

Webes alkalmazások fejlesztése - 1. beadandó

Iványi Patrik

May 23, 2018

1 Feladat

Készítsük el egy étel-futár vállalat rendeléseket kezelő rendszerét.

- A weblap főoldalán megjelennek a kategóriák (pl. levesek, pizzák, üdítők), valamint a 10 legnépszerűbb (legtöbbet rendelt) étel/ital.
- A kategóriát kiválasztva listázódnak a tételek (név és ár kíséretében), amelyek szűrhetők név(részlet)re. Ételek esetén leírás is van. Külön meg vannak jelölve a csípős, illetve vegetáriánus ételek.
- Ételek és italok tetszőleges számban helyezhetők a kosárba egy adott felső korlátig (20.000 Ft), afelett több terméket nem lehet a kosárba helyezni.
- A kosár tartalma bármikor megtekinthető, ekkor látszódnak a felvett tételek, illetve látható az összeg. Bármely tétel kivehető a kosárból.
- A rendelést törölhetjük, illetve leadhatjuk. Utóbbi esetben meg kell adnunk a nevünket, címünket, illetve telefonszámunkat, majd elküldhetjük a rendelést.

Az asztali grafikus felületet az alkalmazottak használják a rendelések, illetve a weblap tartalmának adminisztrálására.

- Az alkalmazott bejelentkezhet (felhasználónév és jelszó megadásával) a programba, illetve kijelentkezhet.
- Bejelentkezve listázódnak a leadott, illetve teljesített rendelések (leadás időpontja, teljesítés időpontja, név, cím, telefonszám, összeg), egy rendelést kiválasztva pedig listázódnak a tételek. A leadott rendelés teljesítettnek jelölhető, ekkor a rendszer rögzíti a teljesítés időpontját is. A lista szűrhető csak teljesített, illetve csak leadott rendelésekre, továbbá a rendelő nevére, illetve cím(részlet)re.

- Lehetőség van új étel, illetve ital hozzáadására (név, ár, illetve étel esetén leírás, csípős/vegetáriánus tulajdonságok megadásával). Az egyértelműség miatt nem engedélyezett több ugyanolyan nevű étel/ital felvitele.

2 Feladat elemzése

2.1 Webes felhasználói felület

A feladatot a három rétegű, Model-View-Controller architektúrában valósítjuk meg.

- Az adatok tárolásához, és a felhasználó kezeléshez létrehozunk egy adatbázist, melyben az ételek/italok, a kategóriák, a rendeléseket, valamint a vásárlás során a kosár tartalmát tároljuk. Az adatbázis modelljét az RestaurantContext biztosítja.
- Az oldal egységes megjelenítésért létrehozunk egy megosztott megjelenítőt.
- A Home könyvtár alatt létrehozuk a kedvenc ételek/italok listázásért felelős nézetet, valamint az oldalsó menüsávban egy lehetőséget, hogy az adott felhasználó szűrhesse kategóriára.
- Külön nézetet hozunk létre a kategóriák megjelenítése érdekében, a nézet a kiválasztott kategória alapján változtatja tartalmát.
- A kontrollerek felelőssége csökkentése érdekében létrehozunk két funkcionalitásért felelős osztályt, az AccountService-t, mely a felhasználói műveletekért felelős, valamint az AuctionService-t, amely a konzisztens licitálást biztosítja, valamint a megfelelő licitek és tárgyak elérést biztosítja.
- A bevásárló kosarat egy ShoppingCart modell segítségével valósítottuk meg, mely az adott session és az adatbázis között épít ki kapcsolatot.
- Külön controller felel a vásárlás lebonyolításáért, valamint az ételek közti keresésekért.

