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

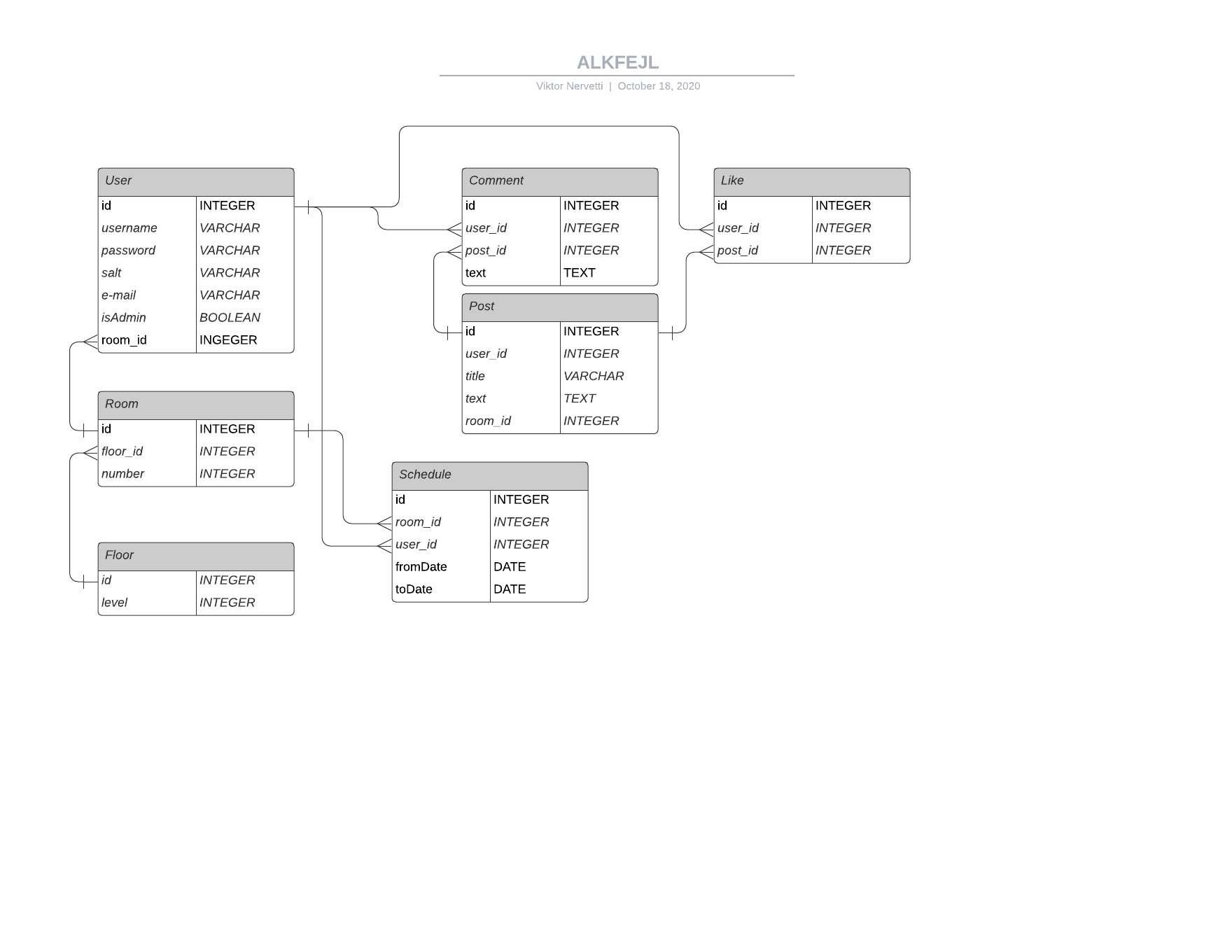
## Fejlesztői környezet bemutatása

* 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)
  + /password/change – jelszó módosítás (POST)
* /post
  + / - összes poszt lekérése (GET)
  + / - poszt készítése (POST)
  + /schedule – szobafoglalásból készűlő pszt (POST)
  + /like – poszt lájkolása (POST)
  + /archive – adott poszt archiválása/törlése (POST)
  + /liked – adott poszt lájkolta-e az aktuális felhasználó (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.

