

Introcafe

dokumentáció

Készítette: Cyberlife Kft.

Lukovics Marcell, Czézár Krisztián

**Tartalomjegyzék**

**Bevezetés**

1.1. Projekt célja

1.2. Felhasználói igények és célcsoport

**Rendszerterv**

2.1. Funkcionális követelmények

2.2. Nem funkcionális követelmények

2.3. Rendszerkomponensek áttekintése

3.1. Tervezési irányelvek

3.2. Alapvető felületek bemutatása

3.2.1. Főoldal

3.2.2. Menü megtekintése

3.2.3. Rendelés leadása

3.2.4. Adminisztrációs felület

3.3. Felhasználói élmény (UX) elemek

**Adatbázis**

4.1. Adatbázisban tervezett elemek

4.2. Táblák és adattípusok

**Backend fejlesztés**

5.1. Használt technológiák és keretrendszerek

5.2. Alapvető API végpontok

5.2.1. Menü lekérdezése

5.2.2. Rendelés kezelése

5.2.3. Felhasználói jogosultságok

5.3. Biztonsági megoldások

**Frontend fejlesztés**

6.1. Használt technológiák és keretrendszerek

6.2. Interaktív elemek és funkciók

6.3. Reszponzív design megvalósítása

**Kód mellékletek**

**Jövőbeli fejlesztési lehetőségek**

9.1. Új funkciók javaslatai

**Bevezetés**

**1.1.: Projekt célja**

A Bomboclat Kft. felkeresett minket weboldalunkon feltűntetett e-mail címünkön keresztül egy kávézó szoftveres megvalósításának igényével. Skibidi János és Gyatt Anna, a cég tulajdonosai egy házaspár, kik friss cégükkel szeretnének belevágni egy vendéglátói vállalkozásba, és megalkotni egy minimalista, relaxáló és munkaorientált kávézót a kornak megfelelő modern megoldásokkal.

**1.2.: Felhasználói igények és célcsoport**

Egy csöndes relaxáció, illetve munkaorientált kávézó megalkotása. A kávézó célja, hogy a legkevésbé szociális ember is nyugodtan fogyaszthasson ezen a helyen, és elmenekülhessen a világ zajaitól egy fárasztó nap után. A kávézó tematikájának legfontosabb eleme a rendelés, illetve kiszolgálás maximális automatizálása az emberi munkaerő leváltása nélkül. Ezen cél eléréséhez elengedhetetlen a kontakt nélküli rendelésleadás és fizetés lehetőségének megvalósítása, az üzlet belsejének tudatos felépítése és elrendezése, illetve a házirend szabályzatának megfontolt megfogalmazása.

**Adatbázis struktúra**

**4.1.: Adatmodell áttekintése**

Az adatbázisban tervezett elemek:

- A regisztrált vásárlók rögzített adatai, például: E-mail cím, bejelentkezéshez szükséges jelszó, név a UI-ban való személyes megszólítás érdekében, egyedi felhasználó id az azonosíthatóság érdekében, illetve az adott felhasználó vásárlás után járó feljegyzett intropontjai, amik később kedvezményekért válthatók be.

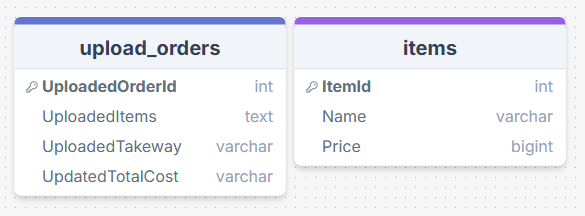
- A kávézó által forgalmazott termékek adatai: Adott termék azonosítója, az ahhoz tartozó terméknév, illetve az eladási ára.

- Az teljesítendő rendelések táblája: Tartalmazza az adott rendelés sorszámát egy 1-től 999-ig terjedő skálán, az sorszámhoz tartozó rendelt termékeket illetve azok mennyiségét, a fogyasztási típust (helyben fogyasztás/elvitel), a végösszeget, illetve a rendelés rögzítésének idejét.

**4.2.: Táblák és adattípusok**

A kávézó adatainak kezelésére „introcafe” néven hoztunk létre adatbázist, a cégtulajdonos igényeinek megfelelően. A megvalósításhoz az MySQL (MariaDB) adatkezelő nyelvet használtuk, a környezet futtatásához illetve az adattárolásához a XAMPP program került használatra (<https://sourceforge.net/projects/xampp/>), illetve a táblák grafikus kezeléséhez a devart dbForge Studio Express (<https://www.devart.com/dbforge/mysql/studio/>) változatát használtuk.

A létrehozott adatbázis táblái, oszlopai, illetve kapcsolatai az alábbi ábrán láthatók:



Az „**items**” tábla tartalmazza:

- a termék azonosítóját, ezt manuálisan kell megadni termék felvételekor, típusa INT(11) (szám) értéke nem lehet üres (NOT NULL),

- a termék nevét, típusa VARCHAR(255) (szöveg), értéke nem lehet üres (NOT NULL),

- illetve a termék árát. típusa BIGINT(20), értéke nem lehet üres (NOT NULL).

