KoliApp dokumentáció

# Projekt ötlet

## KoliApp

Egy webes alkalmazás, amely felületet biztosít kollégistáknak közérdekű információk kiírására és szobák foglalására, mint például mosókonyha, tanulószobák...

## Fejlesztők

* Nervetti Viktor - frontent
* Reitter János - backend

## Feladat funkcionális követelményei

* Főképernyő: posztok megjelenítése, posztokhoz kommentek hozzáfűzése
* Poszt rögzítés: poszt készítése a főképernyőre
* Szobafoglaló képernyő: szobafoglalás igény rögzítése (amiből poszt lesz szintén)
* Regisztrációs felület: regisztrálás
* Bejelentkező felület: bejelentkezés

## Nem funkcionális követelmények

* Front end: Angular (angular material és bootstrap)
* Back end: Spring boot
* Adatbázis: MySQL

## Szakterületi fogalom

* REST API
* Authentication/Authorisation (spring security)
* MVC
* map struct
* Testing

## Szerepkörök

* Admin: bejegyzést tud törölni
* Nem bejelentkezett felhasználó: meg tudja nézni a felvett bejegyzéseket és foglalásokat
* Bejelentkezett felhasználó: Tud létrehozni bejegyzéseket, kommentelni kész bejegyzésekhez, és szobát foglalni

