

Oblikovanje programske potpore

Ak. god. 2018./2019.

Praćenje troškova

Dokumentacija, Rev. 2

Grupa: TheGenerics

Voditelj: Domagoj Jakšić

Datum predaje: 17. 1. 2019.

Nastavnik: Ivana Perčinlić

Sadržaj

1.	Dnevnik promjena dokumentacije	5
2.	Opis projektnog zadatka.....	7
3.	Pojmovnik.....	10
4.	Funkcionalni zahtjevi	11
4.1	Dionici.....	11
4.2	Aktori i njihovi funkcionalni zahtjevi	11
4.3	Opis obrazaca uporabe	12
4.3.1	UC01: Registracija korisnika	12
4.3.2	UC02: Prijava korisnika.....	12
4.3.3	UC03: Dodaj trošak.....	13
4.3.4	UC04: Izmijeni trošak.....	14
4.3.5	UC05: Obriši trošak.....	15
4.3.6	UC06: Dodaj prihod	15
4.3.7	UC07: Izmijeni prihod.....	16
4.3.8	UC08: Obriši prihod	16
4.3.9	UC09: Stvori štednju	17
4.3.10	UC10: Izmijeni štednju	18
4.3.11	UC11: Obriši štednju	18
4.3.12	UC12: Dodaj novac u štednju	19
4.3.13	UC13: Uzmi novac iz štednje	19
4.3.14	UC14: Stvori grupu	20
4.3.15	UC15: Pozovi korisnika u grupu.....	21
4.3.16	UC16: Uredi postavke grupe.....	22
4.3.17	UC17: Napusti grupu	22
4.3.18	UC18: Odgovori na poziv u grupu.....	23
4.3.19	UC19: Izbriši grupu	24
4.3.20	UC20: Dodaj kategoriju troška	24
4.3.21	UC21: Izmijeni kategoriju troška	25
4.3.22	UC22: Obriši kategoriju troška	25
4.3.23	UC23: Podesi korisničke postavke.....	26
4.3.24	UC24: Odjava korisnika	26
4.3.25	UC25: Promjena lozinke	27

4.3.26 UC26: Dodaj kategoriju prihoda	28
4.3.27 UC27: Izmijeni kategoriju prihoda	28
4.3.28 UC28: Obriši kategoriju prihoda	29
4.4 Sekvencijski dijagrami	33
4.4.1 UC01: Registracija korisnika	33
4.4.2 UC02: Prijava korisnika	34
4.4.3 UC03: Dodaj trošak	35
4.4.4 UC04: Izmijeni trošak	36
4.4.5 UC05: Obriši trošak	37
4.4.6 UC06: Dodaj prihod	38
4.4.7 UC07: Izmijeni prihod	39
4.4.8 UC08: Obriši prihod	40
4.4.9 UC09: Stvori štednju	41
4.4.10 UC10: Izmijeni štednju	42
4.4.11 UC11: Obriši štednju	43
4.4.12 UC12: Dodaj novac u štednju	44
4.4.13 UC13: Uzmi novac iz štednje	45
4.4.14 UC14: Stvori grupu	46
4.4.15 UC15: Pozovi korisnika u grupu	47
4.4.16 UC16: Uredi postavke grupe	48
4.4.17 UC17: Napusti grupu	49
4.4.18 UC18: Odgovori na poziv u grupu	50
4.4.19 UC19: Izbriši grupu	51
4.4.20 UC20: Dodaj kategoriju troška	52
4.4.21 UC21: Izmijeni kategoriju troška	53
4.4.22 UC22: Obriši kategoriju troška	54
4.4.23 UC23: Podesi korisničke postavke	55
4.4.24 UC24: Odjava korisnika	56
4.4.25 UC25: Promjena lozinke	57
4.4.26 UC26: Dodavanje kategorije prihoda	58
4.4.27 UC27: Izmijeni kategoriju prihoda	59
4.4.28 UC28: Obriši kategorije prihoda	60
5. Ostali zahtjevi	61
6. Arhitektura i dizajn sustava	62

6.1	Svrha, opći prioriteti i skica sustava.....	62
6.2	Dijagram razreda s opisom.....	66
6.3	Dijagram objekata.....	69
6.4	Ostali UML dijagrami	72
6.4.1	Komunikacijski dijagrami.....	72
6.4.2	Dijagrami aktivnosti.....	73
6.4.3	Dijagram stanja	75
6.4.4	Dijagram komponenti	77
7.	Implementacija i korisničko sučelje	78
7.1	Dijagram razmještaja.....	78
7.2	Korištene tehnologije i alati.....	79
7.3	Isječak programskog koda vezan za temeljnu funkcionalnost sustava	81
7.4	Ispitivanje programskog rješenja	85
7.5	Upute za instalaciju	99
7.5.1	Upute za instalaciju Java Development Kita (JDK)	99
7.5.2	Upute za instalaciju integrirane razvojne okoline (IDE)	101
7.5.3	Upute za instalaciju podrške za rad s bazom podataka	102
7.5.4	Upute za instalaciju Gita	103
7.5.5	Upute za postavljanje aplikacije na poslužitelj	104
7.6	Korisničke upute	107
8.	Zaključak i budući rad.....	119
9.	Popis literature	120
10.	Dodatak A: Indeks (slika, dijagonama, tablica, ispisa kôda).....	121
11.	Dodatak B: Dnevnik sastajanja.....	124
12.	Dodatak C: Prikaz aktivnosti grupe	126
13.	Dodatak D: Plan rada / Pregled rada i stanje ostvarenja	128

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autor(i)	Datum
0.1	Napravljen predložak.	Domagoj Jakšić	16.10.2018.
0.2	Dodan opis projektnog zadatka.	Robert Čiček, Domagoj Jakšić	21.10.2018.
0.3	Dodani funkcionalni zahtjevi.	Domagoj Jakšić	21.10.2018.
0.3.1	Dodani ostali zahtjevi.	Domagoj Jakšić	21.10.2018.
0.4	Dodani obrasci uporabe.	Domagoj Jakšić, Vitomir Marjanović, Vlatko Magjer, Domagoj Kroflin	21.10.2018.
0.4.1	Dodani dijagrami obrazaca uporabe.	Dominik Galić	28.10.2018.
0.4.2	Dodani sekvencijski dijagrami.	Vlatko Magjer, Vitomir Marjanović, Domagoj Kroflin	29.10.2018.
0.5	Dodana svrha, opći prioriteti i skica sustava.	Dominik Galić	29.10.2018.
0.5.1	Dodani dijagrami razreda s opisom.	Domagoj Jakšić	29.10.2018.
0.6	Dodan pojmovnik.	Domagoj Jakšić	29.10.2018.
0.7	Dodan dijagram objekata.	Robert Čiček	29.10.2018.
0.8	Ažuriran dodatak A, B, C, D.	Domagoj Jakšić	15.11.2018.
1.0	Predaja prve verzije projektne dokumentacije.		28.11.2018.
1.1	Dodan dijagram razmještaja.	Domagoj Kroflin	4.1.2019.
1.2	Dodane korištene tehnologije i alati.	Robert Čiček	4.1.2019.
1.3	Dodani komunikacijski dijagrami.	Vlatko Magjer	5.1.2019.
1.4	Dodane upute za instalaciju.	Robert Čiček	6.1.2019.
1.5	Dodani dijagrami aktivnosti.	Vlatko Magjer	11.1.2019.
1.6	Dodani dijagrami komponenti i stanja, izmjena u modelu baze podataka.	Dominik Galić	13.1.2019.

1.7	Dodani isječci programskog koda.	Domagoj Jakšić	13.1.2019.
1.8	Dodano ispitivanje programskog rješenja.	Vitomir Marjanović	13.1.2019.
1.9	Dodane korisničke upute.	Vitomir Marjanović	13.1.2019.
2.0	Predaja druge verzije projektne dokumentacije.		17.1.2019.

2. Opis projektnog zadatka

Cilj projektnog zadatka „Praćenje troškova“ jest razviti web aplikaciju prilagođenu mobilnom uređaju za praćenje i vođenje troškova pojedinca ili grupa većeg broja korisnika. Korisnici će moći upravljati vlastitim prihodima, troškovima i štednjama te dijeliti iste s članovima svojih grupa.

Uvjet za korištenja aplikacije jest registracija ili prijava korisnika u sustav. Registrirati se može bilo koji novi korisnik tako da unese osnovne podatke potrebne za korištenje aplikacije.

Obavezni podatci za registraciju:

- jedinstveno korisničko ime
- lozinka
- e-mail adresa (koju je potrebno verificirati)
- ime korisnika
- prezime korisnika

U sustav se može prijaviti samo registrirani korisnik.

Otvaranjem aplikacije korisniku se prikazuje pregled trenutnog stanja računa te popisi troškova, prihoda, štednji i notifikacija. Korisnik sam odabire razdoblje u kojem će pregledavati svoje troškove i prihode, a to može biti na tjednoj, mjesecnoj i godišnjoj razini. Pritom su mu vidljivi opis, datum, iznos i kategorija troška odnosno prihoda. Kategorije troškova i prihoda stvara korisnik ovisno o osobnim potrebama i željama. Osim pregleda troškova i prihoda prikazivao bi se i pregled aktualnih štednji ako ih korisnik ima kreirane. Tada bi korisnik imao mogućnost pregleda osnovnih informacija štednje poput ciljanog iznosa koji si je zadao, trenutne količine ušteđenog novca i razdoblja u kojem taj ciljani iznos želi dosegnuti.

Odabirom na neku od ponuđenih opcija u glavnom izborniku korisnik može otvoriti jednu od ponuđenih stranica: njegova trenutna grupa, pozivi u grupe i štednje te korisničke postavke.

Korisnik ima pristup svim unesenim prihodima u željenom vremenskom razdoblju. Svaki od tih prihoda korisnik može izbrisati ili izmijeniti podatke o njemu. Također, korisnik može dodati novi prihod specificirajući opis, iznos, datum i kategoriju prihoda. Unošenjem novog prihoda, sustav automatski uvećava iznos trenutnog stanja računa za uneseni iznos prihoda. Analogno prihodima, korisnik može pregledati svoje dosadašnje troškove ovisno o vremenu kada su oni nastali. Svi troškovi se mogu izmijeniti i obrisati, a omogućeno je i dodavanje novih troškova. Za svaki trošak je potrebno unijeti opis, iznos, datum i kategoriju troška. Uneseni iznos troška se oduzima od trenutnog stanja računa. Korisnik može stvoriti novu štednju za koju će definirati razdoblje, ciljani iznos, opis i razlog. Stvaranjem nove štednje korisnik u štednju može uplaćivati novac te mu je omogućeno i uzimanje novca iz štednje. U svakom trenutku je moguće vidjeti trenutni iznos uplaćen u štednju. Postoji mogućnost stvaranja zajedničke štednje više korisnika.

Korisnik će imati mogućnost unosa, izmjene i brisanja kategorije troška. Kategorija troška predstavlja grupu u koju se svrstavaju slični troškovi i svaki će korisnik imati priliku prilagoditi za sebe i svoju grupu. Neke od kategorija mogu biti druženje (troškovi odlazaka u kino, na kavu itd.), režije (troškovi plaćanja struje, vode, plina itd.), kredit (troškovi plaćanja rate kredita), stanarina i slično. Također, korisnik će moći definirati i svoje kategorije prihoda, kao što su npr. mjesecna plaća, povrat poreza, bonusi, džeparac i slično. Sve kategorije će biti moguće izmijeniti i obrisati.

Korisnik može osnovati ili pristupiti u grupu u kojoj korisnici dijele troškove, prihode i sve relevantne postavke te stanje računa u grupi automatski postaje stanje računa svih korisnika. Ovakva grupa simulira kućanstvo u kojem svi članovi doprinose svojim prihodima te koriste i dijele zajedničke resurse. Administrator može pozivati članove u grupu, davati administratorske ovlasti drugim članovima grupe te obrisati grupu. Korisnik može i van grupe imati zajedničku štednju s drugim korisnicima.

Korisnika aplikacije bi se upozoravalo ako troškovi odmiču od prosjeka ili ako je zabilježena rastrošnost i savjetovalo bi ga se na koji način može više uštedjeti,

ovisno o korisničkim postavkama i preferencijama. Ako korisnički troškovi nadilaze prihode te stanje korisničkog budžeta bude „u minusu“, aplikacije će upozoriti korisnika na njegovo financijsko stanje i aktivne štednje. U aplikaciji bi bilo moguće uređivanje postavki upozoravanja i savjetovanja.

Na kraju svakog korištenja aplikacije, korisnik se može odjaviti iz sustava.

3. Pojmovnik

- CSS (Cascading Style Sheets)
 - Jezik koji služi za oblikovanje web-stranica
 - Koristi se za određivanje stila prikazivanja elemenata u jezicima poput HTML-a i XML-a
- Git
 - Raspodijeljeni sustav za upravljanje revizijama izvornog koda
- Gitlab
 - Web-bazirani upravitelj Git repozitorija
- HTML (HyperText Markup Language)
 - Prezentacijski jezik za izradu i prikaz web stranica
- Java
 - Objektno orientirani programski jezik
- Java Spring Framework
 - Okvir (eng. framework) za Java aplikacije čija je uloga izbacivanje ovisnosti iz programskog koda
- PostgreSQL
 - Objektno-relacijski sustav za upravljanje bazama podataka
- React
 - Biblioteka jezika JavaScript korištena za razvoj korisničkog sučelja
- UML (Unified Modeling Language)
 - Standardizirani jezik za modeliranje u programskom inženjerstvu
 - Koristi se za vizualizaciju, specifikaciju i dokumentaciju sustava programske potpore

4. Funkcionalni zahtjevi

4.1 Dionici

- Neregistrirani korisnici
- Registrirani korisnici
- Prijavljeni korisnici

4.2 Aktori i njihovi funkcionalni zahtjevi

- Neregistrirani korisnik, inicijator
 - Može se registrirati na sustav
- Registrirani korisnik, inicijator
 - Može se prijaviti na sustav
- Prijavljeni korisnici, inicijator
 - Pregled svog profila – trenutno stanje računa, pregled unesenih prihoda, troškova, štednji, notifikacija
 - Unos novih prihoda, troškova i štednji
 - Brisanje prihoda, troškova i štednji
 - Izmjena podataka u pojedinim prihodima, troškovima i štednjama
 - Unos novca u postojeće štednje
 - Stvaranje grupe korisnika i pozivanje drugih korisnika u grupu
 - Ulazak u postojeću grupu korisnika
 - Može postati administrator grupe
 - Izlazak iz grupe korisnika
 - Pregled profila grupe – trenutno stanje budžeta grupe, prihodi i troškovi članova grupe, zajedničke štednje
 - Dobiva upozorenja i savjete ovisno o svojim troškovima
 - Odabir postavki upozorenja i savjetovanja
 - Stvaranje kategorija troškova
 - Može se odjaviti
- Baza podataka, sudionik
 - Pohranjuje sve podatke o korisnicima
 - Pohranjuje podatke o grupama korisnika

- Pohranjuje podatke o prihodima, troškovima i štednjama korisnika
- Pohranjuje podatke o korisničkim postavkama

4.3 Opis obrazaca uporabe

4.3.1 UC01: Registracija korisnika

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvaranje korisničkog računa za pristup sustavu
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik ima funkcionalnu e-mail adresu te se nalazi na stranici registracije novih korisnika
- **Rezultat:** Stvoren korisnički račun
- **Željeni scenarij:**
 1. Korisnik ispravno unosi sve potrebne podatke za stvaranje korisničkog računa
 2. Korisnik potvrđuje unesene podatke
 3. Poslužitelj upisuje podatke u bazu podataka
 4. Poslužitelj šalje verifikacijski e-mail na korisnikovu e-mail adresu
 5. Korisnik verificira svoj račun potvrdom verifikacijskog e-maila te se potvrda proslijeđuje poslužitelju
 6. Poslužitelj šalje podatke o uspješnoj verifikaciji bazi podataka
 7. Podaci o uspješnoj verifikaciji se upisuju u bazu podataka
 8. Poslužitelj omogućuje korisniku prijavu na sustav
- **Mogući drugi scenarij:**
 1. Korisnik unosi neispravne podatke
 2. Sustav upozorava na neispravne podatke i omogućuje korisniku ispravak unosa

4.3.2 UC02: Prijava korisnika

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pristup korisnika funkcionalnostima sustava
- **Sudionici:** Poslužitelj, baza podataka

- **Preduvjeti:** Aktiviran korisnički račun te se korisnik nalazi na stranici prijave korisnika u sustav
- **Rezultat:** Pristup korisnika funkcionalnostima sustava
- **Željeni scenarij:**
 1. Korisnik unosi potrebne podatke za prijavu
 2. Korisnik potvrđuje uneseno
 3. Poslužitelj dohvaća korisničke podatke iz baze podataka te provjerava ispravnost unesenih podataka
 4. Poslužitelj omogućuje korisniku pristup funkcionalnostima aplikacije i podacima vezanima uz korisnički račun
- **Mogući drugi scenarij:**
 1. Korisnik unosi potrebne podatke za prijavu
 2. Korisnik potvrđuje uneseno
 3. Poslužitelj dohvaća korisničke podatke iz baze podataka te provjerava ispravnost unesenih podataka
 4. Poslužitelj utvrđuje neispravnost unesenih podataka
 5. Korisniku se omogućuje ponovni unos podataka

4.3.3 UC03: Dodaj trošak

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Dodavanje novog troška korisnika
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je priavljen na sustav te je otvorio stranicu troškova
- **Rezultat:** Uspješno dodavanje troška
- **Željeni scenarij:**
 1. Korisnik odabire opciju dodavanja troška
 2. Korisnik unosi željene parametre i upisuje iznos troška
 3. Korisnik potvrđuje odabrane parametre
 4. Poslužitelj provjerava ispravnost podataka
 5. Poslužitelj upisuje podatke u bazu podataka
 6. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**

1. Korisnik odabire opciju dodavanja troška
2. Korisnik unosi željene parametre i upisuje iznos troška
3. Korisnik potvrđuje odabrane parametre
4. Poslužitelj provjerom utvrđuje neispravnost podataka
5. Poslužitelj obavještava korisnika o neispravnom unosu i omogućuje mu ponovni unos podataka

4.3.4 UC04: Izmjeni trošak

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Izmjena parametara određenog troška korisnika
- **Sudionici:** Poslužitelj, baza podataka
- **Predvjeti:** Korisnik je prijavljen na sustav, otvorio je stranicu troškova te mora postojati barem jedan njegov uneseni trošak
- **Rezultat:** Izmijenjeni željeni parametri određenog troška
- **Željeni scenarij:**
 1. Korisnik odabire opciju izmjene postojećeg troška
 2. Poslužitelj dohvaća podatke iz baze podataka te ih prikazuje korisniku
 3. Korisnik mijenja parametre troška
 4. Korisnik potvrđuje promjene
 5. Poslužitelj provjera ispravnost podataka
 6. Poslužitelj upisuje podatke u bazu podataka
 7. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju izmjene troška
 2. Korisnik mijenja parametre troška
 3. Korisnik potvrđuje odabrane parametre
 4. Poslužitelj provjerom utvrđuje neispravnost podataka
 5. Poslužitelj obavještava korisnika o neispravnom unosu i omogućuje mu ponovnu izmjenu parametara

4.3.5 UC05: *Obriši trošak*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Brisanje odabranog troška
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav, nalazi se na stranici troškova te mora postojati barem jedan njegov unesen trošak
- **Rezultat:** Dolazi do brisanja odabranog troška
- **Željeni scenarij:**
 1. Korisnik odabire opciju brisanja željenog troška
 2. Korisnik potvrđuje svoj odabir
 3. Poslužitelj briše podatke o trošku iz baze podataka
 4. Baza podataka obavještava poslužitelja o uspješnosti brisanja troška
- **Mogući drugi scenarij:** -

4.3.6 UC06: *Dodaj prihod*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Dodavanje prihoda
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav te je otvorio stranicu prihoda
- **Rezultat:** Uspješno dodavanje prihoda
- **Željeni scenarij:**
 1. Korisnik odabire opciju dodavanja prihoda
 2. Korisnik odabire željene parametre i upisuje iznos prihoda
 3. Korisnik potvrđuje odabrane parametre
 4. Poslužitelj utvrđuje ispravnost podataka
 5. Poslužitelj upisuje podatke u bazu podataka
 6. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju dodavanja prihoda
 2. Korisnik unosi željene parametre i upisuje iznos prihoda
 3. Korisnik potvrđuje odabrane parametre

4. Poslužitelj provjerom utvrđuje neispravnost podataka
5. Poslužitelj obavještava korisnika o neispravnom unosu i omogućuje mu ponovni unos

4.3.7 UC07: Izmjeni prihod

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Izmjena parametara određenog prihoda korisnika
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav, otvorio je stranicu prihoda te mora postojati barem jedan unesen prihod
- **Rezultat:** Izmijenjeni željeni parametri određenog prihoda
- **Željeni scenarij:**
 1. Korisnik odabire opciju izmjene postojećeg prihoda
 2. Korisnik mijenja parametre prihoda
 3. Korisnik potvrđuje promjene
 4. Poslužitelj provjera ispravnost podataka
 5. Poslužitelj upisuje podatke u bazu podataka
 6. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju izmjene prihoda
 2. Korisnik mijenja parametre prihoda
 3. Korisnik potvrđuje odabrane parametre
 4. Poslužitelj provjerom utvrđuje neispravnost podataka
 5. Poslužitelj obavještava korisnika o neispravnom unosu i omogućuje mu ponovnu izmjenu parametara

4.3.8 UC08: Obriši prihod

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Brisanje odabranog prihoda
- **Sudionici:** Poslužitelj, baza podataka

- **Preduvjeti:** Korisnik je prijavljen na sustav, otvorio je stranicu prihoda te mora postojati barem jedan unesenih prihoda
- **Rezultat:** Dolazi do brisanja odabranog prihoda
- **Željeni scenarij:**
 1. Korisnik odabire opciju brisanja željenog prihoda
 2. Korisnik potvrđuje svoj odabir
 3. Poslužitelj briše podatke o prihodu iz baze podataka
 4. Baza podataka obavještava poslužitelja o uspješnosti brisanja prihoda
- **Mogući drugi scenarij:** -

4.3.9 UC09: Stvori štednju

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Stvaranje štednje
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav te je otvorio stranicu štednji
- **Rezultat:** Uspješno stvaranje štednje
- **Željeni scenarij:**
 1. Korisnik odabire opciju stvaranja štednje
 2. Korisnik odabire parametre štednje
 3. Korisnik potvrđuje odabrane parametre
 4. Poslužitelj provjerava ispravnost podataka
 5. Poslužitelj upisuje podatke u bazu podataka
 6. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju stvaranja štednje
 2. Korisnik odabire parametre štednje
 3. Korisnik potvrđuje odabrane parametre
 4. Poslužitelj provjerom utvrđuje neispravnost unesenih podataka, obavještava korisnika i omogućuje mu ponovni unos

4.3.10 UC10: Izmjeni štednju

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Izmjena parametara određene štednje korisnika
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav, otvorio je stranicu štednji te mora postojati barem jedna unesena štednja
- **Rezultat:** Izmijenjeni željeni parametri određene štednje
- **Željeni scenarij:**
 1. Korisnik odabire opciju izmjene postojeće štednje
 2. Korisnik mijenja parametre štednje
 3. Korisnik potvrđuje promjene
 4. Poslužitelj provjerava izmijenjene podatka
 5. Poslužitelj upisuje podatke u bazu podataka
 6. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju izmjene parametara štednje
 2. Korisnik odabire parametre štednje
 3. Korisnik potvrđuje odabrane parametre
 4. Poslužitelj provjerom utvrđuje neispravnost podataka, obavještava korisnika te omogućuje ponovnu izmjenu podataka

4.3.11 UC11: Obriši štednju

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Brisanje odabrane štednje
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav, otvorio je stranicu štednji te mora postojati barem jedna unesena štednja
- **Rezultat:** Dolazi do brisanja odabrane štednje
- **Željeni scenarij:**
 1. Korisnik odabire opciju brisanja željene štednje
 2. Korisnik potvrđuje brisanje štednje

3. Poslužitelj briše podatke o štednji iz baze podataka
 4. Baza podataka obavještava poslužitelja o uspješnosti brisanja štednje
- **Mogući drugi scenarij:** -

4.3.12 UC12: Dodaj novac u štednju

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Povećanje iznosa novca u štednji za željeni iznos
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav, otvorio je stranicu štednji te je barem jedna štednja dostupna
- **Rezultat:** Povećanje iznosa novca u štednji za željeni iznos
- **Željeni scenarij:**
 1. Korisnik odabire opciju dodavanja novca u željenu štednju
 2. Korisnik unosi željeni iznos novca
 3. Korisnik potvrđuje unos
 4. Poslužitelj provjerava ispravnost podataka
 5. Poslužitelj šalje podatke bazi podataka
 6. Promjene se upisuju u bazu podataka
 7. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju dodavanja novca u željenu štednju
 2. Korisnik unosi željeni iznos novca
 3. Korisnik potvrđuje unos
 4. Poslužitelj utvrđuje neispravnost unesenih podataka
 5. Poslužitelj javlja neispravnost unesenih podataka korisniku i omogućuje mu ponovni unos podataka

4.3.13 UC13: Uzmi novac iz štednje

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Smanjenje iznosa novca u štednji za željeni iznos

- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav, otvorio je stranicu štednji te je barem jedna štednja dostupna
- **Rezultat:** Smanjenje iznosa novca u štednji za željeni iznos te povećanje stanja računa korisnika
- **Željeni scenarij:**
 1. Korisnik odabire opciju uzimanja novca iz željene štednje
 2. Korisnik unosi željeni iznos novca
 3. Korisnik potvrđuje unos
 4. Poslužitelj dohvaća podatke o štednji iz baze podataka
 5. Poslužitelj potvrđuje ispravnost podataka
 6. Poslužitelj upisuje podatke u bazu podataka
 7. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju uzimanja novca iz željene štednje
 2. Korisnik unosi željeni iznos
 3. Korisnik potvrđuje unos
 4. Poslužitelj dohvaća podatke o štednji iz baze podataka
 5. Poslužitelj utvrđuje da je količina ušteđenog novca manja od željenog iznosa koji korisnik želi uzeti
 6. Poslužitelj javlja pogrešku korisniku i omogućuje mu ponovni unos željene sume novca

4.3.14 *UC14: Stvori grupu*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Stvaranje grupe korisnika koji međusobno dijele resurse
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je registriran u aplikaciji te je otvorio stranicu grupa
- **Rezultat:** Stvorena grupa
- **Željeni scenarij:**
 1. Korisnik odabire opciju za stvaranje nove grupe
 2. Korisnik unosi sve potrebne podatke

3. Korisnik potvrđuje unos
 4. Poslužitelj provjerava ispravnost podataka
 5. Poslužitelj u bazu podataka upisuje podatke o novoj grupi
 6. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju za stvaranje nove grupe
 2. Korisnik unosi sve potrebne podatke
 3. Korisnik potvrđuje unos
 4. Poslužitelj utvrđuje neispravnost unesenih podataka
 5. Poslužitelj obavještava korisnika o neuspjelom upisu i omogućuje mu ponovni unos podataka

4.3.15 UC15: Pozovi korisnika u grupu

- **Glavni sudionik:** Prijavljeni korisnik (administrator grupe)
- **Cilj:** Pozivanje korisnika u prethodno kreiranu grupu
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Grupa je kreirana, korisnik ima ulogu administratora te je otvorio stranicu grupnih postavki
- **Rezultat:** Poziv poslan novom korisniku
- **Željeni scenarij:**
 1. Korisnik (administrator) odabire opciju za pozivanje novog korisnika u grupu
 2. Korisnik (administrator) unosi korisničko ime pozvanog korisnika
 3. Korisnik (administrator) potvrđuje unos
 4. Poslužitelj provjerava ispravnost podataka
 5. Poslužitelj upisuje podatke o pozivu u bazu podataka
 6. Baza podataka potvrđuje poslužitelju uspješan upis podataka
 7. Poslužitelj šalje korisniku poziv u grupu
- **Mogući drugi scenarij:**
 1. Korisnik (administrator) odabire opciju za pozivanje novog korisnika u grupu
 2. Korisnik (administrator) unosi korisničko ime korisnika

3. Korisnik (administrator) potvrđuje unos
4. Sustav utvrđuje da ne postoji korisnik tog korisničkog imena ili je član s tim korisničkim imenom već član grupe
5. Sustav javlja pogrešku pri unosu i omogućuje ponovni unos

4.3.16 ***UC16: Uredi postavke grupe***

- **Glavni sudionik:** Prijavljeni korisnik (administrator grupe)
- **Cilj:** Promjena postavki grupe i spremanje promjena
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Član grupe koji obavlja promjenu je administrator te je otvorio stranicu postavki grupe
- **Rezultat:** Postavke grupe su promijenjene
- **Željeni scenarij:**
 1. Korisnik (administrator) mijenja postavke (uređuje administratorske ovlasti članova, izbacuje članove iz grupe, uređuje kategorije prihoda, uređuje kategorije troškova)
 2. Poslužitelj provjerava ispravnost unesenih podataka
 3. Poslužitelj upisuje promjene u bazu podataka
 4. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik (administrator) mijenja postavke (uređuje administratorske ovlasti članova, izbacuje članove iz grupe, uređuje kategorije prihoda, uređuje kategorije prihoda)
 2. Poslužitelj utvrđuje neispravnost unesenih podataka
 3. Poslužitelj obavještava korisnika o neispravnom unosu i omogućuje ponovni unos podataka

4.3.17 ***UC17: Napusti grupu***

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Član grupe napušta grupu
- **Sudionici:** Poslužitelj, baza podataka

- **Preduvjeti:** Korisnik je član grupe te je otvorio stranicu grupe koju želi napustiti
- **Rezultat:** Korisnik napušta grupu
- **Željeni scenarij:**
 1. Korisnik odabire opciju za napuštanje grupe
 2. Korisnik potvrđuje napuštanje grupe
 3. Poslužitelj briše korisnika iz popisa članova grupe u bazi podataka
 4. Baza podataka obavještava poslužitelja o uspješnosti brisanje korisnika iz popisa članova grupe
- **Mogući drugi scenarij:** -

4.3.18 UC18: Odgovori na poziv u grupu

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Korisnik se pridružuje grupi
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav
- **Rezultat:** Korisnik postaje član grupe
- **Željeni scenarij:**
 1. Korisnik prima obavijest o pozivu u grupu
 2. Korisnik potvrđuje ulazak u grupu
 3. Korisnik postaje član grupe
 4. Poslužitelj upisuje promjene u bazu podataka
 5. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik prima obavijest o pozivu u grupu
 2. Korisnik odbija poziv
 3. Poslužitelj upisuje promjene u bazu podataka
 4. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka

4.3.19 ***UC19: Izbriši grupu***

- **Glavni sudionik:** Prijavljeni korisnik (administrator grupe)
- **Cilj:** Brisanje grupe
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik koji vrši brisanje je administrator grupe te je otvorio stranicu grupnih postavki
- **Rezultat:** Grupa je obrisana
- **Željeni scenarij:**
 1. Korisnik (administrator) odabire opciju za brisanje grupe
 2. Korisnik (administrator) potvrđuje brisanje grupe
 3. Poslužitelj briše grupu iz baze podataka
 4. Baza podataka obavještava poslužitelja o uspješnosti brisanja grupe
- **Mogući drugi scenarij:** -

4.3.20 ***UC20: Dodaj kategoriju troška***

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Stvoriti novu kategoriju troška
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen u sustav te je otvorio stranicu korisničkih postavki
- **Rezultat:** Nova kategorija troška je dostupna za uporabu
- **Željeni scenarij:**
 1. Korisnik odabire opciju dodavanja nove kategorije troška
 2. Korisnik unosi atribute kategorije troška
 3. Korisnik potvrđuje unos
 4. Poslužitelj zapisuje podatke u bazu podataka
 5. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju za dodavanje nove kategorije troška
 2. Korisnik unosi postojeće ime kategorije
 3. Korisnik potvrđuje unos kategorije troška

4. Poslužitelj utvrđuje neispravnost unesenih podataka te omogućuje korisniku ponovni unos podataka

4.3.21 UC21: Izmjeni kategoriju troška

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Promijeniti atribut određene kategorije troška
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav te je otvorio stranicu korisničkih postavki, barem jedna kategorija troška je dostupna
- **Rezultat:** Atributi odabrane kategorije troška imaju nove vrijednosti
- **Željeni scenarij:**
 1. Korisnik odabire opciju izmjene postojeće kategorije troška
 2. Korisnik mijenja vrijednosti atributa kategorije troška
 3. Korisnik potvrđuje izmjenu
 4. Poslužitelj zapisuje podatke u bazu podataka
 5. Baza podataka obavještava poslužitelja o uspješnosti upisa podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju za izmjenu kategorije troška
 2. Korisnik unosi postojeće ime kategorije
 3. Korisnik potvrđuje izmjenu kategorije troška
 4. Poslužitelj provjerom utvrđuje neispravnost unesenih podataka te omogućuje korisniku ponovni unos podataka

4.3.22 UC22: Obriši kategoriju troška

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Brisanje odabrane kategorije troška
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav te je otvorio stranicu korisničkih postavki
- **Rezultat:** Dolazi do brisanja odabrane kategorije troška
- **Željeni scenarij:**
 1. Korisnik odabire opciju brisanja željene kategorije troška

2. Korisnik potvrđuje brisanje kategorije troška
 3. Poslužitelj zahtjeva brisanje podataka iz baze
 4. Baza podataka dojavljuje poslužitelju uspješno brisanje podataka
- **Mogući drugi scenarij:** -

4.3.23 UC23: Podesi korisničke postavke

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Izmjena korisničkih postavki ovisno o osobnim željama korisnika
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav te je otvorio stranicu korisničkih postavki
- **Rezultat:** Napravljene promjene korisničkih postavki
- **Željeni scenarij:**
 1. Korisnik odabire opciju podešavanja korisničkih postavki
 2. Korisnik mijenja željene postavke
 3. Korisnik potvrđuje promjene
 4. Poslužitelj zapisuje podatke u bazu podataka
 5. Baza podataka dojavljuje poslužitelju uspješan unos podataka
- **Mogući scenarij:** -

