

Beadandó feladat dokumentáció

Feladat

Készítsünk egy mozi üzemeltető rendszert, mely két alkalmazásból áll. Az első egy webes felület, melyen keresztül a nézők megtekinthetik a moziműsort, valamint rendelhetnek jegyeket.

- A főoldalon megjelenik a napi program, azaz mely filmeket mikor vetítik a moziban, valamint kiemelve az öt legfrissebb (legutoljára felvitt) film plakátja.
- A filmet kiválasztva megjelenik annak részletes leírása (rendező, főszereplők, hossz, szinopszis), plakátja, továbbá az összes előadás időpontja.
- Az időpontot kiválasztva lehetőség nyílik helyfoglalásra az adott előadásra. Ekkor a felhasználónak meg kell adnia a lefoglalandó ülések helyzetét (sor, illetve oszlop) egy, a mozitermet sematikusan ábrázoló grafikus felületen. Egyszerre legfeljebb 6 jegy foglalható, és természetesen csak a szabad helyek foglalhatóak (amelyek nem foglaltak, vagy eladottak). A felhasználónak ezen felül meg kell adnia teljes nevét, valamint telefonszámát, ezzel véglegesíti a foglalást.

A második egy asztali grafikus felület, melyet az alkalmazottak használnak a mozipénztárakban az előadások meghirdetésére, illetve jegyek kiadására.

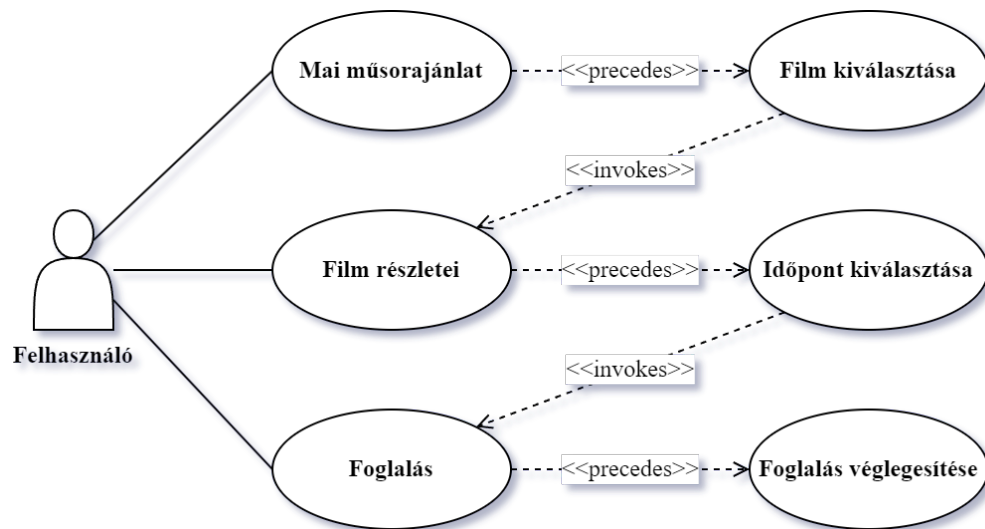
- Az alkalmazott bejelentkezhet (felhasználónév és jelszó megadásával) a programba, illetve kijelentkezhet.
- Új film felvitelekor ki kell tölteni a film adatait (cím, rendező, főszereplők, hossz, szinopszis), valamint feltölthetünk egy képet plakátként.
- Új előadás meghirdetéséhez a felhasználónak ki kell választania a termet, valamint a filmet, és az időpont megadásával hirdetheti meg az előadást. A meghirdetéskor ügyelni kell arra, hogy az előadás ne ütközzön más előadásokkal az adott teremben (figyelembe véve a kezdés időpontját, illetve a film hosszát), illetve két előadás között legalább 15 percnak kell eltelnie a takarítás végett.
- A jegyvásárláshoz ki kell választani a filmet és az előadást. Ezt követően listázódnak a helyek (sor, oszlop, státusz). A szabad, illetve foglalt helyek eladhatóak, illetve a foglalt helyeket kiválasztva meg lehet tekinteni a foglaló adatait (név, telefonszám).

