Hausknecht Bálint
B2ICWH

Webes alkalmazások fejlesztése

1. Feladat:

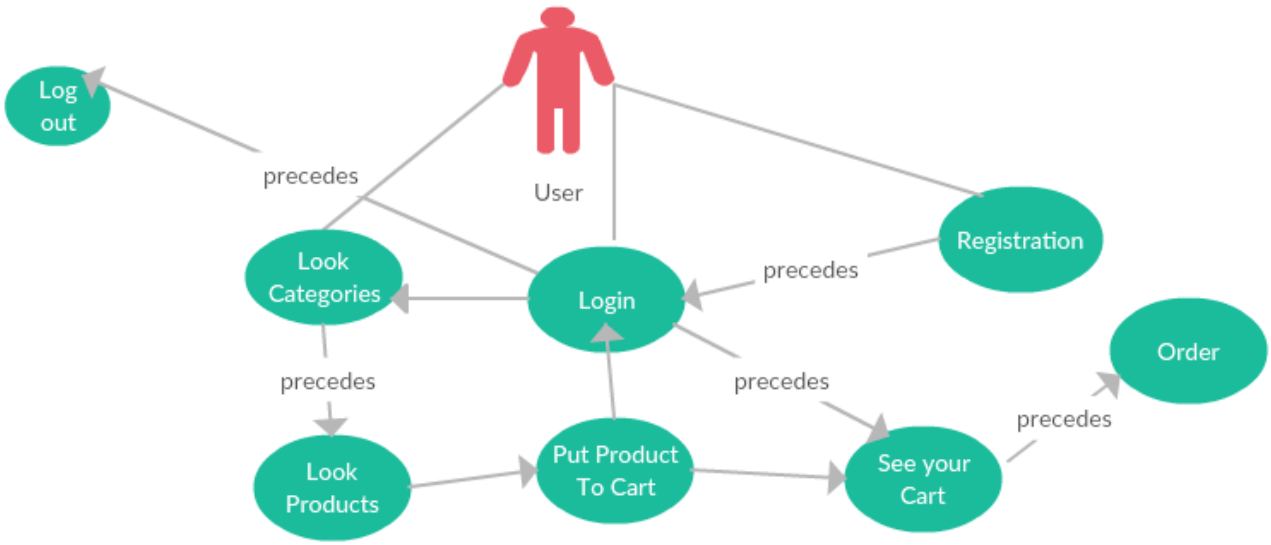
Elektronikai webáruház

Készítsük el egy elektronikus termékekkel foglalkozó cég online rendszert, amelyben a vevők megrendelhetik termékeiket.

1) **részfeladat:** a webes felületen a vásárlók adhatnak le online rendeléseket a cégnek.
- A főoldalon megjelennek a kategóriák (pl. szórakoztató elektronika, számítástechnika, konyhai gépek), és minden kategóriára egy véletlenszerűen választott termék a kategóriából.
- A kategóriát kiválasztva listázódik az összes kategóriabeli termék. Egy oldalon legfeljebb 20 termék jelenik meg, a többit lapozással lehet elérni. A termékek gyártóval, típusszámmal, rövid leírással, valamint nettó/bruttó árral rendelkeznek. A listát ár, illetve gyártó szerint rendezhetjük (növekvő/csökkenő) sorrendbe.
- A vásárló tetszőleges számban helyezheti a kosárba a termékeket (egy termékből több darabot is rendelhet), majd megadhatja adatait (név, cím, telefonszám, e-mail cím), és véglegesítheti a rendelést. A véglegesítés előtt a kosárból lehet törölni is, változtatható a darabszám (amennyiben ez 0-ra csökken, akkor a termék törlődik), illetve az egész kosár kiüríthető egy lépésben. A kosarat bármikor meg lehet tekinteni, illetve látható a nettó/bruttó végösszeg is.
- A webes felületen nem jelennek meg a tartósan hozzáférhetetlen (inaktív) termékek, valamint azok sem, amelyek raktárkészlete üres.

