

**Oblikovanje programske potpore
Ak. god. 2018./2019.**

***Praćenje troškova
Dokumentacija, Rev. 2***

Grupa: *grupaprekoveze*
Voditelj: *Niko Palić*

Datum predaje: *17. siječnja 2019.*

Mentor: *Ivana Perčinlić*

Članovi tima:

Niko Palić – voditelj, upravljanje projektom i kvalitetom, izrada baze podataka, izrada UML dijagrama i dokumentacije

Fabian Leskovec – nadzor dokumentacije, izrada UML dijagrama i dokumentacije

Dominik Marjanović – back-end i arhitektura, izrada UML dijagrama i dokumentacije

Leo Miloš – back-end i arhitektura, izrada UML dijagrama i dokumentacije

Karlo Jagar – front-end, izrada UML dijagrama i dokumentacije

Branimir Pehar – front-end, izrada UML dijagrama i dokumentacije

Sadržaj

1.	Dnevnik promjena dokumentacije	4
2.	Opis projektnog zadatka	5
3.	Pojmovnik	7
4.	Funkcionalni zahtjevi.....	8
5.	Ostali zahtjevi.....	10
6.	Arhitektura i dizajn sustava	43
6.1.	Svrha, opći prioriteti i skica sustava.....	44
6.2.	Dijagram razreda s opisom.....	48
6.3.	Dijagram objekata.....	52
6.4.	Ostali UML dijagrami	53
7.	Implementacija i korisničko sučelje	59
7.1.	Dijagram razmještaja	59
7.2.	Korištene tehnologije i alati.....	60
7.3.	Isječak programskog koda vezan za temeljnu funkcionalnost sustava	61
7.4.	Ispitivanje programskog rješenja	66
7.5.	Upute za instalaciju	76
7.6.	Korisničke upute	79
8.	Zaključak i budući rad	91
9.	Popis literature	92
	Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda).....	93
	Dodatak B: Dnevnik sastajanja.....	95
	Dodatak C: Prikaz aktivnosti grupe.....	96
	Dodatak D: Plan rada / Pregled rada i stanje ostvarenja	98

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodataka	Autor(i)	Datum
0.1	Preuzet predložak	Niko	23.10.2018
0.15	Dodani UC dijagrami i opisi	Fabian	24.10.2018
0.2	Izrađena shema baze	Niko	26.10.2018
0.25	Definirani funkcionalni zahtjevi	Karlo	3.11.2018
0.3	Dodani UC dijagrami i opisi	Branimir	5.11.2018
0.35	Dodani UC dijagrami i opisi	Fabian	5.11.2018
0.4	Dodani UC dijagrami i opisi	Niko	7.11.2018
0.5	Dodani sekvensijski dijagrami i opisi	Fabian, Niko	15.11.2018
0.6	Dodani opis projekta	Niko	21.11.2018
0.65	Popravljena dokumentacija	Fabian	24.11.2018
0.8	Dodani dijagrami razreda i te njihovi opisi	Leo, Dominik	26.11.2018
0.85	Dodan dijagram objekata	Karlo	27.11.2018
0.9	Upotpunjena i provjerena dokumentacija	Niko	28.11.2018
1.0	Dovršena dokumentacija (verzija 1)	svi	28.11.2018
1.1	Izmjene i prepravke	Niko	2.12.2018
1.2	Izmjene i prepravke	Niko	3.12.2018
1.3	Dijagrami stanja, komunikacije i aktivnosti	Fabian	5.1.2019
1.4	Prepravke dijagrama	Niko	13.1.2019
1.5	Izmjene i prepravke	Niko	14.1.2019
1.8	Izmjene i prepravke	Fabian, Niko	16.1.2019
1.9	Dijagram komponenti	Niko	17.1.2019
2.0	Dovršena dokumentacija (verzija 2)	Fabian, Niko	17.1.2019

2. Opis projektnog zadatka

Od režija i raznolikih poreza do svakodnevnih izdataka na kavu, danas je čovjek suočen s konstantno nadolazećim troškovima koji su toliko česti da ih je naporno i nepraktično ručno zapisivati i računati. Kako je održavanje što pozitivnije bilance prihoda i troškova implicitna zadaća svakog odgovornog čovjeka, nazire se potreba za automatiziranim sustavom koji je u stanju olakšati proces praćenja tih zarada i izdataka te ljudima bitno umanjiti nepotrebnu potrošnju vremena i energije na taj posao.

Cilj projekta je razvoj korisne, stabilne, efikasne i lako uporabljive web aplikacije za praćenje troškova. Konačni produkt će omogućiti korisnicima s minimalnim znanjem uporabe internetskih aplikacija da nadziru svoju potrošnju. Ideja aplikacije je ta da korisnik sam unosi koliko je, kada i na što potrošio te da unosi svoje prihode i zaradu. Aplikacija na temelju tih podataka za korisnika izračunava bilancu njegove blagajne. Korisnost ove aplikacije leži u svim proširenjima i funkcionalnostima nadovezanim na tu osnovnu ulogu sustava.



Slika 2.1 Praćenje troškova

Ciljani korisnici su sve punoljetne osobe sa svakodnevnim pristupom računalu i internetu.

Korisnik se može registrirati u sustav uporabom legitimnog e-maila te se s njim i s lozinkom prijaviti. Pri unosima podataka, prihoda i troškova, korisnik može unijeti točno onu količinu novaca koju želi, a sustav će automatski unosu zabilježiti datum i osvježiti stanje blagajne.

Kako bi korisnikove slučajne pogreške bile lako ispraviti, sustav omogućava i izmjenu te brisanje prijašnjih unosa uz sve potrebne popratne akcije.

Korisnički profil će biti u potpunosti izmjenjiv, a sadržavat će ime i prezime te e-mail, kao i lozinku korisnika.

Sustav će pratiti stanje korisnikove blegajne i upozoravati ga o mogućoj pretjeranoj potrošnji i konzultirati ga o tome gdje bi trebao početi štedjeti. Nadalje korisnik može pregledati statistiku o svojim izdacima i primanjima. Može personalizirati svoje unose pridavanjem proizvoljnih opisa i kategorija uz prihode i troškove. U početku će korisnik imati i ponuđene kategorije tako da ih ne mora sam unositi ako ne želi biti previše detaljan.

Da bi se korisniku olakšalo postići njegove financijske ciljeve i dodatno ga motiviralo, sustav omogućuje definiranje štednji, koje djeluju kao zasebna blegajna s unaprijed postavljenim ciljem kojeg se treba postići. Svaka takva štednja će imati prikaz koliko je ostalo do cilja.

Posebno korisna funkcija sustava jest stvaranje zajedničkih štednji, između više ljudi dijeljenih financija, kojima više korisnika može zajedno ulagati prema nekome cilju. Korisnik može pozvati bilo koju registriranu osobu u blegajnu, dok pozvana osoba može odbiti ili prihvati poziv i biti priključena štednji.

Sustav bi se u budućnosti mogao nadograditi automatiziranim unošenjem troškova i prihoda na temelju statistike (npr. ako se svaki dan na kavu potroši 20 kuna sustav bi mogao to primijetiti i početi sam unositi te troškove) ili praćenjem kreditnih kartica i on-line transakcija. Ovaj osnovni produkt bit će napravljen tako da buduća unaprjeđenja budu što izvedivija.

3. Pojmovnik

Astah Pri izradi svih potrebnih UML dijagrama korišten je Astah Community.

Bootstrap je alat koji nam omogućuje lakši rad s HTML-om, CSS-om i JavaScript-om.

CSS je stilski jezik, upotrebljava se za opis prezentacije dokumenta napisanog pomoću markup (HTML) jezika. Korišten je za prikaz aplikacije u web okruženju.

Eclipse je programska razvojna okolina (IDE) pisana u Javi, a može se koristiti za razvoj aplikacija u raznim programskim jezicima (Java, C,C++,PHP...). Eclipse koristi dodatke za pružanje svih funkcija unutar i na vrhu sustava za pokretanje. Njegov radni vremenski sustav temelji se na Equinoxu, implementaciji OSGi jezgrene specifikacije okvira.

Git je sustav za upravljanje inačicama koji omogućuje lakši timski rad i sustavno praćenje koda. Git je uparen s Gitlab središnjim repozitorijem. Također razvojna okolina koju smo odabrali (Eclipse IDE) omogućuje integraciju Git sustava prilikom izrade programskog koda web aplikacije.

Herokuapp Za izvedbu web poslužitelja korišten je Heroku web poslužitelj.

HTML (HyperText Markup Language) je prezentacijski jezik za izradu web stranica. Korišten je za prikaz aplikacije u web okruženju.

HTTP Za komunikaciju između aplikacije i baze podataka se koriste HTTP(Hypertext Transfer Protocol) zahtjevi.

JavaScript je skriptni programski jezik, koji se izvršava u web pregledniku na strani korisnika. Napravljen je da bude sličan Javi, zbog lakšega korištenja, ali nije objektno orijentiran kao Java, već se temelji na prototipu i tu prestaje svaka povezanost s programskim jezikom Java.

Spring Framework je okvir aplikacije i inverzija kontrolnog spremnika za Java platformu. Osnovne značajke okvira mogu se koristiti s bilo kojom Java aplikacijom, ali postoje i proširenja za izgradnju web aplikacija na platformi Java EE (Enterprise Edition). Iako okvir ne nameće bilo koji specifičan programski model, postao je popularan u Java zajednici.

Thymeleaf je Java XML (Extensible Markup Language) / XHTML / HTML5 predložak koji može raditi na webu i ne-web okruženjima. Pogodnije je za posluživanje XHTML / HTML5 na sloju prikaza MVC web aplikacija, ali može obradivati bilo koju XML datoteku čak i u izvanmrežnim okruženjima. Pruža cjelovitu integraciju Spring Framework-a.

4. Funkcionalni zahtjevi

DIONICI:

- Korisnik

KORISNIČKI FUNKCIONALNI ZAHTJEVI

• Korisnik

- Mogućnost registracije te prijave na sustav pomoću e-mail adrese i odabrane lozinke
- Mogućnost uređivanja vlastitog profila – mijenjanje osobnih podataka, lozinke, imena, prezimena
- Raspolaže jednim računom (blagajnom) i može stvoriti proizvoljan broj štednji
- Može pregledavati podatke o stanju vlastitog računa, štednji te prebacivati novac na iste
- Može odabratи predložene kategorije prihoda i troškova te unijeti proizvoljne
- Mogućnost unosa, izmjene i brisanja prihoda i troškova na računu
- Može postaviti cilj štednje, tj. sumu novca koju je potrebno dostići kao i opis štednje
- Može stvoriti više zajedničkih blagajni s više korisnika u kojima transparentno rukuje zajedničkim financijama
- Može zvati korisnike u zajedničke blagajne i biti pozvan u njih
- Može izaći iz zajedničkih blagajni
- Biva obaviješten u slučaju otkrivenog prekomjernog trošenja
- Ima na raspolaganju statistike o unesenim prihodima i troškovima

Baza podataka

Pohranjuje podatke o svim registriranim korisnicima (imena, prezimena, lozinke, korisnička imena, e-mailovi), njihovim blagajnama i štednjama, unosima prihoda i troškova

OPIS OBRAZACA UPORABE:

- **UC1 – Registracija**
 - Glavni sudionik: Korisnik
 - Cilj: Izrada novog korisničkog računa
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: legitimni e-mail
 - Željeni scenarij:
 - I. Korisnik odabire opciju izrade novog računa
 - II. Korisnik unosi tražene podatke, te pritiskom na gumb šalje zahtjev za obradu traženih podataka
 - III. Web aplikacija provjerava valjanost unesenih podataka
 - IV. Ako su uneseni podaci ispravni, Web aplikacija upisuje dane podatke u bazu podataka te na adresu električne pošte korisnika šalje poruku s verifikacijskom poveznicom
 - V. Potvrda registracije na e-mailu omogućuje stvaranje novog računa
 - Mogući scenarij:
 - I. Registracija nije moguća zbog unosa već registrirane ili nepostojeće adrese električne pošte ili zbog krivog formata unesenih podataka
 - II. Aplikacija obavještava korisnika o neuspjelom upisu te ga vraća na stranicu za izradu novog računa
- **UC2 – Prijava**
 - Glavni sudionik: Korisnik
 - Cilj: Pristup korisničkom sučelju
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik mora biti registriran
 - Željeni scenarij:
 - I. Korisnik unosi username i zaporku, te pritiskom na gumb šalje zahtjev za obradu traženih podataka
 - II. Web aplikacija provjerava ispravnost danih podataka
 - III. Prijava u sustav uspješna
 - Mogući scenarij:
 - I. Korisnik je unio netočne podatke
 - II. Prijava u sustav neuspješna, aplikacija obavještava korisnika o neuspjehu te ga vraća na stranicu za prijavu
- **UC3 – Odjava**
 - Glavni sudionik: Korisnik
 - Cilj: Odjava iz sustava
 - Ostali sudionici: Web aplikacija, baza podataka

- Preduvjet: Korisnik mora biti prijavljen
- Željeni scenarij:
 - I. Korisnik pritišće na gumb za odjavu
 - II. Aplikacija uspješno odjavljuje korisnika
- **UC4 – Pogledaj info zajedničke štednje**
 - Glavni sudionik: Korisnik
 - Cilj: Odjava iz sustava
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik mora biti prijavljen, imati zaj. štednju, te biti unutar stranice za prikaz zaj. štednji
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za pregled informacije zaj. štednje
 - II. Aplikacija prikazuje sadržaj dnevnika štednje i popis sudionika
- **UC5 – Uređivanje i pregled profila**
 - Sudionik: Korisnik
 - Cilj: Izmjena i pregled osobnih podataka
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „profil“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za promjenu vrijednosti osobnih podataka
 - II. Aplikacija upisuje izmijenjene podatke u bazu podataka
 - III. Aplikacija šalje potvrdu o uspješnoj promjeni vrijednosti
 - Mogući scenarij:
 - I. Spremanje neuspješno zbog krivog formata podataka
 - II. Korisnik je obaviješten o neuspjehu
- **UC6 – Stvaranje zasebnih kategorija**
 - Glavni sudionik: Korisnik
 - Cilj: Odabir individualnih kategorija
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „kategorije“
 - Željeni scenarij:
 - I. Korisnik odabire opciju izrade i dodavanja individualnih kategorija
 - II. Aplikacija dodaje individualne kategorije u bazu podataka uz već postojeće kategorije na web-stranici
 - III. Aplikacija šalje potvrdu o uspješnom dodavanju
 - Mogući scenarij:
 - I. Dodavanje neuspješno jer željena kategorija već postoji ili zbog maksimalnog broja mogućih kategorija

II. Korisnik je obaviješten o neuspjehu

- **UC7 – Stvaranje štednje**
 - Glavni sudionik: Korisnik
 - Cilj: Prebacivanje određene količine neiskorištenog novca u štednju
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „štednje“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za stvaranje štednje
 - II. Korisnik unosi željeni cilj štednje
 - Mogući scenarij:
 - I. Cilj štednje neispravno unesen
 - II. Korisnik se obavještava o neuspjehu
- **UC8 - Pregled kategorija**
 - Glavni sudionici: Korisnik
 - Cilj: Prikaz svih kategorija koje su vidljive korisniku
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „kategorije“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za pregled kategorija
 - II. Aplikacija dohvata i prikazuje korisnikove kategorije
- **UC9 – Unos prihoda**
 - Glavni sudionik: Korisnik
 - Cilj: Unos prihoda u blagajnu
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „prihodi“
 - Željeni scenarij:
 - I. Korisnik unosi željeni iznos prihoda
 - II. Aplikacija upisuje prihod i pohranjuje izmjene u bazu podataka
 - Mogući scenarij:
 - I. Spremanje neuspješno zbog krivog formata podataka
 - II. Korisnik je obaviješten o neuspjehu
- **UC10- Unos troškova**
 - Glavni sudionik: Korisnik
 - Cilj: Unos troškova u blagajnu
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „troškovi“
 - Željeni scenarij:
 - I. Korisnik unosi željeni iznos troškova

- II. Aplikacija upisuje trošak i pohranjuje izmjene u bazu podataka
 - Mogući scenarij:
 - I. Spremanje neuspješno zbog krivog formata podataka
 - II. Korisnik je obaviješten o neuspjehu
- **UC11 – Brisanje prihoda**
 - Glavni sudionik: Korisnik
 - Cilj: Brisanje prihoda iz blagajne
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav te je unio svoj prihod, ušao u pregled prihoda
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za brisanje prihoda
 - II. Aplikacija briše prihod korisnikove blagajne iz baze podataka
 - III. Stanje računa se umanjuje za iznos obrisanog prihoda
- **UC12 – Brisanje troškova**
 - Glavni sudionik: Korisnik
 - Cilj: Brisanje troškova iz blagajne
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav te je unio svoje troškove, ušao u pregled troškova
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za brisanje troškova
 - II. Aplikacija briše trošak korisnikove blagajne iz baze podataka
 - III. Stanje računa se uvećava za iznos obrisanog troška
- **UC13 – Pregled troškova**
 - Glavni sudionik: Korisnik
 - Cilj: Pregled korisničkih troškova
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za pregled troškova
 - II. Aplikacija dohvata prikaz troškova iz baze
- **UC14 – Pregled prihoda**
 - Glavni sudionik: Korisnik

- Cilj: Pregled prihoda blagajne
- Ostali sudionici: Web aplikacija, baza podataka
- Preduvjet: Korisnik je prijavljen u sustav te je unio svoj prihod
- Željeni scenarij:
 - I. Korisnik pritišće na gumb za pregled prihoda
 - II. Aplikacija dohvata prikaz prihoda iz baze
- **UC15 – Kreiranje zajedničke štednje**
 - Glavni sudionik: Korisnik
 - Cilj: Kreiranje zajedničke štednje više korisnika
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „zajedničke štednje“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za kreiranje zajedničke štednje
 - II. Aplikacija stvara zajedničku štednju te promjene pohranjuje u bazu podataka
 - Mogući scenarij:
 - I. Stvaranje zajedničke štednje neuspješno zbog nepostojanja pridodanih članova
 - II. Korisnik je obaviješten o neuspjehu
- **UC16 – Prijenos s blagajne na štednju**
 - Glavni sudionik: Korisnik
 - Cilj: Prijenos željenog iznosa iz blagajne u štednju
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav i ima neiskorišteni novac, kliknuo „štедnje“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za uređivanje štednje
 - II. Korisnik odabire iznos za prijenos
 - III. Aplikacija prebacuje i zapisuje iznos u bazu podataka
 - Mogući scenarij:
 - I. Korisnik raspolaze nedostatnim financijama
 - II. Korisnik je obaviješten o neuspjehu
- **UC17 – Izmjeni štednju**
 - Glavni sudionik: Korisnik
 - Cilj: Promjena štednje
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav i ima stvorenu štednju, kliknuo „štедnje“
 - Željeni scenarij:

- I. Korisnik pritišće na gumb za izmjenu štednje te unosi željeni iznos za cilj
- II. Aplikacija izmjenjuje štednju te promjene zapisuje u bazu podataka

- Mogući scenarij:

- I. Željeni cilj neispravno unesen
- II. Korisnik je obaviješten o neuspjehu

- **UC18 – Izmijeni trošak**

- Glavni sudionik: Korisnik
- Cilj: Izmjena troška
- Ostali sudionici: Web aplikacija, baza podataka
- Preduvjet: Korisnik je prijavljen u sustav te već ima zadani iznos troška, ušao u pregled troškova
- Željeni scenarij:
 - I. Korisnik pritišće gumb za izmjenu troška te unosi željeni iznos
 - II. Aplikacija izmjenjuje trošak te promjene pohranjuje u bazu podataka
- Mogući scenarij:
 - I. Krivi format
 - II. Korisnik je obaviješten o neuspjehu

- **UC19 – Izmjeni prihod**

- Glavni sudionik: Korisnik
- Cilj: Izmjena prihoda
- Ostali sudionici: Web aplikacija, baza podataka
- Preduvjet: Korisnik je prijavljen u sustav te već ima zadani iznos troška, ušao u pregled prihoda
- Željeni scenarij:
 - I. Korisnik pritišće na gumb za izmjenu prihoda te unosi željeni prihod
 - II. Aplikacija izmjenjuje prihod te promjene pohranjuje u bazu podataka
- Mogući scenarij:
 - I. Krivi format
 - II. Korisnik je obaviješten o neuspjehu

- **UC20 – Izađi iz zajedničke štednje**

- Glavni sudionik: Korisnik
- Cilj: Izlazak iz zajedničke štednje
- Ostali sudionici: Web aplikacija, baza podataka

- Preduvjet: Korisnik je prijavljen u sustav, ima zajedničku štednju, kliknuo „zajedničke štednje“
- Željeni scenarij
 - I. Korisnik pritišće na gumb za izlazak iz zajedničke štednje
 - II. Aplikacija izbacuje korisnika iz zajedničke štednje te promjene pohranjuje u bazu podataka
 - III. Ako je korisnik bio zadnji u toj štednji, prebacuje se sav novac na njegovu blagajnu

- **UC21 – Povlačenje novaca sa štednje**

- Glavni sudionik: Korisnik
- Cilj: Povlačenje novaca iz štednje
- Ostali sudionici: Web aplikacija, baza podataka
- Preduvjet: Korisnik je prijavljen u sustav, ima štednju, kliknuo „štednje“
- Željeni scenarij:
 - I. Korisnik pritišće na gumb za povlačenje novaca iz štednje
 - II. Aplikacija povlači novac te zapisuje promjene u bazu podataka
- Mogući scenarij:
 - I. Korisnik nema odgovarajuće financije
 - II. Korisnik je obaviješten o neuspjehu

- **UC22 – Pozovi korisnika u štednju**

- Glavni sudionik: Korisnik
- Cilj: Poziv korisnika u zajedničku štednju
- Ostali sudionici: Web aplikacija, baza podataka
- Preduvjet: Korisnik je prijavljen u sustav, ima štednju, kliknuo „zajedničke štednje“
- Željeni scenarij:
 - I. Korisnik pritišće na gumb za poziv korisnika u zajedničku štednju
 - II. Aplikacija šalje zahtjev željenom korisniku te ga zapisuje u bazu podataka nakon prihvata poziva
- Mogući scenarij:
 - I. Pozvani korisnik ne postoji
 - II. Korisnik je obaviješten o neuspjehu

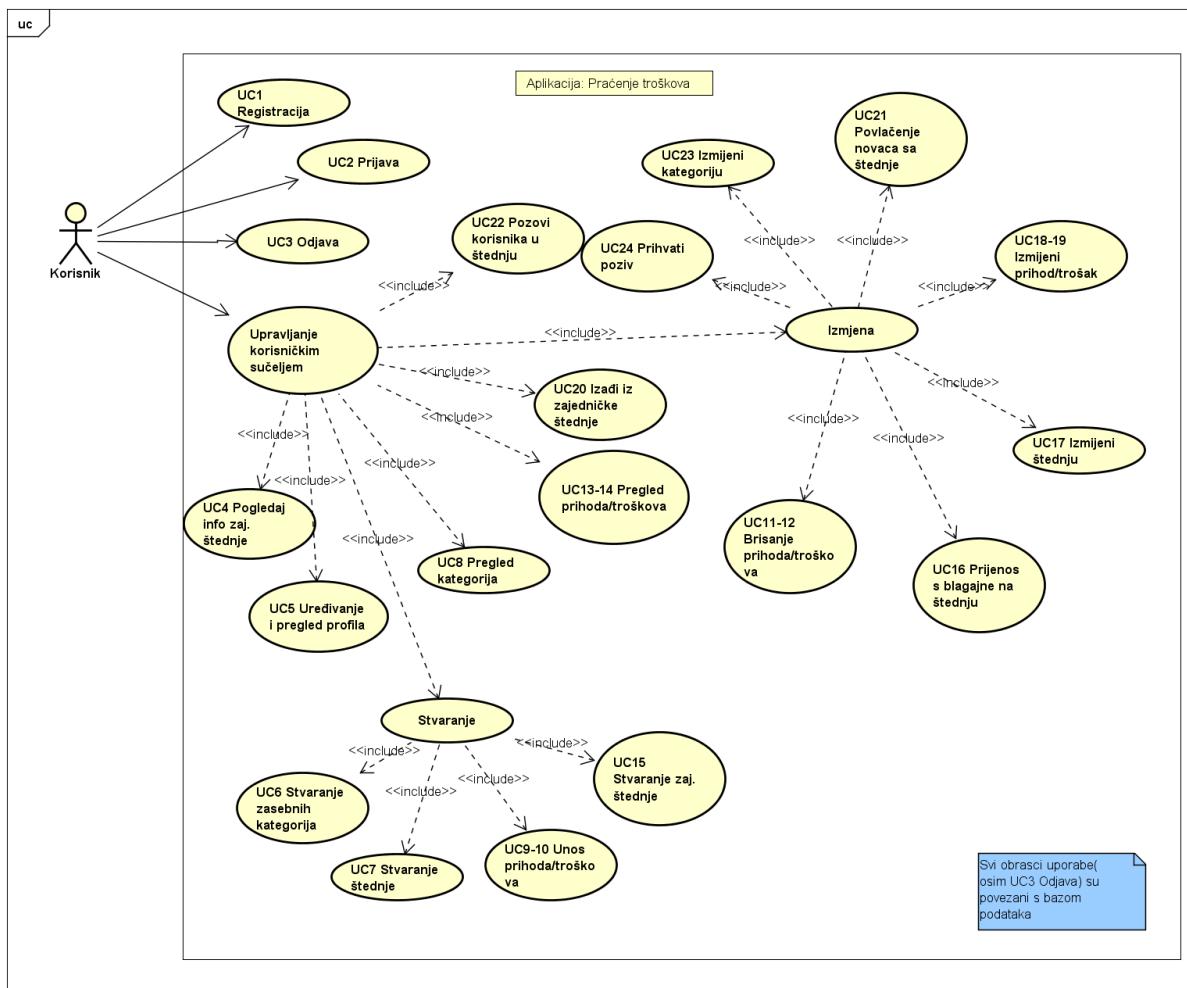
- **UC23 – Izmijeni kategoriju**

- Glavni sudionik: Korisnik
- Cilj: Izmjena kategorija
- Ostali sudionici: Web aplikacija, baza podataka
- Preduvjet: Korisnik je prijavljen u sustav te već ima stvorene zasebne kategorije, kliknuo „kategorije“

- Željeni scenarij:
 - I. Korisnik pritišće na gumb za izmjenu kategorija te unosi željene promjene
 - II. Aplikacija izmjenjuje kategoriju i pohranjuje izmjenu u bazu podataka
- Mogući scenarij:
 - I. Krivo uneseni podaci
 - II. Korisnik je obaviješten o neuspjehu

- **UC24 – Prihvati poziv**

- Glavni sudionik: Korisnik
- Cilj: Pregled statistike o financijama
- Ostali sudionici: Web aplikacija, baza podataka
- Preduvjet: Korisnik je prijavljen u sustav te je pozvan u neku zaj. štednju, ušao u pregled poziva
- Željeni scenarij:
 - I. Korisnik prihvaca poziv
 - II. Aplikacija obavlja potrebne operacije u bazi podataka
 - III. Korisnik je uspješno dodan u zaj. štednju

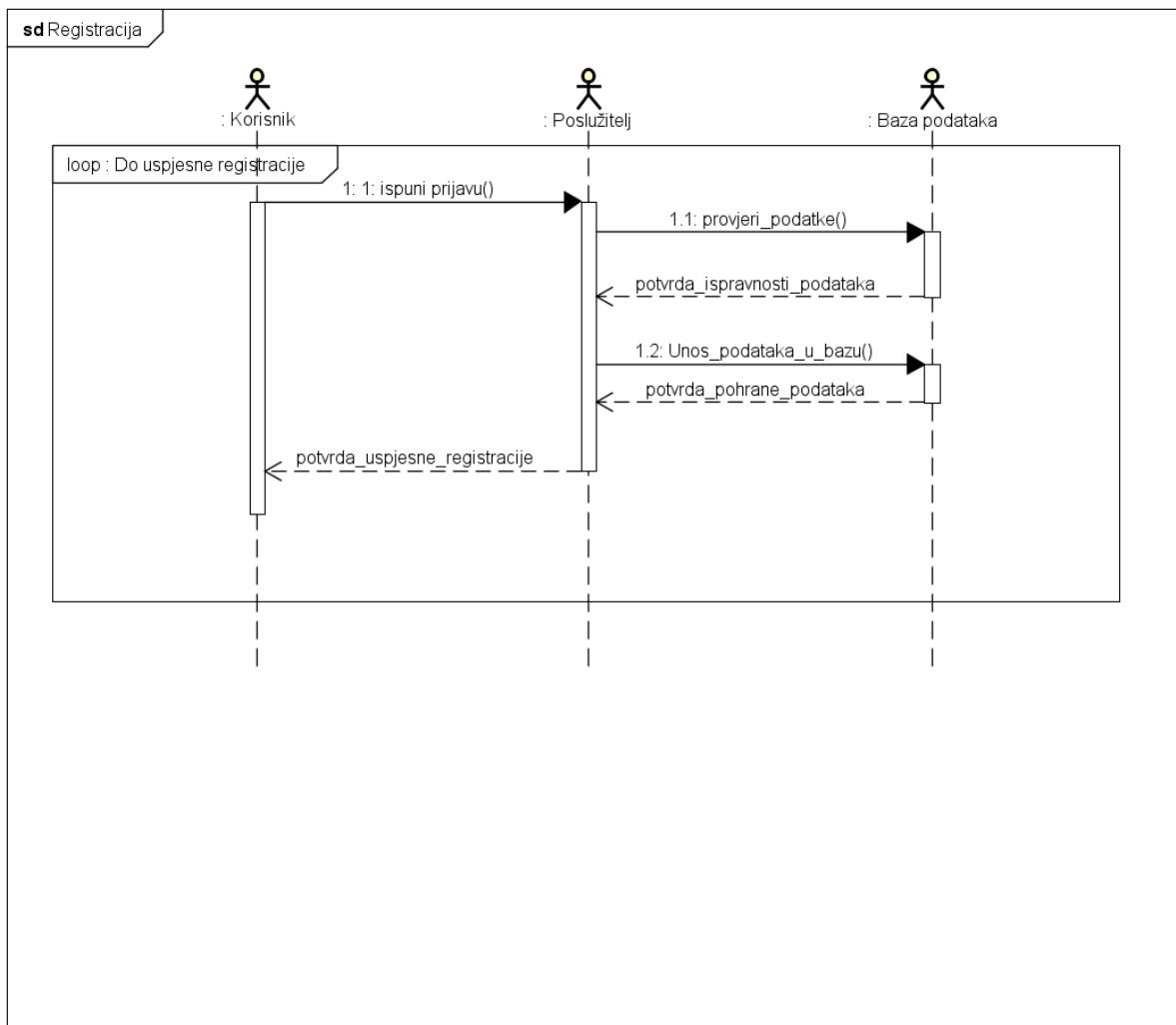


Slika 4.1.1. Dijagram obrazaca uporabe korisnika

SEKVENCIJSKI DIJAGRAMI:

Sekvencijski dijagram za UC1 (Registracija):

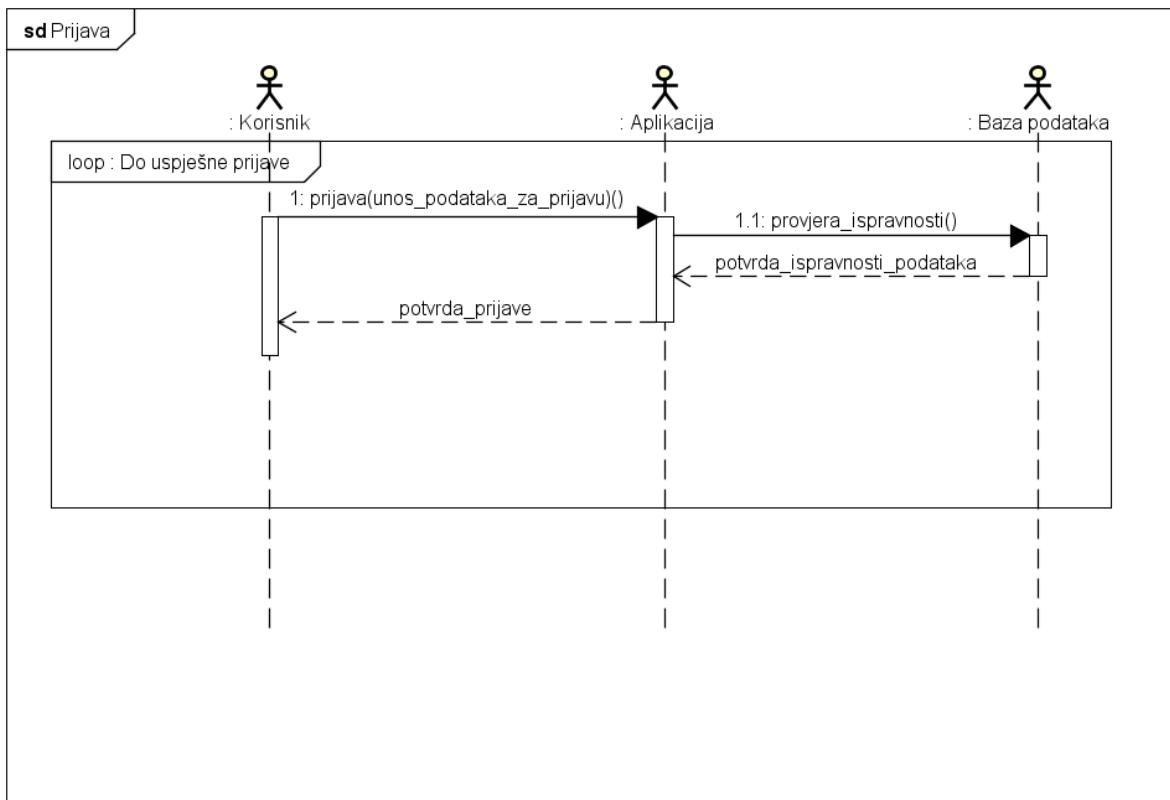
Korisnik se registrira ispunom web obrasca prijave. Njegovi podaci su poslani poslužitelju koji zajedno s bazom podataka provjerava njihovu ispravnost te da li već postoji korisnik sa istim podacima. Ukoliko je potvrđena ispravnost podataka, poslužitelj šalje bazi zahtjev za pohranu istih te će korisnik biti uspješno registriran. Poslužitelj informira korisnika o uspješnoj registraciji. U suprotnom korisnik se može opet pokušati registrirati.



