Szegedi Szakképzési Centrum Vasvári Pál Gazdasági és Informatikai Szakgimnáziuma

Az 54 213 05 számú Szoftverfejlesztő szakképesítés záródolgozata

Készítette:



Lázár Martin

Szeged

2019

Tartalom Jegyzék

[1.Bevezetés 2](#_Toc37292715)

[2. Fejlesztői Dokumentáció 3](#_Toc37292716)

[**2.1 Adatbázis** 3](#_Toc37292717)

[**2.1.1 Adatbázis ábra** 3](#_Toc37292718)

[**2.1.2 Egyedek és tulajdonságaik** 4](#_Toc37292720)

[**2.2 Asztali alkalmazás** 7](#_Toc37292721)

[**2.2.1 Felhasználók** 8](#_Toc37292722)

[**2.2.2 A szoftver** 9](#_Toc37292723)

[**2.2.3 Bejelentkezés** 9](#_Toc37292724)

[**2.2.4 Adminisztrációs Felület** 9](#_Toc37292725)

[**2.2.5 Felhasználói Felület** 11](#_Toc37292726)

[**2.2.6 Tesztelés** 13](#_Toc37292727)

[**2.2.7 Fejlesztési tervek** 14](#_Toc37292728)

[**2.3 Web alkalmazás** 15](#_Toc37292729)

[**2.3.1 Index** 15](#_Toc37292730)

[**2.3.2 Bejelentkezés** 17](#_Toc37292731)

[**2.3.3 Reszponzivitás** 18](#_Toc37292732)

[3. Felhasználói dokumentáció 19](#_Toc37292733)

[**3.1 Szoftver** 19](#_Toc37292734)

[**3.1.1 Belépés** 19](#_Toc37292735)

[**3.1.2 Adminisztrációs Felület** 20](#_Toc37292736)

[**3.1.3 Felhasználói Felület** 21](#_Toc37292737)

[**3.1.5 Keresés Menü** 22](#_Toc37292738)

[**3.2 Web** 23](#_Toc37292739)

[**3.2.1 Web alkalmazás bemutatása** 23](#_Toc37292740)

[**3.2.2 Funkciók** 23](#_Toc37292741)

[**3.2.3 Főoldal** 24](#_Toc37292742)

[**3.2.4 Foglalás** 25](#_Toc37292743)

[**3.3 Rendszerkövetelmény és hibalehetőségek** 26](#_Toc37292744)

[4. Források 26](#_Toc37292745)

[5. Plágium nyilatkozat 27](#_Toc37292746)

[6. Köszönetnyilvánítás 28](#_Toc37292747)

# **1.Bevezetés**

A 21. Századba már szinte elengedhetetlen, hogy az ember ne legyen a közösségi hálózatok aktív tagja, ha nem szeretne lemaradni a legfrissebb hírekről valamint arról a rengeteg információ áradatról, amelyet ez a szolgáltatás nyújtani tud. Az internet képes arra, hogy minden ember megtalálja a saját érdeklődési körének megfelelő szolgáltatásokat. Valamint internet és a számítógépek rohamos fejlődésével, az okoseszközök elterjedésével ma már minden hírről a közösségi hálózatokon keresztül juthatunk tájékoztatáshoz. Korábban leírtakból kiragadnám az érdeklődési kört. Mivel az elkészített szoftver egy hobbi ismeretterjesztéséről szól, igy jött nekem a filmek világa, mint témaválasztási lehetőség. Mivel ez a téma nagyon közel áll hozzám és filmrajongónak mondom magam. Ezért elkészítettem egy szoftvert, ahol az érdeklődők nem csak a mozikról olvashatnak, hanem az azokat kitaláló rendezőkről valamint a benne játszó színészekről, igy tehát ajánlom a filmek szerelmeseinek valamint azoknak, akik egyáltalán nincsenek, otthon a témában mivel rengeteg ismerethez juthatnak. A projektem webes részén a felhasználó jegyet foglalhat az általa kiválasztott filmre, ami a FilmMánia film múzeumba kerül vetítésre

A „Projekt bemutatása” részben kitérek arra, hogy a szoftver jelenleg hogyan is működik valamint mik, azok, amik elvárhatók az alkalmazástól bemutatóm a szoftver fejlesztésének folyamatát valamint a megvalósuló funkciókat.

A „Fejlesztési Dokumentációnál” kitérek a szoftver mögötti adatbázisra annak céljáról, valamint hogy milyen feladatokat kell, hogy elásson, bemutatom azokat az eszközöket, szoftvereket, amelyek segítségével megvalósítottam a záródolgozatom. Milyen technikákat használtam benne és ezeket forráskóddal alátámasztom.

# **2. Fejlesztői Dokumentáció**

## **2.1 Adatbázis**

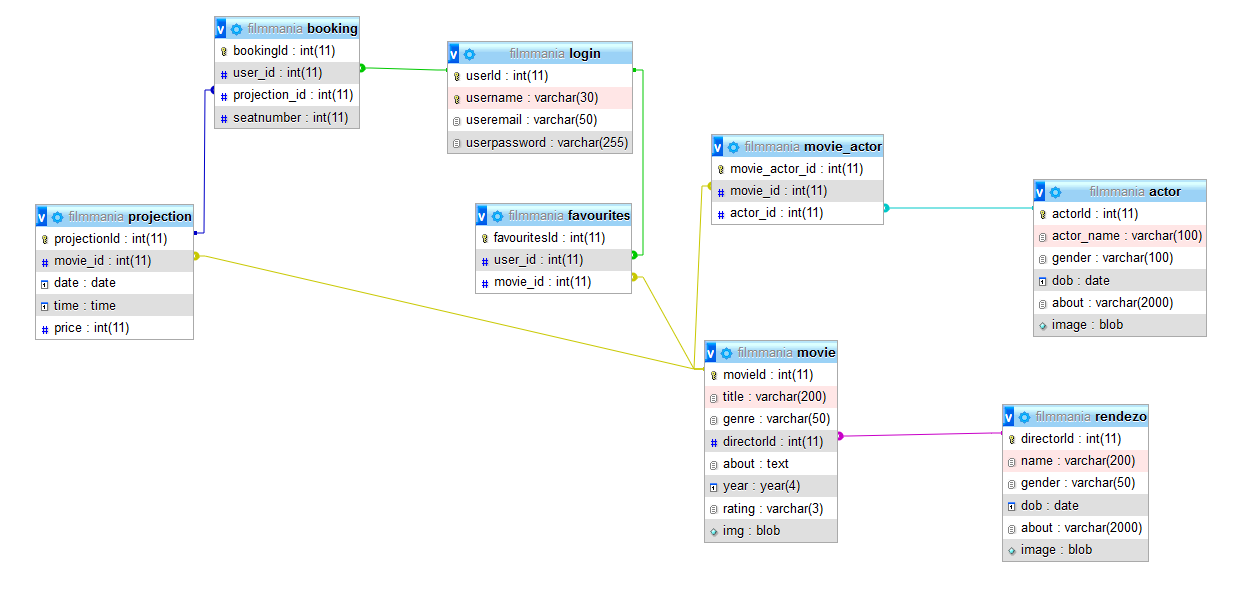
Fejlesztői dokumentációm az adatbázis bemutatásával szeretném kezdeni, mivel a rendszer szerves része az adatbázis. Csak akkor nincs rá szükség, ha az adatok szoftveresen vagy weben vannak eltárolva vagy nincsenek olyan adatok, amelyeket kezelni kellene adatbázis szinten.

Az adatbázis a szoftvernek az asztali és webalkalmazását is egyaránt kiszolgálja. Asztalon célja hogy minden felhasználónak eltárolja a bejelentkezési adatait valamint asztali regisztrációnál felvegye azokat, ezen kívül tárolja a filmek/rendezők/színészek összes adatát, felhasználók kedvenc filmjeinek elmentésére szolgál. Webes felületen a filmek vetítési adatait, amellett a felhasználó által lefoglalt székeket tárolja.

