|  |  |
| --- | --- |
|  | **Eötvös Loránd Tudományegyetem**  **Informatikai Kar**  **Algoritmusok és Alkalmazásaik Tanszék** |

Teremfoglaló rendszer

**Nagy Sára Tomecz Patrik**

mesteroktató Programtervező informatikus BSc

nappali tagozat

**Budapest, 2019**

Tartalomjegyzék

[1. Bevezetés 3](#_Toc529729921)

[2. Felhasználói dokumentáció 4](#_Toc529729922)

[2.1 A célközönség 4](#_Toc529729923)

[2.2 A hardveres és szoftveres követelmények 4](#_Toc529729924)

[2.3 A használati útmutató 5](#_Toc529729925)

[2.3.1 Látogatók számára elérhető felület 5](#_Toc529729926)

[2.3.2 Bejelentkezett felhasználók (tanárok) számára elérhető felület 11](#_Toc529729927)

[2.3.3 Adminisztrátorok számára elérhető felület 14](#_Toc529729928)

[3. Üzemeltetői dokumentáció 15](#_Toc529729929)

[4. Fejlesztői dokumentáció 16](#_Toc529729930)

[4.1 Tesztelés 16](#_Toc529729931)

[4.1.1 Backend 16](#_Toc529729932)

# 1. Bevezetés

Napjainkban nagy népszerűségnek örvend a webes alkalmazások fejlesztése.

A szakdolgozatom témája egy ilyen alkalmazás, egy teremfoglaló rendszer elkészítése. A rendszer megvalósítását modern, a tanulmányaim során megismert eszközökkel valósítottam meg. Törekedtem arra, hogy a felület könnyen átlátható, felhasználóbarát legyen. A fejlesztés közben az alkalmazást folyamatosan teszteltem a hibák elkerülése végett.

Az alkalmazás felülete letisztult, bárki számára könnyen kezelhető. A vendégek megtekinthetik az aktuális foglalásokat, mind táblázat, mind naptár nézetben. Lehetőség nyílik a foglalások közötti keresésre is. A tanárok/felhasználók leadhatnak új tanóra-, esemény foglalásokat, szerkeszthetik saját foglalásaikat. Az adminisztrátor felvehet új épületeket, tantermeket, szemesztereket, tantárgyakat és foglalásokat, valamint lehetősége nyílik ezek szerkesztésére is, továbbá menedzselheti az éppen várakozó kéréseket.

# 2. Felhasználói dokumentáció

## 2.1 A célközönség

Az alkalmazás célközönsége elsősorban a felsőoktatásban tanuló hallgatók és tanárok. A cél, hogy könnyebbé, átláthatóbbá tegyek a diákok számára a tanórákhoz, eseményekhez tartozó szükséges információk megszerzését, megkönnyítse az órarend összeállítását. Ezzel együtt a tanárok számára egyszerű felületet nyújt a foglalások leadására és azok szerkesztésére is.

## 2.2 A hardveres és szoftveres követelmények

A futó alkalmazás használatához az egyetlen követelmény egy webböngésző program.

Ajánlott webböngészők:

* Mozilla Firefox 62.0
* Google Chrome

A Mozilla Firefox rendszerkövetelményei Windowson:

* Operációs rendszerek:
  + Windows 7
  + Windows 8
  + Windows 10
* Hardveres követelmények:
  + Pentium 4-es vagy újabb processzor, amely támogatja az SSE2-t
  + 512MB memória / 2GB memória a 64-bites verzió számára
  + 200MB szabad tárhely

A rendszerkövetelmények Macen:

* Operációs rendszerek:
  + macOS 10.9
  + macOS 10.10
  + macOS 10.11
  + macOS 10.12
  + macOS 10.13
* Hardveres követelmények:
  + Intel x86 processzor
  + 512 MB memória
  + 200 MB szabad tárhely

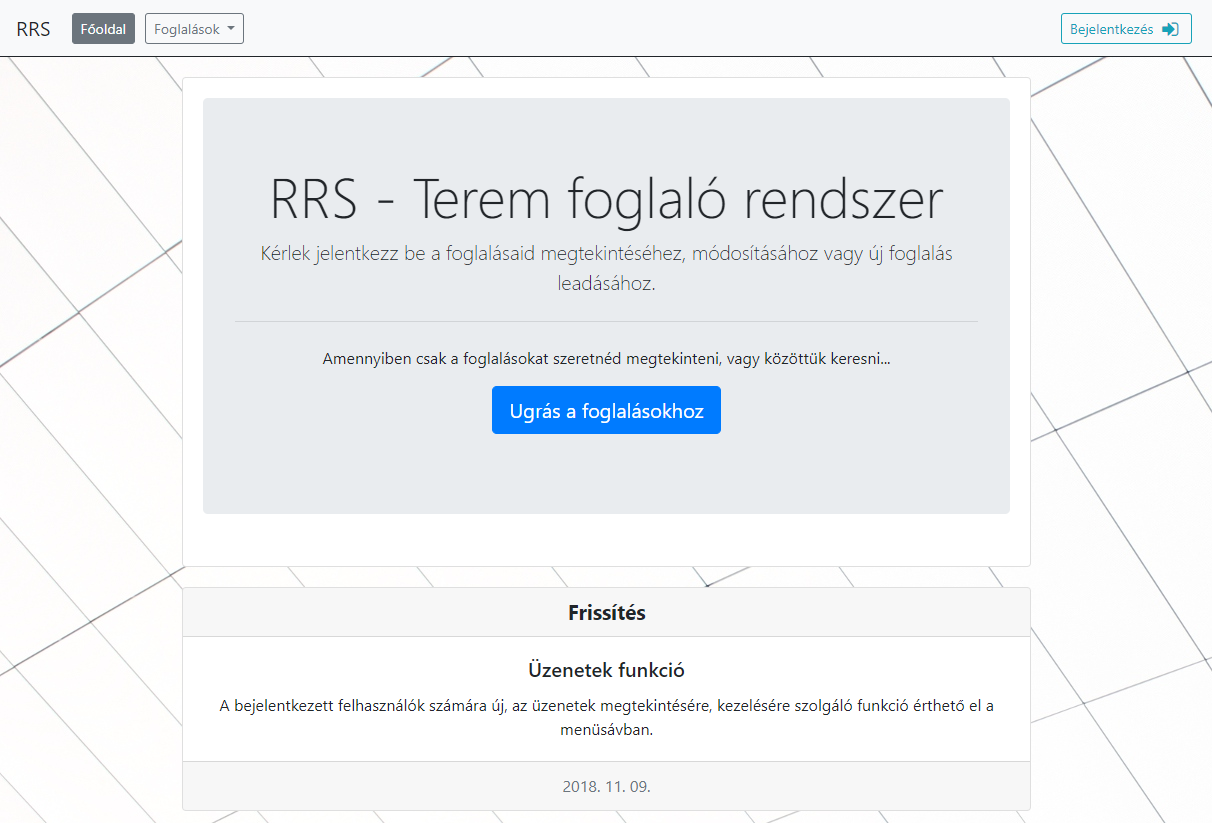
A Google Chrome követelményei hasonlóak.

## 2.3 A használati útmutató

### 2.3.1 Látogatók számára elérhető felület

#### 2.3.1.1 Főoldal

Az weboldal megnyitásakor a kezdőlap tárul a látogató szeme elé. A főoldalon értesülhetünk a hírekről, frissítésekről és a fontos tudnivalókról.



