

Oblikovanje programske potpore

Ak. god. 2016./2017.

Troškovnik života

Dokumentacija, 1.

Grupa: *Teflja*

Voditelj: *Filip Božić*

Datum predaje: *18. Studeni 2016*

Nastavnik: *Alan Jović*

Sadržaj

1. Dnevnik promjena dokumentacije	3
2. Opis projektnog zadatka	4
3. Pojmovnik	6
4. Funkcionalni zahtjevi	7
4.1 Dionici:	7
4.2 Aktori i njihovi funkcionalni zahtjevi:	7
4.3 Popis obrazaca uporabe:	8
4.4 Dijagrami obrazaca uporabe:	15
4.5 Sekvencijski dijagrami:	16
5. Ostali zahtjevi	27
6. Arhitektura i dizajn sustava	28
6.1 Svrha, opći prioriteti i skica sustava	28
6.1.1 Udaljena baza podataka	28
6.1.2 Lokalna baza podataka	30
6.2 Dijagram razreda	34
6.3 Objektni dijagrami	35
6.4 Ostali UML dijagrami	36
6.4.1 Komunikacijski dijagrami	36
6.4.2 Dijagram stanja	36
7. Implementacija i korisničko sučelje	38
7.1. Korištene tehnologije i alati	38
7.2. Isječak programskog koda vezan za temeljnu funkcionalnost sustava	39
7.3. Ispitivanje programskog rješenja	40
7.4. Upute za instalaciju	41
7.5. Korisničke upute	42
8. Zaključak i budući rad	43
9. Popis literature	44

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autor(i)	Datum
0.1	Napravljen predložak	Filip Božić	11.10.2016
0.2	Dodan opis projektnog zadatka	Jakov Živković	22.10.2016.
0.5	Dodani funkcionalni zahtjevi bez dijagrama Dodani ostali zahtjevi	Darin Dašić	26.10.2016
0.51	Uređeni obrasci uporabe	Darin Dašić	27.10.2016
0.55	Dodan UML dijagrama obrazaca uporabe	Vilim Prodanović	30.10.2016.
0.6	Dodani pojmovnik	Ema Zoković	30.10.2016.
0.61	Uređena dokumentacija Dorađeni opisi obrazaca uporabe Dodani sekvencijski dijagrami za UC1	Nikola Gajski Filip Božić	30.10.2016.
0.62	Dorađeni funkcionalni zahtjevi	Darin Dašić Filip Božić	31.10.2016.
0.63	Dodani sekvencijski dijagrami SD10, SD11, SD12	Vilim Prodanović	2.11.2016.
0.64	Dodani novi obrasci uporabe UC12-UC18	Vilim Prodanović	3.11.2016.
0.7	Dodani preostali SD	Nikola Gajski Darin Dašić Filip Božić Vilim Prodanović	8.11.2016.
0.8	Dodan komunikacijski dijagram, dijagram stanja, dijagram objekata te dijagram za paket DAO te opisi	Stjepan Vrljičak	15.11.2016.
0.9	Dodano sve vezano uz relacijske modele baze podataka	Jakov Živković	16.11.2016.
0.92	Uređeno sve vezano uz baze podataka	Jakov Živković	16.11.2016.
0.93	Unos dijagrama i pripadnih opisa	Vilim Prodanović	17.11.2016.
0.94	Dodan novi objektni dijagram	Darin Dašić	18.11.2016.
0.95	Dodano nekoliko funkcionalnih zahtjeva, uređen dokument i gramatika	Nikola Gajski	18.11.2016
1.0	Preuređeni dijagrami razreda i dijagram objekata	Filip Božić	18.11.2016.

2. Opis projektnog zadatka

Cilj ovog projekta je razvoj programske podrške za web aplikaciju pod nazivom: Troškovnik života. Svrha aplikacije je organizacija i olakšano praćenje financijskog stanja krajnjeg korisnika na temelju trenutnih prihoda, odnosno rashoda. Aplikacija bi na temelju statističkih podataka omogućila i učinkovito financijsko planiranje koje za cilj ima unaprjeđenje financijskog stanja korisnika. Pod korisnikom, ovdje se smatraju pojedine osobe, ali i kolektivi ka što su npr. obitelj, poduzeće i slično.

Ovom programskom financijskom rješenju može se pristupiti na dva načina:

- prijava postojećeg korisnika unosom korisničkog imena i lozinke
- novi korisnik registracijom u sustav

Postojeći korisnik u početnom sučelju aplikacije odabire formu za prijavu, dok novi korisnik odabire formu za registraciju gdje upisuje novo korisničko ime i lozinku. Korisničko ime i lozinku moguće je naknadno izmijeniti kroz sučelje programa.

U slučaju registracije novog korisnika, otvara se prozor u kojem se od korisnika traži da navede svoje privatne podatke:

- ime i prezime (obavezno)
- OIB
- datum rođenja
- adresu prebivališta (ulica i broj, naziv mjesta, poštanski broj mjesta)
- telefon
- e-mail (obavezno)

U slučaju prijave postojećeg korisnika ne otvara se prozor za unos privatnih podataka, već se otvara početni prozor u kojem je moguće kroz izbornik: stvoriti novi troškovnik života ili otvoriti postojeći pohranjeni troškovnik, pohraniti izmjene u troškovniku ili izaći iz programa. U izborniku treba također biti omogućeno izmijeniti postojeće privatne podatke te generirati i spremi PDF. Korisniku koji se prijavljuje drugi put ili kasnije ne pojavljuje se više prozor za unos osobnih podataka, ali ih on uvijek može izmijeniti kroz sučelje programa.

Pri stvaranju novog troškovnika života korisniku je omogućeno:

- odabrati jednu ili više kategorija troškova (npr. režije, hrana, kozmetika, auto) ili definirati nove kategorije; a kategorije mogu imati i potkategorije, u obliku stablaste strukture
- za svaku odabranu ili definiranu kategoriju definirati stavku troškova
- definirati vlastitu mjesečnu (neto) plaću i ostale prihode (koje korisnik može imenovati)

Svaka kategorija troška osim svog naziva ima i oznaku je li fiksni trošak svakodnevnog života. Ako korisnik odabere da je ta kategorija fiksni trošak, tada i svaka stavka u toj kategoriji postane fiksna stavka troška. Svaka stavka troškova ima:

- naziv
- period evidentiranja (npr. jednom tjedno, jednom mjesečno, u određene mjesece, jednom godišnje, jednokratno)
- evidentirane iznose
- oznaku je li fiksni trošak
- dodatni komentar stavke

Sučelje programa je napravljeno tako da korisniku omogući pregledan grafički prikaz svih kategorija i stavaka troškova. Vremenska skala koja se prikazuje u nekom trenutku u programu može se mijenjati po potrebi, pri čemu je uobičajeni prikaz po 12 mjeseci unutar jedne godine, no moguće je npr. odrediti i tromjesečni skupljeni prikaz troškova u periodu od 2 godine. Osim definiranja vremenske skale, program omogućuje pomak na raniji period (npr. za prošlu godinu) ili unaprijed, ako za to postoje podaci. Za svaki potperiod unutar perioda zadanog vremenskom skalom (npr. za svaki mjesec u jednoj godini), ispisuje se zbroj stavki troškova u pojedinoj kategoriji troškova kao i ukupan zbroj svih kategorija troškova. Zasebno se ispisuje ukupan zbroj svih fiksnih troškova. Isto tako korisnik može steći uvid i u ukupan zbroj svih evidentiranih prihoda (plaća i ostali prihodi).

Korisniku u svakom trenutku treba biti dostupna statistička analiza troškova i prihoda kroz promatrane periode zadane vremenskom skalom u obliku tri grafa:

1. graf: prihod od plaće, ostali prihodi i ukupni prihodi
2. graf: trošak od fiksnih troškova, trošak od stavki koje nisu fiksni troškovi, ukupni trošak
3. graf: ukupni troškovi i ukupni prihodi

Dodatno, u statistici su posebno naznačeni najskuplji trošak u promatranim periodima te najveći prihod.

Upravo statistička analiza te pregledni grafički prikaz omogućuju krajnjem korisniku uvid u trenutno financijsko stanje i potencijalne stavke koje se mogu unaprijedit primjenom strukturnog financijskog planiranja.

Također, programska podrška omogućuje slanje troškovnika drugoj osobi i primanje troškovnika od drugih osoba putem interneta. Naime, korisnički podaci: korisničko ime, ime i prezime i e-mail se prilikom prve prijave u sustav zapisuju i u udaljenu bazu na poslužitelju. Korisnik može kroz sučelje pronaći drugu osobu prema tim podacima kroz ostvarenu tražilicu i poslati mu trenutno otvoreni troškovnik. Korisnik koji je spojen na internet i koji je prijavljen u program dobiva obavijest o pristiglom troškovniku (u lokalnu bazu) koji može otvoriti, ali samo za pregled (čitanje).

Funkciju razmjene financijskih izvještaja među različitim krajnjim korisnicima koju aplikacija nudi omogućava kvalitetnije financijsko planiranje na temelju usporedbe s dobrim i lošim stavkama financijskog stanja kod drugih korisnika. Također nudi i razmjenu informacija u financijskom kolektivu kao što je npr. poduzeće.

Smatramo da se kvalitetno svakodnevno funkcioniranje u današnjem svijetu temelji upravo na financijskoj stabilnosti i strukturnom financijskom planiranju. Stoga bi ovakva aplikacija imala vrlo veliku praktičnu vrijednost za široki spektar krajnjih korisnika kao pojedinaca, ali i kao kolektiva (obitelj, firma, poduzeće, itd.)

3. Pojmovnik

Eclipse: programska razvojna okolina (IDE), koristi se za razvoj aplikacija u raznim programskim jezicima poput Jave, C++...

Java: objektno orijentirani programski jezik.

Apache Tomcat: aplikacijski web poslužitelj otvorenog koda za pokretanje i prikaz web stranica

Git: distribuirani sustav za upravljanje izvornim kodom

CSS (Cascading Style Sheets): vrsta stilskog jezika, rabi se za opis prezentacije dokumenta napisanog pomoću HTML jezika.

HTML (HyperText Markup Language): prezentacijski jezik za izradu web stranica

UML: standardizirani jezik za slikovno predstavljanje i modeliranje objekata.

JPA (Java Persistence API): Javino sučelje za upravljanje relacijskim modelima

JavaScript: skriptni programski jezik koji se izvršava u web pregledniku na strani korisnika. Napravljen je da bude sličan Javi, međutim nije objektno orijentiran

jQuery: platformski neovisna JavaScript knjižica koja olakšava front-end programiranje

CORS (Cross-origin resource sharing): mehanizam koji omogućava dohvat podataka koji se nalaze na udaljenom serveru

4. Funkcionalni zahtjevi

4.1 Dionici:

- Neregistrirani korisnik
- Registrirani korisnik
- Lokalna baza podataka
- Udaljena baza podataka

4.2 Aktori i njihovi funkcionalni zahtjevi:

- Neregistrirani korisnik, inicijator
 - Omogućuje mu se registracija na sustav
- Registrirani korisnik, inicijator
 - Prijava u sustav
 - Promjena osobnih podataka
 - Odjava iz sustava
 - Kreacija novog troškovnika
 - Otvaranje postojećeg troškovnika
 - Grafički prikaz troškovnika
 - Slanje troškovnika drugom korisniku
 - Generiranje PDF verzije troškovnika
 - Spremanje izmjena troškovnika
- Lokalna baza podataka, sudionik
 - Pohranjuje podatke o registriranom korisniku
 - Pohranjuje troškovnike registriranog korisnika
- Udaljena baza podataka, sudionik
 - Pohranjuje podatke o svim registriranim korisnicima
 - Služi kao međuspremnik pri slanju troškovnika od jednog korisnika prema drugom

4.3 Popis obrazaca uporabe:

- **UC1 – Registriraj se u sustav**
 - **Glavni sudionik:** Neregistrirani korisnik
 - **Cilj:** Izrada novog korisničkog računa
 - **Sudionici:** Lokalna baza podataka, udaljena baza podataka, poslužitelj
 - **Preduvjet:** nema
 - **Željeni scenarij:**
 - Korisnik unosi novo korisničko ime i lozinku
 - Poslužitelj uspoređuje podatke sa podacima iz baze podataka (korisničko ime nije zauzeto).
 - Korisnik se preusmjerava na formu za unos privatnih podataka: ime (obavezno), prezime (obavezno), OIB, datum rođenja, adresa prebivališta, telefon i e-mail (obavezno).
 - Poslužitelj šalje korisničko ime, lozinku i privatne podatke u lokalnu bazu podataka kako bi se registrirao novi korisnik.
 - Lokalna baza podataka kreira i pohranjuje novog korisnika.
 - Udaljena baza pohranjuje osnovne podatke o korisniku
 - Korisnik obavješten o uspješnoj registraciji.
 - **Mogući scenarij:** Ukoliko je korisničko ime zauzeto, potrebno je izabrati novo. Lozinka ne zadovoljava kriterije, potrebno izabrati drugu. Izgubljena veza sa bazom podataka, pričekati ili resetirati web stranicu.
- **UC2 - Prijavi se u sustav**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Prijava korisnika u sustav te rad u aplikaciji
 - **Sudionici:** Lokalna baza podataka, udaljena baza podataka, poslužitelj
 - **Preduvjet:** Registracija
 - **Željeni scenarij:**
 - Korisnik unosi svoje korisničko ime i lozinku.
 - Poslužitelj provjerava korisničko ime i lozinku
 - Poslužitelj obavještava registriranog korisnika o uspješnoj prijavi.
 - Poslužitelj provjerava ima li u udaljenoj bazi podataka troškovnika koji su poslani trenutnom korisniku
 - Ako ima takvih troškovnika, poslužitelj ih briše s udaljene baze podataka i sprema u lokalnu bazu podataka
 - **Mogući scenarij:** Ukoliko korisničko ime ili lozinka nisu ispravni, zahtijeva se ponovni upis.

