

Oblikovanje programske potpore

Ak. god. 2018./2019.

Praćenje troškova

Dokumentacija, Rev. 1

Grupa: *grupaprekoveze*

Voditelj: *Niko Palić*

Datum predaje: *28. studenog 2018.*

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 (Java Spring) i arhitektura, izrada UML dijagrama i dokumentacije

Leo Miloš – back-end (Java Spring) i arhitektura, izrada UML dijagrama i dokumentacije

Karlo Jagar – front-end (HTML, CSS), izrada UML dijagrama i dokumentacije

Branimir Pehar – front-end (HTML, CSS), 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.....	11
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.....	53
6.4. Ostali UML dijagrami	54
7. Implementacija i korisničko sučelje	55
7.1. Dijagram razmještaja.....	56
7.2. Korištene tehnologije i alati.....	57
7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava	58
7.4. Ispitivanje programskog rješenja	59
7.5. Upute za instalaciju	60
7.6. Korisničke upute	61
8. Zaključak i budući rad	62
9. Popis literature	63
Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda).....	64
Dodatak B: Dnevnik sastajanja.....	65
Dodatak C: Prikaz aktivnosti grupe.....	66
Dodatak D: Plan rada / Pregled rada i stanje ostvarenja	68

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	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 sekvencijski dijagrami i opisi	Fabian, Niko	15.11.2018
0.6	Dodani opis projekta	Niko	21.11.2018
0.65	Popravljen 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

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 blagajne 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 blagajna 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 blagajnu, dok pozvana osoba može odbiti ili prihvatiti 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 obrađivati 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, korisničkog imena
 - Mogućnost brisanja profila
 - Raspolaze jednim računom (blagajnom) i može stvoriti proizvoljan broj štednji
 - Može pregledavati podatke o stanju vlastitog računa, štednji te prebacivati novac na iste
 - Može odabrati 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 elektroničke 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 elektroničke 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 adresu elektroničke pošte 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. Pritisak na gumb za odjavu
 - II. Uspješno provedena odjava
- **UC4 – Brisanje korisničkog računa**
 - Glavni sudionik: Korisnik
 - Cilj: Brisanje korisničkog računa
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjeti: Korisnik je prijavljen u sustav
 - Željeni scenarij:
 - I. Pritisak na gumb za brisanje korisničkog računa
 - II. Potvrda odluke
 - III. Podaci o korisničkom računu se brišu iz baze podataka
- **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
 - Željeni scenarij:
 - I. Promjena vrijednosti osobnih podataka
 - II. Izmijenjeni podaci upisani u bazu podataka
 - III. Potvrda 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. Dodavanje individualne kategorije u bazu podataka uz već postojeće kategorije na web-stranici
 - III. Potvrda 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. Pritisak na gumb za stvaranje štednje
 - II. Unos odabranog cilja 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. Dohvaćanje i prikaz korisnikovih kategorija
- **UC9** – Unos prihoda
 - Glavni sudionik: Korisnik
 - Cilj: Unos prihoda u blagajnu
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „blagajna“
 - Željeni scenarij:
 - I. Korisnik unosi željeni iznos prihoda
 - II. Uspješan zapis prihoda i njezina pohrana u bazu podataka
 - Mogući scenarij:
 - I. Spremanje neuspješno zbog krivog formata podataka
 - II. Korisnik je obaviješten o neuspjehu

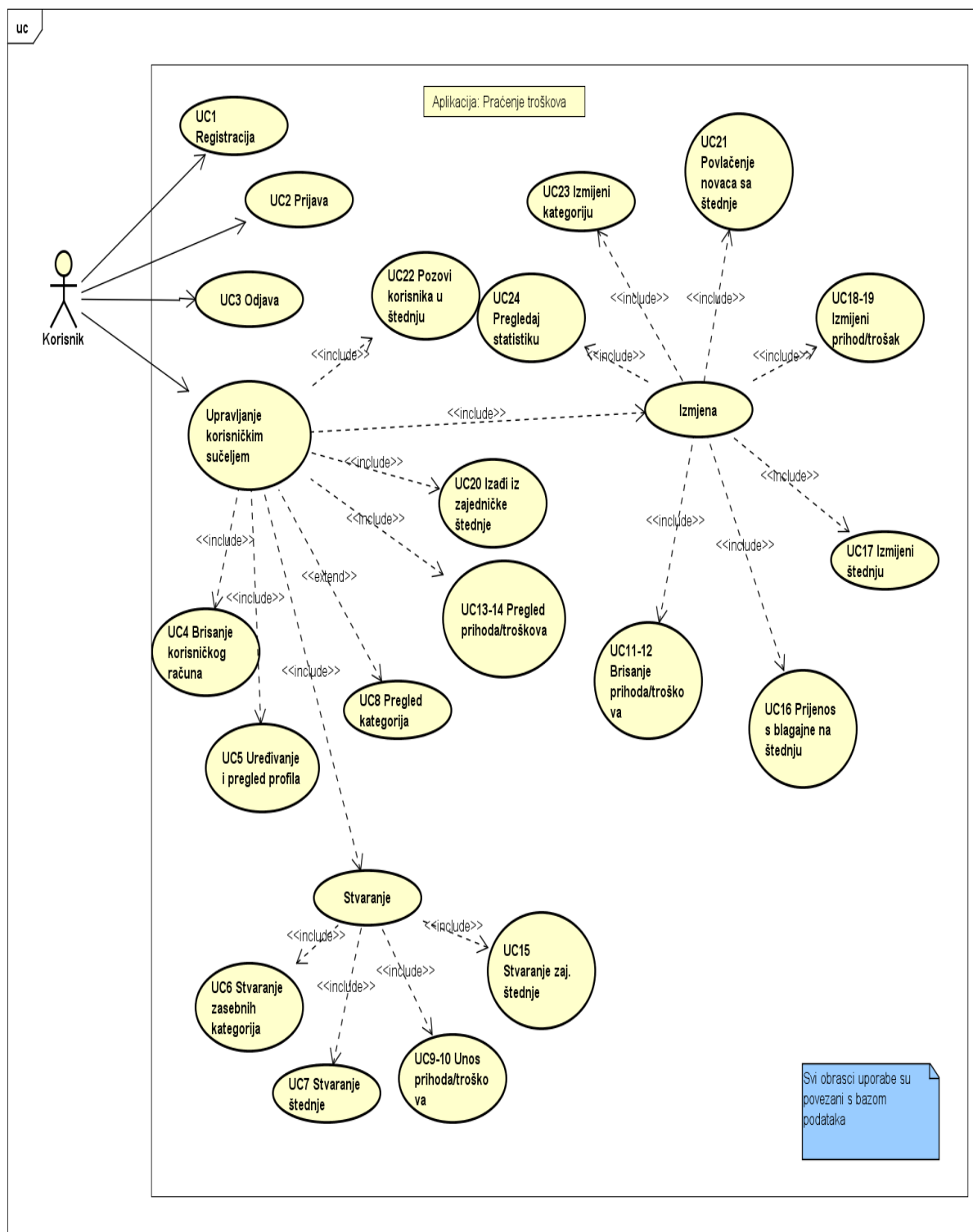
- **UC10-** Unos troškova
 - Glavni sudionik: Korisnik
 - Cilj: Unos troškova u blagajnu
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „blagajna“
 - Željeni scenarij:
 - I. Korisnik unosi željeni iznos troškova
 - II. Uspješan zapis troškova 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. Uspješno brisanje prihoda korisnikove blagajne iz baze podataka
 - III. Stanje računa se umanjuje za iznos obrisano 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. Uspješno brisanje troškova korisnikove blagajne iz baze podataka
 - III. Stanje računa se uvećava za iznos obrisano troška

- **UC13** – Pregled troškova
 - Glavni sudionik: Korisnik
 - Cilj: Pregled korisničkih troškova
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav, kliknuo „blagajna“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za pregled troškova
 - II. Uspješno dohvaćanje iz baze i prikaz troškova
- **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, kliknuo „blagajna“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za pregled prihoda
 - II. Uspješno dohvaćanje iz baze i prikaz prihoda
- **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 „štednje“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za kreiranje zajedničke štednje te odabire ostale članove štednje
 - II. Uspješno stvaranje zajedničke štednje te njezin zapis 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 „blagajna“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za prijenos s blagajne u štednju
 - II. Odabir iznosa za prijenos
 - III. Uspješno prebacivanje i zapis iznosa u bazi podataka
 - Mogući scenarij:
 - I. Korisnik raspolaže nedostatnim financijama
 - II. Korisnik je obaviješten o neuspjehu

- **UC17** – Izmijeni štednju
 - Glavni sudionik: Korisnik
 - Cilj: Promjena štednje
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav i ima stvorenu štednju, kliknuo „štednje“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za izmjenu štednje te unosi željeni iznos za cilj
 - II. Uspješna izmjena štednje te njezin zapis 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. Uspješna izmjena troška te njegova pohrana u bazu podataka
 - Mogući scenarij:
 - I. Krivi format
 - II. Korisnik je obaviješten o neuspjehu
- **UC19** – Izmijeni 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. Uspješna izmjena prihoda te njegova pohrana u bazu podataka
 - Mogući scenarij:
 - I. Krivi format
 - II. Korisnik je obaviješten o neuspjehu

- **UC20** – Izadi 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 „štednje“
 - Željeni scenarij
 - I. Korisnik pritišće na gumb za izlazak iz zajedničke štednje
 - II. Bira koliko novaca želi povući pri izlasku
 - III. Uspješan izlazak korisnika iz zajedničke štednje te brisanje iz baze podataka
 - IV. Transfer odabrane svote novaca u 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. Uspješno povlačenje novaca te zapis 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 „štednje“
 - Željeni scenarij:
 - I. Korisnik pritišće na gumb za poziv korisnika u zajedničku štednju
 - II. Uspješno slanje zahtjeva željenom korisniku te njegov zapis u bazu podataka nakon prihvata poziva
 - Mogući scenarij:
 - I. Pozvani korisnik ne postoji
 - II. Korisnik je obaviješten o neuspjehu

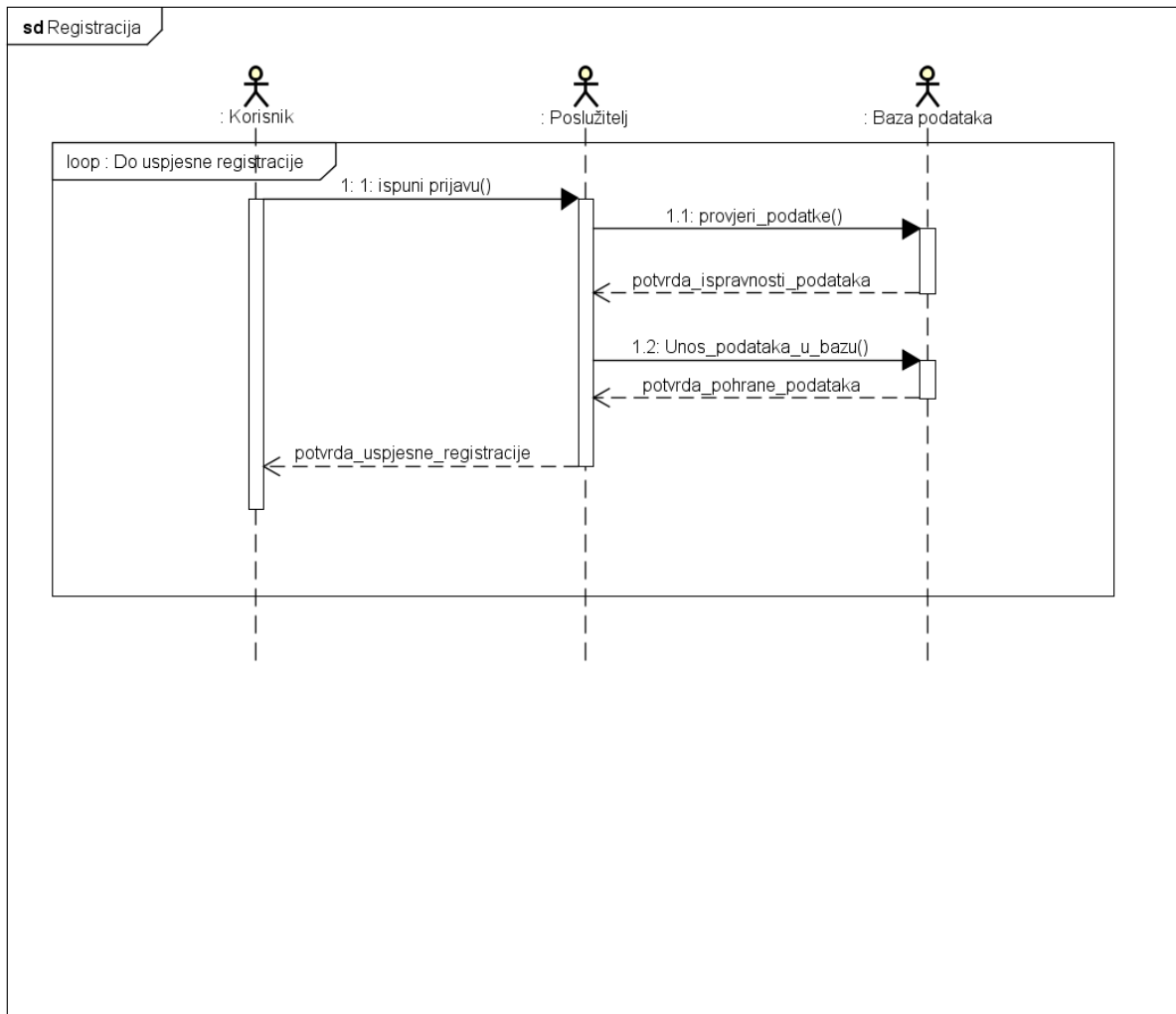
- **UC23** – Izmijeni kategoriju
 - Glavni sudionik: Korisnik
 - Cilj: Izmjena kategorija
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav te već ima stvorene zasebne kategorije, kliknuo „kategorije“
 - Željeni scenarij:
 - I. Korisnik pritisće na gumb za izmjenu kategorija te unosi željene promjene
 - II. Uspješna izmjena kategorija i pohrana izmjena u bazu podataka
 - Mogući scenarij:
 - I. Krivo uneseni podaci
 - II. Korisnik je obaviješten o neuspjehu
- **UC24** – Pregledaj statistiku
 - Glavni sudionik: Korisnik
 - Cilj: Pregled statistike o financijama
 - Ostali sudionici: Web aplikacija, baza podataka
 - Preduvjet: Korisnik je prijavljen u sustav te je odabrao prikaz prihoda/štednje
 - Željeni scenarij:
 - I. Korisnik odabire blagajnu/štednju za koju želi pregledati statistiku
 - II. Pritisće na gumb za pregled statistike
 - III. Uspješno dohvaćanje iz baze i prikaz statistika o stanju blagajne/štednje