Az „**upload\_orders”** tábla tartalmazza:

- a teljesítendő rendelés sorszámát, ez a program többi része által van feltöltve, típusa INT(11) (szám), értéke nem lehet üres (NOT NULL),

- a rendelt termékek listáját, típusa VARCHAR(255) (szöveg), értéke nem lehet üres (NOT NULL),

- a rendelés fogyasztási típusát (helyben/elvitel), típusa VARCHAR(255) (szöveg), értéke nem lehet üres (NOT NULL),

- a fizetendő végösszeget, típusa BIGINT(20) (szám), értéke nem lehet üres (NOT NULL),

- a rendelés rögzítési idejét, típusa DATETIME (dátum, idő), alapértelmezett értéke az adott pillanatbeli rendszeridő a CURRENT\_TIMESTAMP metódussal, értéke nem lehet üres (NOT NULL).

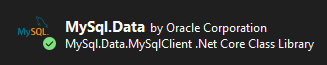
**Asztali alkalmazás, rendeléskezelő**

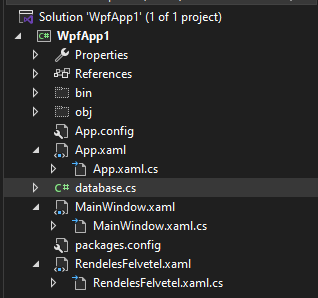
**5.1.: Asztali alkalmazás leírása**

A projekt ezen szegmense a kávézó dolgozóinak szánt felület. Használatával a foglalkoztatottak megtekinthetik az adatbázisban lévő teljesítendő rendeléseket, elkészítés után késznek jelölheti őket, így törölve az adatbázisból, illetve abban az esetben, ha a vendég bármilyen oknál kifolyólag nem képes leadni rendelését a web felületen, abban az esetben a dolgozó manuálisan is fel tudja venni a rendelést.

**5.2.: Asztali alkalmazás szoftveres megvalósítása**

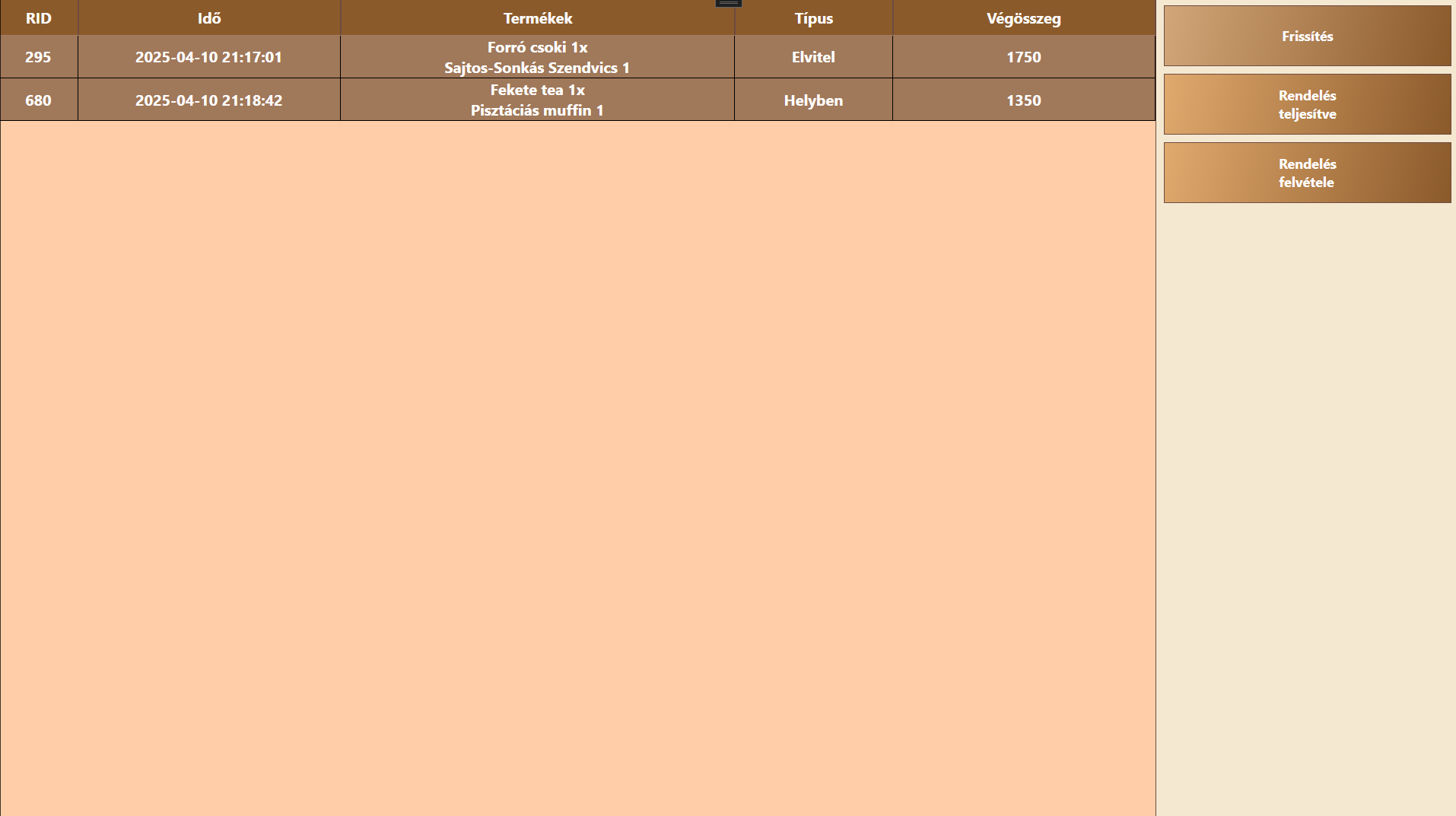
Az applikáció létrehozásához a Microsoft Visual Studio 2022 program ingyenes verzióját használtuk. Ez egy natív WPF Application, illetve C# és XAML programnyelveken lett megírva. A projekt két fő szegmensre bontható, a rendelés kezelő, illetve a rendelés felvétel felületre. Mindkét felülethez tartozik egy-egy dizájn fájl (XAML), ami a szoftver grafikus megjelenésének kódját tartalmazza, ezzel interaktál a felhasználó futtatás során, illetve egy-egy a XAML fájlhoz tartozó C# osztály, ami a mükődéshez szükséges függvényeket tartalmazza, mint például a gombok, szövegdobozok funkciói, stb. Ezen kívül a projekt része egy “database” nevezetű C# osztály is, ami az adatbázisból való lekérdezésekért/feltöltésekért felelős, így lehetővé téve a rendelések kezelését. Az applikáció teljes mértékben reszponzív, így bármilyen felbontású illetve méretű kijelzőn futtatjuk, a tartalma helyesen és olvashatóan fog megjelenni. Az adatbázis közti kommunikációhoz elengedhetetlen a MySQL.Data 9.2.0 bővítmény letöltése:



Az alábbi képen a projekt fájljai, illetve azok kapcsolatai láthatók:  


**5.3.: A rendelés kezelő:**

Az applikáció grafikus megjelenése az alábbi képen látható:

****

A felület érintőképernyős monitorokra lett elsősorban tervezve dízájn szerint. Az ablak két fő elemből áll, egy DataGrid objektumból, amiben a beérkezett rendelések kerülnek kilistázásra, illetve egy StackPanel objektumból amiben három gomb objektum található.

A DataGrid táblázat eleme a rendelés azonosítója, a felvétel időpontja (a rendelések eszerint vannak sorrendbe rakva, a legkorábban felvett lesz legfelül), a rendelt termékek, és azok mennyisége, a fogyasztás típusa, ami lehet elvitel vagy helyben fogyasztás, illetve a fizetendő végösszeg. Ezeket az adatokat az „introcafe” adatbázis „upload\_orders” táblájából kéri le a következő oszlopok segítségével: UploadedOrderId, UploadedOrderTime, UploadedItems, UploadedTakeway, UploadedTotalCost. A DataGrid tartalma automatikus frissül egy adatbázis lekérdezés segítségével, ami minden tizedik másodpercben fut le. Ha kattintással kijelölünk egy sort a DataGrid-ben, kékre változik a színe, és az adott rendelés lesz a kiválasztott elem. Az elem újboli megnyomásával, vagy egy másik rendelésre való kattintással vonhatjuk vissza a kijelölést, utóbbi esetlen az újonnan megnyomott rendelés lesz a kiválasztott.

A második rész egy StackPanel objektum, aminek 3 gomb típusú elem a tartalma. A „Frissítés” gomb a DataGrid manuális frissítésére szolgál, ha esetleg bármilyen okból nem működne az automatikus lekérdezés. Amennyiben egy rendelés ki van választva a DataGrid-ben, a rendelés teljesítése gomb megnyomásával törölhetjük mint a táblából, mint az adatbázisból. A rendelés felvétele gomb megnyomásával új ablakban megnyílik a rendelés felvétele menü.

**5.4.: A rendelés felvétel ablak:**

Az applikáció grafikus megjelenése az alábbi képen látható:



Az ablak 3 részre sorolható:

- Bal szegmens:

- Egy DataGrid objektumból áll, ami az “introcafe” adatbázis “items” táblájából listázza ki az adatokat. Megjeleníti a termék azonosítóját, nevét, illetve árát. A rendelés kezelőben megismert kijelölő rendszer itt is érvényes.

- Középső szegmens:

- Tartalma egy label objektum, ami a “Keresés…” felirat megjelenítésére szolgál, további funkciója nincs.

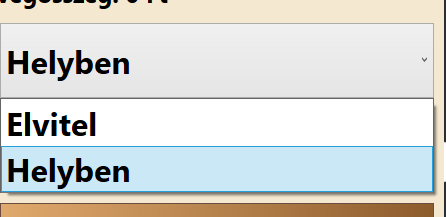
- Egy TextBox objektumból, ami a bal szegmensben való keresésre szolgál, a könnyebb használhatóság érdekében. Bele kattintással, és utána az abba való gépeléssel kereshetünk a táblázat “Név” oszlopában tárolt adatokra.

