1. Beadandó feladat dokumentációja

Készítette:

Név	Peskó Márton
Azonosító	YRQHGX
E-mail cím	peskomarton@hotmail.com

Feladat leírása:

(7.) 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.

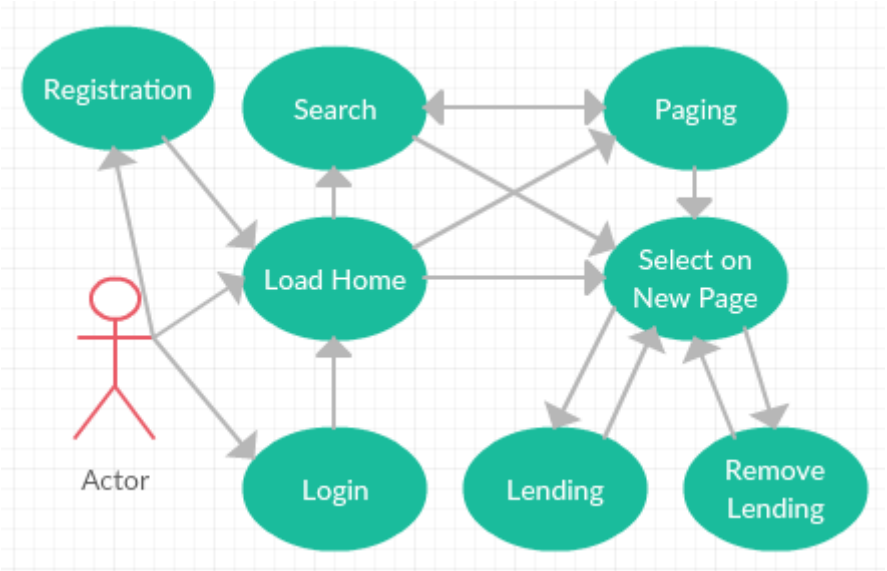
- A főoldalon a könyvtárban elérhető könyvek kerülnek ki listázásra, alapértelmezetten népszerűségi sorrendben (kölcsönzések száma), de a felhasználó választhat cím szerinti lexikografikus rendezést is. Egy oldalon legfeljebb 20 könyv jelenik meg, a többit lapozással lehet elérni. A könyvek címmel, (első) 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 kötetre, a kölcsönzés tervezett kezdő és befejező napját megadva. Az előjegyzés nem fedhet át aktív kölcsönzéssel vagy más előjegyzéssel.

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).

Elemzés:

A megoldást [ASP.NET Core MVC] keretrendszerrel meg lehet valósítani.



Vezérlők:

Kelleni fog egy vezérlő a szűrt könyvek megjelenítéséhez, a felhasználók autentikációjához és autorizációjához valamint egy a foglalások lebonyolításához. Valamint ezeknek alkotunk egy szülőosztályt, melyben tároljuk a teljes alkalmazásra kiterjedő egységes adatokat.

Ezek a vezérlők, fogják visszaadni a nézeteket. Ezek a sablon nézetek fognak valós időben statikus HTML-é lefordulni nézetk modellek segítségével és ezeket fogják a webböngészők megjeleníteni.

Nézetek:

Használni fogunk egy megosztott nézetet, mely arra szolgál, hogy az általunk mindig látni kívánt elemeket jelenítsük meg. Például a fejléc és a felhasználói autentikációs és autorizációs űrlap.

Szükségünk lesz egy kezdő nézetre, amely majd megjeleníti a könyveket a felhasználó által beállított és elküldött adatok alapján. Kell majd egy a könyvet kiválasztva annak részleteit megjelenítő nézet.

Kell a könyvfoglalást lebonyolító nézet.

Kell egy a bejelentkezéshez és a regisztrációhoz.

Valamint egy minimális könyvfoglalás törlési eredményt megjelenítő és navigációs hibát kiíró nézeteket.

Ezekhez a nézetekhez szükség szerint van egy nézet modell kialakítva.

Nézet modellek:

Ezek a nézet modellek egy-egy űrlaphoz kapcsolódnak segítenek azok adatainak ellenőrzésében és továbbításában.

Ezek alapján kelleni fog egy bejelentkezési, regisztrációs, keresési, könyv kölcsönzői nézet modell, valamint egy hiba nézet modell.

Szolgáltatások:

A perzisztencia megalkotásához az [Entity Framework Core]-t használtam.

