



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

**PROGRAMOZÁSI NYELVEK ÉS
FORDÍTÓPROGRAMOK TANSZÉK**

Mozijegyek foglalását kezelő rendszer

Témavezető:

Mészáros Mónika

tanársegéd

Szerző:

Fischer Zsolt

Programtervező informatikus BSc

Budapest, 2018.

Tartalomjegyzék

1	Bevezetés	3
2	Felhasználói dokumentáció.....	4
2.1	Szükséges telepítések	4
2.1.1	Felhasználóként.....	4
2.1.1.1	JRE.....	4
2.1.2	Adminisztrátorként	7
2.1.2.1	JRE.....	7
2.1.2.2	MySQL	7
2.1.2.3	Adatbázis létrehozása.....	10
2.2	Program futtatása	11
2.2.1	Adminisztrátorként	11
2.2.2	Felhasználóként.....	12
2.3	Adminisztrátori funkciók	12
2.3.1	Filmek és előadások	13
2.3.1.1	Új film felvétele	13
2.3.1.2	Új terem felvétele.....	15
2.3.1.3	Új előadás kiírása	16
2.3.2	Keresések/törlések	17
2.3.2.1	Foglalások keresése/törlése.....	17
2.3.2.2	Filmek keresése/törlése	18
2.3.2.3	Felhasználók keresése/törlése	19
2.3.3	Felhasználói vélemények	20
2.3.4	Saját adatok módosítása	21
2.3.5	Kijelentkezés.....	22
2.4	Felhasználói funkciók	23
2.4.1	Regisztráció.....	23
2.4.2	Új foglalás indítása	26
2.4.3	Keresések	30
2.4.3.1	Keresés saját foglalások között.....	30
2.4.3.2	Keresés filmek között	32
2.4.4	Új felhasználói vélemény.....	33
2.4.5	Felhasználói vélemények	34
2.4.6	Saját adatok módosítása	35
2.4.7	Kijelentkezés.....	36
3	Fejlesztői dokumentáció	37

3.1	Feladat leírása	37
3.2	Fejlesztéshez szükséges telepítések	38
3.2.1	Fejlesztő környezet	38
3.2.2	JDK	38
3.2.3	MySQL	38
3.2.4	Felhasznált modulok	38
3.3	Megoldási terv	39
3.3.1	Adatbázis.....	39
3.3.2	Use case diagram	43
3.3.3	Megvalósítandó funkciók.....	43
3.4	Fájlszerkezet	46
3.4.1	Fájlok csoportosítása.....	46
3.5	Tesztelés.....	58
3.5.1	Funkcionális tesztek.....	58
3.5.2	Távoli felhasználás tesztelése	73
3.5.3	Párhuzamos felhasználás tesztelése	73
3.5.4	Unit tesztek	75
4	Továbbfejlesztési lehetőségek	85
5	Összegzés.....	86
6	Irodalomjegyzék	87

1 Bevezetés

Szakdolgozatom témája egy adott mozi adatait és foglalásait kezelő asztali alkalmazás, amely segítségével a felhasználók kényelmesen és könnyedén foglalhatnak mozijegyet kedvenc filmjeikre otthonról, vagy csak nézelődhetnek a filmkínálat között.

Az alkalmazás funkcióinak igénybe vételéhez minden esetben regisztrációra van szükség, hiszen csak bejelentkezést követően válnak elérhetővé számunkra a menüpontok.

Bejelentkezni adminisztrátorként és felhasználóként is be lehet, viszont csak felhasználóként van lehetőségünk regisztrálni, de ettől függetlenül a rendszerbe természetesen több adminisztrátort is fel tudunk venni.

Adminisztrátorként a bejelentkezést követően tudunk új filmet és termeket felvenni különböző adatok és borítókép megadásával. A felvett filmekhez lehetőség van előadás kiírására az előadás időpontjának és a terem számának megadásával.

Lehetőség van keresni filmek, felhasználók és foglalások között különböző szempontok szerint. Ezeket a foglalásokat, filmeket, felhasználókat táblázatos formában is megtekinthetjük, ill. a foglalásokat PDF fájlba exportálhatjuk.

A már nem vetített filmeket, lemondott foglalásokat és szabályszegő felhasználókat törölhetjük is.

Továbbá, a felhasználói vélemények olvasására is lehetőség van, ill. frissíthetjük az email címünket/jelszavunkat is a megfelelő oldalon.

Felhasználóként a bejelentkezés után új foglalást indíthatunk az előadás kiválasztásával és a személyek számának megadásával. Ha kiválasztottuk a megfelelő előadást, akkor az ülőhelyek kiválasztása következik.

Lehetőségünk van keresni filmek és saját foglalások között több szempont alapján. A keresett foglalást törölhetjük is, ill. nyomtatható dokumentummá exportálhatjuk.

Filmek keresésekor véleményezhetjük és pontozhatjuk is a filmeket tetszésünk szerint.

Saját adatlap megtekintésekor, bizonyos adatok módosítását is lehetővé teszi az alkalmazás.

Emellett felhasználóként is böngészhetünk a mások által írt értékelések között a tökéletes film kiválasztásának érdekében.

2 Felhasználói dokumentáció

2.1 Szükséges telepítések

2.1.1 Felhasználóként

2.1.1.1 JRE

A program használatához szükségünk van a legalább 1.8-as verziójú JRE-re (*Java Runtime Environment*). (64 bites Windows 10 operációs rendszert feltételezve a továbbiakban).

A JRE a Java nyelven írt alkalmazások futtatásához szükséges.

Könnyedén elképzelhető, hogy a számítógépünkre már fel van telepítve a Java, ugyanis nagyon sok alkalmazás használatához szükséges. Ezt úgy tudjuk leellenőrizni, hogy a képernyő bal alsó sarkában levő kereső mező megnyitása után begépeljük, hogy „cmd”, megnyitjuk, majd beírjuk, hogy „java -version”, s ENTER-t nyomunk.

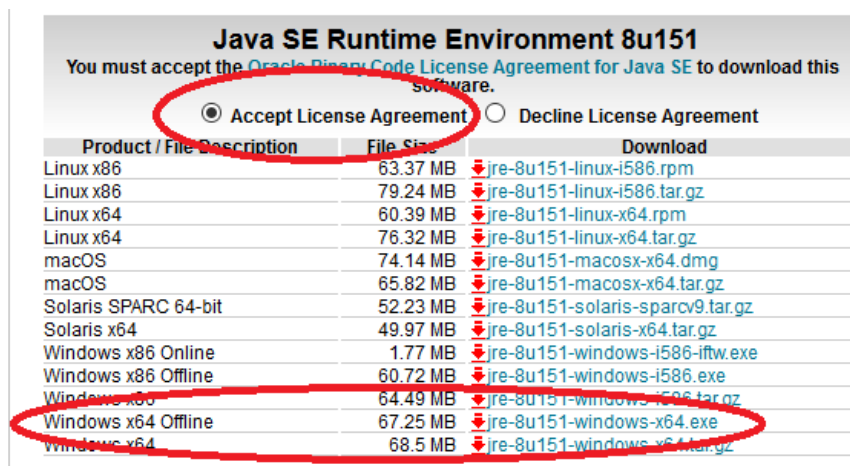
Ha a java nincs telepítve, akkor a következő üzenet fogad minket:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>java -version
'java' is not recognized as an internal or external command,
operable program or batch file.
```

1. ábra

Ebben az esetben töltjük le a megfelelő fájlt a hivatalos oldalról¹.



2. ábra

Letöltést követően nyissuk meg a letöltött *jre-8u151-windows-x64.exe* fájlt, majd a telepítési utasításokat követve telepítjük fel a JRE-t.

¹ <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

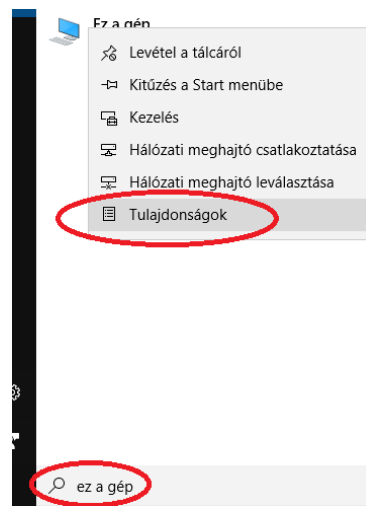
Feltelepítés után a környezeti változókat kell beállítanunk.

Ha a telepítés során nem változtattuk meg a telepítés helyét, akkor alapértelmezetten a Windows a *C:\Program Files\Java\jre1.8.0_151* helyre telepítette.

Ezt az elérési útvonalat kell megadnunk a következőképp:

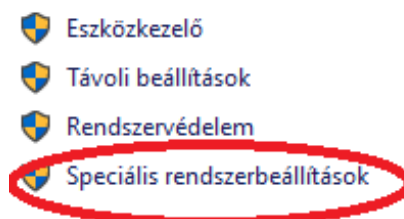
1. A képernyő bal alsó sarkában kattintsunk a „nagyító” jelre.

Gépeljük be, hogy „Ez a gép”. Ezt követően jobb gombbal kattintsunk az „Ez a gép”-re, majd válasszuk ki azt, hogy „Tulajdonságok”.



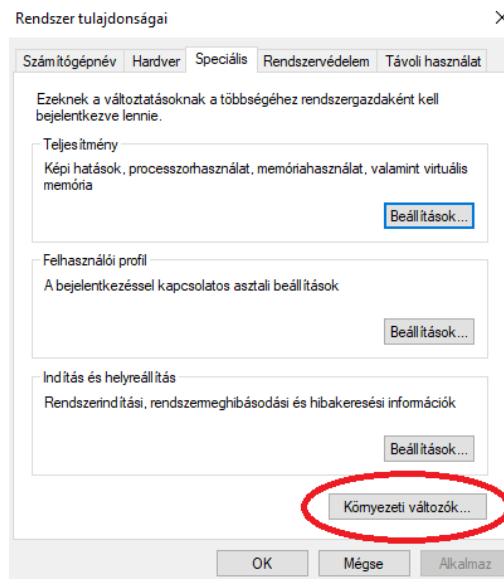
3.ábra

2. A megnyitott ablak bal felső részén látható menüpontok közül válasszuk ki a „Speciális rendszerbeállítások”-at.



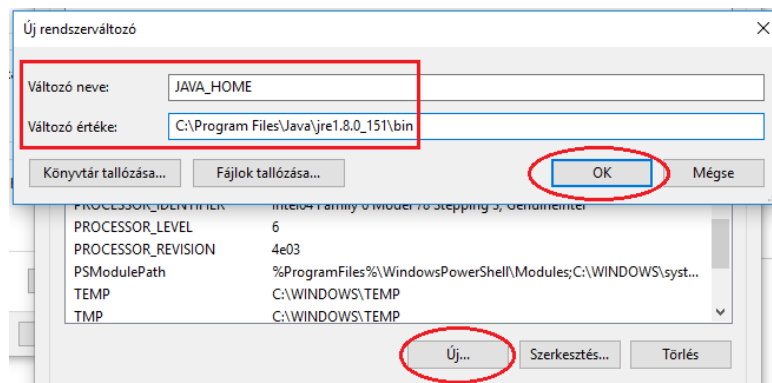
4.ábra

3. Ez követően nyissuk meg az ablak alján található „Környezeti változók”-at.



5. ábra

4. Ezután a „Rendszerváltozók” részben hozzuk létre a JAVA_HOME változót (ha még nincs) az „Új” gombbal. A változó nevének adjuk meg, hogy JAVA_HOME, a változó értékének pedig a JRE elérési útvonalát (jelen esetben *C:\Program Files\Java\jre1.8.0_151\bin*), majd mentjük el.



6. ábra

5. Legvégül pedig ellenőrizzük le a feltelepült java-t.

Ehhez a képernyő bal alsó sarkában nyissuk meg a kereső mezőt és írjuk be, hogy „cmd”, majd nyissuk meg.

A felugró ablakba gépeljük be, hogy „java -version”, majd nyomjunk ENTER-t.

Ekkor sikeres telepítés és beállítások esetén a következőt kell látnunk:

