VMSZC Nádasdy Tamás Közgazdasági, Informatikai és Műszaki Szakgimnáziuma, Szakközépiskolája és Kollégiuma

*SZOFTVERFEJLESZTŐ*

**TeleTaby  
(Rendelés kezelő)**

**Konzulens:** **Készítette:**

Bartos Judit Nagy Sándor

# TARTALOMJEGYZÉK

Tartalomjegyzék

[TARTALOMJEGYZÉK 1](#_Toc36119745)

[BEVEZETÉS 3](#_Toc36119746)

[1 A RENDSZERREL SZEMBEN TÁMASZTOTT ELVÁRÁSOK 5](#_Toc36119747)

[1.1 Program célja 5](#_Toc36119748)

[1.2 Feladatai 5](#_Toc36119749)

[2 FEJLESZTÉSI TECHNOLÓGIÁK 6](#_Toc36119750)

[2.1 Microsoft Visual Studio 2019 Community 6](#_Toc36119751)

[2.2 C# 6](#_Toc36119752)

[2.3 Microsoft SQL Server Management Studio 18 7](#_Toc36119753)

[3 AZ ADATBÁZIS 8](#_Toc36119754)

[4 A RENDSZER AKTORAI ÉS JOGOSULTSÁGAIK 10](#_Toc36119755)

[4.1 Pultos 10](#_Toc36119756)

[4.2 Munkás 10](#_Toc36119757)

[4.3 Tulajdonos 10](#_Toc36119758)

[5 A RENDSZER FUNKCIÓI, KÉPERNYŐ KÉPEK 11](#_Toc36119759)

[5.1 Bejelentkezés 11](#_Toc36119760)

[5.2 Pultos 12](#_Toc36119761)

[5.3 Munkás 13](#_Toc36119762)

[5.4 Tulajdonos 15](#_Toc36119763)

[5.4.1 Felhasználók kezelése 15](#_Toc36119764)

[5.4.2 Termékek kezelése 16](#_Toc36119765)

[5.4.3 Rendelések 18](#_Toc36119766)

[5.4.4 Adatbázis 20](#_Toc36119767)

[6 TESZTELÉS 22](#_Toc36119768)

[6.1 Fehérdobozos tesztelés 22](#_Toc36119769)

[6.2 Feketedobozos tesztelés 22](#_Toc36119770)

[7 TOVÁBBI FEJLESZTÉSI LEHETŐSÉGEK 23](#_Toc36119771)

[ÖSSZEFOGLALÁS 25](#_Toc36119772)

# BEVEZETÉS

Nyári munka keretében kerültem a vendéglátásba, pontosabban egy fürdőn lévő pizzériához. Fontosnak tartom megjegyezni, hogy nem csak pizzákkal foglalkoztunk továbbá az ott készített termékek száma a többi fürdőn lévő vendéglátó létesítményhez képest elég magas volt, ugyan ez igaz volt a rendelések számára is. Első kézből tapasztaltam, hogy amit vendég ként természetesnek vettem az nem is olyan egyszerű, mégpedig, hogy mindenki a saját rendelését kapja meg. Nyilvánvaló hogy kisebb helyeken, mondjuk egy büfé esetében ez nagyon egyszerűen megvalósítható, de nagyobb léptekben ez már komoly problémákat okozhat. Képzeljük el azt a helyzetet, hogy folyamatosan kapunk rendeléseket körülbelül 25 különböző pizza közül, ráadásul több különböző pultból is érkezhetnek, egyetlen rendelés akár több terméket is magába foglalhat. Az éppen bent lévő rendelések száma könnyen elérheti a 15-20 rendelést is, amit nem éppen könnyű időrendben fejben tartani úgy, hogy semmiképpen se hibázzunk, ezt senkitől sem várhatjuk el. Továbbá ugyan ez a probléma fent áll a másik oldalon is, a pultosok oldalán, akik a rendelésekért felelnek. Egyértelmű, hogy erre a problémára valamilyen megoldást kell találni. Az első és legkézenfekvőbb ötlet az lehet, hogy kezdjünk el papír alapon, számon tartani mindent. Ebben az esetben viszont még fellép az a probléma, hogy a papírt A-ból B pontba el kell juttatni, ezáltal lassul az információ áramlásának gyorsasága és ez azt jelentené, hogy a pultosoknak el kellene hagyniuk a pultot rendszeresen, ami nem szerencsés. Igaz lehet, hogy ez a módszer bizonyos esetekben, kisebb rendelés számmal, kevesebb alkalmazottal működő kis helyen működhet, de nagyobb mértékekben ez nem megoldás. Itt jön képben az elektronikus megoldás, ami minden fenti problémára megoldást jelent. Fontos megjegyezni, hogy egy ilyen rendszer kiépítése komoly összegekben kerül, ugyan akkor tekintve a célt, befektetés ként kell rá gondolni annak érdekében hogy több rendelést lehessen kiszolgálni. Mint szoftveres és hardveres anyagi vonzata is van ennek a megoldásnak. A pizzériában erre a problémára egy programot használtunk, ami segítségével a munka gördülékenyen folyhatott anélkül, hogy a rendelések kezelésével különösebb probléma lett volna. Ott töltött időm alatt sokat gondolkoztam rajta, hogy vajon hogyan is működhet, ha engem kértek volna fel, mint leendőbeli programozót, én hogyan oldottam volna meg ezt az összetett problémát, képes lettem-e volna megoldani? Foglalkoztatott továbbá, hogy más hasonló helyeken milyen megoldást alkalmazhattak erre a problémakörre. Erre választ is kaptam egy nyári nyaralás során mikor is több, az én munkahelyemhez hasonló helyen étkeztem. Legtöbb hely sokkal kisebb forgalommal rendelkezett emiatt nekik nem volt olyan égető ez a probléma, a rendelések száma miatt megengedhették maguknak a fejben tartós megoldást. Ugyanakkor volt egy hely, amely nagy szerepet játszott a végső döntésemben. A korábban említetthez képest, itt sokkal nagyobb volt a forgalom emiatt itt a papíros módszert alkalmazták. Ugyanakkor ez nem egy olyan átlátható rendesz mint először gondolnánk. Ennek okán történt az, hogy egy egyszerű kavarodás miatt, indokolatlanul sokat kellett várnom, hogy rendelésem kézhez kapjam. Ekkor gondoltam úgy, hogy ez lesz a szakdolgozatom és megvalósítom a saját programom a problémakör megoldására. Hatalmas előny volt számomra a készítés során, hogy én napi szinten használtam az ottani programot ezáltal, mint felhasználó tudtam teljes pontossággal, hogy mi is a probléma és miként is lenne legcélszerűbb megoldani úgy, hogy a felhasználóknak is kedvező legyen. Mindezen okok miatt döntöttem úgy, hogy egy főként rendelések kezelését végző programot fogok készíteni mint szakdolgozat, minek neve TeleTaby.

