

RoomHub

Egy terem-foglaló alkalmazás specifikációja
Szoftverarchitektúrák házi feladat

Pilinszki-Nagy Csongor, Ghadri Najib

November 2018

Tartalom

| | |
|--|-----------|
| 1 Bevezetés | 3 |
| 1.1 A rendszer célja, funkciói és környezete | 3 |
| 1.2 A rendszer által biztosítandó tipikus funkciók | 3 |
| 1.3 A program környezete | 4 |
| 2 Megvalósítás | 5 |
| 2.1 Architektúra | 5 |
| 2.1.1 Adatbázis réteg | 6 |
| 2.1.2 Adathozzáférési réteg (Data Access Layer) | 6 |
| 2.1.3 Üzleti logikai réteg (Business Logic Layer) | 7 |
| 2.1.4 Felhasználói felület - UI | 7 |
| 2.2 Adat- és adatbázisterv | 8 |
| 2.2.1 Kapcsolatmodell | 8 |
| 2.2.2 Az adatbázis entitás-relációs diagramja | 8 |
| 2.2.3 A program objektummodellje | 8 |
| 2.2.4 Event entitás | 8 |
| 2.2.5 Room entitás | 9 |
| 2.2.6 User entitás | 9 |
| 2.3 Biztonság | 9 |
| 3 A forráskód felépítése | 10 |
| 4 Eredmény és UI Dizájn | 13 |
| 5 Telepítési leírás | 20 |
| 5.1 Rendszerkövetelmények | 20 |
| 5.2 Telepítés Linuxon | 20 |
| 5.3 Telepítés Windowson | 20 |
| 6 A program készítése során felhasznált eszközök | 21 |
| 7 Összefoglalás | 22 |
| 8 Hivatkozások | 23 |

1 Bevezetés

1.1 A rendszer célja, funkciói és környezete

A hallgatók feladata egy online teremfoglaló rendszer készítése, amely lehetővé teszi termek foglalását rendezvényekhez rendelve. Követelmények:

- háromszintű jogosultságkezelés: admin, regisztrált felhasználó, látogató
- az adminisztrátor felvehet termeket
- a regisztrált felhasználó lefoglalhat termeket
- publikus és privát rendezvények vannak
- a látogató látja a publikus rendezvényekhez foglalt termeket naptárba rendezve

1.2 A rendszer által biztosítandó tipikus funkciók

- Látogatók funkciói
 - Publikus események megtekintése
- Regisztrált felhasználók funkciói
 - Új esemény létrehozása
 - * Dátum választása
 - * Terem foglalása
 - * Esemény látogathatóságának beállítása
 - Események kezelése
 - * Meglévő esemény módosítása
 - * Esemény törlése
- Adminisztrátor funkciói
 - Új terem felvétele
 - Termek kezelése
 - * Terem módosítása
 - * Terem törlése

1.3 A program környezete

Annak érdekében, hogy a szoftver több operációs rendszeren (Linux, macOS) is futtatható legyen , platformfüggetlen megoldásokat választottunk. A programot Python nyelven fejlesztettük, a weboldal elkészítéséhez Django használtunk. A Django rengeteg magas szintű tervezési minta használatát teszi lehetővé, ami szoftverarchitektúrális szempontból gördülékeny fejlesztést tesz lehetővé. Az adatbázis rétegen pedig SQL-t használtunk, mely a Django szintaktikán keresztül könnyen elérhető. A webes megjelenítés és a platformfüggetlen környezet garantálja, hogy a szofver széles körben használható legyen.

A program más magától nem értetődő követelményt nem támaszt a számítógépek felé.

2 Megvalósítás

Az általunk elkészített program a RoomHub nevet kapta, a méltán népszerű GitHub után. Ez kifejezi, hogy a szoftver segítségével szobákat lehet kezelni és egy egy kisebb csoport számára elérhetővé tenni. A program felépítése klientszerver alapú.

2.1 Architektúra

Az architektúra első alkalmazott tervezési mintája az MVC, ami három alapvető részre bontja a szoftvert. Az első a model, ami a logikát tartalmazza, ebben történik az adatokhoz való hozzáférés, azok kezelése és továbbítása a controller számára. A view is a controllerrel áll kapcsolatban és megjeleníti azokat az adatokat, amiket az továbbít. A controller képviseli a külvilággal való kapcsolatot és ez jelenti a view és a model közötti kapcsolatot. Ez által jól elszeparálható az adatok tárolása és megjelenítése.

