**ZÁRÓDOLGOZAT**

**Walkowicz Szása**

**5/13SZOFT**

**BAJAI SZC TÜRR ISTVÁN TECHNIKUM**

**Szoftverfejlesztő**

ZÁRÓDOLGOZAT

Food app

**Walkowicz Szása**

2024

**NYILATKOZAT A ZÁRÓDOLGOZATRÓL**

Alulírott *...................Walkowicz Szása......................* (név) tanuló

**kijelentem, hogy**

*..................................FoodApp.........................* című záródolgozatomat (nyomtatott és elektronikus formában) a Bajai SZC Türr István Technikum pedagógusai és tanulói:

* + **felhasználhatják (pl. hivatkozás alapjául, olvasótermi használatra) későbbi munkájukhoz a szerzői jogok tiszteletben tartása mellett).**
  + nem használhatják fel (titoktartási nyilatkozat csatolása mellett).

Ugyanakkor kijelentem, hogy a záródolgozat ***saját munkám eredménye*.**

Baja, 2024. április 29.

………………………………….

Aláírás

Tartalom

[**1.** **Bevezetés** 1](#_Toc165377044)

[**1.1 A dolgozat célkitűzései** 1](#_Toc165377045)

[**2.** **Felhasználói dokumentáció** 1](#_Toc165377046)

[**2.1 Általános specifikáció** 1](#_Toc165377047)

[**2.1.1 Rendszerkövetelmény** 2](#_Toc165377048)

[**2.1.2 XAMPP rendszerkövetelmény** 2](#_Toc165377049)

[**2.1.3 FoodApp rendszerkövetelmény** 2](#_Toc165377050)

[**2.1.4 Használathoz szükséges környezet** 2](#_Toc165377051)

[**2.2 Használati leírás** 2](#_Toc165377052)

[**2.2.1 Bejelentkezés** 3](#_Toc165377053)

[**2.2.2 Rendelés felvétele** 4](#_Toc165377054)

[**2.2.3 Aktív rendelések** 5](#_Toc165377055)

[**2.2.4 Futár nézet** 7](#_Toc165377056)

[**2.2.5 Étel/Ital felvétel** 10](#_Toc165377057)

[**3. Fejlesztői dokumentáció** 13](#_Toc165377058)

[**3.1 Fejlesztői környezet** 13](#_Toc165377059)

[**3.2 MySQL** 13](#_Toc165377060)

[**3.3 XAMPP** 13](#_Toc165377061)

[**3.4 Visual Studio 2019** 13](#_Toc165377062)

[**3.5 Fejlesztési folyamat** 14](#_Toc165377063)

[**3.5.1 Követelmények meghatározása, tervezés** 14](#_Toc165377064)

[**4.** **Összegzés** 31](#_Toc165377065)

[**5.** **Felhasznált források** 33](#_Toc165377066)

[**6.** **Ábrajegyzék** 34](#_Toc165377067)

# **Bevezetés**

Az étteremipar folyamatos növekedése és az új éttermek gyors megjelenése kiemelten fontossá teszi egy könnyen hozzáférhető és átlátható szoftver létrehozását. Ez a szoftver, mely kis tárterületen is hatékonyan működik, kiemelkedően segíti a dolgozók munkáját, felgyorsítva az éttermi folyamatokat. A program létrehozásának alapvető célja, hogy ne kelljen minden alkalmazottól teljes körű szakértelmet elvárni, hiszen könnyen elsajátítható tudásra van szükség a hatékony működtetéshez. Ennek megfelelően, az efféle szoftverek fejlesztése és alkalmazása nem csupán kívánatos, hanem az éttermek hatékony és gazdaságos üzemeltetésének alapvető feltétele is.

A program megtalálható GitHubon az alábbi linken: <https://github.com/WalkowiczSzasa/FoodApp>

## **1.1 A dolgozat célkitűzései**

A záródolgozat célkitűzései négy fő pontba sorolhatók:

* Rendelések felvételének egyszerűsítése éttermi dolgozók számára
* Egyszerű étlap frissítés lehetősége az admin vagy az ezért felelős személy részére
* Minden szükséges információ megosztása a futárokkal a sikeres kiszállításhoz
* Egy könnyen kezelhető felület létrehozása éttermek számára

# **Felhasználói dokumentáció**

## **2.1 Általános specifikáció**

A FoodApp alkalmazást olyan éttermeknek készítettem, amelyek még újak a piacon és egy városon belül terjeszkednek, ezért még nincsen igényük nagy méretű adatbázisokra, amiknek a futtatásához szervergépek szükségesek, hanem akár egy irodai laptopon is futtatható alkalmazást szeretnének.

### **2.1.1 Rendszerkövetelmény**

Az alkalmazás futtatásához két program rendszerkövetelményét kell szem előtt tartanunk, ugyanis mindkettő szükséges ahhoz, hogy az alkalmazás működjön.

### **2.1.2 XAMPP rendszerkövetelmény**

* Operációs rendszer: Windows, Linux vagy macOS
* Processzor: 1 GHz-es vagy gyorsabb processzor
* RAM: 512 MB (1 GB ajánlott)
* Merevlemez-terület: 200 MB az XAMPP telepítéséhez
* További követelmények: PHP, MySQL, Apache szerverkomponensek

### **2.1.3 FoodApp rendszerkövetelmény**

* Operációs rendszer: Windows 7 vagy újabb
* Processzor: 1 GHz-es vagy gyorsabb processzor
* RAM: 32 bites rendszerhez 1 GB, 64 bites rendszerhez 2 GB
* Merevlemez-terület: Akár 10 GB szabad terület, az alkalmazás méretétől és az adattárolási igényektől függően
* Grafika: DirectX 9 vagy újabb verzió, WDDM 1.0 illesztőprogram
* Kijelző: Legalább 1920 x 1080 képpont felbontás

### **2.1.4 Használathoz szükséges környezet**

Szükséges egy számítógép Windows operációs rendszerrel.

## **2.2 Használati leírás**

Mielőtt még elkezdnénk használni az alkalmazást telepítenünk kell az XAMPP-ot az Apache és MySQL komponensekkel. Miután ezt megtettük el kell indítanunk mindkét komponenst az XAMPP-on belül és meg kell nyitnunk a MySQL Admin felületét az Admin gombbal. Itt a felhasználói fiókok menüpontban létre kell hoznunk egy új felhasználót asd névvel, asd jelszóval és az összes engedéllyel. Miután ez megvan létre kell hoznunk egy restaurantapp nevű adatbázist, amibe importálnunk kell a githubon a database mappában lévő restaurantapp.sql fájlt. Végül pedig létre kell hoznunk egy felhasználót akár admin, szakács vagy futár jogkörrel és már használhatjuk is az alkalmazást vagy beléphetünk a már létező admin felhasználóba, aminek a felhasználóneve „admin”, a jelszava pedig „a”.

