**DISTRIBUTED SYSTEMS**

**Assignment 1**

Name: Chis David Adrian

1. Conceptual architecture of the distributed system

2. DB design

3. UML Deployment diagram

4. ReadMe files containing build and execution considerations

1. Conceptual architecture of the distributed system

Diagram

Description automatically generated

DB Design

A picture containing text, map, indoor, several

Description automatically generated

3. UML Deployment diagram

Graphical user interface

Description automatically generated

4. ReadMe files containing build and execution considerations

Aplicatia are la baza 2 parti , partea de BackEnd si partea de FrontEnd

Aplicatia BackEnd (BE) este o aplicatie de sine statatoare scrisa in Java se bazeaza pe Spring Framework si ruleaza pe portul 8080 default, Este o aplicatie care serveste pe tip de server pentru aplicatia de frontEnd (FE) si in acelasi timp este si o aplocatie Client pentru DB Server.

Aplicatia BE face legatura dintre nivelul BD si Client ul astfel este structurata pe Layere si anume DB Model Repository Service Controller .

* 1. DB este primul nivel reprezentat de pachetul Model din aplicatie in care sunt mapate tabelele si legaturile dintre ele.
  2. Repository: pachetul ce contine intergete care fac tranzitia de la obiectul efectiv la tabel prin intermediul ORM (Object-Relational Mapping)
  3. Service: pachetul care contine logica aplicatiei folosind repository-urile
  4. Controller: pachetul unde se face legatura dintre Front-End si Back-End, unde se afla metodele ce gestioneaza request-urile HTTP si folosesc service-urile pentru a apela la baza de date.

Componenta Front-End va rula pe portul 3000 si aceasta va reprezenta interactiunea dintre BackEnd si utilizatori. Pentru obtinerea datelor din baza de date, in Front-End vom face request-uri HTTP catre Back-End si vom obtine datele pe care le vom prelucra mai departe. Aceasta componenta a fost prelucrata utilizand libraria ReactJS iar limbajul a fost Typescript. Aici am impartit fiecare logica in cate o componenta care este continuta in cate un pachet separat. Astfel, avem urmatoarele pachete:

Components : in care sunt incluse componentele reutilizabile in aplicatie ex tabele grafice etc

Pages : in care sunt structurate fiecare Pagina din aplicatie

Services : IN care sunt introduse serviciile FE coresponedente fiecarui Controller din BE astfel se face o organizare a codului mult mai buna si duce la o intelegere cat mai buna. Apelurile catre BackEnd sunt facute utilizand Jquery (Ajax) pentru a face codul cat mai simplu de citit pt developer.

Utils in care am stocat constante sau metode des utilizate in aplicatie

Enums in care s-au stocat enum urile folosite in aplicatie

Dto incare s-au stocat obiectele secundare folosite pentru transmiterea datelor din FE in BE

Model un pachet inc are sunt introduse cate o interfata pt fiecare model din BE astfel facanduse un match 1-1 cu BE ul