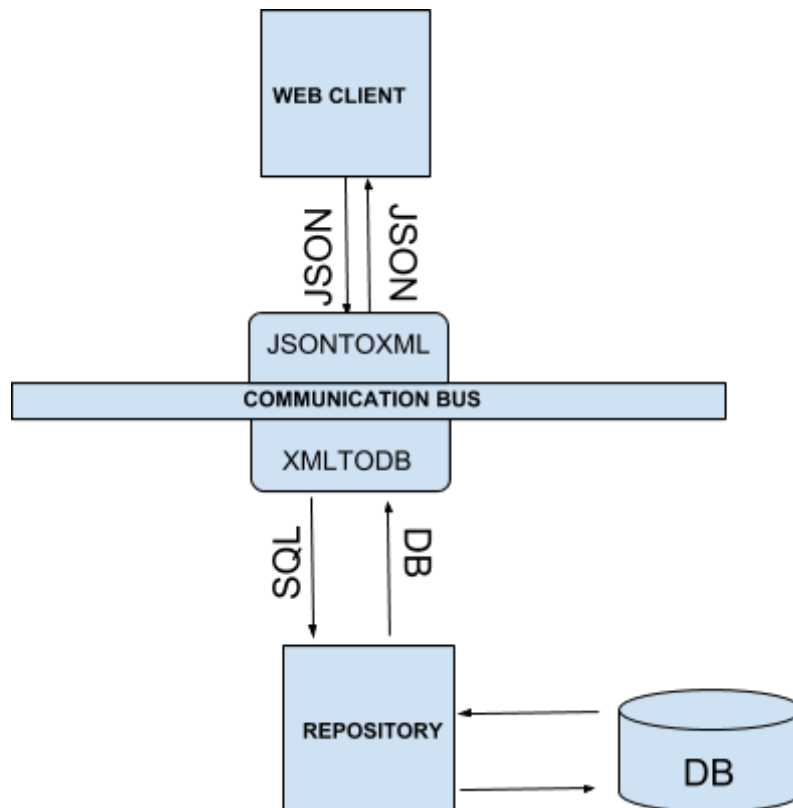


Projektni Zadatak 1

Potrebno je napraviti dizajn sistema, arhitekturu sistema, implementira i istestirati rešenje koji simulira rad i komunikaciju Communication Bus modula.



Communication bus

Communication bus je komponenta koja predstavlja posrednika u komunikaciji između komponenti sistema. Za interne poslove, da bi se obezbedio dovoljan nivo apstrakcije, communication bus koristi XML kao format podataka. On je direktno povezan sa dve komponente: JSONTOXML i XMLTODB. Zadatak ovih komponenti je da podatke u formatu koje razume odgovarajuća komponenta pretvori u format koji razume communication bus i obrnuto.

Web client

Web client je komponenta sistema koja je izložena drugim sistemima na korišćenje putem interneta. Drugi sistemi se obraćaju ovoj komponenti, šalju ju odgovarajuće zahteve, ona te zahteve obrađuje/prosleđuje dalje i vraća odgovarajući odgovor svojim klijentima.

Repository

Repository je komponenta koja prima odgovorajući upit (u vidu SQL-a u slučaju da se radi sa bazom podataka) i izvršava ga nad željenom bazom podataka.

Opis rada sistema:

Web client komponenta dobija od strane svojih klijenata neki od zahteva:

GET, POST, PATCH, DELETE praćen odgovorajućom imenicom.

Na primer: GET /resurs/1. Ona taj zahtev dobija u odgovarajućem JSON formatu:

```
{
  "verb": "GET",
  "noun": "/resurs/1",
  "query": "name='pera'; type=1",
  "fields": "id; name; surname"
}
```

Sama komponenta ima zadatak da proveriti da li je zahtev dobro formatiran (da li prosleđeni glagol postoji, da li su vrednosti polja query i fields formatirana na odgovarajući način). Polja query i fields nisu obavezna, ona dodatno filtriraju resurse.

Kako Communication Bus ne razume JSON kao format podataka, već isključivo XML, neophodno je zahtev pretvori u XML i to je svrha JSOXML adapter komponente. XML od prethodnog JSON zahteva bi bio:

```
<request>
  <verb>GET</verb>
  <noun>/resurs/1</noun>
  <query>name='pera';type=1</query>
  <fields>id; name; surname</fields>
</request>
```

Communication bus dobijeni zahtev samo prosleđuje svojoj adapter komponenti XMLTODB koja treba ovaj XML da pretvori u odgovarajući SQL upit (izgled upita ce da zavisi od baze koju odaberete):

```
SELECT
  id,
  name,
  surname
FROM resurs
WHERE id = 1
AND name = 'pera'
AND surnatypeme = 1
```

Repository komponenta prima zahtev u ovom obliku i izvršava ga nad bazom podataka, dobija odgovarajući odgovor i dobijeni odgovor prosleđuje XMLTODB komponenti da ga pretvori u odgovarajući XML, svaki odgovor mora da poseduje polja: STATUS, STATUS_CODE i PAYLOAD. Dozvoljeni statusi su: REJECTED, BAD_FORMAT, SUCCESS, a status kodovi: 3000, 5000 i 2000 respektivno. Payload bi trebalo da sadrži dobijeni resurs odnosno u slučaju primeru xml polja id, name i surname u slučaju SUCCESS koda odnosno error-message polje sa porukom o grešci. Tako dobijeni XML Communication bus prosleđuje svojoj JSOXML adapter komponenti koja ga transformiše u JSON. Zatim se odgovor prosleđuje web client komponenti koja ga vraća svojim klijentima. U slučaju da zahtev nije bio dobro formatiran web client odmah vraća BAD_FORMAT 5000 sa odgovarajućom porukom

o grešci, ne prosleđuje zahtev dalje.

U bazi podataka se nalaze resursi, svaki resurs ima id, naziv, ime, opis (opis je u json formatu ukoliko izabrana baza podataka podržava json kao tip kolone) i tip. Svaki tip poseduje naziv i id. Postoji i lista veza između resursa koja je opisana svojim id-om, id prvog, id drugog resursa u vezi, tip veze. Svaki tip veze ima svoj naziv i svoj id.

Zahetv pored polja navedenih u primeru može da poseduje i sledeća polja: connectedTo sa spiskom id-ova povezanih resursa(id=1; id=3), pri čemu je ovo dodatni filter, na primer ako postoji više resursa sa imenom Petar vrati mi onog koji je u vezi sa resursima Klavir i Odbojka i connectedType gde se navode id-ovi odgovarajućih tipova, pri čemu je i ovo dodatni filter i koristan je kad na primer imam više resursa sa imenom Petar a potrebni su mi samo oni koji postoje u tipu veze roditelj. Kada klijent web client komponente kreira zahtev obavezan je da unese samo verb i noun, sve ostalo predstavlja samo dodatni filter koji može, a ne mora da unese.

Kriterijum ocenjivanja

1. Dizajn I arhitektura rešenja
2. Korišćenje Scrum metodologije razvoja – definisanje User Story-a i taskova, planiranje i estimacija
3. Implementacija rešenja
4. CI ciklus
 - a. Build
 - b. UnitTestovi
 - c. Pokrivenost koda testovima

