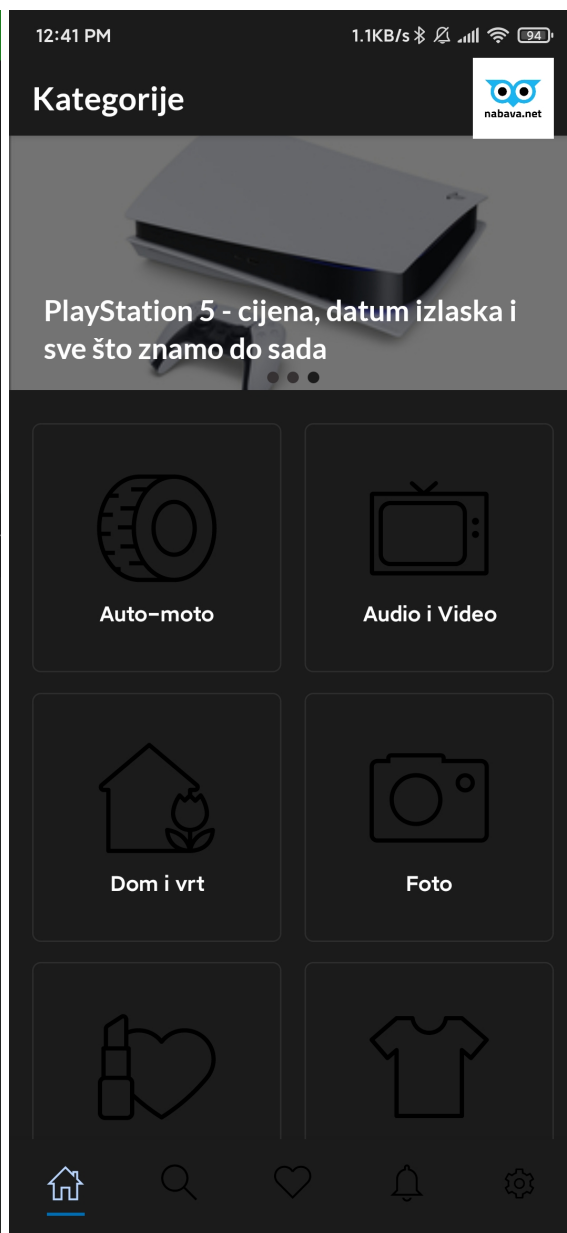
Ova aplikacija (2.1a) je slična našoj po tome što ima funkcionalnost stvaranja više popisa i manipulacije njima. Bitna razlika je to što nema "agregator cijena" nego sve cijene upisuje korisnik. Za ovu aplikaciju smo saznali tek nakon definiranja specifikacije programske potpore, nakon pretrage trgovine *Google Play* za potrebe ove dokumentacije, zato ima vrlo slično ime i neke detalje jer izgleda da slično razmišljamo.

2.4.2 nabava.net

Aplikacija nabava.net (2.1b) ima upravo suprotno - "agregator cijena", ali nema košaricu i usredotočena je na "online" kupovinu. Nije toliko usredotočena na namirnice. Mobilna aplikacija joj je zapravo prilagodba web aplikacije.



(a) SmartCart:Shopping list



(b) nabava.net

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Neprijavljeni korisnici
2. Kupci
3. Trgovci
4. Administratori (nadskup razvojnog tima)

Aktori i njihovi funkcionalni zahtjevi:

1. Neprijavljeni korisnik, tj. gost (inicijator) može:
 - (a) vidjeti trgovine u blizini (TuB) kao popis
 - (b) prikazati TuB na karti
 - (c) odabrati TuB kao željenu za prikaz dostupnosti i cijena artikala
 - (d) vidjeti za svaki artikl s popisa posebno u kojoj je TuB najjeftiniji
 - (e) vidjeti za cijelu košaricu u kojoj TuB je dostupna i najjeftinija
 - (f) upravljati popisima
 - i. stvarati nove popise
 - ii. preimenovati popise
 - iii. odabrati koji popis se prikazuje
 - iv. pretraživati artikle i dodavati ih na prikazani popis
 - v. micati artikle s prikazanog popisa
 - vi. sve artikle s prikazanog popisa kopirati na neki drugi
 - vii. obrisati prikazani popis
 - viii. označiti artikle kao dodane u košaricu
 - (g) označiti artikl kao omiljeni za lakši pristup
 - (h) vidjeti mjeru pouzdanosti informacija o artiklu
 - (i) vidjeti ukupnu cijenu popisa
 - (j) vidjeti cijenu košarice
 - i. prijaviti se u sustav

- ii. registrirati se u sustav, stvoriti korisnički račun za koji su mu potrebni e-mail adresa i lozinka ili Google račun
- iii. promijeniti lozinku

2. Kupac (inicijator) može:

- (a) sve što može i neprijavljeni korisnik osim prijave
- (b) glasati o ispravnosti informacija o artiklu
- (c) uz skeniranje artikla napisati informacije o njemu ako su trenutne neispravne ili ih nema
- (d) obrisati račun
- (e) promijeniti lozinku

3. Trgovac (inicijator) može:

- (a) sve što može i kupac, samo mu je za registraciju potreban dodatni verifikacijski kôd
- (b) upravljati svojim trgovinama
 - i. stvarati nove trgovine (potrebno navesti lokaciju i radno vrijeme)
 - ii. obrisati postojeće trgovine
 - iii. u bilo koju trgovinu dodati artikle s pripadnim cijenama
 - iv. mijenjati cijene artikala
 - v. označiti da su artikli na popustu
 - vi. označiti da je neki artikl rasprodan
 - vii. maknuti artikl iz trgovine
 - viii. mijenjati radno vrijeme trgovine

4. Administrator (inicijator) može:

- (a) sve što može i kupac, samo mu je za registraciju potreban dodatni verifikacijski kôd
- (b) pristupiti administratorskoj nadzornoj ploči preko web-sučelja
- (c) ukloniti korisnika
- (d) ukloniti trgovinu

5. Baza podataka na poslužitelju (sudionik):

- (a) pohranjuje sve podatke o sudionicima i njihovim ovlastima
- (b) pohranjuje sve podatke o artiklima
- (c) pohranjuje sve podatke o trgovinama

6. Baza podataka na uređaju (sudionik):

- (a) pohranjuje podatke o popisima i omiljenim artiklima korisnika

7. Email poslužitelj (sudionik):

- (a) šalje privremenu lozinku korisniku koji je zaboravio svoju
- (b) šalje obavijesti administratorima (o onemogućavanju pristupa jednom od njih)

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC0 - Postavljanje najveće udaljenosti

- **Glavni sudionik:** Korisnik
- **Cilj:** Ograničiti polumjer u kojem mogu biti rezultati pretrage trgovine
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Dozvoljen je pristup lokaciji
- **Opis osnovnog tijeka:**
 1. Korisnik otvara postavke aplikacije
 2. Korisnik postavlja najveću udaljenost unutar koje može biti rezultat pretrage. Udaljenost se računa kao euklidska udaljenost geografske širine i dužine
 3. Ubuduće prilikom svakog upita vezanog uz trgovine osim ručne pretrage baza prvo odstrani one koje su dalje od najveće dopuštene udaljenosti od korisnika ako je pristup lokaciji još uvijek dozvoljen
- **Opis mogućih odstupanja:**
 - 3.a Korisnik makne dozvolu pristupa lokaciji u nekom trenutku
 1. Ne primjenjuje se ograničenje na lokaciju trgovine
- **Opis mogućih odstupanja:**
 - 3.b Korisnik je na polu ili najkraći put do trgovine prelazi meridijan ± 180
 1. Korisnik potencijalno dobiva pogrešne rezultate, ali ih svi dobivaju brzo

UC1 - Stvaranje popisa

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti novi popis artikala
- **Sudionici:** Baza podataka na uređaju
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. korisnik otvori izbornik
 2. odabere opciju "Stvori novi popis"
 3. na mjesto prikazanog popisa se stavlja novi prazni popis s pretpostavljenim imenom "Novi popis"

UC2 - Otvaranje drugog popisa

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikazivanje određenog postojećeg popisa
- **Sudionici:** Baza podataka na uređaju
- **Preduvjet:** Postoji više od jednog popisa
- **Opis osnovnog tijeka:**
 1. Korisnik otvori izbornik sa svim popisima
 2. Korisnik odabire popis s izbornika
 3. Drugi popis zamijenjuje prikazani
- **Opis mogućih odstupanja:**
 - 1.a Nema drugih popisa
 1. Korisnik dobije obavijest da nema drugih popisa i ne otvara se izbornik

UC3 - Brisanje popisa

- **Glavni sudionik:** Korisnik
- **Cilj:** Obrisati popis
- **Sudionici:** Baza podataka na uređaju
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik otvara popis koji želi obrisati
 2. Korisnik pritisne gumb za brisanje
 3. Korisnik potvrdi da želi obrisati popis
 4. Popis se briše i prikazuje se drugi
- **Opis mogućih odstupanja:**
 - 4.a Ne postoji drugi popis
 1. Stvara se novi prazni popis

UC4 - Preimenovanje popisa

- **Glavni sudionik:** Korisnik
- **Cilj:** Preimenovanje popisa
- **Sudionici:** Baza podataka na uređaju
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik dugo pritisne ime otvorenog popisa
 2. Otvori se prikaz za upis teksta
 3. Izlaskom iz prikaza se sadržaj spremi kao novo ime popisa

UC5 - Kopiranje sadržaja popisa

- **Glavni sudionik:** Korisnik
- **Cilj:** Sve podatke s popisa nadodati na kraj drugog popisa
- **Sudionici:** Baza podataka na uređaju
- **Preduvjet:** Postojanje dva popisa (predložak i odredište)
- **Opis osnovnog tijeka:**
 1. Korisnik otvara popis predložak
 2. Korisnik pritisne gumb "kopiraj"
 3. Otvara se izbornik popisa
 4. Korisnik odabire popis s izbornika
 5. Svi artikli i njihove količine se dodaju na kraj popisa odredište
- **Opis mogućih odstupanja:**
 - 3.a Popis odredište ne postoji
 1. Korisnik dobije obavijest da nema drugih popisa i ne otvara se izbornik

UC6 - Dodavanje artikla na popis pomoću barkoda

- **Glavni sudionik:** Korisnik
- **Cilj:** Dodati određeni artikl na popis i u košaricu
- **Sudionici:** Baza podataka na uređaju, baza podataka na poslužitelju
- **Preduvjet:** Veza s poslužiteljem
- **Opis osnovnog tijeka:**
 1. Korisnik pritisne gumb za dodavanje artikla na popis pomoću barkoda
 2. Korisnik skenira barkod
 3. Baza na poslužitelju vrati informacije o artiklu povezanom s barkodom
 4. Artikl se doda na kraj popisa i u košaricu, pretpostavljena količina je 1
- **Opis mogućih odstupanja:**
 - 2.a Korisnik ne može pristupiti kameri, tj. barkod skeneru
 1. Sustav korisniku omogućava ručni upis znamenaka
 - 3.a Nema informacija o artiklu u bazi podataka na poslužitelju
 1. Upisuje se samo cijena ako postoji, inače se ispisuje poruka suosjećanja

UC7 - Dodavanje artikla na popis pretragom

- **Glavni sudionik:** Korisnik
- **Cilj:** Dodati određeni artikl na popis

- **Sudionici:** Baza podataka na uređaju, baza podataka na poslužitelju
- **Preduvjet:** Veza s poslužiteljem
- **Opis osnovnog tijeka:**
 1. Korisnik pritisne gumb za dodavanje artikla na popis pretragom
 2. Korisnik upisuje naziv artikla
 3. Korisnik odabire jedan od ponuđenih artikala
 4. Artikl se doda na kraj popisa, pretpostavljena količina je 1
- **Opis mogućih odstupanja:**
 - 3.a Nema traženog artikla
 1. Ispisuje se poruka suosjećanja na mjestu gdje bi trebali biti rezultati pretrage

UC8 - Dodavanje artikla na popis zadavanjem filtra

- **Glavni sudionik:** Korisnik
- **Cilj:** Dodati na popis artikl koji zadovoljava filter
- **Sudionici:** Baza podataka na uređaju, baza podataka na poslužitelju
- **Preduvjet:** Veza s poslužiteljem
- **Opis osnovnog tijeka:**
 1. Korisnik pritisne gumb za dodavanje artikla na popis pomoću filtra
 2. Korisnik postavlja filter i/ili redukciju (min, max)
 3. Artikl koji zadovoljava filter se doda na kraj popisa, pretpostavljena količina je 1
- **Opis mogućih odstupanja:**
 - 3.a Više artikala zadovoljava filter i/ili redukciju
 1. Odabire se onaj s najnižom cijenom koji je najbliži ako je dozvoljen pristup lokaciji, inače prvi u bazinom odgovoru na upit

UC9 - Dodavanje artikla iz popisa omiljenih artikala

- **Glavni sudionik:** Korisnik
- **Cilj:** Dodati na popis artikl iz popisa omiljenih artikala
- **Sudionici:** Baza podataka na uređaju
- **Preduvjet:** Postoji omiljeni artikl
- **Opis osnovnog tijeka:**
 1. Korisnik otvori bočnu traku
 2. Korisnik pritisne željeni artikl
 3. Artikl se doda na kraj popisa, pretpostavljena količina je 1

4. Popis se osvježi

UC10 - Zamjena artikla na popisu nekim drugim artiklom

- **Glavni sudionik:** Korisnik
- **Cilj:** Zamijeniti artikl na popisu nekim drugim artiklom
- **Sudionici:** Baza podataka na uređaju, baza podataka na poslužitelju
- **Preduvjet:** Postoji artikl na popisu, veza s poslužiteljem
- **Opis osnovnog tijeka:**
 1. Korisnik povuče artikl na popisu ulijevo
 2. Korisnik odabire novi artikl na onaj način na koji je dodan stari artikl
 3. Novi artikl je prikazan na mjestu starog, pretpostavljena količina je 1

UC11 - Micanje artikla s popisa

- **Glavni sudionik:** Korisnik
- **Cilj:** U potpunosti maknuti artikl s popisa
- **Sudionici:** Baza podataka na uređaju
- **Preduvjet:** Postoji artikl na popisu
- **Opis osnovnog tijeka:**
 1. Korisnik povuče artikl na popisu udesno
 2. Artikl se miče s popisa

UC12 - Registracija korisnika

- **Glavni sudionik:** Gost
- **Cilj:** Stvaranje korisničkog računa
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Veza s poslužiteljem
- **Opis osnovnog tijeka:**
 1. Gost otvara izbornik
 2. Gost u izborniku odabire "Registracija"
 3. Gost upisuje email, lozinku i potvrdu lozinke za registraciju kao kupac.
Dodatno tajni broj za registraciju kao trgovac ili administrator
 4. Baza provjerava postoji li već korisnik s istom adresom elektroničke pošte
 5. Korisnik dobiva povratnu informaciju
- **Opis mogućih odstupanja:**

- 3.a Email adresa je u neispravnom obliku, lozinka ima manje od 8 znakova ili se potvrda ne podudara
 - 1. Korisnik dobiva odgovarajuću informativnu poruku, podatci se ne brišu ni ne šalju
- 4.a U bazi već postoji korisnikova mail adresa
 - 1. Korisnik dobiva odgovarajuću informativnu poruku, podatci se ne brišu

