

## **WEBES ALKALMAZÁSOK FEJLESZTÉSE**

### Második beadandó

Készítette: Simon Gergő (BIXTG7)

## Tartalomjegyzék

Feladat.....	4
Adatbázis .....	6
Osztályok .....	7
MainCategory .....	7
SubCategory .....	7
Product .....	7
Product.Packaging .....	7
Product.Packaging.PackagingType .....	8
ProductOrder .....	8
Order .....	8
Employee .....	8
BeerkaContext .....	8
BeerkaService .....	9
DbInitializer .....	10
Komponens diagram .....	10
Első részfeladat.....	11
Felhasználói esetek .....	11
Osztályok .....	12
HomeController .....	12
ProductsController .....	12
ShoppingCartController .....	12
ShoppingCart .....	13
ShoppingCart.ShoppingCartItem .....	13
ShoppingCartViewModel .....	13
SessionHelper .....	13
FinalizationViewModel .....	14
Osztályok kapcsolata .....	14
Második részfeladat .....	15
Felhasználói esetek .....	15
Osztályok .....	16
MainCategoriesController .....	16
SubCategoriesController .....	17

AccountController .....	17
ProductsController .....	17
OrdersController .....	17
BeerkaAPIService.....	18
LoginViewModel.....	18
OrderViewModel.....	18
MainViewModel .....	19
ProductViewModel.....	19
Osztályok kapcsolata .....	20
Web API végpontok.....	21
Tesztesetek.....	22
MainCategoriesController .....	22
OrdersController .....	22
ProductsController .....	22
SubCategoriesController .....	22

## Feladat

### Italnagykereskedés

Készítsük el italnagykereskedés online rendszerét, amelyen keresztül a vásárlók italokat rendelhetnek.

Első részfeladat: a webes felületen a vásárlók adhatnak le online rendeléseket a cégnek.

- A főoldalon megjelennek a főkategóriák (Alkohol mentes italok, Alkoholos italok), illetve az egyes főkategóriához tartozó alkategóriák (Pl. Alkoholos – Whiskyk, Vodka stb.)
  - Mindegyik főkategória tartalmazzon legalább két alkategóriát.
- Egy alkategóriát kiválasztva listázódik az összes kategóriabeli termék.
  - A termékek gyártóval, típuszámmal, rövid leírással, valamint nettó és bruttó árral rendelkeznek (27% ÁFA).
  - Adott termékhez megjelenik, hogy milyen kiszerelésben kapható.
    - Minden termék kapható darabra, illetve az alábbi kiszerelések valamelyikében: zsugorfólia, rekesz, tálca.
  - A termékről megtekinthető, hogy jelenleg hány darab van készleten (A készlet mindig darabban kerül megmutatásra).
  - Egy oldalon legfeljebb 20 termék jelenik meg, a többi lapozással lehet elérni.
  - A listát ár, illetve gyártó szerint rendezhetjük (növekvő/csökkenő) sorrendbe.
- A vásárló tetszőleges számban helyezheti a kosárba egy adott termékeket, a darabszám és a kiszerelés megadásával.
  - Ha a rendelkezésre álló készletnél nagyobb számban ne lehessen a kosárba helyezni.
  - Ha egy termék adott kiszereléssel bekerült a kosárba, akkor más kiszereléssel már ne lehessen rendelni (Nem lehet egyszerre a kosárban darabban és pl. tálcával is).
  - Ha adott termék nem darab kiszereléssel került megrendelésre, akkor a kosárban jelezni kell hogy az adott kiszerelés hány darabot jelent (zsugorfólia = 6 darab, tálca = 24 darab, rekesz = 12 darab), illetve a készletet is ennek megfelelően kell validálni (Pl. 24 db elérhető akkor ne tudjon két tálcával rendelni)
  - A kosárban lévő termék törölhető, illetve egy kattintásra az egész kosár törölhető.
- A kosarat bármikor meg lehet tekinteni, illetve látható a nettó/bruttó végösszeg is.
- A kosárban található rendelés véglegesíthető az adatok alábbi adatok megadásával: név, cím, telefonszám, e-mail cím.

Második részfeladat: az alkalmazottak az asztali grafikus felületen keresztül adminisztrálhatják a rendeléseket, illetve a termékeket és a raktárkészleteket.

- Az alkalmazott bejelentkezhet (felhasználónév és jelszó megadásával) a programba, illetve kijelentkezhet.
- Bejelentkezve az alkalmazás listázza az egyes főkategóriákat és alkategóriákat, és egy alkategóriákon belül az ahhoz tartozó termékeket az alábbi adatok mutatóásával: gyártó, típusszám, rövid leírás, nettó és bruttó ár, elérhető kiszerelések és készlet.
  - Egy adott terméknek módosítható a raktárkészlete.
  - A felületen lehetőség van új termék megadására is, a gyártó, típusszám, rövid leírás, nettó ár, elérhető kiszerelések és készlet megadásával. A bruttó árat a rendszer automatikusan számolja (27% ÁFA).
- 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.
  - A rendelések szűrhetők a megrendelő név(részlet), dátum, valamint teljesítettség állapota szerint.
  - Egy adott rendelés kijelöléssel teljesítettnek jelölhető
    - Teljesítéskor a rendelt mennyiség automatikusan levonódik a készletből.
    - A teljesítés visszavonható, ebben az esetben a rendelt mennyiségek is visszahelyezésre kerülnek a készletbe.

Az adatbázis az alábbi adatokat tárolja:

- Főkategóriák (név)
- Alkategóriák (főkategória, név)
- Termékek (gyártó, modellszám, leírás, kategória, nettó ár, raktárkészlet, kiszerelés (darabon kívül – ha van))
- Munkatársak (teljes név, felhasználónév, jelszó)
- Rendelések (név, cím, telefonszám, e-mail cím, termékek listája, teljesítettség).

## Adatbázis

Az adatok Entity Framework Core segítségével kezelődnek.

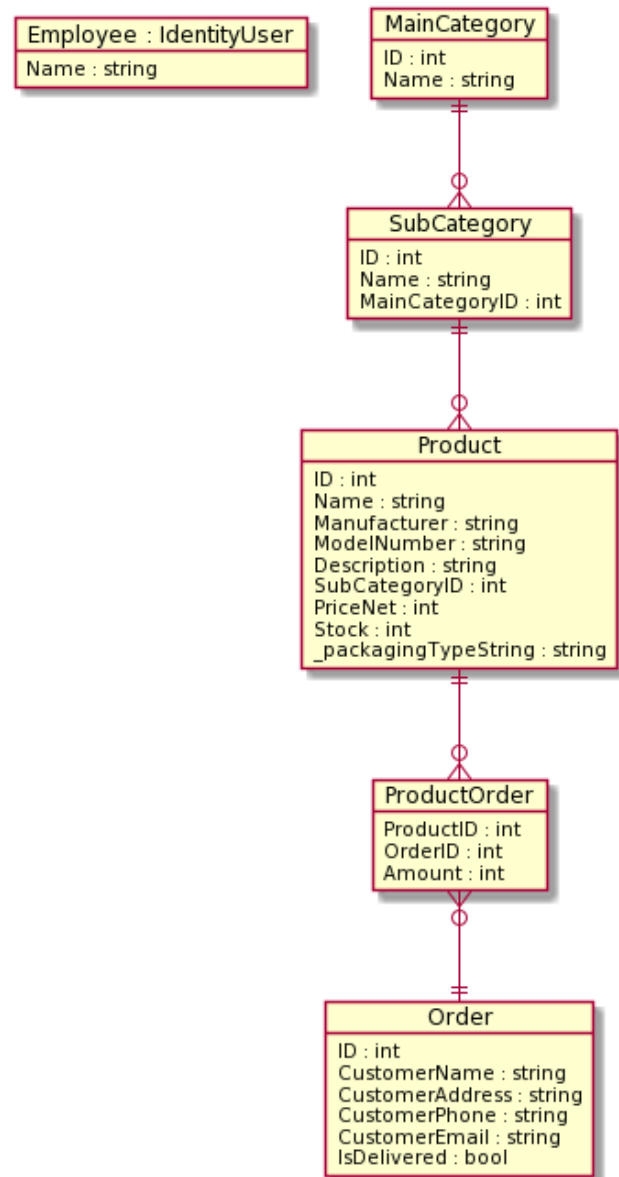
Az adatbázisban minden eltárolandó elemhez tartozik egy osztály, ami az elemhez tartozó adatok tárolásáért felelős.