```
C:\Users\Asus>java -version
java version "1.8.0_151"
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) Client VM (build 25.151-b12, mixed mode)
```

7. ábra

2.1.2 Adminisztrátorként

2.1.2.1 JRE

Ugyanúgy szükségünk van az JRE-re a program használatához adminisztrátorként, mint felhasználóként. A letöltésről és telepítésről az előző (2.1.1.1 JRE) részben részletesen olvashatunk.

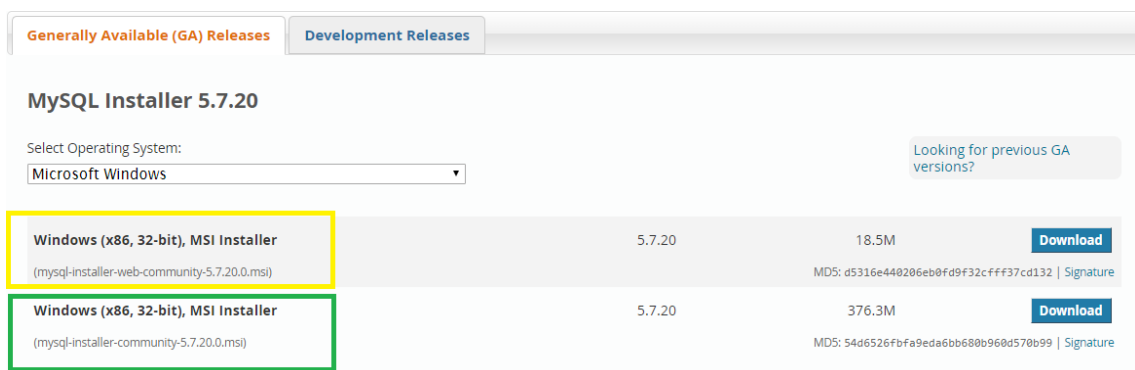
2.1.2.2 MySQL

Az adatainkat MySQL adatbázisban fogjuk tárolni, ehhez pedig szükségünk lesz a MySQL-re.

Első lépésben töltsük le az Installert a hivatalos oldalról².

Ha rendelkezünk internetkapcsolattal miközben telepítjük a MySQL-t, akkor a sárga kereteset, ha nem rendelkezünk, akkor a zöld kereteset töltsük le.

A MySQL Installer 32 bites, de fel fogja telepíteni a 64 biteset verziót is.



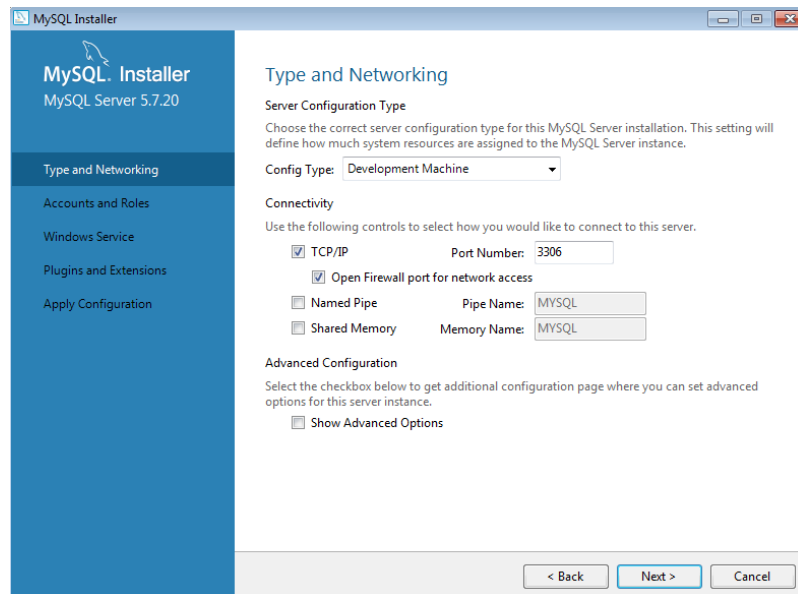
8.ábra

Letöltés után nyissuk meg a letöltött fájlt, s kezdjük meg a telepítést.

² <https://dev.mysql.com/downloads/windows/installer/5.7.html>

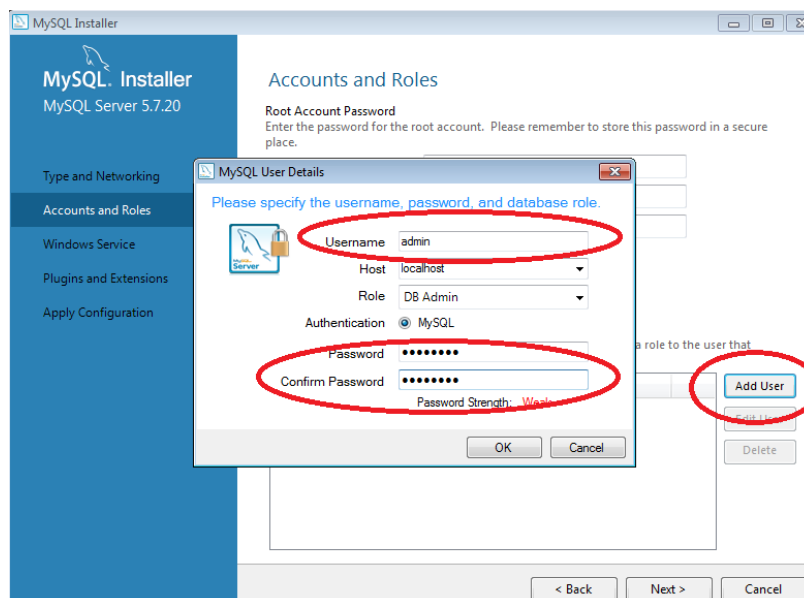
A legtöbb beállítást hagyjuk alapértelmezetten, viszont van pár kérdéses rész, amelyeknek a beállításáról a következő képek szolgálnak segítségül:

Az alap Port szám a **3306**, ezt nyugodtan hagyjuk ezen.

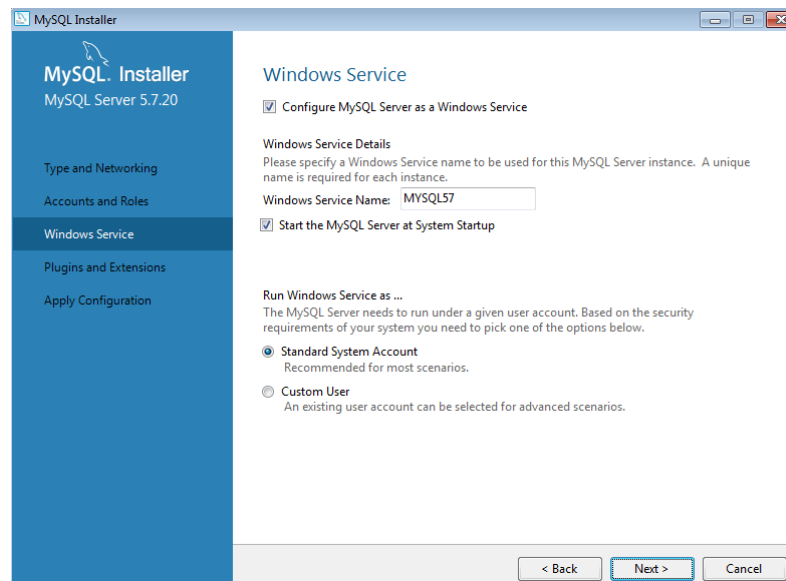


9.ábra

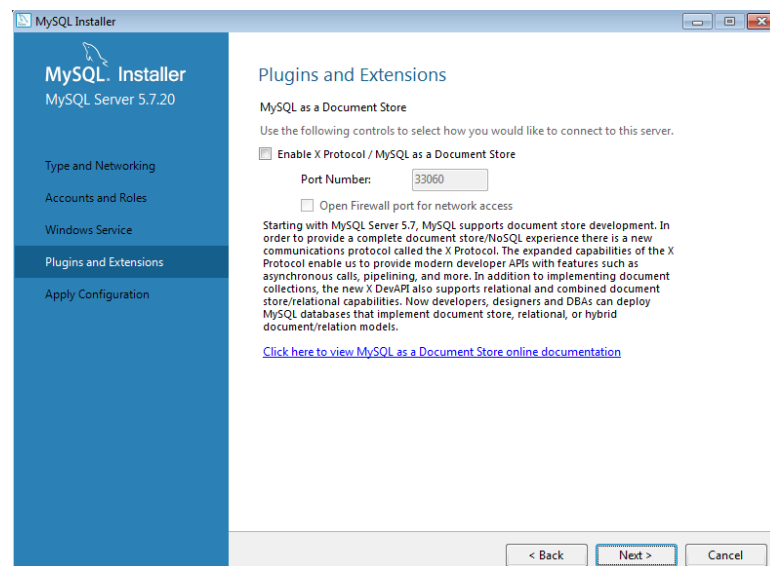
A következő lépésben adjunk hozzá egy új felhasználót (*Add User*), például „**admin**” névvel és „**password**” jelszóval. A Role(szerep): **DB Admin** vagy annál magasabb legyen. Erre később szükségünk lesz!



10.ábra

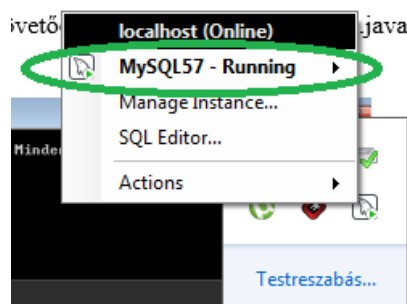


11. ábra



12. ábra

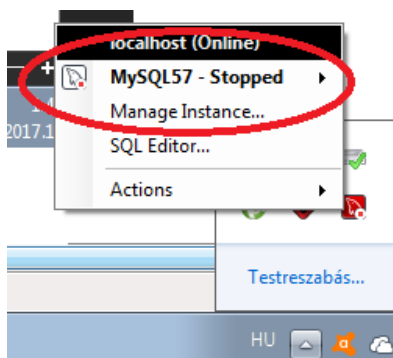
Telepítést követően a jobb alsó sarokban láthatjuk a szerver állapotát.
Running(futó) állapotban legyen!



13. ábra

A MySQL-t nem kell minden indításkor külön elindítani, hiszen az a számítógéppel együtt fog elindulni.

Azonban, ha valamilyen oknál fogva, a MySQL szerver nem futna, tehát a következőt látnánk az előző képen látottak helyett:



14.ábra

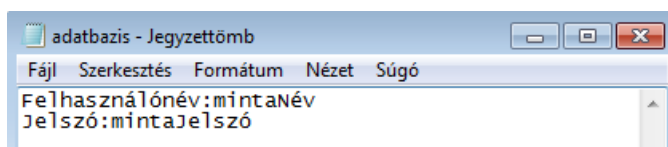
Ekkor csupán annyit kell tennünk, hogy a „*Stopped*” (pirossal bekarikázott) sor végén látható → jelre kattintunk, és kiválasztjuk, hogy „*Start*”. Ezt követően a szerver ismét használatra kész (fut).

2.1.2.3 Adatbázis létrehozása

A program **legelső** használata előtt (tehát nem kell ezt minden alkalommal megtennünk, amikor a gépet újraindítjuk) létre kell hoznunk az adatbázist, amelyből a későbbiekben dolgozni fogunk.

Első lépésben írjuk át a *Database\adatbazis.txt* fájl sorait, ha szükséges (Felhasználónév, Jelszó).

Minta:

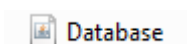


15.ábra

Az első (Felhasználó) sorban a „Felhasználó:” után írjuk be a korábban (vagy még régebben) létrehozott MySQL felhasználónevet, a másodikban pedig a hozzá megadott jelszót a „Jelszó:” után. Mentsük el a változtatásokat!

Az adatbázis mindig a saját gépünkön fog létrejönni, de később egy másik gépről is könnyedén elérhetjük azt.

Ezután hozzuk létre az adatbázisunkat a *Database\Database.jar* fájl futtatásával.



Ekkor létrejönnek az üres táblák a megfelelő oszlopokkal.

A „*Kategóriák* (genres)” és a „*Korhatárok* (ages)” táblák az adatbázis létrejöttével együtt feltöltődnek a megfelelő adatokkal (pl.: Akció, Horror stb. – 12, 16 stb.).

Ezen felül, az adatbázissal együtt egy *adminisztrátor* is létrejön „**admin**” felhasználónévvel és „**asd**” jelszóval, melyeket később kedvünk szerint módosíthatunk.

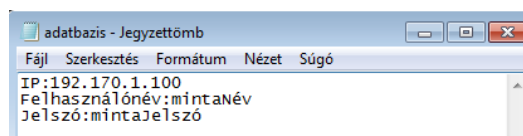
2.2 Program futtatása

2.2.1 Adminisztrátorként

A program **legelső** használata előtt a *Cinema\adatbazis.txt* fájl tartalmát írjuk át az operátortól kapott adatokra, amelynek tartalmaznia kell az adatbázist futtató gép ip címét, illetve az adatbáziscsatlakozáshoz szükséges felhasználónevet és jelszót.

Értelemszerűen az első (IP) sorba a „:” után írjuk az ip-t, a következő két sorba (Felhasználónév, Jelszó) pedig a felhasználónevet és jelszót, szintén a „:” után.

Minta:

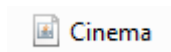


16.ábra

Abban az esetben, ha a használni kívánt adatbázis a saját gépünkön van, akkor az IP sorba írjuk azt, hogy „localhost”, a következő két sorba (Felhasználónév, Jelszó) pedig a korábban létrehozott MySQL fiókhoz tartozó felhasználónevet és jelszót a „:” után.

Ne felejtsük el elmenteni a változtatásokat!

Ezt követően a *Cinema\Cinema.jar* fájl futtatásával tudjuk megkezdeni az alkalmazás használatát, s a bejelentkező felület kell, hogy fogadjon minket.



Bejelentkező oldal:



17.ábra

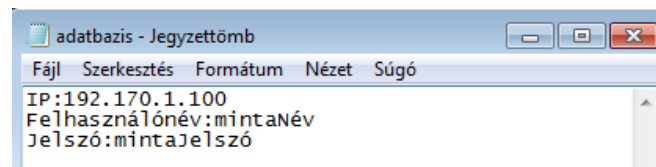
2.2.2 Felhasználóként

Első lépésben az alkalmazás **legelső** használata előtt az adatbáziscsatlakozáshoz szükséges adatokat kell megadnunk.

Ehhez nyissuk meg a *Cinema\adatbazis.txt* fájlt, és ha szükséges, írjuk át a fájl sorait (IP, Felhasználónév és Jelszó) az adminisztrátortól megkapott adatokra (minden sorban a „;” után kell beírni a megfelelő adatot).

Mentsük el a változtatásokat!

Minta:



18.ábra

Végül indítsuk el a programot a *Cinema\Cinema.jar* fájl futtatásával.



Az adatbázis információk helyes megadását követően megkezdhetjük az alkalmazás használatát.

2.3 Adminisztrátori funkciók

Amint már korábban is említettem, adminisztrátor regisztrációjára nincsen lehetőség, csakis felhasználóéra. Minden adminisztrátori funkció csak bejelentkezés után érhető el!

Adminisztrátorként lehetőségünk van:

- új film felvételére
- új terem felvételére
- új előadás kiírására
- foglalások/filmek/felhasználók közötti keresésre különböző szempontok alapján
- foglalások/filmek/felhasználók törlésére, ill. foglalások exportálására PDF fájlba
- felhasználói vélemények olvasására filmek szerint
- saját adatok módosítására

2.3.1 Filmek és előadások

2.3.1.1 Új film felvétele

Adminisztrátorként lehetőségünk van új filmet felvenni a rendszerünkbe.

19.ábra

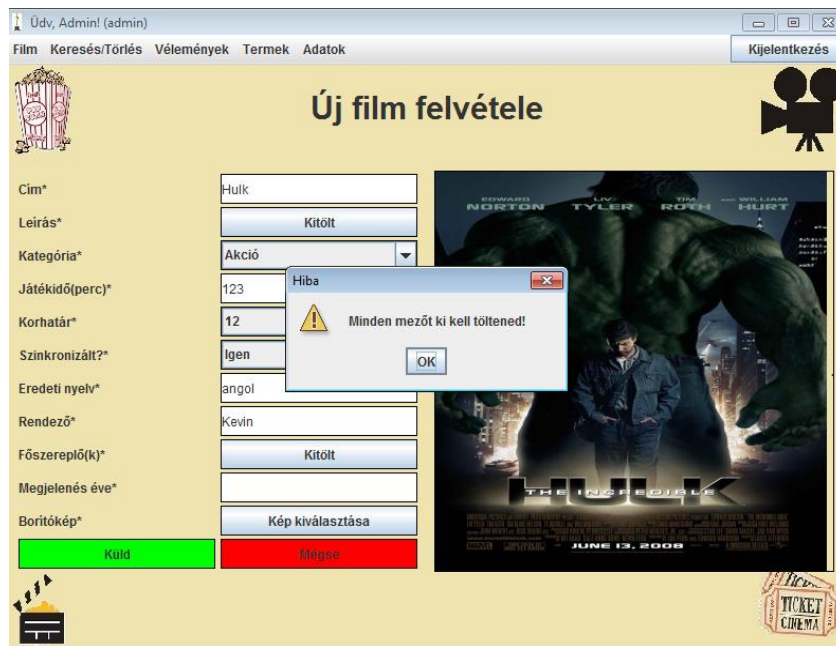
Minden mező kitöltése kötelező, ill. a *Játékidő* (értéke max. 250) és *Megjelenés éve* (nem lehet nagyobb, mint az aktuális év) mezőket csak számokkal tölthetjük ki. A kép mérete nem lehet nagyobb egy adott értéknél.

Vannak további megkötések a mezők hosszaira.

A *Cím* max. 30 karakter, a *Leírás* max. 400 karakter, a *Rendező* max. 50 karakter, a *Főszereplők* max. 100 karakter, az *Eredeti nyelv* pedig max. 15 karakter hosszú lehet.

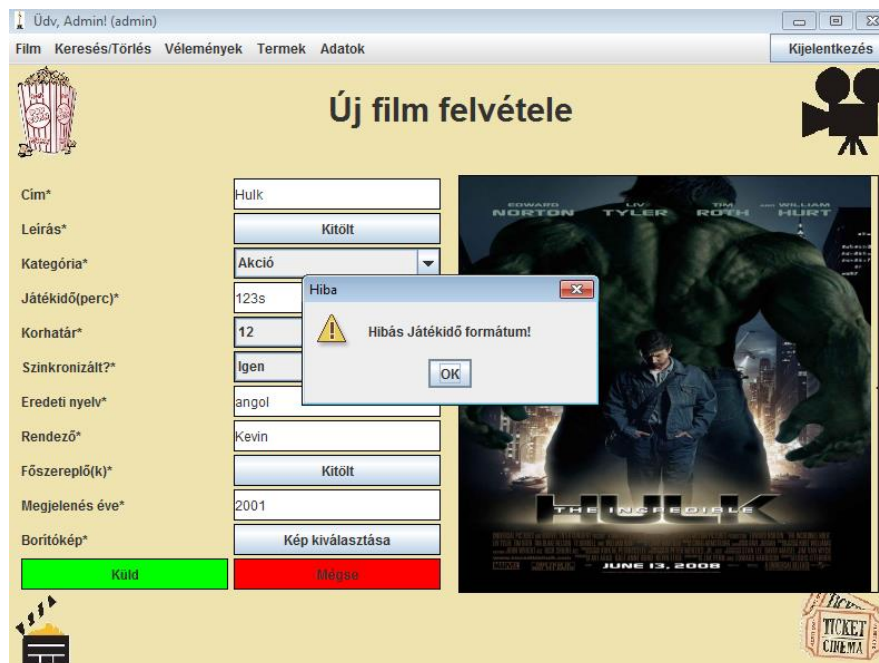
Ha ezen feltételek közül bármelyik nem teljesül, a programunk hiba üzenetet fog dobni, ellenkező esetben a film felvétele sikeres volt, melyet egy üzenettel erősít meg a programunk.

Ha nincs minden mező kitöltve:



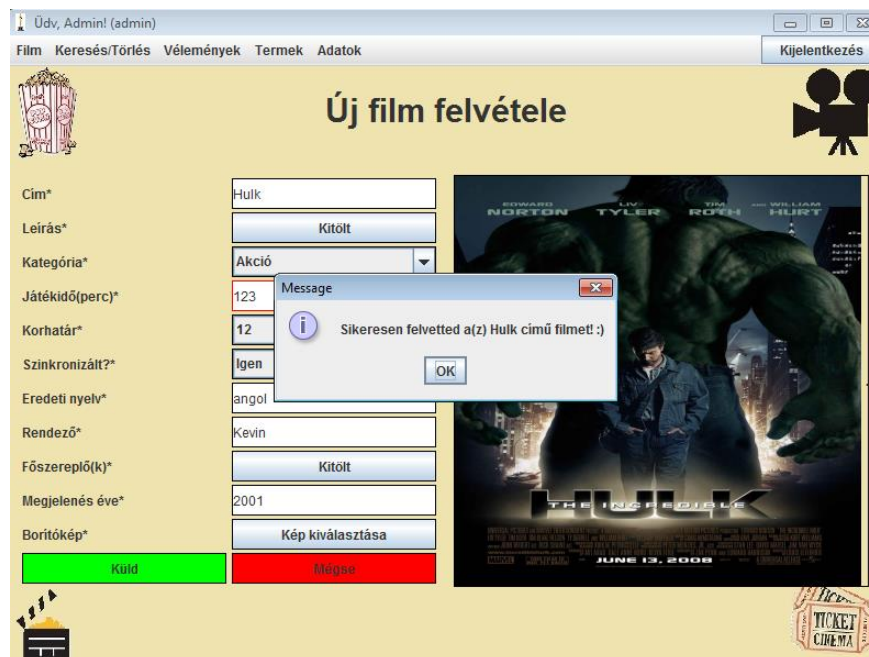
20.ábra

Ha a *Játékidő* vagy *Megjelenés éve* mezők valamelyike nem megfelelő formátumban lett megadva (az OK gombra kattintás után, a hibás mező kerete pirosra változik):



21.ábra

Sikeres filmfelvétel esetén:



22.ábra

2.3.1.2 Új terem felvétele

Ahhoz, hogy előadást tudjunk kiírni, szükségünk van termekre (legalább egyre).



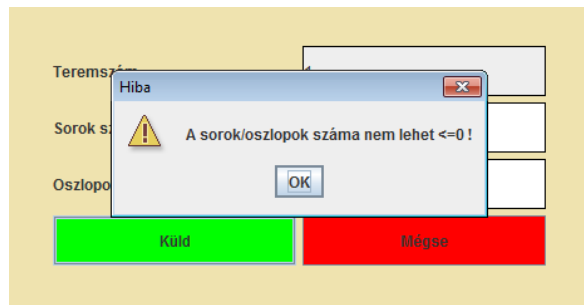
23.ábra

A *Teremszám* értéke mindig a soron következő terem száma.

A *sorok/oszlopok számának* megadása kötelező.

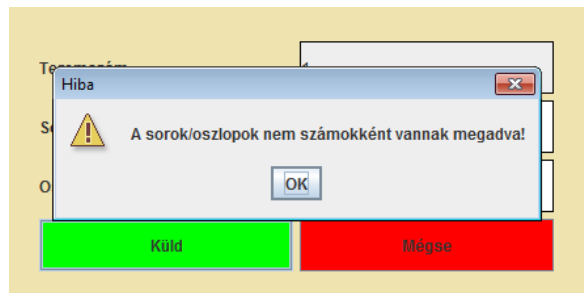
A *sorok/oszlopok* formátuma szám kell, hogy legyen, ill. értékük legalább 1 és legfeljebb 50 lehet. Ellenkező esetben hibaüzenetet kapunk.

Hibás érték:



24.ábra

Hibás formátum:



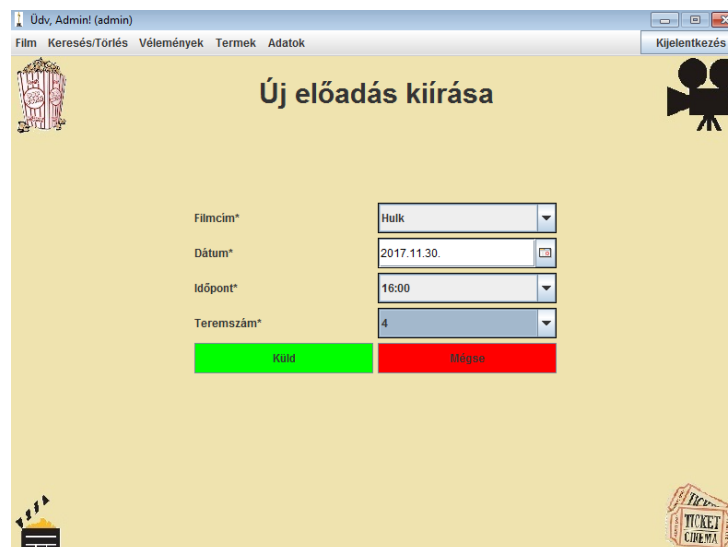
25.ábra

2.3.1.3 Új előadás kiírása

Miután sikeresen felvettünk legalább egy filmet és egy termet, lehetőségünk van új előadás kiírására a létrehozott filmekhez.

Itt is minden mező kitöltése kötelező, ill. csak következő napra vagy későbbi időpontra tudunk új előadást kiírni és nem tudunk felvenni előadást egy adott időpontra egy adott terembe, ha már van kiírva előadás arra az időpontra abban a teremben.

Ugyanazt a filmet egyszerre csak egy teremben lehet játszani.



26.ábra

2.3.2 Keresések/törlések

2.3.2.1 Foglalások keresése/törlése

Lehetőségünk van keresni a foglalások között, ill. törölni is tudunk egy adott foglalást. Alapértelmezetten, ha a *Keres* gombra kattintunk úgy, hogy a *Foglalási szám* és a *Felhasználónév* mezők üresek, akkor az összes foglalást megkapjuk, melyek között az *Előző foglalás* és a *Következő foglalás* gombokkal tudunk lapozni.

Tudunk keresni csak *Foglalási szám* és csak *Felhasználónév* alapján, de akár mindkettő szerint is egyszerre, azonban ügyelni kell, hiszen csak akkor fog találatot adni a keresésünk *Foglalási szám* alapján való kereséskor, ha a mező értéke karakterre pontosan megegyezik a keresett foglalás foglalási számával.

Felhasználónév szerinti kereséskor, az összes olyan foglalást megkapjuk, amelynek foglalójának a felhasználóneve tartalmazza a beírt szöveget (pl.: nd → Andi, Endi stb.).

Üdv, Admin! (admin)

Film Keresés/Törlés Vélemények Termek Adatok Kijelentkezés

Foglalások keresése/törlése

Foglalási szám: 1P45X4BME9

Felhasználónév: Pru

Keres **Töröl**

Foglalási szám: 1P45X4BME9

Foglaló személy: Prumcsi

Filmcím: Aasd

Előadás időpontja: 2017-11-28 19:00:00.0

Teremszám: 4

Foglalás időpontja: 2017-11-25 04:41:00.0

Lefoglalt székek: 48,49,50

Előző foglalás PDF Táblázat Következő foglalás

Foglalás törlése

27.ábra

Továbbá a kilistázott foglalásokat (egyesével, mindig csak az éppen a képernyőn levőt) tudjuk exportálni egy **PDF** fájlba, amelynek neve a foglaló felhasználónevéből és a foglalási számból áll.

Ha a foglalást törölni szeretnénk, akkor erre is van lehetőségünk a *Foglalás törlése* gomb megnyomásával majd jóváhagyásával. Ha nincs kiválasztva foglalás, akkor a törlés gomb megnyomását követően hibaüzenetet kapunk.

Legvégül, ha túl sok eredményt adott a keresésünk a feltételek alapján, akkor a könnyebb átláthatóság érdekében a *Táblázat* gombra kattintva az összes, a feltételnek megfelelő foglalást egy táblázat formájában megtudjuk tekinteni és sorba is rendezhetjük

őket különböző értékek szerint, pl. Foglalás időpontja szerint növekvő/csökkenő sorrendben.

Foglalási szám	Felhasználó	Filmcím	Vetítés időpontja	Teremszám	Foglalás időpontja	Ár(Font)
1P45K4ME9	Prumcsi	Asd	2017-11-28 19:00:00.0	4	2017-11-25 04:41:00.0	4500
A31BQ584BA	Prumcsi	Asd	2017-11-28 19:00:00.0	4	2017-11-25 04:41:11.0	3000

28.ábra

2.3.2.2 Filmek keresése/törlése

A foglalásokhoz hasonlóan itt is az alapértelmezett keresés az összes filmet kilistázza, azonban lehetőségünk van *Filmcím/Kategória* szerint keresni a filmek között külön-külön vagy a két feltétel alapján egyszerre is.

29.ábra

Fontos megemlíteni, hogy a *Filmcím*-nél nem feltétlen kell a teljes filmcímet begépelnünk, a keresés az összes olyan filmet kifogja listázni, amelyeknek címében szerepel a keresett filmcím. A talált filmek között az *Előző film* és *Következő film* gombokkal tudunk lapozni.