### **2.1.1 Adatbázis ábra**

Ez az ábra szemlélteti az egyedek közötti kapcsolatot, a táblák tartalmát és az adattípusokat.

### 



**2.1.2 Egyedek és tulajdonságaik**A **login** táblába kerül eltárolásra a bejelentkezéshez szükséges felhasználónév és jelszó, e-mail címe. A **movie** táblában található meg a film összes adata valamint a rendezőnek azonosítója, **rendezo** táblában megtalálható az adott rendező összes adata. **actor** táblába találjuk a színészek összes megjelenítendő adatát. A **movie\_actor** tábla egy kapcsoló tábla, ami arra szolgál, hogy a színészek tábláját össze kösse a filmek táblájával az adott táblák azonosítójuk alapján ez elengedhetetlen az adatok lekérdezései miatt. **favourites** tábla a felhasználó kedvenc filmjeit tárolja azonfelül ez is egy kapcsoló tábla mivel összeköti a felhasználó azonosítóját a film azonosítójával. **Projection** táblába tárolom a vetítésre kívánt filmek dátumát és árát. Végezetül a **booking** táblába tárolom a már lefoglalt jegyeket és azok helyeit.  **A login** tábla tulajdonságai:

* **userId:** Egész szám, elsődleges kulcs, automatikusan növekvő érték, amely a felhasználók megkülönböztetésére szolgál.
* **username:** Szöveg, a bejelentkezéshez szükséges felhasználónév, egyedi. (30)
* **userpassword:** Szöveg, a bejelentkezéshez szükséges jelszó.(255)
* **useremail:** Szöveg, a felhasználó e-mail címe.(50)

A **movie** tábla tulajdonságai:

* **movieId:** Egész szám, elsődleges kulcs, automatikusan növekvő érték, amely a filmek azonosítója
* **title:** Karakter, a filmek címet jelöli(200)
* **genre:** Szöveg, a filmek kategóriáját mutatja meg(50)
* **directorId:** Egész szám, a rendezők azonosítóját adja meg
* **About:** Szöveg, hosszabb terjedelemű a filmek rövidebb történetét írja le
* **year:** Évszám, filmek megjelenésének dátumát tárolja.
* **rating:** Szöveg, a filmek IMDb pontszámát adja
* **img:** Kép, a film képét tárolja

A **rendező** tábla tulajdonságai:

* **directorId**: Egész szám, elsődleges kulcs, automatikusan növekvő érték, amely a rendezők azonosítására szolgál
* **name**: Szöveg, a rendező nevét raktározzuk (200)
* **gender**: Szöveg: a rendező nemének tárolására szolgál (50)
* **dob**: Dátum típus, a rendező születési dátumát tárolja
* **about**: Szöveg hosszabb terjedelemű, a rendező rövid bemutatását foglalja magába
* **image**: Kép, az adott rendező profilképének tárolására szolgál

A **actor** tábla tulajdonságai:

* **actorId:** Egész szám, elsődleges kulcs, automatikusan növekvő érték, amely a színészek azonosítására szolgál
* **name**: Szöveg, a színész nevét raktározzuk (200)
* **gender**: Szöveg: a színész nemének tárolására szolgál (50)
* **dob**: Dátum típus, a színész születési dátumát tárolja
* **about**: Szöveg, hosszabb terjedelemű, a színész rövid bemutatását foglalja magába
* **image**: Kép, az adott színész profilképének tárolására szolgál

A **movie\_actor** tábla tulajdonságai:

* **movie\_actor\_id:** Egész szám, elsődleges kulcs, automatikusan növekvő érték, amely a film-színész kapcsolatainak megkülönböztetésére szolgál
* **movie\_id:** Egész szám, a film azonosítója
* **actor\_id:** Egész szám, a színész azonosítója

A **favourites** tábla tulajdonságai:

* **favouritesId:** Egész szám, elsődleges kulcs, automatikusan növekvő érték, amely a kedvencek megkülönböztetésére szolgál
* **user\_id:** Egész szám, a felhasználók azonosítóját tárolja.
* **movie\_id:** Egész szám, a filmek azonosítóját tárolja.

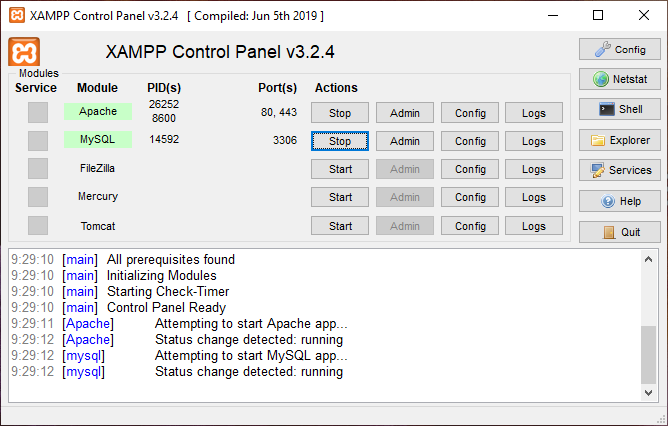
A **projection** tábla tulajdonságai:

* **projectionId:** Egész szám, elsődleges kulcs, automatikusan növekvő érték, amely a vetítések azonosítására szolgál
* **movie\_id:** Egész szám, a filmek azonosítóját tárolja.
* **date:** Dátum típus, a vetítés dátumát adja meg
* **time:** Idő típus, a vetítés időpontja tárolja (óra, perc)
* **price:** Egész szám, a vetítés árának tárolására szolgál

A **booking** tábla tulajdonságai:

* **bookingId:** Egész szám, elsődleges kulcs, automatikusan növekvő érték, amely a foglalás azonosítására szolgál
* **user\_id:** Egész szám, a felhasználó azonosítóját tárolja
* **projection\_id:** Egész szám, a vetítés azonosítóját tárolja
* **seatnumber:** Egész szám, a lefoglalt ülőhelyek számát raktározza

Az adatbáziskapcsolat létrehozásához a Xampp nevű programot használom. Az adatbázisok létrehozásához, módosításához pedig phpMyAdmin-t.



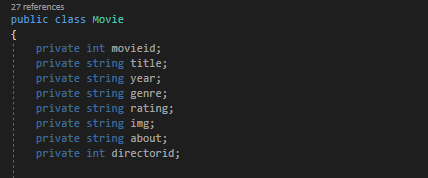
## **2.2 Asztali alkalmazás**

Az asztali alkalmazásom C# nyelven íródott, valamint a Visual Stúdió nevezetű szoftverrel. Azért választottam ezt a nyelvezetet, mert ezen a programozási nyelven van a legtöbb tapasztalatom és az iskolai tanulmányaimban is ez volt a legtöbb projektem alapja. Viszont fontos tudni, hogy ez a programozási nyelv kizárólag egy platformra használatos, tehát nem platform független, mint például a Java vagy PERL. Nem volt számomra fontos tényező a több platformra történő programozás mivel a projektemet eleve egy platformra szántam. Fejlesztői környezet kiválasztása a megszokás jegyében történt ezért választottam a Visual Stúdiót mivel ennek a kezelésével van a legtöbb ismeretem. Illetve ingyenes és a nagy a választék a személyre szabási lehetőségei között, ami kedvező volt projektem megvalósításában. Kezelése a többi fejlesztői környezethez képest egyszerűbb számomra.

Viszont a leglényegesebb szempont, amiért ezt a szoftvert választottam az az, hogy képes kapcsolatot létesíteni olyan verziókövető rendszerekkel, mint például a GitHub vagy a Microsoft saját fejlesztésű Azure DevOps nevezetű felhő alapú tárhelye. Ennek a programnak a 2019-es legújabb verzióját alkalmaztam.

