Alkalmazás fejlesztések GY.

Classroom Application

DOKUMENTÁCIÓ

Foltán Dániel
LC5K0C
foltandani@gmail.com
github.com/foltandaniel

Albirt Mátyás
S6ACON
amatyi96@gmail.com
github.com/amatyi96

Bárkányi Dominik
TIFYTA
barkanyi96@gmail.com
github.com/barkanyidominik

Tartalomjegyzék

Fejlesztői környezet

Bemutatás

Az applikáció fejlesztése Windows 10 -es rendszeren történik és IntelliJ IDEA fejlesztői környezetet használunk, amely egyetemi hallgatók részére ingyenesen elérhető. Érhető és könnyű kezelése mellett nagy segítséget nyújt akár másféle program elkészítésében is.

Beállítás

Spring Boot projektünket a Spring Initializer(https://start.spring.io/) eszközzel készítjük el, ahol a következő beállításokat használjuk:

- Felül a lenyíló listákban:
  - maven
  - java
  - 1.5.7
- Group: hu.elte.alkfejl
- Artifact: classroomApplication

Használt technológiák

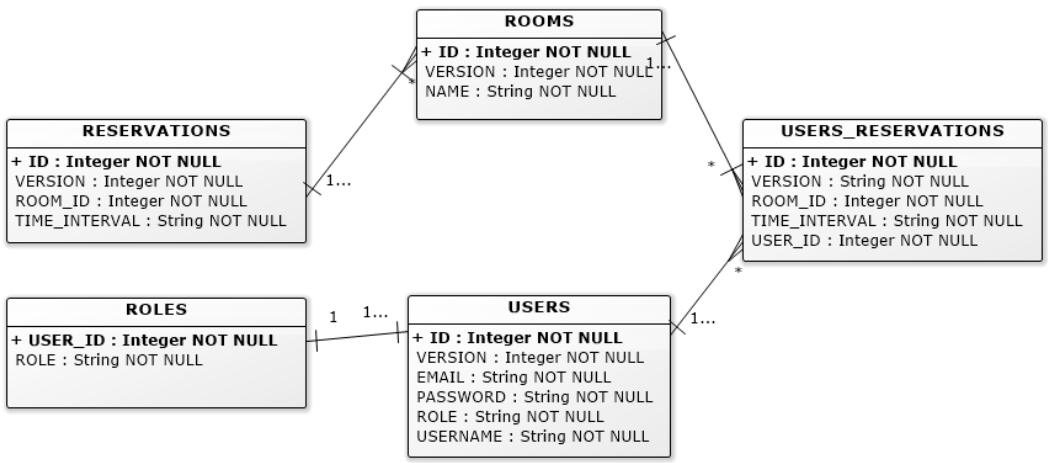
A projektünkhöz több dependencies-t is használunk segítségképpen:

- H2 - in memory database
- JPA - spring-data-jpa -> repositoryk használatához
- Lombok - automatikus getter, setter, constructor generálás
- Thymeleaf - nézet generálása
- DevTools - fejlesztéshez segítség

Beállítások után letölthető a Spring Boot projektünk, amelyet IntelliJ IDEA-ban megnyitva fejleszthetünk is.

Adatbázis-terv

UML diagram



Könyvtárstrúktúra

Bemutatás

Projektünkben MVC architektúrát válósitunk meg, ami egy olyan struktúra, amely segítségével a szoftverünk szerepköreit szétválasztjuk, az egyes rétegeket egymástól külön kezeljük.

A struktúra megtartásával az egyes rétegek könnyebben cserélhetőek, kiegészíthetőek, a szoftver jól struktúrált lesz és átlátható. Az újabb funkciók implementációja során a megoldandó problémát is kisebb részekre tudjuk így bontani.

Model: Az MVC-ben az M a Model-t jelenti. Ebben a rétegben írjuk le az adatbázisban található entitásokat és ezek kapcsolatait.