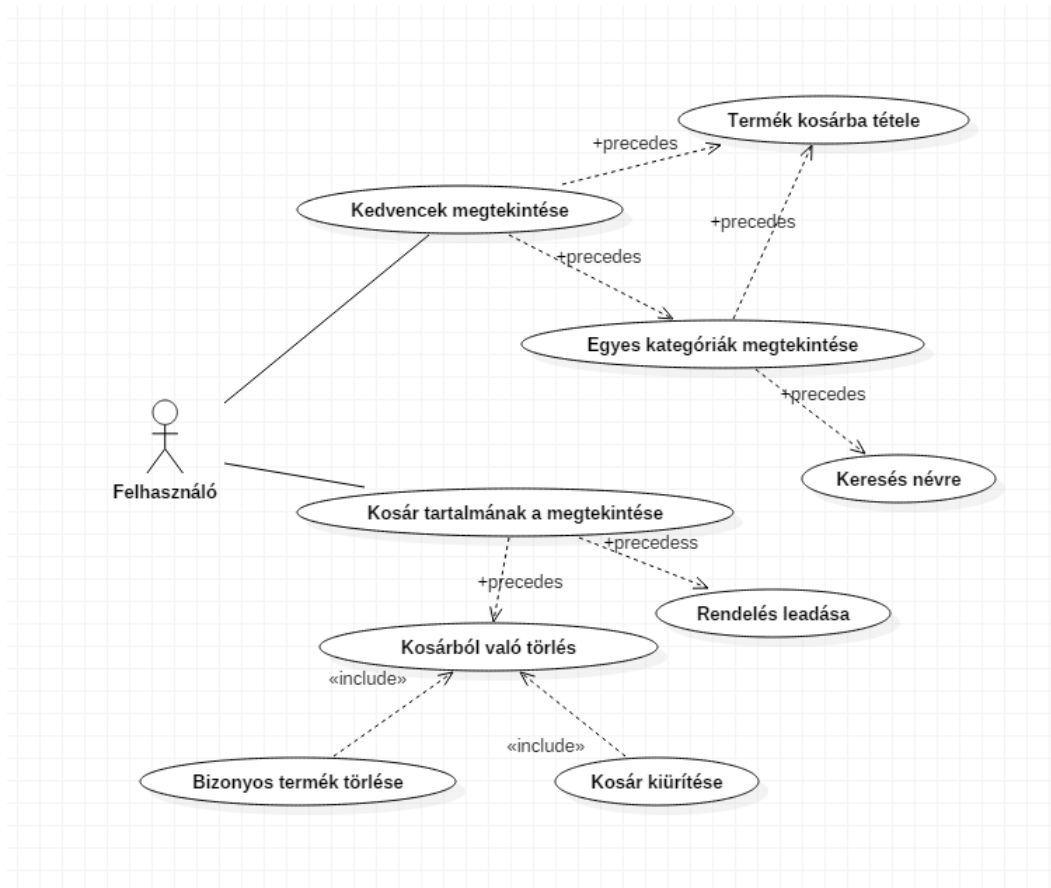
2.2 Grafikus kliensfelület

- Az adminisztratív grafikus felület, mely a WebApi-val kommunikál MVVM architektúrában valósítjuk meg WPF segítségével.

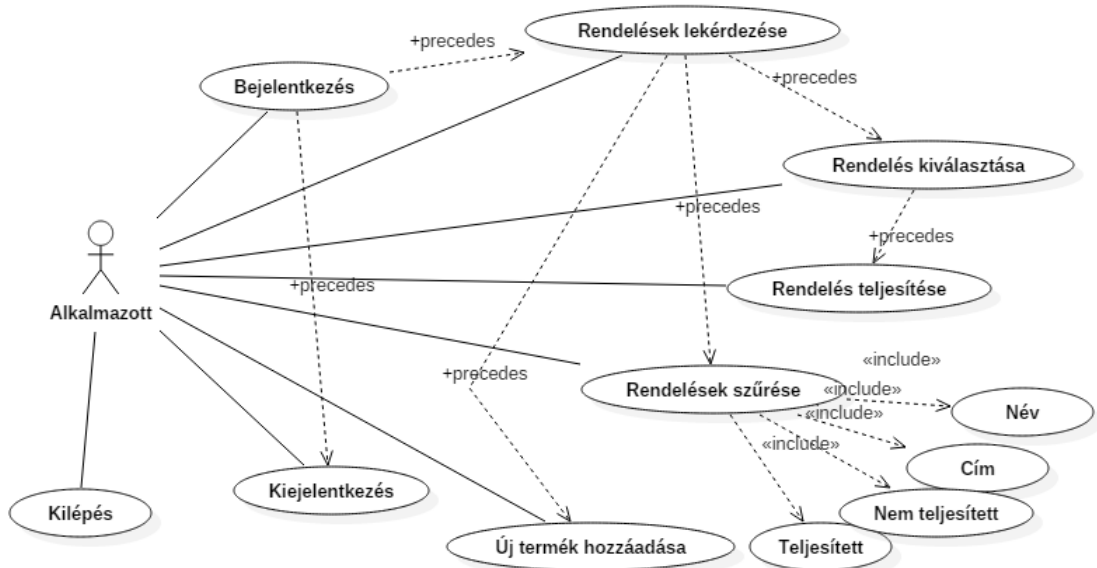
- Az alkalmazottak a bejelentkező ablak felugrásával jelentkezhetnek be az alkalmazásba és érhetik el a főablakot.
- A főablakon egy főmenüt találunk, melynek segítségével lekérdezhetők a rendelések és a termékek, továbbá itt tudjuk menteni a változtatásainkat az adatbázisban és kilépni vagy kijelentkezni az alkalmazásból.
- A rendelések lekérdezése esetén, azok egy táblázatban jelennek meg, kiválasztva közülük egy rendelést egy másik táblázatban megjelennek a rendeléshez tartozó termékek.
- A rendeléseket lehetőségünk van szűrni három szempont alapján. A *Leadott* jelölőnégyzet bekapcsolása esetén csak azon rendelések jelennek meg, amelyek még nincsenek teljesítva, *Teljesített* jelölőnégyzet esetén csak a már teljesített rendelések jelennek meg, továbbá a *Név* mezőbe írva, csak azok a rendelések jelennek meg, melyekben a név attribútum tartalmazza a mező tartalmát. A szűrést a *Szűrés* gombra kattintva tudjuk végrehajtani.
- Egy rendelést kiválasztva lehetőségünk van annak teljesítésére, ez a *Kijelölt teljesítése* gombbal tehető meg.
- Továbbá az *Új termék hozzáadása* gombbal új terméket tudunk hozzáadni az adatbázishoz, ekkor egy másik ablak ugrik fel, ahol megadhatjuk a szükséges adatokat az új termék létrehozásához.