Slika 4.1.2. Sekvencijski dijagram za UC1

Sekvencijski dijagram za UC2 (Prijava):

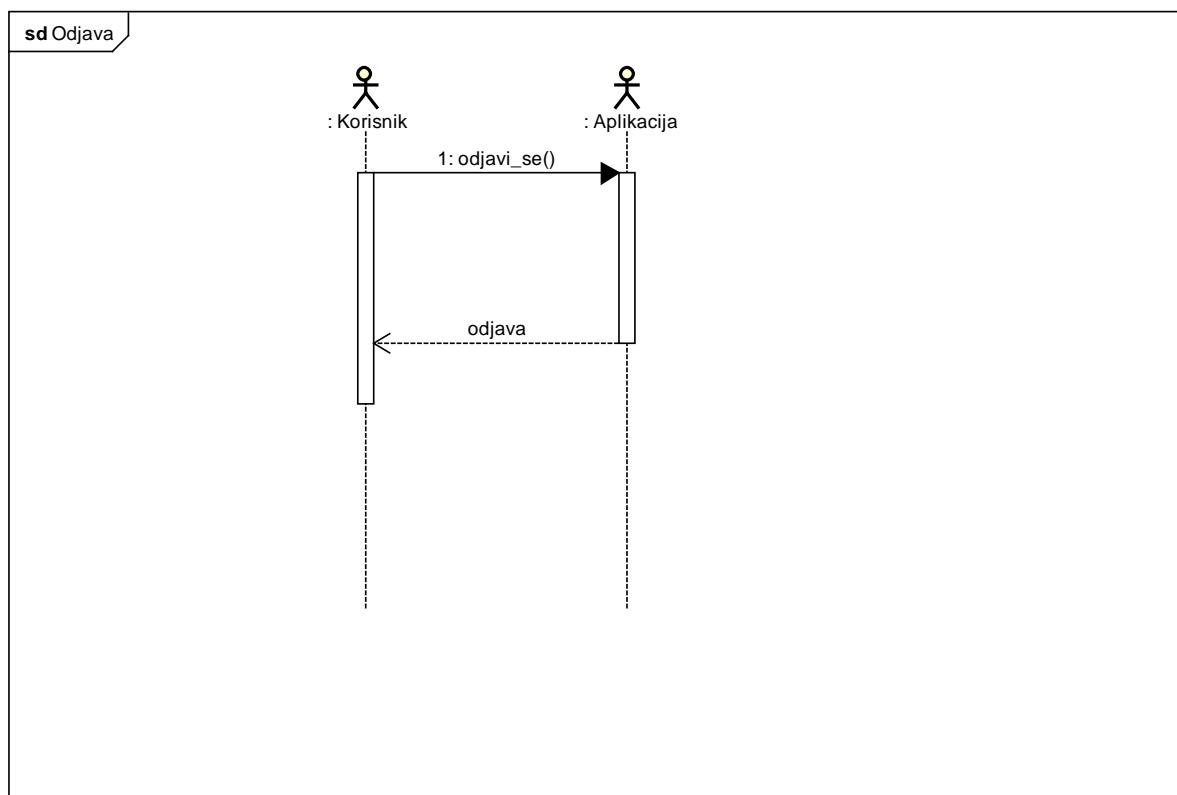
Registrirani korisnik unosi podatke za prijavu. Web aplikacija podatke šalje u bazu na provjeru. Ako su podaci ispravni korisnik se prijavljuje u sustav. U slučaju neispravnih podataka korisnik može ponovo pokušati.



Slika 4.1.3. Sekvencijski dijagram za UC2

Sekvencijski dijagram za UC3 (Odjava):

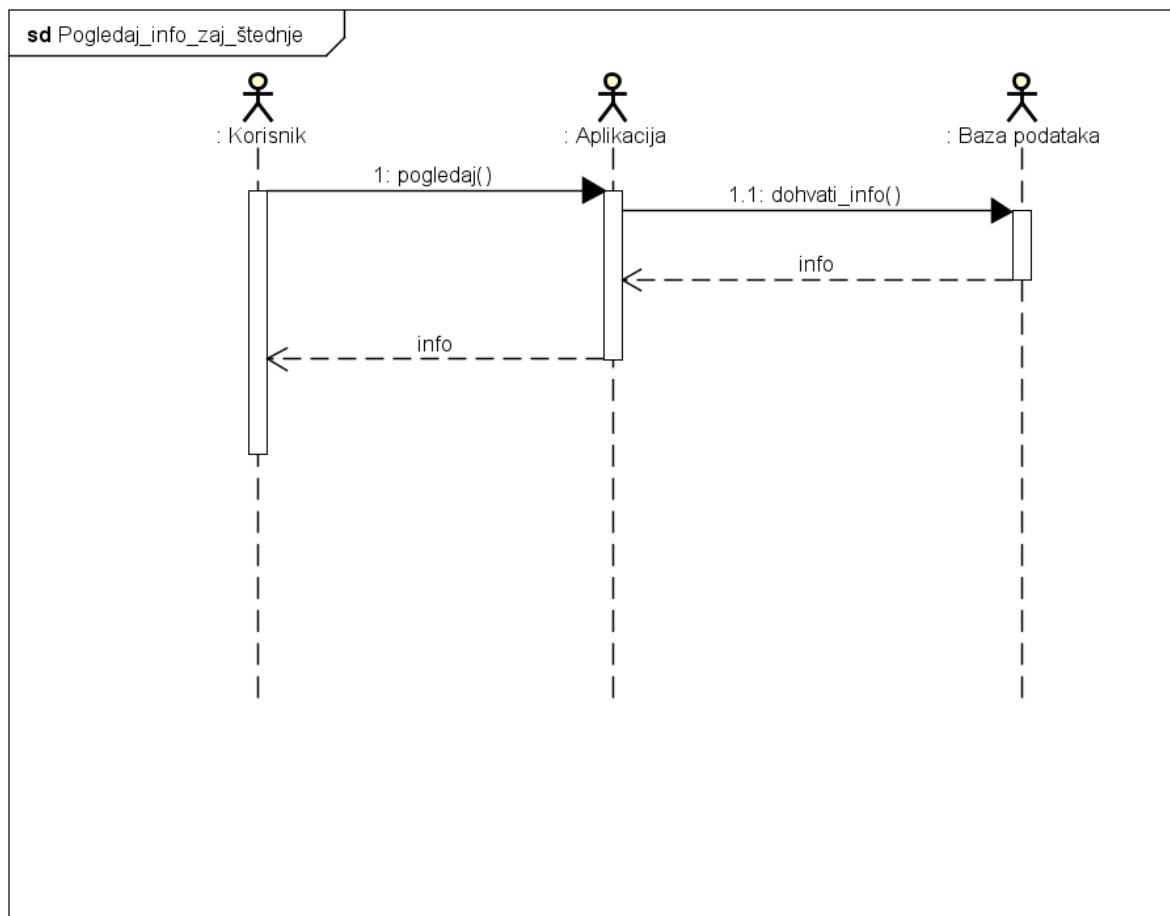
Registrirani korisnik pritiskom na gumb obavještava aplikaciju da se želi odjaviti iz sustava.



Slika 4.1.4. Sekvencijski dijagram za UC3

Sekvencijski dijagram za UC4 (Pogledaj info zajedničke štednje):

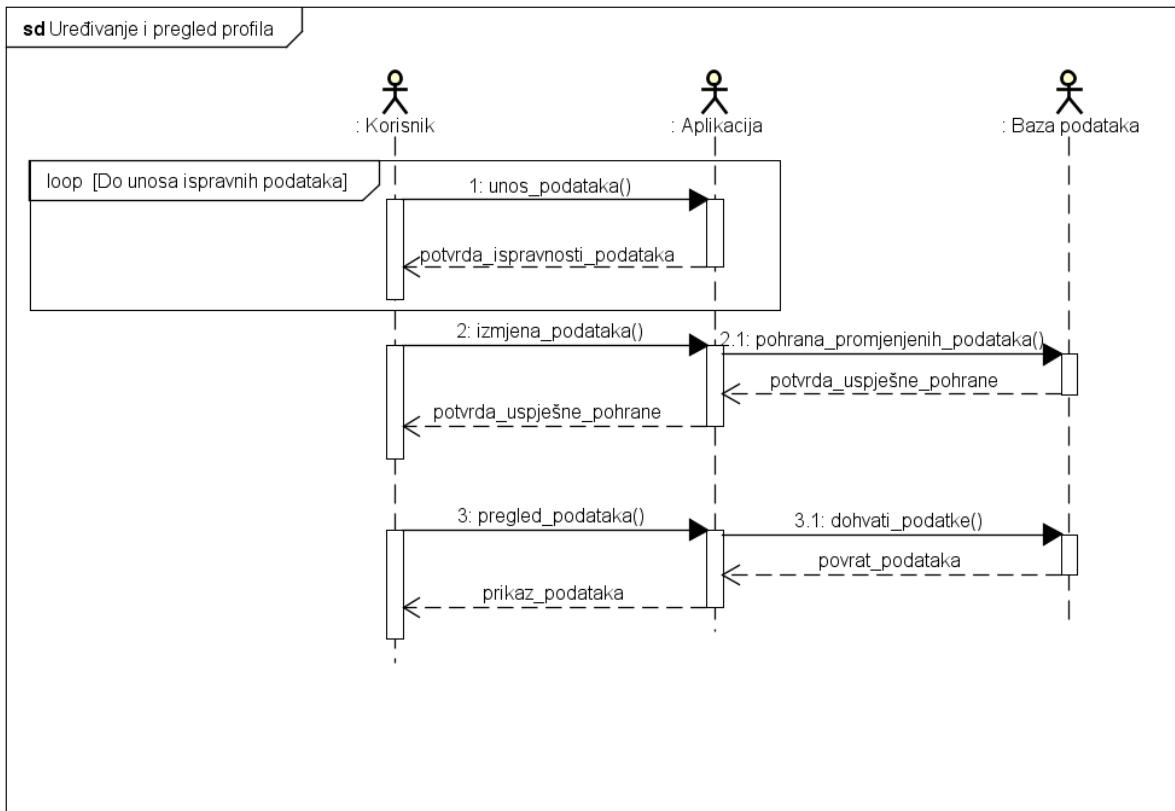
Registrirani korisnik klikće na pregled zaj. štednji i potom na gumb „info“ neke od zaj. štednji. Korisnik pregleđava informaciju o štednji, tj. log i popis korisnika.



Slika 4.1.5. Sekvencijski dijagram za UC4

Sekvencijski dijagram za UC5 (Uređivanje i pregled profila):

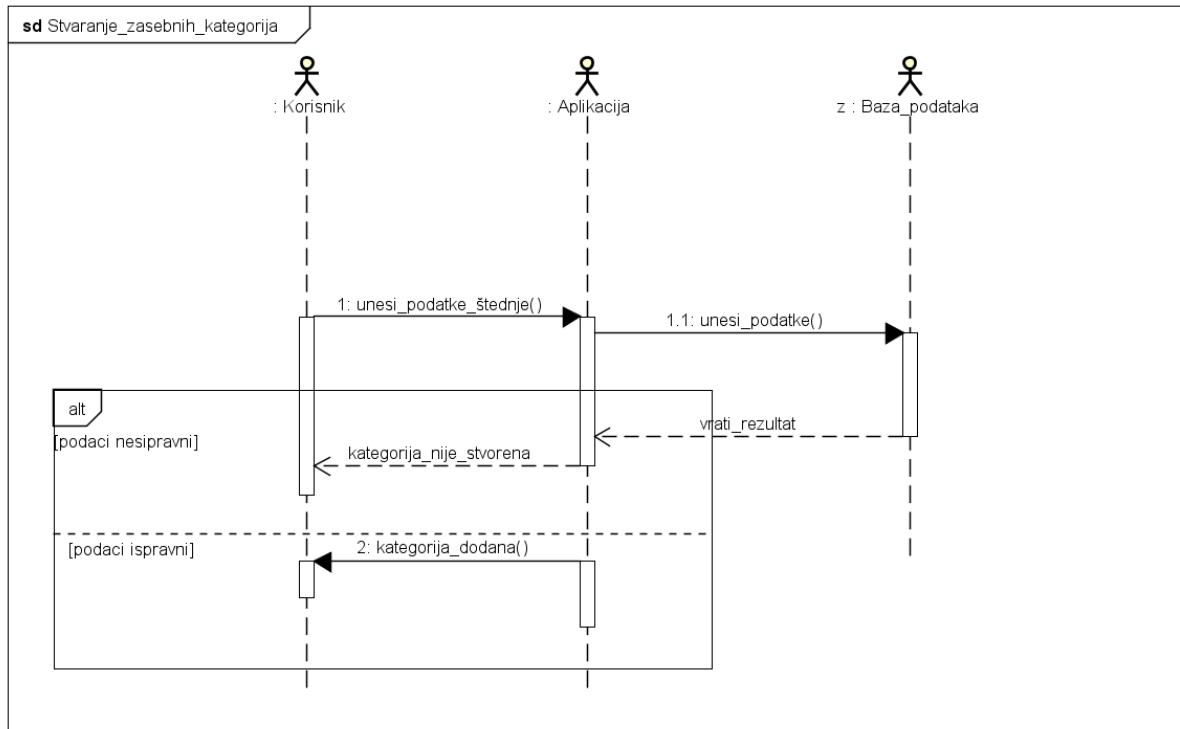
Korisnik unosi podatke sve dok nisu valjani. Poslužitelj provjerava valjanost podataka te korisniku šalje poruku o ispravnosti istih. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka te ih ona pohrani. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika. Prilikom pregledavanja profila korisnik šalje zahtjev poslužitelju te on isti proslijedi bazi podataka i od nje zatraži korisničke podatke. Baza ih zatim proslijedi poslužitelju, a poslužitelj ih prikaže korisniku.



Slika 4.1.6. Sekvencijski dijagram za UC5

Sekvencijski dijagram za UC6 (Stvaranje zasebnih kategorija):

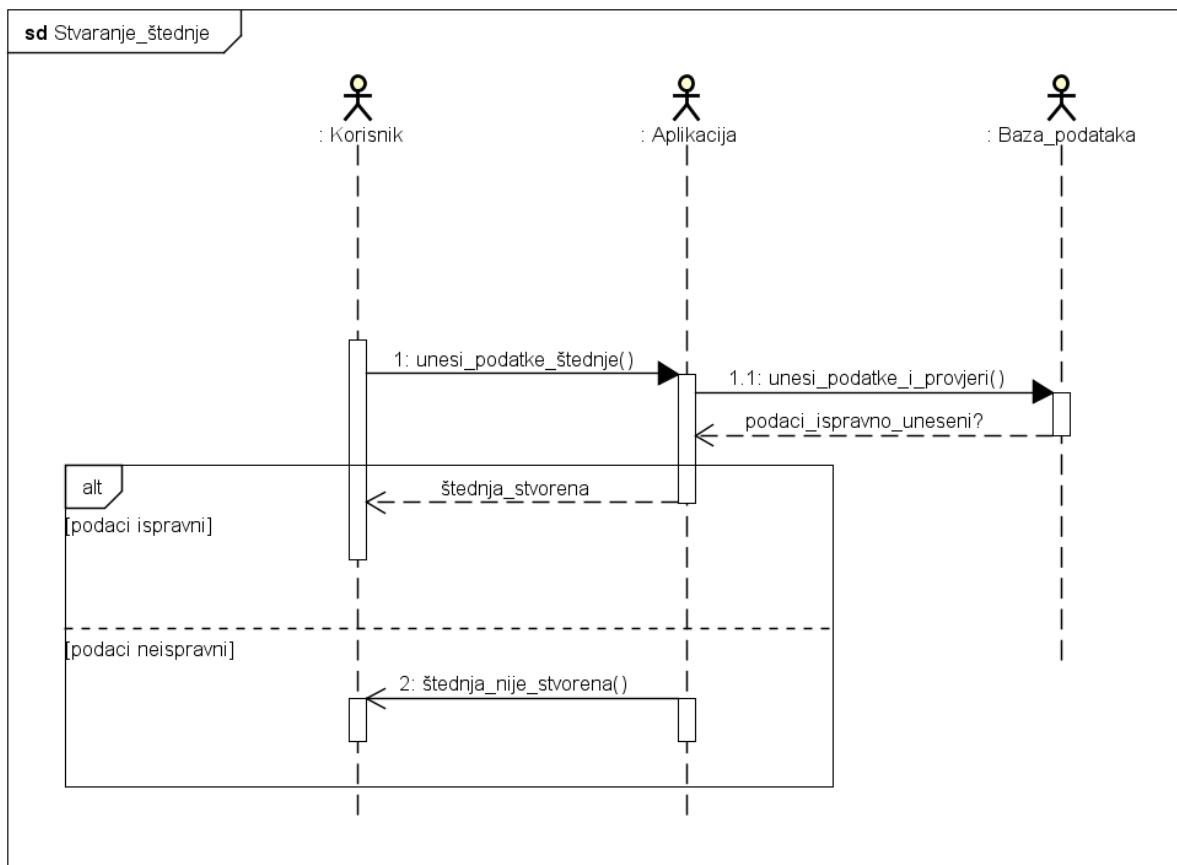
Korisnik pritišće na gumb za odabir zasebnih kategorija blagajne. Nakon unosa podataka poslužitelj ih šalje u bazu podataka te ih ona pohrani. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika. Korisniku se zatim prikažu na nadodane kategorije. U slučaju neispravnih podataka korisniku se javlja greška.



Slika 4.1.7. Sekvencijski dijagram za UC6

Sekvencijski dijagram za UC7 (Stvaranje štednje) :

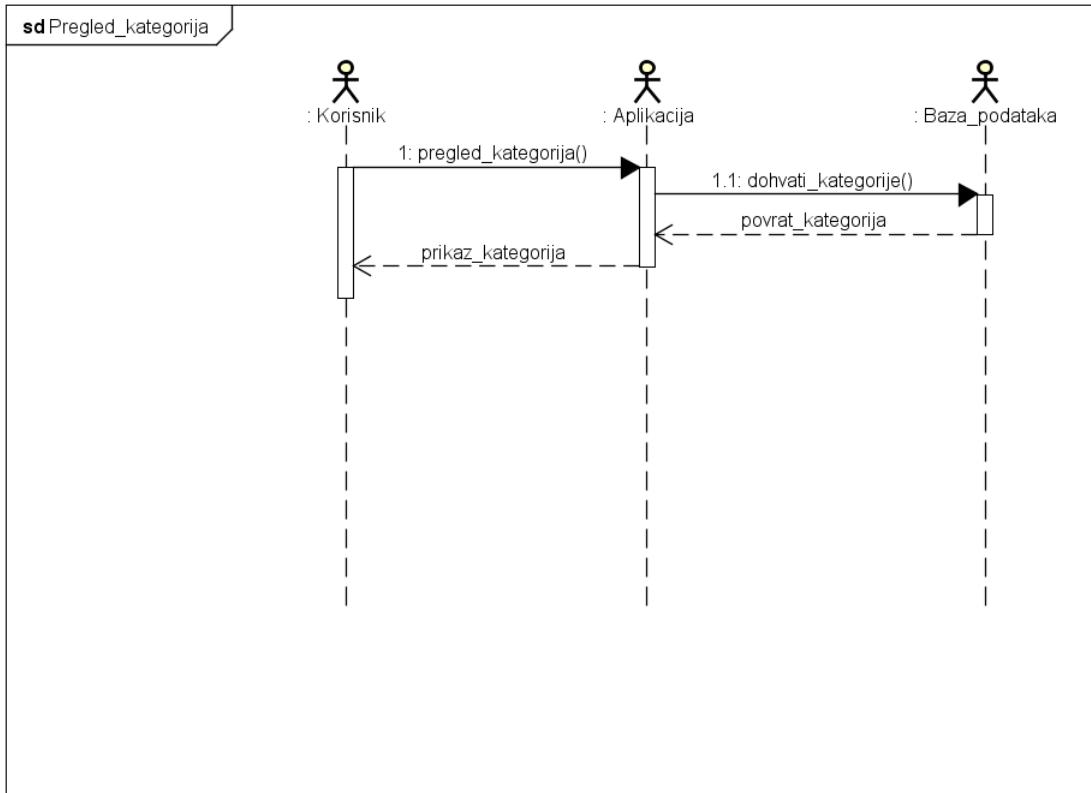
Korisnik odabire opciju stvaranja štednje. Zatim unosi tražene podatke. Poslužitelj provjerava valjanost podataka te korisniku šalje poruku o ispravnosti istih. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka te ih ona pohrani. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika. U slučaju neispravnih podataka korisniku se javlja greška.



Slika 4.1.8.Sekvencijski dijagram za UC7

Sekvencijski dijagram za UC8 (Pregled kategorija):

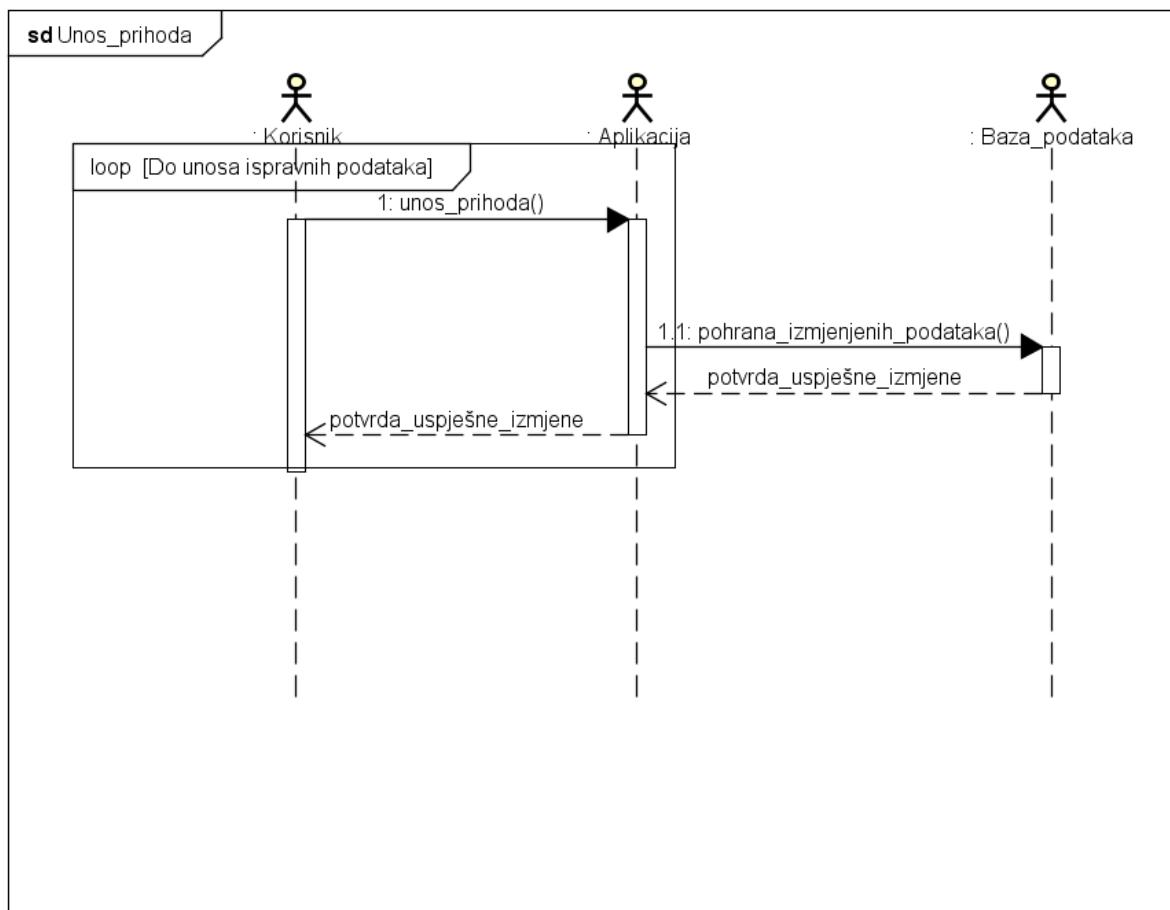
Korisnik odabire opciju pregleda kategorija. Prilikom pregledavanja kategorija korisnik šalje zahtjev poslužitelju te on isti prosljeđuje bazi podataka i od nje zatraži podatke kategorija. Baza ih zatim proslijedi poslužitelju, a poslužitelj ih prikaže korisniku.