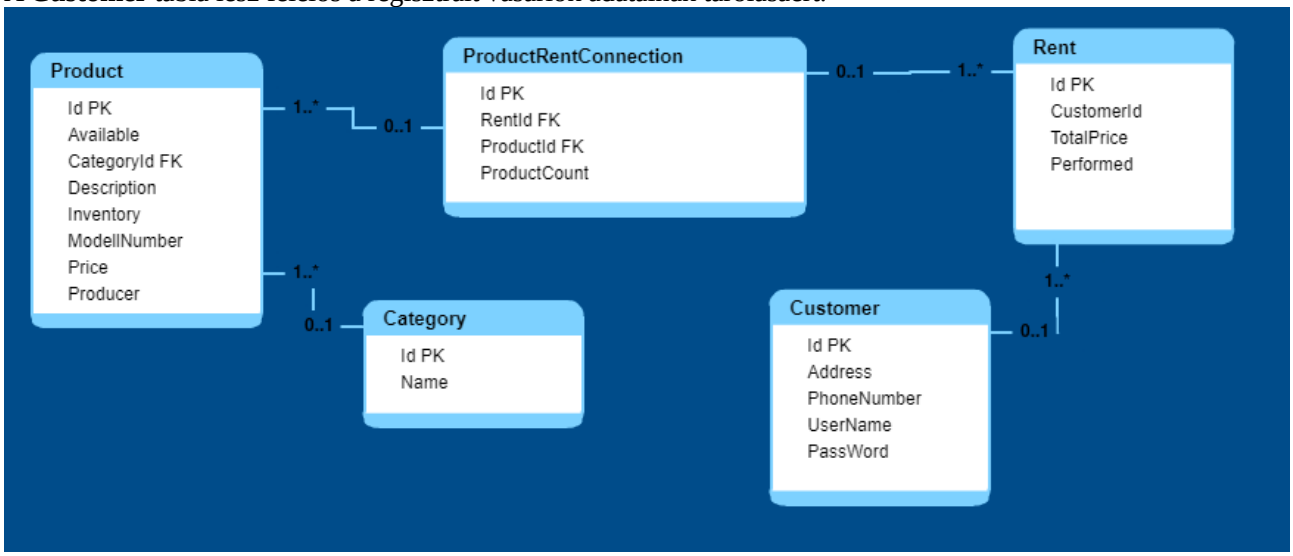
2) **részfeladat**: az alkalmazottak az asztali grafikus felületen keresztül adminisztrálhatják a rendeléseket, illetve raktárkészleteket.
- Az alkalmazott bejelentkezhet (felhasználónév és jelszó megadásával) a programba, illetve kijelentkezhet.
- Bejelentkezve az alkalmazás listázza a termékeket (gyártó, típus, leírás, nettó/bruttó ár, illetve raktárkészlet), és lehetőség van a raktárkészlet növelésére az egyes termékeknél.
- Amennyiben a termék hosszabb távon nem hozzáférhető, lehet inaktív állapotba helyezni. Ez azt jelenti, hogy a webes felületen csak addig jelenik meg, és addig rendelhető, amíg van belőle raktárkészlet. Továbbá az asztali grafikus felületen sem lehet a raktárkészletét növelni. Inaktív terméket természetesen lehet újra aktiválni.
- Az alkalmazás listázza a rendeléseket (dátum, név, cím, telefonszám, e-mail cím, termékek listája). A teljesítést a munkatárs kijelöléssel tudja kezdeményezni, amelyre a rendszer megerősítést kér, és automatikusan módosítja a raktárkészletet. A rendelések szűrhetőek a megrendelő név(részle) szerint.

