Univerzitet Singidunum

Fakultet za informatiku i računarstvo

**Aplikacija za lične finansije**

* 16. maj – 10. jun 2022. –

Profesor : Violeta Tomaševič Studenti: Damjan Belojević, 2019200570,

Asistent : Petar Jakić Darja Nedić, 2019200813

Sadržaj

[1. Kratak opis projekta 3](#_Toc105769589)

[2. Postupak razvoja softvera 3](#_Toc105769590)

[2.1 Model procesa razvoja 3](#_Toc105769591)

[2.2 Plan projekta 3](#_Toc105769592)

[2.2.1 Kritične tačke 3](#_Toc105769593)

[2.2.2 Uloge članova projektnog tima 4](#_Toc105769594)

[3. Analiza zahteva 4](#_Toc105769595)

[3.1 Specifikacija zahteva 4](#_Toc105769596)

[3.1.1 Funkcionalni zahtevi 4](#_Toc105769597)

[3.1.2 Zahtevi u pogledu kvaliteta 5](#_Toc105769598)

[3.1.3 Projektna ograničenja 5](#_Toc105769599)

[3.2 Modelovanje ponašanja 5](#_Toc105769600)

[3.3 Interakcija korisnika sa softverom 6](#_Toc105769601)

[3.4 Validacija zahteva 8](#_Toc105769602)

[4. Projektovanje sistema 8](#_Toc105769603)

[4.1 Stil u projektovanju 8](#_Toc105769604)

[4.2 Strategije projektovanja 8](#_Toc105769605)

[4.2.1 Projektovanje arhitekture sistema 8](#_Toc105769606)

[4.2.2 Prikaz dela funkcionalnosti 9](#_Toc105769607)

[4.2.3 Projektovanje programa 10](#_Toc105769608)

[4.2.4 Završno projektovanje 11](#_Toc105769609)

[5. Implementacija softvera 12](#_Toc105769610)

[5.1 Lokalizacija ulaza i izlaza 12](#_Toc105769611)

[5.2 Struktura podataka 12](#_Toc105769612)

[5.3 Unutrašnja dokumentacija 14](#_Toc105769613)

[6. Testiranje softvera 15](#_Toc105769614)

[6.1 Jedinično testiranje 15](#_Toc105769615)

[6.2 Integraciono testiranje 18](#_Toc105769616)

[6.4 Završetak testiranja 19](#_Toc105769617)

[7. Isporuka softvera i obuka korisnika 20](#_Toc105769618)

[7.1 Uputstvo za korisnike 20](#_Toc105769619)

# 1. Kratak opis projekta

Tema ovog projekta je razvoj aplikacije za evidentiranje finansija. Aplikacija podržava više korisnika i mora da vodi striktno računa o tome da prijavljenom korisniku prikazuje samo podatke iz baze podataka koji se odnose na njega. Korisnik se na aplikaciju prijavljuje sa korisničkim imenom i lozinkom, kao i da samostalno kreira svoje transakcije i da definiše njihovo ime, kategoriju i datum transakcije. Takođe, korisnik može da definiše kategorije troškova i prihoda.

Prilikom evidentiranja prihoda, aplikacija automatski upisuje prihod na račun i automatski koriguje vrednost stanja na računu. Isto se dešava u slučaju evidentiranja rashoda. U oba slučaja, korisnik treba da evidentira jednu od svojih definisanih kategorija. Aplikacija treba da omogući prikaz svih transakcija od najnovije u vrhu prema najstarijoj na dnu. Obezbediti paginaciju prilikom prikaza svih transakcija.

Pored spiska u obliku tabele treba kreirati tri grafikona. Jedan grafikon predstavlja tzv. „Pie Chart“ za prikaz udela određene kategorije u prihodima, drugi za rashode i treći treba da prikaže liniju kretanja vrednosti stanja računa po datumima na X osi i iznosu stanja na Y osi (na mesečnom/godišnjem nivou). Grafički interfejs veb sajta treba da bude realizovan sa responsive dizajnom.

U nastavku ovog dokumenta biće predstavljen detaljan plan razvoja ovog projekta, prikaz implementacije programskog koda, kao i uputsvo za upotrebu datog softvera.

# 2. Postupak razvoja softvera

U narednom poglavlju će biti prikazan model i plan razvoja aplikacije za lične finansije.

## 2.1 Model procesa razvoja

Radi lakše komunikacije u okviru razvojnog tima i dostizanja zajedničkog shvatanja cilja ovog projekta, kreiran je model procesa razvoja. Zbog svog visokog nivoa apstrakcije, lake primene i jednostavnosti, izabrana tradicionalna metoda modelovanja jeste kaskadni model.