Az adatokhoz való hozzáférést a háromrétegű architektúra biztosítja. Az adatbázishoz az üzleti logika kapcsolódik, ami feldolgozva az adatokat továbbítja a megjelenítési réteg felé. Ez az architektúra lehetővé teszi a részek külön való fejlesztését, úgy hogy a többi komponens felé ne változzon a működés. A Django által definiált nézetek Http kéréseket fogadva kommunikálnak, ami Http listenereken keresztül történik. Ez a megközelítés az Acceptor-Connector mintát valósítják meg. Ebben a mintában a kérések fogadása és azok feldolgozása van leválasztva. A Djangoval való fejlesztést annak előre definiált függvényeinek felülírása miatt rendkívül le tud egyszerűsödni. Ez az Interceptor tervezési minta, ami lehetővé teszi a jól definiált templatek használatát. A Django Model ORM egyszerűbb szintaxisát biztosít az egyes Http és SQL kérések lebonyolítására, ez a Wrapper-Facade tervezési minta.

A RoomHub architektúrája 4 különálló modulra bontható:

- Adatbázis réteg (Database Layer, DB)
- Adatelérési réteg (Data Access Layer, DAL)
- Üzleti logika rétege (Business Logic Layer, BLL)
- Felhasználói felület (Graphical User Interface, GUI)

Ezeket a Django által definiált függvények felülírásával Interceptor tervezési minta szerint valósítjuk meg.

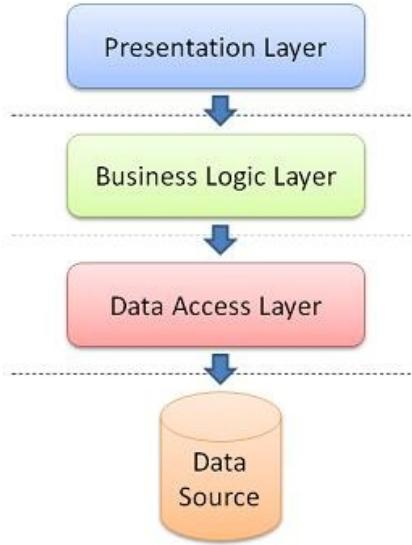


Figure 1: Ábra a szoftver architektúrájáról

2.1.1 Adatbázis réteg

Az adatbázis réteg célja az adatok tárolása. Ezt SQL-el valósítja meg melyet a Django támogat, magas szinten elérhető parancsokkal tudunk SQL kéréseket kiértékelni. Az adatbázis a db.sqlite3.db néven található meg és a Django által használt SQL implementációt használtuk fel.

2.1.2 Adathozzáférési réteg (Data Access Layer)

Célja az adathozzáférés biztosítása a felsőbb rétegek számára.

A réteg feladatai:

- üres adatbázisok létrehozása
- új entitások létrehozása az adabázisban
- adathozzáférés biztosítása a felsőbb rétegeknek

Ez a réteg a models.py kódban jelenik meg melyből a Django automatikusan generál parancsokat.

Nyújtott szolgáltatások:

- Új entitás példányosítása és perzisztálása
 - Event
 - Room

- User
- Entitás törlése
 - Event
 - Room
 - User
- Entitás tulajdonságainak módosítása
 - Event
 - Room
 - User
- Entitás tulajdonságainak lekérdezése
 - Event
 - Room
 - User

2.1.3 Üzleti logikai réteg (Business Logic Layer)

Célja: Az adatkezelési rétegtől kapott adatok alapján kiszolgálni a grafikus felületet.

A réteg lényege, hogy ellenőrzi az egyes adatokhoz való hozzáférések helyességét. minden feltétel, ami szükséges egy egy művelet teljesítéséhez az üzleti logikába került be. Ez az esetünkben két nagy részre bontható, a jogosultságok kezelésére illetve az időpontok egyeztetésére az ütközése elkerülése okán.

Megjelenés a kódban: Ez a rész a views.pyban található, illetve az abból a Django által generált kódokban.

2.1.4 Felhasználói felület - UI

Célja: A felhasználó számára egyszerű, könnyen átlátható, kényelmes felületet nyújtani, amin keresztül az összes funkció gyorsan elérhető és azok eredménye látható.