2. Felhasználói eseteket ábrázoló diagramm:



3. Adatbázis Modell:

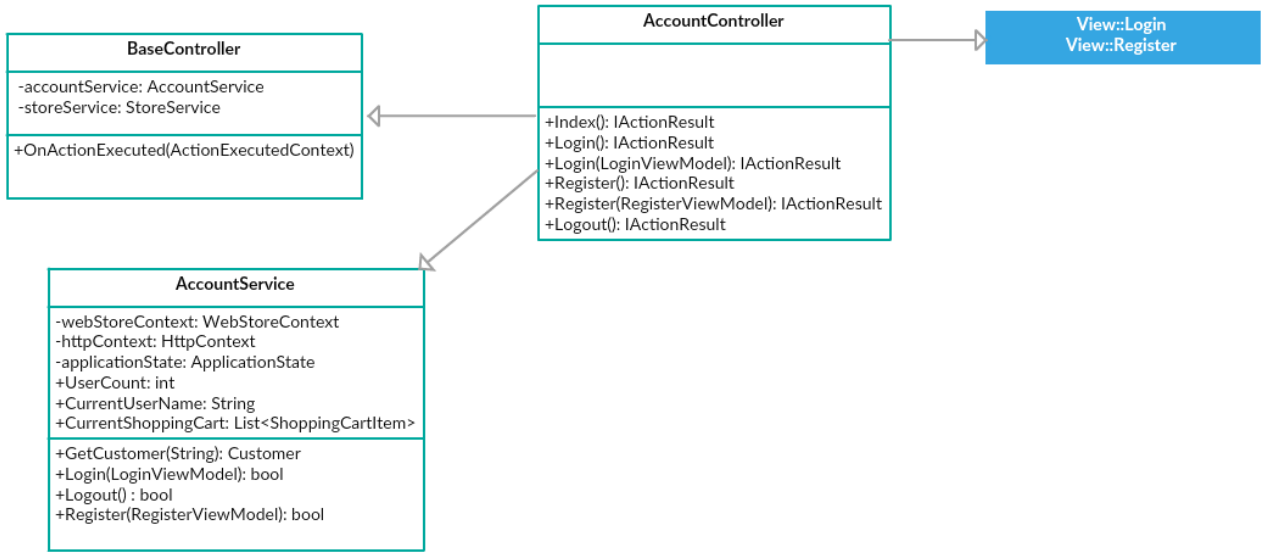
Hausknecht Bálint
B2ICWH
Az adatbázisban 5 táblát tárolok: Categories, Products, Rents, RentProductConnection, Customers.
A **Categories** táblában tárolom el a webshopban található kategóriák nevét, illetve azonosítóját.
A **Products** táblában tárolom el a webshopban vásárolható termékek azonosítóját, nevét, modellszámát, árát, gyártóját, leltár számát, elérhetőségét, kategória azonosítóját (melyik azonosítójú kategóriához tartozik).
A **Rents** táblában tárolom el a rendelések azonosítóját, vásárló azonosítóját, az összegzett árat, illetve, hogy feldolgozták e már.
A **RentProductConnection** tábla egy kapcsoló tábla, amit azért hoztam létre, mert egy adott rendeléshez több termék is tartozhat és mivel listát nem tudok adatbázisban tárolni egyszerűen, így ez a tábla fogja tárolni a Rendelés azonosítót, termék modell számát és mennyiségét és egy saját azonosítót.
A **Customer** tábla lesz felelős a regisztrált vásárlók adatainak tárolásáért.