### **A képen kerék, clipart, jármű, Szárazföldi jármű látható Automatikusan generált leírás2.2.1 Bejelentkezés**

ábra 1 Bejelentkezési felület

A felhasználónév és jelszó mezőkbe egy érvényes felhasználónév jelszó pár beírása után kattintsunk a Bejelentkezés gombra, ha minden rendben volt, akkor a rendszer beenged a fiókunktól függően vagy a Rendelés felvétel vagy pedig a Futár nézet oldalra, ha nem volt helyes a felhasználónév jelszó kombináció, akkor az alábbi hibaüzenetet kapjuk:

 Ebben az esetben vagy a felhasználónevünket vagy pedig a jelszavunkat elírtuk.

ábra 2 Felhasználónév/ jelszó hiba

### **A képen szöveg, képernyőkép, diagram, Betűtípus látható Automatikusan generált leírás2.2.2 Rendelés felvétele**

**\*1**

ábra 3 Rendelés felvétel

\*1 mező: Ebben a mezőben lesznek azok a vásárlók, akik már legalább egyszer rendeltek az étteremből, ha innen választunk egy opciót az automatikusan kitölti a mezőket.

A piros négyzeten belüli helyen lesznek találhatóak az ételek és az italok amiket hozzá lehet adni a rendeléshez.

A képen szöveg, képernyőkép, képernyő, Betűtípus látható

Automatikusan generált leírásMiután megadtuk a rendelés adatait, amik a következők: A vásárló telefonszáma, a vásárló neve, a kiszállítási cím utcája és házszáma (ha elviteles a rendelés, akkor csak be kell pipálni az elvitel melletti fehér négyzetet), az hogy mikorra kell a rendelés, megjegyzés a rendeléshez (opcionális) már csak tételeket kell adnunk hozzá a tételek kis paneljén lévő hozzáad gombbal. Ha látni szeretnénk, hogy eddig milyen tételeket adtunk a rendeléshez csak rá kell kattintanunk a kis kék négyzettel bekeretezett ikonra és a rendelés tételei megjelennek az alatta lévő fehér téglalapban.

ábra 4 Rendelés felvétel tételek

Ha többet szeretnénk hozzáadni egy tételből a rendeléshez, akkor csak rá kell kattintanunk a sorára és meg kell nyomnunk „+” jelet, ha kevesebbet szeretnénk belőle, akkor pedig a „-” jelet kell megnyomnunk.

Ha a rendelés bármely nem opcionális mezőjét üresen hagyjuk akkor az üresen hagyott mezőre releváns hibát kapunk:

****

Ha nem kaptunk hibát és kiürültek a mezők, akkor minden rendben volt és megtörtént a rendelés felvétel. Továbbléphetünk az Aktív rendelések fülre.

### **A képen szöveg, képernyőkép, tervezés látható Automatikusan generált leírás2.2.3 Aktív rendelések**

ábra 5 Aktív rendelések

Az Aktív Rendelések oldalon tudjuk szerkeszteni és törölni a készülő rendeléseket valamint az állapotukat „Készül”-ről át tudjuk állítani „Kész”-re és ezt követően át tudjuk adni őket egy futárnak.

A szerkesztéshez először rá kell kattintanunk a szerkeszteni kívánt rendelés ceruza ikonjára majd a rendelés tételeinek betöltéséhez a kékkel bekeretezett ikonra.

A képen szöveg, képernyőkép, tervezés látható

Automatikusan generált leírás

ábra 6 Aktív Rendelések tétel szerkesztés

A Rendelés szerkesztése fülön tudjuk megnézni a rendelés tételeit és módosítani őket.

A piros téglalapon belüli részen a Kiszállítási díjat tudjuk változtatni, de ez az összes az átállítást követő rendelésnek megváltoztatja a Kiszállítási díját, nem csak az aktuálisaknak.

Ha a rendelés adatait szeretnént módosítani mint példaul telefonszám, vagy kiszállítási cím, akkor a ceruza ikonra kattintás után a lila négyzetben lévő Adatok módosítása gombra kell kattintsunk.

Ezen a fülön az adatok módosítása mellett a rendelés törlésére is módunk van.

A képen szöveg, képernyőkép, tervezés látható

Automatikusan generált leírás

ábra 7 Aktív Rendelések adatok módosítása

Miután elkészült a rendelés ki-pipálhatjuk és ezzel az állapota „Készül”-ről „Kész”-re vált, ezzel feloldva a futárválasztás lehetőségét.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásFutárt a sárga téglalappal bekarikázott legördülő dobozban tudunk. A program a futár kiválasztásakor automatikusan lementi a választásunkat, így ezért külön nem kell újra rákattintsunk a Mentés gombra.

ábra 8 Aktív rendelés

### **2.2.4 Futár nézet**

Ez a nézet elérhető az összes szerepkör számára, de csak bizonyos feltételekkel, amik a következők: az admin szerepkörű felhasználók az összes rendelést látják ami „Kész” állapotra van állítva és hozzá van rendelve egy futárhoz, a konyhai személyzet csak az „ELVITEL” futárra állított rendeléseket látják a felületen, míg a futárok csak azokat a A képen szöveg, képernyőkép, tervezés látható

Automatikusan generált leírásrendeléseket látják, amelyeket átadtak nekik.

ábra 9 Futár nézet

Miután ráklikkeltünk a Futár nézet gombra ez a látvány tárul elénk. Semmi ok a pánikra, ha nem látunk rendeléseket, mivel először rá kell kattintanunk a piros négyzettel bekeretezett ikonra, hogy betöltsük őket.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

ábra 10 Futár nézet 2

A sárga négyzetekben lévő lenyíló dobozzal találunk módot arra, hogy egy egyszerűsített verzióját tekintsük meg a rendelés tételeinek.

Ha már kiszállítottuk a rendelést és el szeretnénk tűntetni a rendszerből vagy csak meg szeretnénk tekinteni a számla előnézetét akkor a megtekintendő rendelés jobb oldalán található nagyító ikonra kattintás után a számla ikonra kattintással azt meg is tehetjük.

Ekkor ez a látvány tárul elénk:

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás **2**

**1**

ábra 11 Futár nézet Számla

Innentől már csak annyi a teendőnk, hogy rá klikkelünk a „Kész” feliratú zöld gombra a „Számla” feliratú gomb mellett és már le is zártuk a rendelést véglegesen.

### **A képen szöveg, képernyőkép, diagram, Betűtípus látható Automatikusan generált leírás2.2.5 Étel/Ital felvétel**

ábra 12 Étel/Ital elem módosítás

Ezen a felületen tudnak az admin szerepkörrel rendelkező felhasználók felvenni ételeket, italokat és allergéneket, valamint módosítani a már meglévőket és a csomagolási árat.

A csomagolási ár módosításához csak át kell írni a „Csomagolás:” mellett szereplő értéket és rá kell kattintani a mentés gombra.

