

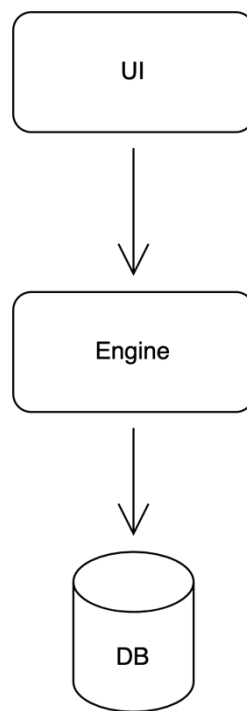
# Projektni zadatak DRES

## Sistem onlajn kupovine

Implementirati projekat koji simulira onlajn kupovinu proizvoda.

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 prilikom kupovine onlajn proizvoda.

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 kreiranje novog proizvoda i povecanje kolicine postojećih proizvoda. Nakon što administrator doda proizvod, on je vidljiv na listi proizvoda. Proizvod treba da sadrži ime, cijenu u jednoj valuti koju administrator odabere i količinu na stanju.

Druga opcija koju administrator može da koristi je praćenje uživo izvršenih kupovina proizvoda (ko je kupio koji proizvod).

Obični korisnik se registuje popunjavanjem istih polja kao i administrator. Nakon uspješne registracije mail porukom obavijestiti administratora o registraciji novog korisnika.

Novoregistrovani korisnik treba da verifikuje svoj nalog tako što će da 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 kupovinu.

Korisnički nalog se vezuje za jednu karticu.

Obični korisnik treba da podrži sledeće mogućnosti:

1. Registracija naloga
2. Prijava na nalog (email i lozinka)
3. Izmena korisnickog profila (svih polja)
4. Pregled računa u svim svojim valutama.
5. Mogućnost uplate na račun u bilo kojoj valuti.
6. Mogućnost konverzije iz stanja računa jedne valute u drugu valutu
7. Mogućnost kupovine proizvoda u bilo kojoj valuti.
8. Pregled istorije kupovine proizvoda.

Korisnike se loguje putem:

- Email
- Lozinka

Proces kupovine:

Korisnik pokreće kupovinu tako što odabira jedan proizvod i nakon toga bira valutu u kojoj želi da plati (samo ako na kartici takvo stanje posjeduje). Nakon toga potvrđuje kupovinu i kupovina 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 kupovine na čekanju. U tom momentu strižu poruke na email administratora (ukoliko je 5 kupovina u obradi sve ih upisati u jedan mail) o izvršenoj kupovini, a sredstva se skidaju sa računa odnosno uplaćuju

na račun. Administrator takođe treba da ima svoje stanje računa na koji se uplaćuju sredstva.

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)