3 Felhasználói estek

3.1 Webes felület



3.2 Adminisztratív kliens

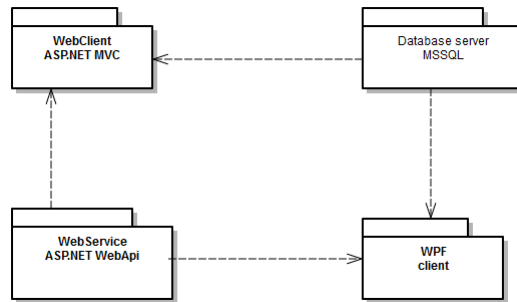


4 A rendszer szerkezete

4.1 Komponensek

Az MVC architektúrának megfelelően a programnak 4 fontos komponense van.

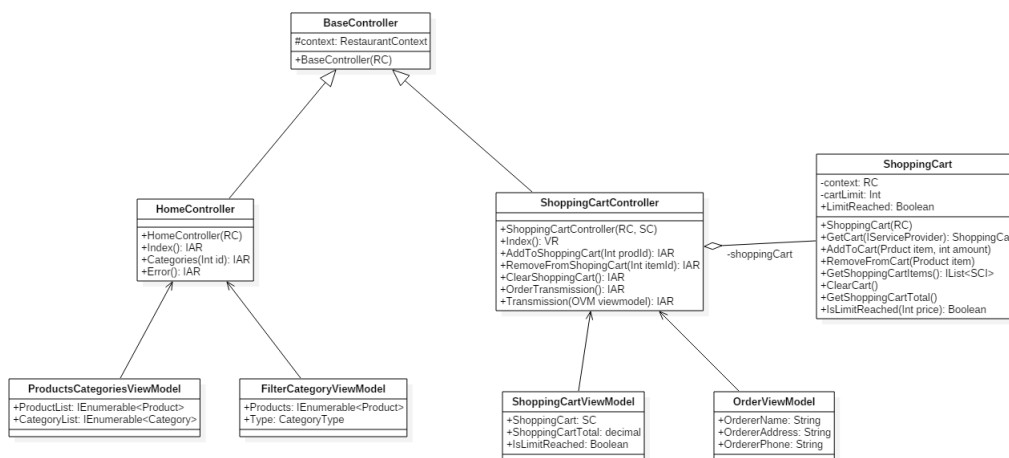
- Az adatbázis felelős a rendszer által használt adatok, és a bevárás lebonyolításához történő információk tárolására. Az adatbázishoz való kapcsolódást az Entity Framework segítségével biztosítjuk.
- A webes alkalmazási felület ASP.NET-ben MVC architektúrával megvalósítva, amely tartalmaz felhasználói nézeteket és ehhez kapcsolódó nézetmodelleket, kontrollereket, valamint az adatbázis modelljét, és a funkcionalitásért felelős kiszolgáló osztályokat.
- Az adminisztratív grafikus alkalmazást WPF felhasználásával íródott, MVVM architektúrában.