## 2.2 Plan projekta

### 2.2.1 Kritične tačke

* Prikupljanje zahteva (rok: 20. maj 2022.)
* Validacija zahteva (rok: 22. maj 2022.)
* Razvoj plana implementacije (rok: 24. maj 2022.)
* Implementacija softvera (rok: 29. maj 2022.)
* Testiranje kvaliteta (rok: 7. jun 2022.)
* Isporuka softvera (rok: 10. jun 2022.)

### 2.2.2 Uloge članova projektnog tima

1. Damjan Belojević
   1. Projektant
   2. Programer (*front-end, back-end* za statistički deo aplikacije)
   3. Inženjer za testiranje
   4. Izrada dokumentacije
2. Darja Nedić
   1. Analitičar zahteva
   2. Programer (baza podataka, *back-end* (korisnički deo aplikacije)
   3. Isporuka sistema
   4. Izrada dokumentacije

# 3. Analiza zahteva

Analiza zahteva je prvi korak u razvoju softvera. U nastavku je prikazan proces prikupljanja zahteva, u saradnji sa naručiocem, kao i njihova specifikacija i verifikacija.

## 3.1 Specifikacija zahteva

Nakon višednevnog razgovora sa naručiocem, pregleda dokumentacije prethodnih projekata i upoznavanja prethodnih sistema, ustanovili smo projektne zahteve, i izvršili njihovu specifikaciju, koja će biti navedena u nastavku.

### 3.1.1 Funkcionalni zahtevi

* Korisniku treba da bude omogućeno kreiranje naloga ličnim odabirom lozinke i jedinstvenog korisničkog imena, na koji se može dalje logovati korišćenjem istih;
* U slučaju da korisnik prilikom prijavljivanja pokuša da izabere korisničko ime koje je već zauzeto, sistem treba da ga spreči i izbaci adekvatnu poruku;
* Nakon kreiranja profila, korisnik treba da bude upućen na stranicu gde može da kreira novi račun;
* Račun se kreira unošenjem njegovog naziva, kao i valute (RSD ili EUR);
* Pri ulasku na početnu stranu korisniku treba da se prikazuje naziv i stanje njegovog trenutno odabranog računa, kao i dugme za dodavanje nove transakcije;
* Pored navedenih informacija, na početnoj strani treba da bude dijagram koji prikazuje odnos prihoda i rashoda trenutnog računa;
* Na posebnoj sekciji početne strane korisniku treba da bude obezbeđen prikaz svih transakcija, podeljenih na prihode i rashode;
* Prilikom odabira opcije za dodavanje transakcije, korisnik može odabrati koji tip transakcije želi da unese(prihod ili rashod), i zatim da evidentira datu transakciju unosenjem njenog naziva, kategorije i vrednosti;
* Tokom dodavanja transakcije, korisnik treba da ima opciju da sopstveno definiše kategorije za dati tip transakcije;
* Sistem korisniku treba da omogući i prikaz statistike za tekući mesec potrošnje i zarade po kategorijama, prikazanih pomoću *bar-chart* i *pie-chart* dijagrama, kao i prikaz promene stanja računa po danima pomoću line chart dijagrama;
* Pored statistike, korisniku treba da se omogući pristup arhivi njegovih transakcija za period od početka korišćenja aplikacije, sa mogućnosti pregleda godišnje statistike prikazane putem *line-chart* dijagrama;

### Zahtevi u pogledu kvaliteta

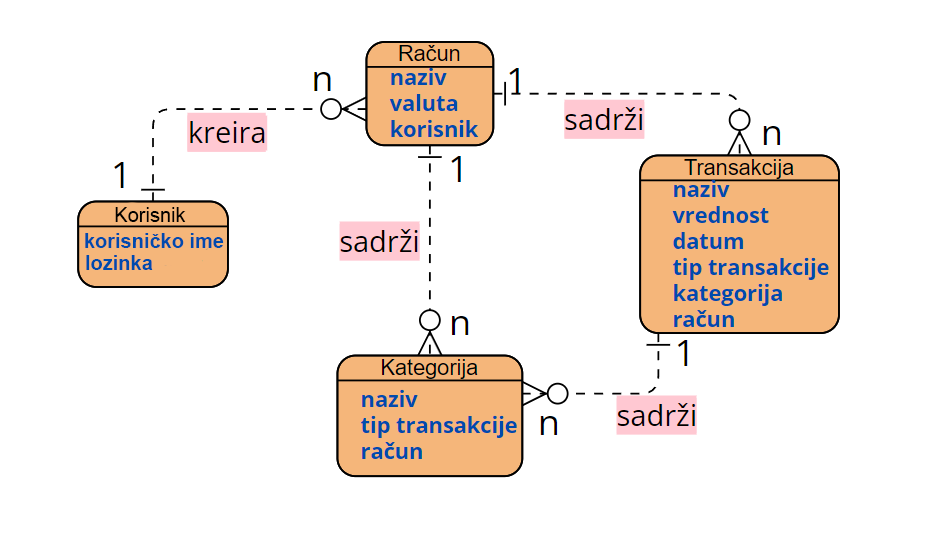
* Sistem je potrebno da isporučuje korisniku željene informacije sa minimalnim zakašnjenjem, i sa najbržim mogućim odzivom;
* U pogledu memorije, aplikacija bi trebalo da zauzima što manji kapacitet, da bi bila lako dostupna svim korisnicima, na svim uređajima;
* Aplikacija je, takođe, potrebno da ima intuitivan interfejs, i nizak nivo kompleksnosti korišćenja;
* Sa stanovišta bezbenosti, neophodno je da lični podaci i informacije o korisnicima budu nepristupačne za bilo koga osim za samog korisnika;
* Uneti podaci moraju biti tačni i dosledni, kao i nepromenjeni u odnosu na oblik u kom su uneti od strane korisnika;
* Sistem mora u svakom trenutku da obezbedi sigurnu kopiju korisničkih podataka da bi, u slučaju dolaska do greške ili do njihovog gubitka, mogao da obezbedi lak povratak na prvobitno stanje korisničkog profila;
* U slučaju pojave greške ili otkaza u softveru, sistem mora imati maksimalnu brzinu otklona istih;