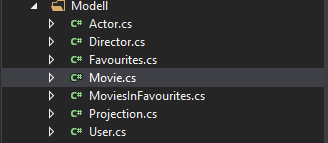
A program tervezése során igyekeztem olyan felhasználóbarát környezetet kialakítani, amelyen a felhasználó könnyedén tud tájékozódni és ne legyen képes végzetes hibát elkövetni annak használata közben. Igyekeztem olyan kifejezéseket/szavakat használni, amelyeket a felhasználó is ismer a könnyebb értelmezhetőség jegyében.

Ügyeltem az esztétikusságra. Mivel a felhasználói felület minőségét befolyásolja a felhasznált színek száma és fajtája és az olvashatóság, így igyekeztem olyan ablakokat létrehozni ahová színes elemeket vittem fel, így vonzóbb, gyorsabban áttekinthető és könnyebben megjegyezhető lett. A program megvalósítása során igyekeztem az objektum orientált programozás mellett maradni, igy nem kellett mindig újra és újra írni az adott kódot, hanem metódusokat készítettem, amiket csak egyszerűen meghívtam, ha szükséges volt. Igy felesleges kódsorokat spóroltam meg és átláthatóbbá vált a programom. Az objektum orientált programozás jóval megkönnyíti a fejlesztők munkáját a programok bonyolultságának csökkentésével. Az objektumok egységbe foglalják az adatokat és a hozzájuk tartozó műveleteket. A program egymással kommunikáló objektumok összességéből áll. A legtöbb objektumorientált nyelv osztály alapú, azaz az objektumok osztályok példányai, és típusuk az osztály. Igy én is ezt a megvalósítást alkalmaztam



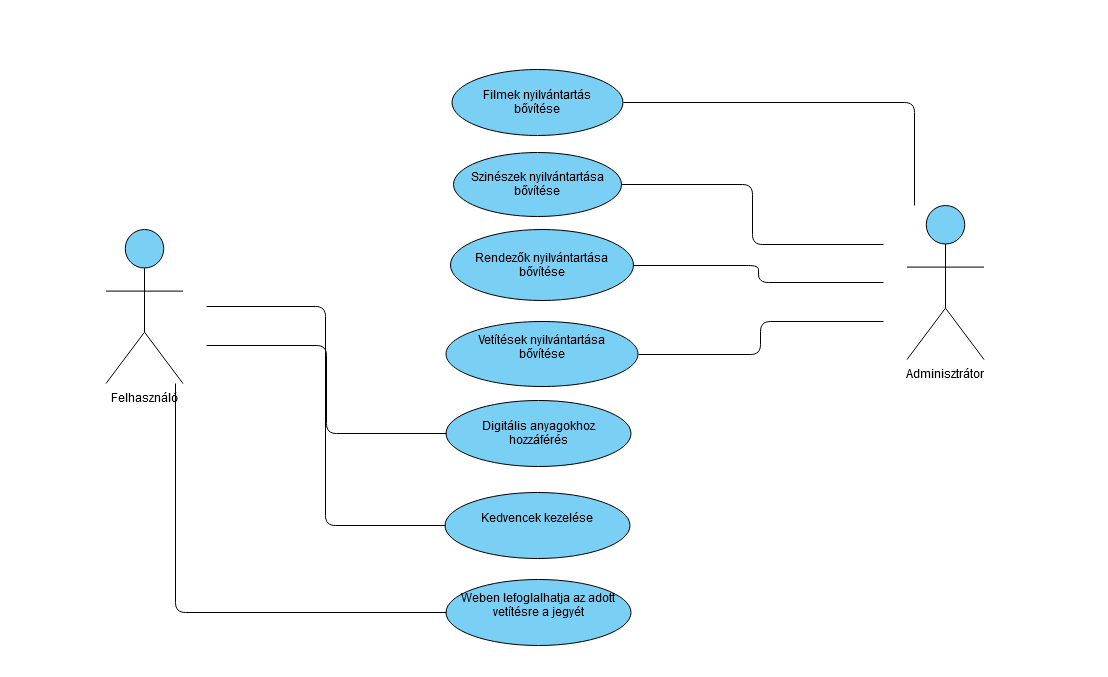
1. ábra Filmek osztálya (Film Objektum hozzátartozó tulajdonságokkal)

Amint látható a Filmek osztályának 27 referencia értéke van, tehát 27 helyen használom ezt az objektumot.



Ezen a képen látható azoknak az objektumoknak a listája, amelyekkel dolgoztam a szoftverem során

### **2.2.1 Felhasználók**

Felhasználók feladatainak szemléltetésre USE-CASE diagrammot használok ez bemutatja a fejlesztő által elkészített program használatra jogosult felhasználók tevékenységi köreit. FilmMánia esetében ez a következők lehetnek:

Ahogy az ábra is szemlélteti, az objektumok nyilvántartási részét egyedül az Adminisztrátor végzi abban a felhasználónak nincs szerepe. A felhasználoknak joguk van digitális anyagot megtekinteni, valamint weben jegyet foglalhatnak. De szerkesztési lehetőségük nincs. Aki nem regisztrált tagja a szoftvernek, az a bejelentkezési felületnél tovább nem is tud lépni. Ám regisztrálni bármikor tud az asztali felületen.

### **2.2.2 A szoftver**

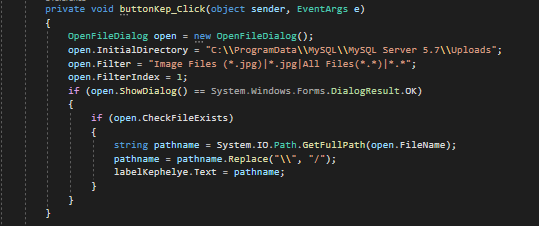
Egy olyan szoftverről van szó amelynek személyre szabottan kell működnie, igy elengedhetetlen az azonosítás, valamint mint a legtöbb program esetében itt is szükséges hogy a felhasználó jogosult-e a program használatára.

### **2.2.3 Bejelentkezés**

A program elindítása után az első felület a bejelentkezés, igy minden egyéb része a programnak rejtve marad egészen addig amig be nem jelentkezett a használó. A bejelentkezés mellett van lehetősége a felhasználónak a regisztrációra. Regisztrációnál könnyedén kiszúrható, hogy valós személy regisztrál-e mivel a rendszer nézi a regisztrálni kívánt adatokat, és ha azok nem felelnek meg a követelményeknek (felhasználónév/email/jelszó) akkor nem engedi tovább az illetőt. A rendszer a belépés során 2 adatot kér le: Felhasználónév, jelszó. A szoftver SQL utasítással kérdezi le az adatokat az adatbázisból, és ha azok nem egyeznek meg egyetlen ilyen párosítással sem akkor hibaüzenet ad ki és a program nem megy tovább. Azonban ha egyezést talál a megadott felhasználónév és jelszópárosítással az adatbázisban lévővel. E két adatot viszi tovább a fő menübe. Adminisztrátori bejelentkezés során a Belépés gomb helyett az Adminisztrátor gombra kell kattintani ahol ugyan is csak adminisztrátori felhasználóval valósulhat meg a belépés. Erre azért van szükség, hogy az egyszerű felhasználó ne érhesse el valamint, hogy adminisztrátor is be tudjon jelentkezni mind a kettő felületre.

