



Internacionalna poslovno-informaciona akademija Tuzla

Visoka škola za savremeno poslovanje, informacione tehnologije i tržišne komunikacije

Program za praćenje finansija u Javi (Projekat iz predmeta Programiranje u Javi)

Smjer/Godina studija:

Informatika i računarstvo / treća godina

Autor:

Ajla Halilčević

Tuzla, decembar, 2025. godina

1. UVOD

Ovaj projekat predstavlja desktop aplikaciju FinanceTracker, napravljenu u programskom jeziku Java uz korištenje grafičkog okruženja Swing i baze podataka MongoDB. Rađena je u IntelliJ IDEA Community Edition 2025.2.5.

Cilj ove aplikacije je omogućiti korisniku jednostavno praćenje ličnih finansija kroz unos prihoda i rashoda, te pregled ukupnih suma i stanja. Pored osnovnih funkcija dodavanja transakcija, aplikacija podržava:

- Uređivanje postojećih transakcija;
- Brisanje transakcija;
- Rad sa kategorijama;
- Export/izvoz izvještaja u tekstualni fajl;
- Automatsko računanje salda.

Aplikacija je napravljena prema smjernicama iz zadatka za Projekat 1 iz predmeta Programiranje u Javi.

2. KAKO POKRENUTI APLIKACIJU

Da bi aplikacija radila ispravno potrebno je instalirati Javu (preporučljivo Java 17), instalirati IntelliJ IDEA i otvoriti projekat, zatim provjeriti da je MongoDB instaliran i da servis radi na podrazumijevanom portu **27017**. U IntelliJ-u pronaći klasu Main.java i pokrenuti aplikaciju klikom na *Run* dugme.

Nakon pokretanja otvara se Swing prozor u kojem korisnik može unositi i prikazivati transakcije.

3. ARHITEKTURA RJEŠENJA

Aplikacija je organizovana u nekoliko jasno odvojenih klasa, od kojih svaka ima svoju odgovornost. U nastavku su navedene najvažnije komponente.

3.1. MongoDBConnection

Ova klasa služi kao centralno mjesto za uspostavljanje veze sa MongoDB bazom podataka. Koristi se Singleton pristup - konekcija se kreira samo jednom.

Glavne karakteristike:

Čuva URI baze mongodb: //localhost:27017;

Omogućava pristup bazi *financeTrackerDB*;

Vraća instancu *MongoDatabase* koja se koristi u ostatku aplikacije.

3.2. Transaction

Transaction je model klase koji predstavlja jednu transakciju u aplikaciji i u bazi.

Sadrži:

- ObjectId id - MongoDB identifikator dokumenta;
- String type - Prihod ili Rashod;
- Double amount - iznos;
- String description - opis
- String category - kategorija.

Zadužen je za pretvaranje objekata u MongoDB dokument:

```
public Document toDocument()
```

3.3. TransactionManager

To je klasa koja predstavlja komunikaciju sa bazom i "poslovnu logiku". Sve operacije nad transakcijama se obavljaju ovdje.

Glavne funkcije:

- Dodavanje transakcije;
- Čitanje svih transakcija;
- Ažuriranje transakcija;
- Brisanje transakcija;
- Računanje ukupnog prihoda, rashoda i salda;
- Sabiranje iznosa po kategorijama.

Ovdje se koristi MongoDB kolekcija *transactions*, iz koje se svi dokumenti čitaju i obrađuju.

3.4. FinanceTrackerForm (GUI)

Ova klasa predstavlja kompletnu Swing formu i interakciju s korisnikom.

Sadrži:

- Polja za unos vrste, iznosa, opisa i kategorije;
- Dugmad za izračunavanje, ažuriranje, brisanje i export/izvoz;
- Tabelu sa svim transakcijama;
- Prikaz ukupnih suma (Prihod, Rashod, Stanje).

Najvažnije metode:

- *loadDataIntoTable()* - učitava sve podatke u GUI tabelu;
- *updateSummary()* - izračunava i prikazuje zbir prihoda, rashoda i stanja;
- *updateTransaction()* - mijenja postojeću transakciju u bazi;
- *deleteTransaction()* - briše selektovani red.

```
public FinanceTrackerForm() { 1 usage
    manager = new TransactionManager();

    loadDataIntoTable();
    updateSummary();
}
```

3.5. Main

Pokreće aplikaciju tako što kreira: *new JFrame* ("Praćenje ličnih finansija") i postavlja *FinanceTrackerForm.mainPanel* kao sadržaj prozora.

4. FUNKCIONALNOST APLIKACIJE

U nastavku su opisane sve funkcionalnosti implementirane prema projektnom zadatku.