### Projektna ograničenja

* Za realizaciju *front-enda* koristiće se *html/css/javascript*/*bootstrap/jQuery* tehnologija;
* Za realizaciju *back-enda* koristiće se *php* tehnologija;
* Za modelovanje i kreiranje baze podataka koristiće se *mysql* tehnologija;

## Modelovanje ponašanja

Grafička notacija za predstavljanje struktuiranog modela koju ćemo predstaviti u nastavku je ER diagram (*Entity-relationship diagram*).

Na diagramu (*slika 1*) su predstavljeni entiteti (žuto), njihovi atributi(plavo), veze između njih(roze), i kardinalnost tih veza.

Slika 1 - ER dijagram

Na gorepomenutom dijagramu se mogu uočiti četiri glavna entiteta, i to su Korisnik, Račun, Transakcije i Kategorije transakcija. Korisnik, koga opisuju atributi korisničko ime i lozinka, preko aplikacije kreira jedan ili više računa. Računi, koje opisuju atributi naziv, valuta, i ID korisnika kojem pripada, sadrže transakcije i korisnički definisane kategorije. Transakcije koje se nalaze na računu opisuju atributi naziv, vrednost, datum dodavanja transakcije, tip transakcije (prihod ili rashod), kategorija transakcije, kao i ID računa kojem pripada. Korisnički definisane kategorije karakterišu atributi naziv, tip transakcije za koji je definisana kategorija, kao i ID računa kojem pripadaju.

# 3.3 Interakcija korisnika sa softverom

U nastavku je prikazan dijagram slučaja korišćenja (*slika 2*) koji opisuje celokupnu interakciju korisnika sa sistemom, kao i detaljan opis jednog dela sistema, odnosno slučaja dodavanja nove transakcije(*slika 3*).

Diagram

Description automatically generated

Slika - Dijagram slučaja korišćenja

Text

Description automatically generated

Slika - Dodavanje transakcije

# 3.4 Validacija zahteva

Nakon dugog razmatranja i prolaska kroz dokumentaciju, formalne inspekcije i timskog ocenjivanja zahteva, došli smo do zaključka da su navedeni specifizirani zahtevi potpuni, ispravni, jednostavni za razumevanje, nedvosmisleni, i dosledni zahtevima korisnika.

# 4. Projektovanje sistema

Nakon prikupljenih zahteva u razgovoru sa naručiocem, njihove specifikacije i verifikacije, sledeća faza je projektovanje sistema, gde je potrebno prevesti prethodno sažeti problem, u njegovo rešenje.

## 4.1 Stil u projektovanju

Kao strukturalnu organizaciju ovog softverskog sistema izabraćemo Klijent-Server arhitekturu, koja je veoma pogodna za web aplikacije, kao i aplikacije koje često komuniciraju sa bazom podataka. Veći broj korisnika u isto vreme može zahtevati veliki broj usluga od strane servera, koji su konstantno aktivni i osluškuju zahteve klijenata.

## 4.2 Strategije projektovanja

Projektovanje ovog sistema izvršeno je u dva nivoa, na nivou arhitekture, i na nivou programa.

### 4.2.1 Projektovanje arhitekture sistema

Nakon prikupljenih informacija iz specifikacije zahteva kreirali smo određene komponente koje će predstavljati srž sistema i temelj za kreiranje softverskog koda kao i veze između njih. Kao komponente odredili smo Korisika, Račun, Transakciju i Kategorije transakcija. Najbolje možemo razumeti značaj ovih komponenata, kao i njihove međusobne relacije prikazom na dijagramu komponenata(*slika 4*).