- **UC3 - Odjavi se iz sustava**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Prijavljeni korisnik se uspješno odjavi iz aplikacije i prekine vezu za poslužiteljem
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** Prijava
 - **Željeni scenarij:**
 - Korisnik odabire opciju odjava iz aplikacije.
 - Poslužitelj prekida trenutnu sjednicu te odjavljuje korisnika.
 - **Mogući scenarij:** aplikacija se zablokira, nestane veza sa bazom podataka

- **UC4 – Promijeni podatke**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Korisnik uređuje svoje osobne podatke
 - **Sudionici:** Lokalna baza podataka, udaljena baza podataka, poslužitelj
 - **Preduvjet:** Prijava
 - **Željeni scenarij:**
 - Korisnik odabire opciju uređivanja svojih osobnih podataka.
 - Poslužitelj iz baze podataka dohvaća podatke o korisniku.
 - Poslužitelj otvara prozor gdje se ipisuju trenutni osobni podatci.
 - Korisnik izmjenjuje podatke.
 - Poslužitelj šalje nove podatke u lokalnu i udaljenu bazu podataka na pohranu.
 - Poslužitelj obavještava korisnika o uspješnoj promjeni podataka.
 - **Mogući scenarij:** Korisnik je unio neispravne podatke, obavezne podatke je ostavio praznima, poslužitelj nije uspješno unio promjene u bazu podataka.

- **UC5 – Odaberi kategorije troškova**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Izrada novog troškovnika
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** Prijavljen u aplikaciju i odabrao je funkciju kreiraj novi troškovnik.
 - **Željeni scenarij:**
 - Korisnik odabire opciju za odabir kategorije troškova.
 - Poslužitelj dohvaća iz baze podataka sve pretpostavljane kategorije troškova i prikazuje ih korisniku.
 - Korisnik selektira željene kategorije.
 - **Mogući scenarij:** Korisnik nije odabrao niti jednu kategoriju.

- **UC6 – Stvori novi troškovnik**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Izrada novog troškownika
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** Prijava
 - **Željeni scenarij:**
 - Korisnik odabire opciju stvori novi troškovnik
 - Poslužitelj preusmjeri korisnika u sučelje za stvaranje novog troškownika
 - Korisnik bira jednu ili više kategoriju troškova iz pretpostavljenih kategorija ili može definirati svoju, novu kategoriju troška. Svakoju kategoriju može dodati i potkategoriju troškova, te svakoju potkategoriju novu. Zatim za svaku odabranu kategoriju korisnik definira stavku troška, je li trošak fiksna, period evidentiranja, iznose evidentiranja te ukoliko želi komentar.
 - Korisnik definira svoju mjesečnu plaću (neto) te ostale prihode
 - **Mogući scenarij:** Korisnik je kao stavku neke kategorije naveo drugu kategoriju troškova/prihoda. Poslužitelj obavještava korisnika te ga traži da popravi eventualne greške.
- **UC7 – Kreiraj nove kategorije troškova**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Definirati novu glavnu kategoriju troškova te ju dodati u troškovnik
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** Prijava, rad na troškovniku
 - **Željeni scenarij:**
 - Korisnik odabire kreiranje nove kategorije troškova
 - Navede naziv kategorije te odabere opciju kreiraj
 - Poslužitelj prihvati novu kategoriju te ju dodaje u troškovnik
 - **Mogući scenarij:** Navedena kategorija postoji kao ponuđena kategorija.
- **UC8 – Grafički prikaži troškovnik**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Prikazati troškovnik grafički
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** Prijava, korisnik je kreirao barem jedan troškovnik
 - **Željeni scenarij:**
 - Korisnik odabire vizualizaciju troškownika kojeg trenutno uređuje,
 - Poslužitelj prikaže grafički prikaz troškownika u periodu od 12 mjeseci
 - Korisnik mijenja raspon perioda, period prikaza te elemente troškownika (kategorije, stavke) koje se prikazuju

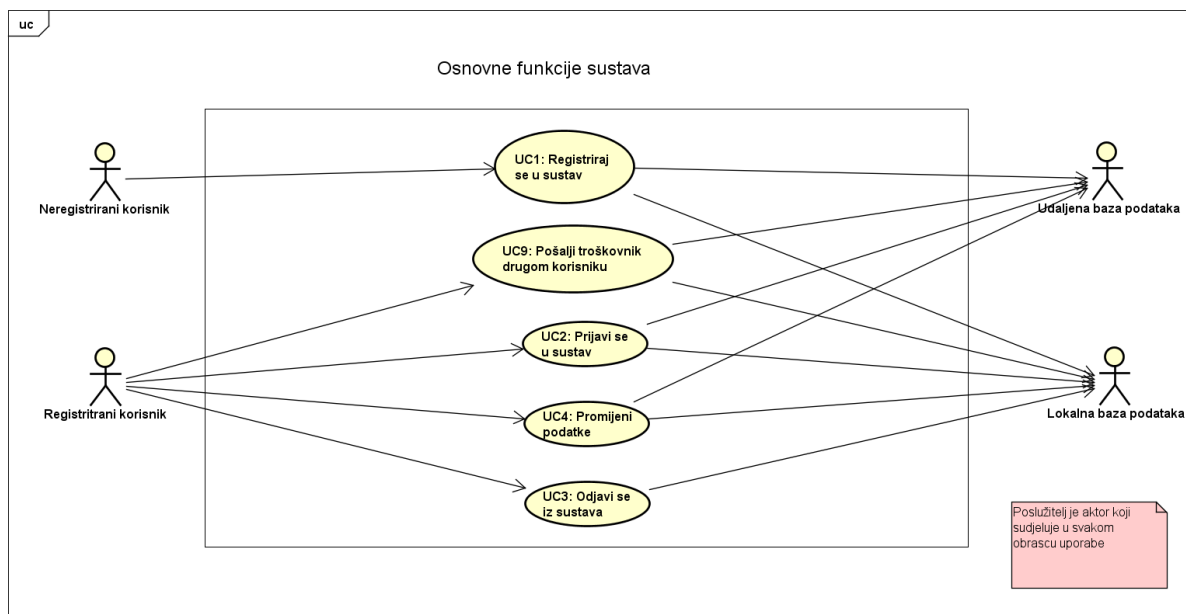
- Prikazuju se sljedeći grafovi: graf prihoda od plaće, ostalih prihoda i ukupnih prihoda, graf troškova od fiksnih troškova, troškova od stavki koje nisu fiksne te ukupnog troška, graf ukupnih troškova i ukupnih prihoda
- **Mogući scenarij:** troškovnik je tek kreiran pa nema dovoljno podataka za graf, graf je nepregledan, prikazuju se krive vrijednosti
- **UC9 – Pošalji troškovnik drugom korisniku**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Slanje i primanje troškovnika kroz aplikaciju
 - **Sudionici:** Lokalna baza podataka, udaljena baza podataka, poslužitelj, drugi registrirani korisnik
 - **Preduvjet:** registracija oba korisnika, pošiljatelj ima barem jedan troškovnik i trenutno je otvoren
 - **Željeni scenarij:**
 - Korisnik odabere opciju slanja troškovnika drugom korisniku
 - Korisnik iz lokalne baze podatak odabire koji troškovnik želi poslati
 - Kroz tražilicu unosom imena i prezimena / korisničkog imena / e-maila pronalazi korisnika kojemu šalje troškovnik
 - Tražilica prikaže željenog korisnika, korisnik ga odabire i šalje troškovnik
 - Poslužitelj dohvaća odabrani troškovnik iz lokalne baze podataka i šalje ga udaljenoj bazi podataka koja ga pohranjuje dok se primatelj ne priključi u sustav
 - Primatelju dolazi obavijest o pristiglom troškovniku te ga može samo pregledavati
 - **Mogući scenarij:** tražilica ne može pronaći primatelja, troškovnik se poslao krivom korisniku, troškovnik se ne pošalje nikome, poslan je krivi troškovnik
- **UC10 – Generiraj PDF verziju troškovnika**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Generirati pdf verziju troškovnika
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** prijava u sustav, korisnik je kreirao barem jedan troškovnik
 - **Željeni scenarij:**
 - Korisnik odabere opciju generiraj PDF
 - Odabire za koji troškovnik želi kreirati PDF
 - Poslužitelj prihvata zahtjev i generira PDF
 - Korisniku se javlja da je PDF uspješno generiran te ga može otvoriti u novom prozoru ili skinuti na svoje računalo.
 - **Mogući scenarij:** aplikacija izgubi vezu sa bazom podataka, nije u mogućnosti generirati pdf, generira pdf krivog formata, server ne omogućava preuzimanje

- **UC11 – Pohrani izmjene na troškovniku**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Pohrana izmjena troškovnika u bazu podataka
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** prijava, izrađeni troškovnik i obavljene promjene na troškovniku
 - **Željeni scenarij:**
 - Korisnik odabire opciju pohrani izmjene
 - Poslužitelj prihvata zahtjev
 - Izmjene se pohranjuju u bazu podataka
 - Poslužitelj javlja uspješnu pohranu izmjena
 - **Mogući scenarij:** Izgubila se veza sa bazom podataka, poslužitelj nije u mogućnosti pohraniti podatke. Javlja se greška i traži korisnika da pokuša ponovno
- **UC12 - Definiraj stavke troškova za kategorije**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Definiranje stavaka troškova za svaku navedenu kategoriju
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** prijava, odabrana nova kategorija troškova u troškovniku
 - **Željeni scenarij:**
 - Korisnik upisuje naziv kategorije
 - Korisnik određuje period evidentiranja
 - Korisnik upisuje evidentirane iznose
 - Korisnik, ako želi, upisuje dodatni komentar stavke
 - **Mogući scenarij:** gubi se veza s Internetom, promjene na kategorijama neće biti zapamćene
- **UC13 - Odredi fiksnost kategorija**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Određivanje fiksnosti stavaka troškova pojedinih kategorija
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** prijava, otvoren troškovnik
 - **Željeni scenarij:**
 - Korisnik pregleda sve kategorije i zaključi koji su troškovi fiksni
 - Označi koje kategorije želi da budu fiksne
 - Poslužitelj sve stavke u označenim kategorijama također označi fiksima
 - **Mogući scenarij:** gubi se veza s Internetom, promijenjena fiksnost na kategorijama neće biti zapamćena

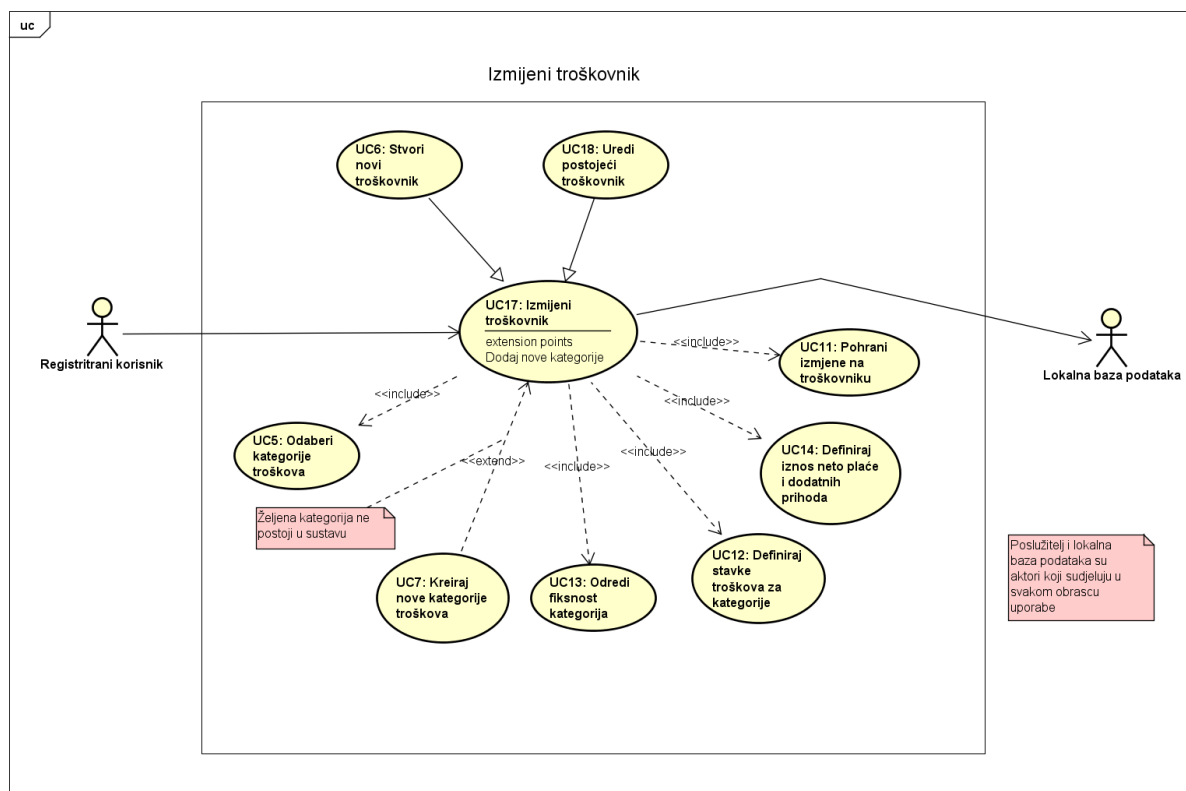
- **UC14 - Definiraj iznos neto plaće i dodatnih prihoda**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Definiranje iznosa mjesečne neto plaće i ostalih dodatnih prihoda
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** prijava, otvoren troškovnik
 - **Željeni scenarij:**
 - Korisnik upisuje iznos mjesečne neto plaće
 - Korisnik upisuje ostale dodatne prihode bitne za ovaj troškovnik, ukoliko ih se može sjetiti
 - **Mogući scenarij:** gubi se veza s Internetom, upisani podaci se neće zapamtiti
- **UC15 - Pregledaj troškovnik**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Pregled odabranog troškownika
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** prijava
 - **Željeni scenarij:**
 - Korisnik se prijavljuje u sustav i odabire troškovnik koji želi pregledati (može birati između vlastitih troškownika i troškownika primljeni od drugih korisnika)
 - **Mogući scenarij:** gubi se veza s bazom podataka, nije moguće dohvatiti nijedan troškovnik
- **UC16 - Definiraj vremensku skalu**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Definiranje vremenske skale za grafički prikaz troškownika
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** prijava, odabran troškovnik za grafički prikaz
 - **Željeni scenarij:**
 - Pretpostavljena skala je 12 mjeseci
 - Korisnik, ako želi, mijenja skalu
 - **Mogući scenarij:** aplikacija se blokira, promijenjena skala se vraća na prethodnu vrijednost
- **UC17 – Izmijeni troškovnik**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Izmjena postojećeg ili stvaranje novog troškownika
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** Prijava
 - **Željeni scenarij:**