A *Táblázat gomb* megnyomására a filmeket egy jól átlátható táblázat formájában láthatjuk, amelynek sorait rendezni tudjuk az oszlopok értéke szerint pl. *Filmcím* szerint abc sorrendben.

Ha esetleg a filmünket ki szeretnénk törölni, akkor erre is van lehetőségünk a *Film törlése* gombbal, abban az esetben, ha a filmhez nincsen jövőbeli foglalás, ellenkező esetben hibaüzenet értesít minket arról, hogy a film nem törölhető.

A *Film törlése* gomb megnyomását követően még egy megerősítő ablak kérdezi minket, hogy biztosan törölni szeretnénk-e az adott filmet.

Abban az esetben, ha a törölni kívánt filmhez csak múltbéli foglalások vannak, akkor minden gond nélkül törölhető.

Ha nincs kiválasztva film és úgy szeretnénk törölni, akkor hibaüzenetet kapunk.

Film törlésekor az összes, a filmhez tartozó foglalás és előadás is törlésre kerül.

2.3.2.3 Felhasználók keresése/törlése

Az utolsó ilyen menüpontunk a felhasználók keresése/törlése.

Az alapértelmezett keresésnél az összes felhasználót megkapjuk, viszont tudunk keresni *Felhasználónév* és *Email cím* alapján külön-külön, de akár mindkettő alapján egyszerre is, viszont itt pontosan kell megadnunk a felhasználónevet/email címet, hiszen csak ekkor lesz sikeres a keresésünk, feltéve, hogy létezik a keresett felhasználó.

30.ábra

A kilistázott felhasználók között az *Előző felhasználó* és a *Következő felhasználó* gombokkal tudunk váltogatni.

A *Táblázat gomb* jelen esetben a feltételnek megfelelő kereséseket fogja táblázat formájában megmutatni, amelynek sorai rendezhetők a különböző oszlopok pl. *Felhasználónév* szerint abc sorrendben.

Felhasználók					
Név	Felhasználónév	Email cím	Regisztráció időpontja	Foglalások száma	Státusz
Első Admin	Admin	zsollfischer@gmail.com	2000-01-01 00:00:00.0	0	admin

31.ábra

Felhasználó törlésére is lehetőségünk van, azonban itt figyelniünk kell, hiszen felhasználót akkor is tudunk törölni, ha van jövőbeli foglalása. Éppen ezért, felhasználót csak különleges esetben, pl. szabálysértés esetén törölhetünk, amelyet meg is kell indokolnunk a *Felhasználó törlése* gomb megnyomását és a törlés véglegesítését követően.

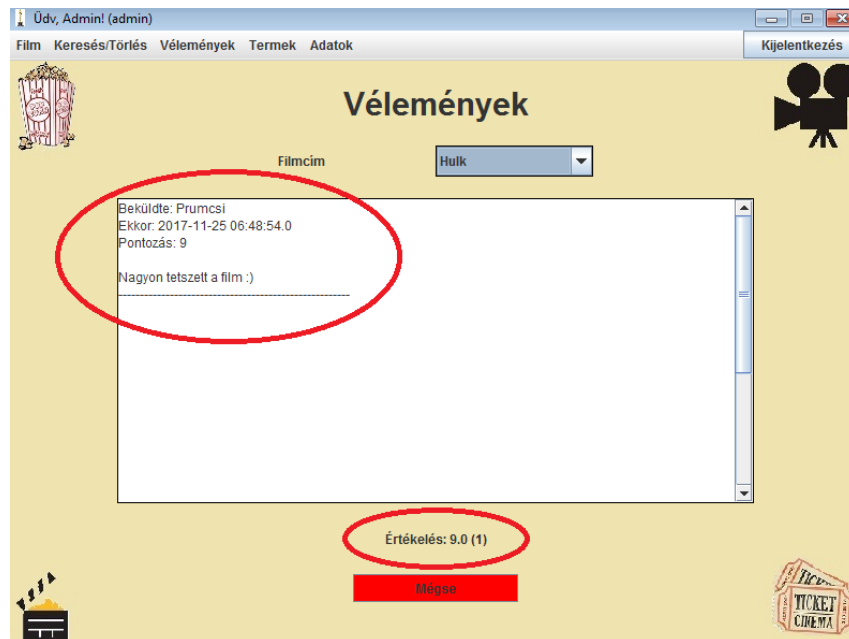
A törölt felhasználó email-ben kap értesítést a törlésről és annak okáról.

Felhasználó törlésekor az összes, a felhasználóhoz tartozó foglalás is törlésre kerül.

Más adminisztrátorokat és saját magunkat nem tudjuk törölni.

2.3.3 Felhasználói vélemények

A felhasználóknak lehetőségük van véleményeket írni a filmekről, amelyeket adminisztrátorként és felhasználóként is egyaránt meg lehet tekinteni.



32.ábra

Egy adott filmről írt véleményeket úgy kaphatjuk meg, hogy a lenyíló listánkból kiválasztjuk a filmet és ha vannak hozzá vélemények írva, akkor azok középen felsorolva jelennek meg elválasztva, a legfrissebb véleménnyel kezdve.

Egy vélemény tartalmazza azt, hogy ki írta, mikor írta, hány pontot adott a filmre 10-ből, és magát a véleményt. Ekkor a középső mező és a *Mégse gomb* között olvashatjuk a film átlagpontját és zárójelben azt is, hogy ez az átlag hány értékelés átlaga.

Abban az esetben, ha nincs vélemény még a kiválasztott filmhez, akkor a „*Nincsenek még kommentek ehhez a filmhez!*” szöveget olvashatjuk a mezőben.

2.3.4 Saját adatok módosítása

Lehetőség van módosítani az *email-címet* és a fiókhoz tartozó *jelszót*.



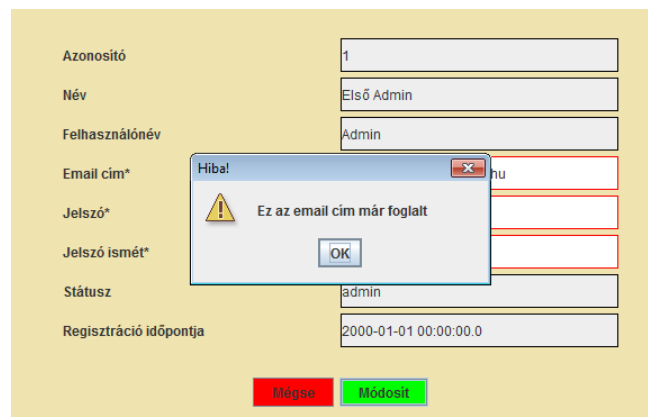
33.ábra

Az *Adatok módosítása* gomb megnyomása után tudjuk módosítani az adatainkat.

- Minden mező kitöltése kötelező
- Email-cím módosítására akkor van lehetőség, ha az újonnan megadott email-cím még szabad, tehát nem regisztrált még azzal az email-címmel senki (max. 40 karakter hosszú).
- Jelszó változtatásra is lehetőség van, de a *Jelszó* és *Jelszó ismét* mezők értékeinek meg kell egyezniük (max. 30 karakter hosszú).

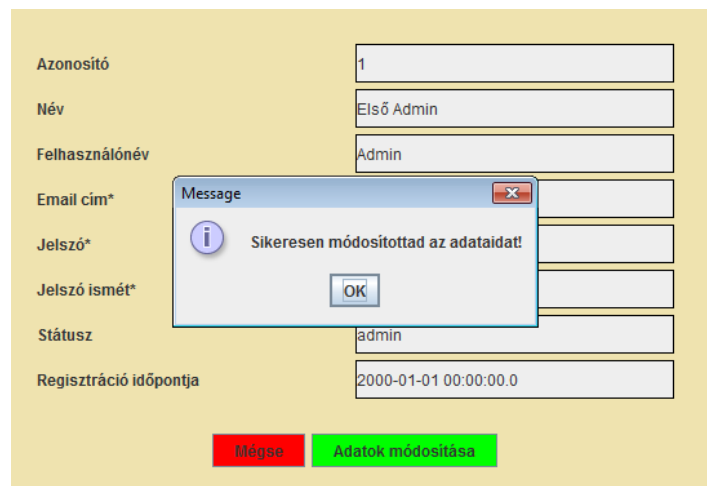
Ha bármelyik feltétel nem teljesül a fentiek közül, akkor hibaüzenetet kapunk, ellenkező esetben az adatmódosítás sikeres.

Példa hibára (foglalt email-cím):



34.ábra

Sikeres adatmódosítás:



The screenshot shows a user profile editing form on a yellow background. The form contains the following fields and values:

Field	Value
Azonosító	1
Név	Első Admin
Felhasználónév	Admin
Email cím*	
Jelszó*	
Jelszó ismét*	
Státusz	admin
Regisztráció időpontja	2000-01-01 00:00:00.0

At the bottom of the form are two buttons: a red "Mégse" (Cancel) button and a green "Adatok módosítása" (Save) button. A blue "Message" dialog box is overlaid on the form, displaying the text "Sikeresen módosítottad az adataidat!" (You successfully modified your data!) and an "OK" button.

35.ábra

2.3.5 Kijelentkezés

A menüsor jobb szélén lévő *Kijelentkezés* gomb segítségével tudunk kijelentkezni.

Ezt követően a bejelentkező oldalra kerülünk vissza, ahol újra bejelentkezhethetünk vagy regisztrálhatunk új felhasználóként.



36.ábra

2.4 Felhasználói funkciók

Ahhoz, hogy a programunk szolgáltatásait igénybe tudjuk venni, regisztrációra van szükség. Regisztrációt, majd bejelentkezést követően érhetők el számunkra a felhasználói funkciók.

Felhasználóként lehetőségünk van:

- regisztrálni
- új foglalást indítani
- filmek/saját foglalások között keresni különböző szempontok alapján
- saját foglalások törlésére, ill. PDF fájlba exportálására
- véleményt írni adott filmről
- felhasználói vélemények olvasására filmek szerint
- saját adatok módosítására

2.4.1 Regisztráció

A regisztráció elengedhetetlen a program szolgáltatásainak igénybevételéhez, hiszen csak bejelentkezést követően érhetők el a funkciók számunkra.

Regisztrációkor automatikusan felhasználóként regisztrálunk, adminisztrátorként nincs lehetőség regisztrációra.



37.ábra

Regisztrációkor minden mező kitöltése kötelező.

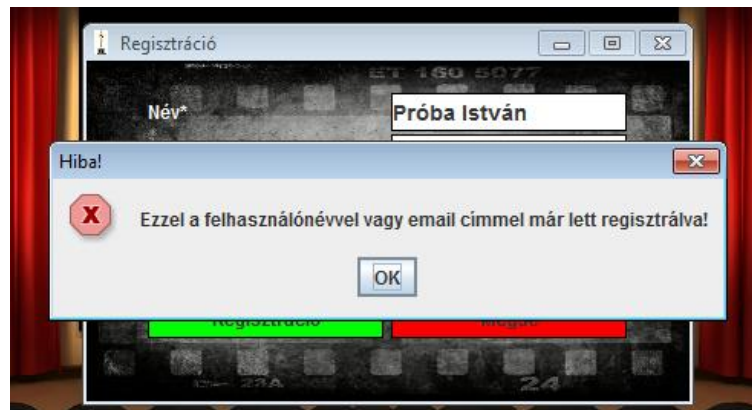
A *felhasználónév* (max. 30 karakter hosszú), *email* (max. 40 karakter hosszú) mezők egyediék, tehát nem lehet olyan felhasználónévvel/email címmel regisztrálni, amivel korábban már valaki beregisztált.

A *név* mező max. 50 karakter hosszú lehet.

A *jelszó* és *jelszó újra* mezők értékének meg kell egyezniük (max. 30 karakter hosszúak). Nagyon **fontos**, hogy az email-cím valós, létező email-cím kell, hogy legyen, hiszen regisztrációt követően erre az email-címre egy üzenetet fog kapni a felhasználó, ami egy **visszaigazoló kódot** tartalmaz. A regisztráció csak abban az esetben lesz sikeres és teljes, ha ezt a kódot megadjuk a programnak!

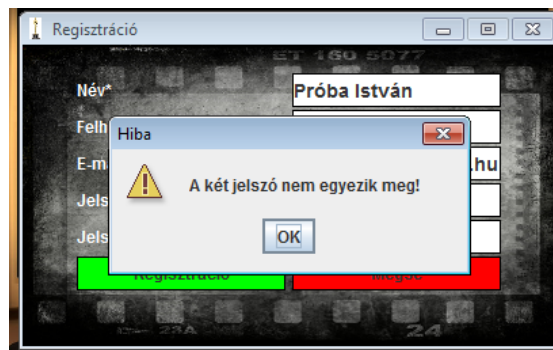
Abban az esetben, ha bármely feltétel nem teljesül, hibaüzenetet kapunk.

Már létező felhasználónév/email-cím:



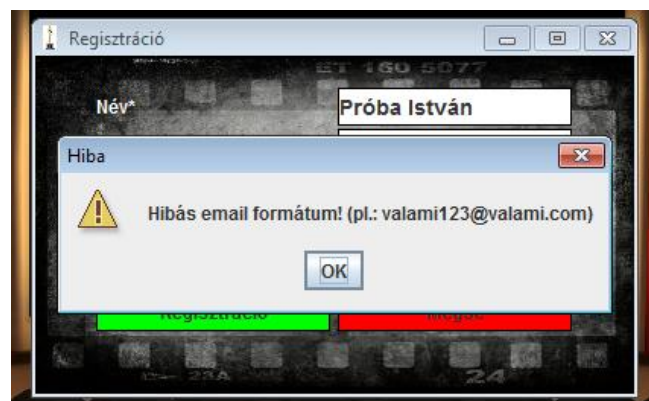
38.ábra

A két jelszó eltér:



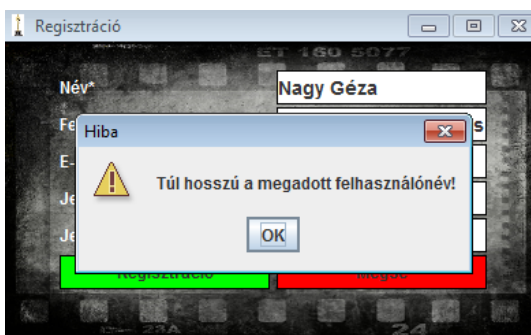
39.ábra

Hibás email formátum:



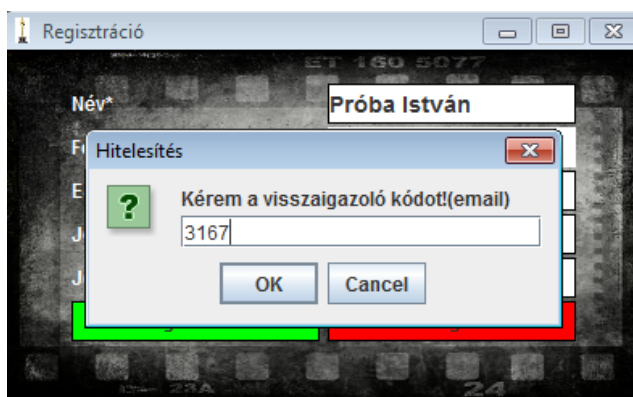
40.ábra

Túl hosszú felhasználónév:



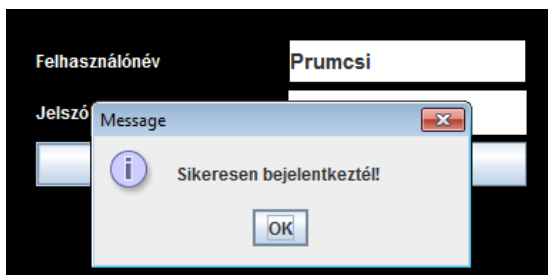
41.ábra

Ha a megadott adatok megfelelőek, akkor a megadott email-címre megérkezik a visszaigazoló kód, amit rögtön kér is a program a sikeres regisztráció érdekében.



42.ábra

A programunk addig fogja kérni a visszaigazoló kódot, amíg azt sikeresen meg nem adjuk vagy ki nem X-eljük a Hitelesítés ablakot. Utóbbi esetben viszont nem lesz sikeres a regisztráció, és az adatbázis tartalma sem változik (tehát nem kerül be új felhasználó). Abba az esetben, ha ki X-eltük vagy a Cancel gombra kattintottunk, akkor a regisztrációt meg kell ismételni. Ekkor a rendszer újabb email-t küld ki egy új visszaigazoló kóddal. Ha a kódot sikeresen megadtuk, akkor a regisztráció sikeresen befejeződik és most már be tudunk jelentkezni az imént megadott felhasználónévvel és jelszóval. Sikeres bejelentkezést követően a főoldal fogad minket.



43.ábra

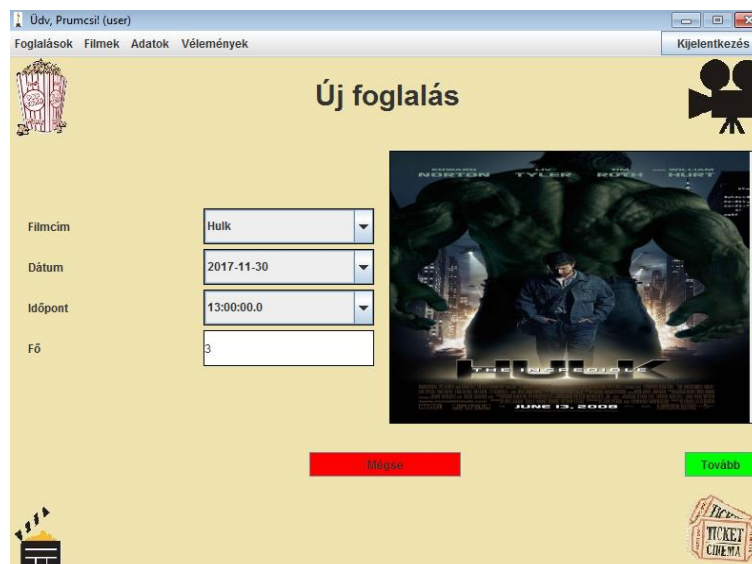
Főoldal:



44.ábra

2.4.2 Új foglalás indítása

Mivel a programunk lényege, hogy filmekre foglaljunk helyeket, ezért nyilván felhasználóként lehetőségünk van erre.



45.ábra

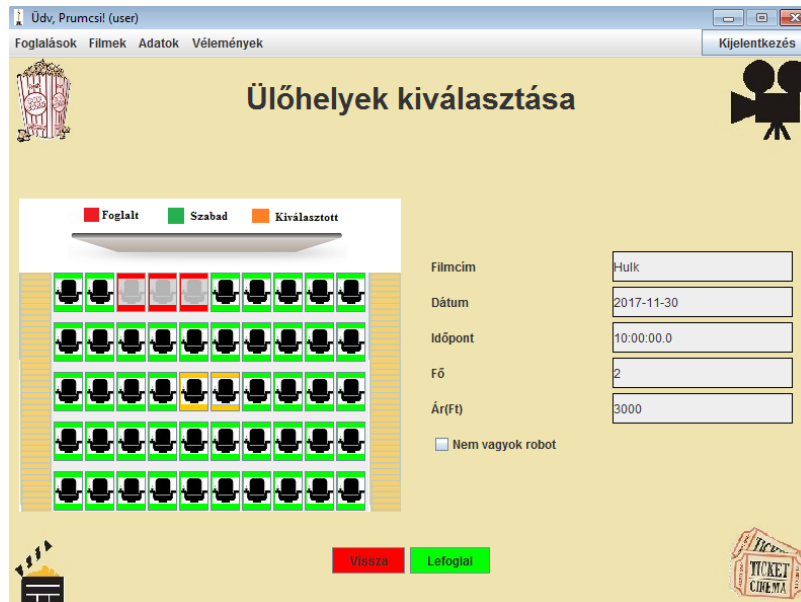
Filmcím, dátum, időpont kiválasztásával és a *fő* megadásával indul a foglalási procedúra.

A *Fő* formátuma szám kell, hogy legyen, ill. nem lehet 1-nél kisebb és 10-nél nagyobb.

Ellenkező esetben hibüzenetet kapunk.

Ha a megadott adatok megfelelőek, akkor a *Tovább* gomb megnyomásával az ülőhelyek kiválasztása következik.

Az alkalmazás a megadott filmcím, dátum és időpont kombinációból kideríti, hogy melyik teremben lesz a vetítés, s a terem sorainak/oszlopainak számának megfelelő mennyiségű szék rajzolódik ki, hiszen különböző méretű termekkel rendelkezhet egy mozi.



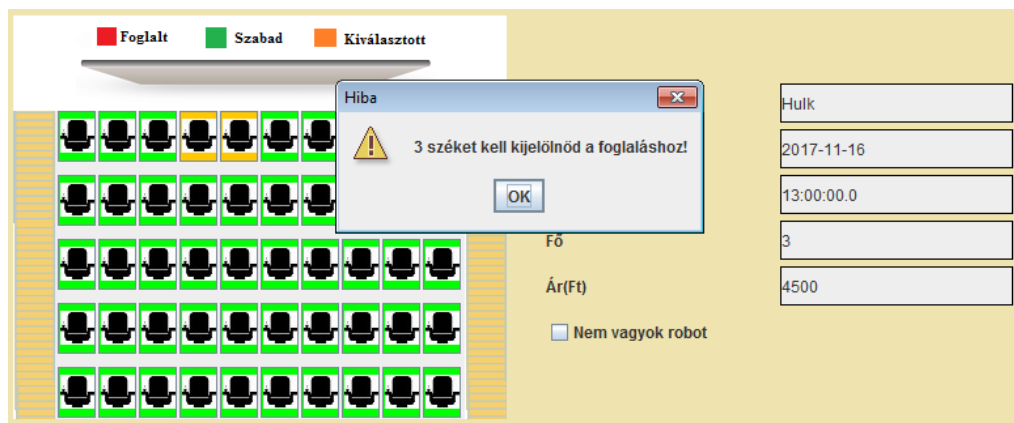
46.ábra

A szabad székek zöld háttérrel, a lefoglalt székek piros háttérrel, a kiválasztott székek pedig narancssárga háttérrel láthatók.

- Csakis annyi ülőhelyet tudunk kiválasztani, ahány $Fő$ -t megadtunk az előző oldalon, de ennyit ki is kell.
- Nem választhatunk ki ülőhelyet úgy, hogy a sor elején/végén vagy két ülés között 1 hely marad szabadon.
- Biztonsági okokból a *Nem vagyok robot* kipipálása is szükséges a sikeres foglaláshoz.

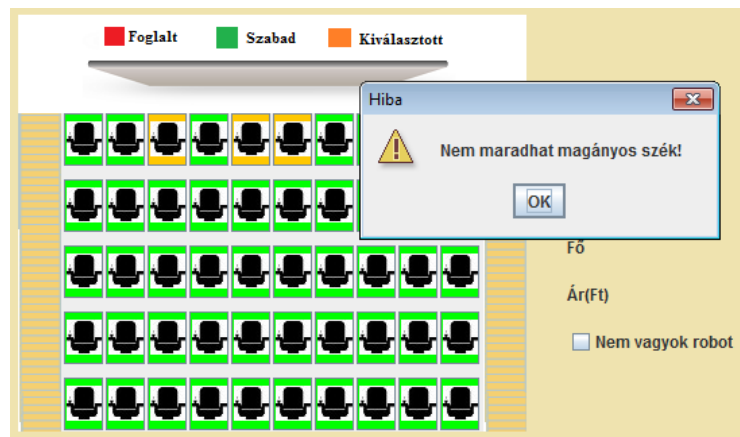
Ha a fenti feltételek közül bármelyik is nem teljesül, akkor a program hibaüzenettel jelzi a problémát és a foglalás sikertelen.

Kevés ülőhely kiválasztása:



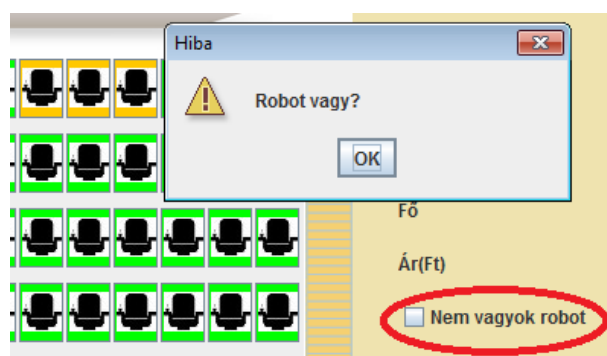
47.ábra

1 hely marad szabadon:



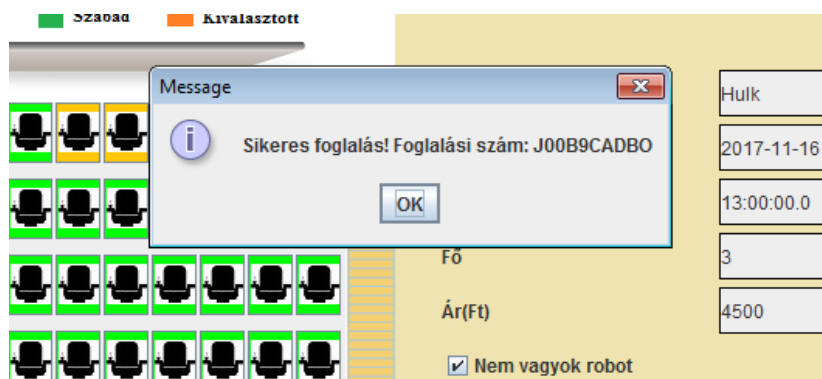
48.ábra

Nem vagyok robot:



49.ábra

Ha minden feltétel teljesül, akkor a regisztráció sikeres, melyről email-t is kapunk a legfontosabb információkkal.



50.ábra

Email:

elte.moziverzum@gmail.com - Sikeres foglalás! J00B9CADBO

Kedves Prumcsi, köszönjük a foglalását!

Foglalási szám: J00B9CADBO

Felhasználónév: Prumcsi

Filmcím: Hulk

Dátum: 2017-11-16

Időpont: 13:00:00.0

Terem: 1

Fő: 3

Székek: 5,6,7

Ár: 4500Ft

Jó szórakozást kíván, a Moziverzum csapata!

51.ábra

2.4.3 Keresések

2.4.3.1 Keresés saját foglalások között

Lehetőségünk van saját foglalásaink között keresni különböző szempontok alapján.

Ilyen szempont a *Foglalási szám* és a *Filmcím*. Ezen szempontok alapján külön-külön vagy akár egyszerre is tudunk keresni.

52.ábra

Foglalási szám esetén csak akkor lesz eredményes a keresés, ha a megadott *Foglalási szám* karakterre pontosan megegyezik a keresett foglalás foglalási számával.

A *Filmcím* esetében viszont nem szükséges a teljes filmcímet megadni, pl.: ha csak egy szót írunk a kereső mezőbe, akkor az összes olyan filmet kilistázza, amelynek címében szerepel az adott szó.

A kilistázott foglalások között az *Előző foglalás* és *Következő foglalás* gombok segítségével tudunk lapozni oda-vissza.

Lehetőségünk van táblázatos nézetre a jobb átláthatóság érdekében a *Táblázat* gomb megnyomásával. A táblázat mindig az adott feltételeknek megfelelő keresés eredményeit tartalmazza. Az alapértelmezett keresés az összes Saját foglalást kilistázza.