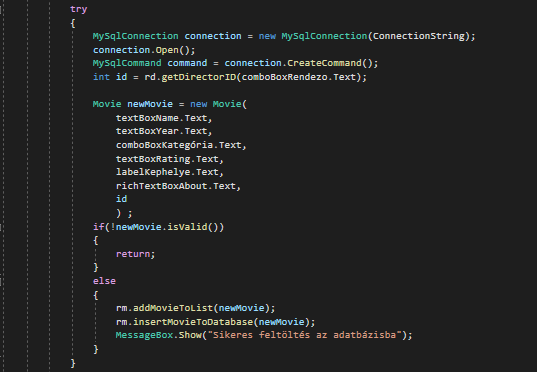
### **2.2.4 Adminisztrációs Felület**

Sikeres bejelentkezés után láthatjuk az adminisztrációs felületet, ezen a felületen történnek az adatok feltöltése. Egy tabPagen vannak elhelyezve, a film, rendező, színész feltöltése. Valamint az adott szinész felvétele egy filmhez, ezentúl a vetitési időpontokat is itt tudjuk felvenni a filmekhez amely a Film Múzeumba kerül lejátszásra.



2. ábra Kép feltöltéshez használt kód

A Xamp egyik hátránya hogy csak a saját gyökérkönyvtárából engedélyezi a képfeltöltést az adatbázisba, ezért a feltölteni kívánt képeket mindenféleképp ott kell tárolni amig nem történt meg a feltöltés



3. ábra Új film felvételének kódja

Ezen a képen példa látható az objektum orientált programozás előnyei, a különböző metódusok meghívásával megkönnyítettem a kód átláthatóságát valamint használhatóságát. IsValid függvény használtam, hogy leellenőrizze a bevitt adatok helyességét, és ha mindent helyesnek talált a program csak utána engedi tovább az információt, hogy feltöltődön az új filmek listájába, amelyből feltöltődik az adatbázisba. Hogy ha hibát talált a program a felvitt adatokkal kapcsolatban akkor azt jelzi a felhasználónak az errorProviderek segítségével



4. ábra errorProviderek felépitése

Az errorProviderek pontosan megmutatják hogy melyik mezőben volt a hiba és leirjak a helyes megadásának feltételét. Minden egyes helyes feltöltés után a szoftver vissza jelzést küld az adminisztrátornak a feltöltés sikerességéről.

### **2.2.5 Felhasználói Felület**

Sikeres bejelentkezés után megjelenik a felhasználói felület. Itt lehetőségünk van látni az összes eddigi felvett Filmek listáját egy DataGridView-ban. Azok címeire kattintva betölti az adott filmnek a nézetét ahol az ismereteket láthatjuk. Kedvencek fülre kattintva láthatjuk egy másik DataGridView-ban a korábban kedvencekhez adott filmek listáját. Lehetőségünk van különböző kereséseket végrehajtani gombok segítségével (film, színész, rendező). Filmek különböző szempontjaira való keresés:

int count = 0;

string query = "SELECT title, name, actor\_name, rating, genre FROM movie INNER JOIN rendezo on movie.directorId = rendezo.directorId " +

"INNER JOIN movie\_actor on movie\_actor.movie\_id = movie.movieId " +

"INNER JOIN actor on actor.actorId = movie\_actor.actor\_id ";

if (textBoxTitle.Text != "")

{

query += "WHERE ";

query += "title LIKE '%" + textBoxTitle.Text + "%' ";

count++;

}

if (textBoxCategory.Text != "")

{

if (count != 0)

{

query += "AND ";

}

else

{

query += "WHERE ";

}

query += "genre LIKE '%" + textBoxCategory.Text + "%' ";

count++;

}

if (textBoxDirector.Text != "")

{

if (count != 0)

{

query += "AND ";

}

else

{

query += "WHERE ";

}

query += " name LIKE '%" + textBoxDirector.Text + "%' ";

count++;

}

if (textBoxActor.Text != "")

{

if (count != 0)

{

query += "AND ";

}

else

{

query += "WHERE ";

}

query += " actor\_name LIKE '%" + textBoxActor.Text + "%' ";

count++;

}

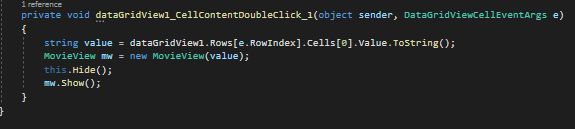
query += ";";

MovieSearchResult movies = new MovieSearchResult(query);

movies.Show();

this.Hide();

Ez egy összetett keresés itt egy vagy akár több tényező megadásával kereshetünk a filmek között. Ezek eredményeit ismét egy DataGridView-ban láthatjuk, és ha rá kattintunk a kiválasztott mezőre bejön az adott Film nézete.



### **2.2.6 Tesztelés**

A szoftver készítése során az egyik legfontosabb feladat az egységtesztek készítése. A teszttel nem csak az elkészített kódot lehet leellenőrizni, de ezeket fel is lehet használni az ellenőrzések során. A projektben nem mindenhez készült teszt vagy ellenőrzés de új film hozzáadása során a feltöltést gomb megnyomásakor ellenőrzésre kerülnek az adatok és ha valami hibás adattal próbálná feltölteni a használó akkor a problémát megjelölve egy errorProvider jelenlétével jelzésre kerül az adott mező mellett.

Üresen hagyott mezőre mutatnék egy jó tesztesetet példaképpen:

[TestMethod()]

public void isValidTestMovieTitleIsEmpty()

{

try

{

Movie m = new Movie("Nagymenők", "1998", "Krimi", "8.6", "asd", "Ez egy jó film", 8);

if (m.isValid())

{

}

}

catch (Exception e)

{

Assert.Fail("A Film első betűje nagybetű mégis hibát dob");

}

Ez a teszt azt vizsgálja, hogy a film címe üres-e ha igen akkor hibát dob

Évszám ellenőrzésre mutatnék egy rossz teszteset példát:

[TestMethod()]

public void isValidTestMovieYearAllCharIsNumber()

{

try

{

Movie m = new Movie("Nagymenők", "199d", "Krimi", "8.6", "asd", "Ez egy jó film", 8);

if (m.isValid())

{

Assert.Fail("Üres név esetén nem dob kivételt!");

}

}

catch (ModellMovieNotValidYearAllCharIsNumber mmnvyacin)

{

if (mmnvyacin.Message != "Az évszám nem tartalmazhat betüket!")

Assert.Fail("Üres név esetén hibát dob, de a hiba szövege rossz!");

}

catch (Exception e)

{

Assert.Fail("Üres név esetén nem dob kivételt vagy rossz kivételt dob!");

}

}

Ez az eset azt vizsgálja, hogy a megadott évszám összes karaktere szám-e, és mivel a példában nem minden karakter szám ezért akkor fogadja el a program, ha rossz a példa értéke.

### **2.2.7 Fejlesztési tervek**

Egyes pontok még a szoftverből kimaradtak, vagy éppen a szemléltetéshez nem igazán szükséges jelenleg. Pont ezek a hiányosságok miatt – és a korábban említett ellenőrzéseket, teszteléseket is beleértve – a szoftver jelenleg még csak az alfa verziónál tart. A program kiadásra jelölt verziója csak igény esetén készül el. E verzió olyanokat hordozna magában, mint:

* Adatbázis bővítés: Filmek/Rendezők/Szinészek száma szinte megszámlálhatatlan mennyiségben létezik ezért az adatbázis folyamatos töltése fontos szempont.
* Módosítási funkció: Felhasználói és egyéb adatokat szoftveresen is meglehetne változtatni pl:.(új jelszó, e-mail-cím változás, Rendező neve)
* Az elfelejtett jelszó menüpont elkészítése