- A főkategóriák tárolásáért a 'MainCategory' osztály felel.
- Az alkategóriák tárolásáért a 'SubCategory' osztály felel.
- Az termékek tárolásáért a 'Product' osztály felel.
- Az munkatársak tárolásáért az 'Employee' osztály felel.
- Az rendelések tárolásáért a 'Order' osztály felel.

Rendelések tárolásakor szükség van egy módra, amivel tetszőleges számban el lehet tárolni a rendelt termékeket, valamint hogy melyik termékből hány került megrendelésre. Ezért felel az összetett kulccsal rendelkező 'ProductOrder' osztály, mely minden eleme eltárolja egy rendelés és egy termék azonosítóját, valamint a mennyiséget is.

Az alkalmazottak adatainak tárolásáért felelős 'Employee' tábla az 'IdentityUser'-ből van származtatva. És ki van egészítve a 'Name' tulajdonsággal, hogy a felhasználónévén kívül az alkalmazott teljes neve is tárolható legyen.

Az adatbázissal a kontrollerek nem közvetlenül a kontextuson keresztül kommunikálnak, hanem szolgáltatásokon keresztül ('BeerkaService', 'BeerkaWebService').

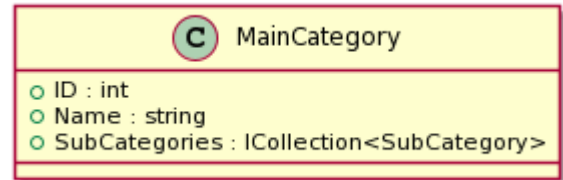


## Osztályok

A program osztályai a továbbiakban kerülnek felsorolásra.

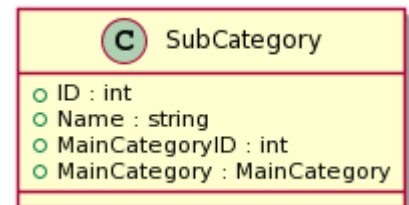
### MainCategory

A főkategóriát megvalósító osztály. A 'SubCategories' navigációs tulajdonságon keresztül gyorsan elérhetők a releváns alkategóriák.



### SubCategory

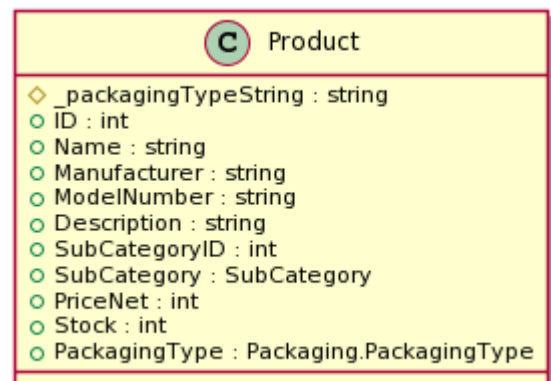
Az alkategóriát megvalósító osztály. A 'MainCategory' navigációs tulajdonságon keresztül elérhető a releváns főkategória.



### Product

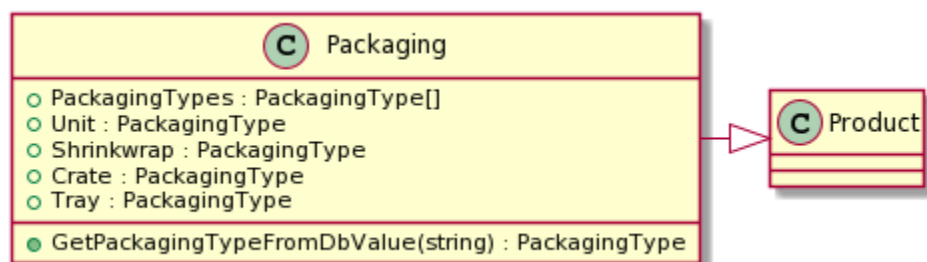
A termék megvalósításáért felelő osztály. Rendelkezik a 'SubCategory' navigációs tulajdonsággal a releváns alkategória gyors elérhetőségének érdekében.

A kiserelést leíró karakterlánc elérése és a későbbi bővítés lehetőségének fenntartása érdekében egy alosztályban rendelkezésre állnak a megfelelő karakterláncok és kiserelés méretek.



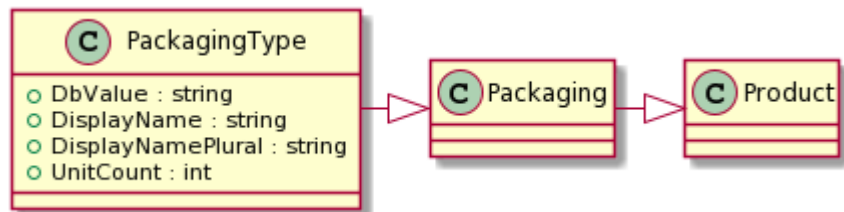
### Product.Packaging

A 'Product' osztály alosztálya. Könnyű elérést és tárolást biztosít a kisereléseket leíró adatoknak. A 'GetPackagingTypeFromDbValue' funkció használatával a megfelelő kisereléstípus gyorsan lekérhető.



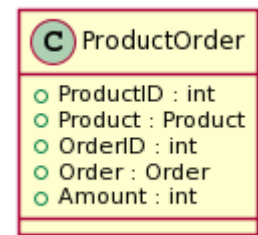
### Product.Packaging.PackagingType

A 'Packaging' osztály alosztálya. Egy kiszereléshez tartozó adatok tárolásához használható. Tárolja az adatbázisba kerülő karakterláncot, a megjeleníthető nevet, a megjeleníthető nevet többesszámban, valamint a kiszerelés által reprezentált darabszámot.



### ProductOrder

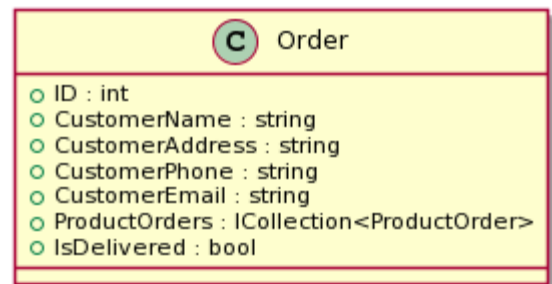
Ez az osztály felelős a rendelt termékek és darabszámaik rendelésenkénti tárolásáért. Ezen kerül tárolásra az összetartozó termékazonosító, rendelésazonosító, és a rendelt mennyiség.