Ha ételt, italt vagy allergént szeretnénk módosítani rá kell kattintanunk a megfelelő fülre az oldal tetején az „Étel/Ital felvétel” felirat alatt és a szerkeszteni kívánt elemhez tartozó „Módosít” gombot kell megnyomnunk. Ezután, ha elindulunk a jobb oldali panel felé a kurzorral az ablak jobb oldalán lévő mezők a kiválasztott elem adataival feltöltődnek a következőképp: A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

ábra 13 Étel/ital elem módosítás 2

Ezeket az adatokat szabadon módosíthatjuk és miután végeztünk csak rá kell kattintanunk a jobb alsó sarokban lévő „Módosít” gomra és már módosultak is az elem adatai.

**A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás**

ábra 14 Új elem felvétele

Új elem felvételéhez csak ki kell választanunk a kívánt kategóriát és be kell írnunk az üres mezőkbe az adatokat majd végül pedig rá kell kattintanunk az „Elem felvétele” gombra.

# **3. Fejlesztői dokumentáció**

## **3.1 Fejlesztői környezet**

Ebben a fejezetben fogom ismertetni a felhasznált programok elméleti hátterét, amivel megindokolom, hogy miért pont ezekre az eszközökre esett a választásom.

## **3.2 MySQL[[1]](#footnote-1)**

MySQL egy nyílt forráskódú relációs adatbázis-kezelő rendszer, melyet gyakran használnak webes és asztali alkalmazások háttértárolójaként. Képes hatékonyan kezelni nagy adatmennyiségeket, könnyen integrálható más szoftverekkel és támogatja a szabványos SQL (Structured Query Language) nyelvet. A MySQL gyors és megbízható adatbázis-kezelési megoldást biztosít különböző célú alkalmazásokhoz.

## **3.3 XAMPP[[2]](#footnote-2)**

XAMPP egy ingyenes, nyílt forráskódú szoftvercsomag, amely egyszerűvé teszi a lokális szerverkörnyezet létrehozását. Tartalmazza az Apache HTTP szervert, a MySQL adatbázis-kezelőt, a PHP-t és más szükséges eszközöket, így ideális fejlesztési környezetet biztosít webalkalmazások fejlesztéséhez és teszteléséhez. A fejlesztés során az Apache-ot és a MySQL-t használtam.

## **3.4 Visual Studio 2019[[3]](#footnote-3)**

Azért választottam a Visual Studio 2019-et, mert erős, integrált fejlesztői környezetet kínál, ami lehetővé teszi a hatékony projektkezelést és kódolást. Rugalmasan támogatja a különböző programozási nyelveket és platformokat, és rendszeresen frissül a legújabb technológiákhoz.

## **3.5 Fejlesztési folyamat**

A harmadik fejezet tartalmazza a fejlesztési folyamat lépéseit és a folyamat során összegyűjtött észrevételeimet, valamint a fejlesztés során felmerülő hibákat és azok megoldásának módját.

### **3.5.1 Követelmények meghatározása, tervezés**

Követelményként az alábbi célokat tűztem ki a program fejlesztése során:

* Fontos, hogy az alkalmazás internetkapcsolat nélkül is zavartalanul működjön, hogy mindig rendelkezésre álljon a felhasználók számára.
* Az alkalmazás használatának könnyen elsajátíthatónak kell lennie, hogy ne okozzon nehézségeket a felhasználónak a betanulás.
* Fontos szempont, hogy az alkalmazás minimális erőforrásigényekkel rendelkezzen, hogy bármilyen eszközön hatékonyan és gyorsan fusson.
* Az alkalmazásnak gyorsan és könnyen elindíthatónak kell lennie, hogy azonnal használható legyen a felhasználók számára.
* Természetesen az alkalmazásnak meg kell felelnie a szükséges rendszerkövetelményeknek és elvárásoknak a hatékony és biztonságos működés érdekében.

A projekt tervezését négy fő pontba bontottam, amelyek az alábbiak:

**Bejelentkezési felület**

A bejelentkezési folyamat megvalósítása során a felhasználónév/jelszó hitelesítési módszert terveztem alkalmazni. Regisztrációs lehetőség nem lesz elérhető az alkalmazásban, azonban az adminisztrátor jogosult lesz új felhasználók létrehozására. Az adminisztrátor feladata továbbá meghatározni az új felhasználók beosztását, mint például futár vagy konyhai személyzet. Minden új felhasználónak egyedi felhasználónévvel és megjelenített névvel kell rendelkeznie, hogy az azonos nevek ne vezessenek félreértésekhez.

**Felhasználói jogkörök**

Az adminisztrátor jogosultságai:

* Módosíthat allergéneket, új allergéneket vehet fel, és törölheti azokat.
* Létrehozhat, törölhet és módosíthat felhasználókat, beleértve a felhasználók jelszavát, felhasználónevét és megjelenített nevét az adatbázis felületén.
* Módosíthatja az ételek árait, létrehozhat új ételeket, valamint módosíthatja a meglévő ételeket.
* Nyomon követheti a rendelések állapotát más felületeken, például a futár és a konyha felületén.
* Rendeléseket vehet fel, állíthatja azok állapotát, és törölheti azokat.

Konyhai személyzet jogosultságai:

* Rendeléseket vehet fel, azok állapotát módosíthatja, valamint a kész rendeléseket átadhatja egy futárnak.
* Látja az elviteles rendeléseket

Futárok jogosultságai:

* Csak azokat a rendeléseket látja, amelyek „Kész” állapotúak és átadták neki.
* Megtekintheti az elérhetőségeket, az úticélt, a végösszeget és a tételek árát, valamint a fizetési mód típusát (kártyás vagy készpénzes).
* Kiszállított állapotba helyezheti a rendelést, ezzel eltüntetve azt az alkalmazás felületéről.

**Rendelések felépítése és kezelése**

Csak az adminisztrátor és a konyhás jogosultak a rendelések létrehozására és szerkesztésére. A rendelések a következő információkat tartalmazzák:

* Azt, hogy a rendelés kiszállításra vagy elvitelre készül
* Telefonszámot
* Szállítási címet (Utca, házszám)
* Jegyzetet a rendeléshez
* Csomagolási díjat
* Szállítási díjat
* Fizetési módot
* Elkészítési határidő
* A rendelés felvételének idejét

A készülő rendeléseket lehet szerkeszteni. A rendelések állapota a felvétel után „Készül” állapotba kerül, majd egy gomb megnyomására a kiválasztott rendelések állapota „Kész”-re vált. A „Kész” rendeléseket ki lehet választani és át lehet adni az egyik futárnak. Végül a futár/adminisztrátor/konyhás ki tudja pipálni a rendeléseket, ezzel eltüntetve őket a felületről. Az adatbázisból csak az adminisztrátor tudja az adatokat törölni manuálisan.

A szállítási cím mezőt a rendszer elmenti, és legördülő combobox-ban lehet kiválasztani a már elmentett személyeket. Telefonszám ütközés esetén a mentett értékek módosulnak az újonnan felvett értékekre.

A csomagolási díj tételenként egy bizonyos összeget (pl.: 200,-Ft) jelent, amit az adminisztrátor tud módosítani.