4.3.24 UC24: Odjava korisnika

- **Mogući drugi scenarij:** -
- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Odjava prijavljenog korisnika
- **Sudionici:** Poslužitelj
- **Preduvjeti:** Korisnik je prijavljen na sustav
- **Rezultat:** Uspješna odjava korisnika
- **Željeni scenarij:**
 1. Korisnik odabire opciju za odjavu iz sustava
 2. Poslužitelj odjavljuje korisnika iz sustava

- **Mogući drugi scenarij:** -

4.3.25 *UC25: Promjena lozinke*

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Promjena lozinke korisnika
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav te je otvorena stranica korisničkih postavki
- **Rezultat:** Uspješna promjena lozinke
- **Željeni scenarij:**
 1. Korisnik otvara korisničke postavke
 2. Korisnik unosi trenutnu lozinku
 3. Korisnik unosi novu lozinku
 4. Korisnik unosi novu lozinku ponovno radi potvrde
 5. Korisnik potvrđuje promjene
 6. Poslužitelj provjerava poklapanje dvije novounesene lozinke te ispravnost stare lozinke u bazi podataka
 7. Baza podataka dojavljuje poslužitelju ispravnost lozinke
 8. Poslužitelj zapisuje podatke u bazu podataka
 9. Baza podataka dojavljuje poslužitelju uspješan upis podataka
- **Mogući drugi scenarij:**
 1. Korisnik otvara korisničke postavke
 2. Korisnik unosi pogrešnu trenutnu lozinku ili unosi dvije različite nove lozinke
 3. Korisnik potvrđuje promjene
 4. Poslužitelj provjerava poklapanje dvije novounesene lozinke te ispravnost stare lozinke u bazi podataka
 5. Baza podataka javlja poslužitelju neispravnost stare lozinke i/ili poslužitelj otkriva nepoklapanje dvije novounesene lozinke
 6. Poslužitelj javlja korisniku da je u pitanju pogreška

4.3.26 ***UC26: Dodaj kategoriju prihoda***

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Dodavanje nove kategorije prihoda
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav te je otvorena stranica korisničkih postavki
- **Rezultat:** Uspješno dodavanje nove kategorije prihoda
- **Željeni scenarij:**
 1. Korisnik odabire opciju za dodavanje nove kategorije prihoda
 2. Korisnik unosi ime kategorije prihoda
 3. Korisnik potvrđuje unos kategorije
 4. Poslužitelj zapisuje podatke u bazu podataka
 5. Baza podataka dojavljuje poslužitelju uspješan unos podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju za dodavanje nove kategorije prihoda
 2. Korisnik unosi postojeće ime kategorije
 3. Korisnik potvrđuje unos kategorije prihoda
 4. Poslužitelj provjerava u bazi podataka postojanje kategorije
 5. Baza podataka potvrđuje poslužitelju postojanje kategorije
 6. Poslužitelj javlja pogrešku korisniku

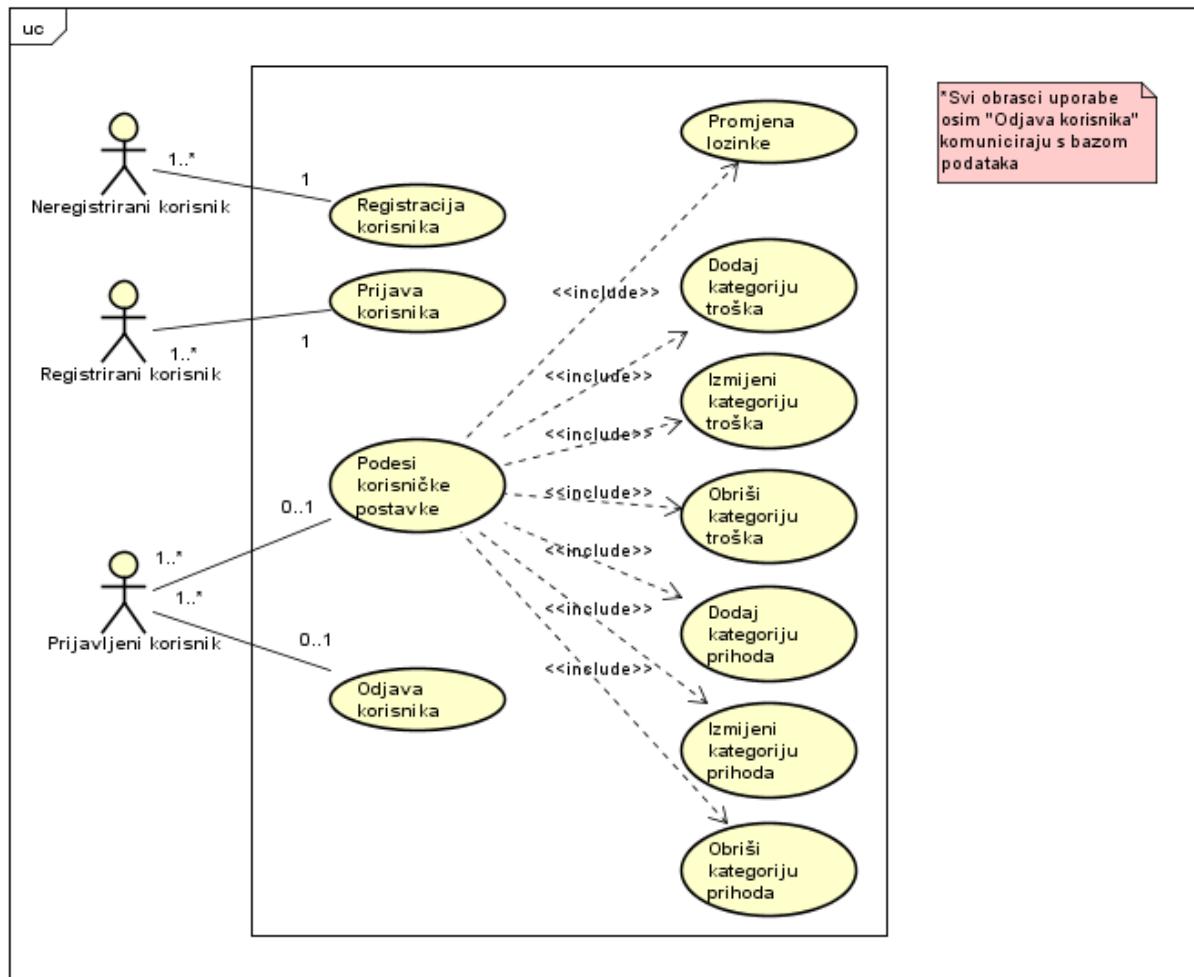
4.3.27 ***UC27: Izmjeni kategoriju prihoda***

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Izmjena kategorije prihoda
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav te postoji barem jedna kategorija prihoda, otvorena je stranica korisničkih postavki
- **Rezultat:** Uspješna izmjena kategorije prihoda
- **Željeni scenarij:**
 1. Korisnik odabire opciju za izmjenu kategorije prihoda
 2. Korisnik unosi ime kategorije prihoda
 3. Korisnik potvrđuje izmjenu kategorije prihoda

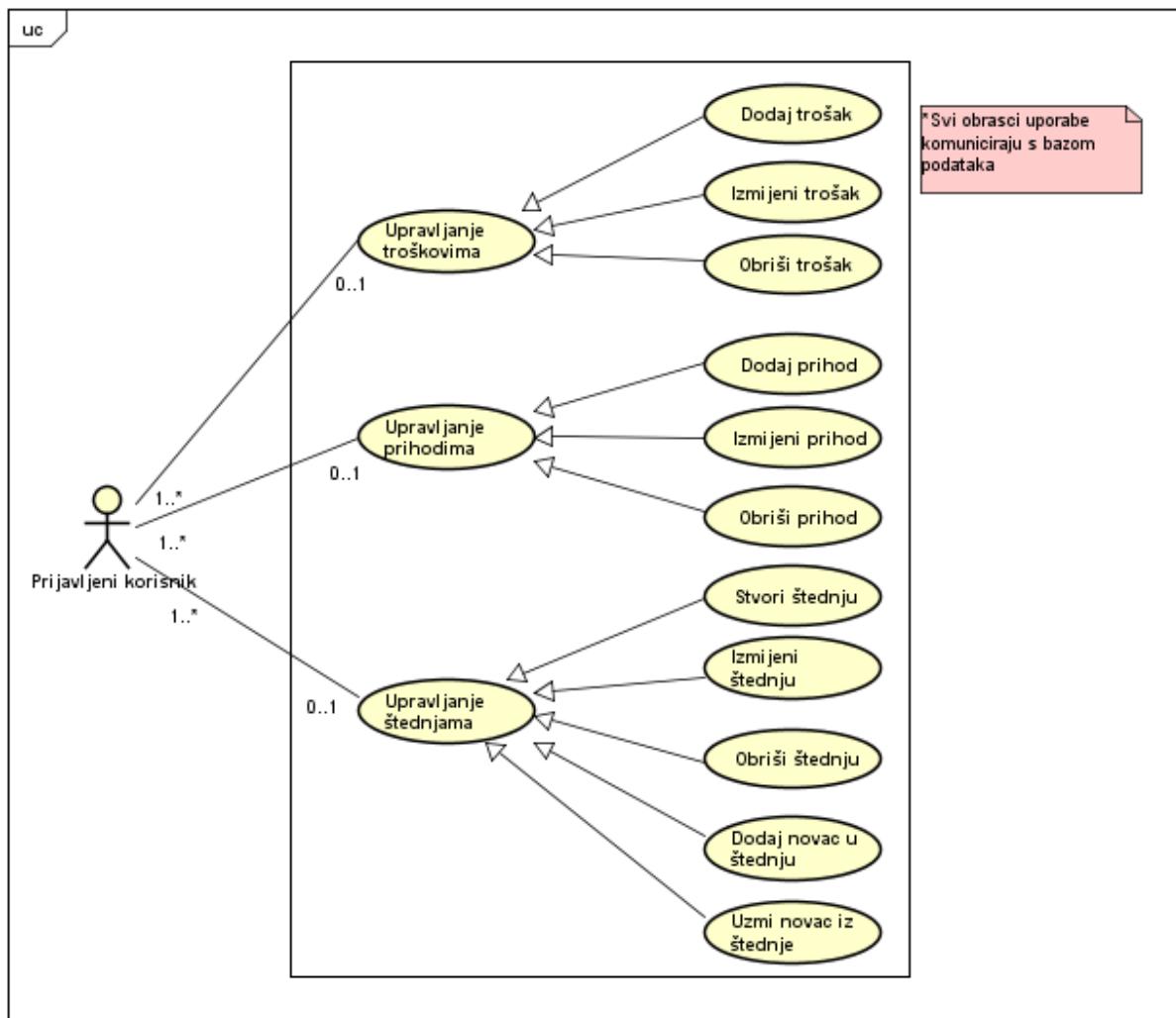
4. Poslužitelj provjerava u bazi podataka postojanje kategorije
 5. Baza podataka dojavljuje poslužitelju nepostojanje kategorije
 6. Poslužitelj zapisuje podatke u bazu podataka
 7. Baza podataka dojavljuje poslužitelju uspješan unos podataka
- **Mogući drugi scenarij:**
 1. Korisnik odabire opciju za izmjenu kategorije prihoda
 2. Korisnik unosi postojeće ime kategorije
 3. Korisnik potvrđuje izmjenu kategorije prihoda
 4. Poslužitelj provjerava u bazi podataka postojanje kategorije
 5. Baza podataka potvrđuje poslužitelju postojanje kategorije
 6. Poslužitelj javlja pogrešku korisniku

4.3.28 UC28: Obriši kategoriju prihoda

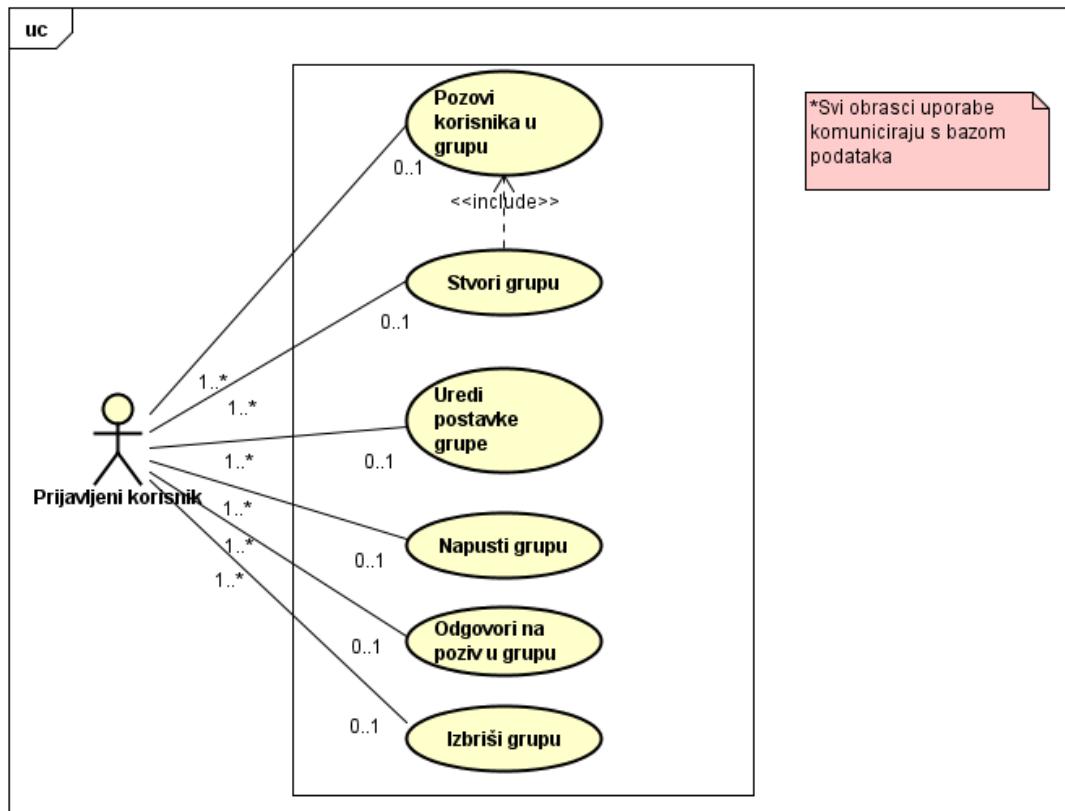
- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:** Brisanje kategorije prihoda
- **Sudionici:** Poslužitelj, baza podataka
- **Preduvjeti:** Korisnik je prijavljen na sustav te postoji barem jedna kategorija prihoda, otvorena je stranica korisničkih postavki
- **Rezultat:** Uspješno brisanje kategorije prihoda
- **Željeni scenarij:**
 1. Korisnik odabire opciju za brisanje kategorije prihoda
 2. Korisnik potvrđuje brisanje kategorije prihoda
 3. Poslužitelj zahtjeva brisanje podataka iz baze
 4. Baza podataka dojavljuje poslužitelju uspješno brisanje podataka
- **Mogući drugi scenarij:** -



Slika 4.3.1 Dijagram obrazaca uporabe za neregistriranog, registriranog i prijavljenog korisnika
 (UC01, UC02, UC20, UC21, UC22, UC23, UC24, UC25, UC26, UC27, UC28)



Slika 4.3.2 Dijagram obrazaca uporabe za prijavljenog korisnika – upravljanje troškovima, prihodima i štednjama (UC03, UC04, UC05, UC06, UC07, UC08, UC09, UC10, UC11, UC12, UC13)

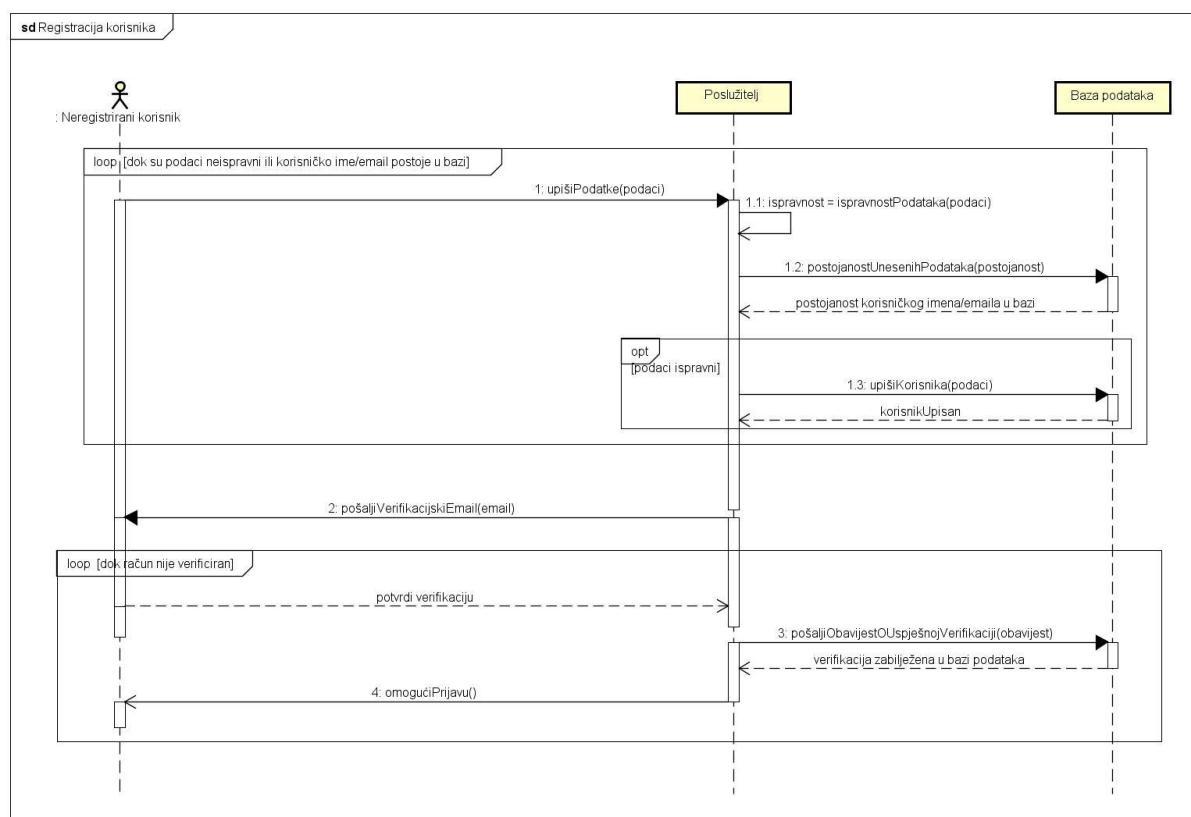


Slika 4.3.3 Dijagram obrazaca uporabe za prijavljenog korisnika – grupe (UC14, UC15, UC16, UC17, UC18, UC19)

4.4 Sekvencijski dijagrami

4.4.1 UC01: Registracija korisnika

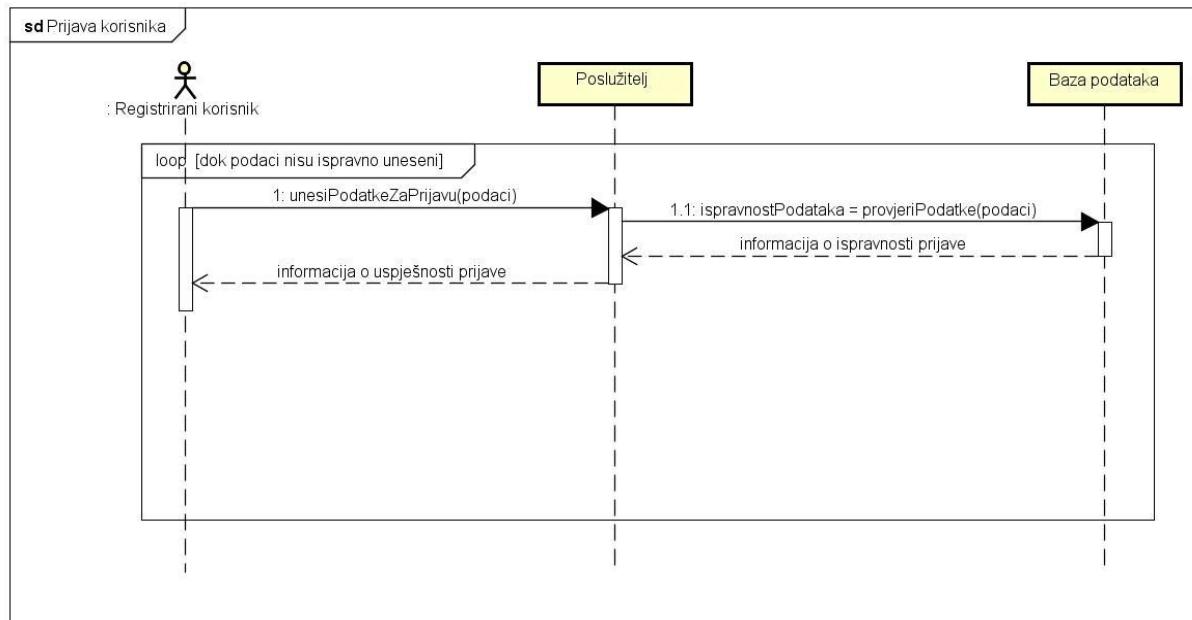
Neregistrirani korisnik želi napraviti račun. Nakon što pristupi stranici za registraciju ispunjava obrazac unoseći korisničko ime, e-mail adresu na koju mu se šalje poveznica za potvrdu registracije i željenu lozinku. Poslužitelj provjerava ispravnost unesenih podataka. U slučaju da u bazi podataka već postoji uneseno korisničko ime ili e-mail adresa, račun neće biti kreiran, u suprotnom se račun uspješno stvara i korisnički podaci upisuju se u bazu podataka. Registrirani korisnik dobiva verifikacijski e-mail na svoju e-mail adresu kako bi mogao pristupiti svom računu. Nakon što korisnik otvorí poveznicu u verifikacijskom e-mailu, poslužitelj javlja bazi da je račun verificiran te je račun spreman za korištenje.



Slika 4.4.1 Sekvencijski dijagram obrasca Registracija korisnika

4.4.2 UC02: Prijava korisnika

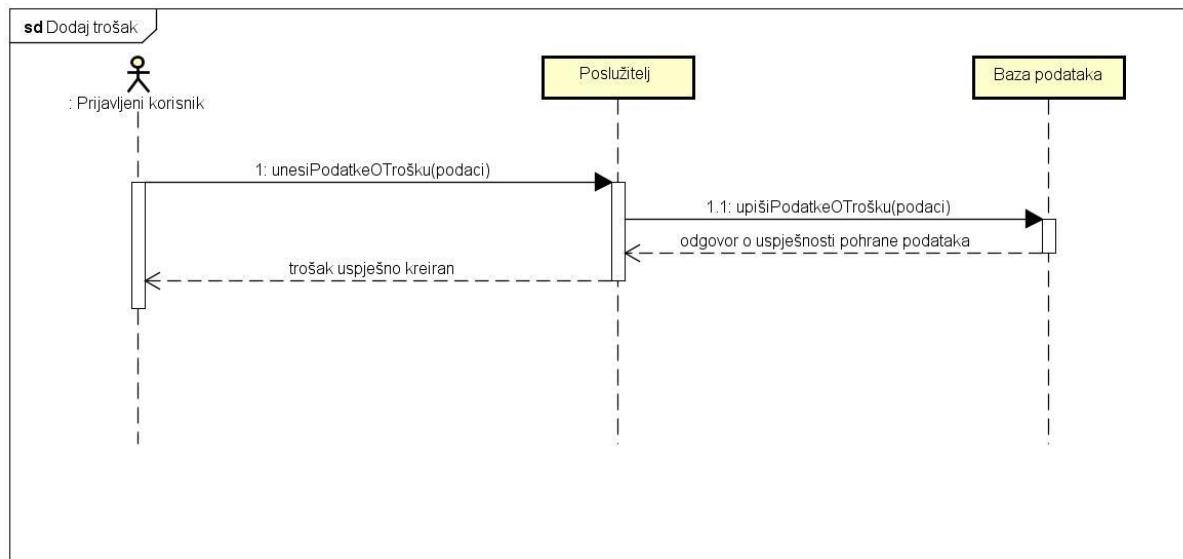
Korisnik prilikom prijave unosi svoje korisničko ime i pripadajuću lozinku. Poslužitelj komunicira s bazom kako bi ispitao ispravnost podataka. U slučaju neispravnosti, korisnik je dužan ponovno unijeti podatke, u suprotnom korisnik uspješno izvršava prijavu.



Slika 4.4.2 Sekvencijski dijagram obrasca Prijava korisnika

4.4.3 UC03: Dodaj trošak

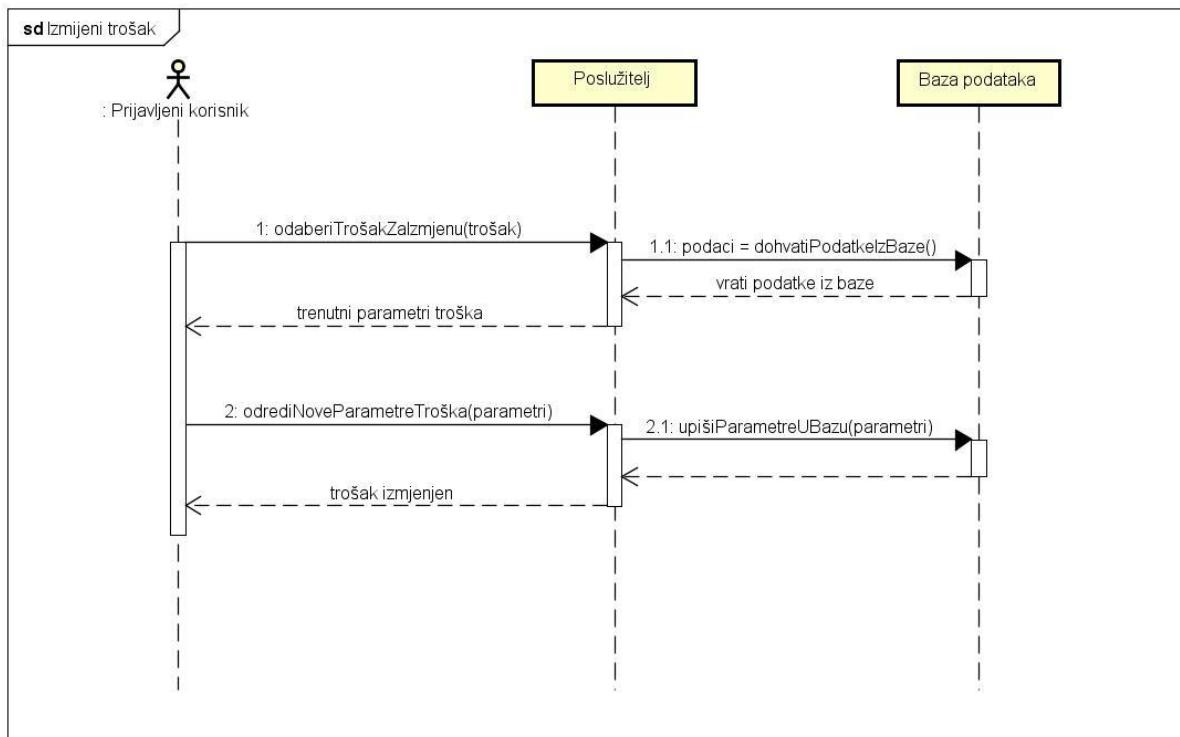
Korisnik dodaje trošak tako što odabire parametre troška te upisuje iznos troška. Komunikaciju između poslužitelja i baze podataka predstavlja upisivanje podataka o trošku u bazu te povratna informacija o uspješnosti pohrane.



Slika 4.4.3 Sekvencijski dijagram obrasca Dodaj trošak

4.4.4 UC04: Izmjeni trošak

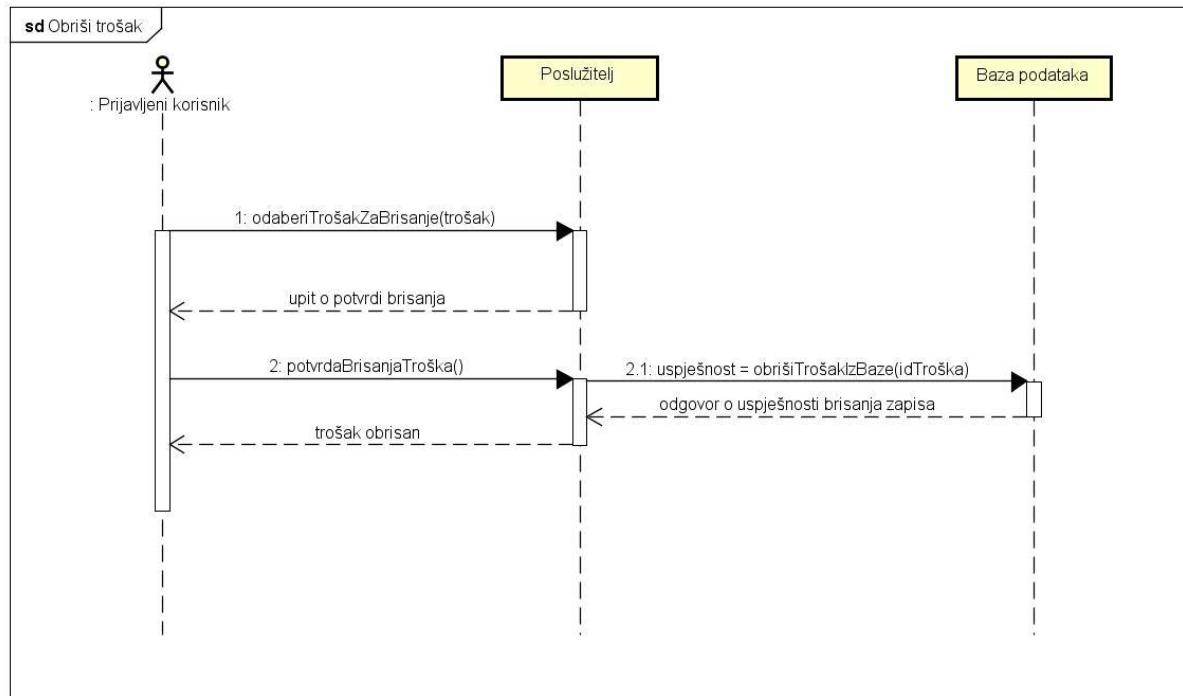
Korisnik mijenja parametre troška tako što odabire nove ili upisuje novi iznos troška. Komunikaciju između poslužitelja i baze podataka predstavlja dohvaćanje zapisa troška iz baze podataka, upisivanje novih podataka o trošku u bazu te povratna informacija o uspjehu.



Slika 4.4.4 Sekvenčni dijagram obrasca Izmjeni trošak

4.4.5 UC05: Obriši trošak

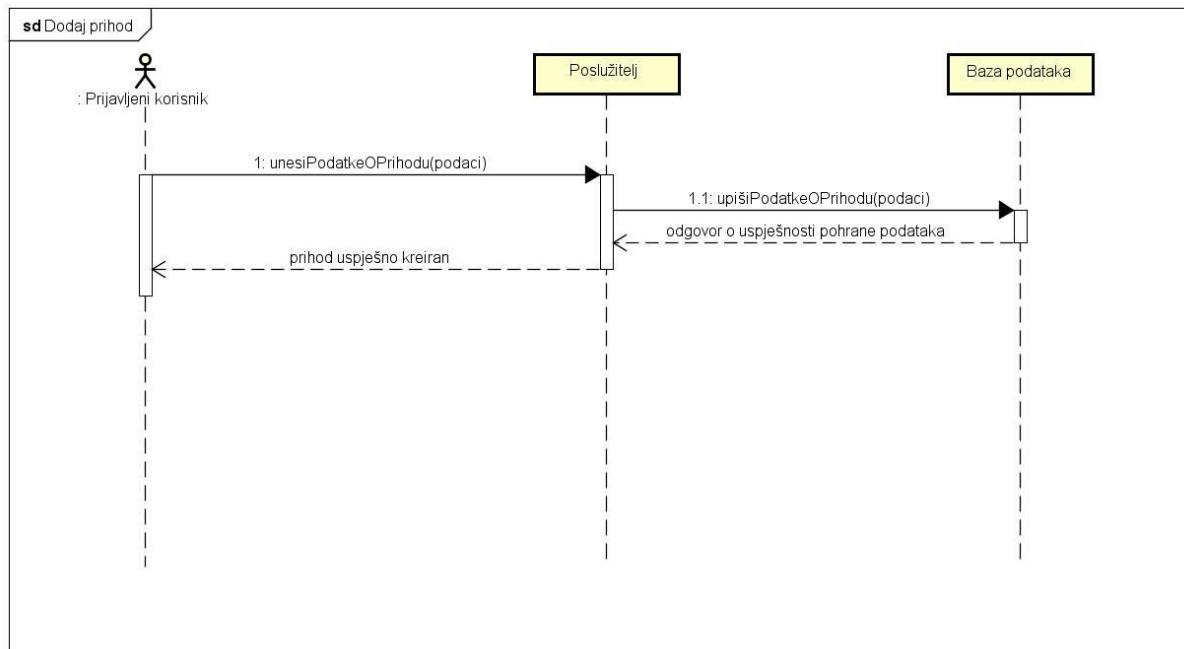
Korisnik briše postojeći trošak tako što ga označava i odabire opciju brisanja. Komunikaciju između poslužitelja i baze podataka predstavlja brisanje zapisa troška iz baze te povratna informacija o uspješnosti brisanja.