Slika 4.1.9.Sekvencijski dijagram za UC8

Sekvencijski dijagram za UC9 (Unos prihoda):

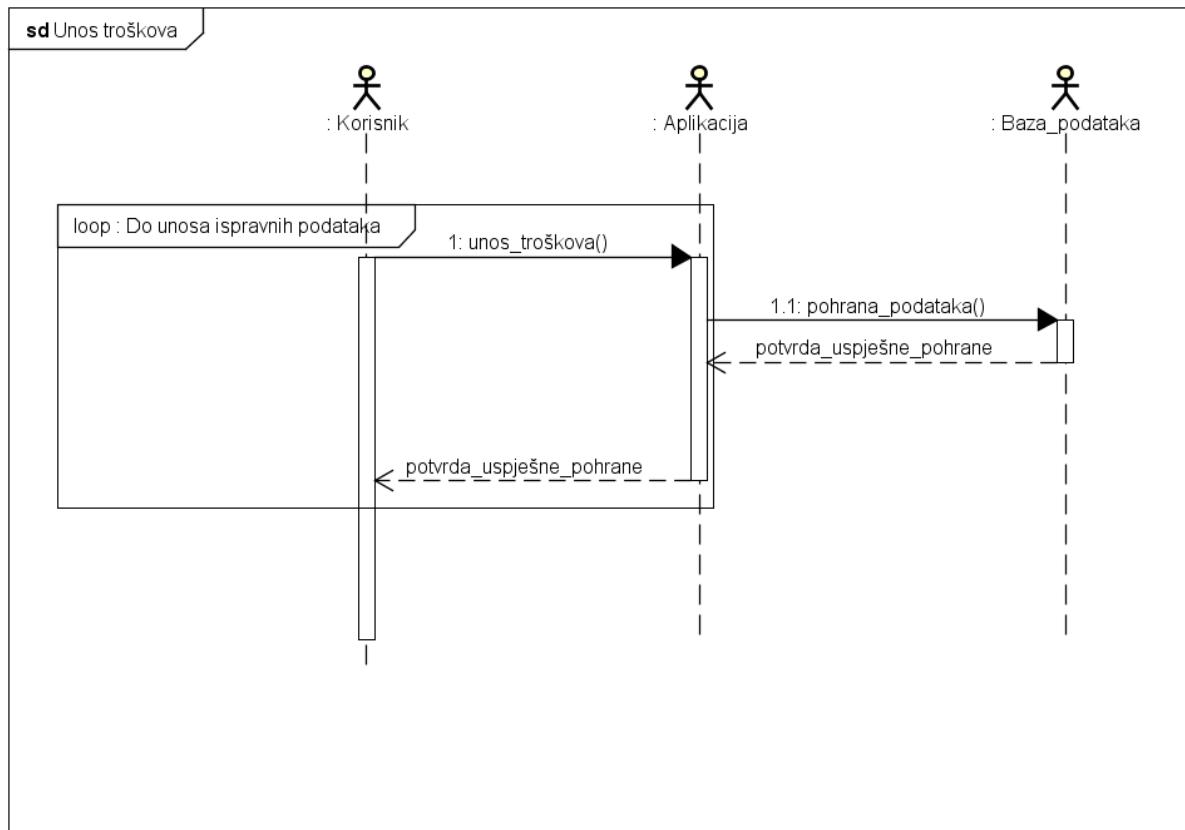
Korisnik odabire opciju unosa prihoda te unosi iznos prihoda. Web aplikacija podatke šalje u bazu na provjeru. Ako su podaci ispravni prihod se unosi u bazu podataka. U slučaju neispravnih podataka korisniku se javlja greška.



Slika 4.1.10.Sekvencijski dijagram za UC9

Sekvencijski dijagram za UC10 (Unos troškova):

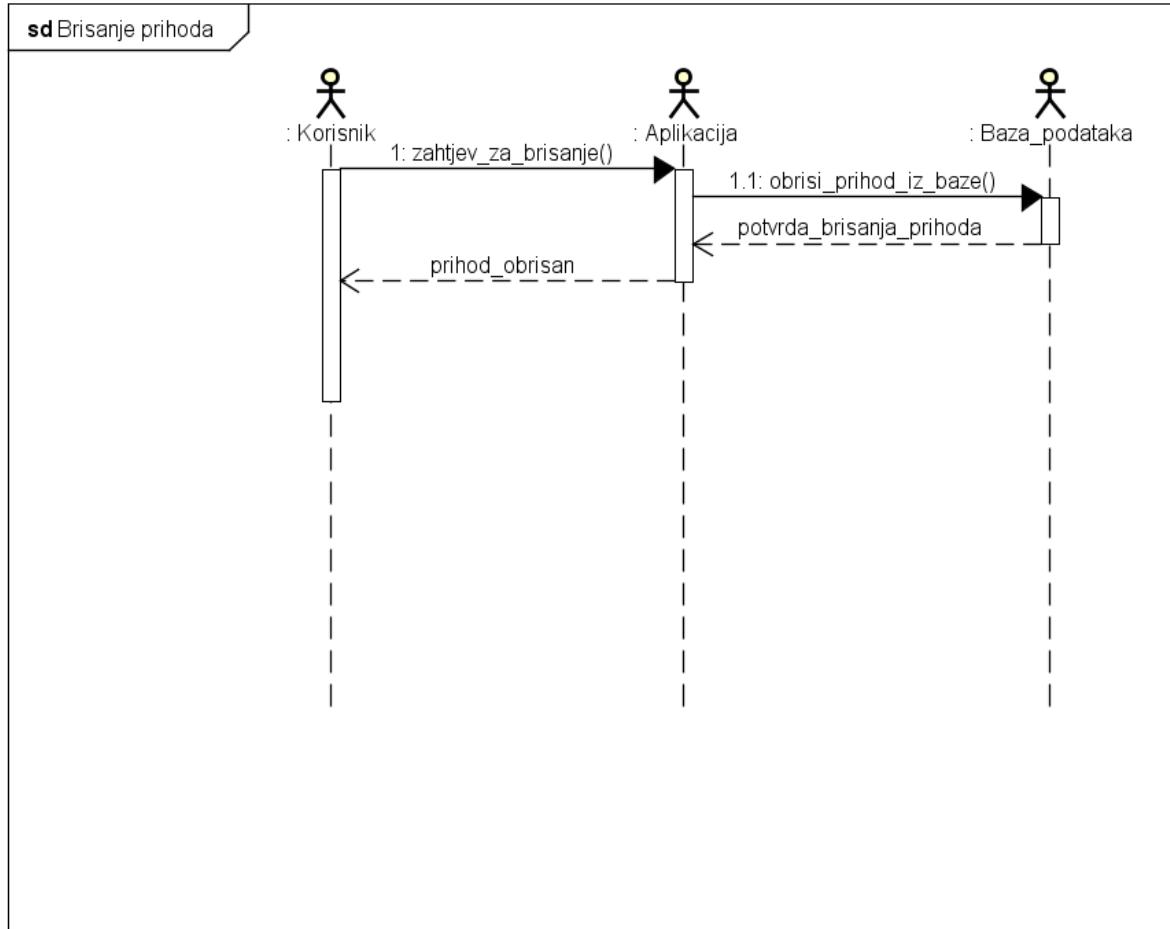
Korisnik odabire opciju unosa troškova te unosi iznos troška. Web aplikacija podatke šalje u bazu na provjeru. Ako su podaci ispravni prihod se unosi u bazu podataka. U slučaju neispravnih podataka korisniku se javlja greška.



Slika 4.1.11.Sekvencijski dijagram za UC10

Sekvencijski dijagram za UC12 (Brisanje prihoda):

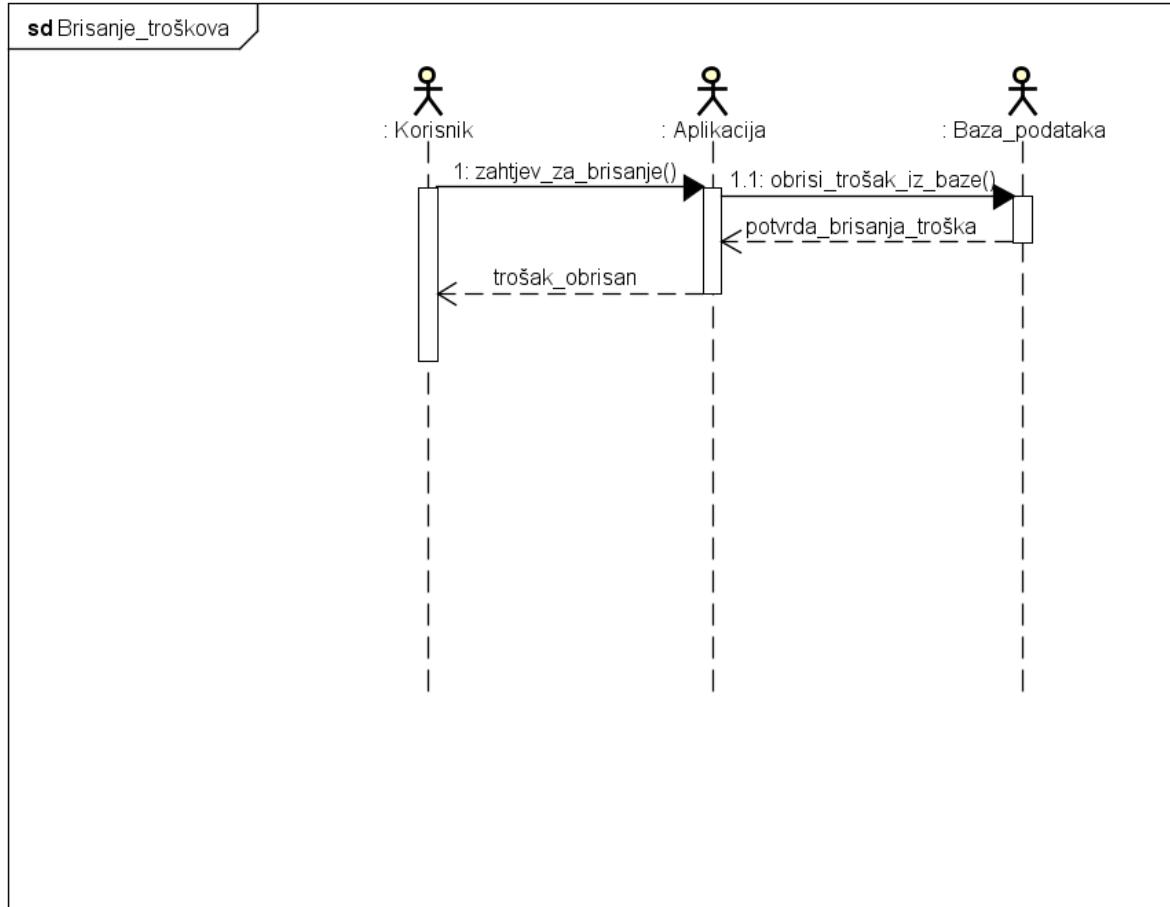
Korisnik odabire opciju brisanja prihoda. Zahtjev za brisanje dolazi poslužitelju koji provjerava ima li korisnik imao ovlasti za brisanje prihoda. Ako ima, šalje bazi zahtjev za brisanje podataka i vraća korisniku povratnu informaciju.



Slika 4.1.12.Sekvencijski dijagram za UC11

Sekvencijski dijagram za UC12 (Brisanje troškova):

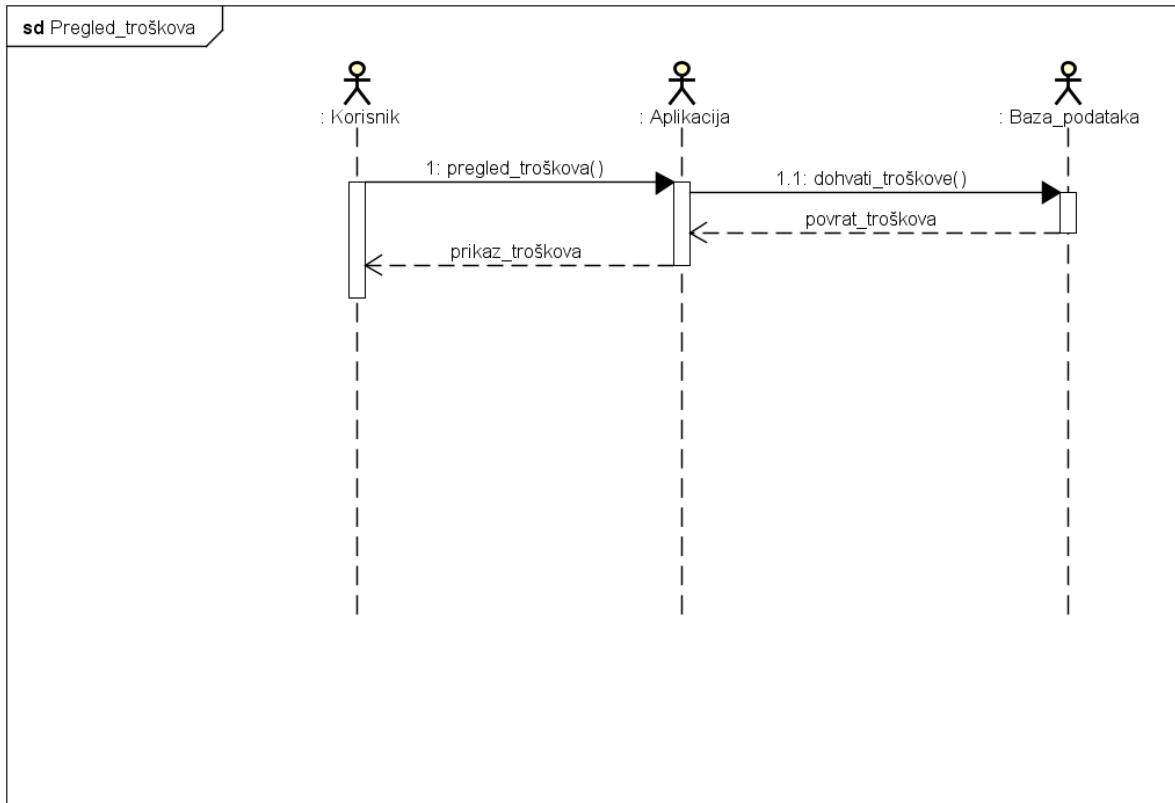
Korisnik odabire opciju brisanja troškova. Zahtjev za brisanje dolazi poslužitelju koji provjerava da li korisnik ima ovlasti za brisanje troškova. Ako ima šalje bazi zahtjev za brisanje podataka i vraća korisniku povratnu informaciju.



Slika 4.1.13.Sekvencijski dijagram za UC12

Sekvencijski dijagram za UC13 (Pregled troškova):

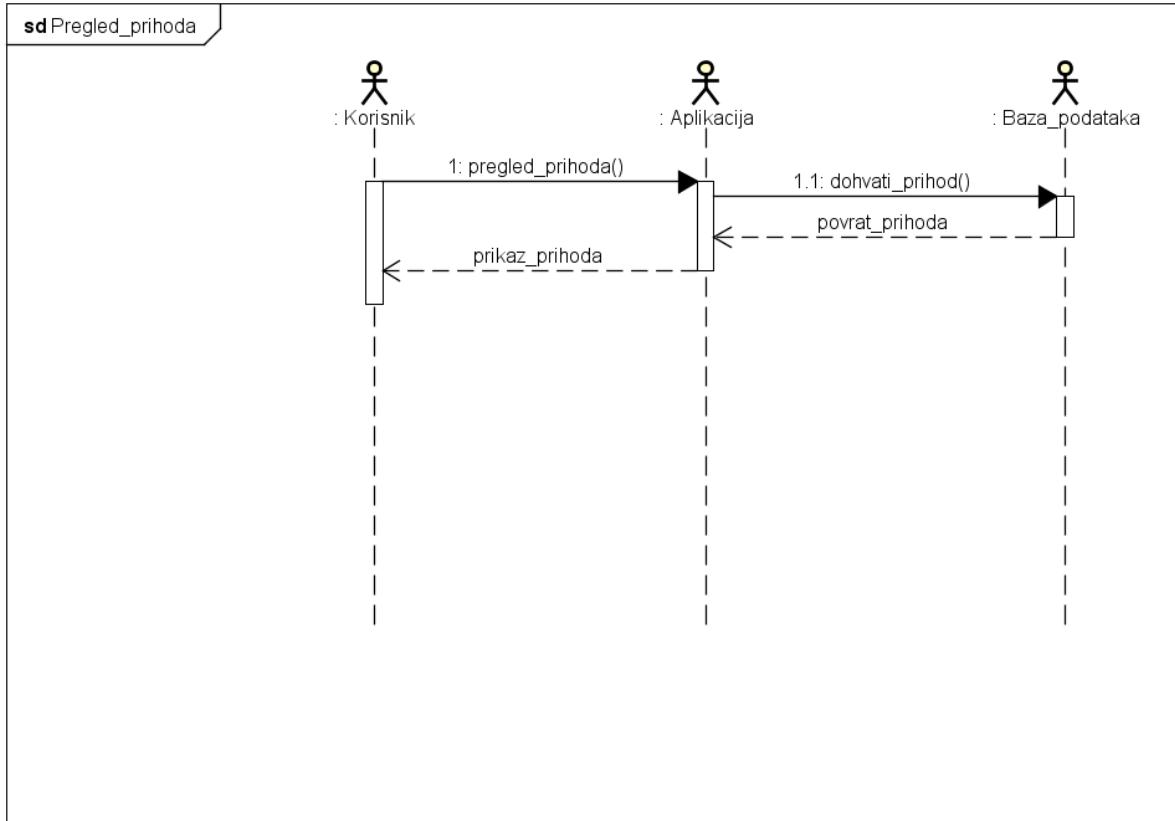
Korisnik odabire opciju pregleda troškova. Prilikom pregledavanja troškova korisnik šalje zahtjev poslužitelju te on isti prosljeđuje bazi podataka i od nje zatraži podatke troškova. Baza ih zatim proslijedi poslužitelju, a poslužitelj ih prikaže korisniku.



Slika 4.1.14.Sekvencijski dijagram za UC13

Sekvencijski dijagram za UC14 (Pregled prihoda):

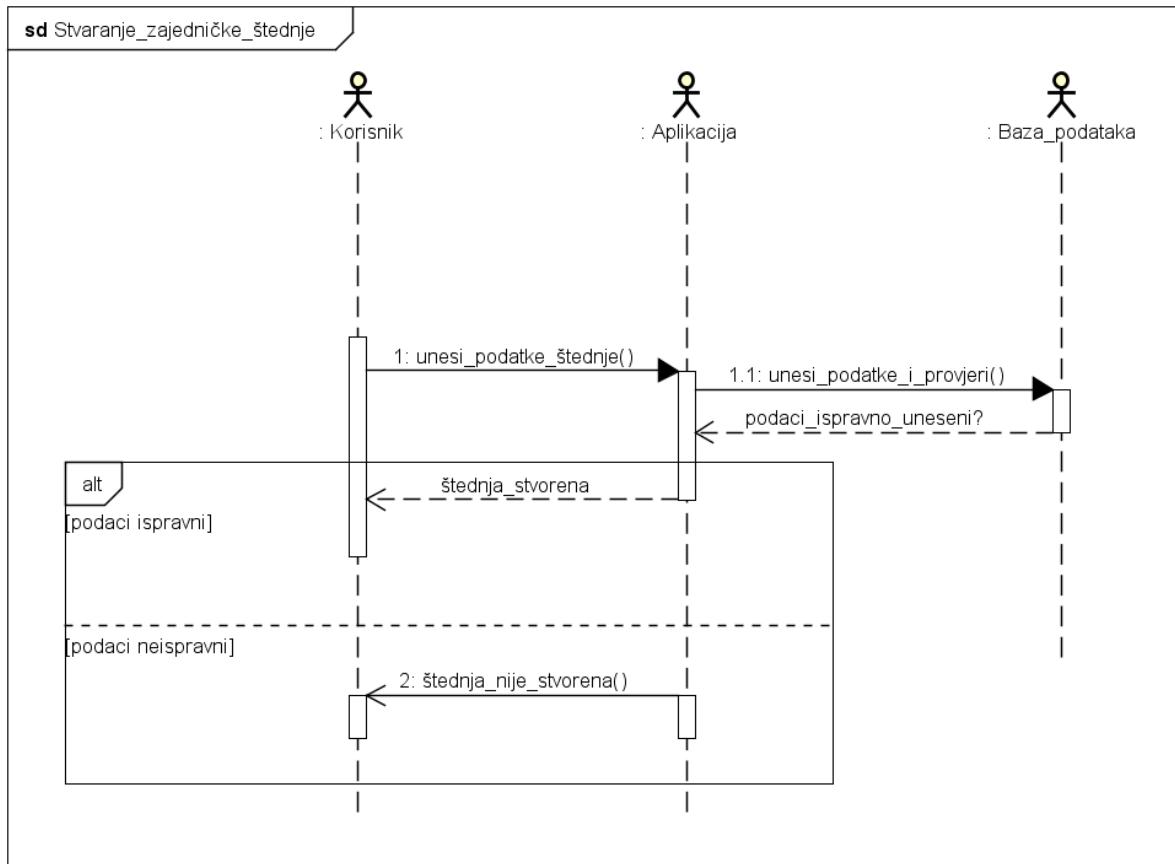
Korisnik odabire opciju pregleda prihoda. Prilikom pregledavanja prihoda korisnik šalje zahtjev poslužitelju te on isti prosljeđuje bazi podataka i od nje zatraži podatke troškova. Baza ih zatim proslijedi poslužitelju, a poslužitelj ih prikaže korisniku.



Slika 4.1.15.Sekvencijski dijagram za UC14

Sekvencijski dijagram za UC15 (Stvaranje zajedničke štednje):

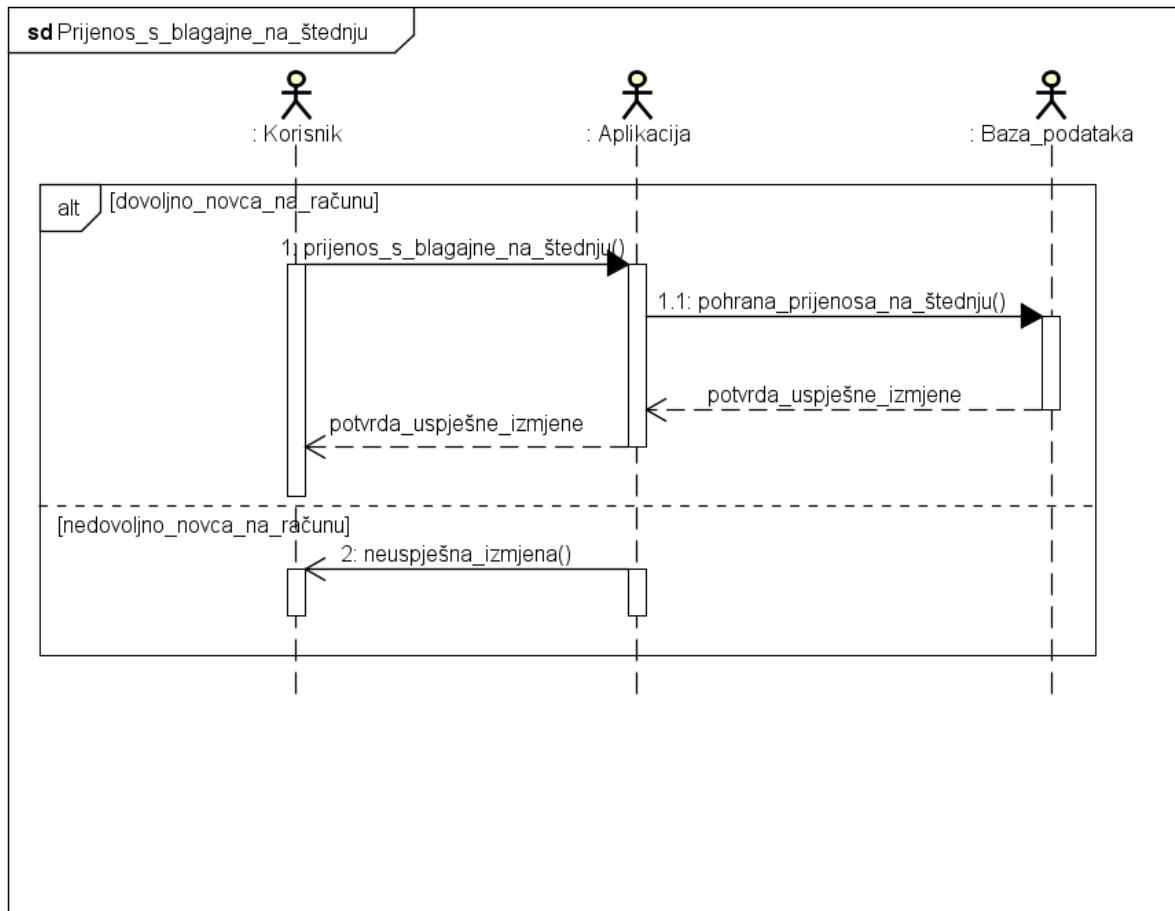
Korisnik odabire opciju kreiranja zajedničke štednje. Poslužitelj provjerava valjanost podataka te korisniku šalje poruku o ispravnosti istih. Nakon unosa podataka poslužitelj ih šalje u bazu podataka te ona kreira štednju i obavještava poslužitelj o uspješnosti koji informaciju prosljeđuje korisniku.



Slika 4.1.16.Sekvencijski dijagram za UC15

Sekvencijski dijagram za UC16 (Prijenos s blagajne na štednju):

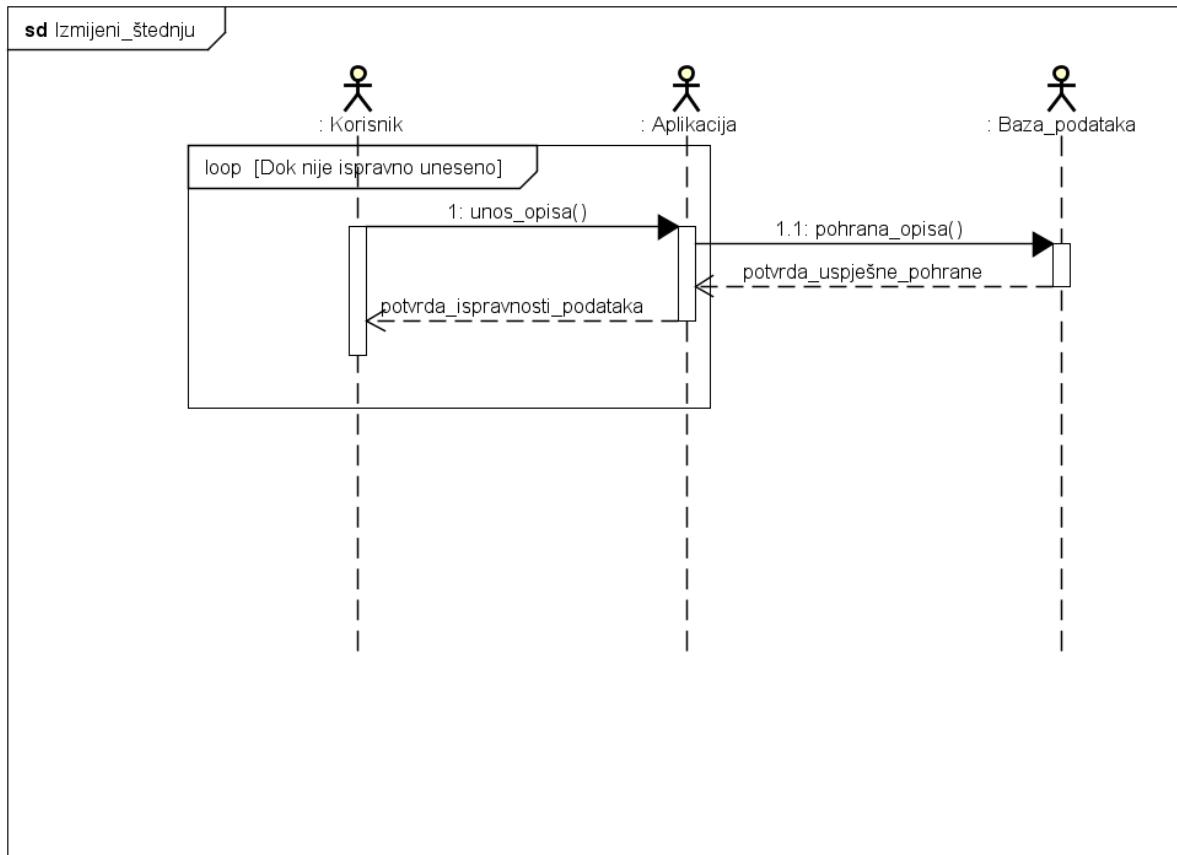
Korisnik odabire opciju izmjene štednje. Zatim unosi željeni iznos. Poslužitelj provjerava valjanost podataka te korisniku šalje poruku o ispravnosti istih. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka te ih ona prebaci u štednju i pohrani. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika. U slučaju neispravnih podataka korisniku se javlja greška.



Slika 4.1.17.Sekvencijski dijagram za UC16

Sekvencijski dijagram za UC17 (Izmjeni štednju):

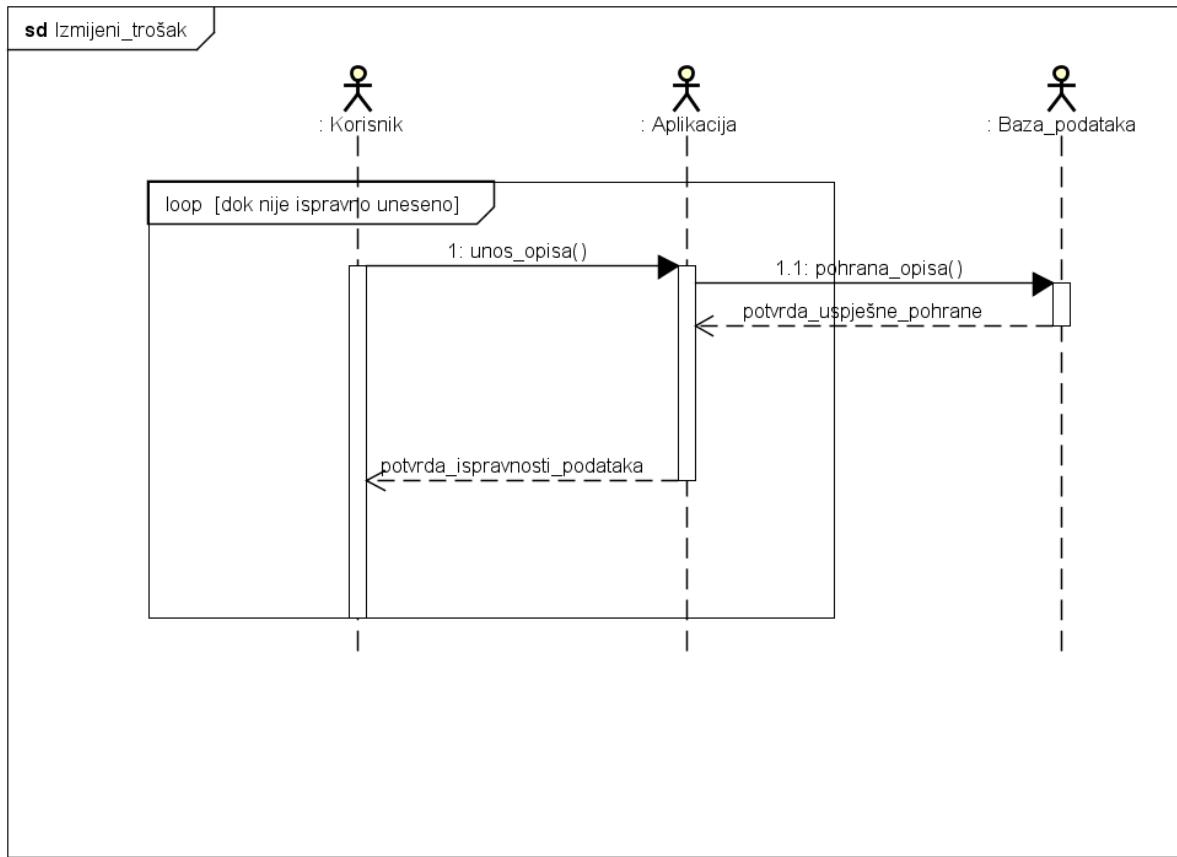
Korisnik odabire opciju izmjene štednje. Zatim unosi željenu promjenu opisa ili cilja štednje. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka gdje se pohranjuju. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika. U slučaju neispravnih podataka korisnik može ponovo pokušati.