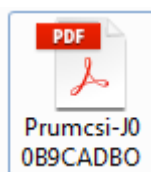
Foglalási szám	Felhasználó	Filmcím	Vetítés időpontja	Teremszám	Foglalás időpontja	Ár(Forint)
J00B9CADBO	Prumcsi	Hulk	2017-11-16 13:00:00.0	1	2017-11-02 12:57:38.0	4500

53.ábra

Ha esetleg mégsem tudunk elmenni a lefoglalt filmre, akkor sincsen probléma. Egyszerűen tudjuk törölni a foglalásunkat a *Foglalás törlése* gomb megnyomásával, majd jóváhagyásával.

Egy nagyon fontos opció a *PDF* gomb, hiszen ezzel képesek vagyunk az adott foglalásunkat (tehát, amelynek adatait épp a képernyőn látjuk → táblázatos nézet esetén nem működik) PDF fájlba exportálni, amelyet utána könnyedén kinyomtathatunk.

A PDF fájl neve a felhasználó nevéből és a foglalási számból áll, és tartalmazza az összes fontos információt a foglalással kapcsolatosan.



54.ábra

Az újonnan exportált PDF fájl megnyitását követően a következő kinézetű (de más tartalmú) dokumentum kell, hogy fogadjon minket:



Foglalás visszaigazolás

Személyes adatok	
Név:	Próba István
Felhasználónév:	Prumcsi
Email cím:	mazsola2vidi@freemail.hu

Vetítési adatok	
Foglalási szám:	J00B9CADBO
Filmcím:	Hulk
Vetítés időpontja:	2017-11-16 13:00:00.0
Teremszám:	1
Lefoglalt székek:	5,6,7
Jegvár(összesen):	4500Ft

Kérjük, hogy vetítés előtt legalább fél órával vegye át jegyét a pénztárnaknál!
A kedvezményre jogosító dokumentumot a pénztárnál kell bemutatni!

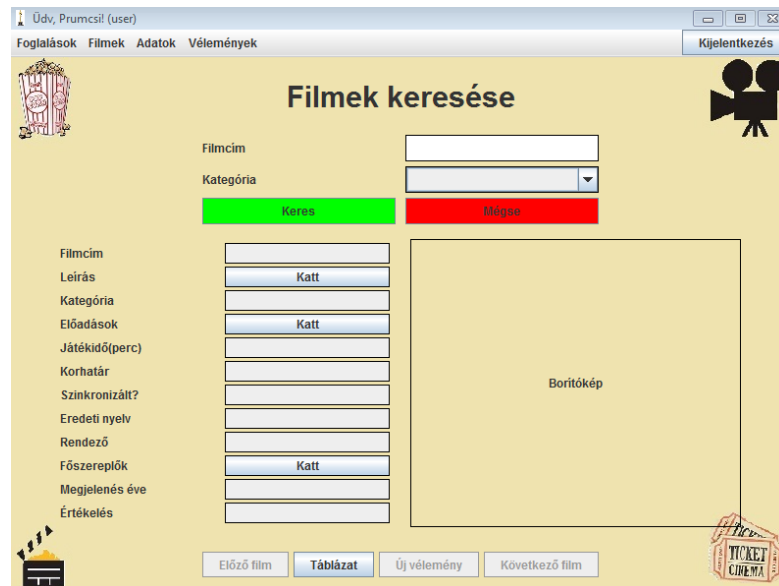
Jó szórakozást kíván a Moziverzum csapata!



55.ábra

2.4.3.2 Keresés filmek között

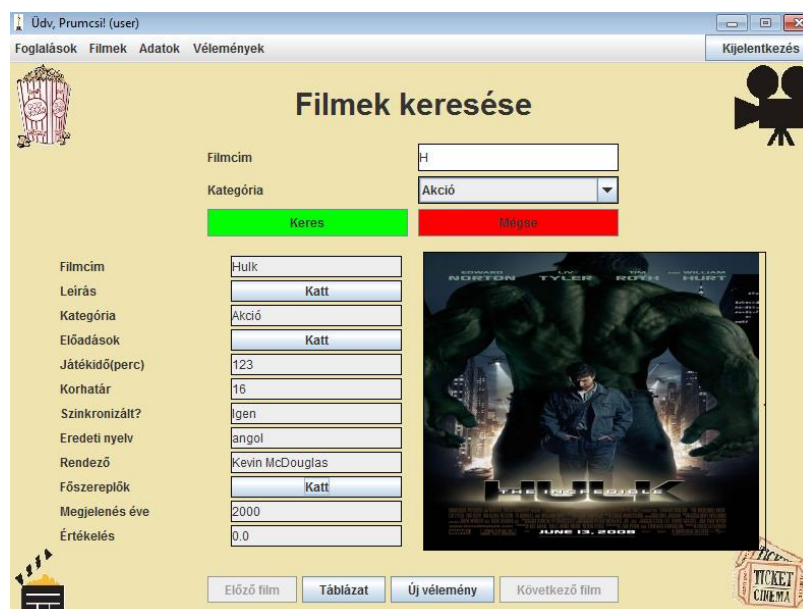
A mozifilmek között is lehetőségünk van keresni különböző szempontok alapján, mint például a *Filmcím* és *Kategória* szerint külön-külön vagy akár egyszerre is.



56.ábra

Ez a keresési oldal szinte ugyanúgy működik, mintha adminisztrátorként lennénk bejelentkezve, azzal a kivétellel, hogy itt nem tudunk filmet törölni, viszont tudjuk véleményezni azokat.

A *Filmcím* alapján való keresésnél itt sem kell a teljes filmcímet megadnunk, elég, ha csak egy szót vagy az első pár betűt írjuk be, s ekkor az összes olyan filmet megkapjuk, amelynek címében szerepel a megadott szó.

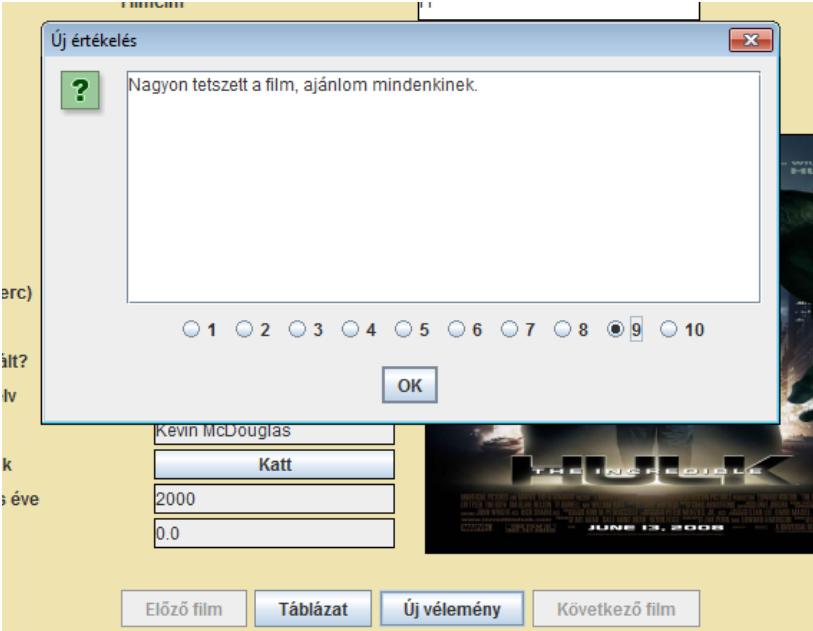


57.ábra

A keresett filmek között az *Előző film* és *Következő film* gombokkal tudunk váltogatni. A *Táblázat* használata esetén átláthatóbb képet kapunk a kilistázott filmekről egy táblázat formájában, melyben a sorokat rendezni is tudjuk a különböző oszlopok szerint. Amint egy film kiválasztásra került, elérhetővé válik az *Új vélemény*, mellyel véleményt tudunk írni az adott filmről és pontozni is tudjuk egy 10-es skálán, de erről részletesebben a következő menüpontban olvasható.

2.4.4 Új felhasználói vélemény

Sokszor választunk filmet az alapján, hogy mások miként értékelték azt. Éppen ezért van lehetőség vélemény írására és pontozásra egy filmmel kapcsolatban.



58.ábra

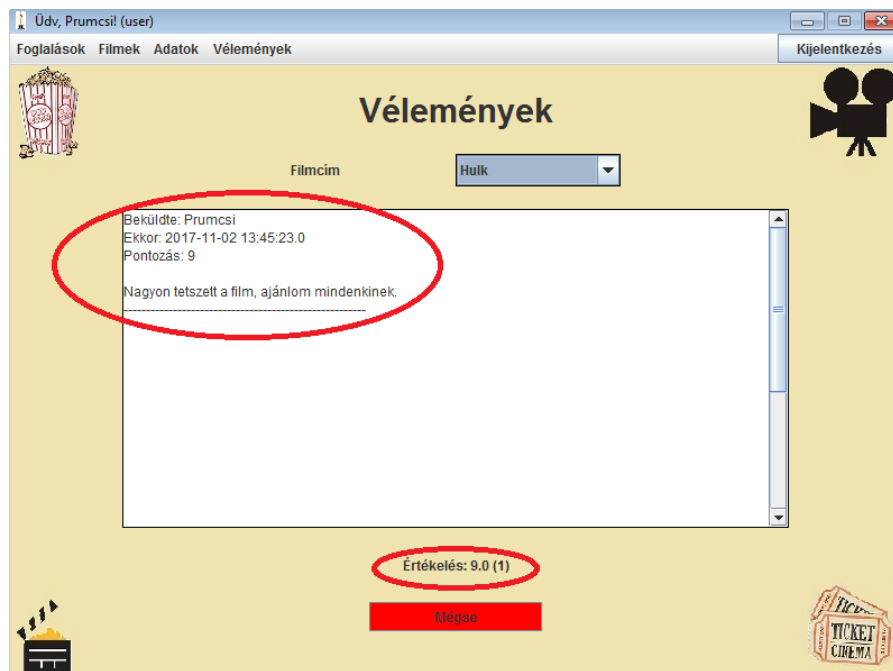
Minden mező kitöltése kötelező egy vélemény írásakor, ellenkező esetben hibaüzenetet kapunk.

Először is le kell írunk a véleményünket (max. 400 karakter), utána pedig egy 10-es skálán pontoznunk kell a filmet, majd az *OK* gomb megnyomásával tudjuk elküldeni az értékelésünket.

Több értékelést is írhatunk egy filmmel kapcsolatban.

2.4.5 Felhasználói vélemények

Felhasználói véleményeket tudunk olvasni adott filmről, amihez annyit kell csupán tennünk, hogy kiválasztjuk melyik filmről szeretnénk elolvasni az értékeléseket.



59.ábra

Amint a fenti ábrán láthatjuk, az imént írt vélemény már olvasható is a film kiválasztását követően.

Egy megjegyzés áll egy *felhasználónévből*, egy *időpontból*, egy *pontozásból*(10-ből) és magából a *véleményből*. Alul pedig olvasható a film átlaga és mellette zárójelben az, hogy hány értékelés alapján kaptuk ezt az átlagot.

2.4.6 Saját adatok módosítása

Lehetőség van módosítani az *email-címét* és az fiókhoz tartozó *jelszavát*.

60.ábra

Az *Adatok módosítása* gomb megnyomása után tudjuk módosítani az adatainkat.

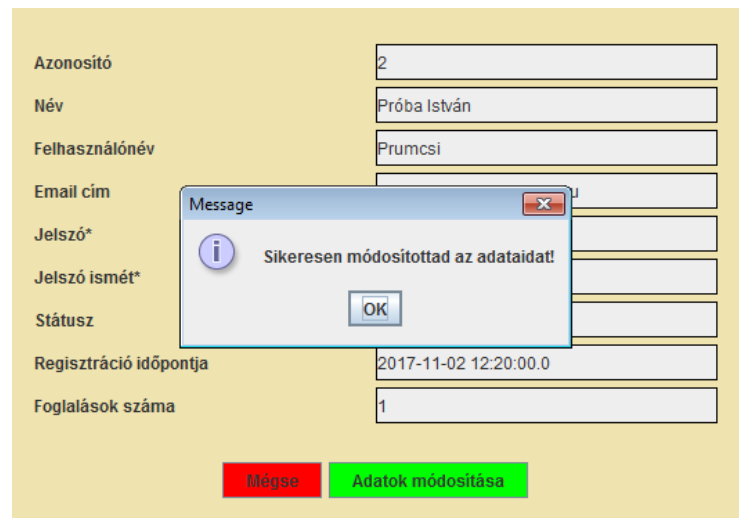
- Minden mező kitöltése kötelező
- Email-cím módosítására akkor van lehetőség, ha az újonnan megadott email-cím még szabad, tehát nem regisztrált még azzal email-címmel senki (max. 40 karakter hosszú).
- Jelszó változtatásra is lehetőség van, de a *Jelszó* és *Jelszó ismét* mezők értékeinek meg kell egyezniük (max. 30 karakter hosszú).

Ha bármelyik feltétel nem teljesül a fentiek közül, akkor hibaüzenetet kapunk, ellenkező esetben az adatmódosítás sikeres.

Példa hibára (foglalt email-cím):

61.ábra

Sikeres adاتمódosítás:



The image shows a web form for updating user data. The form has a light yellow background and contains the following fields:

Field Label	Value
Azonosító	2
Név	Próba István
Felhasználónév	Prumcsi
Email cím	
Jelszó*	
Jelszó ismét*	
Státusz	
Regisztráció időpontja	2017-11-02 12:20:00.0
Foglalások száma	1

At the bottom of the form are two buttons: a red "Mégse" (Cancel) button and a green "Adatok módosítása" (Update data) button.

A modal dialog box titled "Message" is overlaid on the form. It contains an information icon, the text "Sikeresen módosítottad az adataidat!" (You successfully modified your data!), and an "OK" button.

62.ábra

2.4.7 Kijelentkezés

Legvégül pedig értelemszerűen felhasználóként is lehetőségünk van kijelentkezni.

Ekkor a bejelentkező oldalra kerülünk vissza, ahol ismét betudunk jelentkezni vagy regisztrálhatunk új felhasználóként.

3 Fejlesztői dokumentáció

3.1 Feladat leírása

A szakdolgozat elkészítésének célja egy olyan alkalmazás létrehozása, amely egy mozival kapcsolatosan előforduló feladatokat lát el.

Minden funkció csak bejelentkezést követően válik elérhetővé. Bejelentkezni adminisztrátorként és felhasználóként is be lehet, viszont regisztrálni csak felhasználót lehet. Adminisztrátor regisztrálására közvetlenül az alkalmazásból nincs lehetőség.

Adminisztrátorként bejelentkezést követően minden menüpont használhatóvá válik.

Lehetőség van új film felvételére az adatok helyes megadásával. Több különböző méretű terem felvételére is lehetőséget ad az alkalmazás, melyek elengedhetetlenek új előadás kiírásához. A sikeresen felvett filmekhez új előadást is kiírhatunk az előadás időpontjának és helyének (teremszám) megadásával.

A mozifilmek, regisztrált felhasználók és foglalások között különböző szempontok alapján van lehetőség keresésre. A keresett filmet, felhasználót és foglalást adminisztrátorként képesek vagyunk törölni bizonyos feltételek mellett.

A foglalásokat PDF fájlba exportálhatjuk, melyet később könnyedén kinyomtathatunk.

A felhasználók által írt értékeléseket a megfelelő oldalon olvashatjuk a film kiválasztását követően.

Végül pedig lehetőség van az email-címünk és jelszavunk megváltoztatására is, ha idejét érezzük a frissítésnek.

Felhasználóként lehetőség van foglalást indítani a megfelelő előadás, majd az ülőhelyek kiválasztásával. Ezek a foglalások később vissza kereshetők, törölhetők, ill. exportálhatók.

A filmek közötti keresés több szempont szerint is lehetséges, melyek véleményezhetők is. A saját és mások által írt értékelések könnyedén vissza olvashatóak később.

Ha szeretné a felhasználó megváltoztatni az email címét vagy a jelszavát, akkor az alkalmazás erre is lehetőséget ad.

Az alkalmazás kezeli a felmerülő hibákat és kivételeket, s (hiba)üzenetekkel értesíti az alkalmazás használóját minden lépésben.

3.2 Fejlesztéshez szükséges telepítések

3.2.1 Fejlesztő környezet

A program fejlesztéséhez nem szükséges, de ajánlott egy fejlesztő környezet.

Én személy szerint a program írása közben *NetBeans IDE*-t használtam, mivel a használata gyors és egyszerű.

Letölteni a JDK-val (*Java Development Kit*) együtt érdemes a hivatalos oldalról³.

3.2.2 JDK

Java alkalmazások fejlesztéséhez szükségünk van a legalább 1.8-as verziószámú JDK-ra (*Java Development Kit*), melyet vagy letöltünk az Oracle hivatalos oldaláról⁴ külön, vagy pedig a NetBeans-sel együtt töltjük le az előző pontban (3.2.1 Fejlesztő környezet) megadott oldalról, ha azt a fejlesztő környezetet választjuk.

Telepítést követően állítsuk be a megfelelő környezeti változókat.

3.2.3 MySQL

A program használata során az adatainkat MySQL adatbázisban tároljuk. Letöltése könnyedén megtehető a hivatalos oldalról⁵.

A telepítésről részletesen a Felhasználói dokumentációban olvashatunk.

A **legfontosabb**, hogy hozzunk létre egy felhasználót, pl.: „**admin**” névvel és „**password**” jelszóval (ha még nem rendelkezünk MySQL accounttal), s garantáljunk neki **minden jogot** az adatbázishoz!

3.2.4 Felhasznált modulok

A *jcalendar-1.4.jar*⁶ a naptár megjelenítésére és használatára szolgál.

Az *itext5-itextpdf-5.5.12.jar*⁷ a foglалások PDF-be exportálására szolgál.

A *java-mail-1.4.4.jar*⁸ az email küldésére szolgál a program használata során.

A *junit-4.10.jar* pedig⁹ a Unit tesztelést teszi lehetővé.

³ <http://www.oracle.com/technetwork/articles/javase/jdk-netbeans-jsp-142931.html>

⁴ <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

⁵ <https://dev.mysql.com/downloads/windows/installer/5.7.html>

⁶ <http://www.java2s.com/Code/Jar/j/Downloadjcalendar14jar.htm>

⁷ <https://sourceforge.net/projects/itext/files/>

⁸ <http://www.java2s.com/Code/Jar/j/Downloadjavamail144jar.htm>

⁹ <https://sourceforge.net/projects/junit/files/junit/4.10/>

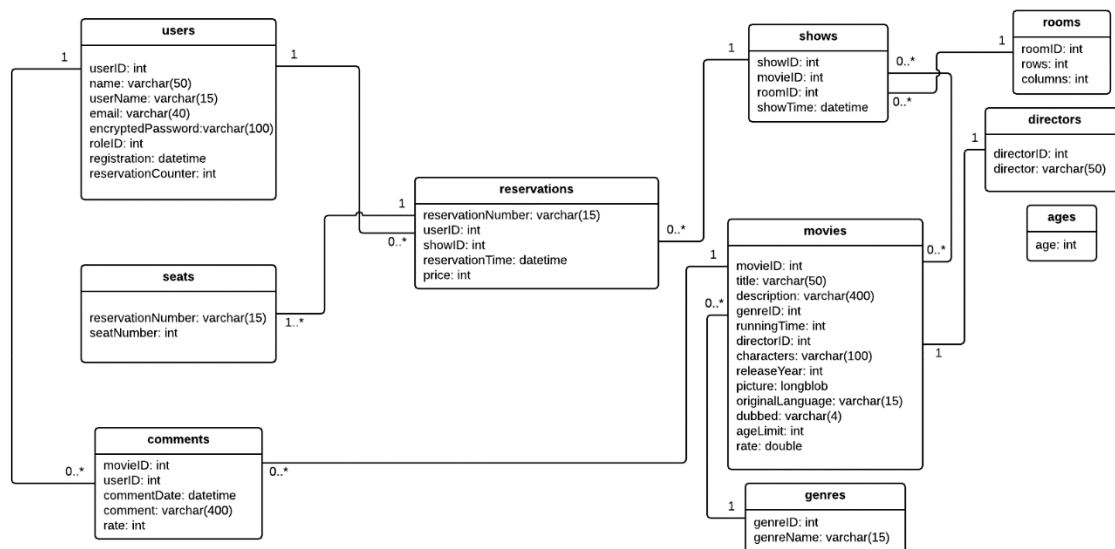
3.3 Megoldási terv

3.3.1 Adatbázis

A program használata során az adatainkat **MySQL** adatbázisban tároljuk.

Az adatbázisunk neve *cinema*.

Az adatbázis felépítése a következő:



63.ábra

Az adatbázisban minden rekord **not null**, tehát értékkel kell rendelkeznie.

Táblák felépítése

users

A *users* táblában tároljuk a felhasználókat.

- **userID** – egyesével növekvő egész típusú elsődleges kulcs, a felhasználó egyedi azonosítószáma.
- **name** – a felhasználó max. 50 karakteres teljes neve (vezetéknév és keresztnév).
- **userName** – a felhasználó max. 15 karakter hosszú egyedi felhasználóneve.
- **email** – a felhasználó max. 40 karakter hosszú egyedi email-címe.
- **encryptedPassword** – titkosított jelszó.
- **roleID** – egész típusú státusz azonosító (1-admin, 2-user).
- **registration** – dátum típusú regisztráció időpontja.
- **reservationCounter** – egész típus, amely számon tartja, hogy az adott felhasználó hányszor foglalt eddig. Ezt azért tároljuk külön az adatbázisban, mert egy film törlésekor az összes, a filmhez tartozó foglalást is töröljük.

A *users* táblánál megemlíteném, hogy ugyebár csak felhasználó regisztrálására ad lehetőséget az alkalmazás, adminisztrátoréra nem.

Adminisztrátort a MySQLWorkbench-en belül manuálisan lehet beszúrni az adatbázisba a *users* tábla megfelelő mezőinek helyes kitöltésével.

Példa:

```
insert into cinema.users (name, userName, email, encryptedPassword, roleID, registration, reservationCounter)
values ('Új Admin', 'UjAdmin', 'admin123@admin.com', '7815696ecbf1c96e6894b779456d330e', 1, '2017-01-01 00:00:00', 0);
/*7815696ecbf1c96e6894b779456d330e az "asd" titkosított változata, de ez később könnyedén megváltoztatható*/
```

64.ábra

Az „asd” jelszó titkosítása MD5, egyirányú titkosítással történt, melyet regisztrációkor egy ennek megfelelő osztály (Encrypt.java) végez el, s ez a titkosított jelszó kerül tárolásra az adatbázisban. Ezután minden bejelentkezéskor a program a beírt jelszót titkosítja, és ezt a titkosított jelszót veti össze az adatbázisban levő, a felhasználóhoz tartozó jelszóval.

rooms

A *rooms* táblában tároljuk a termeket, melyekben a székek téglalap alakban helyezkednek el.

- roomID – egyesével növekvő egész típusú elsődleges kulcs, a terem egyedi azonosítószáma.
- rows – sorok száma, egész típus.
- columns – oszlopok száma, egész típus.

directors

A *directors* táblában tároljuk a rendezőket.

- directorID – egyesével növekvő egész típusú elsődleges kulcs, a rendező egyedi azonosítószáma.
- director – a rendező egyedi max. 50 karakter hosszú neve.

ages

Az *ages* táblában tároljuk a korhatárokat, amelyre azért van szükség, mert a filmek korhatára nem lehet akármilyen szám, hanem előre meg vannak határozva (pl.: 6, 12 stb.).

- age – egyedi korhatár, egész típusú.

genres

A *genres* táblában tároljuk a kategóriákat (pl.: Akció, Horror, Dráma stb.).

- **genreID** – egyesével növekvő egész típusú elsődleges kulcs, a kategória egyedi azonosítószáma.
- **genreName** – a kategória egyedi max. 15 karakter hosszú neve.

movies

A *movies* táblában tároljuk a filmeket.

- **movieID** – egyesével növekvő egész típusú elsődleges kulcs, a film egyedi azonosítószáma.
- **title** – a film egyedi max. 50 karakter hosszú címe.
- **description** – a film max. 400 karakter hosszú leírása.
- **genreID** – a film műfajának egész típusú azonosítója.
- **runningTime** – a film egész típusú hossza (percben).
- **directorID** – a rendező egész típusú azonosítója.
- **characters** – főszereplők, max. 100 karakter hosszú.
- **releaseYear** – megjelenés éve, egész típusú.
- **picture** – a film longblob típusú borítóképe.
- **originalLanguage** – eredeti nyelv, max. 15 karakter hosszú.
- **dubbed** – szinkronizált-e a film vagy sem.
- **ageLimit** – egész típusú korhatár.
- **rate** – a film lebegőpontos átlagértékelése a felhasználók szerint.

shows

A *shows* táblában tároljuk az előadásokat.

- **showID** – egyesével növekvő egész típusú elsődleges kulcs, az előadás egyedi azonosítószáma.
- **movieID** – film egész típusú azonosítója, amelyre az előadás ki lett írva.
- **roomID** – egész típusú teremszám, amelyben az előadás lesz.
- **showTime** – az előadás dátum típusú időpontja.

reservations

A *reservations* táblában tároljuk a foglalásokat.

- reservationNumber – max. 15 karakter hosszú elsődleges kulcs, a foglalás egyedi azonosítószáma.
- userID – felhasználó egész típusú azonosítója, akihez a foglalás tartozik.
- showID – előadás egész típusú azonosítója, amelyikre a foglalás történt.
- reservationTime – foglalás dátum típusú dátuma (időponttal).
- price – foglalás egész típusú összege.

seats

A *seats* táblában tároljuk a lefoglalt székeket.

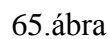
- reservationNumber – a foglalás max. 15 karakter hosszú száma, amelyhez a lefoglalt szék tartozik.
- seatNumber – egész típusú, a lefoglalt szék sorszáma.

comments

A *comments* táblában tároljuk a filmekről írt felhasználó véleményeket.

- movieID – film egész típusú azonosítója, amelyhez a megjegyzés írva lett.
- userID – felhasználó egész típusú azonosítója, aki a megjegyzést írta.
- commentDate – dátum típus, a megjegyzés dátuma (időponttal).
- comment – a max. 400 karakter hosszú vélemény.
- rate – egész típusú pontozás (10-es skálán).

A használati esetek (*use case*) diagramja azt ábrázolja, hogy miként működik együtt a felhasználó a rendszerrel.



1. Adminisztrátori funkciók

- 43

- Lehetőség van keresésre a filmek között foglalási szám és felhasználónév szerint.
- Keresés eredményének megtekintése.
- A kilistázott foglalást törölhetjük/PDF-be exportálhatjuk ill. egy táblázat formájában is megtekinthetjük.
- Keresés filmek között
 - Ebben a menüpontban van lehetőség keresni filmek között filmcím és kategória szerint.
 - A keresés eredményének megtekintése
 - A kilistázott filmeket törölhetjük, ha nincsenek jövőbeli foglalások hozzájuk, ill. táblázatban is megtekinthetjük őket.
- Keresés felhasználók között
 - Felhasználók között tudunk keresni felhasználónév és email cím alapján
 - A keresés eredményét megtekinthetjük.
 - A felhasználót törölhetjük is a törlés indokának megadásával, ill. táblázat formájában is megtekinthetjük a kilistázott felhasználókat.
- Vélemények
 - Ebben a menüpontban van lehetőség a felhasználók által írt vélemények megtekintésére a megfelelő film kiválasztásával.
- Termek
 - Új terem felvételére van itt lehetőség a sorok/oszlopok számának megadásával.
- Adatok
 - Saját adataink megtekintésére és módosítására (email, jelszó) ad lehetőséget ez a menüpont.

2. Felhasználói funkciók

- Bejelentkezés
 - A felhasználónév és jelszó megadását követően lehetőség bejelentkezésre.
- Regisztráció
 - Új felhasználó regisztrálása a mezők kitöltésével.
- Új foglalás
 - Új foglalást tudunk elindítani a film, dátum, időpont kiválasztásával és a személyek számának megadásával.
 - Ezt követően az ülőhelyeket tudjuk kiválasztani a feltételeknek megfelelően.
- Keresés saját foglalások között
 - Saját foglalásaink között tudunk keresni foglalási szám és filmcím alapján
 - A keresési eredményt megtekinthetjük.
 - A keresett foglalást törölhetjük/PDF-be exportálhatjuk vagy táblázat formájában is megtekinthetjük.
- Keresés filmek között
 - Ebben a menüpontban van lehetőség keresni filmek között filmcím és kategória szerint.
 - A keresés eredményének megtekintése
 - A kilistázott filmet táblázat formájában is megtekinthetjük, ill. új véleményt írhatunk hozzá a vélemény és pont megadásával.
- Saját adatlap
 - Lehetőség az adataink (email cím, jelszó) módosítására, ha az adott email cím nem foglalt még.
- Vélemények
 - Ebben a menüpontban van lehetőség a felhasználók által írt vélemények megtekintésére a megfelelő film kiválasztásával.

3.4 Fájlstruktúra

3.4.1 Fájlok csoportosítása

❖ Cinema project

- Adatbázis fájlok
 - Database.java
 - AdminDatabase.java
 - UserDatabase.java
- Adminisztrátori fájlok
 - AdminPage.java
 - NewMovie.java
 - NewShow.java
 - MovieSearch.java
 - UserSearch.java
 - ReservationSearch.java
 - Comments.java
 - AdminDetails.java
 - Rooms.java
- Felhasználói fájlok
 - UserPage.java
 - NewReservation.java
 - SeatsChooser.java
 - OwnReservations.java
 - OwnDetails.java
 - MovieUserSearch.java
 - CommentsUser.java
- Közös fájlok
 - Start.java
 - RegistrationPage.java
 - LoginPage.java
 - MainPanel.java
- Egyéb fájlok
 - Encrypt.java
 - Email.java

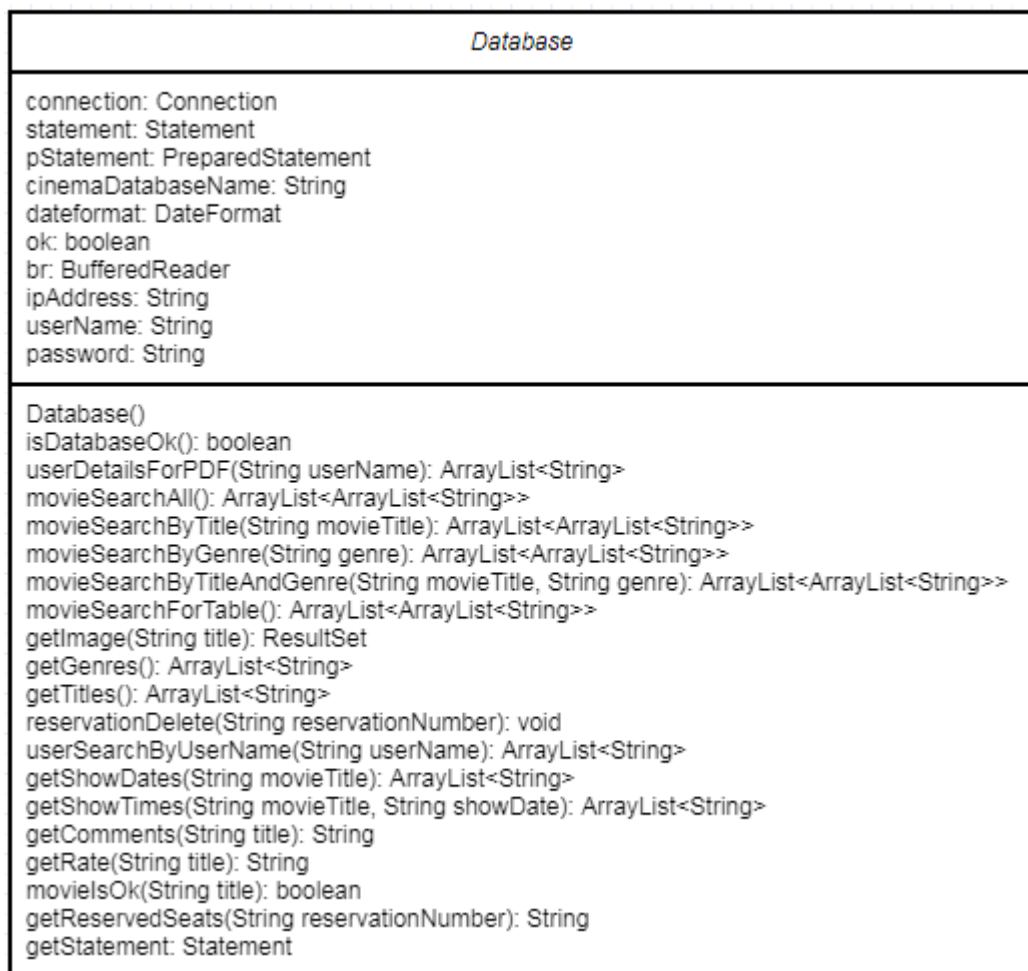
- PDF.java
- Seat.java
- Table.java

❖ Database project

- CreateDatabase.java
- Database.java

Cinema project

Database



66.ábra

A Database egy *abstract* osztály, amely csatlakozik az adatbázishoz. Van 2 leszármazottja, melyek a felhasználói adatbáziskéréseket és az adminisztrátori adatbáziskéréseket végzik, ill. a közös adatbázisműveleteket is ezt az osztály hajtja végre. *isDatabaseOk()* metódus tárolja, hogy a megadott adatbázis információk helyesek-e. *movieIsOk()* metódus visszaadja, hogy a megadott film szerepel-e már az adatbázisban.

AdminDatabase

AdminDatabase
<pre>AdminDatabase() newMovie(String title, String description, String genre, String director, String mainCh, File picture, FileInputStream fis, int releaseYear, int runningTime, String originalLanguage, String dubbed, int ageLimit): void addShow(String title, int roomID, String time, Date showDate): boolean hasReservation(String movieTitle): boolean movieDelete(String movieTitle): void commentsDelete(String movieTitle): void seatsDelete(ArrayList<Integer> showIDs): void reservationDelete(ArrayList<Integer> showIDs): void showDelete(ArrayList<Integer> showIDs): void userSearchByEmail(String email): ArrayList<String> userSearchByUserNameAndEmail(String userName, String email): ArrayList<String> userSearchAll(): ArrayList<ArrayList<String>> userDelete(String userName): void reservationSearchByNumber(String reservationNumber): ArrayList<ArrayList<String>> reservationSearchByUser(String userName): ArrayList<ArrayList<String>> reservationSearchByNumberAndUser(String reservationNumber, String userName): ArrayList<ArrayList<String>> reservationSearchAll(): ArrayList<ArrayList<String>> getAges(): ArrayList<Integer> getRooms(): ArrayList<Integer> addRoom(int rows, int columns): void getNextRoomID(): int</pre>

67.ábra

Az AdminDatabase osztály tartalmazza azokat a metódusokat, amik csak az adminisztrátorok számára érhetők el.

hasReservation() metódus megadja, hogy egy adott filmhez tartozik-e jövőbeli foglalás.

getNextRoomID() metódus a következő felvehető terem sorszámát adja meg.

UserDatabase

UserDatabase
timeStamp: String
<pre> UserDatabase() isAdmin(String userName, String password): boolean validUser(String userName, String password): int newUser(String name, String userName, String email, String password): void validRegistration(String userName, String email): boolean getRoomID(String title, String date): int getRows(int roomID): int getColumns(int roomID): int addSeats(String reservationNumber, ArrayList<Integer> seats): void addReservation(String reservationNumber, String userName, String title, String showDate, int roomID, int price, ArrayList<Integer> seats): void getReservedSeats(String date, String time, int roomID): ArrayList<Integer> getReservationNumber(String date, String time, int roomID): ArrayList<String> getReservationCounter(String userName): int setReservationCounter(String userName): void getEmail(String userName): String reservationSearchByNumber(String reservationNumber, String userName): ArrayList<ArrayList<String>> reservationSearchByMovie(String title, String userName): ArrayList<ArrayList<String>> reservationSearchByNumberAndMovie(String reservationNumber, String title, String userName): ArrayList<ArrayList<String>> reservationSearchAll(): ArrayList<ArrayList<String>> newEmailCheck(String email, String userName): boolean updateUser(String userName, String email, String password): boolean getShowTitles(): ArrayList<String> addComment(String title, String user, String comment, int rate): void </pre>

68.ábra

A UserDatabase osztály tartalmazza az összes, felhasználók által használt metódust, ami az adatbázissal kapcsolatos.

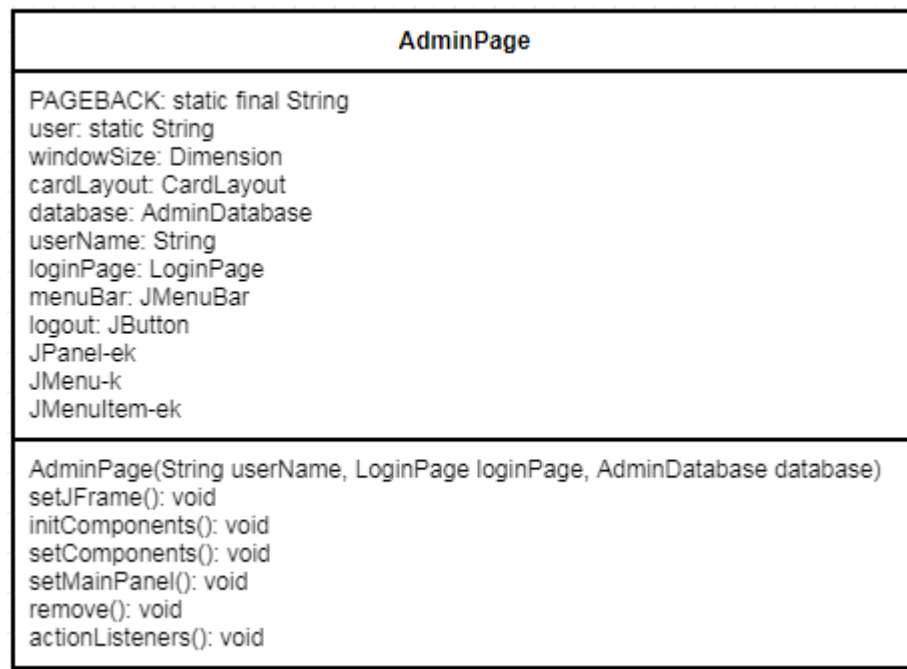
isAdmin() metódus megadja, hogy az adott felhasználó adminisztrátor-e vagy sem.

validUser() metódus a bejelentkezéskor használatos → ha hibásak a megadott adatok, akkor -1-el tér vissza, különben a visszatérési érték 1 (ha adminisztrátor) vagy 2 (ha felhasználó).

validRegistration() ellenőrzi, hogy a megadott email cím vagy felhasználónév foglalt-e már.

getShowTitles() visszaadja azoknak a filmeknek a címét, amelyekhez van kiírva jövőbeli előadás.

AdminPage



69.ábra

Az AdminPage osztály (JFrame leszármazottja) valósítja meg a főoldalt adminisztrátori bejelentkezés esetén.

A *user* bejelentkezett adminisztrátor felhasználónevét tárolja.

setJFrame() beállítja a JFrame paramétereit.

initComponents() inicializálja az adattagokat.

setComponents() beállítja, hogy hogyan is nézzen ki az oldal.

setMainPanel() metódus a főoldalra röpít minket vissza.

actionListeners() metódusban van az összes ActionListener, amelyeket a gombokhoz rendelünk.

A **UserPage** osztály feladata hasonló, csak azt nem adminisztrátori, hanem felhasználói bejelentkezéskor használjuk

A továbbiakban csak azokról az osztályokról csináltam **UML** diagramot, amelyek lényeges eltérnek a többitől. A legtöbb osztály az előző(AdminPage) osztály szerkezetéhez hasonló a metódusok és adattagok tekintetében, persze mindig van némi eltérés, de azokat ki fogom emelni a továbbiakban, s leírom, hogy milyen feladatot lát el az adott metódus.

NewMovie

A NewMovie osztály (MainPanel leszármazottja) valósítja meg azt az oldalt, ahol új filmeket tudunk felvenni.

isNumber() metódus eldönti az adott értékről, hogy szám-e.

NewShow

A NewShow osztály (MainPanel leszármazottja) az új előadás kiírásáért felelős oldalt valósítja meg.

fillComboBox() metódus feltölti értékekkel az oldalon található JComboBox-okat.

MovieSearch

Movie Search
adminPage: AdminPage database: AdminDatabase dimension: Dimension movieList: ArrayList<ArrayList<String>> movieIterator: ListIterator<ArrayList<String>> nextItem: ArrayList<String> previousItem: ArrayList<String> JPanel-ek JLabel-ök JTextField-ek JComboBox-ok JButton-ök
MovieSearch(AdminPage adminPage, Dimension dimesnion, AdminDatabase database) setJPanel(): void initComponents(): void setComponents(): void setDetailFields(String movieTitle, String description, String genre, String runningTime, String director, String characters, String releaseDate, String language, String dubbed, String ageLimit, String rate): void movieRefreshAll(): void movieRefreshByTitle(String title): void movieRefreshByGenre(String genre): void movieRefreshByTitleAndGenre(String title, String genre): void setPicture(ResultSet rs): void actionListeners(): void

70.ábra

A MovieSearch oldal a film-kereső oldal kinézetéért felelős.

setDetailFields() metódus az oldalon található JTextField-ek értékét állítja be.

movieRefreshAll() feltölti a filmeket tartalmazó listát az összes filmmel.

movieRefreshByTitle() csak azokkal a filmekkel tölti fel a listát, amelyeknek címében szerepel a paraméterként megadott érték.

movieRefreshByGenre() csak azokkal a filmekkel tölti fel a listát, amelyek a megadott kategóriába tartoznak.

movieRefreshByTitleAndGenre() csak azokkal a filmekkel tölti fel a listát, amelyeknek címében szerepel a paraméterként megadott érték és a megadott kategóriába tartoznak.

A **MovieUserSearch** osztály (MainPanel leszármazottja) hasonló felépítésű, azzal a különbséggel, hogy kiegészül egy új funkcióval (új vélemény), egy másik(törlés) funkció pedig eltűnik.

A **UserSearch**, **ReservationSearch**, **OwnReservations** (MainPanel leszármazottjai) osztályok felépítése megegyezik ennek az osztálynak a felépítésével, csak ott más feltételek szerint tudunk keresni a felhasználók/foglalások között.

Comments

A Comments osztály (MainPanel leszármazottja) a felhasználói vélemények olvasására alkalmas oldalt valósítja meg.

Rooms

A Rooms osztály (MainPanel leszármazottja) segítségével tudunk új mozitermet felvenni a sorok és oszlopok megadásával.

isNumber() metódus megadja, hogy a sorok/oszlopok számának megadott érték szám-e.

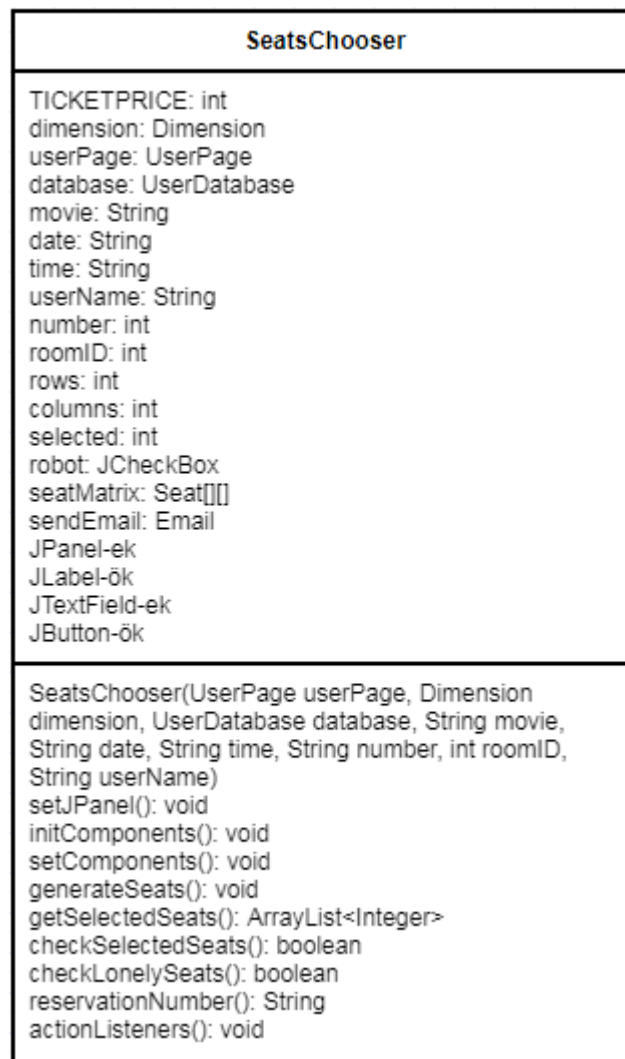
NewReservation