- Korisnik bira između opcija izmjena postojećeg (vlastitog) i stvaranja novog troškovnika
- **Mogući scenarij:** Korisnik želi izmijeniti primljeni (tuđi) troškovnik, to nije moguće
- **UC18 – Uredi postojeći troškovnik**
 - **Glavni sudionik:** Registrirani korisnik
 - **Cilj:** Izmjena i uređivanje postojećeg (vlastitog) troškovnika
 - **Sudionici:** Lokalna baza podataka, poslužitelj
 - **Preduvjet:** Prijava
 - **Željeni scenarij:**
 - Korisnik odabire opciju uredi postojeći troškovnik
 - Poslužitelj preusmjeri korisnika u sučelje za uređivanje troškovnika
 - Korisnik mijenja jednu ili više kategoriju troškova iz kategorija ili može definirati novu kategoriju troška. Svakoj kategoriji može dodati i mijenjati potkategoriju troškova, te svakoj potkategoriji novu. Zatim za svaku odabranu kategoriju korisnik mijenja stavku troška, je li fiksni trošak, period evidentiranja, iznose evidentiranja te, ukoliko želi, komentar.
 - Korisnik mijenja svoju mjesečnu plaću (neto) te ostale prihode
 - **Mogući scenarij:** aplikacija se gasi, izmjene na troškovniku neće biti zapamćene

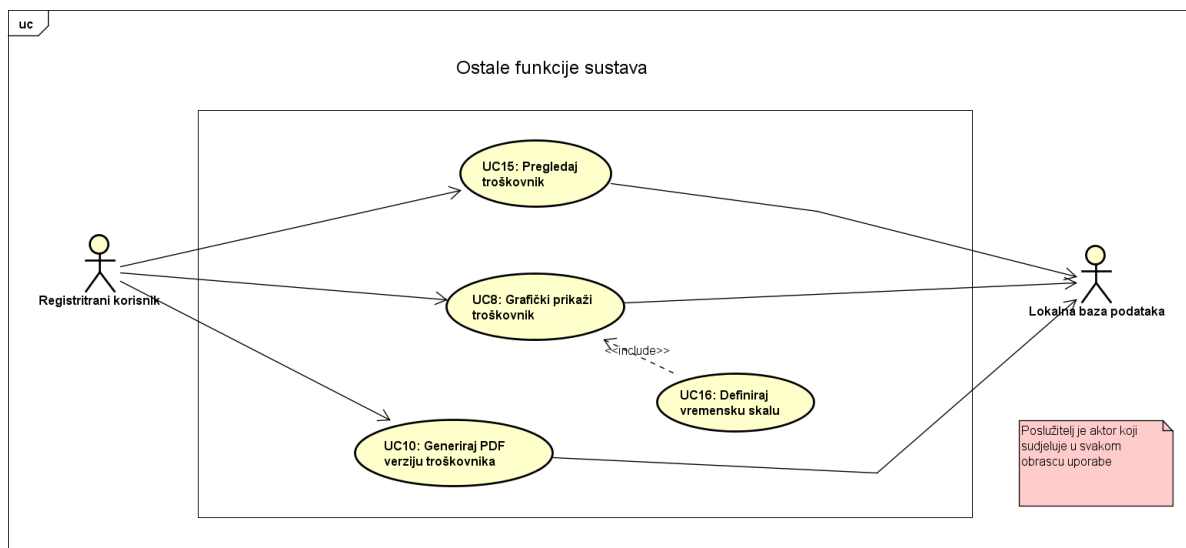
4.4 Dijagrami obrazaca uporabe:



Slika 1 - UC dijagram osnovnih funkcija sustava



Slika 2 - UC dijagram Izmijeni troškovnik

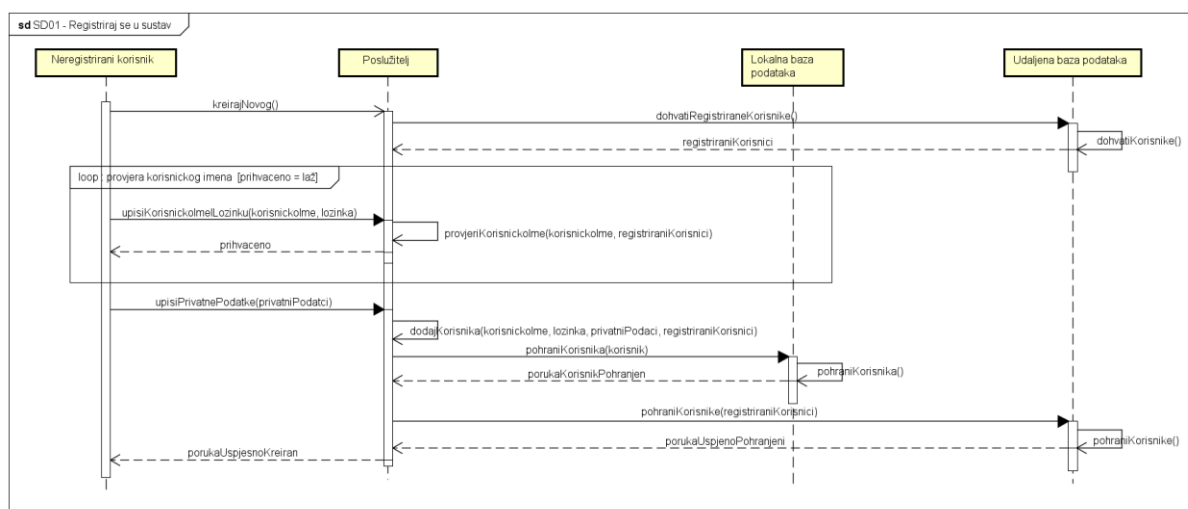


Slika 3 - UC dijagram ostalih funkcija sustava

4.5 Sekvencijski dijagrami:

UC1 – Registriraj se u sustav

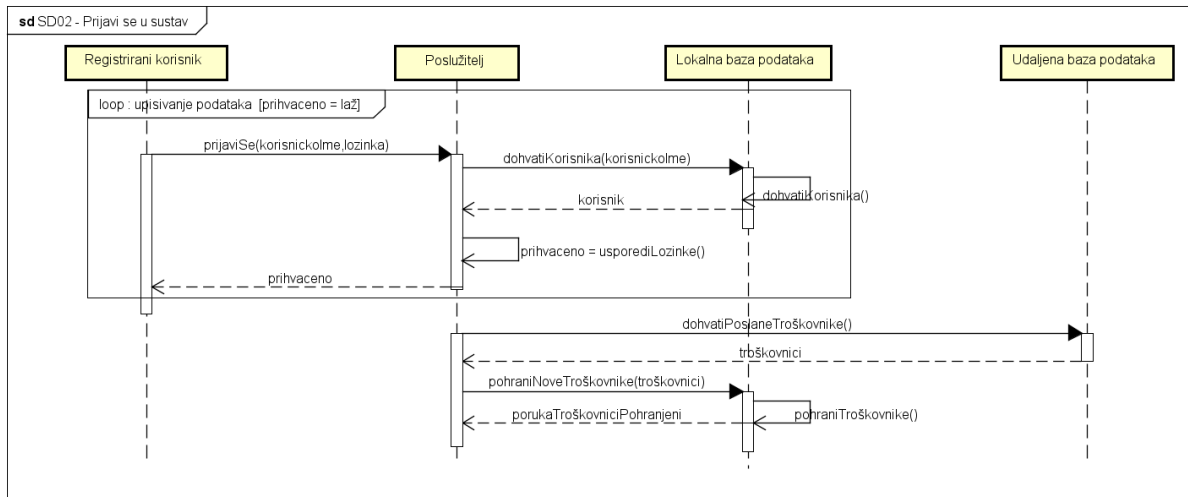
Korisnik zahtijeva registraciju te poslužitelj dohvaća sve trenutno registrirane korisnike iz udaljene baze podataka. Korisnik upisuje novo korisničko ime i lozinku te se korisničko ime uspoređuje s korisničkim imenima svih registriranih korisnika. Ako takvo korisničko ime već postoji, korisnik mora upisati novo. Nakon uspješne provjere korisničkog imena i lozinke, korisnik upisuje ostale privatne podatke te se oni spremaju samo na lokalnu bazu podataka. Dodatno, korisničko ime, lozinka i e-mail adresa korisnika zapisuju se na udaljenu bazu podataka zajedno s ostalim korisnicima.



Slika 4 - SD dijagram za UC1

UC2 - Prijavi se u sustav

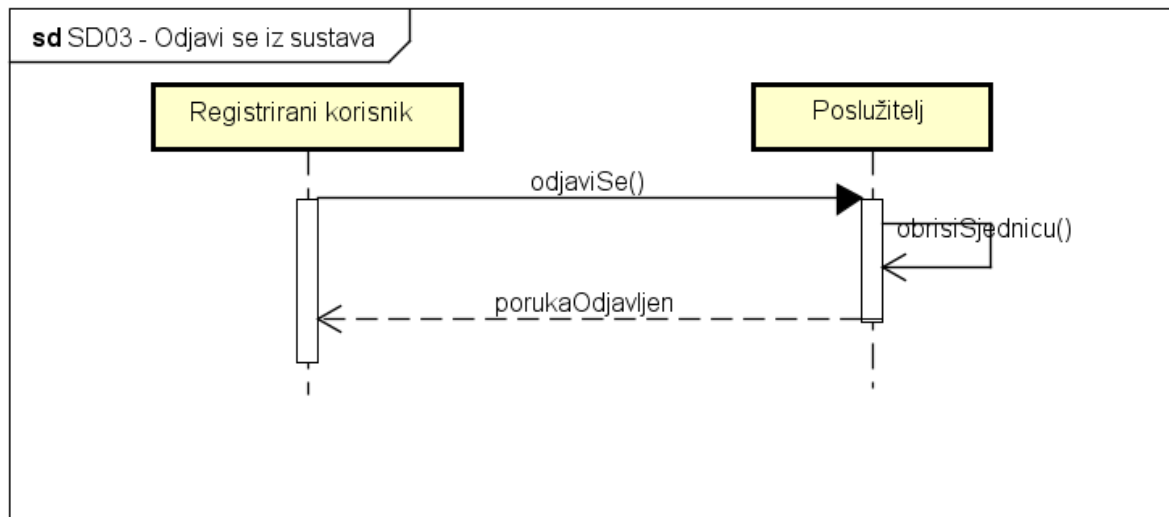
Korisnik upisuje ime i lozinku te se oni provjeravaju s odgovarajućim korisničkim imenom i lozinkom pohranjenim na lokalnoj bazi podataka. Nakon uspješne prijave, poslužitelj u lokalnu bazu podataka sprema pristigle troškovnike od drugih korisnika, ako takvih ima.



Slika 5 - SD dijagram za UC2

UC3 - Odjavi se iz sustava

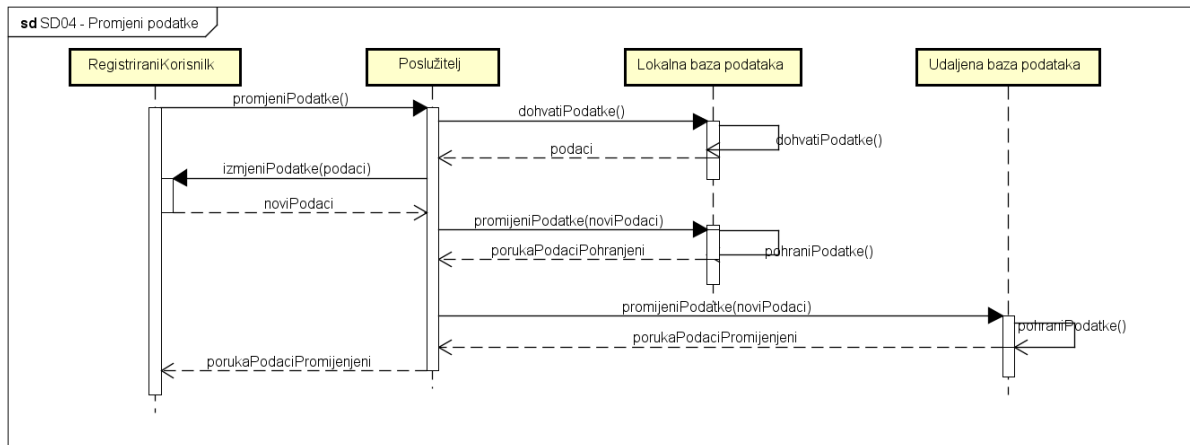
Korisnik se odjavljuje iz sustava, poslužitelj briše sjednicu korisnika.