Slika 4.1.18.Sekvencijski dijagram za UC17

Sekvencijski dijagram za UC18 (Izmjeni trošak):

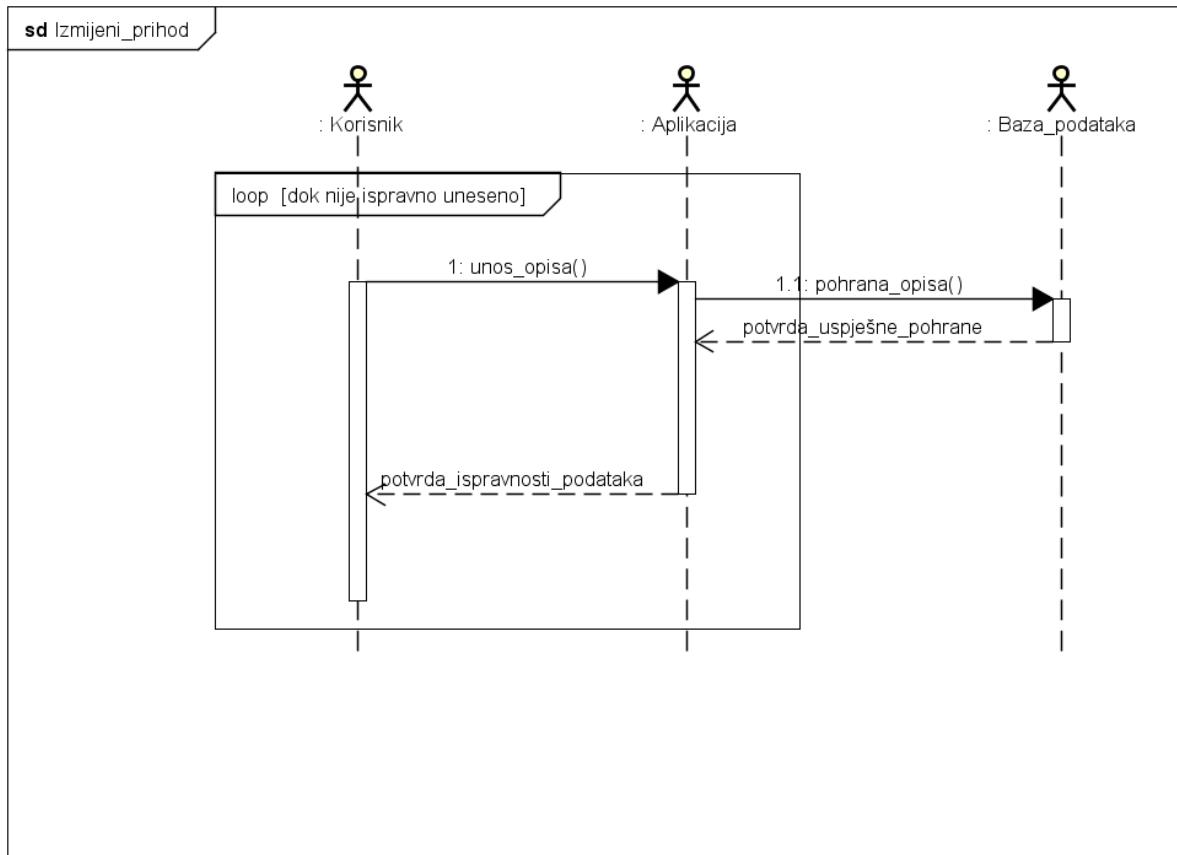
Korisnik odabire opciju izmjene troška. Zatim unosi željeni iznos troška. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka gdje se pohranjuju. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika. U slučaju neispravnih podataka korisnik može ponovo pokušati.



Slika 4.1.19.Sekvencijski dijagram za UC18

Sekvencijski dijagram za UC19 (Izmjeni prihod):

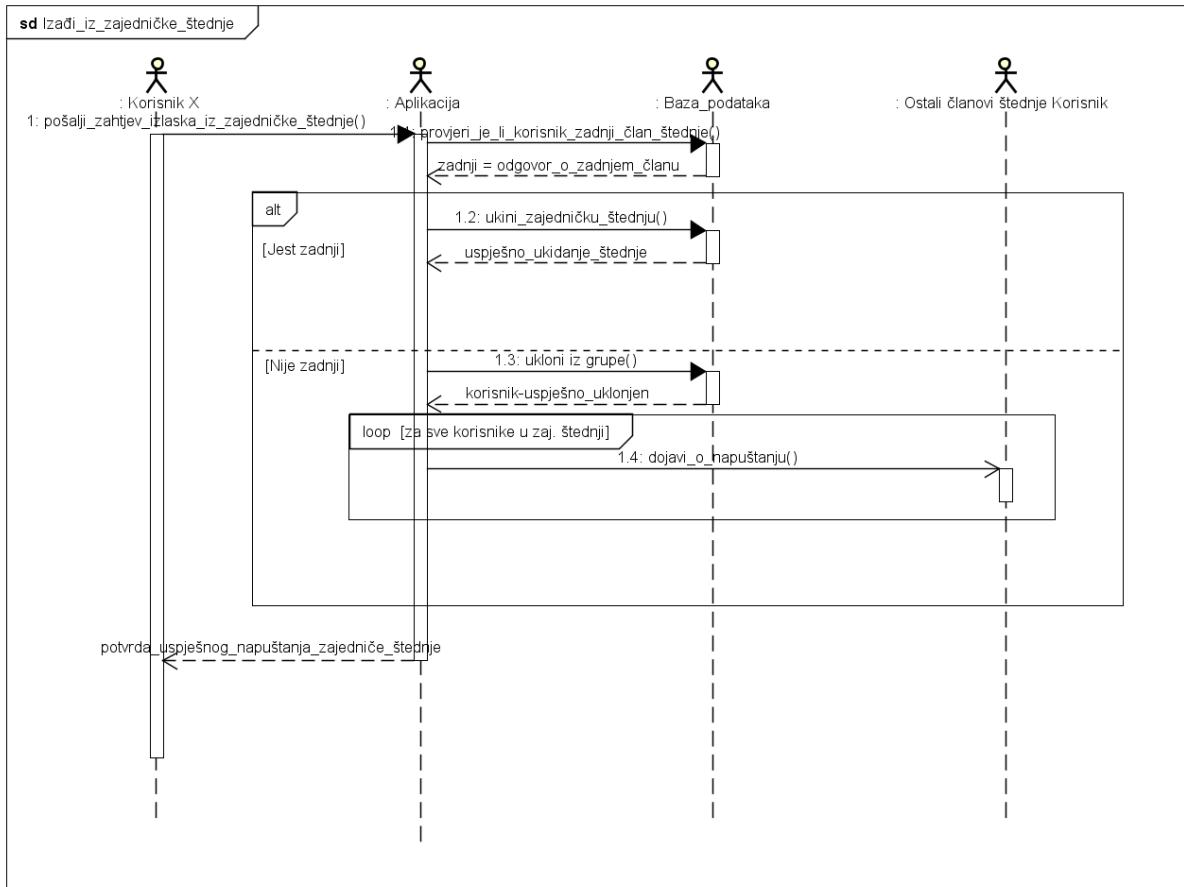
Korisnik odabire opciju izmjene prihoda. Zatim unosi željeni iznos prihoda. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka gdje se pohranjuju. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika. U slučaju neispravnih podataka korisnik može ponovo pokušati.



Slika 4.1.20.Sekvencijski dijagram za UC19

Sekvencijski dijagram za UC20 (Izađi iz zajedničke štednje)

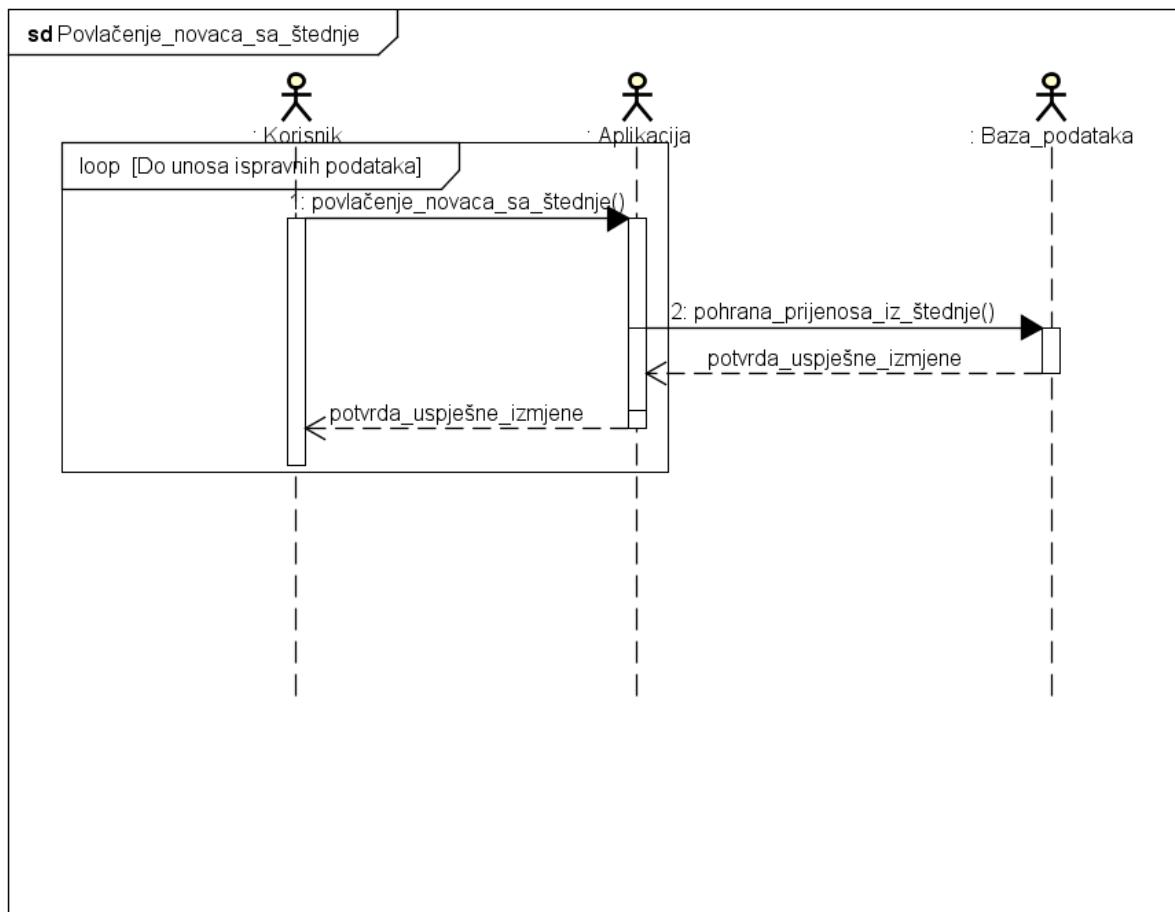
Korisnik odabire opciju izlaska iz zajedničke štednje. Zatim šalje zahtjev sa svojim podacima i podacima o zajedničkoj štednji poslužitelju. Poslužitelj šalje upit prema bazi je li taj korisnik posljednji korisnik grupe, a baza mu šalje potrebnu informaciju. Ako je korisnik posljednji član grupe, poslužitelj prema bazi šalje zahtjev za ukidanjem grupe pri čemu baza vraća poruku o uspješnom ukidanju grupe. Ukoliko korisnik nije posljednji član grupe, poslužitelj prema bazi šalje zahtjev za brisanjem korisnika iz grupe, a baza mu vraća poruku o uspješnom ukidanju grupe. Nakon toga, poslužitelj ostalim članovima grupe šalje informaciju o korisnikovom napuštanju grupe, a samom korisniku šalje informaciju o uspješnom napuštanju grupe.



Slika 4.1.21.Sekvencijski dijagram za UC20

Sekvencijski dijagram za UC21 (Povlačenje novaca sa štednje):

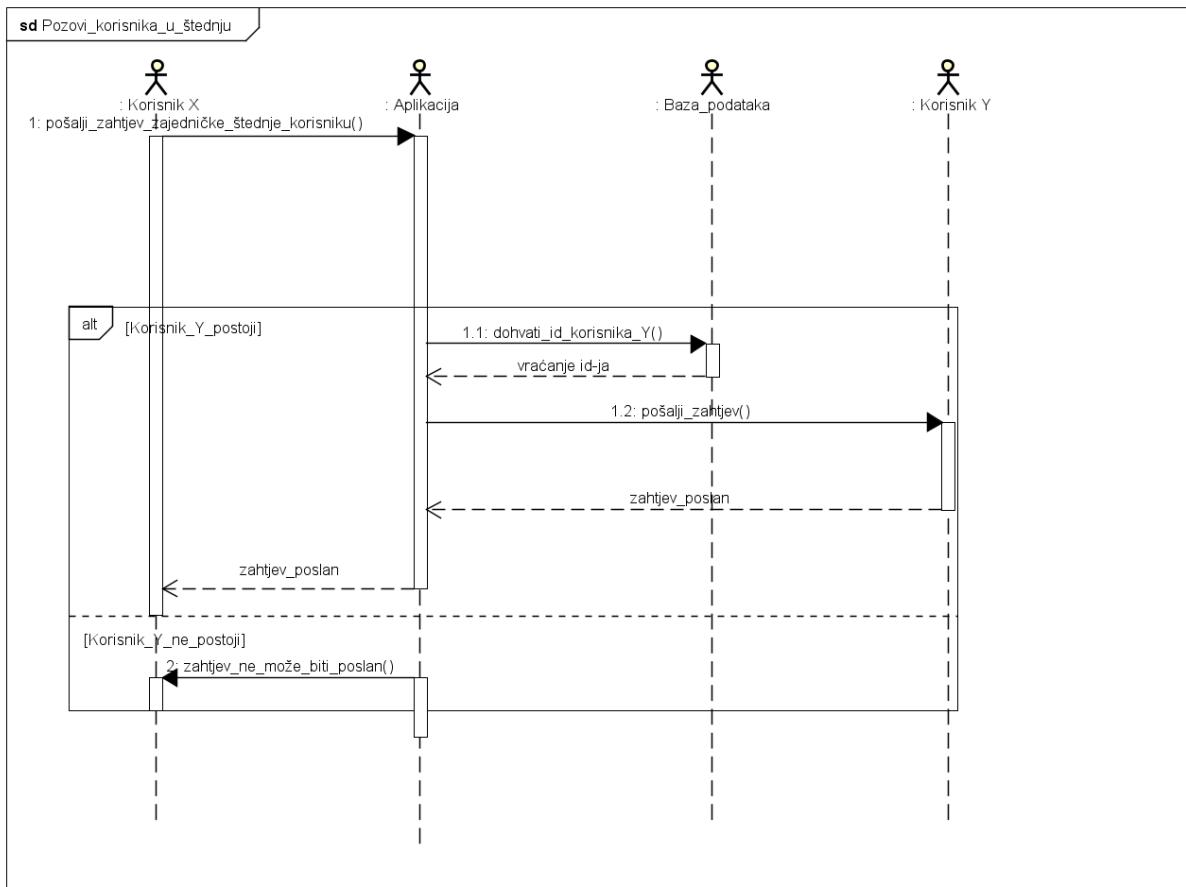
Korisnik odabire opciju uređivanja štednje. Zahtjev se prosljeđuje poslužitelju zajedno s podacima korisnika. Zatim unosi željeni iznos povlačenja. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj šalje zahtjev u bazu podataka koja povlači novce iz štednje te pohranjuje promjene. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika. U slučaju neispravnih podataka korisniku se javlja greška.



Slika 4.1.22.Sekvencijski dijagram za UC21

Sekvencijski dijagram za UC22 (Pozovi korisnika u štednju):

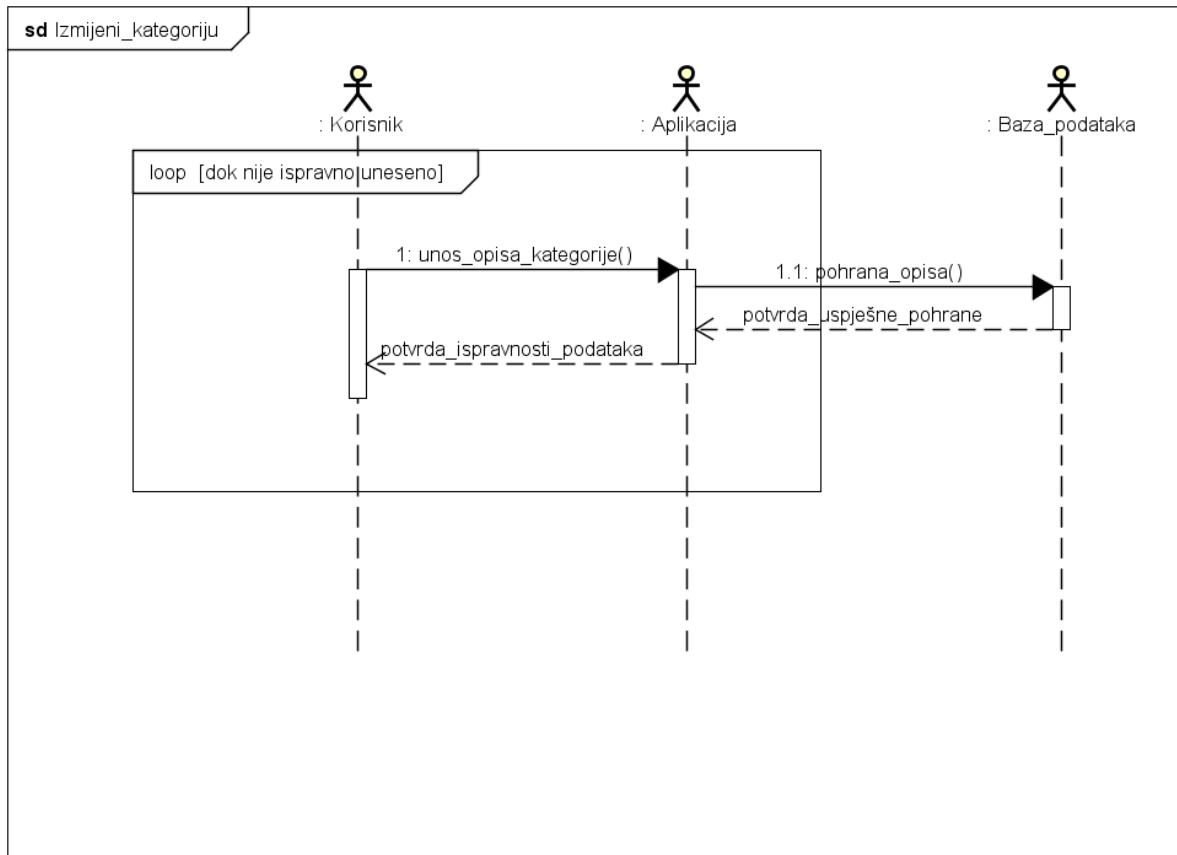
Korisnik odabire opciju pozivanja drugih korisnika u zaj. štednju. Zahtjev dolazi do poslužitelja koji provjerava postojanje korisnika uz pomoć baze. Poslužitelj u bazu podataka pohranjuje zahtjev. Ako korisnik Y postoji poslužitelj mu šalje zahtjev korisnika X te korisnik Y šalje poslužitelju potvrdu o primitku zahtjeva. U slučaju nepostojanja korisnika javlja se greška.



Slika 4.1.23.Sekvencijski dijagram za UC22

Sekvencijski dijagram za UC23 (Izmjeni kategoriju):

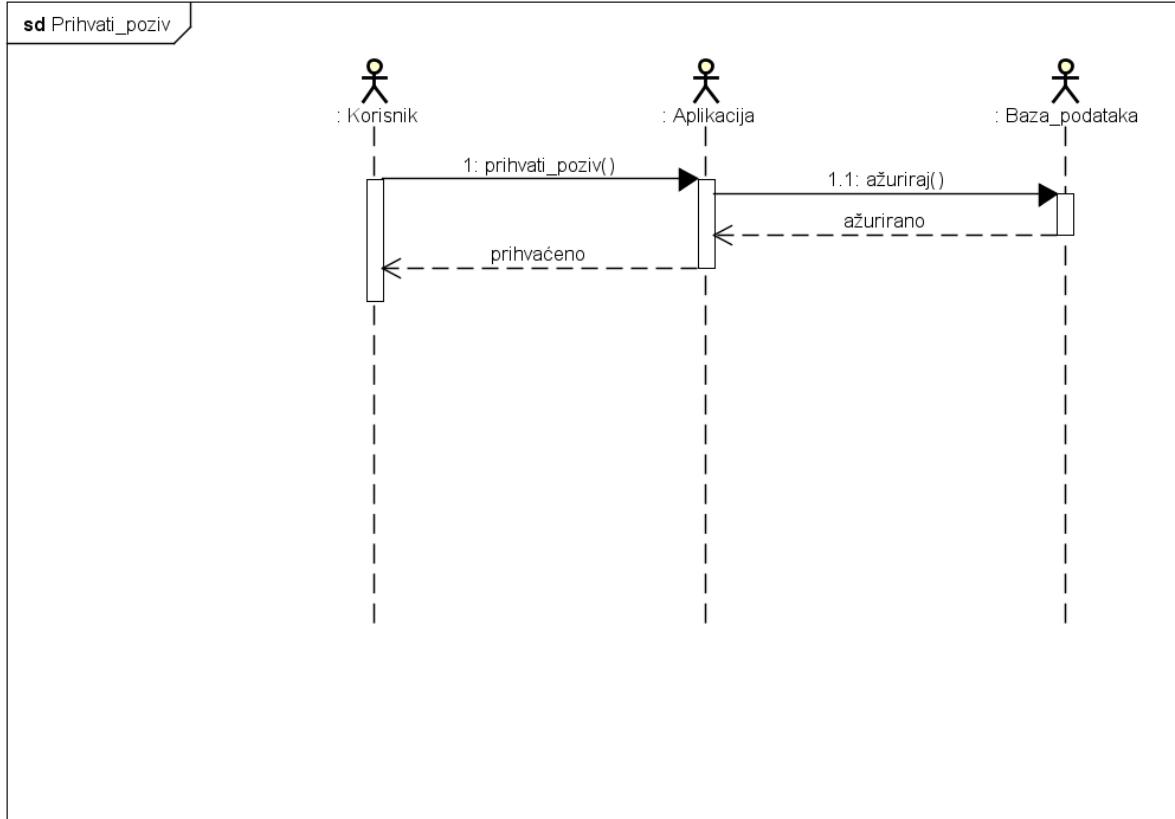
Korisnik odabire opciju izmjene kategorije. Zatim unosi željenu izmjenu opisa kategorije. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka gdje se pohranjuju. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika. U slučaju neispravnih podataka korisniku se javlja greška.



Slika 4.1.24.Sekvencijski dijagram za UC23

Sekvencijski dijagram za UC24 (Prihvati poziv):

Korisnik koji je pozvan uneku zajedničku štednju može jednostavno potvrditi poziv, pri čemu će se njegova odluka reflektirati u bazi podataka, a server će ga obavijestiti o uspjehu.



Slika 4.1.25.Sekvencijski dijagram za UC24

5. Ostali zahtjevi

- Sustav mora ispravno, brzo i konzistentno pružati informacije korisniku
- Svaka promjena koju korisnik napravi mora biti uspješno i trajno pohranjena u sustavu
- Sustav mora podržavati paralelno korištenje, tj. istovremenu uslugu većem broju korisnika
- Sustav mora raditi dovoljno brzo da bude koristan korisniku
- Aplikacija mora biti intuitivna i jednostavna za korištenje
- Pri svakoj registraciji korisnik će na raspolaganju imati nekoliko „defaultnih“ kategorija
- Sustav mora biti otporan na potencijalne pogreške i kvarove
- Korisnici smiju pristupiti isključivo informacijama koje su im dodijeljene
- Nadogradnja dodatnih funkcionalnosti mora biti moguća bez narušavanja strukture sustava
- Aplikacija mora biti prilagođena uporabi na internetu i responzivna

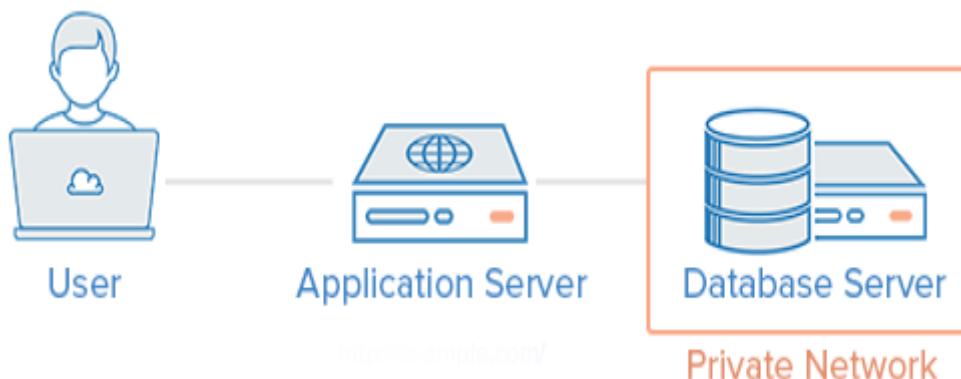
6. Arhitektura i dizajn sustava

Arhitektura je temelj sustava programske potpore i kao takva mora biti pažljivo odabrana. Izbor kvalitetne arhitekture može značiti razliku između osrednjeg i vrhunskog proizvoda. Ona mora u potpunosti i efikasno podržavati sve zahtjeve naše aplikacije, prije svega to je višekorisnički rad putem interneta i paralelna manipulacija podacima. Stoga, koristit ćemo arhitekturu web aplikacije, podržanu objektno-usmjerena programskom paradigmom.

Korisnik neće morati instalirati aplikaciju na vlastito računalo, već će moći iz web preglednika pokrenuti proizvod.

Temelj arhitekture bit će sljedeći elementi:

- Baza podataka
- Web aplikacija
- Web poslužitelj



Slika 6.1 Simboličan prikaz elemenata arhitekture

6.1. Svrha, opći prioriteti i skica sustava

Sustav je predviđen za jednostavan višekorisnički pristup, praćenje, pregledavanje i izmjenu informacija. Da bi udovoljili tim zahtjevima koristit ćemo web poslužitelj povezan s bazom podataka, u koju će sve informacije od korisnika biti pohranjene. Prioritet pri uporabi ovog sustava je sigurnost, koja je osigurana provjerjenim tehnologijama (Java Spring Boot) i dobrim inženjerskim navikama.

Osnova rada sa sustavom je dana ovako:

- I. Korisnik putem web preglednika šalje zahtjev web poslužitelju.
- II. Web poslužitelj pokreće web aplikaciju i prosljeđuje zahtjev.
- III. Aplikacija obrađuje zaprimljeni zahtjev i po potrebi pristupa bazi podataka.
- IV. Putem poslužitelja, aplikacija vraća korisniku odgovor u obliku HTML dokumenta vidljivog u web pregledniku.

Najvažniji podsustavi arhitekture našeg sustava su:

1. Web aplikacija
2. Web poslužitelj
3. Baza podataka

Web aplikacija

Web aplikacija bit će realizirana u objektno orijentiranom programskom jeziku Java (tehnologija Java Spring Boot) i ona će obrađivati i odgovarati na korisnikove zahtjeve, koje oblikuje u operacije sustava, kao i primati podatke koje korisnik unosi. Prilikom primanja zahtjeva putem web poslužitelja, web aplikacija će isti obraditi, dohvatiti podatke iz baze podataka, generirati željeni rezultat te isti vratiti u HTML obliku sa željenim podatcima. Web aplikacija je zadužena i za vizualnu prezentaciju sustava i podataka koju vrši putem pripreme vizualne informacije u HTML obliku, dok web preglednik iste prikazuje u obliku web stranica. Web aplikacija također komunicira i s bazom podataka putem Interneta, upravlja svim podacima potrebnim za rad sustava te ispunjava bazu podataka po potrebi.

Web poslužitelj

Poslužitelj funkcioniра на principu zahtjeva i odgovora, korisnik preko web preglednika šalje određeni zahtjev, a poslužitelj putem HTTP protokola odgovara na navedeni zahtjev.

Naša se aplikacija trenutno izvodi na Heroku web poslužitelju. Web poslužitelj može biti poslužen na lokalnom hardveru ili integriran u cloud-u, on sadržava sveukupnu okolinu (skripte, biblioteke) koja je potrebna za izvođenje web stranice. U našem konkretnom implementaciji web poslužitelj je integriran u cloudu.

Baza podataka

Baza podataka je integralni dio sustava koji je odgovoran za pohranu, dostavu i upravljanje informacijama unesenih od strane korisnika.

Našu bazu podataka čine sljedeće relacije:

- Category
- Expenditure
- Users
- Revenue
- Saving
- Log
- Common_balance
- dodatne relacije koje služe za Many-to-Many povezivanje

Category – ovaj entitet predstavlja zasebne kategorije za evidenciju prihoda/troškova pojedinog korisnika. Sadrži sljedeće atribute:

- cat_id – tekstualni identifikator kategorije,
- name – ime kategorije,
- info- opis kategorije.

Entitet je u vezi Many-to-Many s entitetom Users putem atributa user_id i cat_id.

Expenditure – ovaj entitet predstavlja određene troškove koje je ostvario korisnik. Sadrži sljedeće atribute:

- exp_id – identifikator troškova,
- name – naziv troška,
- info – opis troška,
- date – datum troška,
- amount – iznos troška,
- cat_id – tekstualni identifikator kategorije.