A grafikus felületen keresztül léphet be az ember vagy regisztrálhat. Ezt követően tekintheti meg az események egy részét jogosultságától függően. Az események közötti navigáció egyszerű, azoknak módosítása könnyen elérhető. A user minden pillanatban az oldal tetején látja a lehetőségeit, mint az eseményeinak megtekintése, új esemény létrehozása, vagy profiljának kezelése.

Az oldalt Bootstrap segítségével formáztuk meg, hogy az alap html felépítéshez képes fogyaszthatóbb tartalmat kapjunk

Megjelenése a kódban: views.py, .html és .css fájlok.

2.2 Adat- és adatbáziserv

2.2.1 Kapcsolatmodell

Adatbázisunkban több táblát használunk, egyet-egyet az egyes entitásokra. Ezek közötti kapcsolatot 1:N kapcsolatok segítségével tudjuk kifejezni. Egy szobához tartozhat több esemény, ezek minden a szoba id-jához vannak rendelve. Egy felhasználó szervezhet több eseményt, ezek a felhasználó id-jához vannak rendelve. Ezáltal az események táblája egy junction tablenek is tekinthető, mivel azon keresztül a szobák és felhasználók között egy N:N kapcsolat valósítható meg. Ezek a kapcsolatok egyéb információkkal is el vannak látva, mint pl az esemény ideje, így nem csak egyszerű hozzárendelésről van szó.

2.2.2 Az adatbázis entitás-relációs diagramja

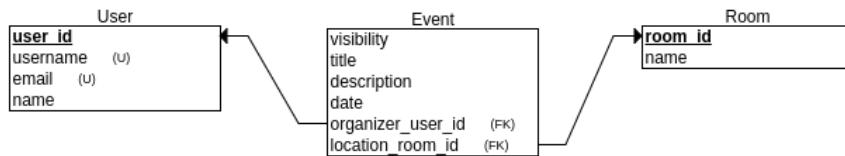


Figure 2: Relációs diagram

2.2.3 A program objektummodellje

Ebben az alfejezetben részletesen is bemutatjuk az egyes entitások felépítését és azok közötti kapcsolatokat.

2.2.4 Event entitás

Célja: Egy egy esemény nevének leírásának időpontjának és termének tárolása, valamint az azt létrehozó felhasználó azonosítása.

Tulajdonságai:

| Mezőnév | Adattípus |
|-------------------|---------------|
| visibility | BoolField |
| title | CharField |
| description | CharField |
| description | Charfield |
| date | DateTimeField |
| organizer_user_id | ForeignKey |
| location_room_id | ForeignKey |

2.2.5 Room entitás

Célja: Célja a szobák tulajdonságainak tárolása, annak nevének és azonosítójának formájában.

Tulajdonságai:

| Mezőnév | Adattípus |
|---------|--------------|
| room_id | IntegerField |
| name | CharField |

2.2.6 User entitás

Célja: A felhasználók adatainak tárolása, azok nevével, címével és azonosítójával.

Tulajdonságai:

| Mezőnév | Adattípus |
|----------|--------------|
| user_id | IntegerField |
| username | CharField |
| email | CharField |
| name | CharField |

2.3 Biztonság

A program tervezésekor a biztonságot is a Django által kínált eszköztárral valósítottuk meg. A bemenő adatok és azok jogosultsága folyamatos ellenőrzés alatt áll, így nem történhet nem megfelelő személy általi adat kinyerés vagy módosítás. Emelett szempont volt a visszatérő hibaüzeneteknél, hogy azok ne okozzanak adatszivárgást, így azok jelentése egységesen jelzi a felhasználónak, hogy egy egy műveletre nem képes, több információt nem szolgáltat a hiba okáról. Ez abban az esetben lehet fontos, amikor valaki fel szeretné deríteni az egyes termekben tartandó privát eseményeket, időpontok próbálgatásával. minden kérést a Django által biztosított HTTP kéréseken keresztül intézünk, ami támogatja az authentifikálás, így minden művelet annak válik elérhetővé, aki jogosult is rá.