### Order

A rendelés megvalósításáért felelős osztály. Ezen kerülnek tárolásra a megrendelő adatai és a rendelés teljesítettsége.

Rendelkezik a 'ProductOrders' navigációs tulajdonsággal, melyen keresztül könnyedén lekérhetők a megrendeléshez tartozó termékek listája.

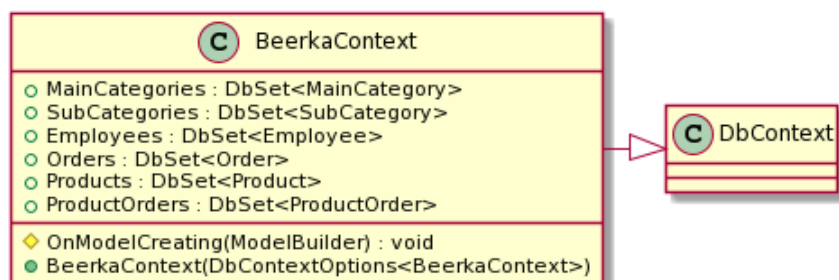


### Employee

Az alkalmazott megvalósításáért felelős osztály. Az alkalmazott adatait tárolja. Az 'IdentityUser' osztály leszármazottja, kiegészítve egy 'Name' tulajdonsággal, az alkalmazott nevének tárolása érdekében.

### BeerkaContext

'DbContext' alosztálya. Az adatbáziskontextus megvalósításáért felelős. Ezen keresztül kérdezhetők le az adatbázis adatai.





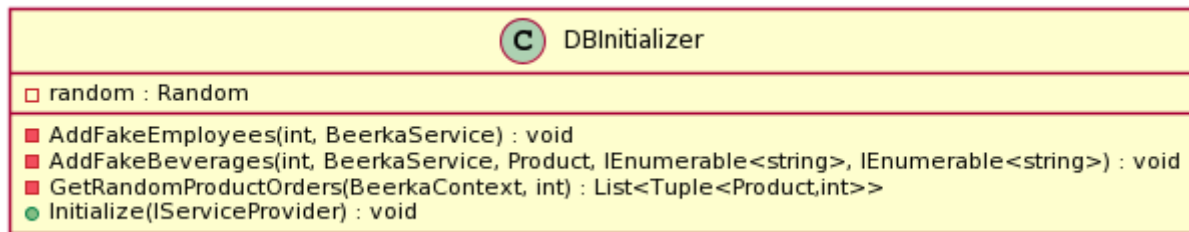
## BeerkaService

Az adatbáziskontextusra épülő szolgáltatás. Az adatbázis adatai könnyű kezeléséért, valamint a jelszavak kódolásáért, ellenőrzéséért, validációért felelős.

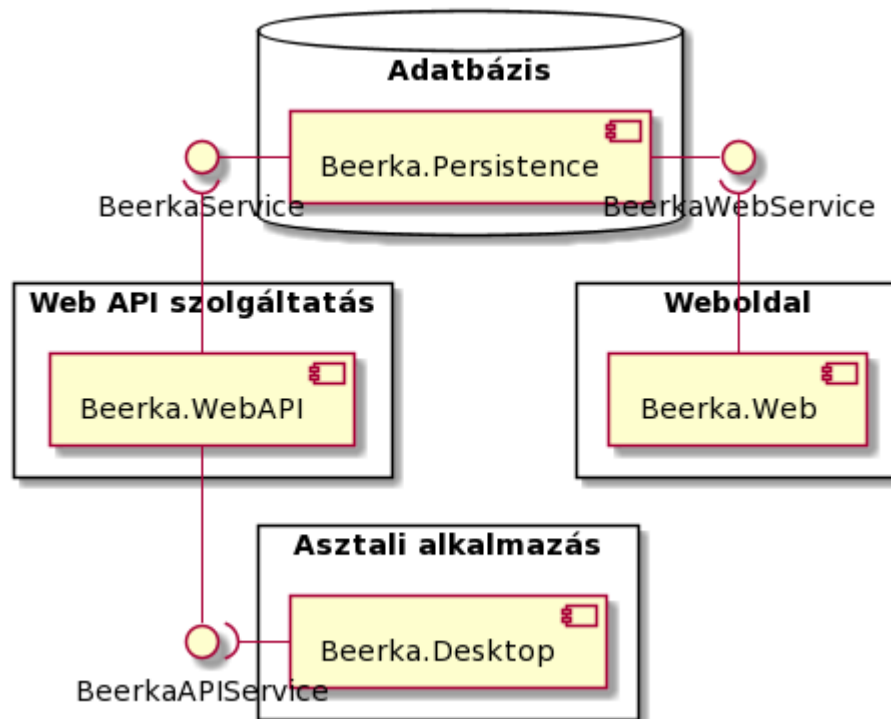
C BeerkaService	
□	_context : BeerkaContext
■	CreatePasswordHash(string, string) : string
■	CreateRandomSalt(int) : string
●	AddMainCategory(string) : void
●	GetProducts() : List<Product>
●	RemoveProduct(Product) : void
●	RemoveProduct(int) : void
●	AddOrder(Order, IEnumerable<Tuple<Product, int>>) : void
●	AddOrder(string, string, string, string, bool, IEnumerable<Tuple<Product, int>>) : void
●	GetOrder(int) : Order
●	GetOrders() : List<Order>
●	RemoveOrder(Order) : void
●	RemoveOrder(int) : void
●	CheckPassword(string, string, string) : bool
●	CheckPassword(string, Employee) : bool
●	AddEmployee(Employee) : void
●	AddEmployee(string, string, string) : void
●	RemoveEmployee(Employee) : void
●	RemoveEmployee(int) : void
●	GetEmployee(int) : Employee
●	GetEmployee(string) : Employee
●	GetEmployees() : List<Employee>
●	AddMainCategory(MainCategory) : void
●	RemoveMainCategory(MainCategory) : void
●	RemoveMainCategory(int) : void
●	RemoveMainCategory(string) : void
●	GetMainCategories() : List<MainCategory>
●	GetMainCategory(int) : MainCategory
●	GetMainCategory(string) : MainCategory
●	AddSubCategory(string, string) : void
●	AddSubCategory(SubCategory) : void
●	RemoveSubCategory(SubCategory) : void
●	RemoveSubCategory(int) : void
●	RemoveSubCategory(string, string) : void
●	GetSubCategories() : List<SubCategory>
●	GetSubCategory(int) : SubCategory
●	GetSubCategory(string, string) : SubCategory
●	AddProduct(Product) : void
●	AddProduct(string, string, string, string, int, int, int, Product.Packaging.PackagingType) : void
●	GetProduct(int) : Product
●	ValidateShoppingCartItem(ShoppingCart, Product, Product.Packaging.PackagingType, int) : ShoppingCartItemError
●	BeerkaService(BeerkaContext)

## DbInitializer

Az adatbázis iniciális feltöltéséért felelős. Számos fikcionális adatot generál az adatbázisba, a tesztelés és bemutatás megkönnyítése érdekében.



## Komponens diagram



## Első részfeladat

### Felhasználói esetek

Az italnagykereskedést webes felületét a Beerka.Web applikáció valósítja meg ASP .NET Core keretrendszerben, MVC architektúrában.