# JWT és refresh token

A felhasználó azonosítást követően jwt tokennel történik a hitelesítés. Amikor a kliensoldal egy hívást indít a backendnek, akkor a http hívásnak a bearer tokenének van megadva a jwt. Ez a rendszer van még egy plusz tokennel biztosítva, mely az időeltelést követően lejár. Amennyiben ez a token le van járva, a felhasználó nem tud semmilyen hívást elvégezni. Amennyiben a refresz/token meghívásra kerül, akkor a lejárati idő újraindul.

# 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

Frontend dokumentáció

Frontend megvalósításhoz angular material és bootstrap technológiákat használtuk fel.

## Webapp szolgáltatások:

1. Bejelentkezés – Userekenek, hogy ne csak megnézni, hanem új posztokat is rögzíteni tudjanak, bejelentkezésre lesz szükségük.
2. Kijelentkezés – Bejelentkezetett userek ki tudjanak jelentkezni.
3. Bejegyzések oldalon:
   1. Bejegyzések megtekintése – összes nem archív bejegyzés megtekinthető
   2. Bejegyzéshez tartozó kommentek megtekintése – bármely kilistázott
   3. Bejegyzéshez komment fűzés – bármely listázott bejegyzéshez bejelentkezett felhasználó kommentet fűzhet.
4. Poszt rögzítés – Poszt rögzítésének lehetősége bejelentkezett felhasználóknak.
5. Szobafoglalás rögzítés – Szobafoglalás igény rögzítés, amelyből automatikus poszt fog generálódni.

## Kapcsolat a szerverrel

A kliensoldal a backenddel Rest API hívásokkal kommunikál. A hívások során a küldendő és fogadó adatok JSON formátumban jönnek/érkeznek.

## Szobafoglalás rögzítés folyamat bemutatása

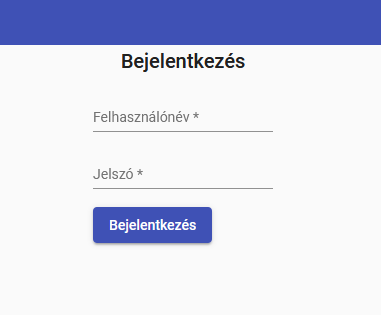
1. A bejelentkezett felhasználó kitölti a foglaláshoz szükséges adatokat a kliensoldalon található űrlapon.
2. Amennyiben az űrlap helyesen van kitöltve, az adatokat elküldi a backendnek, hogy megvizsgálja a backend, hogy abban a megadott idő intervallumban szabad az adott szoba.
3. Amennyiben szabad visszajelez a kliensoldalnak egy true-val, ha nem akkor egy false-sal.
4. Ha Igazat kap a backendből, akkor a frontend szintén rest api hívással létrehoztatja a backenddel a szobafoglalás bejegyzést.
5. Amennyiben elkészült a szobafoglalás bejegyzés, a megadott adatokból generálódni fog egy poszt is.

## Frontend teszt

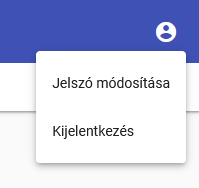
# Felhasználói dokumentáció

A KoliApp webalkalmazás célja, hogy megkönnyítse a kollégisták életét. Ahelyett, hogy különböző nem hivatalos fórumokon, oldalakon írjanak ki kéréseket és közérdemű információkat, lett létrehozva egy egységes felület, mely ellátja ezeket a feladatokat.

## Bejelentkezés



Megfelelő belépési adatok megadása után a felhasználó bejelentkezve lesz, mely azt jelenti, hogy lesz lehetősége, kommentet írni, posztot és szobafoglalást létrehozni.