Entitet je u vezi Many-to-One s entitetima Category putem atributa cat_id i Users putem atributa user_id

Saving – ovaj entitet predstavlja korisničku štednju koju korisnik može pokrenuti te zadati željeni iznos štednje koji želi postići i krajnji datum do kojeg želi ostvariti taj iznos.

Sadrži sljedeće atribute:

- sav_id – tekstualni identifikator štednje,
- start_date – datum početka štednje,
- end_date – datum završetka štednje,
- goal – željeni iznos štednje,
- funds – novčana sredstva u štednji,
- info – opis štednje.

Entitet je u vezi Many-to-One s entitetom Users putem atributa user_id.

Users – ovaj entitet predstavlja registrirane korisnike te služi za pohranjivanje i čuvanje podataka koji opisuju korisnika u našem sustavu web aplikacije. Sadrži sljedeće atribute:

- name – ime osobe,
- surname – prezime osobe,
- e_mail – email adresa korisnika,
- password - šifra,
- username – korisničko ime i
- funds – novčana sredstva.

Entitet je u vezi Many-to-Many s entitetima Category i Common_balance putem atributa user_id, a One-to-Many s entitetima Expenditure, Revenue, Saving i Log putem atributa user_id.

Revenue – ovaj entitet predstavlja prihode pojedinog korisnika. Sadrži sljedeće atribute:

- rev_id – tekstualni identifikator prihoda,
- name – naziv prihoda,
- info – opis prihoda,
- date – datum prihoda,
- amount – iznos prihoda,
- cat_id – tekstualni identifikator kategorije.

Entitet je u vezi Many-to-One s entitetima Category putem atributa cat_id te rev_id i Users putem atributa user_id.

Log – ovaj entitet predstavlja praćenje korisnikovih radnji nad zajedničkom štednjom.

Sadrži atribute:

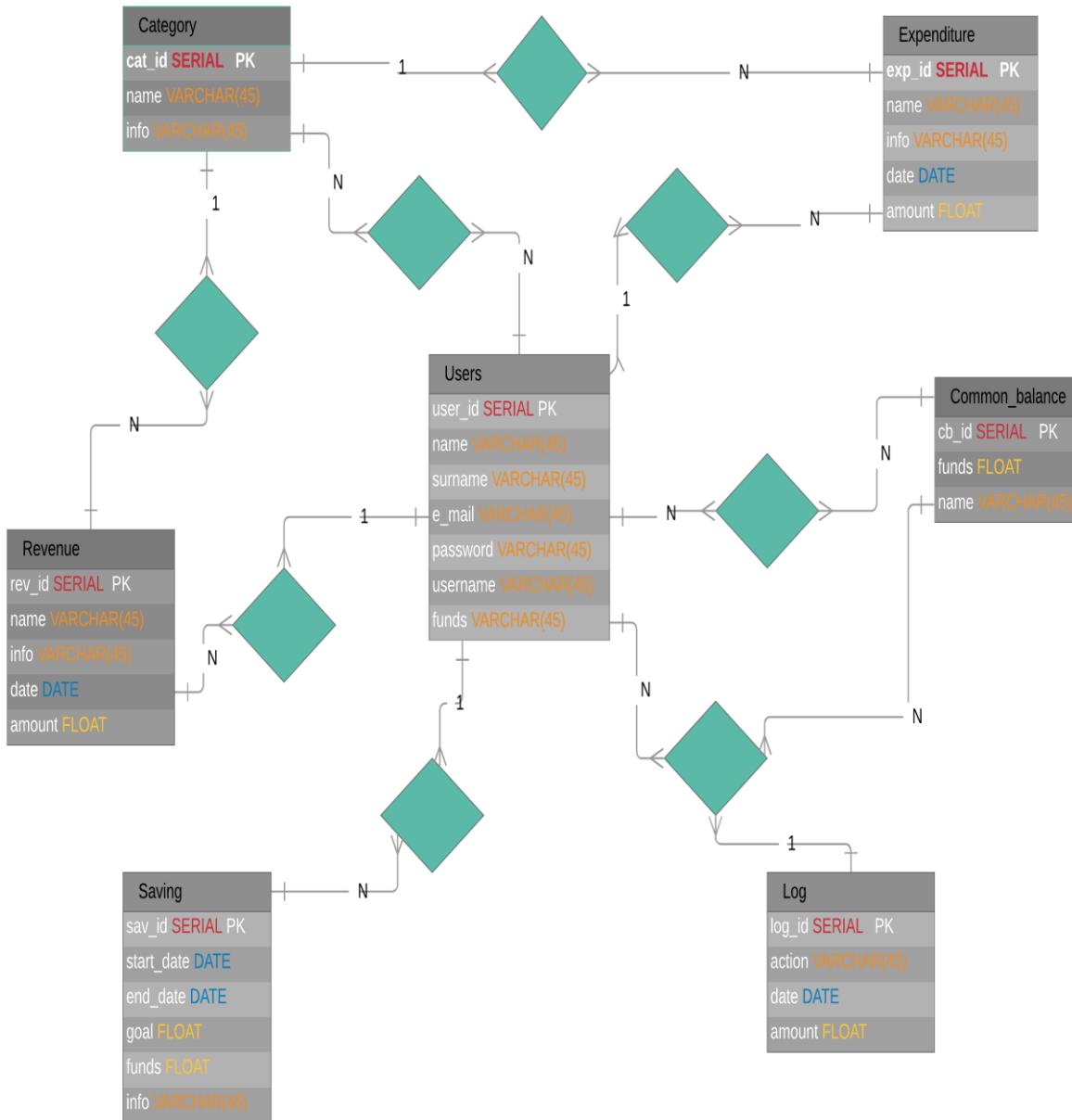
- log_id – tekstualni identifikator,
- action – radnja,
- e_mail – email adresa korisnika,
- date – datum prijave,
- amount – iznos upisanih prihoda/troškova/štednji,
- user_id – tekstualni identifikator korisnika,
- cb_id – tekstualni identifikator zajedničke štednje.

Entitet je u vezi One-to-Many s entitetima Users putem atributa log_id te user_id i entitetom Common_balance putem atributa cb_id.

Common balance – ovaj atribut predstavlja zajednički novčani iznos korisnika koji su u kolektivnoj štednji. Sadrži sljedeće atribute:

- cb_id – tekstualni identifikator zajedničke štednje,
- funds – iznos,
- name – naziv zajedničke štednje.

Entitet je u vezi Many-to-Many s entitetom Users putem atributa cb_id.



Slika 6.1.1: ER model baze podataka

6.2. Dijagram razreda s opisom

Web aplikacija dijeli se na prezentacijski, aplikacijski i podatkovni sloj. Prezentacijski sloj predstavlja interakciju korisnika s aplikacijom pomoću HTTP zahtjeva i omogućuje ispis podataka koje korisnik vidi u web pregledniku. Aplikacijski sloj zadužen je za upravljanje aplikacijom, implementaciju poslovne logike i suradnju s podatkovnim slojem kako bi obradio zahtjeve korisnika i dao im odgovore. Podatkovni sloj komunicira s bazom podataka i obrađuje dobivene korisničke podatke.

Web stranica se dijeli na poglede, tj. dohvatljive dijelove web stranice kojima korisnik može pristupiti u web pregledniku. Pogledi predstavljaju prezentacijski sloj aplikacije i modeliraju se razredima, tako da jedan razred predstavlja jedan pogled web aplikacije.

Pogledi

Upravljači u nastavku služe za mapiranje korisničkih akcija nad aplikacijom s odgovarajućim metodama u programu. Kada npr. korisnik htjedne promijeniti neki prihod, doći će to pogleda za prihode, a upravljač za prihode će se posvetiti obradi tog zahtjeva.

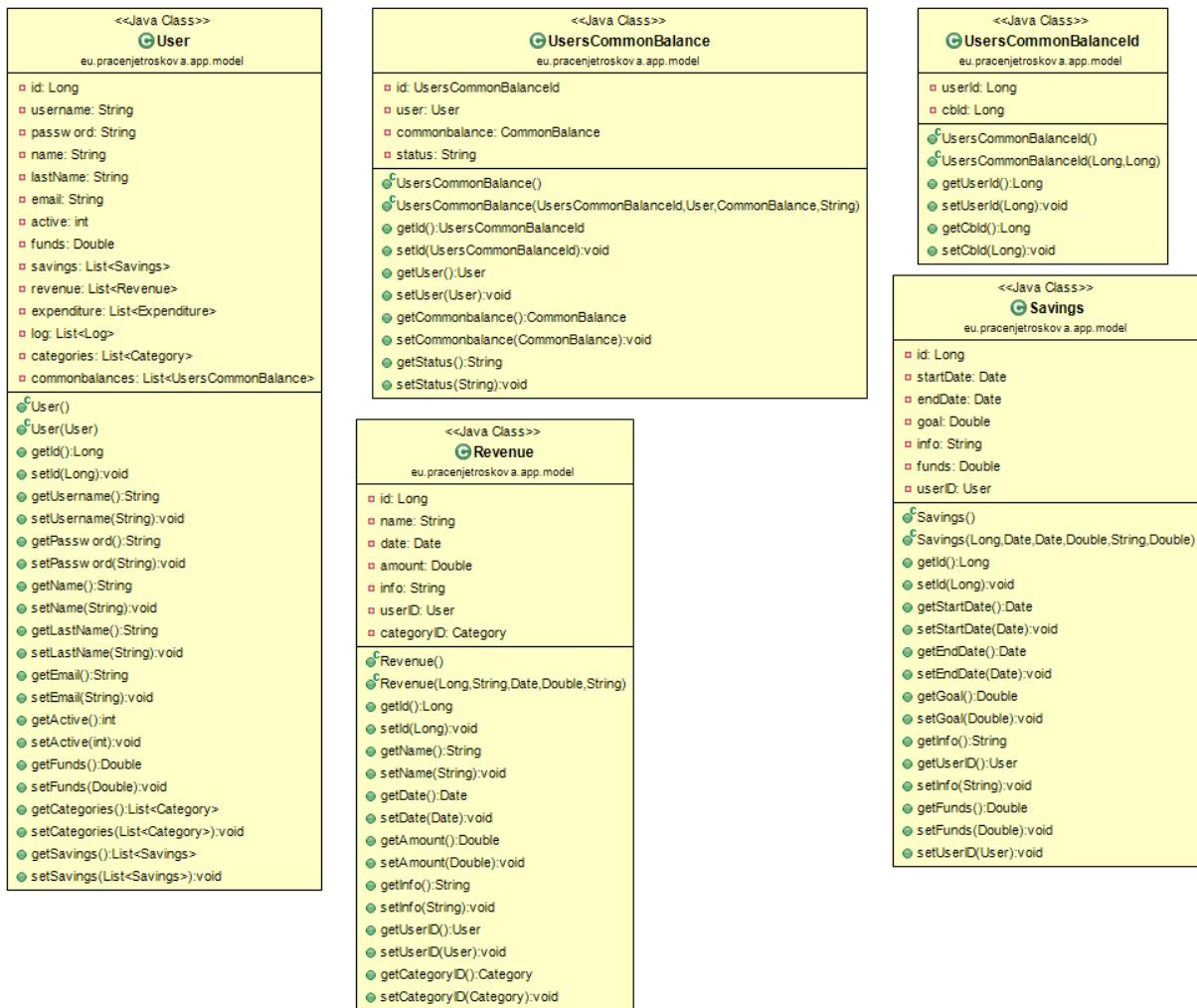


Slika 6.2.1 Razredi paketa kontroler

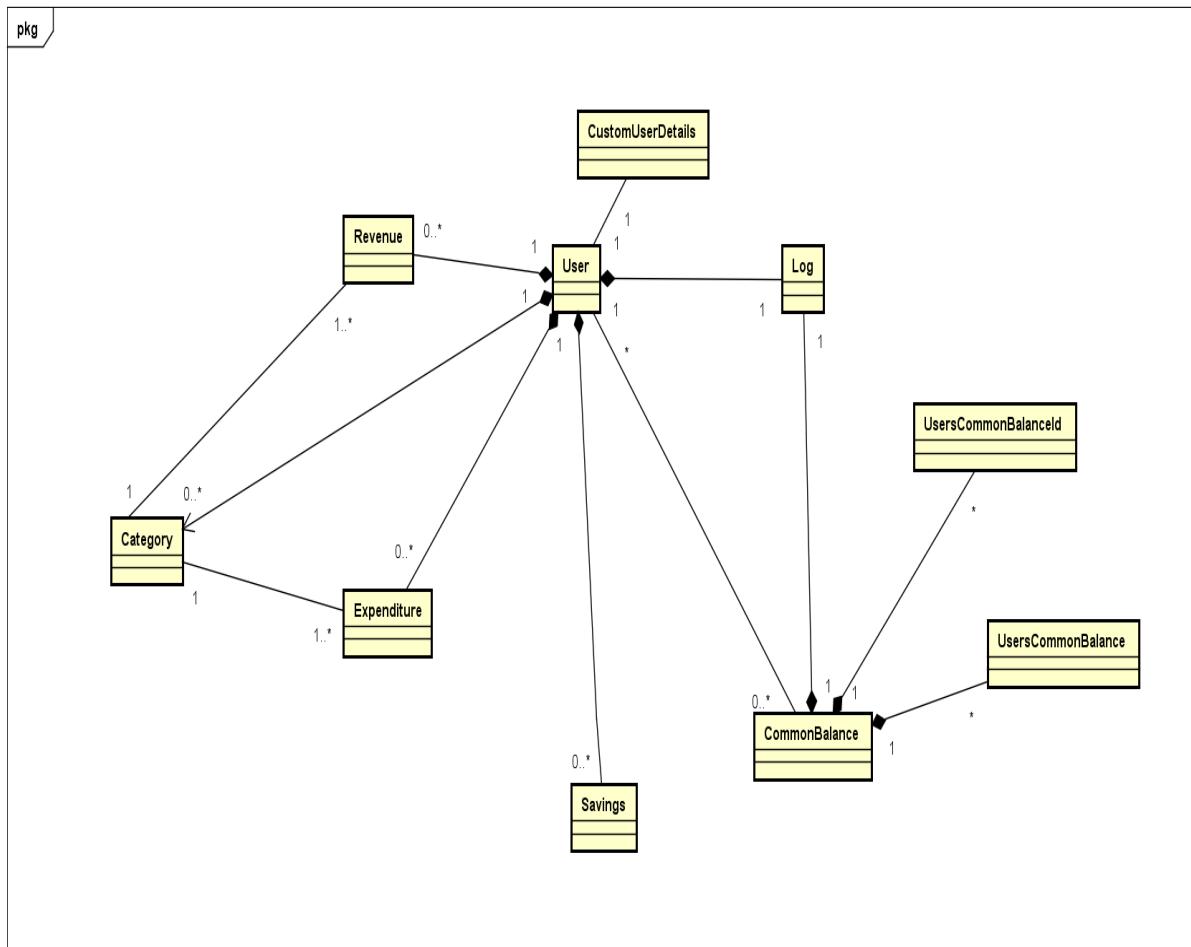
Entiteti

Entiteti su razredi koji odgovaraju tablicama u bazi podataka i sadrže metode koje obavljaju operacije nad tim objektima. Entiteti predstavljaju aplikacijski sloj web aplikacije.

<pre><<Java Class>> C CommonBalance eu.pracenjetroskova.app.model □ id: Long □ funds: Double □ name: String □ info: String □ goal: Double □ date: Date □ log: List<Log> □ users: List<UsersCommonBalance> C CommonBalance() E CommonBalance(Long,Double,String,String,Double,Date) E getid():Long E setid(Long):void E getFunds():Double E setFunds(Double):void E getName():String E setName(String):void E getInfo():String E setInfo(String):void E getGoal():Double E setGoal(Double):void E getLog():List<Log> E setLog(List<Log>):void E getUsers():List<UsersCommonBalance> E setUsers(List<UsersCommonBalance>):void E getDate():Date E setDate(Date):void E deleteMember(UsersCommonBalance):void</pre>	<pre><<Java Class>> C CustomUserDetails eu.pracenjetroskova.app.model □ role: String E CustomUserDetails(User) E getAuthorities():Collection<? extends GrantedAuthority> E getPassword():String E getUsername():String E isAccountNonExpired():boolean E isAccountNonLocked():boolean E isCredentialsNonExpired():boolean E isEnabled():boolean</pre>	<pre><<Java Class>> C Log eu.pracenjetroskova.app.model □ id: Long □ action: String □ date: Date □ amount: Double □ userID: User □ cbID: CommonBalance E Log() E Log(Long,String,Date,Double) E getid():Long E setid(Long):void E getAction():String E setAction(String):void E getDate():Date E setDate(Date):void E getAmount():Double E setAmount(Double):void E getUserId():User E setUserId(User):void E getCbID():CommonBalance E setCbID(CommonBalance):void</pre>
<pre><<Java Class>> C Expenditure eu.pracenjetroskova.app.model □ id: Long □ name: String □ date: Date □ amount: Double □ info: String □ userID: User □ categoryID: Category E Expenditure() E Expenditure(Long,String,Date,Double,String) E getid():Long E setid(Long):void E getName():String E setName(String):void E getDate():Date E setDate(Date):void E getAmount():Double E setAmount(Double):void E getInfo():String E setInfo(String):void E setUserID(User):void E getCategoryID():Category E setCategoryID(Category):void E getUserId():User</pre>	<pre><<Java Class>> C Category eu.pracenjetroskova.app.model □ id: Long □ name: String □ info: String □ type: String □ expenditure: List<Expenditure> □ revenue: List<Revenue> □ users: List<User> E Category() E Category(Long,String,String,Type) E getid():Long E setid(Long):void E getName():String E setName(String):void E getInfo():String E setInfo(String):void E getType():String E setType(String):void</pre>	



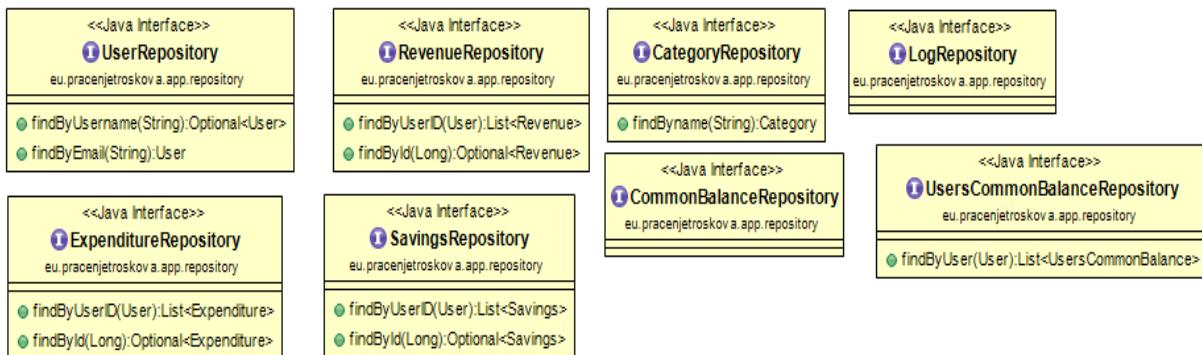
Slika 6.2.2 Razredi paketa entiteti



Slika 6.2.3 Dijagram razreda

Repozitoriji

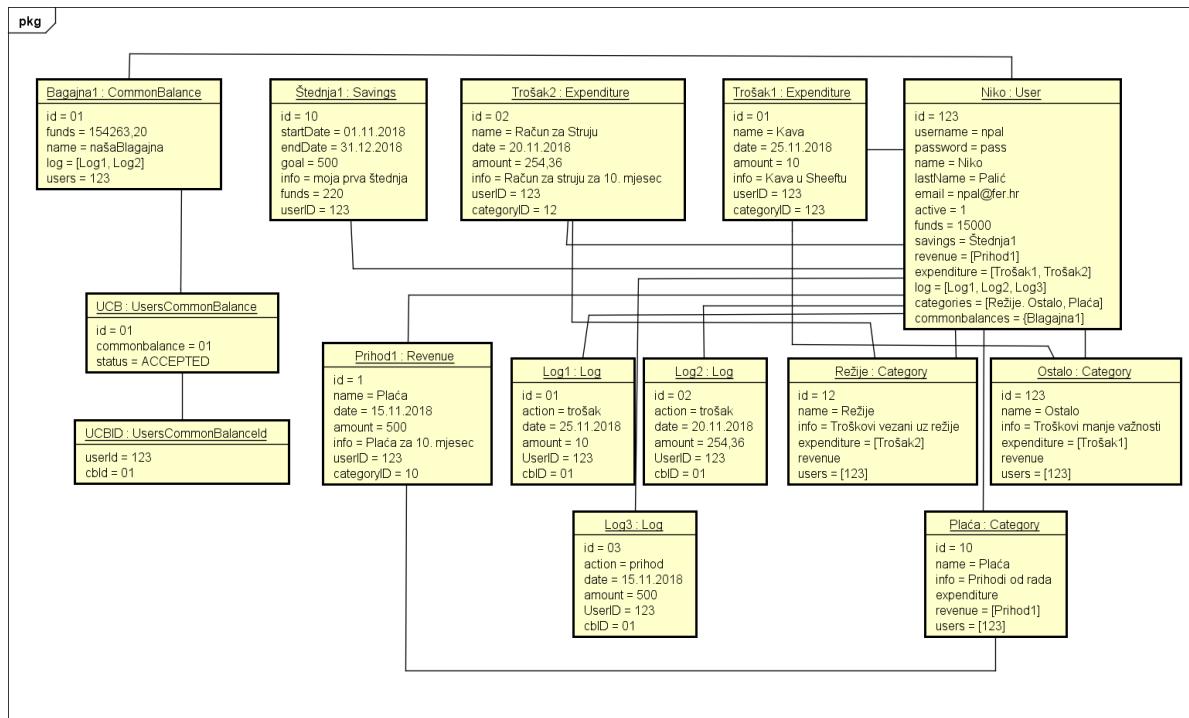
Repozitoriji predstavljaju razrede koji omogućuju izravnu komunikaciju s bazom podataka i dohvaćanje podataka iz baze podataka u okviru Java Spring tehnologije. Repoziotoriji predstavljaju podatkovni sloj.



Slika 6.2.4 Razredi paketa repozitoriji

6.3. Dijagram objekata

Dijagram objekata prikazuje stanje sustava u trenutku pregledavanja podataka korisnika Niko. U sustavu su prikazani svi potrebni podatci. To uključuje: Korisnik, troškovi, prihodi, logovi, blagajne, štednje te kategorije korisnika.



Slika 6.3.1 Dijagram objekata u sustavu

6.4. Ostali UML dijagrami

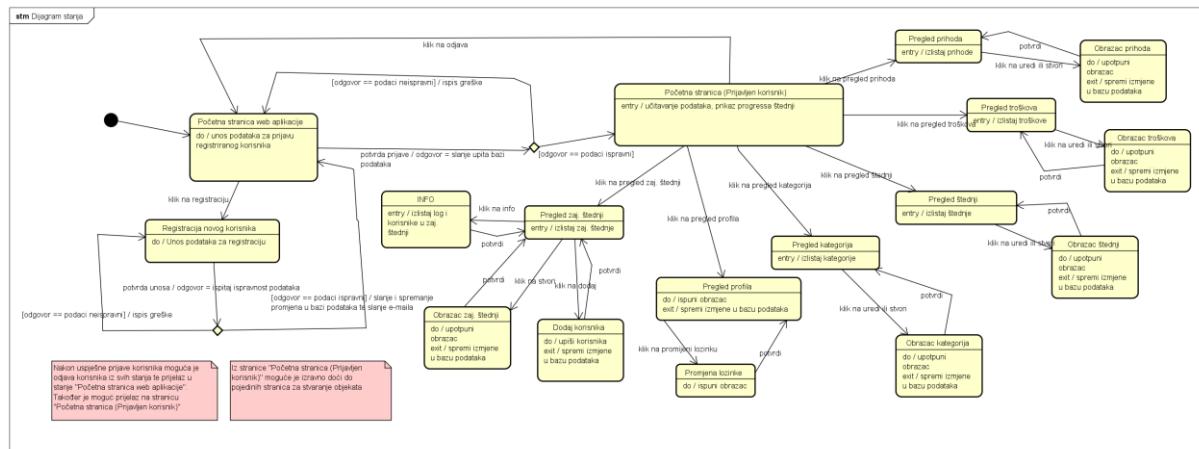
Dijagram stanja

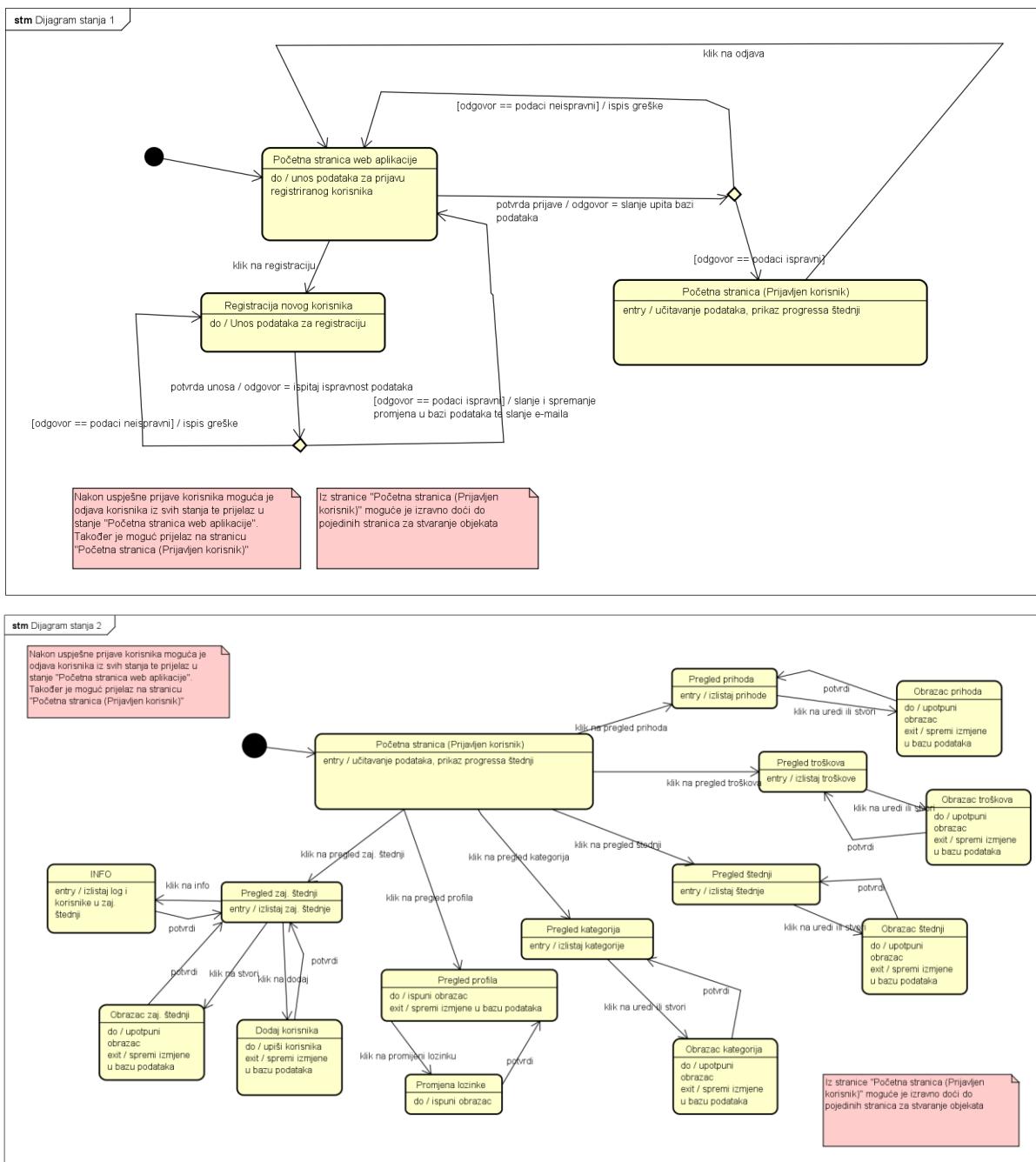
Dijagram stanja na slici 6.4.1 prikazuje stanja programa kao položaj korisnika unutar strukture web-stranica.

Upisivanjem web adrese aplikacije, korisniku se otvara početna stranica web aplikacije. Korisnik se može registrirati pritiskom na gumb za registraciju prilikom čega mu se otvara novi prozor za unos podataka potrebnih za registraciju. Zatim se provjerava ispravnost unesenih podataka te ukoliko su isti ispravni korisnik se proslijede na početnu stranicu, njegovi podaci se spremaju u bazu podataka te mu se šalje e-mail s potvrdom. Ukoliko su podaci neispravni korisnik ostaje u prozoru za registraciju.

Korisnik se na početnoj stranici može prijaviti kao registrirani korisnik pomoću ispravnog username-a i zaporke. Ukoliko su ti podaci neispravni korisniku se dojavljuje greška. Nakon što se korisnik prijavi u sustav učitavaju se njegove individualni podaci i štednje te mu se prikazuje početna stranica u prijavljenom stanju.

Pritom korisniku stoji na raspolaganju znatan broj opcija; može pregledavati (zaj.) štednju („Stranica pregled (zaj.) štednje“), kategorije („Stranica pregled kategorije“), troškove („Stranica pregled troškova“), vlastiti profil („Stranica pregled profila“) i prihode („Stranica pregled prihoda“). Ukoliko je prijavljen korisnik odabrao jednu od prvotno ponuđenih opcija na početnoj stranici („Početna stranica(Prijavljen korisnik)“), sada mu na raspolaganju stoje sljedeće opcije ovisno o stranici na kojoj se nalazi; može stvoriti vlastitu štednju („Stranica stvori štednju“), zajedničku štednju („Stranica stvori zajedničku štednju“), zasebne kategorije („Stranica stvori zasebnu kategoriju“). Također može unijeti svoje prihode i troškove („Stranica izmjeni prihod/trošak“) te izmijeniti korisničku lozinku („Stranica „Promjena lozinke“) i odjaviti se u bilo kojem trenutku, što ga ponovno vraća na početnu stranicu.