Elemzés

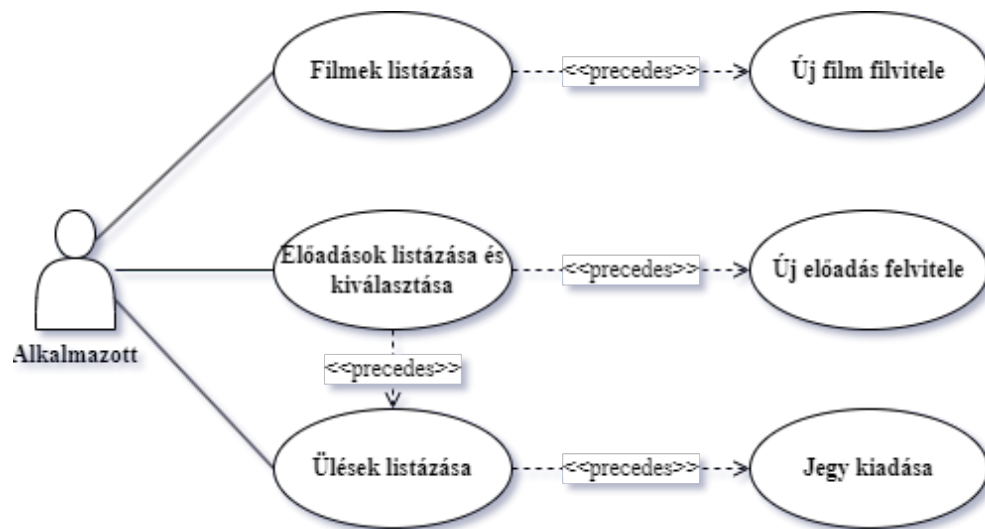
- A feladat megoldása négy komponens, egy adatbázis (illetve annak adatátviteli osztályai), egy webes felhasználói felület, egy webszolgáltatás és az azt használó asztali grafikus felület. Az adatbázis az Entity Framework, a webes felhasználói felület (illetve webszolgáltatás) ASP.NET CORE és Razor, illetve az asztali grafikus felület WPF keretrendszerek segítségével készülnek.
- A webes felhasználói felület három weblapon fog megjelenni:
 1. weblap a főoldal, melyen megjelenik a napi program, azaz mely filmeket mikor vetítik a moziban, valamint kiemelve az öt legfrissebb (legutoljára felvitt) film plakátja.
 2. weblap, mely az 1. weblapon való film választása után megjeleníti annak részletes leírását (rendező, főszereplők, hossz, szinopszis), plakátját, továbbá az összes aznapi előadás időpontját.
 3. weblap, melyen a 2. weblapon való időpont kiválasztása után lehetőség nyílik helyfoglalásra az adott előadásra. Ekkor a felhasználónak meg kell adnia a lefoglalandó ülések helyzetét (sor, illetve oszlop) egy, a mozitermet sematikusan ábrázoló grafikus felületen. Egyszerre legfeljebb 6 jegy foglalható, és természetesen csak a szabad helyek foglalhatóak (amelyek nem foglaltak, vagy eladottak). A felhasználónak ezen felül meg kell adnia teljes nevét, valamint telefonszámát, ezzel véglegesíti a foglalást.
- Az asztali grafikus felület két ablakban, melyből az egyik három lapon (fülön) fog megjelenni:
 1. ablak az alkalmazás elindításakor először megjelenített, ahol az alkalmazott bejelentkezhet a programba.
 2. ablak, mely tartalmazza az egyes funkciókhoz kötött oldalakat
 1. oldala, ahol listázhatóak a filmek és azokhoz adható új, illetve a meglévőket lehet frissíteni és törölni.

2. oldala, ahol listázhatóak az előadások és azokhoz adható új, illetve a meglévőket lehet frissíteni és törölni.
3. oldala, ahol listázhatóak a másodikon választott előadás ülései, melyek közül a szabad és foglalt ülésekre lehet jegyet kiadni.