Slika 6 - SD dijagram za UC3

UC4 – Promijeni podatke

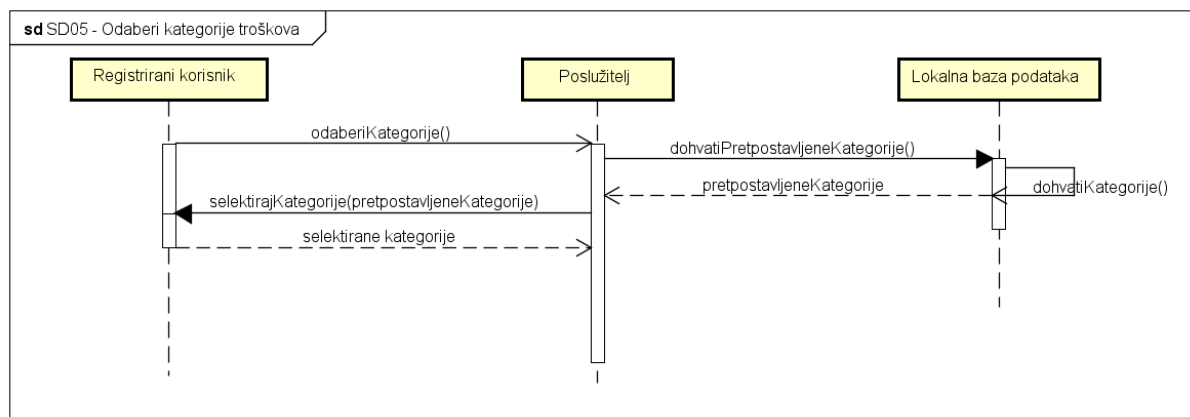
Korisnik zahtijeva promjenu svojih korisničkih podataka. Njegovi trenutni podaci se dohvaćaju iz lokalne baze podataka te ih korisnik mijenja. Novi podaci zapisuju se natrag u lokalnu bazu podataka te u udaljenu bazu podataka ako je promijenjeno korisničko ime, lozinka ili e-mail adresa.



Slika 7 - SD dijagram za UC4

UC5 – Odaberi kategorije troškova

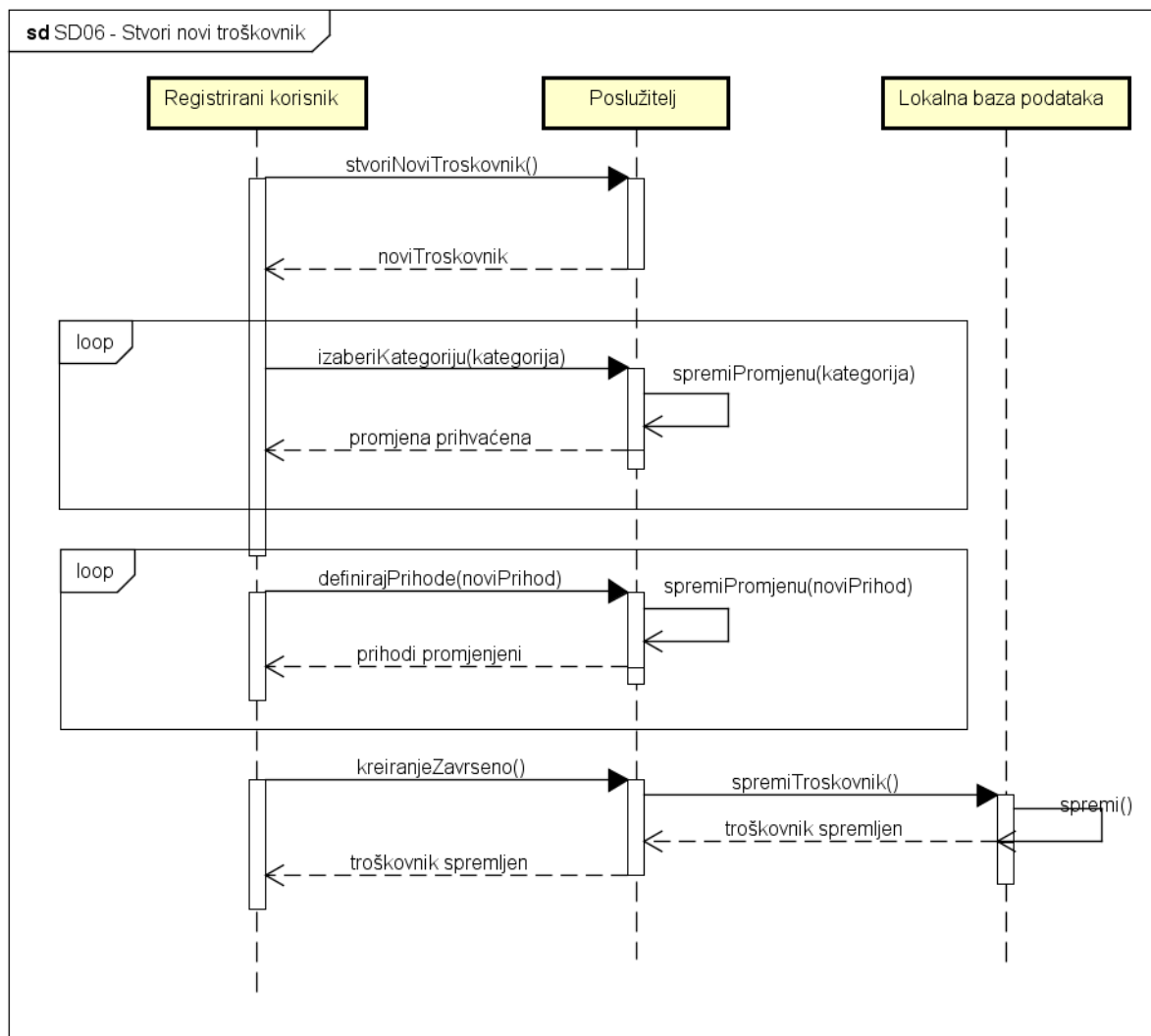
Korisnik zahtijeva odabir kategorija troškova te mu poslužitelj iz baze podataka dohvaća sve pretpostavljene kategorije. Korisnik odabire kategorije koje želi unijeti u svoj troškovnik.



Slika 8 - SD dijagram za UC5

UC6 – Stvori novi troškovnik

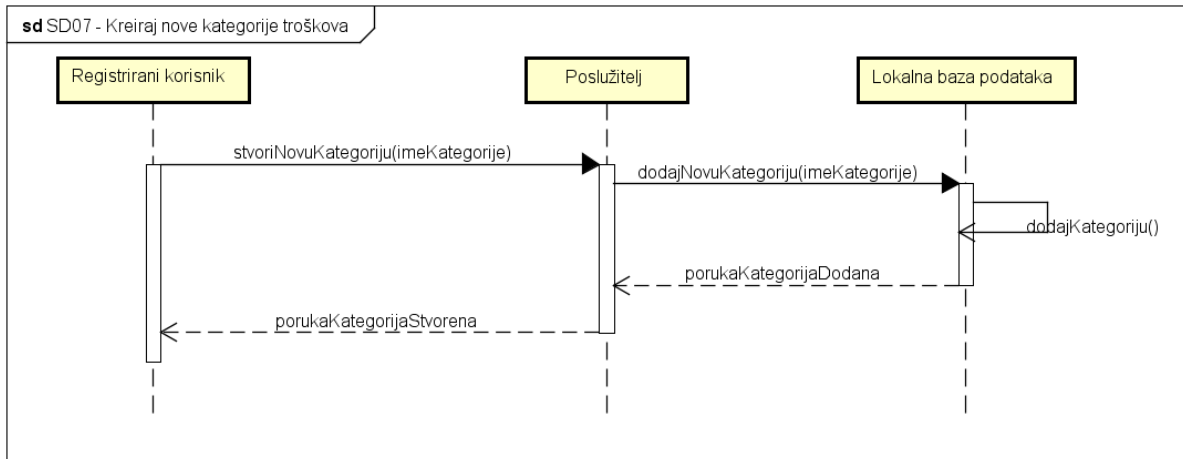
Korisnik želi stvoriti novi troškovnik. Prvo odabire kategorije troškova, a zatim definira prihode. Novi troškovnik se sprema u lokalnu bazu podataka.



Slika 9 - SD dijagram za UC6

UC7 – Kreiraj nove kategorije troškova

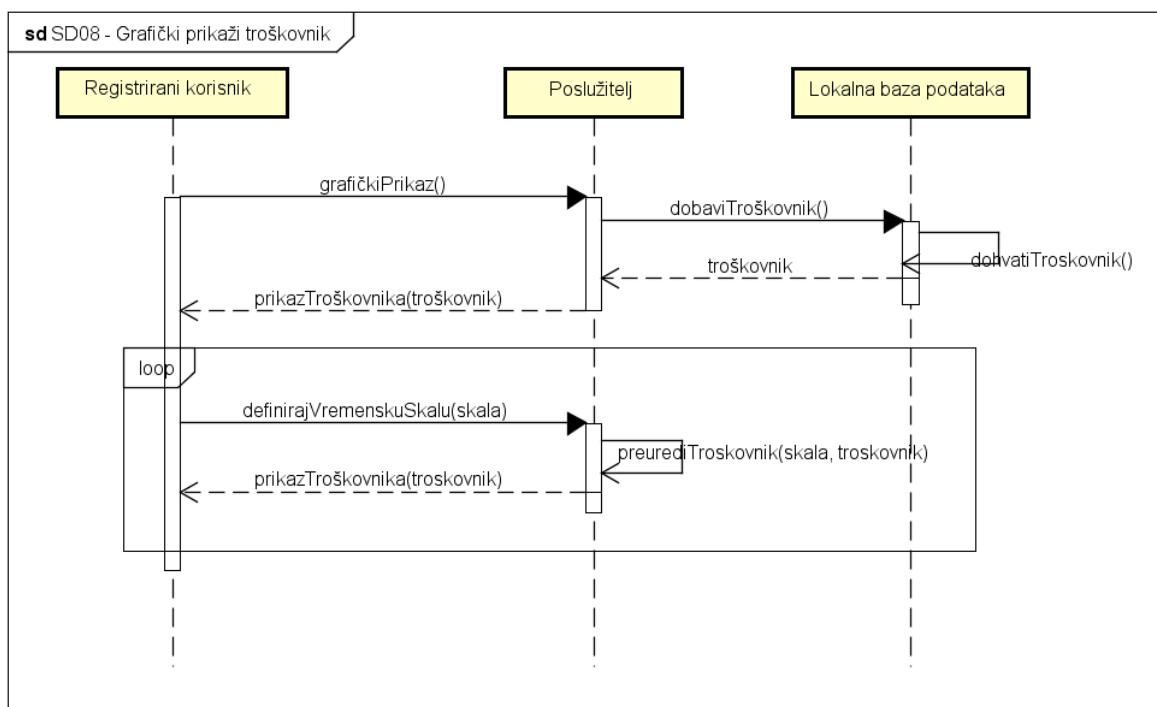
Korisnik želi stvoriti novu kategoriju troškova te ju dodaje u troškovnik. Poslužitelj novu kategoriju troškova sprema u lokalnu bazu podataka.



Slika 10 - SD dijagram za UC7

UC8 – Grafički prikaži troškovnik

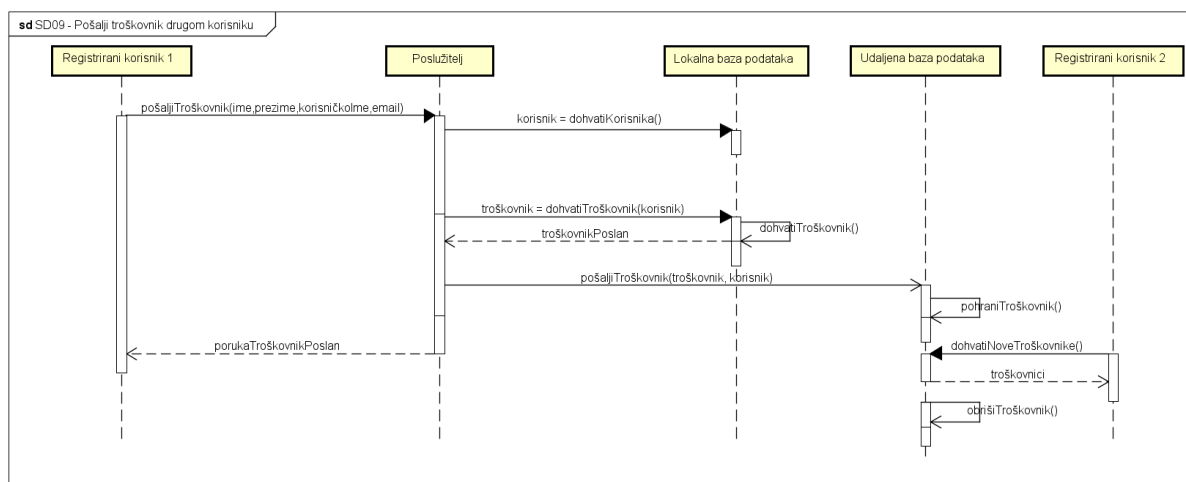
Korisnik zahtijeva grafički prikaz nekog od troškovnika koje mu je poslužitelj dohvatio iz lokalne baze podataka. Nakon što korisnik odabire koji troškovnik odabire koji troškovnika želi grafički prikazati, na zaslonu mu se prikazuje grafički prikaz. Pri tome korisnik može mijenjati vremensku skalu grafa koliko god puta želi.