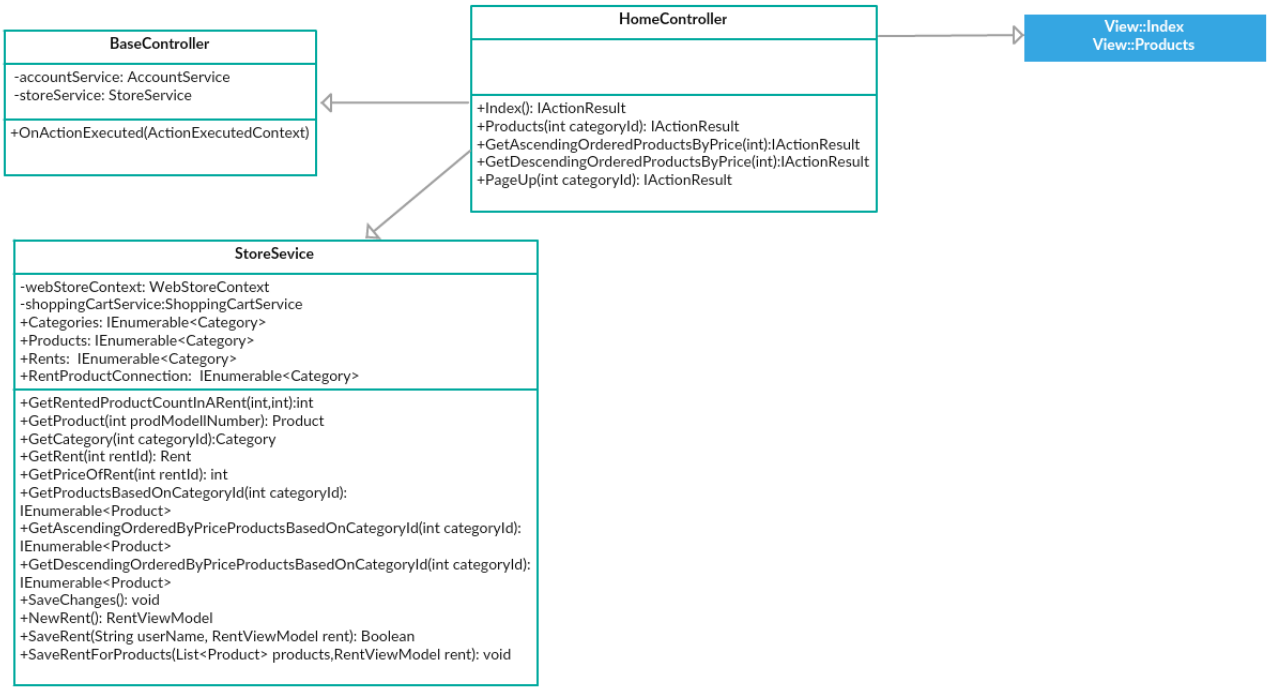
4. <u>Webshopot megvalósító webes alkalmazás:</u>

Alapvetően a felületet bárki eléri és megtekintheti az aktuális elérhető termékeket a különböző kategóriákból, de rendelni csak a regisztrált és bejelentkezett felhasználók tudnak.
Minden Controller osztályomnak szüksége van különböző servicekre, ezért csináltam egy BaseControllert amelyből majd az összes többi Controlleremet származtatni fogom.
A főoldalnak a mindig látható részeit a shared mappában található Layout.cshtml-be raktam bele. A különböző almappákban található cshtml fájlok pedig mindig ebbe a Layout.cshtml-be fognak betöltődni az aktuális eseménytől függően.
A bejelentkeztetéshez létrehoztam egy **AccountControllert**, ami képes fogadni a regisztrációs felületen kitöltött és elküldött formot, illetve a belépésre való igény is egyenesen ide érkezik be. A controllernek a felelőssége alapvetően az, hogy minimális validációt végezzen, illetve továbbítsa a kérést valamilyen Service felé.
Jelen esetben az **AccountService** lesz ez a service ami tud kommunikálni az adatbázissal és egy regisztrálni vágyodó felhasználó adatait belerakni és beléptetésnél pedig ellenőrzini (autorizálni).

