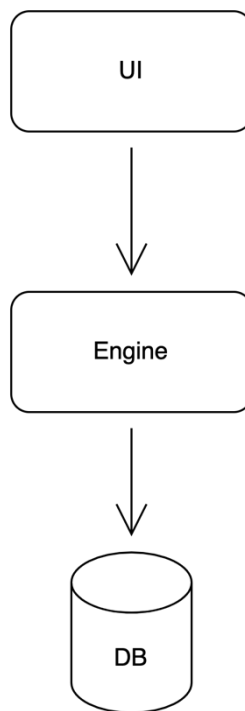


Projektni zadatak DRES Forum

Implementirati sistem za pravljenje različitih tema i komentara.

Implementacija treba da sadrži 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 opsluži korisnika u interakciji sa platnim prometom.

Akcije koje treba podržati na korisničkom interfejsu su:

1. Registracija novog korisnika
2. Logovanje postojećeg korisnika
3. Izmena korisničkog profila
4. Pravljenje teme na koju se drugi korisnici mogu pretplatiti
5. Pravljenje novog komentara od strane bilo kojeg korisnika

6. Slanje obaveštenja svim pretplaćenim korisnicima
7. Upvote \ downvote komentara
8. Omogućiti upvote \ downvote iz samog mejla
9. Omogućiti brisanje neke teme od strane vlasnika
10. Omogućiti pretragu tema i sortiranje po tome koje imaju najviše komentara ili najviše upvotes ili downvotes. Samo ulogovani korisnici mogu da rade upvote - downvote ili da komentarišu.
11. Omogućiti vlasniku teme da u bilo kom momentu može da napravi temu zatvorenom za komentarisanje

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 nakon kreiranja naloga dolazi na stranicu sa temama, gde postoji inicijalno prazan spisak njegovih tema sa mogućnošću pravljenja nove teme, a ispod se nalazi spisak svih tema trenutno koje postoje. U okviru ovog dela može da navigira na bilo koju temu i da se na neku pretplati ili da postavi komentar. Nakon što bilo koji korisnik ostavi komentar na neku temu neophodno je svim pretplaćenim korisnicima poslati mejl sa komentarom i mogućnostima da urade upvote ili downvote. Istu opciju omogućiti i na UI delu. Takođe je neophodno omogućiti korisniku da sopstvene teme zatvori ili otvori.

Slanje mejlova pretplaćenim korisnicima ne sme da poremeti rad sistema. Po mogućnošću koristiti neki servis za slanje mejlova ili slati ručno. Primer sistema za slanje mejlova jesu Postmark ili SendGrid.

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 (implementacija UI-a kroz neku od tehnologija React, Vue, Angular...) – 10 poena
3. Implementirana je baza sa kojom komunicira Engine – 14 poena
4. Koriscenje niti ili procesa prilikom implementacije – 15 poena
5. Dockerizacija aplikacije I pokretanje na vise racunara (distribuiran sistem) – 10 poena