# 1 A RENDSZERREL SZEMBEN TÁMASZTOTT ELVÁRÁSOK

## Program célja

A program legfőbb célja nagyon egyszerűen, tömören megfogalmazható, rendelések kezelése. Ugyan akkor ez első hallásra nem olyan egyértelmű, hogy mit is foglal magába. Az egész folyamat egy vendéggel és a pultossal kezdődik. A vendég leadja a rendelését amit a pultos rögzít egy egyszerű és átlátható menü segítségével és egy felhasználó barát felületen, majd fizetést követően véglegesít és ezáltal felvisz az adatbázisban. Ezt követően a leadott rendelést tartalmazó tételek meg kell, hogy jelenjenek azokon a helyeken, ahol elkészítik őket az éppen ott dolgozó alkalmazott. Miután elkészül egy termék, azt jelzik, ezáltal a pultos tudja hogy egy adott rendelésének egyik terméke kész és kiadható. Mikor egy rendelés minden tétele teljesítésre került, az adott rendelés lezárható. További fontos kikötés a programmal szemben, hogy statisztikai adatokat szolgáltasson, továbbá a felhasználók és a menü, könnyen, programozó beavatkozás nélkül történhessen meg.

## Feladatai

A program célját ismerve a következő lista készíthető a feladatairól:

* bejelentkezés kezelése
* rendelések összeállítása, leadása
* rendelések nyomon követesé
* tételek megjelenítése a megfelelő helyen
* tétel elkészültének jelzése
* rendelések lezárása
* felhasználók, menü változtatása
* statisztikai adatok képzése

Ezek lennének a program fő feladatai, a váza. Ezen feladatok több kisebb részre bonthatók és vannak olyan feldatok is amelyek a program céljának nem szerves részei mégis nélkülözhetetlenek

# FEJLESZTÉSI TECHNOLÓGIÁK

## Microsoft Visual Studio 2019 Community[[1]](#footnote-1)

A Microsoft Visual Studio egy integrált fejlesztői környezet(IDE) a Microsofttól. Számítógépes programok készítésére épp úgy alkalmas, mint weboldalak vagy mobil alkalmazások készítésére. A Microsoft szoftverfejlesztői felületeit használja pl. Windows API, Windows Forms és Microsoft Silverlight. Menedzselt és natív kódok létrehozására is egyaránt alkalmas. A Visual Studio tartalmaz a kód szerkesztési segítséget, ilyen például az IntelliSense ami kiegészíti a megkezdett parancsokat, szavakat a lehetőségeknek megfelelően. Továbbá kód refaktoring funkcióval is rendelkezik, melynek célja egy kódrészlet megváltoztatása annak érdekében, hogy az jobban értelmezhetőbb, átláthatóbb legyen és mindezt anélkül, hogy a kód funkcionalitása megváltozna. Egyéb beépített eszközök közé tartozik még például a kód profájler, tervező felület grafikus felhasználó felület(GUI) létrehozásához, web és adatbázis schema tervező is. Pluginok alkalmazását is lehetővé teszi, hogy fokozza a lehetőségeket. A Visual Studio 36 különböző programozási nyelvet támogat. Beépített nyelvek közé tartoznak például: C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, és a CSS.

## C#[[2]](#footnote-2)

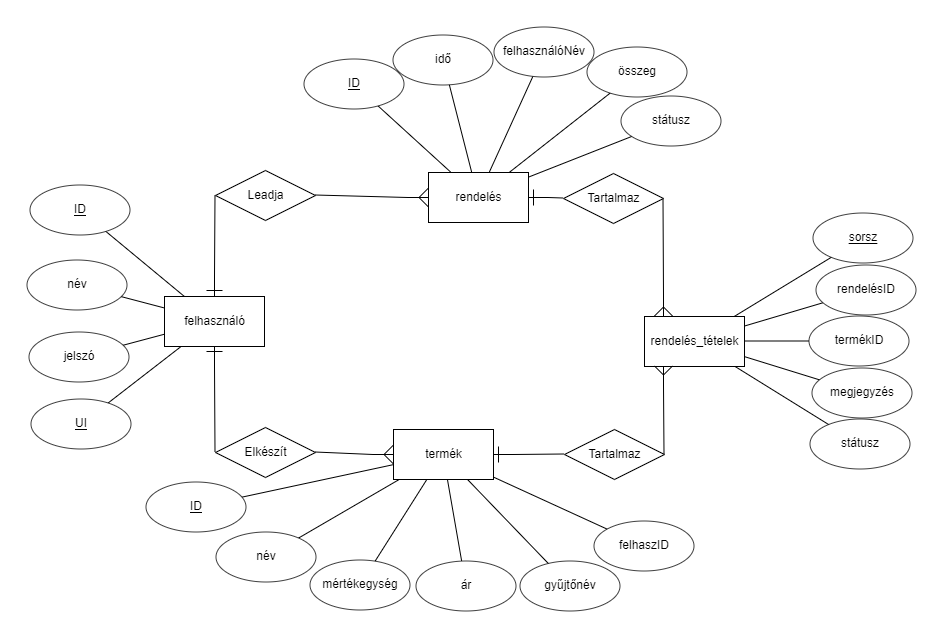
A C# a Microsoft által 2000-ben kifejlesztett objektumorientált programozási nyelv mely a .NET keretrendszerhez és a C nyelvcsaládhoz tartozik. Fejlesztés során példa ként szolgáltak a C++ és a Java nyelvek. Megjelenése óta nagy sikerre tett szert ennek oka lehet, hogy bár régeben leginkább csak asztali alkalmazások fejlesztésére használták, napjainkban például az ASP segítségével webes fejlesztésre is alkalmas. A C# egy erősen típusos, egyszerű és modern.

## Microsoft SQL Server Management Studio 18[[3]](#footnote-3)

SQL Server Management Studio (SSMS) egy integrált környezet SQL infratruktúra kezeléséhez. Széles körben elterjed és használt eszköz az adatbázis fejlesztők körében is. SSMS eszközöket biztosít SQL Serverek és adatbázisok konfigurálásához, monitorozásához. Lehetővé teszi adatbázisok létrehozását, szerkesztését és lekérdezések végrehajtását attól függetlenül, hogy a helyi számítógépeden vannak vagy a felhőben.