Használt modeleink:

- Reservation.ts
- Room.ts
- User.ts
- UserReservation.ts

Controller: A Controller feladata a beérkező HTTP kéréseket feldolgozni, és ezekre válaszolni. Végpontokat definiálunk, amelyeket a külvilág (Angular) alkalmazás elér HTTP kérésekkel és ezekre választ kap. Gyakran készítünk egy Service réteget is a Controller mellé, hogy mg jobban szétválasszuk a felelősségi köröket: A controller csak a kérések fogadásával és a válaszadással foglalkozik, a Service réteg a kérés és válasz közötti adat feldolgozással, adatbázissal való kommunikációval.

Használt controllereink:

- auth.service.ts
- reservaton.service.ts

Service: A Service réteg szigorúbban véve a Controller-hez tartozik, az üzleti logikát tartalmazza. A mi feladatunkban például a felhasználók validálásának menetét és a regisztrálást fogja tartalmazni.

Data Access Layer: Lényegében a Model-be tartozik, de csak most jutottunk el a használatához, ezért mutatom itt be.

View: A View rétegbe kerülnek azok a fájlok, amelyek a megjelenést biztosítják, tehát esetünkbe a html fájlok, amelyekbe a böngésző értelmezni fog.

Használt view-jaink:

- menu.component.ts
- rooms.component.ts
- dashboard.component.ts
- error.component.ts
- login.component.ts
- register.component.ts
- new-reservation.component.ts
- reservation-detail.component.ts
- reservation-list.component.ts

Egyéb osztályaink:

- ServerRoutes.ts
- app.component.ts
- app.module.ts
- MaterialItemsModule.ts
- routes.ts

Végpont-tervek

Leírások

Jelen állapotban 3 db végpontunk van tesztelés céljából, de a későbbiekben bővülni fog.

- Register
- Login
- Rooms

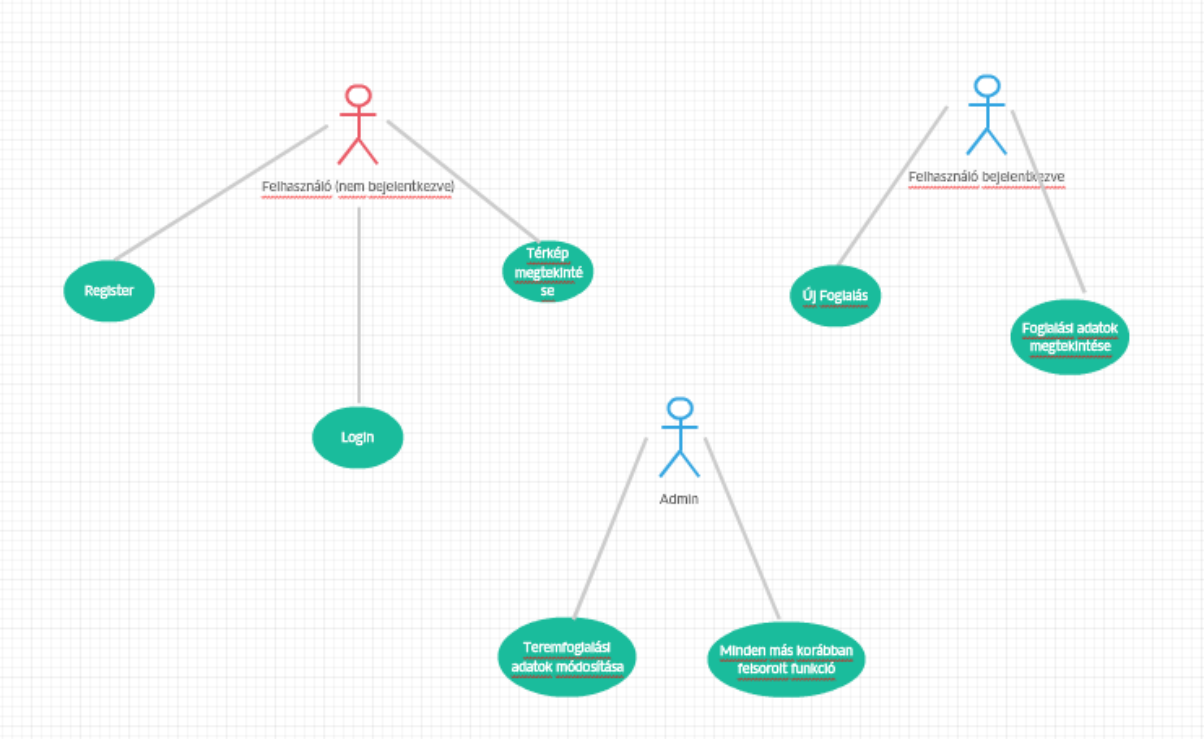
1 db végpont müködésének leírása(szekvenciadiagram)