Slika 4.1.1. Dijagram obrazaca uporabe korisnika

SEKVENCIJSKI DIJAGRAMI:**Obrazac uporabe UC1 (Registracija):**

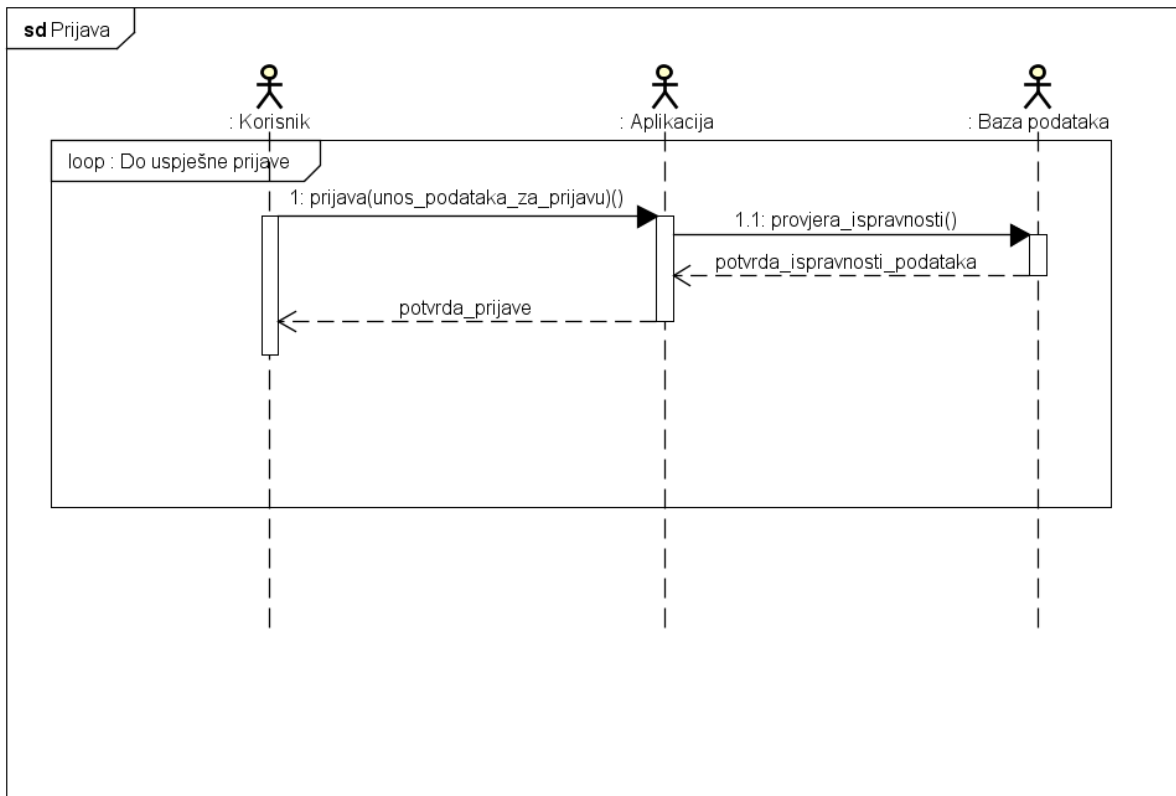
Korisnik se registrira ispunom web obrasca prijave. Njegovi podaci su poslani poslužitelju koji zajedno s bazom podataka provjerava njihovu ispravnost te da li već postoji korisnik sa istim podacima. Ukoliko je potvrđena ispravnost podataka, poslužitelj šalje bazi zahtjev za pohranu istih te će korisnik 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

Obrazac uporabe UC2 (Prijava):

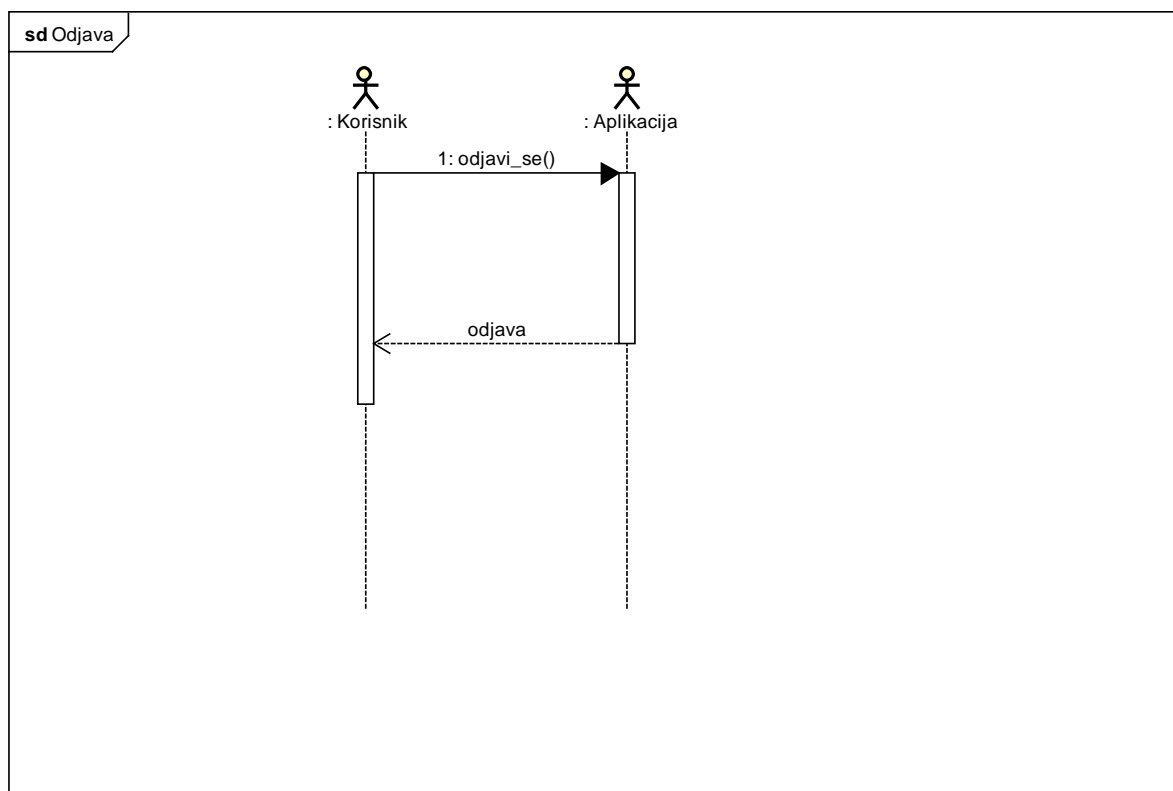
Registrirani korisnik unosi podatke za prijavu. Web aplikacija podatke šalje u bazu na provjeru. Ako su podaci ispravni korisnik se prijavljuje u sustav. U slučaju neispravnih podataka korisnik može ponovo pokušati.



Slika 4.1.3. Sekvencijski dijagram za UC2

Obrazac uporabe UC3 (Odjava):

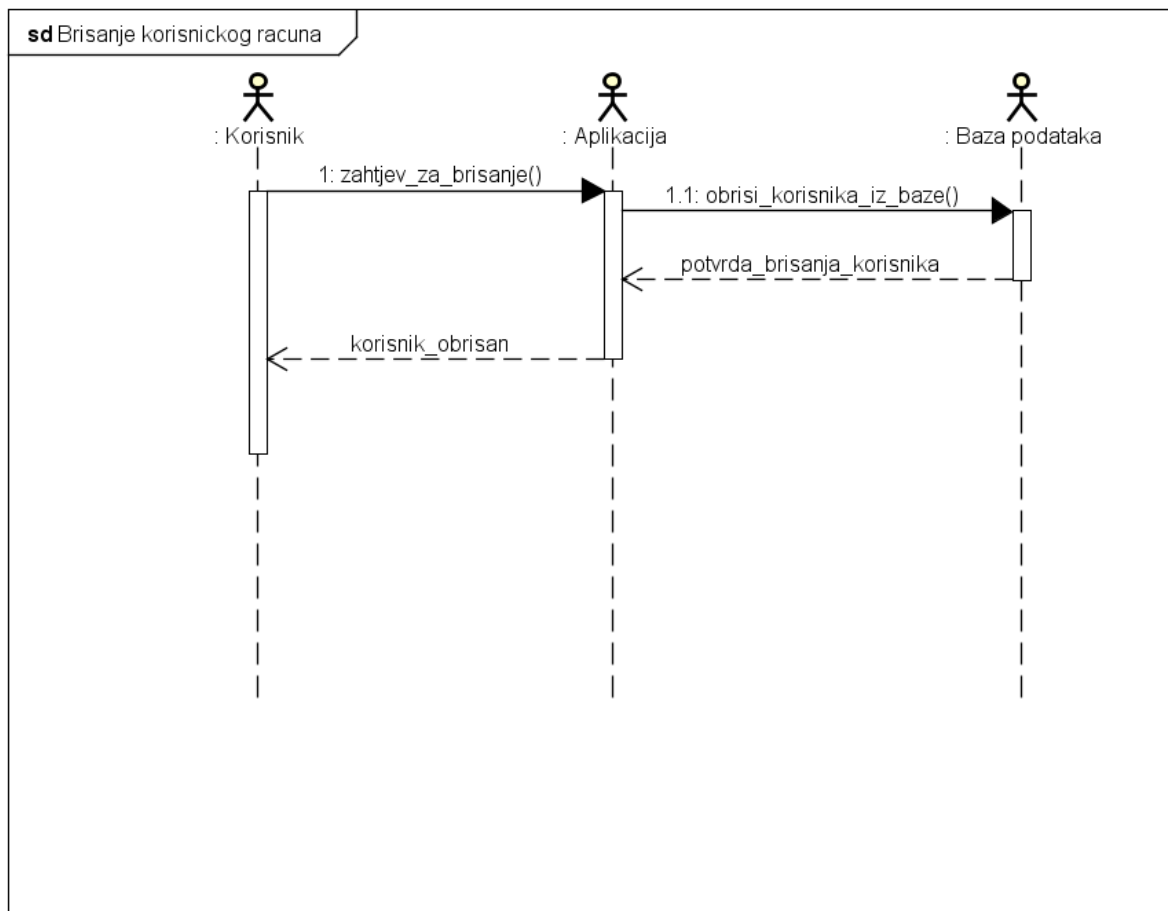
Registrirani korisnik pritiskom na gumb obavještava aplikaciju da se želi odjaviti iz sustava.



Slika 4.1.4. Sekvencijski dijagram za UC3

Obrazac uporabe UC4 (Brisanje korisničkog računa):

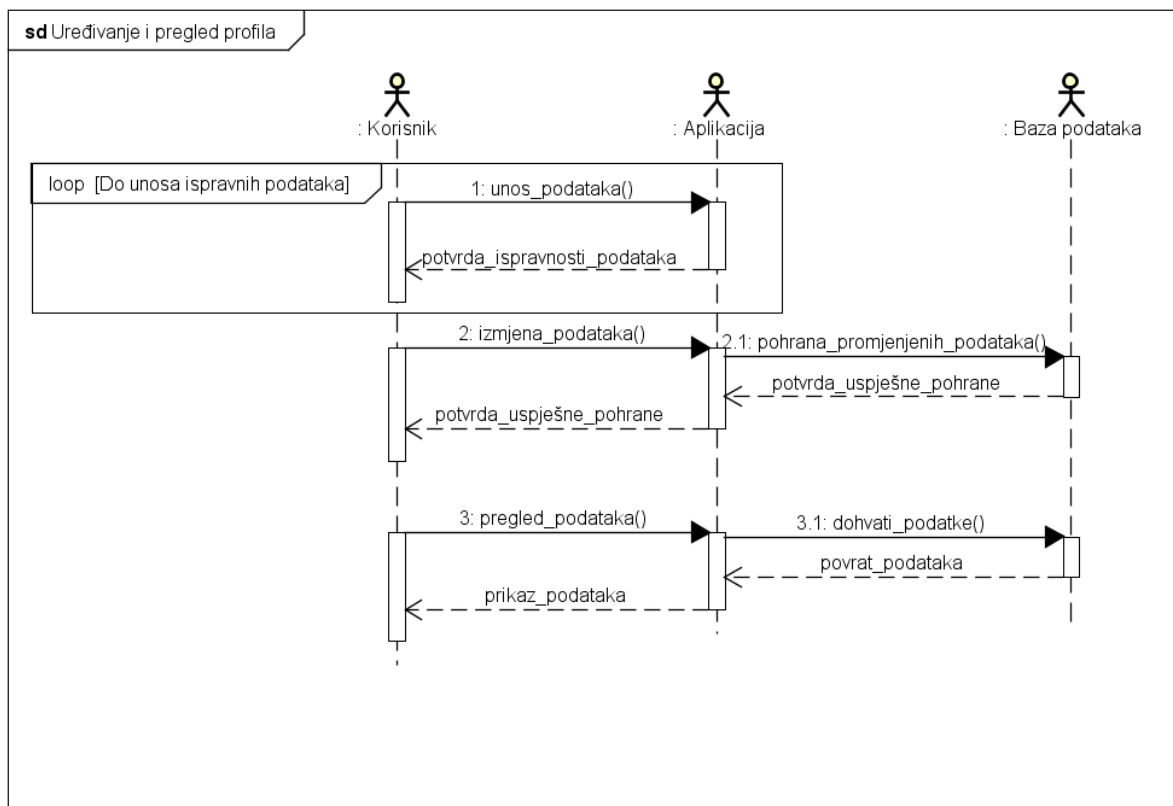
Registrirani korisnik odabire opciju brisanja profila. Zahtjev za brisanje dolazi poslužitelju koji prosljeđuje zahtjev bazi. Ako ima šalje bazi zahtjev za izmjenu podataka i vraća korisniku povratnu informaciju.



Slika 4.1.5. Sekvencijski dijagram za UC4

Obrazac uporabe UC5 (Uređivanje i pregled profila):

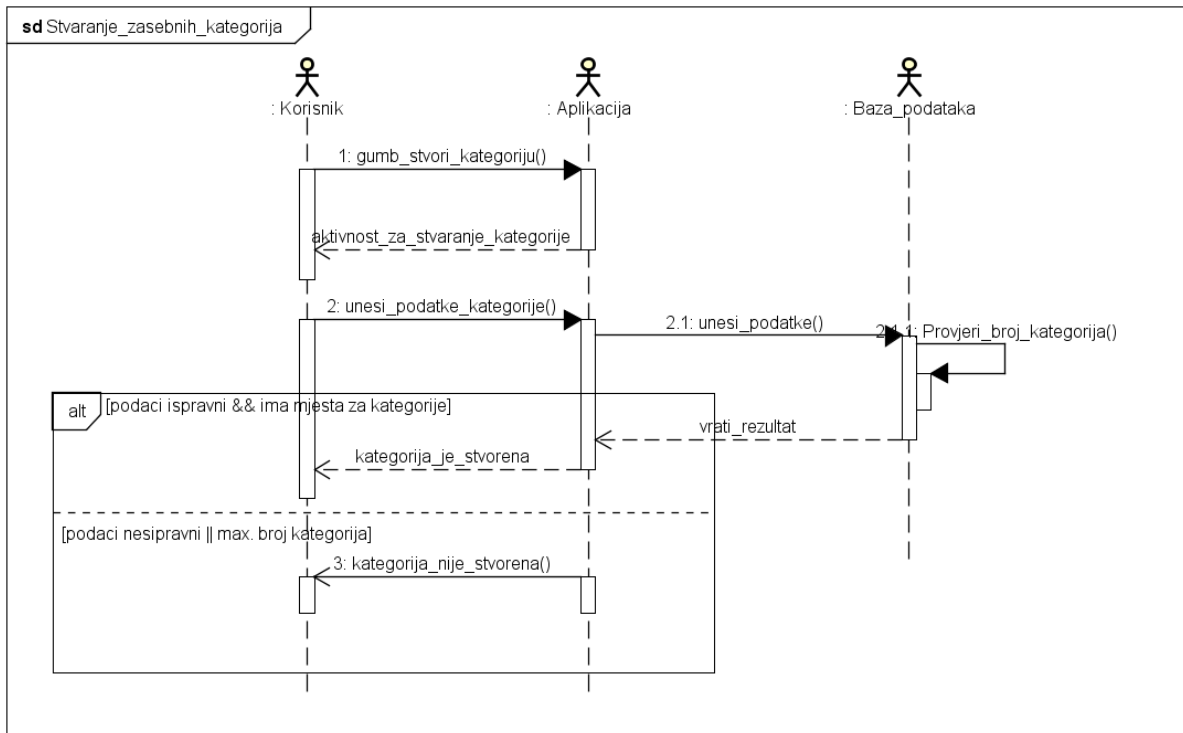
Korisnik unosi podatke sve dok nisu valjani. Poslužitelj provjerava valjanost podataka te korisniku šalje poruku o ispravnosti istih. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka te ih ona pohrani. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj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