# 3 AZ ADATBÁZIS

Az adatbázis helyes megtervezése kulcsfontosságú egy program tervezéseinek folyamatai közül, mondhatni talán a legfontosabb lépés. Ennek alapján fog működni a programunk egésze, és ha nem végzünk ennél a lépésnél elég alapos és körültekintő munkát, későbbiekben komolya bonyodalmakat okozhat. A program funkcióit erre alapozva, az adatbázis tábláira és köztük lévő kapcsolatokra tudjuk alapozni és ezekhez tervezni. Amennyiben valamire nem gondolunk vagy elsiklunk valami felett ebben a fontos fázisban, nagyon sok órás időveszteséggel kell számolnunk a későbbiekben. Egy hibás vagy nem eléggé részletes adatbázis miatt akár a program számos funkcióját is újra kell terveznünk vagy akár az egész adatbázist az alapjairól .Ezért volt fontos, hogy nagyon alaposan járjak el ennél a szakasznál és mindenre gondoljak, ami a későbbiekben kellhet. Ugyanakkor még így sem bizonyultam eléggé körültekintőnek. A fejlesztés későbbi fázisaiban kétszer is szembesültem az eredeti adatbázis hiányosságaival, amelyeket pótolnom kellett. Ezzel elég sok időt veszítettem, de megtanultam, hogy az adatbázis megtervezésére nem lehet túl sok időt fordítani. Az én adatbázisom négy táblából áll. Az egyik tábla a ’felhasználó’ nevet kapta. Ebben tárolom a program felhasználóinak adatait, nevét, jelszavát hozzá tartozó felhasználói felület azonosítóját továbbá egy egyedi azonosítót. A következő táblám a ’rendelések’ tábla. Ez tartalmazza a leadott rendeléseket. Minden rendeléshez tartozik egy időpont mikor leadták, egy falhasználónév aki leadta, a rendelés összege és egy státusz, ami megmutatja, hogy már elkészült-e az adott rendelés, továbbá természetesen minden rendelés egyedi azonosítóval rendelkezik. A rendeléseken túl még a rendelésekben foglalt tételeket is számon kellett tartanom ezért egy ’rendelés\_tételek’ nevű táblát is létrehoztam. Ebben találhatóak az egyes rendelésekhez tartozó tételek és azoknak az adatai. Tárolom itt annak a rendelésnek az azonosítóját melyhez a tétel tartozik, a tétel azonosítóját, a tételhez esetlegesen tartozó megjegyzést, egy sorszámot továbbá egy státuszt, ami mutatja, hogy elkészült-e az adott tétel. Végül pedig még egy táblát létre kellett hoznom, amely a megrendelhető termékeket tartalmazza, ezt ’termék’-nek neveztem el. Minden terméknek van egy egyedi azonosítója, neve, ára, gyűjtőneve. Továbbá egyes termékeknek van mértékegysége és ’tartozása’ is ami azt mutatja meg melyik felhasználó felelős az elkészítéséért



1. ábra ERD

.

# 4 A RENDSZER AKTORAI ÉS JOGOSULTSÁGAIK

A rendszer három különböző típusú aktorral dolgozik. Minden aktornak egy jól, a többitől elkülönített feladata és felhasználói felülete van. A három típus a pultos, munkás és az adminisztrátor.

## 4.1 Pultos

A pultos felhasználók rendelkeznek a rendelések felvételéhez szükséges felülettel és ezáltal a ehhez szükséges jogokkal is. Csakis ilyen felhasználóval rendelkező alkalmazott képesek rendeléseket leadni, kezelni és lezárni.

## 4.2 Munkás

A munkások felelnek a rendelésekben leadott tételek elkészítéséért és ennek jelzéséért is. Minden ilyen felhasználóhoz vannak hozzá rendelt tételek melyek elkészítéséért ő felelős és ő is jogosult rá.

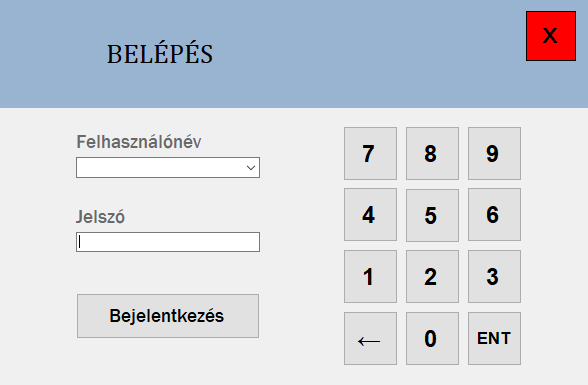
## 4.3 Tulajdonos

A tulajdonos felhasználónak van jogosultsága, lehetősége arra, hogy felhasználókat, termékeket és valamilyen szinten az adatbázist kezelje.

# 5 A RENDSZER FUNKCIÓI, KÉPERNYŐ KÉPEK

## 5.1 Bejelentkezés

A program elindítását követően a következő felület fogad, ami a bejelentkeztetésért felelős. Sikeres bejelentkezést kövezően jelennek meg a további felhasználói felületek. Először a felhasználót kell kiválasztanunk, majd mikor a jelszó mezőbe kattintunk, megjelennek a kép jobb oldalán lévő gombok a jelszó beírásához.

