TeamUp arhitektura i specifikacije

1. Arhitektura aplikacije

TeamUp aplikacija zamišljena je kao mobilna aplikacija koja koristi web servis kako bi dohvatila, izmjenila ili dodala podatke u bazu podataka. Prema tim zahtjevima gradi se arhitektura koja sadrži jedan web poslužitelj koji ima bazu podataka sa podacima važnim za funkcioniranje aplikacije te web servis koji vrši obradu podataka unutar te baze podataka. Na centralni web poslužitelj spajaju se klijenti putem mobilne aplikacije te određenim radnjama unutar mobilne aplikacije vrše pozive prema web servisu, a web servis vrši promjenu podataka unutar baze.

// TODO: slikica

1. Baza podataka

// TODO: ERA i objašnjenje modela

1. Web servis – specifikacija

Web servis leži na poslužitelju te se pokreće kao samostalna aplikacija. Prilikom pokretanja web servisa aplikacija podiže svoj interni (ugrađeni) web poslužitelj putem kojeg prima pozive od klijenta te vraća odgovore. Ugrađeni web poslužitelj pokreće se tako da radi paralelno sa web poslužiteljem na kojem je smješten osluškivajući port 8080.

Kako bismo mogli shvatiti logiku web servisa potrebno je prethodno pojasniti klase koje su napravljene pomoću kojih će klijent komunicirati sa web servisom. U nastavku se nalazi dijagram klasa koji je implementiran unutar klijentske aplikacije.

// TODO: dijagram klasa

Cijela logika web servisa napisana je u RESTful paradigmi gdje se putem zadanog URL-a i HTTP metode (POST, GET, PUT, DELETE) vrše pozivi na web servis, što okida akcije za obradu podataka ovisno o URL-u, metodi te podacima koji su poslani. Svi podaci šalju se u tijelu zahtjeva (request body) u JSON formatu (application/json) što servis čita i koristi pri obradi. Primjerice, pri registraciji novih korisnika koristiti ćemo URL oblika <https://IP_ADRESA:8080/person/signup> te HTTP metodu POST gdje će u tijelu zahtjeva biti Person objekt popunjen podacima i iskazan u JSON formatu. Detaljnija specifikacija web servisa opisana je u tablici u nastavku.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NEŠTO | Operacija | URL | HTTP metoda | Tijelo zahtjeva |
| 1 | Ispis svih korisnika | /person | GET | - |
| 2 | Registracija korisnika | /person/signup | POST | Person objekt |
| 3 | Login korisnika | /person/login | POST | Credentials objekt |
| 4 | Ispis određenog korisnika prema id-u | /person/{id} | GET | - |
| 5 | Izmjena korisnika | /person/{id} | PUT | Person objekt |
| 6 | Brisanje korisnika | /person/{id} | DELETE | - |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

// TODO: tablica i specifikacija do kraja (token etc. ?)

1. Mobilna aplikacija – specifikacija poziva

Klijent koristi mobilnu aplikaciju kako bi kontaktirao web servis i napravio izmjene unutar aplikacije. Kako bi se uspostavio kontakt sa serverom, korištena je standardna HttpURLConnection klasa koja // TODO