UC13 - Prijava korisnika

- **Glavni sudionik:** Gost
- **Cilj:** Prijava u sustav
- **Sudionici:** Baza podataka na poslužitelju, email poslužitelj u slučaju odstupanja 2.a
- **Preduvjet:** Veza s poslužiteljima (sustav, baza podataka i email), gost ima račun
- **Opis osnovnog tijeka:**
 - 1. Gost otvara sučelje za prijavu
 - 2. Gost upisuje email adresu i lozinku ili se prijavi pomoću Google računa
 - 3. Baza podataka provjerava dane informacije
 - 4. Gost dobije povratnu informaciju
- **Opis mogućih odstupanja:**
 - 2.a Korisnik ne zna koju lozinku ima ili više ne želi koristiti postojeću
 - 1. Korisnik pritisne gumb "Promijeni lozinku"
 - 2. Korisnik upiše svoju mail adresu
 - 3. Sustav preko email poslužitelja šalje poruku s privremenom lozinkom koju sprema u bazu
 - 4. Korisnik dobiva novu lozinku na mail adresu
 - 5. Korisnik se prijavi u aplikaciju i promijeni lozinku (UC15)
 - 3.a Email adresa je u neispravnom obliku ili lozinka ima manje od 8 znakova
 - 1. Korisnik dobiva odgovarajuću informativnu poruku, podatci se ne brišu ni ne šalju
 - 3.b U bazi ne postoji korisnikova mail adresa s pripadnom lozinkom ili je račun označen kao onemogućen
 - 1. Korisnik dobiva odgovarajuću informativnu poruku

UC14 - Odjava korisnika

- **Glavni sudionik:** Prijavljeni korisnik (kupac, trgovac ili administrator)
- **Cilj:** Prijeći u način rada neprijavljenog korisnika
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Veza s poslužiteljem, korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Prijavljeni korisnik otvara izbornik
 2. Prijavljeni korisnik u izborniku odabire "Odjava"
 3. Korisnik je odjavljen

UC15 - Promjena lozinke

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Promijeniti lozinku
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Veza s poslužiteljem, korisnik je prijavljen email adresom i lozinkom
- **Opis osnovnog tijeka:**
 1. Prijavljeni korisnik otvara izbornik
 2. Prijavljeni korisnik u izborniku odabire "Promjena lozinke"
 3. Prijavljeni korisnik upiše staru lozinku, novu lozinku i potvrdu nove lozinke
 4. Prijavljeni korisnik dobiva obavijest o uspješnoj promjeni lozinke
- **Opis mogućih odstupanja:**
 - 3.a Korisnikova trenutna lozinka nije ispravna ili se potvrda nove lozinke ne podudara
 1. Korisnik dobiva odgovarajuću informativnu poruku

UC16 - Pregled informacija o artiklu

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati informacije o artiklu
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Veza s poslužiteljem i artikl na popisu
- **Opis osnovnog tijeka:**
 1. Korisnik pritisne ime ili cijenu artikla na popisu
 2. Korisniku se prikažu informacije o artiklu i mjera pouzdanosti tih informacija

UC17 - Izmjena informacija o artiklu

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Pregledati i izmijeniti informacije o artiklu
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Korisnik treba biti prijavljen, imati vezu s poslužiteljem i artikl na popisu
- **Opis osnovnog tijeka:**
 1. Korisnik pregleda informacije o artiklu
 2. Prijavljeni korisnik glasa o ispravnosti informacija
 3. Nakon davanja negativnog glasa, prijavljeni korisnik može odabrati drugi opis ili uređuje postojeći
 4. Prijavljeni korisnik skenira barkod tog proizvoda
 5. Promjena se spremi u bazu podataka
 6. Prijavljeni korisnik ubuduće vidi onaj opis koji je odabrao ili napisao te se tom opisu broj glasova povećava za jedan

UC18 - Omiljeni proizvodi

- **Glavni sudionik:** Korisnik
- **Cilj:** Učiniti artikl lakše dostupnim za dodavanje na popis, poništiti tu akciju
- **Sudionici:** Baza podataka na uređaju
- **Preduvjet:** Artikl je na popisu
- **Opis osnovnog tijeka:**
 1. Korisnik otvori informacije o artiklu
 2. Korisnik pritisne na zvjezdicu
 3. Ako artikl nije na popisu omiljenih proizvoda, pojavit će se na njemu, u suprotnom će se maknuti s njega
 4. Za pregled omiljenih proizvoda korisnik otvara bočnu traku

UC19 - Najjeftinija trgovina

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikazati trgovinu u kojoj je zbroj cijena zadanih artikala najniža
- **Sudionici:** Baza podataka na poslužitelju, baza podataka na uređaju
- **Preduvjet:** Veza s poslužiteljem
- **Opis osnovnog tijeka:**
 1. Korisnik otvara izbornik načina izračna cijena
 2. U izborniku bira "Najjeftinija trgovina"
 3. Cijene se osvježe

4. Ponovno osvježavanje popisa korisnik pokreće povlačenjem popisa prema dolje, pritom se porast cijene označava promjenom njene boje u crvenu, a pad u zelenu (UC18.1)
 5. Za informacije o trgovini korisnik pritisne na cijenu artikla
- **Opis mogućih odstupanja:**
 - 3.a Ne postoji trgovina sa svim artiklima s popisa
 1. Bira se trgovina koja ima najviše artikala na popisu, ako ima više takvih onda ona s najviše artikala općenito, a ako takvih ima više, onda ona koja bude prva u bazinom odgovoru na upit. Artiklima kojih nema u trgovini se zacrveni oznaka količine.
 - 3.b Postoji više trgovina s najnižom cijenom popisa
 1. Odabire se najbliža ako je dozvoljen pristup lokaciji, inače prva u bazinom odgovoru na upit
 - 4.a Cijena cijelog popisa postane manja u drugoj trgovini ili neki artikli postanu nedostupni u trenutnoj trgovini ili cijela trgovina postane nedostupna
 1. Trenutna trgovina se promijeni u onu koja ima sve artikle i najnižu cijenu popisa
 - 4.b Artikel ne postoji više niti u jednoj trgovini
 1. Naziv artikla promijeni boju u crvenu

UC20 - Najjeftiniji artikli

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikazati za svaki artikel najnižu cijenu
- **Sudionici:** Baza podataka na uređaju, baza podataka na poslužitelju
- **Preduvjet:** Veza s poslužiteljem
- **Opis osnovnog tijeka:**
 1. Korisnik otvara izbornik načina izračuna cijena
 2. U izborniku bira "Najjeftiniji artikli"
 3. Cijene se osvježe
 4. Ponovno osvježavanje popisa korisnik pokreće povlačenjem popisa prema dolje, pritom se porast cijene označava promjenom njene boje u crvenu, a pad u zelenu (19.1)
 5. Za informacije o trgovini u kojoj je cijena dostupna korisnik pritisne na cijenu artikla
- **Opis mogućih odstupanja:**

- 3.a Postoji više trgovina s najnižom cijenom artikla
 - 1. Odabire se najbliža ako je dozvoljen pristup lokaciji, inače prva u bazinom odgovoru na upit
- 4.a Artikl s popisa više nije u ponudi u trenutnoj trgovini, skuplji je nego negdje drugo ili trenutna trgovina više ne postoji
 - 1. Cijena se uzme iz trgovine gdje je najjeftinija
- 4.b Artikl ne postoji više niti u jednoj trgovini
 - 1. Naziv artikla promijeni boju u crvenu

UC21 - Prikaz cijena u točno određenoj trgovini

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikazati cijene artikala u točno određenoj košarici
- **Sudionici:** Baza podataka na uređaju, baza podataka na poslužitelju
- **Preduvjet:** Veza s poslužiteljem
- **Opis osnovnog tijeka:**
 - 1. Korisnik otvara izbornik načina izračuna cijena
 - 2. U izborniku bira "Određena trgovina"
 - 3. Korisnik upisuje naziv trgovine
 - 4. Korisnik odabire jednu od ponuđenih trgovina. Ako je dozvoljen pristup lokaciji, poredane su po udaljenosti od korisnika
 - 5. Cijene se osvježe Pipijem
 - 6. Ponovno osvježavanje popisa korisnik pokreće povlačenjem popisa prema dolje
 - 7. Za informacije o trgovini korisnik pritisne na cijenu artikla
- **Opis mogućih odstupanja:**
 - 4.a Nema tražene trgovine
 - 1. Ispisuje se poruka suosjećanja na mjestu gdje bi trebali biti rezultati pretrage
 - 5.a Nema artikla s popisa u zadanoj trgovini
 - 1. Artiklu se oznaka količine zacrveni označavajući da ga nema u skladištu
 - 6.a Artikl s popisa više nije u ponudi u zadanoj trgovini
 - 1. Oznaka količine artikla promijeni boju u crvenu
 - 6.b Artikl ne postoji više niti u jednoj trgovini i/ili trgovina više ne postoji
 - 1. Naziv artikla i/ili trgovine promijeni boju u crvenu

UC22 - Dodavanje trgovine

- **Glavni sudionik:** Trgovac
- **Cilj:** Dodati informacije o trgovini
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Trgovac je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Trgovac otvara svoju nadzornu ploču
 2. Trgovac odabire opciju "Dodaj trgovinu"
 3. Za dodavanje trgovine trgovac upisuje naziv trgovine, njenu adresu, radno vrijeme i koordinate

UC23 - Brisanje trgovine

- **Glavni sudionik:** Trgovac ili administrator
- **Cilj:** Obrisati trgovinu iz sustava
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Trgovac je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Trgovac ili administrator otvara svoju nadzornu ploču
 2. Trgovac ili administrator odabire trgovinu koju ju želi obrisati
 3. Trgovac ili administrator pritisne gumb za brisanje i potvrdi odluku
 4. Baza obriše podatke o trgovini
 5. Trgovac dobiva potvrdu na mail o brisanju trgovine

UC24 - Mijenjanje podataka o trgovini

- **Glavni sudionik:** Trgovac
- **Cilj:** Promijeniti podatke o trgovini ili artiklima u njima
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Trgovac je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Trgovac otvara svoju nadzornu ploču
 2. Trgovac odabire trgovinu kojoj želi promijeniti podatke
 3. Trgovac promijeni željene podatke ili doda nove artikle kao petorku (barkod, cijena, popust, dostupnost, email_zeljenog_opisivaca) ili ih doda više odjednom kao csv datoteku
 4. Baza podataka spremi promjene
- **Opis mogućih odstupanja:**
 - 3.a Informacije o artiklima ne odgovaraju zadanom formatu

1. Akcija ili transakcija se prekida i trgovac dobiva odgovarajuću informativnu poruku
- 3.b Polje "email_zeljenog_opisivaca" je "None" ili mail nekoga tko nije opisao dani artikl
 1. S trgovinom neće biti povezan određeni opis artikla nego će se prikazivati pretpostavljeni (onaj s najviše pozitivnih glasova)

UC25 - Onemogućenje pristupa

- **Glavni sudionik:** Administrator
- **Cilj:** Onemogućavanje pristupa kupcu
- **Sudionici:** Baza podataka na poslužitelju, email poslužitelj u slučaju odstupanja 3.a
- **Preduvjet:** Administrator je prijavljen u sustav na web sučelju, veza s poslužiteljima (email, baza podataka i sustav)
- **Opis osnovnog tijeka:**
 1. Administrator otvara svoju nadzornu ploču na web-sučelju
 2. Administrator odabire opciju "Zabrani pristup korisničkom računu"
 3. Administrator odabire račun kojem želi zabraniti pristup
 4. Baza podataka označava račun kao onemogućen
 5. Korisnik je odjavljen čim uspostavi vezu s poslužiteljem
- **Opis mogućih odstupanja:**
 - 3.a Odabran je administratorski račun
 1. Račun je označen jednom, ali ne gubi funkcionalnost dok ga ne označi bar 50% administratora u istom danu
 2. Šalje se email poruka svim ostalim administratorima tko koga pokušava onemogućiti

UC26 - Uklanjanje opisa artikla

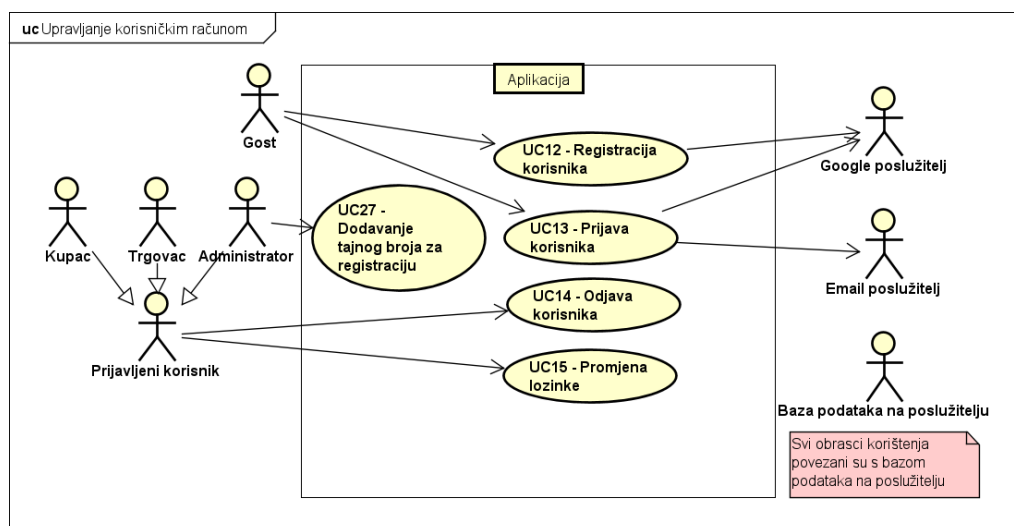
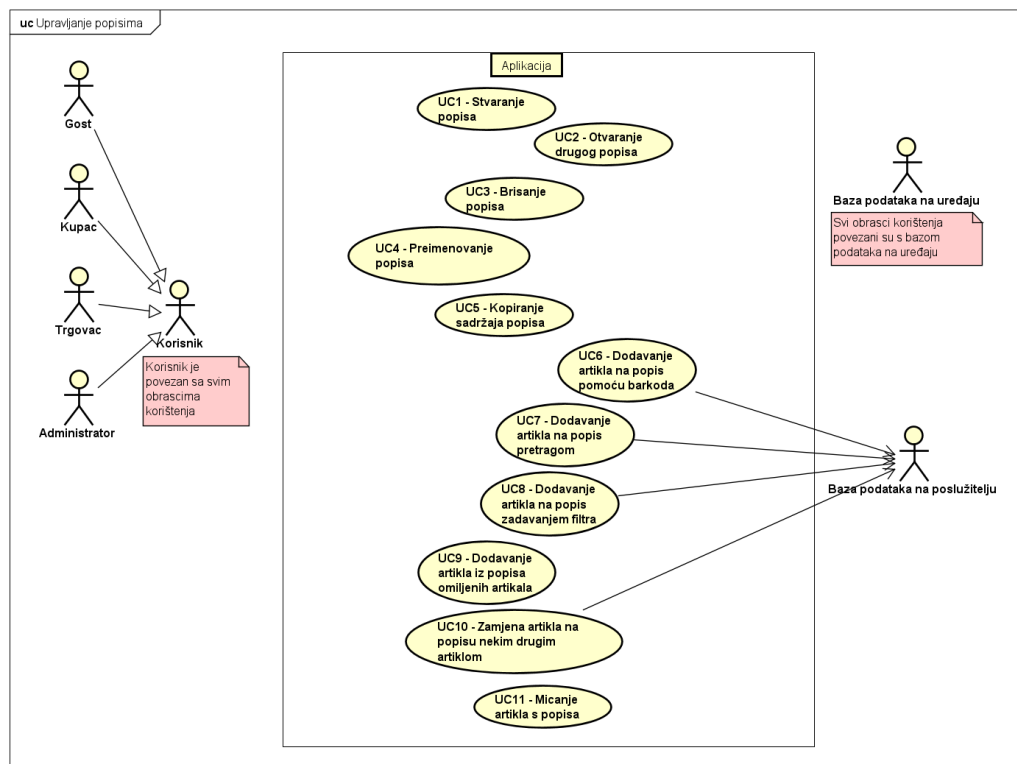
- **Glavni sudionik:** Administrator
- **Cilj:** Uklanjanje opisa artikla iz sustava
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Administrator je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Administrator otvara svoju nadzornu ploču na web-sučelju
 2. Administrator odabire opciju "Uredi artikle"
 3. Administrator odabire opis artikla koji želi maknuti

