

WAF Beadandó

Hartyányi Kevin
hartyanyi.kevin@gmail.com
C0S0RJ

Május 2020

Contents

1	Feladat	3
2	Elemzés	3
3	Tervezés	4
4	WebAPI	8
5	Client	8
6	Tesztelés	10

1 Feladat

Készítsük el egy elektronikus termékekkel foglalkozó cég online rendszert, amelyben a vevők megrendelhetik termékeiket. 1. részfeladat: a webes felületen a vásárlók adhatnak online rendeléseket a cégnek.

1. A főoldalon megjelennek a kategóriák (pl. szórakoztató elektronika, számítástechnika, konyhai gépek), és minden kategóriára egy véletlenszerűen választott termék a kategóriából.
2. 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 és bruttó árral rendelkeznek (27% ÁFA). A listát ár, illetve gyártó szerint rendezhetjük (növekvő/csökkenő) sorrendbe.
3. 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 kosárkiüríthető egy lépésben. A kosarat bármikor meg lehet tekinteni, illetve látható a nettó/bruttó végösszeg is.
4. A webes felületen nem jelennek meg a tartósan hozzáférhetetlen (inaktív) termékek, valamint azok sem, amelyek raktárkészlete üres.

2 Elemzés

- A weblapot ASP.NET Core MVC keretrendszerben valósítjuk meg.
- A főoldalon a kategóriák listázására egy táblázatot használunk, ahol jól átlátható mind a kategória neve, mind egy random kép az ott előforduló áruk közül.
- Az oldalak közötti mozgásra az ablakok alján és tetején linkeket helyezünk el.
- A kosár tartalmának a változtatására gombokat rakunk az áruk kilistázásához, hogy könnyen lehessen hozzáadni azokat a kosár tartalmához.
- Az adatokat adatbázisban tároljuk az Entity Framework Core segítségével, amit megfelelő számú próbadattal is feltöltünk a későbbi tesztelés céljából.
- A felhasználói esetek a figure 1 ábrán láthatóak.

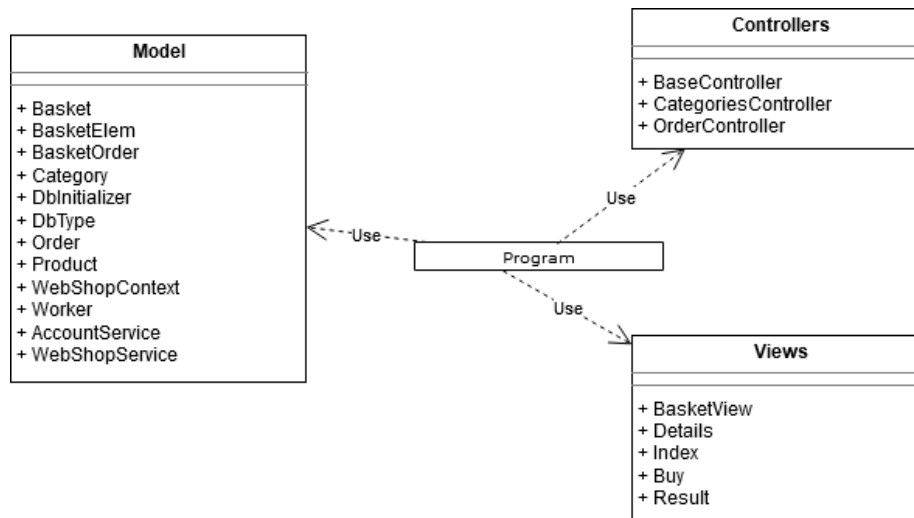


Figure 2: Az alkalmazás csomagdiagramja

- Model

- A Model tartalmaz minden adatbázisbeli elemhez egy osztályt ami annak tartalmát írja le. Ezek a következők:
 1. **Categorie:** Ebben találhatóak a különböző kategóriák amibe az árukat soroltuk. Egy név és egy árú listából állnak.
 2. **Product:** Ez maga az árú mely az adott árúhoz tartozó gyártót, modellszámot, rövid leírását, nettó árát, azt, hogy raktárkészleten van-e, és elérhetőségét tartalmazza. Amennyiben egy árú nincs raktárkészleten, vagy nem elérhető, akkor az nem fog megjelenni a vásárlók számára.
 3. **Basket:** Ez az aktuális felhasználóhoz tartozó kosár, amiben a felhasználónév (a felhasználó azonosítására használt szöveg) és a jelenleg a kosarában lévő áruk listája szerepel melyeket egy `BasketElem` által tárolunk. Ezt fogjuk majd felhasználni a felhasználó rendelésének leadásakor.
 4. **BasketElem:** A `Basket` ennek a listáját tartalmazza. Két mezőből áll ezek: az adott árú (`Product`), és egy egész szám mely azt jelzi, hogy az adott árúból mennyit kíván a felhasználó megrendelni.
 5. **Order:** Ez a már leadott rendeléseket tartalmazza. Mikor a felhasználó, úgy dönt, hogy megrendeli a kosarában lévő árukat, akkor ki kell töltenie a rendelési információkat melyek a következők: A felhasználó valódi neve (vagy annak a neve aki átfogja venni a csomagot), az e-mail címe, a cím ahová az árukat kell szállítani, és egy telefonszám a további kommunikáció fentartásának céljából. Ezeket az információkat tartalmazza ez az osztály.