A NewReservation osztály (MainPanel leszármazottja) által megvalósított oldalon tudunk új foglalást indítani.

Az *isNumber()* metódus ellenőrzi, hogy a megadott érték szám-e.

A *setPicture()* az oldalon található borítóképet állítja be a kiválasztott film borítóképére.

SeatsChooser



71.ábra

Új foglalás indításakor a megfelelő előadás kiválasztása után az üléshelyek kiválasztása következik. Ezt teszi lehetővé ez az osztály, amely a MainPanel leszármazottja.

TICKETPRICE az egy konstans érték, amely tárolja a mozijegy árát.

A *seatMatrix* tárolja az üléshelyeket.

A *generateSeats()* metódus legenerálja az üléshelyeket a sorok/oszlopok számának megfelelően.

getSelectedSeats() visszaadja a az általunk kiválasztott székeket.

checkSelectedSeats() ellenőrzi foglalás előtt, hogy a kiválasztott üléshelyek még szabadok-e.

checkLonelySeats() ellenőrzi, hogy nem maradt-e a kiválasztás után magányosan hagyott üléshely.

reservationNumber() metódus legenerálja a foglalási számot.

OwnDetails

Ezzel az osztállyal (MainPanel leszármazottja) tudjuk megváltoztatni a saját adatainkat (email, jelszó).

setEditable() metódus bizonyos mezőket szerkeszthetővé vagy nem szerkeszthetővé állít (a paraméterben megadott boolean érték alapján).

setDetailFields() beállítja az oldalon található mezők értékeit az adott felhasználó értékeire.

emailMatch() metódus ellenőrzi, hogy a megadott email-cím formátuma megfelel-e egy előre megadott sablonnak¹⁰.

Start

A Start osztály feladata a program indítása.

RegistrationPage

A regisztráló oldal (JFrame leszármazottja) kinézetét és logikáját valósítja meg.

emailMatch() metódusa ellenőrzi, hogy a regisztrálni kívánt email-cím formátuma megfelel-e az oldal alján található sablonnak.

LoginPage

A LoginPage osztály/oldal (JFrame leszármazottja) a felhasználók és adminisztrátorok számára is egyértelműen elérhető. A bejelentkező oldal kinézetéért és megfelelő működéséért felelős.

MainPanel

A MainPanel osztály a JPanel leszármazottja, és összesen két metódusa van.

felülírja a JPanel *paintComponent()* metódusát, és beállítja a háttérét a *setPictureName()* paramétereként megadott képre.

Encrypt

Az Encrypt osztály egyetlen metódussal rendelkezik.

public static String toMD5(String password) – a paraméterként megadott jelszót titkosítja egyirányú titkosítással.

¹⁰ `^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}$`

Email

Email
from: String userName: String password: String session: Session random: Random confirmationNumber: int
Email() sendRegistrationEmail(String name, String to): void sendReservationEmail(String userName, String movie, int roomID, int number, String date, String time, String seats, String reservationNumber, int price, String to): void sendUserDeleteEmail(String name, String reason, String to): void setProperties(): void getConfirmationNumber(): String

72.ábra

Az Email osztály állítja össze és küldi ki az emaileket regisztrációkor, foglaláskor és felhasználó törlésekor.

A *from*(elte.moziverzum kukac gmail.com), *userName*(elte.moziverzum), *password*(eltemozi) beégetett értékek.

A *sendRegistrationEmail()* kiküldi a regisztráló emailt új felhasználó regisztrációjakor.

A *sendReservationEmail()* metódus új, sikeres foglalás esetén küld ki email a foglalás lefontosabb adataival.

A *sendUserDeleteEmail()* akkor küld ki emailt adott felhasználónak, ha valamilyen oknál fogva (pl.: szabályszegés) a felhasználó törlésre kerül.

A *setProperties()* beállítja az email küldéshez szükséges tulajdonságokat (host, port stb.)

PDF

PDF
<pre>titleFont: Font tableHeaderFont: Font confirmationFont: Font haveFunFont: Font cellFont: Font endFont: Font fileName: String name: String userName: String email: String confirmNumber: String movieTitle: String time: String room: String seats: String price: int document: Document pageIcon: Image dialIcon: Image pageTitle: Paragraph end: Paragraph haveFun: Paragraph personalTable: PdfTable movieTable: PdfTable</pre>
<pre>PDF(String name, String userName, String email, String confirmNumber, String movieTitle, String time, String room, String seats, int price) setDocument(): void initComponents(): void setComponents(): void addComponents(Element add): void</pre>

73.ábra

A foglalt mozijegyeket PDF-be is lehet exportálni. Az exportált fájl kinézetét valósítja meg ez az osztály.

A PDF fájl elkészítéséhez **itext**-et használtam, ami könnyedén és érthetően használható.

Az *addComponent()* metódus új elemet ad a fájlunkhoz.

A *setDocument()* létrehozza az új dokumentumot, melynek neve a felhasználónévből és a foglalási számból áll. Majd megnyitja ezt a dokumentumot, hogy írni lehessen bele.

Seat

A Seat osztály a székeket valósítja meg és a JButton leszármazottja.

Egy adattagja van, az egész típusú *seatNumber*, amely tárolja a szék sorszámát.

A *setIcon()* beállítja a szék hátterét.

Table

Table
model: DefaultTableModel table: JTable scrollPane: JScrollPane centerRenderer: DefaultTableCellRenderer
Table() setJFrame createTableForMovies(ArrayList<ArrayList<String>> movies): void createTableForReservations(ArrayList<ArrayList<String>> reservations): void createTableForUsers(ArrayList<ArrayList<String>>): void

74.ábra

A Table osztály valósítja meg a táblázatos nézeteket felhasználók, filmek és foglalások esetén a paraméterként kapott értékek alapján.

Database project

CreateDatabase

Lefuttatja a *Database\Database.java* osztályt, amely létrehozza az adatbázist.

Database

Database
firstUser: String connection: Connection statement: Statement br: BufferedReader cinemaDatabaseName: String dropDatabase: String createCinemaDatabase: String createUsersTable: String createMoviesTable: String createGenresTable: String createReservationsTable: String createShowsTable: String createSeatsTable: String createRoomsTable: String createDirectorsTable: String createAgesTable: String createCommentsTable: String
Database() addGenres(): void addAges(): void getDatabaseDetails(): String

75.ábra

Létrehozza az adatbázist(*cinema*).

addGenres(), *addAges()* metódusok hozzáadják az adatbázishoz a kategóriákat és a korhatárokat. A *getDatabaseDetails()* kiolvassa a szükséges adatbázis információkat a *database.txt* fájlból.

3.5 Tesztelés

3.5.1 Funkcionális tesztek

A program futtatását követően a **bejelentkező** oldal kell, hogy elérhető legyen számunka. Vannak külön *adminisztrátori*, külön *felhasználói* és *közös* funkciók is.

1. Közös funkciók

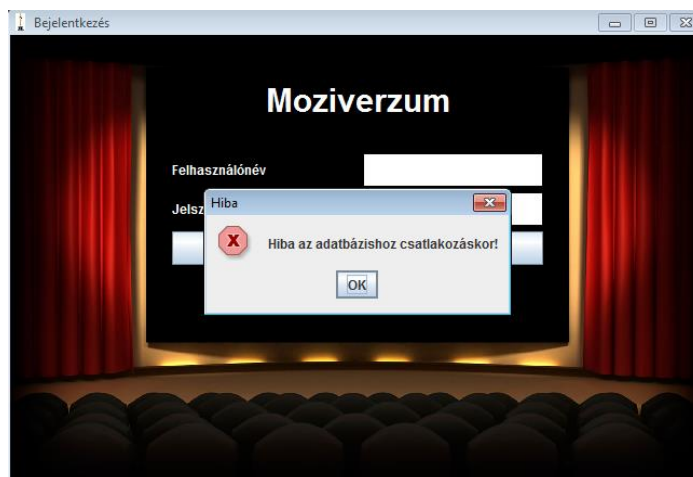
- **Adatbázis adatok megadása fájlban**

A programunk az ip címet, amelyik gép adatbázisához csatlakozik, és a csatlakozáshoz szükséges felhasználónevet és jelszót az *adatbazis.txt* fájlból olvassa ki.

Kezelt esetek:

- üres fájl → hiba
- kevesebb, mint 3 sor → hiba
- 3 sor, de valamelyik üres → hiba
- nincs „:” jel a sorban → hiba
- az első 3 sor helyes, de több sor is van → helyes
- továbbá az alkalmazás tartalomellenőrzést is végez a fájlban, tehát pl.:IP:asd → hiba stb.

Hiba esetén a program hibaüzenetet ad, majd leáll.



76.ábra

- **Bejelentkezés**

A Felhasználó és Jelszó mezők kitöltését követően a Bejelentkezés gomb megnyomásával tudunk bejelentkezni, ha megadott információk helyesek, tehát a beírt felhasználónév létezik és a hozzá tartozó jelszó megegyezik a begépelt jelszóval.

Bejelentkezést követően a bal felső sarokban a logó, ill. az Üdv, <<Felhasználónév>>!(<<szerep>>) olvasható.



77.ábra

- **Kijelentkezés**

Bejelentkezést követően a menüsor jobb oldali végén lévő Kijelentkezés gomb megnyomásával, majd jóváhagyásával tudunk kijelentkezni. Ezt követően a Bejelentkező oldalra kerülünk vissza.

- **Kilépés**

A jobb felső sarokban lévő X megnyomását követően a program bezáródik.

2. Adminisztrátori funkciók

A menüsorunk a következőképpen áll fel:

- Film
 - Új film felvétele
 - Új előadás kiírása
- Keresés/Törlés
 - Foglalások
 - Filmek
 - Felhasználók
- Vélemények
 - Felhasználói vélemények
- Termek
 - Új terem felvétele

- Adatok
 - Saját adatlap

Új film felvétele

78.ábra

Minden mező kitöltése kötelező.

A leírás mező kitöltése a *Kitölt* gomb megnyomását követően lehetséges. A megnyomás után a mezőbe tudjuk beírni a film leírását (max. 400 karakter), majd az *OK* gombbal tudjuk azt elmenteni. Ezt követően a *Kitölt* újbóli megnyomását követően az imént beírt szöveg kell, hogy a mezőben megjelenjen.

A *Főszereplő(k)* mező kitöltése ugyan ezen az elven alapul.

A *Kategória*, *Korhatár*, *Szinkronizált* mezők értékét lenyíló listából tudjuk kiválasztani.

A *Játékidő(perc)*, *Megjelenés éve* mezők értéke szám kell, hogy legyen, ellenkező esetben nem lesz sikeres a filmfelvétel és hibaüzenetet kapunk.

A film borítóképét a *Kép kiválasztása* gombbal tudjuk kiválasztani. Ekkor egy file kiválasztó ablak ugrik fel elénk, ahol ki tudjuk keresni a kívánt képet. Kiválasztás után az *Open* gomb hatására a kiválasztott kép a jobb oldali keretben kell, hogy látható legyen.

Ha a kiválasztott kép mérete nagyobb a megengedettnél, akkor arról egy hibaüzenet értesít minket azonnal.

A *Küld* gomb hatására, ha minden mező megfelelő módon lett kitöltve, akkor sikeresen felvesszük a filmet az adatbázisunkba, melyet egy visszaigazoló üzenet is megerősít. Ellenkező esetben, hibaüzenetet kapunk és a film felvétele sikertelen.

A *Mégse* gombbal a főoldalra kerülünk vissza.

Új előadás kiírása

79.ábra

Minden mező kitöltése kötelező.

A *Filmcím*, *Időpont*, *Teremszám* mezők értékét lenyíló listából tudjuk kiválasztani.

A *Filmcím* lista az adatbázisban található összes filmet tartalmazza.

A *Teremszám* a korábban felvett termeket tartalmazza.

Ezekből következik, hogy új műsort csak akkor tudunk felvenni, ha

- van legalább egy felvett film
- van legalább egy felvett terem

A *Dátum* esetén pedig egy naptárból tudjuk kiválasztani a megfelelő dátumot, amely csak következő napi vagy későbbi lehet.

A *Küld* gomb megnyomásakor az összes feltétel ellenőrzése megtörténik.

Hiba esetén hibaüzenet fogad minket a hibáról és az előadás felvétele sikertelen.

Ellenkező esetben pedig az előadás sikeresen felvételre kerül.

A *Mégse* gombbal a főoldalra kerülünk vissza.

Foglalások keresése/törlése

80.ábra

Kezdetben csak a *Keres*, *Mégse*, *Táblázat* és *Foglalás törlése* gombok érhetők el.

Két feltétel alapján tudunk keresni:

- Foglalási szám
- Felhasználónév

Ha a két mező értéke üres, akkor a *Keres* gomb megnyomását követően az összes foglalás kilistázódik, és az első foglalás adatai töltődnek be a megfelelő mezőkbe. Ha nincs egyetlen foglalás sem, akkor üzenet értesít minket róla.

Csak a *Foglalási szám* megadása esetén, a megfelelő foglalást kapjuk meg, ha létezik, ellenkező esetben pedig hibaüzenetet kapunk.

Csak *Felhasználónév* megadása esetén az összes foglalást megkapjuk, ami az adott felhasználóhoz tartozik, ha létezik, ellenkező esetben hibaüzenetet kapunk.

Foglalási szám és *Felhasználónév* megadásakor megkapjuk azt a foglalást, ami az adott felhasználóhoz tartozik és amelynek száma megegyezik a keresett számmal, ha van ilyen, ellenkező esetben hibaüzenet fogad minket.

Ha a keresést követően egynél több foglalást kapunk, akkor a *Következő foglalás* gomb elérhetővé válik, majd a megnyomását követően a következő foglalás adatai kell, hogy betöltődjenek a megfelelő mezőkbe, és az *Előző foglalás* gomb elérhetővé kell, hogy váljon, amellyel visszafele tudunk lépkedni a foglalások között.

Ha van keresett foglalás, akkor a *PDF* gomb hatására az adott foglalás egy PDF fájlba exportálódik, amely tartalmazza a foglalás fontos adatait.

A *Táblázat* gomb hatására egy táblázat ugrik fel elénk, amely a keresési feltételeknek megfelelő foglalásokat tartalmazza.

Ha nincs foglalás kiválasztva, akkor a *Foglalás törlése* gomb egy hibaüzenettel tájékoztat minket, hogy a törléshez ki kell választani egy foglalást.

Ellenkező esetben pedig a jóváhagyást követően a foglalást sikeresen törli.

A *Mégse* gombbal a főoldalra kerülünk vissza.

Felhasználók keresése törlése

81.ábra

A Felhasználók keresése/törlése oldal ugyanúgy működik, mint az előző (foglalások keresése/törlése) oldal, azzal a különbséggel, hogy itt felhasználókra tudunk keresni *Felhasználónév* és *Email cím* alapján.

A gombok elvárt működése megegyezik az előző oldaléval, itt viszont értelemszerűen nincs PDF-be exportálás.

Felhasználó törlése esetén meg kell adnunk a törlés okát is, ezután pedig a törölt felhasználó emailben kap értesítést a törlésről és annak okáról. Más adminisztrátorokat és saját magunkat nem tudjuk törölni.

elte.moziverzum@gmail.com - Törölt account

Kedves Prumcsi!

Tájékoztatjuk, hogy accountja és minden eddigi foglalása a mai napon törlése került!
A törlés oka: Szabálysértés

További sok sikert!
Moziverzum csapata

82.ábra

Filmek keresése/törlése

83.ábra

A gombok működése itt is ugyanaz, mint az előző két oldalon, azonban ez az oldal még kiegészül pár funkcióval.

Filmre *Filmcím* és *Kategória* szerint tudunk keresni.

Filmcím alapján való keresés esetén elegendő a címnek csak egy részét beírni, s ekkor az összes olyan filmet megkapjuk, amelynek címében a keresett részlet előfordul.

Kategória szerint való keresés esetén egy lenyíló listából tudjuk kiválasztani a keresett kategóriát.

Az oldalon 3 *Mutat* gomb található:

- Leírás
- Előadások
- Főszereplők

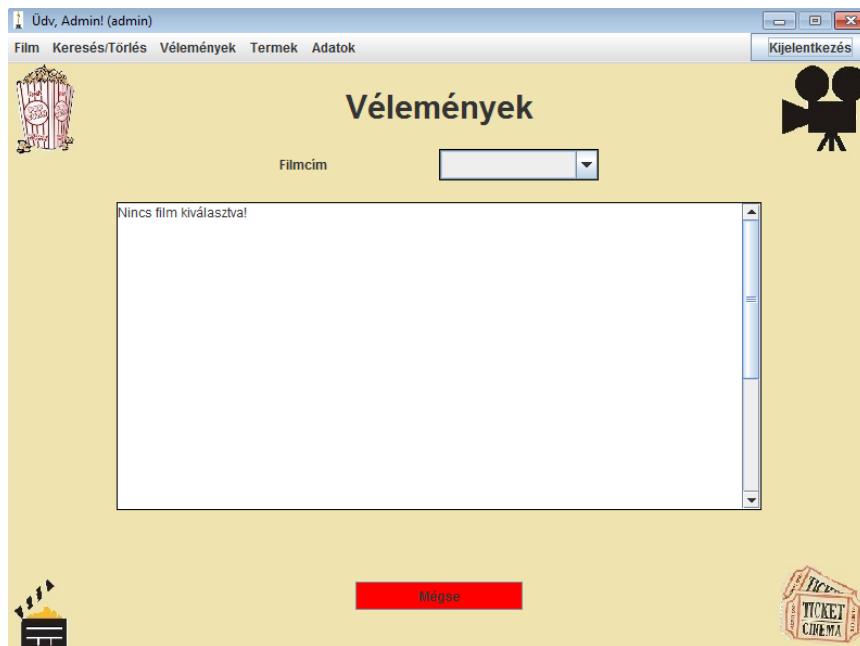
Leírás esetén a *Mutat* gomb hatására egy ablak ugrik fel, amely tartalmazza a film leírását. Előadások esetén a *Mutat* gomb hatására a filmhez kiírt előadásokat láthatjuk időrendi sorrendben dátumok szerint csoportosítva.

Főszereplők esetén a *Mutat* gomb hatására a főszereplőket olvashatjuk egy felugró ablakban.

Keresések során a keresett film adatai a megfelelő mezőkbe töltődnek be.

Előző film és *Következő film* esetén pedig a filmeket tartalmazó lista előző/következő elemének adatai töltődnek be a mezőkbe.

Felhasználói vélemények



84.ábra

A Felhasználói vélemények menüpont kiválasztását követően a *Vélemények* oldalra kerülünk.

Alap helyzetben a „*Nincs film kiválasztva*” üzenetet olvashatjuk középen a mezőben.

A keresett film címét a *Filmcím* mező lenyíló listájából tudjuk kiválasztani.

Ha van vélemény írva a kiválasztott filmhez, akkor azokat középen, a mezőben olvashatjuk ----- vonallal elválasztva. Továbbá a *Mégse* gomb felett a film értékelését olvashatjuk, zárójelben a vélemények számával.

Ha még nincsenek megjegyzések írva a kiválasztott filmről, akkor a „*Nincsenek még kommentek ehhez a filmhez!*” üzenetet olvashatjuk. Ekkor a *Mégse* gomb felett az *Értékelés: 0 (0)* olvasható.

A *Mégse* gombbal a főoldalra kerülünk vissza.

Új terem felvétele

Üdv, Admin! (admin)

Film Keresés/Törlés Vélemények Termek Adatok Kijelentkezés

Új terem felvétele

Teremszám

Sorok száma*

Oszlopok száma*

85.ábra

Minden mező kitöltése kötelező.

A *Teremszám* mező értéke mindig a soron következő teremszám.

A *Sorok/Oszlopok száma* esetén van pár feltétel

- szám kell, hogy legyen