1. ábra A főoldal

#### 2.3.1.2 Navigáció

A felső állandó sávban található a menü, alatta pedig a weboldal tartalma. A menüsávban a szürke színnel történik az aktuális oldal jelölése. Kétféle menüpont található: sima és legördülő menü. A menü dinamikusan változik attól függően, hogy a történt-e bejelentkezés.

A be nem jelentkezett felhasználók számára az alábbi menüsáv érhető el:



2. ábra A vendégek számára elérhető menüsáv

Bejelentkezés után attól függően, hogy milyen engedélyekkel rendelkezik az adott felhasználó, az alábbi két menü közül fogadja az egyik:



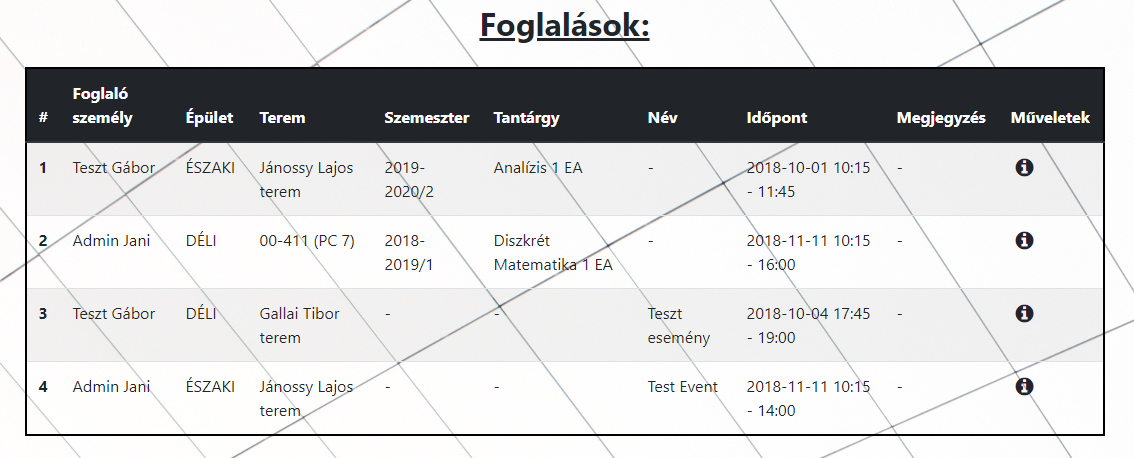
3. ábra A felhasználók számára elérhető menüsáv



4. ábra Az adminisztrátorok számára elérhető menüsáv.

#### 2.3.1.3 Foglalások

A foglalások menüpont alatt, mely egy legördülő menü, található minden, az elfogadott foglalásokkal kapcsolatos általános funkció. Az első menüpontra kattintva a foglalások táblázatos formában jeleníthetjük meg.

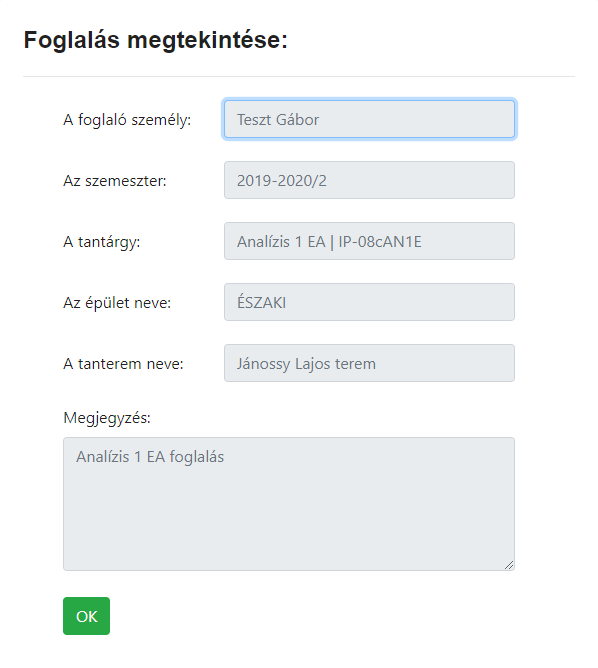


5. ábra Az elfogadott foglalások táblázatos nézetben

A táblázat egy sora egy adott foglalást jelöl és az alábbi információkat tartalmazza:

* A foglalás sorszáma. Fontos, hogy ez nem a foglalás azonosítója, ez csak a táblázaton belüli sorszám jelölésére szolgál.
* A foglaló személy neve
* Az épület neve
* A tanterem neve
* A szemeszter, ha tanórára vonatkozó foglalásról van szó
* A tantárgy neve, ha tanórára vonatkozó foglalásról van szó
* Az esemény neve, ha eseményre vonatkozó foglalásról van szó
* A megjegyzés
* Az adott eseményre vonatkozó műveletek

A „Műveletek” oszlopban a „Részletek” gombra kattintva egy információs ablak jelenik meg, amely tartalmazza az adott foglalás részleteit. Ez az ablak némileg eltérhet attól függően, hogy tanórára vagy eseményre vonatkozó foglalást vizsgálunk.



6. ábra Tanórára vonatkozó foglalás megtekintése

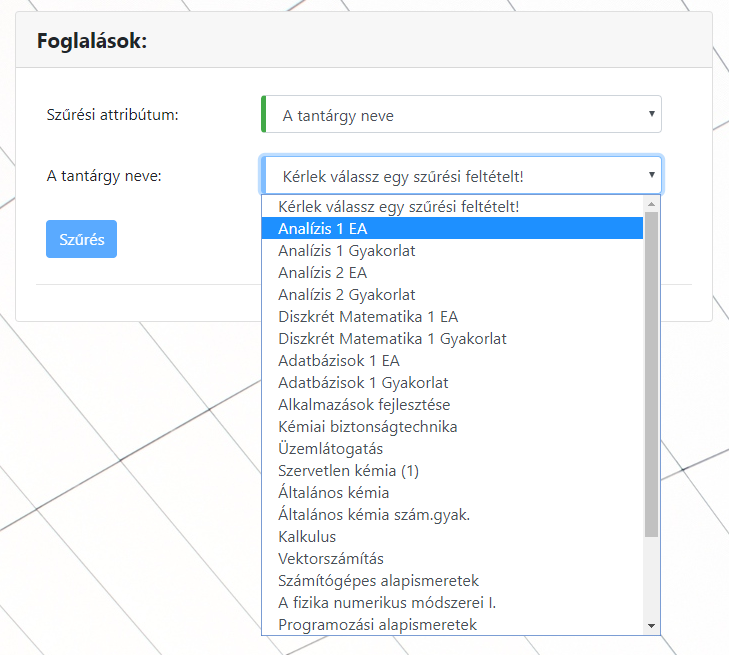


7. ábra Eseményre vonatkozó foglalás megtekintése

A második pontot választva a felhasználóbarát naptári nézethez navigálhatunk. Ahhoz, hogy a foglalások megjelenjenek először meg kell adnunk valamilyen szűrési feltételt az alábbiak közül:

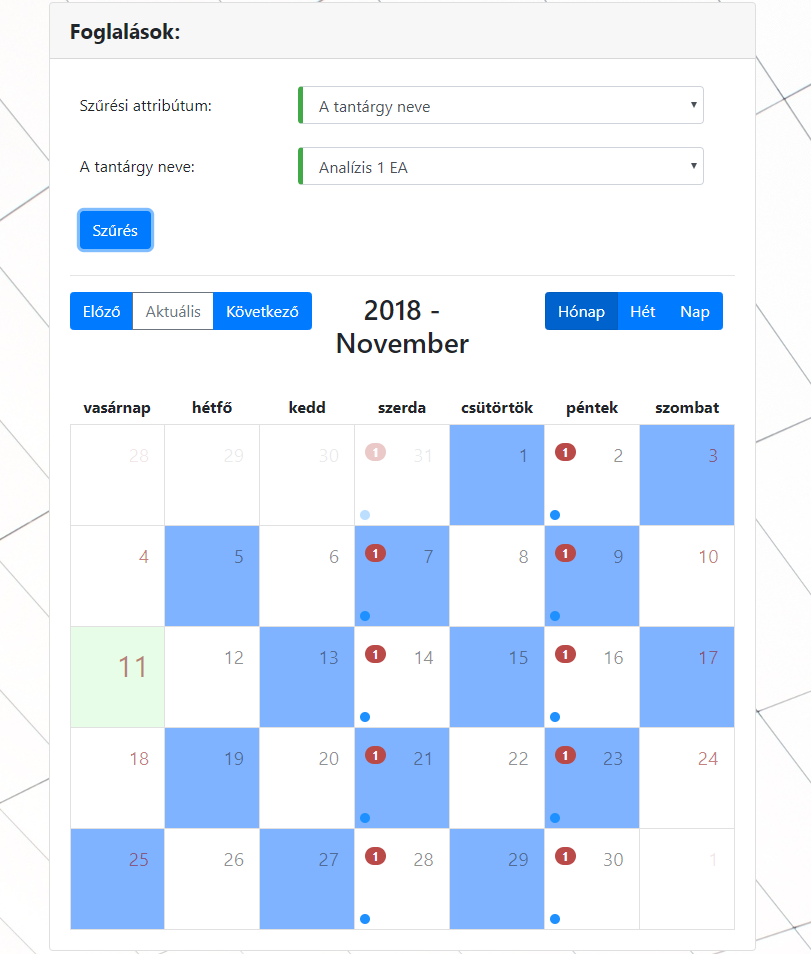
* A foglaló személy neve
* Az épület neve
* A tantárgy neve
* A szemeszter
* Az esemény neve

Abban az esetben, ha a szűrési attribútum kiválasztása sikeres, megjelenik egy második legördülő menü, ahol kiválaszthatjuk a kívánt értéket.



8. ábra A foglalásokra szűrése

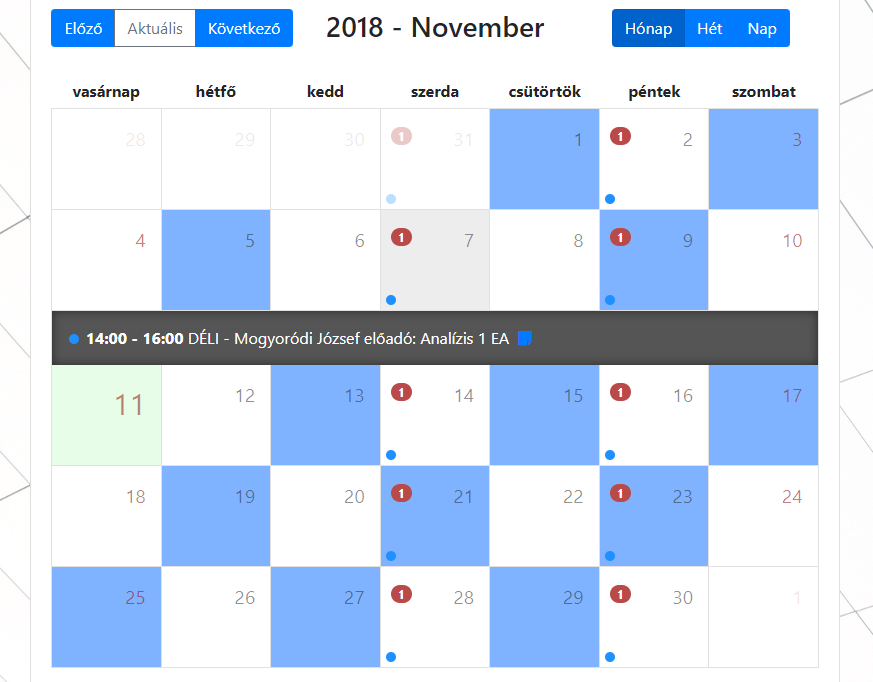
Az érték kiválasztása után, a „Szűrés” gomb aktiválódik, rá kattintva megtörténik a keresés. Ha a keresés sikeres volt, akkor lent megjelenik a Naptár nézet a megfelelő foglalásokat tartalmazva.



9. ábra Naptár (havi) nézet

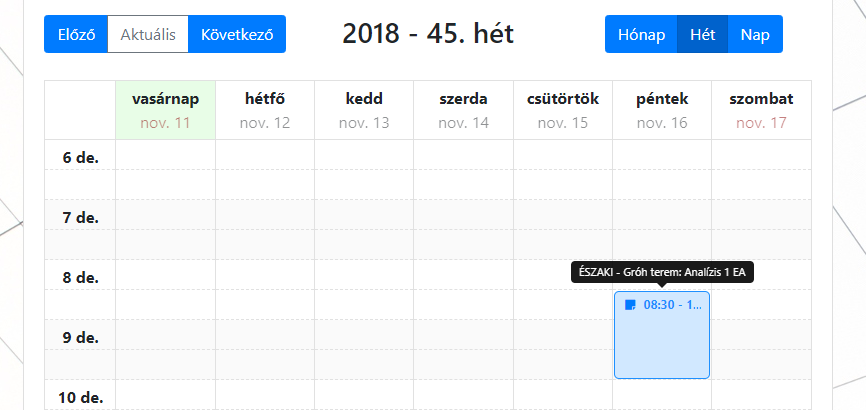
A naptár nézet több, a vendég számára hasznos, funkciót tartalmaz. A naptár lapozása az „Előző” és „Következő” gombok segítségével történik. Az „Aktuális” gombra kattintva a naptár aktuális naphoz tartozó oldala kerül betöltésre. A nézetek váltása a „Hónap”, „Hét” és „Nap” gombok segítségével valósítható meg.

A havi nézet esetén minden egyes napnál, ahol található valamilyen esemény, megjelenik egy piros háttérrel rendelkező ovális gomb. A gömbben található az adott napra vonatkozó foglalások darabszáma. A napra kattintva megtekinthetővé válik a foglalások kezdete, vége, az épület, a tanterem és a foglalás neve. A bővebb részletek megtekinthetők a foglalás neve melletti ikonra kattintva.



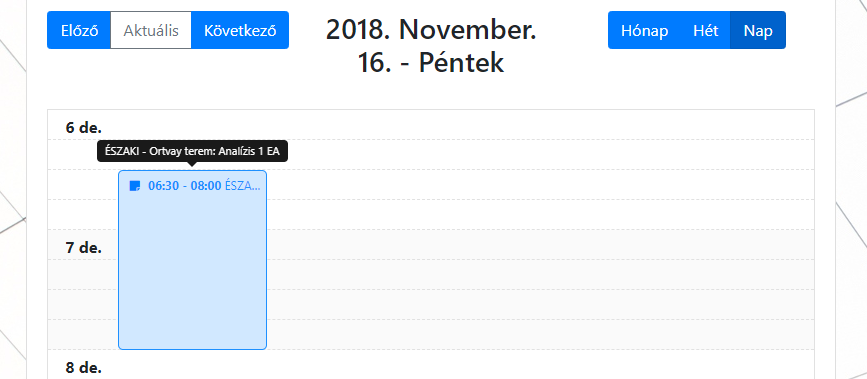
10. ábra A havi nézet

A heti nézetet megnyitva az összes, adott héthez tartozó, foglalás jelenik meg a felhasználó szeme előtt.