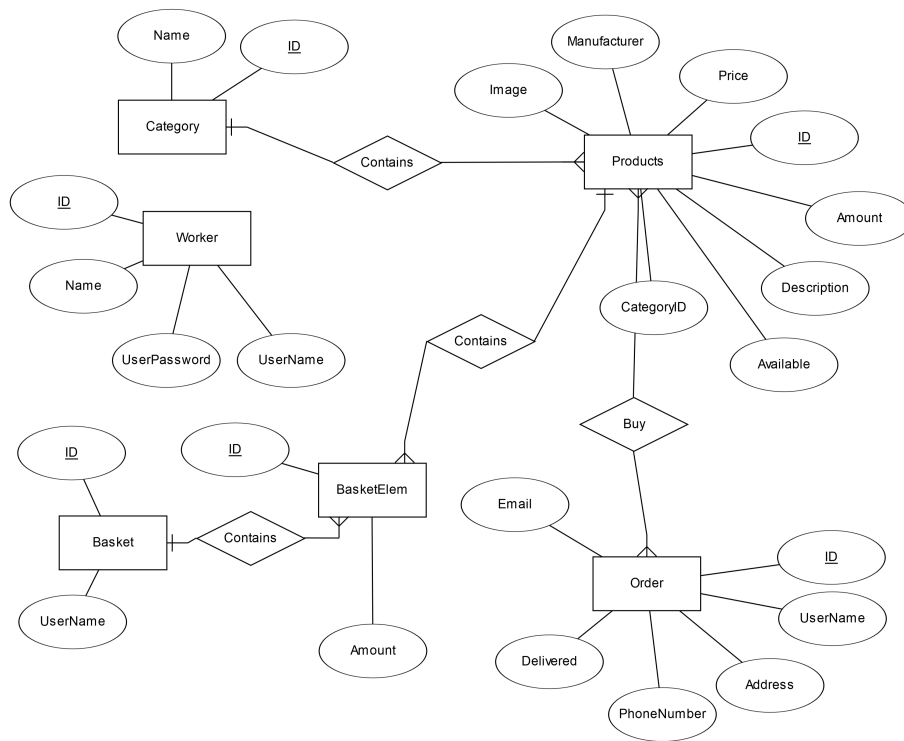


Figure 3: Adatbázis

6. Worker: Ez tartalmazza a dolgozókat akik a weblapot irányítják. A következő elemekből állnak: Név: a dolgozó valódi neve, Felhasználónév, a dolgozó bejelentkezési neve, Jelszó: a bejelentkezéshez használt jelszó.

7. A model osztálydiagramja a Figure 4 ábrán látszik.

- BasketOrder: Ez a rendelés során a felhasználó által kötelezően kitöltendő adatokat tárolja (név, cím ,e-mail, telefonszám), továbbá egy összármezőt, amin összesítjük a rendelés összegét.
- AccountServices: Ez kezeli a süti létrehozását és betöltését a felhasználó azonosítására tokenek segítségével.

- Controller

- BaseController: Ősosztály melyből a többi controllert származtatjuk, mely két taggal rendelkezik, a WebShopService-el (amivel az adatbázist tudjuk elérni), és az AccountServic-el (mellyel a felhasználót tudjuk azonosítani).
- OrderController: Ez a controller végzi a felhasználó rendelése során

az adatok bekérését és ellenőrzését, majd, ha minden adat megfelelő, akkor a rendelés leadását.

- CategoriesController: Ez a controller tartalmazza a kategóriák és az áruk megjelenítését, valamint ezen áruk a kosárba helyezését és annak kezelését, hogy minden felhasználó a neki megfelelő kosarat érje el.
- Ennek az osztálydiagramja a Figure 5-en látszik.