Obrazac uporabe UC6 (Stvaranje zasebnih kategorija):

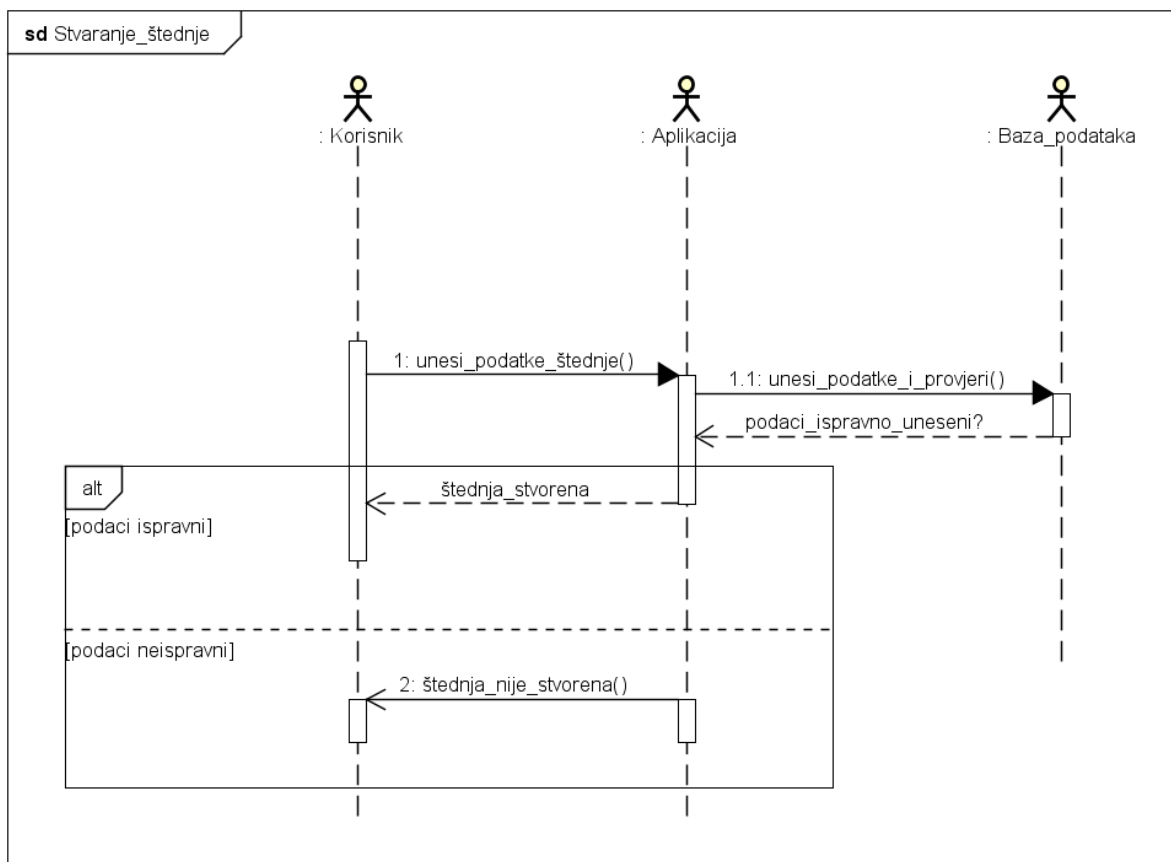
Korisnik pritisće na gumb za odabir zasebnih kategorija blagajne. Zatim se provjerava da li je prekoračen maksimalan broj zasebnih kategorija. Nakon unosa podataka poslužitelj ih šalje u bazu podataka te ih ona pohrani. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj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



Slika 4.1.7. Sekvencijski dijagram za UC6

Obrazac uporabe UC7 (Stvaranje štednje) :

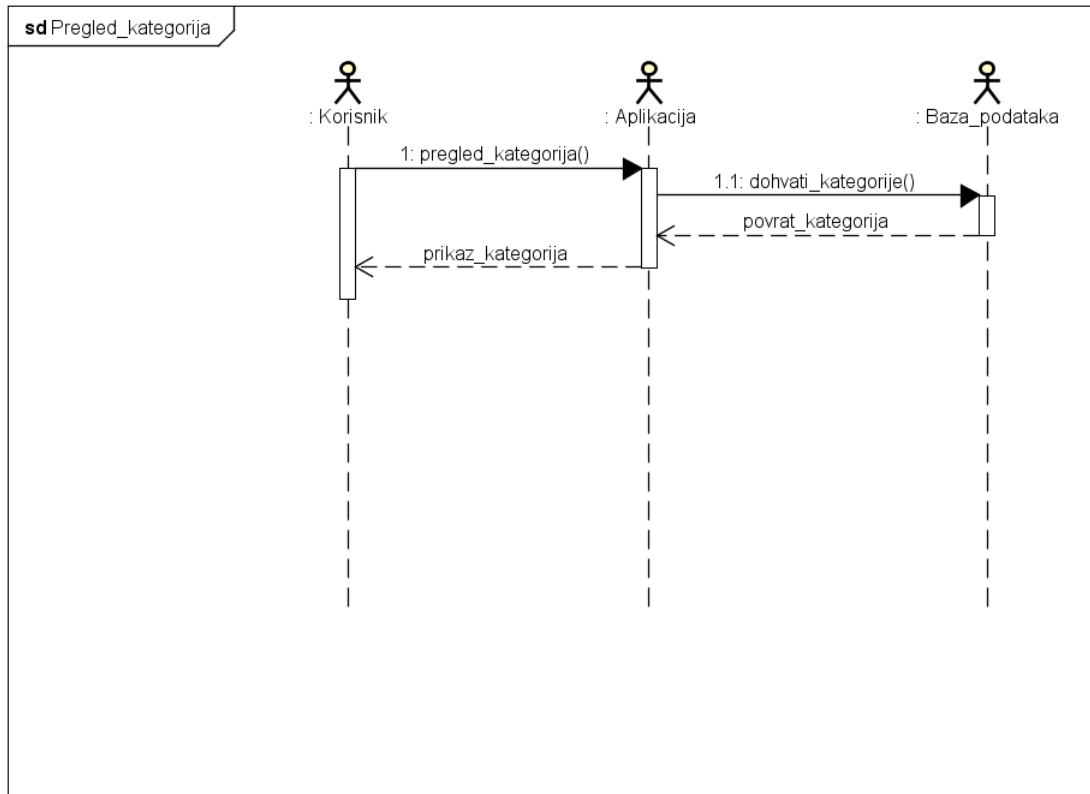
Korisnik odabire opciju stvaranja štednje. Zatim unosi tražene podatke. Poslužitelj provjerava valjanost podataka te korisniku šalje poruku o ispravnosti istih. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka te ih ona pohrani. Nakon uspješne pohrane podataka, baza o tome obavještava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



Slika 4.1.8.Sekvencijski dijagram za UC7

Obrazac uporabe UC8 (Pregled kategorija):

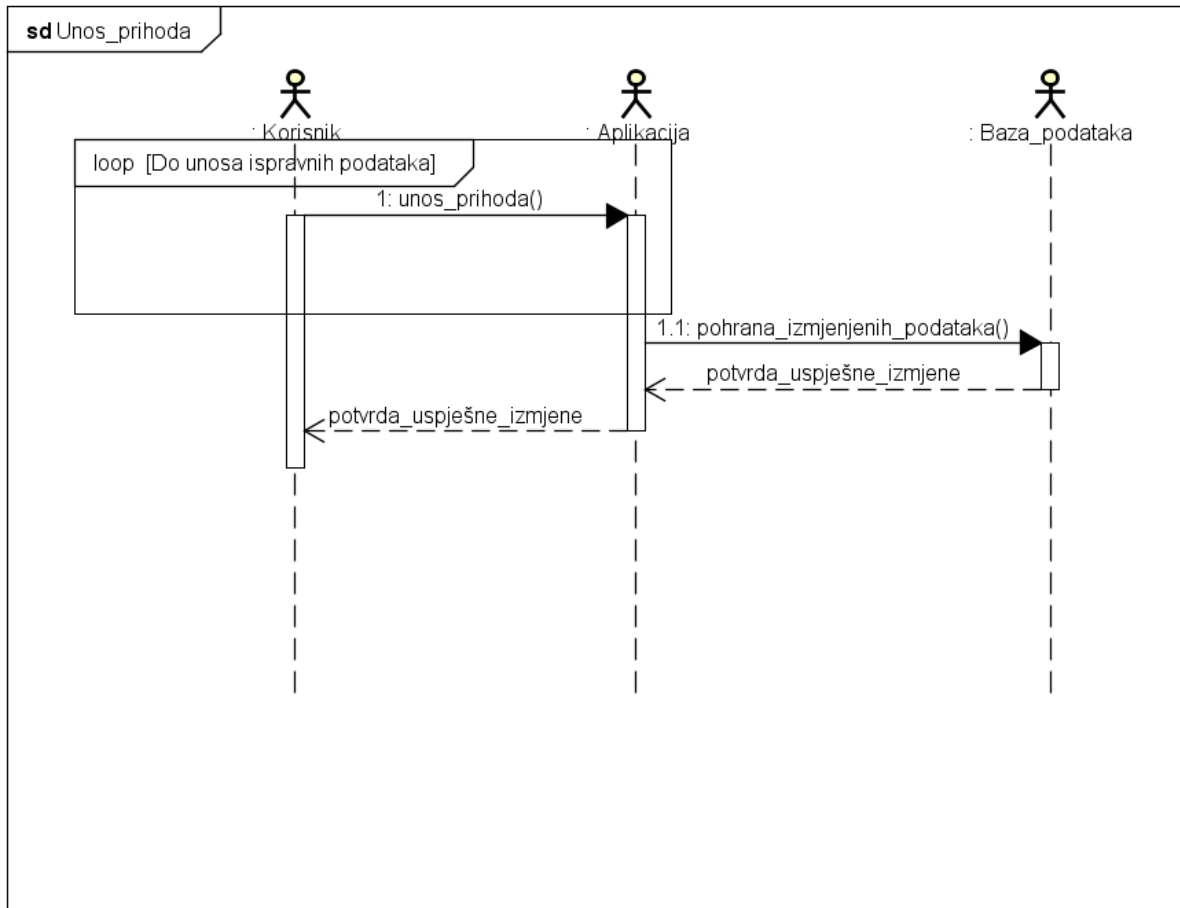
Korisnik odabire opciju pregleda kategorija. Prilikom pregledavanja kategorija korisnik šalje zahtjev poslužitelju te on isti prosljeđuje bazi podataka i od nje zatraži podatke kategorija. Baza ih zatim proslijedi poslužitelju, a poslužitelj ih prikaže korisniku.



Slika 4.1.9. Sekvencijski dijagram za UC8

Obrazac uporabe UC9 (Unos prihoda):

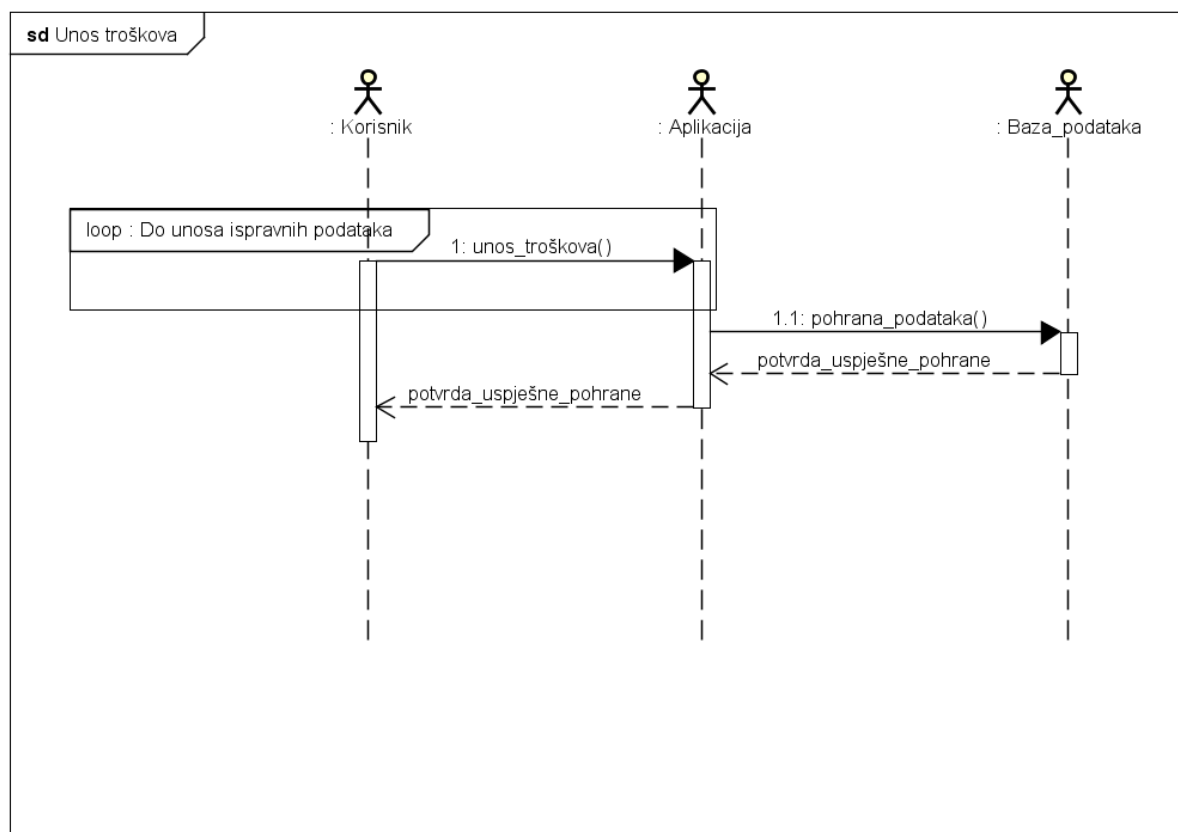
Korisnik odabire opciju unosa prihoda te unosi iznos prihoda. Web aplikacija podatke šalje u bazu na provjeru. Ako su podaci ispravni prihod se unosi u bazu podataka. U slučaju neispravnih podataka korisniku se javlja greška.