Slika 11 - SD dijagram za UC8

UC9 – Pošalji troškovnik drugom korisniku

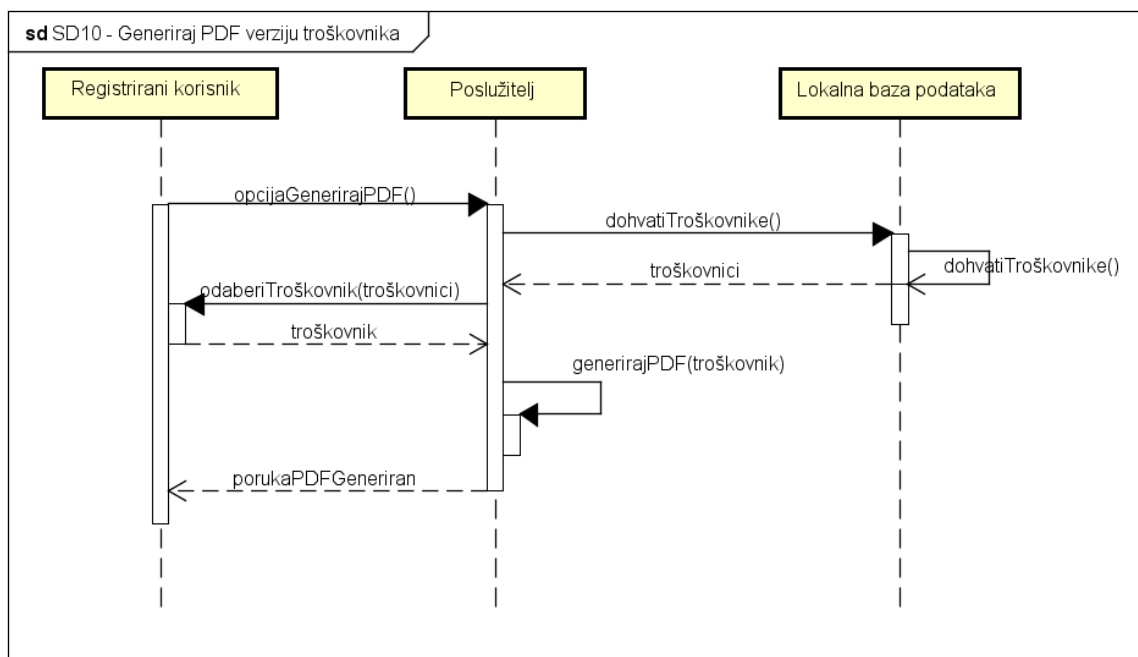
Korisnik odabire opciju slanja troškovnika drugom korisniku te navodi korisničke podatke drugog korisnika. Zatim korisnik odabire koji troškovnik želi poslati te poslužitelj taj troškovnik dohvaća iz lokalne baze podataka. Nakon što dohvati troškovnik, poslužitelj u udaljenu bazu podataka šalje troškovnik zajedno s korisničkim podacima korisnika kojem je potrebno dostaviti troškovnik. Udaljena baza primljeni troškovnik privremeno sprema sve dok se korisnik kojemu je troškovnik poslan ne prijavi u sustav. Nakon što se on prijavi u sustav, drugi korisnik, tj. njegov poslužitelj s udaljene baze podataka dohvaća primljeni troškovnik, briše ga iz udaljene baze podataka i sprema u lokalnu bazu.



Slika 12 - SD dijagram za UC9

UC10 – Generiraj PDF verziju troškovnika

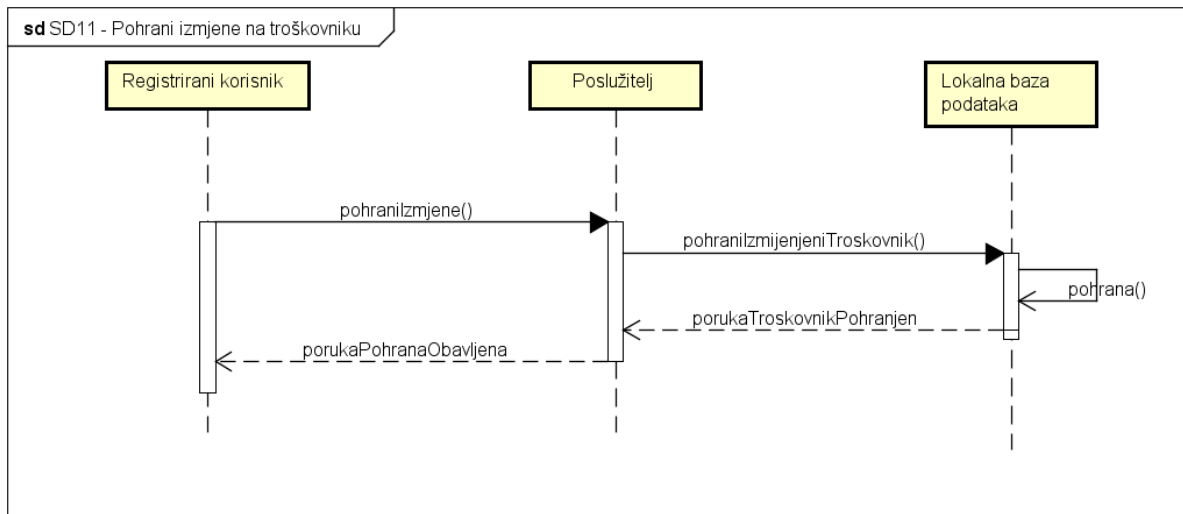
Korisnik odabire koji troškovnik iz lokalne baze podataka želi generirati u PDF dokument. Poslužitelj nakon toga generira PDF dokument te ga šalje korisniku, tj. sprema ga na korisnikovo računalo.



Slika 13 - SD dijagram za UC10

UC11 – Pohrani izmjene na troškovniku

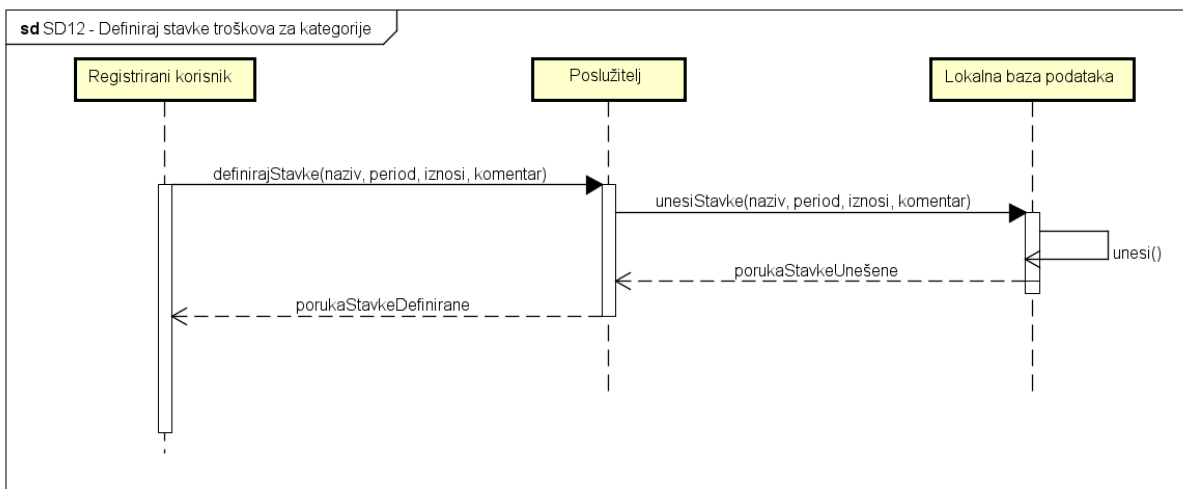
Korisnik zahtijeva pohranu napravljenih promjena na troškovniku. Poslužitelj lokalnoj bazi podataka šalje izmijenjeni troškovnik te ga ona pohranjuje na mjesto dosadašnjeg.



Slika 14 - SD dijagram za UC11

UC12 - Definiraj stavke troškova za kategorije

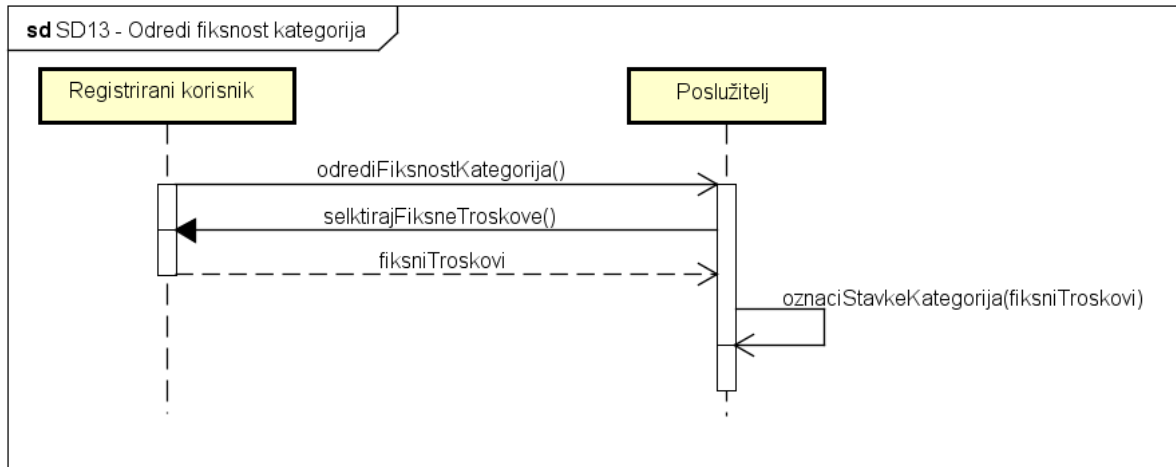
Korisnik definira i unosi dijelove stavaka troškova koji pripadaju određenim kategorijama. Poslužitelj nove stavke prosljeđuje lokalnoj bazi podataka koja ih pohranjuje.



Slika 15 - SD dijagram za UC12

UC13 - Odredi fiksnost kategorija

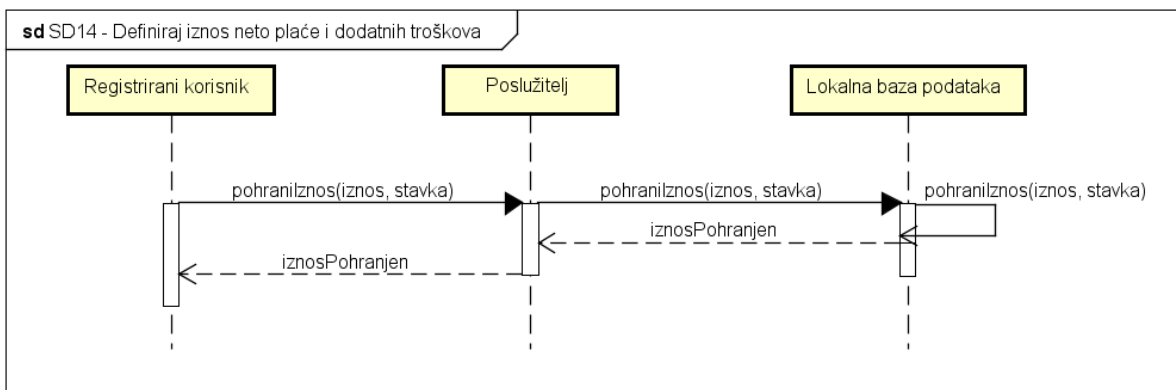
Korisnik označava koje kategorije predstavljaju fiksne troškove te poslužitelj fiksima označava i sve njihove stavke.



Slika 16 - SD dijagram za UC13

UC14 - Definiraj iznos neto plaće i dodatnih prihoda

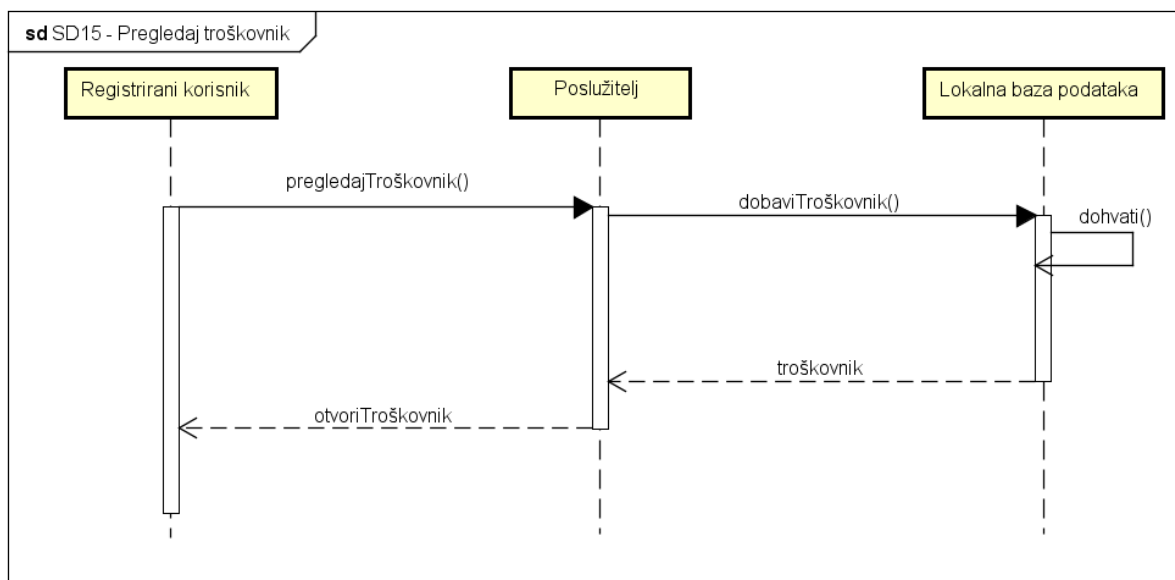
Korisnik unosi iznose neto plaće i dodatnih prihoda te ih poslužitelj prosljeđuje u lokalnu bazu podataka koja ih pohranjuje.