• View

- Index: Ez a főoldal amelyen a lehetséges kategóriák elérhetőek.
- Details: Ezen az oldalon tudjuk kilistázni az egyes kategóriákhoz tartozó árukat. Egy oldalon csak 20 termék jelenhet meg egyszerre, és a további termékeket lapozással tudjuk elérni, továbbá rendezni tudunk, gyártó és ár szerint.
- BasketView: Itt érhető el a kosár jelenlegi tartalma, amelyből tudunk elvenni vagy hozzáadni még árukat, majd ha elégedettek vagyunk, akkor megrendelhetjük az adott tartalmat.
- Buy: Ez a rendelési felület, melyen a felhasználó adatait kérjük be, melyek a rendeléshez szükségesek, amennyiben ezek helyes kerültek be, úgy elmentjük az adott rendezést és a felhasználót átirányítjuk a Result oldalra.
- Result: Ez egy rövid ablak amelyen csak értesítjük a felhasználót a sikeres rendelés rögzítéséről, továbbá még egyszer kiírjuk a rendelés teljes összeget.

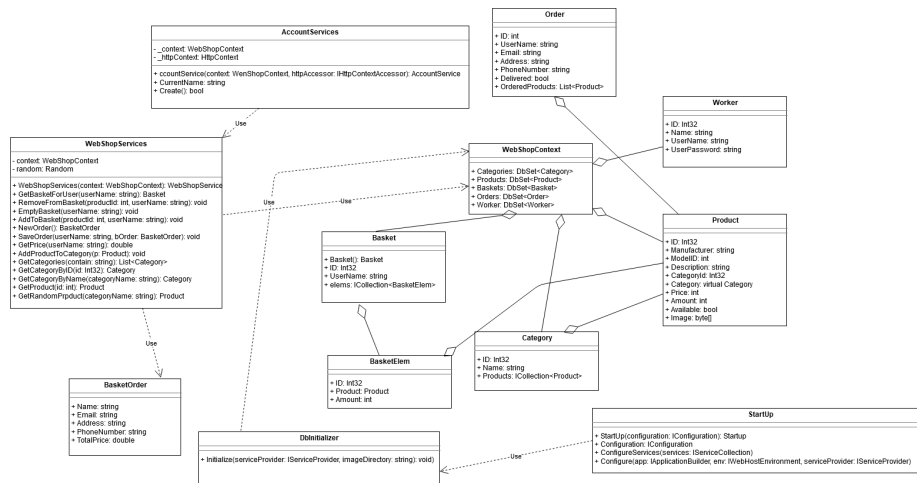


Figure 4: A model osztálydiagramja.

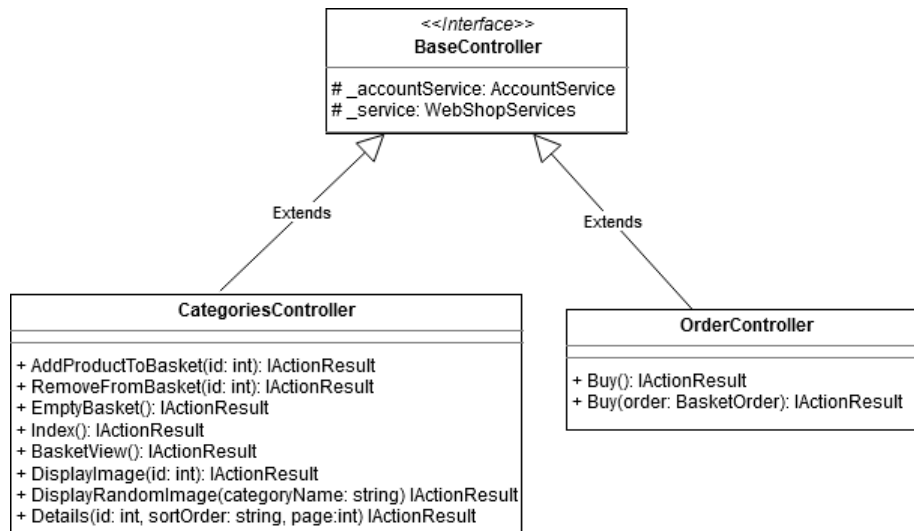


Figure 5: A controller osztálydiagramja.