Diagram

Description automatically generated

Slika - Dijagram komponenata

### 4.2.2 Prikaz dela funkcionalnosti

U nastavku ovog poglavlja biće ilustrovan deo funkcionalnosti sistema dijagramom aktivnosti(*slika 6*). Prikazana funkcionalnost je proces prijavljivanja korisnika na platformu.

Prvi korak jeste sam ulazak u aplikaciju. Sledeći korak je postavljanje pitanja da li korisnik ima već kreiran nalog, ili je potrebno da izvrši dodavanje novog ličnog naloga.

U slučaju da korisnik već ima nalog, on se loguje na aplikaciju unošenjem svog korisničkog imena i lozinke, i zatim klikom na dugme *Log in* završava proces logovanja.

U slučaju da korisnik nema nalog, klikom na odgovarajući link *Sign in* biće usmeren na stranicu gde može uneti željeno korisničko ime i lozinku, i da se prijavi na aplikaciju klikom na dugme *Sign in*. Sistem nakon toga proverava da li u bazi već postoji to korisničko ime, i ukoliko postoji, korisnik biva obavešten da mora da izabere drugo korisničko ime koje je jedinstveno.

Prilikom uspešne prijave ili logovanja na aplikaciju, korisnik biva upućen na početnu stranicu aplikacije gde može da pregleda svoje transakcije i informacije o trenutnom računu.

Diagram

Description automatically generated

Slika 6 - Dijagram aktivnosti

### 4.2.3 Projektovanje programa

Tehnologija koja će se koristiti za upravljanje bazom podataka jesu relacione baze i MySQL. Tehnologija relacionih baza je pogodna za ovu aplikaciju zbog načina organizovanja podataka. Najadekvatnije je čuvati transakcije tabelarno, a ne u vidu kolekcija.

Za kreiranje *back-enda* biće korišćena PHP tehnologija, koja veoma dobro funkcioniše zajedno sa MySQL-om.

Za određene delove *front-end*-a biće korišćena bootstrap tehnologija (grafikoni za prikaz statistike). Razlog korišćenja bootstrap-a je ušteda vremena kreiranja kompleksnih komponenata, ali takođe i sigurnost korišćenja već istestiranih komponenata. Za ostatak *front-end*-a biće korišćena *html/css/javascript/jQuery* tehnologija.

Na slici (*slika 5*) u nastavku je prikazana predviđena organizacija direktorijuma i datoteka koje će sadržati izvorni kod. Ova organizacija će poslužiti programerima kao alat za navođenje kroz sam razvoj i implementaciju softvera.

A picture containing graphical user interface

Description automatically generated

Slika 5 - Struktura datoteka i direktorijuma

### 4.2.4 Završno projektovanje

U aplikaciji koja radi sa transakcijama korisnika velika pažnja treba biti posvećena formatima podataka. Treba biti obezbeđeno ograničenje unosa pojedinih podataka preko ENUM vrednosti (biranje valute) i takođe unosa broja karaktera za tekstualne podatke.

Zbog velike količine podataka koje sadrži ova aplikacija, potrebno je obratiti dosta pažnje ka izbegavanju redudantnosti podataka. Konkretno, ukoliko već postoji određena kategorija, korisnik ne bi trebalo da ima mogućnost da je ponovo unese u bazu, kako ne bi dolazilo do opterećenja. Takođe, potrebno je ograničiti sve transakcije po računima u kojima su one sadržane, kako ne bi došlo do mešanja podataka između različitih računa, ili, još gore, različitih korisnika.

# 5. Implementacija softvera

Nakon verifikacije prikupljenih zahteva, i uspešnog projektovanja sistema, naredni korak u razvoju softvera jeste prevođenje samih projektnih zahteva u izvorni kod.

## 5.1 Lokalizacija ulaza i izlaza

Prilikom razvoja softvera potrebno je izvršiti lokalizaciju ulaznih i izlaznih podataka, odnosno kreirati odvojene komponente gde korisnik može unositi ulazne podatke, i komponente gde sistem prikazuje izlazne podatke korisniku;

Pri kreiranju novog profila, logovanju na postojeći profil, kao i pri kreiranju novih računa, transakcija, i kategorija, sistem prima ulazne podatke od korisnika putem forme;