3 A forráskód felépítése

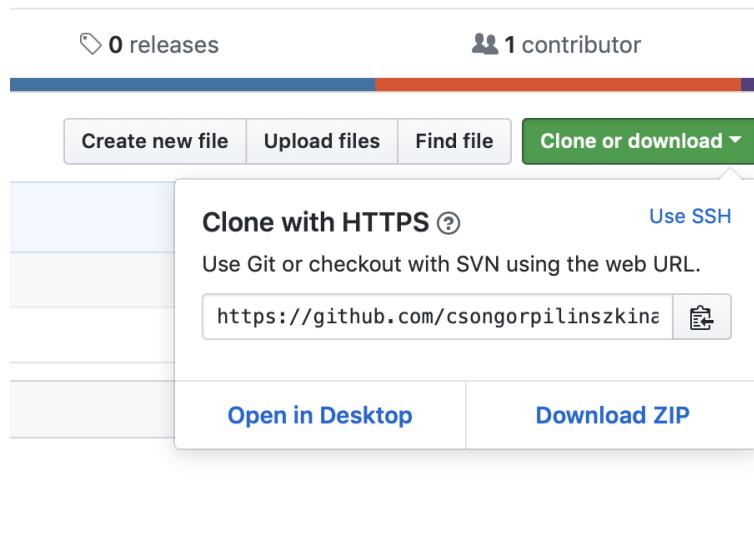


Figure 3: Github klónozás

A kód a GitHubon keresztül a <https://github.com/csongorpilinszkinagy/roomhub.git> címen érhető el. Ezt klónozva használható is.

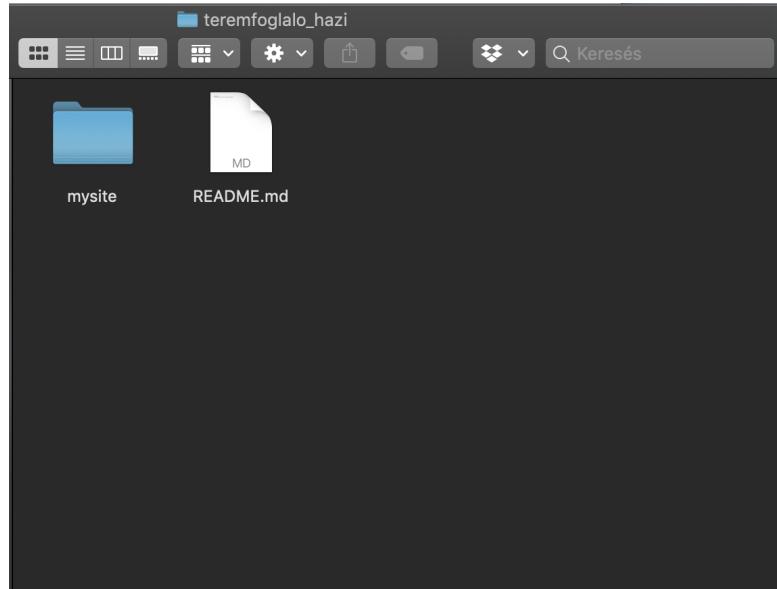


Figure 4: Fájlok a roomhub mappában

A fájlok mappaszerkezete a következő képeken látható. Két fő részre botható egy mysite és egy rooms mappára. Az első a kinézetért felelős, míg a másodikhoz köthető a logika.

