



Eötvös Loránd Tudományegyetem

Informatikai Kar

Webes alkalmazások fejlesztése

Dokumentáció a 2. beadandóhoz

Csonka Szilvia
V3ZAUG

2019. május 25.

1. Kitűzött feladat

Könyvtári kölcsönző

Készítsük el egy könyvtár online kölcsönzői és nyilvántartó rendszerét, amellyel a látogatók kölcsönzését, előjegyzését, valamint a könyvtárosok adminisztratív munkáját tudjuk támogatni.

A könyvtárosok az asztali grafikus felületen keresztül adminisztrálhatják a könyveket és a kölcsönzéseket.

- A könyvtáros bejelentkezhet (felhasználónév és jelszó megadásával) a programba, illetve kijelentkezhet; a további funkciók csak bejelentkezett állapotban elérhetők.
- Az alkalmazás listázza a könyveket, valamint a hozzájuk tartozó köteteket. Lehetőség van új könyv, illetve kötet rögzítésére.
- A könyvtáros selejtezhet egy kötetet, de csak akkor, ha nincsen aktuálisan kikölcsönözve. Az esedékes jövőbeni előjegyzések törlésre kerülnek és az adott kötet továbbá nem lesz kölcsönözhető.
- Az alkalmazás listázza az aktív kölcsönzéseket és a jövőben esedékes előjegyzéseket. A könyvtáros egy aktív kölcsönzést inaktívvá tehet (visszavitték a könyvet), valamint egy inaktív előjegyzést aktív kölcsönzésnek jelölhet (elvitték a könyvet). Egy kölcsönzés státuszának változtatása nincs a tervezett kezdő és befejező naphoz kötve (gondolva pl. a késedelmes visszavitelre), azonban egy kötetnek egyszerre legfeljebb egy aktív kölcsönzése lehet.

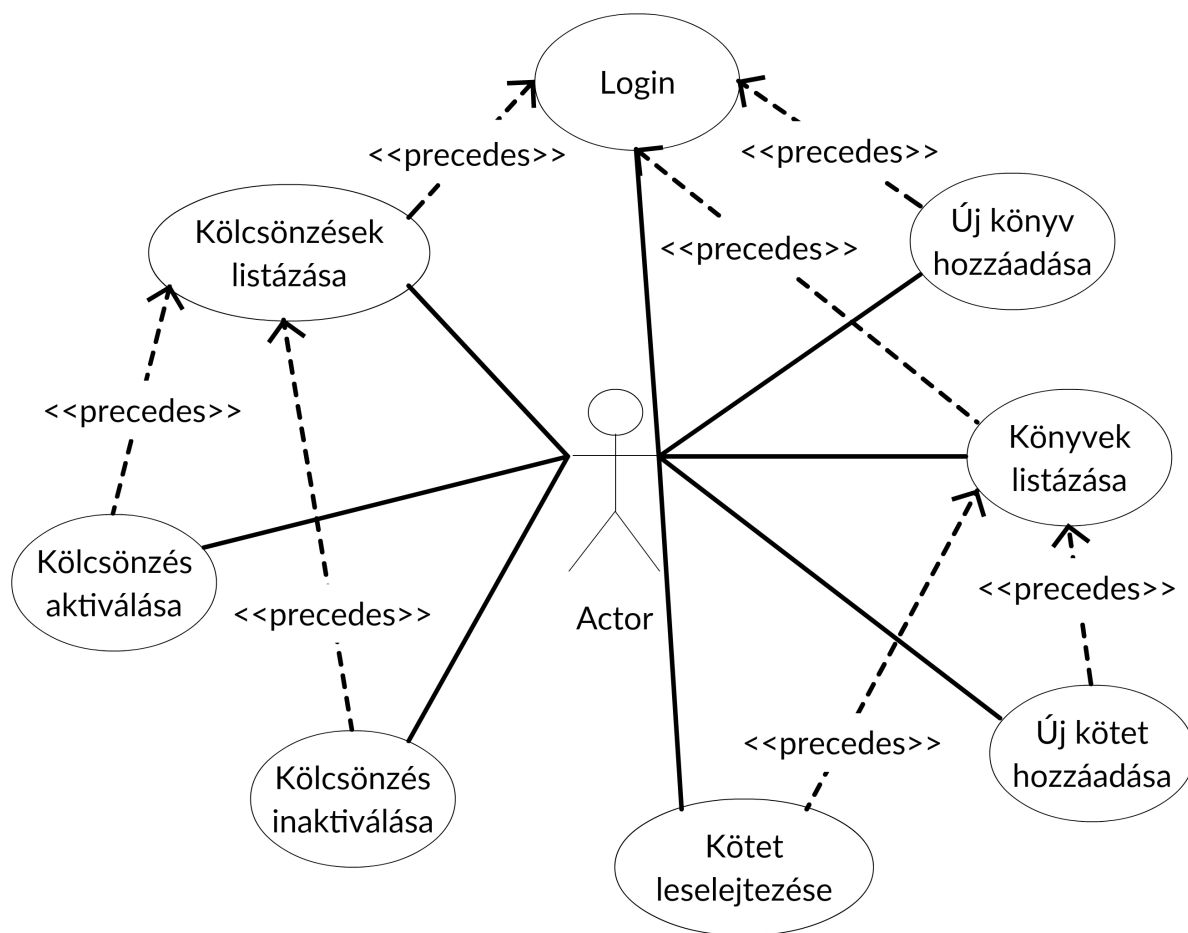
Az adatbázis az alábbi adatokat tárolja:

