

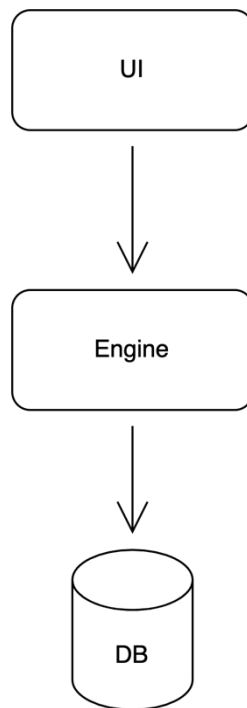
# Projektni zadatak DRES

## Sistem razmjene novčanih sredstava

Implementirati projekat koji simulira transakcije izmedju korisničkih računa.

Implementacija treba da sadrzi 3 komponente:

1. Korisnicki interfejs (UI)
2. Servis za obradu zahteva I podataka (Engine)
3. Bazu podataka (DB)



Skica sistema

### Korisnicki interfejs (UI)

Korisnicki interfejs je Flask web aplikacija koja treba da opsluzi korisnika u interakciji sa platnim prometom.

Inicijalno aplikacija treba da sadrži administratorski nalog. Administrator ima svoje:

1. Ime
2. Prezime
3. Adresa
4. Grad
5. Drzava
6. Broj telefona
7. Email
8. Lozinka

Prva od opcija koja je omogućena administratoru nakon prijave je registracija novog korisnika. Nakon što administrator registruje korisnika, korisnik treba da dobije mail sa kredencijalima za prijavu.

Druga opcija koju administrator može da koristi je praćenje uživo izvršenih transakcija između korisnika (bez osvježavanja stranice).

Novoregistrovani korisnik treba da verifikuje svoj nalog tako što će da unese bar jednu karticu na svoj nalog što zapravo predstavlja verifikaciju.

Kako bi verifikacija bila uspješna administrator treba da prihvati povezivanje kartice za nalog i tada je korisnički nalog verifikovan i spreman za transakcije.

Korisnički nalog može da ima više kartica. Svaka kartica ima svoje stanje računa.

Obični korisnik treba da podrži sledeće mogućnosti:

1. Prijava na nalog (email i lozinka)
2. Izmena korisnickog profila (svih polja)
3. Pregled računa u svim svojim valutama.
4. Mogućnost uplate na račun u bilo kojoj valuti (u okviru jedne kartice).
5. Mogućnost konverzije iz stanja računa jedne valute u drugu valutu (u okviru jedne kartice)
6. Mogućnost vršenja transakcije sa drugim računima (korišćenjem jedne od kartica koje je verifikovao).

Korisnike se loguje putem:

- Email
- Lozinka

Proces transakcije:

Korisnik treba da pokrene transakciju tako što će da odabere jednu od svojih kartica i jednu od valuta koju posjeduje na toj kartici, odabere iznos i broj računa (kartice) korisnika kome želi da uplati novac. Uz to potrebno je unijeti i podatke o email adresi, imenu i prezimenu primaoca. Nakon toga transakcija će se kreirati i biti u stanju čekanja tek kada se provjeri da li postoje novčana sredstva za njeno izvršenje. Svako jedan minut potrebno je izvršiti sve transakcije na čekanju. U tom momentu strižu poruke na email pošiljaoca i primaoca o izvršenoj transakciji, a sredstva se skidaju sa računa odnosno uplaćuju na račun.

Ukoliko primaoc nema na svom računu (kartici) stanje u datoj valuti, kreirati ga. Prilikom konverzije valuta potrebno je koristiti trenutno stanje kursne liste.

### Servis za obradu zahteva I podataka (Engine)

Engine je servis implementiran kao flask API aplikacija. Engine ima svoje endpointe koje prikazuje eksternom svetu (UI aplikaciji) za koriscenje. UI deo poziva endpointe Engine-a radi obrade raznih zahteva I podataka. Pri tome samo Engine komunicira sa bazom, a UI sa Engine-om.

### Baza podataka (DB)

Baza podataka je u komunikaciji sa Engine-om za svrhu skladistenja podataka o aplikaciji. U bazi se skladiste svi esencijalno bitni podaci za rad aplikacije.

Model baze kao I tip baze (NoSQL, SQL) je proizvoljan (ne može biti SQLite).

### Nacina ocenjivanja

1. Aplikacija je funkcionalna I postoji Flask aplikacija – 51 poen
    - a. Aplikacija se sastoji od 1 aplikacije bez baze koja je potpuno funkcionalna
  2. Implementiran Engine kao posebna Flask aplikacija gde UI komunicira sa Engine-om putem API-a – 10 poena
  3. Implementirana je baza sa kojom komunicira Engine – 9 poena
  4. Koriscenje procesa prilikom implementacije – 15 poena
  5. Dockerizacija aplikacije I pokretanje na vise racunara (distribuiran sistem) – 15 poena
- Deploy aplikacije na Heroku – gratis 5 poena (moguće samo su implementirane sve funkcionalnosti korisnika i administratora)