Slika 4.1.10.Sekvencijski dijagram za UC9

Obrazac uporabe UC10 (Unos troškova):

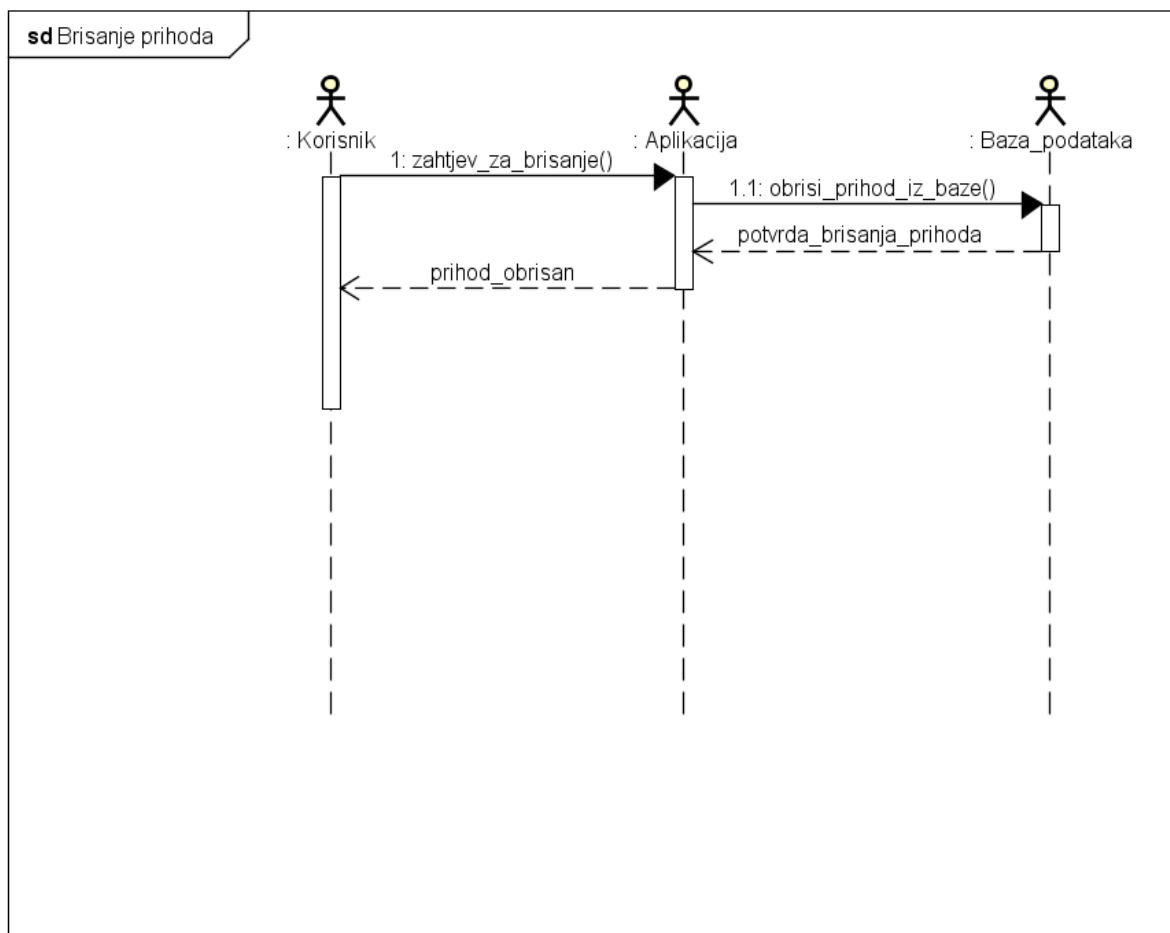
Korisnik odabire opciju unosa troškova te unosi iznos troška. Web aplikacija podatke šalje u bazu na provjeru. Ako su podaci ispravni prihod se unosi u bazu podataka. U slučaju neispravnih podataka korisniku se javlja greška.



Slika 4.1.11. Sekvencijski dijagram za UC10

Obrazac uporabe UC12 (Brisanje prihoda):

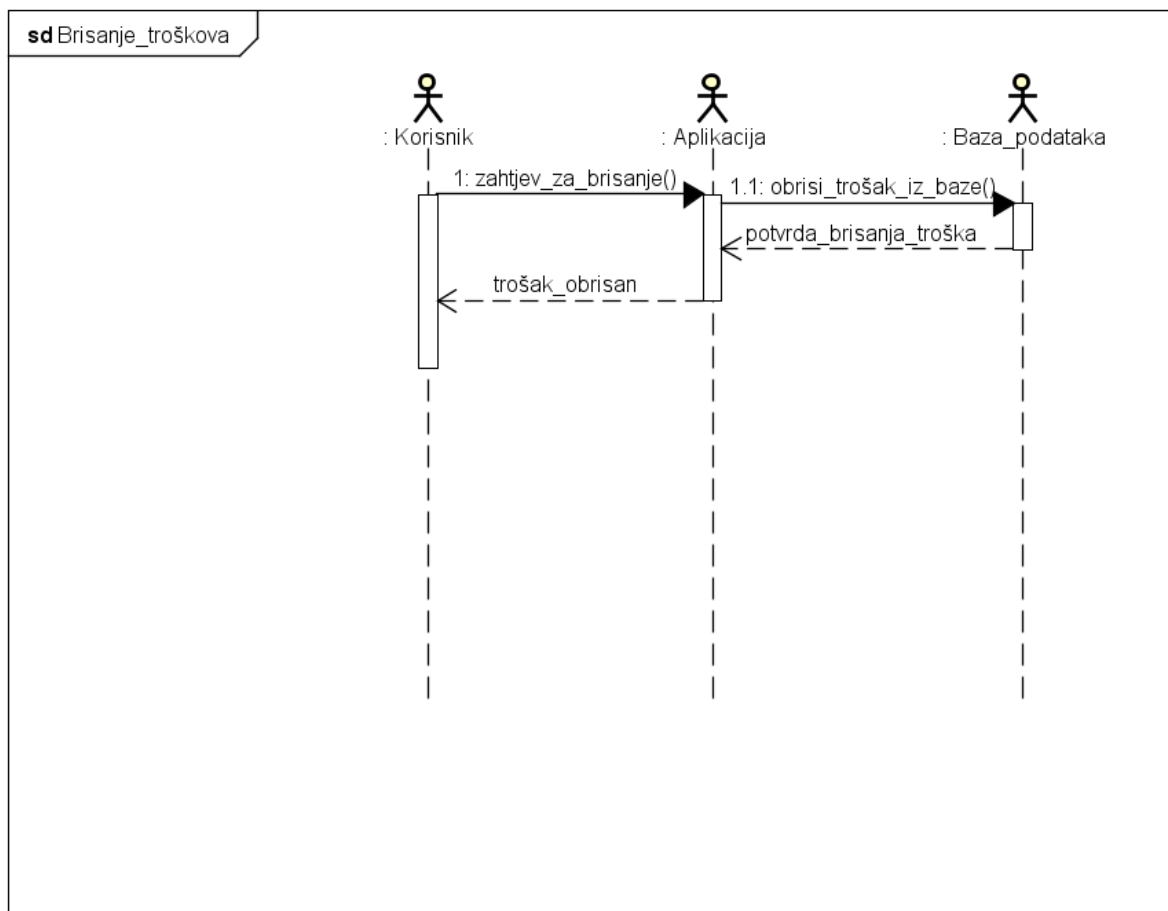
Korisnik odabire opciju brisanja prihoda. Zahtjev za brisanje dolazi poslužitelju koji provjerava ima li korisnik ima ovlasti za brisanje prihoda. Ako ima, šalje bazi zahtjev za brisanje podataka i vraća korisniku povratnu informaciju.



Slika 4.1.12.Sekvencijski dijagram za UC11

Obrazac uporabe UC12 (Brisanje troškova):

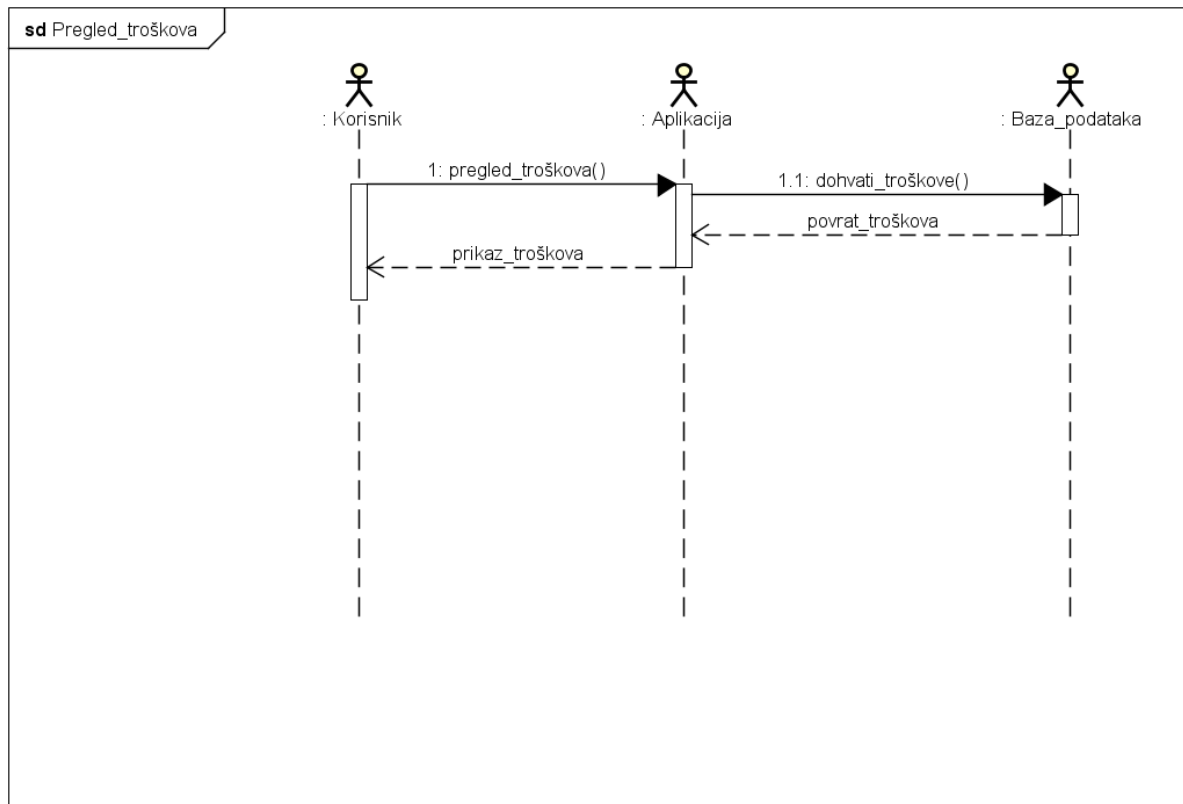
Korisnik odabire opciju brisanja troškova. Zahtjev za brisanje dolazi poslužitelju koji provjerava da li korisnik ima ovlasti za brisanje troškova. Ako ima šalje bazi zahtjev za brisanje podataka i vraća korisniku povratnu informaciju.



Slika 4.1.13.Sekvencijski dijagram za UC12

Obrazac uporabe UC13 (Pregled troškova):

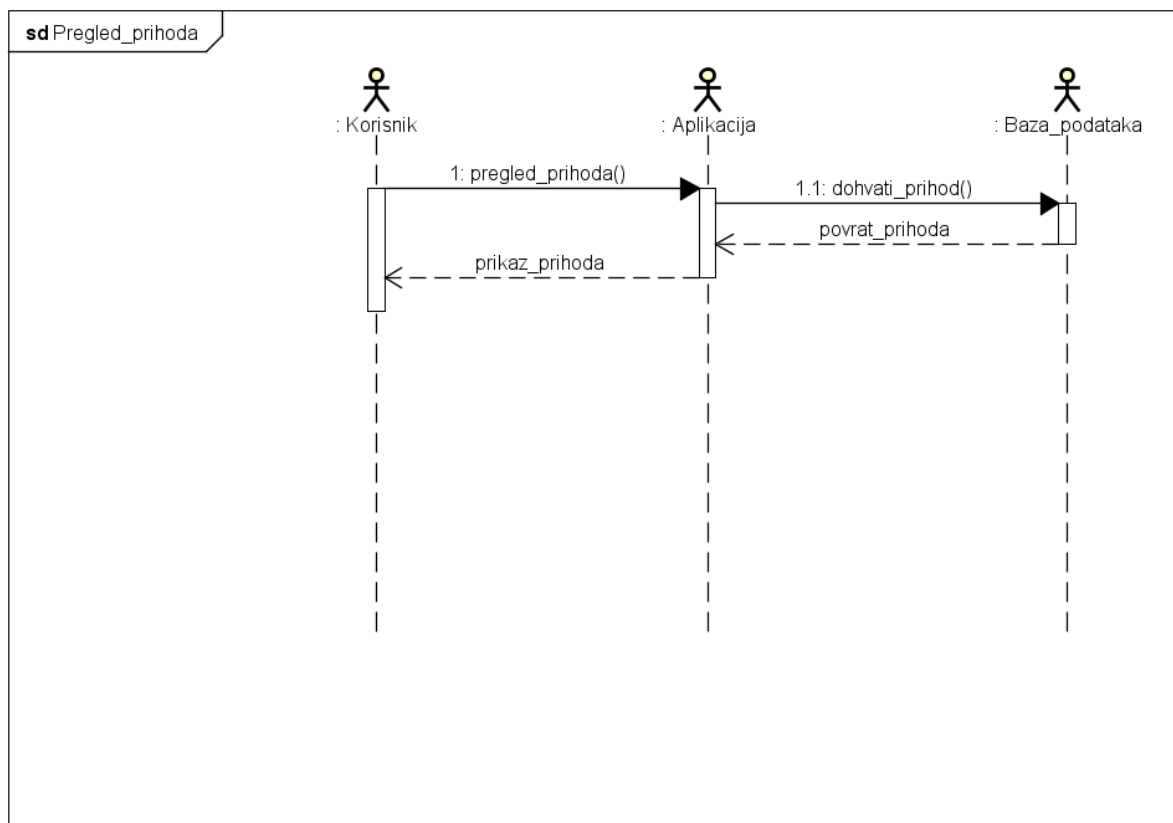
Korisnik odabire opciju pregleda troškova. Prilikom pregledavanja troškova korisnik šalje zahtjev poslužitelju te on isti prosljeđuje bazi podataka i od nje zatraži podatke troškova. Baza ih zatim proslijedi poslužitelju, a poslužitelj ih prikaže korisniku.