**Ételek, italok és allergének felvétele és kezelése**

Az allergének, italok és ételek felvételét kizárólag az adminisztrátori fiókban lehet elvégezni egy különálló felületen. Ezek szerkeszthetőek és törölhetőek az adminisztrátori felületen.

* Ételek felvétele:

Az ételek rögzítése során szükséges információk: Megnevezés, Áfakulcs, Ár és allergének, valamint elérhetőség.

* Italok felvétele:

Az italok regisztrálásakor szükséges adatok: Megnevezés, Ár és elérhetőség.

* Allergének felvétele:

Az allergének hozzáadásánál csak a Megnevezés adatot kell megadni.

**Adatbázistervek**

Az alábbiakban fogom részletezni a programban használt adatbázis felépítését és a benne lévő táblák, mezők jelentését, tartalmát és felhasználását.

Az adatbázis relációs modellen alapul, és különböző táblákból áll, amelyek között kapcsolatok vannak definiálva. A táblák szervezése optimalizálva van az adatok hatékony tárolása és lekérdezése érdekében.

**allergene:**

* „allergeneID” int(11): Az adott allergén egyedi azonosítója, automatikusan növekvő szám. Azért hoztam létre, hogy megkönnyítse az adott allergénekre való hivatkozást a programban.
* „allergeneName” varchar(255): 255 karakter hosszúságú szöveg. Az allegén neve.

**customer:**

* „ID” int(11): A vásárlók egyedi azonosítója, automatikusan növekvő szám. Ezzel a számmal lehet más táblákból hivatkozni a vásárlókra, valamint ez alapján lehet szerkeszteni az adott vásárlók adatait.
* „name” varchar(40): Ez a mező tartalmazza a vásárló nevét.
* „phoneNumber” varchar(30): A vásárló telefonszámát tartalmazó mező.

**deliveryfee:**

Ez az egyik egy értékkel rendelkező tábla az adatbázisban, amit azért hoztam létre, hogy a kiszállítási díjat lehessen a grafikus felületen szerkeszteni.

* „ID” int(11): Erre az értékre hivatkozik a program amikor az admin szerkeszti kiszállítási díjat.
* „price” double: A kiszállítási díj.

**destination:**

* „ID” int(11): Erre az értékre hivatkozik a program új rendelés felvételénél az orders táblában orderDestID néven, automatikusan növekvő érték, a tábla elsődleges kulcsa.
* „orderType” tinyint(1): Logikai mező, ide kerül beírásra a rendelés típus az alábbi módon: ha az érték 0, akkor a rendelés elviteles lesz, ha az érték 1, akkor kiszállításos. A kettő között az a különbség, hogy az elviteles rendelésért a megrendelő eljön és nem kerül kiszállítási díj felszámolásra, a kiszállításos rendelést viszont az étterem egyik futára házhoz viszi és kiszállítási díj kerül felszámolásra.
* „orderStreet” varchar(50): A kiszállítási cím utcáját tartalmazó mező.
* „orderStreetNum” varchar(11): A kiszállítási címhez tartozó házszámot tartalmazó mező.
* „customerID” int(4): Ebben a mezőben hivatkozik a program a rendelést leadó személyre az ID-ja alapján.

**drink:**

* „ID” int(11): Az italok egyedi azonosítója, erre az értékre hivatkozik a program az orders táblából a drinkID mezőben.
* „drinkName” varchar(40): Az ital nevét tartalmazó mező.
* „drinkPrice” int(5): Az ital árát tartalmazó mező.
* „drinkStatus” int(1): Logikai mező, ennek a mezőnek az értékétől függ, hogy az ital elérhető-e vagy sem. 1 = elérhető, 0 = nem elérhető.

**food:**

* „ID” int(11): Az ételek egyedi azonosítója, erre az értékre hivatkozik a program az orders táblából a foodID mezőben.
* „foodName” varchar(40): Az étel nevét tartalmazó mező.
* „foodDesc” varchar(100): Ez a mező egy rövid leírást tartalmaz az étel alapanyagairól, hogy segítséget nyújtson a diszpécsereknek és a futároknak.
* „foodVAT” varchar(5): Az étel áfakulcsát tartalmazó mező.
* „foodPrice” int(5): Ez a mező tartalmazza étel árát.
* „foodAllergenID” varchar(40): Ebben a mezőben találhatók az ételben fellelhető allergének ID-jai szóközzel elválasztva.
* „foodStatus” int(1): Eldöntő mező, ennek a mezőnek az értékétől függ, hogy az étel elérhető-e vagy sem. 1 = elérhető, 0 = nem elérhető.

**orders**

* „orderID” int(11): A rendelések egyedi azonosítója, automatikusan növekvő szám.
* „orderNote” varchar(150): A rendelésekhez tartozó megjegyzések mezője, a rendeléseket megjegyzés nélkül is le lehet adni.
* „orderTime” datetime: Ebbe a mezőbe kerül az, hogy mikor vették fel a rendelést. A program automatikusan rögzíti ezt az adatot és a grafikus felületen nincs lehetőség módosítani.
* „orderDueTime” datetime: Ez a mező tartalmazza azt az időpontot, amikorra a rendelést kérték. Az orderTime mezővel ellentétben ennek az adatnak a módosítása lehetséges a grafikus felületen.
* „orderDestID” int(4): A rendeléshez tartozó kiszállítási cím egyedi azonosítójának mezője. Automatikusan rögzítésre kerül a program által.
* „orderDispatchID” int(3): A rendelést kiszállító futár egyedi azonosítóját tartalmazó mező. Módosítható adat amíg a rendelés ki nem lett szállítva.
* „orderStatus” int(1): Eldöntő mező, a rendelés állapotát tartalmazza. 0 = Készül, 1 = Kész.
* „foodID” varchar(255): Ebben a mezőben találhatóak az adott rendelésben megrendelt ételek azonosítójai. A mezőben tárolt adatok módosíthatók grafikus felületen.
* „drinkID” varchar(255): Ebben a mezőben találhatóak az adott rendelésben megrendelt italok azonosítói. A mezőben tárolt adatok módosíthatók grafikus felületen.
* „paymentID” int(4): Az adott rendeléshez tartozó fizetési információkat tartalmazó adatsor azonosítója, a payment tábla ID-ja szerint hozzárendelt adat.

**finishedorders**

* „forderID” int(11): A kiszállított rendelések egyedi azonosítója.
* orderNote” varchar(150): A kiszállított rendelésekhez tartozó megjegyzések mezője, a rendeléseket megjegyzés nélkül is le lehet adni.
* „orderTime” datetime: Ebbe a mezőbe kerül az, hogy mikor vették fel a rendelést. A program automatikusan rögzíti ezt az adatot és a grafikus felületen nincs lehetőség módosítani.
* „orderDueTime” datetime: Ez a mező tartalmazza azt az időpontot, amikorra a rendelést kérték. Az orderTime mezővel ellentétben ennek az adatnak a módosítása lehetséges a grafikus felületen.
* „orderDestID” int(4): A kiszállított rendeléshez tartozó kiszállítási cím egyedi azonosítójának mezője. Automatikusan rögzítésre kerül a program által.
* „orderDispatchID” int(3): A rendelést kiszállító futár egyedi azonosítóját tartalmazó mező. Módosítható adat amíg a rendelés ki nem lett szállítva.
* „foodID” varchar(255): Ebben a mezőben találhatóak az adott rendelésben megrendelt ételek azonosítójai. A mezőben tárolt adatok módosíthatók grafikus felületen.
* „drinkID” varchar(255): Ebben a mezőben találhatóak az adott rendelésben megrendelt italok azonosítójai. A mezőben tárolt adatok módosíthatók grafikus felületen.
* „paymentID” int(4): Az adott rendeléshez tartozó fizetési információkat tartalmazó adatsor azonosítója, a payment tábla ID-ja szerint hozzárendelt adat.