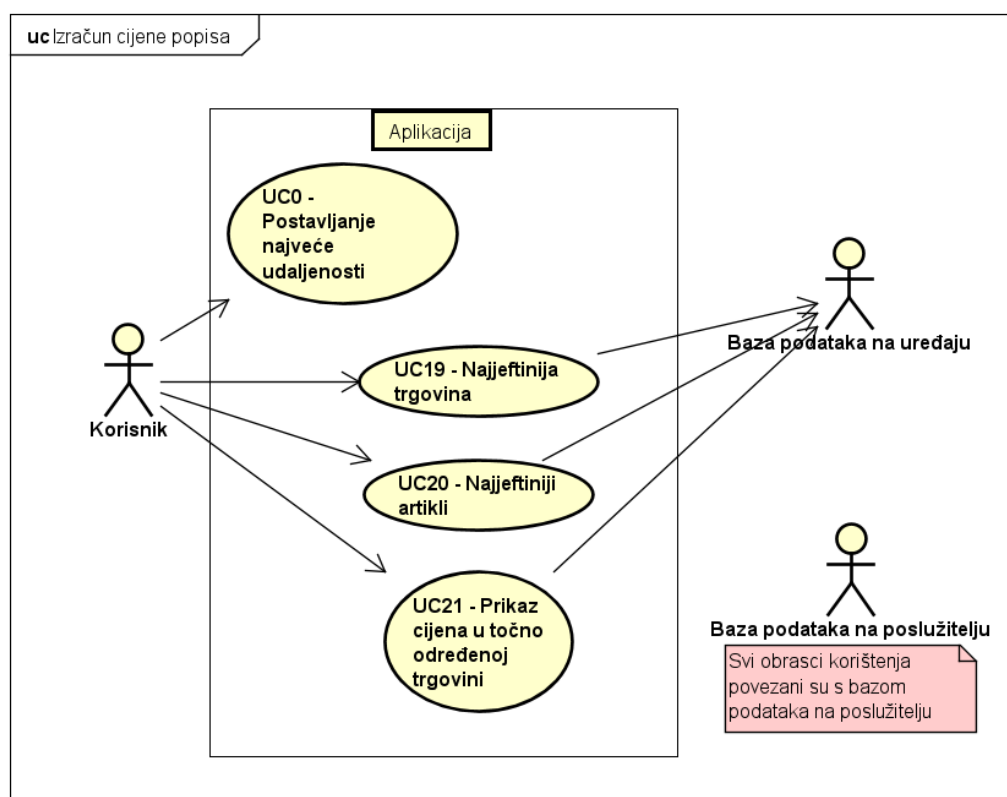
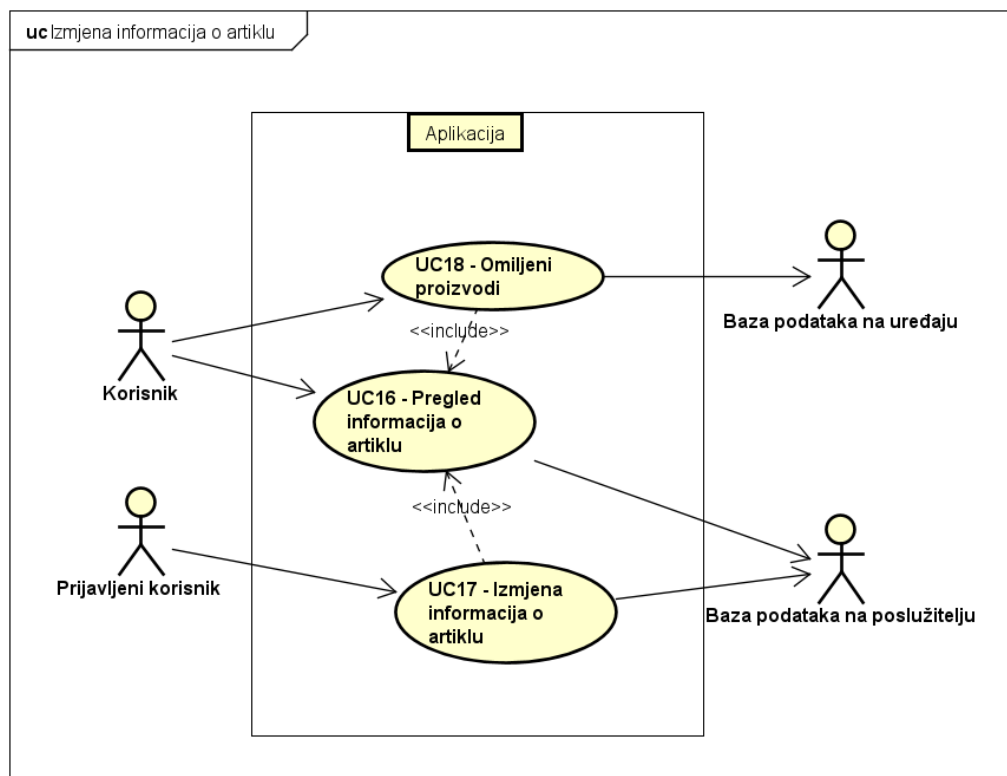
4. Administrator vidi mail adresu koorisnika koji je napisao sporni opis
5. Administrator pritisne gumb "Ukloni opis artikla"
6. Baza uklanja opis artikla

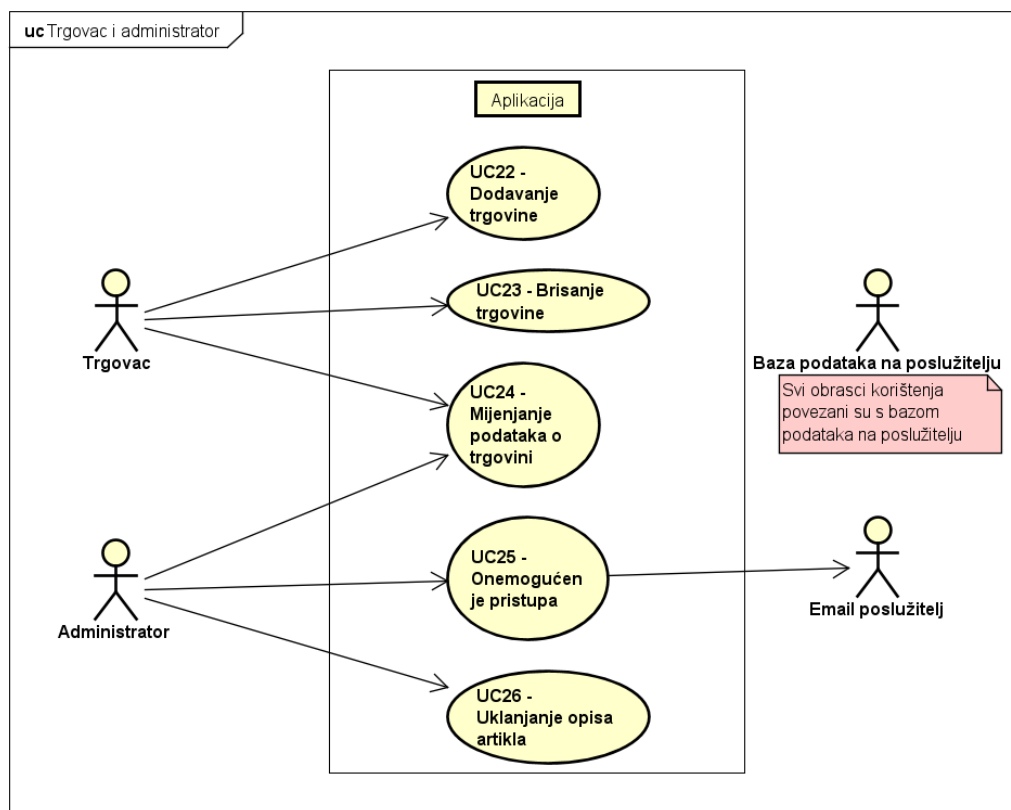
UC27 - Dodavanje tajnog broja za registraciju

- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti registraciju trgovcu i/ili administratoru
- **Sudionici:** Baza podataka na poslužitelju
- **Preduvjet:** Administrator je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Administrator otvara svoju nadzornu ploču na web-sučelju
 2. Administrator odabire opciju "Dodaj verifikaciju"
 3. Administrator dodaje tajni broj i email za registraciju
 4. Podatci se spremaju u bazu podataka

Dijagrami obrazaca uporabe



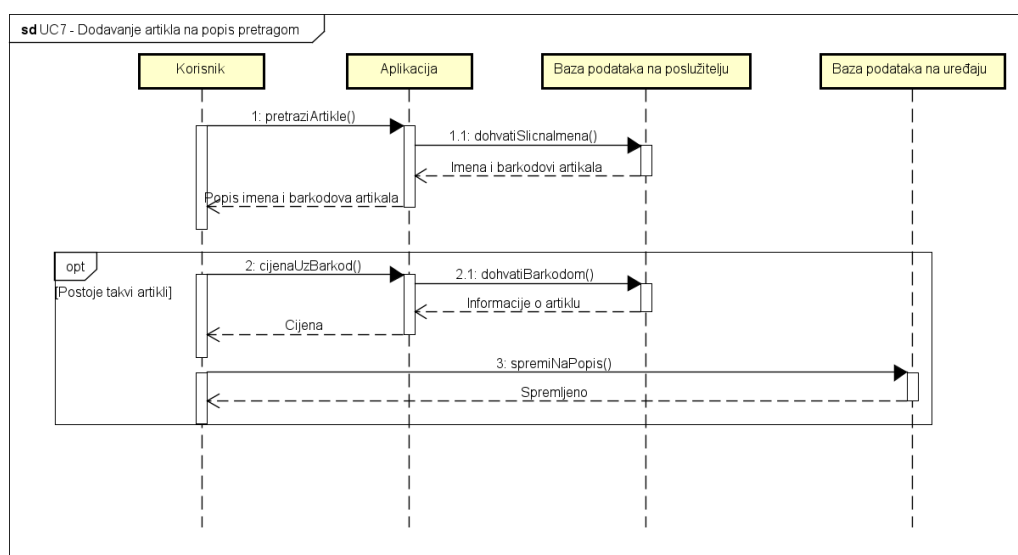




3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC7 - Dodavanje artikla na popis pretragom

Korisnik pošalje traženo ime artikla. Poslužitelj iz baze podataka dohvaća barkodove i imena artikala sličnog imena. Prosljeđuje ih u listi korisniku. Ako lista nije prazna, korisnik odabire jedan od ponuđenih artikala i šalje poslužitelju njegov barkod i način izračuna cijene popisa, poslužitelj na temelju toga dohvati iz baze podataka informacije o artiklu i prosljeđuje ih cijenu korisniku. Korisnik ima sve potrebne informacije o stavci popisa i sprema ju u bazu podataka na uređaju.



Slika 3.1: Sekvencijski dijagram za UC7

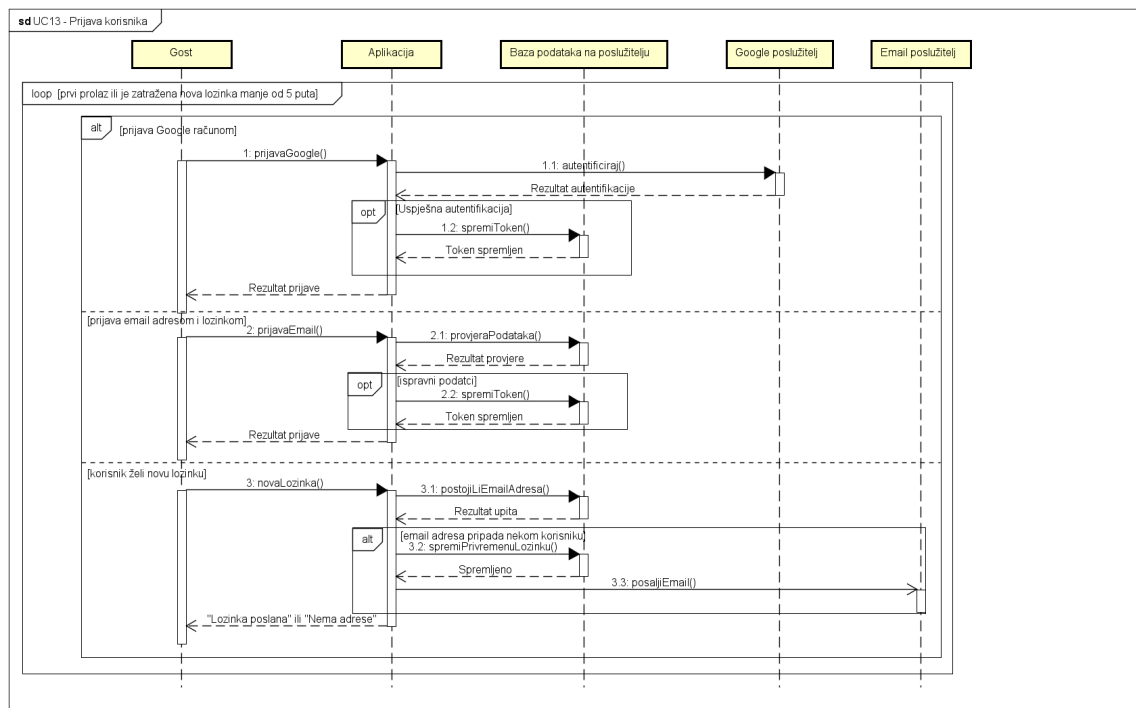
Obrazac uporabe UC13 - Prijava korisnika

Ako se korisnik odlučio prijaviti Google računom, pošalje zahtjev za to poslužitelju koji provjerava njegove vjerodajnice zahtjevom prema Google poslužitelju. Ako su ispravne, u bazu podataka na poslužitelju sprema se token za korisnika i prijava je uspješna. Poslužitelj vraća rezultat prijave korisniku.

Ako se korisnik odlučio prijaviti email adresom i lozinkom, podatke za prijavu šalje na poslužitelj. Poslužitelj u bazi podataka provjerava jesu li ispravni. Ako jesu, poslužitelj sprema sjednički token u bazu podataka. na kraju poslužitelj vraća rezultat prijave korisniku.