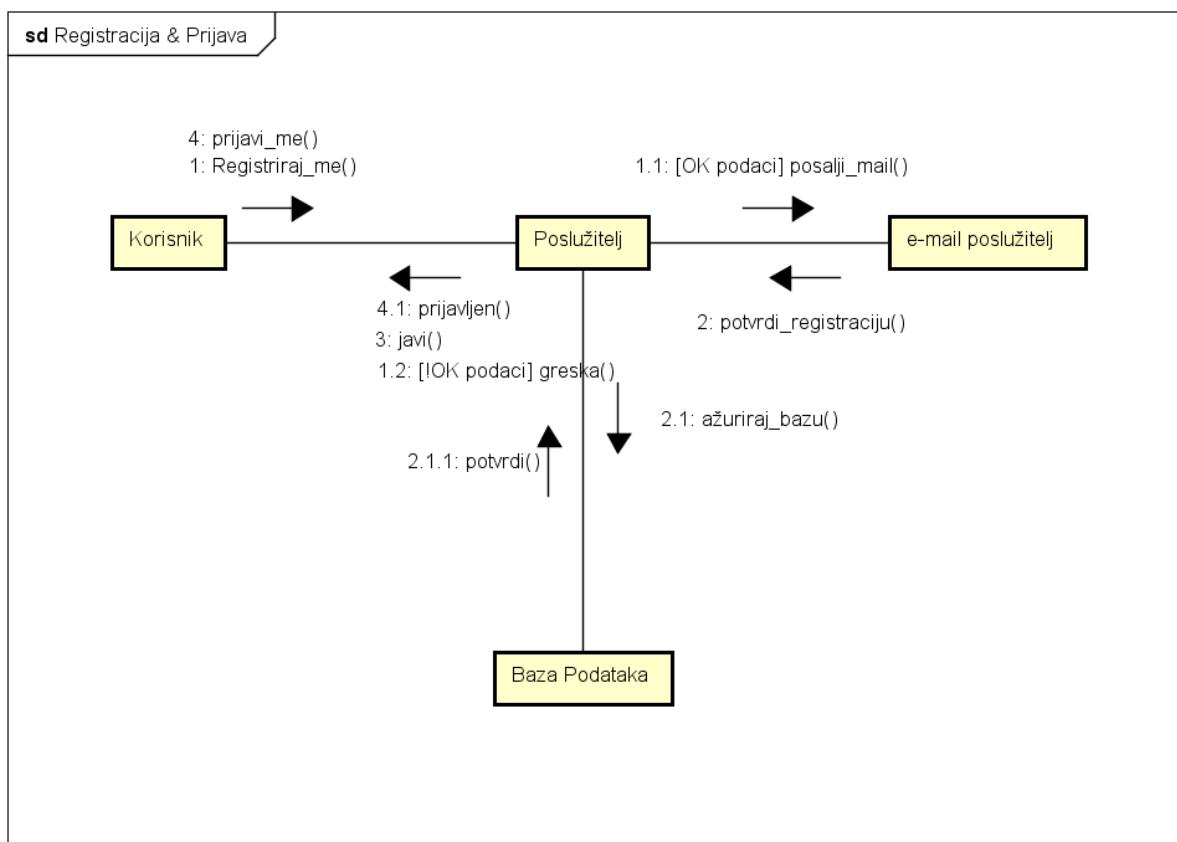


Slike 6.4.1 – 6.4.3 Dijagram stanja

Prva slika prikazuje cijeli dijagram, a preostale dvije su dijelovi dijagrama odvojeni radi preglednosti

Komunikacijski dijagram

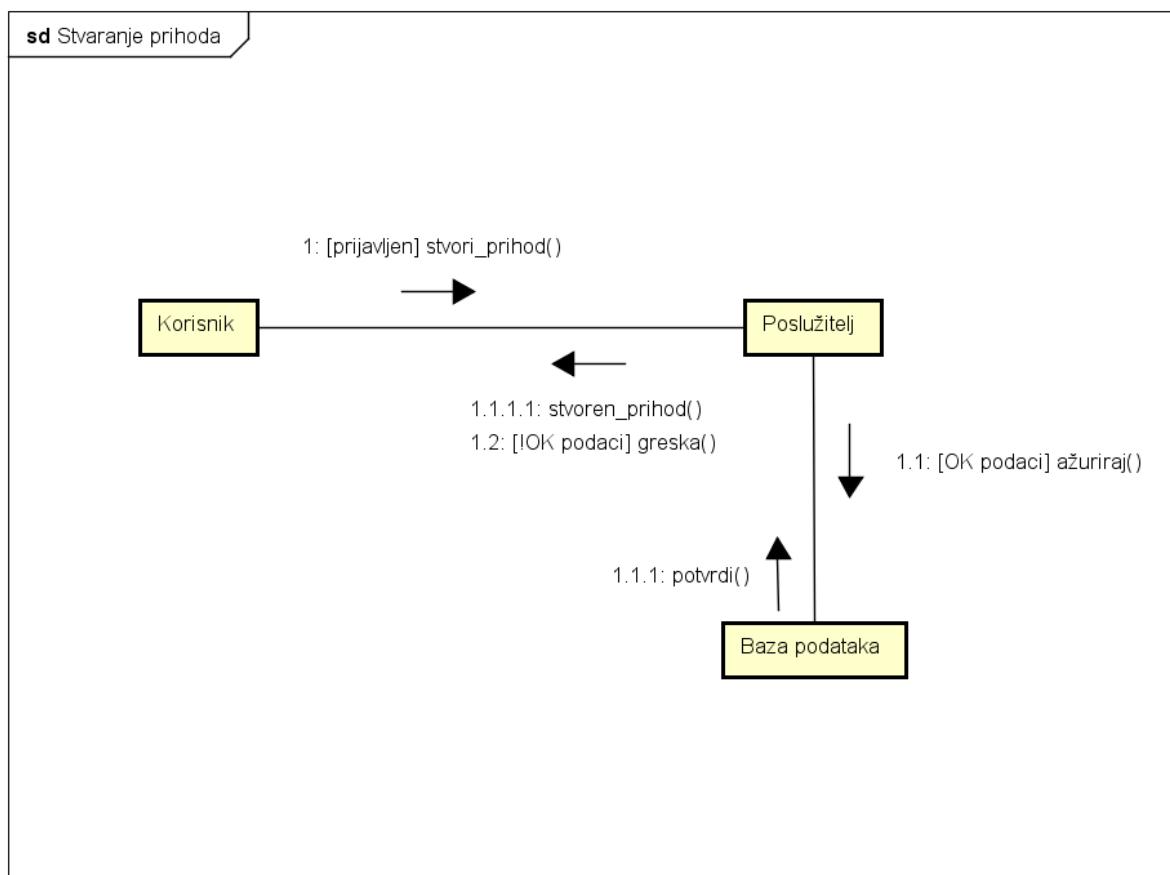
Komunikacijski dijagram na slici 6.4.2. prikazuje slijed događanja prilikom registracije korisnika te njegove prijave. Korisnik pritisne na gumb na registraciju, zatim mu se učita stranica za registraciju te korisnik ispunjava tražene informacije (adresu elektroničke pošte i lozinku). Poslužitelj podatke prosljeđuje bazi podataka na provjeru te ukoliko su podaci ispravni poslužitelj registrira korisnika u bazu podataka. Korisnik je tada obaviješten o uspjehnosti registracije te prima potvrdni email. Korisnik zatim putem web stranice unosi tražene podatke(adresu elektroničke pošte i zaporka) te pritišće na gumb za prijavu. Poslužitelj prosljeđuje zahtjev bazi podataka koja provjerava njihovu ispravnost te ovisno o rezultatu provjere, prijavljuje korisnika.



Slika 6.4.2. Komunikacijski dijagram registracije

Komunikacijski dijagram na slici 6.4.3 prikazuje slijed događaja prilikom stvaranja prihoda za sve korisnike koji su prijavljeni u sustav te se nalaze na „Stranica izmjeni prihod“.

Korisnik unosi tražene podatke te potvrđuje stvaranje prihoda, pritom se taj zahtjev šalje web poslužitelju. Poslužitelj putem baze podataka provjerava ispravnost unesenih podataka te obavještava korisnika o uspješnosti stvaranja prihoda ili ukoliko su podaci neispravni, javlja grešku.

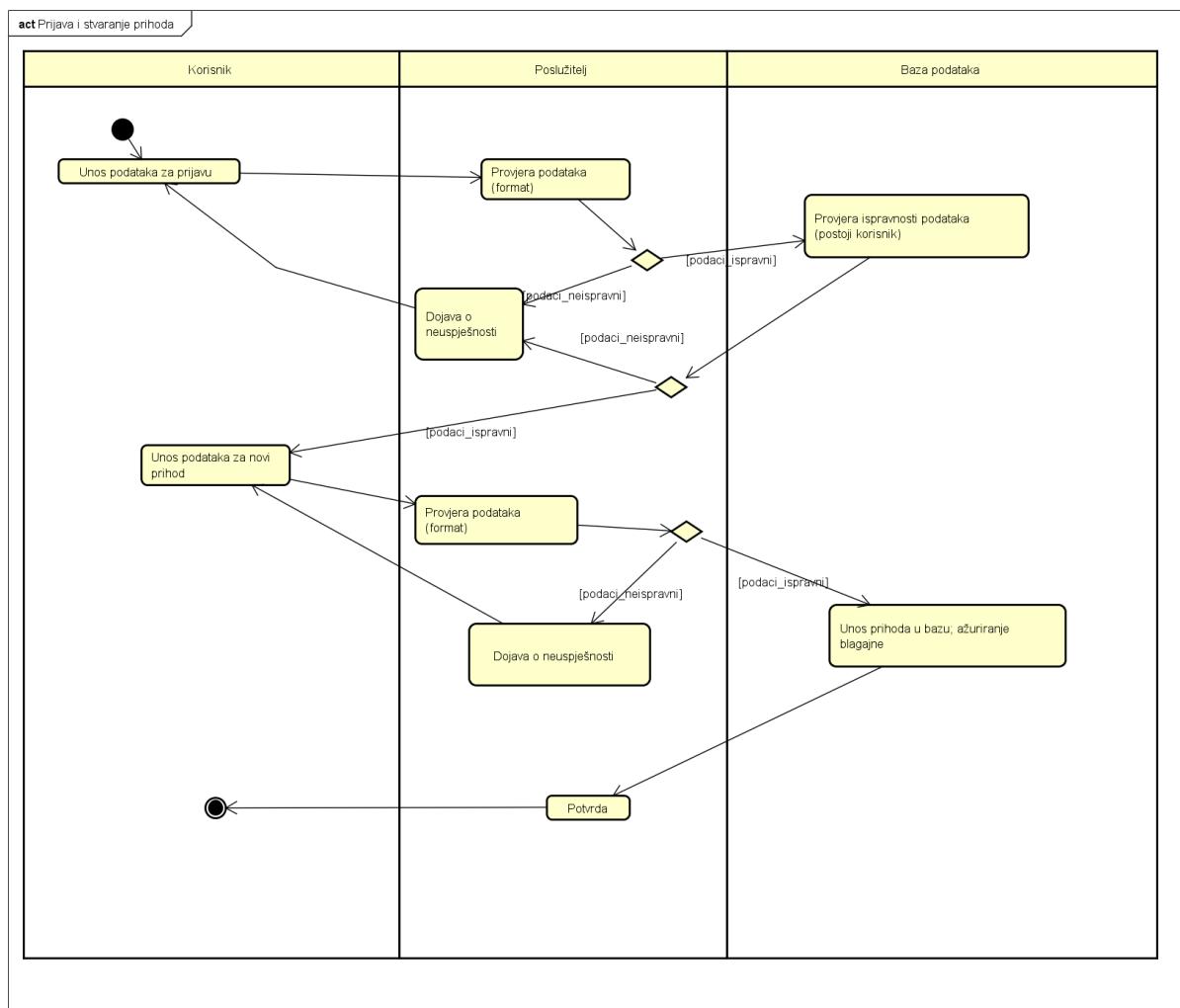


Slika 6.4.3. Komunikacijski dijagram stvaranja prihoda

Dijagram aktivnosti

Slika 6.4.4. prikazuje aktivnosti prilikom prijave registriranog korisnika i stvaranja prihoda. Korisnik na početnoj stranici web aplikacije unosi korisničke podatke korisničko ime i zaporka) te pritiskom na gumb za prijavu šalje web poslužitelju zahtjev za prijavu. Poslužitelj putem baze podataka provjerava ispravnost korisničkih podataka te ukoliko su podaci neispravni javlja grešku korisniku.

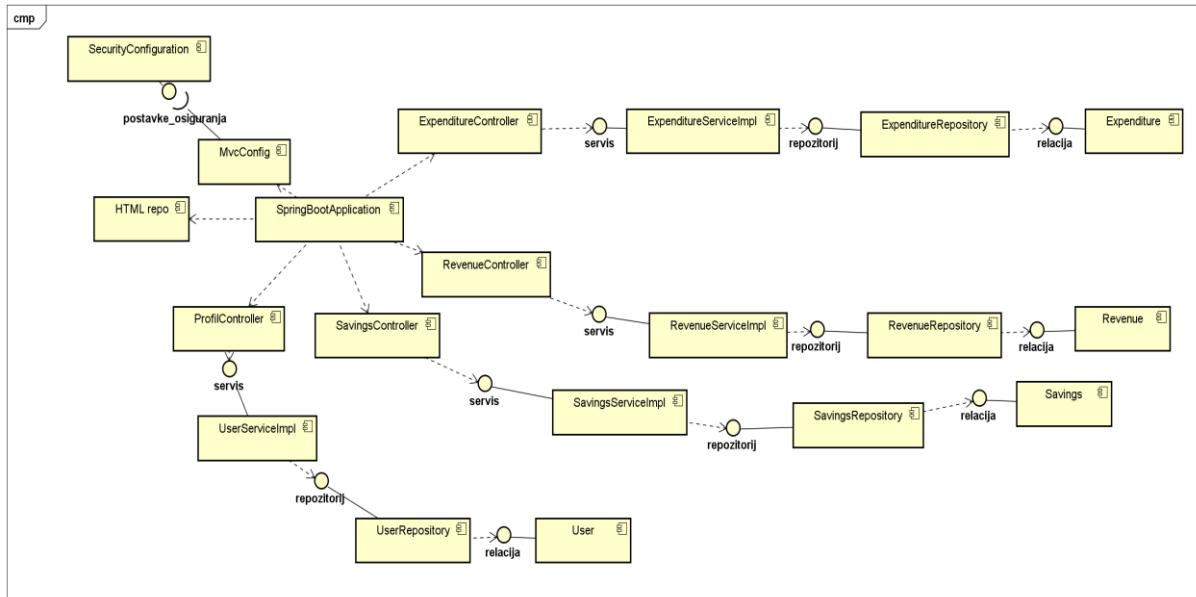
Zatim korisnik pritišće na gumb za stvaranje prihoda, unosi tražene podatke i potvrđuje stvaranje prihoda. Poslužitelj putem baze podataka provjerava ispravnost unesenih podataka, ukoliko su podaci ispravni spremaju se u bazu podataka te se obavještava korisnika o uspješnosti stvaranja prihoda. Ako su podaci neispravni, javlja se greška korisniku.



Slika 6.4.4. Dijagram aktivnosti
prilikom prijave i stvaranja
prihoda

Dijagram komponenti

Sljedeći dijagram komponenti prikazuje komponente od kojih je sustav izgrađen. Kako je program realiziran kao web aplikacija u radnom okviru Java Spring, dijagram ilustira komponente koje se koriste unutar tog radnog okvira. Zbog velikog broja komponenti i složenosti sustava ovdje je prikazan samo jedan njihov podskup.



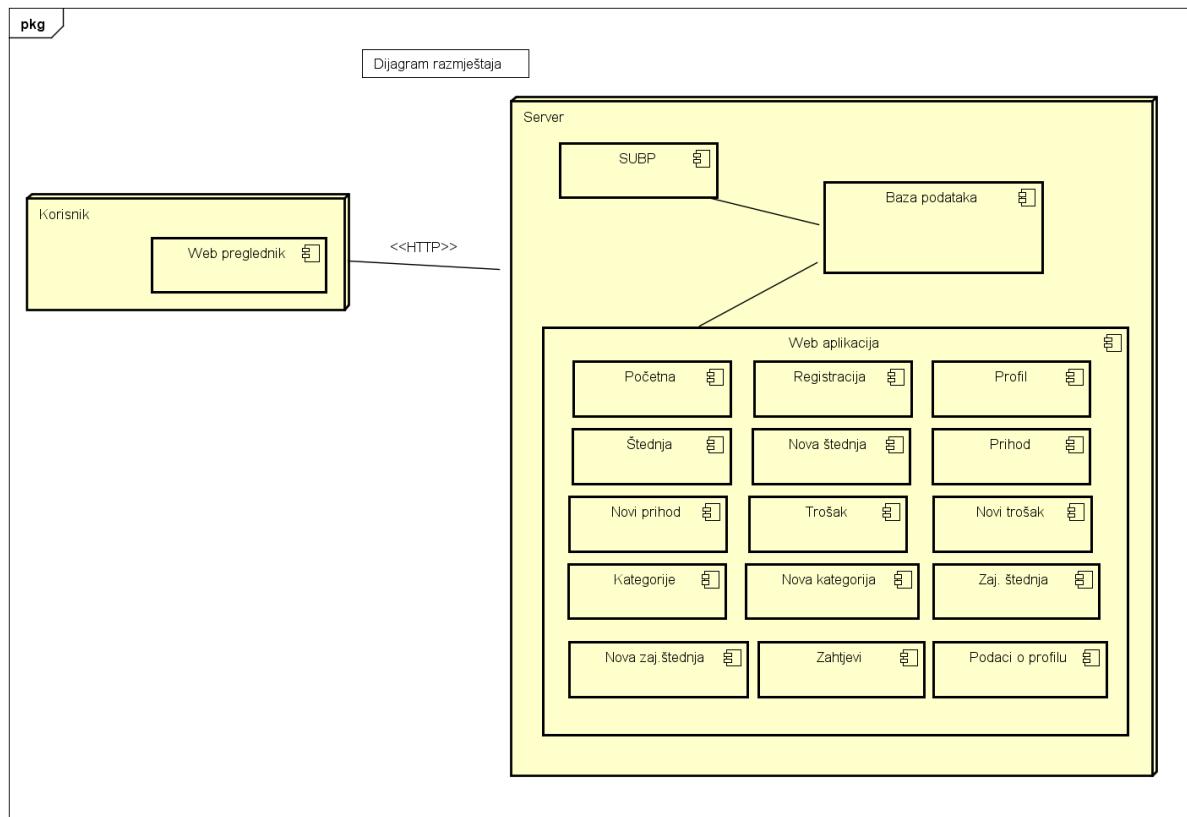
Slika 6.4.5. Dijagram komponenti

Na dijagramu je prikazan centralni pokretač aplikacije koji ovisi o nizu drugih komponenata. Prvo ovisi o komponenti MvcConfig, koja, zajedno sa SecurityConfiguration, upravlja korisničkim pogledima u sustavu. Nadalje, ovisi o upravljačima pojedinih podatkovnih objekata. Oni omogućuju upravljanje informacijama unutar sustava. Primjerice, ExpenditureController zadužen je za upravljanje nad troškovima u sustavu, a ovisi o ExpenditureServiceImpl, komponenti koja pruža uslugu manipulacije podacima upravljaču. Nadalje, ExpenditureServiceImpl ovisi o repozitoriju ExpenditureRepository koji realizira skup troškova, a sam ovisi o komponenti Expenditure. Konačno, komponenta Expenditure modelira n-torku troška u bazi podataka.

7. Implementacija i korisničko sučelje

7.1. Dijagram razmještaja

Na slici 7.1.1 prikazana je generalna topologija web-aplikacije bazirana na „klijent-poslužitelj“ arhitekturi. Komunikacija se u sustavu odvija između korisnika i poslužitelja putem HTTP veze. Web aplikacija je povezana s bazom podataka iz koje povlači podatke.



Slika 7.1.1. Dijagram razmještaja

7.2. Korištene tehnologije i alati

Bootstrap je alat koji nam omogućuje lakši rad s HTML-om, CSS-om i JavaScript-om.

CSS je stilski jezik, upotrebljava se za opis prezentacije dokumenta napisanog pomoću markup (HTML) jezika. Korišten je za prikaz aplikacije u web okruženju.

Eclipse je programska razvojna okolina (IDE) pisana u Javi, a može se koristiti za razvoj aplikacija u raznim programskim jezicima (Java, C,C++,PHP...). Eclipse koristi dodatke za pružanje svih funkcija unutar i na vrhu sustava za pokretanje. Njegov radni vremenski sustav temelji se na Equinoxu, implementaciji OSGi jezgrene specifikacije okvira.

Git je sustav za upravljanje inačicama koji omogućuje lakši timski rad i sustavno praćenje koda. Git je uparen s Gitlab središnjim repozitorijem. Također razvojna okolina koju smo odabrali (Eclipse IDE) omogućuje integraciju Git sustava prilikom izrade programskog koda web aplikacije.

Herokuapp Za izvedbu web poslužitelja korišten je Heroku web poslužitelj.

HTML (HyperText Markup Language) je prezentacijski jezik za izradu web stranica. Korišten je za prikaz aplikacije u web okruženju.

JavaScript je skriptni programski jezik, koji se izvršava u web pregledniku na strani korisnika. Napravljen je da bude sličan Javi, zbog lakšega korištenja, ali nije objektno orijentiran kao Java, već se temelji na prototipu i tu prestaje svaka povezanost s programskim jezikom Java.

Spring Framework je okvir aplikacije i inverzija kontrolnog spremnika za Java platformu. Osnovne značajke okvira mogu se koristiti s bilo kojom Java aplikacijom, ali postoje i proširenja za izgradnju web aplikacija na platformi Java EE (Enterprise Edition). Iako okvir ne nameće bilo koji specifičan programski model, postao je popularan u Java zajednici.

Thymeleaf je Java XML (Extensible Markup Language) / XHTML / HTML5 predložak koji može raditi na webu i ne-web okruženjima. Pogodnije je za posluživanje XHTML / HTML5 na sloju prikaza MVC web aplikacija, ali može obradivati bilo koju XML datoteku čak i u izvanmrežnim okruženjima. Pruža cjelovitu integraciju Spring Framework-a.

7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava

Sustav je temeljen na Java Spring Boot radnom okviru. On čini kostur aplikacije pa će zato i u ovom poglavlju biti fokus na njemu. Spring Boot povezuje front-end, tj. vanjski prikaz aplikacije (web-stranice) s back-endom, tj. bazom podataka u kojoj se trajno spremaju sve vrijednosti tijekom interakcije s programom.

Pokretanje programa:

```
@SpringBootApplication
public class PracenjeTroskovaApplication {

    public static void main(String[] args) {
        SpringApplication.run(PracenjeTroskovaApplication.class, args);
    }
}
```

Postavke sigurnosti aplikacije i obrana od neautoriziranog pristupa:

```
@Override
protected void configure(HttpSecurity http) throws Exception {

    http.csrf().disable();

    http.authorizeRequests()
        .antMatchers("/js/**")
        .permitAll();

    http.authorizeRequests()
        .antMatchers("/signup")
        .permitAll();

    http.authorizeRequests()
        .antMatchers("/login")
        .permitAll();

    http.authorizeRequests()
        .antMatchers("/**")
        .fullyAuthenticated()
        .anyRequest()
        .permitAll()
        .and()
        .formLogin().loginPage("/login")
        .permitAll();

    http.requiresChannel()
        .requestMatchers(r -> r.getHeader("X-Forwarded-Proto") != null)
        .requiresSecure();
```

```
}
```

Kod omogućuje pristup resursima stranice jedino u slučaju da je korisnik autoriziran (`.fullyAuthenticated()`). Pristup „login“ stranici je svima dopušten (`http.authorizeRequests().antMatchers("/login").permitAll();`).

Upravljači:

Upravljači (engl. *controller*) su jedan od sastavnih dijelova programa. Aplikacija sadrži upravljače za više funkcionalnosti, poput onih za registraciju, štednje, prihode.., a ovdje je prikazan dio upravljača za troškove:

```
@Controller
@RequestMapping("troskovi")
public class ExpenditureController {

    private final ExpenditureService expenditureService;
    private final UserService userService;

    @Autowired
    public ExpenditureController(ExpenditureService expenditureService,
        UserService userService) {
        super();
        this.expenditureService = expenditureService;
        this.userService = userService;
    }

    @PostMapping("/izbrisni/{id}")
    public String deleteTrosak(@PathVariable Long id, Principal principal) {
        User user=userService.findByUsername(principal.getName()).get();

        user.setFunds(user.getFunds()+expenditureService.findById(id).get().getAmount());
        userService.updateUser(user);
        expenditureService.deleteExpenditure(id);
        return "redirect:/profil/troskovi";
    }

    .....

    @RequestMapping("troskovi")
```

Kada korisnik dodje na stranicu za pregled troškova povezat će se na ovaj dio sustava.

```
@PostMapping("/izbrisni/{id}")
```

Kada korisnik pokrene akciju brisanja nekog troška pokrenut će se povezana funkcija `deleteTrosak` koja će kao argument automatski primiti id troška kojeg je potrebno obrisati, a u bazi podataka su troškovi identificarni id-jem.

```
User user=userService.findByUsername(principal.getName()).get();
```

Dohvaća se korisnik koji je pokrenuo akciju.

```
user.setFunds(user.getFunds() + expenditureService.findById(id).get().getAmount());
```

Ažurira se korisnikova varijabla „funds“ u bazi podataka koja prati stanje njegove blagajne. Pošto se trošak briše, program će povećati stanje blagajne za iznos troška, kao da nikad nije ni bio dodan.

```
userService.updateUser(user);
```

Korisnik će biti ažuriran u cijelom sustavu.

```
expenditureService.deleteExpenditure(id);
```

Iz baze podataka će se obrisati trošak.

Svi ostali upravljači funkcioniraju na sličan način, mapirajući korisničke zahtjeve (njegove klikove po stranici) s odgovarajućim funkcijama.

Klase baze podataka:

Svaka relacija u bazi je implementirana jednom klasom u Javi. Ovdje je dan primjer klase troškova:

```
@Entity
@Table(name = "expenditure")
public class Expenditure {

    @Id
    @GeneratedValue
    @Column(name="expenditure_id")
    private Long id;

    @NotBlank(message="Polje ne može ostati prazno")
    @Column(name="expenditure_name")
    private String name;

    @DateTimeFormat(pattern = "yyyy-MM-dd")
    @Column(name="expenditure_date")
    private Date date;

    @NotNull(message="Polje ne može ostati prazno")
    @NumberFormat(style=Style.NUMBER)
    @DecimalMin(value="0.0",message="Iznos mora biti pozitivan broj")
    @Column(name="expenditure_amount")
    private Double amount;

    @NotBlank(message="Polje ne može ostati prazno")
    @Column(name="expenditure_info")
    private String info;

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="user_id")
    private User userID;

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="category_id")
```

```

private Category categoryID;

public Expenditure() {}

public Expenditure(Long id, String name, Date date, Double amount, String info) {
    super();
    this.id = id;
    this.name = name;
    this.date = date;
    this.amount = amount;
    this.info = info;
}

```

Svaki stupac, tj. atribut relacije je programski realiziran kao privatna varijabla. Primjerice,

```

@Id
@GeneratedValue
@Column(name="expenditure_id")
private Long id;

```

ovako je implementirana „id“ varijabla relacije „trošak“. Pošto je ona ujedno i identifikator (ključ u terminologiji SQL-a) Iznad njega se stavlja oznaka @Id. Da ona predstavlja atribut relacije je označeno s @Column, uz koji je dano i ime koje će se koristit dalje u programu za referenciranje tog atributa.

Za atribute postoji razna ograničenja korespondentna onima u samoj bazi podataka. Tako npr. oznaka @NotNull osigurava da vrijednost atributa uvijek bude definirana.

Ovim klasama se program povezuje s (u našem slučaju) Postgres bazom podataka.

Repozitoriji:

Java Spring koristi posebna sučelja, tzv. repozitorije koji predstavljaju relacije baze. Metode deklarirane u repozitoriju ostvaruju se automatski.

```

public interface ExpenditureRepository extends JpaRepository<Expenditure, Long> {

    List<Expenditure> findByUserID (User userID);
    Optional<Expenditure> findById (Long id);
}

```

Tako metoda findByUserID realizira svoju funkcionalnost traženja relacije po ključu bez da je igdje eksplisitno implementirana. Java Spring zna koja je funkcionalnost tražena zbog sintakse naziva metode.

Servisi:

Servisi služe za manipulaciju relacijama baze podataka. Sadrže metode brisanja, dodavanja, traženja.. n-torki neke relacije.

```

public interface ExpenditureService {

```

```
void createExpenditure(Expenditure expenditure);
List<Expenditure> findByUserID (User userID);
void deleteExpenditure (Long id);
Optional<Expenditure> findById(Long id);
void saveExpenditure(Expenditure expenditure);
}
```

Iznad je sučelje koje implementira klasa servisa za troškove. Primjerice metoda te klase `createExpenditure` je implementirana ovako:

```
@Override
public void createExpenditure(Expenditure expenditure) {
    expenditureRepository.save(expenditure);

}
```

Metoda koristi repozitorij za troškove da bi spremila novi trošak.

U ovom odjeljku prikazan je maleni dio aplikacije, a cijeli kod je znatno opširniji i kompleksniji tako da nije prikazan u potpunosti u ovome dokumentu. Uz prikazane klase postoje i razne druge, primjerice, za validaju e-maila pri registraciji korisnika i enumeriranje akcija koje korisnik može poduzeti u zajedničkoj štednji.

7.4. Ispitivanje programskog rješenja

Ispituju se temeljne funkcionalnosti svih razina korisnika radi utvrđivanja pogrešaka u programskom rješenju te sprječavanja nepoželjnog rada programske podrške. Također se ispituju rubni uvjeti rada sustava koji bi mogli uzrokovati rušenje sustava.

Test 1

Opis ispitnog slučaja: Stvaranje korisničkog računa. Potrebni su ime, prezime, adresa elektroničke pošte, lozinka i potvrda lozinke.