**packaging**

Ez a másik egy értékkel rendelkező tábla az adatbázisban, amit azért hoztam létre, hogy a csomagolási díjat lehessen a grafikus felületen szerkeszteni.

* „ID” int(11): Erre az értékre hivatkozik a program amikor az admin szerkeszti a csomagolási díjat.
* „price” double: A csomagolás ára.

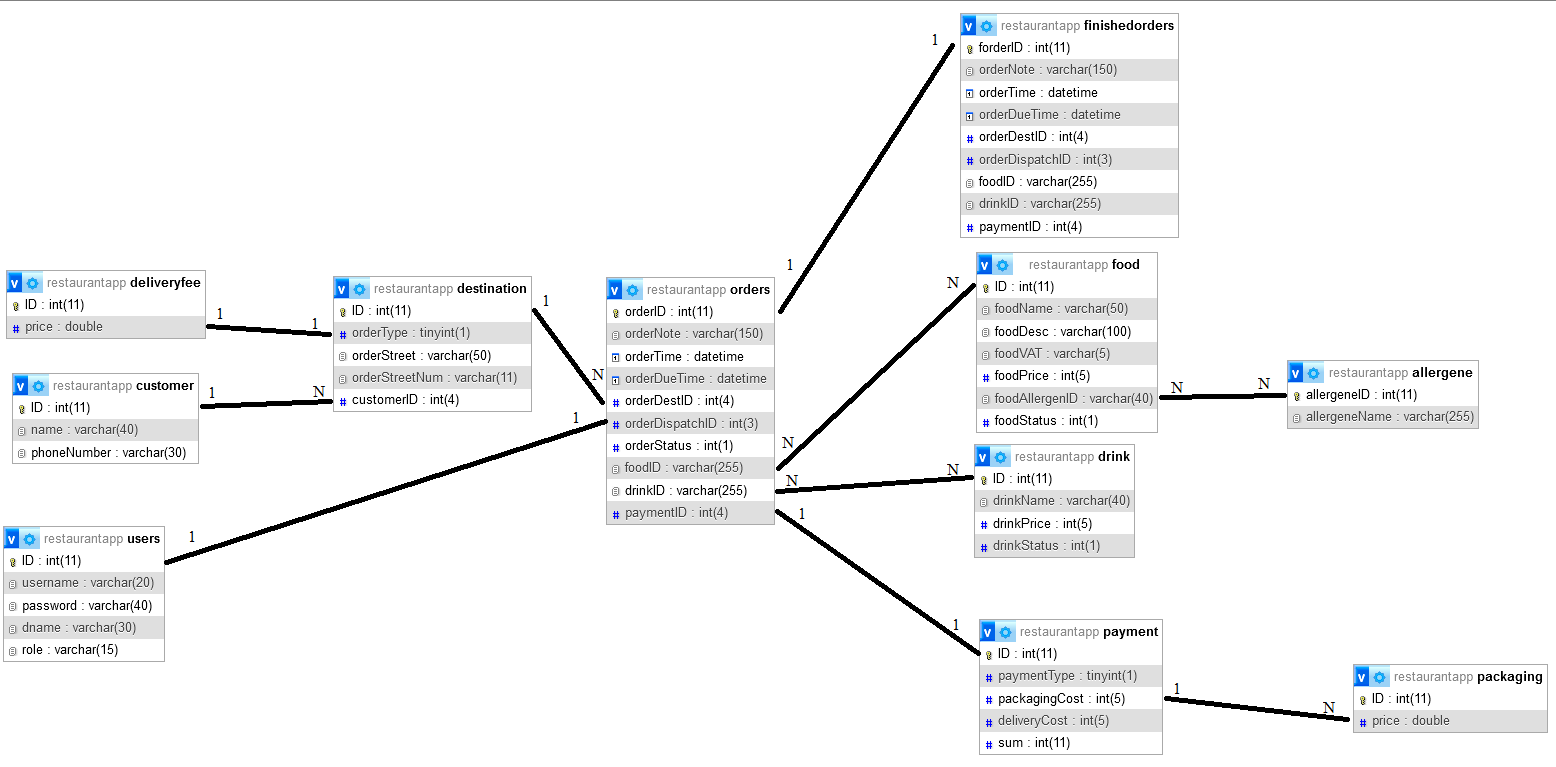
**payment**

* „ID” int(11): A rendelésenkénti fizetések egyedi azonosítója, automatikusan növekvő szám, ezzel az adattal kerül meghívásra a végösszeg és a tételek ára az orders táblában.
* „paymentType” tinyint(1): Logikai mező, itt kerül feljegyzésre a fizetőeszköz. 0 = Készpénz, 1 = Kártya.
* „packagingCost” int(5): A rendelés csomagolásainak összesített ára. Beleszámít a végösszegbe.
* „deliveryCost” int(5): A rendelés kiszállítási díja, elviteles rendelés esetén 0, egyébként a grafikus felületen szerkeszthető összeg.
* „sum” int(11): Az ételek és italok összesített árát tartalmazó mező, nincsen beleszámolva a csomagolás és a kiszállítási díj.

**users**

* „ID” int(11): A felhasználók egyedi azonosítója, automatikusan növekvő szám.
* „username” varchar(20): Ebben a mezőben találhatóak a felhasználók felhasználó nevei. Ez az egyik adat, amely szükséges a programba való bejelentkezéshez.
* „password” varchar(40): Az adott felhasználókhoz tartozó jelszót tartalmazó mező, fontos eleme a bejelentkezési adatoknak. Ez a másik adat amely szükséges a programba való belépéshez.
* „dname” varchar(30): A felhasználó programban megjelenített neve, nem összekeverendő a felhasználónévvel, ami a bejelentkezéshez szükséges, ennek kizárólag esztétikai szerepe van.
* „role” varchar(15): A felhasználó szerepkörét tartalmazó mező, fontos eleme a hitelesítésnek, ettől függ, hogy az adott felhasználónak mihez van hozzáférése a programban. A szerepkör lehet: „cook” : „konyhai személyzet”, „dispatch” : „futár” és „admin” : „tulaj / rendszergazda”.

**Adatbázis kapcsolatok**

Az alábbiakban beszúrtam egy képet az adatbázis tábláinak kapcsolatairól.