Ako je korisnik odlučio zatražiti novu lozinku, šalje svoju email adresu poslužitelju. On provjerava u bazi podataka postoji li korisnik s tom email adresom. Ako postoji, generira privremenu lozinku, sprema ju u bazu podataka i šalje emailom korisniku

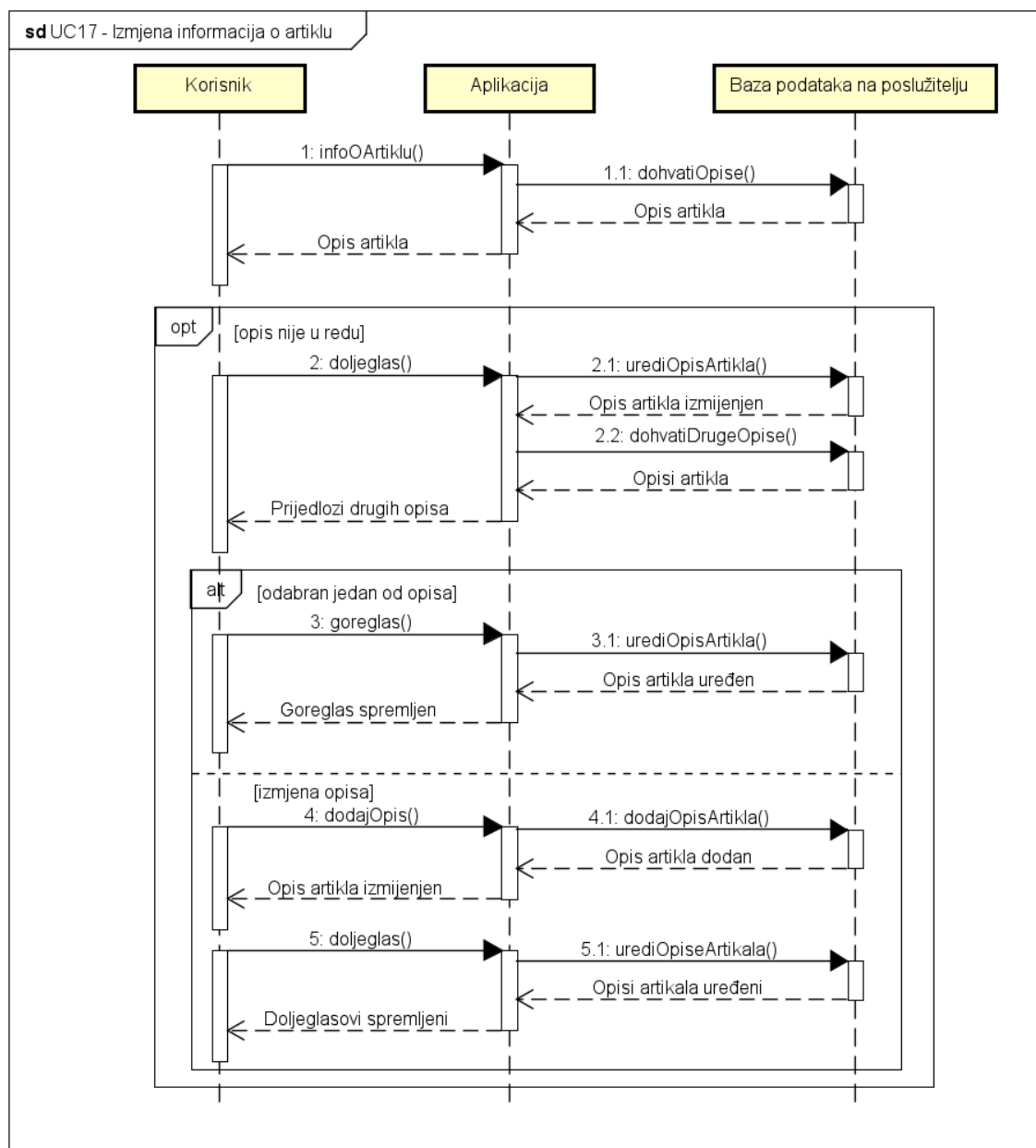
preko email poslužitelja. Korisnik dobiva informaciju o tome je li lozinka poslana.



Slika 3.2: Sekvencijski dijagram za UC13

Obrazac uporabe UC17 - Izmjena informacija o artiklu

Korisnik zatraži od poslužitelja informacije o artiklu. Korisnik dohvati informacije u bazi podataka i prosljeđuje ih korisniku. Ako korisnik zaključi da opis nije u redu, šalje zahtjev za smanjenjem broja glasova opisa. Poslužitelj ažurira broj glasova u bazi podataka i dohvati alternativne opise u bazi podataka koje zatim šalje korisniku. Ako se korisniku sviđa jedan od ponuđenih opisa, bira ga i tada se poslužitelju šalje zahtjev za povećanjem broja glasova opisa. Poslužitelj u bazi ažurira broj glasova i šalje korisniku povratnu informaciju. Ako se pak korisniku ne sviđa niti jedan od ponuđenih opisa, šalje svoj opis poslužitelju koji se sprema u bazu podataka i šalje zahtjev za smanjenjem broja glasova ponuđenih artikala. Poslužitelj ažurira broj glasova u bazi podataka i šalje povratnu informaciju korisniku.



Slika 3.3: Sekvencijski dijagram za UC17

3.2 Ostali zahtjevi

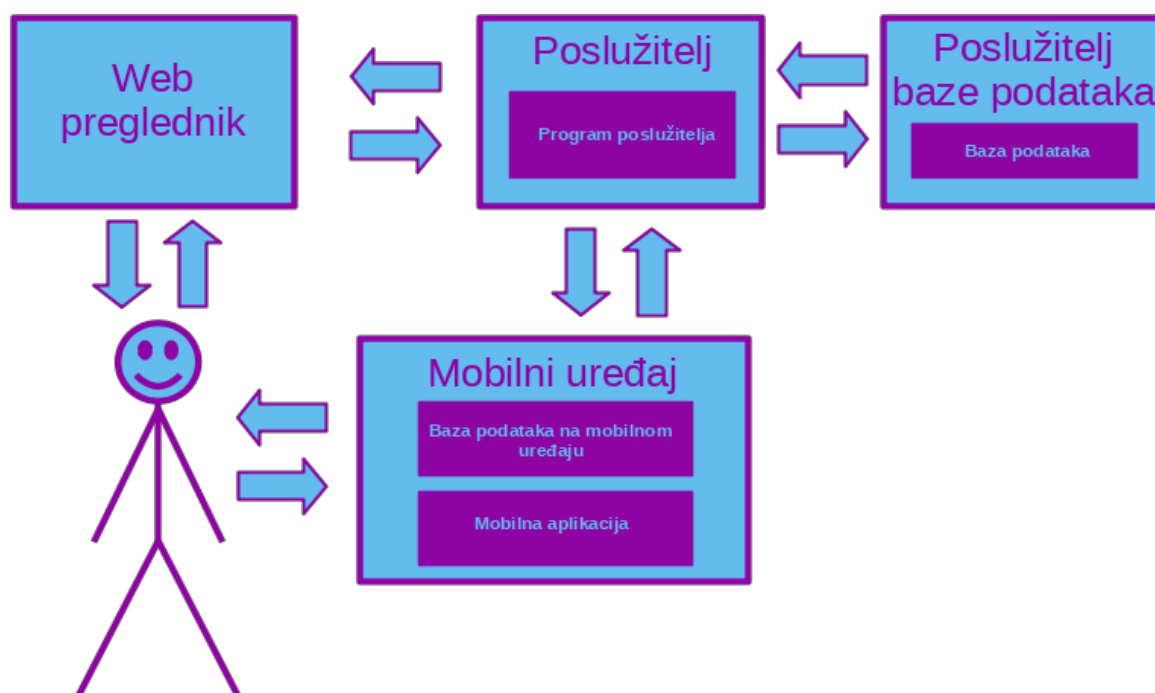
- Aplikacija mora podržavati istovremeni rad 1000 korisnika
- Baza podataka mora odgovoriti na upit unutar pet sekundi
- Baza podataka mora podržavati unos 100 000 opisa artikala
- Baza podataka mora biti zaštićena od SQL injekcija
- Baza podataka mora biti otporna na napade duginom tablicom (rainbow table attack) dodavanjem predmetka lozinci
- Baza podataka mora lozinke raspršiti (hashing) pomoću funkcije SHA-256
- Pretpostavljena valuta sustava je HRK
- Korisničko sučelje mora podržavati hrvatski jezik i dijakritičke znakove
- Korisničko sučelje mora biti intuitivno
- Korisničko sučelje mora minimizirati broj koraka potrebnih da se učini neka akcija
- Neispravno korištenje korisničkog sučelja ne smije ugroziti rad sustava
- Aplikaciju mora biti moguće pokrenuti na Android uređajima inačice 5.0 naviše

4. Arhitektura i dizajn sustava

Arhitektura je podijeljena na tri podsustava:

- Mobilna aplikacija
- Poslužitelj
- Poslužitelj baze podataka

Okviri i jezici koristimo izabrani su imajući na umu, između ostaloga, njihovu relativno dugotrajnu popularnost, što znači da osim male vjerojatnosti score zastare postoji i veliki skup korisnika koji pružaju podršku, kao i problema koje su ti korisnici već riješili, a dostupni su na stranicama poput stackoverflow.com. Također su izvrsno dokumentirani.



Slika 4.1: Skica arhitekture sustava

Aplikacija na mobilnom uređaju napravljena je za sustav Android. Dio je front-end sloja, ali i u sklopu operativnog sustava komunicira s ugrađenom lokalnom

bazom podataka (dakle nema zasebni poslužitelj). Preko protokola HTTP komunicira s poslužiteljem. Mobilna aplikacija napisana je u strojno podržanom (native) jeziku Androida - Java - koji je objektno orijentirani jezik što nam omogućuje podjelu u razrede koji su u paketima grupirani u međusobno povezane elemente, a pritom nam omogućuje i zadržavanje apstrakcije. Omogućuje nam i ponovnu iskoristivost jer bilo koji uređaj koji ima Java virtualni stroj može pokrenuti isti kôd. Android kao programski okvir omogućava veliku fleksibilnost (kroz razmjешtanje i imenovanje resursa prema dogovoru), poput automatske promjene jezika ovisno o korisničkim postavkama ili rezolucije slika ovisno o rezoluciji ekrana.

Kôd na poslužitelju dio je backend sloja i napisan je u Pythonu 3 koji se, osim spomenutim prednostima objektno orijentirane paradigme, odlikuje i jednostavnošću kodiranja što značajno smanjuje vrijeme razvoja. Pogotovo uz backend okvir *django* koji omogućava fleksibilnost ne koristeći "Convention over configuration" paradigmu, ali je i dalje jednostavan jer je pisan u Pythonu. Njegovo svojstvo "batteries included" smanjuje količinu posla razvojnog timu akcijama poput automatskog stvaranja sučelja za administratora kojemu se može pristupiti iz web preglednika korištenjem HTTP protokola te jednostavnog povezivanja s email poslužiteljem pomoću SMTP protokola.

Koristimo stilističku varijaciju arhitekture zasnovane na događajima: MVC (*Model, View, Controller*) obrazac. Kako mu i ime kaže, dijeli aplikaciju na model, pogled (View) i nadglednik (Controller). Time se omogućava odvajanje korisničkog sučelja (kroz pogled) od ostatka sustava. Za upravljanje zahtjevima korisnika služi nadglednik, a model opisuje strukturu i pravila vezane uz podatke. Podržava ga *django*.

4.1 Baze podataka

Za spremanje podataka koristimo, zbog njihove uvriježnosti i uglavnom velike sličnosti s našim poimanjem onoga što predstavljaju, relacijske baze podataka. One dijele informacije o objektima u attribute, a vrste objekata u tablice, tj. relacije. Takve baze podataka upravo postoje i optimirane su za učinkovito spremanje čitanje i izmjenu podatka, pa ih zato za to i koristimo. Konkretno, na poslužitelju smo se odlučili za PostgreSQL, a na Android uređaju imamo SQLite3.

Slijede opisi tablica i njihovih veza. Imena atributa primarnog ključa deblje su otisnuta, a pozadinska boja ćelije u kojoj je ime stranog ključa promijenjena je u plavu.

4.1.1 Opis tablica na uređaju

Tablica omiljeni sadrži šifre stavaka koje su označene kao omiljene. Tablica omiljeni je u *Many-to-one* vezi s tablicom stavka (preko atributa sifStavka).

omiljeni		
sifStavka	INT	šifra stavke koja je označena kao omiljena

Tablica popis sadrži nazive popisa i njihove načine izračuna cijene. Tablica popis je u *One-to-many* vezi s tablicom stavka (preko atributa sifPopis).

popis		
sifPopis	INT	šifra popisa
nazivPopis	NCHAR VARYING	naziv popisa
izrCijene	BIT	način izračuna cijene
sifTrgovina	INT	šifra trgovine vezane uz popis (ako je primjenjivo)

Tablica stavka sadrži stavke na popisu. Tablica stavka je u *One-to-many* vezi s tablicom omiljeni (preko atributa sifStavka). Tablica stavka je u *Many-to-one* vezi s tablicom popis (preko atributa sifPopis).

stavka		
sifStavka	INT	šifra stavke
sifPopis	INT	šifra popisa na kojem je stavka
barkod	DECIMAL	barkod artikla
cijena	DECIMAL	cijena artikla
filtarFunkcija	VARCHAR	sadrži sažeto napisane filter funkcije koje se primjenjuju
uKosarici	BIT	oznaka je li stavka dodana u košaricu
kolicina	DECIMAL	količina artikla u intervalu [0,000 - 999,999]
naziv	NCHAR VARYING	naziv artikla

stavka		
sifTrgovina	INT	šifra trgovine vezane uz cijenu

4.1.2 Opis tablica na poslužitelju

Tablica artikl sadrži popis svih barkodova artikala. Tablica artikl je u *One-to-many* vezi s tablicom opisArtikla (preko atributa barkod) i tablicom artiklUTrgovini (preko atributa barkod).

artikl		
barkod	DECIMAL	barkod artikla

Tablica artiklUTrgovini sadrži popis artikala u trgovini. Tablica artiklUTrgovini je u *Many-to-one* vezi s tablicom artikl (preko atributa barkod), tablicom trgovina (preko atributa sifTrgovina) i tablicom opisArtikla (preko atributa barkod i email).

artiklUTrgovini		
barkod	DECIMAL	barkod artikla
sifTrgovina	CHAR	šifra trgovine
email	VARCHAR	email vezan uz željeni opis
cijena	DECIMAL	cijena artikla u trgovini
popust	FLOAT	popust na cijenu artikla u trgovini
dostupnost	BIT	ima li artikla u trgovini

Tablica kategorija sadrži popis kategorija artikala. Tablica kategorija je u *One-to-many* vezi s tablicom potkategorija (preko atributa sifKategorija).

kategorija		
sifKategorija	INT	šifra kategorije artikla
nazivKategorija	NCHAR VARYING	naziv kategorije artikla

Tablica korisnik sadrži informacije o prijavljenim korisnicima aplikacije. Tablica korisnik je u *One-to-many* vezi s tablicom onemoguceniRacun (preko atributa email), tablicom opisArtikla (preko atributa email), tablicom privremenaLozinka (preko atributa email) i tablicom trgovina (preko atributa email). Tablica korisnik je u *Many-to-one* vezi s tablicom uloga (preko atributa sifUloga).

korisnik		
email	VARCHAR	email korisnika
sifUloga	INT	šifra uloge korisnika
lozinka	VARCHAR	hash lozinke korisnika (SHA 256)
token	CHAR	token sjednice korisnika

Tablica onemoguceniRacun sadrži popis onemogućenih računa. Tablica onemoguceniRacun je u *Many-to-one* vezi s tablicom korisnik (preko atributa adminEmail).

onemoguceniRacun		
onemoguceni	VARCHAR	email osobe kojoj je onemogućen pristup
adminEmail	VARCHAR	email admina koji je onemogućio osobi pristup
datum	DATE	datum onemogućenja