4 WebAPI

A WebAPI biztosítja a dolgozók számára az adatok elérését, továbbá a dolgozók regisztrációit és bejelentkezéseit kezeli. Elemei:

- **CategoriesController:** A kategóriák lekérdezését végzi.
- **ProductsController:** A termékek lekérdezéseit végzi, ide értve az összes termék, egyéni termék, és egy kategóriába tartozó termékek lekérdezését. Lehet továbbá frissíteni is a termékeket.
- **OrderController:** A rendelések lekérdezését végzi. Lehet az összes rendelést és egyéni rendelést is lekérdezni. Továbbá lehet frissíteni is egy rendelést.
- **AccountController:** Ez a kontrollor végzi a dolgozók be- illetve kijelentkezését, továbbá ez végzi a regisztrációt is.

5 Client

Egy WPF felületet biztosít a dolgozók számára. Három fő részre osztható. View, ViewModel, és Model. A Model tartalmazza a logikai részek leírását, a View a grafikus felület biztosítását, a ViewModel pedig a kettő összekapcsolását.

- **Model**
 - **ClientAPIService:** Szervíz osztály, ami kommunikál a WebAPI-al.

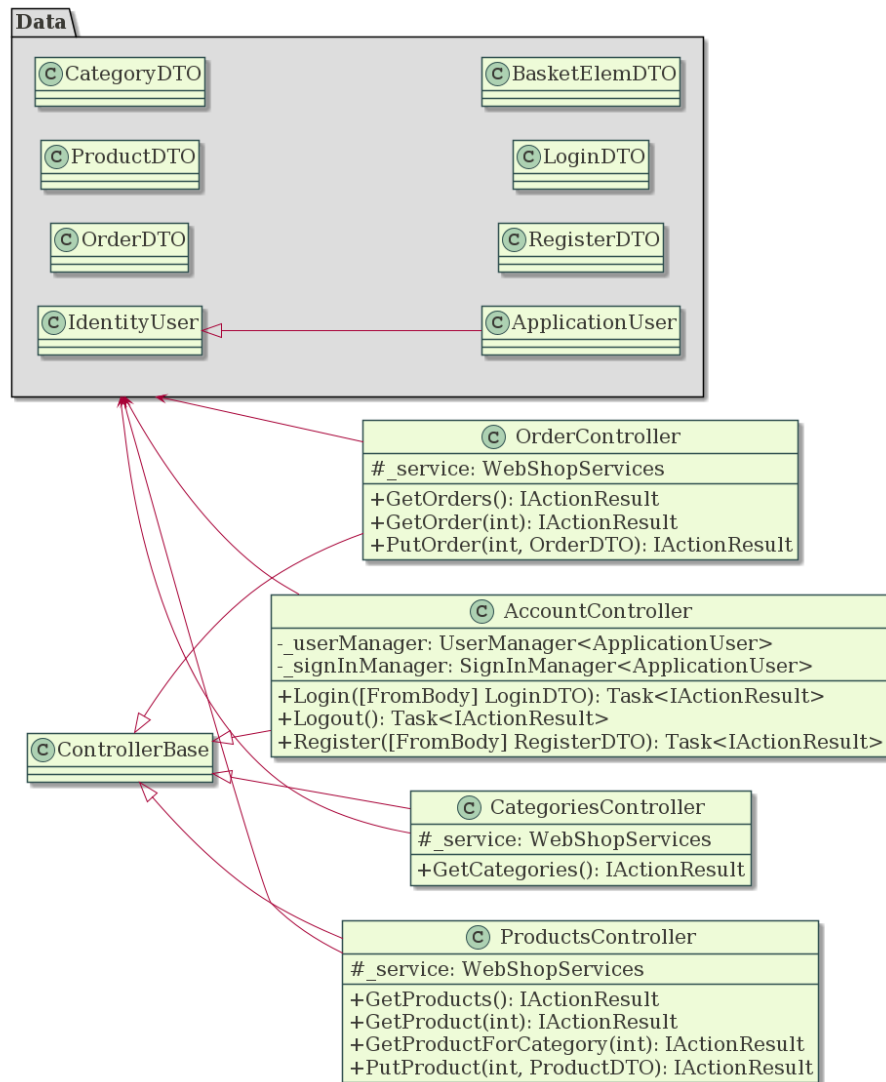


Figure 6: A WebAPI controller osztálydiagramja.

- NetworkException: Hibakezelésre szolgáló osztály
- View
 - LoginWindow: Bejelentkezési felület.
 - RegisterWindow: Regisztrációs felület
 - OrderWindow: Rendelések kezelésére felület.
 - MainWindow: Termékek listázása kategóriák alapján és kezelésére felület.
- ViewModel
 - CategoryViewModel: Kategóriák megjelenítésére segéd.
 - ProductViewModel: Termékek megjelenítésére segéd.
 - LoginViewModel: Bejelentkezési adatok megjelenítésére segéd. Dolgozók bejelentkezését végzi.
 - OrdViewModel: Rendelések megjelenítésére segéd.
 - RegisterViewModel: Regisztráció megjelenítésére segéd. Dolgozók regisztrációit végzi.
 - MainViewModel: Termékek kezelését biztosítja, lehet inaktív/aktívá tenni egy terméket továbbá növelni az elérhető mennyiséget.
 - OrderViewModel: Rendelések kezelését végzi. Lehet keresni a megrendelő neve, címe, továbbá a rendelés állapota alapján. Teljesíteni is lehet egy megrendelést.