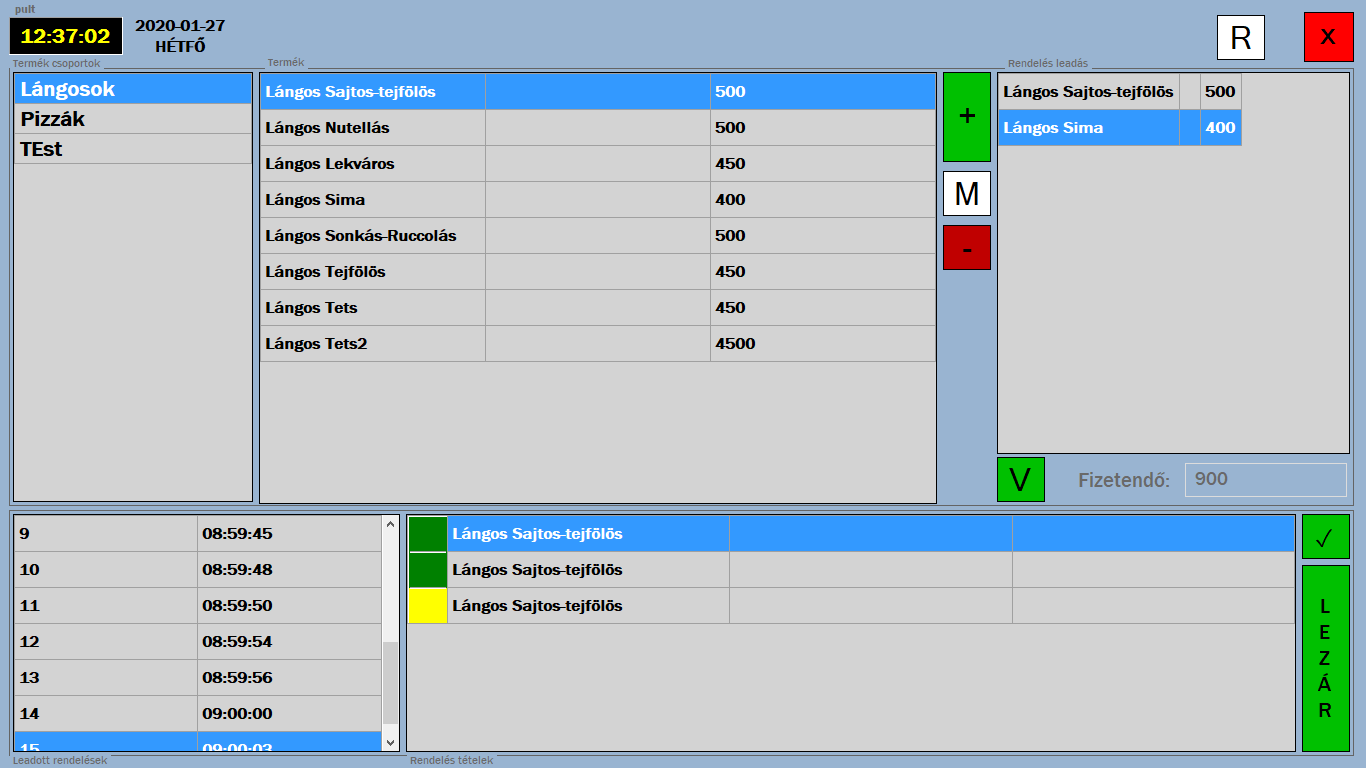


2. ábra Belépés

* Fekete: A belépéshez szükséges vezérlők láthatóak. Az első, ahova a felhasználó nevet kell megadni, az egy lenyíló menü ahonnan kiválaszthatjuk azt a felhasználó nevet, amellyel be szeretnénk jelentkezni. Ebbe a menüben indításkor kerülnek feltöltésre az adatbázisban szereplő felhasználónevek. Ezt követően egy szöveges dobozba adhatjuk meg a jelszót. A bejelentkezés gombra kattintva, vagy Enter billentyű lenyomásával pedig bejelentkezhetünk. Amennyiben helyes felhasználónév és jelszó párost adtunk meg, a felhasználóhoz tartozó kezelő felület jelenik meg számunkra. Rossz adatok megadása esetében vagy meg nem adása esetében hiba üzenetet kapunk.
* Kék: 12 gomb összessége melyek egy numerikus billentyűzetre hasonlítanak, annak érdekében, hogy a jelszót meg tudjuk adni. Erre azért van szükség, mivel a munkaállomásokon nincs hely egér és billentyűzet elhelyezésére ezért érintőképernyőt kell használni. 10 számot látunk, amik segítségével megadhatjuk a jelszót továbbá egy vissza nyilat, ami segítségével törölhetünk és ’ENT’ gombot amivel bejelentkezhetünk.
* Piros: A programból való kilépéshez szükséges bezáró gomb.

## 5.2 Pultos

Az alábbi felhasználói felület a pultosok felülete. Ez felelős a rendelések felvételéért és nyomon követeséért. Mindezeket a lehető legegyszerűbb és legátláthatóbb módon teszi lehetővé.

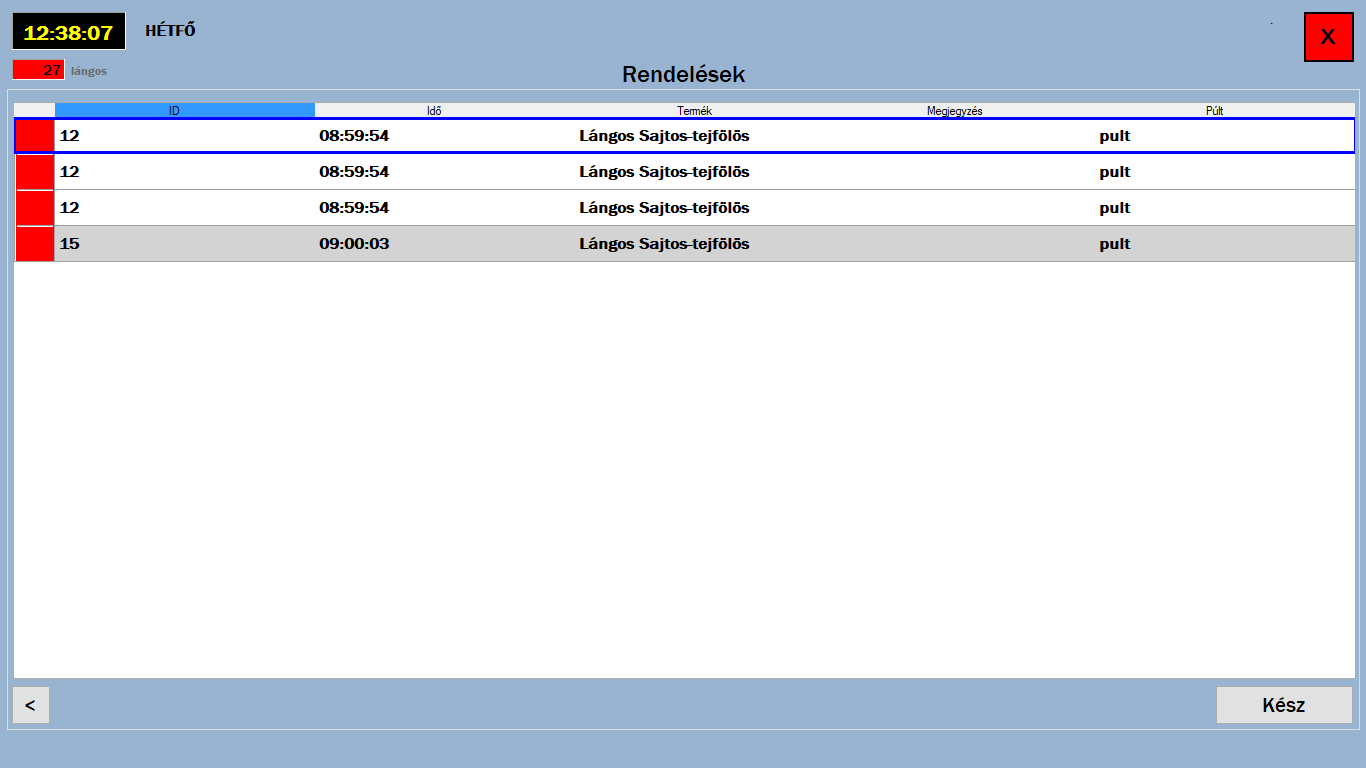


3. ábra Pult

* Fekete: Egy rendelés összeállításához és leadásához szükséges elemek láthatóak itt. Az első táblázat tartalmazza, a termék csoportokat melyek megkönnyítik egy adott termék megtalálását a lehető leggyorsabban. A második táblázatban találhatóak az éppen kiválasztott csoporthoz tartozó termékek, azoknak a neve, mértékegysége és ára. Harmadik a rendelés táblázat, ami azokat a termékeket tartalmazza, amelyeket kiválasztottunk és hozzáadtunk annak érdekében, hogy rendelés formájában leadjuk. Négy gomb látható, a ’+’ gomb segítségével adhatjuk hozzá az éppen táblázatban kiválasztott terméket a rendelés táblázathoz. A ’-’ gomb teszi lehetővé, hogy a rendelés táblázatból eltávolítsunk elemeket. A ’M’ gomb segítségével pedig úgy tudunk terméket a rendeléshez adni mintha a ’+’ gombot használnánk azzal a különbséggel, hogy itt mielőtt a rendeléshez hozzá adódna a termék, egy megjegyzést fűzhetünk hozzá. A rendelés táblázat alatt láthatjuk, hogy a táblázathoz adott termékekért mennyit kell fizetni és miután a fizetés megtörtént a ’V’ gombbal véglegesíthetjük a rendelést, azaz leadjuk.
* Zöld: A bal felső sarokban láthatjuk, hogy éppen milyen felhasználóval vagyunk bejelentkezve. Az aktuális időt, dátumot és napot.
* Sárga: A bal oldali táblázatban az általunk leadott rendeléseket láthatjuk, azok azonosítóját és leadás időpontját. A jobb oldali táblázat jeleníti meg az bal oldalon éppen kiválasztott rendeléshez tartozó termékek adatait. Láthatjuk a nevét, a hozzá tartozó megjegyzést illetve a mértékegységét. Továbbá minden termék előtt egy zöld vagy sárga színű négyzet látható. A zöld szín jelzi, hogy az adott termék elkészült a sárga pedig azt, hogy a termék készítése még folyamatban van. A ’ ✓’ gomb teszi lehetővé a pultos számára, hogy az olyan termékek elkészültét jelezze, amelyek elkészítéséért ő maga felelős. Egy rendelésen belül egy kiválasztott terméken tudja ezt használni, amennyiben az a termék hozzá tartozik. A ’LEZÁR’ gomb feladata, hogy az elkészült rendeléseket lezárja. Abban az esetben zárható le az éppen kiválasztott rendelés ha minden magában foglalt termék már elkészült.
* Kék: A ’R’ lenyomásával egy email küldésére lesz lehetőségünk egy a config fájlban megadott email címre amelyet a főnök felügyel. Ennek a funkciónak a célja, hogy jelezni tudjuk a lehető legegyszerűbben ha valami elfogyott, elromlott vagy probléma kérés van.
* Piros: A programból való kilépéshez szükséges bezáró gomb.

