

# WAF Beadandó

Hartyányi Kevin  
hartyanyi.kevin@gmail.com  
C0S0RJ

Április 2020

## Contents

<b>1</b>	<b>Feladat</b>	<b>3</b>
<b>2</b>	<b>Elemzés</b>	<b>3</b>
<b>3</b>	<b>Tervezés</b>	<b>4</b>

# 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 többi 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.

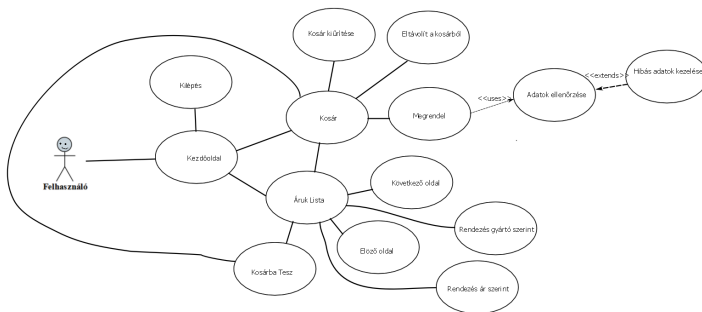


Figure 1: Felhasználói esetek

### 3 Tervezés

- Programszerkezet:
  - A weblapot ASP.NET Core MVC keretrendszerben valósítjuk meg, ennek megfelelően három fő részből fog állni ezek: Model (mely a program logikai működéséért felel), View (mely a felhasználó számára jeleníti meg a programot, a felhasználó ezen keresztül lép interakcióba a programunkal), és Controller (mely a View és a Model összekapcsolásáért felelős).
  - A felhasználó azonosítására sütit használunk, melyebe a felhasználó első bejelentkezése során elmentünk egy token, amivel azonosítani tudjuk. Ezzel a tokennel tudjuk lehetővé tenni, hogy a felhasználó későbbi visszatérése után is megmaradjon a kosárában lévő termékek listája.
- Perzisztencia
  - Az adatkezelés feladata az adatbázisban lévő információk betöltése és a Model számára elérhetővé tétele.
  - Mivel Entity Framework-öt használunk ezért egy speciális osztályban (WebShopContext) tárolódik az adatbázis és ennek a segítségével tudjuk azt elérni.
  - Azért, hogy ezt az osztályt ne közvetlenül módosítsák a controllerek egy külön osztályt (WebShopServices) készítünk, mely az adatbázisra vonatkozó wrapper függvényeket tartalmaz.
  - Az adatbázis egyedkapcsolati diagramja a Figure 3 képen látszik.

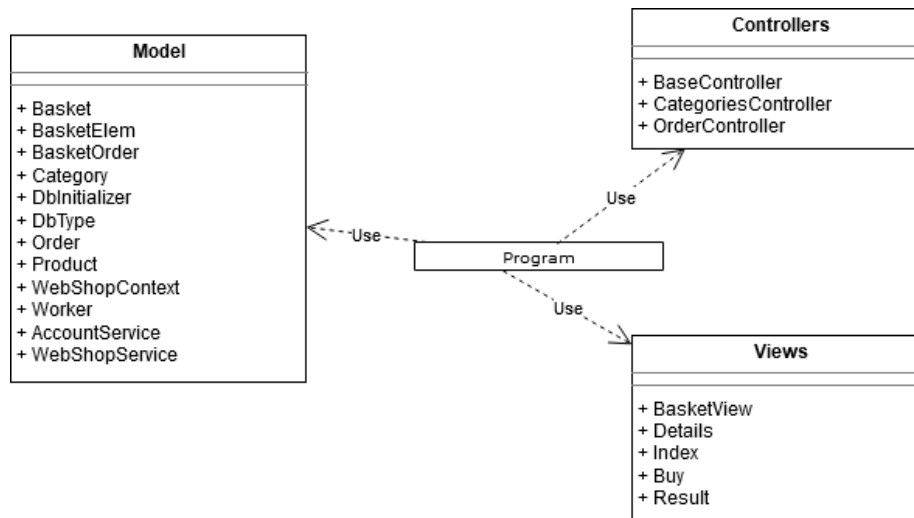


Figure 2: Az alkalmazás csomagdiagramja

- 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.