Slika 4.4.5 Sekvencijski dijagram obrasca Obriši trošak

4.4.6 UC06: Dodaj prihod

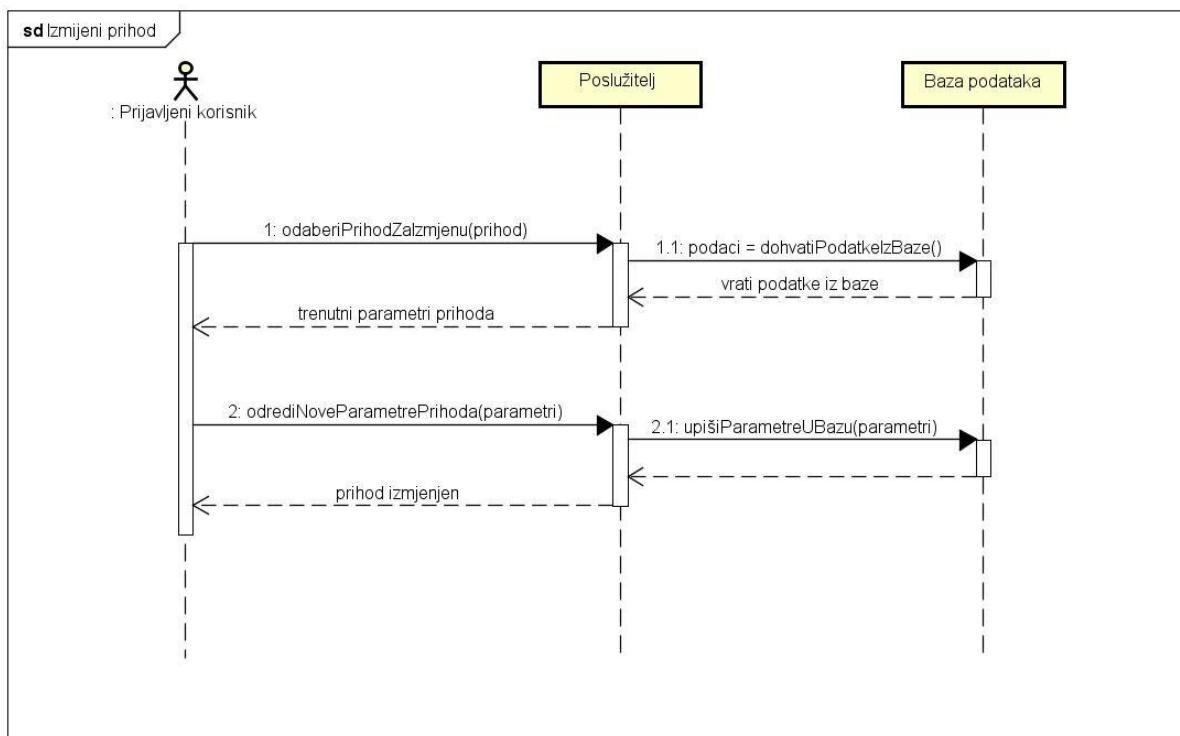
Korisnik dodaje prihod tako što odabire parametre prihoda te upisuje iznos prihoda. Komunikaciju između poslužitelja i baze podataka predstavlja upisivanje podataka o prihodu u bazu te povratna informacija o uspješnosti pohrane.



Slika 4.4.6 Sekvencijski dijagram obrasca Dodaj prihod

4.4.7 UC07: Izmjeni prihod

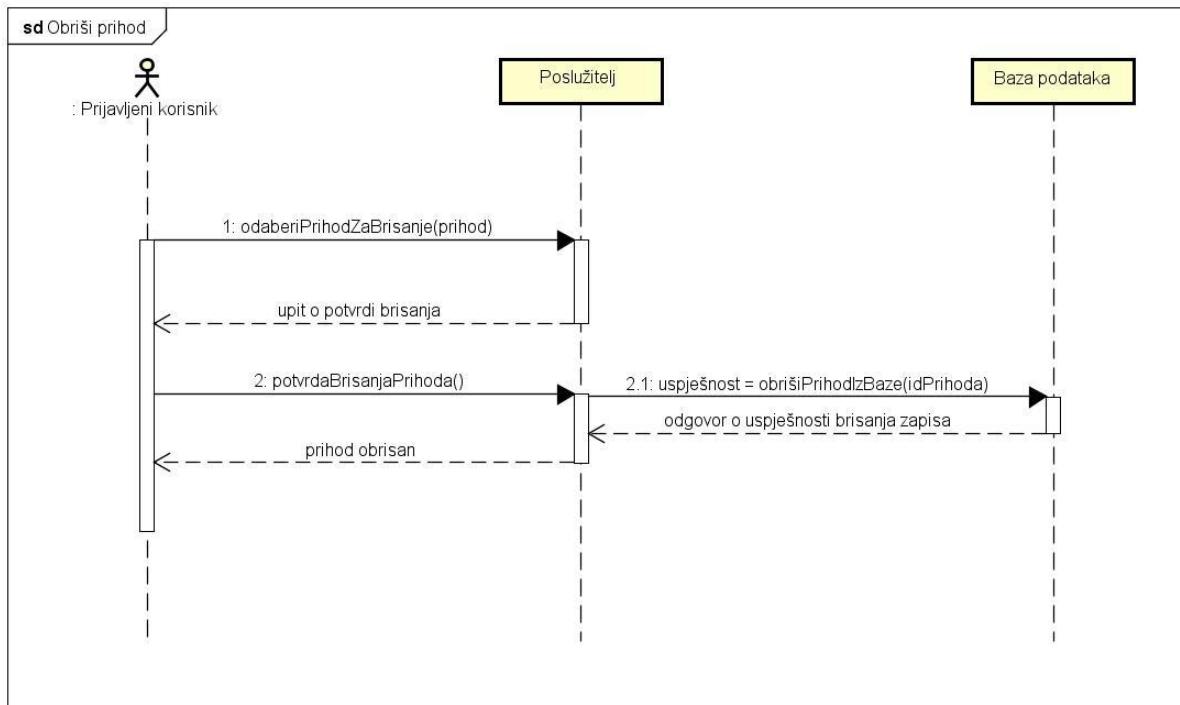
Korisnik mijenja parametre prihoda tako što odabire nove ili upisuje novi iznos prihoda. Komunikaciju između poslužitelja i baze podataka predstavlja dohvaćanje zapisa prihoda iz baze podataka, upisivanje novih podataka o prihodu u bazu te povratna informacija o uspješnosti pohrane.



Slika 4.4.7 Sekvencijski dijagram obrasca Izmjeni prihod

4.4.8 UC08: Obriši prihod

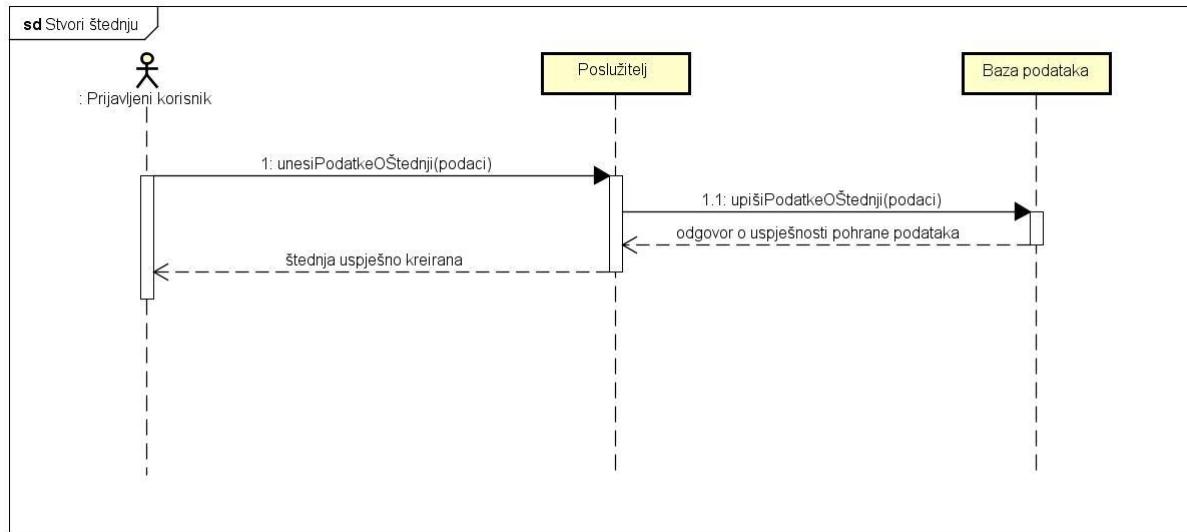
Korisnik briše postojeći prihod tako što odabire opciju brisanja. Komunikaciju između poslužitelja i baze podataka predstavlja brisanje zapisa prihoda iz baze te povratna informacija o uspješnosti brisanja.



Slika 4.4.8 Sekvencijski dijagram obrasca Obriši prihod

4.4.9 UC09: Stvori štednju

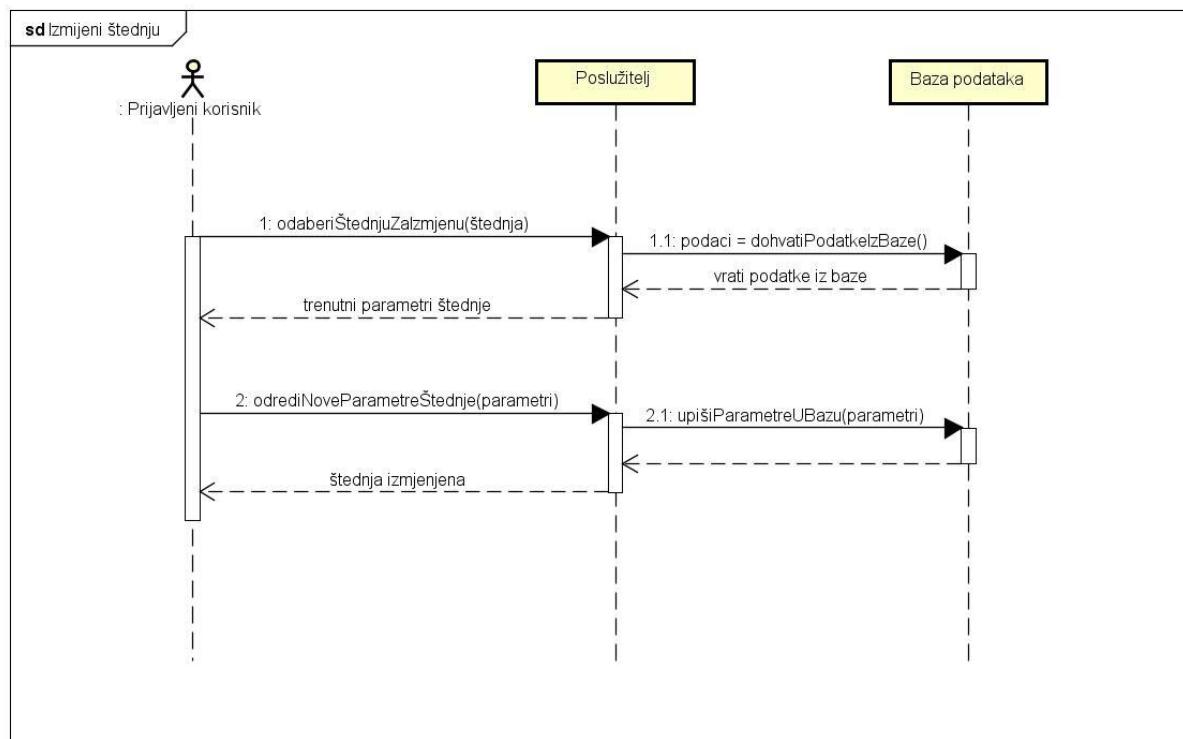
Korisnik stvara štednju tako što odabire parametre štednje te upisuje ciljni iznos štednje. Komunikaciju između poslužitelja i baze podataka predstavlja upisivanje podataka o štednji u bazu te povratna informacija o uspješnosti pohrane.



Slika 4.4.9 Sekvencijski dijagram obrasca Stvori štednju

4.4.10 UC10: Izmjeni štednju

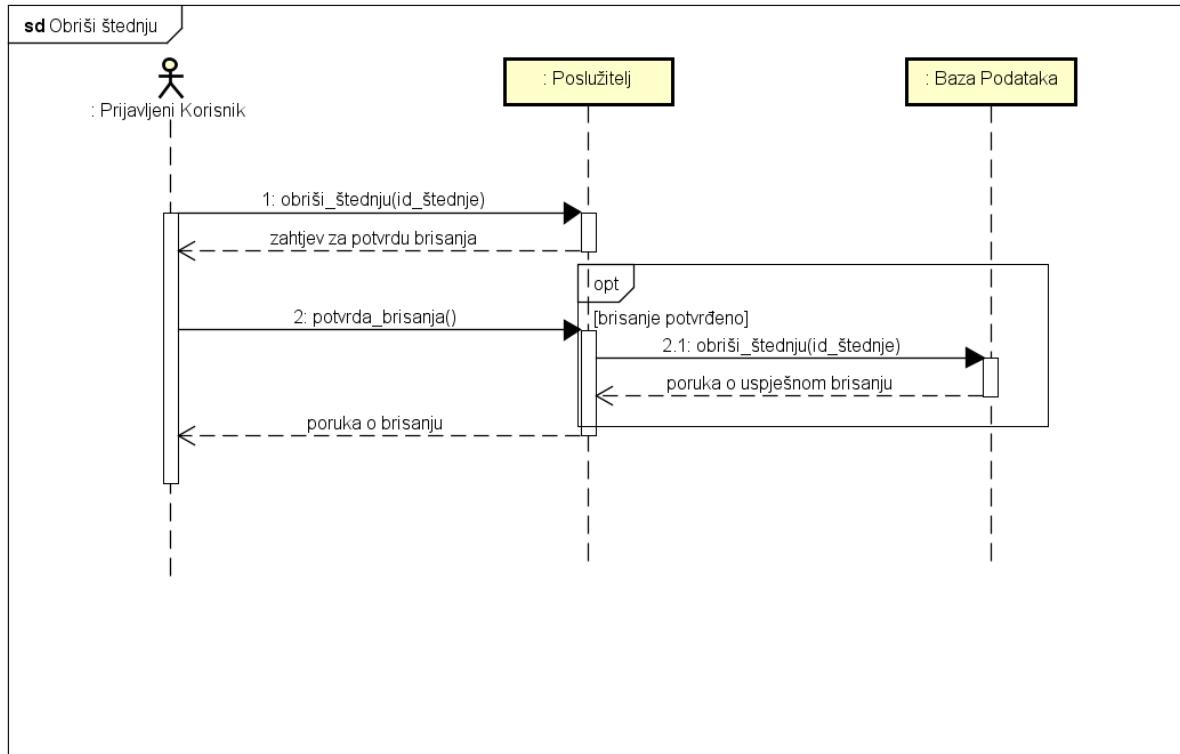
Korisnik mijenja parametre štednje tako što odabire nove ili upisuje novi ciljni iznos štednje. Komunikaciju između poslužitelja i baze podataka predstavlja dohvaćanje zapisa štednje iz baze podataka, upisivanje novih podataka o štednji u bazu te povratna informacija o uspješnosti pohrane.



Slika 4.4.10 Sekvencijski dijagram obrasca Izmjeni štednju

4.4.11 UC11: Obriši štednju

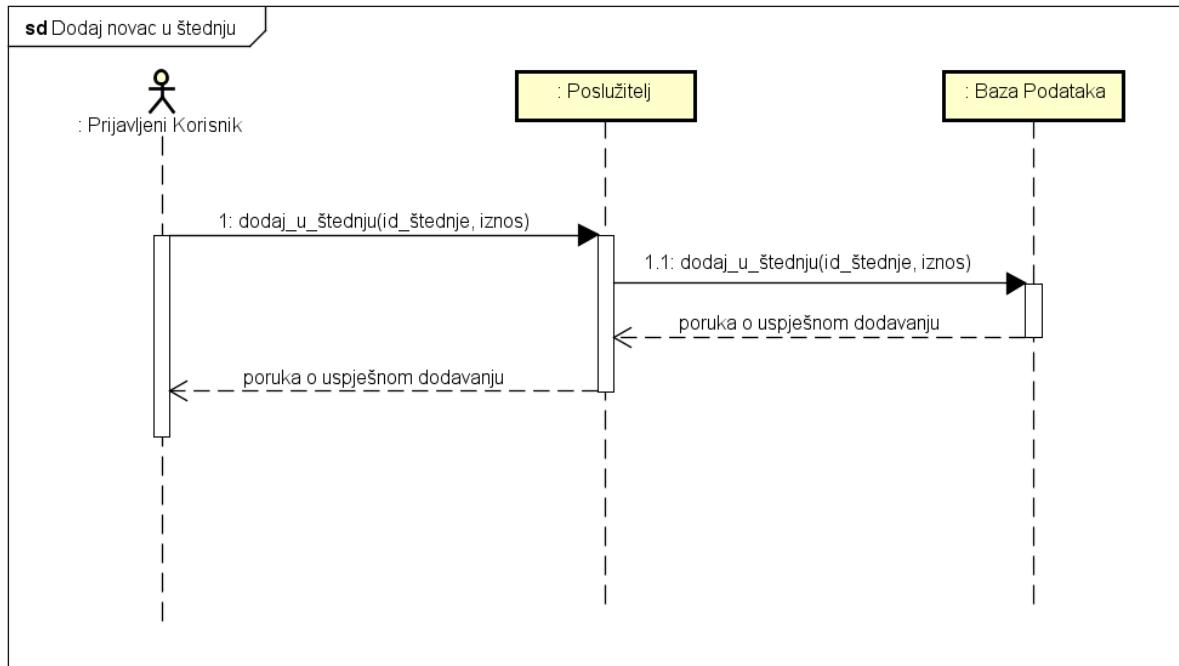
Prijavljeni korisnik šalje zahtjev za brisanje štednje. Poslužitelj zatim traži dodatnu potvrdu tog korisnika nakon koje, u slučaju pozitivnog odgovora, započinje s brisanjem štednje iz baze i odgovara s porukom o uspješno obrisanoj štednji.



Slika 4.4.11 Sekvencijski dijagram obrasca Obriši štednju

4.4.12 UC12: Dodaj novac u štednju

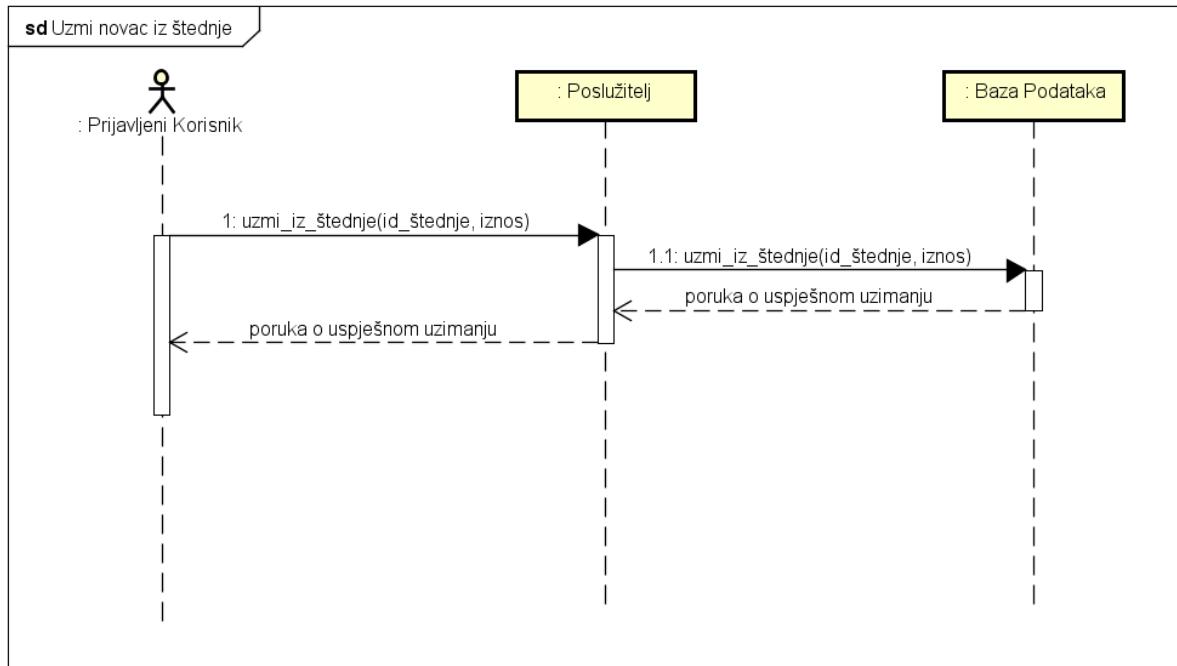
Prijavljeni korisnik šalje zahtjev za dodavanje određenog iznosa novca. Poslužitelj, zatim, započinje uređivanje štednje u bazi podataka te odgovara porukom o uspješnom dodavanju novca u odabranu štednju.



Slika 4.4.12 Sekvencijski dijagram obrasca Dodaj novac u štednju

4.4.13 UC13: Uzmi novac iz štednje

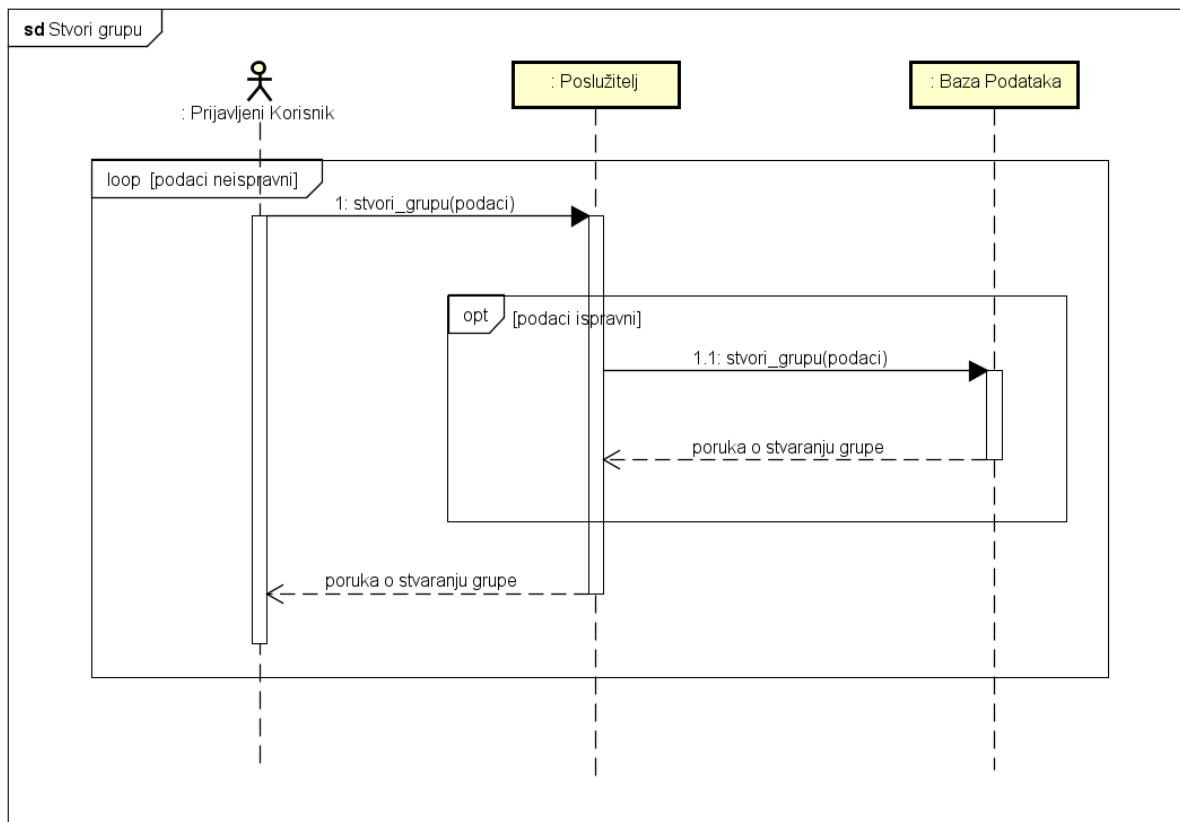
Prijavljeni korisnik šalje zahtjev za uzimanje određenog iznosa novca. Poslužitelj, zatim, započinje uređivanje štednje u bazi podataka te odgovara s porukom o uspješnom uzimanju novca iz odabrane štednje.



Slika 4.4.13 Sekvencijski dijagram obrasca Uzmi novac iz štednje

4.4.14 UC14: Stvori grupu

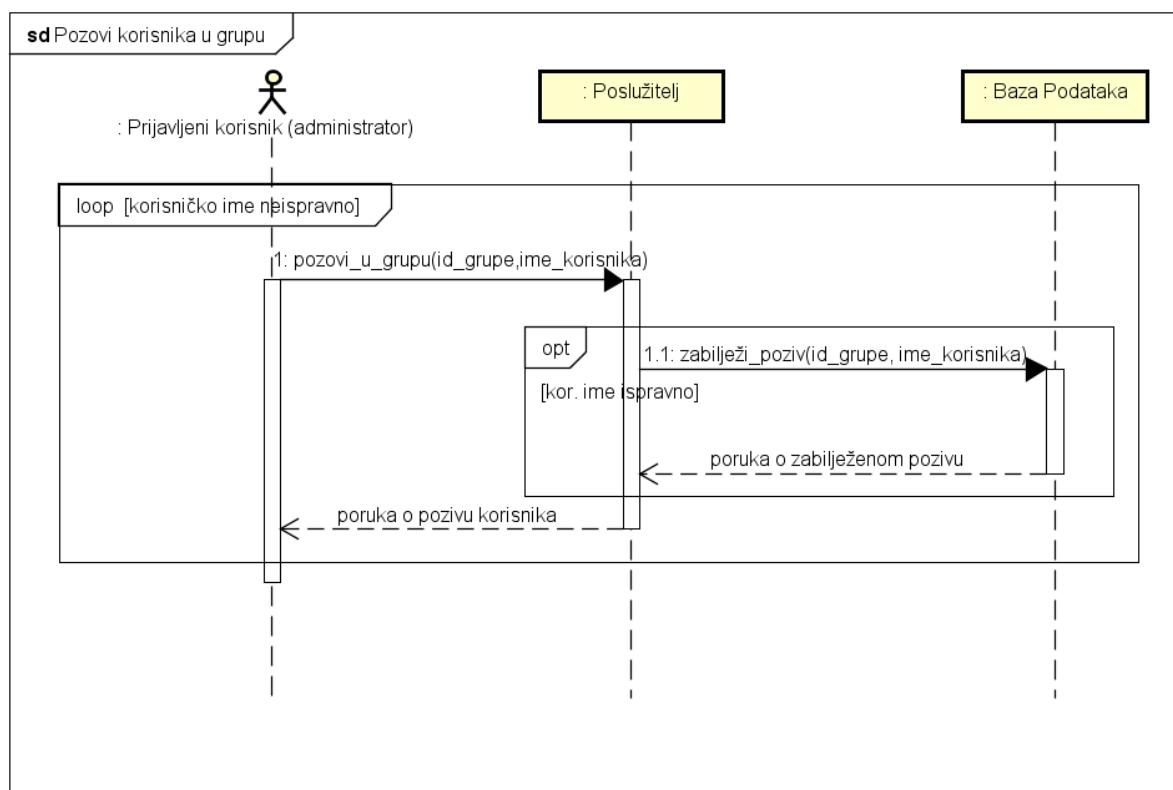
Prijavljeni korisnik unosi podatke potrebne za stvaranje grupe te šalje zahtjev za stvaranje grupe. Poslužitelj, ako su podaci ispravni, stvara grupu na temelju unosa u bazi podataka te odgovara porukom o uspješnom stvaranju grupe. U suprotnom obavještava korisnika o neispravnosti podataka.



Slika 4.4.14 Sekvencijski dijagram obrasca **Stvori grupu**

4.4.15 UC15: Pozovi korisnika u grupu

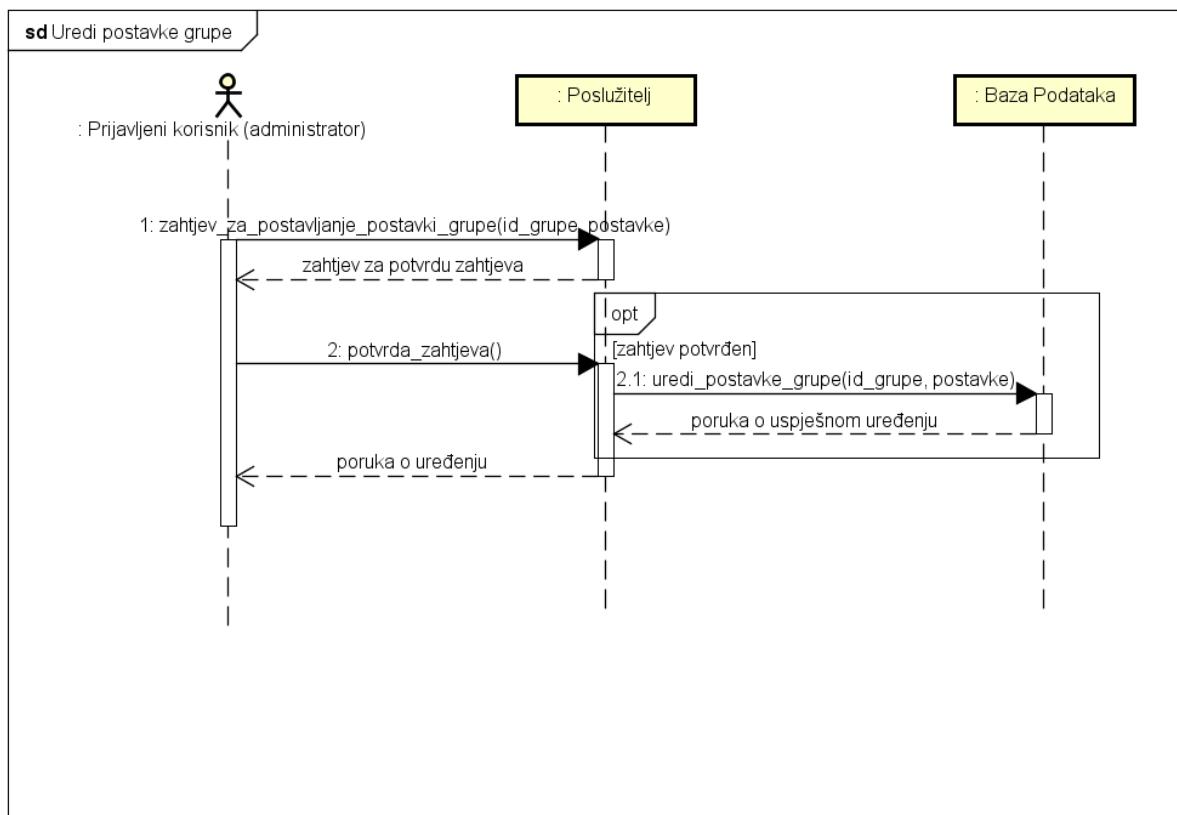
Prijavljeni korisnik, administrator grupe, unosi korisničko ime drugog registriranog korisnika i šalje zahtjev za poziv u grupu. Poslužitelj, ako navedeni korisnik postoji, šalje upit tome korisniku za prihvatanje poziva u grupu. U suprotnom obavještava pozivatelja o nepostojanju korisnika s navedenim imenom i omogućuje ponovni upis imena.



Slika 4.4.15 Sekvencijski dijagram obrasca Pozovi korisnika u grupu

4.4.16 UC16: Uredi postavke grupe

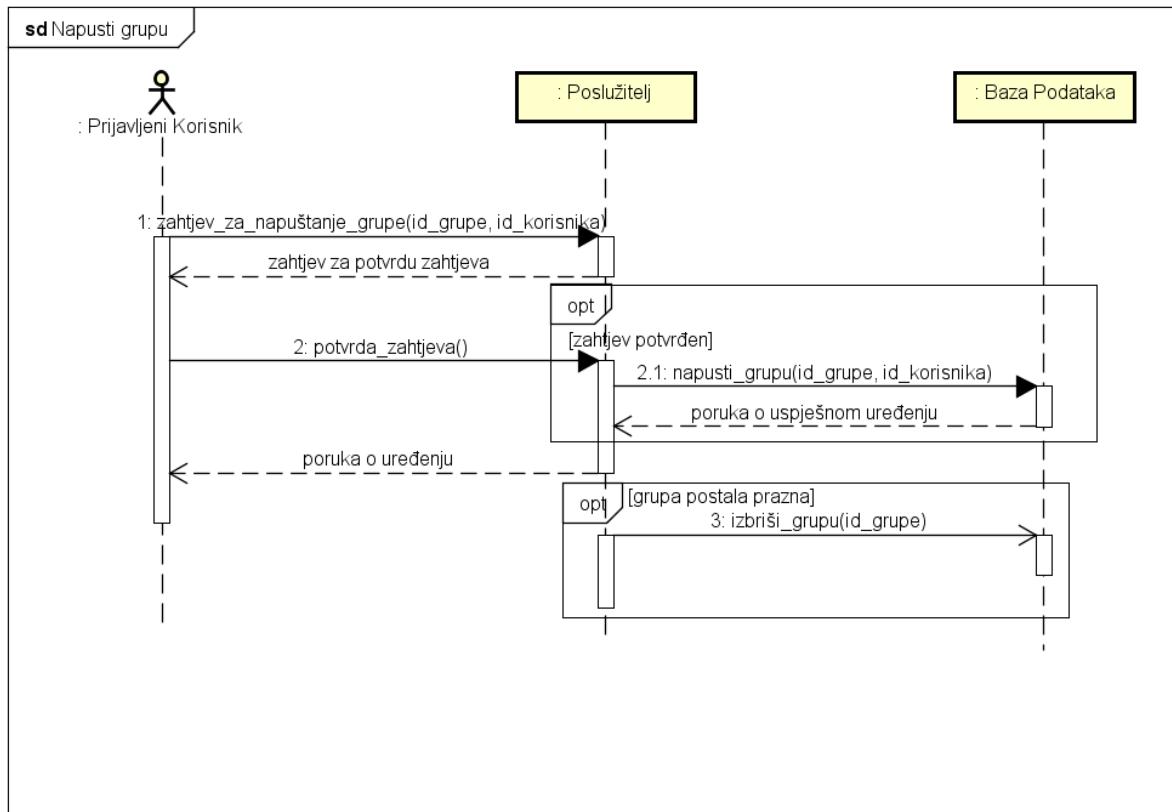
Prijavljeni korisnik, administrator grupe, mijenja postavke grupe i šalje zahtjev za promjenom postavki grupe. Poslužitelj, zatim, traži dodatnu potvrdu administratora nakon koje, u slučaju pozitivnog odgovora, uredi bazu podataka te odgovara s porukom o uspješnoj promjeni.