## 5.3 Munkás

Az alábbi a munka állomások felülete. Minden állomás egy ilyennel rendelkezik, de mindegyikük csak a hozzájuk tartozó termékeket látja benne, minden hozzájuk szükséges adattal.



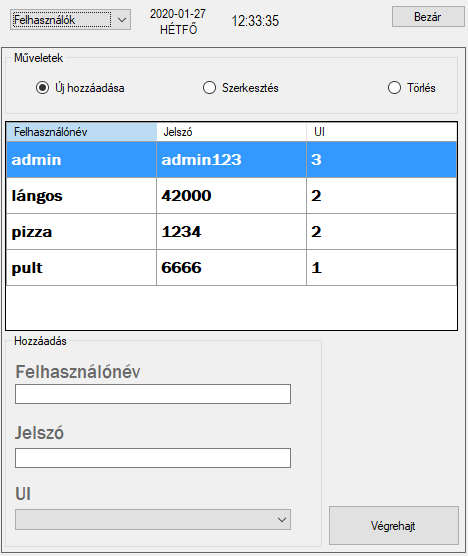
4. ábra Munkás

* Fekete: Táblázatba foglalva láthatjuk a rendelésként beérkező tételetek. Látható, hogy melyik rendeléshez tartoznak, vagyis a rendelés azonosítója, a leadás időpontja, a tétel neve, hozzá tartozó megjegyzést illetve, hogy melyik pultból érkezett be. Továbbá minden sor előtt megjelenik egy színes négyzet aminek három színe lehet. Fehér ami azt jelenti hogy a rendelés leadása óta nem tellett el több mint 15 perc. Sárga ami szerint a leadás óta több mint 15, de kevesebb mint 30 perc tellett el és piros ami azt mutatja, hogy az eltelt idő több mint 30 perc. Ez a négyzet nagyon szemléletessé teszi hogy éppen mennyire vagyunk elmaradva a rendelésekkel.  
  A sorok háttér színe rendelésenként megkülönböztetésre kerül. Fehér és világos szürke színek váltakoznak úgy, hogy azonos rendeléshez tartozó tételeket jelző sorok színe azonos, egészen addig amíg egy új rendelés tételei nem következnek, azok új színnel láthatóak.  
   A bal alsó sarokban a ’<’ gomb segítségével váltani tudunk a táblázatban megjelenített adatok között. Két opció van, vagy az éppen elkészítésre váró tételek láthatóak vagy a már elkészült rendeléseket nézhetjük meg, abban az esetben ha valamilyen keveredés lenne.  
  Jobb alsó sarokban a ’Kész’ gomb segítségével jelezhetjük, hogy a táblázatban éppen kijelölt(kék kerettel rendelkező) sorban szereplő tétel elkészítésre került. A gomb megnyomását követően, a tétel eltűnik a táblázatból és kész ként lesz jelezve a pultos számára.
* Zöld: A bal felső sarokban láthatjuk, hogy éppen milyen felhasználóval vagyunk bejelentkezve. Az aktuális időt és napot, továbbá egy számlálót, ami az aznap, az adott felhasználó által elkészített tételek számát mutatja.
* Piros: A programból való kilépéshez szükséges bezáró gomb.

## 5.4 Tulajdonos

Ezt a felület a tulajdonos számára készült annak érdekében, hogy a program fentartható és üzemképes legyen. Statisztikai adatok képzése is a feladatai közé tartozik. Mivel több feladatot lát el, ezért a korábbiakhoz képest aprólékosabb és bővebb.

### 5.4.1 Felhasználók kezelése



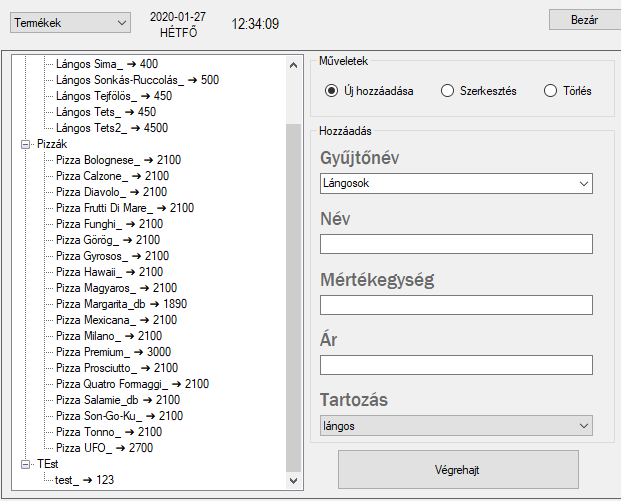
5. ábra Felhasználók kezelése

A fent látható felület felelős a program felhasználóinak kezelésére, szerkesztésére, törlésére vagy új felhasználó létrehozására.

* Fekete: Az adatbázisban szereplő felhasználók adatai láthatóak táblázatba foglalva. Mindegyikük rendelkezik egy felhasználónévvel, jelszóval és egy UI azonosítóval. Az UI azonosító felelős azért, hogy bejelentkezést követően a felhasználó számára melyik felhasználói felület jelenjen meg a már korábban említettek közül.
* Sárga: A felhasználókkal végezhető műveletek láthatóak itt. Három művelet végrehajtása lehetséges. Új felhasználó felvétele, egy már meglévő felhasználó adatainak módosítása vagy törlése. Mindig az éppen kiválasztott művelet érvényes és hajtódig végre.
* Lila: A bal felső sarokban egy szöveget olvashatunk, ami mindig az aktuális kiválasztott műveletnek felel meg. Alatta szövegdobozok láthatóak melyekben a felhasználók adataik adhatók meg vagy éppen módosíthatók. Az UI azonosító kiválasztása nem maga az azonosító szám megadása által történik, hanem egy lenyíló menü segítségével, ami a lehetséges opciókat tartalmazza(pl.pultos).  
  Jobb alsó sarokban található ’Végrehajt’ gomb megnyomásával hajhatjuk végre az éppen kiválasztott műveletet. Szerkesztés és törlés esetében a kiválasztott terméken hajtódig végre, új létrehozása esetén pedig a megadott adatokon.
* Zöld: A bal felső sarokban láthatjuk az aktuális dátumot és időt. Továbbá lenyíló menü segítségével navigálhatunk a tulajdonos felületen belül.
* Piros: A programból való kilépéshez szükséges bezáró gomb.