11. ábra A heti nézet

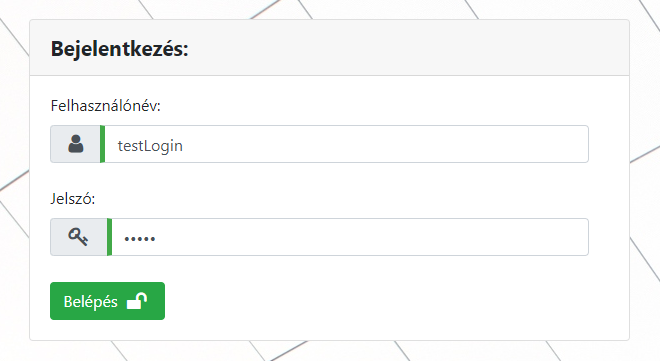
A „Nap” gombra kattintva az aktuális napra vonatkozó információk kerülnek megjelenítésre.



12. ábra A napi nézet

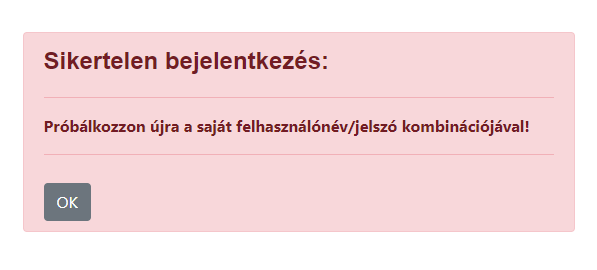
#### 2.3.1.4 Bejelentkezés

A menüsáv jobb szélén található a bejelentkezés gomb, mely a megfelelő űrlapra navigálja a látogatót.



13. ábra A bejelentkezés

Az űrlap kitöltése után a „Belépés” gombra kattintva kísérelheti meg a vendég a bejelentkezést. Sikeres bejelentkezés esetén a rendszer a főoldalra navigálja a felhasználót, azonban, ha a bejelentkezés sikertelen, az alábbi felugró ablak figyelmeztet:

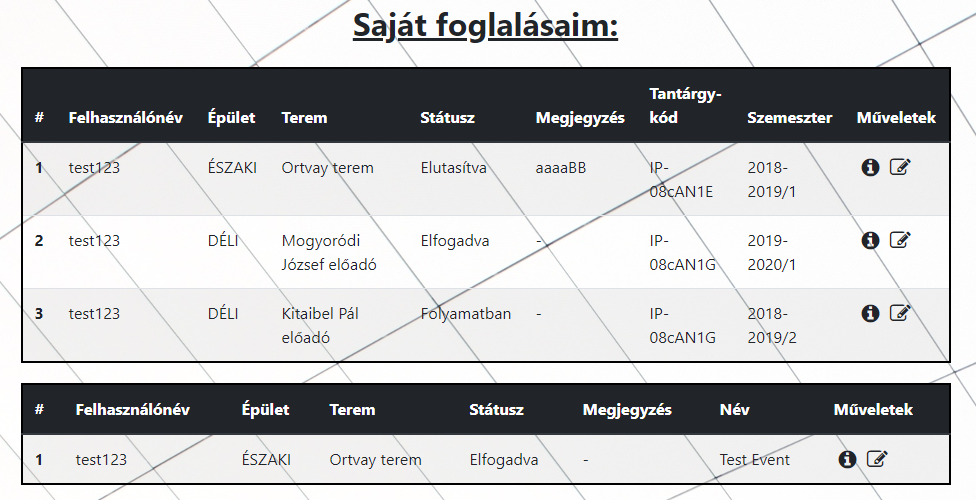


14. ábra Sikertelen bejelentkezés

### 2.3.2 Bejelentkezett felhasználók (tanárok) számára elérhető felület

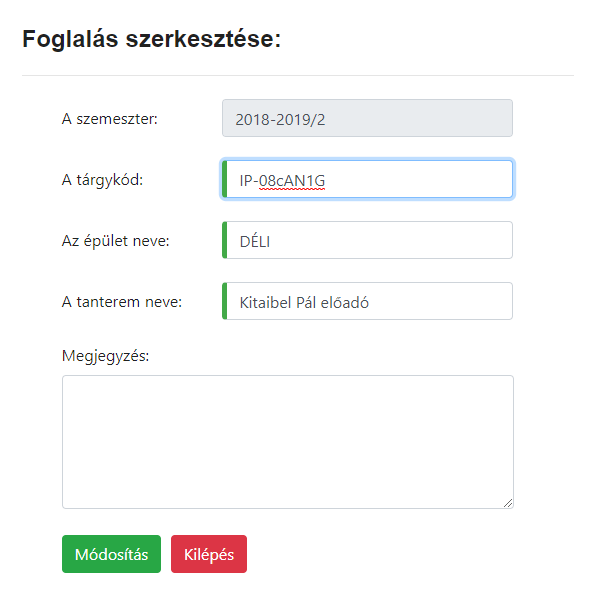
#### 2.3.2.1 Saját foglalásaim

A sikeres bejelentkezés után a felhasználót a rendszer a főoldalra navigálja, a menüsáv újabb, az alapvető elemeket kiegészítő funkciókkal bővül. A foglalásaim menüpont alatt kilistázhatjuk a saját foglalásainkat táblázatos formában státusztól függetlenül.



15. ábra A saját foglalásaim

A bejelentkezett felhasználónak lehetősége nyílik a saját foglalásai szerkesztésére. Ehhez a „Műveletek” oszlopban a „Szerkesztés” gombra kell kattintani. Ekkor az alábbi felugró ablak jelenik meg:



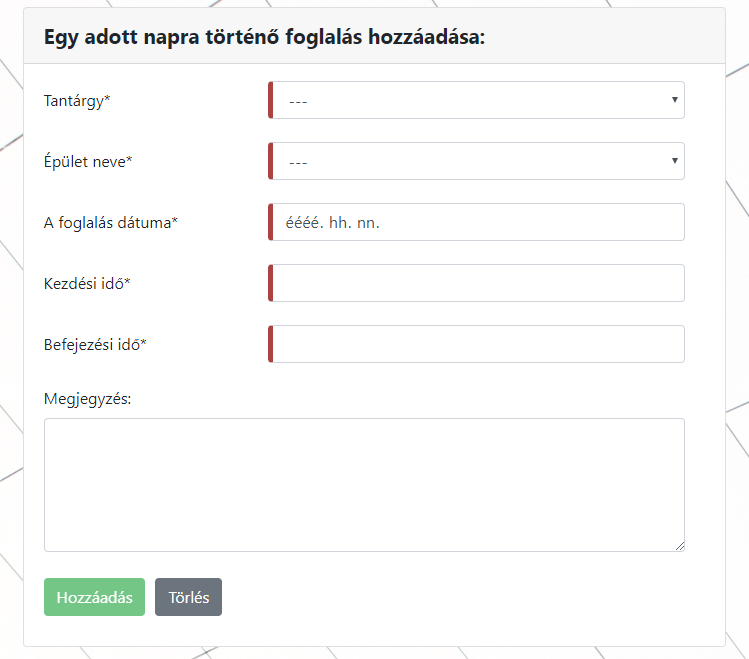
16. ábra Saját foglalás szerkesztése

#### 2.3.2.2 Űrlapok

Az űrlapok menüpont alatt három lehetőség közül választhat a felhasználó:

* Új egyszerű foglalás létrehozása
* Új szemeszterre vonatkozó foglalás létrehozása
* Új eseményre vonatkozó foglalás létrehozása

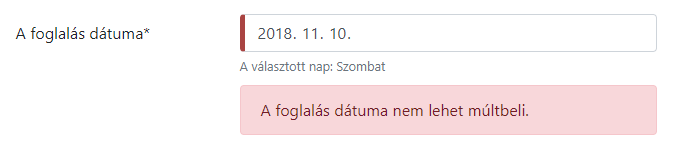
Mindhárom menüpont esetén egy-egy űrlap tárul a felhasználó szeme elé. Abban az esetben, ha a felhasználó az első menüpontot választja, azaz új egyszerű foglalást szeretne létrehozni, az alábbi űrlap jelenik meg:



17. ábra Egyszerű foglalás létrehozása

Az űrlap kitöltése során néhány adatot legördülő menük segítségével lehet kiválasztani, néhányat be kell vinni és van, amikor naptárral kell dolgozni. Ha ez sikeresen megtörtént, akkor a „Hozzáadás” gomb segítségével lehet rögzíteni a foglalást. A „Törlés” gombra kattintva az űrlap az eredti állapotba állítható vissza. A szemeszterre és eseményre vonatkozó foglalások regisztrálása hasonlóképpen történik.

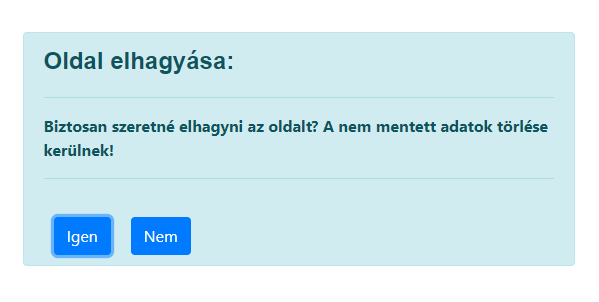
Minden egyes űrlap leadásához szükséges, hogy érvényes legyen. A mezők melletti szín jelöli az adott mező állapotát. Piros, ha a mező érvénytelen és zöld, ha érvényes. Abban az esetben, ha egy mező már érintett és mégis érvénytelen, akkor a mező alatt piros alapon hibaüzenet jelenik meg. Ennek a célja, hogy segítse a felhasználót a mező helyes kitöltésében.



18. ábra Egy érvénytelen mező

Az űrlap akkor számít érvényesnek, ha minden mező érvényes és egyetlen hibaüzenet sincs.

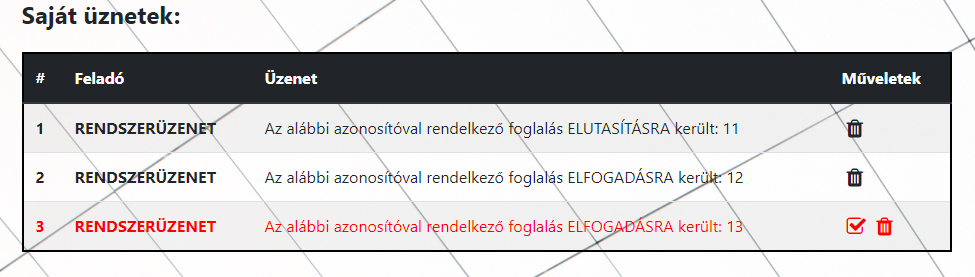
Abban az esetben, ha egy űrlap kitöltése már megkezdődött, de a felhasználó mégis megpróbál elnavigálni az oldalról, egy felugró ablak figyelmezteti arról, hogy a nem mentett adatok törlésre kerülnek.



19. ábra Oldal elhagyása figyelmeztetés

#### 2.3.2.3 Üzenetek

A felhasználónak lehetősége nyílik a saját üzeneti megtekintésére az „Üzenetek” menüpont alatt. A megjelenő táblázat minden sora egy-egy üzenetet jelöl. Az olvasatlan üzenetekhez tartozó sor piros, míg az olvasottakhoz tartozó fekete. Továbbá az oldal lehetőséget nyújt az üzenetek törlésére (az „Üzenete törlése” gomb segítségével), valamint az olvasatlan üzenetek státuszának állítására (a „Megjelölés olvasottként” gombra kattintva).



20. ábra Saját üzenetek

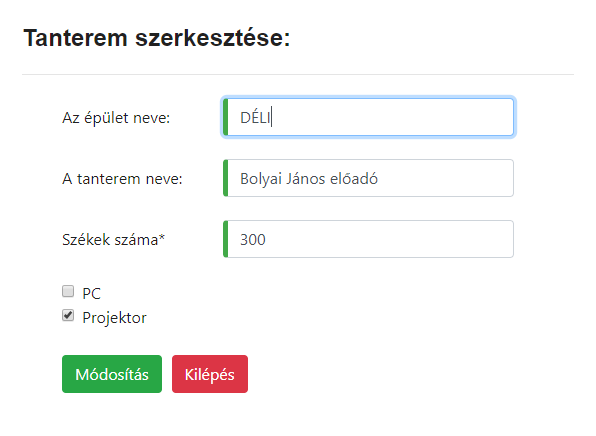
### 2.3.3 Adminisztrátorok számára elérhető felület

#### 2.3.3.1 Táblázatok

A „Táblázatok” menüpont alatt lehetősége nyílik az adminisztrátornak az adatbázisban található adatok táblázatos formában történő kilistázására. A legördülő menü az alábbi menüpontokat tartalmazza:

* Felhasználók
* Tantermek
* Félévek
* Tantárgyak
* Épületek
* Tantárgy foglalások
* Esemény foglalások

Minden egyes táblázat lehetővé teszi az adminisztrátor számára az adott sorhoz tartozó entitás szerkesztését. Ezt a „Műveletek” oszlopban a „Szerkesztés” gombra kattintva tehetjük meg. Ekkor egy, az adott entitáshoz tartozó, felugró ablak jelenik meg. A változtatásokat a „Módosítás” gombra kattintva lehet véglegesíteni.



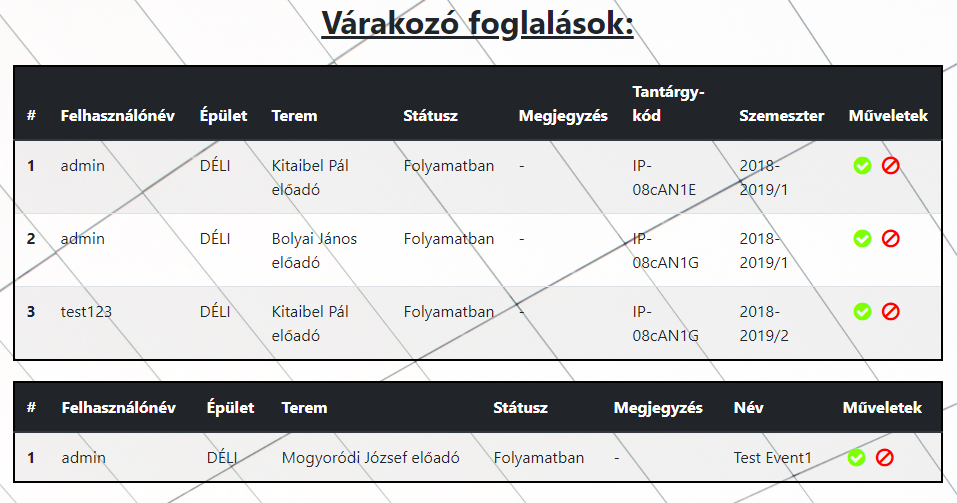
21. ábra A tanterem szerkesztéséhez tartozó felugró ablak

#### 2.3.3.2 Űrlapok

Az űrlapok menüpont bővült, újabb, kényelmesen használható funkciókat tartalmaz, lehetőséget nyújtva új szemeszter, felhasználó, tanterem, tantárgy vagy épület létrehozására, az érvényességre nagy hangsúlyt fektetve.