* Ulazni podaci za korisničko ime, lozinku, naziv računa, naziv transakcije i naziv kategorije su tekstualnog tipa;
* Ulazni podaci za vrednost transakcije su brojevnog tipa;
* Ulazni podaci za valutu računa su tipa ENUM i sadrže predefinisane vrednosti RSD ili EUR;
* Izazni podaci na početnoj strani dinamički prikazuju trenutni mesec i godinu(date tip), korisničko ime, naziv računa(tekstualni podaci), valutu odabranog računa(ENUM tip), kao i trenutno stanje računa(brojevni tip) koji se dinamički ažurira svakom novom dodatom transakcijom;
* Izlazni podaci na *bar-chart* dijagramu početne strane dinamički prikazuju ukupno stanje svih prihoda i svih rashoda na trenutnom računu, i automatski se ažuriraju svakom dodatom transakcijom;
* Izlazni podaci u sekciji za transakcije daju tabelarni prikaz svih transakcija odvojenim na prihode i rashode, sa nazivom transakcije, nazivom kategorije(tekstualni podaci), vrednosti transakcije(brojevni tip), i datumom unosenja te transakcije(date tip);
* Izlazni podaci u sekcijama za statisticki pregled i pregled arhive transakcija na datom računu, korisniku treba biti prikazani u vidu *bar-chart*, *pie-chart* i *line-chart* dijagrama sa dinamički promenljivim vrednostima za trenutni mesec ili odabranu godinu;

## 5.2 Struktura podataka

Struktura podataka koja je izabrana za ovu aplikaciju jeste sistem relacionih baza podataka. Softver konstantno komunicira sa bazom podataka, u kojoj se čuvaju svi podaci o korisnicima, njihovim računima, transakcijama i kategorijama. Pregled ovih komponenata i odnosa između njih najbolje ćemo razumeti prikazom modela baze podataka (*slika 7*).

Na slici možemo uočiti da se baza sastoji iz četiri tabele, tabele ’*user’* u kojoj se čuvaju podaci o korisnik, tabele ’racun’ u kojoj se čuvaju podaci o računima, tabele ’transakcija’ u kojoj se čuvaju podaci o unetim transakcijama, i tabele ’kategorija’ u kojoj se čuvaju podaci o korisnički definisanim kategorijama. Na slici je takođe ilustrovan odnos pomenutih tabela, odnosno relacije u kojima se one nalaze.