Slika 4.4.16 Sekvencijski dijagram obrasca Uredi postavke grupe

4.4.17 UC17: Napusti grupu

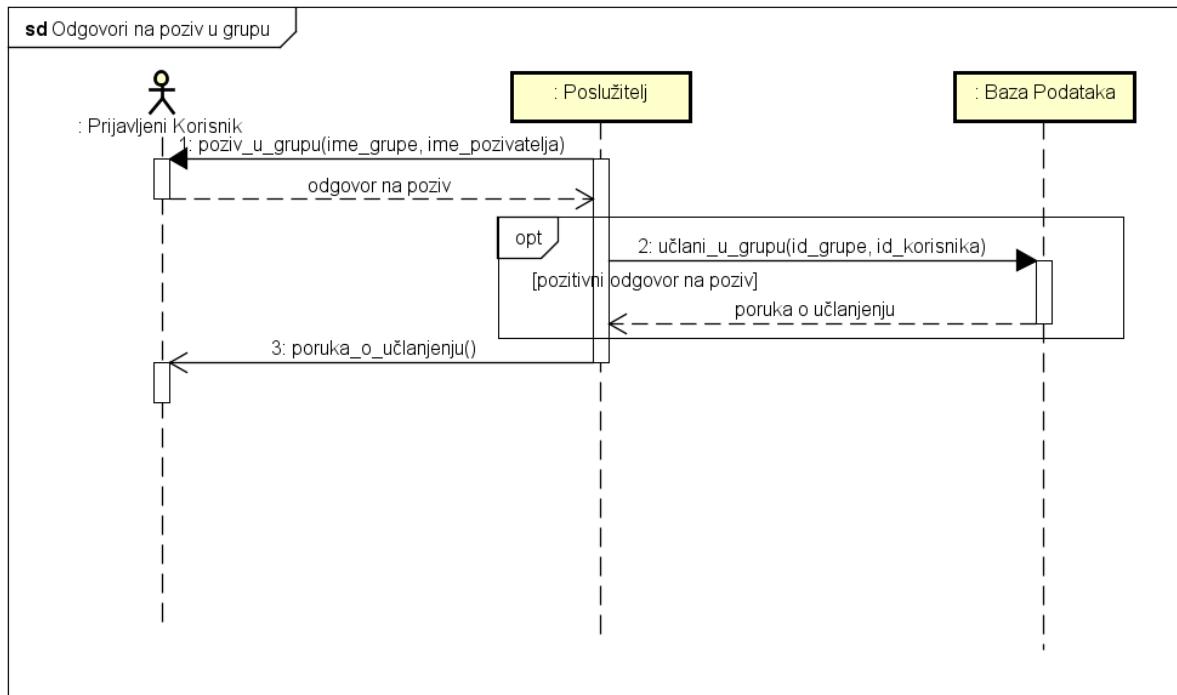
Prijavljeni korisnik šalje zahtjev za izlazak iz grupe čiji je član. Poslužitelj, zatim, traži dodatnu potvrdu korisnika nakon koje, u slučaju pozitivnog odgovora, uredi bazu podataka te odgovara s porukom o uspješnoj promjeni. Ako grupa nakon toga postane prazna, grupa se briše iz baze podataka.



Slika 4.4.17 Sekvencijski dijagram obrasca Napusti grupu

4.4.18 UC18: Odgovori na poziv u grupu

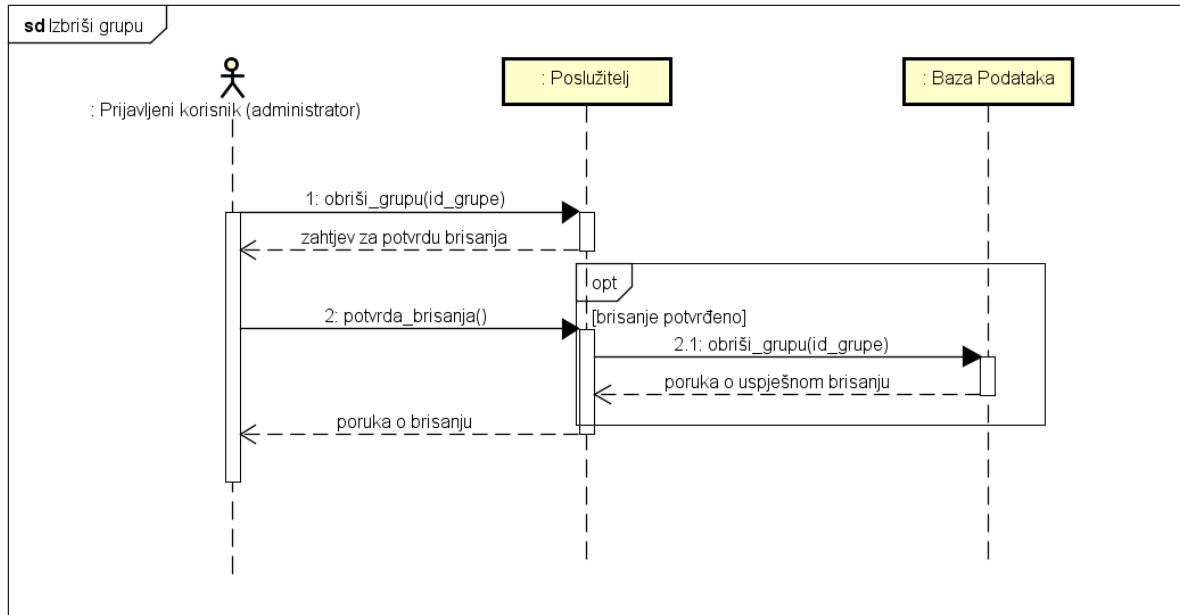
Prijavljeni korisnik odgovara na poslužiteljev upit o pozivu u grupu. Poslužitelj, ako je odgovor pozitivan, uređuje bazu podataka. Korisnik je obaviješten o posljedicama.



Slika 4.4.18 Sekvencijski dijagram obrasca Odgovori na poziv u grupu

4.4.19 UC19: Izbriši grupu

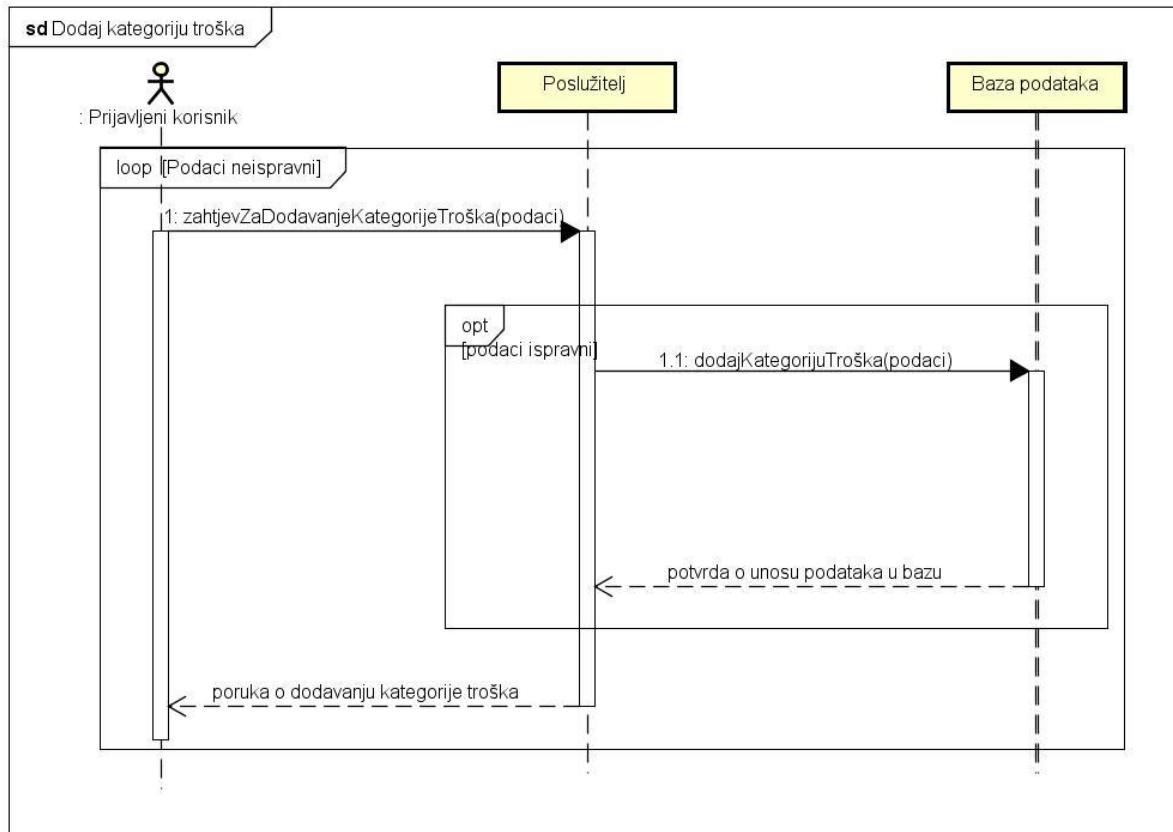
Prijavljeni korisnik, administrator grupe, šalje zahtjev za brisanje grupe. Poslužitelj, zatim, traži dodatnu potvrdu administratora nakon koje, u slučaju pozitivnog odgovora, uredi bazu podataka te odgovara s porukom o uspješnoj promjeni.



Slika 4.4.19 Sekvencijski dijagram obrasca Izbriši grupu

4.4.20 UC20: Dodaj kategoriju troška

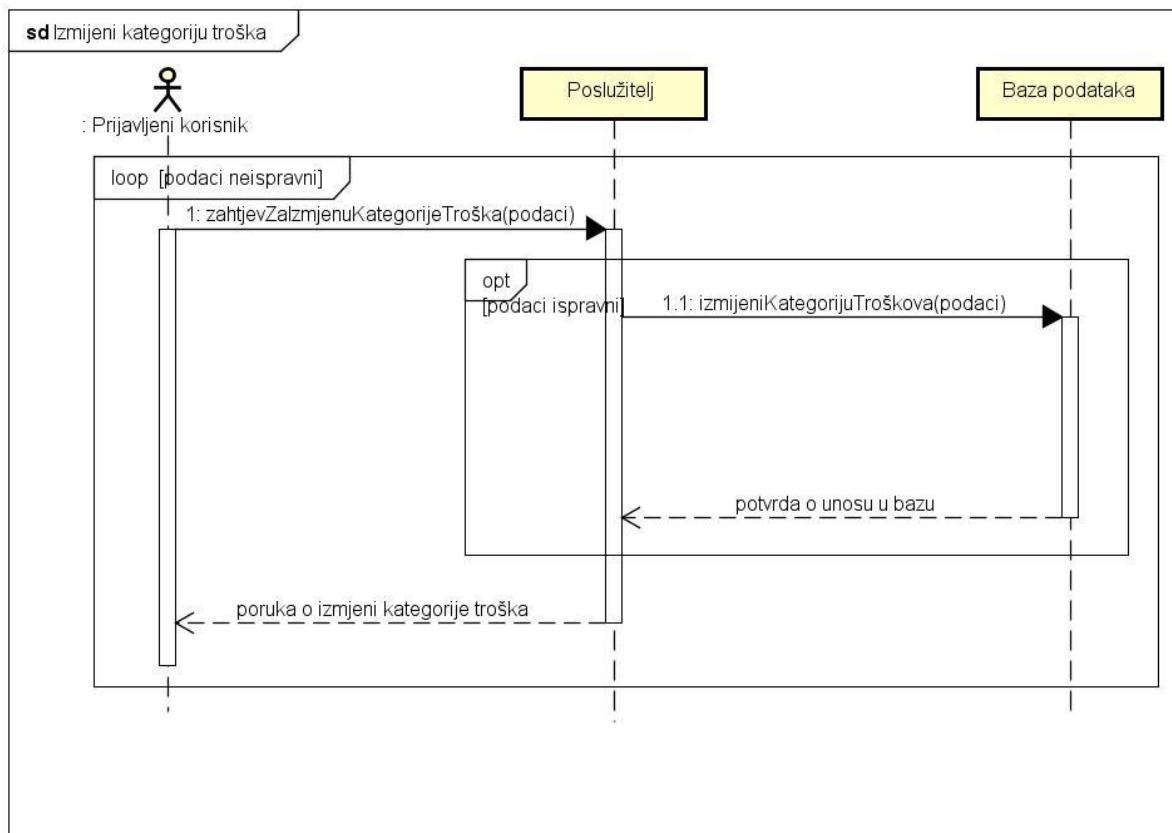
Prijavljeni korisnik unosi podatke o dodavanju nove kategorije troška. Poslužitelj traži potvrdu unosa podataka i pri njenom primitku kategorija troška se upisuje u bazu. Poslužitelj dobiva povratnu informaciju od baze, koju prosljeđuje korisniku. Ukoliko korisnikov unos podataka nije pravilan postupak se ponavlja.



Slika 4.4.20 Sekvenčni dijagram obrasca Dodaj kategoriju troška

4.4.21 UC21: Izmjeni kategoriju troška

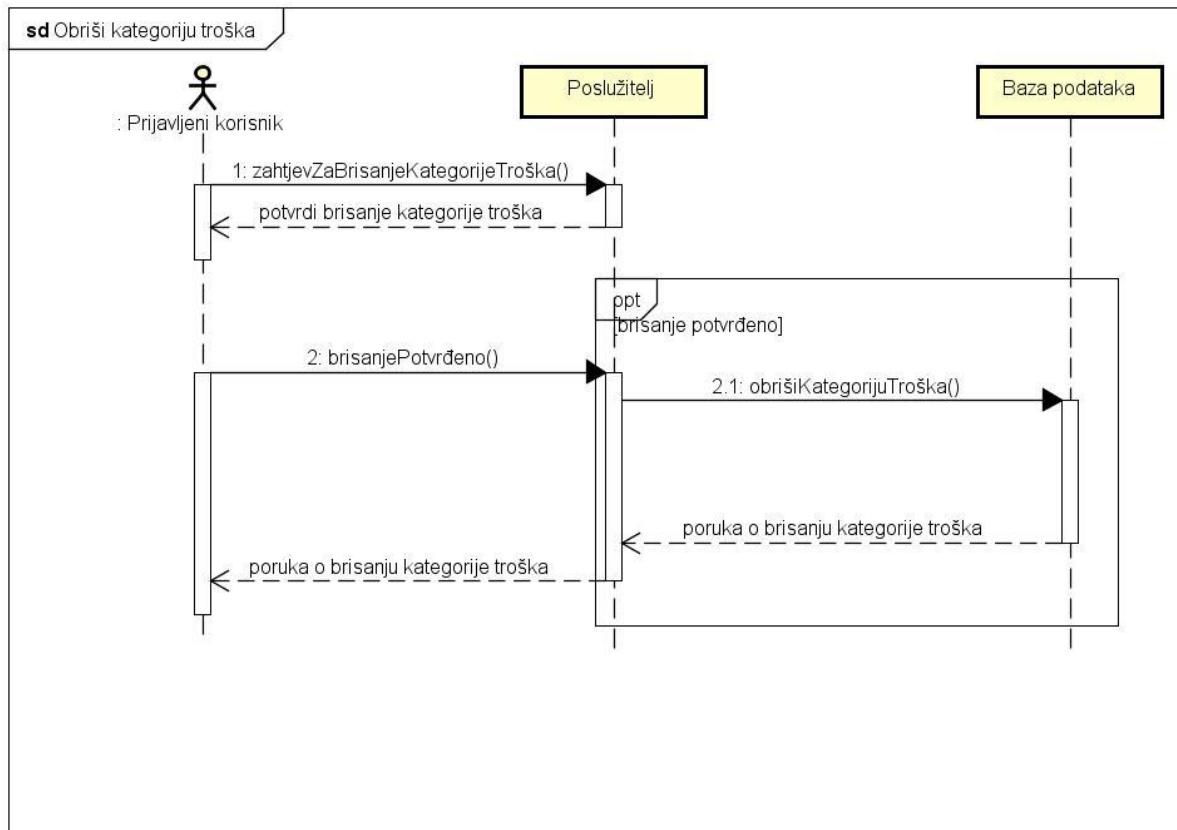
Prijavljeni korisnik unosi podatke za izmjenu kategorije troška. Poslužitelj traži potvrdu unosa podataka i pri njenom primitku u bazu se upisuju izmjenjeni podaci o kategoriji troška. Poslužitelj prima povratnu informaciju o upisu u bazu, koji prosljeđuje korisniku. Ukoliko korisnikov unos podataka nije pravilan postupak se ponavlja.



Slika 4.4.21 Sekvencijski dijagram obrasca Izmjeni kategoriju troška

4.4.22 UC22: Obriši kategoriju troška

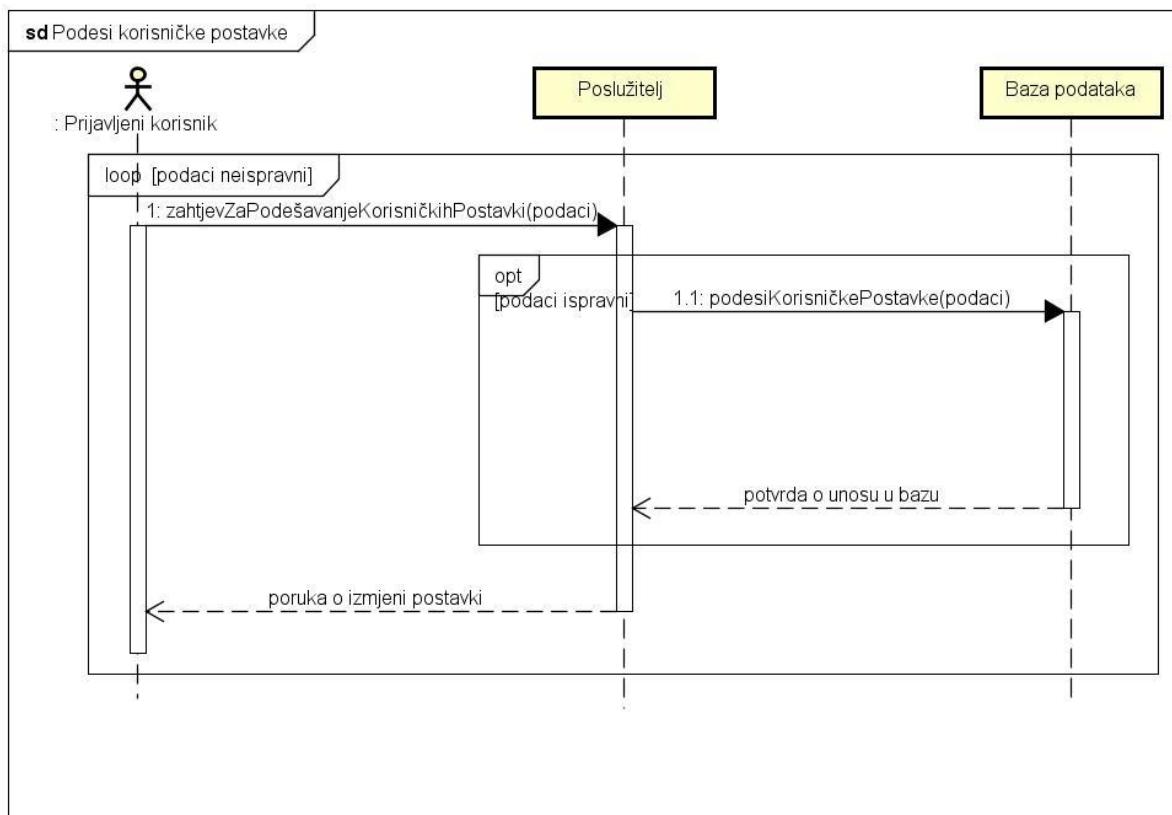
Prijavljeni korisnik podnosi zahtjev za brisanje kategorije troška. Poslužitelj traži potvrdu brisanja kategorije troška od korisnika i pri njenom primitku iz baze se briše kategorija troška. Poslužitelj prima povratnu informaciju o brisanju od baze koju prosljeđuje korisniku.



Slika 4.4.22 Sekvenčni dijagram obrasca Obriši kategoriju troška

4.4.23 UC23: Podesi korisničke postavke

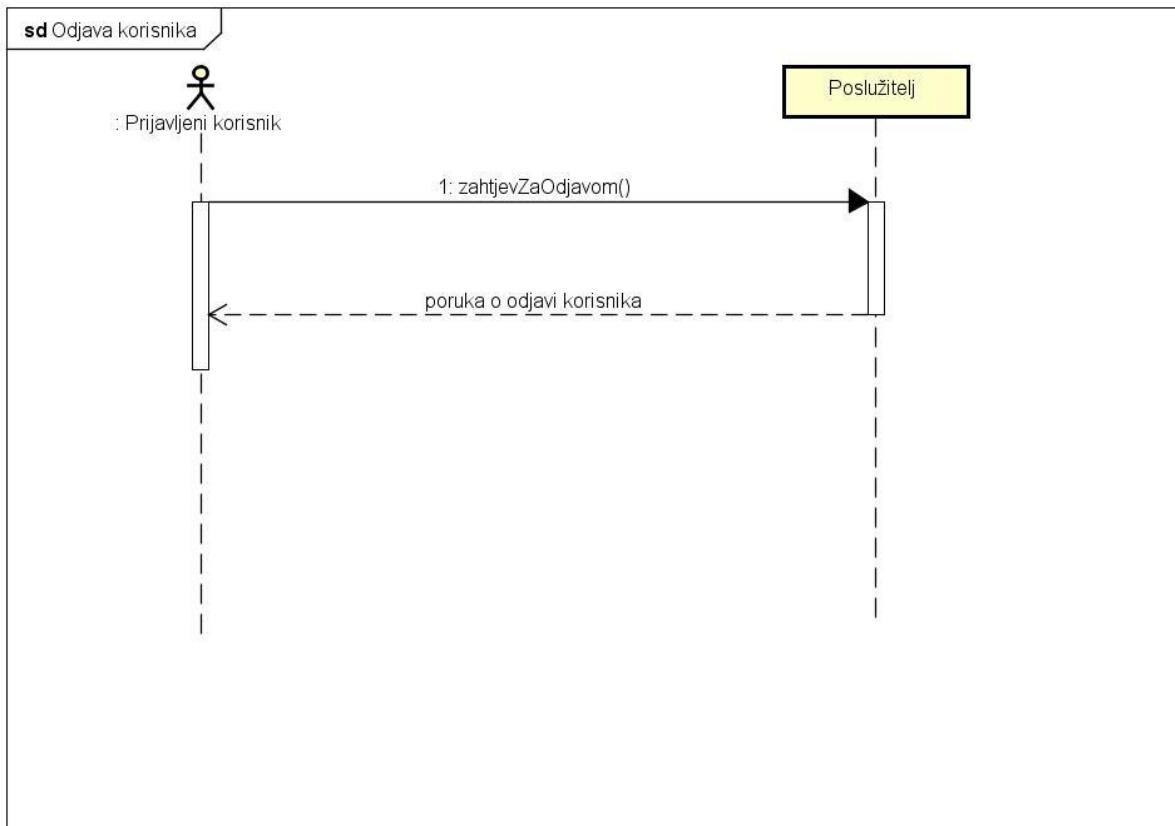
Prijavljeni korisnik unosi nove postavke za svoj korisnički račun. Poslužitelj traži potvrdu o unosu podataka i pri njenom primitku u bazu se upisuju nove postavke korisnika. Poslužitelj prima poruku o unosu podataka od baze i prenosi ju korisniku. Ukoliko unos podataka nije ispravan postupak se ponavlja.



Slika 4.4.23 Sekvencijski dijagram obrasca Podesi korisničke postavke

4.4.24 UC24: Odjava korisnika

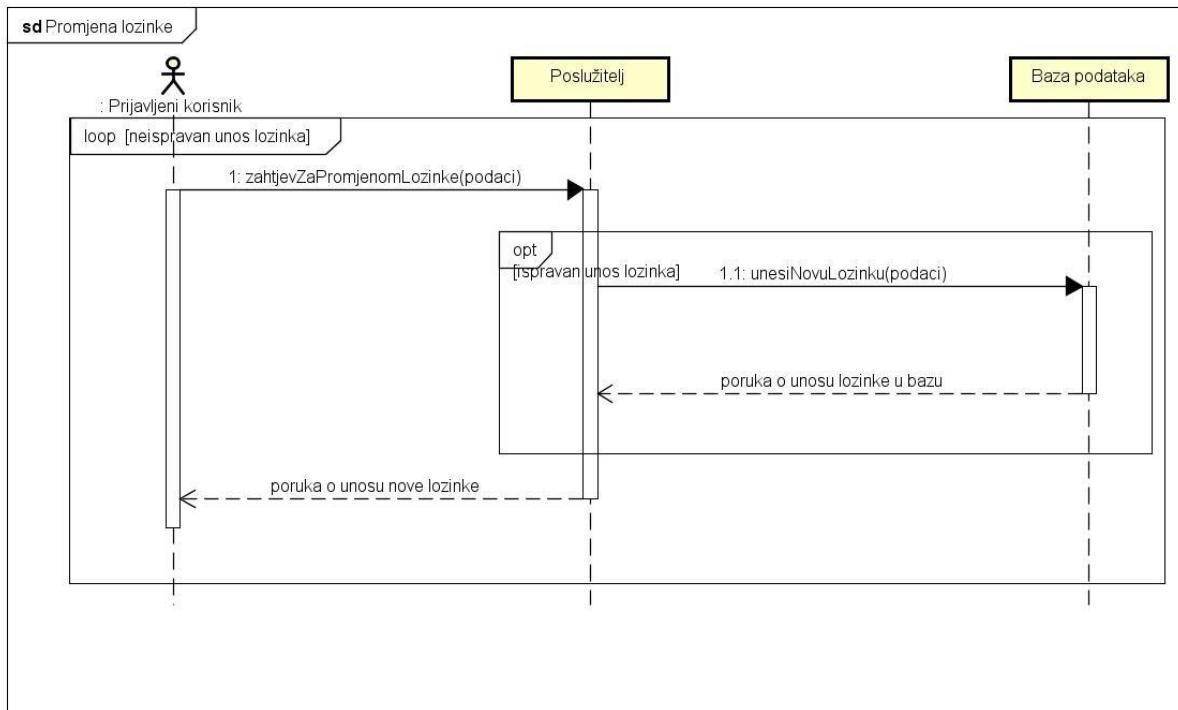
Prijavljeni korisnik šalje poslužitelju zahtjev za odjavu. Poslužitelj prihvata zahtjev i korisnik se odjavljuje.



Slika 4.4.24 Sekvencijski dijagram obrasca Odjava korisnika

4.4.25 UC25: Promjena lozinke

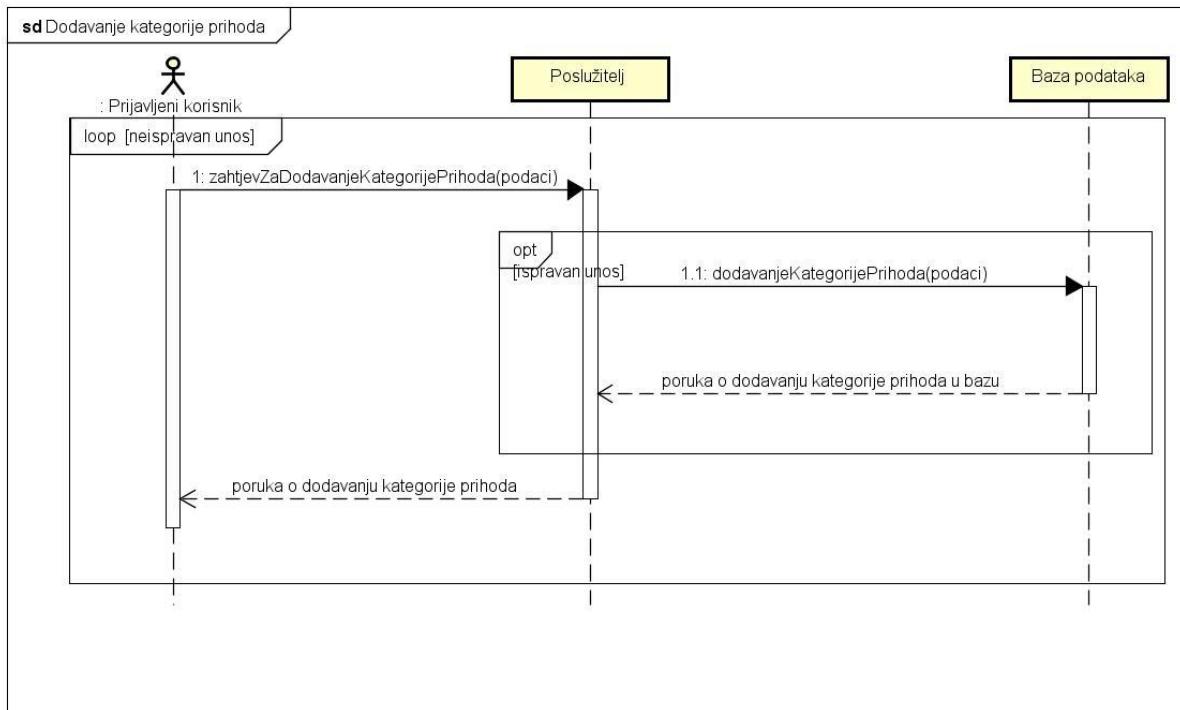
Prijavljeni korisnik odlazi u korisničke postavke i otvara poveznicu za promjenu lozinke. Unosi trenutnu lozinku i novu koju ponavlja. Poslužitelj od korisnika traži da potvrdi unos i primitkom potvrde poslužitelj šalje podatke za upis nove lozinke u bazu podataka. Nova lozinka se upisuje u bazu i o tome se obavještava korisnika. U slučaju krivog unosa trenutne ili nove lozinke, postupak se ponavlja.



Slika 4.4.25 Sekvencijski dijagram obrasca Promjena lozinke

4.4.26 UC26: Dodavanje kategorije prihoda

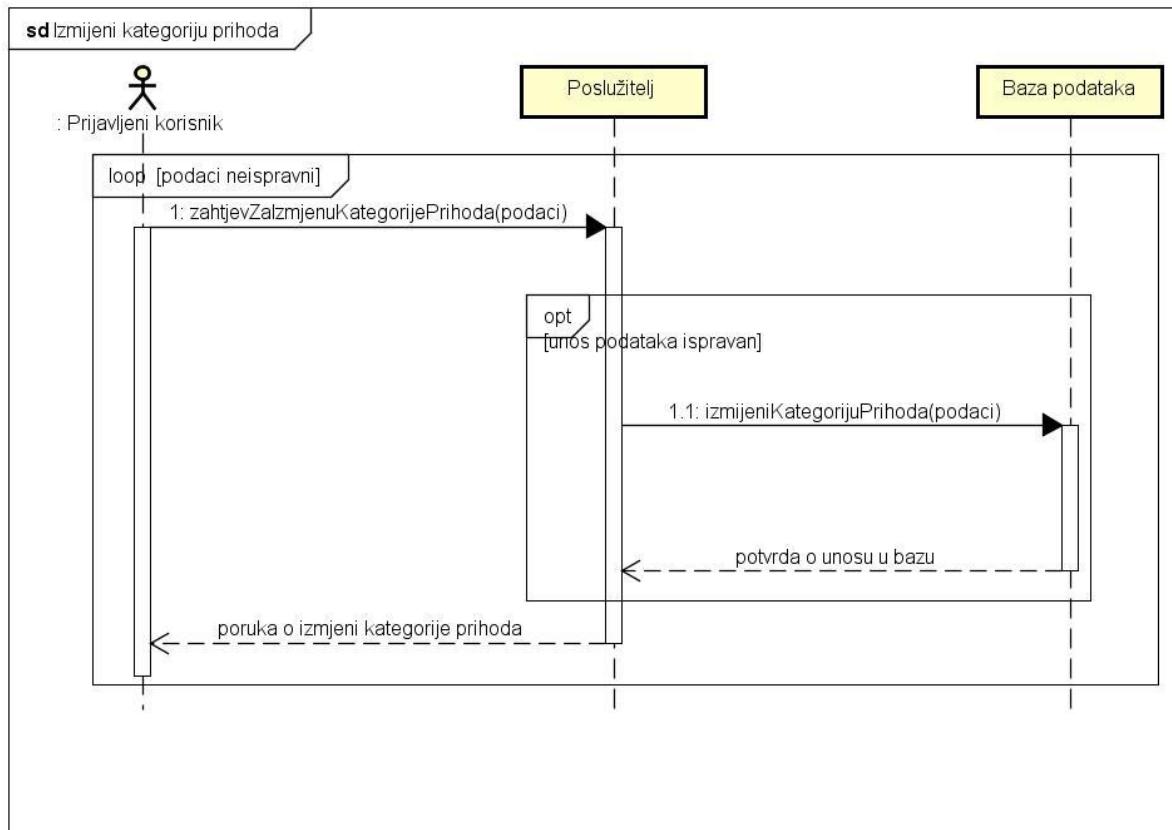
Prijavljeni korisnik upisuje podatke za stvaranje nove kategorije prihoda. Poslužitelj zahtjeva potvrdu unosa od korisnika i pri njenom primitku nova kategorija prihoda unosi se u bazu. Baza podataka vraća poruku u unosu podataka poslužitelju koji ju prosljeđuje korisniku. Ukoliko kategorija prihoda već postoji ili unos podataka nije ispravan postupak se ponavlja.