# Backend megvalósítása

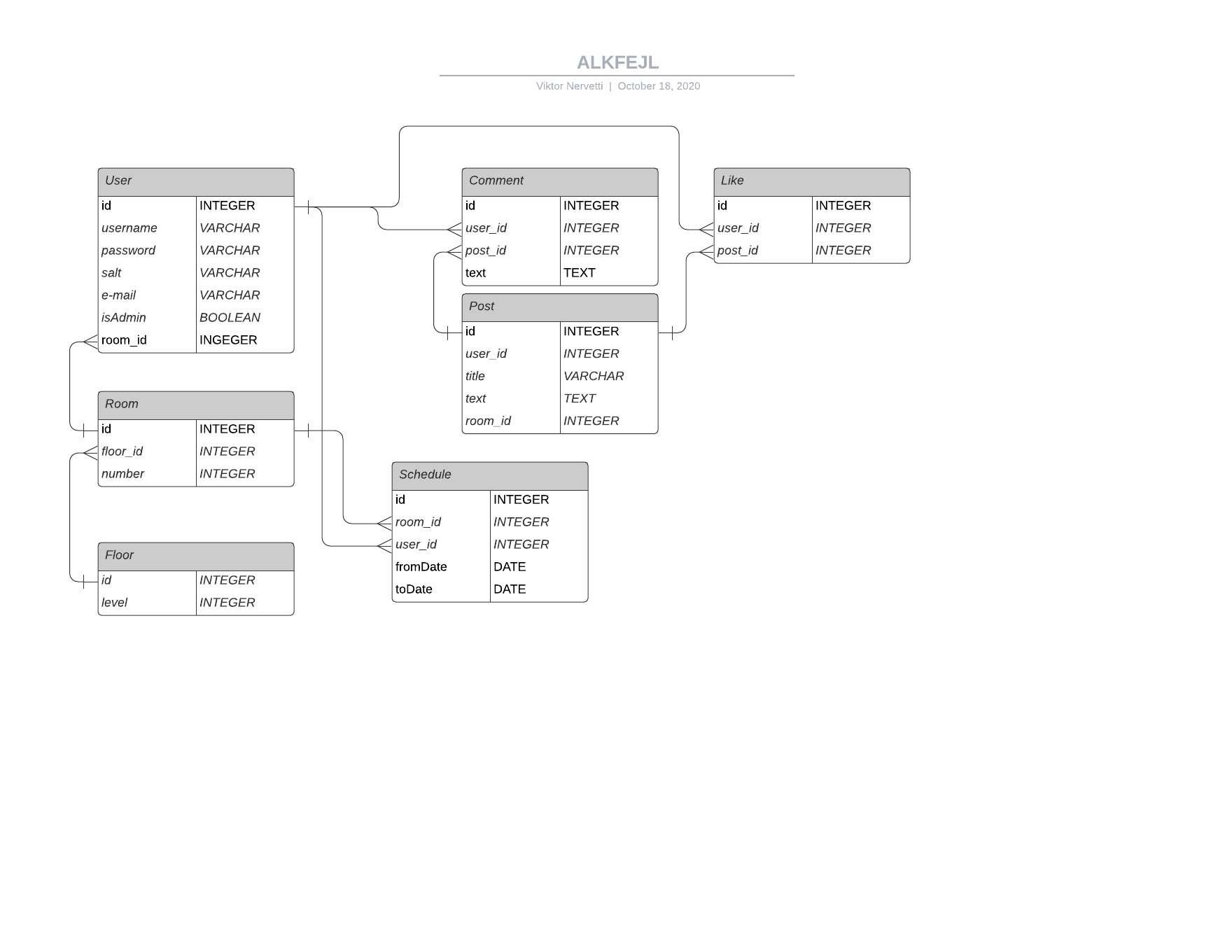
## Fejlesztői környezet bemutatása

* Eclipse IDE for Enterprise Java Developers 2020-09 (4.17.0) fejlesztői környezet

## Felhasznált technológiák

* Maven Project
* Spring boot
* Lombok
* JWT
* Refreshtoken
* Rest API
* Spring Security
* map struct

## Adatbázis terv



## Alkalmazás könyvtárstruktúra

* KoliApp
  + controller: végpontok
  + entity: adatbázis entitások
  + repository: entitásokhoz tartozó repositoryk
  + security: autentikáció és autorizáció
  + exception: KoliAppException implementálása
  + mapper: dto és entitások közötti mapper interfészek
  + service: biznisz logika
  + dto: data transfer object

## Végpont-tervek

* /auth
  + /signup – regisztráció (POS)
  + /login – bejelentkezés (POS)
  + /refresh/token – új refreshtoken kiosztás (POS)
  + /logout – kijelentkezés (POS)
* /post
  + / - összes poszt lekérése (GET)
  + / - poszt készítése (POST)
  + /schedule – szobafoglalásból készűlő pszt (POST)
  + /like – poszt lájkolása (POST)
  + /archive – adott poszt archiválása/törlése (POSt)
* /comment
  + / - komment beszúrása (POST)
* /room
  + / - összes szoba lekérése (GET)
* /schedule
  + / - szobafoglalás rögzítése (POST)
  + /check – szobafoglalás előtti időintervallum vizsgálat

# Egy végpont részletezése

Poszt készítésének részletezése:

1. Kliensoldalon az összerakott PostCreateDto entitást elküldi a backend részére.
2. Backend a service layeren a PostCreateDto-ból a PostMapper osztály segítségével előállítja a Post entitást.
3. A poszt entitás a Poszt respository segítségével kap egy egyedi azonosítót és elmenti az adatbázisba.
4. A létrehozott és elmentett poszt entitást visszaküldi a backend a kliensoldalnak.

# Map struct

A kliensoldalról érkező (eltérő struktúrájú) entitások és az adatbázis entitások közötti mapper fájlok létrehozását a map struct technológia könnyíti meg. Ennek segítségével elegendő csak egy interfészt (vagy absztrakt osztályt) létrehoznunk és ha deklaráljuk a megfelelő függvényeket a map struct kigenerálja hozzá nekünk a függvény implementációját. Amennyiben az implementáció nem egyértelmű, akkor @Mapping annotációkkal segíthetjük a folyamatot.

# Backend tesztek

1. Összes posztot visszadó végpont
2. Összes szobát visszadó végpont
3. Szobafoglalás időpont ellenőrzés (mindkét esetre)
4. Poszt létrehozása majd összes poszt lekérdezése, egyik poszt archívvé alakítása majd újból összes poszt méretének lekérdezése
5. Posztok lájkolása
6. Kommentelés