Felhasználói esetek

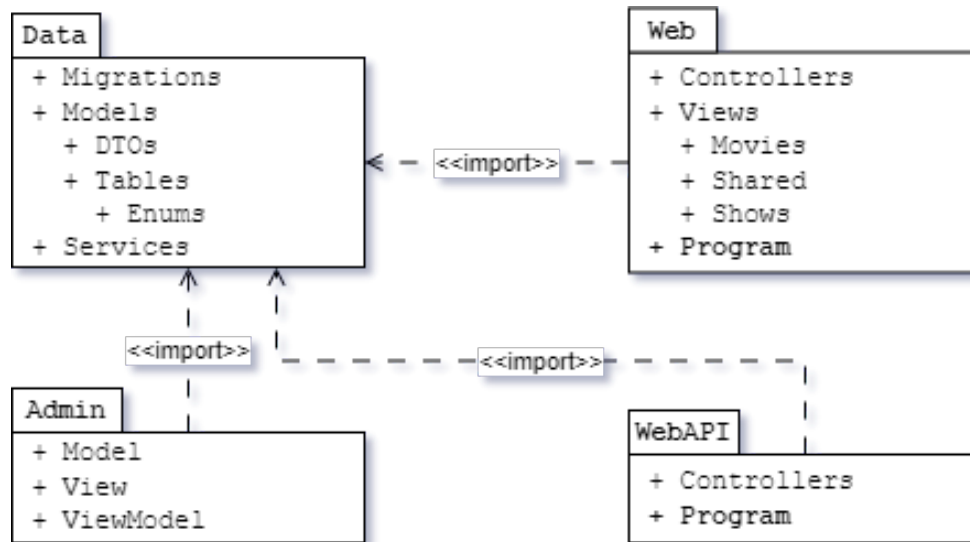


Ábra 1: Webes felület felhasználói eseteinek UML diagramja



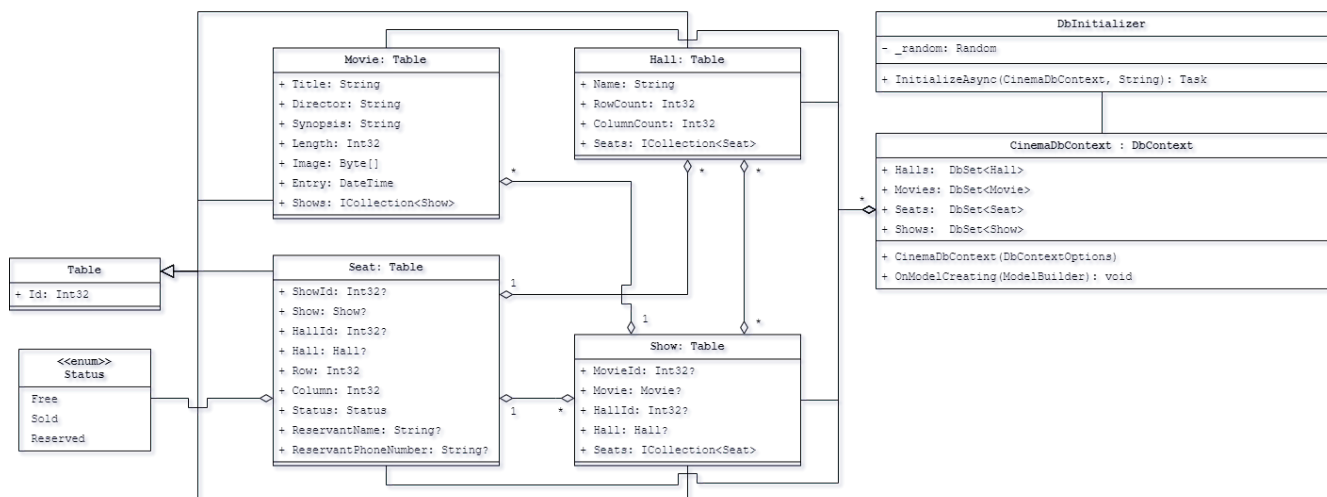
Ábra 2: Asztali grafikus felület felhasználói eseteinek UML diagramja

