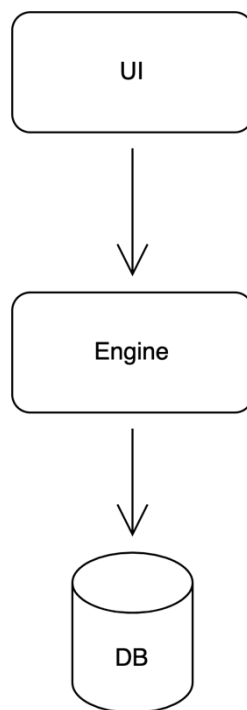


## Projektni zadatak DRES

Implementirati projekat koji simulira crypto menjacnicu I on-line crypto racun za licne uplate.

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.

Akcije koje treba podrzati na korisnickom interfejsu su:

1. Registracija novog korisnika
2. Logovanje postojećeg korisnika
3. Izmena korisnickog profila
4. Pregled stanja
5. Ubacivanje sredstava putem platne kartice na on-line crypto racun
6. Pregled istorije transakcija sa mogucnoscu sortiranja I filtriranja

7. Iniciranje nove transakcije drugom korisniku
  - a. Koji ima otvoren on-line crypto racun
8. Izbor crypto valuta – sa osvezavanjem kursne liste sa interneta prema glavnoj valuti \$
9. Zamena valute

Korisnik se registruje unoseci:

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

Korisnike se loguje putem:

- Email
- Lozinka

Novi korisnik ima stanje 0. On tada treba da zatrazi verifikaciju naloga. Za verifikaciju je potrebno da unese svoju platnu karticu I bice mu skinuto 1\$. Nakon toga korisnik moze da uplati sredstva sa kartice na svoj on-line racun.

Test platna kartica:

Broj: 4242 4242 4242 4242

Ime: <Ime Korisnika>

Datum isteka kartice: 02/23

Sigurnosni kod: 123

Korisnik inicira transakciju drugom korisniku unoseci podatke njegovoj email adresi ukoliko drugi korisnik ima registrovan nalog.

Kad se inicira transakcija, ona treba da se obradi na strani Engine-a. Transakcija ima stanja:

1. U obradi
2. Obradjeno
3. Odbijeno

Potrebno vreme da se transakcija majnuje (validira) je 5min. Za to vreme sistem mora da bude sposoban da odgovori na ostale zahteve. Svaka transakcija mora da ima svoj Hash (ID) koji je znak da je zapisana na blockchain-u. Za hash-ovanje koristiti keccak256 funkciju.

U hash stringu treba da bude:

- Posiljalac transakcije
- Primalac transakcije
- Iznos
- Random.int

Konverzija crypto valute se vrši po principu da korisnik uplaćuje sa kartice sa on-line racun u Dolarima. Kursna lista se dovlaci sa eksternog API-a kursne liste. Nakon dobijanja liste, korisnik bira valutu i iznos. Nakon uspesne konverzije korisnik ima novo stanje u novoj crypto valuti. Korisnik moze da ima neogranicen broj valuta i stanja racuna u crypto valutama.

### 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.

## 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 niti prilikom implementacije – 10 poena
  5. Koriscenje procesa prilikom implementacije – 10 poena
  6. Dockerizacija aplikacije i pokretanje na vise racunara (distribuiran sistem) – 10 poena
- Deploy aplikacije na Heroku – gratis 5 poena (moguće samo ako je svih 6 tacaka implementirano)