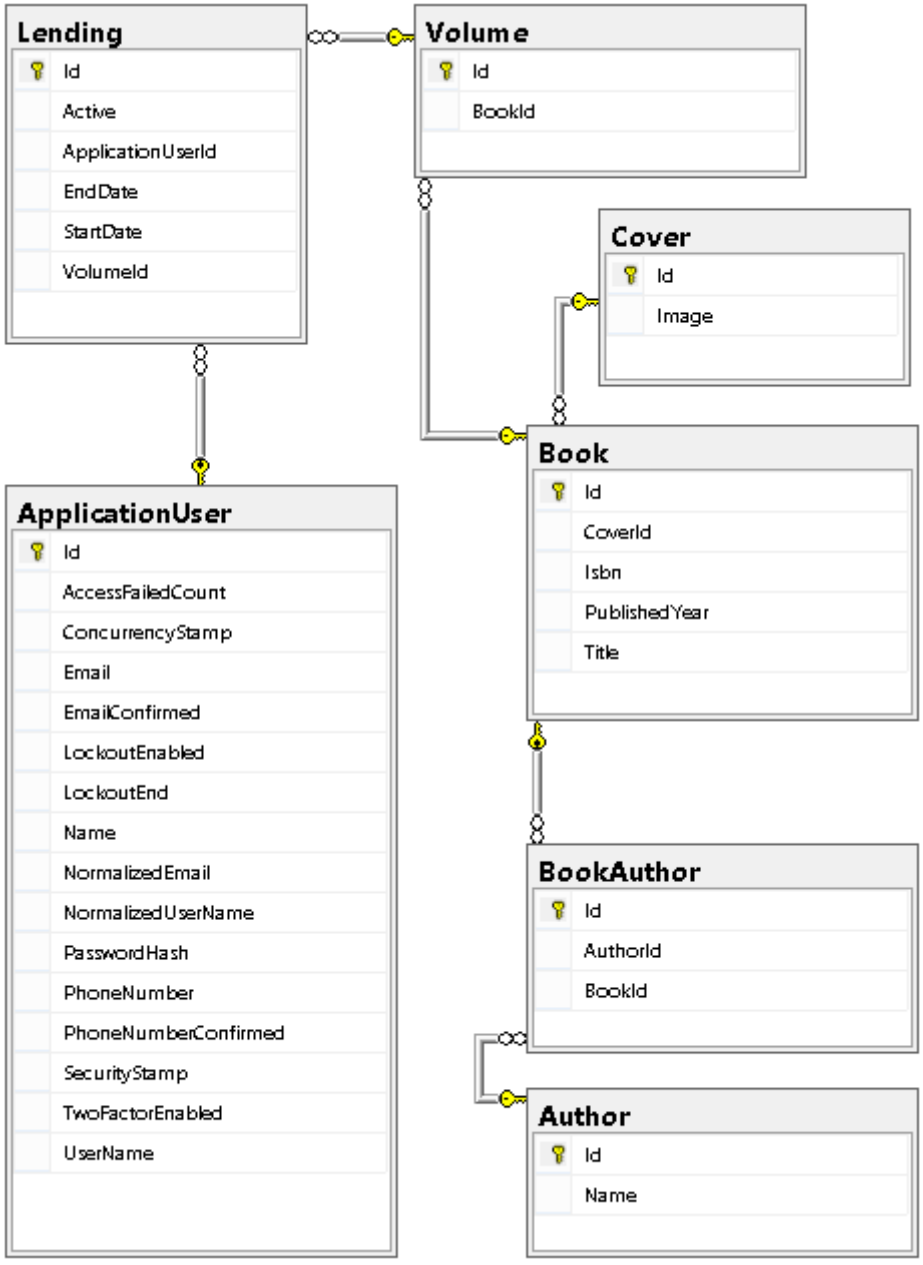
Ezzel add kapcsolatot a könyvtár nevű szolgáltatás, a következő lehetőségekkel:

- Könyvek objektum lista

- Könyvek objektum lista szűkítése szerző és cím szerint
- Könyvek objektum lista szűkítése lapozás szerint
- Könyv könyv index szerint
- Könyv index kölcsönzés index szerint
- Könyv index kötet index szerint
- Kötet kötet index szerint
- Kölcsönzés mentése
- Kölcsönzés törlése

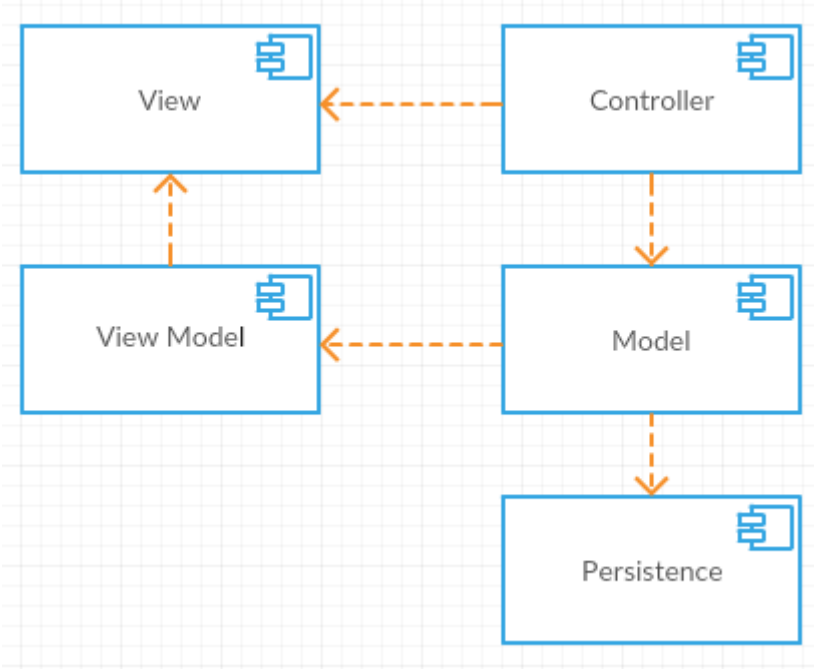
Perzisztencia:

Egy adatbázist leíró modell szerkezet, a feladat alapján, kiegészítve segéd funkciókkal, mint például: egy könyv objektumhoz tartozó első írót visszaadó függvény.