Rendszer szerkezete



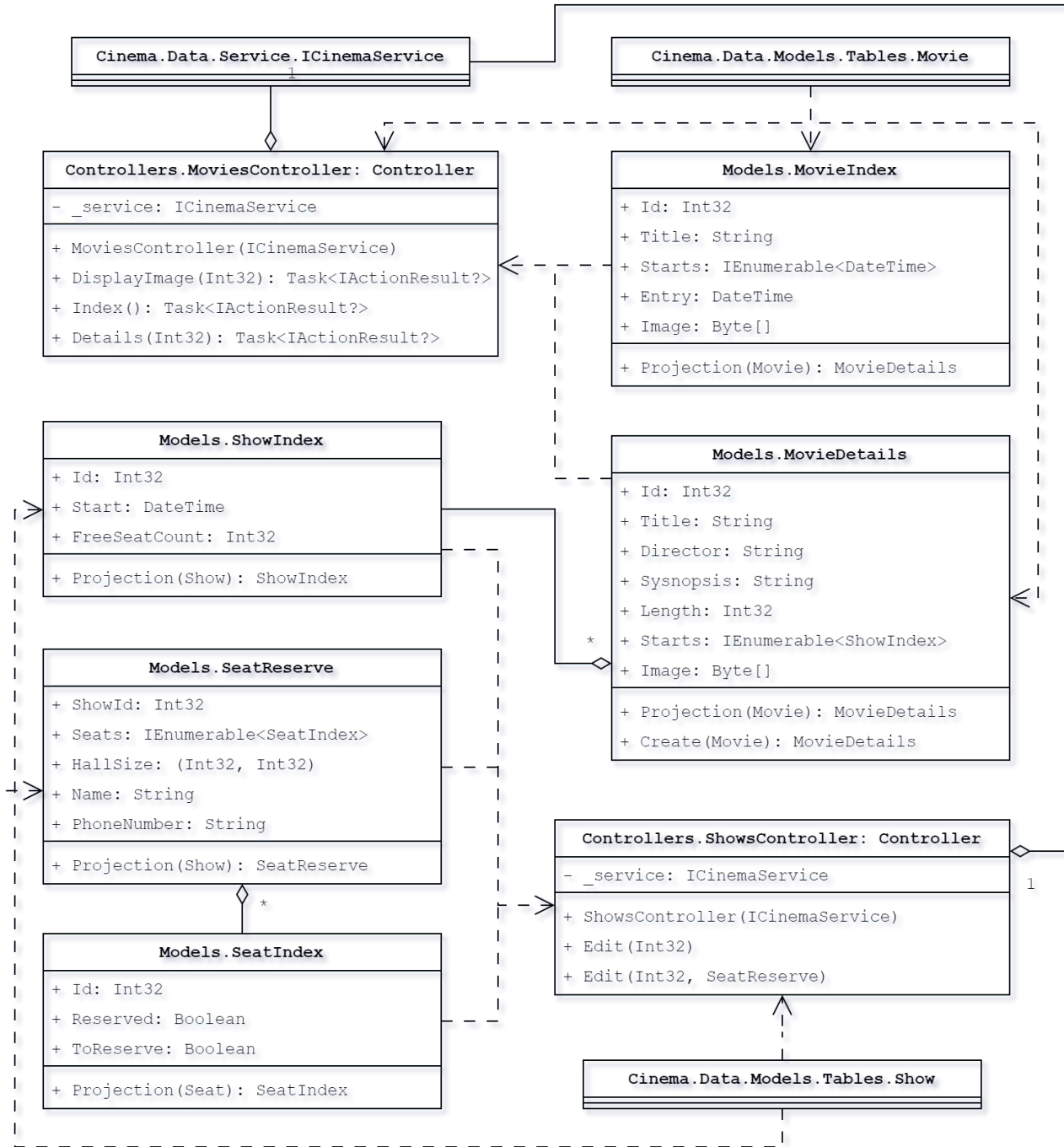
Ábra 3: Komponensek UML diagramja

- A rendszer a részeknek megfelelően négy projektből és azok névtereiből épül fel:
 1. Az adatbázis modell (Models és Migrations névterek) és annak adatátviteli osztályait (DTOs névtér) tartalmazó Cinema.Data projekt
 - (a) Cinema.Data.Models névtér az adatbázis sémájának és azok tábláinak definícióit tartalmazza, melynek tartalmát a Cinema.Data.Migrations névtérbe fordítja egy migráció esetén, mellyel futtatáskor áll fel az adatbázis
 - (b) Cinema.Data.DTOs névtér az adatbázis adatátviteli osztályait tartalmazza, amely előre megírt szűrőként szolgálnak az asztali grafikus felülethez