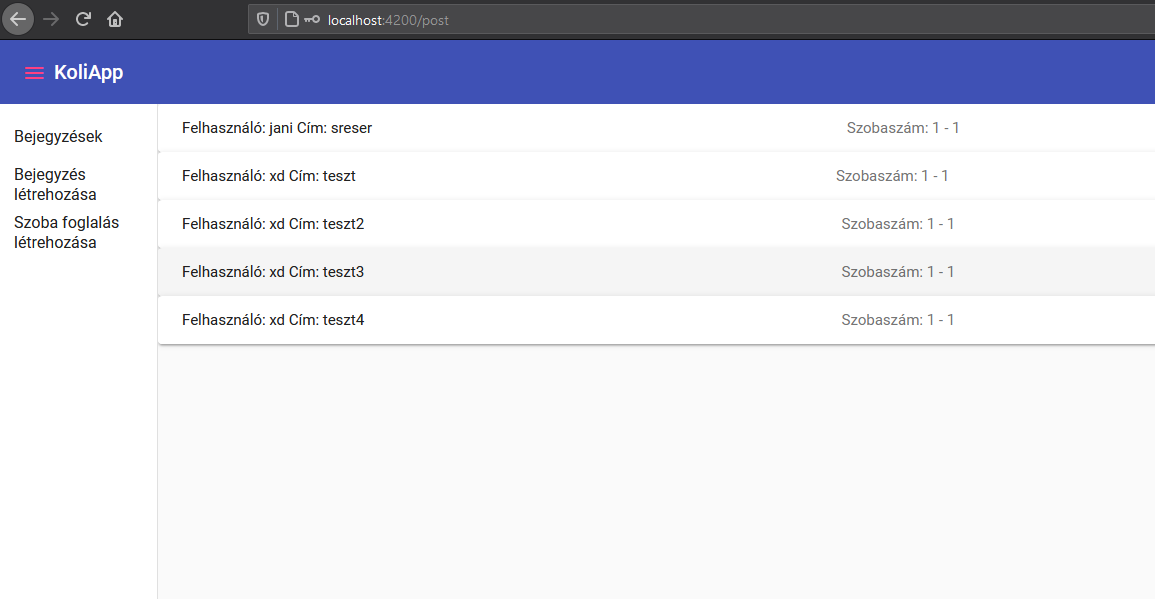
Bejelentkezést követően lehetőség van kijelentkezni vagy jelszót módosítani.

Amennyiben nem vagyunk bejelentkezve, akkor a fent látható lenyíló ablakon a bejelentkezés és regisztrálás pontok fognak megjelenni.

## Posztok

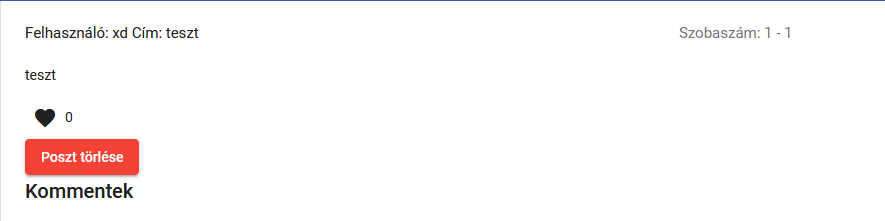
Bejelentkezést követően el leszünk navigálva a posztok főoldalra.

Itt vannak kilistázva az összes poszt.