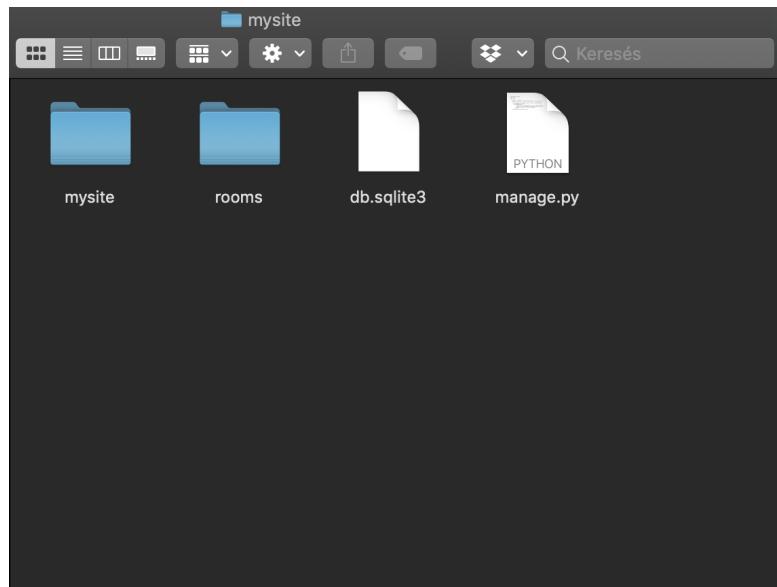


Figure 5: Fájlok a mysite mappában

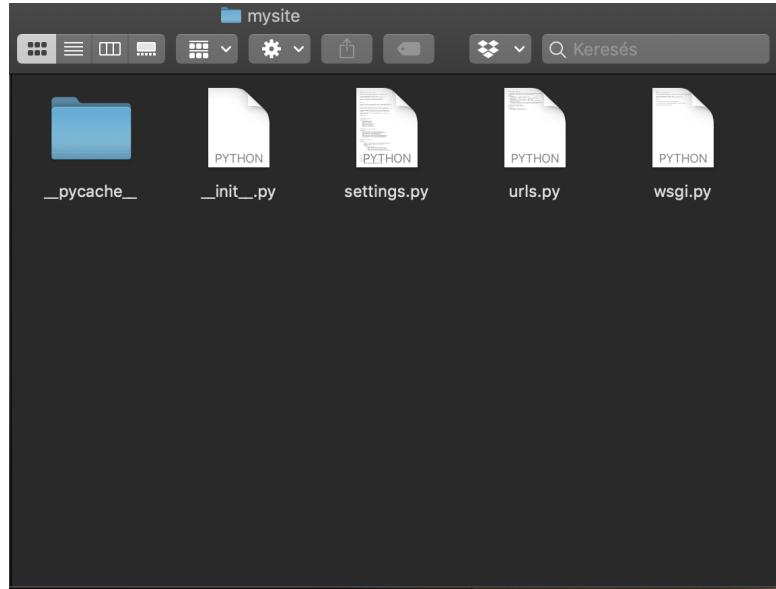


Figure 6: Fájlok a mysite/mysite mappában

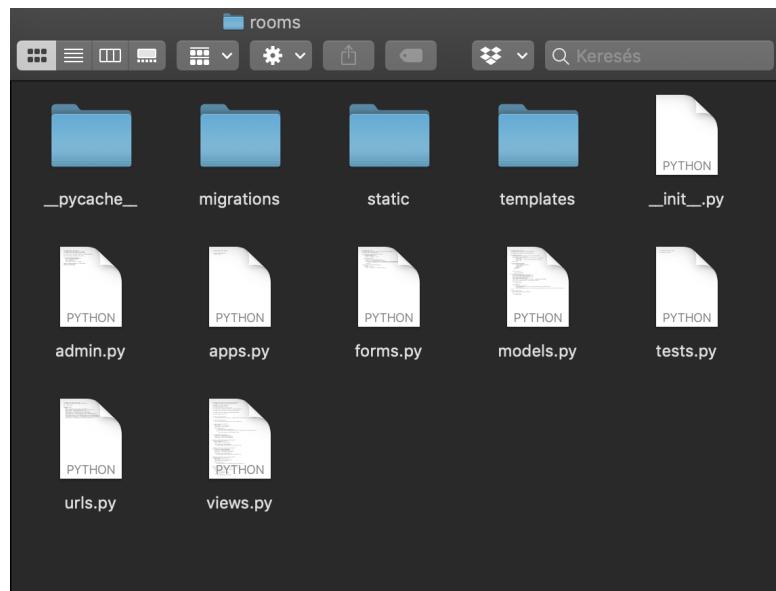


Figure 7: Fájlok a rooms mappában

```

Pilinszki-Nagy-MacBook-Pro:teremfoglalo_hazi pilinszki-nagycsongor$ cd mysite
Pilinszki-Nagy-MacBook-Pro:mysite pilinszki-nagycsongor$ python3 manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
November 30, 2018 - 07:31:14
Django version 2.1.3, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

```

Figure 8: Szerver indítása

A server indítása a python3 manage.py runserver parancssal elérhető, majd a konzolon írt címen keresztül a <http://127.0.0.1:8000/roomhub> elérési úton keresztül tekinthető meg a kezdőoldal.