#### 2.3.3.3 Várakozó foglalások

A „Várakozó foglalások” menüpont célja, hogy az adminisztrátor könnyedén, grafikus felületen tudja kezelni az újonnan felvett foglalásokat. Az összes éppen „Folyamatban” státusszal rendelkező foglalás itt tekinthető meg. A foglalás státusza az „Elfogadás” gombra kattintva „Elfogadott” értékre módosul, az „Elutasítás” gomb hatására pedig „Elutasított” értéket vesz fel. Minden egyes módosítás azonnal frissül az adatbázisban, és mellékhatásként generál egy üzenetet a foglaló személy számára a státusz változásról.



22. ábra Várakozó foglalások

# 3. Üzemeltetői dokumentáció

## 3.1 Hardveres követelmények

A szoftvereknek megfelelő minimum és ajánlott hardveres követelmények az alábbi táblázatban tekinthetők meg:

|  |  |  |
| --- | --- | --- |
| Operációs rendszer | Minimum követelmény | Ajánlott követelmény |
| Windows x86 32-bit  Windows x86 64-bit | **Processzor:** 2 mag  **Memória:** 2 GB  **Merevlemez:** 10 GB szabad tárhely | **Processzor:** 4 mag vagy több  **Memória:** 8 GB vagy több  **Merevlemez:** 20 GB szabad tárhely |
| Linux x86 32-bit  Linux x86 64-bit | **Processzor:** 2 mag  **Memória:** 2 GB  **Merevlemez:** 10 GB szabad tárhely | **Processzor:** 4 mag vagy több  **Memória:** 8 GB vagy több  **Merevlemez:** 20 GB szabad tárhely |
| Mac OS X | **Processzor:** 2 mag  **Memória:** 2 GB  **Merevlemez:** 10 GB szabad tárhely | **Processzor:** 4 mag vagy több  **Memória:** 4 GB vagy több  **Merevlemez:** 20 GB szabad tárhely |
| Solaris x86 64-bit  Solaris Sparc 64-bit | **Processzor:** 2 mag  **Memória:** 2 GB  **Merevlemez:** 10 GB szabad tárhely | **Processzor:** 4 mag vagy több  **Memória:** 8 GB vagy több  **Merevlemez:** 20 GB szabad tárhely |

## 3.2 Szoftveres követelmények

Az alkalmazás tetszőleges környezetben üzemeltethető. A szükséges szoftverek listája a következő:

* Apache Maven 3.5.4+
* Apache Tomcat 8.0+
* Java 1.8+
* MySQL 8.0+
* NodeJs 8.12+ / Npm 6.4+

## 3.3 Telepítés

* Első lépésként indítsuk el a MySQL szervert.
* Az első indításkor szükség van az adatbázis létrehozására (ajánlott a „roomreservationdb” név használata). Abban az esetben, ha más adatbázis név kerül megadásra, akkor a rrs-server\src\main\resources\application.properties fájlban az adatforrás URL-t ennek megfelelően kell beállítani.
* Az alkalmazás egyéb fontos beállításainak módosítására szintén, a már előbb említett, application.properties fájlban van lehetőség.
* A gyökér könyvtárban található build.sh script futtatásával, vagy a mvn clean install parancs kiadásával tudjuk a backend és frontend fordítását elindítani.
* Ha ez megtörtént, és sikeres volt, akkor a rrs-server\deploy.sh futtatásával, vagy a mvn spring-boot:run parancs kiadásával tudjuk az alkalmazást elindítani az alapértelmezett 8080-as porton.
* Az alkalmazás első indítása létrehozza a megfelelő táblákat és a közöttük lévő kapcsolatokat.
* A mintaadatokkal történő feltöltéshez az rrs-server\src\main\resources\insert.sql fájl futtatása szükséges.

# 4. Fejlesztői dokumentáció

## 4.1 A fejlesztés menete

### 4.2.1 A fejlesztői környezet

#### 4.2.1.1 Backend

A backend fejlesztése során a tanulmányaim közben már megismert NetBeans-re esett a választásom. A NetBeans egy integrált fejlesztői környezet, ami a Java nyelven alapul. A program grafikus fejlesztőfelületet kínál a különböző alkalmazások, Appletek vagy akár JavaBeanek elkészítéséhez, amelynek segítségével könnyebben, gyorsabban tudjuk fejleszteni saját programjainkat. Megszámlálhatatlan kényelmi funkcióval is rendelkezik, melyek többek közt a kódgenerálás, automatikus formázás és a kódkiegészítés.

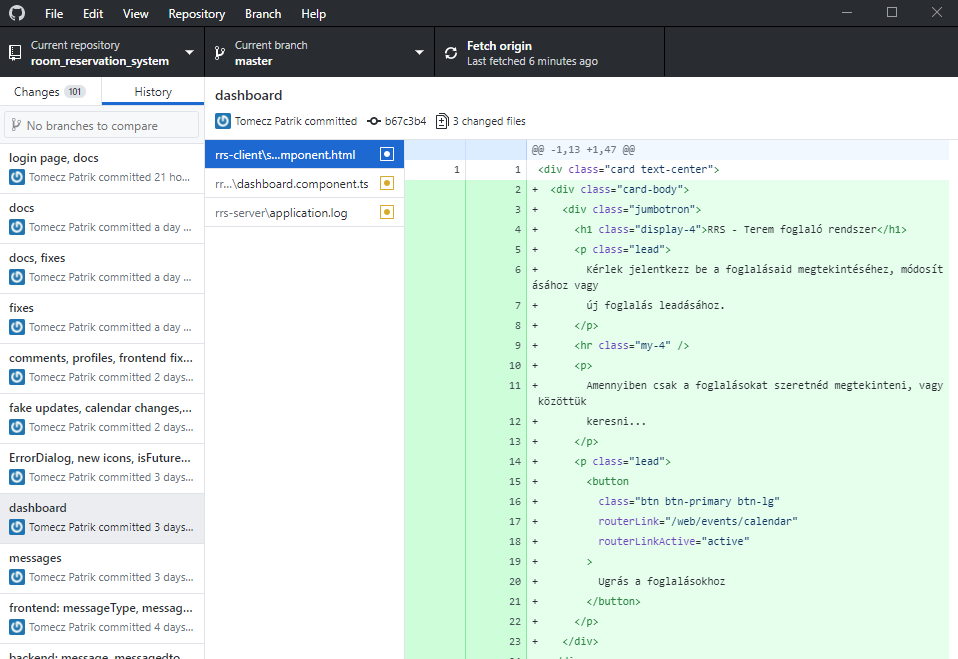
#### 4.2.1.2 Frontend

A megjelenítési réteg kódolása során a Visual Studio Code szerkesztő mellett döntöttem. A Visual Studio Code, melyet gyakran csak VS Code-nak rövidítenek, egy ingyenes, nyílt forráskódú kódszerkesztő, melyet a Microsoft fejleszt Windows, Linux és OS X operációs rendszerekhez. Támogatja a hibakeresőket, valamint beépített Git támogatással rendelkezik, továbbá képes az intelligens kódkezelésre az IntelliSense segítségével. Ezen felül testre szabható, így a felhasználók megváltoztathatják a kinézetet, megváltoztathatják a szerkesztő gyorsbillentyű-kiosztását, az alapértelmezett beállításokat és még sok egyebet. A program megnyitása után a kiegészítők menüpont alatt lehetőség nyílik saját kiegészítők telepítésére, ki- és bekapcsolására.