### 5.4.2 Termékek kezelése

Termékek hozzáadásáért, módosításáért és törléséért felelős felület. A termékeket egy fastruktúrába rendezve láthatjuk az átláthatósága és könnyű kezelés érdekében. A struktúrában csoport nevek(pl.levesek) alattuk pedig a hozzájuk tartozó termékek láthatóak(pl. húsleves).

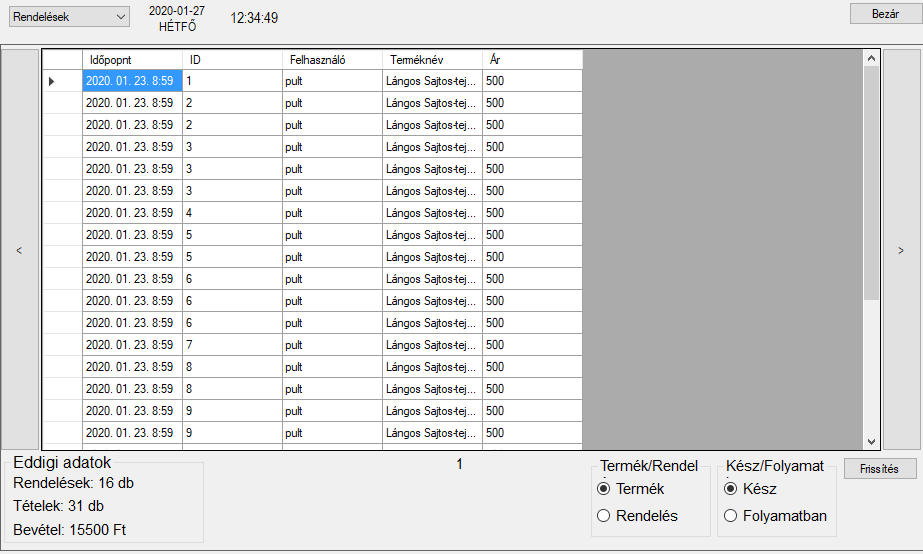


6. ábra Termékek kezelése

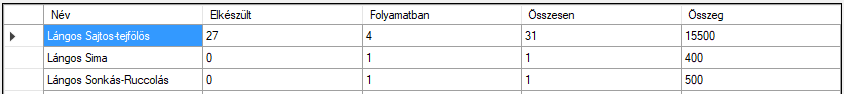
* Fekete: A fastruktúrába rendezett termékek láthatóak. A csoportokhoz tartozó termékek a csoportok neve alatt, beljebb szedve helyezkednek el, itt nevük és áruk látható. Minden csoportnév előtt látható egy ’-’ jel, ami segítségével az adott csoport termékeit elrejthetjük, ezáltat az egész rendszer jobban átlátható és kezelhető.
* Lila: A bal felső sarokban egy szöveget olvashatunk, ami mindig az aktuális kiválasztott műveletnek felel meg. Alatta szövegdobozok láthatóak melyekbe a termékek adataik adhatók meg. Minden termék rendelkezik gyűjtőnévvel, névvel és árral, tartozás és mértékegység tulajdonsággal nem feltétlen. A gyűjtőnevük azt a csoportot jelenti, amelybe a termék tartozik(pl.húsleves->levesek).  
  Tartozás tulajdonság jelzi, hogy a termék melyik felhasználóhoz tartozik, kinek a feladata az elkészítés. Azért nem muszáj ezt megadni, mivel több pultos esetében ha a pultosok termékeit(pl.kávé) egy felhasználóhoz csatolnánk, az a többi pultos számára nem jelenne meg. Ezt a tulajdonságot egy lenyíló menüből válaszhatjuk ki, ami a felhasználókat tartalmazza.  
  Mértékegység megadására azért van lehetőség, mert vannak olyan termékek, amelyek léteznek különböző kivitelben(pl.Fanta üdítő->330ml,1,5l), emiatt van rá szükség, hogy meg tudjuk őket különböztetni még ha nevük azonos is.  
  Lent pedig a ’Végrehajt’ gomb megnyomásával hajhatjuk végre az éppen kiválasztott műveletet. Szerkesztés és törlés esetében a kiválasztott terméken hajtódig végre, új létrehozása esetén pedig a megadott adatokon.
* Sárga: A termékekkel végezhető műveletek láthatóak itt. Három művelet végrehajtása lehetséges. Új termék felvétele, egy már meglévő termék adatainak módosítása vagy törlése. Mindig az éppen kiválasztott művelet érvényes és hajtódig végre.
* Zöld: A bal felső sarokban láthatjuk az aktuális dátumot és időt. Továbbá lenyíló menü segítségével navigálhatunk a tulajdonos felületen belül.
* Piros: A programból való kilépéshez szükséges bezáró gomb.

### 5.4.3 Rendelések

A tulajdonos számára biztosít lehetőséget az adatbázisba való bepillantásra, a rendelések megtekintésére. Több módon képez az adatokból jól átlátható és értelmezhető információt ez a felület. Tökéletes arra, hogy statisztikát képezzen és, hogy átfogó képet alkosson a forgalomról, fogyasztásról és bevételről.



7. ábra Statisztika

* Fekete: Középen a felület legfontosabb eleme, a táblázat látható, amely tartalmazza az adatbázisból éppen lekért adatokat rendezve, csoportosítva az egyszerűség és átláthatóság érdekében. A táblázat két oldalán egy-egy gomb látható melyek feladata a navigáció a különböző lekérdezések közöl. A ’>’ gomb segítségével a következő adatokat mutatja meg a táblázatban, ez egészen addig működik, míg végig nem érünk a lehetséges opciókon, ugyanis ekkor vissza ugrik az első lehetőségre. A ’<’ gomb ugyan ezt a lehetőséget biztosítja azzal a különbséggel, hogy ez mindig az előzőt mutatja egészen addig, míg az elsőhöz nem ér, mert az után az utolsóra ugrik. Négy lehetséges opció van amik között váltogathatunk.  
  A táblázat alatt középen egy számot láthatunk, ami egytől négyig vehet fel értékeket. Ez, mint egy oldalszám segít tájékozódni mikor a fent már említett gombokkal navigálunk az adatok lekérdezése között.  
  Az első eset az, ami a képen is látható. Itt a legáltalánosabb adatok tekinthetőek meg, az adott napi rendelések és tételek. A táblázat alatti választógombok segítségével dönthetjük el, hogy a rendeléseket vagy a tételeket szeretnénk megtekinteni, majd miután ezt elöntöttük, választhatunk, hogy a már kész vagy még folyamatban lévőkre vagyunk kíváncsiak. Ez az opció csak ebben az első esetben áll fent.  
  