Login

```
import {Component, OnInit} from '@angular/core';
import {User} from "../../model/User";
import {AbstractControl, FormControl, FormGroup, Validators} from '@angular/forms';
import {AuthService} from "../../services/auth.service";
import {Router} from "@angular/router";

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
  user: User;
  loginForm: FormGroup = new FormGroup({
    username: new FormControl('', [Validators.required]),
    password: new FormControl('', [Validators.required])
  });

  constructor(private loginService: AuthService, private router: Router) {}

  ngOnInit() {
    this.user = new User();
  }

  submit() {
    this.loginService.login(new User(this.username.value, this.password.value))
      .subscribe(
        res => this.router.navigate(['/reservation']),
        err => console.log(err))
  }

  get username(): AbstractControl {
    return this.loginForm.get('username');
  }

  get password(): AbstractControl {
    return this.loginForm.get('password');
  }
}
```

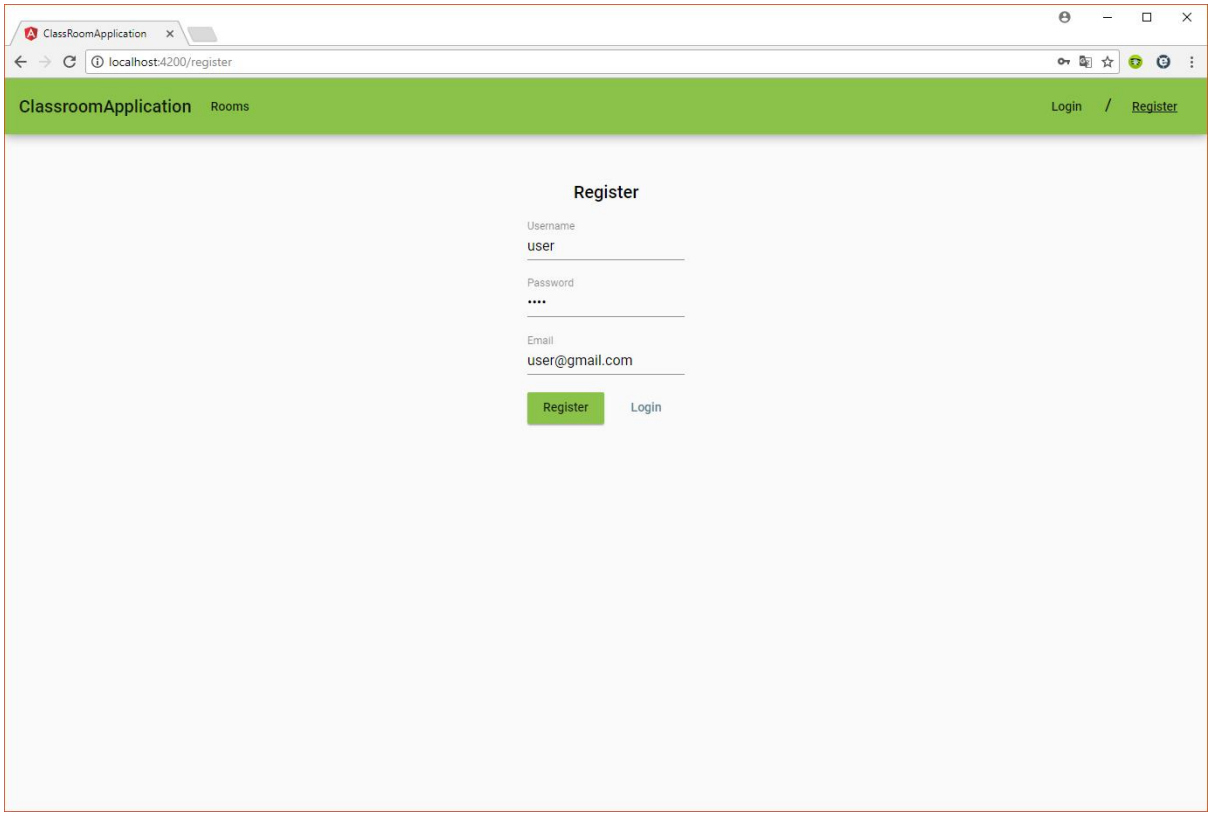
Fentebb látható a loginért felelős osztály, amely a loginForm-ról a megfelelő get metódusokkal lekéri a megadott felhasználó nevet és jelszót, majd a submit segítségével a loginService működésbe lép, amely leellenőrzi, hogy az adott felhasználó szerepel-e az adatbázisban és ha igen, akkor a hozzá tartozó jelszó megfelelő-e, ha ezek mind teljesülnek, akkor a felhasználó átirányításra kerül a reservation oldalra, ellenkező esetben, pedig a log-ban kiírásra kerül a hibaüzenet.