- könyvek (cím, szerző, kiadás éve, ISBN szám, borítókép);
- kötetek (könyv, könyvtári azonosító);
- kölcsönzések (kötet, látogató, kezdő nap, befejező nap, aktív-e);
- látogatók (név, cím, telefonszám, azonosító, jelszó);
- könyvtárosok (név, azonosító, jelszó).

2. Elemzés

- A program elindítását követően először egy, a bejelentkezésre szolgáló ablakot jelenítünk meg.
 - Az ablakban elhelyezünk két szövegdobozt, amelybe a felhasználó a nevét illetve a jelszavát írhatja.

- A szövegdobozok felett elhelyezünk egy-egy címkét, amely felirata jelöli, hogy mely szövegdobozba mely adatot szükséges megadni.
 - A szövegdobozok és a címkék alatt elhelyezünk egy nyomógombot, amely segítségével a felhasználó a szervernek beküldheti adatait, ahol azok ellenőrzésre kerülnek és sikeres bejelentkezés esetén elérhetővé válik az adminisztrációs felület.
 - Az ablak aljában elhelyezünk egy szövegblokkot, amely a sikertelen bejelentkezést követő hibaüzenet megjelenítésére szolgál.
- A sikeres bejelentkezést követően egy újabb ablakot jelenítünk meg és a bezárjuk a bejelentkeztető ablakot.
 - Az ablak bal felső sarkában elhelyezünk egy menüt, amely segítségével a felhasználó listázhatja a könyveket illetve a kölcsönzéseket, továbbá kijelentkezhet az alkalmazásból.
 - Ha a menüből a felhasználó kiválasztja a kijelentkezést, akkor a program kijelentkezteti őt, bezárja az adminisztrációra szolgáló ablakot, és megnyitja a bejelentkezésre szolgáló ablakot.
 - A kölcsönzések listázása.
 - * Az ablak bal oldalán elhelyezünk egy adattáblázatot, amely a kölcsönzések megjelenítésére szolgál. Ez a táblázat alapból üres.
 - * Amennyiben a felhasználó a menüből kiválasztja a kölcsönzések listázását, az adattáblázatot feltöltjük az adatbázisban szerepelő kölcsönzések adataival.
 - * A táblázat alatt két nyomógombot helyezünk el, melyek segítségével a felhasználó aktiválhat illetve inaktiválhat egy, a táblázatban kijelölt kölcsönzést.
 - * A fent említett két nyomógomb alatt elhelyezünk egy szövegblokkot, amelyben hibaüzenetet jelenítünk meg, amennyiben a kijelölt kölcsönzés aktiválása vagy inaktiválása sikertelen volt, vagy nem volt kiválasztva kölcsönzés, illetve arról is értesítjük a felhasználót, amennyiben a művelet sikeres volt. Ez utóbbi esetben frissítjük a táblázatbeli adatokat, hogy a legutóbbi művelet eredményének megfelelően jelenjenek meg benne az adatok.
 - A könyvek listázása.
 - * Az ablak jobb oldalán, az utóbbihoz hasonló adattáblázatot helyezünk el, amely azonban a könyvek adatainak megjelenítésére szolgál.
 - * Amennyiben a felhasználó a menü segítségével listázza a könyveket, az adattáblázatban megjelenítjük az aktuálisan az adatbázisban tárolt könyvek adatait.