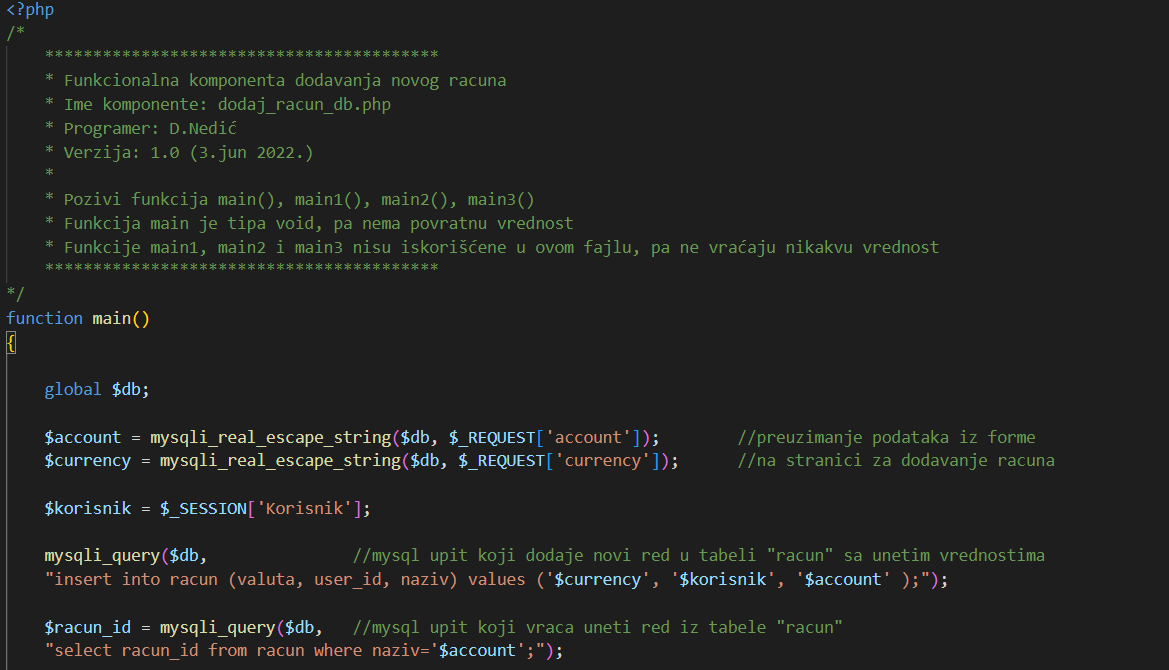
Diagram

Description automatically generated

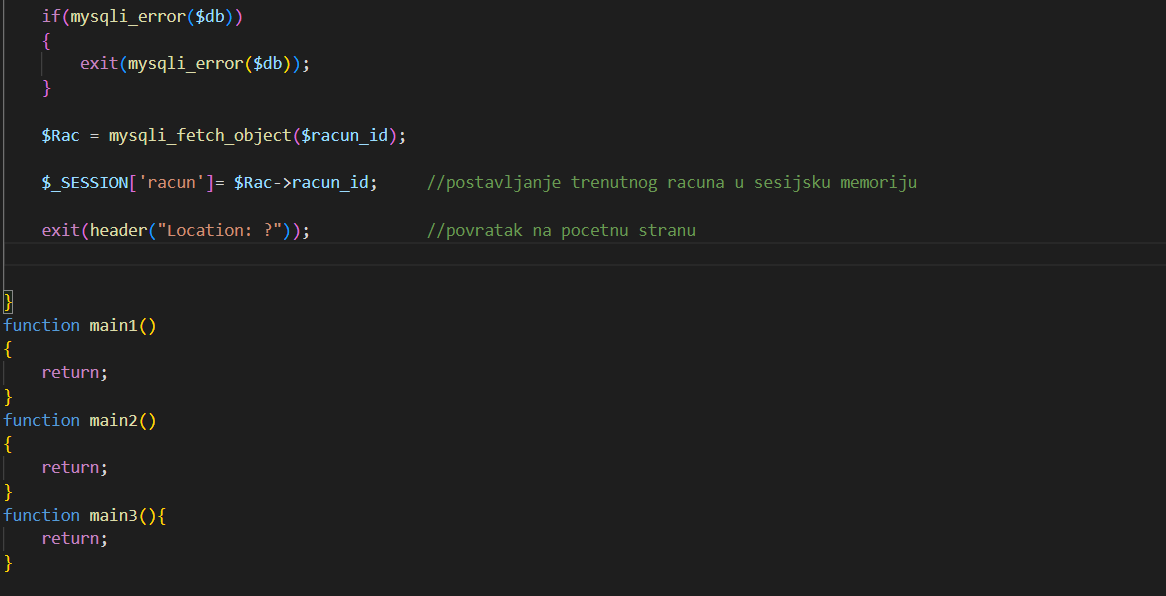
Slika 7 - Model baze podataka

## 5.3 Unutrašnja dokumentacija

U nastavku (*slika 8* i *slika 9*) će biti prikazan izvorni programski kod funkcionalne komponente za dodavanje računa, sa unutrašnjom dokumentacijom koja obuhvata komentare, kao i informacije u zaglavlju dokumenta.



Slika 8 - Dodavanje računa 1



Slika 9 - Dodavanje računa 2

# 6. Testiranje softvera

Nakon izvršene implementacije prešli smo na testiranje pojedinačnih delova aplikacije. Posle jediničnog testiranja, prešli smo na integraciono testiranje korišćenjem strategije “od vrha ka dnu”.

## 6.1 Jedinično testiranje

Za pronalazak grešaka koristili smo nekoliko postupaka. Testiranje svakog modula počeli smo sa prolaskom kroz kod u vidu letimičnog pregleda. Svako je svoj deo koda predstavljao ostatku tima nakon čega su se vodile diskusije.

Nakon pregleda koda generisali smo test primere sa datim ulaznim podacima kako bismo mogli dobijeni izlaz da upoređujemo sa očekivanim izlazom.

Takođe za testiranje ruta i zahteva naše aplikacije koristili smo Postam platformu.

**Prijava korisnika**

Prilikom testiranja prijavljivanja korisnika naišli smo na problem da korisnik koji nema nalog uspešno izvršava prijavu na aplikaciji. Dodali smo uslov koji onemogućava izvršavanje takve akcije time što proverava da li takav korisnik već postoji u bazi. Ukoliko ne postoji, aplikacija korisnika vraća na stranicu za prijavljivanje sa ispisanom porukom kao sto se može videti na slici (*slika 10*).

Graphical user interface, application

Description automatically generated

Slika 10 – Prijava korisnika

**Dodavanje transakcije**

Kao metod testiranja ovog modula aplikacije koristili smo metod “crne kutije”. Konkretno logika je bila takva da ukoliko korisnik izabere “prihod” kao tip transakcije, ukupna suma na računu će se povećati za tu vrednost. Suprotno tome, ako korisnik kao tip transakcije izabere “rashod”, ukupna suma treba da se smanji za tu vrednost. Ovu funkciju smo istestirali preko tehnike podele na klase ekvivalencije.

Ulazne podatke smo podelili na legalne i nelegalne klase. Za oba tipa transakcija je važilo :

* **Legalna klasa** : svi pozitivni brojevi
* **Nelegalna klasa** : 0 i svi negativni brojevi

Unosom negativne vrednosti poremetila bi se vrednost stanja na računu, a izvršavanje transakcije od 0 RSD/EUR ne bi imala smisla.

**Dodavanje računa**

Testiranjem dela aplikacije za dodavanje računa naleteli smo na grešku prilikom definisanja rute. U ovu svrhu smo koristili Postman platformu. Kliktanje na dugme “dodaj račun” unutar početne stranice dovodio je do 404 errora kao što se vidi na slici (*slika 11*).

Text

Description automatically generated

Slika 11 – 404 error

Prvo smo proveravali rutu preko Postman-a. Kao što se vidi na slici(*slika 12*) GET metoda normalno funkcioniše (Status 200 OK).

