

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

2020.05.22.

DOKUMENTÁCIÓ

2. beadandó feladat

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Herczegfalvy Veronika
L6GLK1

Feladatleírás

Könyvtári kölcsönző

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

1. részfeladat

- A webes felületen a látogatók adhatnak le online előjegyzéseket, illetve böngészhetik a kínálatot.
- A főoldalon a könyvtárban elérhető könyvek kerülnek kilistázásra, alapértelmezetten népszerűségi sorrendben (kölcsönzések száma), de a felhasználó választhat cím szerinti lexikografikus rendezést is. Egy oldalon legfeljebb 20 könyv jelenik meg, a többi lapozással lehet elérni. A könyvek címmel, (első) szerzővel, kiadási évvel, ISBN számmal és borítóképpel rendelkeznek, amelyek megjelenítésre kerülnek.
- A látogatók a könyvek között cím és szerző szerint, szabad szavas beviteli mező segítségével szűrhetnek.
- Egy könyvet kiválasztva megjelenítésre kerülnek a kötetei, valamint esetlegesen aktív kölcsönzésük és jövőbeni előjegyzéseik.
- A látogatók regisztrációt (név, telefonszám, e-mail cím, felhasználónév, jelszó, megerősített jelszó) és bejelentkezést követően előjegyzést adhatnak le egy Webes alkalmazások fejlesztése 2019/2020 tavaszi félév 9 kötetre, a kölcsönzés tervezett kezdő és befejező napját megadva. Az előjegyzés nem fedhet át aktív kölcsönzéssel vagy más előjegyzéssel.

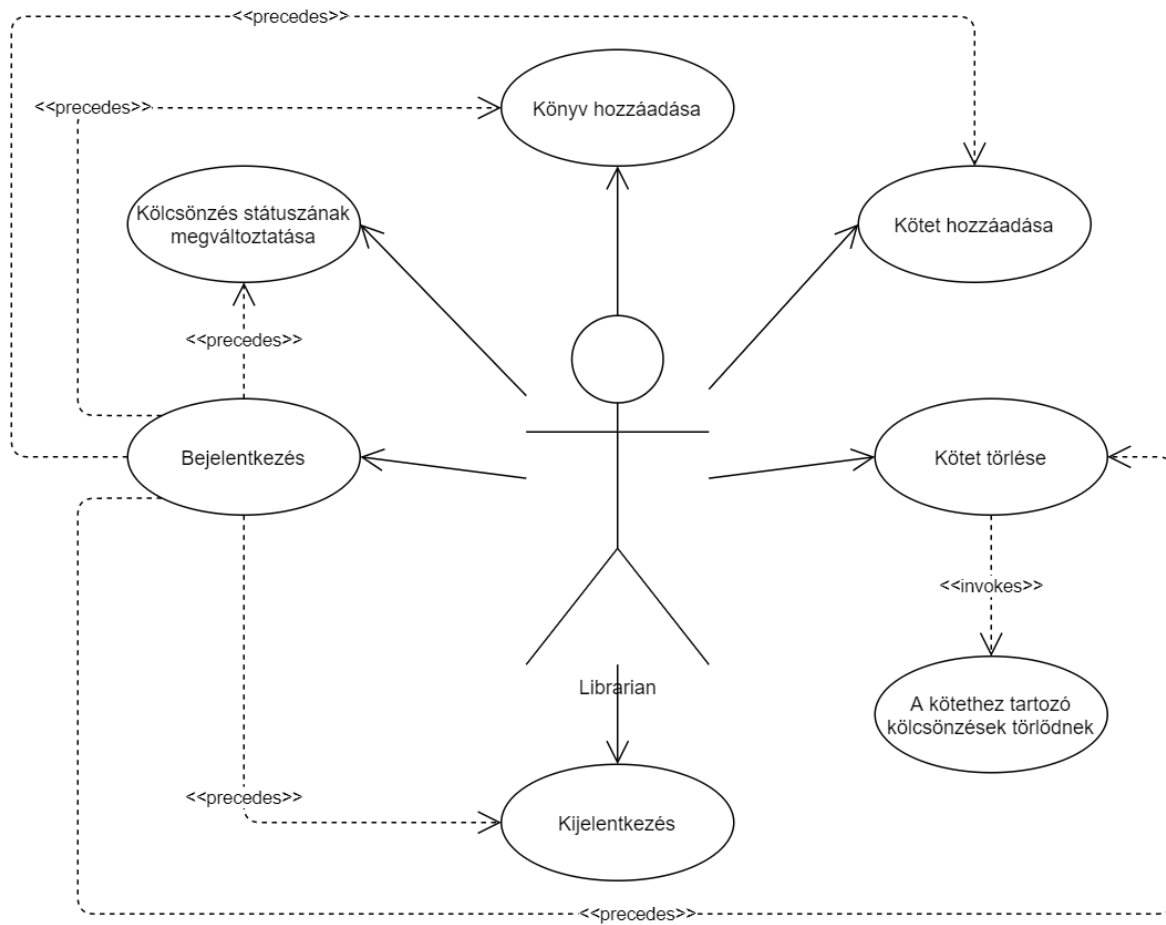
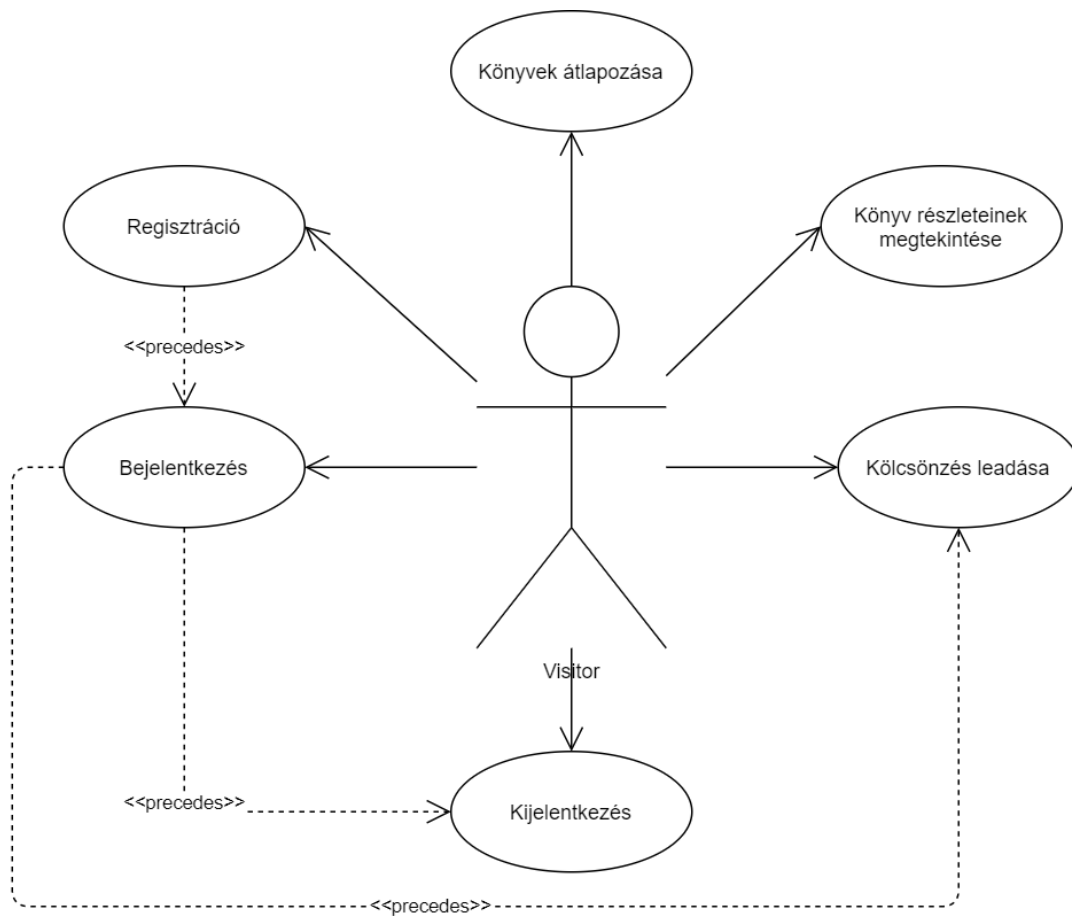
2. részfeladat

