

## 2. beadandó feladat dokumentáció

### DoorBash

#### Készítette:

Szalóki Sándor

H8L59S

[setr@inf.elte.hu](mailto:setr@inf.elte.hu)

#### Feladat:

Készítsük el egy étel-futár vállalat rendeléseket kezelő rendszerét.

1. részfeladat: a megrendeléseket a vásárlók a webes felületen keresztül adhatják le a cég felé.

- 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 szűrhetőek név(részlet)re. Ételek esetén leírás is van. Külön meg vannak jelölve a csípős, illetve vegetáriánus ételek.
- Ételek és italok tetszőleges számban helyezhetőek 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 összár. Bármely tétel kivehető a kosárból.
- A rendelést törölhetjük, illetve leadhatjuk. Utóbbi esetben meg kell adnunk a nevünket, címünket, illetve telefonszámunkat, majd elküldhetjük a rendelést.

2. részfeladat:

- 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.

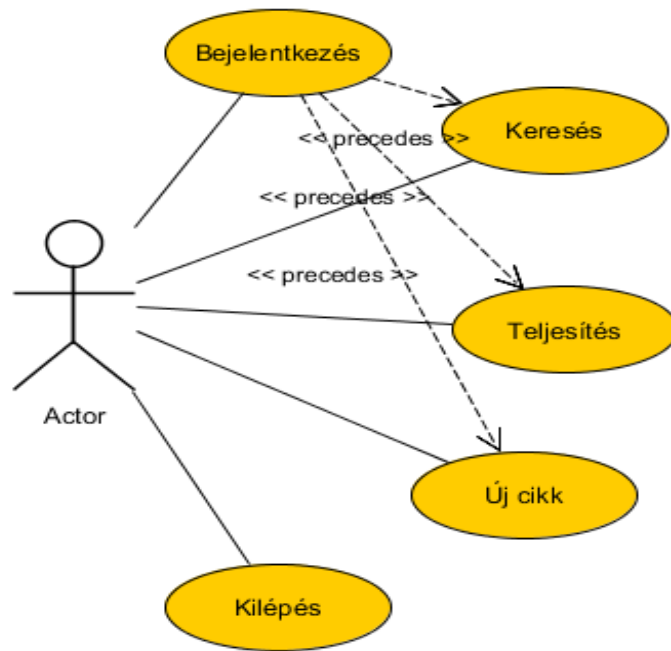
- Bejelentkezve listázódnak a leadott, illetve teljesített rendeléseket (leadás időpontja, teljesítés időpontja, név, cím, telefonszám, összeg), egy rendelést kiválasztva pedig listázódnak a tételeket. A leadott rendelés teljesítettnek jelölhető, ekkor a rendszer rögzíti a teljesítés időpontját is. A 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

Az adatbázis az alábbi adatokat tárolja:

- kategóriák (név);
- ételek és italok (név, kategória, leírás, ár, csípős-e, vegetáriánus-e);
- munkatársak (teljes név, felhasználónév, jelszó);
- rendelések (név, cím, telefonszám, megrendelt ételek és italok, teljesített-e).

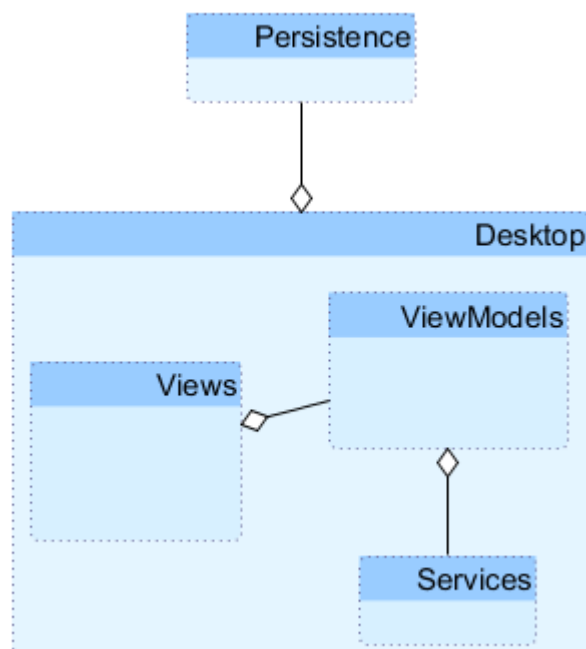
#### **Elemzés:**

- Az alkalmazás perzisztencia részét egy külső C# könyvtárként kezeljük. A weblap részt MVC segítségével hozzuk létre. A grafikus felületet WPF-el MVVM architektúrában.
- Több ablakra is szükségünk van:
  - Bejelentkező felület
  - Egy ablak, ahol listázzuk az elemeket, új cikkeket tudunk felvenni
- Az alkalmazáshoz szükséges egy adatbázis is. A feladatban leírt adatbázis jól megfogalmazható SQL adatbázisként.