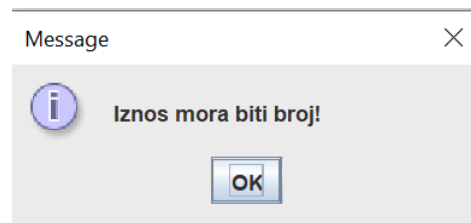
4.1. Dodavanje/Izračun transakcija

Korisnik unosi sledeće:

- Tip transakcije (Prihod / Rashod);
- Iznos prihoda/rashoda;
- Opis prihoda/rashoda;
- Kategoriju.

Nakon klika na “Izračunaj” podaci se validiraju i čuvaju se u MongoDB kolekciji, tabela se osvježava i zatim se izračunavaju nove sume.

Ako je unesen nevažeći broj, korisnik dobija poruku:



4.2. Ažuriranje transakcije

Korisnik izabere red iz tabele koji želi ažurirati, podaci se automatski učitaju u polja. Nakon uređivanja vrijednosti i klika na “Ažuriraj”, transakcija se pronalazi u bazi putem njenog ObjectId. Poslije toga podaci se mijenjaju i spremaju, tabela i sume se ponovo učitavaju.

4.3. Brisanje transakcije

Klikom na dugme “Obriši” prikazuje se dijalog za potvrdu, ako korisnik potvrdi onda se dokument briše iz baze i tabela se osvježava.

4.4. Pregled i kategorije

Svaka transakcija pripada jednoj od ovih kategorija: Plata, Hrana, Racuni, Zabava, Prijevoz, Ostalo.

Kategorije se prikazuju kao grafičko polje izbora i kao kolona u tabeli.

4.5. Export izvještaja

Aplikacija generiše tekstualni fajl koji sadrži:

- Ukupni prihod;
- Ukupni rashod;
- Stanje;
- Sumarizirane iznose po kategorijama.

Fajl se kreira u root direktoriju projekta pod nazivom *izvjestaj.txt* i izgleda ovako:

```
Ukupni prihod:100.0
Ukupni rashod:35.0
Stanje:65.0

Suma po kategorijama:
Plata:100.0
Hrana:10.0
Racuni:25.0
Zabava:0.0
Prijevoz:0.0
Ostalo:0.0
```

5. IZGLED APLIKACIJE

1. Glavni prozor aplikacije:

The screenshot shows the main window of the application. It has a title bar with the text 'Praćenje ličnih finansija' and standard window controls. The main area is titled 'PRAĆENJE LIČNIH FINANSIJA'. It contains several input fields and buttons. The first input field is labeled 'Unesite iznos vašeg prihoda/rashoda:'. The second is labeled 'Opišite ovaj izvor prihoda/rashoda:'. Below these are two dropdown menus for 'Kategorija:' with 'Racuni' and 'Prihod' selected. There are four buttons: 'Izračunaj', 'Ažuriraj', 'Obriši', and 'Export'. Below the buttons is a table with 4 columns: 'Vrsta', 'Iznos', 'Opis', and 'Kategorija'. The table contains three rows of data. At the bottom, there are three lines of text: 'Prihod: 100.0', 'Rashod: 35.0', and 'Stanje: 65.0'.

Vrsta	Iznos	Opis	Kategorija
Prihod	100.0	plata	Plata
Rashod	10.0	plata	Hrana
Rashod	25.0	plata	Racuni

Prihod: 100.0
Rashod: 35.0
Stanje: 65.0

2. Unešene transakcije, primjer kategorija i korištenja prihoda/rashoda:

This screenshot is identical to the first one, but the 'Kategorija:' dropdown menu is now set to 'Plata' instead of 'Racuni'. The table and summary text at the bottom remain the same.

Vrsta	Iznos	Opis	Kategorija
Prihod	100.0	plata	Plata
Rashod	10.0	plata	Hrana
Rashod	25.0	plata	Racuni

Prihod: 100.0
Rashod: 35.0
Stanje: 65.0

6. ZAKLJUČAK

Projekat uspješno obavlja sve zahtjeve iz upustva. Aplikacija omogućava jednostavno i intuitivno praćenje ličnih finansija koristeći tehnologije MongoDB i Java Swing GUI.

Svi zahtjevi su realizovani:

- Unos, prikaz i uređivanje podataka;
- Brisanje i validacija;
- Rad s kategorijama;
- Kreiranje izvještaja i stabilna komunikacija s bazom.

7. GITHUB LINK

https://github.com/AjlaHalilcevic/PUJ_FinanceTracker_AjlaHalilcevic