8. ábra Termék statisztika

* Második esetben a fent látható kép szerint látjuk az aznap már legalább egyszer rendelt tételeket és hozzájuk tartozó információkat. Itt egyszerűen áttekinthető, hogy egyes termékekere mekkora a kereslet, mekkora bevétel származik belőlük és az arra is következtethetünk, hogy az elkészítéshez szükséges hozzávalók milyen gyorsan fogynak.  
  

9. ábra Pultos statisztika

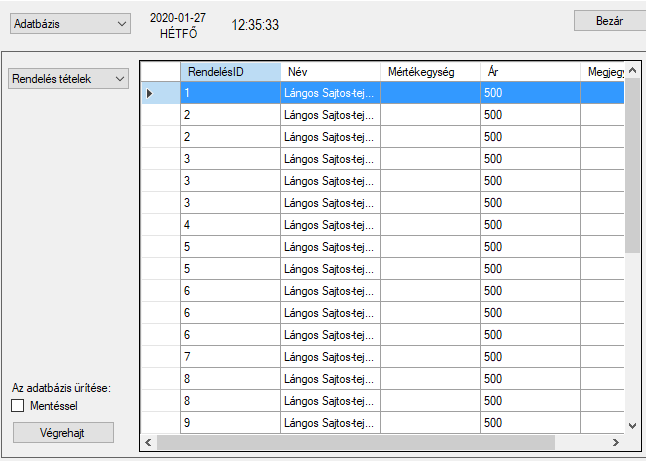
* A harmadik esetben láthatóak a pultos felhasználók adataik. A táblázatban jól észrevehető, hogy melyik pultoknál mekkora a forgalom. Megjelenik, hogy az adott felhasználónak hány rendelése van éppen folyamatban, eddig a nap során mennyit készített el majd ezek összessége. továbbá a teljes bevétele is. Több pultos esetében mindezen információk nagyon hasznosak lehetnek, könnyen kiderülhet, ha valamely pultra nincs is nagy szükség vagy akár az is ha egy újra van szükségünk a rendelések felvételének meggyorsításának érdekében. Továbbá napvégén elszámoláskor hasznos látni, hogy melyik pult mekkora bevételre tett szert, mennyivel kell elszámolniuk.  
  

10. ábra Munkás statisztika

* A negyedik eset a harmadikhoz nagyon hasonló. Az a különbség a kettő között, hogy itt nem a pultosok, hanem a tételek elkészítéséért felelős felhasználókról láthatjuk ugyanazon adatokat. Hasznos információ a napok végén látni hogy melyik munkaállommás mennyi tétellel készül el, mekkora bevételt termelnek. Ezek tudatában el tudjuk dönteni, hogy egy adott munkaállomásra több vagy éppen kevesebb alkalmazott szükséges annak érdekében hogy a munka gördülékenyen történjen. Továbbá nap közben a folyamatban lévő termékek számából következtethetünk arra, hogy mennyire elfoglaltak, leterheltek az alkalmazottak.
* Lila: A bal alsó sarokban a leglényegesebb adatokat láthatjuk, amit az adatbázisból kaphatunk. Rendelések darabszáma, ami mutatja, hogy addig a pillanatig, mikor nézzük ezt a felületet, mennyi rendelés érkezet be. A tételek száma a rendelésekben foglalt tételek darabszámát mutatja, ami szintén nagyon hasznos. Végül pedig a tételek árának összege látható, a bevétel. Fontos tudni, hogy az előző adatok leképzése közben nem tesz különbséget a program elkészült és még el nem készült tételek és rendelések között, mindent beleszámol. Ezen adatok adják vissza legjobban azt, hogy az adott pillanatig mi történt.
* Sárga: Jobb alsó sarokban egy ’Frissítés’ gomb található, melynek feladata az éppen táblában látható adatok újbóli lekérdezése az adatbázisból, ezáltal aktuálisabb adatokat kapunk.
* Zöld: A bal felső sarokban láthatjuk az aktuális dátumot és időt. Továbbá lenyíló menü segítségével navigálhatunk a tulajdonos felületen belül.
* Piros: A programból való kilépéshez szükséges bezáró gomb.

### 5.4.4 Adatbázis

Az alábbi felület legfőbb célja az adatbázis ürítése minden munkanap végén annak érdekében, hogy másnap ne zavarjanak be az előző nap rendelései a statisztikákba. Ugyanakkor, ha esetleg egy munkahét adatait szeretnénk látni, az is megtehető, hogy csak akkor ürítjük ki, ezáltal nagyobb időintervallumot magába foglaló adatokat kapunk. A felület másik célja az archiválás, de a rendelések és tételek ömlesztett megtekintésére is alkalmas.



11. ábra Adatbázis ürítés

* Fekete: Bal fent látható elnyíló menüből válaszhatjuk ki, hogy a rendeléseket vagy a tételeket szeretnénk megtekinteni a táblázatban. Itt mindenféle szűrés, csoportosítás nélkül láthatjuk az adatokat majdnem ugyan, úgy ahogyan az adatbázisban is találhatóak.
* Sárga: ’Az adatbázis ürítése: ’ szöveg után egy jelölő négyzetet és egy gyombot találhatunk. A jelölő négyzet után a ’Mentéssel’ felirat látható. Amennyiben a jelölő négyzetet kipipáljuk, és úgy kattintunk a ’Végrehajt’ gombra, úgy a config fájlban megadott helyre, excel fájlba importálja a rendelésekkel és hozzájuk tartozó tételekkel kapcsolatos információkat majd üríti az adatbázis e részeit. Az excel fájlt az épen aktuális dátummal látja el mint név. Amennyiben nem pipáljuk ki a jelölő négyzetet és úgy kattintunk a gombra, akkor excel fájlba való mentés nélkül, töröli az adatokat az adatbázisból.

# 6 TESZTELÉS

A tesztelés a programkészítés egyik legfontosabb folyamata. Kódolás és tervezés közben sok hiba keletkezhet. A hibákat általánosságban két csoportra oszthatjuk, szemantikai és szintaktikai hibákra. Az utóbbira azonnal fény derül kódolás közben, hiszen a fejlesztői környezet felhívja rá a figyelmünk és egy ilyen hibával a program sem futtatható. A probléma forrása a szemantikai hibák, ilyen hiba esetében a program lefut csak nem az elvártaknak megfelelően működik. Az ilyen hibák miatt szükséges és elengedhetetlen a tesztelés folyamata. A program tesztelése során két számomra tetszőleges és szimpatikus módszert választottam melyekről úgy gondoltam, hogy elég hatékonysággal fogják a programban lévő hibákat kiszűrni. Ez a két módszer a feketedobozos és fehérdobozos tesztelés. Ez a két módszer nagyon hasonló, egyetlen dologban különböznek, de úgy hiszem, hogy nagyon jól kiegészítik egymást.