- Az utolsó objektum egy label, ami a rendeléshez eddig hozzáadott termékek listájának megjelenítése szolgál. Kiírja a termék nevét, illetve ha hozzáadás során érzékeli a program, hogy az adott termék már szerepel a listában, akkor szorzóként hozzáadja az adott termék mennyiségéhez. Feltöltéskor ez lesz az adatbázis UploadedItems mező értéke.

Jobb szegmens:

- Tartalma egy végösszeg label, ami az eddig felvett termékek összesített árát tárolja, később ez lesz az adatbázisba feltöltött UploadedTotalCost mező értéke. Tartalma minden hozzáadott termék után automatikusan frissül.

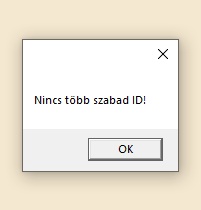
- Egy ComboBoc objektum, ami a fogyasztás típusának kíválasztására szolgál. Rákattintással egy legördülő menü nyílik meg, ahol a kívánt típus megnyomásával választhatunk:



- Egy hozzáadás gomb, aminek megnyomásával adhatjuk hozzá a rendelt termékek listájába a DataGrid-ben aktív kijelölt elemet. Kijelölés hiányában a gomb nem funkcionális.

- Egy felvétel gomb, aminek megnyomásával feltölthetjük a rendelés adatait az adatbázis “upload\_orders” táblájába. Az elküldött elemek a következők:

- A rendelés azonosítója: Ez lesz az adatbázis UploadedOrderId mezőjének értéke. Automatikusan generált véletlenszerű szám egy 1-től 999-ig terjedő skálán. Ez az érték a gomb megnyomásakor jön létre, de feltöltés előtt annyiszor futtat le egy ellenőrzést az űadatbázisban már szereplő rendelés azonosítók között, illetve egy érték újragenerálást amíg nem talál egy szabad értéket, amihez még nem tartozik rendelés. Így egyszerre maximum 999 rendelés lehet az adatbázisban. Amennyiben az összes generálható érték foglalt (999 rendelés található az adatbázisban), abban az esetben megszakítja a feltöltési folyamatot a program és egy MessageBox-ban értesíti a felhasználót a hibáról:



- A rendelt elemek listáját, aminek tartalma lesz az adatbázis UploadItems mezőjének értéke.

- A fogyasztás típusát amit a ComboBox kiválaszott eleméből választ ki, ez lesz az UploadedTakeway mező értéke az adatbázisban.

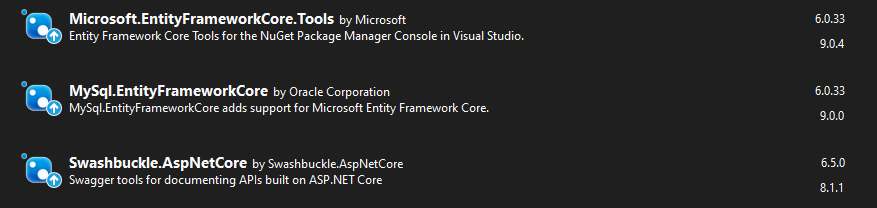
- A fizetendő végösszeget, ez lesz az UploadedTotalCost mező értéke az adatbázisban.

-A MÉGSEM gomb lenullázza az összes beviteli mezőt és bezárja az ablakot, ezzel megszakítva az új rendelés felvételének folyamatát.

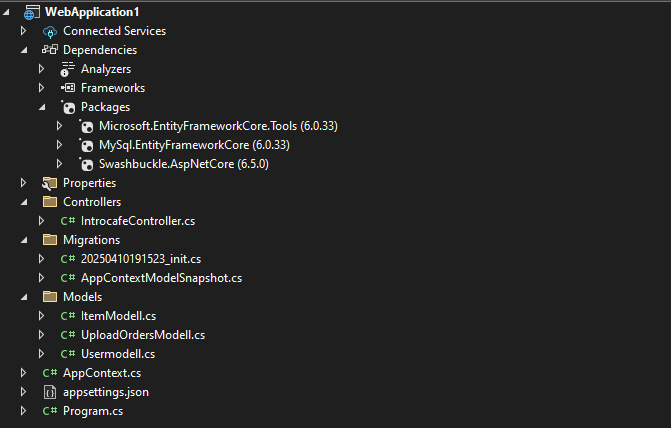
**Web API (szerveroldal)**

**6.1.: A Web API leírása:**

A projekt ezen részének feladata, hogy lehetővé tegye a kávézó weboldala és az adatbázis közötti kommunikációt. A Web API a Microsoft Visual Studio 2022 programban lett megírva. Létrehozásához a .NET keretrendszer 6.0-ás verzióját használtuk. A projekt létrehozása után elengedhetetlen az alábbi NuGet csomagok letöltése:



A projekt fájljai és azoknak kapcsolatai az alábbi képen láthatók:



**Program.cs:**

A kapcsolat létrehozása a feladata, illetve ezen belül állítjuk be a Swagger grafikus környezet feladatát, illetve a hozzáférési engedélyeket Cors engedélyek segítségévél.