Hausknecht Bálint
B2ICWH

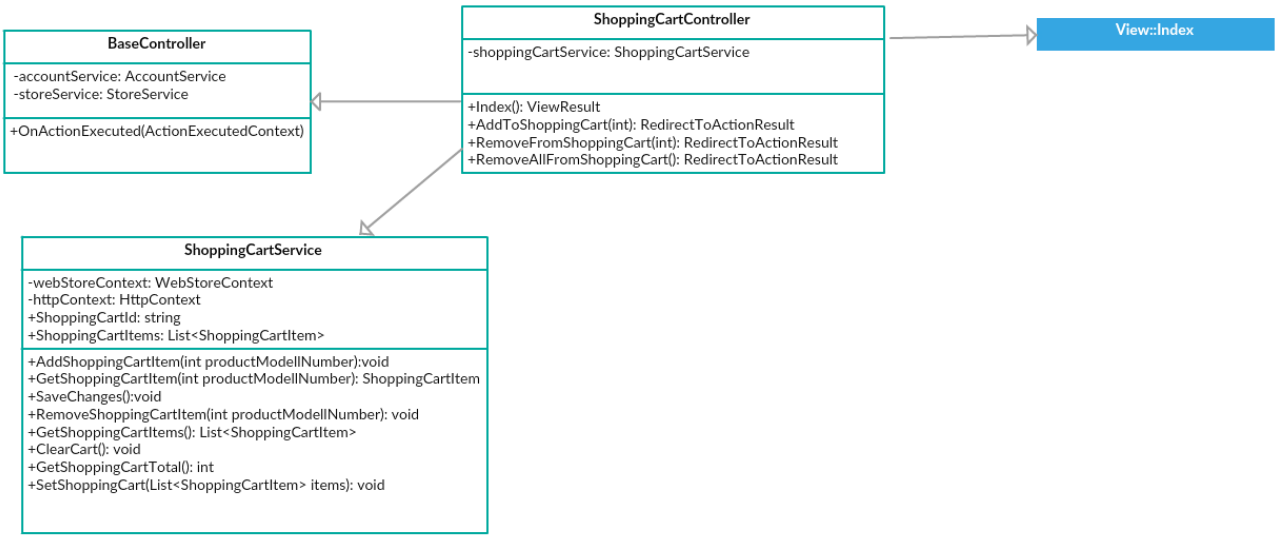


A főoldal megnézéshez szükséges kérések kezelésért a **HomeController** lesz a felelős, mely a beérkező kérést továbbítja validálás után a **StoreServicenek**. Ez a controller is a BaseControllerből származik.

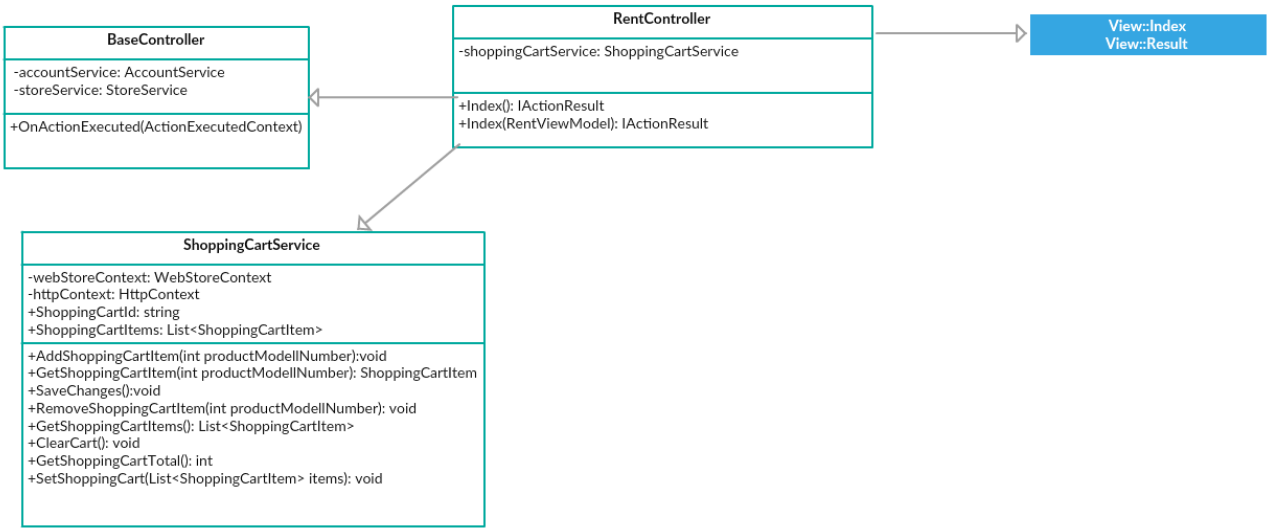


Ha egy regisztrált és bejelentkezett felhasználónak az egyik termék megtetszik a webshopban, akkor a kosárba ezt a terméket a kosarába ezt bele tudja rakni. Az egyes felhasználók kosarainak a kezeléséért a **ShoppingCartController** a felelős, mely a **ShoppingCartService** felé közvetíti a kéréseket. Az egyes kosarakat Sessionokben tároltam termék azonosító, mennyiség párok listájaként. Ahhoz, hogy Sessionben tudjak listát tárolni kiterjesztettem az alap Session Get és Set metódusait. Ennek implementációja a **SessionExtension** osztályban található, mely egy statikus osztály, így egyszerűen lehet használni bárhol.