ábra 15 Relációs modell

Magyarázat:  
destination 1 - 1 deliveryfee, mert egy úticélnak csak egy szállítási díja van

destination N - 1 customer, mert egy vásárló rendelhet több helyre

destination 1 - 1 order, mert egy rendeléshez csak egy úti cél tartozhat

order 1 - 1 user, mert egy rendelést csak egy felhasználó vehet fel

order 1 - 1 finishedorder, mert egy rendelésből csak egy kész rendelés lehet

order M - N food, mert egy rendeléshez tartozhat több étel és fordítva

food M - N allergene, mert egy ételhez tartozhat több allergén és fordítva

order M - N drink, mert egy rendeléshez tartozhat több ital és fordítva

order 1 - 1 payment, mert egy rendeléshez csak egy fizetés tartozhat

payment 1 - N packaging, mert egy végösszeghez tartozhat több csomagolási ár is

**Tesztelés**

Hiba #1: Az Aktív Rendelések nevű fülön a rendelés módosításánál a program megpróbálta elmenteni a rendelést még úgy is, hogy nem volt kijelölve egy sem, ezt természetesen kijavítottam, ugyanis egy kritikus hiba volt és azóta a program le ellenőrzi, hogy van-e kijelölt rendelés és csak ezután tölti fel a változtatásokat.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

ábra 16 Hiba #1 kép

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Automatikusan generált leírásHiba #2: A programba be lehetett lépni üres adatokkal az „ELVITEL” felhasználó miatt, ezt a kritikus hibát is kezeltem, azzal, hogy mostantól a program minden bejelentkezési kísérlet előtt ellenőrzi, hogy a felhasználónév és jelszó mezők ki vannak-e töltve.

ábra 17 Hiba #2 kép

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírásA képen képernyőkép, Téglalap látható

Automatikusan generált leírásHiba #3: Mivel ugyanazt a módszert és mezőt használtam a rendelés felvétele és az aktív rendelések tételeinél, ezért duplán jelenítette meg a tételeket a program és keresztben is megjelenítette őket, a hibát javítottam azzal, hogy átneveztem a változókat és a listboxokat.

ábra 18 Rendelés felvétele, Tételek Hiba #3

ábra 19 Aktív Rendelések, Tételek Hiba #3

**Fejlesztéshez felhasznált források**

Az alábbiakban fogom felsorolni a fejlesztéshez felhasznált forrásokat:

* A programban található ikonokat az alábbi oldalról töltöttem le: [Iconfinder](https://www.iconfinder.com/)[[4]](#footnote-4).
* Az alábbi youtube videó segített megérteni a flowlayout panelek feltöltésének hogyanját és miértjét: a videó[[5]](#footnote-5).
* Hasznos segítség volt a kódolásban az alábbi oldal: Stackoverflow[[6]](#footnote-6).

**A programban használt funkciók magyarázata**

Bejelentkezes.cs/ bejelentkezes():

Egy egyszerű ellenőrző funkció, amely egy adattáblába tölti be a MySQL-től visszakapott információt(ha a táblában van olyan, amely egyezik a keresett információval) és ennek a táblának ellenőrzi a funkció a hosszát az alábbi módon:

* Ha a tábla hossza = 1, akkor van ilyen felhasználó és beenged a rendszer
* Ha a tábla hossza != 1, akkor nincs ilyen felhasználó és hibaüzenetet dob a program

Fo\_Form.cs/ role\_kezeles():

Fontos része a programnak, ugyanis ez a funkció kezeli a felhasználói jogköröket. A bejelentkezésnél átadott role-t veszi alapul és annak megfelelően teszi láthatóvá vagy rejtetté a program egyes ablakait és füleit.

Fo\_Form.cs/ allergene\_lekerdezes():

Egy egyszerű SQL lekérdezés, ami visszaadja az összes allergént és azok ID-ját és beírja őket egy listboxba.

Fo\_Form.cs/ Ablak\_mozgatas region:

Ez a programrész azt csinálja, amit a neve is elmond róla: lehetővé teszi az ablak mozgatását a kis kéz ikon segítségével.

Rendeles\_felvetel.cs/ ures\_rendeles\_kezeles():

A rendelés felvétel ablak hibakezelő egysége, ellenőrzi, hogy minden szükséges adatot megadtunk-e és ha nem akkor a kihagyott adatra releváns hibát dob.

Rendeles\_felvetel.cs/ szalldij\_betolt():

Egy egyszerű MySQL lekérdezés, ami a szállítási díj összegét adja vissza.

Rendeles\_felvetel.cs/ customerek\_betolt():

Egy MySQL lekérdezés, ami betölti a már az adatbázisban lévő vásárlókat egy combobox-ba.

Rendeles\_felvetel.cs/ uj\_rendeles():

Ez a funkció MySQL lekérdezések, UPDATE-ek és INSERTEK láncolata, ezzel a funkcióval hozza létre a program az új rendeléseket, de a programon belül minden egysége kommentálva van a funkciónak az érthetőség érdekében.

Rendeles\_felvetel.cs/ mezok\_kiurit():

Ez egy egyszerű mező kiürítő funkció, amit a rendelés felvétel végén hívtam meg. Sokkal esztétikusabb lesz tőle az adatfelvétel és így nem kell fölöslegesen azzal foglalkozni, hogy egyesével kitörölgessük az adatokat a különböző mezőkből.

Rendeles\_felvetel.cs/ etelek\_betolt():

Ez a funkció egy MySQL adatlekérésből és User Control példányosításból áll, ez a funkció tölti fel az ételek menüt az adatbázisban lévő ételekkel.

Rendeles\_felvetel.cs/ italok\_betolt():

Az etelek\_betolt() funkció italos megfelelője.

Rendeles\_felvetel.cs/ rend\_tetelekFeltoltes():

A program ezen része felelős a jelenlegi rendelés elemeit tartalmazó lista feltöltéséért.

Rendeles\_felvetel.cs/ customer\_select():

Ez a kódrészlet a comboBox2 nevű combobox változásait figyeli és létrehoz egy változót, amiben eltárolja az aktuálisan kiválasztott vásárló indexét.

Rendeles\_felvetel.cs/ elvitel\_ellenorzes():

Ez a funkció az elvitel checkbox(checkBox1) változásait ellenőrzi és annak hatására tölti ki az utca és házszám mezőket az alábbiak szerint:

* Ha a checkBox be van pipálva, akkor az elvitel változó értékét 0-ra állítja és az utca és házszám mezőket feltölti az étterem utcájával és az étterem házszámával.
* Ha a checkBoxot-ból kiszedjük a pipát, akkor az elvitel értékét 1-re állítja és az utca és házszám mezőkbe „Utca:” és „Házszám:” helyettesítő szövegek kerülnek.

Rendeles\_felvetel.cs/ Makeshift PLaceholder region:

Mivel régi NET Frameworkot használtam a program írása közben, ezért kénytelen voltam megoldani a placeholder attribútum hiányából keletkező hiányosságokat a szoftverben, ez rész ezt kezeli le.