Slika 4.4.26 Sekvencijski dijagram obrasca Dodavanje kategorije prihoda

4.4.27 UC27: Izmjeni kategoriju prihoda

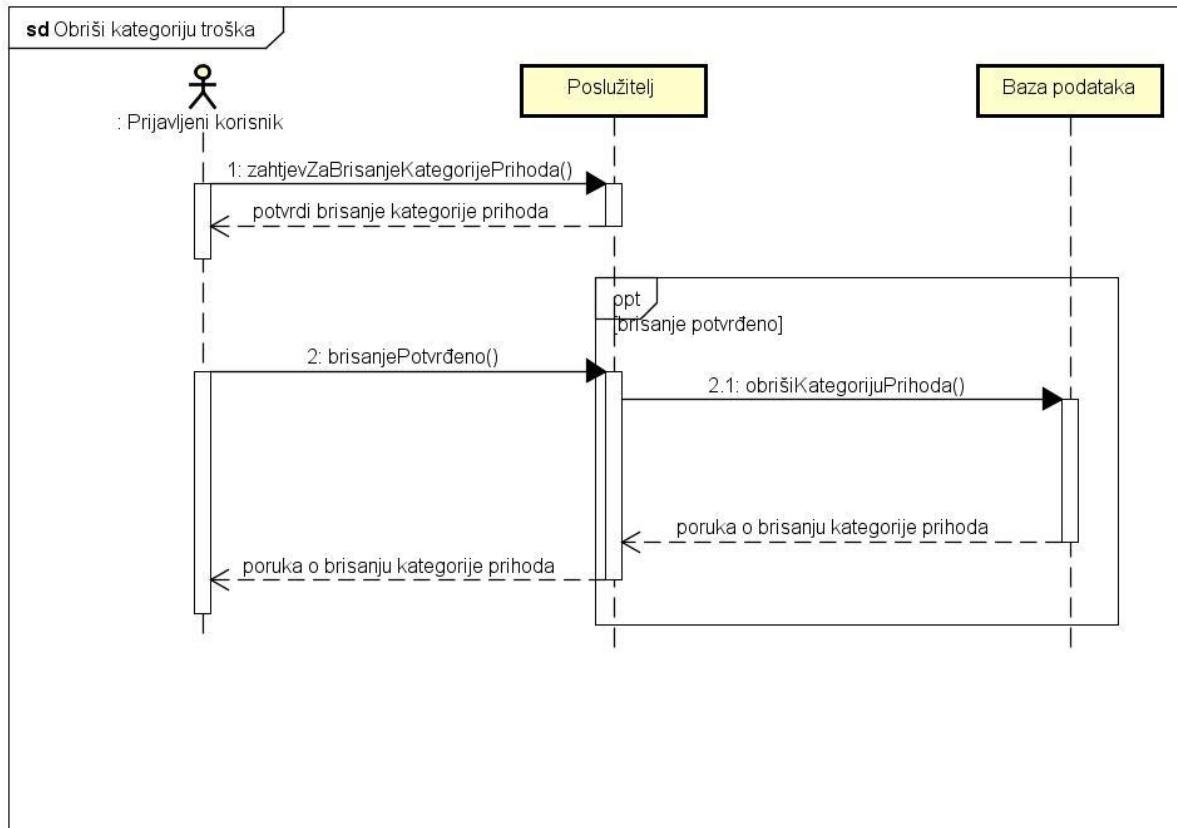
Prijavljeni korisnik odabire opciju za izmjenu kategorije prihoda i unosi promjene. Poslužitelj od korisnika zahtjeva da potvrdi unos podataka i pri primitku potvrde izmijene se unose u bazu podataka. Korisnik je obaviješten o unosu promjena u bazu. Ukoliko unos podataka nije ispravan postupak se ponavlja.



Slika 4.4.27 Sekvencijski dijagram obrasca Izmjeni kategoriju prihoda

4.4.28 UC28: Obriši kategorije prihoda

Prijavljeni korisnik odabire opciju za brisanje kategorije prihoda. Poslužitelj od njega traži potvrdu o brisanju i pri primjeku potvrde poslužitelj bazi šalje zahtjev za brisanje kategorije prihoda iz baze. Korisnik prima povratnu informaciju o brisanju.



Slika 4.4.28 Sekvencijski dijagram obrasca Obriši kategorije prihoda

5. Ostali zahtjevi

- Aplikacija treba biti izvedena kao web aplikacija
- Web aplikacija mora biti prilagođena prikazu na mobilnom uređaju
- Korisnici pristupaju sustavu putem korisničkog imena i lozinke
- Sustav mora podržati istovremeni rad više korisnika
- Prilikom testiranja sustava trebaju biti uneseni podaci za barem pet troškova te minimalno jedan prihod i jednu štednju
- Neispravno rukovanje korisnika aplikacijom ne smije utjecati na stabilnost aplikacije i baze podataka, odnosno sustav mora biti neosjetljiv na korisničke pogreške
- Sustav mora podržavati hrvatske dijakritičke znakove
- Dohvaćanje podataka iz baze podataka ne smije trajati duže od 10 sekundi
- Sustav podržava korištenje valute Hrvatska kuna (HRK)
- Prilikom unosa novih prihoda ili troškova, trenutno stanje računa mora biti automatski ažurirano
- Prilikom dodavanja ili oduzimanja novca iz štednje, trenutno stanje štednje i trenutno stanje računa moraju biti automatski ažurirani
- Korisnici ne smiju imati pristup funkcionalnostima za čije korištenje nisu ovlašteni
- Sustav treba jamčiti pouzdanost prikazanih podataka
- Vrijeme izrade aplikacije ne smije premašiti period od 3 mjeseca

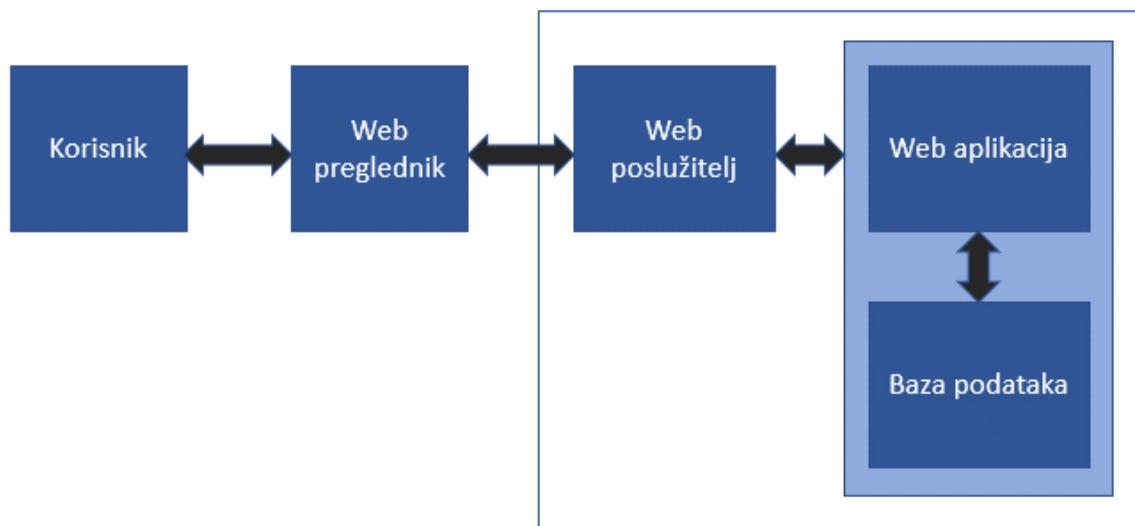
6. Arhitektura i dizajn sustava

6.1 Svrha, opći prioriteti i skica sustava

Budući da je potrebno osigurati jednostavan rad više korisnika s istovremenim pristupom, za izvedbu projektnog zadatka odabrana je arhitektura **web aplikacije**. Ovaj pristup također olakšava upotrebu aplikacije za korisnika jer mu je za korištenje potreban klasični web preglednik kojeg većina računala s pristupom internetu već posjeduje.

Glavne komponente sustava su:

- Web poslužitelj
- Web aplikacija
- Baza podataka



Slika 6.1.1 Skica sustava

Web poslužitelj

Osnovna zadaća web poslužitelja je obrada zahtjeva korisnika koji stižu preko web preglednika. Komunikacija se odvija kroz HTTP, koji je mrežni protokol.

Web aplikacija

Primanje korisničkih zahtjeva te njihovo obrađivanje je glavna zadaća web aplikacije jer ona preoblikuje korisničke zahtjeve u operacije sustava. Web aplikacija upravlja podacima koje pohranjuje i dohvaća iz baze podataka. Jedna od zadaća je i vizualni prikaz sustava korisniku. Aplikacija priprema vizualne elemente koje potom web preglednik prikazuje korisniku u obliku web stranice.

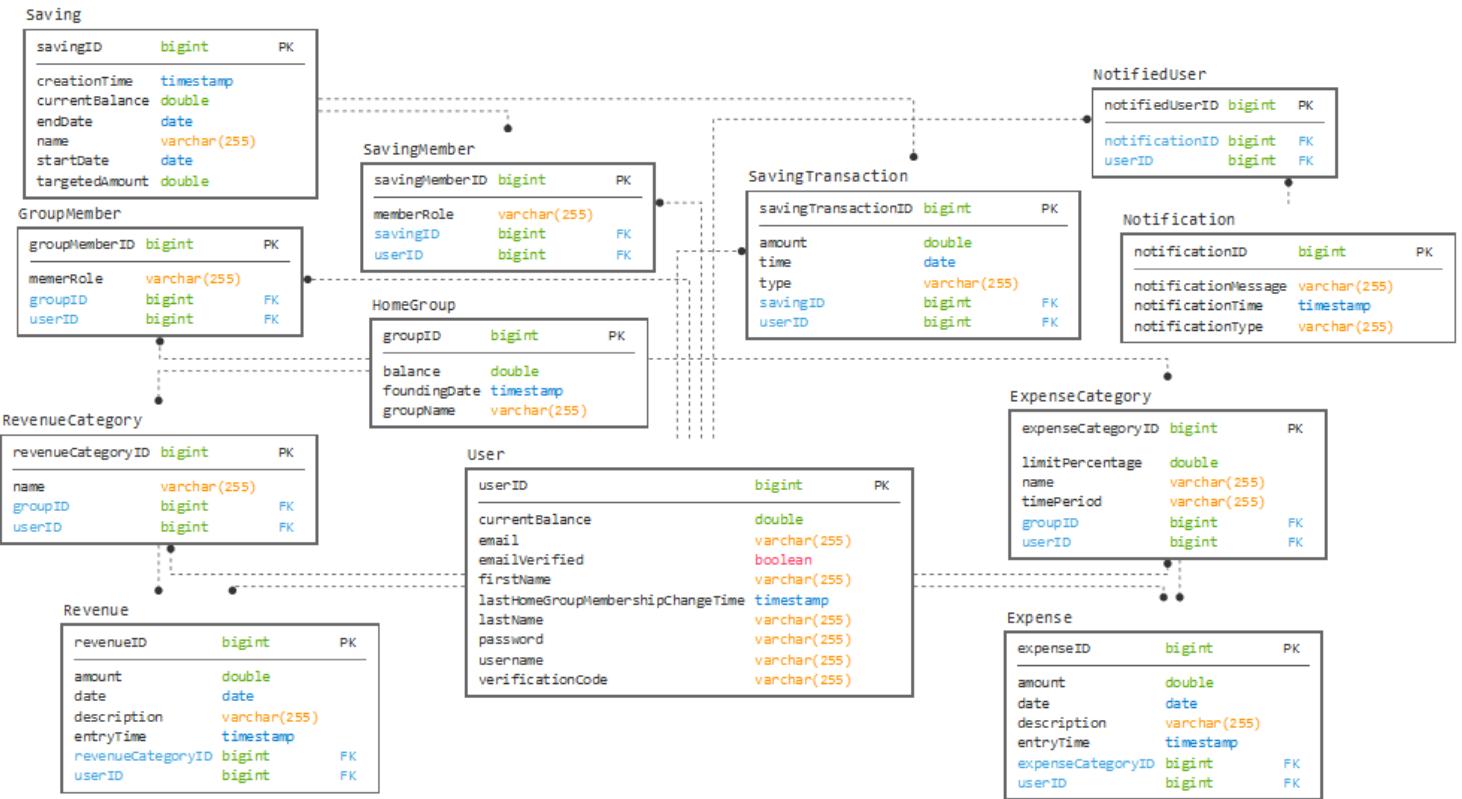
Sustav je izведен u **Model-View-Controller arhitekturi** koja odvaja komponente sustava u 3 cjeline:

- **Model (Model)**
 - Model je dio MVC arhitekture koji obuhvaća podatkovnu strukturu sustava, odnosno zadužen je za upravljanje podacima i poslovnom logikom aplikacije.
- **View (Pogled)**
 - Pogled je dio MVC arhitekture koji je zadužen za prikaz informacija korisniku aplikacije te služi kao poveznica korisnika i sustava preko koje korisnik postavlje zahtjeve sustavu.
- **Controller (Upravitelj)**
 - Upravitelj upravlja zahtjevima koje šalju ostale komponente MVC arhitekture. Zadužen je za dohvaćanje podataka iz baze podataka te proslijedivanje podatkovnih resursa onoj komponenti koja ih je zahtjevala.

Baza podataka

Baza podataka bilježi sve podatke potrebne za rad sustava i čini ih sigurnima. Baza podataka koristi relacijski model, odnosno podaci su pohranjeni u obliku tablica i relacija među njima. Svaka tablica se sastoji od redaka (pohranjene n-torce) i stupaca (atributi pripadne tablice). Model je doveden u 3. normalnu formu. Za upravljanje bazom podataka korišten je *PostgreSQL*. Baza se sastoji od sljedećih tablica:

- **User** – sadrži potrebne podatke o korisniku aplikacije
- **ExpenseCategory** – sadrži podatke o kategoriji troškova
- **Expense** – sadrži podatke o trošku
- **RevenueCategory** – sadrži podatke o kategoriji prihoda
- **Revenue** – sadrži podatke o prihodu
- **Notification** – sadrži podatke o obavijestima koje se javljaju pri korištenju aplikacije
- **NotifiedUser** – podaci koji povezuju obavijest s korisnikom
- **SavingTransaction** – prikazuje transakcije vezane uz pojedinu štednju
- **HomeGroup** – sadrži podatke o tipu grupe koja simulira kućni budžet
- **SavingMember** – prikazuje članstvo u štednji
- **Saving** – sadrži sve potrebne podatke o štednji
- **GroupMember** – prikazuje članstvo u grupi



Slika 6.1.2 Model baze podataka

6.2 Dijagram razreda s opisom

Osnovna podjela razreda u ovom projektu je definirana načelima Model-View-Controller arhitekture. Razredi model dijela MVC arhitekture predstavljaju centralni dio aplikacije. To su entiteti koji definiraju podatke i poslovnu logiku sustava te predstavljaju sponu između upravitelj (controller) i pogled (view) komponenti MVC arhitekture i baze podataka.

User (*korisnik*) je entitet koji opisuje korisnika aplikacije. Svaki korisnik ima jedinstveno korisničko ime i adresu elektroničke pošte. Prilikom registracije korisnik unosi svoje osnovne osobne podatke kao što su ime i prezime te postavlja lozinku kojom je osiguran pristup korisničkom računu. E-mail adresa korisnika se mora verificirati prije korištenja usluga aplikacije. Uz korisnika se vežu kategorije troškova i prihoda koje on sam definira. Za svaku kategoriju troška, korisnik definira postavke obavijesti kojima može regulirati parametre usluga upozorenja i savjetovanja u slučaju prekomjernog trošenja. Sustav pamti sve korisnikove troškove, prihode i štednje. U skladu s unosom prihoda i troškova aplikacija prati trenutačno stanje korisničkog računa. Korisnik može pristupiti jednoj grupi.

HomeGroup (*kućna grupa*) je entitet koji opisuje grupu čiji su članovi najčešće ukućani nekog kućanstva koji dijele sve prihode, troškove i štednje. Članovi grupe imaju prikazano zajedničko stanje računa te zajedničke kategorije prihoda i troškova. Osnivač grupe je ujedno i administrator koji može dati administratorske ovlasti i drugim članovima grupe.

GroupMember (*član grupe*) predstavlja vezni entitet između korisnika i grupe kojim se označava članstvo u grapi te korisnikov status i uloga u toj grapi. Korisnikova uloga može biti član, administrator ili samo pozvani član.

ExpenseCategory (*kategorija troška*) je entitet koji opisuje kategoriju troška. Svaka kategorija troška ima ime te ovisno o tome tko ju koristi, korisnika ili grupu koji ju je napravio. Također, uz svaku kategoriju dolaze i postavke za notifikacije koje se

aktiviraju pri određenim prekoračenjima u troškovima ovisno o postavkama koje je definirao korisnik.

RevenueCategory (*kategorija prihoda*) je entitet koji opisuje kategoriju prihoda. Svaka kategorija sadrži ime te korisnika ili grupu, ovisno o tome tko ju je napravio.

Expense (*trošak*) je entitet koji definira trošak. Svakom trošku je pridružen iznos, vrijeme troška, korisnik koji ga je napravio, opis troška, vrijeme unosa te kategorija.

Revenue (*prihod*) je entitet kojim je opisan prihod. Svaki prihod ima definiran iznos, kategoriju prihoda, vrijeme prihoda, vrijeme unosa, opis te korisnika koji je stvorio prihod.

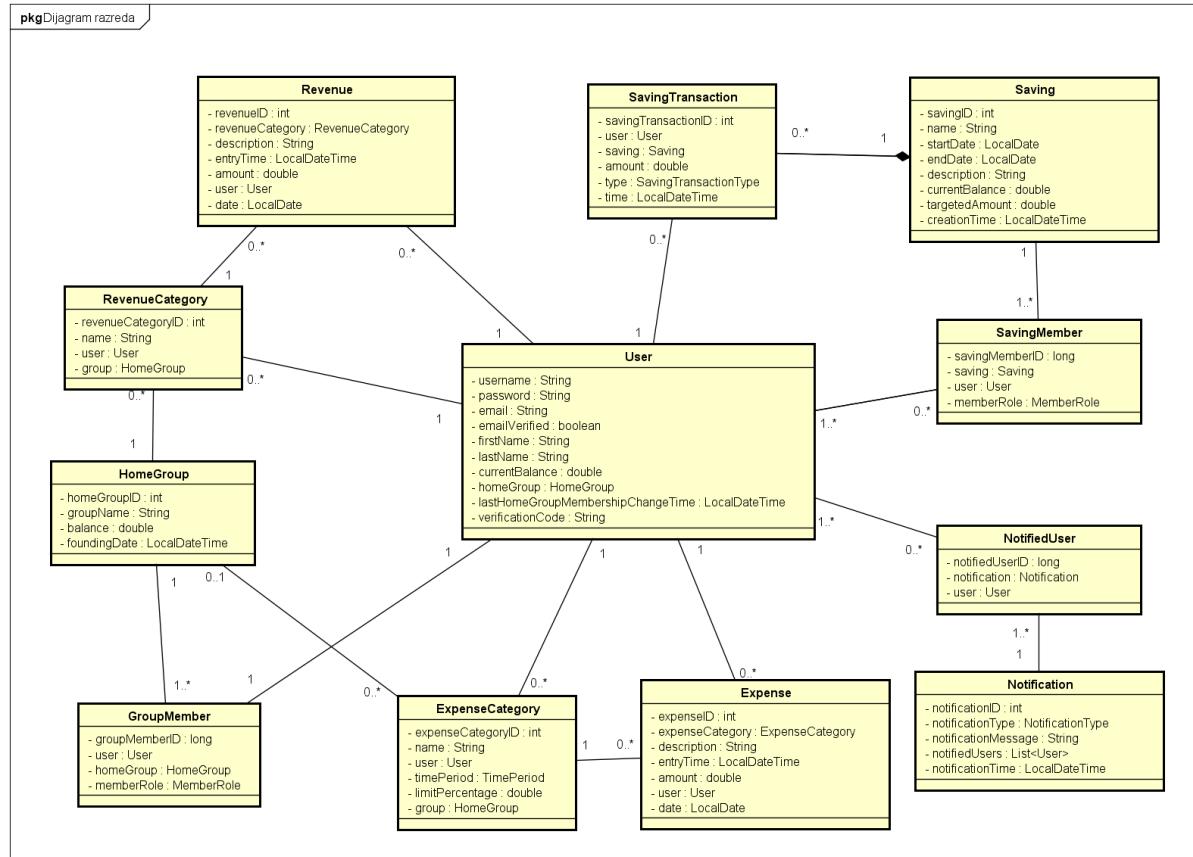
Saving (*štednja*) opisuje štednju koja ima ime, opis, datum početka, datum završetka, odnosno vremenski rok u kojem se treba prikupiti ciljni iznos. U štednji je definiran i trenutni iznos novca koji je prikupljen. U jednoj štednji može štedjeti više korisnika, a postavkama štednje upravljaju administratori. Vrijeme upisa štednje je također zabilježeno.

SavingMember (*član štednje*) predstavlja vezni entitet između korisnika i štednje kojim se označava članstvo u štednji te korisnikov status i uloga u toj štednji. Korisnikova uloga može biti član, administrator ili samo pozvani član.

SavingTransaction (*novčana transakcija vezana uz štednju*) je entitet koji opisuje jednu uplatu ili uzimanje novca iz štednje. Za svaku transakciju poznat je korisnik koji ju napravio, štednja na koju se odnosi, iznos, vrijeme transakcije te tip transakcije. Dva su moguća tipa transakcije: dodavanje novca u štednju ili uzimanje novca iz štednje.

Notification (*obavijest*) je razred koji opisuje obavijest koju korisnik dobiva u slučaju ispunjenja definiranih parametara u sustavu. Svaka obavijest ima svoju vrstu, vrijeme nastanka, listu korisnika na koje se odnosi te poruku.

NotifiedUser (*obaviješteni korisnik*) je vezni entitet između obavijesti i korisnika kojim se definira na koje korisnike se odnosi pojedina obavijest.



Slika 6.2.1 Dijagram razreda modela

6.3 Dijagram objekata

Da bi se demonstrirao rad sustava u stvarnoj situaciji, dijagramom je prikazan izgled razrednog modela u tipičnoj situaciji, tj. na primjeru dva korisnika koji imaju jednu zajedničku štednju.

Postoje dvije instance razreda User, a to su u ovom slučaju korisnici *i22* i *pperic1*. Svaki korisnik ima svoj identifikacijski broj (userID), korisničko ime (userName), lozinku (password), email adresu (email), podatak o potvrđenosti email adrese (emailVerified), kod kojom je potvrđena (verificationCode), ime (firstName), prezime (lastName), trenutačno stanje (currentBalance), te vrijeme posljedne promjene članstva u kućnoj grupi (lastHomeGroupMembershipChangeTime).

Instanca razreda Revenue (prihod) sadrže podatke o prihodu: identifikacijski broj prihoda (revenueID), njegovu kategoriju (revenueCategory), iznos (amount), datum (date), opis (description), datum i vrijeme unosa (entryTime) i informaciju o korisniku kojemu prihod pripada (user). Nadalje, instanca razreda RevenueCategory (kategorije prihoda) sadrži podatke: identifikacijski broj kategorije (revenueCategoryID), naziv kategorije (name), korisnika na koji se odnosi (user) i informaciju o kućnoj grupi (homeGroup).

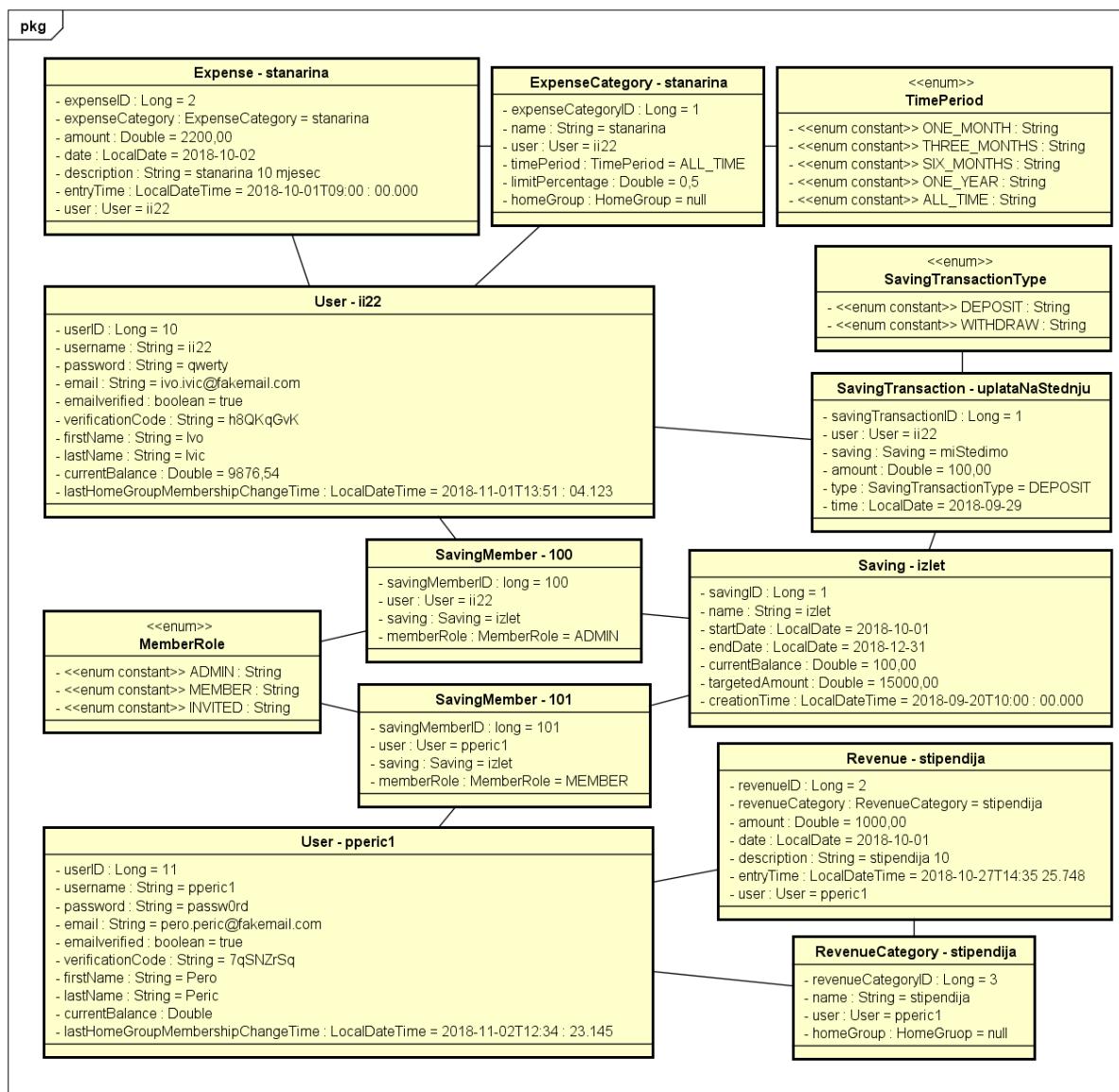
Slično, instanca razreda Expense (trošak) sadrži podatke o trošku: identifikacijski broj (expenseID), njegovu kategoriju (expenseCategory), iznos (amount), datum (date), opis (description), datum i vrijeme unosa (entryTime) i informaciju o korisniku kojem trošak pripada (user). Nadalje instanca razreda ExpenseCategory (kategorija troškova) sadrži podatke: identifikacijski broj kategorije (expenseCategoryID), naziv (name), korisnika na koji se odnosi (user), vremensko razdoblje za izračunavanje parametara koji uzrokuju slanje obavijesti (timePeriod) za što postoji enumeracija vremensko razdoblje (TimePeriod), ograničavajući postotak (limitPercentage) i informaciju o grupi (homeGroup).

Klasa Saving (štедnja) obuhvaća identifikacijski broj štednje (savingID), naziv štednje (name), datum početka (startDate), datum završetka (endDate), trenutačno stanje (currentBalance), ciljni iznos (targetedAmount), i datum kreiranja (creationTime). Za

svaku uplatu (deposit) ili isplatu (withdraw) na štednju stvara se objekt transakcija štednje (savingTransaction) s informacijama: identifikacijski broj transakcije (savingTransactionID), korisnik na kojeg se odnosi (user), štednja na koju se transakcija odnosi (saving), iznos transakcije (amount), tip (može biti uplata ili isplata, a za to postoji enumeracija tip transakcije (SavingTransactionType)), i vrijeme transakcije (time).

Za povezivanje korisnika i štednje postoji klasa SavingMember (član štednje) koja sadrži podatke: identifikacijski broj korisnika štednje (savingMemberID), korisnika (user), štednju (saving) i korisničku ulogu (memberRole) za koju se ujedno koristi enumeracija korisnička uloga (MemberRole). Instanca tog razreda kreira se za svakog korisnika koji ima neku štednju.

Korisnik može podešavati postavke obavijesti, a za podešavanje postavki grupe zaduženi su administratori grupe.

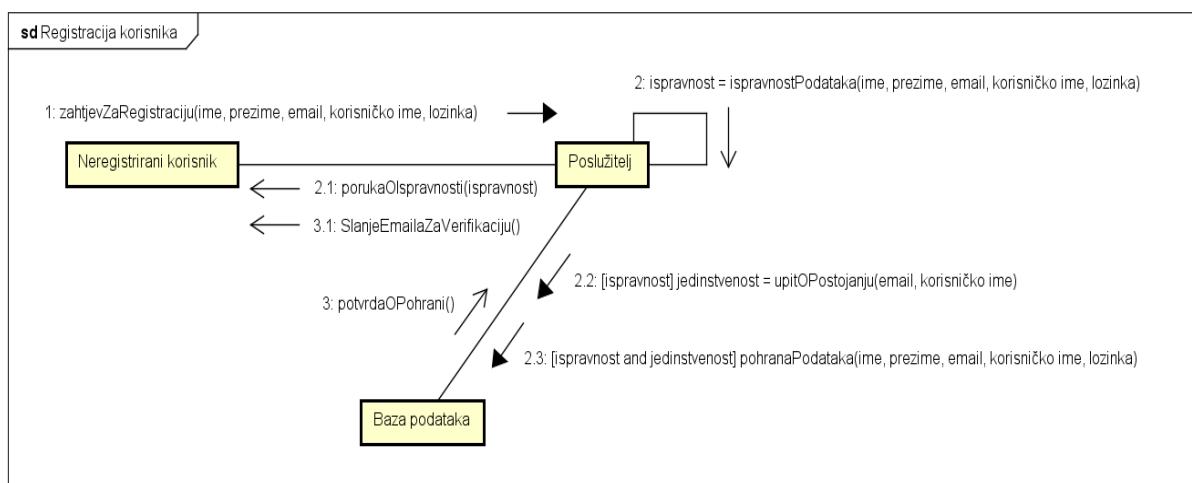


Slika 6.3.1 Dijagram objekata

6.4 Ostali UML dijagrami

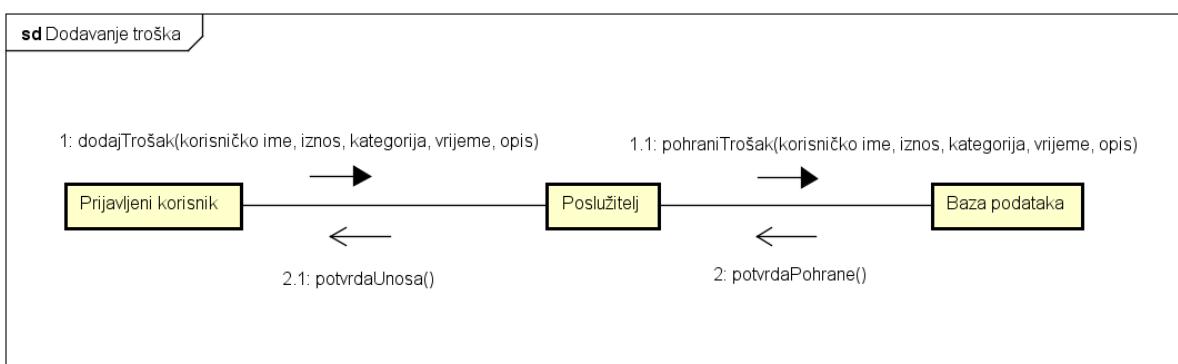
6.4.1 Komunikacijski dijagrami

Nakon upisivanja potrebnih podataka u formular za registraciju, korisnik zatraži registraciju. Poslužitelj provjerava ispravnost podataka i postojanje korisničkog imena ili email adrese u bazi podataka, ako podaci ne postoje stvara se novi registrirani korisnik. Korisniku se zatim šalje elektronička pošta u kojoj se nalazi ključ kojim će korisnik potvrditi svoju registraciju u sustav i omogućiti prijavu u sustav.