## 6.1 Fehérdobozos tesztelés

Ennek a tesztelési módszernek a lényege, hogy a forráskód ismeretében teszteljük a programot. Tisztában vagyunk vele mi hogyan és miért működik és ennek tekintetében próbálunk hiba esetek generálni. A programműködésének ismeretben könnyen találhatunk olyan részeket melyek hibásan vagy nem az elvártaknak megfelelően működnek. Ez a módszer egyértelműen egy szakembert igényel, valakit, aki teljes egészében ismeri a programot vagy a program azon részét, amelynek teszteléséért felelős. Az én esetemben ez a személy természetesen én voltam. Számtalan hibát találtam ezzel a módszerrel, hiszen tudtam a kódom gyengeségeit, esetleges hiányosságait.

## 6.2 Feketedobozos tesztelés

Az előzővel ellentétben ezt a fajta tesztelést olyan személyek végzik akik nem ismerik a forráskódot, olyanok akik a program felhasználói is lehetnének adott esetben akár. Ez a megközelítés is nagyon hasznos hiszen felhasználói szemmel is le lesz tesztelve a program. Egy felhasználó teljesen máshogy látja, gondolja a dolgokat mint a fejlesztő éppen ezért nagyon könnyen olyan hibákat találhat amelyre a fejlesztő talán sose gondolt volna. Új szemszögből tekint a programra és annak működésére, ezáltal nagyon könnyen tud hibákat találni a programunkban. Ezen tesztelés végrehajtásához leginkább barátaim segítségét kértem és nagyon hasznosnak bizonyult.

# 7 TOVÁBBI FEJLESZTÉSI LEHETŐSÉGEK

Számos módon és féle képpen lehetne tovább fejleszteni a programot. Sok funkció és lehetőség hozzáadásával még jobbá lehetne tenni hisz egy program sosincs kész, sohasem tökéletes. Mindig van egy jobb, gyorsabb, erőforrás hatékonyabb megoldás vagy egyszerűen olyan funkciók, amikre eddig még nem gondoltunk, ami segítené a felhasználót. Így van ez az én programomban is. A fejlesztés közben is sok olyan dolog jutott eszembe, ami biztosan hasznos funkció lenne és biztos vagyok benne, hogy az adatbázis lekérdezéseimet is lehetne jobban, gyorsabban csinálni.   
Az egyik funkció, amire gondoltam az a menük létrehozása. Lehetőség arra, hogy az adatbázisban szereplő tételeket, a termékeket, amiket felvittünk, menükbe lehessen foglalni. Menüknek nevet adni és árat. A menük ára pedig a bennük foglalt termékek árától független lenne hisz ez a lényegük, a spórolás. Biztos vagyok benne, hogy ez a funkció nagyon hasznos, sőt szükségszerű lenne, ha megrendelésre készíteném ezt a programot és nem szakdolgozatnak. Azért nem valósítottam meg ezt a funkciót, mert az adatbázisom nem lenne erre alkalmas jelen állapotában és még a grafikus felhasználó felületemen is változtatnom kellene. Tudom, hogy sose lenne kész és lennék vele teljesen megelégedve ezért itt meghúztam a határt és azt mondtam, hogy ezt már nem.  
További jó lehetőség lenne egy új felhasználói felület létrehozása a vendégek számára. Egy érintőképernyő és a nekik létrehozott felhasználói felület segítségével a vendégek számára is lehetővé tennénk a rendelések leadását. Ezzel felgyorsíthatnánk a rendelések felvételét illetve kevesebb alkalmazottra is lenne szükség. Ennek érdekében egy a többinél is egyszerűbb, átláthatóbb és felhasználóbarát felületet kellene létrehozni mely segítségével mindenféle nehézség, félreértés nélkül tudnának a vendégek is rendelést leadni. Miután végeztek rendelésükkel egy blokkot kapnának a rendelésük sorszámával és a tételekkel, amiket tartalmaz. Ezután a kasszához fáradva csak fel kell azt mutatni és fizetni, ezt követően pedig a pultos érvényesítné a rendelést és bekerülne az az adatbázisba. Ehhez egy teljesen új felület és meglévők változtatása lenne szükséges, ami szintén csak egy nagy feladat lenne. Továbbá ez magában foglal egy önmagában is szükséges funkciót, amire sajnos nem volt lehetőségem pedig szerettem volna. A blokkok nyomtatása és nem az általános mindenhol kapott blokkra gondolok, hanem egy sajátra.  
Az egész program szerves része lenne, hogy rendelés leadásakor a vendég egy blokkot kap a rendeléséről, ami a tételek és rendelése sorszámát tartalmazza, később pedig ezzel igazolhatja, hogy övé a termék mikor elkészült és veheti át. Sajnálom, hogy erre sajnos nem volt lehetőség amiatt, hogy nem áll rendelkezésemre az ehhez szükséges hardware.  
Továbbá egy hatalmas továbbfejlesztése lehetőség lenne, ha az egész programot webes alapokra helyeznénk. Manapság egyre több és több szolgáltatás érhető el online. Képek, videók szerkesztése, különböző fájl típusok szerkesztése, megnyitása. Ennek oka pedig a kényelem. Sokkal egyszerűbb mindent egy böngészőből elérni, mint minden feladatra külön-külön programot letölteni. Amennyiben webes alapokon működne a program, a gépeknek csak internet kapcsolatra lenne szüksége és egy közös azonosítóra. Ennek a megoldásnak is megvannak a hátrányai és lehet valamely esetben ez nem is lenne előnyös, de ezek ellenére is úgy gondolom, hogy általánosságban lehet ez a lehetőség előnyösebb lenne.

# ÖSSZEFOGLALÁS

Örülök, hogy egy valós életből vett problémát választottam, mint szakdolgozat téma. Ez számomra is egy kisebb fajta megerősítés, a tudat, hogy erre a problémára, amit mint programozó kaphattam volna, képes voltam megoldást találni és egy működő programot készíteni hozzá. Jó volt egy kicsit nagyobb terjedelmű projecten dolgozni és az ezzel járó nehézségeket megtapasztalni. Eddig csak kisebb, pár száz soros programokat kellett készítenem, kisebb órai feladatok megoldására. Egy teljesen új dolog volt megtapasztalni azt, hogy egyetlen probléma megoldásán hónapokon keresztül kell hosszú órákat dolgozni. Pusztán a terjedelme miatt sok egyszerű problémába ütköztem, amik folyamatos kis odafigyeléssel könnyen kiküszöbölhetőek lettek volna, de mostantól ezekre mind figyelni fogok a jövőben. A fejlesztés minden egyes fázisát végig kellett járnom mire elkészült a program, ezáltal annak egészére bepillantást nyertem, még ha csak alap szinten is. A project során sok kisebb nagyobb nehézségbe ütköztem, melyek során sok új dolog megtanulására voltam rákényszerítve annak érdekében, hogy haladni tudjak, de minden alkalommal, valamivel bővült a tudásom. Vissza tekintve sok olyan része van a programnak, amit már máshogyan csinálnék és a jövőben máshogyan is fogok. Egy pozitív élményként éltem meg a fejlesztés egészét a sok nehézség ellenére is.

1. Microsoft Visual Studio (2020.03.21) <https://en.wikipedia.org/wiki/Microsoft_Visual_Studio> [↑](#footnote-ref-1)
2. C# (2020.03.21) <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)> [↑](#footnote-ref-2)
3. Microsoft SQL Server Management Studio (2020.03.21) <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15> [↑](#footnote-ref-3)