## **2.3 Web alkalmazás**

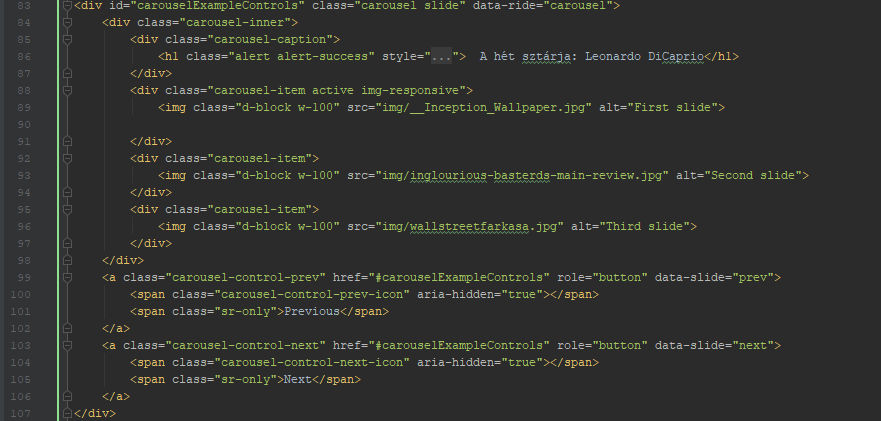
A weboldal elkészítésé során törekedtem arra hogy a felhasználó egyszerűen és gyorsan tudja igénybe venni az oldal tisztségeit. Olvastam a színek helyes használatáról és az elemek térbeli elhelyezkedéseinek lényegéről és annak a látógatóra gyakorolt hatásáról és próbáltam a szerint kialakítani a weboldalam megfelelő összhatás érdekében. Az oldal szerkezetét a HTML jelölőnyelv adja. Emellett a dizájnban fontos szempontot játszott a Bootstrap 4-es verziója valamint CSS stílus lapok használata is, továbbá felhasználtam front-end oldalon a Jquery és Javascript függvényeit. Az összes oldal PHP formátumban íródott mivel kellett egy olyan nyelv, amely képes a szerver-kliens kommunikációra, és a legalapvetőbb erre szolgáló nyelv a PHP, valamint ennél van a legtöbb tapasztalatom. Egyes funkciókat Ajax technológia segítségével oldottam meg mivel számomra az volt a legkézenfekvőbb megoldás. Az Ajax technológiával tudtam növelni a honlap gyorsaságát és kezelhetőségét.

Fejlesztői környezet kiválasztásánál a programok sokfélesége miatt a legtöbb funkcióval ellátott környezet használtam, amely a PhpStorm nevezetű szoftver ugyanis több olyan funkcióval rendelkezik, amiket hasznosnak találtam a munkám során, ilyen például a kód kiegészítés. A program felismeri azt a kódot, amit használni szeretnék és kiegészíti azt. Továbbá egy gombkombinációra rendezi, addig irt soraimat, az átláthatóság nevében. Dizájn során is hasznom vált mivel minden színskálának ismeri az értékét igy sokkal folyékonyabban történt a megfelelő stílus megtalálása.

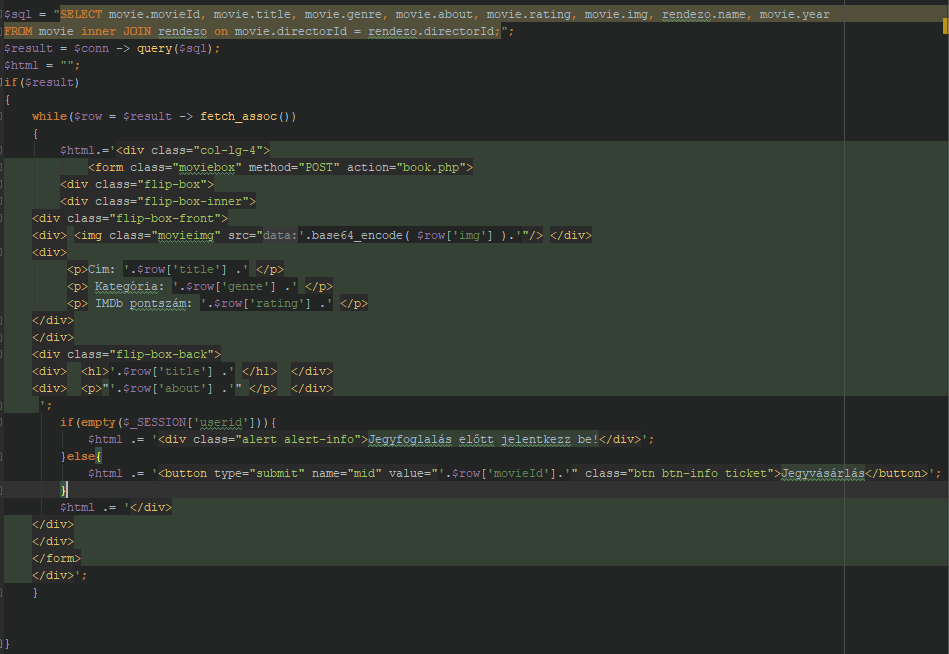
### **2.3.1 Index**

A kezdőoldalt másnéven indexnek nevezzük. Egy oldal betöltődésekor ez lesz az amit automatikusan keresni fog a böngésző a fájlok között. A kezdőoldal volt nálam a leglényegesebb szempont mivel ezen helyezem el a legfontosabb információkat valamint egy összefüggő képet ad magáról az egész alkalmazásról. Amikor betöltődik az oldal a felhasználó egy Image slider-el találja magát szemben, amin elhelyeztem az éppen promotált, vetített filmeket.

Ezen felül a kezdőlapon a Carousel alatt megtalálható az összes film Flip box-ja. Ezeken megtalálhatók a filmek adatai, amelyet egy SQL lekérdezés majd beágyazással valósitok meg. Valamint a hátoldalán, ha **be van jelentkezve** az olvasó „Jegyvásárlás” gomb.



5. ábra Carousel megvalósitása



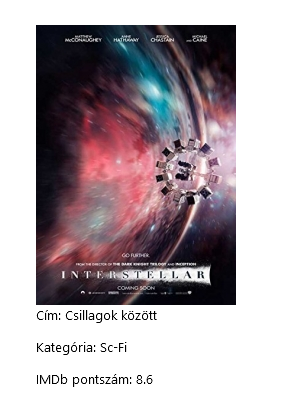
6. ábra Flip Boxok feltöltése

### **2.3.2 Bejelentkezés**

A bejelentkezés az egész oldal alapja. A kezdőlapon kívül más funkciója az oldalnak nem valósítható meg amig nincs bejelentkezve a felhasználó. Bejelentkezési adatok ellenőrzésére egy Validációt hoztam létre, hogy csak olyan adatokkal történhessen meg a bejelentkezés, amelyet az oldal enged. A bejelentkezést úgy oldottam meg, hogy a menüsávban a belépés gombra kattintva egy „Modális” ablak jelenik meg. Az első mezőbe a felhasználónév a másodikba a jelszó kerül. Ha helyesen van kitöltve akkor egy Ajax kéréssel fut le, majd átirányításra kerül, az oldal. Ha a bejelentkezés sikertelen nem töltődik újra az oldal, hanem a kliens oldalról hiba üzenetet jelenitek meg. Sikeres bejelentkezés után Ajax segítségével küldöm át szerver oldalra a felhasználó azonosítóját, és utána onnan dolgozok vele tovább. Fontos hogy weboldalnak minden olyan oldala, amelyhez bejelentkezés szükséges vizsgálja, hogy a felhasználó valóban bevan a jelentkezve. Ezzel is megakadályozva azt hogy kijátsszák a weboldal.