Slika 6.4.1 Komunikacijski dijagram registracije korisnika

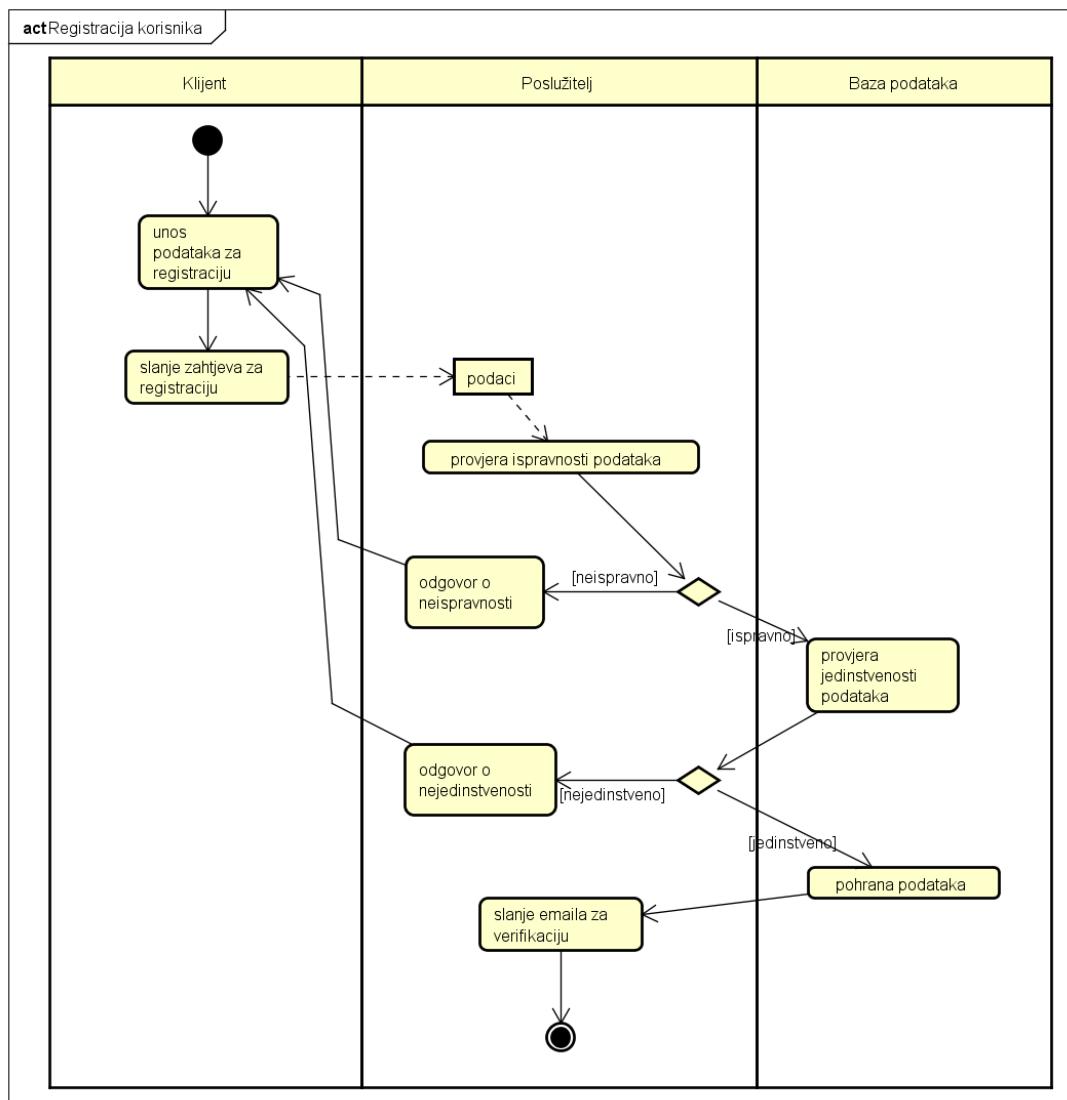
Korisnik šalje potrebne podatke sa zahtjevom za dodavanje troška. Poslužitelj zatim traži pohranu troška u bazi podataka.



Slika 6.4.2 Komunikacijski dijagram dodavanja troška

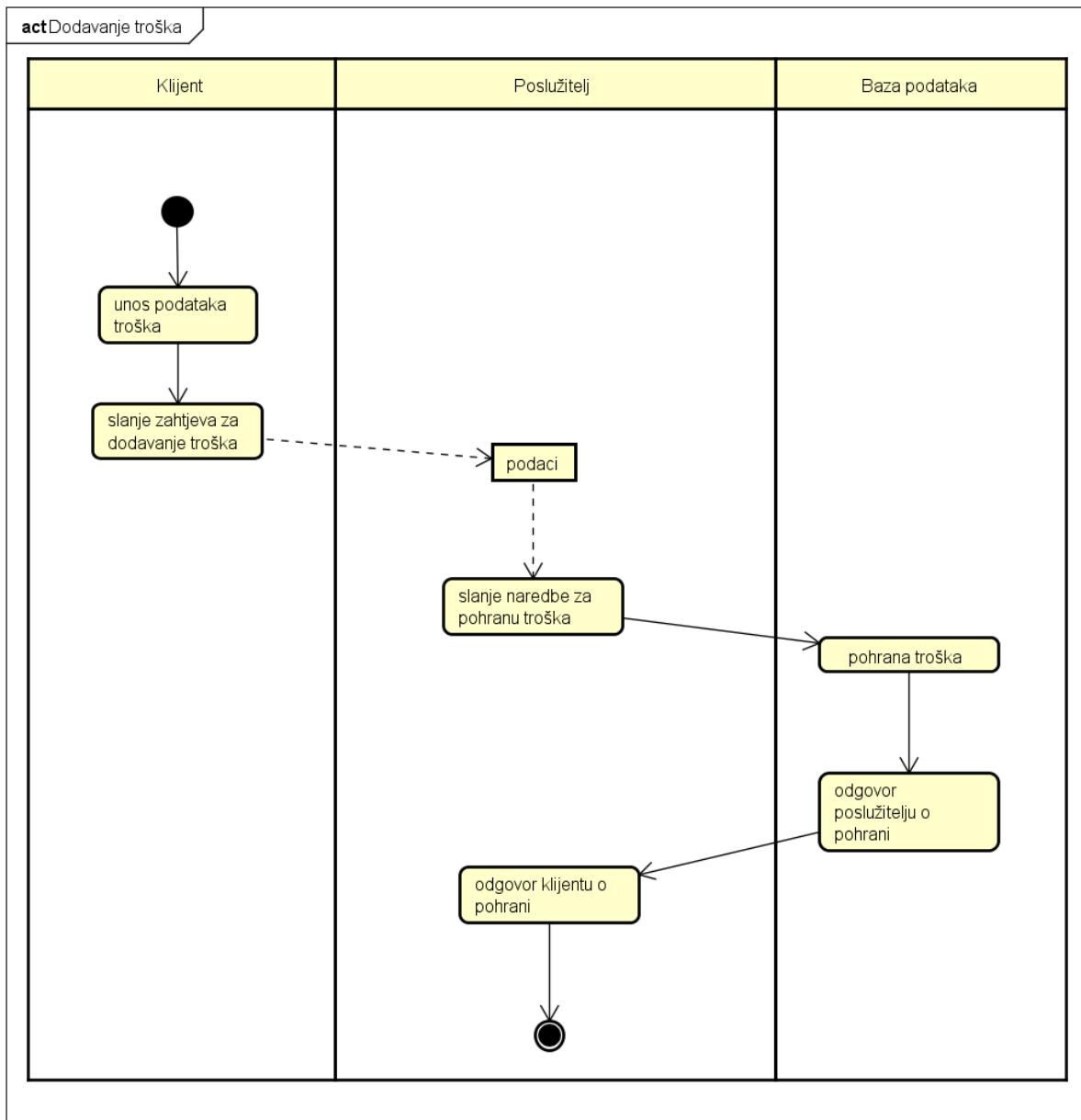
6.4.2 Dijagrami aktivnosti

Dijagram aktivnosti za registraciju korisnika podjelit će se u tri plivačke staze. Jedna plivačka staza je za klijenta, druga za poslužitelj, a treća za bazu podataka. Prva aktivnost se provodi na klijentu unošenjem podataka za registraciju. Podaci se šalju središnjem poslužitelju koji provodi verifikaciju podataka i koji zatim te podatke šalje bazi podataka. Baza podataka provjerava postoje li primljeni podaci već upisani u bazi i vraća odgovor središnjem poslužitelju. Ako podaci već postoje u bazi poslužitelj sastavlja i šalje informaciju korisniku, a ako podaci ne postoje u bazi oni se pohranjuju u bazu podataka. Baza podataka na kraju šalje obavijest poslužitelju da su podaci uspješno pohranjeni, pa poslužitelj sastavlja i šalje email za verifikaciju korisniku.



Slika 6.4.3 Dijagram aktivnosti Registracija korisnika

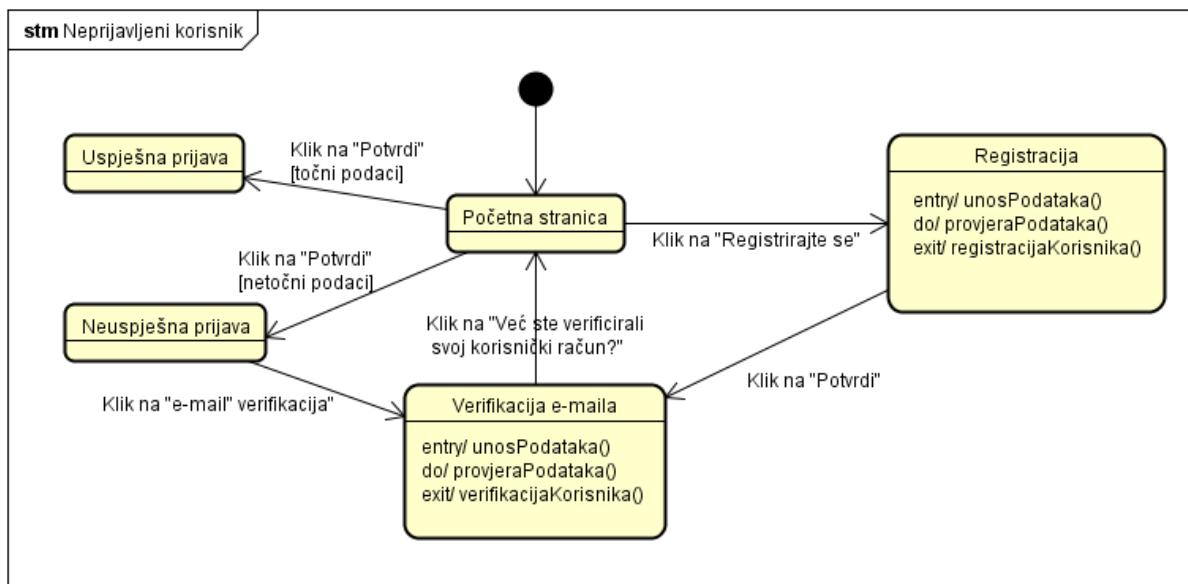
Dijagram aktivnosti za dodavanje troška će se također podijeliti u tri plivačke staze. Jedna plivačka staza je za klijenta, druga za poslužitelj, a treća za bazu podataka. Prva aktivnost se provodi na klijentu unošenjem podataka troška. Podaci se šalju središnjem poslužitelju koji naređuje bazi da ih pohrani. Baza podataka ih pohrani i vrati odgovor poslužitelju. Zatim poslužitelj obavijesti klijenta o pohrani.



Slika 6.4.4 Dijagram aktivnosti Dodavanja troška

6.4.3 Dijagram stanja

Dijagram stanja prikazuje stanja objekata te prijelaze između stanja temeljene na događajima. Prilikom dolaska na početnu stranicu, neprijavljeni korisnik ima mogućnost registracije u sustav, koja traži verifikaciju preko e-mail adrese. Ukoliko korisnik ima korisnički račun, i točno upiše podatke za prijavu, tada prelazi u stanje „Uspješna prijava“ koje se razmatra na dijagramu stanja prijavljenog korisnika.



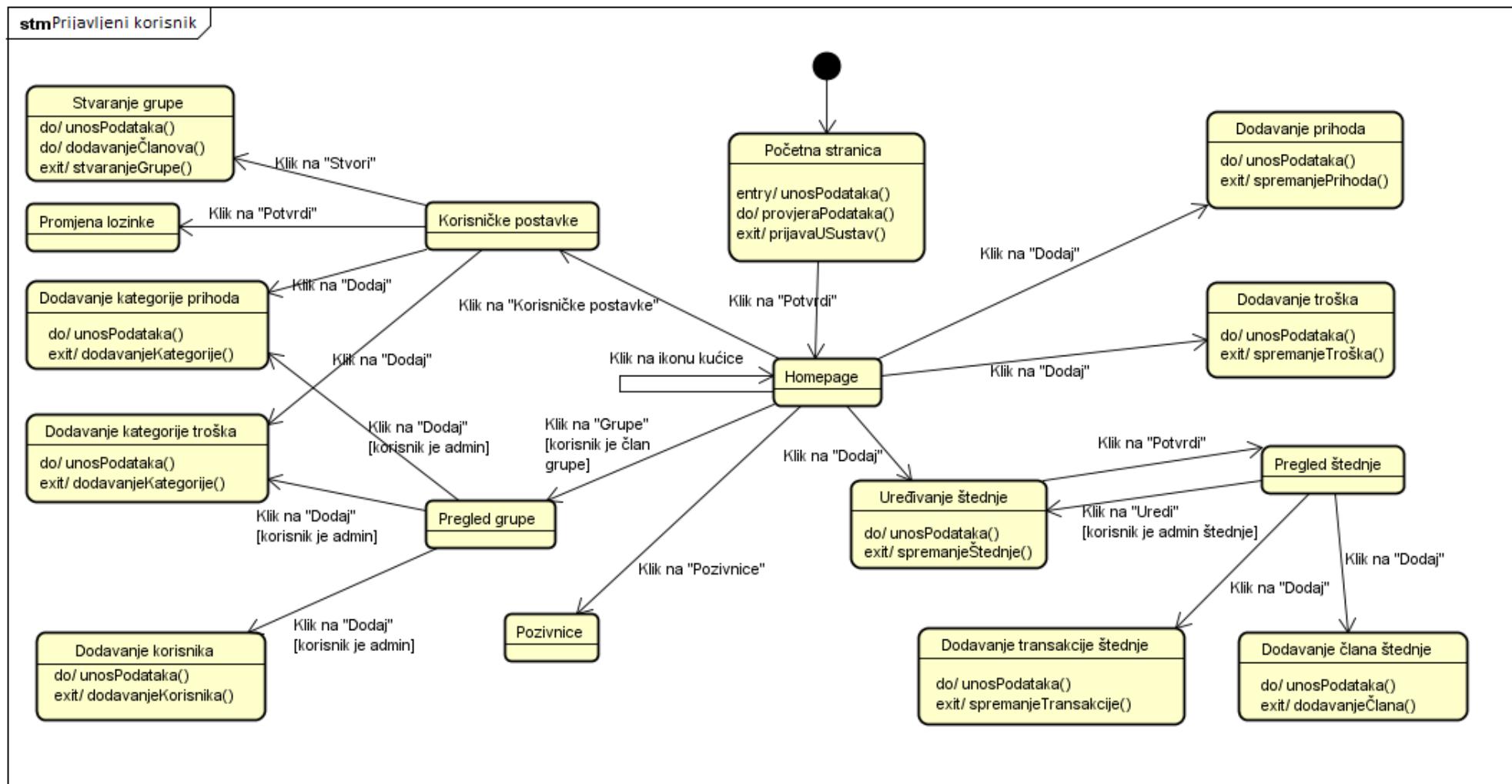
Slika 6.4.5 Dijagram stanja neprijavljenog korisnika

Nakon prijave u sustav korisnik dolazi na glavnu stranicu aplikacije (Homepage) u kojoj je glavni pregled obavijesti, troškova, prihoda i štednji. Zbog preglednosti, neki od mogućih prijelaza za prijavljenog korisnika nisu prikazani, na primjer, korisnik u svakom trenutku može kliknuti na kartice iz zaglavlja i prijeći u pripadajuća stanja („Grupe“, „Korisničke postavke“, „Pozivnice“, „Odjava“). Također, za svaki postojeći trošak i prihod, korisnik ima mogućnost uređivanja klikom na „uredi“ te brisanja klikom na „Obriši“.

U pregledima štednje i grupe, korisnik može kliknuti na „Napusti“ i tako prijeći na Homepage, a ukoliko je admin može ukloniti člana, promijeniti administratorski status te kliknuti na „Obriši“.

Na Homepage, korisnik može kliknuti na ime štednje u kojoj je član, te prijeći u stanje za uređivanje štednje.

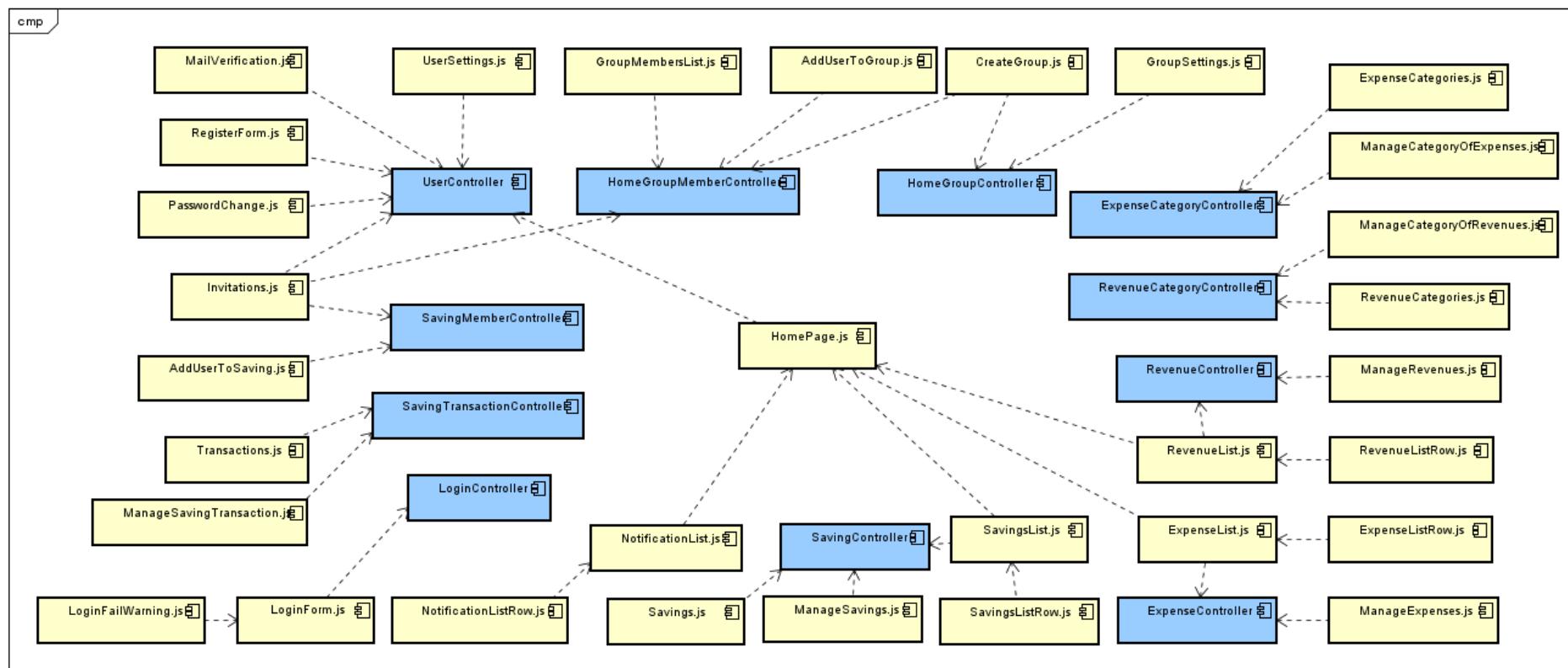
Klikom na pozivnice, korisniku se prikazuju pozivnice za grupe i štednje, korisnik može prihvatiti klikom na „Prihvati“ ili odbiti klikom na „Odbij“.



Slika 6.4.6 Dijagram stanja za prijavljenog korisnika

6.4.4 Dijagram komponenti

Dijagram komponenti opisuje međuovisnost komponenti i interne strukture sustava. Služi za vizualizaciju odnosa među komponentama sustava. Radi preglednosti su prikazane važnije ovisnosti između komponenti sustava. Ovisnost se upotrebljava kada jedna komponenta zahtijeva uporabu druge zbog ostvarenja svoje potpune funkcionalnosti.

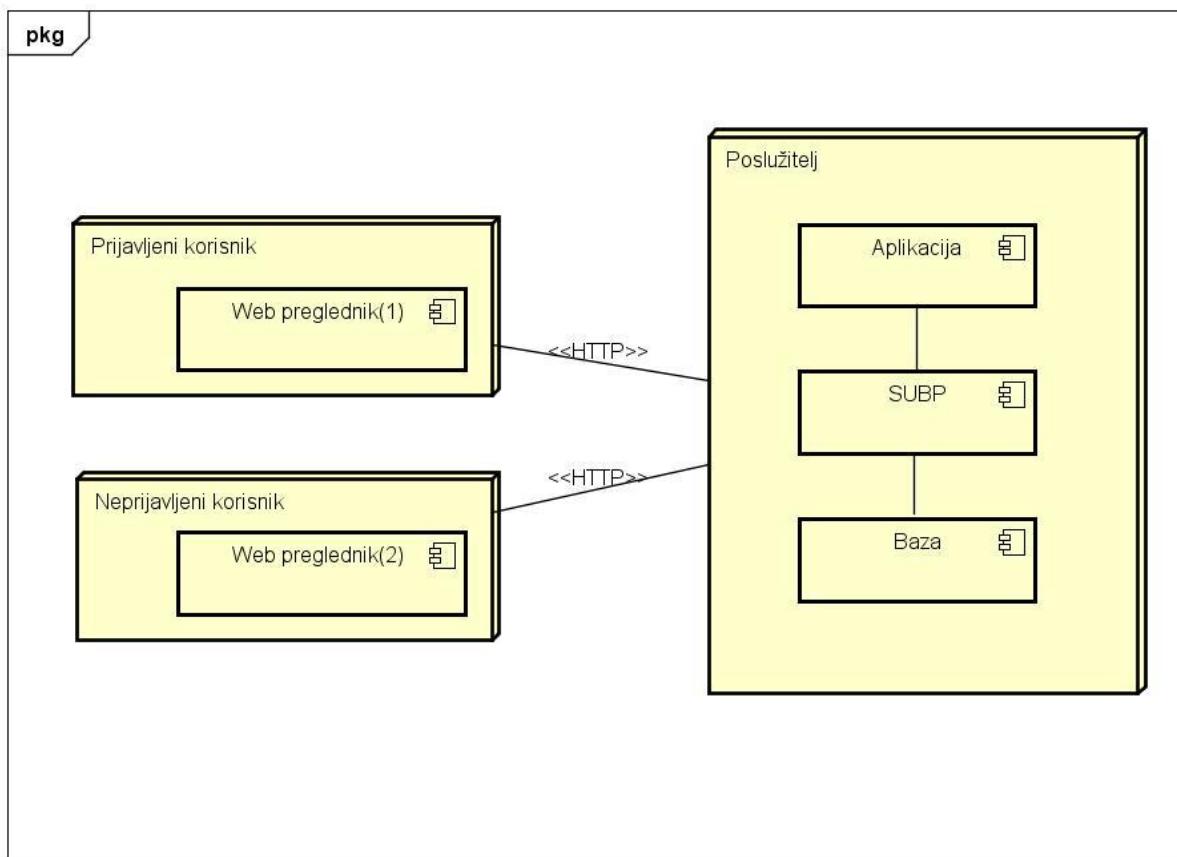


Slika 6.4.7 Dijagram komponenti

7. Implementacija i korisničko sučelje

7.1 Dijagram razmještaja

Na slici 7.1 prikazana je generalna topologija sustava. Sustav je baziran na arhitekturi "klijent – poslužitelj", a komunikacija između računala klijenta (prijavljeni i neprijavljeni korisnik) i poslužitelja odvija se preko HTTP veze. Aplikacija preko sustava za upravljanje bazom podataka komunicira s bazom podataka.



Slika 7.1.1 Dijagram razmještaja

7.2 Korištene tehnologije i alati

Za razvoj aplikacije korišten je programski jezik Java, odnosno Java Spring Framework za pozadinski dio, a jezik JavaScript, odnosno React.js za razvoj korisničkog sučelja. Lokalno testiranje obavljeno je korištenjem Apache Tomcata, a korišten je i sustav za kontrolu izmjena Git, odnosno GitLab, dok je za upravljanje bazom podataka korišten sustav PostgreSQL. Tijekom pisanja dokumentacije za izradu UML dijagrama korišten je alat Astah.

Apache Tomcat

- HTTP web poslužitelj s okruženjem za Javu
- <http://tomcat.apache.org/>

Astah

- alat za modeliranje UML dijagrama
- <http://astah.net/student-license-request>

Git

- sustav za kontrolu izmjena koda
- <https://git-scm.com/>

GitLab

- repozitorij za Git baziran na webu
- <https://gitlab.com/>

Java

- objektno orijentirani programski jezik
- <https://www.java.com/en/>

Java Spring Framework

- razvojni okvir za inverziju kontrole za Javu
- <https://spring.io/>

JavaScript

- interpretirani programski jezik
- <https://www.javascript.com/>

PostgreSQL

- sustav za upravljanje bazom podataka
- <https://www.postgresql.org/>

React

- biblioteka za izradu korisničkih sučelja za JavaScript
- <https://reactjs.org/>

SQL

- strukturirani upitni jezik korišten u relacijskim bazama podataka

7.3 Isječak programskog koda vezan za temeljnu funkcionalnost sustava

Sljedeći isječci koda predstavljaju primjere implementacije backend komponente projektne aplikacije koja je izvedena u razvojnem okviru Spring Boot.

Prvi isječak koda prikazuje razred TheGenericsPracenjeTroskova u kojem je implementirana main metoda kojom se pokreće aplikacija.

```
package hr.fer.opp.project;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@SpringBootApplication
public class TheGenericsPracenjeTroskova {

    @Bean
    public PasswordEncoder pswdEncoder() {
        return new BCryptPasswordEncoder();
    }

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

Naredni isječak koda prikazuje primjer razreda upravitelja koji je označen anotacijom `@RestController`. Razredi upravitelji implementiraju metode kojima se pristupa putem slanja zahtjeva na specifični URI te se njima ostvaruju funkcionalnosti aplikacije. Zahtjevi također mogu sadržavati tijelo poruke (`@RequestBody`) koje se metodama predaju kao argumenti. Upravitelj na zahtjev vraća odgovor koji je specifičan za svaku posebnu metodu. Anotacija `@Autowired` dohvaća instancu servisa.

```
@RestController
@RequestMapping("/savings")
public class SavingController {

    @Autowired
    private SavingService savingService;

    @Autowired
    private SavingTransactionService savingTransactionService;

    @Autowired
    private UserService userService;

    @GetMapping("")
    public List<Saving> listSavings() {
        return savingService.listAll();
    }

    @GetMapping("/{id}")
    @Secured("ROLE_USER")
    public Saving fetchSaving(@PathVariable("id") long id) {
        return savingService.fetchSaving(id);
    }

    @PostMapping("")
    @Secured("ROLE_USER")
    public ResponseEntity<Saving> createSaving(@RequestBody Saving saving) {
        User user = userService.fetchCurrentUser();
        Saving saved = savingService.createSaving(saving, user);
        return ResponseEntity.created(URI.create("/savings/" +
            saved.getSavingID())).body(saved);
    }
}
```

Ovaj isječak koda predstavlja jedan entitet, odnosno model koji je označen anotacijom `@Entity`. Entitet je definiran kao klasa koja se sastoji od atributa te metoda gettera i settera. Za svaki atribut je moguće definirati dodatne informacije i ograničenja u anotacijama koje se vežu uz te attribute.

Prema entitetima se stvaraju tablice u bazi podataka. Anotacijama je moguće definirati primarne ključeve, strane ključeve, jedinstvene attribute i razne druge značajke u bazi podataka.

```

@Entity
public class RevenueCategory {

    /**
     * Revenue category ID
     */
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long revenueCategoryID;

    /**
     * Name of the revenue category
     */
    @Column
    @NotNull
    private String name;

    /**
     * User that created revenue category
     */
    @ManyToOne
    @JoinColumn(name="user_ID", referencedColumnName="userID")
    private User user;

    /**
     * Group that uses this category - can be null
     */
    @ManyToOne
    @JoinColumn(name = "group_ID")
    private HomeGroup homeGroup;
}

```

Ovaj isječak koda prikazuje primjer sučelja koje je označeno anotacijom `@Service` kojom se definiraju metode koje će biti implementirane u klasi koja implementira ovo sučelje.

```

@Service
public interface ExpenseService {

    List<Expense> listAll();

    List<Expense> findByUser(User user);

    Optional<Expense> findById(long expenseID);

    List<Expense> findByExpenseCategory(ExpenseCategory expenseCategory);

    Expense fetchExpense(long expenseID);

    Expense createExpense(Expense expense);

    Expense updateExpense(Expense expense);

    Expense deleteExpense(long expenseID);

    void deleteExpenseCategoryForExpenses(long expenseCategoryID);
}

```

U ovom isječku koda je prikazan primjer implementacije servisa u obliku razreda koji implementira sučelje označeno anotacijom `@Service`. Razredi poput ovog su veza između upravitelja i repozitorija. U njima se odvija dohvata podataka iz repozitorija te njihova provjera i obrada.

```
@Service
public class ExpenseServiceJpa implements ExpenseService {

    @Autowired
    private ExpenseRepository expenseRepo;

    @Autowired
    private UserService userService;

    @Autowired
    private ExpenseCategoryService expenseCategoryService;

    @Autowired
    private HomeGroupService homeGroupService;

    @Override
    public List<Expense> listAll() {
        return expenseRepo.findAll();
    }

    @Override
    public List<Expense> findByUser(User user) {
        return expenseRepo.findByUser(user).orElseGet(ArrayList::new);
    }
}
```

Ovo sučelje je primjer implementacije repozitorija kojim se omogućuje komunikacija aplikacije i baze podataka. Ovakva sučelja su označena anotacijom `@Repository`.

```
@Repository
public interface UserRepository extends JpaRepository<User, Long> {

    int countByUsername(String username);

    int countByEmail(String email);

    Optional<User> findByUsername(String username);

    boolean existsByUsernameAndUserIDNot(String username, Long userID);

