Beadandó feladat dokumentáció

Feladat

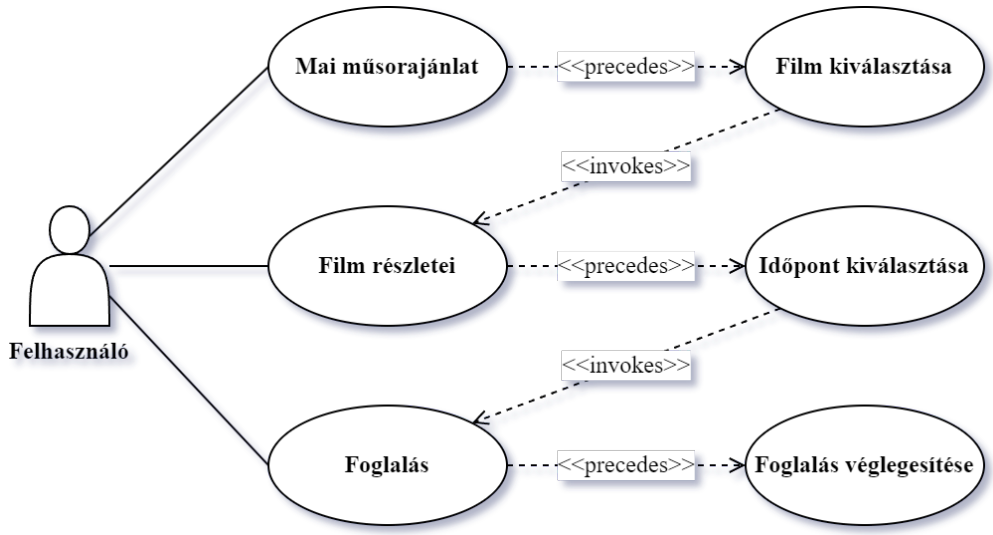
Készítsünk egy mozi üzemeltető rendszert, amelyben egy webes felületen keresztül a nézők tekinthetik meg a moziműsort, valamint rendelhetnek jegyeket.

- 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.

Elemzés

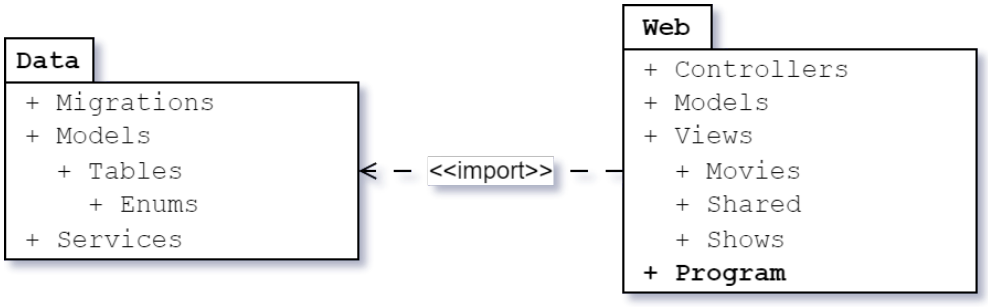
- A feladatot kettő komponens, egy adatbázis és egy webes felhasználói felület, mely utóbbi az Entity Framework segítségével az adatbázishoz kapcsolódva, ASP.NET Core-ban, MVC architekturával kerül megvalósításra.
- A feladat három weblapon fog megjelenni:
  1. weblap a főoldal, melyen megjelenik a napi 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.

Felhasználói esetek



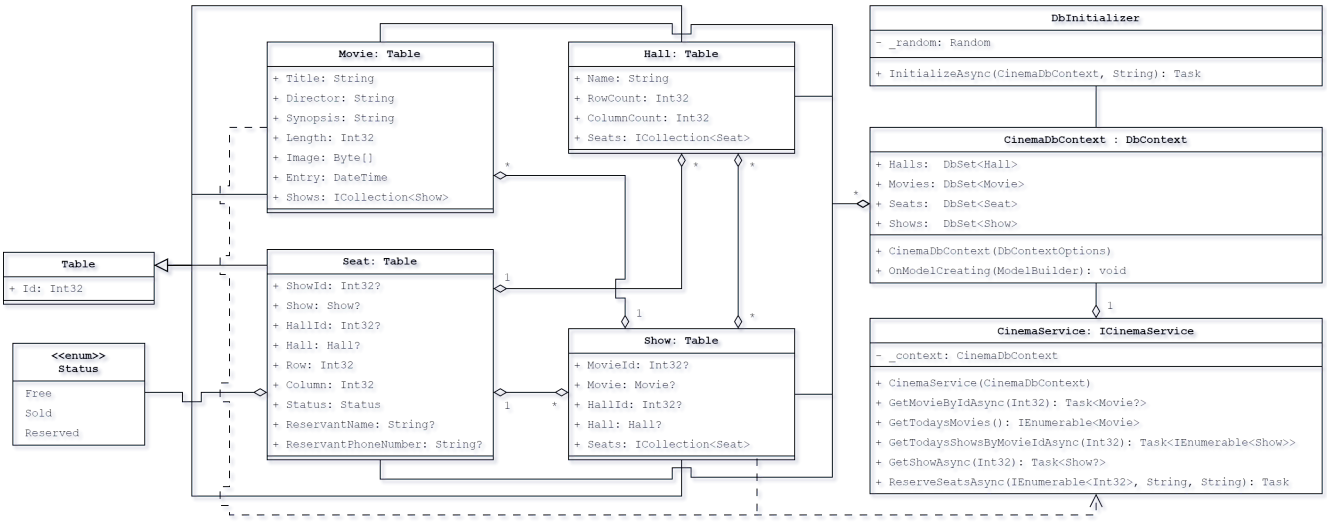
Ábra 1: Felhasználói esetek UML diagramja