Slika 4.1.14.Sekvencijski dijagram za UC13

Obrazac uporabe UC14 (Pregled prihoda):

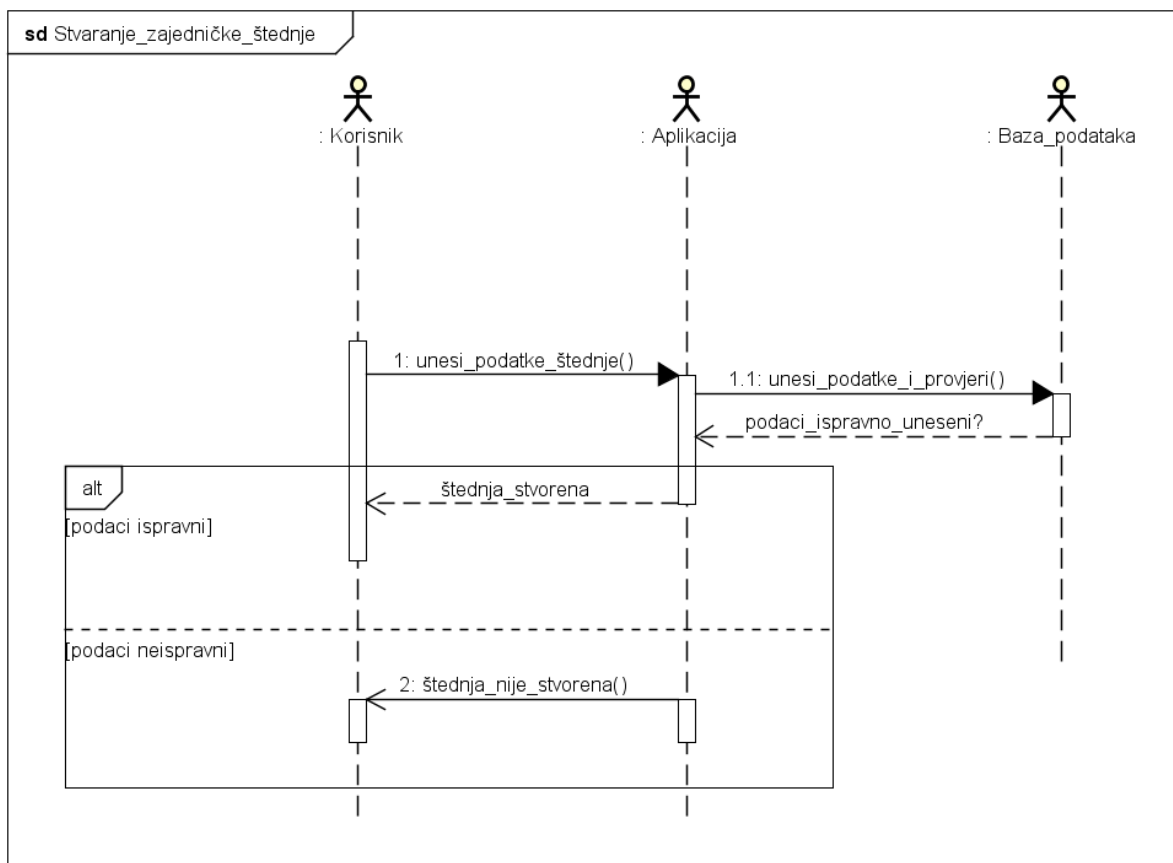
Korisnik odabire opciju pregleda prihoda. Prilikom pregledavanja prihoda korisnik šalje zahtjev poslužitelju te on isti prosljeđuje bazi podataka i od nje zatraži podatke troškova. Baza ih zatim proslijedi poslužitelju, a poslužitelj ih prikaže korisniku.



Slika 4.1.15.Sekvencijski dijagram za UC14

Obrazac uporabe UC15 (Stvaranje zaj. štednje):

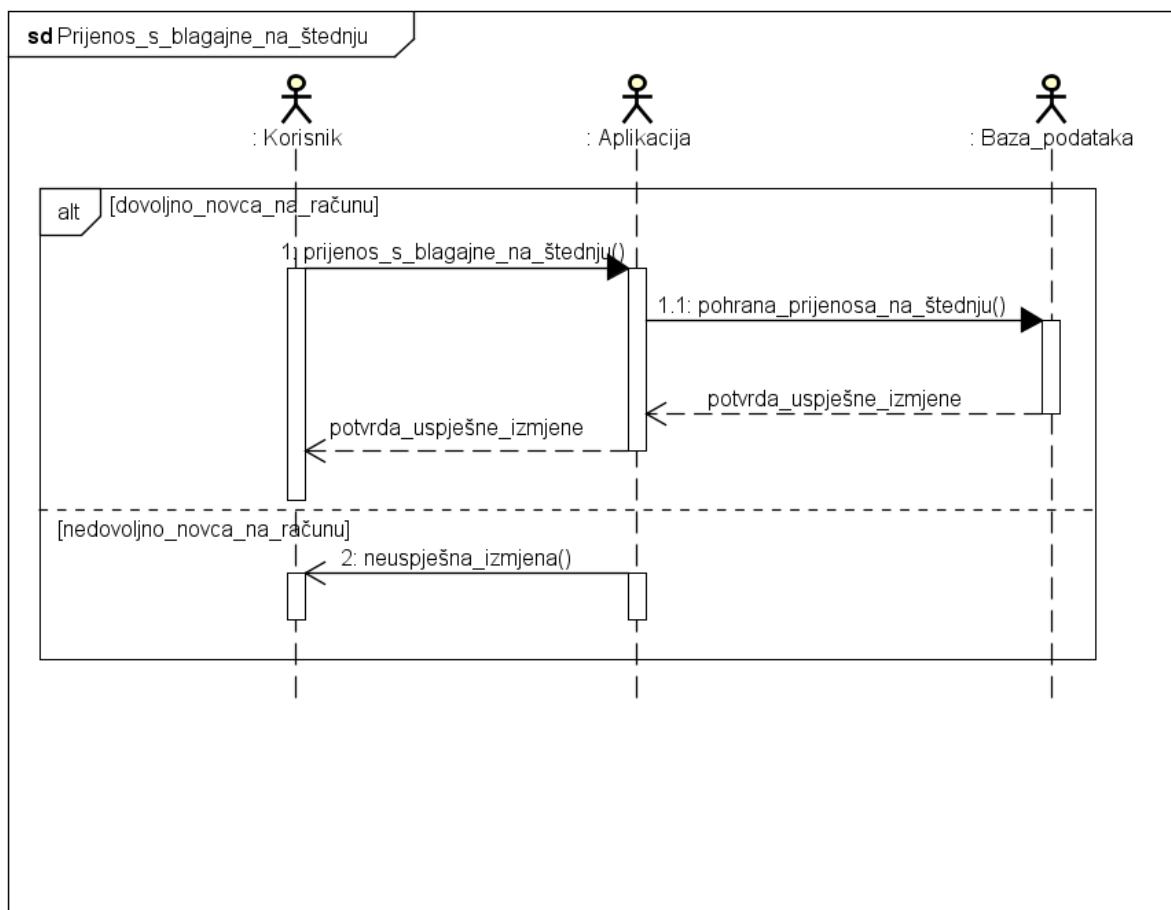
Korisnik odabire opciju kreiranja zajedničke štednje. Poslužitelj provjerava valjanost podataka te korisniku šalje poruku o ispravnosti istih. Nakon unosa podataka poslužitelj ih šalje u bazu podataka te ona kreira štednju i obavještava poslužitelj o uspješnosti koji informaciju prosljeđuje korisniku.



Slika 4.1.16.Sekvencijski dijagram za UC15

Obrazac uporabe UC16 (Prijenos s blagajne na štednju):

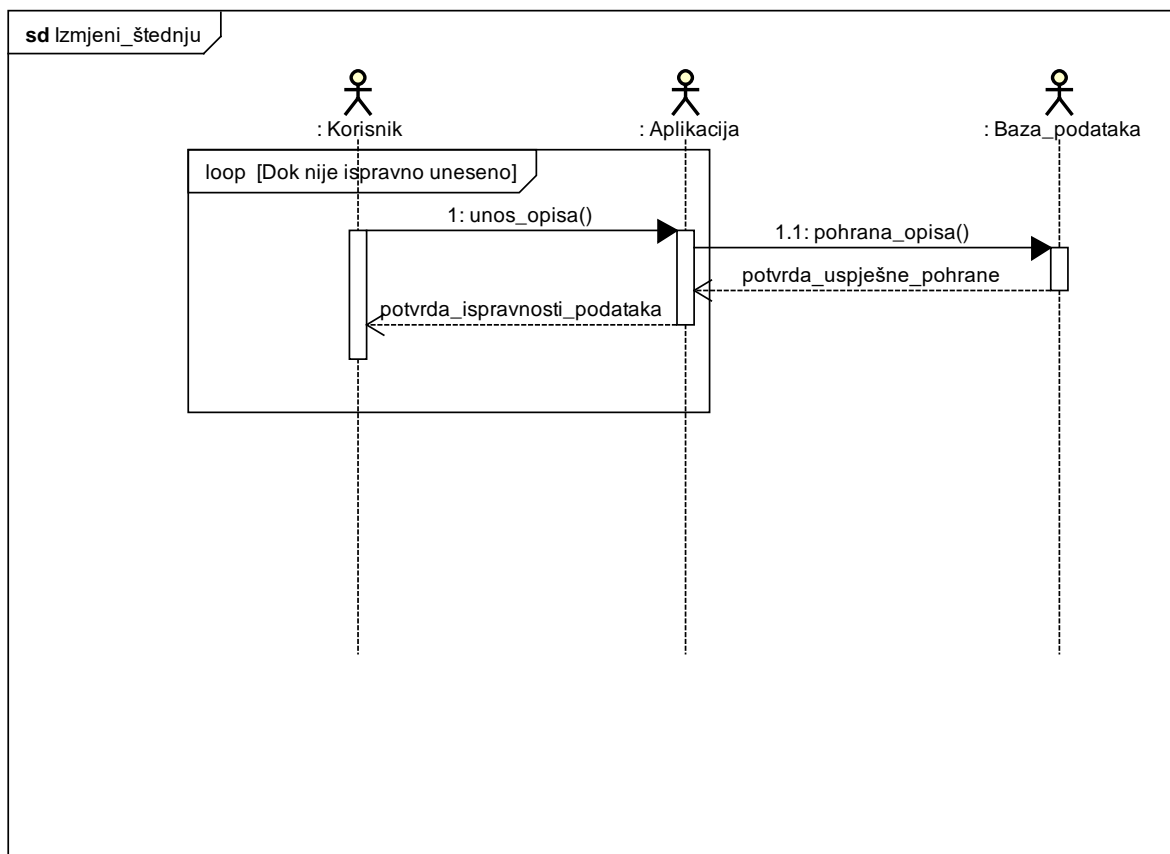
Korisnik odabire opciju prijenosa novca iz blagajne u štednju. Zatim unosi željeni iznos. Poslužitelj provjerava valjanost podataka te korisniku šalje poruku o ispravnosti istih. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka te ih ona prebaci u štednju i pohrani. Nakon uspješne 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

Obrazac uporabe UC17 (Izmijeni štednju):

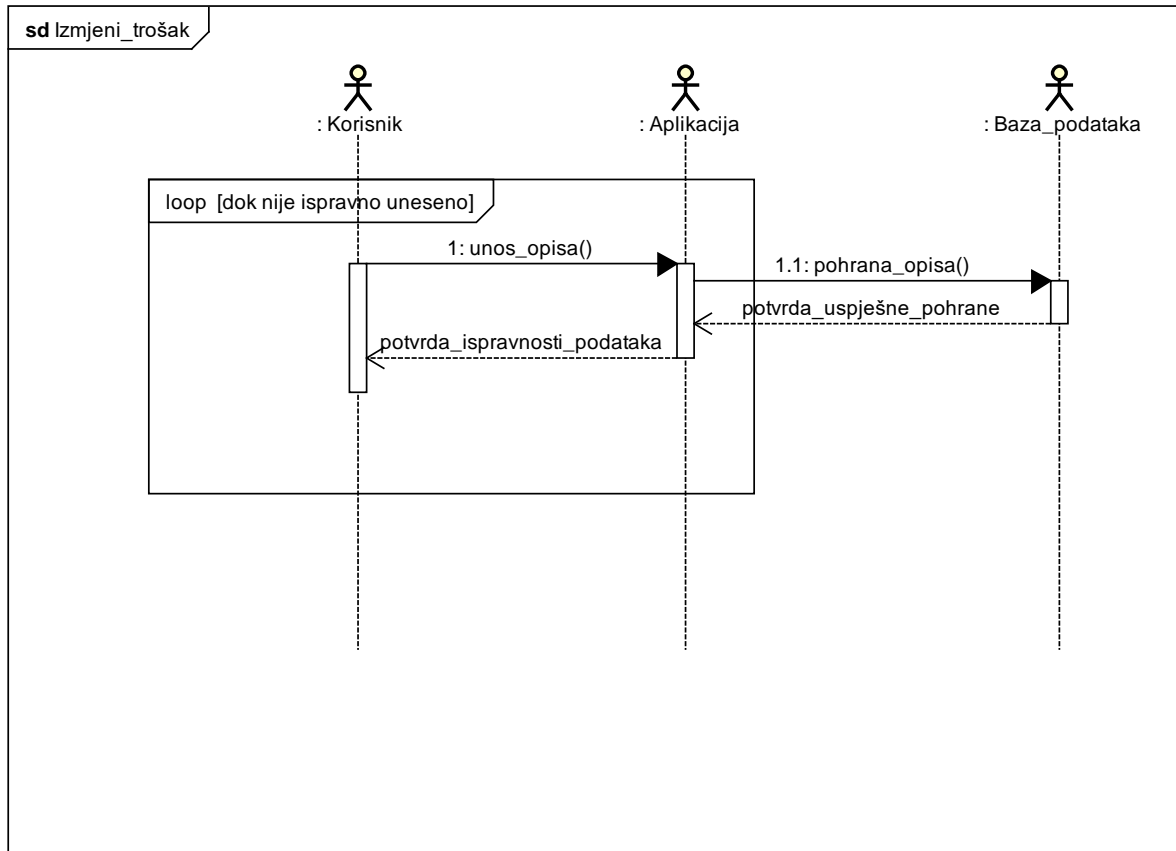
Korisnik odabire opciju izmjene štednje. Zatim unosi željenu promjenu opisa ili cilja štednje. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka gdje se pohranjuju. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj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



Slika 4.1.18.Sekvencijski dijagram za UC17

Obrazac uporabe UC18 (Izmijeni trošak):

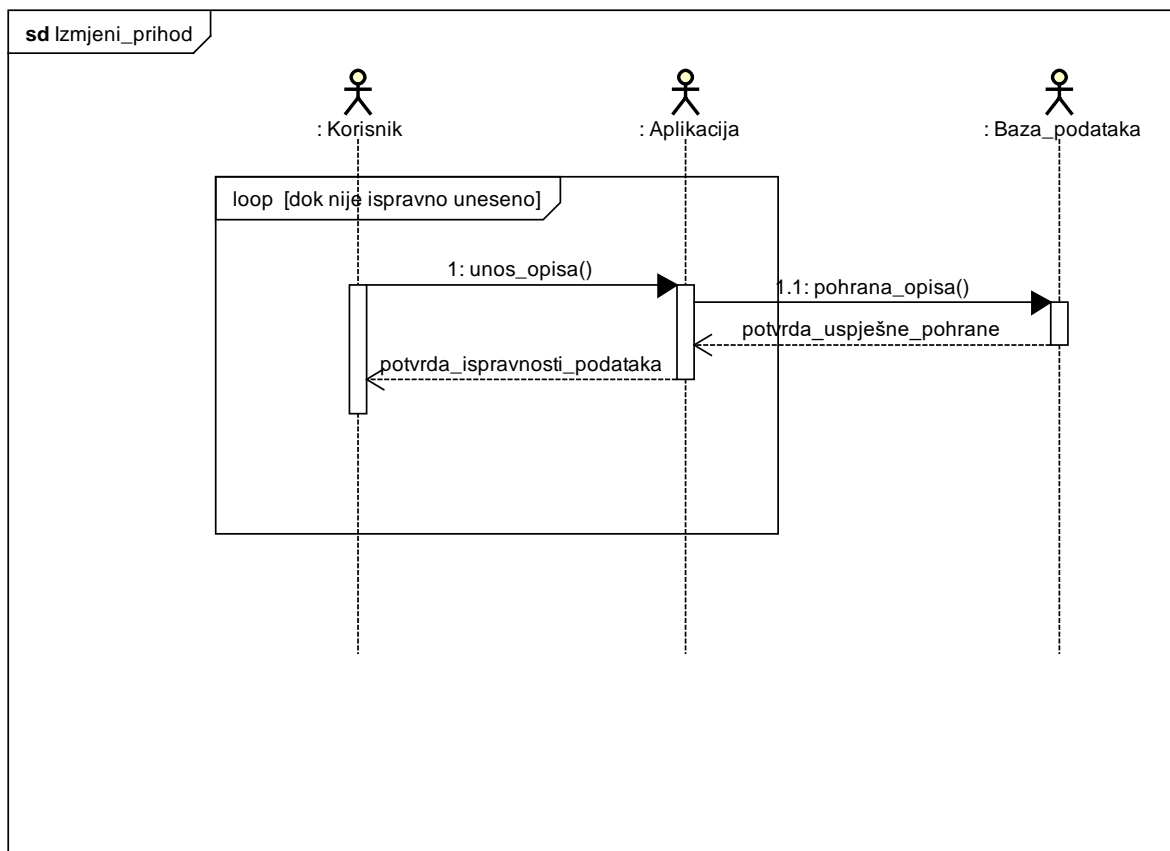
Korisnik odabire opciju izmjene troška. Zatim unosi željeni iznos troška. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka gdje se pohranjuju. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika.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