- 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 elérhetőek.
- 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.

Javasolt adatbázissémák

- 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)

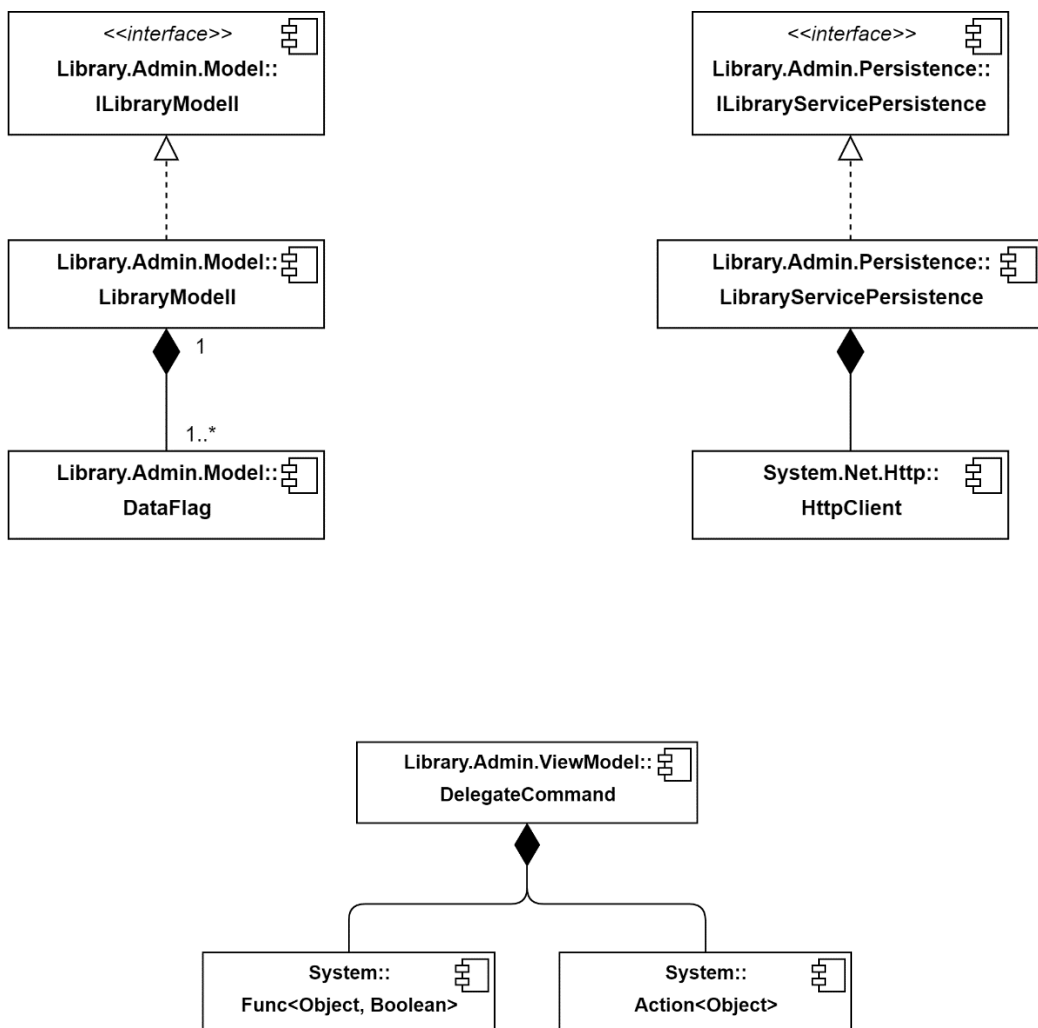
Felhasználói esetek

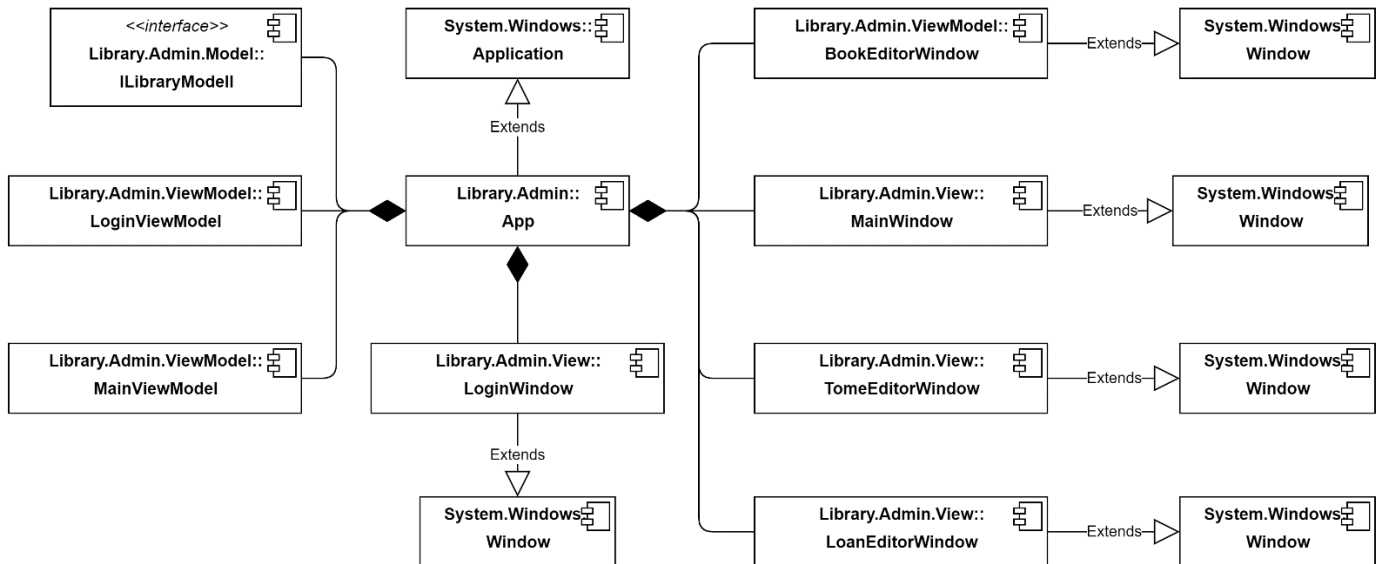
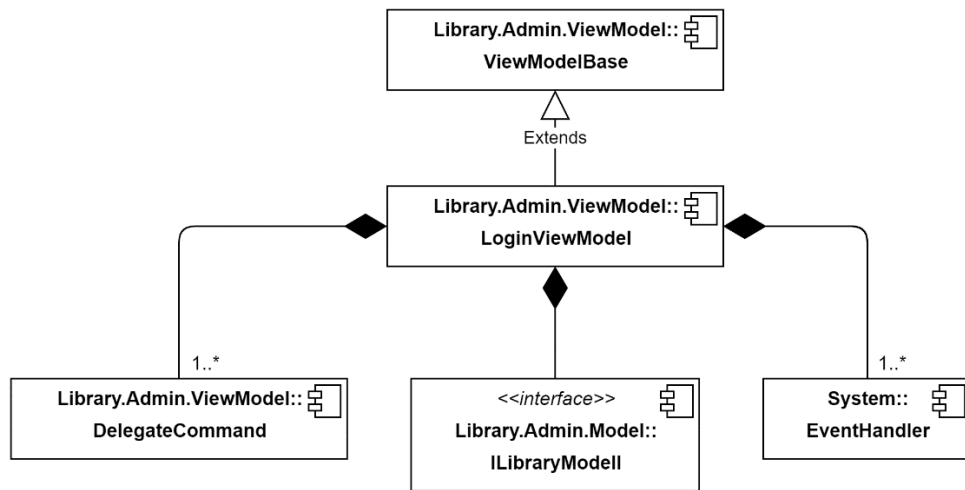


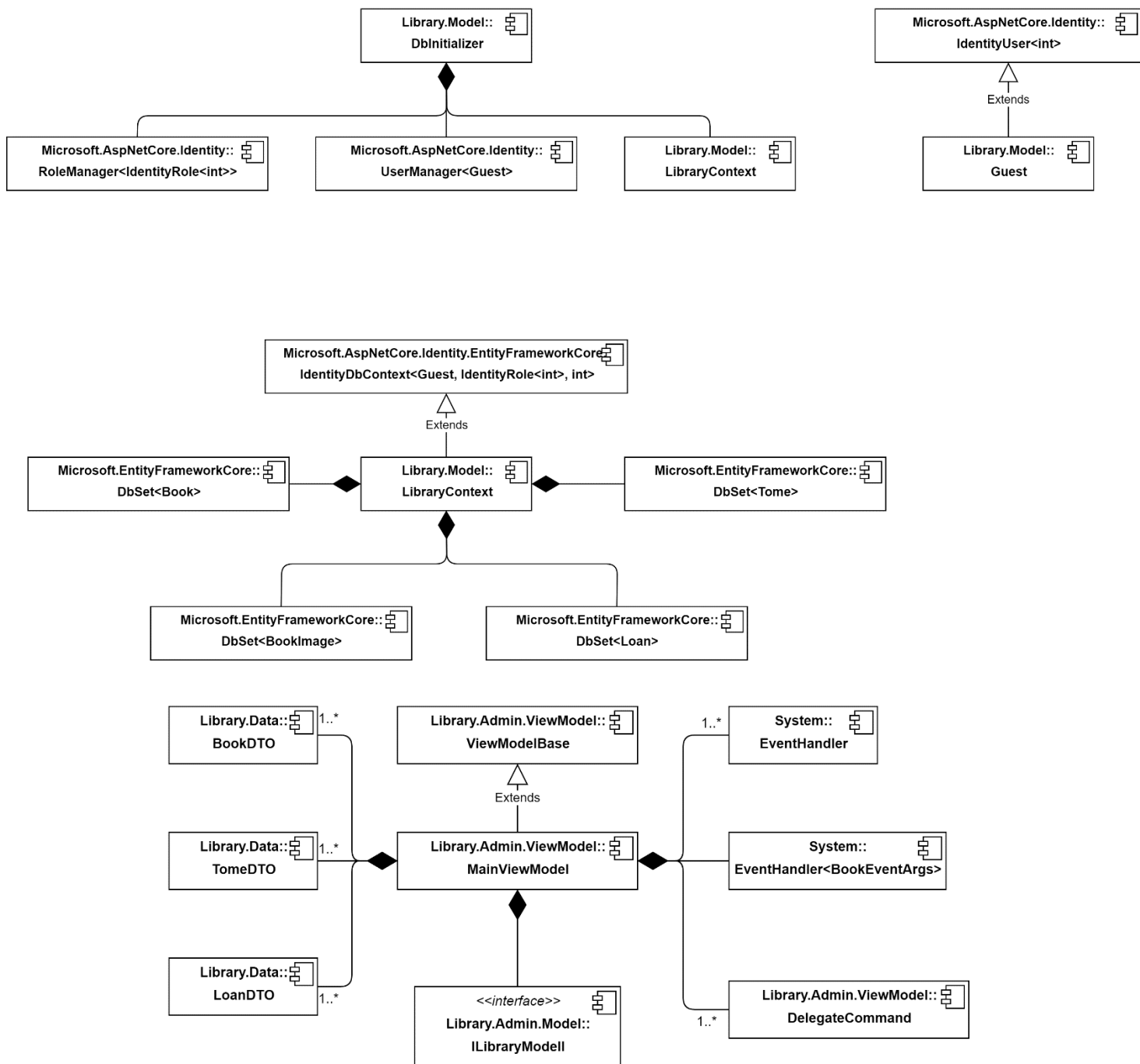
Tervezés

A programban négy nagy egységet lehet elkülöníteni:

- az adatbázist, amely tartalmazza az alkalmazás által használt adatokat (beleértve a felhasználói információkat);
- a webes felhasználói felületet ASP.NET Core-ban, MVC architektúrával, amely biztosítja a felhasználói funkciókat és Entity Framework segítségével kapcsolódik az adatbázishoz;
- a webszolgáltatást ASP.NET Core-ban, MVC architektúrával, amely az adminisztratív funkciókat biztosítja, és Entity Framework segítségével kapcsolódik az adatbázishoz;
- és az adminisztratív klienst .NET Core-ban, WPF grafikus felülettel, MVVM architektúrában, amely az adminisztrátori funkciókat biztosítja, és a webszolgáltatáshoz csatlakozik.