    Optional<User> findByEmail(String email);
}
```

7.4 Ispitivanje programskog rješenja

1. Ispitni slučaj: Registracija korisnika

Opis: U ovom ispitnom slučaju ispitan je pokušaj registracije novog korisnika.

Ulaz: Ime, prezime, email adresa, korisničko ime, lozinka, ponovljena lozinka.

Očekivani izlaz: Uspješna registracija korisnika i poslan verifikacijski email.

Izlaz: Ako je korisnik upisao sve potrebne informacije: korisničko ime i email adresu koji već ne postoje u bazi te lozinku i ponovljenu lozinku koje su identične, korisniku je na email adresu poslan mail s linkom za verifikaciju. Također, potrebno je da lozinka bude duga minimalno 5 znakova.

Rubni slučaj: Ako korisnik ne unese neku od potrebnih informacija aplikacija mu dojavljuje što treba učiniti. Također mu dojavljuje i grešku u slučaju uporabe imena koje je već u bazi i emaila koji je već u bazi ili je neispravnog formata. Također, ako lozinke nisu jednake ili im je duljina manja od 5 znakova, aplikacija mu dojavljuje pogrešku.

Registracija

Ime: Prezime:
Ime mora sadržavati samo slova abecede! Prezime mora sadržavati samo slova abecede!
Ime mora biti dugo barem dva znaka! Prezime mora biti dugo barem dva znaka!

E-mail adresa:
Neispravna e-mail adresa!

Korisničko ime:
Korisničko ime mora sadržavati barem 3 znaka!
Ovo je jedinstveno ime kojim ćete biti prepoznавани unutar aplikacije.

Lozinka:
Lozinka mora biti duga barem 5 znakova!

Ponovite lozinku:

POTVRDI [Već imate korisnički račun?](#)

Slika 7.4.1 Rubni slučaj: podaci neispravnog formata

Registracija

Ime: Prezime:

E-mail adresa: X

Korisnik s ovom e-mail adresom već postoji!

Korisničko ime:

Korisnik s ovim korisničkim imenom već postoji!
Ovo je jedinstveno ime kojim ćete biti prepoznavani unutar aplikacije.

Lozinka: X

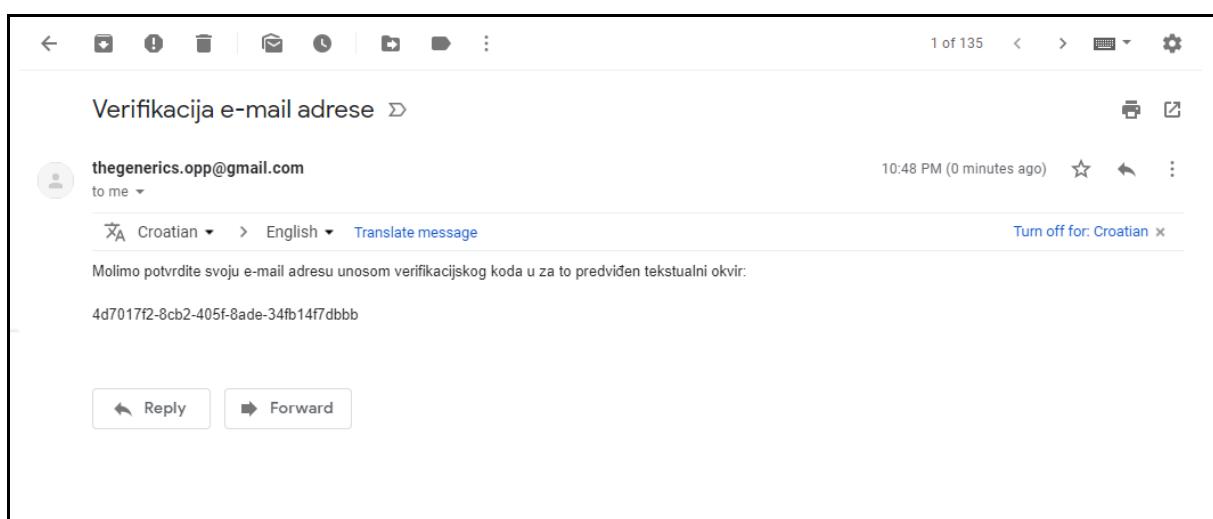
Ponovite lozinku: X

Lozinke se ne podudaraju!

POTVRDI

[Već imate korisnički račun?](#)

**Slika 7.4.2 Rubni slučaj: korisničko ime i e-mail adresa su zauzeti
i lozinke se ne podudaraju**



Slika 7.4.3 Verifikacijski e-mail nakon uspješne registracije

The screenshot shows a form titled "Verifikacija e-maila". It contains two text input fields: "Unesite korisničko ime:" (User name) and "Unesite verifikacijski kod:". Below the fields are two buttons: "POTVRDI" (Confirm) and a link "Već ste verificirali Vaš korisnički račun?".

Slika 7.4.4 Polje za unos verifikacijskog koda

2. Ispitni slučaj: Prijava korisnika

Opis: U ovom ispitnom slučaju ispitan je pokušaj prijave registriranog i verificiranog korisnika.

Ulaz: Korisničko ime i lozinka.

Očekivani izlaz: Uspješna prijava korisnika i preusmjeravanje na početni zaslon.

Izlaz: Potrebno je, naravno, da su korisničko ime i lozinka valjani, što ga preusmjerava na početni zaslon.

Rubni slučaj: Ako korisnik ne unese valjani par korisničkog imena i lozinku dobiva obavijest o nemogućnosti prijave.

The screenshot shows a login form titled "Prijava". It has two text input fields: "Korisničko ime:" (User name) containing "ivoivic" and "Lozinka:" (Password) containing "*****". Below the fields is an error message: "Prijava je neuspješna. Ukoliko niste verificirali Vaš korisnički račun možete to učiniti odabirom poveznice." At the bottom are two buttons: "POTVRDI" (Confirm) and a link "e-mail verifikacija" (Email verification).

Slika 7.4.5 Rubni slučaj: obavijest o neuspješnoj prijavi

Slika 7.4.6 Početni zaslon nakon uspješne prijave

3. Ispitni slučaj: Dodavanje kategorije prihoda ili troška

Opis: U ovom ispitnom slučaju ispitan je pokušaj dodavanja kategorije prihoda/troška.

Ulaz: unos imena kategorije prihoda ili imena kategorije troška, postotka potrošnje koji uzrokuje pojavu obavijesti te vremensko razdoblje koje se odnosi na obavijesti za trošak.

Očekivani izlaz: Uspješno kreiranje kategorije prihoda ili troška.

Izlaz: Kategorija troška ili prihoda je uspješno dodana te je korisnik preusmjeren na prikaz korisničkih postavki.

Rubni slučaj: Ponovljen ili neodabran unos parametra prihoda ili troška uzrokuje obavijest o nemogućnosti dodavanja kategorije.

Unesite podatke o kategoriji troška:

Ime kategorije troška:

Uredite postavke notifikacija koje su vezane za kategoriju troška

Postotak potrošnje koji uzrokuje notifikaciju:

Vremensko razdoblje na koje se odnose notifikacije

POTVRDI

Slika 7.4.7 Obrazac za dodavanje kategorije troška

Unesite podatke o kategoriji prihoda:

Ime kategorije prihoda:

POTVRDI

Slika 7.4.8 Obrazac za dodavanje kategorije prihoda

Unesite podatke o kategoriji troška:

Ime kategorije troška:

Uredite postavke notifikacija koje su vezane za kategoriju troška

Postotak potrošnje koji uzrokuje notifikaciju:

Vremensko razdoblje na koje se odnose notifikacije

Dodavanje kategorije neuspješno.

POTVRDI

Slika 7.4.9 Obavijest o neuspješnom dodavanju kategorije troška

Unesite podatke o kategoriji prihoda:

Ime kategorije prihoda:

Dodavanje kategorije neuspješno.

POTVRDI

Slika 7.4.10 Rubni slučaj: obavijest o neuspješnom dodavanju kategorije prihoda

4. Ispitni slučaj: Dodavanja prihoda

Opis: U ovom ispitnom slučaju ispitan je pokušaj dodavanja prihoda.

Ulaz: Unos datuma, kategorije, opisa te iznosa prihoda.

Očekivani izlaz: Uspješno kreiranje prihoda.

Izlaz: Prihod je uspješno dodan te je korisnik preusmjeren na početni zaslon.

Rubni slučaj: Neodabran unos parametara prihoda uzrokuje obavijest o neuspješnosti dodavanja prihoda.

Popis prihoda						
Datum	Korisnik	Kategorija	Opis	Iznos(HRK)	Uredi	Obrisi
2019-01-30	pmeric	obiteljska ljubaznost	džeparac	250	uredi	Obrisi
2019-01-10	pmeric	obiteljska ljubaznost	posjet baki	100	uredi	Obrisi

Slika 7.4.11 Lista prikaza prihoda i troškova

Unesite podatke o prihodu

Datum:

Opis:

Kategorija prihoda:

Iznos:

POTVRDI

Slika 7.4.12 Obrazac za dodavanje prihoda

Unesite podatke o prihodu

Datum:

Opis

Kategorija prihoda

Iznos:

POTVRDI

Neuspješno dodavanje prihoda.

Slika 7.4.13 Rubni slučaj: obavijest o neuspješnom dodavanju prihoda

5. Ispitni slučaj: Brisanje prihoda

Opis: U ovom ispitnom slučaju ispitan je pokušaj brisanja prihoda.

Ulaz: Odabir gumba za brisanje prihoda/troška na popisu prihoda.

Očekivani izlaz: Uspješno brisanje prihoda.

Izlaz: Prihod je obrisan te ga korisnik nije više u mogućnosti vidjeti na popisu.

Rubni slučaj: Nemogućnost brisanja prihoda koji je korisnik stvorio u slučaju da je pripadnik grupe, gumb za brisanje je automatski onemogućen.

Popis prihoda						
Datum	Korisnik	Kategorija	Opis	Iznos(HRK)	Uredi	Obriši
2019-01-31	vm97	grupnaKat1	prodaja zbirke pjesama	20000	uredi	Obriši
2019-01-22	pperic	grupnaKat1	prodaja meda	7350	uredi	Obriši

Slika 7.4.14 Brisanje prihoda pritiskom na gumb „Obriši“

6. Ispitni slučaj: Dodavanje štednje

Opis: U ovom ispitnom slučaju ispitan je pokušaj dodavanja štednje.

Ulaz: Odabir gumba za dodavanje štednje na popisu štednji početnog zaslona, odabir naziva štednje, ciljanog datuma, iznosa štednje te optionalnog dodavanja članova u štednju.

Očekivani izlaz: Uspješno dodavanje štednje.

Izlaz: Štednja je uspješno dodana, korisnicima označenima u obrascu poslan je zahtjev za pristupom navedenoj štednji te je korisnik preusmjeren na zaslon s informacijama i postavkama navedene štednje.

Rubni slučaj: Nemogućnost dodavanja štednje radi neunesenih ili pogrešno unesenih parametara štednje.

Popis štednji				
Datum početka	Ciljani datum	Trenutni iznos	Ciljani iznos	Ime
2019-01-12	2019-01-31	0	1000	moja prva štednja

Slika 7.4.15 Popis štednji

Unesite podatke o štednji

Naziv štednje:

Ciljani datum:

Iznos:

Dodaj člana štednje:

Dodaj

POTVRDI

Pozvani korisnici:

Korisnik	Poništi
----------	---------

Slika 7.4.16 Obrazac za dodavanje štednje

Unesite podatke o štednji

Naziv štednje:
putovanje u Amsterdam

Ciljani datum:
mm/dd/yyyy

Iznos:
5000

Dodaj člana štednje:
vm97 Dodaj

Dodavanje štednje neuspješno.

POTVRDI

Pozvani korisnici:

Korisnik	Poništi
vm97	Poništi

Slika 7.4.17 Rubni slučaj: obavijest o neuspješnom dodavanju štednje

7. Ispitni slučaj: Dodavanje transakcije štednje

Opis: U ovom ispitnom slučaju ispitan je pokušaj dodavanja transakcije štednje.

Ulag: Odabir gumba za dodavanje transakcije štednje na popisu transakcija štednje zaslona s informacijama o navedenoj štednji, odabir iznosa transakcije te tipa transakcije.

Očekivani izlaz: Uspješno dodavanje transakcije štednje.

Izlaz: Transakcija štednje je uspješno dodana te je korisnik preusmjeren na zaslon s informacijama i postavkama navedene štednje.

Rubni slučaj: Nemogućnost dodavanja transakcije štednje radi neunesenih ili pogrešno unesenih parametara transakcije.

The form has fields for 'Iznos:' (Amount) containing '100', and 'Vrsta transakcije:' (Type of transaction) with a dropdown menu showing 'Odaberite vrstu transakcije', 'Odaberite vrstu transakcije', 'Uplata' (Deposit), and 'Isplata' (Withdrawal). The 'Uplata' option is highlighted.

Slika 7.4.18 Obrazac za dodavanje transakcije štednje

The form has fields for 'Iznos:' (Amount) containing '100', and 'Vrsta transakcije:' (Type of transaction) with a dropdown menu showing 'Odaberite vrstu transakcije'. Below the form is a button labeled 'POTVRDI' (Confirm). A message at the bottom states 'Dodavanje transakcije štednje neuspješno.' (Saving deposit transaction failed).

Slika 7.4.19 Rubni slučaj: neuspješno dodavanje transakcije štednje

8. Ispitni slučaj: Postupak stvaranja grupe

Opis: U ovom ispitnom slučaju ispitan je pokušaj stvaranja grupe.

Ulaz: Odabir gumba za stvaranje grupe unutar korisničkih postavki, odabir naziva grupe te dodavanja članova grupe (opcionalno).

Očekivani izlaz: Uspješno stvaranje grupe.

Izlaz: Grupa je uspješno stvorena, poslane su pozivnice za pristup grupi odabranim članovima te je korisnik preusmjeren na zaslon s informacijama i postavkama navedene grupe.

Rubni slučaj: Nemogućnost stvaranja grupe radi neunesenog imena grupe i nemogućnost slanja pozivnice članu u slučaju da navedeni ne postoji.

Unesite podatke o grupi:

Naziv grupe:

Dodaj člana grupe:

POTVRDI

Pozvani korisnici:

Korisnik	Poništi

Slika 7.4.20 Obrazac za stvaranje grupe

Unesite podatke o grupi:

Naziv grupe:

Dodaj člana grupe:

Dodavanje grupe neuspješno.

POTVRDI

Pozvani korisnici:

Korisnik	Poništi
vm97	<input type="button" value="Poništi"/>

Slika 7.4.21 Rubni slučaj: neuspješno stvaranje grupe

Unesite podatke o grupi:

Naziv grupe:
TheGenerics

Dodaj člana grupe:
peroperic Dodaj

Korisnik s ovim korisničkim imenom ne postoji.

POTVRDI

Pozvani korisnici:

Korisnik	Poništi
vm97	Poništi

Slika 7.4.22 Rubni slučaj: slanje pozivnice nepostojećem korisniku

9. Ispitni slučaj: Napuštanje grupe

Opis: U ovom ispitnom slučaju ispitan je pokušaj napuštanja grupe.

Ulaz: Odabir gumba za napuštanje grupe unutar grupnih postavki.

Očekivani izlaz: Uspješno napuštanje grupe.

Izlaz: Grupa je uspješno napuštena te je korisnik preusmjeren na zaslon s korisničkim postavkama.

Rubni slučaj: -

Praćenje troškova �

Grupa Korisničke postavke Pozivnice Odjava

<p>Promijeni lozinku</p> <p>Nova lozinka: <input type="text"/></p> <p>Ponovi novu lozinku <input type="text"/></p> <p>Potvrdi</p>	<p>Grupa: NAPUSTI</p>
--	--

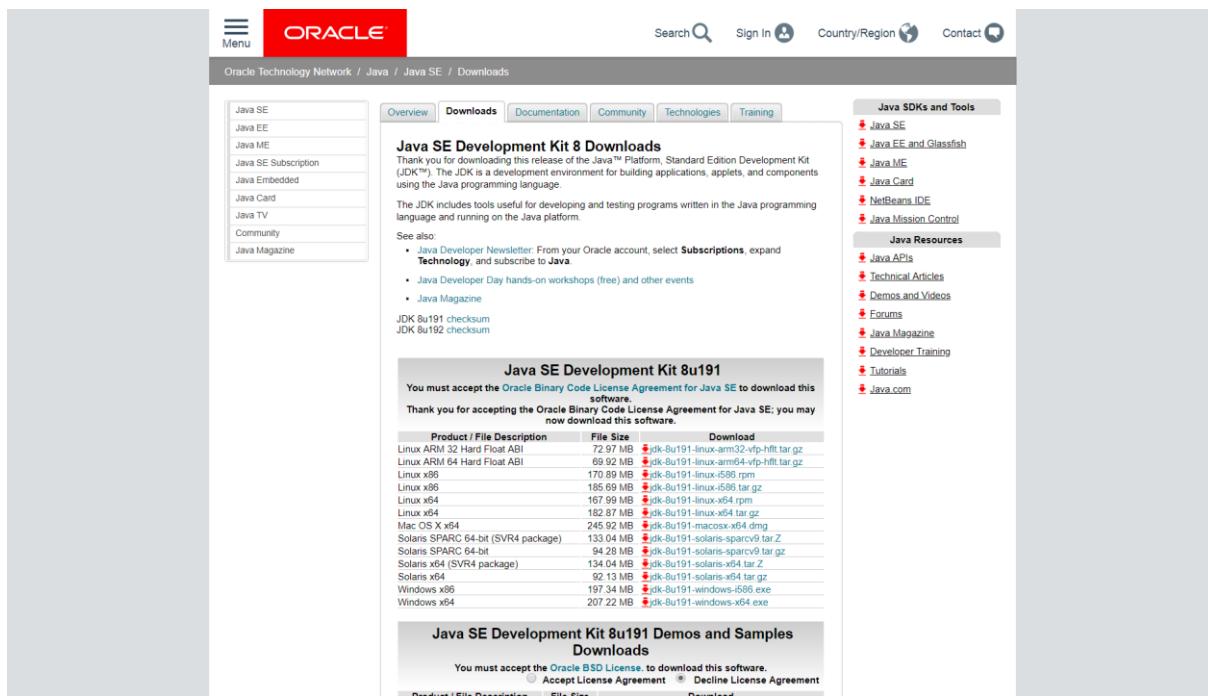
Slika 7.4.23 Napuštanje grupe

7.5 Upute za instalaciju

7.5.1 Upute za instalaciju Java Development Kita (JDK)

Osnovni je preduvjet za rad s Javom, pa tako i Java Spring Frameworkom, je instalirana inačica Java Development Kita (JDK-a). Instalacijski paket JDK-a može se preuzeti na internetskoj stranici:

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>



Slika 7.5.1 Preuzimanje Java Development Kita

Potrebno je pročitati i prihvati odredbe ugovora o korištenju i odabrati odgovarajuću inačicu JDK-a ovisno o operacijskom sustavu. Nakon što se preuzimanje dovrši, potrebno je obaviti instalaciju.



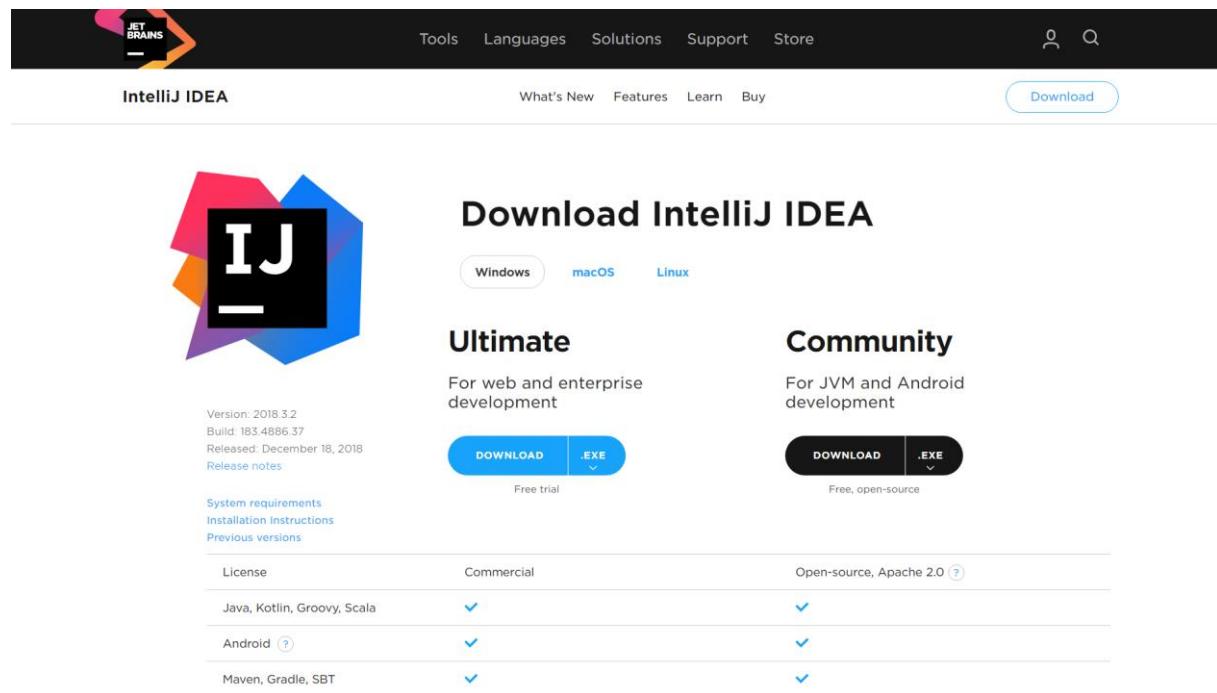
Slika 7.5.2 Instalacija Java Development Kita

Svaki korisnik prilikom instalacije može sebi prilagoditi postavke instalacije, ali je preporučeno držati se predloženih postavki i redom odabratи "Next" te zatim opet "Next". Potrebno je pričekati da se instalacija dovrši i pritisnuti gumb "Close".

Na kraju, potrebno je i postaviti globalne varijable PATH i JAVA_HOME. Njima se može pristupiti tako da se u glavnom izborniku pristupi varijablama okruženja.

7.5.2 Upute za instalaciju integrirane razvojne okoline (IDE)

Za uređivanje Java koda potrebno je koristiti uređivač. Preporučuje se koristiti integriranu razvojnu okolinu (IDE). Jedna od njih je IntelliJ IDEA koja je dostupna za preuzimanje na sljedećoj povezniци: <https://www.jetbrains.com/idea/download/>



Slika 7.5.3 Preuzimanje razvojne okoline IntelliJ

Prije početka preuzimanja potrebno je odabrati operacijski sustav na koji će se instalirati i inačicu koju se želi preuzeti: "Ultimate" ili "Community". Nakon što se preuzimanje dovrši, provodi se instalacija.

7.5.3 Upute za instalaciju podrške za rad s bazom podataka

Za rad i upravljanje s bazom podataka potrebno je koristiti sustav za upravljanje bazom podataka (SUBP). SUBP koji je korišten za izradu aplikacije je PostgreSQL, a može ga se preuzeti na poveznici: <https://www.postgresql.org/download/>

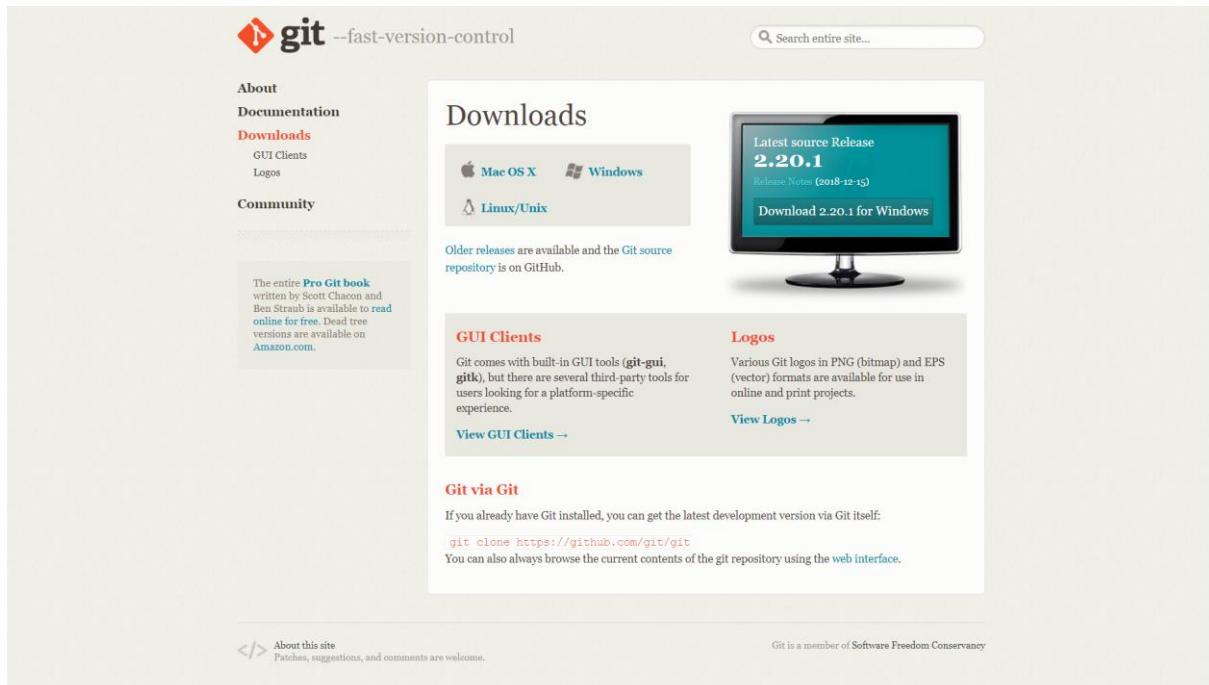
The screenshot shows the PostgreSQL download page. At the top, there's a navigation bar with links for Home, About, Download, Documentation, Community, Developers, Support, Donate, and Your account. A search bar is also at the top right. Below the navigation, a banner indicates '8th November 2018: PostgreSQL 11.1, 10.6, 9.6.11, 9.5.15, 9.4.20, and 9.3.25 Released!'. The main content area has a 'Quick Links' sidebar on the left with links for Downloads (Binary, Source), Software Catalogue, and File Browser. The main content area is titled 'Downloads' with a download icon. It features a section for 'PostgreSQL Core Distribution' which states: 'The core of the PostgreSQL object-relational database management system is available in several source and binary formats.' It then branches into 'Binary packages' and 'Source code'. The 'Binary packages' section lists supported operating systems: BSD (FreeBSD, OpenBSD), Linux (Red Hat family Linux, Debian, Ubuntu, SuSE, OpenSuSE, Other Linux), macOS, Solaris, and Windows. The 'Source code' section notes that source code can be found in the file browser or directly from git.postgresql.org. Below this, a section for 'Beta/RC Releases and development snapshots (unstable)' is mentioned, stating that it contains source code and binary packages for testing purposes only.

Slika 7.5.4 Preuzimanje PostgreSQL-a

Na stranici je potrebno odabrati operacijski sustav nakon čega treba odabrati i inačicu sustava koju se želi instalirati. Preporučuje se odabrati najnoviju dostupnu inačicu. Po završetku preuzimanja potrebno je pokrenuti i dovršiti instalaciju. Prilikom instalacije potrebno je postaviti korisnika i lozinku za bazu.

7.5.4 Upute za instalaciju Gita

Sustav za kontrolu izmjena Git može se preuzeti na poveznici: <https://git-scm.com/downloads>. Potrebno je odabratи operacijski sustav i preuzimanje započinje automatski.



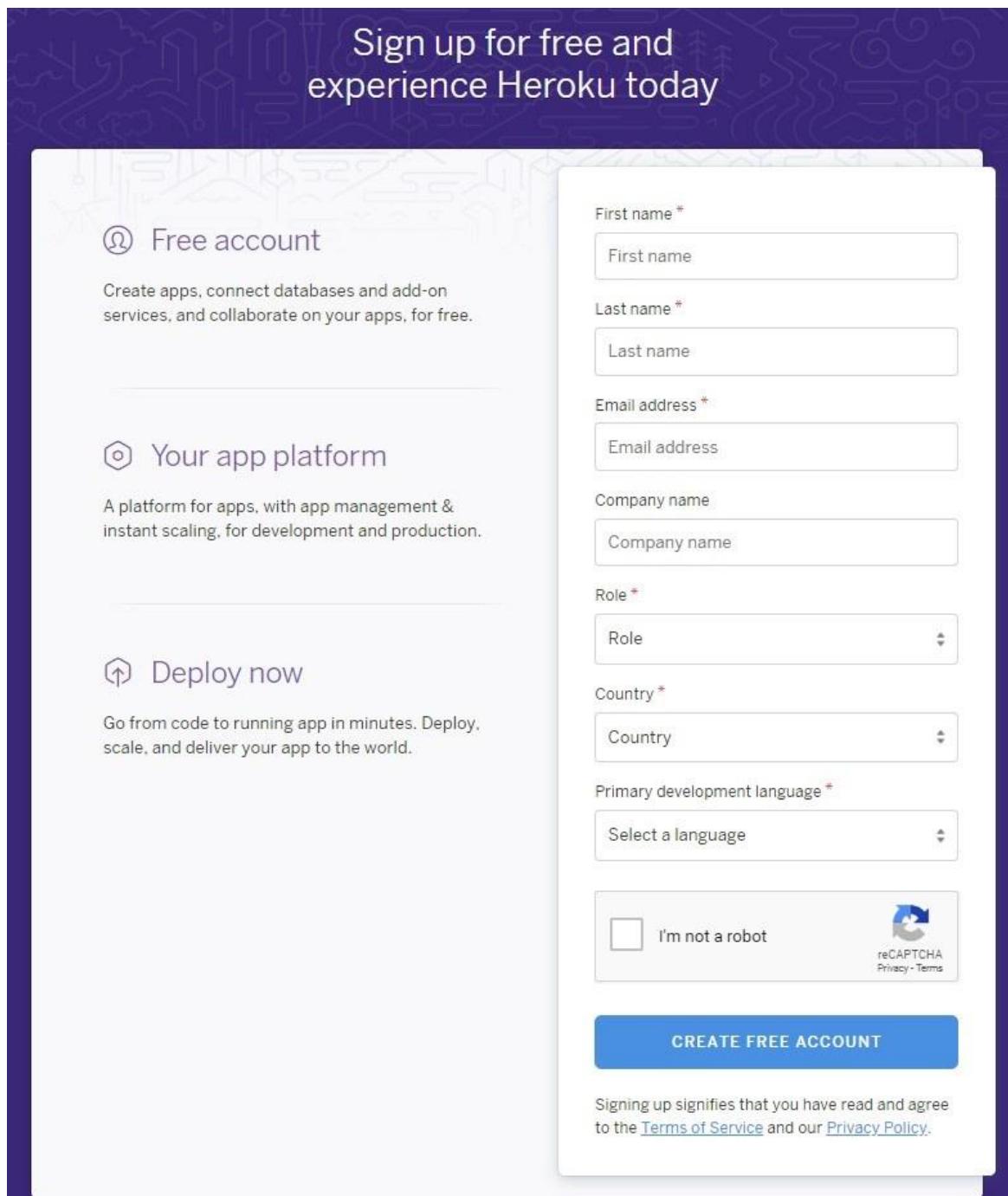
Slika 7.5.5 Preuzimanje Git-a

Nakon što je preuzimanje dovršeno, potrebno je pokrenuti i dovršiti instalaciju, a potom pokrenuti lјusku Gita (Git Bash). Prije korištenja Gita potrebno je podesiti postavke pomoću lјuske:

```
git config --global user.name "Korisnicko ime"  
git config --global user.email email.adresa@mail.com
```

7.5.5 Upute za postavljanje aplikacije na poslužitelj

Projektna aplikacija je postavljena na besplatni poslužitelj Heroku koji omogućuje integraciju aplikacije sa sustavom za upravljanje bazom podataka PostgreSQL. Za korištenje Heroku usluga je potrebna registracija.



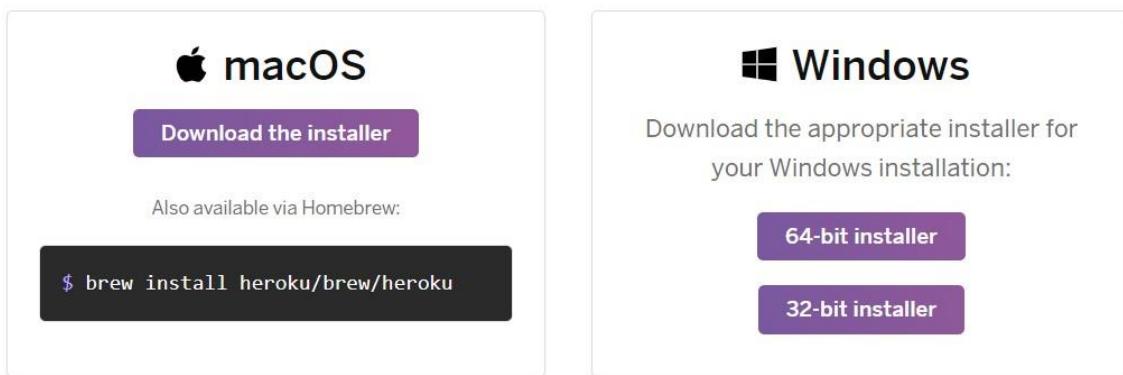
Slika 7.5.6 Registracija na Heroku

Nakon registracije potrebno je preuzeti i instalirati HerokuCLI paket kako bi se Heroku usluge mogle koristiti iz konzole.

Download and install

! The Heroku CLI requires [Git](#), the popular version control system. If you don't already have Git installed, complete the following before installing the CLI:

- [Git installation](#)
- [First-time Git setup](#)



Slika 7.5.7 Preuzimanje HerokuCLI-a

Nakon instalacije, u konzoli je potrebno unijeti:

```
heroku login
```

Nakon toga je potrebno inicijalizirati git repozitorij u slučaju da prilikom izrade aplikacije nije korišten sustav Git. To je moguće napraviti ovim naredbama.

```
cd opp-pracenje-troskova/  
git init  
heroku git:remote -a opp-pracenje-troskova
```

Zatim je potrebno napraviti prvi commit te zatim postaviti aplikaciju na poslužitelj.

```
git add .  
git commit -am "Initial commit"  
git push heroku master
```

Ukoliko već postoji git repozitorij onda je dovoljno unijeti samo jednu naredbu.

```
heroku git:remote -a opp-pracenje-troskova
```

Kako bi se dodala podrška za PostgreSQL potrebno je u pom.xml datoteku aplikacije dodati sljedeći isječak koda.

```
<dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
</dependency>
```

Zatim je potrebno dodati u application.properties dodati sljedeće kako bi aplikacija imala informacije o korištenoj bazi podataka.

```
spring.datasource.driverClassName=org.postgresql.Driver
spring.datasource.maxActive=10
spring.datasource.maxIdle=5
spring.datasource.minIdle=2
spring.datasource.initialSize=5
spring.datasource.removeAbandoned=true
spring.datasource.jdbc-url=${JDBC_DATABASE_URL}
spring.jpa.hibernate.ddl-auto=update
```

Također potrebno je stvoriti i razred DatabaseConfig kako bi se konfigurirala baza podataka.

```
@Configuration
public class DatabaseConfig {
    @Bean
    @Primary
    @ConfigurationProperties(prefix = "spring.datasource")
    public DataSource dataSource() {
        return DataSourceBuilder.create().build();
    }
}
```

Zatim se u konzoli unose ove naredbe.