6 Tesztelés

Ebben a szekcióban a tesztesetek lesznek ismertetve. A tesztelés UnitTest-ek felhasználásával történt az xUnit package használatával. A tesztesetek a WebAPI kontrollerei alapján kerültek megvalósításra.

- CategoriesControllerTest: A kategóriákat teszteli
 - GetCategoriesTest: Az összes kategória lekérdezését teszteli.
- OrderControllerTest: A rendeléseket teszteli
 - GetOrdersTest: Az összes rendelés lekérdezését teszteli.
 - GetOrderTest: Egy rendelés lekérdezését teszteli az id alapján.
 - GetInvalidOrderTest: Hibás rendelés lekérdezését teszteli.
 - PutOrderTest: Rendelés frissítését teszteli.
- ProductsControllerTest: A termékek lekérdezését teszteli

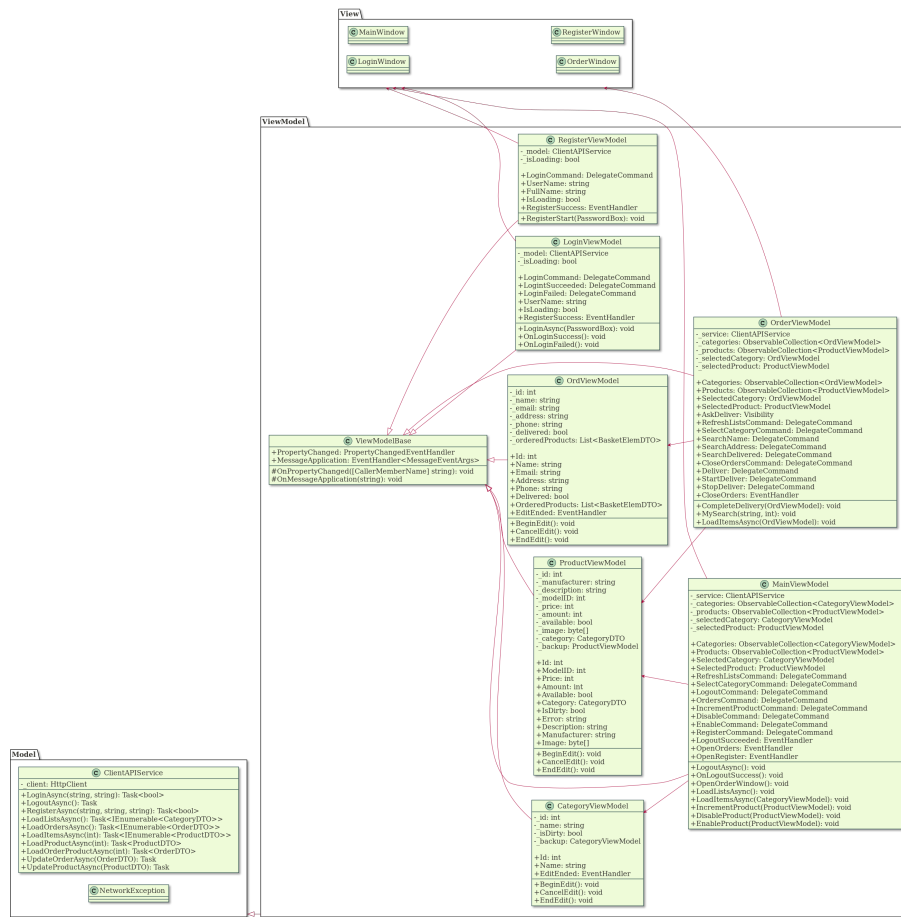


Figure 7: A Client osztálydiagramja.

- `GetProductsTest`: Az összes termék lekérdezését teszteli.
- `GetProductByIDTest`: Egy termék lekérdezését teszteli az id alapján.
- `GetInvalidProductTest`: Hibás termék lekérdezését teszteli.
- `GetProductByCategoryTest`: Egy adott kategóriába tartozó termékek lekérdezését teszteli a kategória id alapján.
- `PutProductTest`: Termék frissítését teszteli.