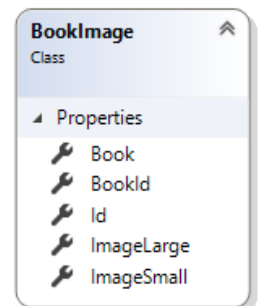
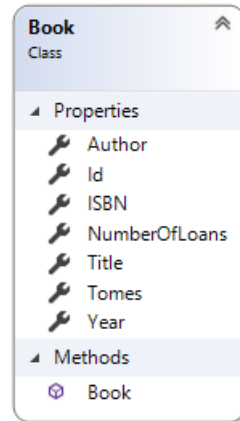
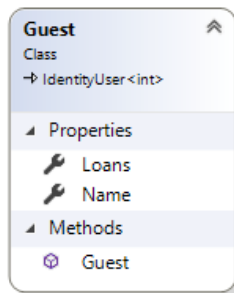
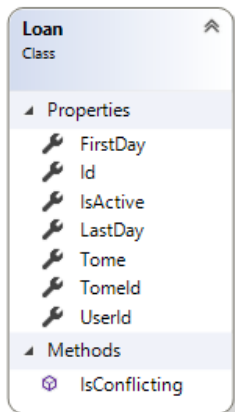
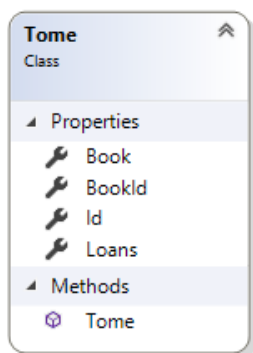


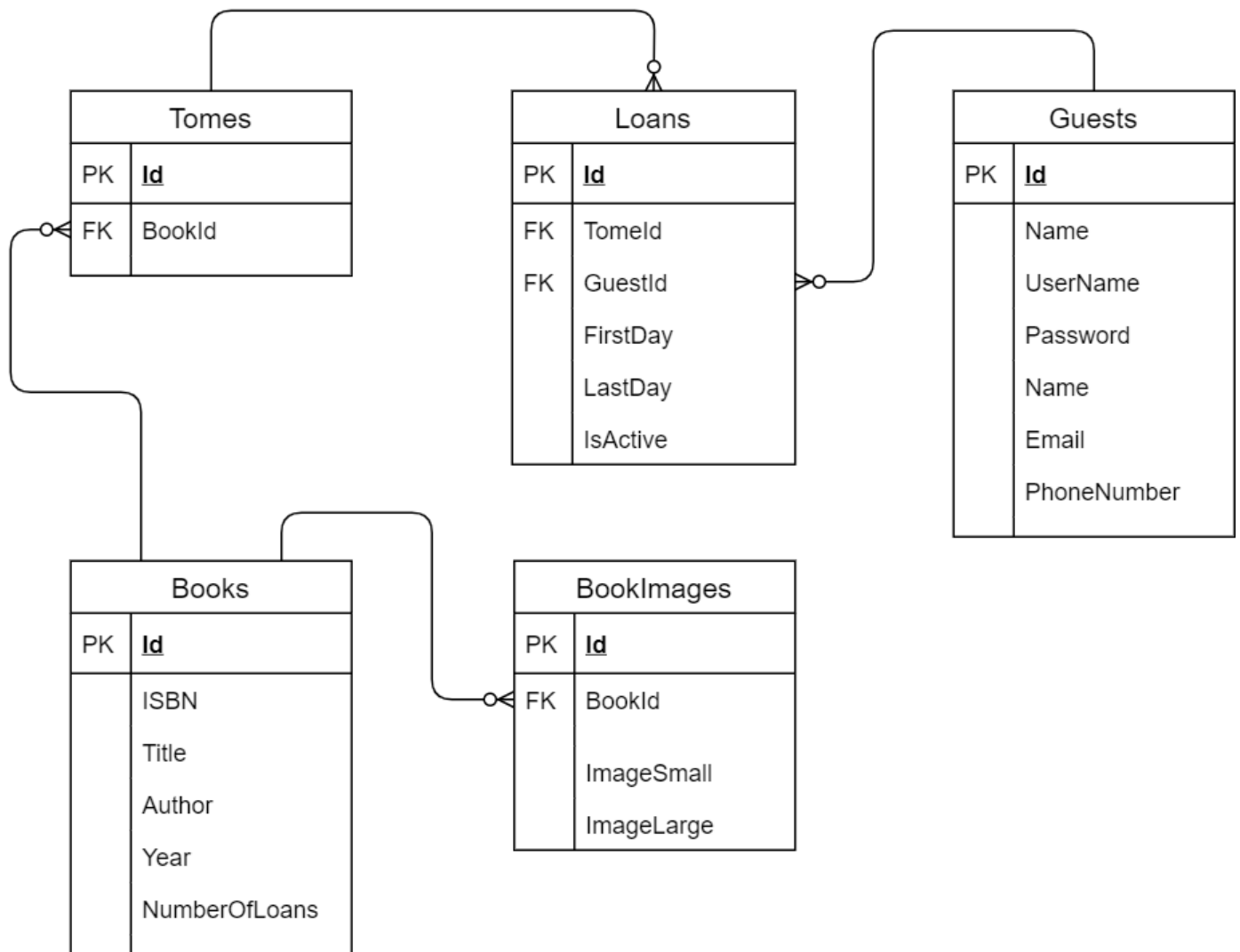
Adatbázis

Az adatbázisban öt manuálisan létrehozott tábla áll rendelkezésre az adatok tárolásához, melyek a következők:

- **Books:** az adatbázisban lévő könyvek tárolása,
- **BookImages:** a könyvekhez tartozó képek tárolása,
- **Tomes:** az adatbázisban lévő kötetek tárolása,
- **Loans:** az adatbázisba mentett kölcsönzések tárolása,

- Guests: a regisztrált felhasználók adatainak tárolása.