```
heroku addons:create heroku-postgresql
heroku config
heroku pg
```

Aplikacija se na kraju može otvoriti putem URI-a koji se dobije kao rezultat uspješnog izvođenja prikazanih naredbi.

7.6 Korisničke upute

Za korištenje aplikacije za praćenje troškova potrebno je biti povezan s Internetom. Korištenjem web preglednika (prema odabiru korisnika, na računalu ili mobitelu) korisnik pristupa pristupnoj stranici. Registrirani korisnici inicijalno imaju jednake mogućnosti, koje se mijenjaju ovisno o ulozi u grupi, što će biti naknadno objašnjeno.

Napomena: neregistrirani i neprijavljeni korisnici ne mogu pristupiti praćenju troškova.

The screenshot shows a login form titled 'Prijava'. It contains two input fields: 'Korisničko ime:' with placeholder 'Unesite korisničko ime' and 'Lozinka:' with placeholder 'Unesite lozinku'. Below the fields are two buttons: a dark grey 'POTVRDI' button and a blue 'Registrirajte se' button.

Slika 7.6.1 Pristupna stranica web aplikacije

Neregistrirani korisnik:

Pritiskom na opciju „*Registrirajte se*“ korisnik je s pristupnog zaslona preusmjeren na obrazac za registraciju.

The screenshot shows a registration form titled 'Registracija'. It includes several input fields: 'Ime:' and 'Prezime:' with placeholder 'Ime' and 'Prezime'; 'E-mail adresa:' with placeholder 'Email'; 'Korisničko ime:' with placeholder 'Korisničko ime' and a note below stating 'Ovo je jedinstveno ime kojim ćete biti prepoznavani unutar aplikacije.'; 'Lozinka:' and 'Ponovite lozinku:' both with placeholder 'Lozinka'; and a note below the second password field stating 'Ponovite lozinku:'. At the bottom are two buttons: a dark grey 'POTVRDI' button and a blue 'Već imate korisnički račun?' link.

Slika 7.6.2 Obrazac za registraciju korisnika

Ispunivši obrazac (ispravnim) podacima korisniku je poslan verifikacijski e-mail unutar kojeg se nalazi verifikacijski kod te je preusmjeren na obrazac za verifikaciju računa.

Verifikacija e-maila

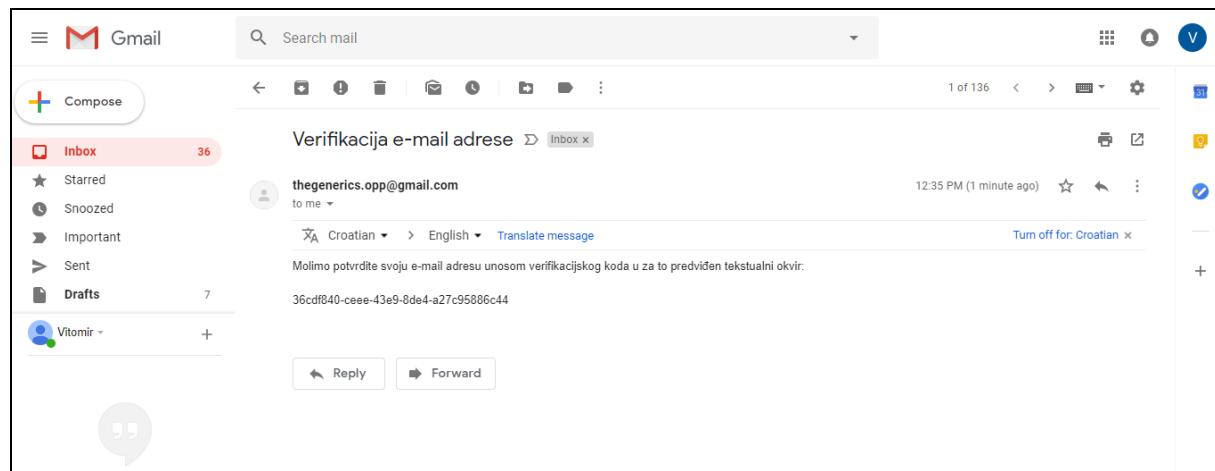
Unesite korisničko ime:

Unesite verifikacijski kod:

POTVRDI

[Već ste verificirali Vaš korisnički račun?](#)

Slika 7.6.3 Obrazac za verifikaciju korisničkog računa



Slika 7.6.4 E-mail s verifikacijskim kodom

Registrirani i verificirani korisnik:

Nakon unesenog korisničkog imena i ispravnog verifikacijskog koda u obrazac, korisnik je preusmjeren natrag na obrazac za prijavu. Unošenjem ispravnog korisničkog imena i lozinke, korisnik dolazi na početnu stranicu s prikazom navigacijske trake, stanja računa, eventualnog članstva u grupi, odabira razdoblja prikaza obavijesti, prihoda, troškova te štednji. Navigacijom niže po stranici korisnik je u mogućnosti vidjeti popise vlastitih (ili grupnih, ovisno o statusu pripadnosti grupi) obavijesti, troškova, prihoda te štednji.

The screenshot shows the application's main dashboard. At the top, there is a header bar with the title "Praćenje troškova" and a house icon. On the right side of the header are links for "Grupa", "Korisničke postavke", "Pozivnice", and "Odjava". Below the header, there are two boxes: one showing "Stanje računa: 0 kn" and "Grupa: privatni korisnik", and another for selecting a time period with a dropdown menu set to "cijelo vrijeme". A large central area is titled "Obavijesti" and contains a table with columns for "Datum" and "Poruka".

Slika 7.6.5 Početni zaslon nakon prijave

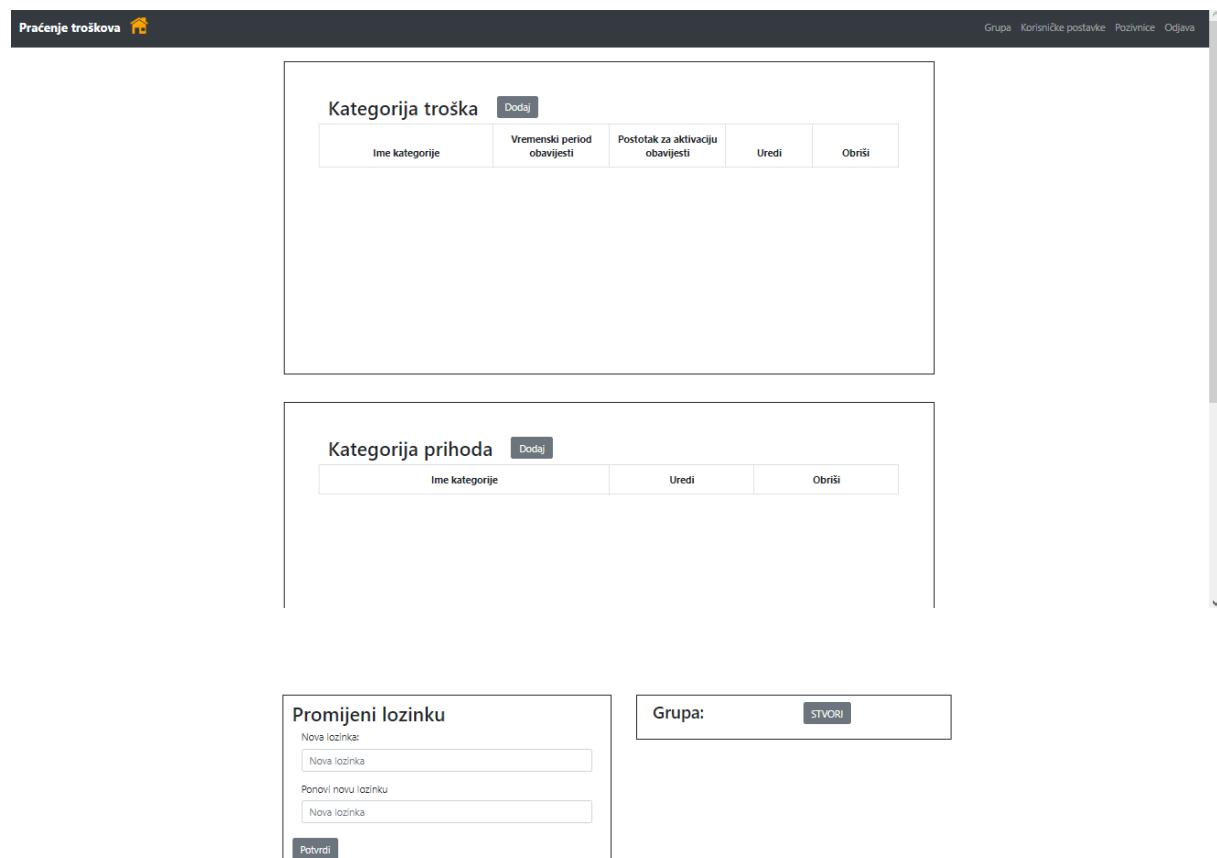
The image displays three separate sections of the application's interface, each with its own header and table:

- Popis prihoda**: Shows a table with columns: Datum, Korisnik, Kategorija, Opis, Iznos(HRK), Uredi, and Obrisi.
- Popis troškova**: Shows a table with columns: Datum, Korisnik, Kategorija, Opis, Iznos(HRK), Uredi, and Obrisi.
- Popis štednji**: Shows a table with columns: Datum početka, Ciljani datum, Trenutni iznos, Ciljani iznos, and Ime.

Slika 7.6.6 i Slika 7.6.7 Ostatak početnog zaslona dobiven navigacijom niz stranicu

Stvaranje kategorija troškova/prihoda:

Prije samog dodavanja prihoda ili troškova korisnik mora stvoriti jednu ili više kategorija prihoda/troškova, što može učiniti odabirom opcije „Korisničke postavke“ na navigacijskoj traci, čime biva preusmijeren na korisničke postavke s popisom kategorija prihoda i troškova, obrascem za promjenu lozinke te opcijama za stvaranje ili brisanje grupe (ovisno o statusu članstva).



Slika 7.6.8 i Slika 7.6.9 Prikaz korisničkih postavki

Pritiskom na gumb za stvaranje kategorije prihoda/troška, korisnik je preusmijeren na obrazac za stvaranje željene kategorije.

Unesite podatke o kategoriji troška:

Ime kategorije troška:

Uredite postavke notifikacija koje su vezane za kategoriju troška

Postotak prosječne potrošnje u zadanom vremenskom razdoblju koji uzrokuje stvaranje obavijesti:

Vremensko razdoblje na koje se odnose notifikacije

POTVRDI

Slika 7.6.10 Obrazac za stvaranje kategorije troška

Unesite podatke o kategoriji prihoda:

Ime kategorije prihoda:

POTVRDI

Slika 7.6.11 Obrazac za stvaranje kategorije prihoda

Napomena: Korisnik nije u mogućnosti stvoriti kategoriju istog imena više puta.

Ispunivši (ispravno) obrazac korisnik je preusmjeren natrag u korisničke postavke, ovaj put s vidljivom dodanom kategorijom na popisu.

Kategorija troška				
Ime kategorije	Vremenski period obavijesti	Postotak za aktivaciju obavijesti	Uredi	Obrisi
zabava	6 mjeseci	30	uredi	Obrisi

Kategorija prihoda		
Ime kategorije	Uredi	Obrisi
instrukcije iz matematike	uredi	Obrisi

Slika 7.6.12 Korisničke postavke nakon stvorenih kategorija troška i prihoda

Kategorije troškova/prihoda moguće je pregledati, urediti (čime je korisnik prebačen na obrazac za izmjenu kategorije) te obrisati.

Stvaranje troškova/prihoda:

U slučaju da korisnik želi stvoriti prihod ili trošak, potrebno se putem navigacijske trake preusmjeriti na početni zaslon te odabratи gumb „Dodaj“ pri vrhu popisa troškova/prihoda, čime korisnik pristupa obrascu za stvaranje prihoda/troška.

Unesite podatke o prihodu

Datum:

Opis

Kategorija prihoda

Iznos:

POTVRDI

Slika 7.6.13 Obrazac za stvaranje prihoda

Unesite podatke o trošku

Datum:

Opis

Kategorija troška

Iznos:

POTVRDI

Slika 7.6.14 Obrazac za stvaranje troška

Ispunjavanjem obrasca, korisnik je ponovno preusmjeren na početni zaslon te se stvoreni trošak/prihod pojavljuje na popisu i korisničko stanje računa ažurira.

Popis prihoda						
Datum	Korisnik	Kategorija	Opis	Iznos(HRK)	Uredi	Obrisi
01.01.2019.	ivoivic	plaća	plaća za siječanj	5000	uredi	Obrisi

Popis troškova						
Datum	Korisnik	Kategorija	Opis	Iznos(HRK)	Uredi	Obrisi
03.01.2019.	ivoivic	obuća	tenisice	500	uredi	Obrisi

Slika 7.6.15 Popis prihoda i troškova nakon stvaranje prihoda i troška

Prihode i troškove s popisa moguće je uređivati i brisati na identičan način kao i kategorije prihoda/troškova , poseban je slučaj kad je korisnik član grupe, tada nije u mogućnosti brisati prihode i troškove koje nije sam stvorio.

Stvaranje štednji:

Na početnom zaslonu moguće je pristupiti i popisu štednji, koje mogu biti stvorene na sličan način kao prihodi i troškovi. U obrascu za dodavanje štednje potrebno je upisati naziv štednje, ciljani datum ostvarenja cilja (zaključenja štednje) te ciljani iznos. Također možemo dodati postojećeg člana u štednju, čime mu je poslana pozivnica koju može (i ne mora) prihvati.

Slika 7.6.16 Obrazac za stvaranje štednje

Nakon potvrđivanja ispravno unesenih podataka, korisnik je preusmjeren na postavke upravo stvorene štednje, u kojima može vidjeti podatke o štednji poput popisa transakcija (uplata/isplata, stvaraju se na sličan način kao sve gore navedene kategorije), popisa članova štednje te postavki navedene štednje (mogućnosti brisanja i uređivanja), koje su omogućene samo ako je korisnik postavljen za administratora unutar štednje (stvaranjem štednje korisnik automatski postaje administrator). Korisnik također može napusnit štednju.

The screenshot displays two main sections of a web application:

Transakcije (Transactions)

Korisnik	Datum	Vrsta	Iznos	Uredi	Obrisi
peroperic	16.01.2019.	Upłata	200	Uredi	Obrisi

Popis članova (Member List)

Korisničko ime	Ukloni člana	Postavi za admina
peroperic	Obrisi	<input checked="" type="checkbox"/>

At the bottom of the page, there are three buttons: **Uredi**, **Napusti štednju**, and **Obrisi štednju**.

Slika 7.6.17 i Slika 7.6.18 Informacije o štednji i postavke štednje

Navedenim informacijama i postavkama štednje može se pristupiti i odabirom željene štednje na popisu štednji početnog zaslona.

Popis štednji				
Datum početka	Ciljni datum	Trenutni iznos	Ciljni iznos	Ime
14.01.2019.	25.01.2019.	200	5000	stednja

Slika 7.6.19 Popis štednji početnog zaslona nakon stvorene štednje

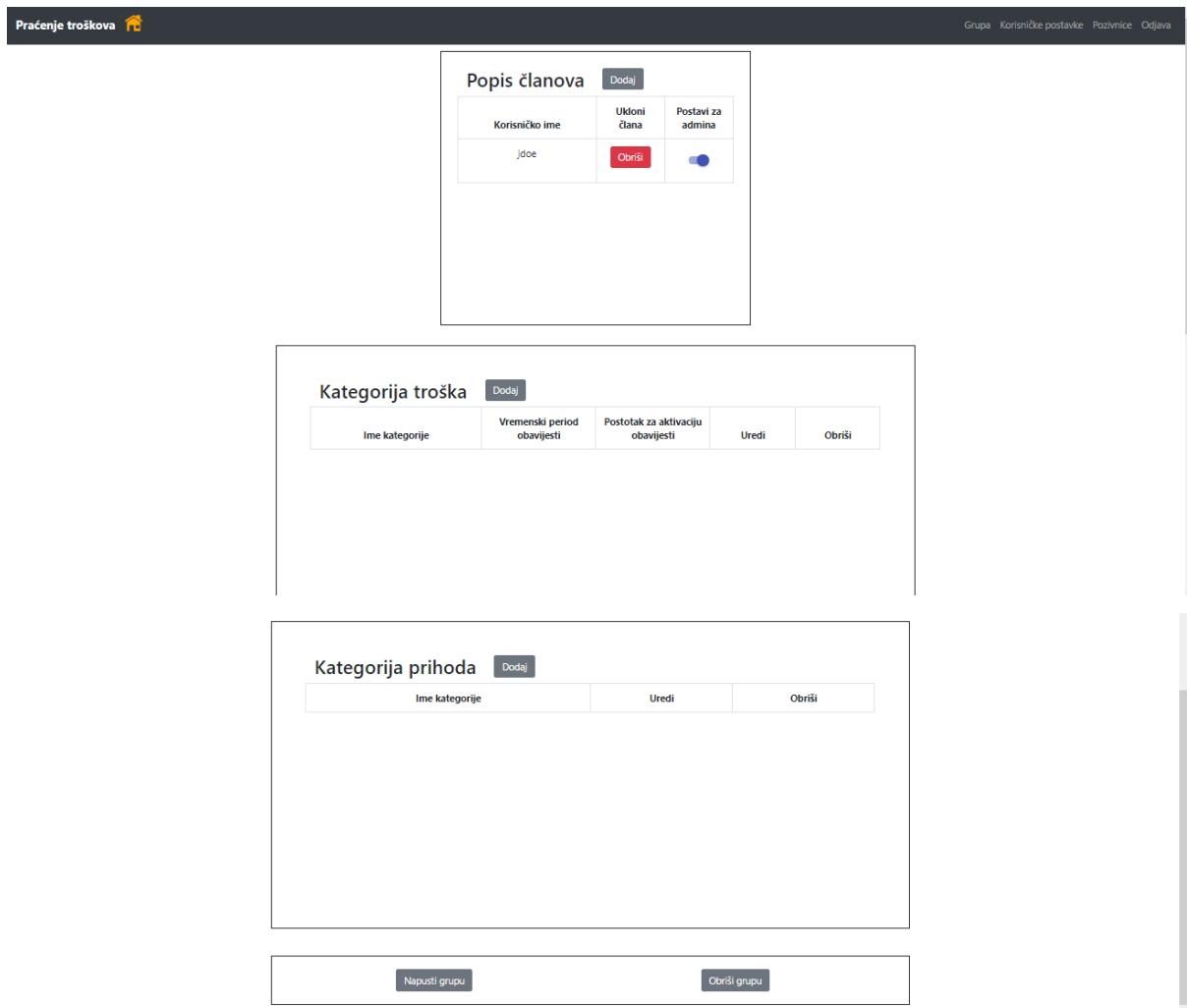
Rad s grupama:

Opciji stvaranja grupe korisnik pristupa odabirom opcije „Stvori“ unutar korisničkih postavki, čime je preusmjeren na obrazac za stvaranje grupe.

Napomena: Grupu je moguće stvoriti samo ako korisnik već nije član neke grupe (čime mu opcija „Stvori“ neće biti omogućena te će mu se pojaviti opcija „Napusti“).

Slika 7.6.20 Obrazac za stvaranje grupe

Prilikom stvaranja grupe moguće je poslati zahtjev bilo kojem postojećem članu. Ispravnim ispunjavanjem obrasca i potvrdom unesenih podataka, korisnik je preusmjeren na postavke grupe. Analogno postavkama štednje, korisnik koji kreira grupu automatski dobiva status administratora te je u mogućnosti dodijeliti taj status preostalim članovima. Ovisno o statusu administratora, omogućena je opcija brisanja grupe. Napustiti grupu mogu svi članovi.



Slika 7.6.21 i Slika 7.6.22 Informacije o grupi i postavke grupe

Pozivnice:

Korisnik prikazu pozivnica može pristupiti odabirom opcije „Pozivnice“ na navigacijskoj traci, nakon čega će biti u mogućnosti vidjeti raspoložive pozivnice za pristup grupama i štednjama.

Napomena: Korisnik je u mogućnosti pristupiti grupi jedino u slučaju kad trenutno nije član grupe.

Pozivnice u štednju		
Ime štednje	Odbaci	Prihvati
stednja	ODBIJ	PRIHVATI

Pozivnice u grupu		
Ime grupe	Odbaci	Prihvati
grupa10	ODBIJ	PRIHVATI

Slika 7.6.23 Pozivnice**Obavijesti:**

Na početnom zaslonu unutar popisa obavijesti u mogućnosti smo vidjeti obavijesti o pozivnicama, o prekoračenjima zadanih limita, ulasku u minus ili o promjeni statusa administratora.

Obavijesti	
Datum	Poruka
16.01.2019.	Stanje Vaše kućne blagajne je trenutačno manje od 0 kuna.
16.01.2019.	Pozvani ste u štednju stednja.
16.01.2019.	Pozvani ste u grupu grupa10.

Slika 7.6.24 Obavijesti

8. Zaključak i budući rad

Zadaci vezani uz prvu verziju projektne dokumentacije su odrđeni. Napisan je opis projektnog zadatka, definirani su funkcionalni zahtjevi s pripadnim opisima obrazaca uporabe, sekvencijskim dijagramima i dijagramima obrazaca uporabe. Opisani su i nefunkcionalni zahtjevi te arhitektura i dizajn sustava s pripadnom skicom baze podataka, dijagramom razreda entiteta i dijagramom razreda s opisom.

Zadaci druge verzije projektne dokumentacije su također uspješno izvršeni. U 6. poglavlju su ažurirani dijagrami razreda te je napisano potpoglavlje 6.4 *Ostali UML dijagrami*. 7. poglavlje kojim se opisuju korištene tehnologije, upute za instalaciju, implementacijski detalji, izgled korisničkog sučelja te korisničke upute je napisano u skladu s izgledom aplikacije i ostvarenim funkcionalnostima projektnog zadatka. 8. i 9. poglavlje te dodaci su ažurirani u skladu s projektnim ostvarenjima u drugom ciklusu.

Projektni zadatak je uspješno implementiran. Kroz prikladno korisničko sučelje se korisniku omogućuje snalaženje u aplikaciji i korištenje svih funkcionalnosti koje su obuhvaćene u sklopu ovog projektnog zadatka.

9. Popis literature

- 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 Software engineering ,Rutgers University, <http://www.ece.rutgers.edu/~marsic/Teaching/SE>
- ⁶ I. Marsic, „Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
- ⁷ Concepts: Requirements, http://www.upedu.org/upedu/process/gcncpt/co_req.htm
- ⁸ UML 2 Class Diagram Guidelines, <http://www.agilemodeling.com/style/classDiagram.htm>
- ⁹ Domain Class Diagram Modeling Standards and Guidelines, <http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf>
- ¹⁰ Astah Community, <http://astah.net/editions/community/> <http://argouml.tigris.org/>

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

Slika 4.3.1 Dijagram obrazaca uporabe za neregistriranog, registriranog i prijavljenog korisnika (UC01, UC02, UC20, UC21, UC22, UC23, UC24, UC25, UC26, UC27, UC28) ..	30
Slika 4.3.2 Dijagram obrazaca uporabe za prijavljenog korisnika – upravljanje troškovima, prihodima i štednjama (UC03, UC04, UC05, UC06, UC07, UC08, UC09, UC10, UC11, UC12, UC13)	31
Slika 4.3.3 Dijagram obrazaca uporabe za prijavljenog korisnika – grupe (UC14, UC15, UC16, UC17, UC18, UC19)	32
Slika 4.4.1 Sekvencijski dijagram obrasca Registracija korisnika.....	33
Slika 4.4.2 Sekvencijski dijagram obrasca Prijava korisnika.....	34
Slika 4.4.3 Sekvencijski dijagram obrasca Dodaj trošak	35
Slika 4.4.4 Sekvencijski dijagram obrasca Izmijeni trošak.....	36
Slika 4.4.5 Sekvencijski dijagram obrasca Obriši trošak	37
Slika 4.4.6 Sekvencijski dijagram obrasca Dodaj prihod.....	38
Slika 4.4.7 Sekvencijski dijagram obrasca Izmijeni prihod	39
Slika 4.4.8 Sekvencijski dijagram obrasca Obriši prihod	40
Slika 4.4.9 Sekvencijski dijagram obrasca Stvorи štednju	41
Slika 4.4.10 Sekvencijski dijagram obrasca Izmijeni štednju	42
Slika 4.4.11 Sekvencijski dijagram obrasca Obriši štednju	43
Slika 4.4.12 Sekvencijski dijagram obrasca Dodaj novac u štednju	44
Slika 4.4.13 Sekvencijski dijagram obrasca Uzmi novac iz štednje	45
Slika 4.4.14 Sekvencijski dijagram obrasca Stvorи grupu.....	46
Slika 4.4.15 Sekvencijski dijagram obrasca Pozovi korisnika u grupu	47
Slika 4.4.16 Sekvencijski dijagram obrasca Uredi postavke grupe	48
Slika 4.4.17 Sekvencijski dijagram obrasca Napusti grupu	49
Slika 4.4.18 Sekvencijski dijagram obrasca Odgovori na poziv u grupu	50
Slika 4.4.19 Sekvencijski dijagram obrasca Izbriši grupu	51
Slika 4.4.20 Sekvencijski dijagram obrasca Dodaj kategoriju troška	52
Slika 4.4.21 Sekvencijski dijagram obrasca Izmijeni kategoriju troška	53
Slika 4.4.22 Sekvencijski dijagram obrasca Obriši kategoriju troška.....	54
Slika 4.4.23 Sekvencijski dijagram obrasca Podesi korisničke postavke	55

Slika 4.4.24 Sekvencijski dijagram obrasca Odjava korisnika	56
Slika 4.4.25 Sekvencijski dijagram obrasca Promjena lozinke.....	57
Slika 4.4.26 Sekvencijski dijagram obrasca Dodavanje kategorije prihoda	58
Slika 4.4.27 Sekvencijski dijagram obrasca Izmjeni kategorije prihoda	59
Slika 4.4.28 Sekvencijski dijagram obrasca Obriši kategorije prihoda.....	60
Slika 6.1.1 Skica sustava	62
Slika 6.1.2 Model baze podataka	65
Slika 6.2.1 Dijagram razreda modela	68
Slika 6.3.1 Dijagram objekata	71
Slika 6.4.1 Komunikacijski dijagram registracije korisnika	72
Slika 6.4.2 Komunikacijski dijagram dodavanja troška.....	72
Slika 6.4.3 Dijagram aktivnosti Registracija korisnika.....	73
Slika 6.4.4 Dijagram aktivnosti Dodavanja troška.....	74
Slika 6.4.5 Dijagram stanja neprijavljenog korisnika	75
Slika 6.4.6 Dijagram stanja za prijavljenog korisnika.....	76
Slika 6.4.7 Dijagram komponenti	77
Slika 7.1.1 Dijagram razmještaja	78
Slika 7.4.1 Rubni slučaj: podaci neispravnog formata.....	85
Slika 7.4.2 Rubni slučaj: korisničko ime i e-mail adresa su zauzeti	86
Slika 7.4.3 Verifikacijski e-mail nakon uspješne registracije	86
Slika 7.4.4 Polje za unos verifikacijskog koda.....	87
Slika 7.4.5 Rubni slučaj: obavijest o neuspješnoj prijavi.....	87
Slika 7.4.6 Početni zaslon nakon uspješne prijave	88
Slika 7.4.7 Obrazac za dodavanje kategorije troška.....	89
Slika 7.4.8 Obrazac za dodavanje kategorije prihoda	89
Slika 7.4.9 Obavijest o neuspješnom dodavanju kategorije troška	90
Slika 7.4.10 Rubni slučaj: obavijest o neuspješnom dodavanju kategorije prihoda	90
Slika 7.4.11 Lista prikaza prihoda i troškova.....	91
Slika 7.4.12 Obrazac za dodavanje prihoda	91
Slika 7.4.13 Rubni slučaj: obavijest o neuspješnom dodavanju prihoda	92
Slika 7.4.14 Brisanje prihoda pritiskom na gumb „Obriši“	93
Slika 7.4.15 Popis štednji	93
Slika 7.4.16 Obrazac za dodavanje štednje	94
Slika 7.4.17 Rubni slučaj: obavijest o neuspješnom dodavanju štednje	95

Slika 7.4.18 Obrazac za dodavanje transakcije štednje.....	96
Slika 7.4.19 Rubni slučaj: neuspješno dodavanje transakcije štednje.....	96
Slika 7.4.20 Obrazac za stvaranje grupe	97
Slika 7.4.21 Rubni slučaj: neuspješno stvaranje grupe	97
Slika 7.4.22 Rubni slučaj: slanje pozivnice nepostojećem korisniku	98
Slika 7.4.23 Napuštanje grupe.....	98
Slika 7.5.1 Preuzimanje Java Development Kita	99
Slika 7.5.2 Instalacija Java Development Kita.....	100
Slika 7.5.3 Preuzimanje razvojne okoline IntelliJ	101
Slika 7.5.4 Preuzimanje PostgreSQL-a	102
Slika 7.5.5 Preuzimanje Git-a.....	103
Slika 7.5.6 Registracija na Heroku.....	104
Slika 7.5.7 Preuzimanje HerokuCLI-a	105
Slika 7.6.1 Pristupna stranica web aplikacije	107
Slika 7.6.2 Obrazac za registraciju korisnika.....	107
Slika 7.6.3 Obrazac za verifikaciju korisničkog računa	108
Slika 7.6.4 E-mail s verifikacijskim kodom	108
Slika 7.6.5 Početni zaslon nakon prijave.....	109
Slika 7.6.6 i Slika 7.6.7 Ostatak početnog zaslona dobiven navigacijom niz stranicu	109
Slika 7.6.8 i Slika 7.6.9 Prikaz korisničkih postavki.....	110
Slika 7.6.10 Obrazac za stvaranje kategorije troška.....	111
Slika 7.6.11 Obrazac za stvaranje kategorije prihoda	111
Slika 7.6.12 Korisničke postavke nakon stvorenih kategorija troška i prihoda	112
Slika 7.6.13 Obrazac za stvaranje prihoda	112
Slika 7.6.14 Obrazac za stvaranje troška.....	113
Slika 7.6.15 Popis prihoda i troškova nakon stvaranje prihoda i troška	113
Slika 7.6.16 Obrazac za stvaranje štednje	114
Slika 7.6.17 i Slika 7.6.18 Informacije o štednji i postavke štednje	115
Slika 7.6.19 Popis štednji početnog zaslona nakon stvorene štednje	116
Slika 7.6.20 Obrazac za stvaranje grupe	116
Slika 7.6.21 i Slika 7.6.22 Informacije o grupi i postavke grupe	117
Slika 7.6.23 Pozivnice	118
Slika 7.6.24 Obavijesti	118

11. Dodatak B: Dnevnik sastajanja

1. sastanak

- **Vrijeme sastanka:** 15.10.2018. 12:00 - 14:00
- **Prisutni:** Domagoj Jakšić, Dominik Galić, Vitomir Marjanović, Robert Čiček, Vlatko Magjer, Domagoj Kroflin
- **Sažetak sastanka:** Analiziran je projektni zadatak te su definirani funkcionalni i nefunkcionalni zahtjevi. Određena je osnovna podjela zadataka, odabrane su tehnologije za rad na projektu, stvoren osnovni nacrt web aplikacije i skica korisničkog sučelja aplikacije.

2. sastanak

- **Vrijeme sastanka:** 23.10.2018. 16:00 - 18:00
- **Prisutni:** Domagoj Jakšić, Dominik Galić, Vitomir Marjanović, Robert Čiček, Vlatko Magjer
- **Sažetak sastanka:** Analiziran posao oko funkcionalnih zahtjeva (opisa obrazaca uporabe) te su raspodijeljeni zadaci članovima grupe vezani uz taj dio projektne dokumentacije za narednih tjedan dana. Stvoreni su osnovni planovi vezani uz izradu sekvencijskih dijagrama. Detaljnije su skicirani izgledi pojedinih view-ova korisničkog sučelja u aplikaciji, razrađeni su prvotni nacrti entiteta te je dogovoren princip upravljanja notifikacijama u aplikaciji. Definirana je arhitektura sustava.

3. sastanak

- **Vrijeme sastanka:** 30.10.2018. 15:00 - 16:30
- **Prisutni:** Domagoj Jakšić, Domagoj Kroflin, Vitomir Marjanović, Robert Čiček, Vlatko Magjer
- **Sažetak sastanka:** Analizirani su napravljeni opisi obrazaca uporabe, sekvencijski dijagrami i opis arhitekture sustava. Ispravljane su pogreške i problemi u dokumentaciji. Dodatno je dogovaran izgled aplikacije. Razgovaralo se o budućim planovima te raspodjeli posla za naredno razdoblje.

4. sastanak

- **Vrijeme sastanka:** 4.12.2018. 12:00 – 14:00
- **Prisutni:** Domagoj Jakšić, Domagoj Kroflin, Vitomir Marjanović, Robert Čiček, Vlatko Magjer, Dominik Galić
- **Sažetak sastanka:** Dogovor i podjela posla oko implementacije projektnog zadatka.

5. sastanak

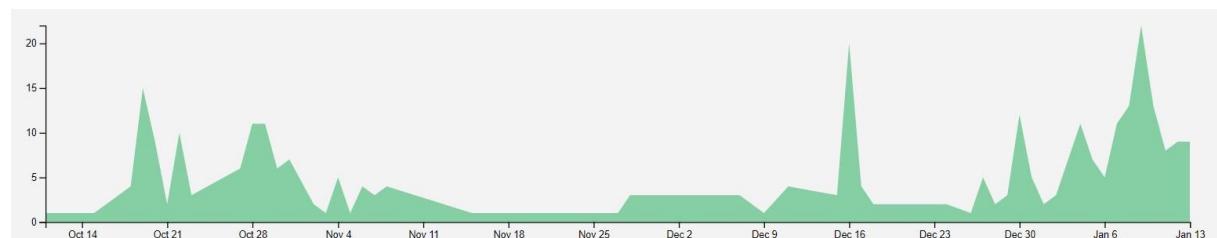
- **Vrijeme sastanka:** 8.1.2019. 18:00 – 20:00
- **Prisutni:** Domagoj Jakšić, Dominik Galić, Vitomir Marjanović
- **Sažetak sastanka:** Dogovor oko dizajna korisničkog sučelja i povezivanja korisničkog sučelja i backend komponente.

12. Dodatak C: Prikaz aktivnosti grupe

Popis aktivnosti	Domagoj Jakšić	Dominik Galić	Vitomir Marjanović	Domagoj Kroflin	Vlatko Magjer	Robert Čiček
Upravljanje projektom	100%	0%	0%	0%	0%	0%
Opis projektnog zadatka	30%	0%	0%	0%	0%	70%
Rječnik pojmova	100%	0%	0%	0%	0%	0%
Opis funkcionalnih zahtjeva	20%	5%	25%	25%	25%	0%
Opis ostalih zahtjeva	100%	0%	0%	0%	0%	0%
Arhitektura i dizajn sustava						
Svrha, opći prioriteti i skica sustava	0%	100%	0%	0%	0%	0%
Dijagram razreda s opisom	100%	0%	0%	0%	0%	0%
Dijagram objekata	0%	0%	0%	0%	0%	100%
Ostali UML dijagrami	0%	50%	0%	0%	50%	0%
Implementacija i korisničko sučelje						
Dijagram	0%	0%	0%	100%	0%	0%

razmještaja						
Korištene tehnologije i alati	0%	0%	0%	0%	0%	100%
Isječak programskog kôda	100%	0%	0%	0%	0%	0%
Ispitivanje programskog rješenja	0%	0%	100%	0%	0%	0%
Upute za instalaciju	20%	0%	0%	0%	0%	80%
Korisničke upute	0%	0%	100%	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	16.67%	16.67%	16.67%	16.67%	16.67%	16.67%
Dodaci						
Indeks	100%	0%	0%	0%	0%	0%
Dnevnik sastajanja	100%	0%	0%	0%	0%	0%

Prikaz rada na projektu s obzirom na broj predaja (*commitova*) u sklopu sustava Gitlab.



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

U prvom ciklusu napisano je prvih 6 poglavlja dokumentacije, izuzev potpoglavlja 6.4 *Ostali UML dijagrami*. Opisan je cilj i planiran način izvedbe projekta, definirani su funkcionalni zahtjevi za koje su dodano navedeni opisi obrazaca uporabe, dijagrami obrazaca uporabe te sekvencijski dijagrami. Navedeni su i nefunkcionalni zahtjevi te opis arhitekture i dizajna sustava s pripadnom skicom baze podataka i dijogramom razreda modela i dijogramom razreda s opisom.

U drugom ciklusu dovršeno je 6. poglavlje u kojem su dodani dijagrami aktivnosti, stanja, komponenti i komunikacijski dijagrami. Napisano je 7. poglavlje u kojem su dodani dijagrami razmještaja, korištene tehnologije i alati, isječci koda iz aplikacije, upute za instalacije, korisničke upute i primjeri testiranja funkcionalnosti sustava. Ažurirani su 8. poglavlje te dodaci.

Implementacija projekta je završena te su aplikaciji ostvarene sve zadane funkcionalnosti sustava obuhvaćene ovim projektnim zadatkom.

Budući rad na ovoj aplikaciji bi u svakom slučaju uključivao dodavanje novih funkcionalnosti kao što je npr. dodavanje chata u aplikaciju ili dodavanje profilne stranice korisnika kako bi drugi korisnici mogli vidjeti više podataka o drugim korisnicima. Također, postojeće funkcionalnosti je moguće nadograditi, npr. slanjem e-mail poruke korisnicima ukoliko istekne vremenski rok njihove štednje ili se dostigne željeni iznos u štednji.