4 Eredmény és UI Dizájn

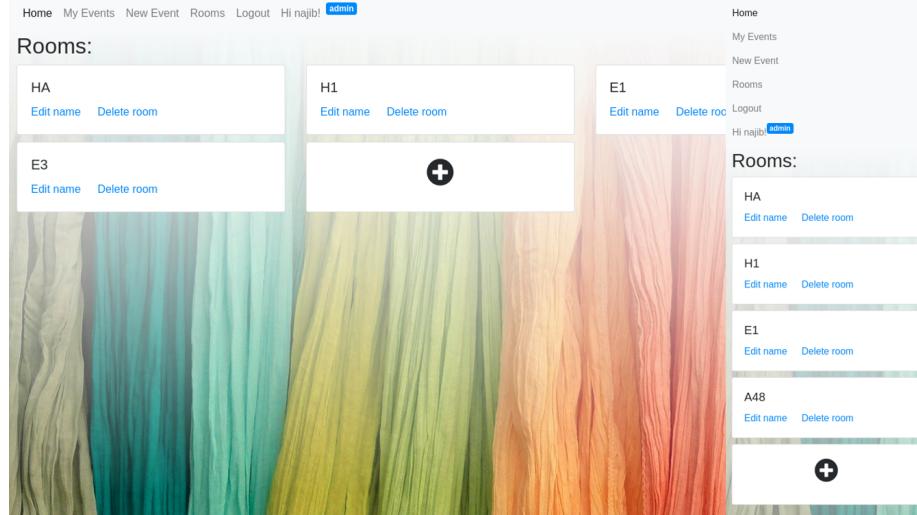


Figure 9: Kezdőképernyő adminon és reszponzív nézet

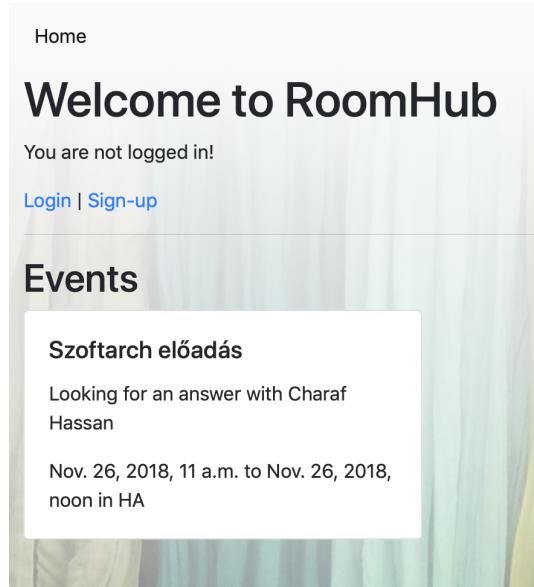


Figure 10: Kezdőképernyő

A kezdőképernyőn láthatóak a publikus mindenki számára elérhető események. Innen beléphetünk vagy regisztrálhatunk, illetve megtekinthetjük az egyes események részleteit is.

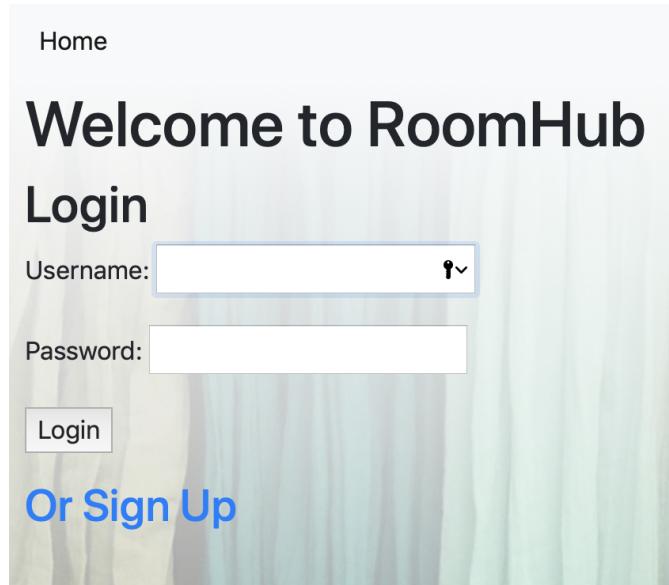


Figure 11: Login

A bejelentkezési felület a felhasználói adatok megadásával a felhasználói fiókhoz vezet.