Hausknecht Bálint
B2ICWH



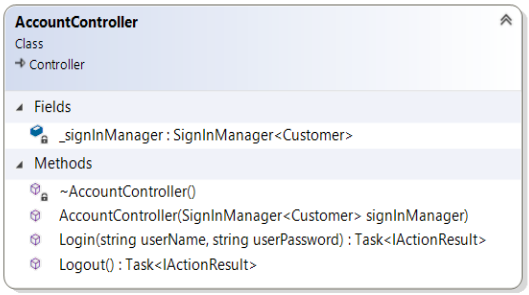
Ha a felhasználó a kosarában elemeket megrendelné, akkor az ezen kérések a **RentControllerhez** fognak becsapódni, mely lényegében a **ShoppingCartServicevel** fog kommunikálni.

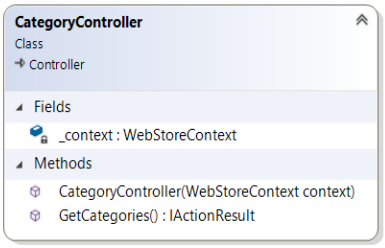


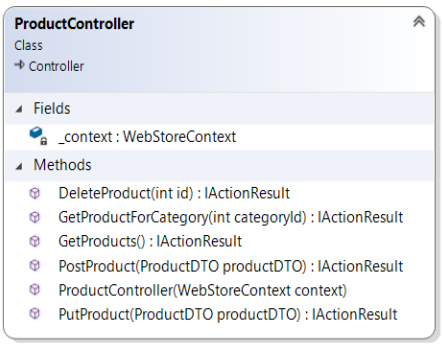
5. WebShopot megvalósító asztali alkalmazás:

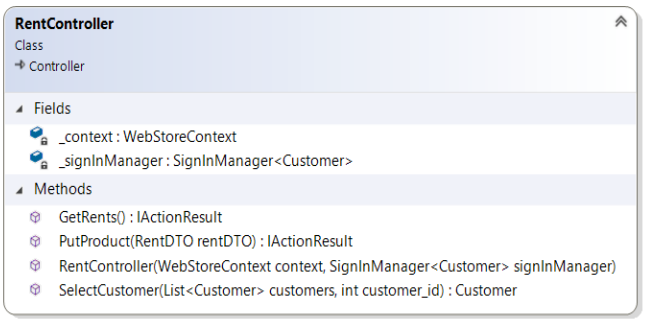
- A kliens egy WPF alkalmazás lesz (WebShop.Admin), amely egy WebAPI szolgáltatáshoz (WebShop.Service) fog csatlakozni.