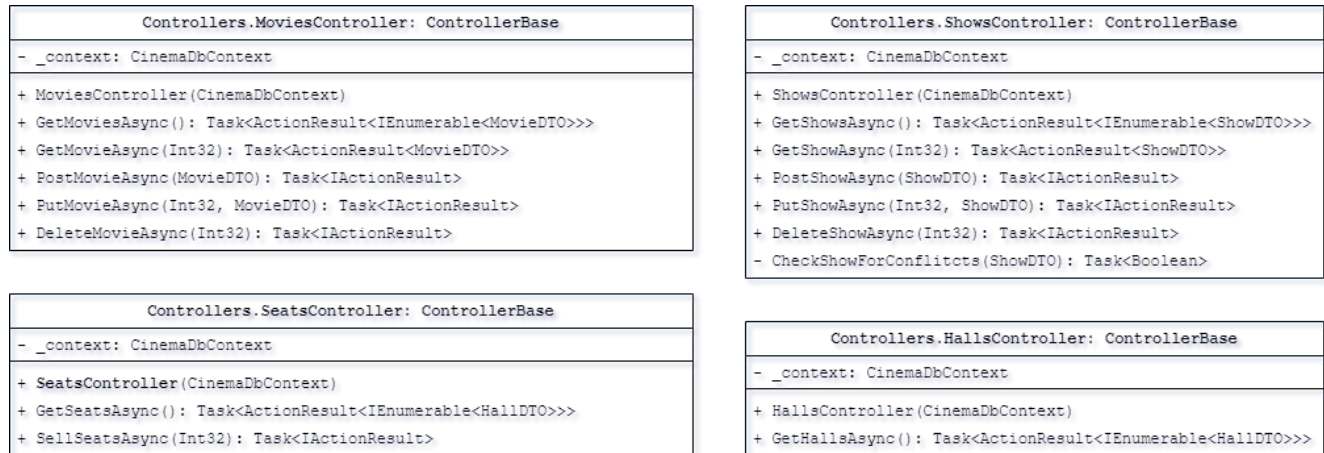
Ábra 4: A Cinema.Data projekt UML osztálydiagramja

2. A kontrollerosztályokat (**Controllers** névtér), az adatátviteli osztályokat (**Models** névtér) és a webes felhasználói felületet (**Views** névtér) tartalmazó **Cinema.Web** projekt
- Cinema.Web.Controllers** névtér az adatbázisból kapott szervízosztály előre megírt lekérdezéseinek az eredményét a **Cinema.Web.Models** névtérben definiált adatátviteli objektumokká alakítva továbbítja a webes felhasználói felület "dinamikus összeállítójának"
 - Cinema.Web.Views** névtér a webes felhasználói felület weblapjainak a "tervezőjeit" tartalmazza .cshtml kiterjesztésben, amely a kapott adatoknak megfelelően szerveroldalon állítja össze a weblapot, majd küldi azt tovább a felhasználónak