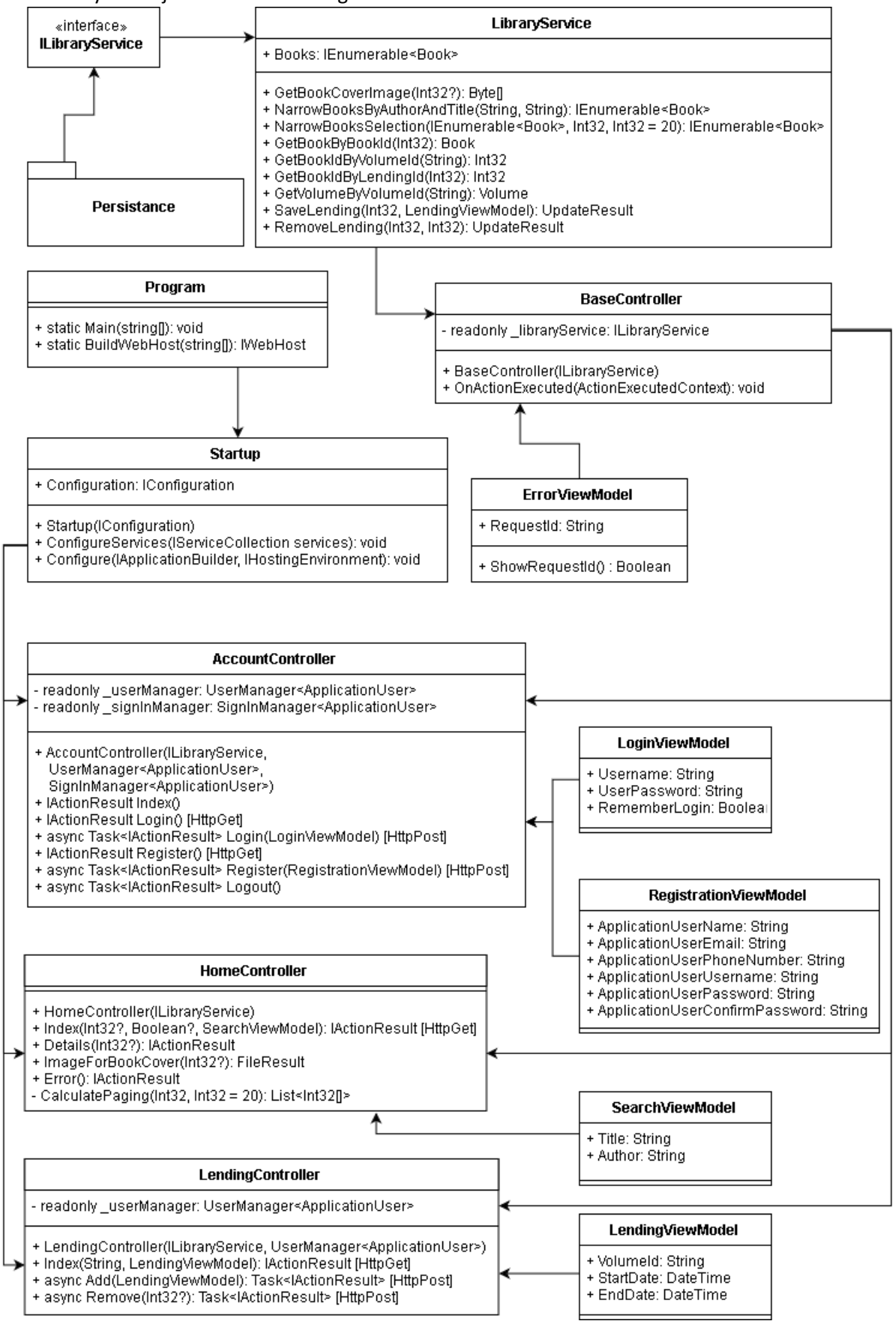


Az ApplicationUser táblához még kapcsolódnak más táblák is melyeket az Identity keretrendszer generált a felhasználó kezeléshez. Ezekkel a ebben a dokumentációban nem foglalkozunk.

Az adatbázist első indításnál vagy adatbázis hiányában létrehozzuk és feltöltjük kezdeti adatokkal, de nem felhasználókkal.



A C# osztályok tartják az összetevő diagramban látható szerkezetet.



Ebben a diagramban a Perzisztencia osztályok kimaradtak, mivel adottakká válnak az adatbázis szerkezetből.