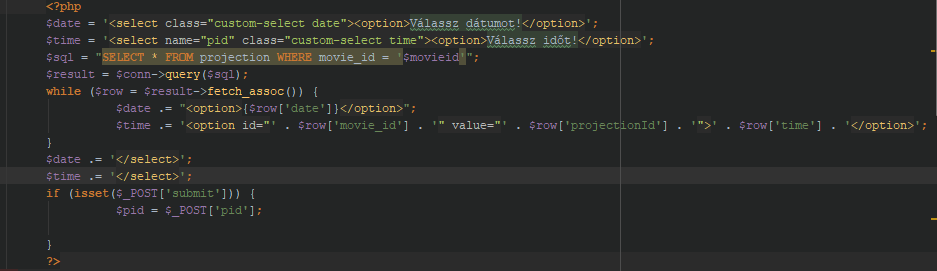


7. ábra Validációs kódrészlet



8. ábra Film Flipbox

Bejelentkezés után a megtetszett filmnek Flip Box-jának hátoldalán lévő „Jegyfoglalás” gombra kattintva előjön, a foglalási oldal ahol a megfelelő időpont kiválasztásával lehetőségünk van rákattintani az ülőhelyválasztás gombra ami ugyanis átirányítja a felhasználót a Vetítőterem nézetre ahol kiválasztásra kerülhet a megfelelő szék a kezelő számára azután ha megtörtént a kiválasztás elmentheti az adatbázisba a foglalás gombbal és onnantól a foglalás érvényes. Ezután a szerver visszairányítja a kezdőlapra.



9. ábra Dátum és időpont lekérdezése

### **2.3.3 Reszponzivitás**

Olyan megközelítéssel terveztem meg a weboldalam, amelynek célja, hogy optimális megjelenést biztosítson - könnyű olvashatóság, egyszerű navigáció a lehető legkevesebb átméretezéssel és görgetéssel a legkülönfélébb eszközökön (az asztali számítógép monitorjától egészen a mobiltelefonokig). Mivel Bootstrap-et használtam, így segítségemre volt a keretrendszerben létrehozott rácsszerkezet, amelynek támogatásával könnyedén alakítottam ki reszponzív oldalakat.

# **3. Felhasználói dokumentáció**

## **3.1 Szoftver**

A program elindításához szükséges első lépés az az, hogy futtatni kell az adatbázist. Amelyet már a korábban említett Xampp nevezetű applikáció segítségével tudunk megtenni azon belül is a MYSQL valamint az APACHE szolgáltatásoknak kell működni-e



Ezeket után már nyugodtan indíthatjuk a FilmMánia.exe nevezetű állományt.

### **3.1.1 Belépés**

A program elindulása után a bejelentkezési felület fogad minket amelyen megkell adni a helyesen megadott felhasználónév jelszó párosunkat. Ha nincs még Fiókunk a regisztrációs gombra kattintva bármikor létre tudunk hozni egyet.

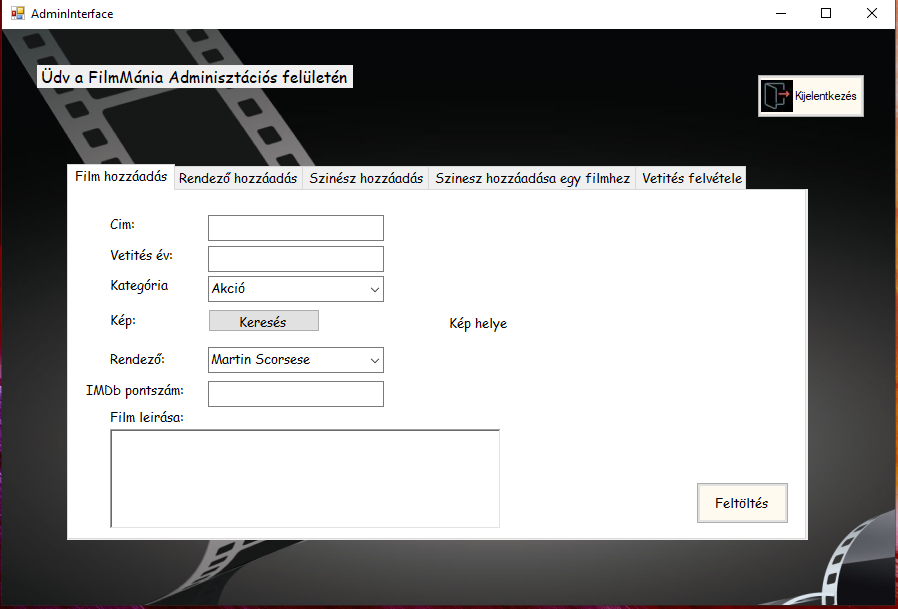




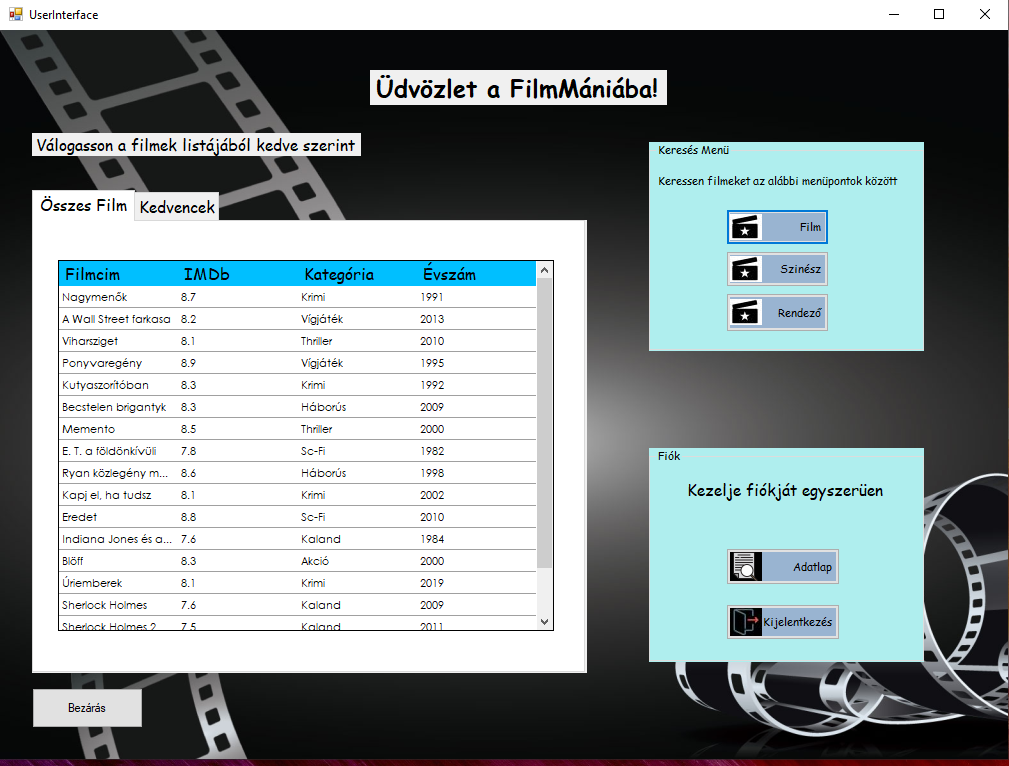
Sikertelen bejelentkezésnél egy hibaüzenet fog adni. Azonban ha a bejelentkezés sikeres átirányit minket a felhasználói kezelőfelületre. Adminisztrátor gombra kattintva, ha adminisztrátori fiókkal próbál belépni, akkor az adminisztrációs felületre visz tovább. Azonban oda ha felhasználói fiókkal próbál a kliens belépni hibát fog dobni a rendszer.

