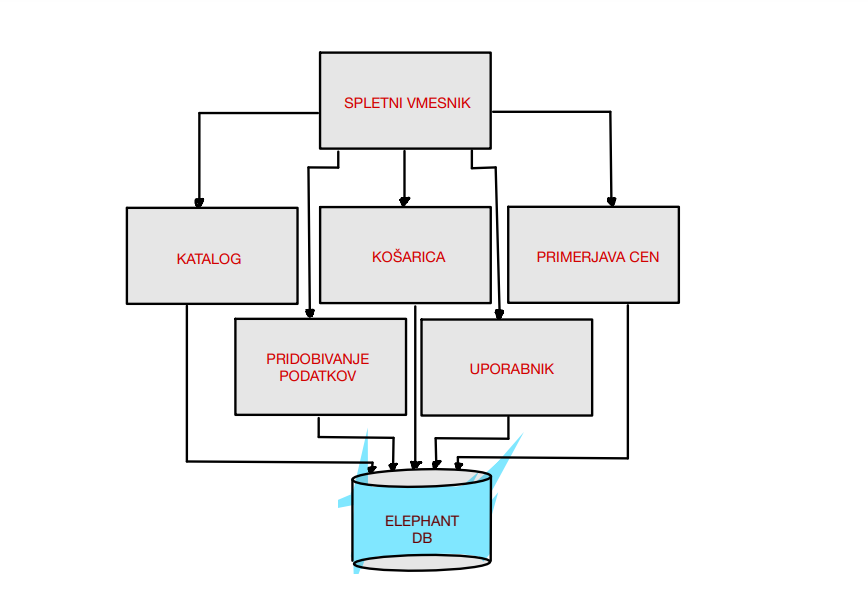
PRIMERJALNIK CEN IZDELKOV

Člani skupine: Damijan Randl, Marko Marinković in Tit Arnšek

Projekt, ki ga predstavljamo, ponuja uporabniku primerjavo cen izdelkov iz različnih trgovin po Sloveniji.

Ogrodje aplikacije je napisano v jeziku Python z uporabo knjižnice Flusk.

Uporabljamo en repozitorij(<https://github.com/titoo1234/RSO-Primerjalnik-cen-izdelkov>) ; vsako mikrostoritev na svojem branchu. Avtomatizirano imamo avtomatsko generiranje in pošiljanje Docker slik na Docker hub. Vse shranjujemo v naslednji repozitorij: <https://hub.docker.com/repository/docker/tito1234/primerjalnik_cen>

Mikrostoritve so naslednje:

* Pridobivanje podatkov (skrbi za iskanje podatkov iz spleta/osveževanje cen. V naprej pripravimo seznam izdelkov, ki jih želimo imeti. Mikrostoritev poskrbi, da vse izdelke poišče v treh različnih trgovinah in poišče cene le teh. Podatke zapiše tudi v bazo)
* Katalog(Namenjen 'naslovni' strani. Skrbi prikaz vseh izdelkov iz baze)
* Primerjava izdelkov(Namenjen primerjavi nekega izdelka. Za nek izdelek prikaže/primerja razliko v ceni v različnih trgovinah)
* Košarica(Prikaz košarice za nekega uporabnika; vstavljanje podatkov v košarico; brisanje podatkov iz košarice)
* Uporabnik(Prijava in registracija uporabnika)
* Spletni vmesnik(Povezuje zgornje mikrostoritve; spletni prikaz, piškotki…)

Spletna stran uporabniku omogoča:

* Prikaz izdelkov, ki jih lahko primerja
* Primerjava samih izdelkov
* Dodajanje izdelkov v košarico (samo če je uporabnik prijavlen); brisanje izdelkov iz košarice
* Prijavo in registracijo uporabnika
* Prikaz izdelkov v košarici(samo če je uporabnik prijavlen)

Na AKS smo nameščali mikrostoritve s pomočjo yaml datotek. Te imamo shranjene na repozitoriju. Za vsako mikrostoritev je narejen Deployment in Service(ClusterIp). Nmaeščen je tudi LoadBalancer 'ingress-nginx-contoller', ki je povezan z Ingressom 'Ingress'. Do storitve lahko dostopamo preko naslova <http://20.74.91.160/SV/> (to je začetna stran)

Konfiguracijo smo uporabili pri git\_action-ih, kjer smo uporabili secrete iz samega git-a. V samih mikrostoritvah, smo konfigurirali dostop do baze in dnevnikov. Ker je povezava konstantna, nismo delali konf. strežnika.

Pri vsaki MS smo implementirali kontrolo zdravja (razen pri SV); metrike smo implementirali pri MS uporabnik in katalog.

Uporabili smo 3 zunanje API-je, ki omogočajo:

* Pridobivanje lokacije
* Pridobivanje podatkov o vremenu
* Prevod

Za dokumentacijo OpenApi-ja smo uporabili avtomatsko generiranje, ki ga omogoča knjižnica Flusk\_restx.

Loge zbiramo na platformi logit.io.