4.2 Osztályok

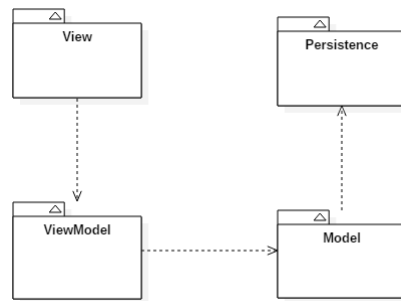
4.2.1 Webes felhasználói felület

- Külön project felelős az adatbázis modell megvalósításáért, mely kapcsolatban van az alkalmazást megvalósító modellel.
- A Models névtéren belül helyezkednek el a nézetmodellek, valamint a bevásárlást végző osztályok.
- A Controllers névtéren belül a kontrollerek helyezkednek el. A program 2 különböző kontrollere oszlik, valamint a közös funkcionalitást biztosító közös kontrollere.
- A View-en belül helyezkedik el a közös megjelenítésért felelős nézet, a kedvezmények listázásáért felelős főoldali nézet, a kosár tartalmát megjelenítő és a rendelés leadásáért felelős nézetek, valamint az egyes kategóriákhoz tartozó ételek/italokrészleteit megadó felület



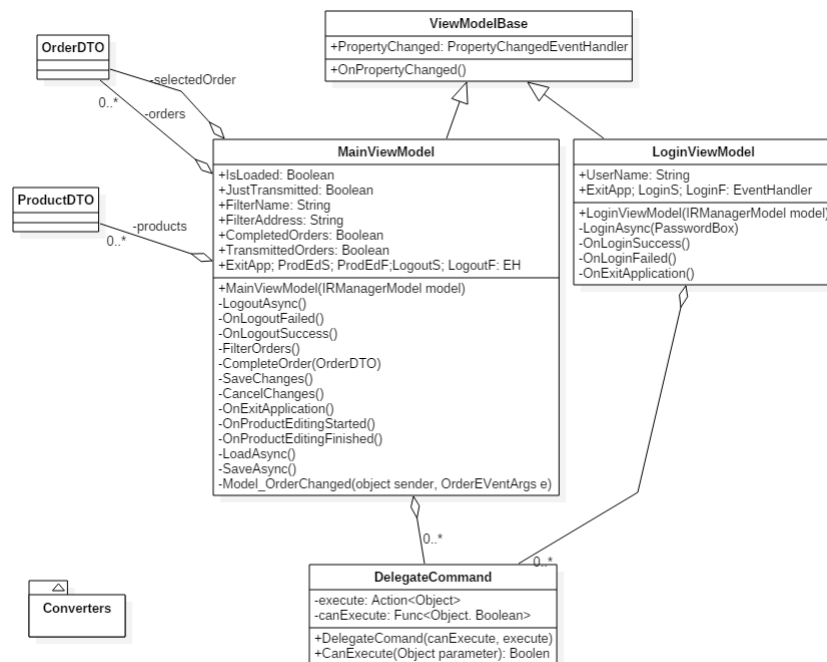
4.2.2 Adminisztratív kliens

- Az *Admin* project felelős az alkalmazás megvalósításáért.
- A grafikus konézetért a View névtérben lévő nézetek felelnek, ezeknek a dinamikus kinézete a ViewModel-hez tartozó osztályokhoz van kötve.
- A funkciók megvalósításáért a Model névtér felelős, mely közvetlenül hívja a Persistence réteget, mely az adatbázissal kommunikál.
- Mivel az adatok JSON formátumban utaznak a szerveres és az applikáció között így ezek konvertálásáért a DTO osztályok felelnek.

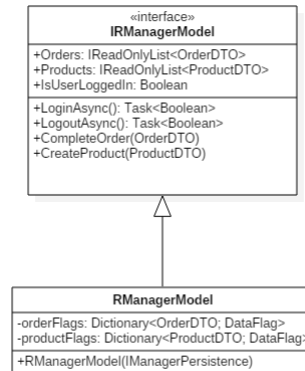


Az egyes komponensek felépítése:

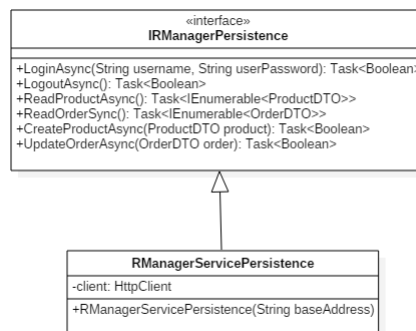
- ViewModel:



- Model:



- Persistence:



5 Adatbázis felépítése

- Az adatbázisban tároljuk az egyes ételeket/italokat.
- A különböző kategóriákat
- Az adott felhasználó által leadott rendeléseket
- A bevásárlás lebonyolításához szükséges adatokat