**appsettings.json:**

Alapértelmezetten létrehozott fájl, egyedüli módosítás benne, az alapértelmezett adatbázis kapcsolat deklarálása.

**AppContext.cs:**

Feladata az adatbázis beállítása az általunk megadott tábla modellek, illetve az adatbázis elérési útvonalának megadásával.

**Modells mappa:**

Az adatbázisban szereplő táblák, illetve azok adattípusait hozzuk létre benne, hogy adatmegosztáskor a megfelelő értékek kerüljenek megfelelő típussal feltöltésre. A mappában található három C# osztály, a táblákat reprezentálja:

- ItemModell.cs = items tábla

- UploadOrdersModell = upload\_orders tábla

- Usermodell = users tábla

Ezeken a fájlokon belül adjuk meg az oszlopok alapértelmezett értékeit is, illetve az elsődleges kulcsokat.

**IntrocafeController.cs:**

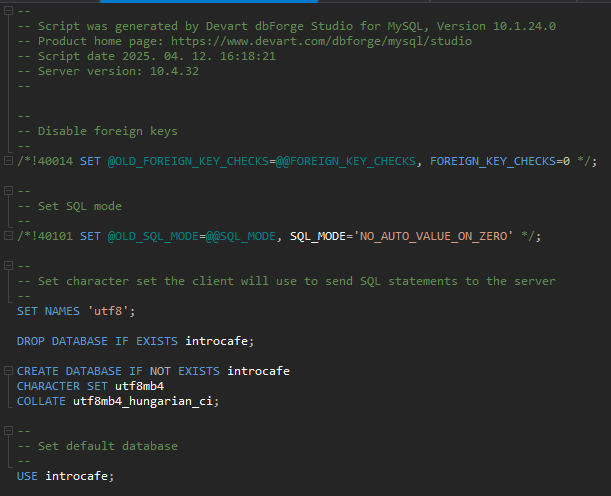
A müködéshez szükséges HttpRequesteket tartalmazza, ilyen például a lekérés, feltöltés, törlés metódus. Bármilyen kommunikáció az adatbázis és a weboldal között ezen keresztül történik. Függvényeket tárol, amik szükség esetén meghívhatóak.

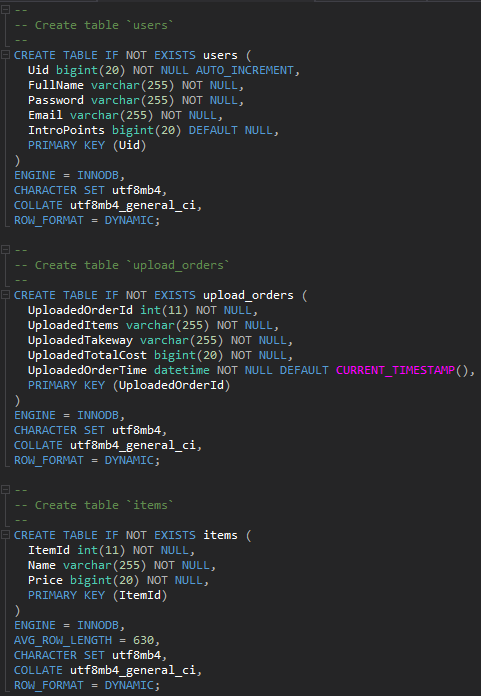
**Migrations mappa:**

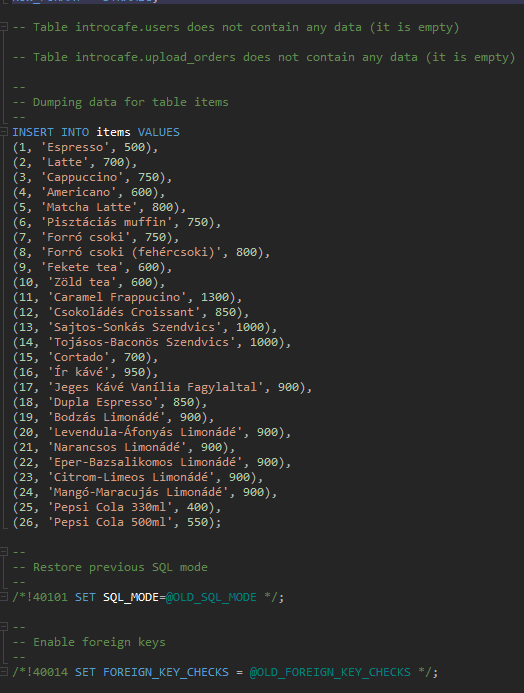
Az adatbázis létrehozásához szükséges fájlokat tartalmazza, további szerepe nincs.