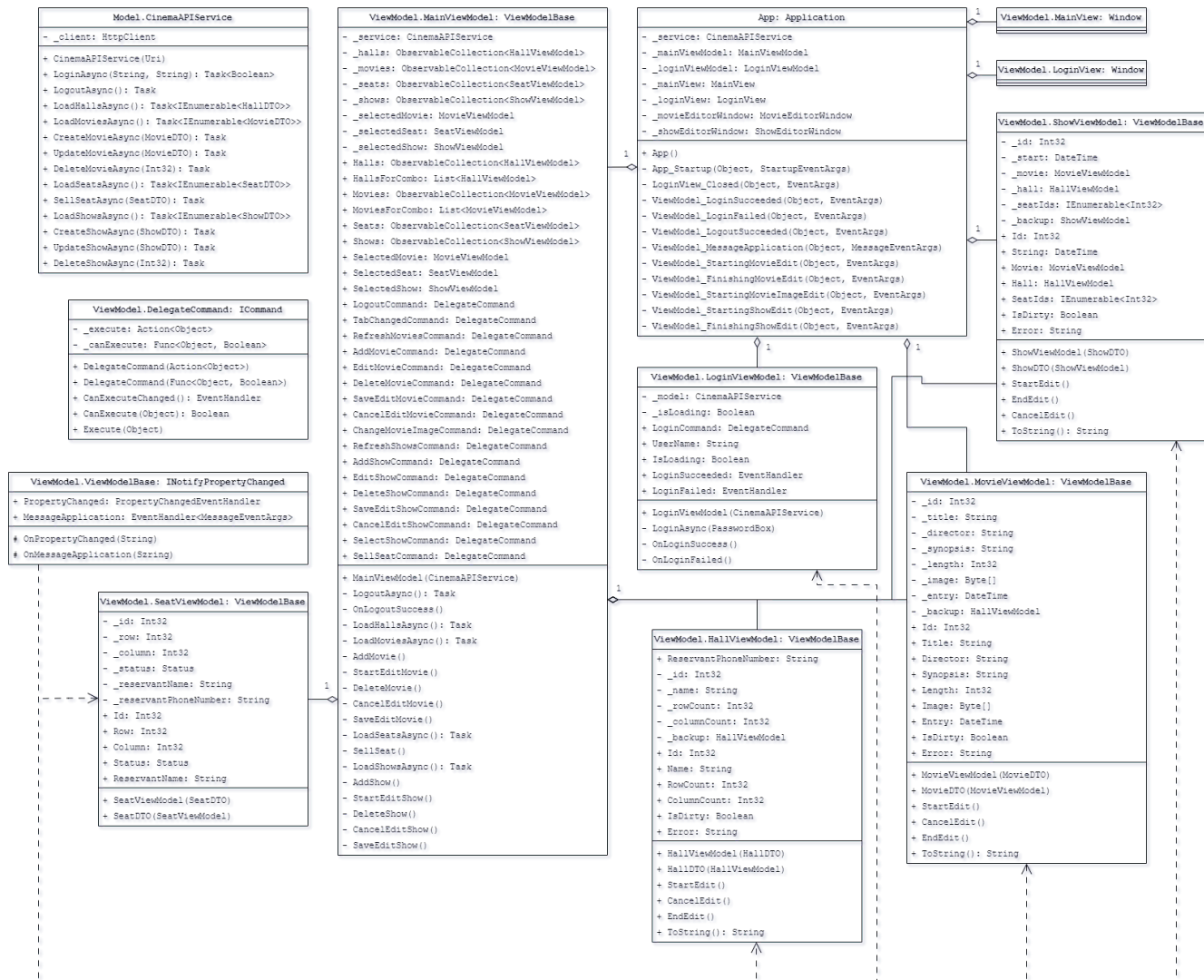
Ábra 5: A Cinema.Web projekt UML osztálydiagramja

3. A webszolgáltatás kontrollerosztályait (`Cinema.WebAPI.Controllers` névtér) tartalmazó `Cinema.WebAPI` projekt, mely az asztali grafikus felület számára továbbít adatokat az adatbázisból.