Rendszer szerkezete



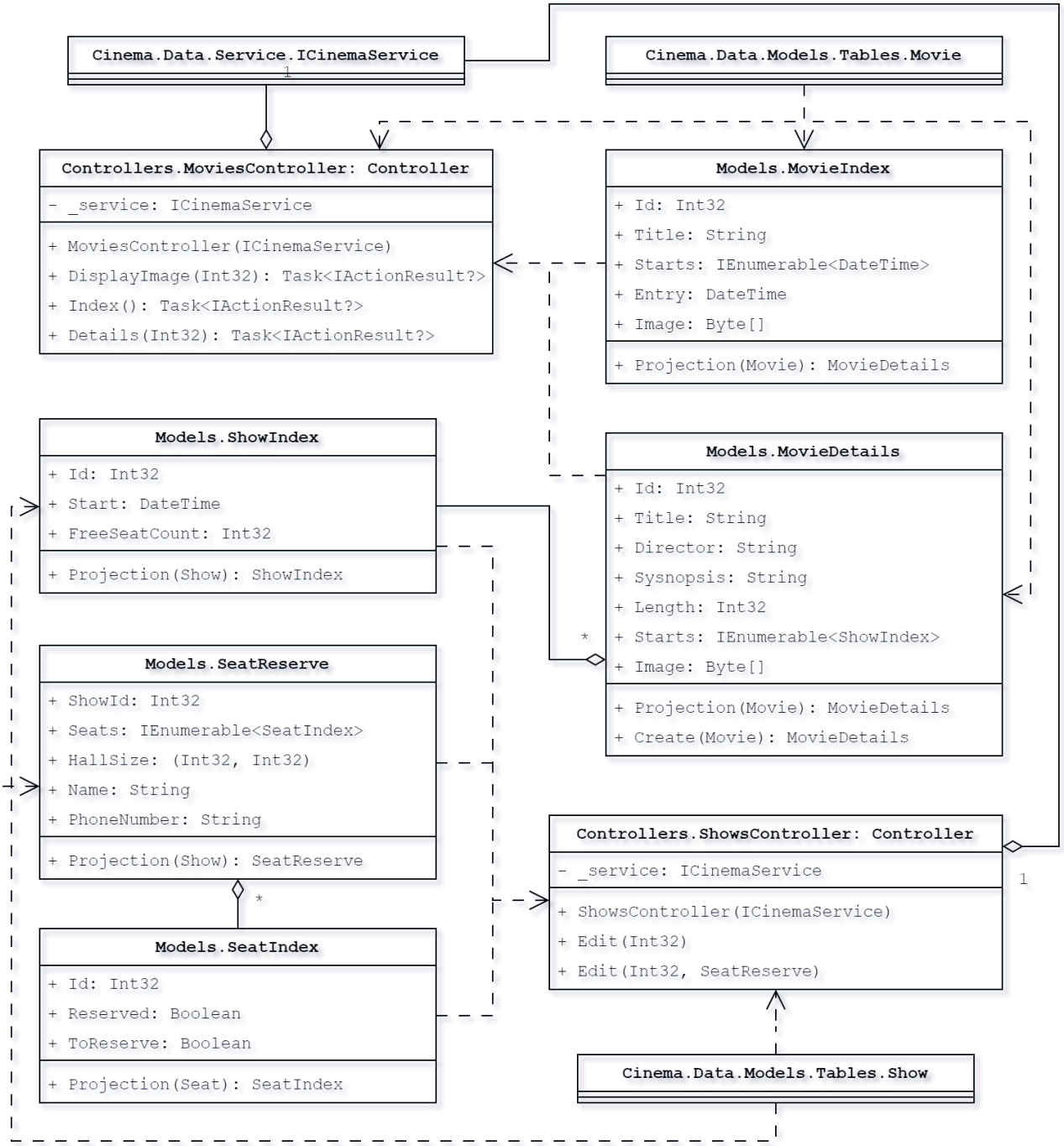
Ábra 2: Komponensek UML diagramja

- A rendszer kettő projektből és azok névtereiből épül fel:
  1. Az adatbázis modellt (`Models` és `Migrations` névterek) és annak szervízosztályát (`Services` 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.Services` névtér az adatbázis szervízosztályát tartalmazza, amely előre megírt lekérdezésekkel kommunikál a webes felület irányába