Hausknecht Bálint
B2ICWH







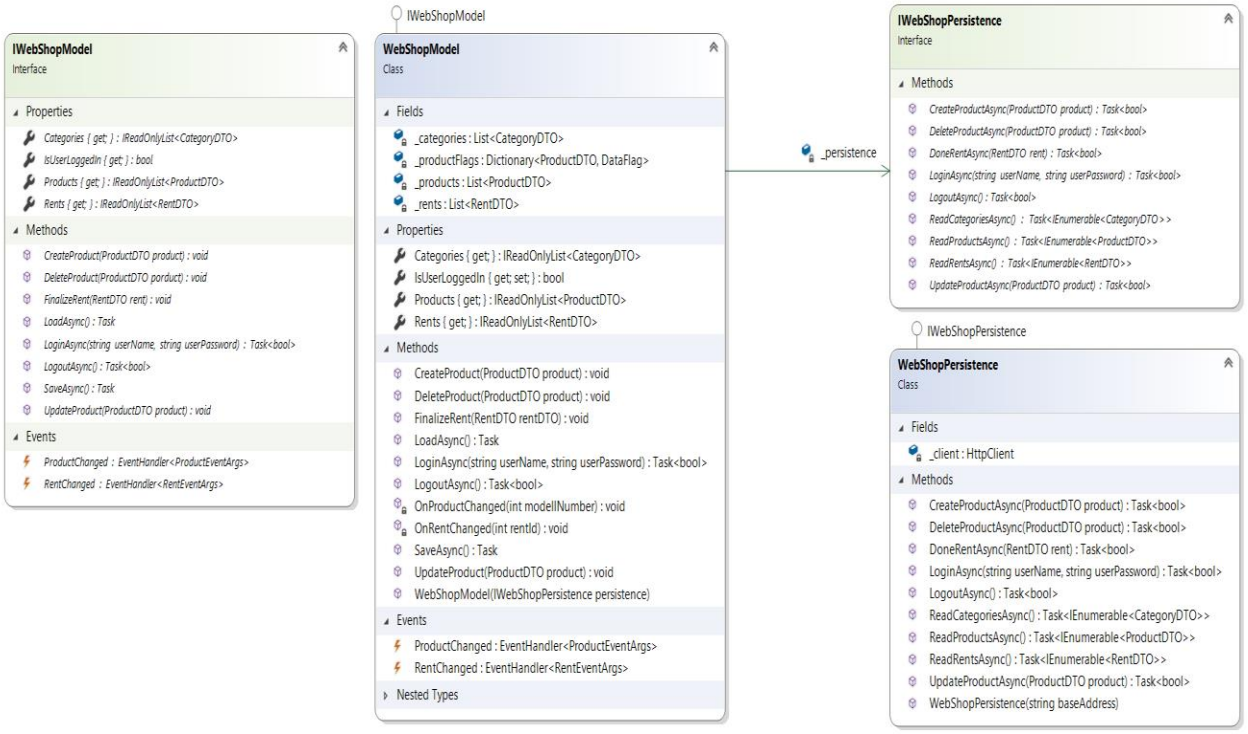


-
- A kliens alkalmazást MVVM architektúrában készítjük el, ahol a perzisztencia (WebShopPersistence) biztosítja a hálózati kommunikációt
- Az adatátvitelhez külön típust hozunk létre (ProductDTO,RentDTO,CategoryDTO), és egy külön osztálykönyvtárba helyezzük el (WebShop.Data), amely megosztásra kerül mindkét projekt számára.
- A szolgáltatásban egy vezérlő (ProductController) biztosítja a CRUD műveleteket az egyes termékeken.
- Egy vezérlő (RentController) felelős a rendelések véglegesítésére vonatkozó kérések kezeléséért. Egy HTTP PUT kérés törzsében megadom a véglegesíteni kívánt rendelést és a rendelés véglegesítésre kerül, illetve a rendelt termékek számát levonom a raktárkészletekből.

A kliens aszinkron adatkezelést biztosít, a modell (IWebShopModel) felügyeli a kliensbeli állapotot állapotjelzőkkel (DataFlag), ez alapján tudjuk mentéskor a megfelelő műveletet elvégezni. A perzisztencia (IWebShopPersistence) feladata az adatok betöltése, mentése és konvertálása aszinkron műveletekkel. Az új

termékeknek létrehozunk egy ideiglenes azonosítót (a megkülönböztetés végett), amely helyett a szerver visszaad egy végleges azonosítót.