**Kód mellékletek**

**6.1.: Adatbázis export fájlja:**

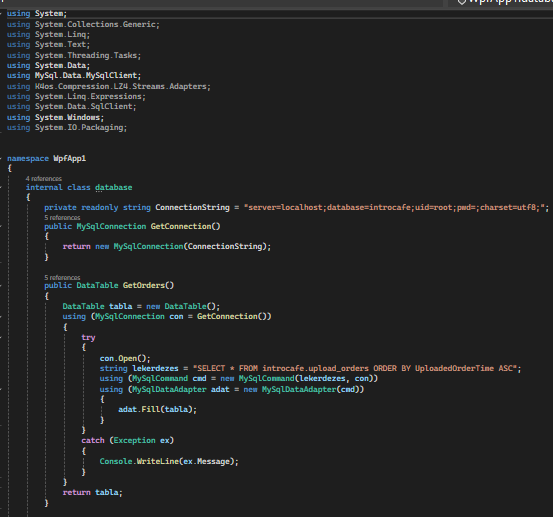


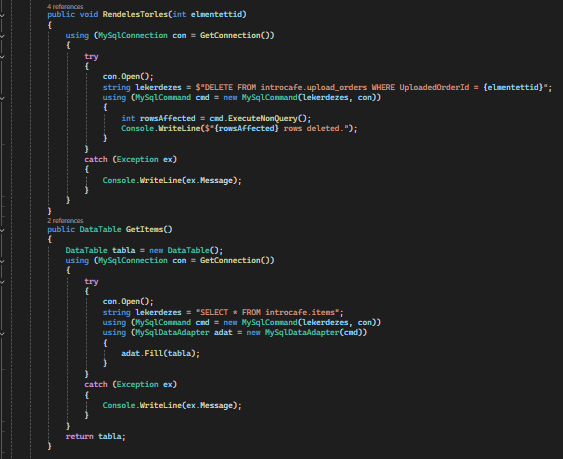


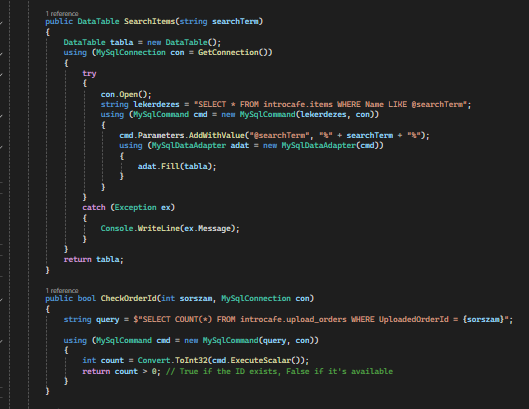


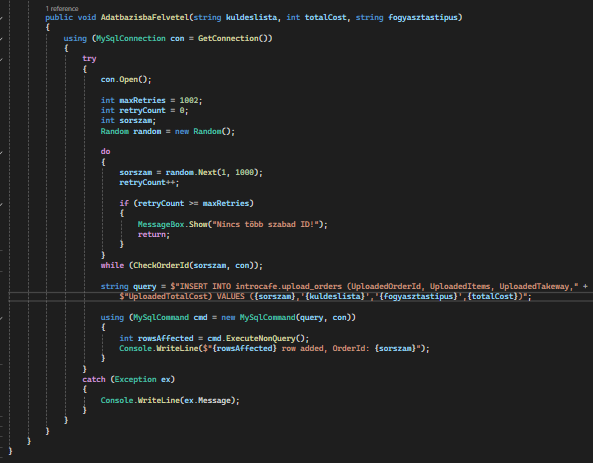
**6.2.: Asztali applikáció kódjai és osztályai:**