A felületen a felhasználó tudja a főoldalon az főkategóriákat és az alkategóriákat böngészni.

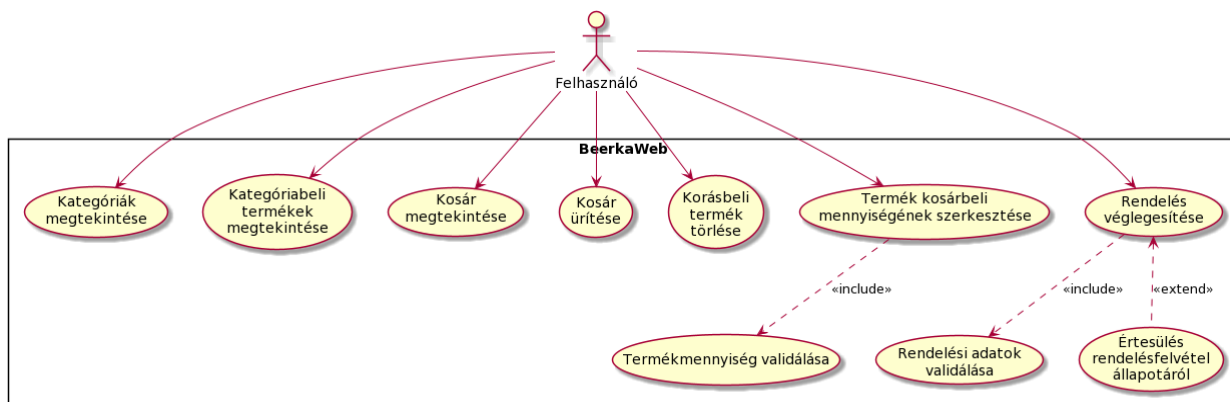
A főoldalról a felhasználó tetszőleges kategória kiválasztása után átirányításra kerül a terméklistát tartalmazó oldalra. A terméklista maximum 20 elemet mutat egyszerre, lapozásra az oldal tetején és alján lévő nyilakkal kerülhet sor. Ezen nyilak csak akkor látszanak, ha több, mint egy oldalnyi termék rendelkezésre áll. A kifogyott termékek nem kerülnek listázásra.

Tetszőleges listázott termék a kosárba helyezhető egy új oldalon a kiszereles és mennyiség megadásával. Már kosárban lévő terméket nem lehet a kosárba tenni, csak módosítani. Amennyiben egy termékből 0 "mennyiség" helyeződik a kosárba, a termék a kosárból törlésre kerül.

Minden oldalról elérhető a főoldal valamint a kosár.

A kosárban a feladatban kért adatok mind kiírásra kerülnek. Rendelkezésre áll minden termék mellett az annak módosításáért/törléséért felelős oldalra átirányító link, valamint az egész kosarat is üríthetjük a termékösszefoglaló alatt elhelyezett linkre kattintva. A rendelés véglegesítésekor meg kell adni a rendelő adatait. Ezek formátumának helyessége esetén a felhasználó átirányításra kerül egy oldalra, ahol a rendelésvétel eredményéről értesülhet. Amennyiben egy termék kosárba helyezése után egy másik felhasználó rendelése miatt kifogy, a rendelésvétel sikertelen.

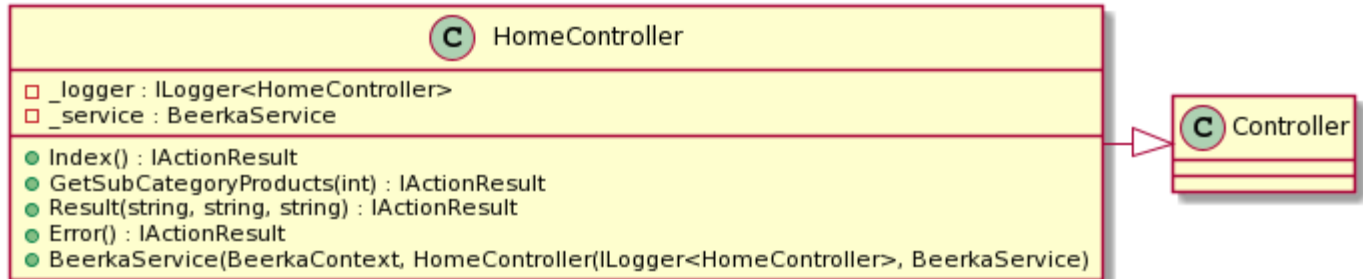
Sikeres rendelésvétel után a szolgáltatás felveszi az adatbázisba a rendelést, valamint módosítja az adatbázisban az érintett termékek készletét.



## Osztályok

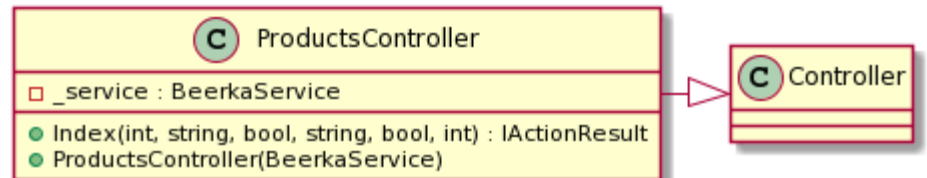
### HomeController

A főoldalhoz tartozó vezérlő. A kategórianézeten kívül lehetőség van tetszőleges üzenetek megjelenítésére a 'Result' akción keresztül. Ez az akció felel a rendelésvétel eredményéről való értesítésről is.



### ProductsController

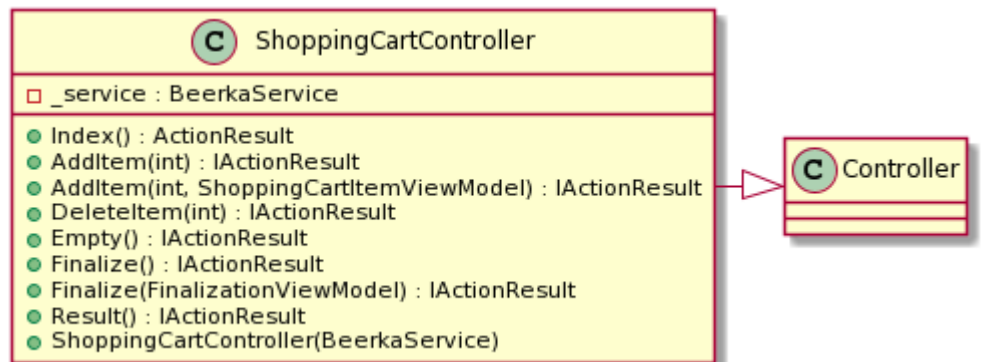
A termékmegjelenítésért felelő oldal vezérlője. A főoldalt megjelenítő akció paraméterei opcionálisan alkalmasak a megjelenített termékek számának és sorrendjének befolyásolására.



### ShoppingCartController

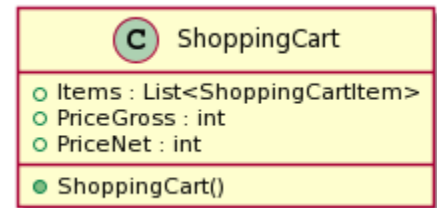
A bevásárlókosarat megjelenítő oldal vezérlője.