Tablica opisArtikla sadrži opis i informacije vezane uz artikl zadan barkodom koje je napisao neki prijavljeni korisnik. Tablica opisArtikla je u *One-to-many* vezi s tablicom artiklUTrgovini (preko atributa barkod i email). Tablica opisArtikla je u *Many-to-one* vezi s tablicom artikl (preko atributa barkod), tablicom korisnik (preko atributa email), tablicom vrsta (preko atributa sifKategorija, sifPotkategorija i sifVrsta) i tablicom zemlja (preko atributa sifZemlja).

opisArtikla		
barkod	DECIMAL	barkod artikla
email	VARCHAR	email osobe koja je napisala opis
sifKategorija	INT	šifra kategorije artikla
sifPotkategorija	INT	šifra potkategorije artikla
sifVrsta	INT	šifra vrste artikla
sifZemlja	CHAR	šifra zemlje podrijetla
kratkiOpis	NCHAR VARYING	kratki opis artikla (do 255 znakova)
nazivArtikla	NCHAR VARYING	naziv artikla (do 32 znaka)
brojGlasova	INT	broj glasova o opisu artikla (goreglasovi - doljeglasovi)
masa	INT	masa artikla

Tablica potkategorija sadrži popis potkategorija artikala. Tablica potkategorija je u *One-to-many* vezi s tablicom vrsta (preko atributa sifKategorija i sifPotkategorija). Tablica potkategorija je u *Many-to-one* vezi s tablicom kategorija (preko atributa sifKategorija).

potkategorija		
sifKategorija	INT	šifra kategorije kojoj potkategorija artikla pripada
sifPotkategorija	INT	šifra podkategorije artikla
nazivPotkategorija	VARCHAR VARYING	naziv potkategorije artikla

Tablica privremenaLozinka sadrži privremene lozinke poslane na email korisnika. Tablica privremenaLozinka je u *Many-to-one* vezi s tablicom korisnik (preko atributa email).

privremenaLozinka		
email	VARCHAR	email korisnika koji je zatražio promjenu
lozinka	CHAR	hash privremene lozinke
istice	DATE	datum isteka lozinke

Tablica tajniBroj sadrži popis tajnih brojeva za registraciju. Tablica tajniBroj je u *Many-to-one* vezi s tablicom uloga (preko atributa sifUloga).

tajniBroj		
broj	INT	tajni broj
sifUloga	INT	šifra uloge koju će korisnik poprimiti
idKorisnika	INT	neki identifikator osobe koja će iskoristiti tajni broj

Tablica trgovina sadrži informacije o trgovinama. Tablica trgovina je u *One-to-many* vezi s tablicom artiklUTrgovini (preko atributa sifTrgovina). Tablica trgovina je u *Many-to-one* vezi s tablicom korisnik (preko atributa email).

trgovina		
sifTrgovina	CHAR	šifra trgovine
email	VARCHAR	email vlasnika trgovine

trgovina		
lat	DECIMAL	geografska širina trgovine
lon	DECIMAL	geografska dužina trgovine
nazivTrgovina	NCHAR	naziv trgovine
adresa	VARCHAR	adresa trgovine
radnoVrijemePočetak	TIME	početak radnog vremena trgovine
radnoVrijemeKraj	TIME	kraj radnog vremena

Tablica uloga sadrži popis uloga koje prijavljeni korisnik može poprimiti (administrator, trgovac ili kupac). Tablica uloga je u *One-to-many* vezi s tablicom korisnik (preko atributa sifUloga) i tablicom tajniBroj (preko atributa sifUloga).

uloga		
sifUloga	INT	šifra uloge korisnika
nazUloga	VARCHAR	naziv uloge prijavljenog korisnika

Tablica vrsta sadrži popis vrsta artikala. Tablica vrsta je u *One-to-many* vezi s tablicom opisArtikla (preko atributa sifKategorija, sifPotkategorija i sifVrsta). Tablica vrsta je u *Many-to-one* vezi s tablicom potkategorija (preko atributa sifKategorija i sifPotkategorija).

vrsta		
sifKategorija	INT	šifra kategorije kojoj vrsta artikla pripada
sifPotkategorija	INT	šifra potkategorije kojoj vrsta artikla pripada
sifVrsta	INT	šifra vrste
nazivVrsta	NCHAR VARYING	naziv vrste artikla

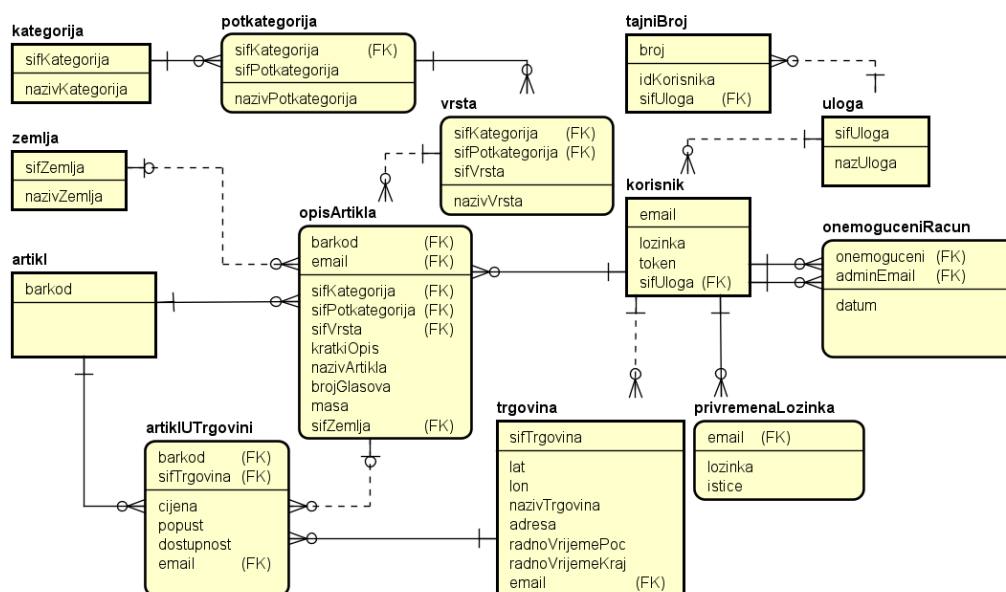
Tablica zemlja sadrži popis zemalja koje korisnik može odabrati kao zemlju podrijetla. Tablica zemlja je u *One-to-many* vezi s tablicom opisArtikla (preko atributa sifZemlja).

zemlja		
sifZemlja	CHAR	šifra zemlje podrijetla
nazivZemlja	NCHAR VARYING	naziv zemlje podrijetla

4.1.3 Dijagrami baze podataka



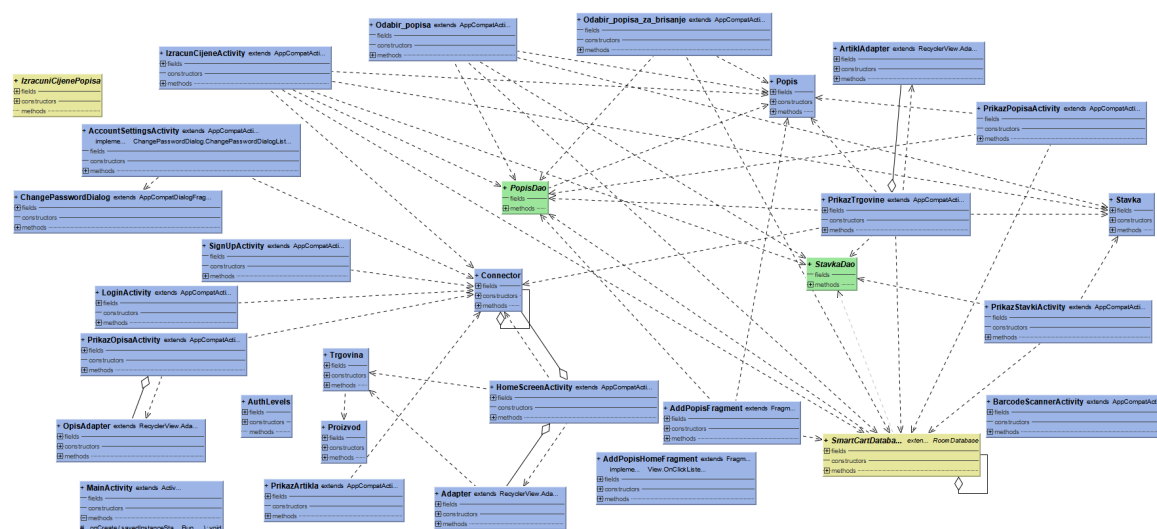
Slika 4.2: Baza podataka na uređaju



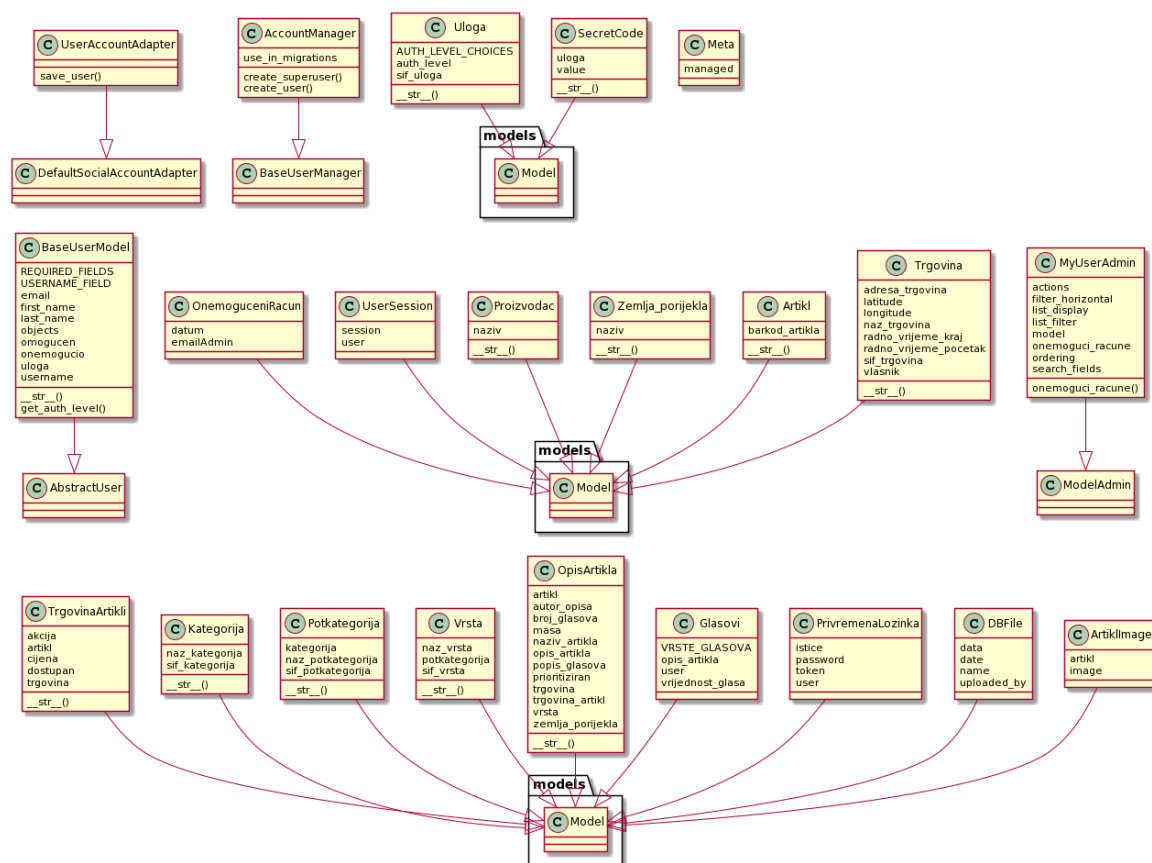
Slika 4.3: Baza podataka na poslužitelju

4.2 Dijagrami razreda

Dijagram razreda na uređaju sadrži klase koje nisu dio generične funkcionalnosti, ali su ubačene radi preglednosti. To su List, Listitem, RoomDB, Favourites i About



Slika 4.4: Dijagram razreda na uređaju

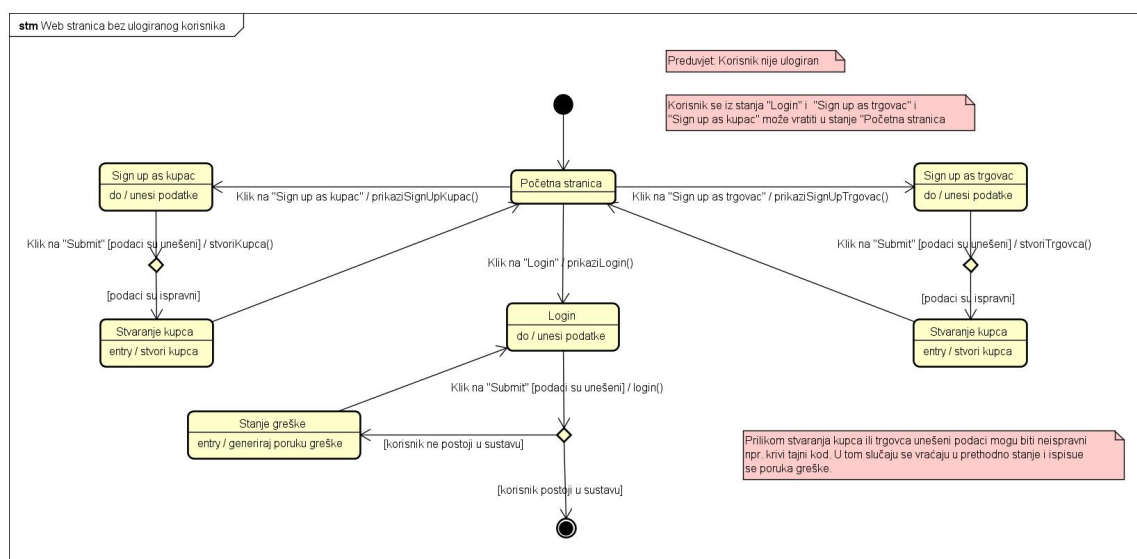


Slika 4.5: Dijagram razreda na poslužitelju

4.3 Dijagram stanja

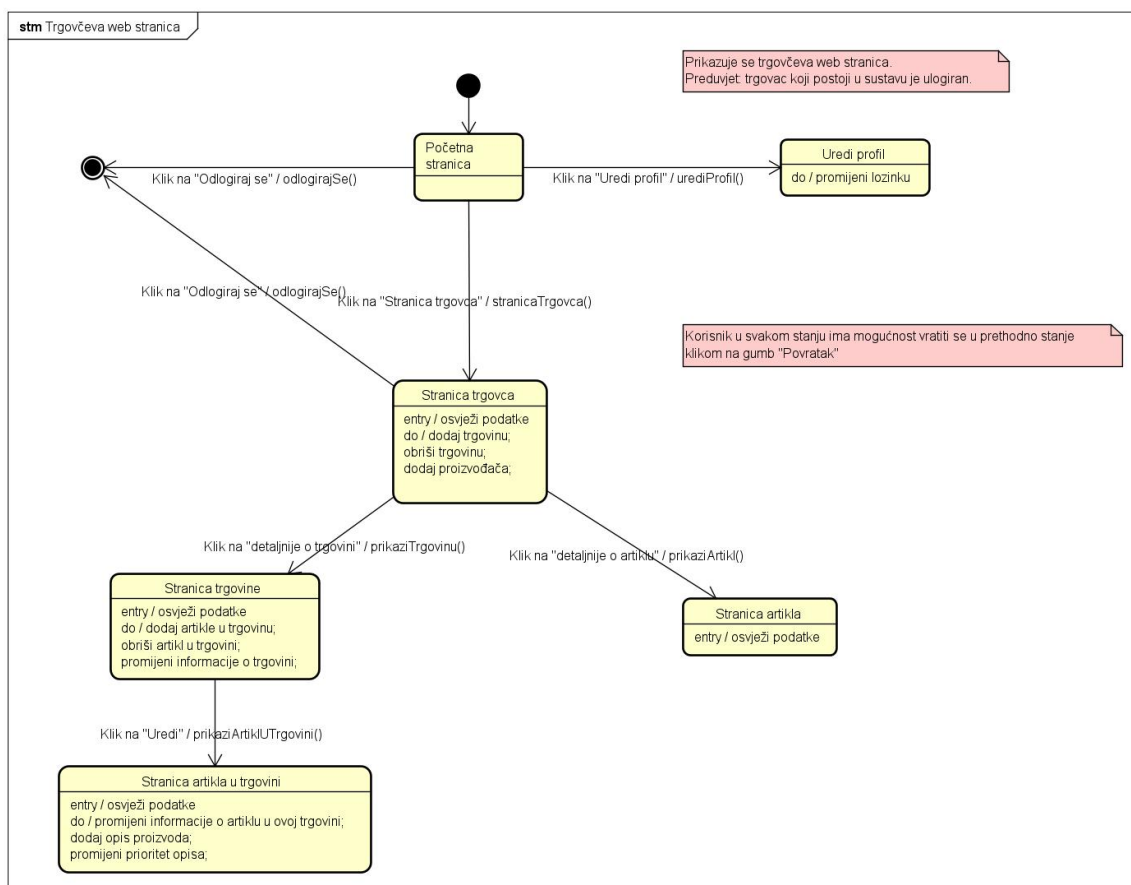
Priložena su dva dijagrama stanja. Jedan prikazuje web stranicu na kojoj korisnik nije ulogiran, a drugi dijagram stanja prikazuje web stranicu na kojoj je ulogiran trgovac.