**database.cs:**

****

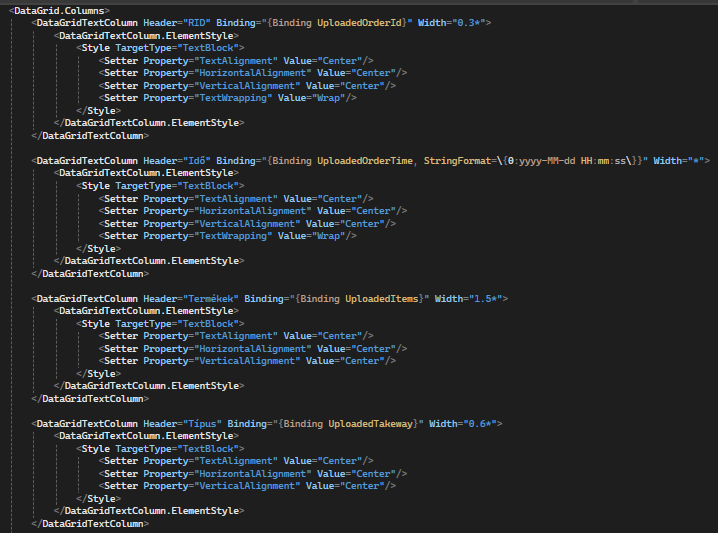
****

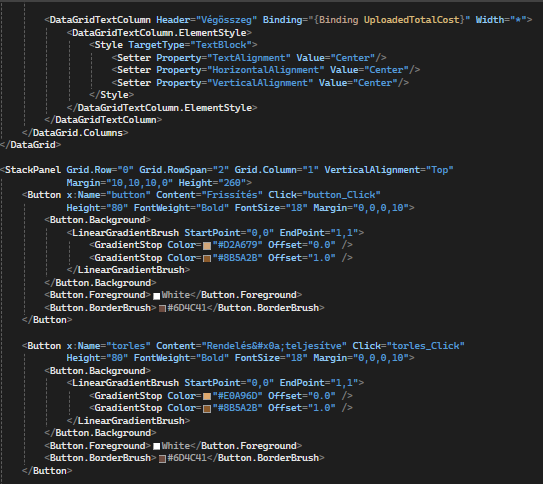
****

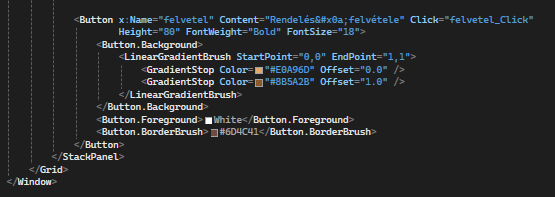
****

**MainWindow.xaml:**

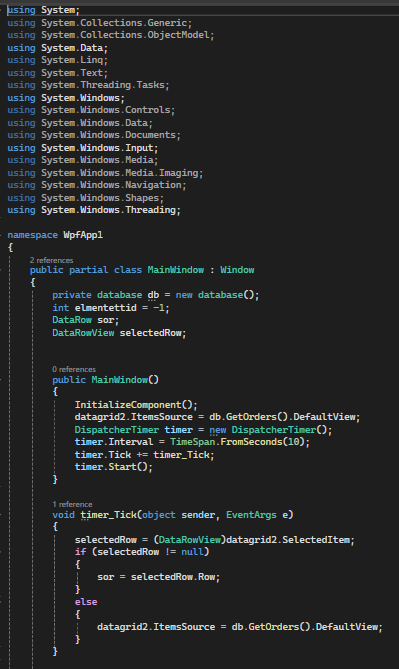
****

****

****

****

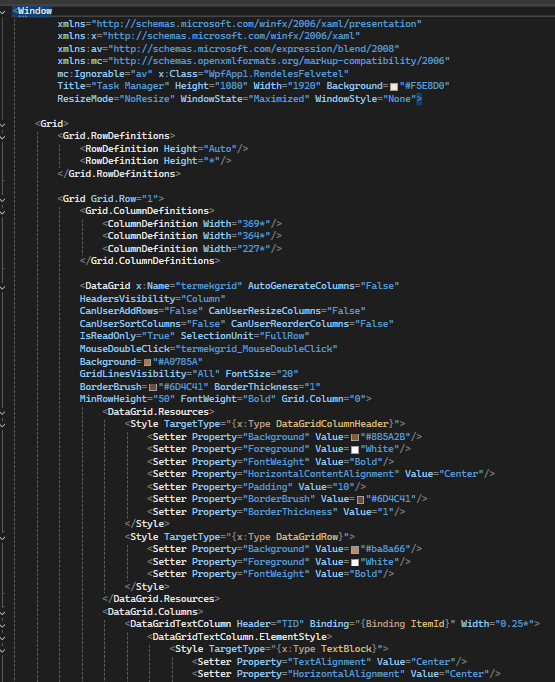
**MainWindow.xaml.cs:**

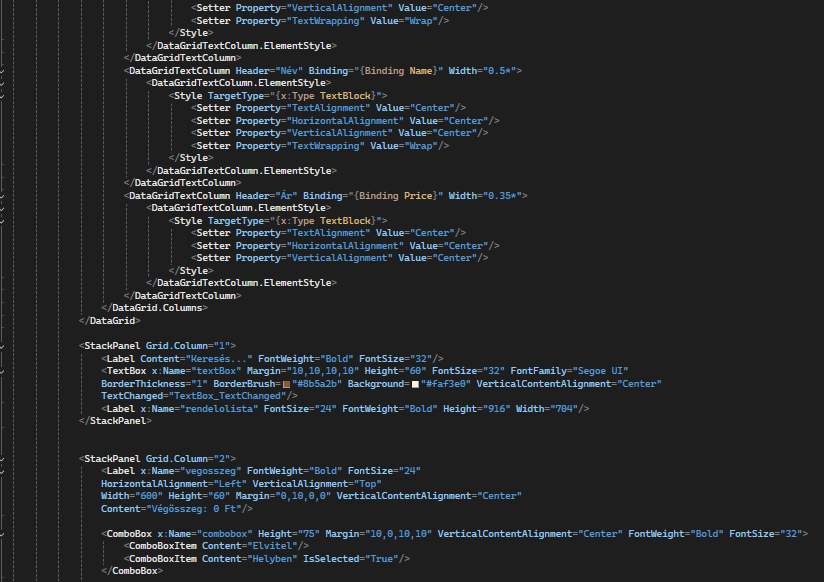