### **3.1.2 Adminisztrációs Felület**

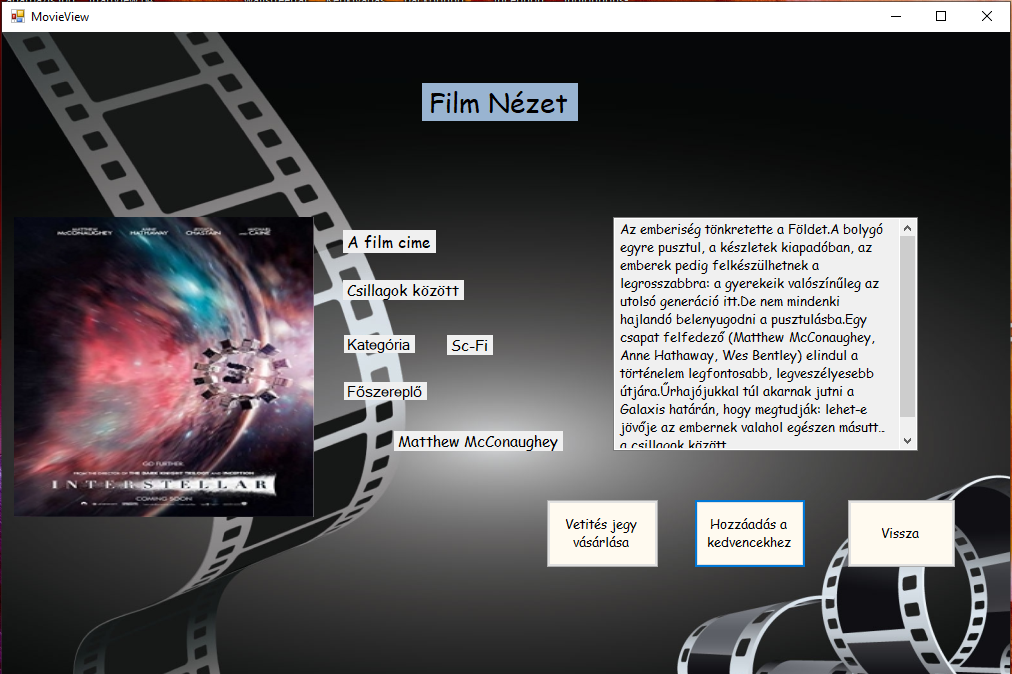
Ezen a felületen történek minden egyes adatfelvétel az adatbázisba. Adatok felvitele a Rendezőkkel kezdődik, ugyanis Rendező nélkül nem lehet filmet felvenni. Ezek után hozzáadható a film, mivel ha sikeresen feltöltődött a rendező a film hozzáadása menüpontnál megjelenik a „Rendező” kiválasztási lehetőség. Utána feltölthetők a színészek az adatbázisba. Egy színész több filmben is szerepelhet, több színész is játszhat egy filmben ezért a Szinész hozzáadása egy filmhez” menüpont ezek felvételére szolgál. Majd végül a Film Múzeumi vetítés hozható létre a „Vetítés felvétele” menüpontban. Minden megadott adatot ellenőrzi, a rendszer ezért például nem lehet a pontszám helyére betűt írni. Fontos tudni, hogy az IMDb pontszám az az Internetes Film Adatbázis felhasználói által pontozott érték. Világszinten elismert pontozási rendszer 0-10-ig terjedő skálán.



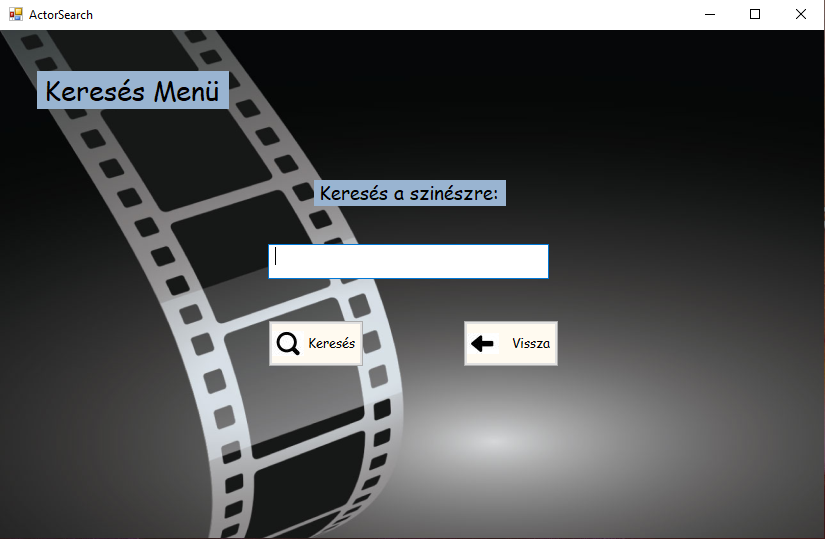
### **3.1.3 Felhasználói Felület**



Bejelentkezés után fogad a fő képernyő. Itt láthatjuk az összes eddigi filmet, Továbbá a kedvencek felületre kattintva azokat a filmek láthatók, amelyek már előzetesen hozzá lett adva a kedvencekhez. Minden filmnek meglehet tekinteni a Nézetét azzal hogy a nevére dupla kattint és ezáltal a program átirányit a Filmnézetre. Itt olvasható a minden információ a filmről, látható a boritóképe majd a gombok segítségével hozzáadható a kedvencek listához vagy weboldalról történő átirányítása során megtekinthető hogy műsoron van-e jelenleg az adott alkotás. „Vissza” gomb segítségével vissza lehet térni a fő képernyőre



### **3.1.5 Keresés Menü**



Keresés menü használatához először tudunk kell mire is akarunk pontosan keresni. Majd annak a megfelelő gombjára kattintva elő jön a keresés menüpont. Ezek után be kell gépelni az keresni kívánt objektumot (Nem szükséges teljes nevet használni a kereséshez pl:. Steven Spielberg nevénél elegendő a keresztneve). „Keresés” gombra kattintva előjön a találat amely egy DataGridViewba látható

## **3.2 Web**

### **3.2.1 Web alkalmazás bemutatása**

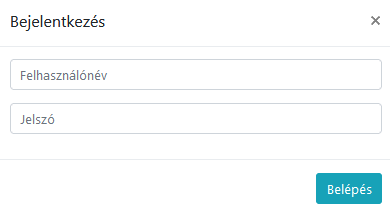
A weboldalt probáltam olyan módon kialakitani hogy a felhasználók számára könnyen kezelhető és a követelményeknek megfeleljen.

A következő oldalak találhatók benne:

