**5. TEL. SĄSKAITŲ UŽDAVINYS**

Sukurti web-aplikaciją su testais pagal žemiau nurodytą aprašymą. Kūrimui naudoti Spring Boot frameworką bei Eclipse IDE for Java EE Developers ar kitą programavimo aplinką.

Atsiskaitant reiks žodžiu paaiškinti suprogramuotą aplikacijos kodą ir testus, naršyklėje pademonstruoti aplikacijos veikimą, bei paleisti testus. Postman pagalba aplikacijai siųsti GET / POST užklausas ‚request‘, parodyti ir pakomentuoti atsakymo ‚response‘ rezultatus.

1. Duomenis saugoti duomenų bazės lentelėse. Galima naudoti in-memory duomenų bazę pvz. H2.

Sukurti dvi lenteles su duomenimis:

a) TelNr su stulpeliais: telNr, userIds (vienas žmogus gali turėti kelis tel. numerius).

b) Saskaita, kurioje saugomi įrašai apie mokėtinas sumas už pokalbius kiekvieną mėnesį.

Lentelėse laukeliai id yra unikalūs, jų reikšmės generuojamos automatiškai.

Lenteles užpildyti įvairiais duomenimis, ne tik tais, kurie pateikti šiame pvz.:

TelNr

|  |  |  |
| --- | --- | --- |
| **id** | **telNr** | **userId** |
| 1 | +370111 | 1 |
| 2 | +370222 | 2 |
| 3 | +370110 | 1 |
| 4 | +370333 | 3 |
| 5 | +370330 | 3 |
| 6 | +370300 | 3 |
|  | … | … |

telNr yra unikalūs.

Saskaita

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **telNrId** | **mėnuo** | **suma** |
| 1 | 1 | 1 | 10 |
| 2 | 2 | 1 | 5 |
| 3 | 3 | 1 | 20 |
| 4 | 4 | 1 | 10 |
| 5 | 5 | 1 | 15 |
| 6 | 1 | 2 | 5 |
| 7 | 2 | 2 | 20 |
|  | … | … | … |

1. Sukurti kiekvieną lentelę atitinkančią @Entity klasę. Šias klases saugoti atskirame pakete pavadinimu pvz. model.
2. Sukurti kiekvieną lentelę atitinkančią CrudRepository tipo klasę / interface‘ą, kuri atliktų CRUD (create, read, update, delete) veiksmus su duomenų bazės įrašais, pvz.:   
   save(S entity); // save new or update existing if id is given

findAll(); findById(ID id); findByTelNr(String telNr); deleteById(ID id); delete(T entity);

Šias klases saugoti atskirame pakete pavadinimu pvz. repository.

1. Sukurti kiekvieną Entity klasę atitinkančią serviso klasę, kurioje būtų metodai, reikalingi atitinkamiems kontroleriams (žr. žemiau). Pvz. metodas, kuris paduoda visus lentelės įrašus kontroleriui, metodas kuris suranda tel. numerius pagal atitinkamus telNrId, metodas, kuris trina pirmos lentelės įrašą ir trina id atitinkančius antros lentelės įrašus. Šias klases saugoti atskirame pakete pavadinimu pvz. service.
2. Web aplikacija turi naršyklėje html pavidalu atvaizduoti pirmos lentelės duomenis, html formos pagalba leisti įrašyti / redaguoti įrašą, leisti trinti egzistuojantį įrašą. Šiuo tikslu naudoti MVC kontrolerį. Jei trinamas kuris nors įrašas pirmoje lentelėje, kartu turi būti ištrinti ir visi trinamą id atitinkantys įrašai antroje lentelėje. Kontroleris turi naudoti serviso klases.
3. Web aplikacija turi naršyklėje html pavidalu atvaizduoti antros lentelės duomenis, html formos pagalba leisti įrašyti / redaguoti įrašą, leisti trinti egzistuojantį įrašą. Šiuo tikslu naudoti MVC kontrolerį. Be to antroje lentelėje šalia telNrId laukelio rodyti telNr reikšmę, kuri pagal telNrId būtų imama iš pirmos lentelės. Kontroleris turi naudoti serviso klases.
4. Web aplikacija turi naršyklėje json pavidalu atvaizduoti pirmos ar antros lentelės duomenis, POST request pagalba leisti įrašyti naują įrašą, leisti trinti egzistuojantį įrašą. Šiuo tikslu naudoti REST kontrolerius. Jei trinamas kuris nors įrašas pirmoje lentelėje, kartu turi būti ištrinti ir visi trinamą id atitinkantys įrašai antroje lentelėje. POST užklausas galima siųsti per Postman įrankį. Kontroleris turi naudoti serviso klases.
5. Kontrolerių klases saugoti atskirame pakete pavadinimu pvz. controller.
6. Model klasėms ir jų metodams parašyti unit testus.
7. Service klasėms ir jų metodams parašyti unit testus, panaudojant Mock objektus.
8. Repository klasėms ir jų metodams parašyti unit testus naudojant Spring Boot @DataJpaTest
9. Controller klasėms parašyti testus, naudojant Spring Boot @WebMvcTest ir Mock objektus.
10. Controller klasėms parašyti integration testus, naudojant Spring Boot @SpringBootTest