****

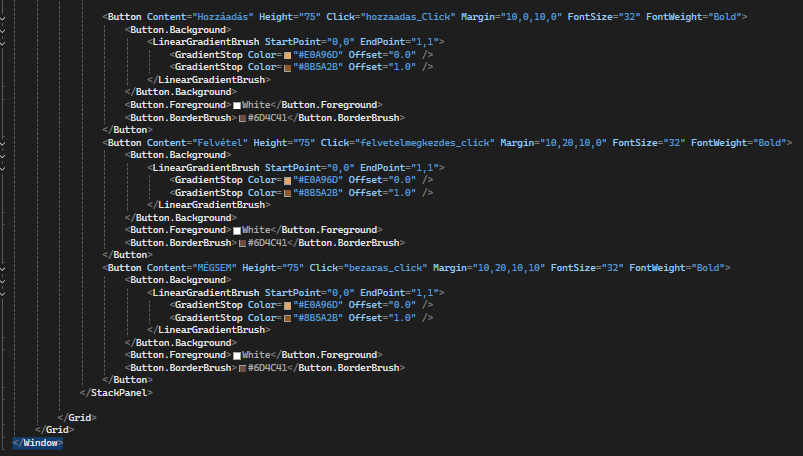
****

****

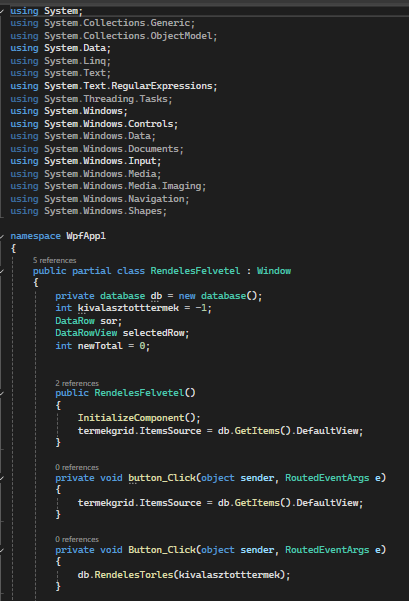
**RendelesFelvetel.xaml:**

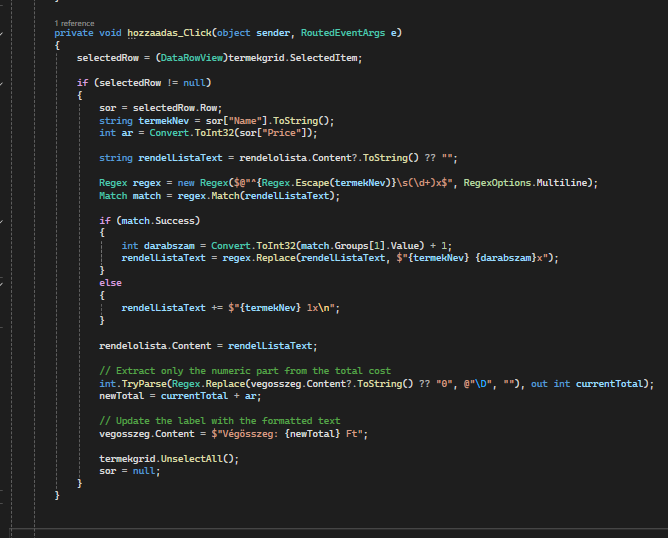
****

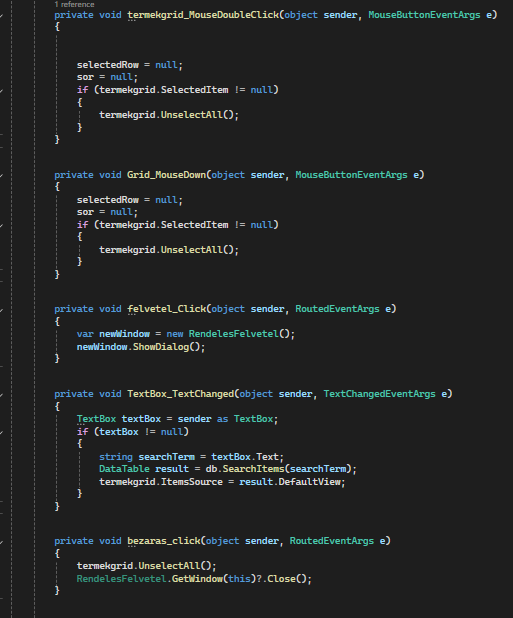
****

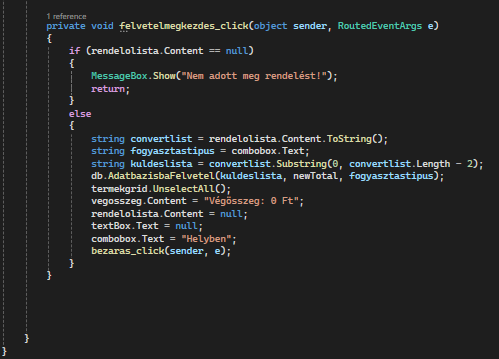
****

**RendelesFelvetel.xaml.cs:**

****

****

****

****