- ≥ 0 ≤ 10 kell, hogy legyen

Ellenkező esetben hibaüzenet fogad minket.

A *Küld* gomb hatására ezek a feltételek ellenőrzésre kerülnek, és ha nincs hiba, akkor a terem felvétele sikeres.

A *Mégse* gombbal a főoldalra kerülünk vissza.

Saját adatlap

Azonosító	1
Név	Admin Admin
Felhasználónév	Admin
Email cím*	zsoltitscher@gmail.com
Jelszó*
Jelszó ismét*	
Státusz	admin
Regisztráció időpontja	2000-01-01 00:00:00.0

[Mégse](#) [Adatok módosítása](#)

86.ábra

Középen az fiókhoz tartozó információkat láthatjuk.

Az *Adatok módosítása* gomb hatására az *Email cím*, *Jelszó* és *Jelszó ismét* mezők keretei pirosra változnak, ezzel jelezve, hogy ezeket a mezőket lehet megváltoztatni. Minden mező kitöltése kötelező a sikeres adاتمódosításhoz.

Email cím megváltoztatása csak akkor lehetséges, ha az email cím, amelyre meg akarjuk változtatni, az még nincsen beregisztrálva egy másik felhasználó által, tehát szabad.

Jelszó megváltoztatása csak akkor lesz sikeres, ha a *Jelszó* és *Jelszó ismét* mezők értékei megegyeznek.

A *Módosít* gomb megnyomásával ezek a feltételek ellenőrzésre kerülnek és ha minden megfelelő, akkor az adاتمódosítás sikeres, ellenkező esetben pedig hibaüzenet fogad minket és nem változik meg egyik adat sem.

A *Mégse* gombbal a főoldalra kerülünk vissza.

3. Felhasználói funkciók

A menüsorunk a következőképpen áll fel:

- Foglalások
 - Új foglalás
 - Saját foglalások
- Filmek
 - Keresés
- Adatok
 - Saját adatlap
- Vélemények
 - Felhasználói vélemények

Regisztráció



87.ábra

Minden mező kitöltése kötelező.

Az *E-mail cím* formátuma ellenőrzött, tehát regisztrálni csak a mintának (pl.:valami123@valami.com) megfelelő email címmel lehet, viszont csak létező email címmel érdemes, hiszen a *Regisztráció* gomb megnyomását követően a megadott email címre egy visszaigazoló email érkezik, amely tartalmaz egy visszaigazoló kódot, amelyet meg kell adni a sikeres regisztráció befejezéséhez.



88.ábra

elte.moziverzum@gmail.com - Visszaigazoló email

Kedves Próba István, üdvözlünk a Moziverzum világában!

Köszönjük, hogy regisztráltál, reméljük jól fogod érezni magad a filmek világában!
Visszaigazoló kód: 2325

Köszönettel, a Moziverzum csapata!

89.ábra

A *Jelszó* és *Jelszó újra* mezők értékeinek meg kell egyezniük.

A *Felhasználónév* és *E-mail cím* pedig egyediek kell, hogy legyenek.

A *Regisztráció* gomb megnyomásakor ezek a feltételek ellenőrzésre kerülnek és hiba esetén hibaüzenet fogad minket.

A *Mégse* gombbal visszakerülünk a bejelentkező felületre.

Új foglalás

90.ábra

Értelemszerűen itt is minden mező kitöltése kötelező.

A *Filmcím*, *Dátum*, *Időpont* mezők értékei egy lenyíló listából tudjuk kiválasztani.

A *Filmcímmel* kell kezdenünk. A *Filmcím* lista azokat a filmeket tartalmazza, amelyekhez van kiírva jövőbeli előadás.

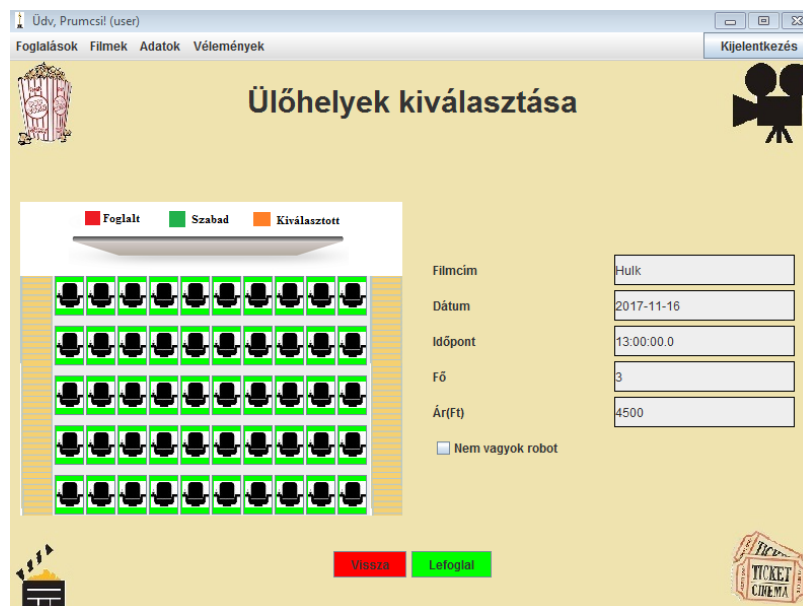
A film kiválasztását követően a jobb oldali keretben a film borítóképe olvasható, ill. a *Dátum* lista feltöltődik olyan dátumokkal, amelyek az adott filmhez tartozó előadásokhoz tartoznak és jövőbeliek.

A dátum kiválasztását követően az *Időpont* lista feltöltődik azokkal az időpontokkal, amelyek az adott film adott dátumához tartoznak és jövőbeliek!

A *Fő* megadása szám és nagyobb 0 kell, hogy legyen.

A *Mégse* gombbal a főoldalra kerülünk vissza.

A *Tovább* gombbal, ha minden mezőt sikeresen kitöltöttünk, akkor az ülőhelyek kiválasztása oldalra kerülünk.



91.ábra

Jobb oldalt láthatjuk a kiválasztott előadás adatait.

A szabad székek zölddel, a lefoglalt székek pirossal, a kiválasztott székek pedig narancssárga színnel jelennek meg.

Csak annyi ülőhelyet tudunk kiválasztani, amennyit az előző oldalon megadtunk.

A foglalt székeket nem tudjuk kiválasztani. A kiválasztott széket pedig újbóli rákattintással tudjuk szabaddá tenni.

A székek kiválasztásakor figyelembe kell venni, hogy sor elején és végén, ill. két kiválasztott/lefoglalt ülőhely között nem maradhat magányos ülés.

Foglalás előtt a „*Nem vagyok robot*” kipipálása is kötelező, biztonsági okokból.

A *Lefoglal* gomb megnyomása esetén az összes feltétel ellenőrzése megtörténik és ha minden megfelelő, akkor sikeres a foglalás, amelyet egy visszaigazoló üzenet és email erősít meg, amelyek tartalmazzák a foglalási számot, mellyel a foglalásunkra tudunk rákeresni.

Kedves Prumcsi, köszönjük a foglalását!

Foglalási szám: 8DCMO4OM36
Felhasználónév: Prumcsi
Filmcím: Hulk
Dátum: 2017-11-16
Időpont: 13:00:00.0
Terem: 1
Fő: 3
Székek: 4,5,6
Ár: 4500Ft

Jó szórakozást kíván, a Moziverzum csapata!

92.ábra

A *Vissza* gombbal az előző oldalra kerülünk vissza, ahol megváltoztathatjuk a fő-k számát vagy akár előadást is válthatunk.

Saját foglalások

Ez az oldal ugyanúgy működik, mint az adminisztrátoroknál a Foglalások keresése/törlése oldal, azzal a különbséggel, hogy itt minden kereséskor csak a saját foglalásaink közül kapjuk meg a feltételeknek megfelelő foglalást, nem pedig az összes foglalás közül.

Emiatt különbség még az is, hogy itt *Felhasználónév* helyett, *Filmcím* alapján tudunk keresni a *Foglalási szám* mellett. A filmcím megadásakor nem kell az egész filmcímet beírunk, elég csak egy részét, s ekkor az összes olyan foglalást megkapjuk, amely olyan filmre szól, amelynek címében szerepel a megadott szórészlet.

Filmek keresése

A Filmek keresése oldal ugyanúgy működik, mint az adminisztrátoroknál a Filmek keresése/törlése oldal, azzal a különbséggel, hogy felhasználóként nem tudunk filmet törölni, viszont egy új funkcióval egészül ki az oldal, az *Új vélemény*-nyel.

Ezzel a funkcióval új véleményt tudunk írni a filmhez.

Az *Új vélemény* megnyomására egy ablak ugrik fel, amelyben minden mező kitöltése kötelező.

Először magát a véleményt kell leírunk, amely max. 400 karakter lehet, utána pedig egy 1-10-es skálán kell a filmet pontoznunk, majd az *OK* gombbal tudjuk a véleményt elküldeni, melynek sikerességét egy igazoló üzenet erősít meg. Ezt követően a véleményezett film átlaga megváltozik.

Saját adatlap

93.ábra

Középen az accountunkhoz tartozó információkat láthatjuk.

Az *Adatok módosítása* gomb hatására az *Email cím*, *Jelszó* és *Jelszó ismét* mezők keretei pirosra változnak, ezzel jelezve, hogy ezeket a mezőket lehet megváltoztatni. Minden mező kitöltése kötelező a sikeres adatomódosításhoz.

Email cím megváltoztatása csak akkor lehetséges, ha az email cím, amelyre meg akarjuk változtatni, az még nincsen beregisztrálva egy másik felhasználó által, tehát szabad.

Jelszó megváltoztatása csak akkor lesz sikeres, ha a *Jelszó* és *Jelszó ismét* mezők értékei megegyeznek.

A *Módosít* gomb megnyomásával ezek a feltételek ellenőrzésre kerülnek és ha minden megfelelő, akkor az adatomódosítás sikeres, ellenkező esetben pedig hibaüzenet fogad minket és nem változik meg egyik adat sem.

A *Mégse* gomb csak az adatomódosítás szakaszban látható, ha ezt a funkciót használjuk, akkor minden mező értéke visszaáll az eredeti értékre.

A *Mégse* gombbal a főoldalra kerülünk vissza.

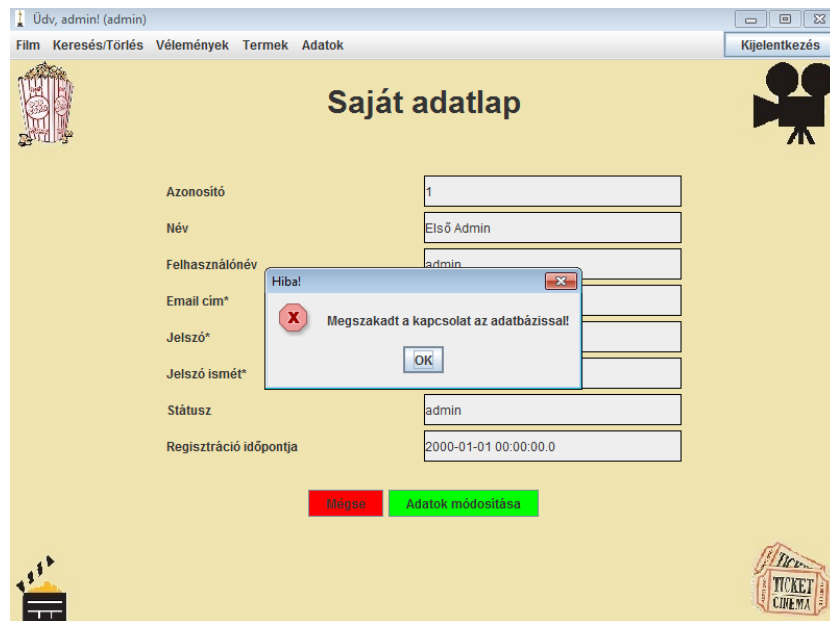
Felhasználói vélemények

Ez az oldal egy az egyben úgy működik minden funkciójával, mint az adminisztrátorok esetében lévő *Vélemények* oldal.

3.5.2 Távoli felhasználás tesztelése

Az alkalmazást lehet használni saját gépen levő, ill. egy másik gépen levő adatbázissal is. Mindkét esetben a program működése teljesen megegyezik.

Azonban, ha egy távoli gép adatbázisát használjuk, akkor fennáll annak a kockázata, hogy az adatbázissal való kapcsolat megszakad. Ekkor az alkalmazás hibaüzenettel jelzi ezt, majd leáll.



94.ábra

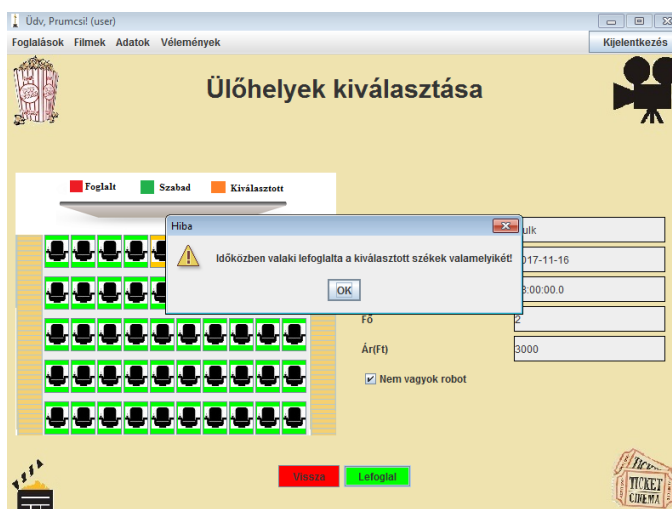
3.5.3 Párhuzamos felhasználás tesztelése

A programot egyszerre többen is képesek használni. Ebből kifolyólag felmerülhet pár probléma, amit kezelni kell.

Kezelt esetek:

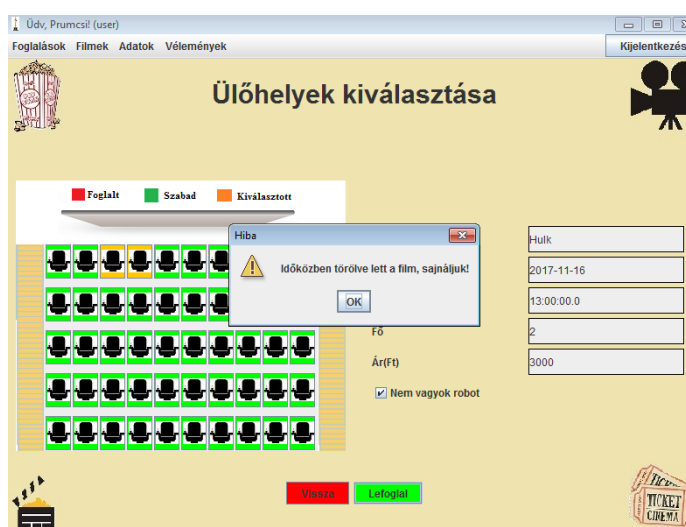
1. Az az eset, amikor 2 vagy több felhasználó egy időben próbál helyet foglalni ugyanarra az előadásra és legalább két felhasználó ugyanazokat az üléseket szeretné lefoglalni. Ha ilyen eset merülne fel, akkor hibaüzenet értesít minket arról, ha valaki lefoglalta már időközben a kiválasztott székeket.

Minden esetben az tudja lefoglalni a székeket, aki korábban nyomott rá a „Lefoglal” gombra.



95.ábra

2. Egy másik eset, amikor egy felhasználó már az ülőhelyek kiválasztásánál jár, de időközben egy adminisztrátor kitörölte azt a filmet, amire a felhasználó éppen az ülőhelyeket választja ki. Ekkor hibaüzenet értesíti a felhasználót erről.



96.ábra

3. Egymással párhuzamosan két különböző filmre is lehet foglalni.
4. Amíg egy felhasználó éppen helyeket foglal, azalatt az adminisztrátor képes új film felvételére.
5. Két adminisztrátor képes egyszerre felvinni két különböző filmet. Ha két ugyanolyan című filmet szeretnének felvinni, akkor annak az adminisztrátornak a filmfelvétele lesz sikeres, amelyik előbb rányomott a „Küld” gombra. Ekkor a másik adminisztrátort hiba értesíti arról, hogy létezik már film olyan címmel.

Illetve, ha két adminisztrátor egyszerre akar előadást kiírni, akkor is van néhány kérdéses eset, amelyek mind kezelve lettek:

- ugyanarra a filmre, ugyanarra az időpontra, ugyanabba a terembe → hiba
- ugyanarra a filmre, ugyanarra az időpontra, de különböző terembe → hiba
- különböző filmekre, azonos időpontban, azonos terembe → hiba
- különböző filmekre, azonos időpontban, de különböző termekben → helyes

Mindig annak az adminisztrátornak az előadása lesz kiírva, amely előbb nyomott rá a „Küld” gombra, s a másik adminisztrátor hibaüzenetet kap.

6. Illetve az az eset is tesztelve lett, amikor két felhasználó ugyanarra az előadásra foglal különböző ülőhelyeket. Ez minden akadály nélkül működik, tehát ekkor a program zavartalanul fut tovább.

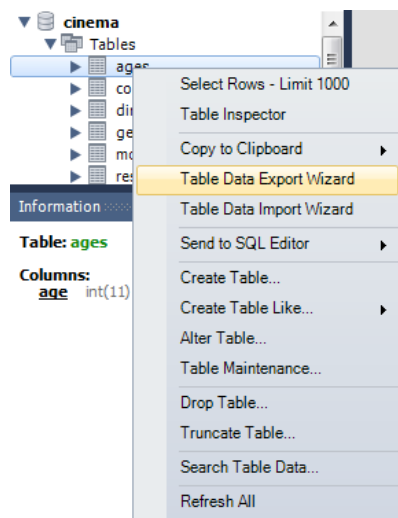
3.5.4 Unit tesztek

A Unit tesztek során a *Database*, *AdminDatabase* és *UserDatabase* osztályok metódusait teszteltem, tehát azt, hogy mindig csak az kerül az adatbázisba, az kerül ki az adatbázisból és az változik meg az adatbázisban, aminek kell.

A teszt fájlok a *Cinema\test\Tests* mappában érhetők el.

Fontos, hogy a tesztelés előtt (adatbázis törlése majd újbóli létrehozása) az adatokat magunknak kell lementeni az adatbázisból, különben elvesznek!

Egy lehetséges megoldás, hogy nyissuk meg a MySQL Workbench-et, amely a MySQL grafikus felhasználói felülete (GUI), majd keressük ki az adatbázisunkat (jelen esetben *cinema*), és táblánként exportáljuk ki őket:



97.ábra

Kattintsunk a **Table Data Export Wizard** gombra (mikor az adatokat vissza akarjuk importálni, akkor természetesen a Table Data Import Wizard-ra), válasszuk ki az exportálni kívánt oszlopokat, majd adjuk meg a helyet, ahova az exportált fájl kerülni fog.

A tesztek futtatása nagyon egyszerű, azonban van három feltétel, ami elengedhetetlen a sikeres tesztelés érdekében:

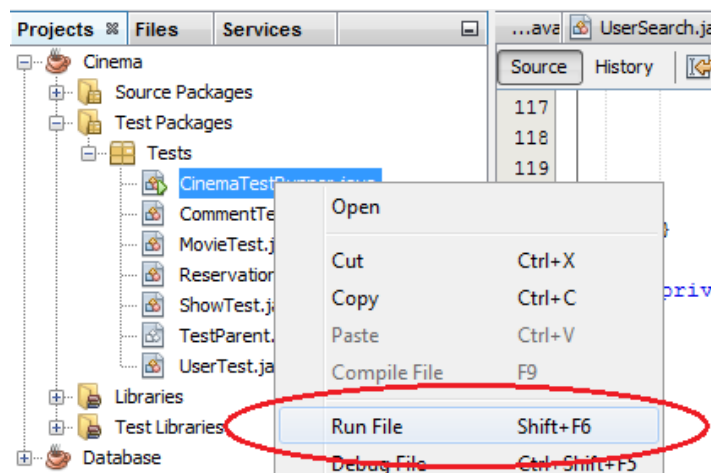
- újonnan létrehozott adatbázison kell tesztelni
- az adatbázis üres kell, hogy legyen
- a *Database* project *Database.java* file-ban kommenteljük ki azt a sort¹¹, amelyben létrehozuk az első adminisztrátort, az üres adatbázis érdekében, majd hozzuk létre az adatbázist újra a *Database\CreateDatabase.java* file futtatásával.

Tehát tesztelés előtt, ha az adatbázisunk üres, de korábban már voltak benne adatok, akkor is kikell törölnünk az adatbázist és újra létrehozni az AUTO_INCREMENT rekordok miatt.

Most már megkezdhetjük a tesztelést.

A *Cinema\Test Packages\Tests* mappa tartalmaz egy *CinemaTestRunner.java* fájlt, amely lefuttatja az összes *Test* osztályunkat.

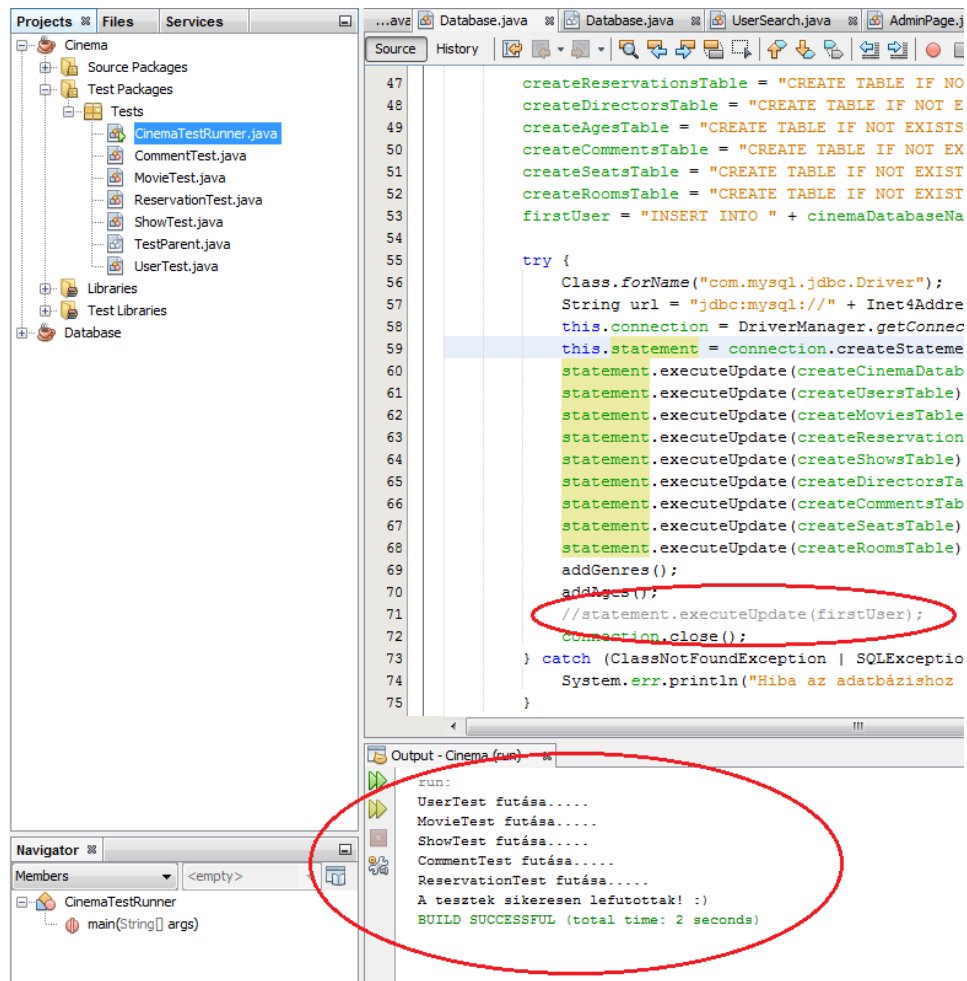
Futtatás:



98.ábra

¹¹ statement.executeUpdate(firstUser);

Sikeres tesztelés esetén a következőt kell, hogy lássuk az Output-on:



99.ábra

Ha a tesztek sikertelenül futnak le, akkor a „*A tesztek sikeresen lefutottak!*” üzenet helyett a „*A tesztek futása sikertelen volt!*” üzenet fogad minket a felbukkanó hibákkal és azok számával.

Összesen **5** Test osztályunk van, amelyet a tesztelés során lefuttatunk:

1. **UserTest** – A felhasználókkal kapcsolatos metódusok tesztelése.
2. **MovieTest** – A filmekkel kapcsolatos metódusok tesztelése.
3. **ShowTest** – Az előadásokkal kapcsolatos metódusok tesztelése.
4. **CommentTest** – A felhasználói véleményekkel kapcsolatos metódusok tesztelése.
5. **ReservationTest** – A foglalásokkal kapcsolat metódusok tesztelése.

Minden Test osztályunkra érvényesek a következők:

- a TestParent osztály leszármazottja, amely csatlakozik az adatbázishoz.

- tartalmaz egy visszatérési érték nélküli `setUp()` metódust, ami `@BeforeClass` annotációval van ellátva, emiatt statikus a metódus. Az olyan metódus, ami `@BeforeClass` annotációval van ellátva, az osztály lefutásának legelején fut le, de csak egyszer (a `@Before` minden `@Test` előtt lefutna). Jelen esetben ennek a metódusnak az a dolga, hogy kiírja, hogy éppen melyik teszt osztályunk fut.
- tartalmaz egy visszatérési érték nélküli, `@Test` annotációval ellátott `tests()` metódust, ami lefuttatja az osztályon belül levő tesztelő metódusinkat. Ez az egyetlen `@Test` annotációval ellátott metódus az osztályban.

1. UserTest

A *UserTest* osztályunk teszteli az összes felhasználóval kapcsolatos metódust.

addUserIfNotExistsTest() – teszteljük a `validRegistration()` metódust, amely megadja, hogy egy felhasználó adatai jók-e vagy sem (true-t kell kapnunk). Ezután a `newUser()` metódus tesztelésével hozzáadunk 2 felhasználót az adatbázisunkhoz. Leellenőrizzük, hogy a méret 2-re változott-e, ill., hogy a felhasználók a megfelelő adatokkal lettek beszúrva.

deleteUserIfExistsTest() – a `userDelete()` metódus tesztelésével kitöröljük az első felhasználót. Ezután megnézzük, hogy tényleg törölve lett-e, és, hogy csak az az egy felhasználó került törlése, nem pedig az összes.

addUserIfExistsTest() – ismét a `validRegistration()` metódust teszteljük, de olyan felhasználónévvel és email címmel, ami már a rendszerben van (false-t kell kapnunk). Ezután a `newUser()` metódussal megpróbálunk felvenni felhasználókat olyan email címmel és felhasználónévvel, amelyek már léteznek. Ekkor `SQLException`-t fogunk kapni a `UNIQUE` adattagok miatt, de ezt kezeljük a `tests()` metódusnál egy (`expected = SQLException.class`) sorral a `@Test` annotáció után.

Ezután ellenőrizzük, hogy tényleg nem szűrt be új sorokat, tehát a méret maradt 2, ill. a már létező sorok értékét sem írta át.

validUserTest() – ebben a metódusban a `validUser()` metódusunkat teszteljük, ami a bejelentkezéskor hívódik meg, s azt dönti el, hogy a megadott felhasználónév és jelszó létezik-e és, hogy összetartoznak-e.

Több esetet is tesztelek:

- létező felhasználónév és helyes jelszó hozzá.

- nem létező felhasználónév.
- létező felhasználónév, de hibás a jelszó hozzá.
- nem létező felhasználónév és jelszó.

userSearchByUserNameTest() – a felhasználók közötti keresés oldal metódusa, amelyben a *userSearchByUserName()* metódust teszteljük. Ez a metódus felhasználónév alapján keres a felhasználók között.