1. ábra. Felhasználói esetek diagramja

- * A táblázat alatt a kijelölt könyv adatait jelenítjük meg szövegdobozokban, amelyek módosíthatóak.
- * Az adatok alatt három nyomógombot helyezünk el, amelyek segítségével törölhetjük a kijelölt könyvhöz tartozó kiválasztott kötetet amennyiben az nincsen kölcsönözve, a kijelölt könyvhez új kötetet adhatunk hozzá, illetve új könyvet hozhatunk létre a szövegdobozban szereplő adatok módosításával.
- * A nyomógombok alatt elhelyezünk egy szövegblokkot, amelyben hibaüzenetet jelenítünk meg ha valamely művelet sikertelen volt illetve tájékoztatjuk a művelet sikerességéről a felhasználót amennyiben ez indokolt.

2.1. Tervezés

- **Programszerkezet:** A programot modell/nézet/nézetmodell (Model-View-ViewModel, MVVM) architektúra valósítjuk meg, amely egy többrétegű felépítést definiál.
- **Perzisztencia:** a `Library.Admin` projekt nem tartalmaz saját adatbázist, hanem a szerverként funkcionáló `Library` projektbeli adatbázisokat használja.

- Az adatkezelést a `LibraryContext` adatbázis valósítja meg, amely táblázatokban tárolja a szükséges információkat, melynek felépítését már kifejtettük az első beadandó dokumentációjában. Ezen programrész szempontjából fontosabb a `LibrarianContext` adatbázis, amelybe kiszerveztük a `Librarian` táblázatot, amely az adminisztrációs felület felhasználóinak adatait tárolja és segíti elő a bejelentkezéshez szükséges autentikációt.
 - * `Librarian` táblázat
 - A könyvtárosok adatait tárolja.
 - Felépítése: `ID`, `Name`, `Password`, további autentikációhoz szükséges, örökölt oszlopok.
 - * `AdminController`: A `Library` projekt kontrollere, amely az adatbázisokbeli adatokhoz való hozzáférést biztosítja és a szerverrel való kommunikációt segíti elő.
 - * A `Library.Data` projektbeli adattípusok segítik elő a `Library.Admin`, kliensként funkcionáló projekt és `Library`, szerverként funkcionáló projekt közötti kommunikációt.
- Modell: a logikát a `CommunicateModel` valósítja meg, amely a szerverrel kommunikál és annak segítségével biztosítja a program funkcióit. Metódusai:
 - Listázások:
 - * `ListBooks`: `BookDTO` típus által reprezentálva adja vissza az adatbázisbeli könyvek listáját.
 - * `ListLendings`: `LendingDTO` típus által reprezentálva adja vissza az adatbázisbeli kölcsönzések listáját.
 - Adatbázist módosító metódusok:
 - * `DeleteVol`: a megadott `ID`-val rendelkező kötetet törli az adatbázisból. Amennyiben a művelet sikeres volt, logikai `igaz`-zal tér vissza.
 - * `AddVol`: a megadott `ID`-val rendelkező könyvhöz ad még egy további kötetet. Amennyiben a művelet sikeres volt, logikai `igaz`-zal tér vissza.
 - * `AddBook`: a megadott `BookDTO` által reprezentált könyvet adja hozzá az adatbázishoz. Amennyiben a művelet sikeres volt, logikai `igaz`-zal tér vissza.
 - * `ActivateLending`: a megadott `ID`-val rendelkező kölcsönzést aktiválja. (Azaz az adatbázisban a megfelelő kölcsönzés `IsActive` logikai típusú adattagját `igaz`-ra állítja.) Amennyiben a művelet sikeres volt, logikai `igaz`-al tér vissza.
 - * `InactivateLending`: a megadott `ID`-val rendelkező kölcsönzést inaktíválja. Amennyiben a művelet sikeres volt, logikai `igaz`-al tér vissza.