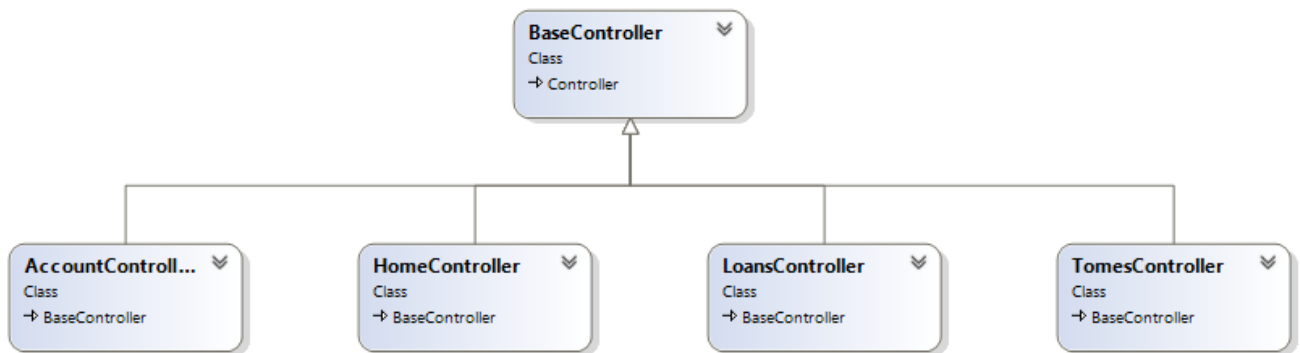


- Egy könyvhöz több kép is tartozhat.
- Egy vendég több kölcsönzést is rögzíthet.
- Egy könyvhöz több kötet is tartozhat.
- Egy kötetre több kölcsönzést is lehet adni.
- A Books tábla nyilvántartja a könyvre leadott kölcsönzések számát is. Ez megkönnyíti a népszerűség szerinti rendezést.

Webes felhasználói felület

A megjelenítési rétegben öt vezérlőt definiálunk, melyekhez nézetek tartoznak.

- **BaseController**: a többi vezérlő ősszánya. Bármelyik vezérlőnek bármelyik metódusának meghívása után összegyűjt aktuális adatokat, melyeket később nézetek felhasználnak.
- **HomeController**: a hozzá tartozó nézetek megjelenítik a főoldalt, illetve megjelenítik egy-egy kiválasztott könyv részleteit.
- **AccountController**: megjeleníti és vezérli a regisztráció ill. bejelentkezés paneleket. Ellenőrzi, hogy a megadott adatok típushelyesek-e, és regisztrálja/bejelentkezteti a felhasználót.
- **LoansController**: megjeleníti a kölcsönzési panelt, majd elmenti az adatokat, végül egy kölcsönzés sikerességét megállapító nézetet ad vissza.
- **LoanDateController**: lekérdezi egy adott hónap olyan napjait, amire az adott kötet még szabad.



Közös megjelenítés

A minden oldalon közösen megjelenő adatokat egy közös, `_Layout.cshtml` nézetben definiáljuk.

- A fejlécben a könyvtár neve található.
- Alatta feltételes megjelenítés: ha be van jelentkezve a látogató, akkor egy link látható a kijelentkezéshez, ha nincs bejelentkezve a felhasználó, akkor két linket – 'Bejelentkezés' és 'Kijelentkezés' – találhatunk.

Kezdőoldal

A `HomeController` Index nézete a következőképpen épül fel:

- A könyvek között lehetőség van szabadszavas keresőmező segítségével cím vagy szerző szerint keresni. A két keresési típus nincs elkülönítve, a megadott keresőszó egyaránt keres a címek és szerzők között is.

- A könyvek alapértelmezetten népszerűség szerinti csökkenő sorrendben vannak megjelenítve, ez a rendezés invertálható, illetve választható cím szerinti lexikografikus rendezés is.
- A könyvek listája táblázatszerűen van megjelenítve, minden könyvhöz megjelenítésre kerül a borítóképe, címe, szerzője, és a kiadás éve.
- Egyszerre 20 könyv kerül megjelenítésre, a többi előre- ill. visszalapozással érhetjük el.

Egy könyv részleteinek megjelenítése

- A főoldalon egy könyvet kiválasztva megtekinthetjük annak részleteit.
- Egy könyv részleteihez tartozik a főoldalon megjelenítésre került tulajdonság mellett az ISBN száma is, amely egy egyedi, 9-11 hosszúságú számsorozat.
- Itt láthatók továbbá a könyvhöz tartozó kötetek is, melyekhez rendre megadjuk a rá regisztrált kölcsönzéseket, és ezeknek tulajdonságait.
- Egy kölcsönzés tulajdonságai alatt csak a publikus információk értendők. Azaz nem kerül megjelenítésre a kölcsönző neve, sem más személyes adata, csak a kötetre vonatkozó adatok, mint a kölcsönzés kezdeti időpontja, utolsó napja, és a kölcsönzés státusza.
- Minden kötet mellett található egy link, amin keresztül kölcsönzést lehet leadni az adott kötetre. Nem lehet másik kölcsönzéssel átfedő kölcsönzést regisztrálni.
- Az oldal alján egy 'Vissza a főoldalra' feliratú link található, mely a könyvek listájához navigál.

Regisztráció

- A `Register.cshtml` nézet megjeleníti a regisztrációs panelt, amelyen input mezőkön keresztül adhatja meg a felhasználó az adatait.
- Regisztráció után a `Login.cshtml` nézetre irányít át.