Dijagram stanja koji opisuje web stranicu na kojoj korisnik nije ulogiran nudi nekoliko opcija: prijava, registracija kao kupac i registracija kao trgovac. Registrirati se kao može bilo tko samo jednom s istom email adresom. Za registraciju kao kupac nisu potrebni nikakvi preduvjeti, dok je za registraciju kao trgovac potreban odgovarajući kod.



Slika 4.6: Dijagram stanja web stranice za neulogiranog korisnika

Sljedeći dijagram stanja opisuje web stranicu na kojoj je ulogiran trgovac. On ima već nabrojane opcije kao što su pregled trgovina, dodavanje trgovine, pregled proizvoda u trgovini, brisanje proizvoda u trgovini itd. Preduvjet za svaku akciju brisanja je naravno da proizvod koji se želi obrisati postoji. Također, što se tiče pregleda trgovine, trgovac može pregledati samo onu trgovinu za koju je on zadužen.

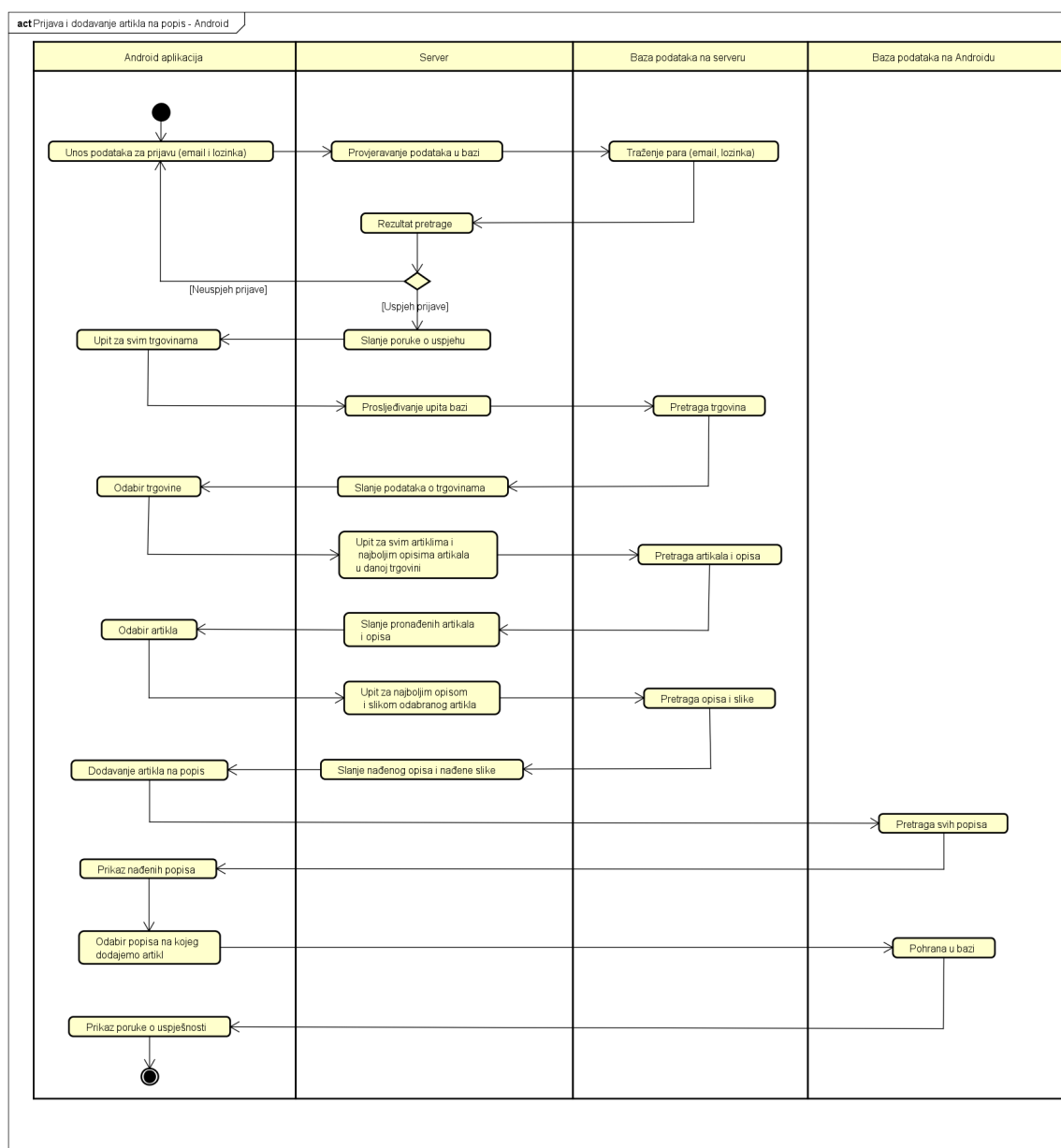


Slika 4.7: Dijagram stanja web stranice za trgovca

4.4 Dijagram aktivnosti

Na slici prikazan je dijagram aktivnosti. On obuhvaća sve aktivnosti koje se događaju na Android aplikaciji, serveru, serverskoj i Android bazi podataka. Obuhvaćeni skup aktivnosti obuhvaća sve od ulogiravanja korisnika na Android uređaju pa do dodavanja artikla na popis.

Korisnik se dakle prvo ulogira, zatim odabere trgovinu te proizvod koji je ponuđen u toj trgovini. Odabirom proizvoda, prikazuje se detaljni opis tog proizvoda, njegova slika te opcije "upvote", "downvote" te "Dodaj na popis". Ovdje sam prikazao postupak dodavanja na popis. Jedan od preduvjeta je taj da postoji barem jedan popis na uređaju.



Slika 4.8: Dijagram aktivnosti za prijavu i dodavanje artikla na popis (Android)

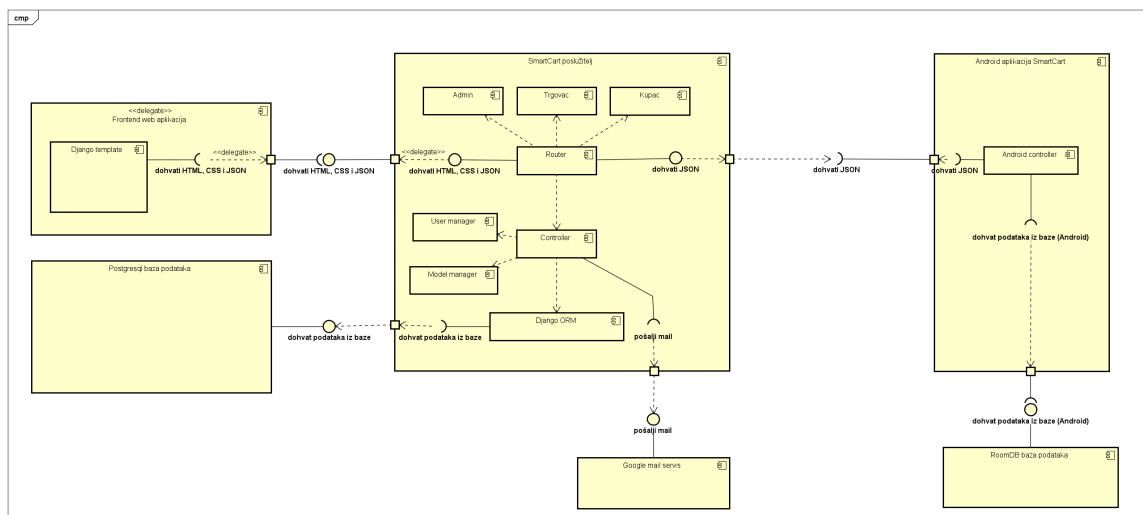
4.5 Dijagram komponenti

Dijagram prikazuje komponente aplikacije koje komuniciraju preko sučelja. Web aplikacija od servera, uz podatke u JSON formatu, mora dobiti i izgled stranice - HTML i CSS. Android aplikacija ne traži HTML i CSS zbog toga što je prikaz primljenih podataka od servera (koji se također dobijaju u JSON formatu) definiran na samom Androidu.

Server ima vlastitu bazu podataka s kojom komunicira preko sučelja koje dolazi

u paketu Django, a svaki Android uređaj komunicira sa svojom bazom podataka preko sučelja definiranim paketom DBRoom.

Također, Google pruža svoje sučelje koje se može iskoristiti za automatsko slanje mailova. Tu funkcionalnost koristi server kako bi korisniku poslao potvrdu o promjeni lozinke.



Slika 4.9: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Programsko rješenje sastoji se od dva dijela - Android aplikacija i web aplikacija.

Android dio pisan je u okruženju AndroidStudio (<https://developer.android.com/studio>), u nativnom jeziku Java. Za testiranje aplikacije koristili smo ili Android Virtual Device kojeg smo pokretali u okruženju AndroidStudio ili smo aplikaciju direktno instalirali i pokretali na Android uređaju. Baza podataka koju smo koristili za Android je RoomDB koja koristi jezik MySQL.

Web dio pisan je u jeziku Python 3, a korišteni framework je Django. Django omogućava lakše razvijanje i upravljanje bazom podataka, funkcijama kao što su login i logout te nudi dinamičko generiranje HTML stranica. Okruženje koje je smo koristili za razvoj serverskog dijela većinom je ovisilo od osobe do osobe. Neki su preferirali Visual Studio Code (<https://code.visualstudio.com/>), a neki PyCharm (<https://www.jetbrains.com/pycharm/>), okruženje za Python temeljeno na okruženju IntelliJ. I VSCode i PyCharm nude ugrađene funkcionalnosti za VCS tj. Version Control System.

Frontend na web dijelu pisan je u HTML-u uz korištenje bootstrap-a i css-a kako bi aplikacija bila responzivna i za mobilne uređaje i uređaje s velikim zaslonima.

Baza podataka na serverskom dijelu (tj. glavna baza podataka) je Postgres baza podataka. Korišteni SQL jezik je dakle PostgreSQL, a korišteni DB Manager je pgAdmin (<https://www.pgadmin.org/>).

Komunikacija članova tima je realizirana preko aplikacije Slack (<https://slack.com/>) i Whatsapp, a samim verzijama koda se upravljalo preko sustava git stranici Gitlab (<https://about.gitlab.com/>).

Dokumentacija je pisana u okruženju TexStudio (<https://www.texstudio.org/>)

i u xed-u.

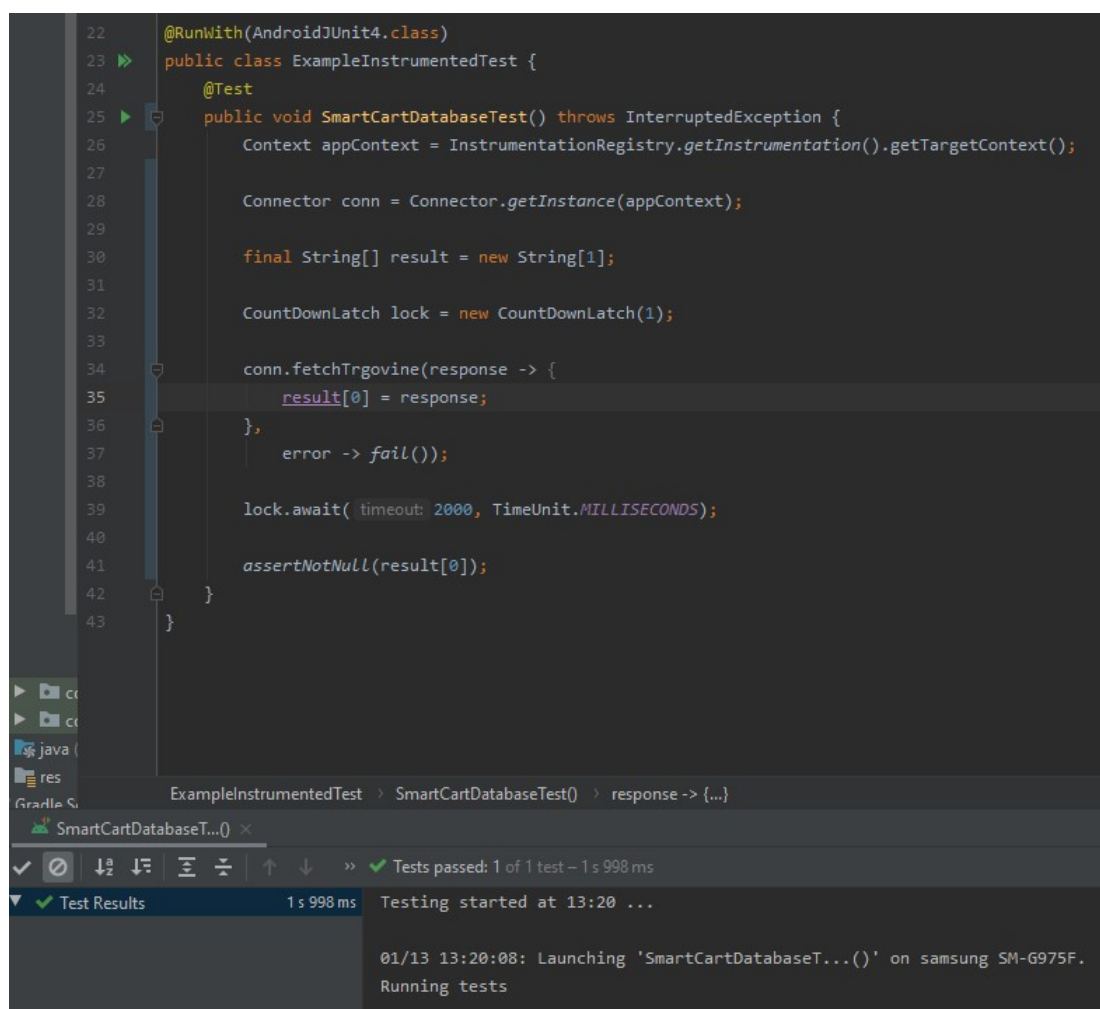
5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