- Felhasználó autentikációját megvalósító metódusok:
 - * **Login:** a megadott felhasználónév, jelszó párost ellenőrzi. Amennyiben az autentikáció sikeres, a metódus gond nélkül végrehajtódik. Sikertelen autentikáció esetén kivételt vált ki, amelyet később kezelünk és hibaüzenetet küldünk a felhasználó felé.
 - * **Logout:** kijelentkezteti az aktuálisan bejelentkezett felhasználót. Sikeres kijelentkezés esetén gond nélkül végrehajtódik a metódus, ellenkező esetben kivételt vált ki.
 - * **Register:** alapértelmezett teszt felhasználót regisztrál, amely a manuális tesztelést segíti elő. A végleges program által nem használt metódus.
- Nézetek
 - A nézetet két ablak által valósítjuk meg. Az egyik a felhasználó autentikációjára szolgál, a másik a könyvtári adminisztrációs funkciókat teszi elérhetővé.
 - **MainWindow:** az alkalmazás futtatásakor először megjelenő ablak, azaz kezdőablak, amely a felhasználó autentikációjára szolgál.
 - **AdminWindow:** az adminisztrációs alkalmazás szolgáltatásait biztosító grafikus felület.
- Nézetmodell: a **ViewModel** osztály valósítja meg a **DelegateCommand** osztály segítségével.
 - Privát adattagok:
 - * **_model:** a **CommunicateModel** modell osztály egy példánya, amelyen keresztül elérhetőek annak metódusai.
 - * **_adminWindow:** a **AdminWindow** egy példánya, amely segítségével létrehozható az ablak.
 - * **_selectedBookIndex:** az aktuálisan kijelölt könyv indexét tartalmazza.
 - * **_selectedLendingIndex:** az aktuálisan kijelölt kölcsönzés indexét tartalmazza.
 - * **_selectedBook:** az aktuálisan kijelölt könyvet tartalmazza.
 - * **_selectedLending:** az aktuálisan kijelölt kölcsönzést tartalmazza.
 - * **_books:** az adatbázisból betöltött könyvek listáját tartalmazza.
 - * **_lendings:** az adatbázisból betöltött kölcsönzések listáját tartalmazza.
 - A nézethez kötött (*bind*) tulajdonságok (*property*):
 - * **LoginUserName:** a bejelentkezésre szolgáló ablak felhasználónév megadására hivatott szövegdozoz tartalmához van kötve, a nézetmodell ennek segítségével éri el a felhasználó által megadott nevet.