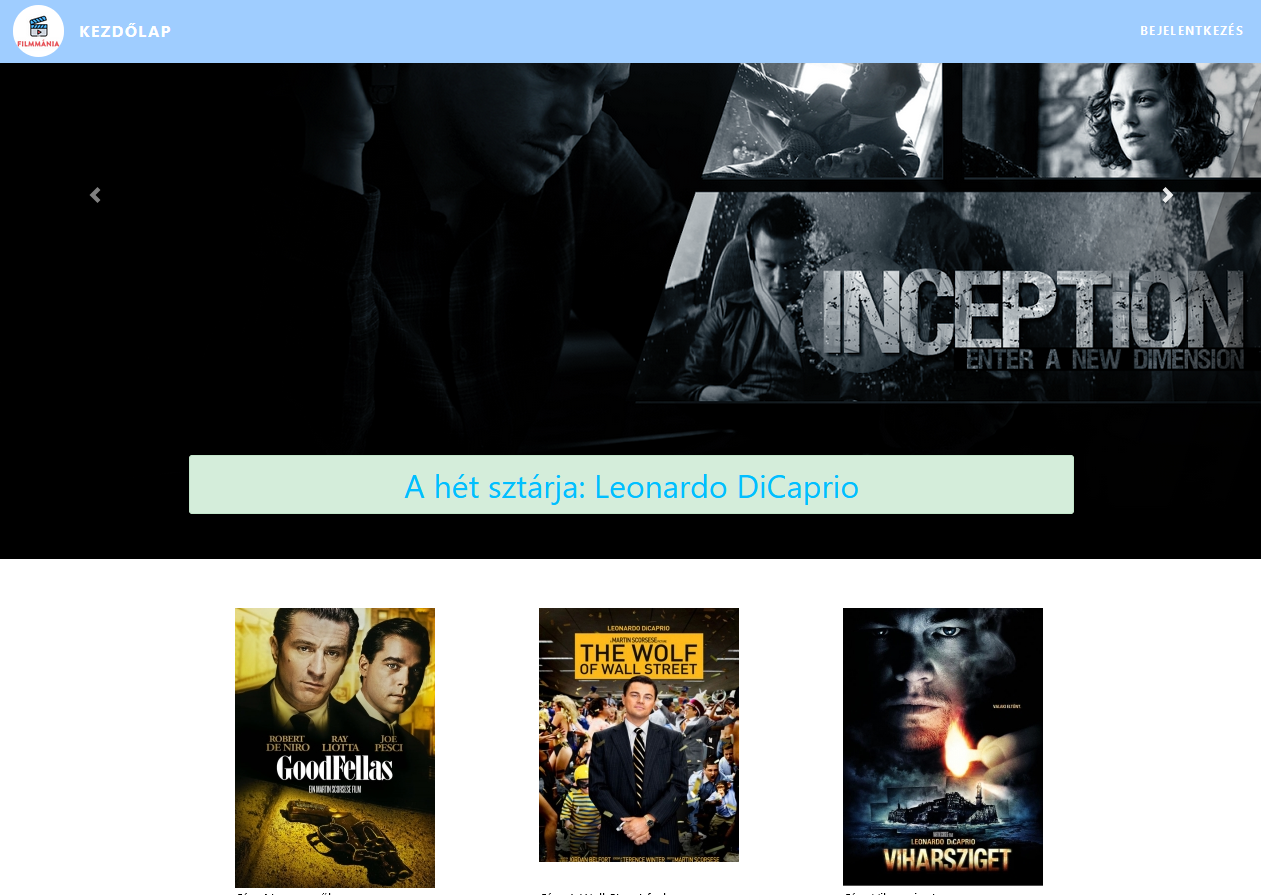
* Főoldal: Itt láthatjuk a filmeket flipcardok segítségével
* Booking: Itt választhatjuk az adott filmek vetítési dátumát és foglalhatjuk le helyünket a vetítésre.
* Foglalásaim: Ezen az oldalon láthatjuk a jelenlegi foglalásokat és azok árait. Törölni tudjuk őket a „Törlés” gomb segitségével

### **3.2.2 Funkciók**

A digitális anyagok megtekintésén kívül minden funkcióhoz bejelentkezés szükség, amely a felső menüsávban található egy modális ablakban a helyes felhasználónév és jelszó beírása után belépés gombra kattintva beléptünk a weboldalra. Hibás felhasználó esetén a rendszer jelez, hogy rossz adatokkal próbálunk belépni.



Sikeres bejelentkezés után megjelenik, a Főoldal teljes képében valamint a felső menüsávban megjelenik a foglalásaim menüpont ahol a korábban lefoglalt jegyeket lehet megtekinteni.



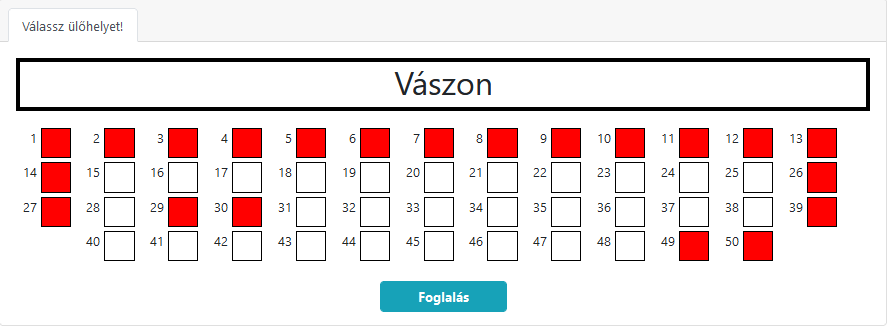
### **3.2.3 Főoldal**

A Főoldalon találjuk a filmeket, amelyekről röviden a főbb információkat találjuk, majd ha rávisszük az kurzort, akkor hátulján megjelenik a története illetve a Jegyvásárlás gomb feltéve, ha be vagyunk jelentkezve, ha nem vagyunk, bejelentkezve a gomb helyett egy üzenet fogadja a látogatót azzal a szöveggel, hogy jegyfoglalás előtt jelentkezzen be. A gombra kattintva a weboldal átirányit a foglalás oldalra.



### **3.2.4 Foglalás**

A foglalás oldalán megtaláljuk a kiválasztott film nézetét, amelyen láthatjuk a film vetítésének dátumait/időpontjait. Ezek kiválasztásával aktiválásra kerül a „Választok ülőhelyet!” gomb mely tovább minket a vetítő terem nézetére ahol láthatjuk a már foglalt(pirossal jelölve) és szabad székeket. Majd kiválaszthatjuk a számunkra megfelelő helyű széket/székeket és ezeket lefoglalhatjuk a „Foglalás gombbal”



Ha lefoglaltuk a székünket, akkor a rendszer automatikusan visszairányit minket a fő képernyőre majd a felső menüsorban a foglalásaim menüre kattintva bejön a korábbi foglalások oldala ahol az időpont kiválasztása után láthatjuk az arra, időpontra a foglalt filmek nevét, időpontját székek számát és az árat egy táblázatban. Ezeket törölni tudjuk, a kuka ikonra kattintva ahol az oldal visszajelzést küld a törlés sikereségéről.



## **3.3 Rendszerkövetelmény és hibalehetőségek**

A program 64 bites Windows 10-en lett tesztelve és írva, De akár Windows 7-en is képes futni ezért annak a rendszerkövetelménye bőven elegendő ahhoz, hogy az szoftver megfelelően fusson

Az Windows 7 számítógépen való futtatásához a következők szükségesek:

* 1 gigahertzes (GHz) vagy gyorsabb 32 bites (x86) vagy 64 bites (x64) processzor\*
* 1 gigabájt (GB) RAM (32 bites rendszerhez) vagy 2 GB RAM (64 bites rendszerhez)
* 16 GB (32 bites rendszerhez) vagy 20 GB (64 bites rendszerhez) szabad lemezterület
* DirectX 9 grafikus eszköz WDDM 1.0 vagy újabb illesztőprogrammal

Hiba alakulhat ki, ha:

* A Xampp nincs elindítva
* Az adatbázist nem ismeri fel vagy nincs rendesen importálva
* Adatfeltöltésnél a kép nem a Xampp mappájában található
* A szoftver nem a C meghajtóról van indítva
* Weboldali állományok nem htdocs mappában található

# **4. Források**

* <https://stackoverflow.com/>
* <https://getbootstrap.com/>
* <https://jquery.com/>
* <https://www.w3schools.com/>
* <https://www.iconfinder.com/>
* <https://online.visual-paradigm.com/app/diagrams/#proj=0&type=UseCaseDiagram>

# **5. Plágium nyilatkozat**

Alulírott, Lázár Martin András, aki az SZSZC Vasvári Pál Gazdasági és Informatikai Szakgimnázium tanulója, hogy az általam készített FilmMánia asztali alkalmazás a saját munkám

Kelt:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
aláírás

# **6. Köszönetnyilvánítás**

Szeretném megköszönni minden felkészítő tanáromnak, hogy idejükkel és tudásukkal segítették a tanulmányaimat a képzés során. Legfőképp Bálint Róbert osztályfőnökömnek, és Gyuris Csaba Szoftverfejlesztési tanáromnak illetve Kádár Tünde Tanárnőnek.