Slika 4.1.19.Sekvencijski dijagram za UC18

Obrazac uporabe UC19 (Izmijeni prihod):

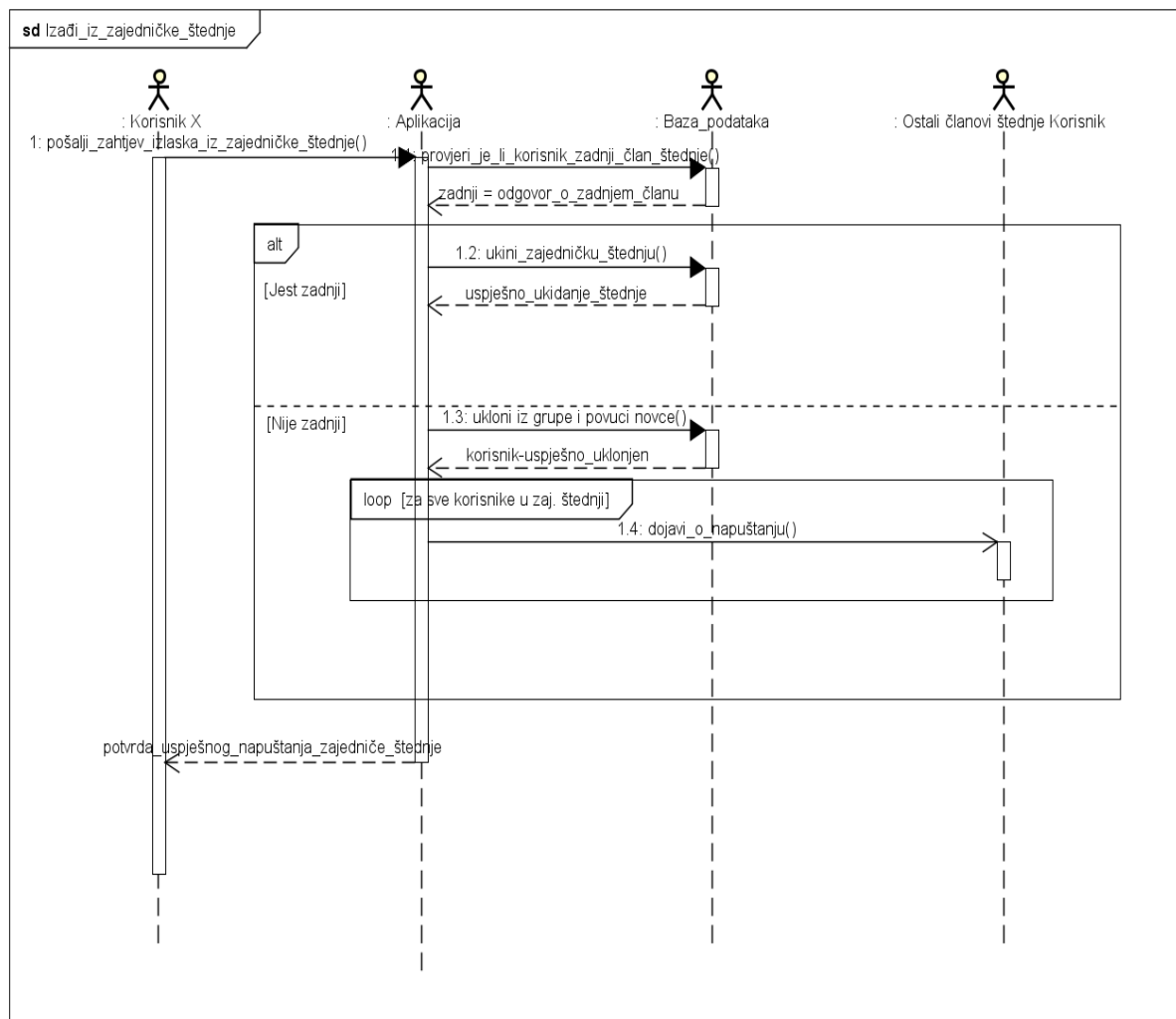
Korisnik odabire opciju izmjene prihoda. Zatim unosi željeni iznos prihoda. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka gdje se pohranjuju. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj korisnika.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



Slika 4.1.20.Sekvencijski dijagram za UC19

Obrazac uporabe UC20 (Izađi iz zajedničke štednje)

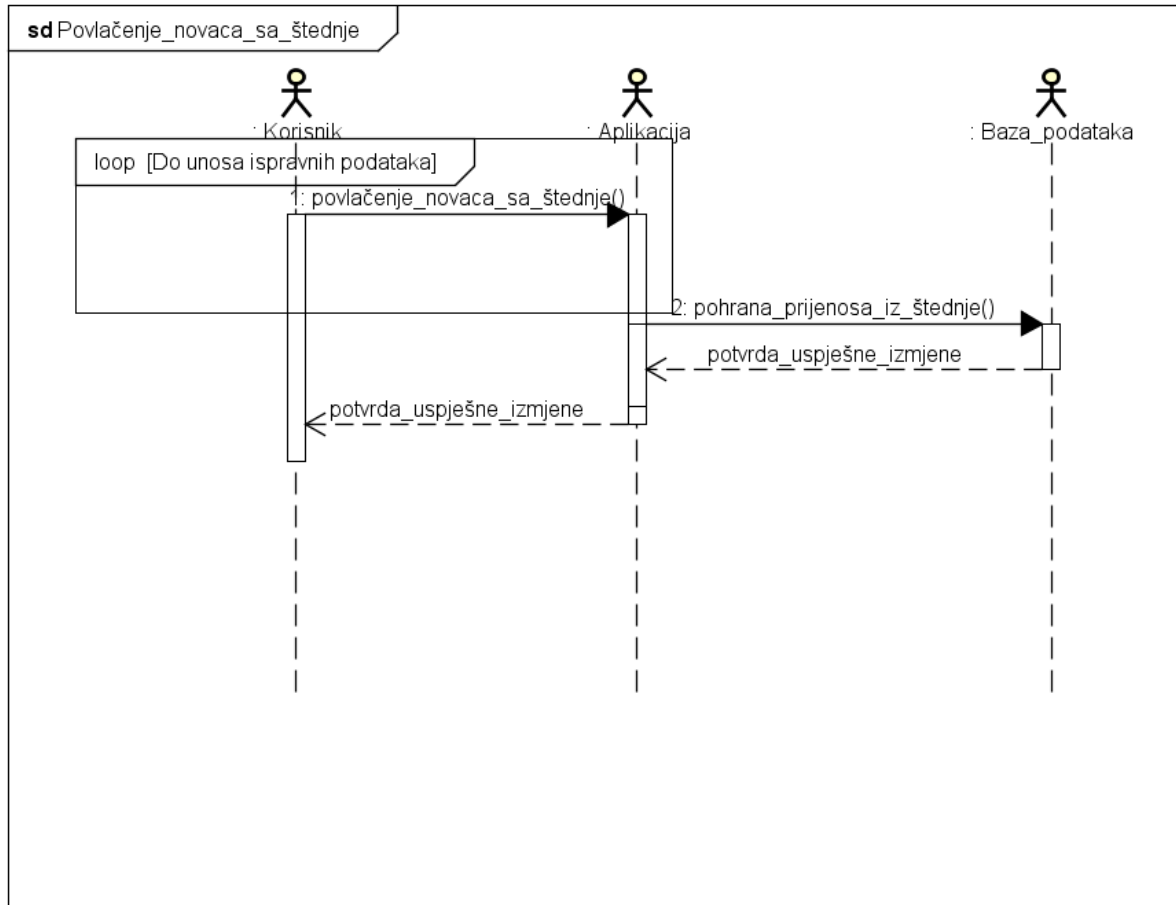
Korisnik odabire opciju izlaska iz zajedničke štednje. Zatim šalje zahtjev sa svojim podacima i podacima o zajedničkoj štednji poslužitelju. Poslužitelj šalje upit prema bazi je li taj korisnik posljednji korisnik grupe, a baza mu šalje potrebnu informaciju. Ako je korisnik posljednji član grupe, poslužitelj 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

Obrazac uporabe UC21 (Povlačenje novaca sa štednje):

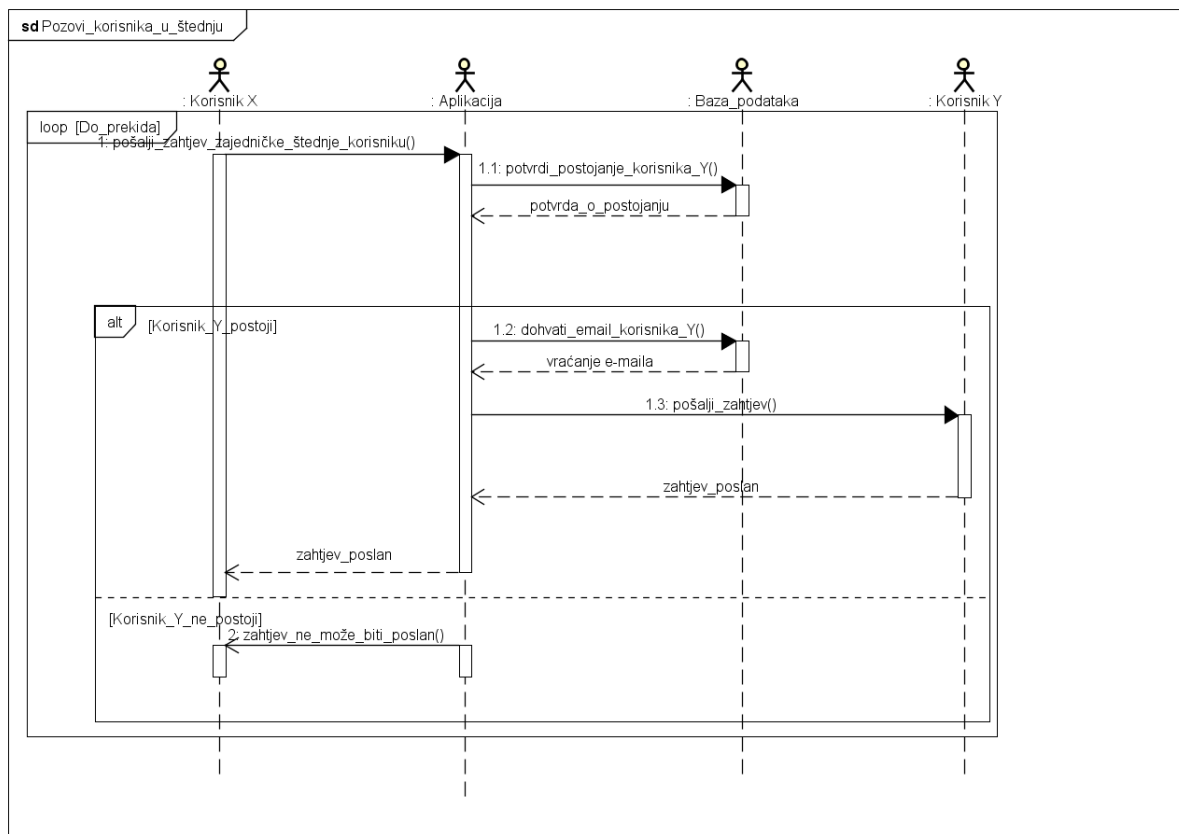
Korisnik odabire opciju povlačenja novaca iz štednje. Zahtjev se proslijeđuje poslužitelju zajedno s podacima korisnika. Zatim unosi željeni iznos povlačenja. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj šalje zahtjev u bazu podataka koja povlači novce iz štednje 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

Obrazac uporabe UC22 (Pozovi korisnika u štednju):

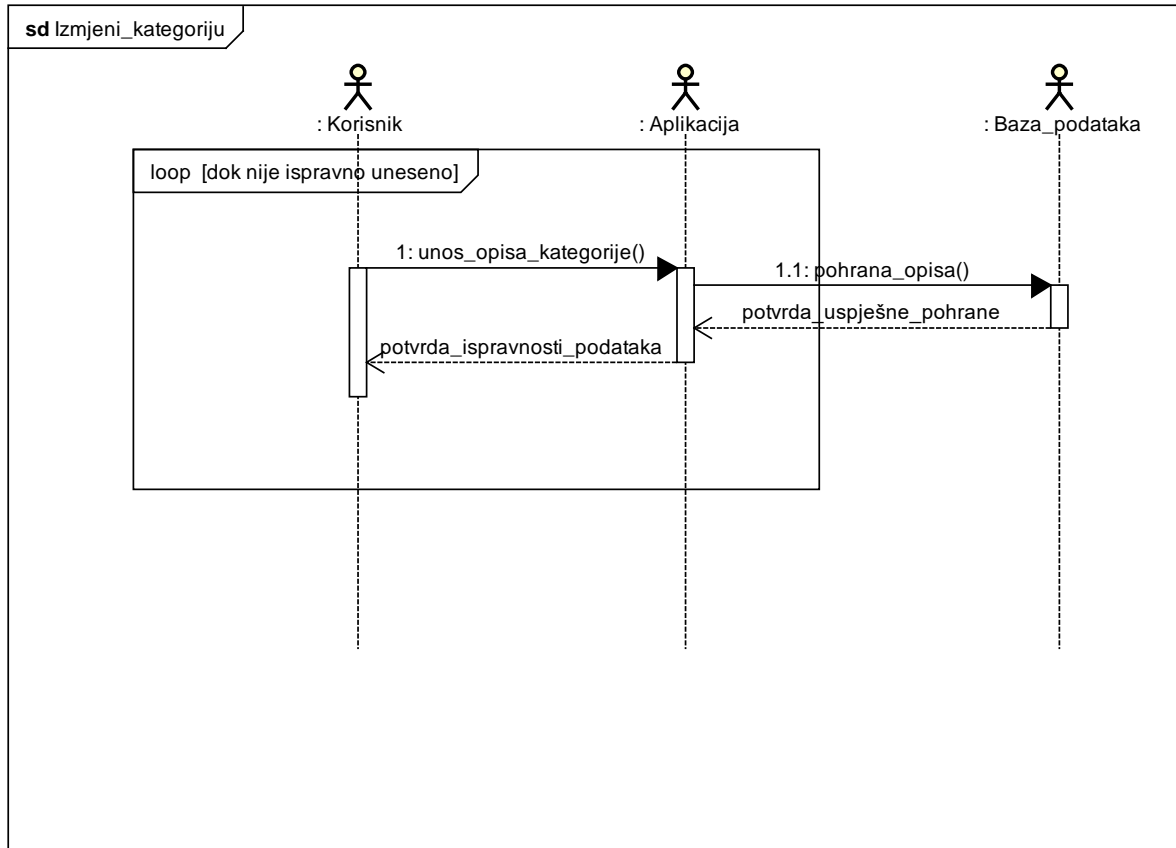
Korisnik odabire opciju pozivanja drugih korisnika u štednju te unosi kratku poruku. Poslužitelj provjerava da li je kreirana zajednička štednja. Zahtjev s porukom koji šalje korisnik X prvo dolazi do poslužitelja zajedno s korisničkim imenom korisnika Y. Poslužitelj također provjerava postojanje pozvanog korisnika Y tako da šalje zahtjev bazi podataka koja mu vraća informaciju o postojanju korisnika Y te njegovoj e-mail adresi. Poslužitelj u bazu podataka pohranjuje poruku. Ako korisnik Y postoji poslužitelj mu šalje zahtjev korisnika X na e-mail adresu te korisnik Y šalje poslužitelju potvrdu o primitku zahtjeva. Poslužitelj informira korisnika X o uspješnom slanju poruke. U slučaju neslanja zahtjeva korisniku se javlja greška.