Graphical user interface, text, application, email

Description automatically generated

Slika 12 – Postam rezultat GET metode

Greška je na kraju nađena u kodu unutar “href” atributa. Zaboravljen je početni deo rute koji je bio neophodan kako bi onda funkcionisala (“?akcija=“). Greška se može videti na slici (*slika 13*)

Text

Description automatically generated

Slika 13 – Greška u definisanju rute

**Statistika računa**

Na stranici sa statistikom računa postojao je problem prikaza potrošnje na mesečnom nivou. Kao što se vidi na slici (*slika 14*) prikazivala se samo vrednost za jedan dan u mesecu.

Chart, line chart

Description automatically generated

Slika 14 – Statistika računa

Nakon detaljnog pregleda koda primetili smo da je u pitanju sintaksna greška unutar petlje. Petlja je služila da prikupi sve troškove iz baze za dati mesec. Istostavljanjem “break“ naredbe u kodu došlo je do upisa pogrešnih vrednosti koje su se zatim koristile za kreigranje grafa. Ispravka greške se može videti na slici *(slika 15*).

Text

Description automatically generated

Slika 15 – Prikaz ispravke greške u kodu

## 6.2 Integraciono testiranje

Nakon završenog jediničnog testiranja krenuli smo izvršavanje integracionog testiranja. U ovu svrhu opredelili smo se za integraciju od vrha ka dnu. Hijerarhija koja prikazuje redosled u kome su moduli testirani data je na slici (*slika 16*).

Diagram

Description automatically generated

Slika 16 – integracija od vrha ka dnu

Komponente smo u process testiranja uključivali strategijom “po širini” tj. uključivali smo komponente po nivoima koji su označeni crvenim isprekidanim linijama na slici (*slika 16*). Drugim rečima prvo smo tesitali da li se sve ispravno prikazuje na početnoj stranici. Zatim smo vršili integraciju modula kreiranje profila, dodavanje transakcije, dodavanje računa i prikaz transakcija. I na samom kraju integracija modula dodavanje kategorije, arhiva i statistika.

## 6.4 Završetak testiranja

Sistem smo testirali sve dok nismo pronašli određeni broj grešaka u našoj aplikaciji. U našem primeru izabrali smo da to bude 2 greške po nedelji. U grafiku zavisnosti na slici (*slika 17*) može se videteti broj pronađenih grešaka u svakoj nedelji od početka testiranja. Odlučili smo da prekinemo testiranje na početku 4 nedelje jer smo distigli željeni broj grešaka.

Chart, line chart

Description automatically generated

Slika 17 – Dijagram zavisnosti

# 7. Isporuka softvera i obuka korisnika

Nakon uspešno završene faze testiranja možemo reći da softver zadovoljava sve projektne funkcionalne i nefunkcionalne zahteve, i da se može preći na fazu isporuke softvera u korisničko okruženje, i na obuku korisnika. U nastavku prilažemo kratko uputstvo za korišćenje aplikacije kojim se korisnici mogu voditi radi lakšeg i bržeg snalaženja i upoznavanja platforme.

## 7.1 Uputstvo za korisnike

U narednih nekoliko koraka ćemo dati kratko uputstvo korišćenja jednog dela aplikacije za nove korisnike.

1. Pri prvom ulasku na aplikaciju prikazaće Vam se stranica za logovanje(*slika 18*);

Graphical user interface, text, application

Description automatically generated

Slika 1 - Log in stranica

1. Ukoliko nemate već napravljen nalog, potrebno je da kliknete na link *Sign up* koji će Vas odvesti na stranicu za prijavljivanje;
2. Na stranici za prijavljivanje putem forme možete uneti odabrano korisničko ime(*username*) i lozinku(*password*) za Vaš novi korisnički profil i prijaviti se klkom na dugme *Sign up* (*slika 19*);

Graphical user interface, application

Description automatically generated

Slika 1 - Sing up stranica

1. Svaki naredni put kada se budete logovali, to možete raditi na stranici za logovanje(*Log-in*) unošenjem Vašeg korisničkog imena i lozinke, i zatim klikom na dugme *Log in*;
2. Nakon prvog ulaska na novi profil, pošto nemate kreiran račun, aplikacija će Vas odvesti na stranicu gde ga možete kreirati(*slika 20*);

Graphical user interface, application

Description automatically generated

Slika - Stranica za dodavanje računa

1. Na stranici za kreiranje računa, unošenjem naziva(*account* *name*) i odabirom valute(*currency*) putem forme, i zatim klikom na dugme *Submit* možete dodati svoj prvi račun;
2. Nakon toga ćete biti sprovedeni na početnu stranu(*home page*), na kojoj će vam biti prikazane osnovne informacije o vašem računu(*slika 21*):  
   - Trenutni datum i godina za koji su prikazani obračuni i stanje računa;  
   - Vaše korisničko ime;  
   - Naziv trenutno odabranog računa(*account name*);  
   - Valuta trenutno odabranog računa(*account currency*);  
   - Stanje trenutno odabranog računa(*account balance*);  
   - Grafikon sa prikazom odnosa ukupnih prihoda(*income*) i rashoda(*expense*) na računu;