Slika 17 - SD dijagram za UC14

UC15 - Pregledaj troškovnik

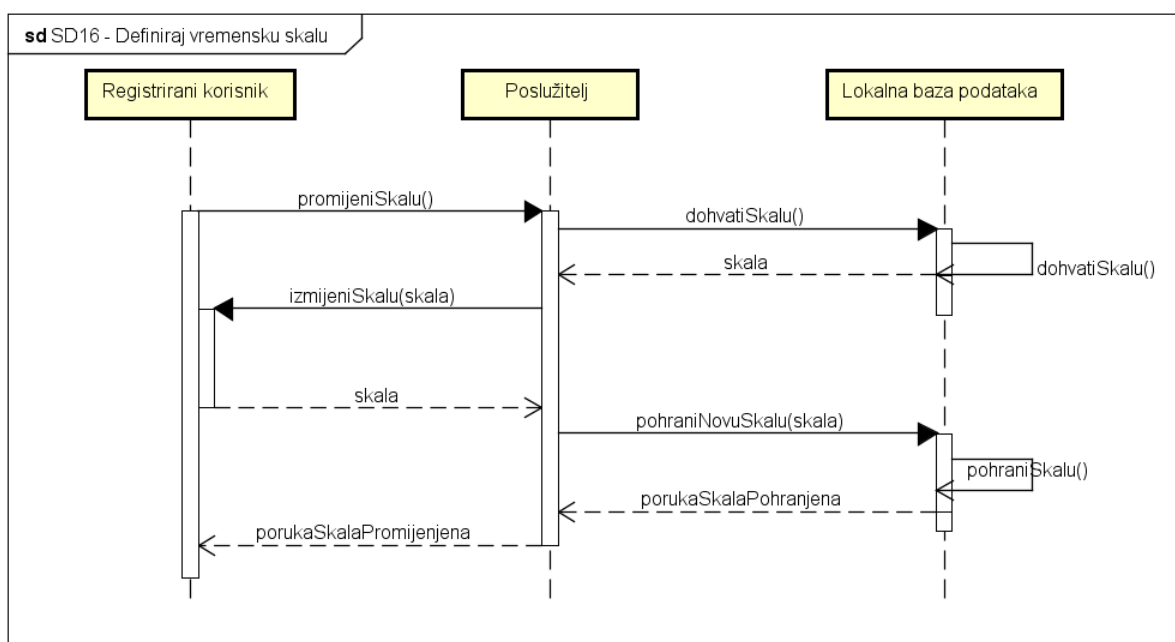
Korisnik zahtijeva pregled troškovnika te ga poslužitelj dohvaća iz lokalne baze podataka i predaje korisniku na pregled.



Slika 18 - SD dijagram za UC15

UC16 - Definiraj vremensku skalu

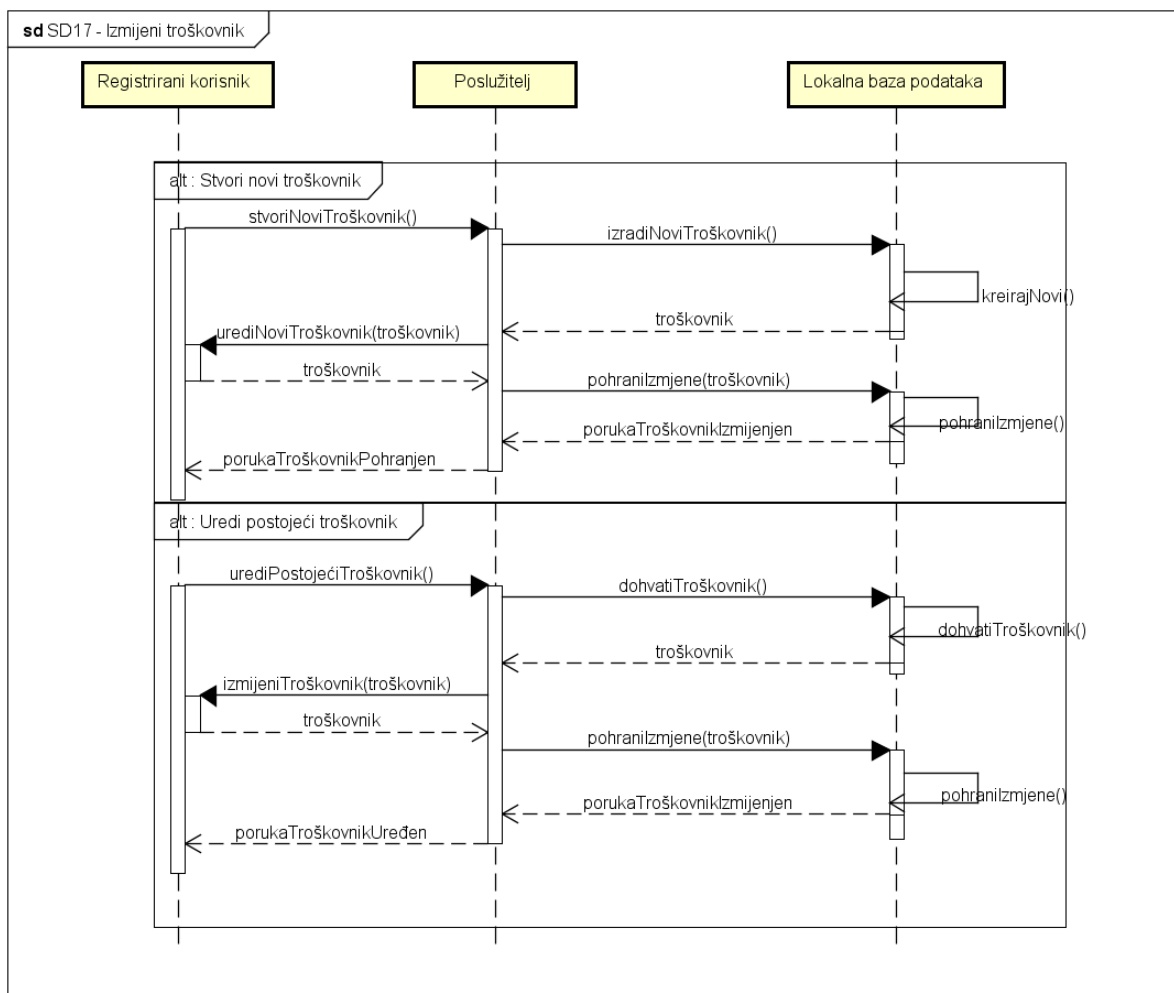
Korisnik zahtijeva izmjenu vremenske skale, poslužitelj iz lokalne baze podataka dohvaća postojeću skalu te ju predaje korisniku na izmjenu. Nakon izmjene, poslužitelj ju šalje bazi podataka na pohranu.



Slika 19 - SD dijagram za UC16

UC17 – Izmijeni troškovnik

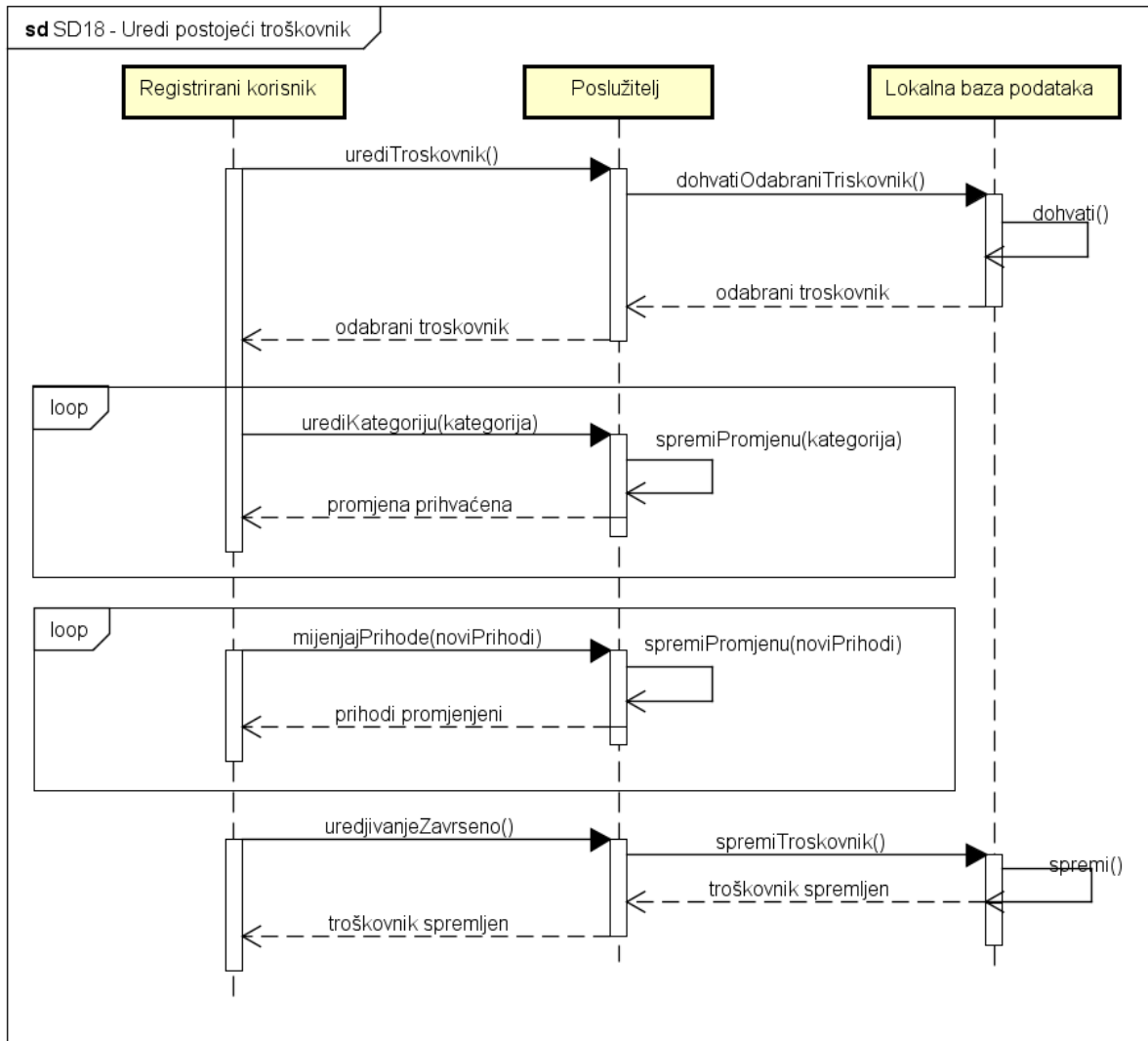
Korisnik izabire želi li stvoriti novi troškovnik ili urediti postojeći troškovnik. Ako stvara novi, lokalna baza stvara novi troškovnik te ga predaje poslužitelju koji ga predaje korisniku na uređivanje. Nakon što korisnik uredi troškovnik, poslužitelj šalje lokalnoj bazi podataka troškovnik na pohranu. Ako korisnik uređuje postojeći troškovnik, poslužitelj iz baze dohvaća troškovnik i predaje korisniku na mijenjanje. Nakon izmjena, poslužitelj šalje u lokalnu bazu novi troškovnik.



Slika 20 - SD dijagram za UC17

UC18 – Uredi postojeći troškovnik

Korisnik zahtijeva uređenje nekog postojećeg troškownika te poslužitelj odabrani troškovnik dohvaća iz lokalne baze podataka. Korisnik tada uređuje kategorije troškova u troškovniku i mijenja prihode. Nakon što završi s promjenama, korisnik zahtijeva pohranu nastalih promjena te poslužitelj lokalnoj bazi podataka predaje izmijenjeni troškovnik na pohranu.



Slika 21 - SD dijagram za UC18

5. Ostali zahtjevi

- Sustav mora podržavati hrvatske grafeme.
- Sustav mora garantirati točnost informacija i podataka.
- Osigurano podržavanje paralelnog rada više korisnika istovremeno.
- Nepravilno postupanje korisnika s aplikacijom ne smije narušavati njezinu funkcionalnost.
- Osigurana jednaka kvaliteta rada na svim operacijskim sustavima.
- Korektno i precizno napisane upute radi jednostavnog korištenja aplikacije.
- Ograničena razina dostupnosti prema korisničkim ovlastima.
- Omogućena nadogradnja sustava, ali pri tome se ne smije narušiti dosadašnji rad sustava.
- Troškovnici se pohranjuju i učitavaju iz lokalne baze podataka na korisničkom računalu
- Određene situacije u troškovnicima posebno označiti zbog bolje preglednosti (ukupni trošak je veći od prihoda, određeni troškovi nisu uneseni ili su neuobičajeni...)
- Ispitati rad programa sa barem 2 gotova troškovnika u lokalnoj bazi podataka
- Ispitati rad programa sa barem 2 gotova troškovnika poslana drugim korisnicima

6. Arhitektura i dizajn sustava

6.1 Svrha, opći prioriteti i skica sustava

Budući da je potrebno imati udaljenu bazu podataka na poslužitelju, za implementaciju odabiremo arhitekturu **web aplikacije**. Za tu vrste arhitekture smo se opredijelili jer su sučelja izrađena za web aplikaciju najjednostavnija za oblikovati, no glavni razlog nam je bio taj da je potrebno omogućiti dohvat podataka iz udaljene baze podataka, a to lako možemo implementirati CORS (Cros-Origin Resource Sharing) mehanizmom izravno u .html datoteci.

Cijeli sustav možemo podijeliti na dva glavna podsustava:

- Web aplikacija s lokalnom bazom podataka kojoj se pristupa preko web sučelja
- Udaljeni poslužitelj s udaljenom bazom podataka kojem se pristupa pomoću CORS mehanizma

U projektnom zadatku koriste s dvije baze podataka, obje temeljene na relacijama. Sama struktura relacijske baze podataka nam olakšava lakši (grafički) prikaz entiteta i sadržaje te baze. Podaci u ovakvoj bazi organiziraju se u relacije između kojih se definiraju određene veze. Relacija je definirana kao skup n-torki s istim atributima definiranim nad istom domenom iz koje se uzimaju vrijednosti. Za naš zadatak koristimo dvije baze, lokalnu i udaljenu. Lokalna se sprema na korisničko računalo (disk), dok je udaljena pohranjena na uređaju gdje se nalazi i poslužitelj. Za grafički prikaz ER-modela baze podataka koristimo program MySQL Workbench. U nastavku su opisane relacije, relacijski atributi i njihov sadržaj, te su priloženi grafički prikazi.