A kosár a munkamenetben van eltárolva. A termék kosárba helyezéséért, szerkesztéséért, törléséért, a kosár ürítéséért valamint a rendelés véglegesítését végző akciókat valósítja meg.



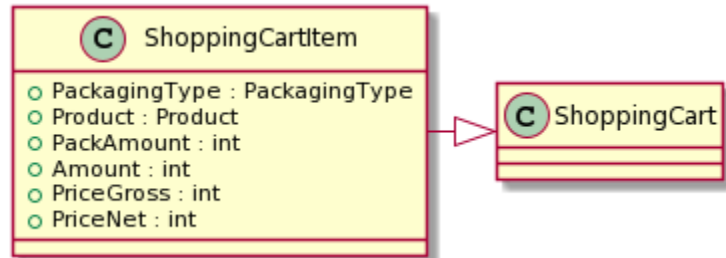
### ShoppingCart

A bevásárlókosár adatainak tárolásáért felelős. A 'PriceGross' és 'PriceNet' tulajdonságokon keresztül a végösszegek gyorsan lekérdezhetők.



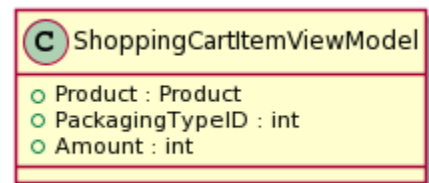
### ShoppingCart.ShoppingCartItem

A bevásárlókosárbeli termék adatainak tárolásáért felelős.



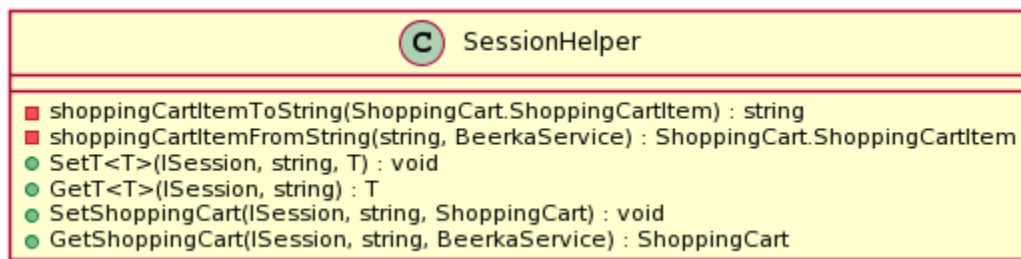
### ShoppingCartItemViewModel

A kosárbeli termék felhasználó által való szerkesztésekor a 'ShoppingCartItemViewModel' osztály felelős az adatok szállításáért és részleges validációjáért.



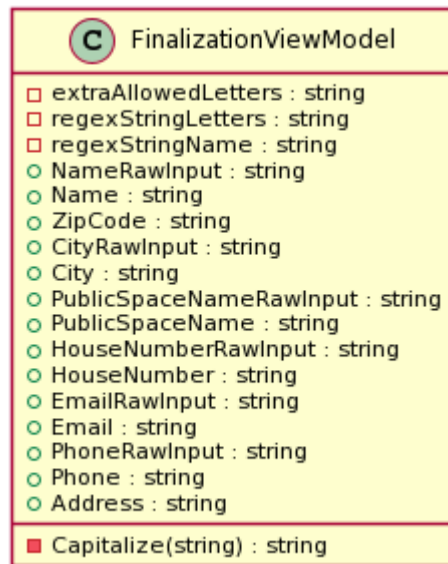
### SessionHelper

A munkamenettel kapcsolatos műveletekhez nyújt segítséget. Használatával a bevásárlókosár adatainak JSON szerializációja és deszerializációja könnyen elvégezhető.



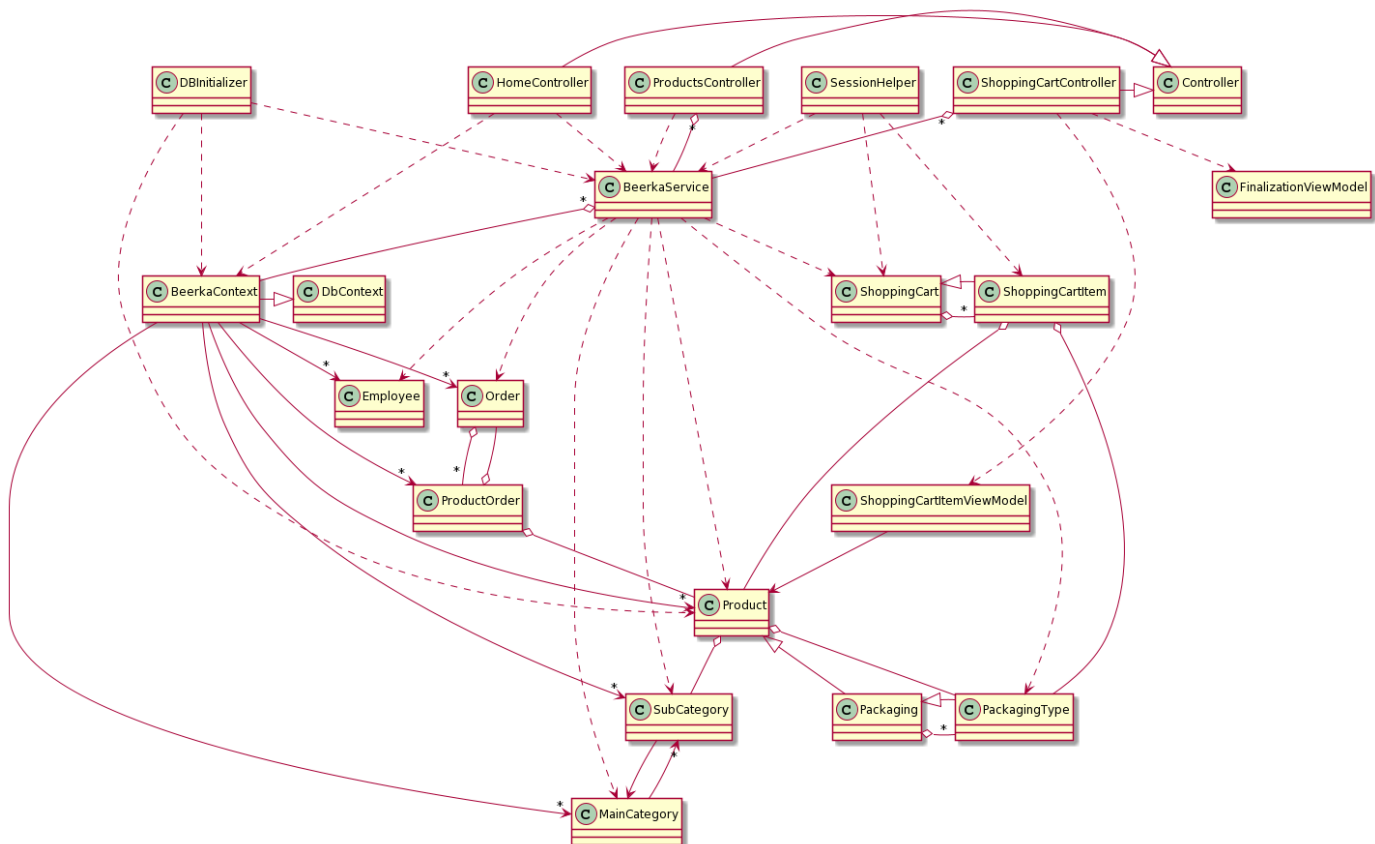
## FinalizationViewModel

A rendelés véglegesítésekor az adatok szállításáért és validációjáért felelős osztály.