Bejelentkezés

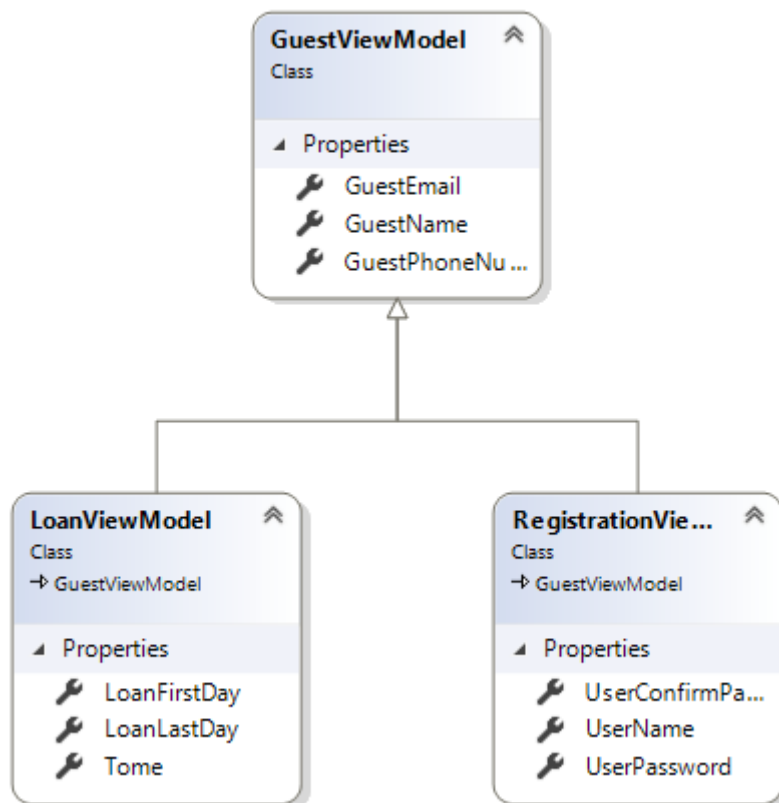
- A `Login.cshtml` nézet megjeleníti a bejelentkező panelt, amelyen felhasználónév és jelszó megadása után a felhasználó bejelentkezhet.
- Az főoldalra irányít át, amelyen a könyvek listája található.
- A fejlécben megjelenik a bejelentkezett felhasználó felhasználóneve.

Kölcsönzés leadása

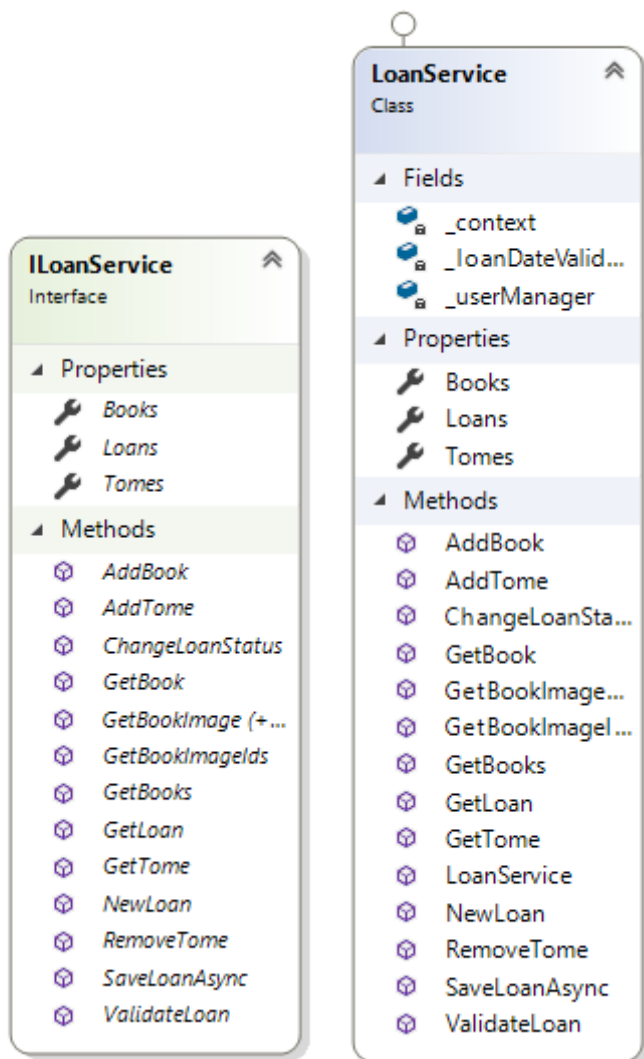
- A `LoansController/Index.cshtml` nézet megjeleníti a kölcsönzés adatait rögzítő formot. Ezen keresztül lehet megadni a kölcsönzés első ill. utolsó napját.
- Ez a nézet csak akkor érhető el, ha a felhasználó be van jelentkezve.
- Ez a nézet tartalmazza, hogy a felhasználó melyik kötetre kívánja leadni a kölcsönzést, tehát ezt a felhasználónak itt nem kell megadnia.
- Ha a megadott időpontban már van előjegyzés a kötetre, akkor a kölcsönzés nem lehetséges.
- Sikeres kölcsönzés esetén a `Result.cshtml` nézetre irányít át, ahol megbizonyosodhatunk a kölcsönzés sikerességéről.

ViewModelek

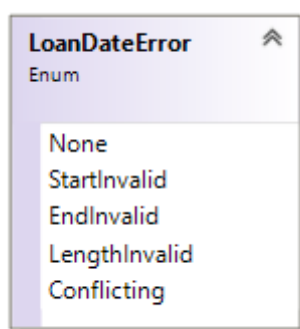
- A `GuestViewModel`, `RegisterViewModel`, ill. `LoginViewModel` a regisztrációhoz ill. bejelentkezéshez szükséges adatokat validálják.



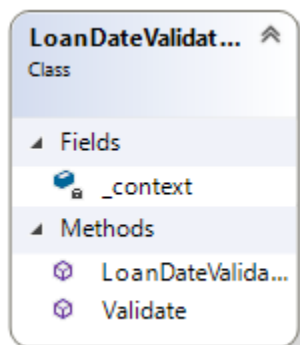
- **LoanService**: az **ILoanService** interfészt implementálja. A nézetek vezérlőivel nem érjük el közvetlenül az adatbázist, ezért ezen az osztályon keresztül juthatunk hozzá a tárolt adatokhoz.



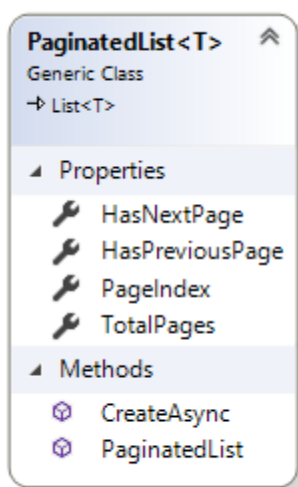
- **LoanDateError**: enumeráció, amely tartalmazza az előjegyzés során felmerülhető hibákat.



- `LoanDateValidator`: ellenőrzi a leadandó kölcsönzés dátumait, hogy ütközik-e más kölcsönzéssel, és ha valami hibát talál, visszatér egy `LoanDateError` enumerációban definiált hibával.