The screenshot shows the 'Welcome to RoomHub' sign-up page. At the top left is a 'Home' link. Below it, the title 'Welcome to RoomHub' is displayed in large, bold, black font. Underneath the title is the heading 'Sign up'. There are three input fields: 'Username' (with placeholder text 'Required. 150 characters or fewer. Letters, digits and @./+/-/_ only.'), 'Email address', and 'Password'. Below the password field is a list of four password requirements: 'Your password can't be too similar to your other personal information.', 'Your password must contain at least 8 characters.', 'Your password can't be a commonly used password.', and 'Your password can't be entirely numeric.'. To the right of the password field is a note: 'Enter the same password as before, for verification.' Below these fields is a blue 'Go!' button. At the bottom left is a link 'Or Log In'.

Figure 12: Signup

A signup a szükséges adatok megadásával történik, ami a bejelentkezési felületre irányítja a felhasználót, sikeres regisztráció után.

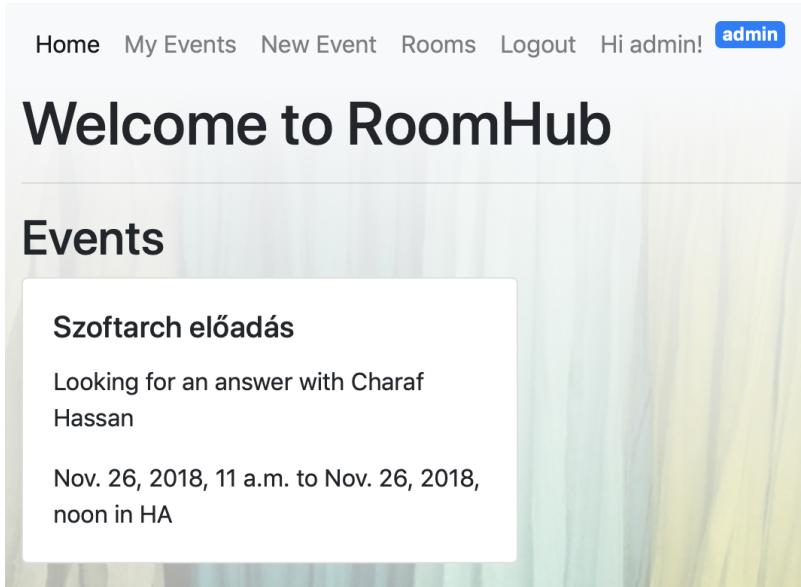


Figure 13: Admin felület

Az admin felületen elérhető minden felhasználó terem és esemény, ezeket módosíthatjuk. Itt lehet termeket felvenni, ez az admin feladata.

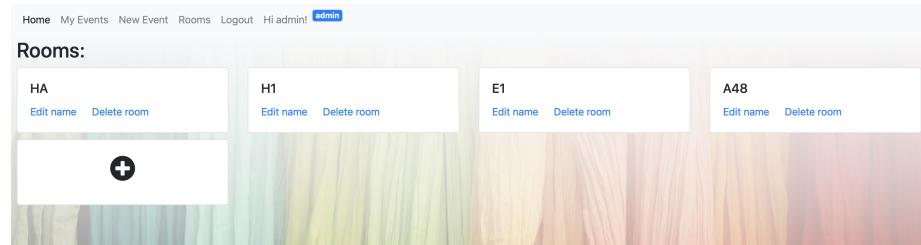


Figure 14: Szobák nézete

Home My Events New Event Rooms Logout Hi admin! **admin**

Edit room

Name: HA

Save

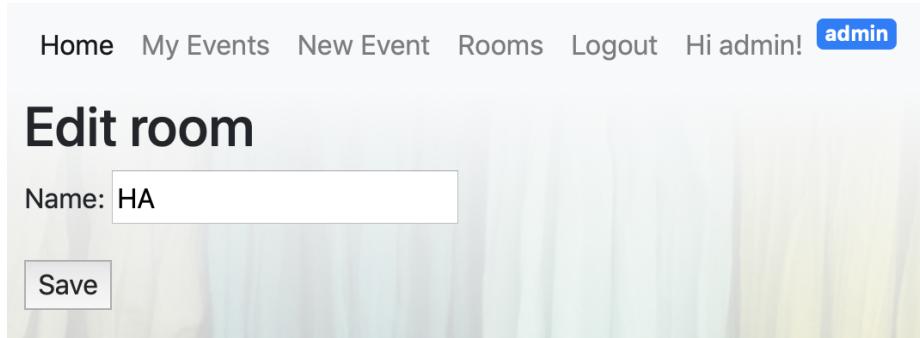


Figure 15: Szoba módosítása

Home My Events New Event Logout Hi csongor!