Chart

Description automatically generated

Slika - Početna stranica

1. Pored osnovnih informacija na početnoj strani se nalaze dva dugmeta, dugme za dodavanje transakcije (*Add transaction*)i dugme za pregled svih transakcija (*All transactions*);
2. Klikom na dugme *Add transaction* na početnoj strani aplikacija će Vam izbaciti *pop-up* prozor sa pitanjem da li želite da dodate transakciju tipa prihod(*income*) ili tipa rashod(*expense*) (*slika 22*);

A screenshot of a computer

Description automatically generated with low confidence

Slika - Odabir tipa transakcije

1. Odabirom željenog tipa transakcije bićete odvedeni na stranicu za dodavanje transakcije(*slika 23*);

Graphical user interface, application, website

Description automatically generated

Slika - Dodavanje transakcije

1. Za kreiranje transakcije potrebno u polja forme uneti njen naziv(*name*), vrednost(*value*) i kategoriju(*category*);
2. Ukoliko na Vašem trenutno odabranom računu nemate kreiranih kategorija za odabrani tip transakcije, ili ako želite da proširite skup već postojećih kategorija, možete to učiniti klikom na link *Add category*;
3. Klikom na link za dodavanje kategorije bićete odvedeni na stranicu gde možete dodati kategoriju za odabrani tip transakcije unošenjem njenog naziva(*name*) u polje forme i zatim klikom na dugme *Submit* (*slika 24*);

Graphical user interface, application

Description automatically generated

Slika - Dodavanje kategorije

1. Nakon što ste kreirali kategoriju možete kreirati transakciju unošenjem naziva(*name*), vrednosti(*value*) i odabirom kategorije putem *drop-down* menija, i zatim klikom na dugme *Submit*;
2. Ukoliko želite da pregledate unetu transakciju, kao i bilo koje prethodno unete transakcije, to možete uraditi klikom na dugme *All transactions*;
3. Klikom na dugme *All transactions* bićete usmereni na sekciju početne stranice gde možete videti tabelarni prikaz svih transakcija, podeljenim na tabove za prihode(*income*) i rashode(*expense*) (*slika 25*);

Table

Description automatically generated

Slika - Prikaz tabele rashoda

1. U tabeli sa transakcijama obezbeđena je paginacija za lakši pregled velikog broja stavki, kao i filtracija i sortiranje po svim parametrima;
2. Transakcije možete brisati klikom na ikonicu za brisanje, nakon čega će vas aplikacija pitati da li ste sigurni da želite da izvršite tu izmenu;
3. Radi lakše navigacije u aplikaciji sa strane se nalazi navigacioni bar sa ikonicama za početnu stranu, stranicu za pregled statistike i arhivu (*slika 26*);

Icon

Description automatically generated with low confidence

Slika - Navigacioni bar

1. Na dnu navigacionog bara se nalazi ikonica Vašeg profila, i klikom na nju možete videti prikaz svih računa na Vašem profilu, link ka dodavanju novog računa, link za odjavljivanje(*Log out*) i Vaše korisničko ime(*slika 27*);

Graphical user interface, application

Description automatically generated

Slika - Informacije o korisničkom profilu

1. Klikom na ikonicu za statistiku, koja se nalazi druga po redu u navigacionom baru, odlazite na stranicu za pregled statistike za trenutni mesec;

Chart

Description automatically generated

Slika 28 - Udeo kategorija u rashodima

Chart, sunburst chart

Description automatically generated

Slika 29 - Udeo kategorija u prihodima

Chart, line chart

Description automatically generated

Slika 30 - Potrošnja po danima za dati mesec

1. Na stranici za pregled statistike aplikacija vam prikazuje raspored udela kategorija za rashode putem *bar-chart* dijagrama(*slika 28*), raspored udela kategorija za prihode putem *pie-chart* dijagrama(*slika 29*), kao i prikaz dnevne potrošnje za taj mesec putem *line-chart* dijagrama(*slika 30*);
2. Klikom na ikonicu za arhivu, koja se nalazi treća po redu u navigacionom baru, odlazite na stranicu za pregled arhive;

Chart, line chart

Description automatically generated

Slika 31 - Mesečna potrošnja za datu godinu

1. Na stranici za preged arhive aplikacija vam prikazuje prikaz mesečne potrošnje za trenutnu godinu putem *line-chart* dijagrama(*slika 31*), kao i opciju da odaberete neku od prethodnih godina u kojima ste imali transakcije za prikaz na dijagramu;