Očekivani rezultat: Korisnik unosi sve tražene podatke i ukoliko su svi podaci ispravni, stvara se novi korisnički račun i korisniku se šalje potvrđena poruka na adresu elektroničke pošte. Ukoliko postoji greška korisniku će se dojaviti zašto stvaranje novog računa nije provedeno, odnosno što je krivo u popunjavanju podataka.

Dobiveni rezultat: U slučaju ispravnog popunjavanja svih polja novi račun je stvoren. Ukoliko sva polja nisu bila popunjena korisniku se ispisuje greška. Također unos u polja lozinke i ponovi lozinku moraju se podudarati te je potrebna jaka lozinka (min. duljina 8 znakova, barem 1 veliko slovo, barem 1 broj i barem 1 poseban znak), ukoliko neki od tih uvjeta nisu zadovoljeni korisnik dobiva prikladnu poruku. Ako postoji greška u adresi elektroničke pošte (ne može se poslati poruka, postoji već korisnik sa tom adresom) ili neka druga greška sustav o tome obavještava korisnika.

Registracija

The screenshot shows a registration form with several input fields and their corresponding error messages:

- First name:** "Unesite ime" (Enter first name) - Error message: "First name required".
- Last name:** "Unesite prezime" (Enter last name) - Error message: "Last name required".
- Email:** "Unesite email" (Enter email) - Error message: "Invalid email address!". Below this, a red box contains the message: "must not be empty".
- Password:** "Unesite lozinku" (Enter password) - Error message: "Password required". Below this, a red box contains the message: "Password must be at least 8 characters in length., Password must contain at least 1 uppercase characters., Password must contain at least 1 digit characters., Password must contain at least 1 special characters.". Below this, another red box contains the message: "size must be between 1 and 2147483647".
- Re-enter password:** "Ponovno unesite lozinku" (Re-enter password) - Error message: "size must be between 1 and 2147483647".

Slika 7.4.1. Ispis greške zbog nepotpunjenih polja za registraciju

Registracija

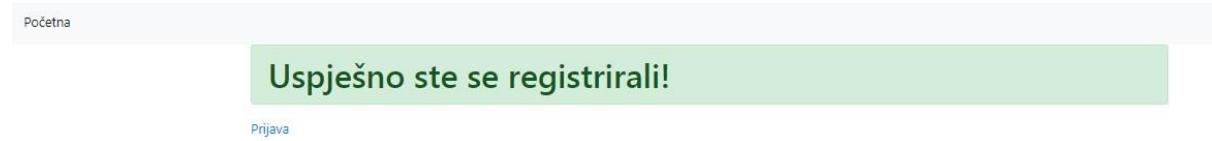
The screenshot shows a registration form with several input fields and error messages. The fields contain the following values:

- First Name: Fabian Leskovec
- Last Name: Nikić
- Address: Nic
- Email: leskovec.fabian@gmail.com
- Password (Visible): ****
- Password (Error Message): Your password is too short
- Confirm Password (Visible): ****
- Confirm Password (Error Message): size must be between 1 and 2147483647

A red callout box highlights the password error message, which states: "Password must be at least 8 characters in length., Password must contain at least 1 uppercase characters., Password must contain at least 1 digit characters., Password must contain at least 1 special characters."

At the bottom right is a green "Pošalji" button.

Slika 7.4.2. Ispis greške zbog
upisa već zauzete adrese el. pošte
te zbog nezadovoljavajuće
lozinke



Slika 7.4.3. Uspješno stvaranje
novog korisničkog računa

Test 2

Opis ispitnog slučaja: Korisnik se prijavljuje u sustav.

Očekivani rezultat: Korisnik je prijavljen u sustav i otvara mu se („Početna stranica (Prijava korisnik)“) ukoliko su njegovi uneseni podaci (korisničko ime i lozinka) ispravni.

Dobiveni rezultat: Uspješna prijava korisnika u sustav ukoliko su uneseni podaci ispravni. Ispis poruke o grešci prilikom upisa neispravnih podataka.

The screenshot shows a login form with fields for 'Korisničko ime/lozinka' (Username/password) and 'Potvrditi email' (Confirm email). Below the form, a red error message box displays the text 'Krivo korisničko ime/lozinka ili niste potvrdili email!' (Wrong username/password or you have not confirmed your email!). To the right of the error message are two buttons: 'PRIJAVA' (Login) in green and 'REGISTRACIJA' (Registration) in blue.

Slika 7.4.4. Ispis poruke o grešci
prilikom upisa neispravnih
podataka

The screenshot shows a welcome page with the heading 'Dobro došli na Praćenje troškova'. Below it is a sub-headline 'Smanjite svoje troškove i uštedite s nama.' (Save your expenses and save with us.) and a call-to-action 'Pridružite nam se i pomozite sami sebi.' (Join us and help yourself.)

On the right side, there is a user profile section with the name 'Korisnik Fabian' and a green button labeled 'Odjavi se' (Logout).

At the bottom, there is a section titled 'Smanjite troškove, uštedite.' (Save money, save) with a descriptive text about the service's purpose.

Slika 7.4.5. Otvaranje početne
stranice prilikom uspješne prijave

Test 3

Opis ispitnog slučaja: Korisnik stvara štednju.

Očekivani rezultat: Korisnik pritišće na tipku „Nova štednja“, pritom mu se otvara stranica za stvaranje štednje i korisnik unosi tražene podatke za stvaranje štednje. Ukoliko neki od podataka nisu ispravni korisnik dobiva poruku o pogrešci.

Dobiveni rezultat: Sustav je uspješno obavijestio korisnika o pogreškama te nakon ispravka istih o uspješnosti stvaranja štednje.

Štednje

16.01.2019.	
14.01.2019.	Krajnji datum je prije početnog
SD	Iznos mora biti broj
dsa	Iznos mora biti broj
štедnja za stan	
Stvori	

[Natrag na pregled štednji](#)

Slika 7.4.6. Ispis i opisi pogrešaka prilikom unosa podataka

Štednje

16.01.2019.

16.01.2019.

1.500

Nedovoljno sredstva u blagajni

1.500

štедnja za stan

Stvor

[Natrag na pregled štednji](#)

Slika 7.4.7. Ispis poruke o grešci (provjera sredstva u blagajni)

Početna Profil Nova štednja Pretraga Korisnik Fabian Odjavi se

Uspješno ste stvorili novu štednju!

Datum početka	Datum kraja	Cilj	Opis	Sredstva	Uplata	Isplata	Uredi	Obriši
16.01.2019	16.01.2019	2000.0	Štednja za stan	1500.0				

Slika 7.4.8. Uspješno stvaranje štednje

Test 4

Opis ispitnog slučaja: Korisnik stvara prihod.

Očekivani rezultat: Korisnik pritišće na tipku „Novi prihod“, pritom mu se otvara stranica za stvaranje prihoda i korisnik unosi tražene podatke za stvaranje prihoda. Ukoliko neki od podataka nisu ispravni korisnik dobiva poruku o pogrešci.

Dobiveni rezultat: Sustav je uspješno obavijestio korisnika o pogreškama te nakon ispravka istih o uspješnosti stvaranja štednje.

Prihodi

Unesite naziv prihoda

Unesite iznos

16.01.2019.

Unesite opis

Stvori

Natrag na pregled prihoda

Slika 7.4.9. Ispis greške zbog nepotpunjениh polja za registraciju

Prihodi

Prihod

ds

16.01.2019.

Prihod1

Stvori

Natrag na pregled prihoda

Slika 7.4.10. Ispis greške zbog krivog formata



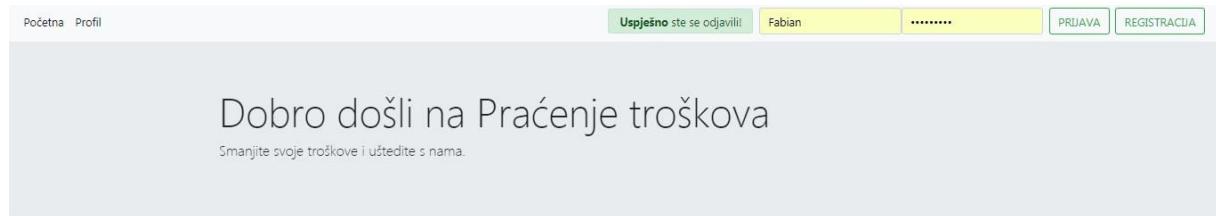
Slika 7.4.11. Uspješno stvoren prihod

Test 5

Opis ispitnog slučaja: Korisnik odabire da se želi odjaviti iz sustava. Nakon promjena u bazi korisnika se vraća na početnu stranicu.

Očekivani rezultat: Nakon odjave korisnik mora biti vraćen na početnu stranicu.

Dobiveni rezultat: Korisnik je uspješno odjavljen iz sustava i vraćen na početnu stranicu.



Slika 7.4.12. Prikaz uspješne odjave

Test 6

Opis ispitnog slučaja: Korisnik odabire postavke profila. Zatim mu se otvara stranica sa podacima korisničkog računa (ime, prezime, adresa el. pošte) te korisnik unosi i potvrđuje željene promjene. Nakon promjena u bazi korisnika se vraća na početnu stranicu.

Očekivani rezultat: Nakon što korisnik uneše željene podatke te potvrdi spremanje promjena, iste se spremaju u bazu podataka te se korisnika vraća na početnu stranicu.

Dobiveni rezultat: Korisnik je uspješno promijenio podatke korisničkog računa i vraćen je na početnu stranicu.

Početna Profil Sredstva: 0,00 Korisnik Fabian Odjavi se

Podaci o profilu

Korisničko ime: Fabian
[Promjeni lozinku](#)

Spremi

[Natrag na profil](#)

Slika 7.4.13. Prikaz korisničkih podataka

Početna Prihod ▾ Štednja ▾ Zajednička štednja ▾ Trošak ▾ Kategorija ▾ Sredstva: 0,00 Korisnik Josip Odjavi se

Dobro došli na praćenje troškova

Smanjite svoje troškove i užtedite s nama.

Štednje
Štednja: Štednja za stan
75,00 %

Slika 7.4.14. Prikaz uspješne promjene korisničkih podataka

Test 7

Opis ispitnog slučaja: Korisnik odabire ikonu za uplatu u štednju. Zatim mu se otvara stranica za unos željenog iznosa uplate. Korisnik unosi i potvrđuje željenu uplatu. Nakon promjena u bazi korisniku se prikaže ažurirana štednja.

Očekivani rezultat: Nakon što korisnik pritisne na ikonu za uplatu, otvara mu se nova stranica za unos iznos uplate. Korisnik unosi željeni iznos, sustav mu javlja ispravnost formata te nakon što korisnik odabire potvrdu promjena iste se spreme u bazu podataka. Korisnik se vraća na stranicu s ažuriranom štednjom.

Dobiveni rezultat: Korisnik je uspješno izveo uplatu u štednju i vraćen je na stranicu štednje.

Datum početka	Datum kraja	Cijj	Opis	Sredstva	Uplata	Isplata	Uredi	Obrisi
16.01.2019	16.01.2019	2000.0	Štednja za stan	1500.0				

Slika 7.4.15. Prikaz odabira ikone za uplatu u štednju

ds

Iznos mora biti broj

Upłati

Natrag na pregled štednji

Slika 7.4.16. Prikaz ispisa greške zbog krivog formata

ds

Iznos mora biti broj

Upłati

Natrag na pregled štednji

200

Nedovoljno sredstava u blagajni

Upłati

Natrag na pregled štednji

Slika 7.4.17. Prikaz ispisa greške zbog nedovoljnih sredstava blagajne

The screenshot shows a user interface for managing savings goals. At the top, there are navigation links: Početna, Profil, Nova štednja, Pretraga, Sredstva: 1.300,00, Korisnik Fabian, and Odjavi se. A green banner at the top center says "Uspješno ste uplatili 200,0 u štednjusu: Štednja za stan". Below this is a table with the following data:

Datum početka	Datum kraja	Cilj	Opis	Sredstva	Uplata	Isplata	Uredi	Obrisi
16.01.2019	16.01.2019	2000,0	Štednja za stan	1700,0				

Slika 7.4.18. Prikaz uspješne
uplate u štednju

7.5. Upute za instalaciju

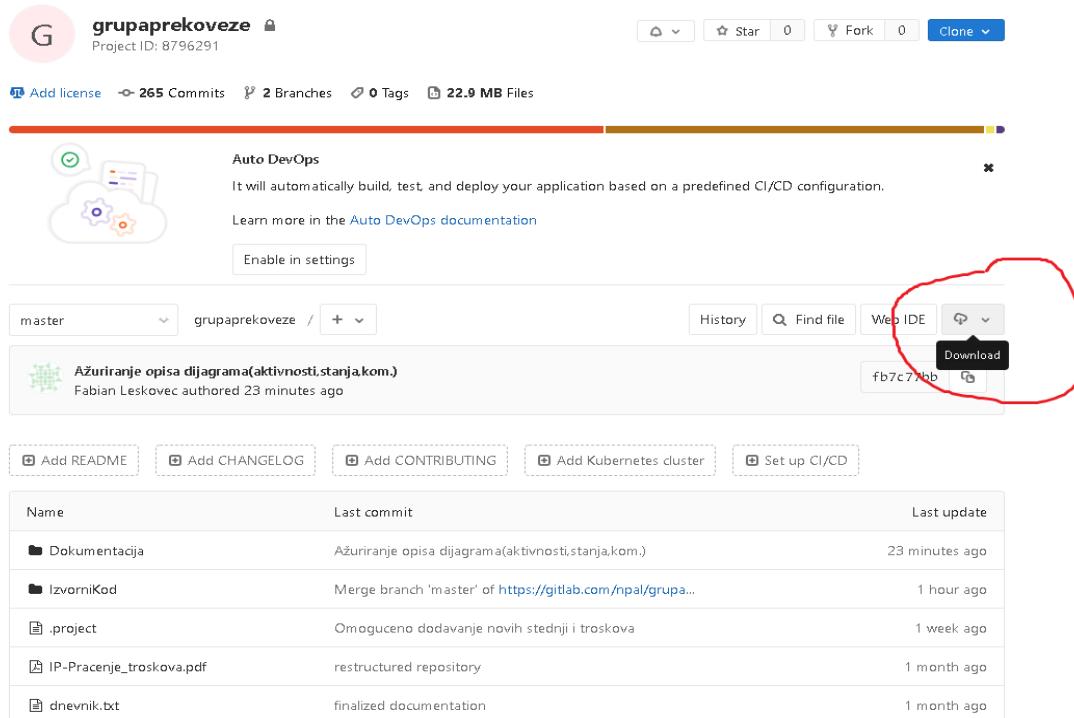
Aplikacija se može ispravno pokrenuti na više različitih načina, a ovdje je prikazan jedan od jednostavnijih.

Aplikacija se izvodi u programskom jeziku Java stoga je potrebno imati odgovarajuću potporu za izvođenje samog jezika. Prije instalacije potrebno je na računalu imati instaliranu Javu: <https://www.java.com/en/download/>

Za jednostavnije pokretanje i rukovanje programom, predlaže se koristiti jednu od platformi za programiranje u Javi. Korisno je imati instaliran JAVA IDE, primjerice Eclipse: <https://www.eclipse.org/downloads/>

Sustav baze podataka koju aplikacija koristi je Postgres pa je i njega potrebno instalirati: <https://www.postgresql.org/download/>

Sa GitLab stranice treba se skinuti repozitorij i iz njega uzeti folder „Praćenje troškova“ koji sadrži izvorni kod programa. Taj folder nalazi se unutar foldera IzvorniKod.



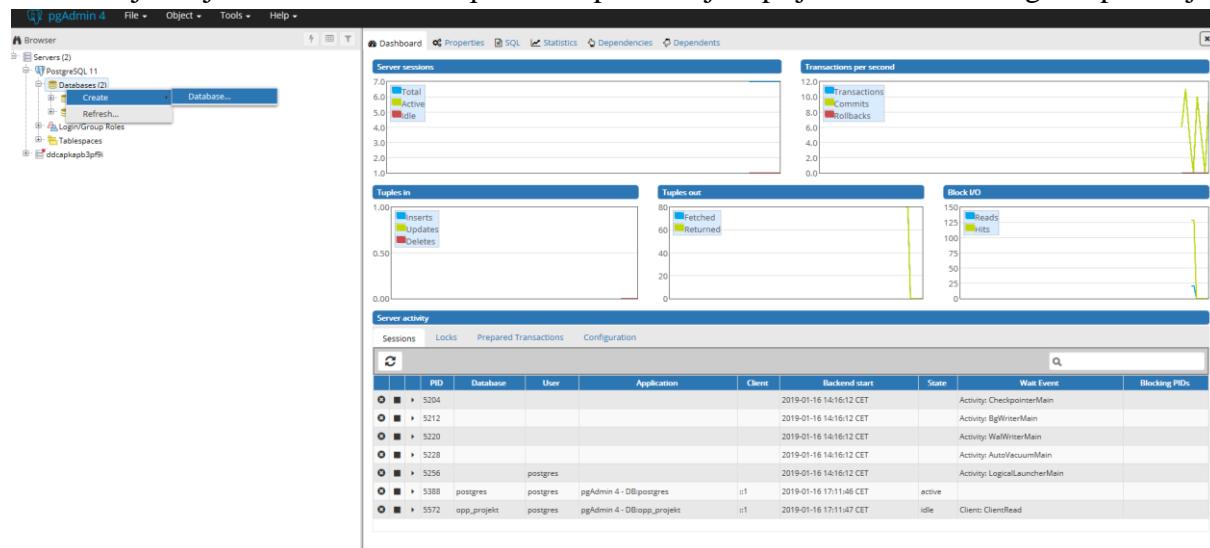
Slika 7.5.1. GitLab sučelje

Ukoliko u programu Eclipse nije već integriran Maven, potrebno je i njega instalirati. Nove verzije Eclipse-a već podržavaju Maven, ali u slučaju da nije tako, ovdje se nalazi tutorial:

<http://www.vogella.com/tutorials/EclipseMaven/article.html#installation-and-configuration-of-maven-for-eclipse>

U programu Eclipse potrebno je importati novi „Maven project“ tako se da desnim klikom miša klikne na odjeljak za projekte i pritisne Import. Nakon toga bit će ponuđen obrazac za stvaranje novog projekta. Iz prikaza potrebno je odabrat „Existing Maven Project“ i kao izvor projekta odabrat prije locirani folder „Praćenje troškova“. Sada je projekt učitan u Eclipse.

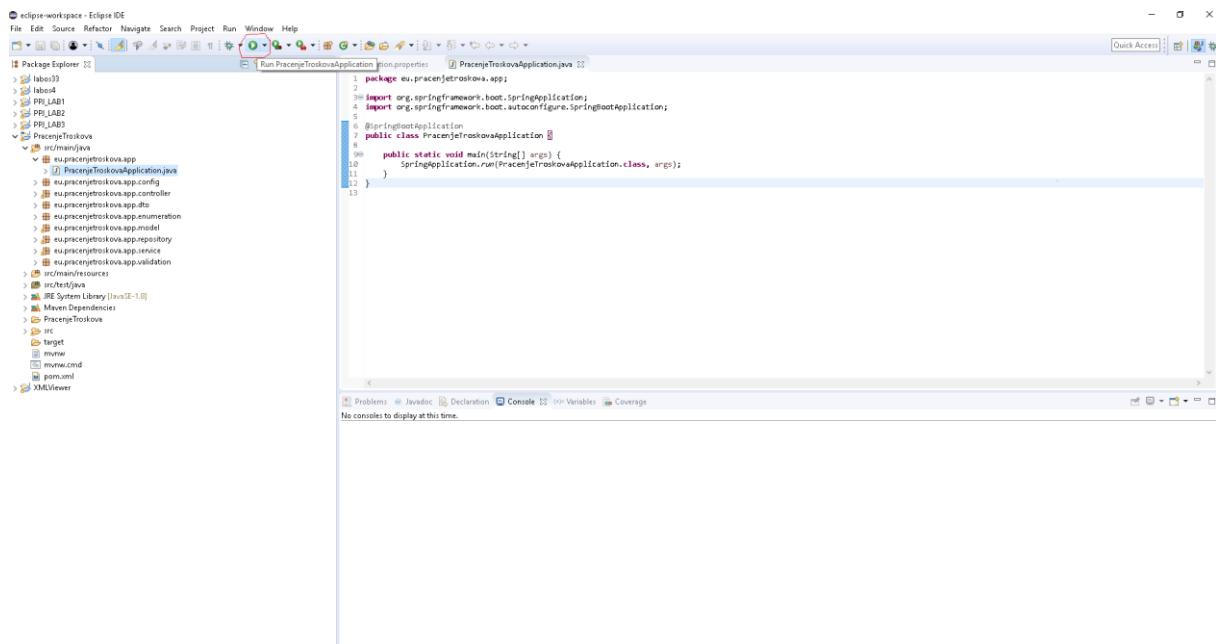
Potrebno je uključiti server za bazu podataka pokretanjem prije instalirane Postgres aplikacije.



Slika 7.5.2. Stvaranje baze podataka

Potrebno je stvoriti novu bazu podataka imena „opp_program“. Program pristupa bazi kao korisnik „postgres“ kojemu je nužno postaviti lozinku na „pass“ naredbom „ALTER USER postgres PASSWORD 'pass'“. Naredba se jednostavno upiše tako da se pokrene „Query Tool“ desnim klikom miša nad bazom i pokrene pritskom na tipku F5. Sada je baza podataka spremna.

Aplikacija se konačno pokreće pokretanjem java programa „PracenjeTroskovaApplication.java“ tako da se u Eclipsu on otvori i pritisne na gumb za pokretanje. Aplikacija sada radi!



Slika 7.5.3. Pokretanje aplikacije

Aplikaciji se na računalu može pristupiti bilo kojim internet preglednikom upisivanjem „localhost:8080/“ kao adresu.

7.6. Korisničke upute

Koristeći bilo koji web preglednik korisnik može pristupiti web aplikaciji. Sama aplikacija je napravljena tako da bude intuitivna za korištenje prosječnom korisniku Interneta. Na početnoj stranici korisnik ima sljedeće 2 mogućnosti: može se registrirati ili ukoliko već ima korisnički račun može se prijaviti.

The screenshot shows a login form with the following elements:

- Top navigation bar with links "Početna" and "Profil".
- Input fields for "Korisničko ime" and "Lozinka".
- Buttons for "PRIJAVA" and "REGISTRACIJA".
- Main content area with the heading "Dobro došli na Praćenje troškova" and the subtext "Smanjite svoje troškove i uštedite s nama."

Slika 7.6.1. Prikaz sučelja za prijavu

Registracija

The screenshot shows a registration form with the following fields:

- "Unesite ime"
- "Unesite prezime"
- "Unesite korisničko ime"
- "Unesite email"
- "Unesite lozinku" (password field)
- A progress bar indicating the status of the registration process.
- "Ponovno unesite lozinku" (re-enter password field).

Slika 7.6.2. Prikaz sučelja za registraciju

Ukoliko korisnik posjeduje korisnički račun prijavu vrši tako da u polje „Korisničko ime“ unosi ime svog korisničkog računa dok u polje „Lozinka“ unosi lozinku svog računa te zatim pritišće na gumb „Prijava“.

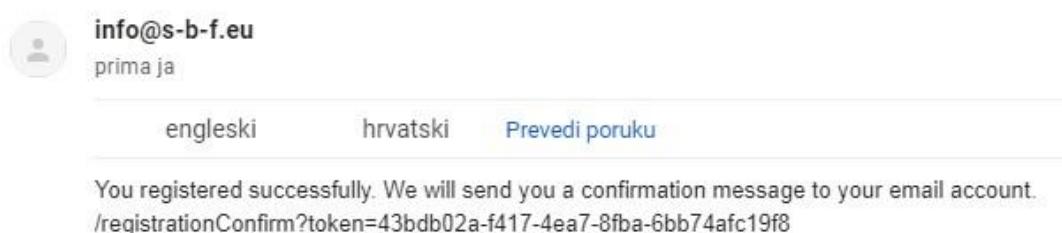
Ako korisnik nema vlastiti korisnički račun, tada pritiskom na gumb „Registracija“ može stvoriti jedan. Zatim se otvara nova stranica (Slika 7.6.2.) na kojoj korisnik u određena polja, u kojima su označeni traženi podaci, unosi podatke.

Napomena: Za uspješnu registraciju sva polja moraju biti popunjena i korisnik mora unijeti istu lozinku u oba polja!

Nakon unosa svih potrebnih podataka korisnik pritišće na gumb „Pošalji“ kako bi stvorio korisnički račun. Korisnik dobiva potvrdu poruku prilikom uspješnog stvaranja korisničkog računa, ili poruku greške ukoliko je došlo do pogreške.

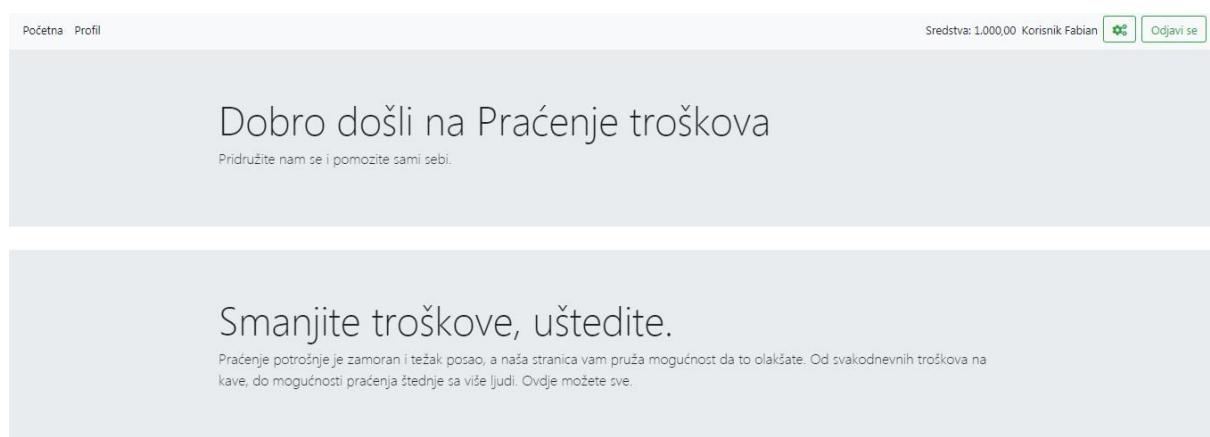
Ukoliko je korisnički račun uspješno stvoren poslužitelj šalje verifikacijskih mail na korisnikovu adresu el. pošte. Korisnik zatim pritišće na poveznicu u verifikacijskom mail-u čime se dovršava stvaranje novog korisničkog računa.

Registration Confirmation



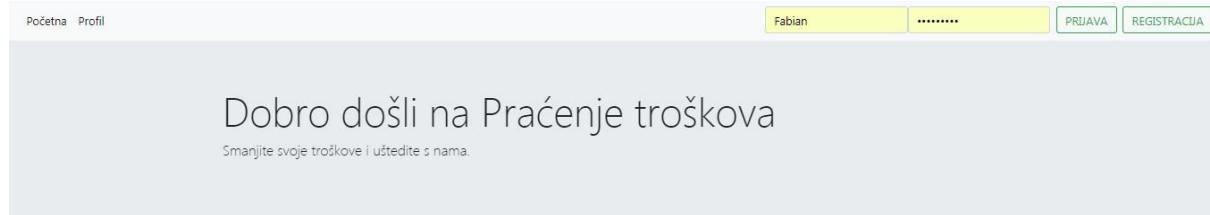
Slika 7.6.3. Potvrda o verifikaciji
adrese el. pošte

Nakon što se korisnik prijavi ili stvori novi korisnički račun (i zatim se prijavi) otvara mu se početna stranica („Prijavljen korisnik“) web aplikacije (Slika 7.6.4.). U gornjem desnom dijelu stranice korisnik vidi iznos raspoloživih sredstava te svoje korisničko ime.

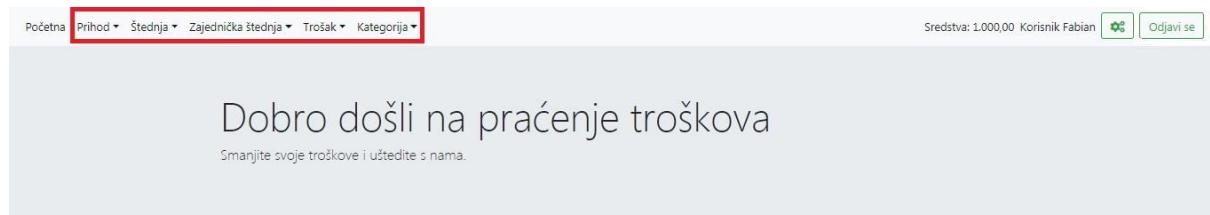


Slika 7.6.4. Prikaz početne
stranice(„Prijavljen korisnik“)
nakon prijave

Nadalje, može pritisnuti na gumb „Početna“ čime se vraća na početnu stranicu web aplikacije (Slika 7.6.5.), može pritisnuti na gumb „Profil“ čime mu se otvara stranica njegovog profila (Slika 7.6.6.) na kojemu mu se nude dodatne mogućnosti te prikaz aktualnih štednji.



Slika 7.6.5. Prikaz početne stranice web aplikacije



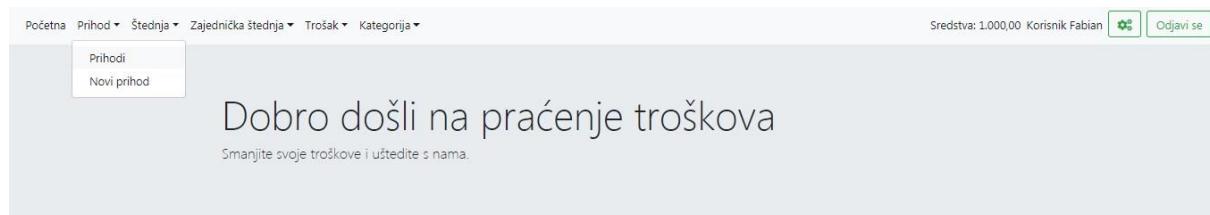
Štednje

Štednja: Štednja za stan

100.00 %

Slika 7.6.6. Prikaz stranice nakon pritiska na gumb „Profil“ i dodatne mogućnosti(označeno crvenim)

Korisnik pritiskom na jednog od ponuđenih gumba („Prihod“, „Štednja“, „Trošak“,...) otvara padajući prozor (Slika 7.6.7.) koji mu nudi opcije pregleda i promijene prihoda, troškova, (zaj.)štедnji i kategorija.