Aktiv\_Rendelesek.cs/ szallDij\_betolt():

Ez a funkció tölti be az aktuális kiszállítási díjat a kiszDijNumericUpDown mezőbe.

Aktiv\_Rendelesek.cs/ rendelesek\_betolt():

Ez a funkció tulajdonképpen az Aktív Rendelések fül fő alkotóeleme. Ez a programrész felelős az aktív rendelések betöltéséért és az Aktiv\_rendeles User Control példányosításáért. A rendeléseket egy MySQL lekérdezéssel kéri le.

Aktiv\_Rendelesek.cs/ etelek\_betolt():

Ugyan az, mint a Rendeles\_felvetel.cs-ben szereplő etelek\_betolt().

Aktiv\_Rendelesek.cs/ italok\_betolt():

Ugyan az, mint a Rendeles\_felvetel.cs-ben szereplő italok\_betolt().

Aktiv\_Rendelesek.cs/ rend\_tetelekFeltoltes():

Miután ráklikkeltünk a ceruza ikonra az egyik aktív rendelésen, ez a funkció felelős az adatok betöltéséért rend\_tetelek listBox-ba.

Aktiv\_Rendelesek.cs/ rend\_adatokBetolt():

Ez a funkció is azután kap értelmet miután rákattintottunk a ceruza ikonra az egyik rendelésen, majd átléptünk az Adatok módosítása fülre. A funkció ezek után betölti a szerkesztendő adatokat.

Aktiv\_Rendelesek.cs/ elvitel\_ellenorzes():

Ugyan az, mint a Rendeles\_felvetel.cs-ben szereplő elvitel\_ellenorzes().

Aktiv\_Rendelesek.cs/ adatmodositas():

Ez a funkció felelős az adatok módosításáért az Aktiv\_Rendelesek Adatok módosítása fülén, az adatokat tartalmazó textBoxok-ból veszi a frissítendő adatokat és egy MySQL UPDATE-el tölti fel őket az adatbázisba, majd meghívja az adat\_mezok\_kiurit() funkciót, ami üríti a mezőket és kiírja egy MessageBox-ba hogy „Adatok sikeresen módosítva”, ha sikerült az adatmódosítás.

Aktiv\_Rendelesek.cs/ rend\_szerk():

Ez a funkció felelős a kiválasztott rendelés tételeinek frissítéséért. Egy stringet készít a rendelés tételeiből és azt tölti fel egy UPDATE-tel az adatbázisba.

Aktiv\_Rendelesek.cs/ Dictionary megszamlalas():

Ez a dictionary felelős a végösszeg kiszámításáért és a tételek megszámlálásáért. A dictionary a tétel nevét veszi kulcs értékként, az elemek számát count-ként és az azonos kulcsú elemek összegét a tételek szummájaként

Aktiv\_Rendelesek.cs/ kiszDijMentesBtn\_CLick():

Ez a programrész felelős a kiszállítási díj összegének frissítéséért. Egy MySQL UPDATE-el történik a feltöltés.

Aktiv\_Rendelesek.cs/ torlesBtn\_Click():

Ez a funkció felelős a rendelés adatbázisból való törléséért. Ezzel egy időben törli a payments táblából is a recordot.

Aktiv\_Rendeles.cs/ combobox\_kezeles():

Ez a funkció kapcsolja ki a combobox görgethetőségét az aktív rendeléseken.

Aktiv\_Rendeles.cs/ vegosszeg\_beiras():

Ez a kódrész tölti fel a payments táblába az aktuális adatsor szummáját és csomagolásának árát.

Aktiv\_Rendeles.cs/ vegosszeg\_szamitas():

Ez a funkció egy összesítő része a végösszeg kiszámításához szükséges funkcióknak. Meghívja az id\_lekeres(), rend\_tetelekLekeres(), csomagar\_lekeres() és az adat\_beiras() funkciókat

Aktiv\_Rendeles.cs/ csomagar\_lekeres():

Ez a funkció kéri le a csomagolás aktuális árát az adatbázisból

Aktiv\_Rendeles.cs/ id\_lekeres():

Ez a funkció kéri le az aktuális rendelés tételeinek ID-ját egy MySQL SELECT-el.

Aktiv\_Rendeles.cs/ rend\_tetelekLekeres():

Ez a funkció felelős az aktuális rendelés tételeinek eltárolásáért és szummájuknak kiszámolásáért

Aktiv\_Rendeles.cs/ fiztip\_lekerdezes():

Ez a programrész felelős az aktuális rendelés fizetéstípusának és kiszállítási díjának az adatbázisból való lekéréséért.

Aktiv\_Rendeles.cs/ futarok\_betolt():

A program ezen része tölti be a futárokat a rendelések comboBoxjaiba és ha az aktuális rendeléshez már van hozzárendelve futár akkor azt automatikusan ki is választja.

Futar\_nezet.cs/ rendelesek\_betolt():

Ez a funkció először lekéri a kiszállítandó rendelések adatait az adatbázisból, majd szűri és példányosítja őket a Futar\_cimek User Control és a bejelentkezett felhasználó ID és szerepkörének segítségével, így csak azokat a rendeléseket jeleníti meg, amik az adott felhasználóra tartoznak. Ez alól az admin kivétel, ő mindegyik rendelést látja.

Futar\_nezet.cs/ adatbeiras():

Ez a kódrészlet segítségével tudja a program megjeleníteni a számla előnézetét.

Futar\_nezet.cs/ fiztip\_kezeles():

Ez a funkció egy kisegítő része a kódnak, ami azt dönti el, hogy a számlán milyen fizetőeszköz legyen megjelenítve.

Futar\_nezet.cs/ datumkezeles():

Ez a funkció is egy kisegítő része a kódnak, ami azt dönti el, hogy a határidő csak órát és percet jelenítsen meg vagy dátumot is.

Futar\_nezet.cs/ keszorder\_atadas():

A kód ezen része felelős azért, hogy a kiszállított rendelések eltűnjenek az orders táblából és átkerüljenek a finishedorders táblába.

Futar\_nezet.cs/ order\_lekeres():

Ez a kódrész kéri le az aktuális rendelés adatait, hogy azok kiszállítás esetén átadásra kerüljenek a finishedorders táblába.

Menu\_elemek.cs/ hozzaadasBtn\_Click():

Ez a kódrész felelős a rendelés felvétel tételeinek feljegyzéséért.

Menu\_elemek\_Aktiv\_rend.cs/ hozzaadasBtn\_Click():

Ez a kódrész felelős az aktív rendelés tételeinek bővítéséért.

Menu\_elemek\_Etetl\_ital\_felvetel.cs/ modositBtn\_Click():

Ez a kódrész adja át az adatokat az Etel\_ital\_felvetel formnak adatmódosításhoz.

Etel\_ital\_felvetel.cs/ csomagolas\_betolt():

Ez a kódrész tölti be a csomagolás aktuális árát az adatbázisból.

Etel\_ital\_felvetel.cs/ allergenek\_betolt():

Ez a funkció kéri le az allergéneket az adatbázisból és példányosítja őket az Allergen.cs User Control segíłtségével.