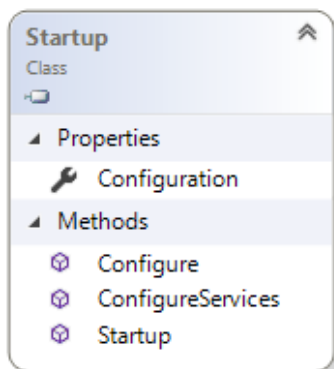


- `PaginatedList<T>`: a főoldalon a lapozást valósítja meg. Sablonparamétere jelen esetben a `Book`.



Webszolgáltatás

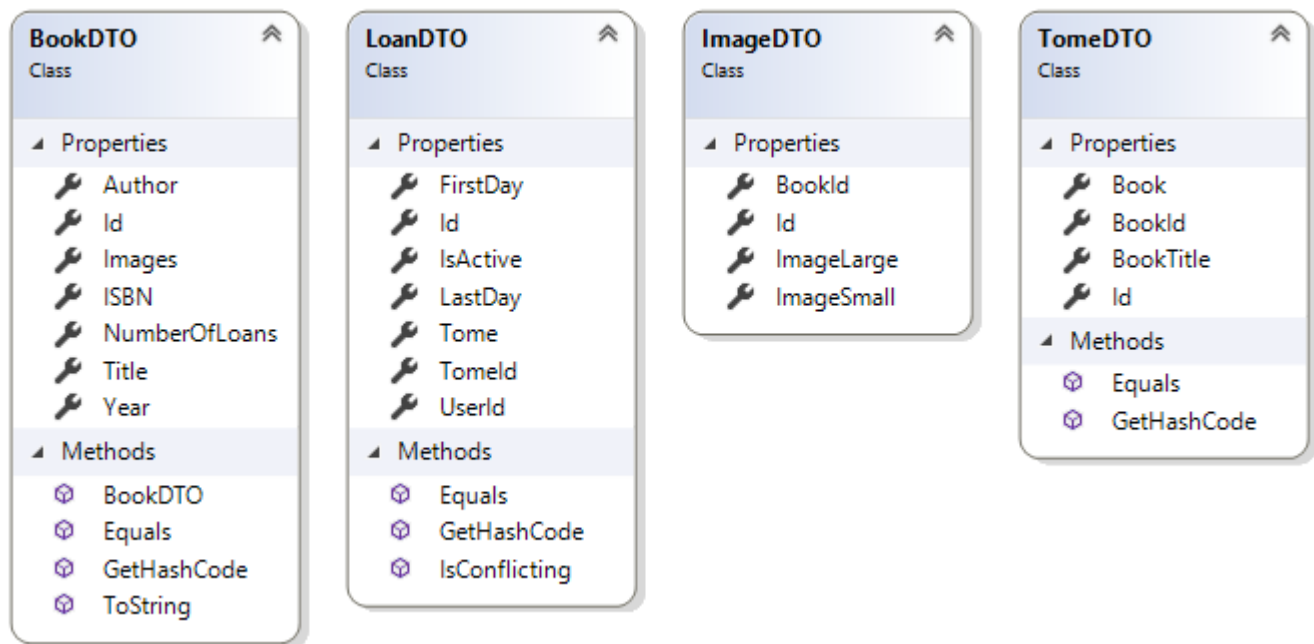
- Az alkalmazást a `Startup` osztályban konfiguráljuk. Itt kerül meghívásra az adatbázis inicializálását végző függvény.



- **BooksController:**
 - `GetBooks()`: lekéri az adatbázisban tárolt könyveket
 - `GetBook(int id)`: lekéri az adatbázisból az id azonosítóhoz tartozó könyvet
 - `PostBook([FromBody] BookDTO book)`: elmenti az adatbázisba a kapott könyvet
- **BookImagesController:**
 - `GetImage(int bookId)`: lekéri a megadott azonosítójú könyvhöz tartozó képeket
 - `PostImage([FromBody] ImageDTO image)`: elmenti a megadott képet
 - `DeleteImage(int id)`: törli a megadott képet
- **AccountController:** `Login()` és `Logout()` metódusai elvégzik a ki- és bejelentkezést
- **LoansController:**
 - `GetLoans()`: törli az adatbázisból a már lejárt és inaktív kölcsönzéseket, majd visszaadja a megmaradt kölcsönzéseket
 - `GetLoansForBook(int id)`: törli az adatbázisból a már lejárt és inaktív kölcsönzéseket, majd visszaadja a megmaradt kölcsönzéseket, amik a megadott könyvre lettek foglalva
 - `GetTomesForBook(int id)`: törli az adatbázisból a már lejárt és inaktív kölcsönzéseket, majd visszaadja a megmaradt kölcsönzéseket, amik a megadott kötetre vonatkoznak
 - `PutLoan([FromBody] LoanDTO loanDTO)`: módosítja a paraméterben kapott kölcsönzést. Technikailag csak a kölcsönzés státusza változhatott meg.
- **TomesController:**
 - `GetTomes()`: visszaadja a létező köteteket
 - `GetTome(int id)`: visszaadja a megadott azonosítójú kötetet
 - `GetTomesForBook(int id)`: visszaadja a megadott azonosítójú könyvhöz tartozó köteteket
 - `PostTome([FromBody] TomeDTO tomeDTO)`: létrehozza és elmenti a megadott kötetet
 - `DeleteTome(int id)`: törli a megadott kötetet
- **App_Data:** ebből a könyvtárból töltődnek be az alkalmazáshoz tartozó képek.

Library.Data

Osztálykönyvtár. Tartalmazza a megjelenítendő típusokat és az adatkonverzióhoz szükséges metódusokat.



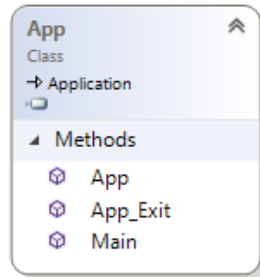
Library.Model

Tartalmazza az adatbázisban tárolt adatok típusát és a migrációkat. A `DbInitializer` osztály feltölti adatokkal az adatbázist, ha abban még nincs adat.

Adminisztrációs kliens

App osztály

Az Application osztály leszármazottja. Adattagként tartalmazza az ablakokat, a hozzájuk tartozó nézetmodelleket, és egy modell interfészt.