6.1.1 Udaljena baza podataka

Slijedi opis entiteta udaljene baze podataka:

User:

Atribut	Domena	Opis atributa
id	long, not null, unique	id korisnika
username	string, not null, unique	korisničko ime u sustavu
firstName	string, not null	ime korisnika
lastName	string, not null	prezime korisnika
email	string, not null, unique	email korisnika

PK = {id}

ExpenseList:

Atribut	Domena	Opis atributa
id	long, not null, unique	id troškovnika
name	string, not null	ime troškovnika
userOwnerID	long, FK{user}	id user-a koji je stvorio troškovnik

PK={id}

ExpenseCategory:

Atribut	Domena	Opis atributa
id	long, not null, unique	id stavke prihoda
name	string, not null	ime stavke prihoda
fixed	boolean, not null	varijabla koja označava fiksnošć stavke prihoda
superCategoryID	long, FK {ExpenseCategory}	id expense category-a koji je nadkategorija ovoj
expenseListID	long, FK {ExpenseList}	id expense list-a kojem pripada ova kategorija

PK={id}**ExpenseItem:**

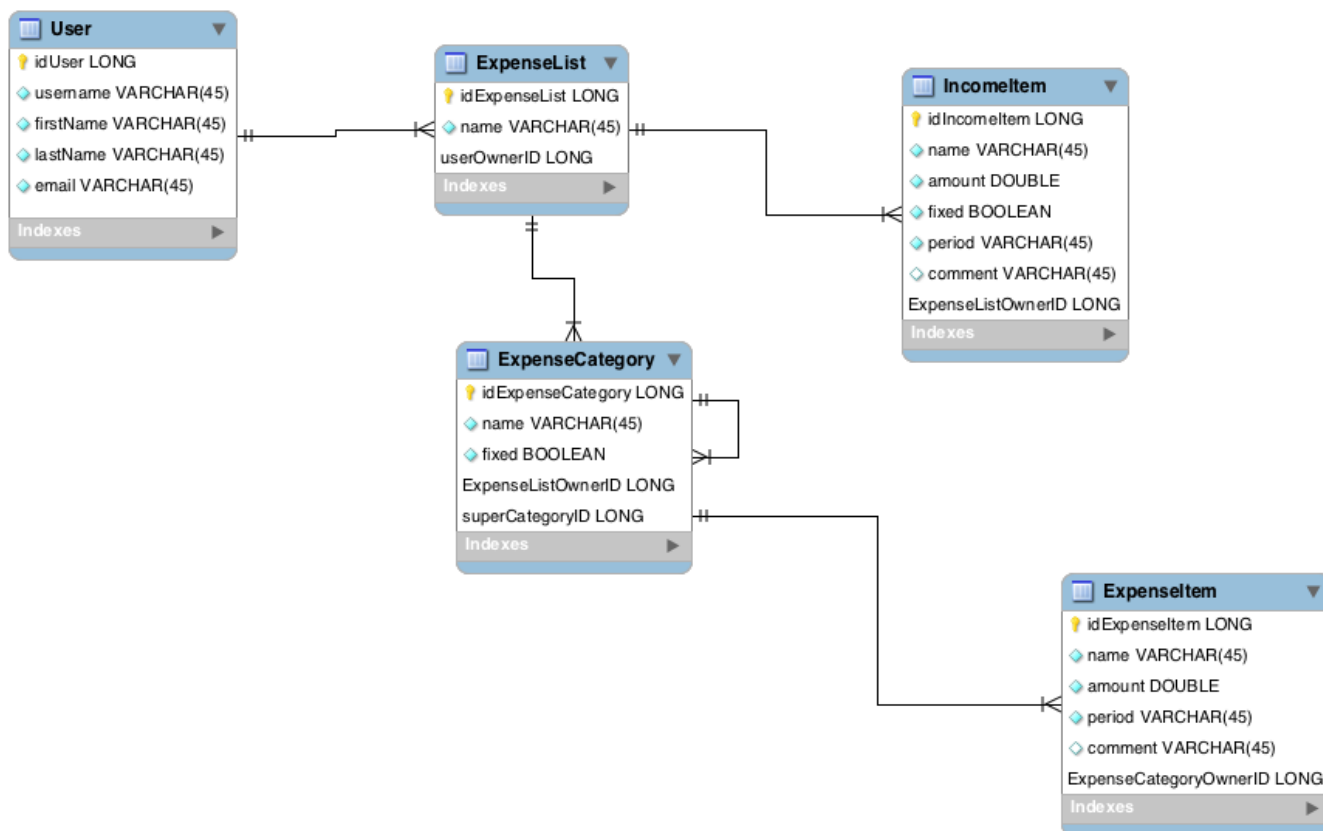
Atribut	Domena	Opis atributa
id	long, not null, unique	id stavke troška
name	string, not null	ime stavke troška
amount	double, not null	iznos stavke troška
period	string, not null	period zaračunavanja
comment	string	dodatni komentar stavke troška
expenseCategoryOwnerID	long FK {ExpenseCategory}	id kategorije kojoj stavka troška pripada

PK = {id}**IncomeItem:**

Atribut	Domena	Opis atributa
id	long, not null, unique	id stavke prihoda
name	string, not null	ime stavke prihoda
amount	double, not null	iznos stavke prihoda
fixed	boolean, not null	varijabla koja označava fiksnošć stavke prihoda
period	string, not null	period zaračunavanja
comment	string	dodatni komentar stavke prihoda
expenseListOwnerID	long FK {ExpenseList}	id troškovnika kojem stavka prihoda pripada

PK = {id}

Grafički prikaz ER modela udaljene baze podataka:



Slika 22 - ER model udaljene baze podataka

6.1.2 Lokalna baza podataka

U nastavku je lokalna baza podataka koja se nalazi na korisničkom računalu

User:

Atribut	Domena	Opis atributa
id	long, not null, unique	id korisnika
username	string, not null, unique	korisničko ime u sustavu
password	string, not null	korisnička lozinka
firstName	string, not null	ime korisnika
lastName	string, not null	prezime korisnika
OIB	string, unique	OIB korisnika
dateOfBirth	Date	datum rođenja korisnika
adressID	long, FK{Adress}	id adrese korisnika
telefon	int, unique	telefon korisnika
email	string, not null, unique	email korisnika

PK = {id}

Adress :

Atribut	Domena	Opis atributa
id	long, not null, unique	id adrese
street	string, not null	ulica
houseNumber	int, not null,unique	ulični broj
city	string	grad
postCode	int, not null	poštanski broj

PK = {id}**ExpenseList:**

Atribut	Domena	Opis atributa
id	long, not null, unique	id troškovnika
name	string, not null	ime troškovnika
userOwnerID	long, FK{user}	id user-a koji je stvorio troškovnik

PK={id}**ExpenseCategory:**

Atribut	Domena	Opis atributa
id	long, not null, unique	id stavke prihoda
name	string, not null	ime stavke prihoda
fixed	boolean, not null	varijabla koja označava fiksnost stavke prihoda
superCategoryID	long, FK {ExpenseCategory}	id expense category-a koji je nadkategorija ovoj kategoriji
expenseListID	long, FK {ExpenseList}	id expense list-a kojem pripada ova kategorija

PK={id}

ExpenseItem:

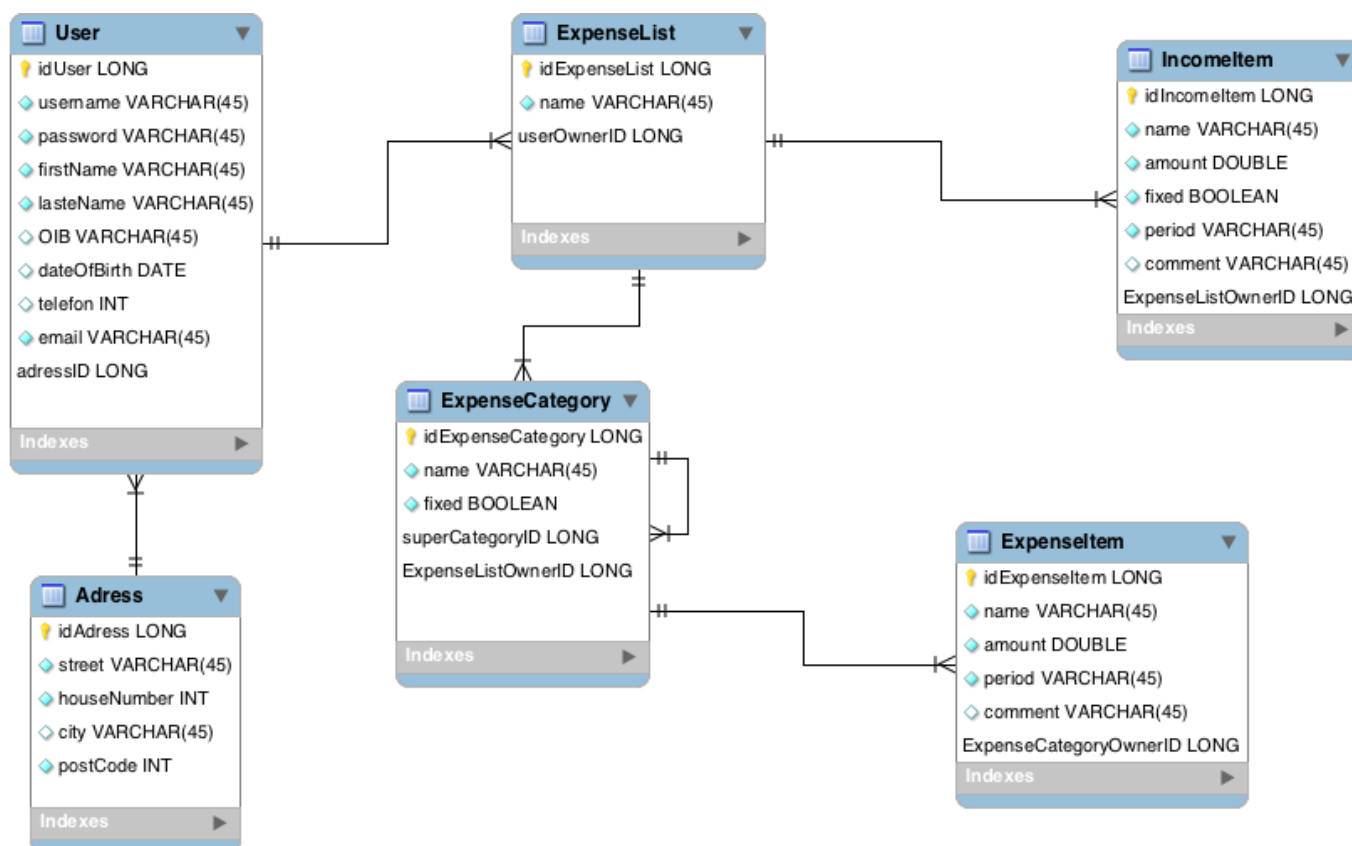
Atribut	Domena	Opis atributa
id	long, not null, unique	id stavke troška
name	string, not null	ime stavke troška
amount	double, not null	iznos stavke troška
period	string, not null	period zaračunavanja
comment	string	dodatni komentar stavke troška
expenseCategoryOwnerID	Long, FK {ExpenseCategory}	id kategorije kojoj stavka troška pripada

PK = {id}**IncomeItem:**

Atribut	Domena	Opis atributa
id	long, not null, unique	id stavke prihoda
name	string, not null	ime stavke prihoda
amount	double, not null	iznos stavke prihoda
fixed	boolean, not null	varijabla koja označava fiksnost stavke prihoda
period	string, not null	period zaračunavanja
comment	string	dodatni komentar stavke prihoda
expenseListOwnerID	Long, FK{ ExpenseList}	id troškownika kojem stavka prihoda pripada

PK = {id}

Grafički prikaz ER modela lokalne baze podataka:



Slika 23 - ER model lokalne baze podataka

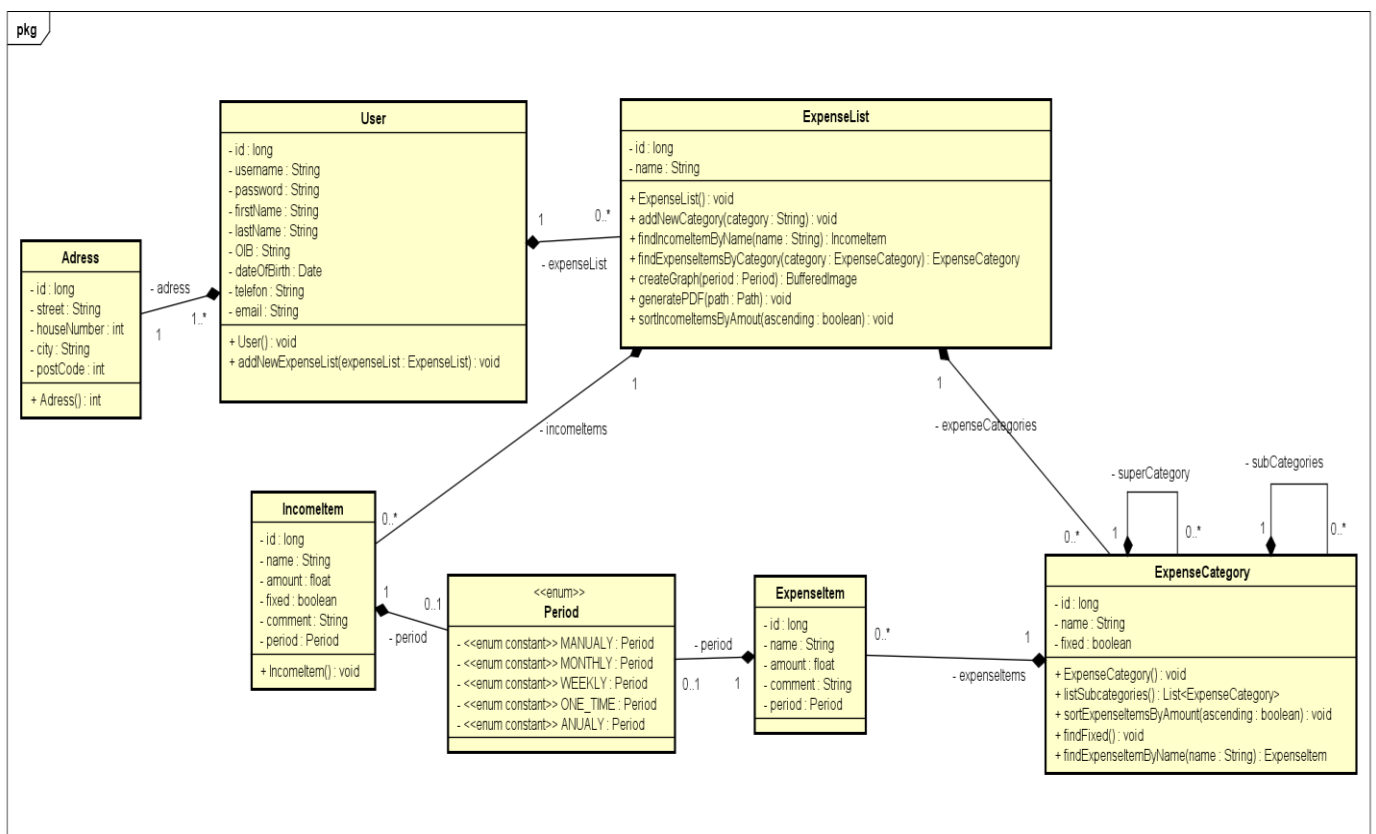
6.2 Dijagram razreda

U ovom potpoglavlju opisani su razredi koji se koriste u projektu te su grafički prikazani preko UML dijagrama.