## Osztályok kapcsolata

A fent felsorolt osztályok kapcsolatait a következő ábra írja le:



## Második részfeladat

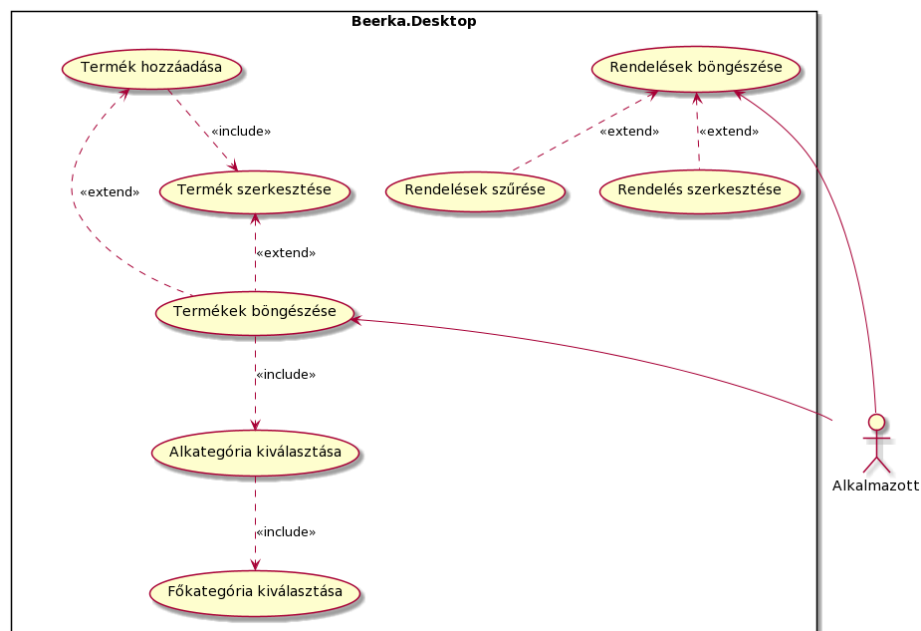
### Felhasználói esetek

Az italnagykereskedést webszolgáltatás felületét a Beerka.WebAPI, az asztali felületet a Beerka.Desktop valósítja meg.

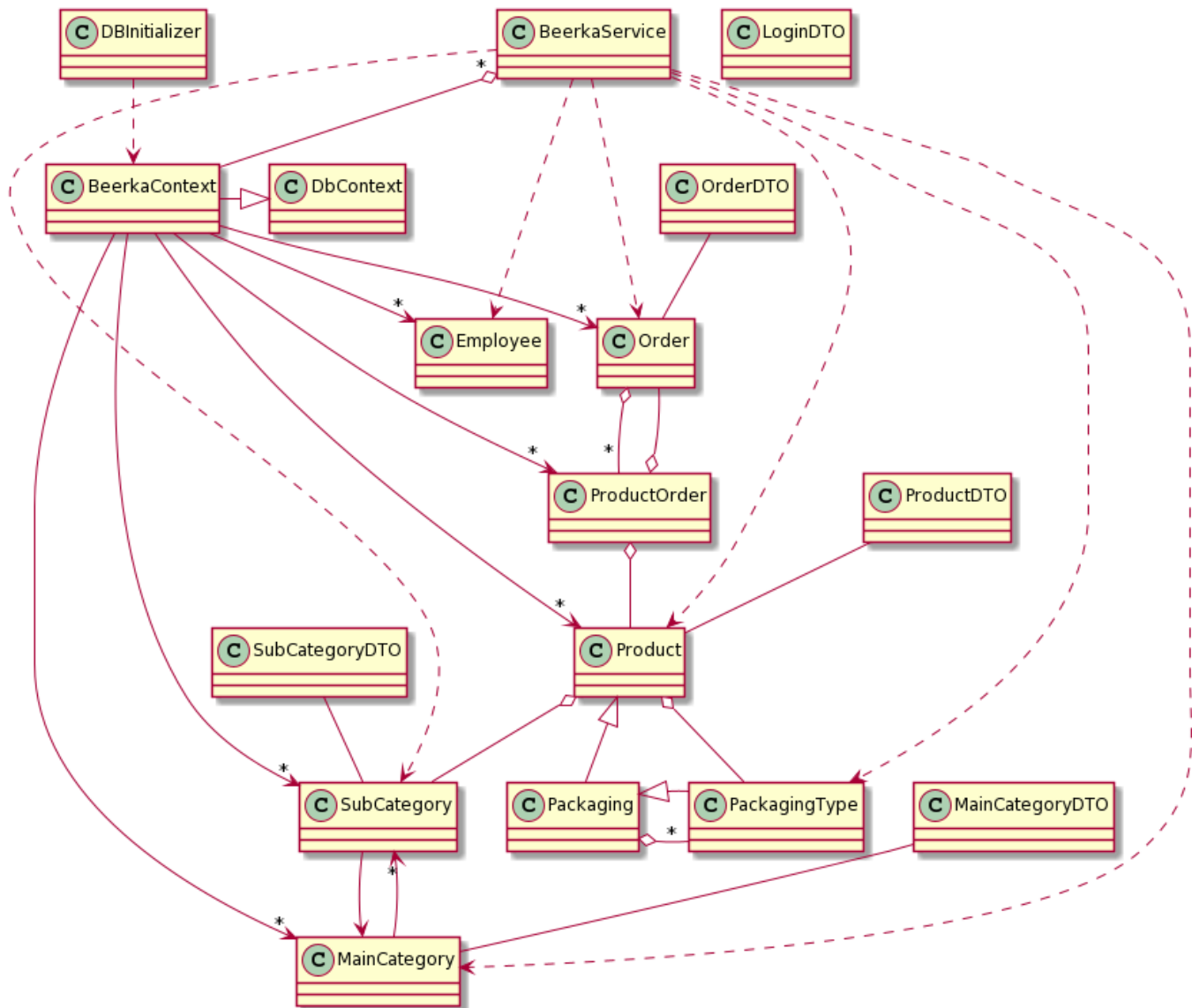
Az asztali alkalmazás induláskor bejelentkezteti a felhasználót, aki így hozzá tud férni a főablakhoz. A főablakban lehetőség van a termékek kategória szerinti böngészésére, illetve a rendelések a feladatban leírt módon való megtekintésére.

A termékeket a szerkesztés gombra kattintva lehet szerkeszteni, a készlet módosítása mellett a többi attribútum is manipulálható.

Az adatbázisban a rendelések időpontja a feladatleírás szerint nem tárolt, ezért dátum szerint szűrni nem lehet őket.



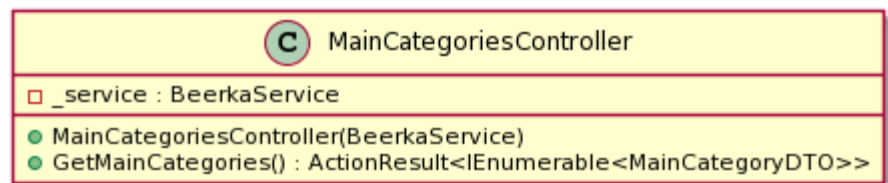
## Osztályok



Az adatbázis osztályaihoz adatszolgáltató osztályok tartoznak. Ezen osztályok lényegében megegyeznek az adatbázisban található osztályokkal, azonban a complex tulajdonságok helyett könnyen serializálható attribútumokkal rendelkeznek, valamint hiányoznak róluk a navigációs tulajdonságok. A webszolgáltatás és az asztali applikáció közötti kommunikáció az elsődleges felhasználásuk, azonban alkalmasak a nézetek és a modell közötti adatátvitelre is.