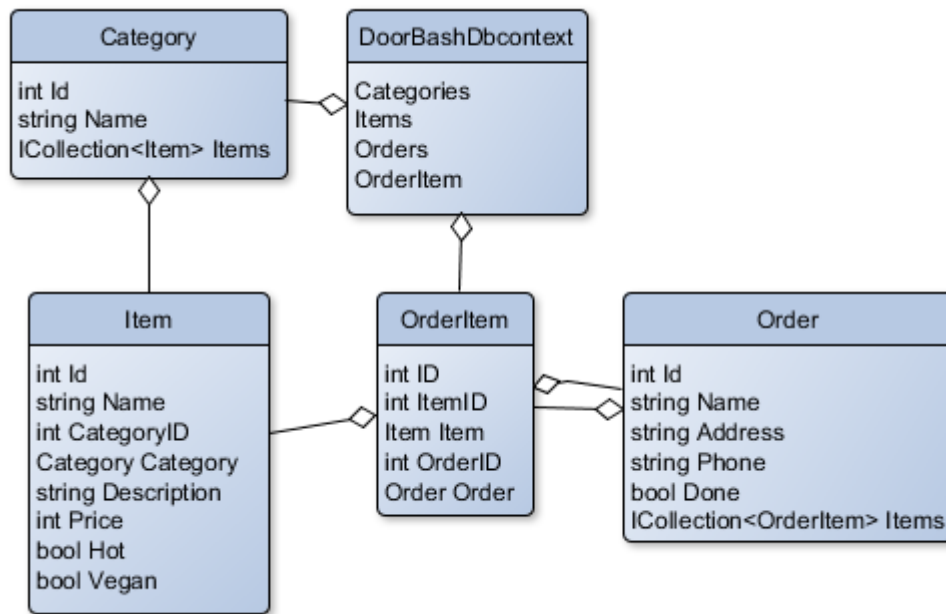


### Tervezés:

- Programszerkezet:
  - A programot MVVM segítségével valósítjuk meg.
  - A feladathoz a program kódját két projektre oszthatjuk szét. Egy DoorBash.Persistence projekt, amely felelős a háttérben működő adatbázis létrehozásért, ehhez csatlakozásért típusaival.
  - A másik projekt a DoorBash.Desktop felelős az app futtatásáért, illetve a funkcióiért. Itt találhatóak a View-ViewModel párok, melyekkel megvalósítjuk a különböző oldalakat.



- Perzisztencia:
  - Az adatbázis létrehozása, ennek kapcsolatai kidolgozása, megvalósítása.
  - Az adatbázissal való kommunikációhoz megfelelő típusok megalkotása.
  - A feladatban leírt követelményeknek megfelelő adat tárolása. Ezekhez a következő típusokat hozzuk létre: Order, Item, Category. Ezen felül szükséges egy OrderItem típus (megfelelő SQL táblával), ami egy many-many kapcsolattábla a rendelésekhez szükséges Rendelés – Áru kapcsolat tárolásához.



- Desktop:
  - Az app megvalósítása MVVM, WPF használatával.
  - Services, azaz szerver oldali hívások itt kerülnek elő.
  - 2 View szükséges a 2 oldalhoz:
    - Login – Bejelentkezés
    - MainWindow – az ablak, ahol listázunk, illetve új elemeket vehetünk fel.
  - Minden View-hoz tartozik egy-egy ViewModel is.

## WebAPI:

api/Category:

HttpGet – A kategóriák lekérdezése

api/Items/New:

HttpPost – Új étel/ital felvétele

api/Orders/?flag=&name=&address=:

HttpGet – Rendelések lekérdezése név, illetve cím szerinti szűréssel. Lehetőség csak teljesített/nem teljesítettek szűrése

api/Orders/Items/:

HttpGet – Adott rendelés életei/italai

api/Orders/Finish:

HttpPost – Adott rendelés teljesítése

## Tesztelés:

Tesztelés folyamán minden API hívás tesztelünk. Adatbázishoz egy teszt adatbázist feltöltünk. Bejelentkezett User egy teszt user middleware segítségével.