Razred **User** predstavlja korisnika koji se koristi sustavom. Atributi razreda se sastoje od osnovnih informacija o korisniku kao što su ime, prezime, OIB, korisničko ime... Korisnik također ima atribut zasebnog razreda **Adress** koji predstavlja njegovu adresu. Članske varijable ovog razreda opisuju korisničku adresu (ulica, kućni broj, poštanski broj...). Svaki korisnik ima svoju listu troškovnika koji su opisani razredom **ExpenseList**.

ExpenseList je jednostavan razred koji sadrži ime troškovnika te dvije liste. Prva lista je lista kategorija troškova koji se nalaze u obliku objekata razreda **ExpenseCategory**. Druga lista je lista prihoda koji se nalaze u obliku objekata razreda **Incomeltem**. Razred ExpenseCategory je razred koji ima ime, listu podkategorija koja je po tipu isto ExpenseCategory, varijablu nadkategorija koja je po tipu isto ExpenseCategory, te listu **ExpenseItem**a. Razred ExpenseItem i razred Incomeltem su dva slična razreda. Razred ExpenseItem modelira stavku troška, dok razred Incomeltem modelira stavku prihoda. Glavna razlika između ta dva razreda je to što u razredu ExpenseItem fiksnost te stavke se nasljeđuje od razreda ExpenseCategory kojem ta stavka pripada, dok u razredu Incomeltem fiksnost se mora definirati zasebno.

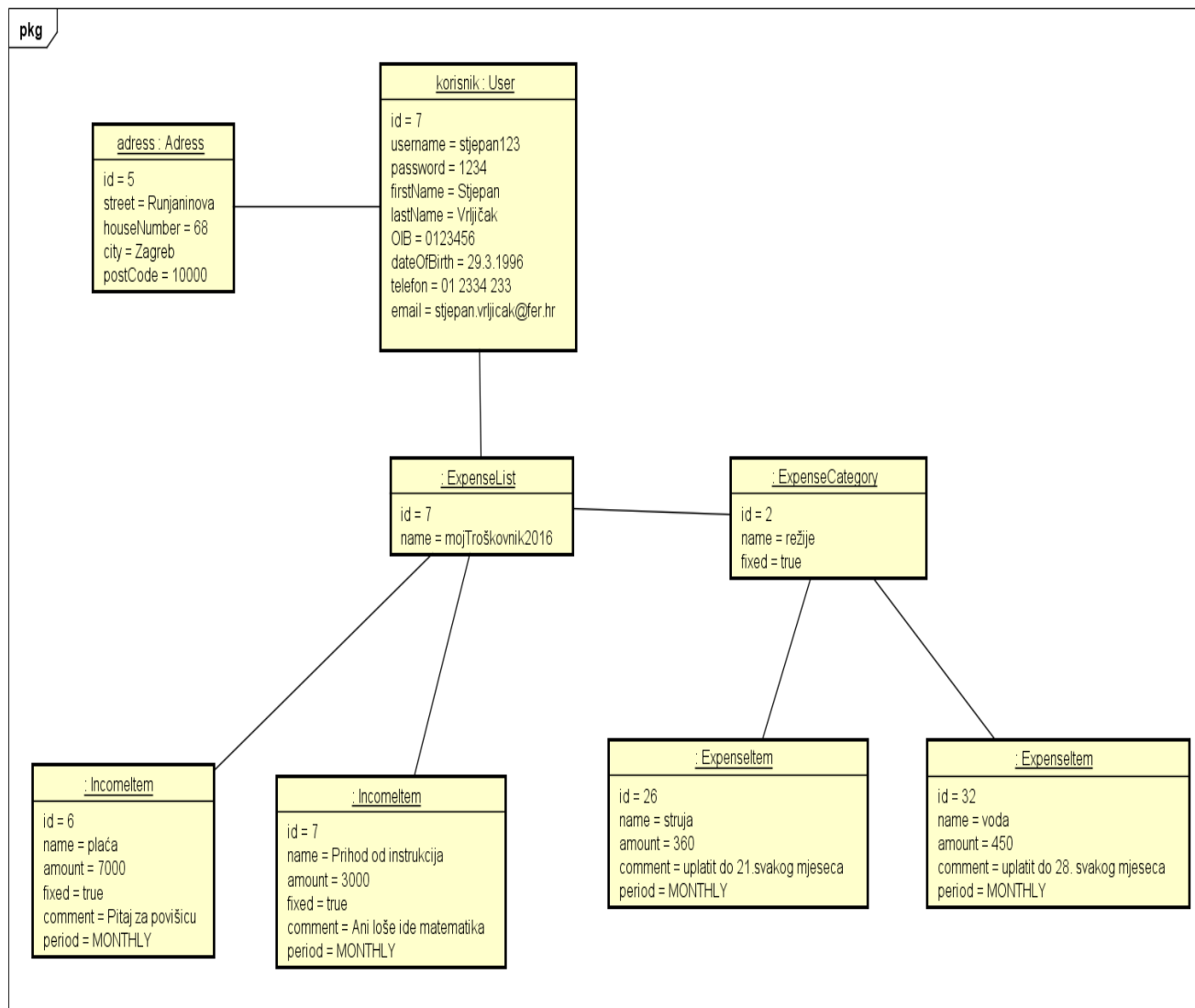
Dijagram razreda također sadrži 1 enum: **Period**. Ta enumeracija služi za određivanje perioda zaručavanja stavke prihoda i stavke troška.



Slika 24 - Dijagram razreda

6.3 Objektni dijagrami

Objektnim dijagramom prikazujemo stanja sustava u nekom trenutku. Na ovom dijagramu je prikazano stanje u kojemu se registrirani korisnik (stjepan123) nalazi u sustavu te ima jedan troškovnik naziva mojiTroškovnik2016. Za sad je u taj troškovnik dodao jednu kategoriju troška (režije) koja ima dvije stavke: struju i vodu. Također troškovnik sadži dvije stavke prihoda: plaća i prihod od instrukcija.



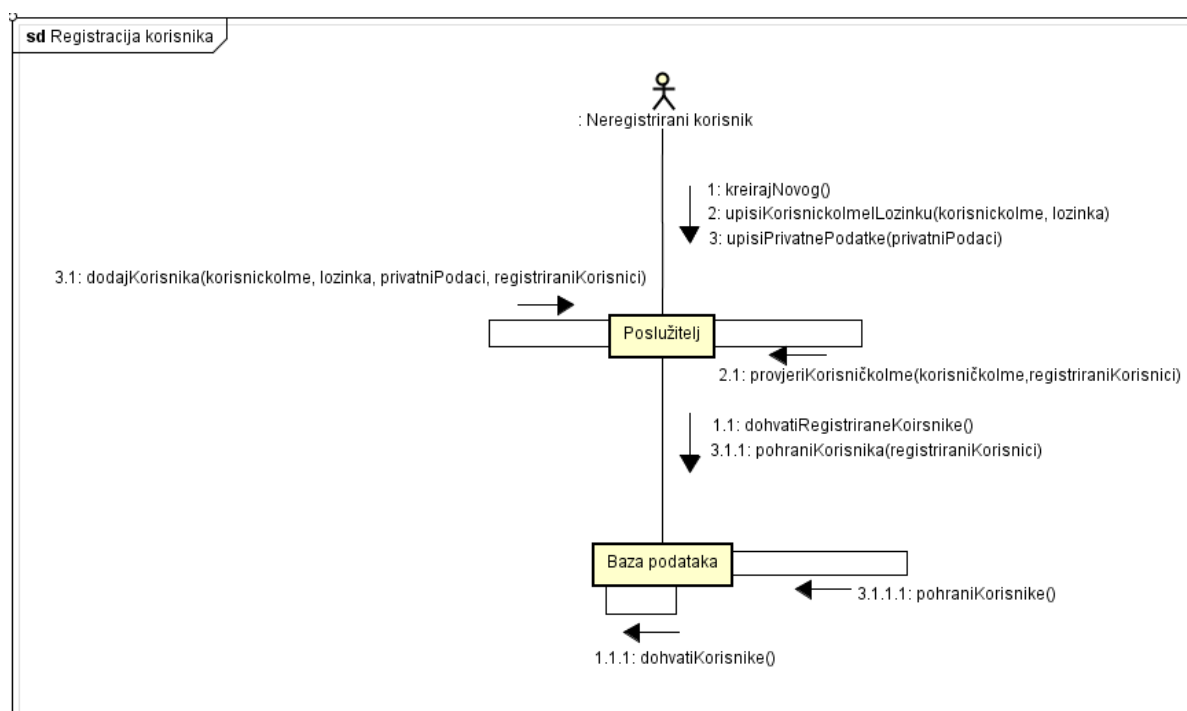
Slika 25 - Objektni dijagram

6.4 Ostali UML dijagrami

6.4.1 Komunikacijski dijagrami

Komunikacijski dijagrami su UML dijagrami koji prikazuju tok poruka između objekata u objektno orijentiranom sustavu te osnovne relacije između razreda i njihovih objekata. Ovi dijagrami obuhvaćaju dinamičko ponašanje sustava koje uključuje već spomenute poruke i uloge instanci danih razreda. Ovi dijagrami ne prikazuju vremenske odnose.

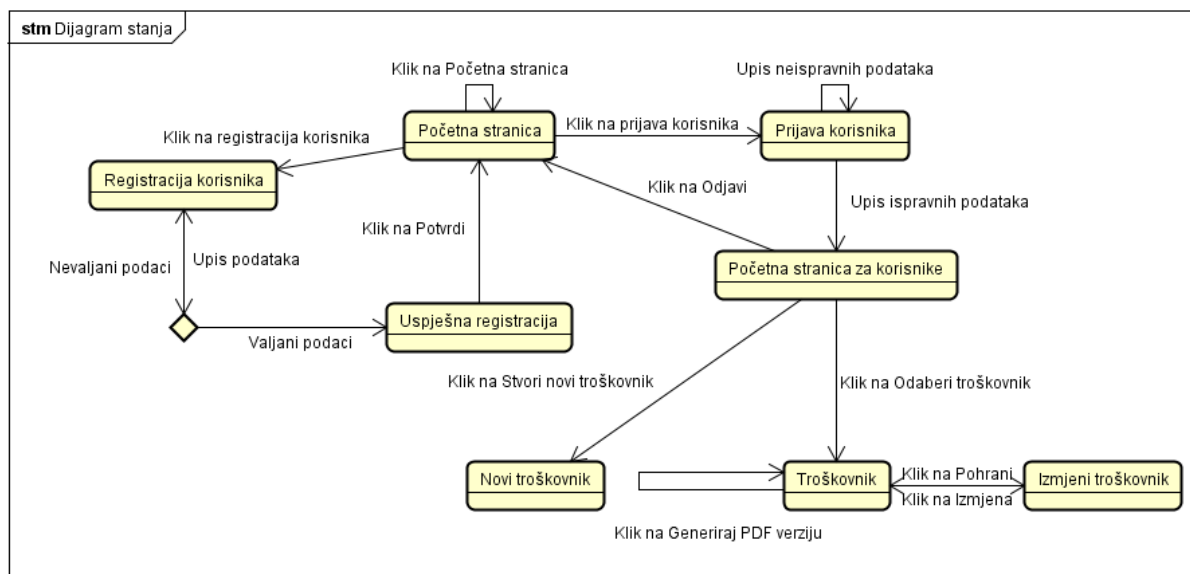
Sljedeći dijagram prikazuje registraciju korisnika u sustav. Prikazan je tok poruka te svi akteri. Neregistrirani korisnik šalje zahtjev sustavu za registraciju. Ukoliko su dani podaci valjani, sustav mu odobrava registraciju i pohranjuje ga u bazu podataka kao novog korisnika.



Slika 26 - Komunikacijski dijagram registracije novog korisnika

6.4.2 Dijagram stanja

Dijagram stanja opisuje dinamičko ponašanje jednog objekta u danom vremenu. Ovakav dijagram pogodan je za opis značajnijeg dinamičkog ponašanja izoliranog objekta a izlaz mu ne ovisi samo o trenutnim ulazima nego i o povijesti. Dijagram stanja se koristi za opis diskretnog ponašanja. Na ovom dijagramu prikazuje se sekvenca stanja objekta te prijelazi iz jednog stanja u drugo na temelju događaja.



Slika 27 - Dijagram stanja

7. Implementacija i korisničko sučelje

7. 1. Korištene tehnologije i alati

Navedi sve tehnologije i alate koji su primijenjeni pri izradi projekta te ih ukratko opisati; njihovo značenje i mjesto i način primjene.

7.2. Isječak programskog koda vezan za temeljnu funkcionalnost sustava

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

7.3. Ispitivanje programskog rješenja

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

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

7.4. Upute za instalaciju

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

7.5. Korisničke upute

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

8. Zaključak i budući rad

U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.

9. Popis literature

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

- 1 Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
- 2 Oblikovanje programske potpore, FER ZEMRIS, <http://www.zemris.fer.hr/predmeti/opp>
- 3 I. Sommerville, „Software engineering“, 8th ed, Addison Wesley, 2007.
- 4 T.C.Lethbridge, R.Langaniere, „Object-Oriented Software Engineering“, 2nd ed. McGraw-Hill, 2005.
- 5 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/>