## MainCategoriesController

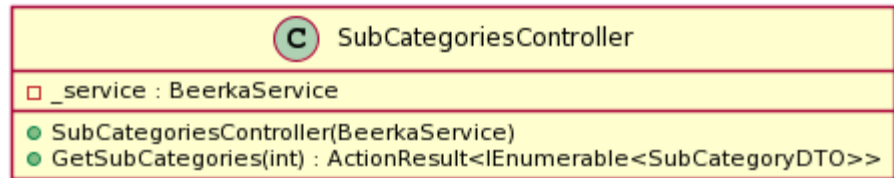
A 'MainCategoriesController' vezérlőn keresztül elérhetők a főkategóriák.





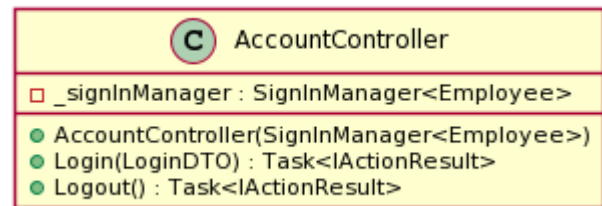
### SubCategoriesController

A 'SubCategoriesController' vezérlőn keresztül elérhetők az alkategóriák.



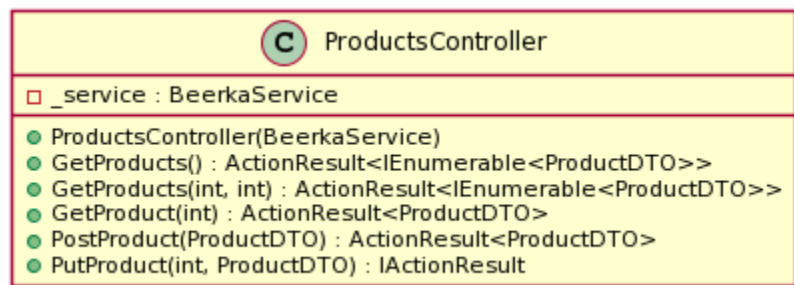
### AccountController

Az 'AccountController' felel az autentikációért, a bejelentkeztetésért.



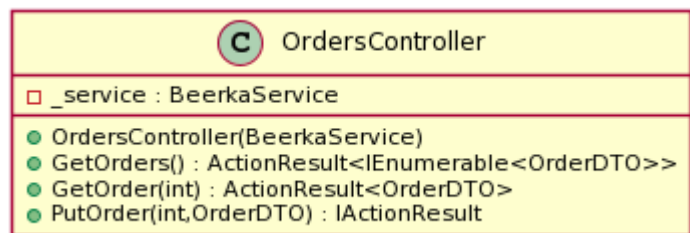
### ProductsController

A 'ProductsController' vezérlőn keresztül lehet lekérdezni a terméklistát. Ezen a vezérlőn keresztül lehet új terméket felvenni a listába, illetve meglévő terméket szerkeszteni.



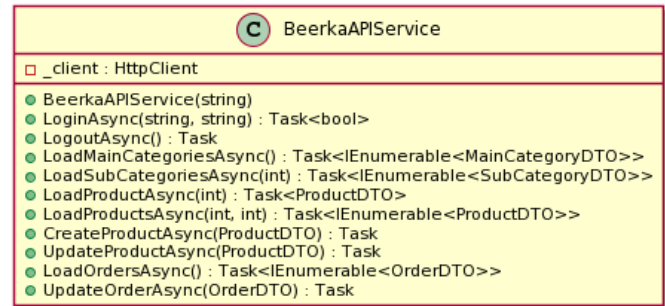
### OrdersController

Az 'OrdersController' felel a rendelések lekérdezhetőségének biztosításáért, valamint a rendelések kézbesítési státuszának frissíthetőségéért.



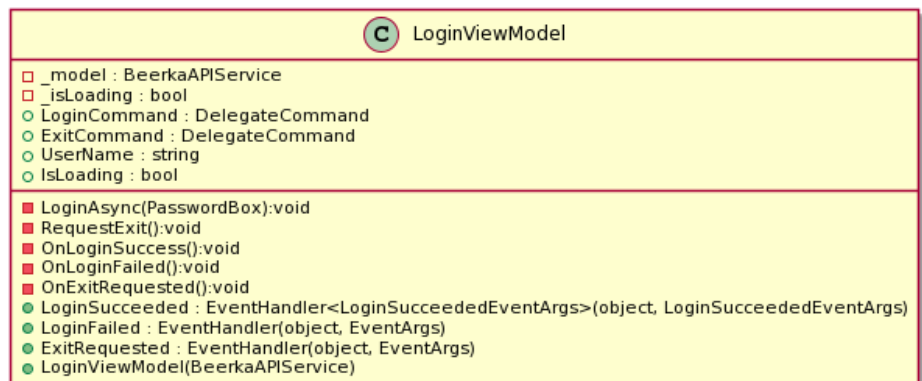
### BeerkaAPIService

Ezen keresztül lehet felvenni a kapcsolatot a webszolgáltatással. Aszinkron módon lehet lekérni, frissíteni és létrehozni új objektumokat az adatbázisban. Az adatok a korábban említett adatátviteli objektumokká konvertálódnak, majd JSON formátmba szerializálva kerülnek továbbításra, illetve fogadásra.



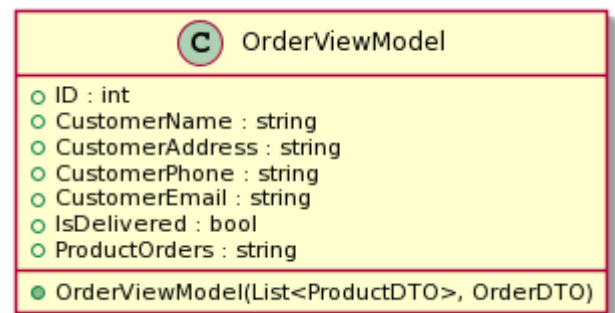
### LoginViewModel

A bejelentkezési felület nézetmodellje. Ezen keresztül tud a felhasználó belépést nyerni a főablakba.



### OrderViewModel

A rendelések adatainak megjelenítéséhez kell. Konstruktórában kerülnek átadásra azon termékek adatátviteli objektumai, melyekhez tartozik rendelés.



## MainViewModel

A főablak nézetmodellje. A főablakon kívül ez az osztály tárolja a szükséges adatokat a termékszerkesztő-ablak megjelenítéséhez.