Etel\_ital\_felvetel.cs/ adatmodositas\_ellenorzes():

Ez a funkció felelős azért, hogy előre hozza a kiválasztott kategóriát az ablakban. A kategóriák az alábbiak: Étel, Ital, Allergén.

Etel\_ital\_felvetel.cs/ checkbox\_atalakitas():

Ez a kódrészlet dönti el, hogy az adott étel/ital elérhető-e az étteremben.

Etel\_ital\_felvetel.cs/ elem\_modositas():

Ez a funkció szükséges a módosított adatok feltöltéséhez mind az étel, ital és allergén kategóriában.

Etel\_ital\_felvetel.cs/ elemFelvetelButton\_CLick():

Ez a kódrészlet felelős az új ételek, italok és allergének felvételéért.

Etel\_ital\_felvetel.cs/ csomagMentesBtn\_CLick():

Ez a kódrész felelős a csomagolás árának módosításáért.

**Mappaszerkezet**

* database: Az adatbázist tároló mappa.
* icons: Az alkalmazásban felhasznált ikonokat tartalmazó mappa.
* FoodApp: Az alkalmazás fájljait tartalmazó mappa.

# **Összegzés**

A negyedik fejezet tartalmazza a program fejlesztése közben elért eredményeimet, azt hogy mit sikerült megvalósítanom a kitűzött célok közül.

1. Kitűzött cél: „Rendelések felvételének egyszerűsítése éttermi dolgozók számára”

Véleményem szerint sikeresen teljesítettem ezt a célt, több emberrel is teszteltettem a programot különböző korosztályokból és a felhasználói dokumentáció elolvasása után egyiküknek sem volt gondja a szoftver működtetésével és felhasználásával.

1. Kitűzött cél: „Egyszerű étlap frissítés lehetővé tétele”

Az innovatív kezelőfelületnek hála az ételek, italok és allergének frissítése a felhasználói dokumentáció elolvasása után nem okozhat gondot egy felhasználónak sem, ezzel a kitűzött célt elértnek tekintem.

1. Kitűzött cél: „Minden szükséges információ megosztása a futárokkal a sikeres kiszállításhoz”

A futárok számára egy könnyen értelmezhető informatív felületet biztosítottam a program fejlesztése közben, hogy megkönnyítsem számukra a rendelés adatai után való értékes perceket felemésztő keresgélés fejfájását, ugyanis az étel akkor a legjobb ha friss!

1. Kitűzött cél: „Egy könnyen kezelhető felület létrehozása éttermek számára”

Ez a cél tulajdonképpen magába foglalja az előbb elmondottakat, amelyeket sikeresen teljesítettem. A program a felhasználói dokumentáció elolvasása után teljes mértékben értelmezhető és a dokumentációból elsajátított tudás könnyen átadható, így az új munkaerő könnyen betaníthatóvá válik a továbbiakban ezzel egy könnyen kezelhető felületet megteremtve az éttermek számára.

Összességében úgy vélem, hogy hatékonyan használtam az rendelkezésre álló időt, és sikerült majdnem teljes mértékben megvalósítanom a tervezett célokat. Fontosnak tartom azonban továbbfejlesztési lehetőségek kiaknázását annak érdekében, hogy a rendszer még hatékonyabb és biztonságosabb legyen.

Először is, az adatbiztonság kiemelt fontosságú. Azért gondolom a legfontosabb továbbfejlesztési lehetőségnek az adatbázis titkosítását, hogy megvédjük az érzékeny információkat illetéktelen hozzáférés ellen. Ez segít megelőzni az adatlopást és bizalmat épít a felhasználókban.

Másodszor, egy GPS rendszer integrálása nagyobb funkcionalitást és kényelmet nyújthatna mind az étterem tulajdonosoknak, mind a kiszállítóknak. Ennek segítségével a tulajdonosok nyomon követhetnék a kiszállítások útját, míg a kiszállítók könnyebben navigálhatnának a címekhez. Ez növelné az ügyfélélményt és a hatékonyságot is.

Végül, a program kiterjesztése több településre lehetővé tenné a szolgáltatás bővítését és a piac növelését. Ennek részeként a kiszállítási díjak tárolása és használata több településre vonatkozóan további lehetőségeket teremtene a vállalkozás számára. Ezáltal rugalmasabbá és versenyképesebbé válna a szolgáltatás. Természetesen ez csak abban az esetben szükséges, ha nagyobb célközönséget szeretnénk elérni vagy a célközönségünk már igényli az efféle bővülést.

Az ilyen irányú fejlesztésekkel a rendszer további előnyökhöz juthat a piacon. Emellett javulhat az ügyfélélmény és az adatbiztonság is, ami mindig előnyös a vállalkozás számára.

# **Felhasznált források**

Az alábbiakban mellékelem a felhasznált forrásokat.

**Weboldalak**

MySQL

<https://www.mysql.com/> [letöltés: 2024. 04. 10]

XAMPP

<https://www.apachefriends.org/hu/index.html> [letöltés: 2024. 03. 15]

Visual Studio 2019

<https://visualstudio.microsoft.com/vs/older-downloads/> [letöltés: 2024. 04. 07]

Stackoverflow

<https://stackoverflow.com/> [letöltés: 2024. 03. 19]

Iconfinder

<https://www.iconfinder.com/> [letöltés: 2024. 03. 29]

**Videó**

YouTube

<https://www.youtube.com/watch?v=u71RJZm7Gdc> [letöltés: 2024. 03. 05]

# **Ábrajegyzék**

ábra 1 Bejelentkezési felület 3

ábra 2 Felhasználónév/ jelszó hiba 3

ábra 3 Rendelés felvétel 4

ábra 4 Rendelés felvétel tételek 4

ábra 5 Aktív rendelések 5

ábra 6 Aktív Rendelések tétel szerkesztés 6

ábra 7 Aktív Rendelések adatok módosítása 7

ábra 8 Aktív rendelés 7

ábra 9 Futár nézet 8

ábra 10 Futár nézet 2 9

ábra 11 Futár nézet Számla 10

ábra 12 Étel/Ital elem módosítás 10

ábra 13 Étel/ital elem módosítás 2 11

ábra 14 Új elem felvétele 12

ábra 15 Relációs modell 22

ábra 16 Hiba #1 kép 23

ábra 17 Hiba #2 kép 23

ábra 18 Rendelés felvétele, Tételek Hiba #3 23

ábra 19 Aktív Rendelések, Tételek Hiba #3 23

1. <https://www.mysql.com/> [↑](#footnote-ref-1)
2. https://www.apachefriends.org/hu/index.html [↑](#footnote-ref-2)
3. https://visualstudio.microsoft.com/vs/older-downloads/ [↑](#footnote-ref-3)
4. <https://www.iconfinder.com/> [↑](#footnote-ref-4)
5. <https://www.youtube.com/watch?v=u71RJZm7Gdc> [↑](#footnote-ref-5)
6. <https://stackoverflow.com/> [↑](#footnote-ref-6)