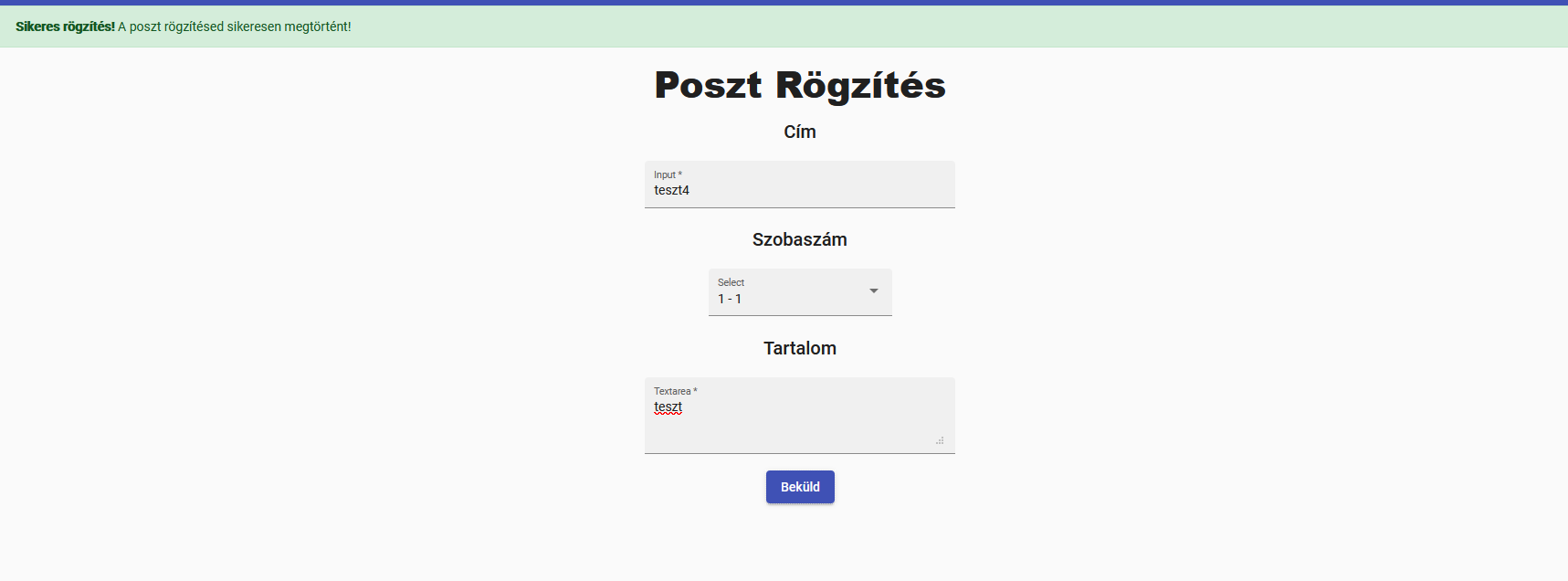
Amennyiben be vagyunk jelentkezve, akkor egy poszt lenyitást követően megnézhetjük a poszt tartalmát, illetve lehetőségünk van kommentet hagyni és lájkolni.

Amennyiben a bejelentkezett felhasználó egy admin, akkor neki lehetősége van a posztokat törölni.



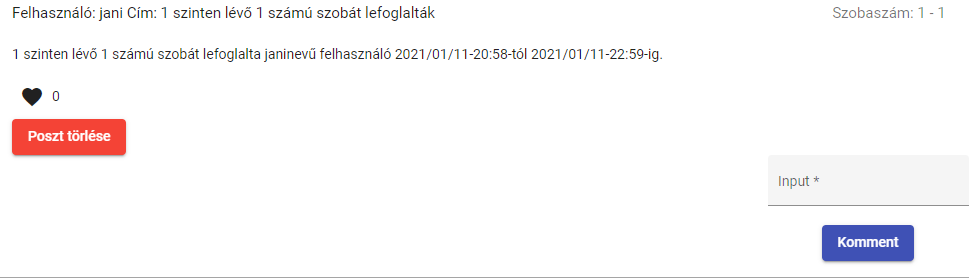
## Poszt rögzítés

A poszt rögzítés képernyőn tudunk a főoldalra posztot rögzíteni. Fontos kritérium, hogy minden mező az űrlapon legyen kitöltve! Amennyiben mindent jól kitöltöttünk és sikeres volt a poszt rögzítés, azt a képernyő jelezni fogja számunkra, szintúgy, ha valamit elfejeltettünk kitölteni.

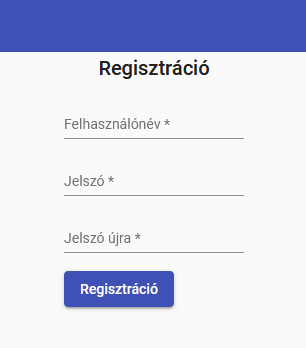


## Szoba foglalás rögzítés

A poszt létrehozásához hasonlóan lehetőség van szobafoglalást is rögzíteni. Itt is fontos, hogy minden mező megfelelően ki legyen töltve! További kritérium, hogy szobafoglalás csak akkor jöhet létre, ha az adatbázisban a kijelölt napra a kijelölt szoba nem lehet szerepelhet olyan időintervallummal, amely belecsúszna a rögzíteni kívántba.



## Regisztráció és jelszó módosítás



Regisztráció és jelszó módosításnál hasonlóan, ha minden mezőt megfelelően kitöltöttünk, akkor a kért módosítás/fiók létrehozás véghez fog menni.