Két esetet nézünk:

- létező felhasználóra keresünk rá
- nem létező felhasználóra keresünk rá

userSearchByEmailTest() – a *userSearchByEmail()* metódust teszteljük, amely a felhasználók között keres email cím alapján.

Két esetet nézünk:

- létező email címre keresünk
- nem létező email címre keresünk

userSearchByUserNameAndEmailTest() – a *userSearchByUserNameAndEmail()* metódust teszteljük, ami a felhasználók között keres felhasználónév és email cím alapján egyszerre.

Több esetet is megnézünk:

- létező felhasználónév és hozzá tartozó email cím → *assertNull()*
- létező felhasználónév de nem hozzátartozó email cím → *assertNull()*
- nem létező felhasználónév de létező email cím → *assertNull()*
- nem létező felhasználónév és nem létező email cím → *assertNull()*

userSearchAllTest() – a *userSearchAll()* metódust teszteljük, ami az összes felhasználóval tér vissza. Megnézzük, hogy a meghívását követően az összes (jelen esetben 2) felhasználót megkapjuk-e a megfelelő adatokkal.

updateUserTest() – a saját adatok módosításakor meghívódó *updateUser()* metódust teszteljük, ami *false*-al tér vissza, ha a megadott email cím már foglalt. Ha *true*, akkor megváltoztatja az email címet és jelszót a megadottakra.

Tesztelt esetek:

- csak a jelszót változtatjuk meg. Megnézzük, hogy tényleg átírta-e.

- megpróbáljuk olyan email címre változtatni, amit már más valaki használ. Itt megnézzük azt is, hogy tényleg nem írta-e át se az email címet, se a jelszót.
- nem foglalt email-címre és új jelszóra való módosítás. Megnézzük, hogy tényleg sikeresen átírta-e mindkét adatot.

Segédfüggvények:

- *getUserDetails()* – visszaadja egy felhasználó adatait.
- *userCounter()* – megadja a felhasználók számát.

2. MovieTest

Ebben az osztályban a filmekkel kapcsolatos metódusokat teszteljük.

newMovieTest() – a *movieIsOk()* metódussal azt teszteljük, hogy egy film szerepel-e már az adatbázisban. A *newMovie()* metódussal felveszünk két új filmet. Megnézzük, hogy tényleg beszúrta-e őket (méret 2), ill., hogy a megfelelő adatokkal lettek beszúrva. Ebben a metódusban teszteljük még a *movieSearchByTitle()* metódust is, ami a filmek között keres filmcím alapján. Meghívjuk létező filmcímrel és nem létezővel is.

Legvégül pedig a *directors* táblát teszteljük. Mivel a beszúrt 2 film rendezője ugyanaz, ezért megnézzük, hogy a rendezőt csak egyszer szúrta-e be az adatbázisba (méret 1).

movieSearchByGenreTest() – a *movieSearchByGenre()* metódust teszteljük, ami a filmek között keres kategória szerint.

A beszúrt filmek kategóriája Akció és Horror.

Tesztelt esetek:

- olyan kategóriára keresünk, amilyen kategóriájú film létezik (Akció) → méret 1
- olyan kategóriára keresünk, amilyen kategóriájú film létezik (Horror) → méret 1
- olyan kategóriára keresünk, amilyen kategóriájú film még nincs → méret 0

movieSearchByTitleAndGenreTest() – a *movieSearchByTitleAndGenre()* metódust teszteljük, amely a filmek között keres filmcím és kategória alapján egyszerre.

Tesztelt esetek:

- létező filmcím és megfelelő kategória → méret 1
- a megadott filmcím csak részben egyezik meg (LIKE) annak a filmnek a címével, aminek a kategóriája megegyezik a megadott kategóriával → méret 1
- létező film, de rossz kategória → méret 0

- létező kategória, de nem létező film → méret 0
- nem létező film, nem létező kategória → méret 0

movieSearchAllTest() – a *movieSearchAll()* metódust teszteljük, ami az összes létező filmmel tér vissza. Megnézzük, hogy a lista, amivel visszatér, annak a mérete 2, és hogy az egyes elemek adatai megfelelőek-e.

movieDeleteTest() – a *movieDelete()* metódust teszteljük, amely a megadott filmet törli az adatbázisból.

Törölünk egy olyan filmet, amihez nincs jövőbeli foglalás → sikeres törlés → méret 1 marad. Itt megnézzük azt is, hogy tényleg csak azt az 1 filmet törölte-e és nem pedig az összeset.

genreTest() – ebben a metódusban teszteljük a *getGenres()* metódust, amellyel az összes kategóriát kapjuk meg. Emellett még teszteljük azt is, hogy be tudunk-e szűrni kategóriát olyan névvel, ami már létezik → hiba, UNIQUE miatt *SQLException*-t kell kapnunk.

titleTest() – a *getTitles()* metódust teszteljük, amely visszatér az összes létező film címével, ami az adatbázisunkban van → méret 2.

rateTest() – a *getRate()* metódust teszteljük, amely megadja egy film értékelését.

Mivel eddig egyik filmhez sem lett vélemény írva, ezért mindkét film értékelése 0.0.

ageTest() – a *getAges()* metódust teszteljük, amely megadja a filmeknél megadható korhatárokat (0,6,12,16,18).

Emellett még megpróbálunk beszűrni egy olyan korhatárt, ami már létezik → nem sikerül az UNIQUE miatt → nem változik a korhatárok darabszáma.

Segédfüggvény:

- *movieCounter()* – megadja, hogy hány film van az adatbázisba.

3. ShowTest

A *ShowTest* osztályban az előadásokkal kapcsolat metódusokat teszteljük.

addRoomTest() – az *addRoom()*, *getRows()*, *getColumns()* metódusokat teszteljük.

Az *addRoom()* metódussal új termeket tudunk felvenni a sorok és oszlopok számának megadásával.

Felveszünk két új termet → méret 2, ill. a `getRows()` és `getColumns()` metódusokkal leellenőrizzük, hogy megfelelő a beszúrt termék sorainak/oszlopainak száma.

addShowTest() – az `addShow()` metódust teszteljük, amellyel új előadást tudunk felvenni.

Tesztelt esetek előadások felvételére:

- szabad időpontban szabad teremre → siker
- ugyanarra az időpontra és teremre, mint az előző pontban → sikertelen
- a terem és a dátum megegyezik, de az időpont eltér → siker
- a dátum és idő megegyezik, de a terem más → siker

getShowTitlesTest() – a `getShowTitles()` metódust teszteljük, ami azoknak a filmeknek a címét adja vissza, amelyre van kiírva jövőbeli előadás.

getShowDatesTest() – a `getShowDates()` metódust teszteljük, ami visszaadja azokat a jövőbeli dátumokat, amelyekre van kiírva előadás megadott filmhez.

getShowTimeTest() – a `getShowTime()` metódust teszteljük, ami visszaadja azokat a jövőbeli időpontokat, amelyekre van kiírva előadás a megadott filmhez és naphoz.

4. **CommentTest**

A *CommentTest* osztályban a felhasználói véleményekkel kapcsolat metódusokat teszteljük.

addCommentTest() – az `addComment()` metódust teszteljük, ami egy adott filmhez szúr be új véleményt.

Két véleményt szúrunk be ugyanahhoz a filmhez, ezután megnézzük, hogy a megjegyzések száma 2 lett-e.

Ebben a metódusban nézzük meg azt is, hogy új vélemény felvételekor annak a filmnek az értékelése is megváltozik-e, amelyhez a vélemény írva lett.

Jelen esetben a 2 vélemény pontozása 6 és 8. Mivel ugyanahhoz a filmhez lettek írva, ezért a film értékelése 0.0-ról $6+8/2=7.0$ -ra kell, hogy változzon.

getCommentTest() – a `getComments()` metódust teszteljük, amely visszaadja a megadott filmhez tartozó felhasználói véleményeket.

Megnézzük, hogy a visszatérési értéke üres String e, ha a filmhez még nem tartozik megjegyzés, ill., hogy ha tartozik hozzá (jelen esetben 2 is), akkor a visszatérési érték megegyezik-e a beszúrt megjegyzésekkel → `Assert.assertEquals()`;

5. ReservationTest

A *ReservationTest* osztályban a foglalásokkal kapcsolatos metódusokat teszteljük.

newReservationTest() – első sorban a *getReservationCounter()* metódust teszteljük, ami megadja, hogy egy adott felhasználónak hány foglalása van eddig → jelen esetben 0, mert még egyik felhasználó sem foglalt.

Ezután felveszünk 2 új foglalást különböző előadásokra különböző felhasználóktól ugyanarra a filmre.

Megnézzük, hogy a foglalásokat felvette-e a rendszer → méret 2.

a *getReservationCounter()* mindkét felhasználó esetében 1-re változott.

Továbbá ellenőrizzük, hogy a megfelelő székek foglalódtak-e le. Ezt a *getReservedSeats()* metódus tesztelésével tudjuk megtenni.

reservationSearchByUserTest() – a *reservationSearchByUser()* metódust teszteljük, ami a foglalások között keres felhasználónév alapján.

Tesztelt esetek:

- olyan felhasználónévre keresünk, akihez tartozik foglalás → ekkor ellenőrizzük, hogy a kapott értékek megegyeznek-e az elvárttal.
- olyan felhasználóra keresünk, akihez nem tartozik foglalás (illetve a felhasználó nem is létezik).

reservationSearchByNumberTest() – a *reservationSearchByNumber()* metódust teszteljük, ami a foglalások között keres foglalási szám alapján.

Tesztelt esetek:

- olyan foglalási számra keresünk, ami létezik → ekkor ellenőrizzük, hogy a kapott értékek megegyeznek-e az elvárttal.
- olyan foglalási számra keresünk, ami nem létezik.

reservationSearchByNumberAndUserTest() – a *reservationSearchByNumberAndUser()* metódust teszteljük, ami a foglalások között keres felhasználónév és foglalási szám alapján egyszerre.

Tesztelt esetek:

- létező foglalási szám és hozzá tartozó felhasználó → ekkor ellenőrizzük, hogy a kapott értékek megegyeznek-e az elvárttal.

- létező felhasználó, de nem tartozik hozzá ilyen foglalási szám
- létező foglalási szám, de nem ez a felhasználó tartozik hozzá

reservationSearchAllTest() – a *reservationSearchAll()* metódust teszteljük, amely az összes foglalást adja meg.

Leellenőrizzük elsősorban, hogy megfelelő számú foglalás van a rendszerben, tehát 2. Ezután pedig mindkét foglalás adatait összevetjük az elvárt adatokkal.

reservationDeleteTest() – a *reservationDelete()* metódust teszteljük, amely törli az adott foglalást.

Mielőtt a foglalást törölnénk, először azt ellenőrizzük a *hasReservation()* metódussal, hogy egy adott filmhez tartozik-e jövőbeli foglalás. Ha igen, akkor a film nem törölhető. Mivel jelen esetben igen, ezért megpróbáljuk törölni a filmet → sikertelen.

Most kitöröljük a filmhez tartozó foglalást a *reservationDelete()* metódus meghívásával, majd leellenőrizzük, hogy tényleg csak azt az egy foglalást törölte-e, s nem pedig az összeset. Kitöröljük a másik foglalást is → foglalások mérete 0.

movieDeleteIfHasReservationTest() – ebben a metódusban is teszteljük a *hasReservation()* függvényt → jelen esetben hamissal tér vissza, hiszen az előző pontban kitöröltük az összes foglalást az adott filmhez → ezután megpróbáljuk törölni a filmet → sikeresen törölni fogja, hiszen már nincsenek foglalások a filmhez.

Egy másik esete, amikor felveszünk egy foglalást múltbeli előadásra.

Most megpróbáljuk törölni a filmet ismét → sikerülni fog, hiszen csak múltbeli foglalás van hozzá.

Legvégül pedig leellenőrizzük, hogy a film törlésével együtt kitörlődik-e az összes foglalás, vélemény, előadás és az előadásokhoz tartozó lefoglalt ülőhelyek is, amelyek a kitörölt filmhez kapcsolódtak → igaz

Segédfüggvények:

- *reservationCounter()* – megadja a foglalások darabszámát.
- *seatCheking()* – megadja az adott filmhez tartozó lefoglalt székeket.
- *getMovie()* – adott film adatait adja meg.

Összességében elmondható, hogy a tesztelés sikerrel zárult. A metódusok az elvártak megfelelően működnek a programban.

4 Továbbfejlesztési lehetőségek

A továbbfejlesztési lehetőségek száma szinte végtelen:

- Kereséseknél több keresési feltétel megadása, hogy a keresés hatékonyabb legyen (pl.: évszám/alapján történő keresés).
- Adatlap kibővítése, pl.: profilképpel, amely a „Foglalások keresése” oldalon megjelenne adott foglalás mellett.
- Felhasználói fórum létrehozása, ahova kérdéseiket/kéréseiket tudják beküldeni a felhasználók.
- Lehetőség a felhasználói vélemények moderálására, szerkesztésére és törlésére adminisztrátorként, ill. saját vélemény módosítására felhasználóként.
- Terem felvételekor a filmtípusok megadása, amelyek az adott teremben lejátszhatók. (3D, 4D, IMAX 3D).
- Kedvezmények:
 - Napi kedvezmény.
 - Kedvezmény X. foglalás után.
- A foglalást követően annak kifizetése a felületen.
- Adott film felvétele a kedvencekhez, vagy törlése onnan.
- Jelszóemlékeztető.
- Lehetőség új adminisztrátor felvételére az adminisztrátori felületen.
- Adminisztrátorként váltogatás a felhasználói/adminisztrátori funkciók között.
- Filmek adatainak (pl.: megjelenés éve, játékidő stb.) szerkesztése adminisztrátorként.

5 Összegzés

Szakdolgozatom elkészítésének célja egy olyan alkalmazás létrehozása volt, melynek segítségével könnyedén lehet minden mozikkal kapcsolatos feladatot ellátni.

Regisztrációt, majd bejelentkezést követően felhasználóként lehetőség van kiválasztani a számunkra megfelelő előadást, és jegyeket foglalni rá, amelyeket a későbbiekben visszakereshetünk, törölhetünk vagy exportálhatunk. Kereshetünk az alkalmazás által kínált filmek között különböző szempontok alapján, melyekről véleményt is írhatunk, hogy segítsünk másokat a döntésben. Ezeket a véleményeket bármikor vissza olvashatjuk a megfelelő oldalon.

Adminisztrátorként miután bejelentkeztünk fel tudunk venni újonnan megjelenő filmeket, amelyekhez előadásokat írhatunk ki. A filmek vetítéséhez szükségünk van termekre, melyek felvételére szintén lehetőség van. A felvett filmek, foglalások és felhasználók között kedvünk szerint keresgélhetünk vagy akár törölhetjük is őket, ill. megtekinthetjük a filmekkel kapcsolatos felhasználói véleményeket is, illetve megváltoztathatjuk egyes adatainkat.

Az alkalmazás elkészítése során a legnagyobb kihívást a borítóképek tárolása és megjelenítése, ill. az oldalak kinézetének kialakításához szükséges ismeretek megszerzése jelentette.

Az általam kitűzött célt elértem, megoldottam, s eközben még inkább elmélyítettem a grafikus felület, adatbázis és felhasználóbarát alkalmazások elkészítéséhez szükséges ismereteket, továbbá megismerkedtem az email küldéssel, PDF fájl létrehozásával és a jelszavak titkosításával is programozási szinten.

6 Irodalomjegyzék

[1] Nyékyné Dr. Gaizler Judit: JAVA 2 I-II. – Útikalauz programozóknak 5.0, ELTE TTK Hallgatói Alapítvány, 2009, [1408], ISBN-978-9630640923

[2] Kathy Sierra, Bert Bates: Head First Java, 2nd Edition, O'Reilly Media, Inc., 2009, [720], ISBN-978-0596009205

[3] Jeffrey D. Ullmann - Jennifer Widom: Adatbázisrendszerek – Alapvetés – Második, átdolgozott kiadás, Panem Kiadó, 2009, [600], ISBN-978-9635454815

[4] Adatbázis diagram elkészítése:

https://www.lucidchart.com/documents#docs?folder_id=home&browser=icon&sort=saved-desc

Elérés dátuma: 2017.11.10.

[5] SQL lekérdezések:

<https://www.w3schools.com/sql/>

Elérés dátuma: 2017.11.10.

[6] Email küldés:

https://www.tutorialspoint.com/javamail_api/javamail_api_sending_emails.htm

Elérés dátuma: 2017.11.10.

[7] PDF fájl felépítése:

<https://developers.itextpdf.com/content/itext-5-examples>

Elérés dátuma: 2017.11.10.

[8] Tesztelési ismeretek elsajátítása:

<https://www.tutorialspoint.com/junit/>

Elérés dátuma: 2017.11.10.

[9] Grafikus felület megvalósításával kapcsolatos kérdések:

<https://docs.oracle.com/javase/tutorial/uiswing/index.html>

Elérés dátuma: 2017.11.10.

[10] Use case diagram jelölések:

http://swap.web.elte.hu/progtech2/progtech2_ea02.pdf

Elérés dátuma: 2017.11.10.

[11] Jelszó titkosítás:

<http://www.java2s.com/Code/Java/Security/UseMD5toencryptstring.htm>

Elérés dátuma: 2017.11.10.

[12] Osztály diagram:

<https://www.gliffy.com/uses/uml-software/>

Elérés dátuma: 2017.11.10