Slika 4.1.23.Sekvencijski dijagram za UC22

Obrazac uporabe UC23 (Izmijeni kategoriju):

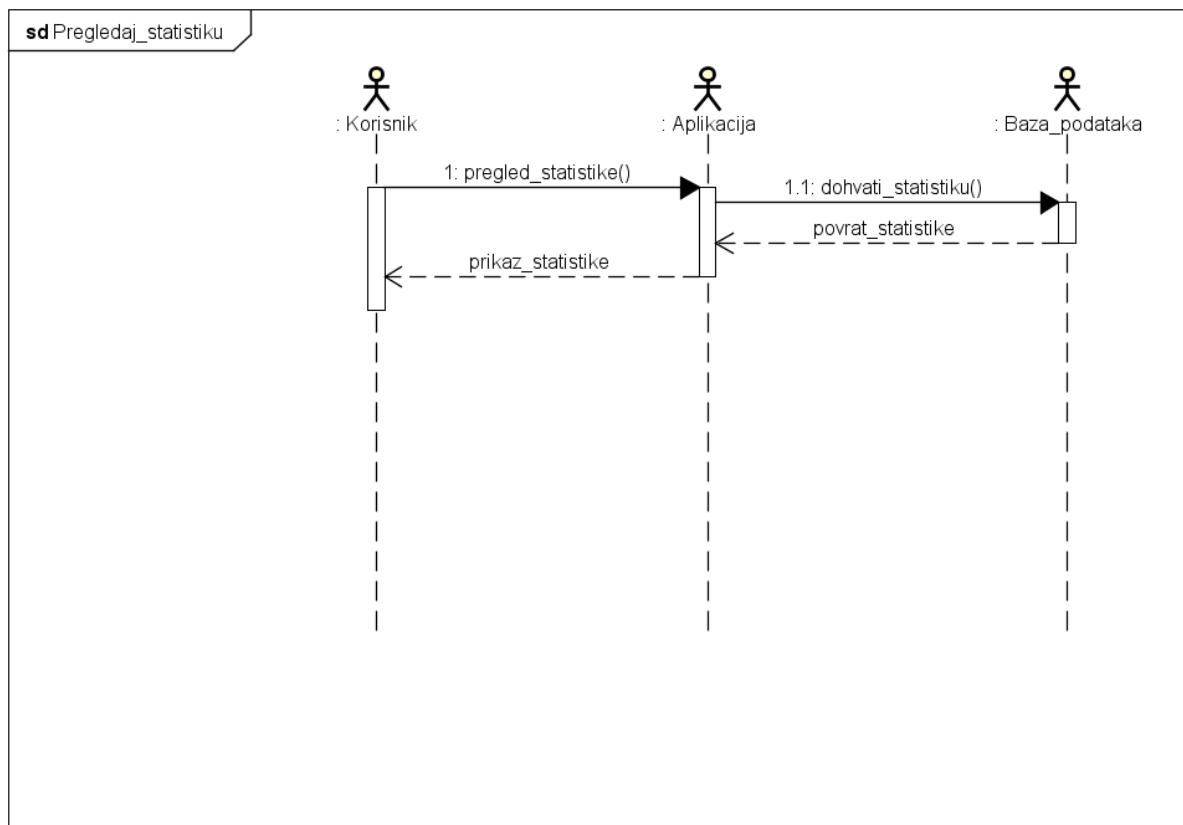
Korisnik odabire opciju izmjene kategorije. Zatim unosi željenu izmjenu opisa kategorije. Poslužitelj provjerava valjanost podataka. Nakon unosa ispravnih podataka poslužitelj ih šalje u bazu podataka gdje se pohranjuju. Nakon uspješne pohrane podataka, baza o tome obavještava poslužitelja, a poslužitelj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



Slika 4.1.24.Sekvencijski dijagram za UC23

Obrazac uporabe UC24 (Pregledaj statistiku):

Korisnik odabire blagajnu/štednju te zatim opciju pregleda statistika o istima. Prilikom pregledavanja statistika korisnik šalje zahtjev poslužitelju te on isti prosljeđuje bazi podataka i od nje zatraži podatke blagajne/štednje. Baza ih zatim proslijedi poslužitelju, a poslužitelj ih prikaže korisniku.



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

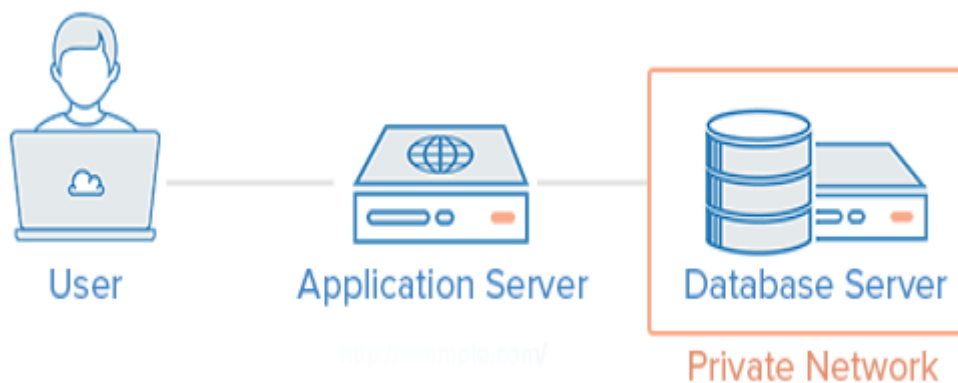
6. Arhitektura i dizajn sustava

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-usmjerenom 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 provjerenim 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 podacima. 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 funkcionira na 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 attribute:

- 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 attribute:

- 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 attribute:

- 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 attribute:

- 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 attribute:

- 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 attribute:

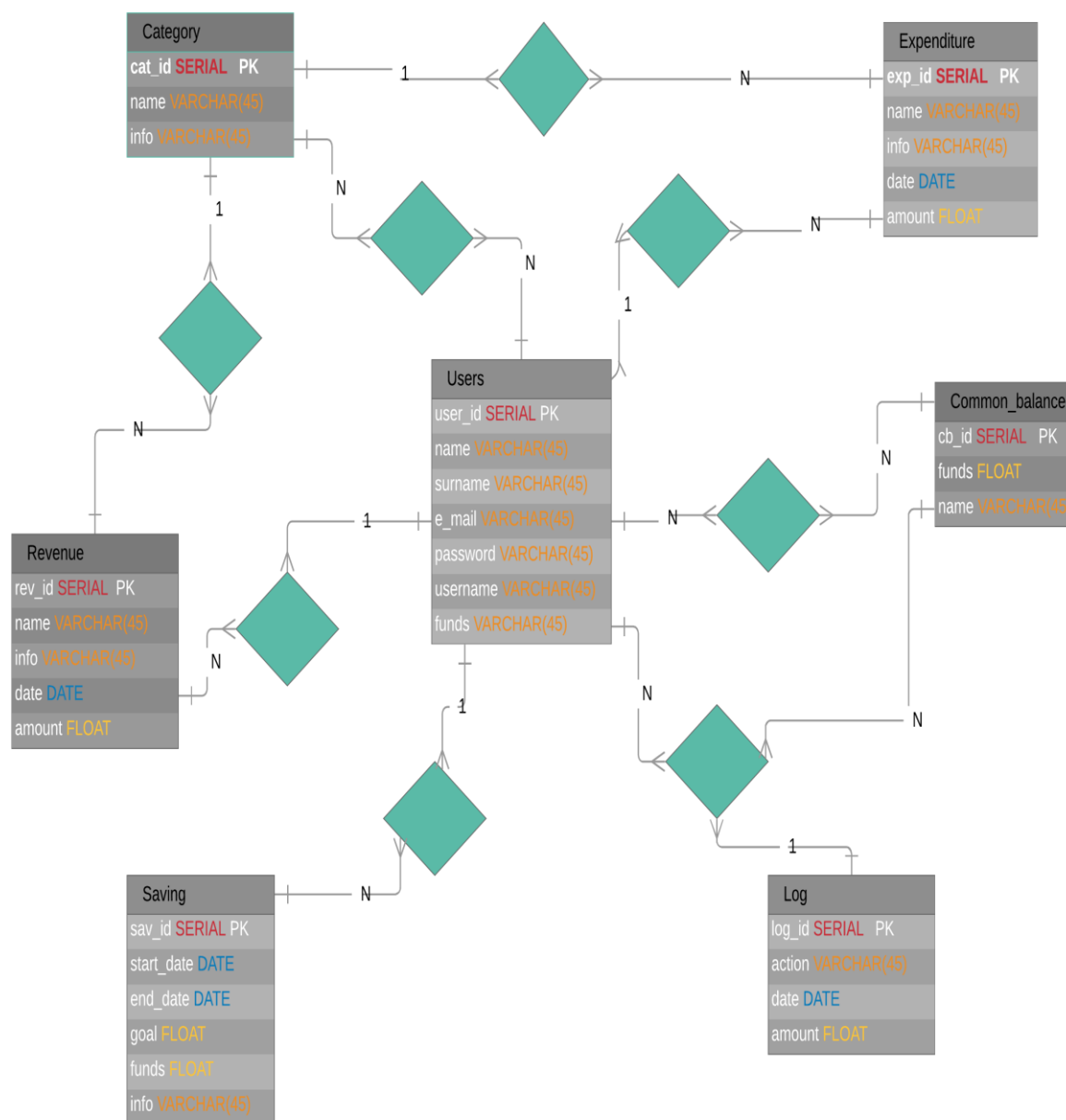
- log_id – tekstualni identifikator,
- action – radnja,
- e_mail – email adresa korisnika,
- date – datum prijave,
- amount – iznos upisanih prihoda/troškova/štednji,
- user_id – tekstualni identifikator korisnika,
- cb_id – tekstualni identifikator zajedničke štednje.

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 attribute:

- 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

Registration - Pogled u kojem korisnik stvara svoj račun

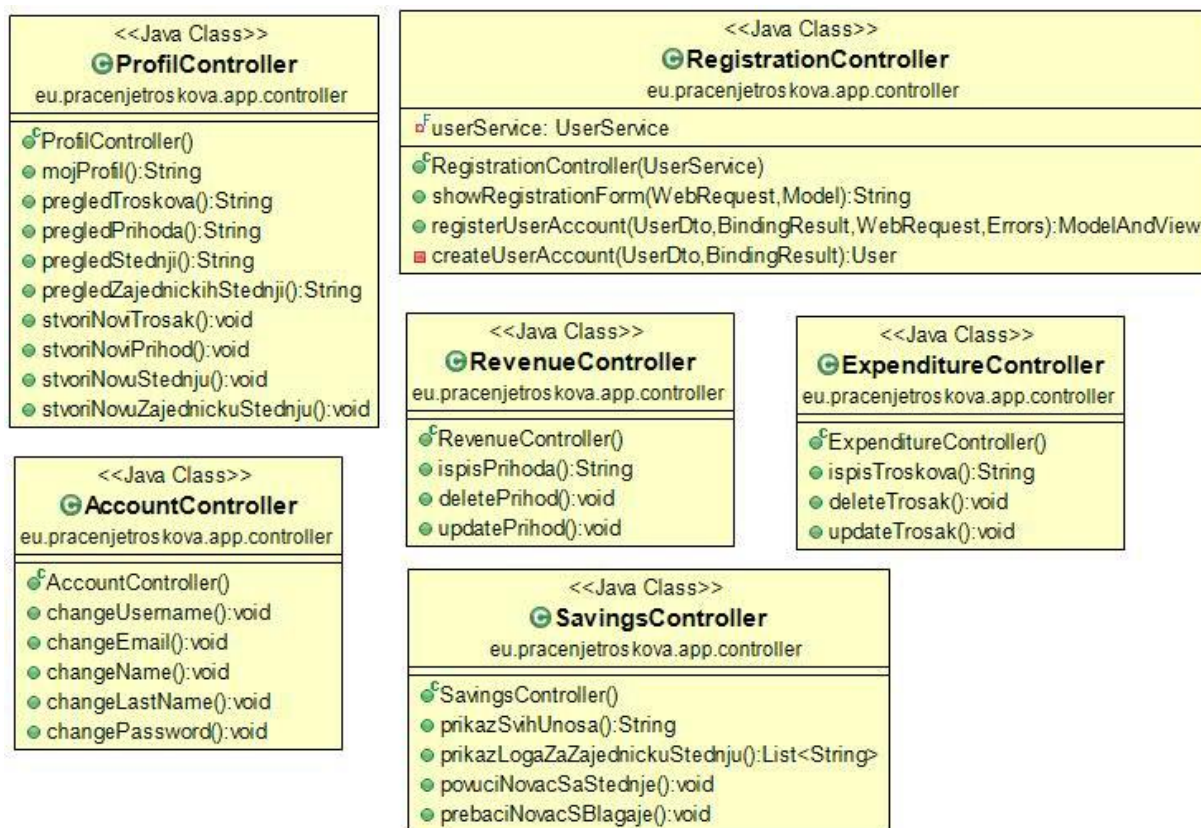
Profil - Pogled u kojem korisnik vidi prikaz svog profila