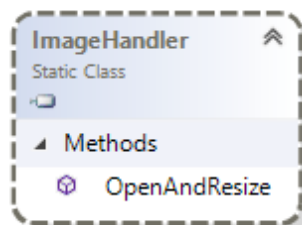
Nézetek

- LoginWindow: bejelentkező ablak
- MainWindow: láthatjuk rajta a könyvek, kötetek és kölcsönzések listáját. Egy könyvet kiválasztva megjelennek a kötetei, egy kötetet kiválasztva megjelennek a rá leadott kölcsönzések. Lehetőség van könyv hozzáadására, kötet hozzáadására vagy törlésére, és kölcsönzés módosítására
- BookEditorWindow: könyv hozzáadására alkalmas
- TomeEditorWindow: kötet hozzáadására alkalmas
- LoanEditorWindow: egy kölcsönzés státuszának megváltoztatására alkalmas

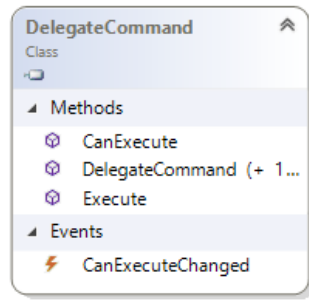
Nézetmodellek

- ViewModelBase: Az INotifyPropertyChanged interfészt implementálja
- MainViewModel: kezeli a főablakból érkező parancsokat és továbbítja a modellnek
- LoginViewModel: kezeli a bejelentkezést

ImageHandler: formázza és visszaadja a képet



DelegateCommand: nézetről érkező parancsok osztálya

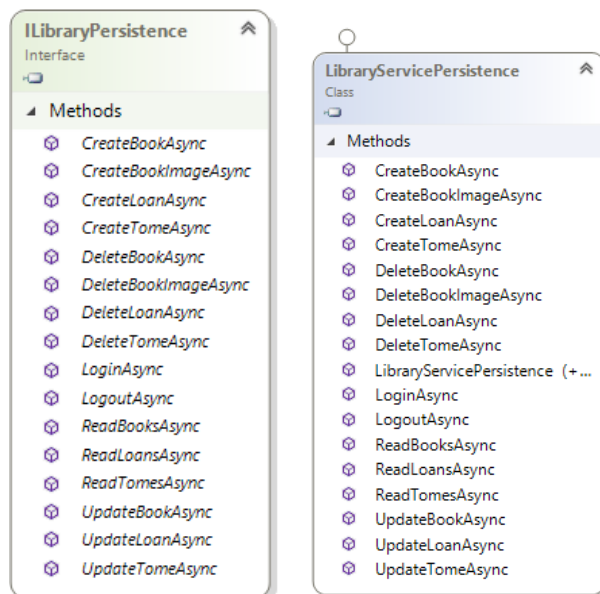


Konverterek

- **BookImageConverter:** könyvek képének átalakítója
- **BookCollectionConverter:** a kapott épületlistából előállítja a címek listáját
- **BookConverter:** megadott id-ra visszaadja magát a könyvet
- **BoolToActivationStateConverter:** boolean-ból stringet állít elő. Igaz esetén „aktív”, hamis esetén „inaktív” stringet ad vissza.

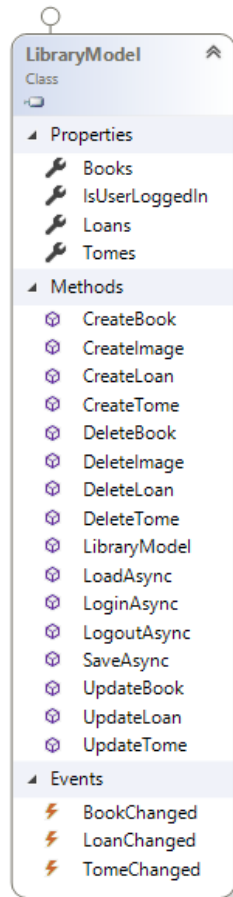
Perzisztencia

- **LibraryServicePersistence:** A webszolgáltatással kommunikálva visszaadja egy-egy művelet sikerességét.



Model

- LibraryModel: az adminisztrációs kliens modellje



Tesztelés

A tesztelést mock teszttel végzem. A BooksController, TmesController és LoansController vezérlőinek metódusait tesztelem vele a webszolgáltatás komponensében.

Tesztesetek

BooksController

- **GetBooksTest**
 - metódus: GetBooks
 - összehasonlítások: valóban a visszaadja az összes könyvből álló listát
- **GetBookTest**
 - metódus: GetBook
 - összehasonlítások: az 1-es azonosítójú könyvet adja vissza
- **GetBookTest2**
 - metódus: GetBook
 - összehasonlítások: az 2-es azonosítójú könyvet adja vissza
- **CreateBookTest**
 - metódus: CreateBook
 - összehasonlítások: az új könyv az elvárásnak megfelelő

TmesController

- **GetTomesTest**
 - metódus: GetTomes
 - összehasonlítások: valóban a visszaadja az összes kötetből álló listát
- **GetTomeTest**
 - metódus: GetTome
 - összehasonlítások: az 1-es azonosítójú kötetet adja-e vissza
- **GetTomeTest2**
 - metódus: GetTome
 - összehasonlítások: az 2-es azonosítójú kötetet adja-e vissza
- **GetTomeForBookTest**
 - metódus: GetTomeForBook
 - összehasonlítások: a 2-es azonosítójú könyvekhez tartozó köteteket adja-e vissza
- **GetTomeForBookTest2**
 - metódus: GetTomeForBook
 - összehasonlítások: az 1-es azonosítójú könyvekhez tartozó köteteket adja-e vissza
- **CreateTomeTest**
 - metódus: CreateTome
 - összehasonlítások: a szimulált adatbázis mérete nő-e eggyel, a hozzáadott kötet valóban az új kötet-e
- **DeleteTomeTest**
 - metódus: DeleteTome
 - összehasonlítások: A válasz megfelelő típusú-e, a szimulált adatbázis mérete eggyel kisebb-e, a törölt kötet id-ja megfelelő-e

LoansController

- **GetLoansTest**
 - metódus: GetLoans
 - összehasonlítások: valóban a visszaadja az összes kölcsönzésből álló listát
- **GetLoansForBookTest**
 - metódus: GetLoansForBook

- összehasonlítások: a szimulált adatbázis mérete megfelelő-e, a visszaadott kölcsönzések listája megfelelő-e
- GetLoansForBookTest
 - metódus: GetLoansForBook
 - összehasonlítások: a visszaadott kölcsönzések listája megfelelő-e
- GetLoansForTomeTest
 - metódus: GetLoansForTome
 - összehasonlítások: a visszaadott kölcsönzések listája a 2-es azonosítójú kötethez tartozó kölcsönzések listája-e
- GetLoansForTomeTest2
 - metódus: GetLoansForTome
 - összehasonlítások: a visszaadott kölcsönzések listája a 3-as azonosítójú kötethez tartozó kölcsönzések listája-e
- EditLoanTest
 - metódus: EditLoan
 - összehasonlítások: a visszatérési típus megfelelő-e, a szimulált adatbázis mérete nem változik