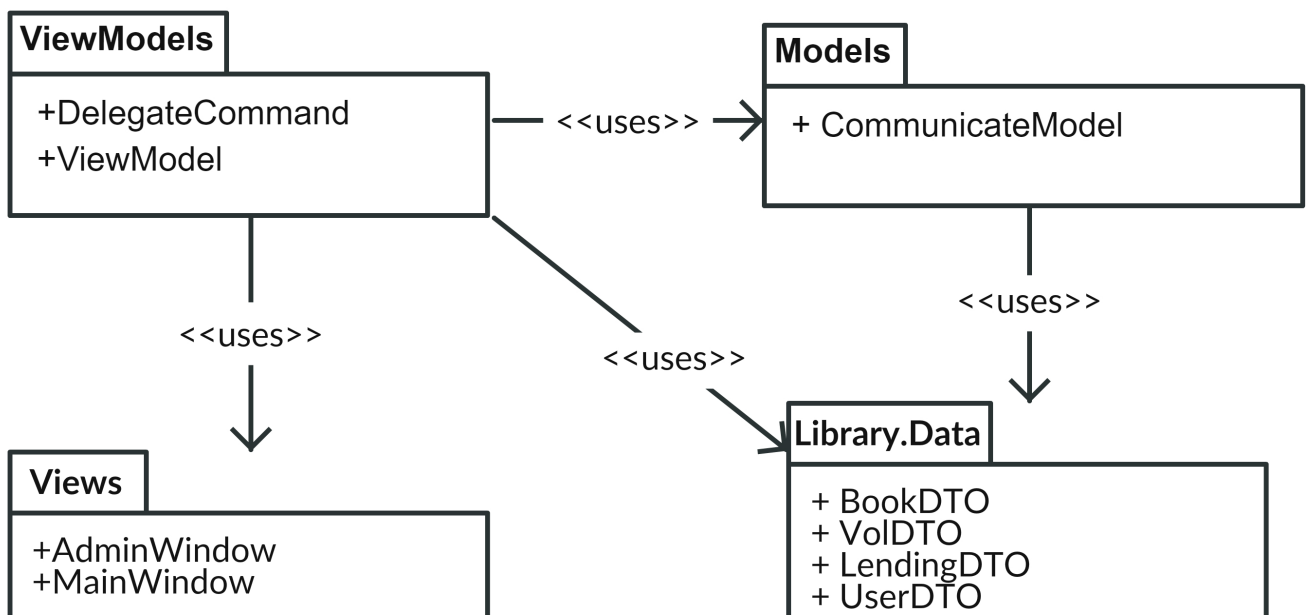
- * **LoginUserPassword**: a bejelentkezésre szolgáló ablak jelszó megadására hivatott szövegdozoz tartalmához van kötve, a nézetmodell ennek segítségével éri el a felhasználó által megadott jelszavat.
- * **LoginErrorMessage**: a bejelentkeztetés során felmerült hibák közlésére szolgál. A bejelentkeztető ablak alján lévő szövegblokkhoz van kötve.
- * **LendingErrorMessage**: a kölcsönzések aktiválásával illetve inaktiválásával kapcsolatos üzeneteket továbbítja a felhasználónak. A kölcsönzések alatt, az **AdminWindow** bal oldalának alján elhelyezkedő szövegblokkhoz van kötve.
- * **BookErrorMessage**: a könyvek módosítására szolgáló metódusokkal kapcsolatos üzeneteket továbbítja a felhasználónak. A könyvek táblázata alatt, az **AdminWindow** jobb oldalának alján elhelyezkedő szövegblokkhoz van kötve.
- * **CurrentVolID**: A könyvek listája alatt elhelyezkedő adatok közül a kötet azonosítóját tartalmazó mezőhöz van kötve. Segítségével érhető el a törlésre kiválasztott kötet.
- * **SelectedBookIndex**: A kiválasztott könyv indexét tartalmazza. Tartalmát a könyvek táblázatához való kötöttsége segítségével automatikusan kapja a nézettől.
- * **SelectedLendingIndex**: A kiválasztott kölcsönzés indexét tartalmazza. Tartalmát a kölcsönzések táblázatához való kötöttsége segítségével automatikusan kapja a nézettől.
- * **SelectedBook**: A nézetmodellbeli hozzá tartozó index változásakor automatikusan töltjük bele a kiválasztott könyv adatait.
- * **SelectedLending**: A nézetmodellbeli hozzá tartozó index változásakor automatikusan töltjük bele a kiválasztott kölcsönzés adatait.
- * **Books**: A könyvek táblázatához van kötve, ennek tartalmát jeleníti meg a táblázat, és ezt a listát töltjük fel, amikor a könyvek listázását végezzük.
- * **Lendings**: A kölcsönzések táblázatához van kötve, ennek tartalmát jeleníti meg a táblázat, és ezt a listát töltjük fel, amikor a kölcsönzések listázását végezzük.