Account - Pogled u kojem korisnik može mijenjati postavke svog profila

Savings - Pogled u kojem korisnik može unositi i vidjeti svoje štednje

Revenue - Pogled u kojem korisnik može unositi i vidjeti svoje prihode

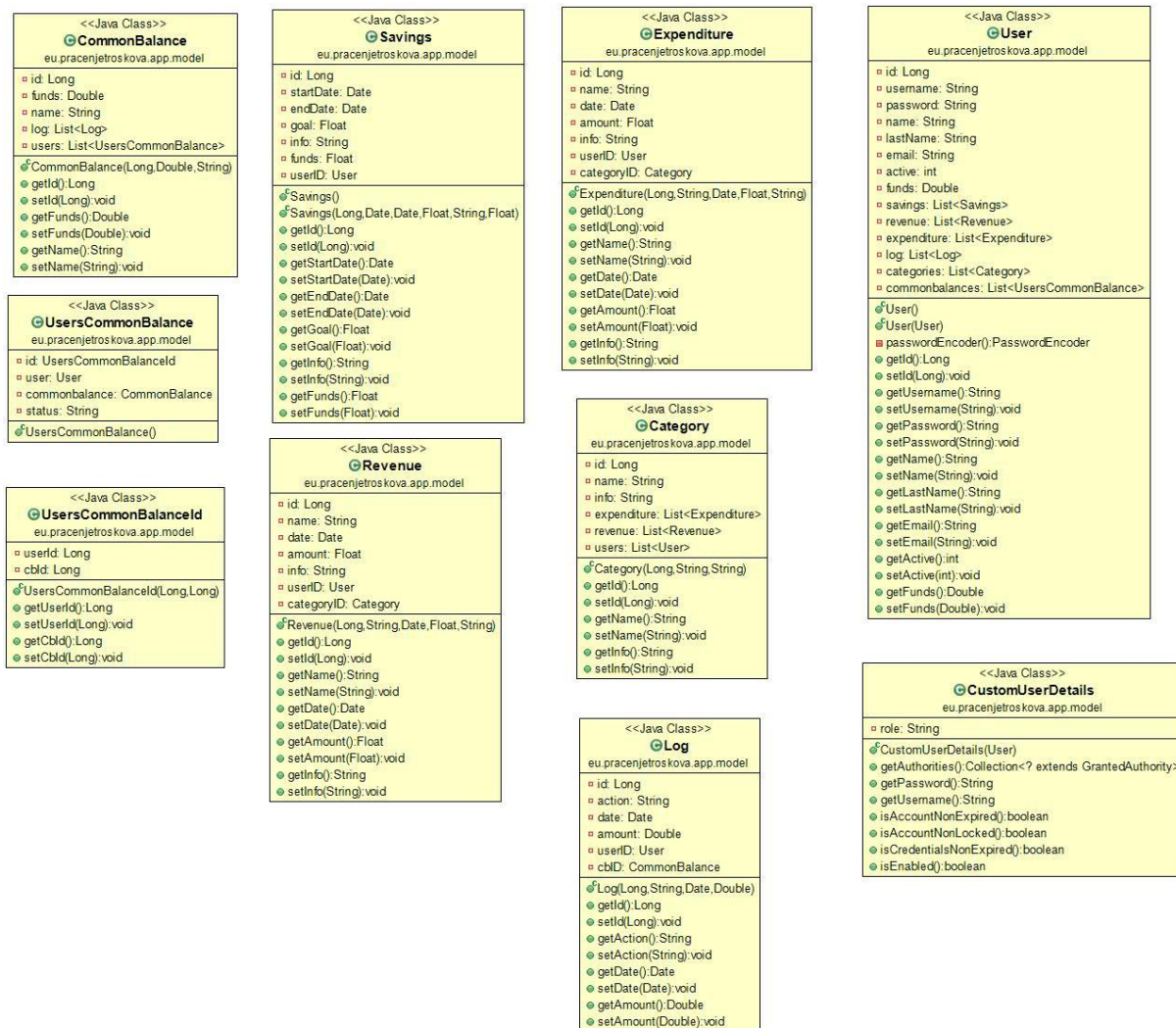
Expenditure - Pogled u kojem korisnik može unositi i vidjeti svoje troškove



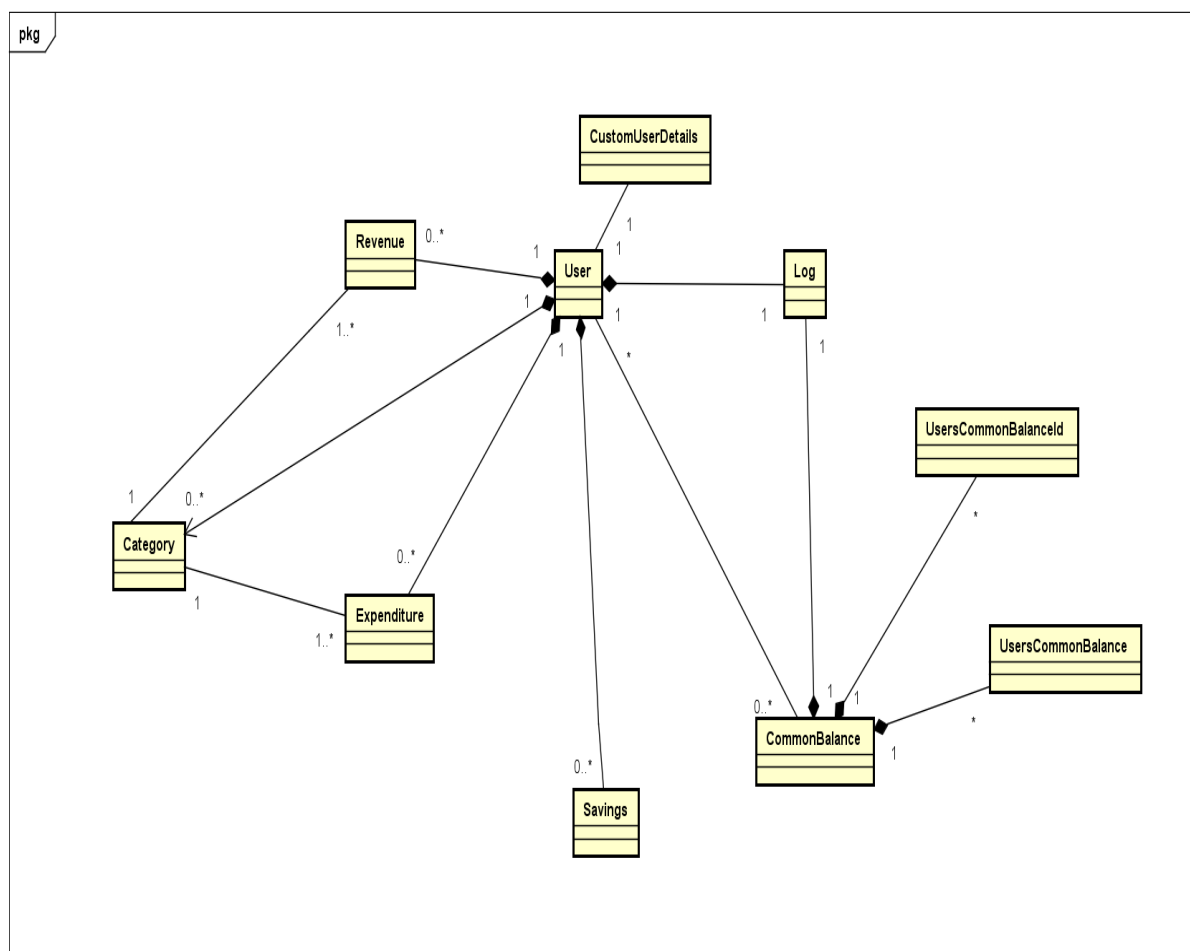
Slika 6.2.1 Razredi paketa pogledi

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.



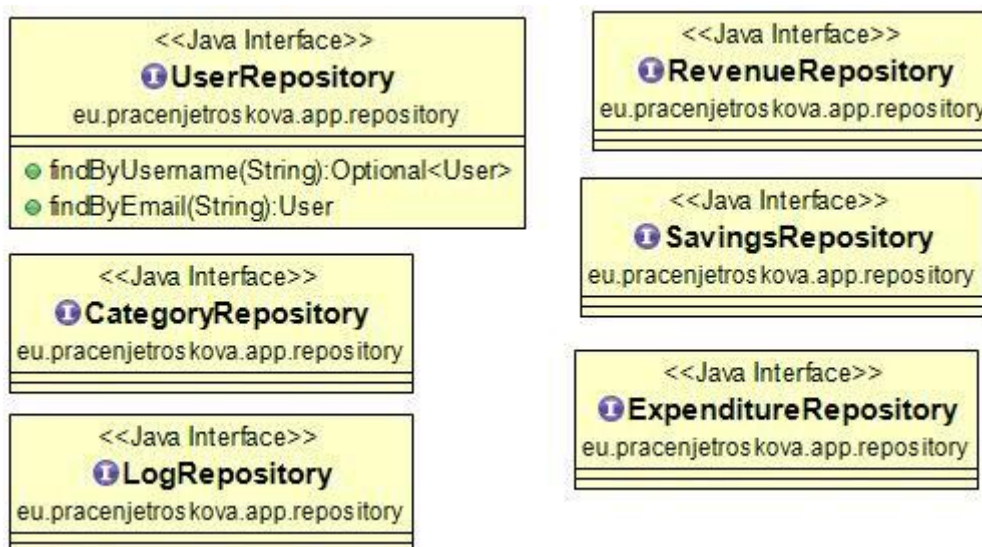
Slika 6.2.2 Razredi paketa entiteti



Slika 6.2.3 Dijagram razreda

Repozitoriji

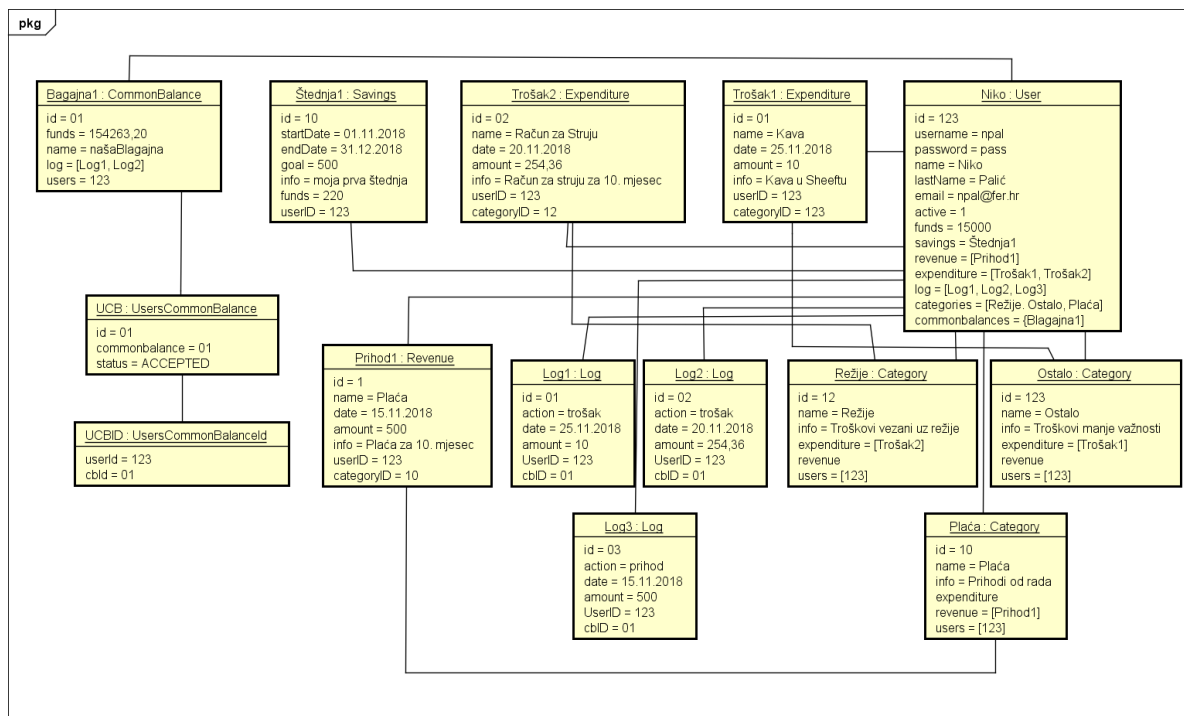
Repozitoriji predstavljaju razrede koji omogućuju izravnu komunikaciju s bazom podataka i dohvaćanje podataka iz baze podataka u okviru Java Spring tehnologije. Repozitoriji predstavljaju podatkovni sloj.



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

Ovdje počinju sadržaji Revizije 2.

U ovom potpoglavlju potrebno je uvrstiti dodatna četiri dijagrama koji prikazuju arhitekturu sustava i to: komunikacijski dijagram, dijagram stanja, dijagram aktivnosti i dijagram komponenti. Potrebno je dati barem po jedan primjerak svakog od dijagrama. Potrebno je da svaki od dijagrama prikazuje (jedan) bitan dio funkcionalnosti sustava. Dijagram komponenti treba prikazivati sve komponente sustava. Prema potrebi, ovdje se mogu dodati i neki dodatni sekvencijski dijagrami za objekte.

7. Implementacija i korisničko sučelje

U ovom poglavlju potrebno je:

dati dijagram razmještaja (deployment dijagram)

navesti koje su tehnologije i alati korišteni u razvoju sustava

dati isječak programskog koda koji implementira neku od temeljnih funkcionalnosti u sustavu

objasniti kako je ispitano implementirano rješenje i pokazati bar 4 ispitna scenarija

dati upute za instalaciju

dati upute za korištenje

7.1. *Dijagram razmještaja*

Potrebno je umetnuti dijagram razmještaja i po potrebi ga opisati.

7.2. Korištene tehnologije i alati

7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava

U ovom poglavlju potrebno je prikazati isječak programa koji prema mišljenju studenta ostvaruje temeljnu funkcionalnost u sustavu (ili nekom modulu).

7.4. Ispitivanje programskog rješenja

U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti s prikazom odabira ispitnih slučajeva.

Prilikom prezentacije svojih Ispitnih scenarija (minimalno četiri) studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete. Poželjno je da se napravi i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane te da se vidi na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Različiti ulazi za ispitne scenarije trebaju pokrivati temeljnu funkcionalnost nekog modula i nekoliko rubnih uvjeta.

7.5. Upute za instalaciju

U ovom poglavlju potrebno je dati upute za instalaciju ostvarenog prototipa.

7.6. Korisničke upute

Korisničke upute ovisit će o količini implementirane funkcionalnosti. Očekuje se da upute budu na oko pet A4 stranica koje će dati potpuni opis funkcionalnosti sustava sa stajališta krajnjeg korisnika.

8. Zaključak i budući rad

Projekt je analiziran i osnovno rješenje je zamišljeno. Definirane su karakteristike, zahtjevi i procesi sustava te sada slijedi implementacija istoga!

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

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)

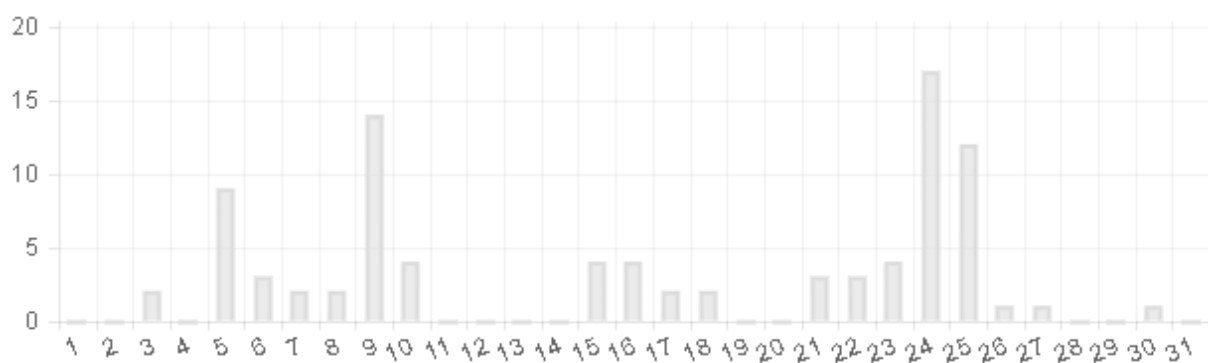
#7.11.2018

Sastanak s asistenticom

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 pojmova	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						
Implementacija i korisničko sučelje						
Dijagram razmještaja						
Korištene tehnologije i alati	0	100%	0	0	0	0
Isječak programskog kôda						
Ispitivanje programskog rješenja						
Upute za instalaciju						
Korisničke upute						
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

Do sada je napravljena većina dokumentacije pa time je i definiran sustav, njegovi zahtjevi i arhitektura. Plan je u drugoj reviziji nadopuniti dokumentaciju dodatnim dijagramima i razviti program sukladno dokumentaciji!