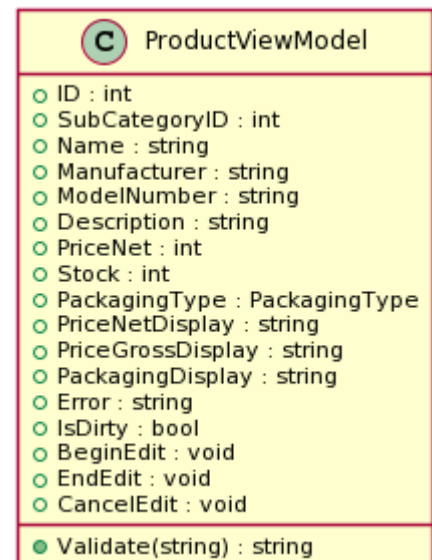
Számos tulajdonságon keresztül ellenőrizhető, hogy adott pillanatban mely kollekciók töltenek éppen, így kötésekkel könnyen megoldható, hogy a felhasználó ne tudjon például termékek lekérdezése közben kategóriát váltani.

A 'UserName' tulajdonság a főprogramban történik beállításra, mikor a felhasználó sikeresen bejelentkezik.

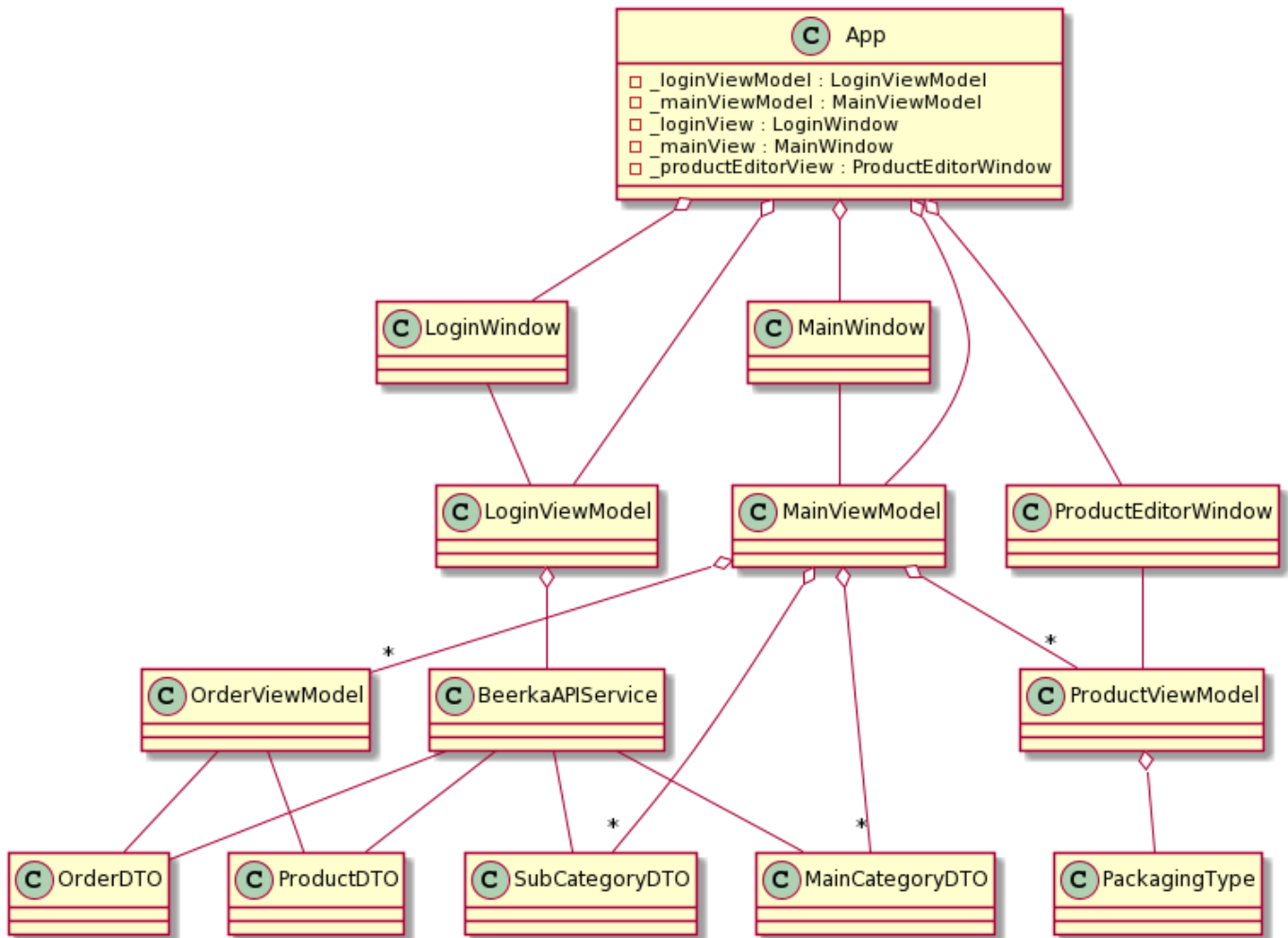


## ProductViewModel

A termékmegjelenítéshez és szerkesztéshez használt nézetmodell. Az 'IEditableObject' és 'IDataErrorInfo' interfészek implementálásának köszönhetően a szerkesztés, valamint a szerkesztés közbeni adat-validáció rendkívül könnyen, külön függvényhívások nélkül, automatikusan végbemegy.



## Osztályok kapcsolata



## Web API végpontok

<b>AccountController</b>	
api/Account/	Post, bejelentkezéshez kell. Vár egy LoginDTO-t, és visszaadja, hogy valid-e a megadott felhasználónév és jelszó.
api/Account/	Post, kijelentkezteti a felhasználót.
<b>MainCategoriesController</b>	
api/MainCategories/	Get, megadja a főkategóriák listáját.
<b>OrdersController</b>	
api/Orders/	Get, megadja a rendelések listáját.
api/Orders/{id}/	Get, megadja az 'id' ID-val rendelkező rendelést.
api/Orders/{id}/	Put, az 'id' mellett vár még egy OrderDTO-t. Frissíti az adott 'id'-val rendelkező rendelést.
<b>ProductsController</b>	
api/Products/	Get, megadja a termékek listáját.
api/Products/{mainCategoryID}/{subCategoryID}/	Get, megadja a 'subCategoryID' alkategória-ID-val rendelkező termékek listáját.
api/Products/{id}/	Get, megadja az 'id' ID-val rendelkező terméket.
api/Products/	Post, vár egy ProductDTO-t. Hozzáad egy új terméket a termékek listájához.
api/Products/{id}/	Put, az 'id' mellett vár még egy ProductDTO-t. Frissíti az adott 'id'-val rendelkező terméket.
<b>SubCategoriesController</b>	
api/SubCategories/{id}/	Get, az 'id' alkategória-ID-val rendelkező termékek listáját adja meg.

## Tesztesetek

### MainCategoriesController

Főkategóriák lekérdezése

### OrdersController

Rendelések lekérdezése.

Rendelés lekérdezése hibás ID-val.

Rendelés frissítése.

Rendelés frissítése hibás ID-val.

### ProductsController

Termékek lekérdezése.

Termékek lekérdezése fő- és alkategória ID-val.

Termékek lekérdezése hibás fő- és alkategória ID-val.

Termékek lekérdezése hibás főkategória és hibás alkategória ID-val.

Termék lekérdezése termék ID-val.

Termék lekérdezése hibás termék ID-val.

Termék frissítése.

Termék frissítése hibás ID-val.

Termék létrehozása.

### SubCategoriesController

Alkategóriák lekérdezése főkategória ID-val.

Alkategóriák lekérdezése hibás főkategória ID-val.