### 4.2.2 Verziókezelés

A fejlesztés kezdetétől a program forráskódját a [www.github.com](http://www.github.com) oldalon tároltam. A célom az volt, hogy bármilyen, a forráskódban történt módosítás visszakereshető, ellenőrizhető, visszaállítható legyen. A gitnek köszönhetően követni tudtam minden egyes változást, a projektet több számítógépről is el tudtam érni, valamint mindig volt egy biztonsági másolatom, ha esetlegesen valamely hardver meghibásodott volna.

Kezdetben a gitet parancssori felületen keresztül használtam, majd kis idő elteltével áttértem a GitHub Desktop alkalmazásra. Ezen program segítségével, grafikus felületen könnyedén végrehajtható volt minden, amit



23. ábra A GitHub Desktop alkalmazás

## 4.3 A felhasznált technológiák

|  |  |
| --- | --- |
| Technológia | Leírás/Indoklás |
| Java Spring Boot | A Spring egy nyílt forráskódú, inversion of controlt megvalósító Java alkalmazás keretrendszer.  A kontroll megfordítása (angolul inversion of control, röviden IoC) főleg objektumorientált programozási nyelvekben használt technika a komponensek összeillesztésére, konfigurálására és kezelésére. A technika lényege, hogy a komponenskezelést (pl. létrehozást, példányosítást, paraméterezést, megszüntetést, metódus hívás) kiemeljük a programkódból, és általában egy külső keretrendszerre bízzuk, mint pl. a Spring. Célja a modularitás növelése és bővíthetővé tétele.  A Spring keretrendszer magját képező szolgáltatásokat főként Java alkalmazás fejlesztésére használják a programozók. Ugyanakkor a Java EE platformra is elérhetők a bővítményei, amelyek web-alkalmazás fejlesztését segítik elő. |
| Maven | Az Apache Maven (röviden Maven) egy szoftver, amelyet szoftverprojektek menedzselésére és a build folyamat automatizálására lehet használni.  A Maven bevezeti a POM, azaz a Projekt Objektummodell (angolul: Project Object Model) fogalmát. Egy POM egy buildelendő projektet ír le és annak függőségeit. Az egyes lépéseket céloknak, angolul goal-oknak nevezik. Vannak előre definiált célok a tipikus feladatokra, mint például a kód fordítása és csomagolása, de a felhasználónak lehetősége van saját célokat is definiálni a projektspecifikus lépések végrehajtására.  A Maven hálózatképes, tehát szükség esetén dinamikusan is le tud tölteni komponenseket. Repository névvel illetik a különböző hosztok fájlrendszereinek azon mappáit, ahol a letölthető komponensek találhatók. A Maven nem csak a repository-kból való letöltést támogatja, hanem a készült szoftvercsomag feltöltését is. |
| Hibernate | A Hibernate egy objektum-relációs leképezést (ORM) megvalósító programkönyvtár Java platformra. Segítségével osztályokat és a relációs adatbázisok tábláit tudjuk egymásba leképezni, az adatbázisban lévő rekordokat objektumokként kezelhetjük, és az objektumainkat egyszerűen tárolhatjuk állapotmegőrző módon adattáblákban.  A Hibernate használható önálló Java alkalmazásokban is, ugyanakkor tipikusan Java EE környezetben alkalmazzák. |
| JPA | A Java Persistence API, vagy röviden JPA, egy keretrendszer a Java programozási nyelvhez, melynek fő feladata a relációs adatok kezelése a Java-t használó alkalmazásokban. |
| MySQL | Az adatbázis motor szerepére a MySQL-t választottam, mivel ez az egyik legelterjedtebb adatbázis-kezelő napjainkban. Telepítése könnyű, Javas környezetben könnyen használható a MySQL Conncetor segítségével. Az adatok vizsgálatára, módosítására és ellenőrzésére biztosít egy MySQL Workbench névvel ellátott programot. |
| Angular 6 | Az Angular egy TypeScript alapú, nyílt forráskódú frontend keretrendszer, melyet a Google Angular csapata fejleszt. Segítségével egyszerű módon bármilyen komplex webalaklamazás létrehozása lehetséges. |
| Bootstrap 4 | A Bootstrap egy ingyenes, nyílt forráskódú frontend keretrendszer, melyet weboldalak, web alkalmazások tervezésére használnak. Megannyi előre megírt HTML és CSS alapú komponenst tartalmaz, többek között űrlapokat, gombokat, navigációs elemeket, táblázatokat és csúszkákat. |

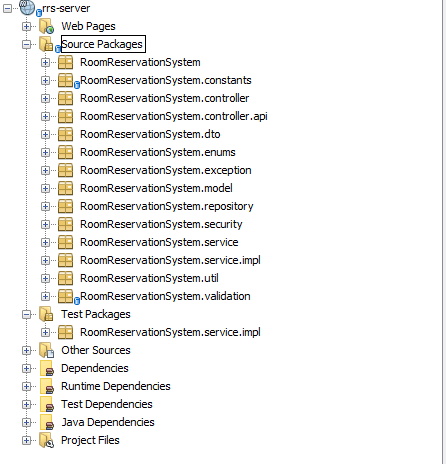
## 4.4 Az alkalmazás architektúrája

### 4.4.1 A backend

#### 4.4.1.1 A szerver felépítése

A fontosabb csomagok és leírásuk:

* constants: A konstansokat tartalmazó csomag
* controller / controller.api: Az entitásokhoz tartozó vezérlő osztályok. A vezérlő dolgozza fel a felhasználói adatbevitelt. Függvényhívásokká képezi le azokat. Ezek fogják előidézni az adatok módosítását, törlését, vagy a nézetek megváltozását.
* dto: Az adatátviteli (DTO, azaz Data Transfer Object) osztályokat tartalmazó csomag.
* enums: A felsorolási típusokat tartalmazó csomag
* exception: A kivételeket tartalmazó csomag
* model: Az entitásokat tartalmazó csomag
* repository:



24. ábra A backend szerkezete

### 4.4.2 A frontend

## 4.1 Tesztelés

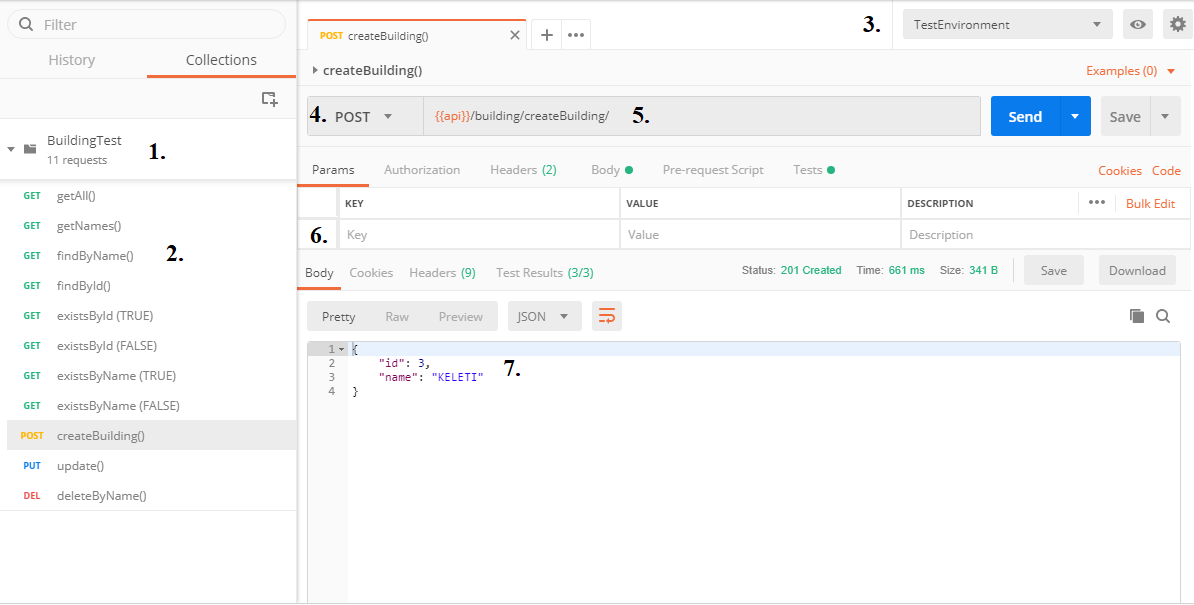
### 4.1.1 Backend

A backend tesztelése során külön kitértem a szerviz és kontroller osztályokra.

A szerviz osztályok tesztelését JUnit és Mockito keretrendszerek segítségével valósítottam meg. Ezek a tesztek automaták, minden egyes build esetén lefutnak. A tesztek felépítése általában azonos. Az osztályok elején találhatóak a Mockito számára fontos adattagok a megfelelő annotációkkal ellátva, ezeket követik a tesztekhez szükséges (osztályszintű) segédváltozók, majd végül az egyes függvények tesztjei. Egy függvényhez legalább egy, de akár több teszt is tartozhat a tartalmától, kivételektől függően.

A végpontok tesztelését a Postman nevezetű alkalmazás segítségével valósítottam meg. Azért esett erre az alkalmazásra a választásom, mert kezelőbarát felhasználói felülettel rendelkezik, automatizálható a tesztelés és széleskörű lehetőségeket biztosít.

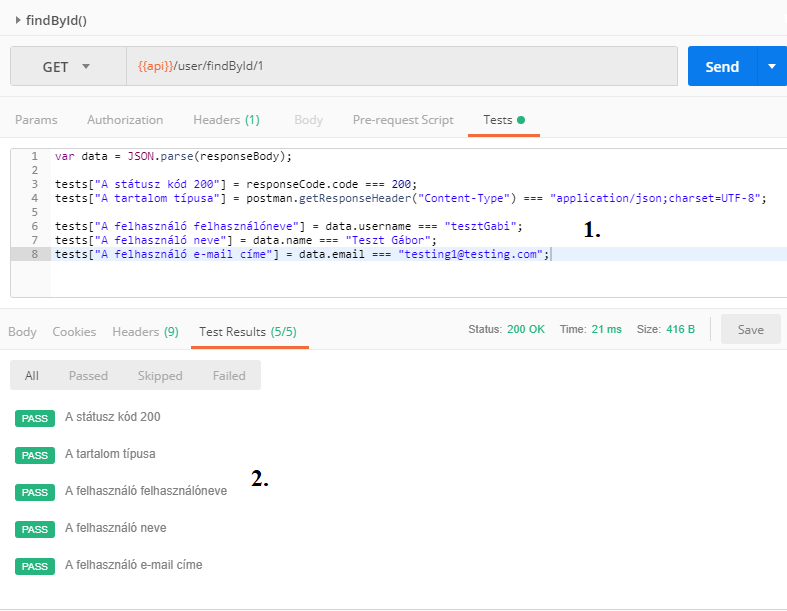
Az alkalmazás használata:



25. ábra A Postman használata

1. A megfelelő gyűjtemény kiválasztása. Ezen gyűjteményekbe szolgálnak a kérések csoportosítására, könnyen importálhatók, exportálhatok.
2. A gyűjteményen belül kiválasztjuk a megfelelő kérést.
3. Annak érdekében, hogy a kéréseket, tesztelést könnyebben, dinamikusabban tudjuk végrehajtani a Postman-ben használhatunk változókat, amelyeket a környezetekben definiálhatunk. A környezet minden kérés futtatása előtt kiválasztható.
4. A kérés típusának kiválasztása
5. A kéréshez tartozó URL megadása
6. A kérés paramétereinek megadása (opcionális). A paraméterek kulcs-érték párok, ezek segítségével generálódik a megfelelő URL.
7. A kérés elküldése
8. A válasz, melynek formátuma beállítható (pl.: JSON)

Tesztelés a Postmanben:



26. ábra Tesztelés a Postmanben

1. A Postman segítségével Javascript nyelven írt teszteket tudunk készíteni. A tesztek során nem csak a válasz objektum attribútumainak vizsgálatát tudjuk végrehajtani, hanem akár a válasz státuszát, tartalmának típusát is ellenőrizhetjük.
2. A kérés futtatása után a definiált tesztek automatikusan lefutnak és grafikusan is megjelennek az eredmények.

A tesztek futtatásának több módja is van:

1. A tesztek egyéni futtatása a Postman alkalmazásból
2. Egy adott gyűjtemény futtatása a Postman alkalmazásból. Ebben az esetben minden egyes a gyűjteményben található kérés és a hozzájuk tartozó tesztek is lefutnak.
3. Script segítségével történő futtatás
   1. Az eredmények szöveges formátumban történő mentéséhez a „text\_tests.sh” script futtatása szükséges
   2. Ha HTML formátumban szeretnénk a teszteket megkapni, akkor a „html\_test.sh” scriptre van szükségünk

Fontos, hogy a tesztek nem csak a kérések sikeres végrehajtása, a státuszkódot, hanem a válasz törzsét is ellenőrzik. Ebből kifolyólag a tesztesetek akkor lehetnek sikeresek, ha azok a minta adatbázis állapotra futnak le. Ezt az állapotot a „TEST\_DB.sql” fájl futtatásával érhetjük el.

A tesztek eredményei a mintaadatbázison HTML formátumban:

# 5. Irodalomjegyzék

* <https://hu.wikipedia.org/wiki/NetBeans>
* <https://hu.wikipedia.org/wiki/Visual_Studio_Code>
* https://hu.wikipedia.org/wiki/Spring\_keretrendszer
* <https://hu.wikipedia.org/wiki/A_kontroll_megford%C3%ADt%C3%A1sa>
* <https://hu.wikipedia.org/wiki/Apache_Maven>
* <https://hu.wikipedia.org/wiki/Hibernate>
* <https://hu.wikipedia.org/wiki/Java_Persistence_API>
* <https://hu.wikipedia.org/wiki/MySQL>
* <https://en.wikipedia.org/wiki/Angular_(application_platform)#Version_6>
* https://en.wikipedia.org/wiki/Bootstrap\_(front-end\_framework)