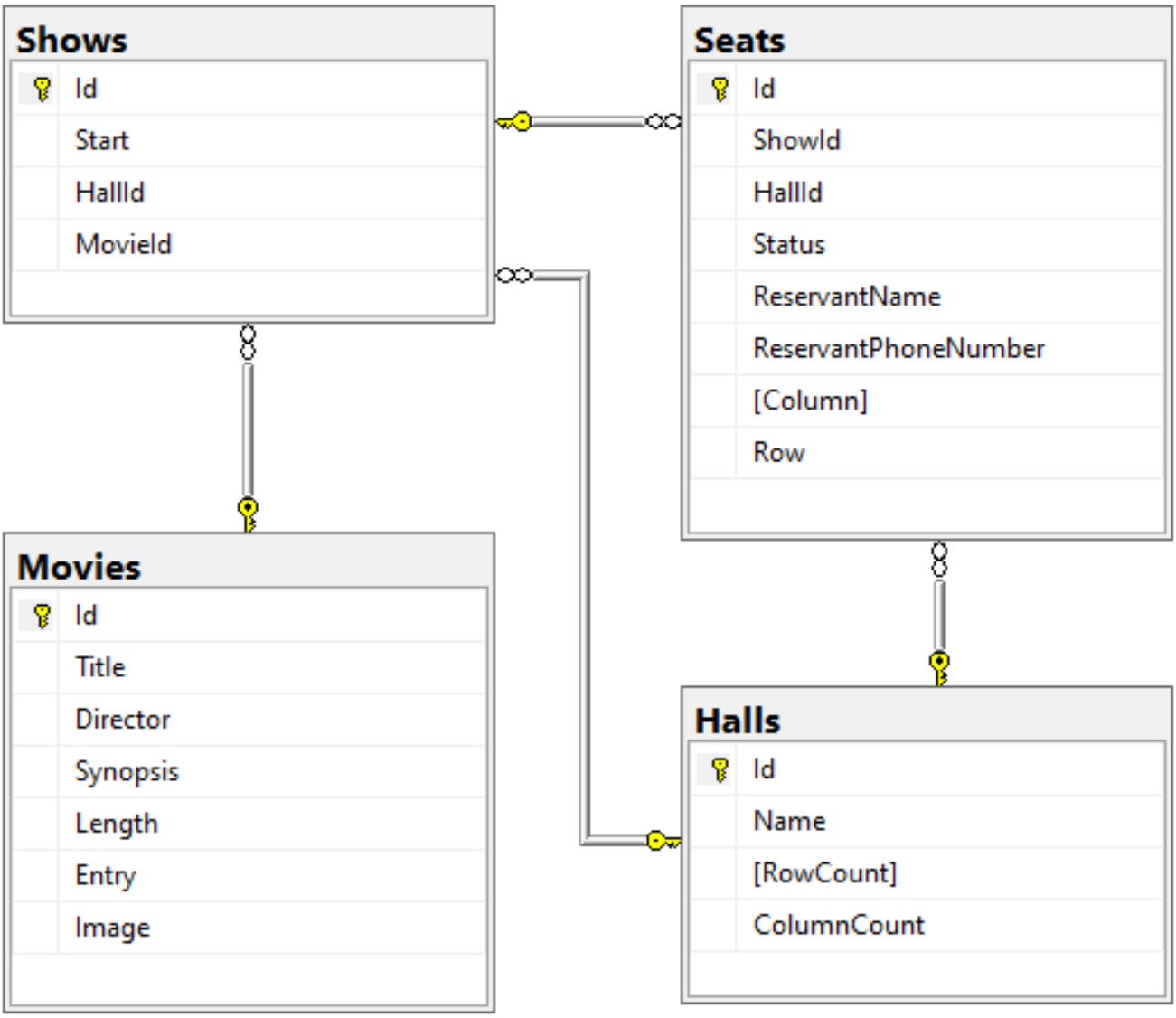
Ábra 3: A `Cinema.Data` projekt UML osztálydiagramja

2. A kontrollerosztályokat (`Controllers` névtér), az adatátviteli osztályokat (`Models` névtér) és a webes felhasználói felületet (`Views` névtér) tartalmazó `Cinema.Web` projekt
   (a) `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"
   (b) `Cinema.Web.Views` névtér a webes felhasználói felület weblapjainak a "tervrajzait" tartalmazza `.cshtml` kiterjesztésben, amely a kapott adatoknak megfelelően szerveroldalon állítja össze a weblapot, majd küldi azt tovább a felhasználónak



Ábra 4: A `Cinema.Web` projekt UML osztálydiagramja

Adatbázis felépítése



Ábra 5: Az adatbázis sémájának diagramja