Felhasználói esetek

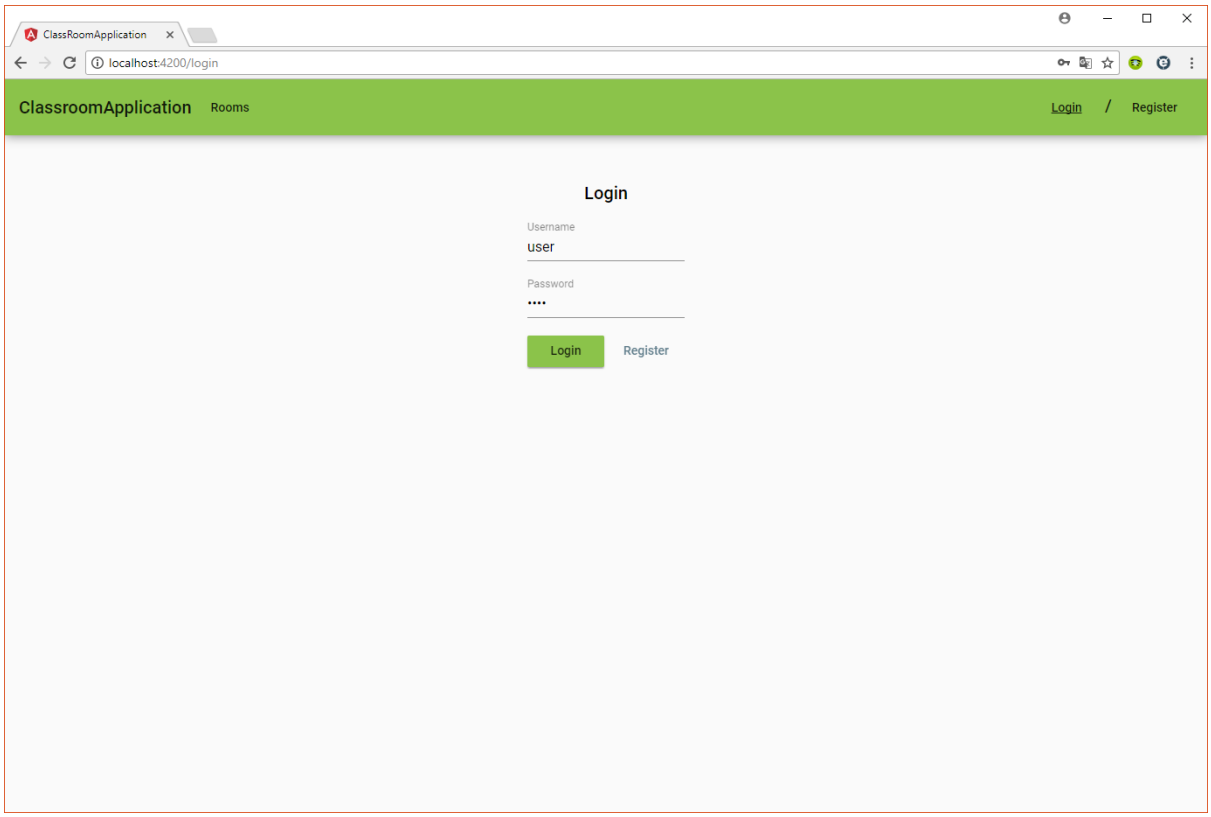


Mockups

Register mockup

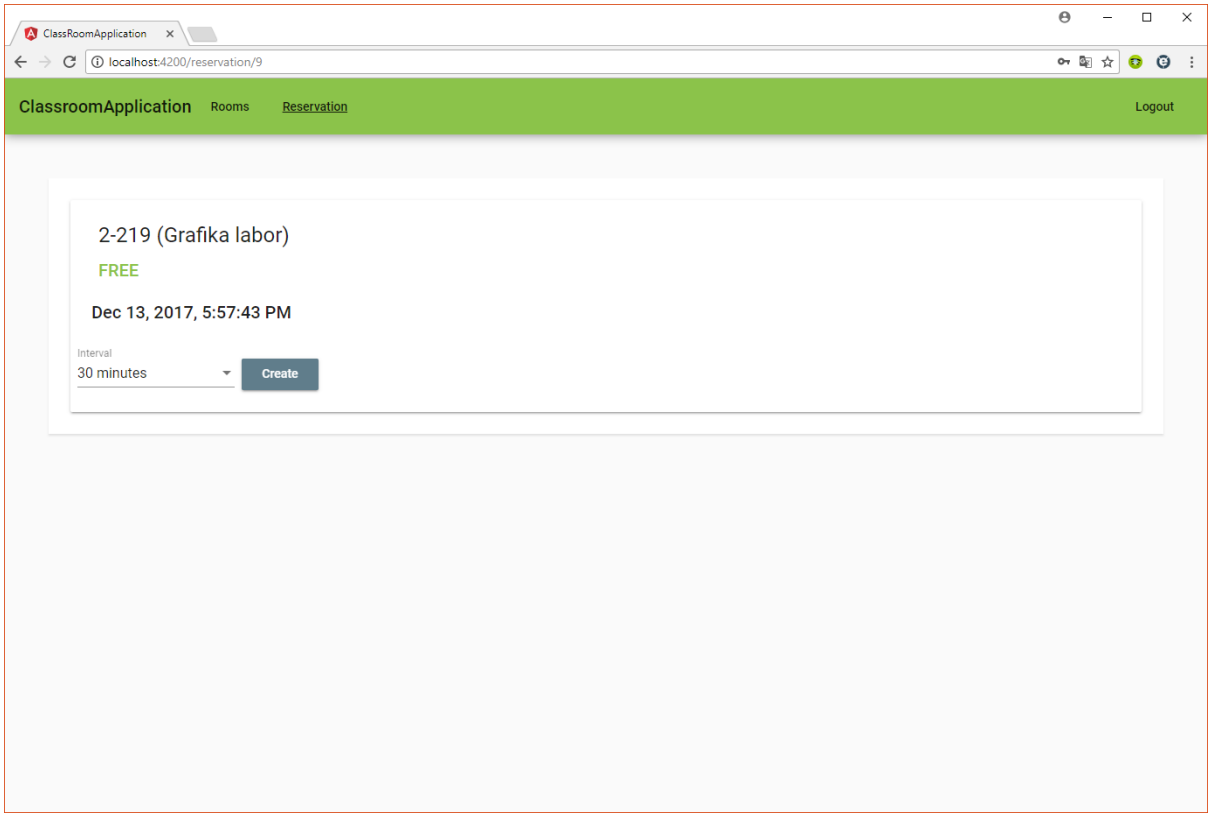


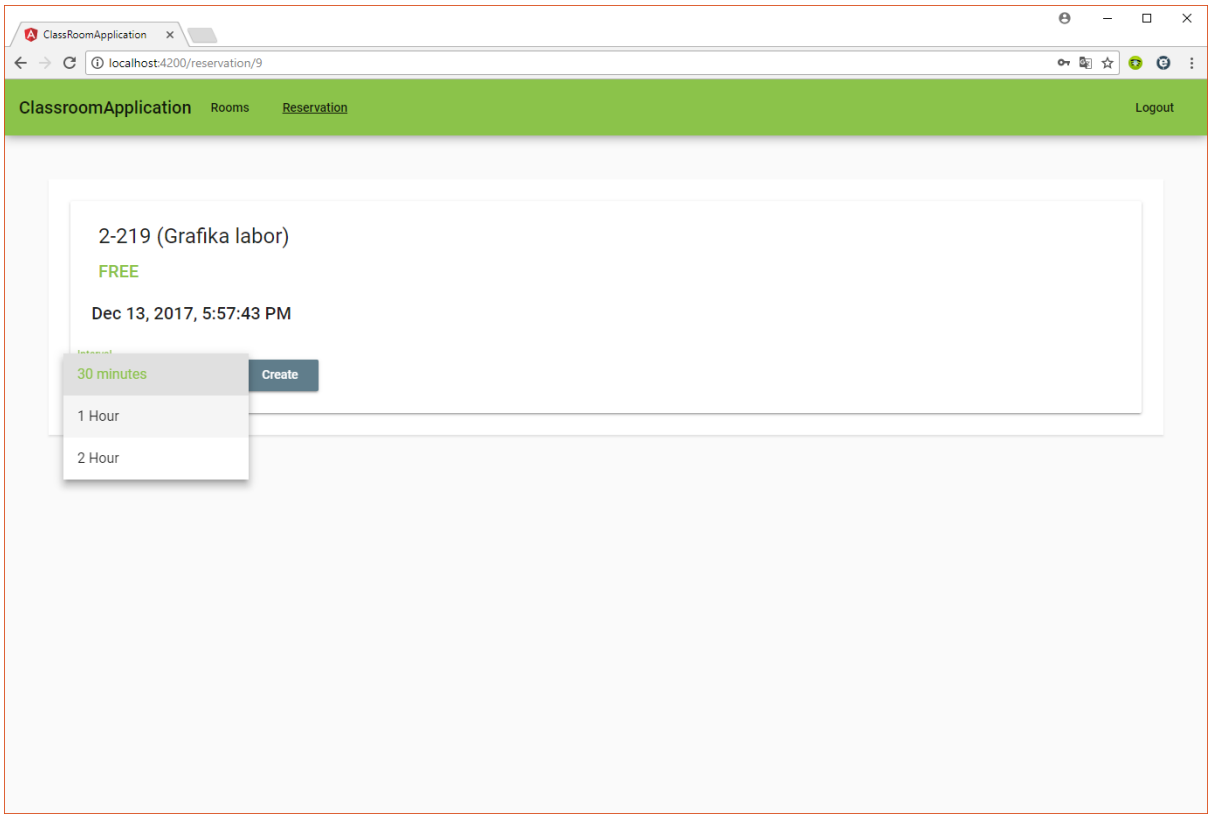
Login mockup



Rooms mockup

Reservation mockup





Felhasználói dokumentáció

Ajánlott követelmények

A program bármilyen operációs rendszeren futtatható. Platformfüggetlen.

Telepítés

Telepítés előtt pár szükséges lépés

- Kegy NodeJS és az npm amit itt lehet letölteni : www.npmjs.com/get-npm
- Internet elérés

Telepítés

1. Látogasson el a github.com/foltandaniel/beadando oldalra
2. Itt kattintson a **"Clone and Download"** gombra, és azon belül kattintson a **"Download as Zip"** gombra.
3. A letöltött állományt csomagoljuk ki
4. **npm i** parancsot adjuk ki parancssorban a kicsomagolt állomány mappájában.
5. **.env.example**-t nevezzük át **.env**-re és állítsuk be tetszés szerint
6. **npm start**-al elindíthatjuk a programot

Használata

1. Böngészőben a keresősávba írjuk be az **.env** fájlban megadott elérést.
2. Regisztrálás és bejelentkezés után használhatjuk a programot.