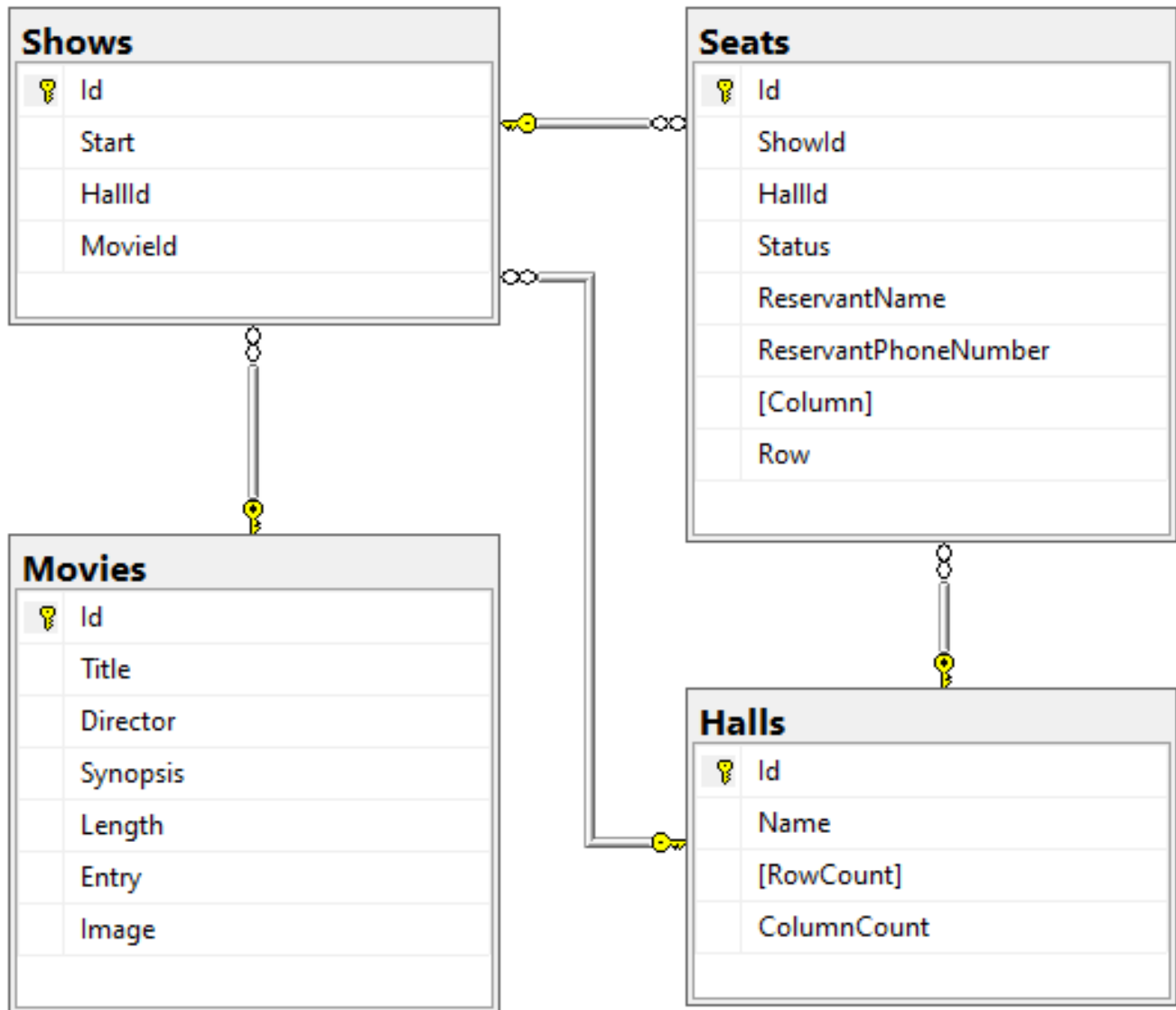
Ábra 6: A `Cinema.WebAPI` projekt UML osztálydiagramja

4. Az asztali grafikus felület modelljét (`Model` névtér), nézetmodelljét (`ViewModel` névtér) és nézetét (`View` névtér) tartalmazó `Cinema.Admin` projekt
- `Cinema.Admin.Model` névtér a webszolgáltatástól megkapott adatokat konvertálja a projekt számára használható formátumúra
 - `Cinema.Admin.ViewModel` névtér definiálja a nézet által használható formátumú adatokat, mely nem csak tárolja azokat, hanem különféle események kezelésével jelzi, ha azok változnak
 - `Cinema.Admin.View` névtér definiálja az ablakok és azok oldalainak megjelenését és tartalmainak elrendezését



Ábra 7: Az Cinema.Admin modeljének és nézetmodeljének sémájának diagramja

Adatbázis felépítése



Ábra 8: Az adatbázis sémájának diagramja

Tesztelés

A webszolgáltatás funkcionalitása egységtesztek segítségével lett ellenőrizve a `Cinema.WebAPI.Test` projektben található `CinemaTest` osztályban. Ehhez 6 függvény áll a rendelkezésre:

- `PostMovieTest`: Új film felvitele esetén az valóban megjelenik az adatbázisban tároltak között.
- `PostMovieWithEmptyTitleTest`: Új film felvitele esetén ha annak nem adunk meg címet, akkor az nem jelenik meg az adatbázisban.
- `PostShowTest`: Új előadás felvitele esetén az valóban megjelenik az adatbázisban tároltak között.
- `PostShowTwiceTest`: Két ugyanabban az időben és helyen rendezett új előadás felvitele esetén az első valóban megjelenik az adatbázisban tároltak között, viszont a második nem.
- `SellSeatTest`: Jegy eladása esetén az sikeresen eladásra kerül és az ülés állapota is eladott lesz.
- `SellSeatTwiceTest`: Jegy kétszeri eladása esetén először sikeres, majd sikertelen lesz az akció.