Welcome to RoomHub

Events

Szoftarch előadás

Looking for an answer with Charaf Hassan

Nov. 26, 2018, 11 a.m. to Nov. 26, 2018, noon in HA

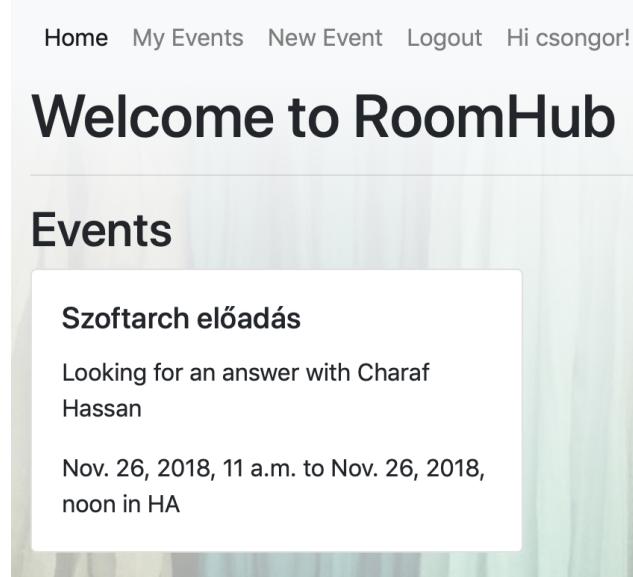


Figure 16: User felület

A felhasználói felületen elérhetőek az összes felhasználói funkciók

Home My Events New Event Logout Hi csongor!

New event

Title:

Description:

Start:

End:

Room:

Is public:

Figure 17: Esemény felvétele

Home My Events New Event Logout Hi csongor2!

Edit My new event

Title: My new event

Description: This is the new event.

Start: 2018-12-01 10:00:00

End: 2018-12-01 12:00:00

Room: Q.1

Is public:

Figure 18: Esemény módosítása

[Home](#) [My Events](#) [New Event](#) [Logout](#) Hi csongor2!

Are you sure you want to delete "My new event"?

[Confirm](#)

Figure 19: Esemény törlése

5 Telepítési leírás

5.1 Rendszerkövetelmények

- Python 3
- Linux, macOS, Windows
- Valamilyen processzor
- Memóriaigény
- Lemezterület igény

5.2 Telepítés Linuxon

Először klónoljuk a git repository-t

```
git clone https://github.com/csongarpilinszkinagy/roomhub.git
```

Telepítük a django 2.1-et

```
pip install --e django/
```

Opcionálisan készítsünk egy virtualenv-et

```
virtualenv env
```

Majd futassuk a következőket:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

```
python manage.py runserver
```

5.3 Telepítés Windowson

A repository klónolása után:

Python installálása

```
https://www.python.org/downloads/release/python-371/
```

Virtualenv installálása és konfigurálása

```
pip install virtualenvwrapper-win
```

```
mkvirtualenv myproject
```

```
workon myproject
```

Django installálása:

```
pip install django
```

Oldal konfigurálása:

```
python manage.py migrate  
python manage.py runserver
```

6 A program készítése során felhasznált eszközök

- Python
- Django
- SQL
- Chrome
- Safari
- Overleaf
- GitHub
- Messenger
- Skype
- Google Slides

7 Összefoglalás

Bemutattuk, hogy a Djangoval hogyan fejleszthető webapplikáció termek foglalására és rendezvények szervezésére. A Django által implementált magas szintű eszközök sokat segítettek a tervezési minták könnyed fejlesztésében. Bemutattuk a három használt entitást, azokkal való megvalósítást a foglaláshoz. Bemutattuk ezeknek a felépítését, az architektúra mögötti döntéseket és azok működését a példán keresztül.

8 Hivatkozások

- Django
 - <https://www.djangoproject.com/> (2018.11.30)
- Django tutorial
 - <https://docs.djangoproject.com/en/2.1/intro/tutorial01/> (2018.11.30)
- Bootstrap
 - <https://getbootstrap.com/> (2018.11.30)
- Virtualenv
 - <https://virtualenv.pypa.io/en/latest/> (2018.11.30)
- SQL Lite
 - <https://www.sqlite.org/index.html> (2018.11.30)
- GitHub
 - <https://github.com/> (2018.11.30)