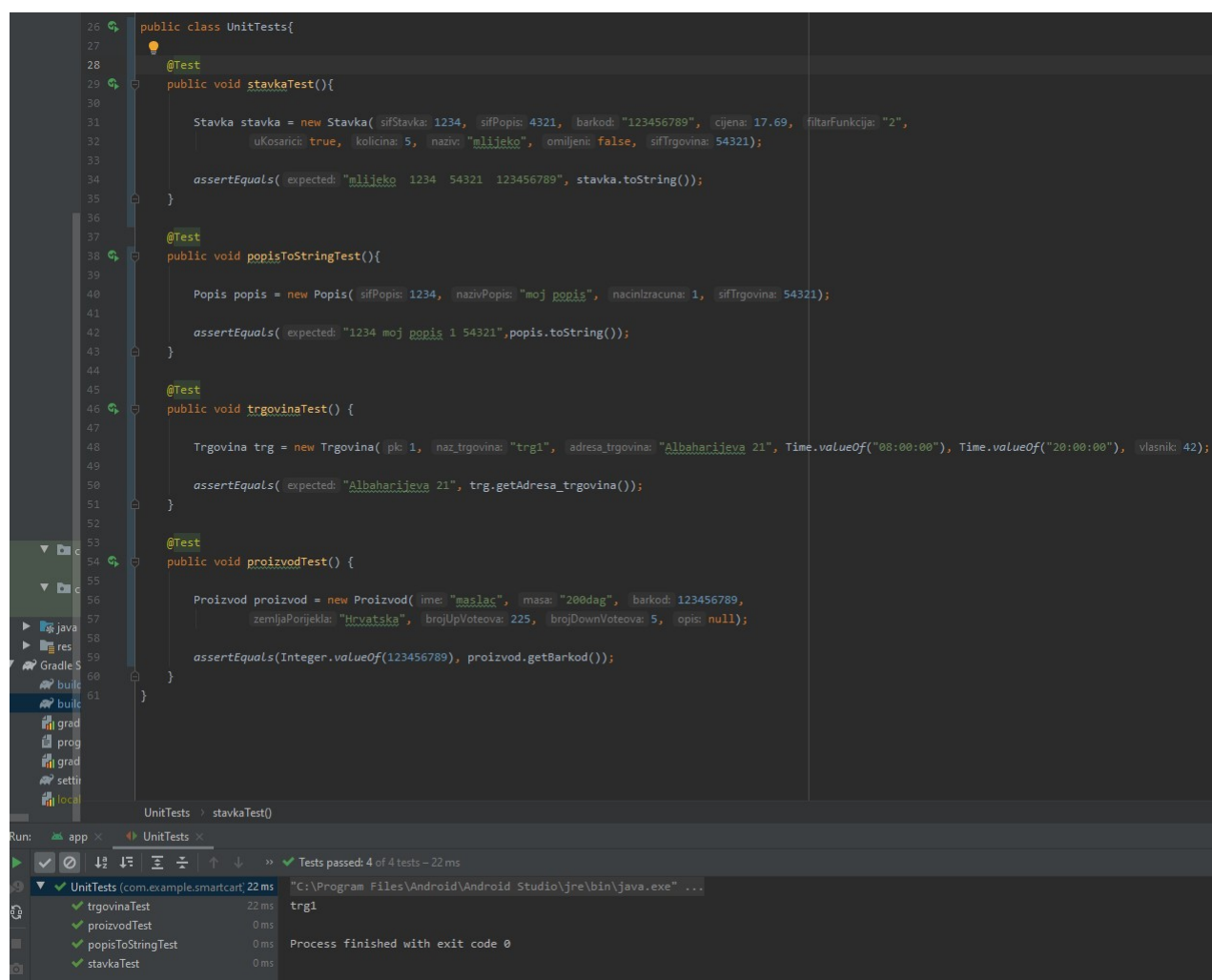
Budući da ja aplikacija napisana za Android uređaje ispitivanje komponenti se provodi pomoću razreda `AndroidJUnit4` (<https://developer.android.com/reference/androidx/test/ext/junit/runners/AndroidJUnit4>) uz korištenje funkcionalnosti dostupnih u modulu `JUnit5` (<https://junit.org/junit5/>)

Testiraju se razne funkcionalnosti dostupne na android uređaju, među kojima vrijedi izdvojiti testiranje dohvata podataka iz baze podataka na uređaju, testiranje prikazivanja stavki na popisu, testiranje prikazivanja popisa, testiranje prikazivanja adrese trgovina i testiranje prikazivanja proizvoda

Primjeri testiranja:



Slika 5.1: Testiranje baze podataka na uređaju



Slika 5.2: Testiranje razreda na uređaju

5.2.2 Ispitivanje sustava

Ispitivanje sustava je provedeno koristeći funkcionalnosti iz paketa unittest koji dolazi u instalaciji programskog jezika Python u kombinaciji sa alatom Selenium WebDriver koji pruža podršku za automatsko upravljanje internetskim preglednikom. Za instalaciju podrške za ispitivanje sustava potrebno je sa interneta skinuti Selenium za programski jezik Python (<https://pypi.org/project/selenium/>, također je moguće instalirati ako se unutar naredbenog retka pozove komanda *pip install selenium*). Uz Selenium za Python potrebno je instalirati i Selenium WebDriver. U našem projektu koristimo geckodriver 0.28.0, verziju Selenium WebDriver-a za internetski preglednik Mozilla Firefox, koja se može skinuti sa sljedeće web-adrese: (<https://github.com/mozilla/geckodriver/releases/tag/v0.28.0>). Nad sustavom se provode razna testiranja. Neka od najvažnijih su:

- Testiranje izrade novog profila i pokušaja prijave na web-stranicu pomoću točne i krive kombinacije (relevantan isječak iz koda):

```
def fill_login_form(self, username, password):
    self.driver.get(HOME_PAGE)
    self.driver.find_element(By.XPATH, '//*[@text()="Log
        in"]').click()
    self.driver.find_element(By.NAME, "username").send_keys(
        username)
    self.driver.find_element(By.NAME, "password").send_keys(
        password)
    self.driver.find_element(By.NAME, "submit_button").click
        ()

def test_successful_login(self):
    self.fill_login_form("ante@fer.hr", "pwd")
    self.assertEqual(self.driver.current_url, f"{HOME_PAGE}/
        trgovac")

def test_wrong_login(self):
    self.fill_login_form("ante@fer.hr", "abc")
    self.assertTrue(str(self.driver.current_url).startswith(f
        "{HOME_PAGE}/login"))

def test_create_new_profile(self):
    self.driver.get(HOME_PAGE)
    self.driver.find_element(By.XPATH, '//*[@text()="Sign
        up as kupac"]').click()
    self.driver.find_element(By.NAME, "email").send_keys(
        NEW_USERNAME)
    self.driver.find_element(By.NAME, "password").send_keys(
        NEW_PASSWORD)
    self.driver.find_element(By.NAME, "confirm_password").
        send_keys(NEW_PASSWORD)
    self.driver.find_element(By.NAME, "submit_button").click
        ()
    self.fill_login_form(NEW_USERNAME, NEW_PASSWORD)
    self.assertTrue(f"Logged in as {NEW_USERNAME}" in self.
        driver.page_source)
```

- Testiranje dodavanja novog proizvoda (relevantan isječak iz koda):

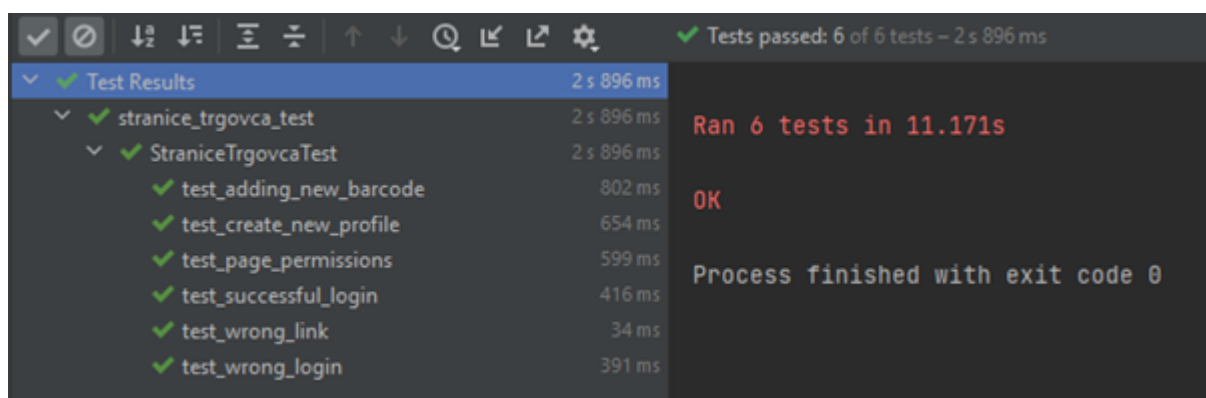
```
def test_adding_new_barcode(self):
    self.fill_login_form("ante@fer.hr", "pwd")
    self.driver.find_element(By.NAME, "barkod_artikla").
        send_keys(BARCODE)
    self.driver.find_element(By.NAME, "new_barcode_button").
        click()
    list_of_products = self.driver.find_element(By.ID, "
        products")
    self.assertTrue(BARCODE in list_of_products.text)
```

- Testiranje „sigurnosti“ stranice (nepostojeći linkovi i pristup linkovima koji zahtijevaju posebnu dozvolu) (relevantan isječak iz koda):

```
def test_wrong_link(self):
    self.driver.get(f"{HOME_PAGE}/another_link/
        that_doesnt_exist")
    self.assertEqual(self.driver.current_url, f"{HOME_PAGE}/")
    )
```

```
def test_page_permissions(self):
    self.fill_login_form(NEW_USERNAME, NEW_PASSWORD)
    self.driver.get(f"{HOME_PAGE}/trgovac")
    self.assertEqual(self.driver.current_url, f"{HOME_PAGE}/")
    )
```

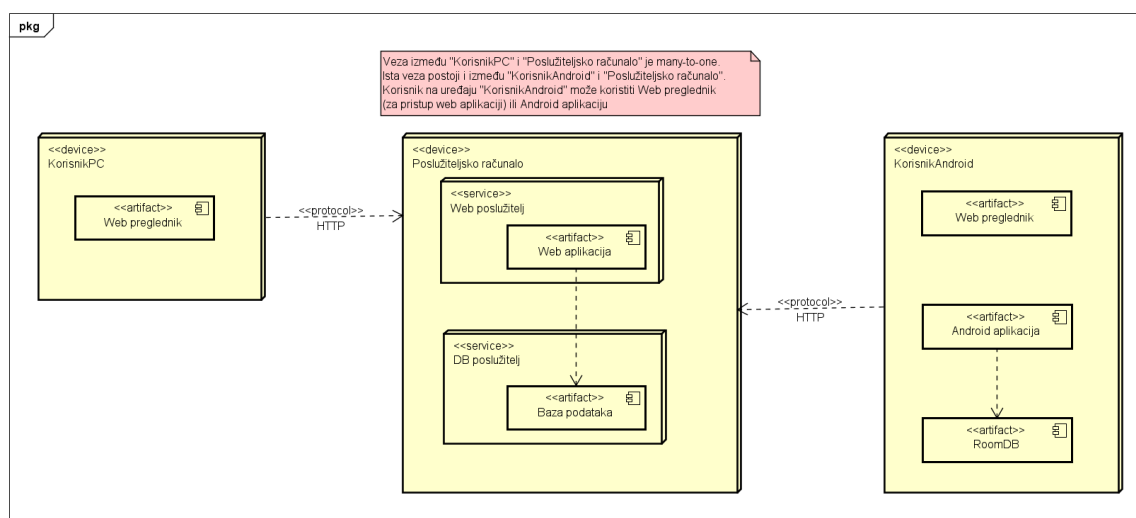
Razultati testiranja:



Slika 5.3: Razultati testiranja servera

5.3 Dijagram razmještaja

Server i serverska baza podataka nalaze se na jednom računalu. Korisnik ima nekoliko opcija ako želi pristupiti aplikaciji. Korisnik može web aplikaciji pristupiti iz bilo kojeg web preglednika. Nadalje, korisnik može aplikaciji pristupiti i preko Android uređaja na kojem je instalirao SmartCart aplikaciju. Android aplikacija, kao i web aplikacija, komunikaciju sa serverom obavljaju HTTP protokolom.



Slika 5.4: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Upute za puštanje u pogon napisane su za linux operacijski sustav. Za pokretanje aplikacije na javnom poslužitelju potrebno je napraviti račun za Heroku na stranici <https://signup.heroku.com/>. Zatim treba instalirati heroku na računalo i ulogirati se naredbama:

```
sudo snap install --classic heroku
heroku login
```

Nakon toga treba se pozicionirati u korijensku mapu repositoryja (preljevstoga) i stvoriti heroku projekt naredbom:

```
heroku create ime-projekta
```

Za stvaranje baze podataka pokrenite

```
heroku addons:create heroku-postgresql
```

Dohvatite url baze podataka i kopirajte ga u varijablu okruženja na Heroku.

```
heroku pg:info
```

```
heroku config:set DATABASE_URL=postgres://kopirani/url CREATE_DB=True FILL_DB=True
```

Zatim je potrebno stvoriti ključ za django aplikaciju naredbom

```
python3 izvorniKod/backend/generate_django_key.py
```

Taj ključ spremljen je u datoteci temp u mapi izvorniKod/backend. Spremite ga na Heroku, ali ne kopiranjem, da ne ostane u povijesti terminala nego sljedećom naredbom:

```
heroku config:set SECRET_CODE=$(cat izvorniKod/backend/temp)
```

Kako biste omogućili Google sign in u Vašoj aplikaciji, stvorite pomoću Vašeg Google računa Google ID i ključ za svoj projekt prema uputama u odlomku "Create authorization credentials" na stranici <https://developers.google.com/identity/sign-in/web/sign-in>. Google ID zalijepite u red 192 u izvorniKod/backend/settings.py, a gmail adresu tog računa 166. red iste datoteke. Tajni ključ kopirajte u neku novu datoteku (npr. umjesto django ključa u izvorniKod/backend/temp. Pošaljite taj ključ na Heroku naredbom:

```
heroku config:set GOOGLE_CLIENT_SECRET=$(cat izvorniKod/backend/temp)
```

Maknite datoteke s ključevima iz repozitorija! Commitajte promjene i pushajte na Heroku.

```
git add .
```

```
git commit -m "Promijenjena gmail adresa za slanje poruka"
```

```
git push heroku master
```

Na kraju onemogućite stvaranje i punjenje baze podataka jer već sadrži sve što treba.

```
heroku config:set CREATE_DB=False FILL_DB=False
```

Ako je sve prošlo kako treba, aplikacija je dostupna na ime-projekta.herokuapp.com.

Za pokretanje aplikacije na mobitelu otvorite izvorniKod/android pomoću Android studija i u klasi Connector izmijenite varijablu host u url vašeg Heroku projekta. Spojite mobitel i pokrenite aplikaciju pritiskom na "run".