– Utasítások:

- * **LoginCommand**: a bejelentkeztetésre szolgáló utasítás. A modell **Login** metódusát használja.
- * **LogoutCommand**: a kijelentkeztetésre szolgáló utasítás. A modell **Logout** utasítását használja.
- * **ListBooksCommand**: a könyvek listázása szolgáló utasítás. A modell **ListBooks** metódusát használja.

- * **ListLendingsCommand**: a kölcsönzések listázására szolgáló utasítás. A modell **ListLendings** utasítását használja.
- * **DeleteVolCommand**: adott kötet törlésére szolgáló utasítás. A modell **DeleteVol** utasítását használja.
- * **AddBookCommand**: adott könyv adatbázisba vitelére szolgáló utasítás. A modell **AddBook** utasítását használja.
- * **AddVolCommand**: adott könyvhöz új kötet adására szolgáló utasítás. A modell **AddVol** utasítását használja.
- * **ActivateLendingCommand**: adott kölcsönzés aktiválására szolgáló utasítás. A modell **ActivateLending** utasítását használja.
- * **InactivateLendingCommand**: adott kölcsönzés inaktiválására szolgáló utasítás. A modell **InactivateLending** utasítását használja.

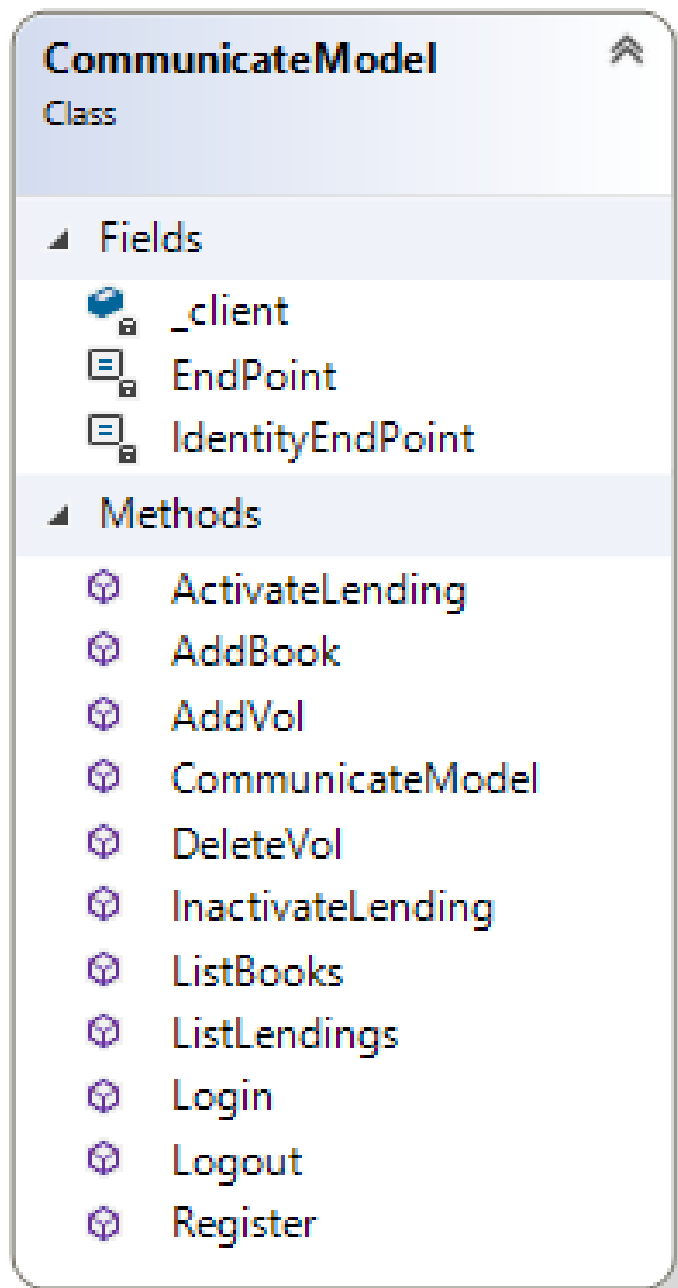
3. Komponensdiagram



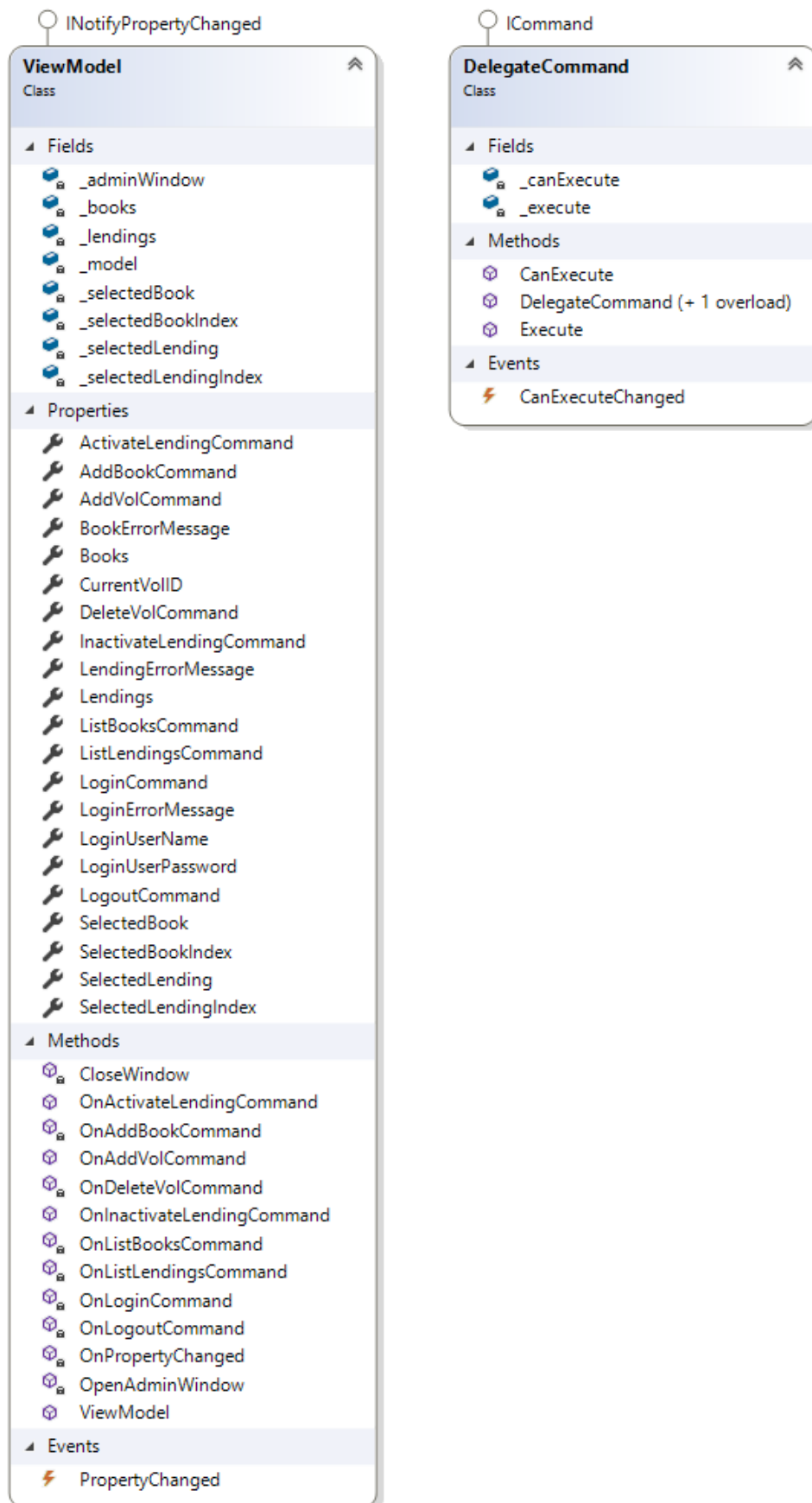
2. ábra. Komponensdiagram

4. Osztálydiagram

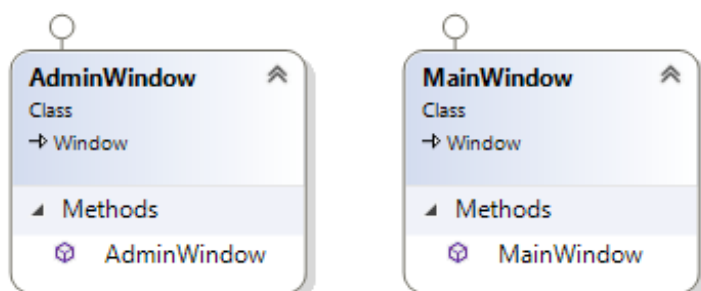
4.1. Modell



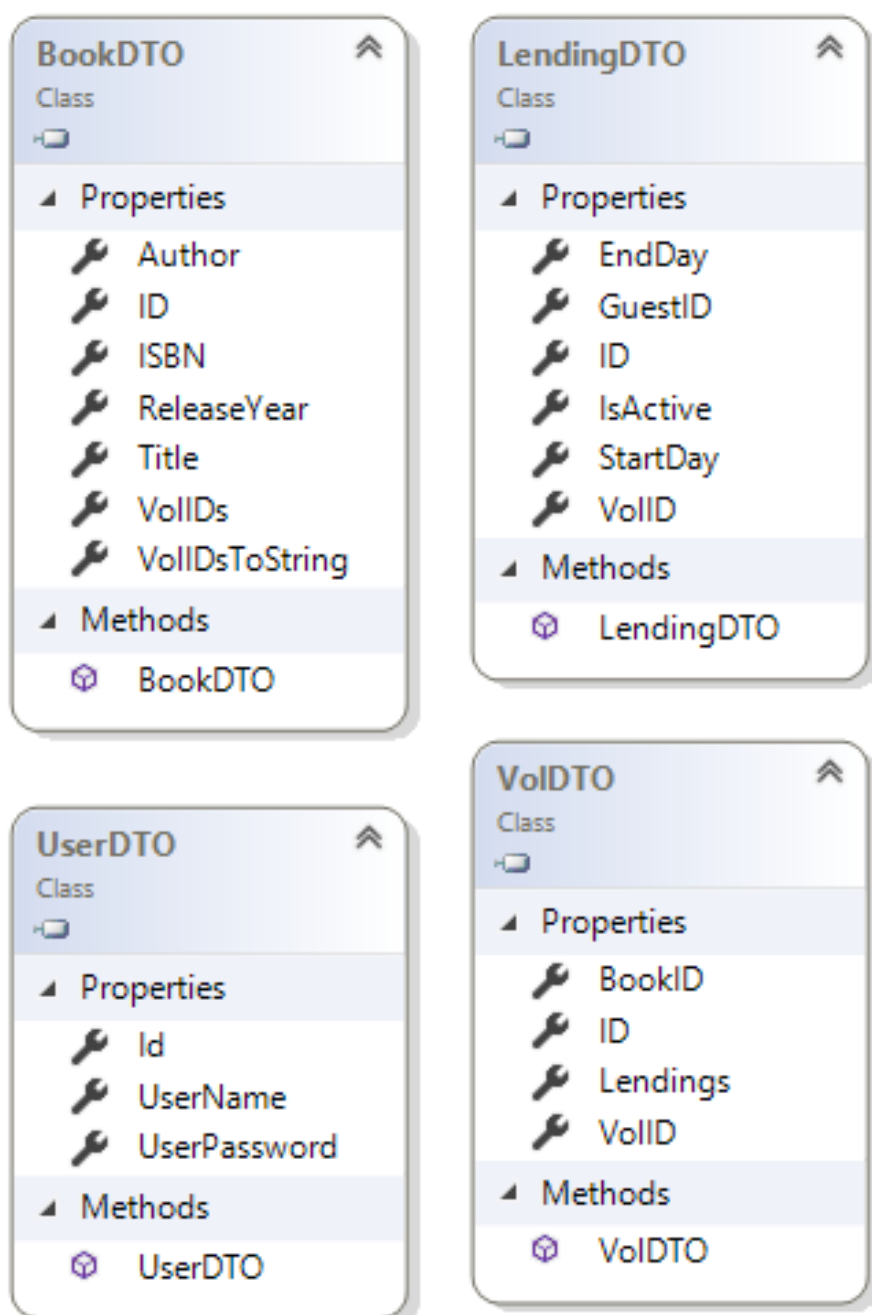
4.2. Nézetmodell



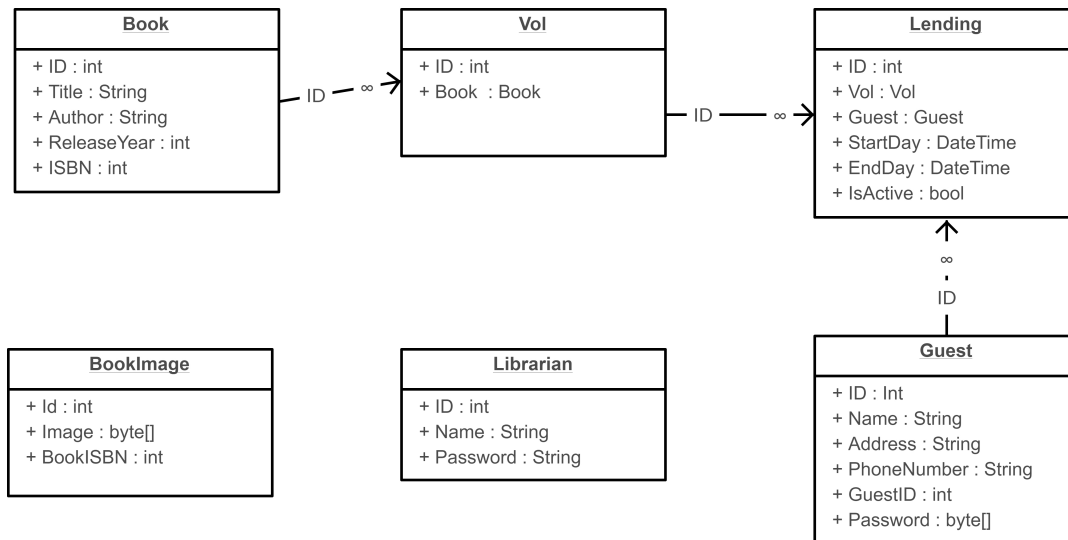
4.3. Nézet



4.4. Adatfelépítés



5. Egyedkapcsolati diagram



6. Tesztesetek

A tesztelést egy ideiglenes adatbázis segítségével valósítjuk meg, amelyet a tesztelés előtt inicializálunk, majd befejeztével törölünk minden alkalommal.

Teszt eset neve	Tesztelt metódus	Teszt leírása
ListBooksTest	ListBooks	Könyvek listázásának tesztelése.
ListLendingsTest	ListLendings	Kölcsönzések listázásának tesztelése.
DeleteVolTestWithOneVol	DeleteVol	Egy kötet törlése az ideiglenes adatbázisból.
DeleteVolTestWithTwoVols	DeleteVol	Két kötet törlése az ideiglenes adatbázisból.
AddOneBookTest	AddBook	Egy könyv hozzáadása az ideiglenes adatbázishoz.
AddTwoBookTest	AddBook	Két könyv hozzáadása az ideiglenes adatbázishoz.
AddOneVolTest	AddVol	Egy kötet hozzáadása az ideiglenes adatbázishoz.
AddTwoVolTest	AddVol	Két kötet hozzáadása az ideiglenes adatbázishoz.
ActivateLendingTest	ActivateLending	Két kölcsönzés aktiválása az ideiglenes adatbázisban.
InactivateLendingTest	InactivateLending	Két kölcsönzés inaktiválása az ideiglenes adatbázisban.
ActivateAndInactivateLendingTest	ActivateLending, InactivateLending	Egy kölcsönzés aktiválása, majd inaktiválása majd ismételt aktiválása az ideiglenes adatbázisban.