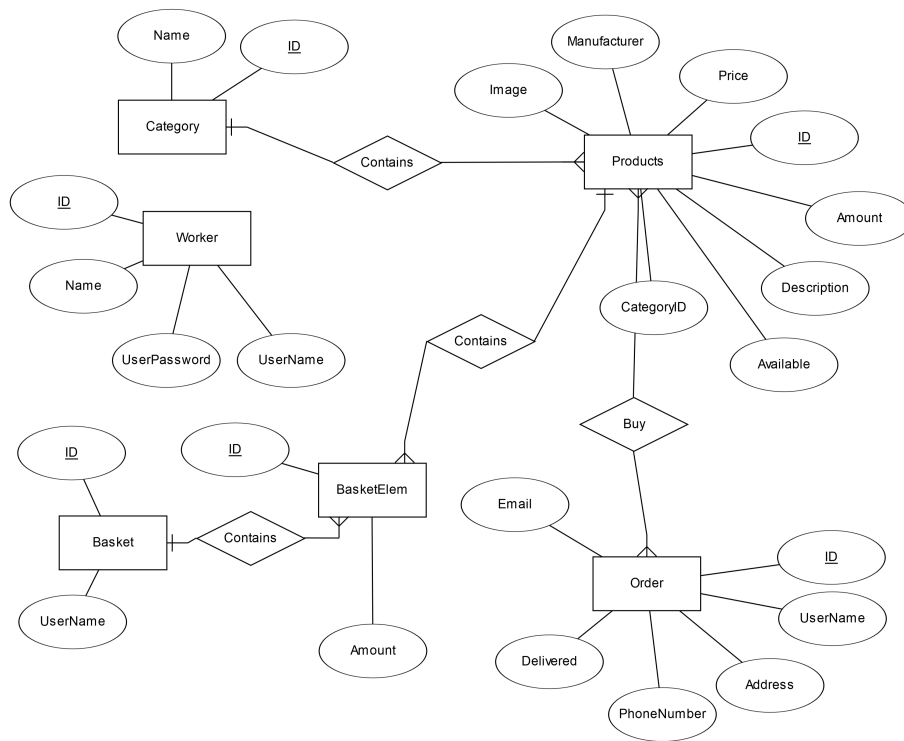


Figure 3: Adatbázis

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 sütik 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.

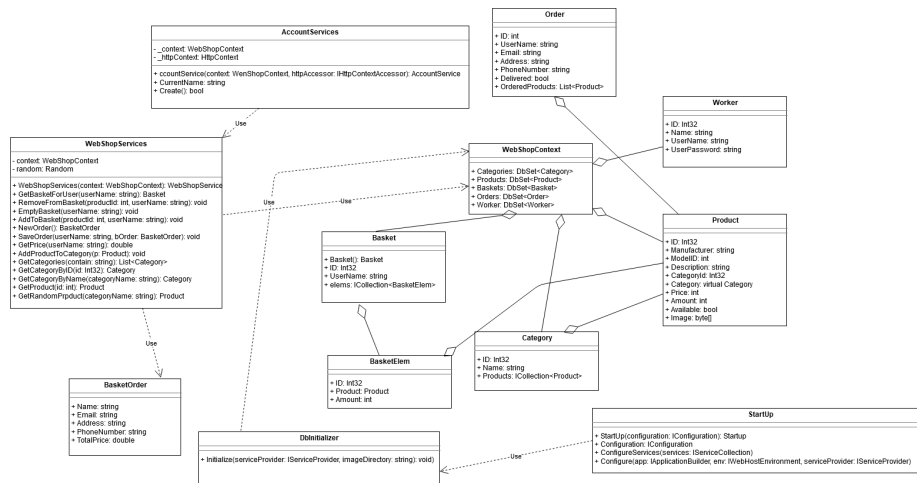


Figure 4: A model osztálydiagramja.

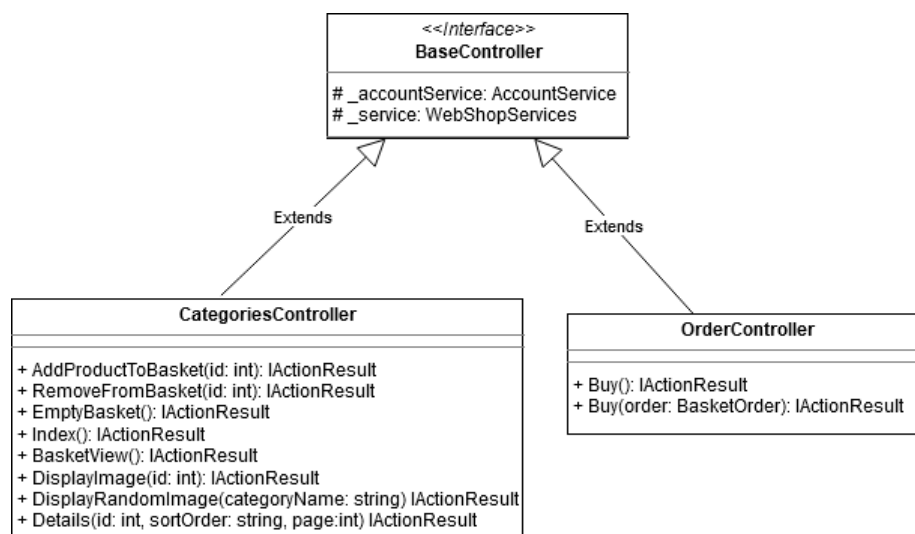


Figure 5: A controller osztálydiagramja.