Štednje

Štednja: Štednja za stan

100.00 %

Slika 7.6.7. Prikaz padajućeg prozora pritiskom na gumb „Prihodi“

Dalnjim pritiskom na gumb „Prihodi“ korisnik otvara novu stranicu (Slika 7.6.8.) na kojoj su mu prikazani svi njegovi prihodi sa nazivom, datumom unosa, iznosom prihoda, opisom, kategorijom te mogućnostima uređivanja i brisanja prihoda.

Naziv	Datum	Iznos	Opis	Kategorija	Uredi	Obrisi
Prihod	16.01.2019	1500.0	Prihod1	NIJE UPISANA		
ds	16.01.2019	1500.0	12	NIJE UPISANA		

Slika 7.6.8. Prikaz uneseni
prihoda, ikona za uređivanje
(označena crvenim) i ikona za
brisanje (označena zelenim)

Korisnik može urediti prihode pritiskom na ikonu oblika olovke kojim mu se otvara stranica za uređivanje prihoda (Slika 7.6.9.). Zatim unosi željene promjene prihoda te ih potvrđuje pritiskom na gumb „Spremi“ čime se ažuriraju sredstva. Također može obrisati prihod pritiskom na ikonu oblika kante za smeće, zatim se od korisnika traži da potvrdi brisanje prihoda (Slika 7.6.10.) te se nakon potvrde prihod obriše i ažuriraju korisnikova sredstva (Slika 7.6.11.).

Uređivanje prihoda

Prihod

1.500

16.01.2019.

Prihod1

Natrag na pregled prihoda

Slika 7.6.9. Prikaz uređivanja
prihoda

Na web-lokaciji pracenje-troskova.herokuapp.com navodi se sljedeće

Jeste li sigurni da želite obrisati označeni prihod?

[U redu](#)

[Odustani](#)

Slika 7.6.10. Prikaz dodatnog potvrđivanja brisanja prihoda

Naziv	Datum	Iznos	Opis	Kategorija	Uredi	Obrisi
Prihod14	14.01.2019	5000.00	Prihod1	NIJE UPISANA		

Slika 7.6.11. Prikaz uređivanja i brisanja prihoda te ažuriranja sredstva

Korisnik može dodati nove prihode pritiskom na gumb „Novi prihodi“ koji se nalazi u padajućem prozoru (Slika 7.6.7.) ili lijevom gornjem kutu stranice „Prihodi“. Zatim se otvara nova stranica i korisnik može unijeti tražene podatke za stvaranje prihoda (Slika 7.6.12.). te se pritiskom na gumb „Stvor“ stvara novi prihod (Slika 7.6.13.).

Prihodi

Unesite naziv prihoda

Unesite iznos

17.01.2019.

Unesite opis

Stvor

Natrag na pregled prihoda

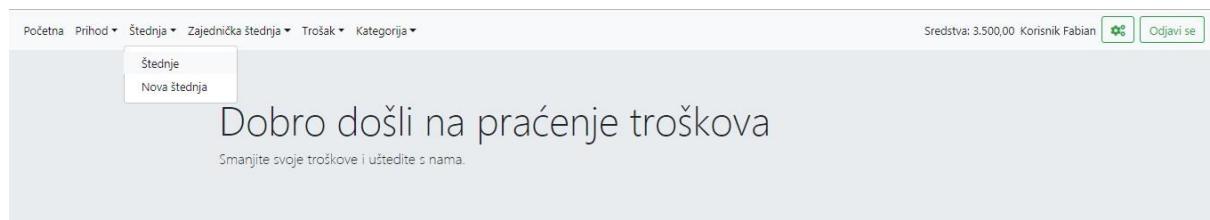
Slika 7.6.12. Prikaz stranica za stvaranje prihoda

The screenshot shows a table with two rows of data. The columns are labeled: Naziv (Name), Datum (Date), Iznos (Amount), Opis (Description), Kategorija (Category), Uredi (Edit), and Obrisi (Delete). The first row contains 'Prihod14' as the name, '14.01.2019' as the date, '5000.0' as the amount, 'Prihod1' as the description, 'NIJE UPISANA' as the category, and edit and delete icons. The second row contains 'NoviPrihod' as the name, '17.01.2019' as the date, '500.0' as the amount, 'Prihod od 500kn' as the description, 'NIJE UPISANA' as the category, and edit and delete icons. Above the table, a green banner says 'Uspješno ste stvorili novi prihod!' (Successfully created a new income!).

Naziv	Datum	Iznos	Opis	Kategorija	Uredi	Obrisi
Prihod14	14.01.2019	5000.0	Prihod1	NIJE UPISANA		
NoviPrihod	17.01.2019	500.0	Prihod od 500kn	NIJE UPISANA		

Slika 7.6.13. Prikaz stvorenog prihoda

Nadalje korisnik pritiskom gumb „Štednja“ otvara padajući prozor (Slika 7.6.14.). Zatim može pritiskom na gumb „Štednje“ otvoriti novu stranicu na kojoj su mu prikazane trenutne štednje (Slika 7.6.15.) ili pritiskom na gumb „Nova štednja“ otvoriti stranicu za stvaranje nove štednje (Slika 7.6.16.).



Štednje

Štednja: Štednja za stan

100.00 %

Slika 7.6.14. Prikaz padajućeg prozora pritiskom na gumb „Štednja“

The screenshot shows a table with one row of data. The columns are labeled: Datum početka (Start Date), Datum kraja (End Date), Cilj (Goal), Opis (Description), Sredstva (Funds), Uplata (Deposit), Isplata (Withdrawal), Uredi (Edit), and Obrisi (Delete). The row contains '16.01.2019' as the start date, '16.01.2019' as the end date, '2000.0' as the goal, 'Štednja za stan' as the description, '2000.0' as funds, and icons for deposit, withdrawal, edit, and delete.

Datum početka	Datum kraja	Cilj	Opis	Sredstva	Uplata	Isplata	Uredi	Obrisi
16.01.2019	16.01.2019	2000.0	Štednja za stan	2000.0				

Slika 7.6.15. Prikaz aktualnih štednji

Štednje

17.01.2019.

17.01.2019.

Unesite sredstva Polje ne može ostati prazno

k Iznos mora biti broj

Unesite opis Polje ne može ostati prazno

Stvori

[Natrag na pregled štednji](#)

Slika 7.6.16. Prikaz stranice za stvaranje nove štednje

Ispravnim unosom traženih podataka za stvaranje štednje te pritiskom na gumb „Stvori“ korisnik stvara novu štednju te mu se otvara ažurirana stranica s prikazom trenutnih štednji (Slika 7.6.17.)

Datum početka	Datum kraja	Cilj	Opis	Sredstva	Uplata	Isplata	Uredi	Obriši
16.01.2019	16.01.2019	2000.0	Štednja za stan	2000.0				
17.01.2019	17.01.2019	100000.0	Štednja za jahtu	1000.0				

Slika 7.6.17. Stvorena nova štednja

Korisnik ima mogućnost uplate novčane svote iz vlastite blagajne u štednju, isplate iz štednje, uređivanja štednje (naziv, cilj...) i brisanja štednje. Pritiskom na ikonu nalik na novčanicu otvara se nova stranica za uplatu novca iz blagajne u štednju te korisnik unosi željeni iznos (ne veći od iznosa u blagajni) i pritiskom na gumb „Uplati“ potvrđuje transakciju (Slika 7.6.18.) te ga se vraća na stranicu sa prikazom štednji (Slika 7.6.)

Transakcija

Unesite iznos

[Natrag na pregled štednji](#)

Slika 7.6.18. Prikaz stranice za uplatu u štednju

Pritisak na ikonu nalik na šaku i dolar otvara se nova stranica za isplatu novca iz štednje u blagajnu te korisnik unosi željeni iznos(ne veći od iznosa u štednji) i pritiskom na gumb „Isplati“ potvrđuje transakciju (Slika 7.6.19.). te ga se vraća na stranicu sa prikazom štednji (Slika 7.6.17.).

Transakcija

Unesite iznos

[Natrag na pregled štednji](#)

Slika 7.6.19. Prikaz stranice za isplatu iz štednje

Pritiskom na gumb nalik na olovku otvara se nova stranica za uređivanje štednje (Slika 7.6.20.). Korisnik zatim unosi željene promjene i pritiskom na gumb „Spremi“ potvrđuje izmjene te ga se vraća na stranicu sa prikazom štednji (Slika 7.6.17.).

Uređivanje štednje

16.01.2019.

16.01.2019.

2.000

2.000

Štednja za stan

[Natrag na pregled štednji](#)

Slika 7.6.20. Prikaz stranice za uređivanje štednje

Brisanje štednje moguće je pritiskom na gumb nalik na kantu za smeće, zatim se od korisnika traži da potvrdi brisanje štednje (Slika 7.6.21.) te se nakon potvrde štednja obriše i vraća na stranicu sa prikazom štednji (Slika 7.6.17.).

Na web-lokaciji pracenje-troskova.herokuapp.com navodi se sljedeće

Jeste li sigurni da želite obrisati označenu štednju?

[U redu](#)

[Odustani](#)

Slika 7.6.21. Prikaz stranice za potvrdu brisanja štednje

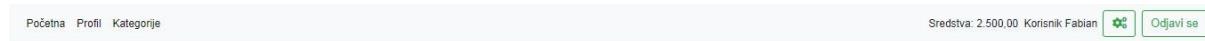
Korisnik pritiskom na gumb „Kategorija“ otvara padajući prozor. Zatim ima dvije mogućnosti. Može pritisnuti na gumb „Kategorije“ prilikom čega mu se otvara nova stranica sa prikazom svih korisnikovih kategorija (Slika 7.6.22.) ili pritiskom na gumb „Nova kategorija“ otvara novu stranicu za stvaranje kategorije (Slika 7.6.23.) te zatim unosi tražene podatke i pritiskom na gumb „Stvori“ stvara novu kategoriju te ga se vraća na stranicu sa prikazom kategorija (Slika 7.7.24.). Kategorija se briše pritiskom na ikonu nalik na kantu za smeće te se od korisnika traži dodatna potvrda brisanja (Slika 7.6.25.). Pritiskom na gumb „U redu“ briše se kategorija i vraća na stranicu sa prikazom kategorija (Slika 7.6.22.)



The screenshot shows a table with columns: Naziv (Name), Opis (Description), Tip (Type), Uredi (Edit), and Obrisi (Delete). There are two rows: one for 'Fotografija' (Description: Prihod od fotografiranja, Type: Prihod) and one for 'Sport' (Description: Trošak sporta, Type: Trošak).

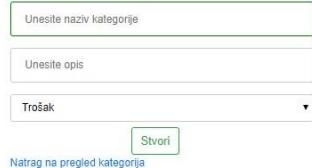
Naziv	Opis	Tip	Uredi	Obrisi
Fotografija	Prihod od fotografiranja	Prihod		
Sport	Trošak sporta	Trošak		

Slika 7.6.22. Prikaz svih kategorija



The screenshot shows a form with fields for 'Naziv' (Name), 'Opis' (Description), and a dropdown 'Tip' (Type) set to 'Trošak'. Below the form is a green 'Stvori' (Create) button.

Kategorije



The screenshot shows a form with fields for 'Unesite naziv kategorije' (Enter category name), 'Unesite opis' (Enter description), and a dropdown 'Tip' (Type) set to 'Trošak'. Below the form is a green 'Stvori' (Create) button.

Slika 7.6.23. Prikaz stranice za stvaranje nove kategorije

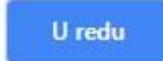
The screenshot shows a user interface for managing expenses. At the top, there are links for 'Početna', 'Profil', and 'Nova kategorija'. On the right, there's a search bar, a balance indicator 'Sredstva: 2,500,00', a user name 'Korisnik Fabian', and a 'Odjavi se' button. A green banner at the top center says 'Uspješno ste stvorili novu kategoriju!' (Successfully created a new category!). Below it is a table with columns: Naziv (Name), Opis (Description), Tip (Type), Uredi (Edit), and Obrisí (Delete). The table contains three rows: 'Fakultet' (Faculty) with 'Troškovi obrazovanja' (Education expenses) as description, 'Tip' as 'Trošak' (Expense), and edit/delete icons; 'Fotografija' (Photography) with 'Prihod od fotografiranja' (Income from photography) as description, 'Tip' as 'Prihod' (Income), and edit/delete icons; and 'Sport' (Sports) with 'Trošak sporta' (Sports expense) as description, 'Tip' as 'Trošak' (Expense), and edit/delete icons.

Naziv	Opis	Tip	Uredi	Obrisí
Fakultet	Troškovi obrazovanja	Trošak		
Fotografija	Prihod od fotografiranja	Prihod		
Sport	Trošak sporta	Trošak		

Slika 7.6.24. Stvaranje nove kategorije i njezin prikaz

Na web-lokaciji pracenje-troskova.herokuapp.com navodi se sljedeće

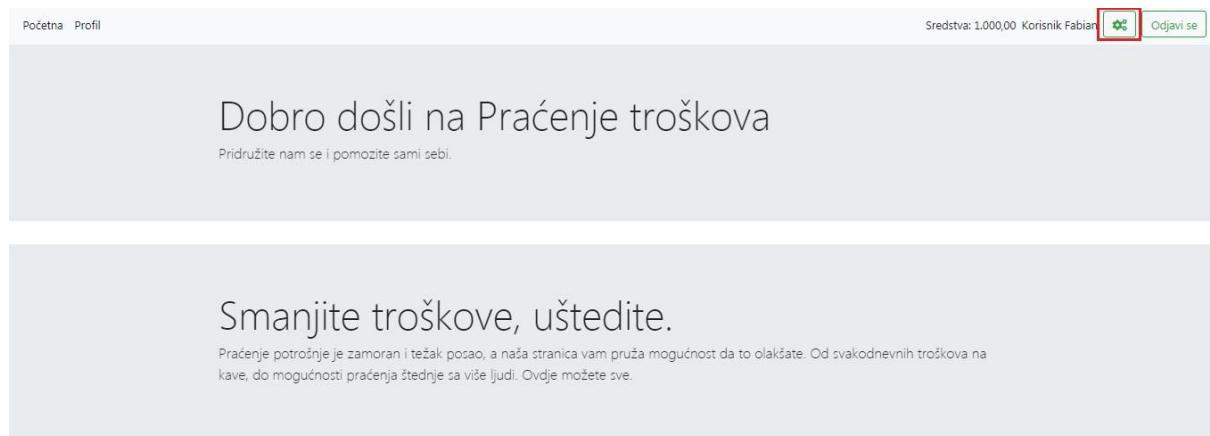
Jeste li sigurni da želite obrisati označenu kategoriju?

 U redu

 Odustani

Slika 7.6.25. Prikaz stranice za potvrdu brisanja kategorije

U gornjem desnom kutu može pritisnuti na ikonu koja izgleda kao tri zupčanika (Slika 7.6.26.) čime mu se otvara stranica sa podacima svojeg profila (Slika 7.6.27.) i nudi mu se mogućnost izmjene tih podataka te izmjene lozinke pritiskom na gumb „Promijeni lozinku“ čime se otvara stranica za izmjenu lozinke (Slika 7.6.28.)



Slika 7.6.26. Gumb „Podaci o profilu“ (označen crvenim)

Podaci o profilu

Korisničko ime: Fabian

[Promijeni lozinku](#)

Fabian
Leskovec
leskovec.fabian@gmail.com

[Natrag na profil](#)

Slika 7.6.27. Prikaz podataka o profilu i gumba „Promijeni lozinku“

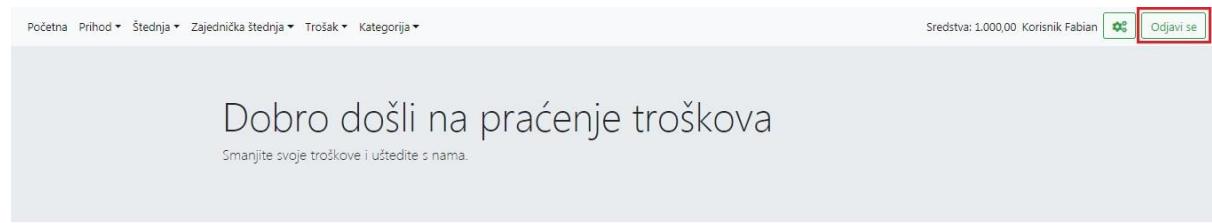
Promjena lozinke

Unesi staru lozinku
Unesi novu lozinku
Potvrdi novu lozinku ponovnim unosom

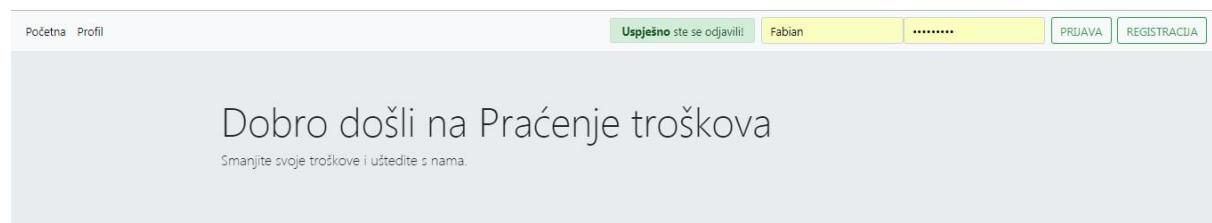
[Natrag na postavke](#)

Slika 7.6.28. Prikaz stranice za promjenu lozinke

Korisnik se može odjaviti pritiskom na gumb „Odjavi se“ (Slika 7.6.29.) čime ga se vraća na početnu stranicu web aplikacije i obavještava o uspješnoj odjavi (Slika 7.6.30.).



Slika 7.6.29. Gumb „Odjavi se“
(označen crvenim)



Slika 7.6.30. Početna stranica web aplikacije nakon uspješne odjave

8. Zaključak i budući rad

Projekt je realiziran i dokumentacija dovršena. Sva logika na kojoj se temelji koncept programa je ispravno ugrađena i testirana. Program se spreman za komercijalnu uporabu!

Korisnik ove aplikacije može efikasno barati svojim financijama, svakodnevnim troškovima i prihodima. Aplikacija je intuitivna i jednostavna za uporabu te ugodna oku. Na raspolaganju su razne dodatne mogućnosti izuzev osnovnog unošenja podataka o svojim primanjima i izdavanjima poput zajedničkog praćenja financija s drugim korisnicima i pregleda statistike o svojoj blagajni.

Program se može dalje proširiti s dodatnim funkcionalnosti. Primjerice, povezivanjem sa stvarnim bankovnim računom ili implementacijom dodatnih korisnih funkcionalnosti poput praćenja vrijednosti međunarodnih valuta.

9. Popis literature

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

- ¹ Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
- ² Oblikovanje programske potpore, FER ZEMRIS, <http://www.zemris.fer.hr/predmeti/opp>
- ³ I. Sommerville, „Software engineering“, 8th ed, Addison Wesley, 2007.
- ⁴ T.C.Lethbridge, R.Langaniere, „Object-Oriented Software Engineering“, 2nd ed. McGraw-Hill, 2005.
- ⁵ UML 2 Class Diagram Guidelines, <http://www.agilemodeling.com/style/classDiagram.htm>
- ⁶ Domain Class Diagram Modeling Standards and Guidelines, <http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf>
- ⁷ Astah Community, <http://astah.net/editions/community/>
- ⁸ Git installation <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
- ⁹ Spring tutorial <https://www.tutorialspoint.com/spring/>
- ¹⁰ Spring tutorial <https://www.youtube.com/watch?v=msXL2oDexqw>
- ¹¹ Maven tutorial <https://www.tutorialspoint.com/maven/>

I. Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda)

Slika 2.1 Praćenje troškova

Slika 4.1.1. Dijagram obrazaca uporabe korisnika

Slika 4.1.2. Sekvencijski dijagram za UC1

Slika 4.1.3. Sekvencijski dijagram za UC2

Slika 4.1.4. Sekvencijski dijagram za UC3

Slika 4.1.5. Sekvencijski dijagram za UC4

Slika 4.1.6. Sekvencijski dijagram za UC5

Slika 4.1.7. Sekvencijski dijagram za UC6

Slika 4.1.8. Sekvencijski dijagram za UC7

Slika 4.1.9. Sekvencijski dijagram za UC8

Slika 4.1.10. Sekvencijski dijagram za UC9

Slika 4.1.11. Sekvencijski dijagram za UC10

Slika 4.1.12. Sekvencijski dijagram za UC11

Slika 4.1.13. Sekvencijski dijagram za UC12

Slika 4.1.14. Sekvencijski dijagram za UC13

Slika 4.1.15. Sekvencijski dijagram za UC14

Slika 4.1.16. Sekvencijski dijagram za UC15

Slika 4.1.17. Sekvencijski dijagram za UC16

Slika 4.1.18. Sekvencijski dijagram za UC17

Slika 4.1.19. Sekvencijski dijagram za UC18

Slika 4.1.20. Sekvencijski dijagram za UC19

Slika 4.1.21. Sekvencijski dijagram za UC20

Slika 4.1.22. Sekvencijski dijagram za UC21

Slika 4.1.23. Sekvencijski dijagram za UC22

Slika 4.1.24. Sekvencijski dijagram za UC23

Slika 4.1.25. Sekvencijski dijagram za UC24

Slika 6.1. Simboličan prikaz elemenata arhitekture

Slika 6.1.1. ER model baze podataka

Slika 6.2.1. Razredi paketa pogledi

Slika 6.2.2. Razredi paketa entiteti

Slika 6.2.3. Dijagram razreda

Slika 6.2.4. Razredi paketa repozitoriji

Slika 6.3.1. Dijagram objekata u sustavu

Slika IV.1. Graf frekvencije rada po danu u mjesecu

Slike 6.4.1 – 6.4.3 Dijagram stanja

Slika 6.4.2. Komunikacijski dijagram registracije

Slika 6.4.3. Komunikacijski dijagram stvaranja prihoda

Slika 6.4.4. Dijagram aktivnosti prilikom prijave i stvaranja prihoda

Slika 6.4.5. Dijagram komponenti

Slika 7.1.1. Dijagram razmještaja

Slika 7.4.1. Ispis greške zbog nepotpunjene polja za registraciju

Slika 7.4.2. Ispis greške zbog upisa već zauzete adrese el. pošte te zbog nezadovoljavajuće lozinke

Slika 7.4.3. Uspješno stvaranje novog korisničkog računa

Slika 7.4.4. Ispis poruke o grešci prilikom upisa neispravnih podataka

Slika 7.4.5. Otvaranje početne stranice prilikom uspješne prijave

Slika 7.4.6. Ispis i opisi pogrešaka prilikom unosa podataka

Slika 7.4.7. Ispis poruke o grešci (provjera sredstva u blagajni)

Slika 7.4.8. Uspješno stvaranje štednje

- Slika 7.4.9. Ispis greške zbog nepotpunjenih polja za registraciju
Slika 7.4.10. Ispis greške zbog krivog formata
Slika 7.4.11. Uspješno stvoren prihod
Slika 7.4.12. Prikaz uspješne odjave
Slika 7.4.13. Prikaz korisničkih podataka
Slika 7.4.14. Prikaz uspješne promjene korisničkih podataka
Slika 7.4.15. Prikaz odabira ikone za uplatu u štednju
Slika 7.4.16. Prikaz ispisa greške zbog krivog formata
Slika 7.4.17. Prikaz ispisa greške zbog nedovoljnih sredstava blagajne
Slika 7.4.18. Prikaz uspješne uplate u štednju
Slika 7.5..1 GitLab sučelje
Slika 7.5.2. Stvaranje baze podataka
Slika 7.5.3. Pokretanje aplikacije
Slika 7.6.1. Prikaz sučelja za prijavu
Slika 7.6.2. Prikaz sučelja za registraciju
Slika 7.6.3. Potvrda o verifikaciji adrese el. pošte
Slika 7.6.4. Prikaz početne stranice(„Prijavljen korisnik“) nakon prijave
Slika 7.6.5. Prikaz početne stranice web aplikacije
Slika 7.6.6. Prikaz stranice nakon pritiska na gumb „Profil“ i dodatne mogućnosti(označeno crvenim)
Slika 7.6.7. Prikaz padajućeg prozora pritiskom na gumb „Prihodi“
Slika 7.6.8. Prikaz izvršenih prihoda, ikona za uređivanje (označena crvenim) i ikona za brisanje (označena zelenim)
Slika 7.6.9. Prikaz uređivanja prihoda
Slika 7.6.10. Prikaz dodatnog potvrđivanja brisanja prihoda
Slika 7.6.11. Prikaz uređivanja i brisanja prihoda te ažuriranja sredstva
Slika 7.6.12. Prikaz stranica za stvaranje prihoda
Slika 7.6.13. Prikaz stvorenog prihoda
Slika 7.6.14. Prikaz padajućeg prozora pritiskom na gumb „Štednja“
Slika 7.6.15. Prikaz aktualnih štednji
Slika 7.6.16. Prikaz stranice za stvaranje nove štednje
Slika 7.6.17. Stvorena nova štednja
Slika 7.6.18. Prikaz stranice za uplatu u štednju
Slika 7.6.19. Prikaz stranice za isplatu iz štednje
Slika 7.6.20. Prikaz stranice za uređivanje štednje
Slika 7.6.21. Prikaz stranice za potvrdu brisanja štednje
Slika 7.6.22. Prikaz svih kategorija
Slika 7.6.23. Prikaz stranice za stvaranje nove kategorije
Slika 7.6.24. Stvaranje nove kategorije i njezin prikaz
Slika 7.6.25. Prikaz stranice za potvrdu brisanja kategorije
Slika 7.6.26. Gumb „Podaci o profilu“ (označen crvenim)
Slika 7.6.27. Prikaz podataka o profilu i gumba „Promijeni lozinku“
Slika 7.6.28. Prikaz stranice za promjenu lozinke
Slika 7.6.29. Gumb „Odjavi se“ (označen crvenim)
Slika 7.6.30. Početna stranica web aplikacije nakon uspješne odjave

II. Dodatak B: Dnevnik sastajanja

\$5.10.2018

Uvodni sastanak. Upoznavanje, predlaganje teme za projekt.

#10.10.2019

Sastanak s asistenticom

\$11.10.2018

Okvirna raspodjela poslova.

#17.10

Sastanak s asistenticom

\$20.10.2018

Detaljnije definiranje sustava (zahtjeva i arhitekture)

#7.11.2018

Sastanak s asistenticom

\$17.12.2018

Dogovaranje oko implementacije i dizajna

\$3.1.2019

Dogovaranje oko implementacije i dizajna

#9.1.2019

Sastanak s asistenticom

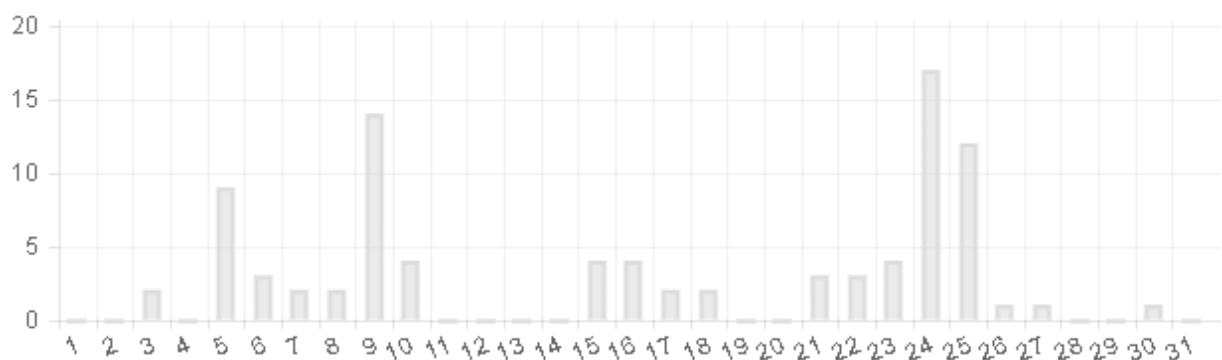
#16.1.2019

Sastanak s asistenticom

III.Dodatak C: Prikaz aktivnosti grupe

Popis aktivnosti	Niko Palić	Fabian Leskovec	Dominik Marjanović	Leo Miloš	Karlo Jagar	Branimir Pehar
Upravljanje projektom	100%	0	0	0	0	0
Opis projektnog zadatka	100%	0	0	0	0	0
Rječnik pojmoveva	0	100%	0	0	0	0
Opis funkcionalnih zahtjeva	20%	20%	0	0	60%	0
Opis ostalih zahtjeva	100%	0	0	0	0	0
Arhitektura i dizajn sustava	25%	25%	25%	25%	0	0
Svrha, opći prioriteti i skica sustava	30%	30%	0	0	20%	20%
Dijagram objekata			10%	10%	80%	
Dijagram razreda s opisom	0	0	70%	30%	0	0
Ostali UML dijagrami	80%	20%	0	0	0	0
Implementacija i korisničko sučelje	100%	0	0	0	0	0
Dijagram razmještaja	10%	90%	0	0	0	0
Korištene tehnologije i alati	0	100%	0	0	0	0
Isječak programskog kôda	100%	0	0	0	0	0
Ispitivanje programskog rješenja	10%	90%	0	0	0	0
Upute za	100%	0	0	0	0	0

instalaciju						
Korisničke upute	10%	90%	0	0	0	0
Plan rada	100%	0	0	0	0	0
Pregled rada i stanje ostvarenja	100%	0	0	0	0	0
Zaključak i budući rad	100%	0	0	0	0	0
Popis literature	0	0	0	0	50%	50%
Dodaci	20%	0	0	0	30%	50%
Indeks	100%	0	0	0	0	0
Dnevnik sastajanja	100%	0	0	0	0	0



Slika IV.1 Graf frekvencije rada po danu u mjesecu

IV. Dodatak D: Plan rada / Pregled rada i stanje ostvarenja

Program je realiziran, a dokumentacija dovršena.

U prвome ciklusu napravljena je većina dokumentacije i manji dio programa, a u drugome ciklusu je sve napravljeno do kraja.