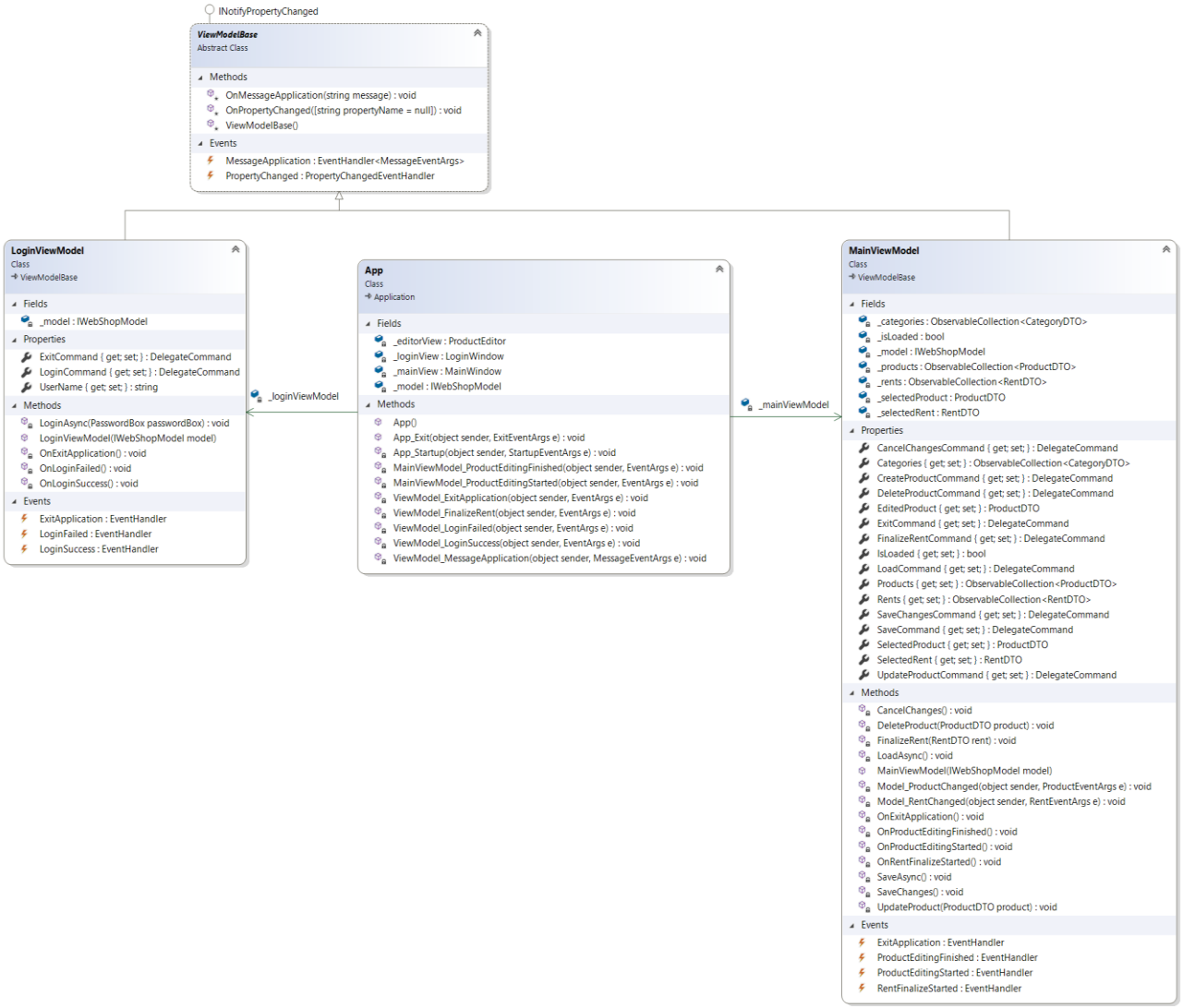
Hausknecht Bálint
B2ICWH



Létrehozunk egy vezérlőt a felhasználó-kezeléshez (AccountController), ebben lehetőséget adunk bejelentkezésre (Login) és kijelentkezésre (Logout). Kliens oldalon megjelenik a két új művelet a modellben (LoginAsync, LogoutAsync). A bejelentkezéshez egy külön nézetet (LoginWindow), valamint nézetmodellt (LoginViewModel) hozunk létre.

Ezen felül létrehoztam egy MainWindow-t és egy PorudctEditorWindowt. A MainWindow fogja mutatni nekünk két táblát, az egyik a termékek listáját a másik pedig a rendelések listáját. Illetve a termékekhez/rendelésekhez tartozó műveleteket egy-egy azoknak megfelelő gombra kötöttem rá. A ProductEditorWindow pedig a termék szerkesztéséhez, illetve új termék létrehozásához kell.

Hausknecht Bálint
B2ICWH



6. Tesztelés

A webszolgáltatás funkcionalitását egységtesztekkel ellenőriztem. Létrehoztam egy test szervert, melyhez a WebStoreContext-emet behúztam, mint service. Illetve a test szerverhez létrehoztam egy test klienst is, melynek segítségével tudok majd HTTP kéréseket küldeni adott végpontokra.

Tesztesetek:

- Termékek lekérdezése, rossz kategória id-val
- Termékek lekérdezése jó kategória id-val
- Termék postolása és adatbázis ellenőrzése