6. Zaključak i budući rad

U ovom projektu koji je trajao 15 tjedana smo trebali izraditi sustav *SmartCart*, sustav za pronalaženje najjeftinije nabavne košarice u korisnikovoj blizini. U to vrijeme je uračunato osmišljavanje ideje za projekt, razradu te ideje, napraviti specifikaciju sustava, implementirati te dokumentirati, testirati i staviti sustav u pogon. Aplikacija posjeduje frontend dio (baziran na operacijskom sustavu *Android*) i backend dio (baziran na programskom jeziku *Python*).

Naš najveći problem je bila sama veličina našeg projekta naspram dodijeljenog vremena i upoznavanje s alatima za izradu našeg sustava. S vremenom smo počeli ovladavati alatima tako da je s vremenom postajalo lakše raditi. Primarno smo se koristili internetom za upoznavanje s našim alatima i metodama kako bi postigli željenu funkcionalnost i u konačnici naš cilj za ovaj projekt. Sastanci grupe su se održavali redovito kroz cijeli semestar i često više nego inicijalno dogovoreno. Primijetili smo bitnost kod točnog specificiranja što se traži od svakog člana grupe jer smo imali slučajeve da se nismo posve razumjeli pa zbog toga neke dijelove projekta smo morali ispočetka raditi ili prepravljati. Primijetili smo bitnost i ostalih predmeta koje smo dosad imali u sklopu našeg visokoškolskog obrazovanja (posebice predmeta *Vještine komuniciranja*, *Objektno orijentirano programiranje*, *Razvoj programske potpore za web i mobilne uređaje*, *Baze podataka* i *Komunikacijske mreže*) bez kojih bi ovaj projekt bio teško izvediv. Smatramo da je ovaj projekt pridonio našem poštovanju tuđih mišljenja, prihvaćanju činjenica da neke naše osobne ideje neće doći u konačni proizvod jer se s tom idejom ne slaže natpolovična većina svih članova grupe i potrebi da se kontinuirano radi jer inače se ne bi ovaj projekt mogao ostvariti do zacrtanog roka.

Sustav bi još mogli nadograditi podrškom za *iOS* sustav i poboljšati estetiku samog projekta. Neki članovi grupe su iskazali interes da se i nakon završetka ovog projekta nastavi daljnji razvoj te da se krene monetizirati projekt.

Popis literature

1. StackOverflow, <https://stackoverflow.com/>
2. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
3. The Unified Modeling Language, <https://www.uml-diagrams.org/>
4. Astah Community, <http://astah.net/editions/uml-new>
5. Django dokumentacija, <https://docs.djangoproject.com/en/3.1/>
6. Android dokumentacija, <https://developer.android.com/docs>
7. LaTeX dokumentacija, <https://www.latex-project.org/help/documentation/>

Indeks slika i dijagrama

3.1	Sekvencijski dijagram za UC7	28
3.2	Sekvencijski dijagram za UC13	29
3.3	Sekvencijski dijagram za UC17	30
4.1	Skica arhitekture sustava	32
4.2	Baza podataka na uređaju	39
4.3	Baza podataka na poslužitelju	39
4.4	Dijagram razreda na uređaju	40
4.5	Dijagram razreda na poslužitelju	41
4.6	Dijagram stanja web stranice za neulogiranog korisnika	42
4.7	Dijagram stanja web stranice za trgovca	43
4.8	Dijagram aktivnosti za prijavu i dodavanje artikla na popis (Android)	44
4.9	Dijagram komponenti	45
5.1	Testiranje baze podataka na uređaju	48
5.2	Testiranje razreda na uređaju	49
5.3	Razultati testiranja servera	51
5.4	Dijagram razmještaja	52
6.1	Dijagram pregleda promjena 1	63
6.2	Dijagram pregleda promjena 2	64

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 14. listopada 2020.
- Prisustvovali: cijeli PreljevStoga
- Teme sastanka:
 - diskusija zadatka
 - upoznavanje s git-om i GitLab-om

2. sastanak

- Datum: 15. listopada 2020.
- Prisustvovali: dr.sc.M.Krhen, I.Rissi, cijeli PreljevStoga
- Teme sastanka:
 - diskutiranje zadatka koji si je grupa zadala
 - savjeti za rad na projektu

3. sastanak

- Datum: 22. listopada 2020.
- Prisustvovali: dr.sc.M.Krhen, I.Rissi, D.Grubelić, J.Komljenović, A.Lakoš, L.Pranjić, M.Vladić
- Teme sastanka:
 - upit vezan uz ispravno pisanje dokumentacije

4. sastanak

- Datum: 24. listopada 2020.
- Prisustvovali: cijeli PreljevStoga
- Teme sastanka:
 - odabir tehnologija
 - raspoređivanje poslova
 - definiranje zadataka za naredni tjedan

5. sastanak

- Datum: 28. listopada 2020.

- Prisustvovali: T.Bjelčić, D.Grubelić, J.Komljenović, A.Lakoš, L.Pranjić, M.Vladić
- Teme sastanka:
 - demonstracija prve inačice dokumentacije
 - definiranje strukture git *repositoryja* i raščišćavanje drugih nedoumica oko git-a
 - priprema za izradu temeljne inačice Android dijela aplikacije
 - definiranje daljnjeg rasporeda poslova

6. sastanak

- Datum: 29. listopada 2020.
- Prisustvovali: D.Grubelić, J.Komljenović, A.Lakoš
- Teme sastanka:
 - demonstracija rada s djansom J.Komljenovića

7. sastanak

- Datum: 29. listopada 2020.
- Prisustvovali: dr.sc.M.Krhen, I.Rissi, T.Bjelčić, D.Grubelić, J.Komljenović, A.Lakoš, L.Pranjić, M.Vladić
- Teme sastanka:
 - dogovor oko izmjene funkcionalnih zahtjeva
 - upit vezan uz pravilan izgled UML dijagrama

8. sastanak

- Datum: 1. studenog 2020.
- Prisustvovali: T. Bjelčić, D.Grubelić, J.Komljenović, A.Lakoš, M.Vladić
- Teme sastanka:
 - demonstracija napretka na backend dijelu
 - dogovor oko daljnjeg rasporeda poslova

9. sastanak

- Datum: 2. studenog 2020.
- Prisustvovali: T.Bjelčić, D.Grubelić, J.Komljenović, A.Lakoš, L.Pranjić, M.Vladić
- Teme sastanka:
 - rasprava o budućim planovima s obzirom na zadaće iz drugog predmeta
 - dogovor o prebacivanju SQL upita o artiklima i trgovinama na poslužitelj

10. sastanak

- Datum: 6. studenog 2020.
- Prisustvovali: cijeli PreljevStoga
- Teme sastanka:
 - dogovor oko opsega generičkih funkcionalnosti za prvu predaju
 - raspored poslova
 - kratki prolazak kroz dokumentaciju

11. sastanak

- Datum: 7. studenog 2020.
- Prisustvovali: cijeli PreljevStoga
- Teme sastanka:
 - priprema za prvu predaju

12. sastanak

- Datum: 13. prosinca 2020.
- Prisustvovali: cijeli PreljevStoga
- Teme sastanka:
 - raspored poslova nakon prve predaje

13. sastanak

- Datum: 23. prosinca 2020.
- Prisustvovali: cijeli PreljevStoga
- Teme sastanka:
 - komunikacija poslužitelja i androida

14. sastanak

- Datum: 28. prosinca 2020.
- Prisustvovali: cijeli PreljevStoga
- Teme sastanka:
 - prikaz podataka na Androidu

15. sastanak

- Datum: 2. siječnja 2021.
- Prisustvovali: cijeli PreljevStoga
- Teme sastanka:
 - implementacija prijave pomoću Google računa

16. sastanak

- Datum: 8. siječnja 2021.
- Prisustvovali: cijeli PreljevStoga

- Teme sastanka:
 - priprema za testiranje

17. sastanak

- Datum: 12. siječnja 2021.
- Prisustvovali: cijeli PreljevStoga
- Teme sastanka:
 - priprema za drugu predaju

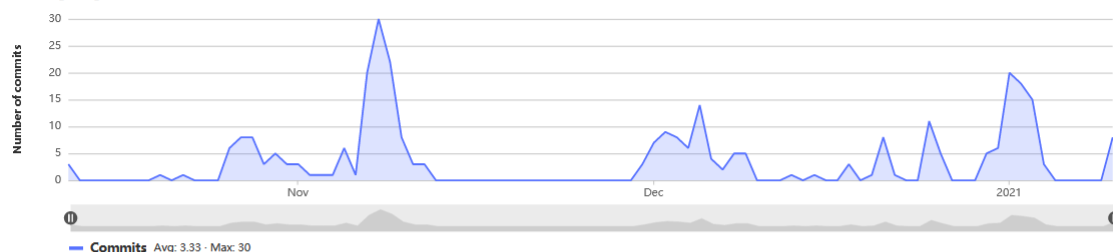
Tablica aktivnosti

	Tomislav Bjelčić	Dominik Brdar	Damjan Grubelić	Josip Komljenović	Antonio Lakoš	Luka Pranjic	Marko Vladić
Upravljanje projektom	5		3	2	25	3	
Opis projektnog zadatka	6	1	3		12	5	2
Funkcionalni zahtjevi	5	1	2	3	2	3	2
Opis pojedinih obrazaca			24		3		
Dijagram obrazaca		1	5		3	2	
Sekvencijski dijagrami		2	4				
Opis ostalih zahtjeva		2	1		2		1
Arhitektura i dizajn sustava	2	2	8		15	4	2
Baza podataka		3	4	5	14		
Dijagram razreda						4	2
Dijagram stanja					1		
Dijagram aktivnosti					1		
Dijagram komponenti					1		
Korištene tehnologije i alati	4	4			1	7	2
Ispitivanje programskog rješenja	5	12	4	36	45	18	16
Dijagram razmještaja					1		
Upute za puštanje u pogon		2	4		1		
Dnevnik sastajanja	3	1	2		1		1
Zaključak i budući rad						6	
Popis literature			1				
Demonstracija nove tehnologije			3		1		
Back end		13	2	30	95	2	18
Dizajn sučelja	8	7	4	1	2	6	5
Front end	19	13	24	3	15	21	23

Dijagrami pregleda promjena

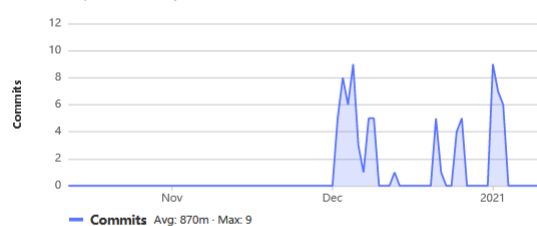
Commits to master

Excluding merge commits. Limited to 6,000 commits.



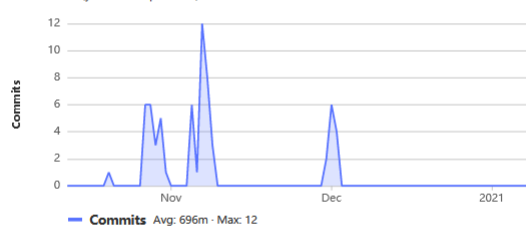
Antonio

80 commits (al51355@fer.hr)



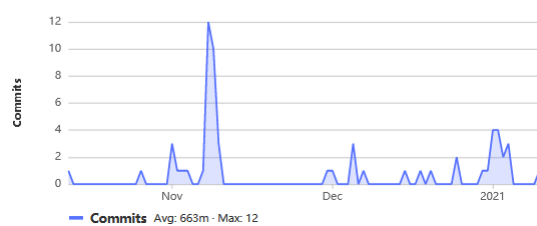
Antonio

64 commits (you@example.com)



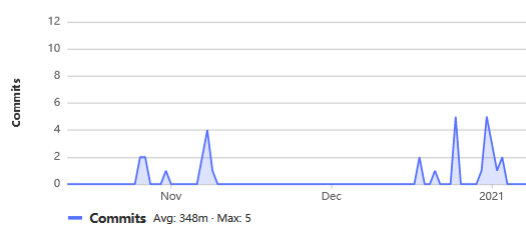
MightyJosip

61 commits (josip.komljenovic@fer.hr)



PajserDivizija

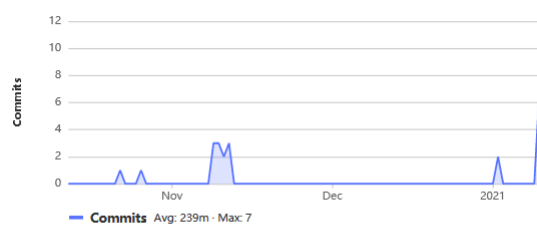
32 commits (luka.pranjic@fer.hr)



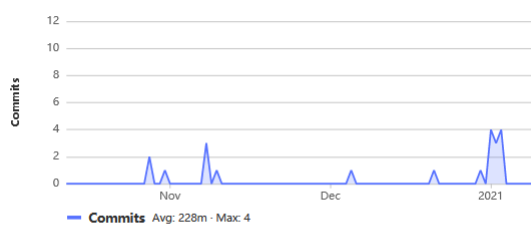
Slika 6.1: Dijagram pregleda promjena 1

Damjan Grubelic

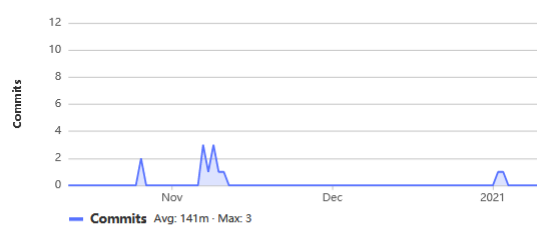
22 commits (damjan.grubelic@fer.hr)

**marko.vladic**

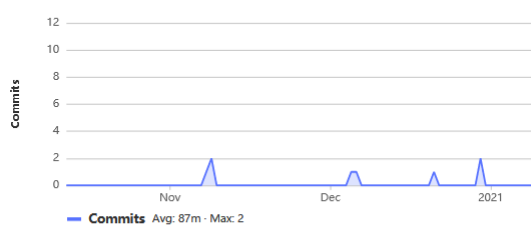
21 commits (marko.vladic@fer.hr)

**grubelic**

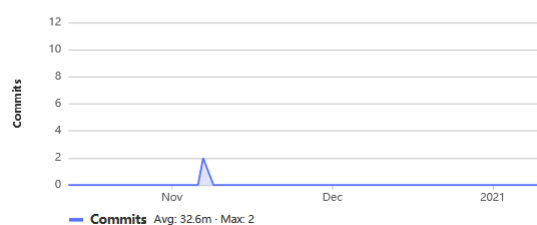
13 commits (damjangrubelic@gmail.com)

**Tomislav Bjelčić**

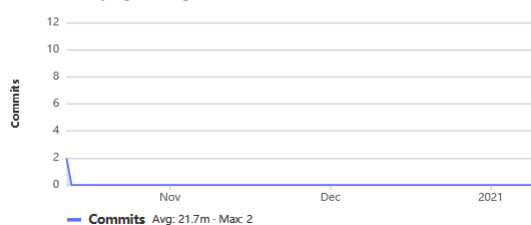
8 commits (tomislav.bjelcic@fer.hr)

**DominikBrdar**

3 commits (db51407@fer.hr)

**Damjan Grubelic**

2 commits (damjan.grubelic@gmail.com)



Slika 6.2: Dijagram pregleda promjena 2