Széchenyi István Egyetem

Gépészmérnöki, Informatikai és Villamosmérnöki Kar

Informatika Tanszék

**SZAKDOLGOZAT**

**Kiss Vince Gergely**

**Mérnök informatikus BSc szak**

2024

|  |
| --- |
| [Gerincen:] Kiss Vince Gergely, 2024 |



**SZAKDOLGOZ****AT**

**Ételkifőzde online rendelési és leltár felület tervezése és megvalósítása**

**Kiss Vince Gergely**

**Mérnök Informatikus BSc szak**

**2024**

A képen szöveg, Betűtípus, dokumentum látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Automatikusan generált leírás

**Nyilatkozat**

Alulírott, Kiss Vince Gergely (A2UWSD), mérnök informatikus BSc szakos hallgató kijelentem, hogy az Ételkifőzde online rendelési és leltár felület tervezése és megvalósítása című szakdolgozat feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Győr, [beadás dátuma]

hallgató

**Kivonat**

Ételkifőzde online rendelési és leltár felület tervezése és megvalósítása

A rendszer, egy konyha napi folyamatait hivatott támogatni rendelés kezelés, kiszállítás és leltár tevékenységek elősegítésében. Azért esett erre a választásom, mert korábbi tanulmányaimban, illetve itt az egyetemen is sajátítottam el ezzel kapcsolatban releváns tudást, emellett egy ismerős megkeresése okán szükség volt egy ilyen rendszer megvalósítására.

A szakdolgozatom első és második fejezetében, bemutatom az ételkifőzdét, betekintést nyújtok egy napra a jelenlegi rendszerben. A harmadikban erre építkezve a jelenlegi rendszert elemzem. A negyedik rész az új rendszer szükségességét hivatott alátámasztani. Megoldási javaslatokat mutatok be, illetve hasonló már kész rendszereket is.

Az ötödik fejezetben kezdődik meg a tervezés folyamata meghatározom a funkcionális és nem funkcionális követelményeket egyaránt, ehhez szorosan kapcsolódóan a hatodik fejezetben Gantt diagramon mutatom be a fejlesztés ütemtervét, melyen meglennek fontos munkafolyamatok tervezett végzése hetekre bontva. A hetedik részben az általam választott fejlesztői környezetről ejtek néhány szót.

A nyolcadik fejezet az adatmodell tervezése és megvalósítása, ebben bemutatom a teljes adatbázis szerkezetét kapcsolatokkal, valamint, hogy hogyan felel meg a normálformáknak. Ezen kívül az egyes táblák felépítése is itt található.

A következő a fejezet a fejlesztés életciklusát mutatja be. Az általam választott és követett MVC minta használatát és a tesztelés folyamatát. A tizedik fejezet a felhasználói dokumentáció itt található minimális rendszerkövetelmények, általános specifikáció. Emellett oktatóanyag a rendszer felhasználói számára, felhasználói, illetve adminisztrátori szinten egyaránt. Ezen fejezet végén esik szó a fejlesztési lehetőségekről, a fejlesztési folyamat utolsó lépéseként, a rendszer karbantartása.

Az utolsó, tizenegyedik fejezetben összegzem a szakdolgozatomban végrehajtott munkát.

**Abstract**

Design and Implementation of an Online Ordering and Inventory System for a Food Service

The system is intended to support the daily processes of a kitchen in facilitating order management, delivery, and inventory activities. I chose this topic because I acquired relevant knowledge in my previous studies and here at the university, and due to an acquaintance's request, there was a need to implement such a system.

In the first and second chapters of my thesis, I introduce the food service, providing insight into a day in the current system. In the third chapter, I analyze the current system based on this. The fourth part is intended to justify the necessity of a new system. I present solution proposals and similar existing systems.

The planning process begins in the fifth chapter, where I define both functional and non-functional requirements. In close connection, in the sixth chapter, I present the development schedule on a Gantt chart, showing important work processes planned to be completed on a weekly basis. In the seventh chapter, I discuss the development environment I have chosen.

The eighth chapter involves the design and implementation of the data model, where I present the entire database structure with relationships and how it meets normal forms. Additionally, the structure of each table is also detailed here.

The next chapter presents the development lifecycle, the MVC pattern I chose and followed, and the testing process. The tenth chapter is the user documentation, containing minimum system requirements, general specifications, and instructional materials for users at both user and administrator levels. The chapter ends with development possibilities as the last step of the development process and the system maintenance.

In the final, eleventh chapter, I summarize the work completed in my thesis.

**Tartalomjegyzék**

[1 Bevezetés 1](#_Toc182911886)

[2 Ételkifőzde bemutatása 2](#_Toc182911887)

[2.1 Egy nap az ételkifőzdében 2](#_Toc182911888)

[3 Üzleti igény definiálása 3](#_Toc182911889)

[3.1 Összegezve 4](#_Toc182911890)

[3.2 SWOT-analízis a jelenlegi rendszerről 4](#_Toc182911891)

[4 Az új rendszert előkészítő elemzés 6](#_Toc182911892)

[4.1 A jelenlegi rendszer hibáinak definiálása 6](#_Toc182911893)

[4.1.1 Ishikawa-diagram 6](#_Toc182911894)

[4.2 Megoldási javaslatok vizsgálata 7](#_Toc182911895)

[4.2.1 Wolt/Foodora 7](#_Toc182911896)

[4.2.2 iBar leltározó app 8](#_Toc182911897)

[4.2.3 Simpra 8](#_Toc182911898)

[4.2.4 Saját rendszer fejlesztése 8](#_Toc182911899)

[4.3 Hasonló megoldások a piacon 9](#_Toc182911900)

[4.3.1 Egészségpajzs 9](#_Toc182911901)

[4.3.2 e-food 9](#_Toc182911902)

[5 Funkcionális és nem funkcionális követelmények meghatározása 10](#_Toc182911903)

[5.1 Funkcionális követelmények 10](#_Toc182911904)

[5.1.1 Hozzávalók, ételek, étlapok nyilvántartása 10](#_Toc182911905)

[5.1.2 Login rendszer 11](#_Toc182911906)

[5.1.3 Rendelés 12](#_Toc182911907)

[5.1.4 Adminisztratív folyamatok támogatása 13](#_Toc182911908)

[5.2 Nem funkcionális követelmények 14](#_Toc182911909)

[5.3 SWOT-analízis az új rendszerről 14](#_Toc182911910)

[6 Fejlesztés ütemterve - Gantt diagram 16](#_Toc182911911)

[7 Alkalmazott fejlesztői környezet 18](#_Toc182911912)

[7.1 XAMPP 18](#_Toc182911913)

[7.2 Visual Studio 2022 18](#_Toc182911914)

[7.3 PhpMyAdmin 18](#_Toc182911915)

[8 Adatmodell tervezése és megvalósítása 19](#_Toc182911916)

[8.1 Az adatbázis szerkezete kapcsolatokkal 19](#_Toc182911917)

[8.1.1 Normálformák 20](#_Toc182911918)

[8.2 Adatbázistáblák bemutatása 20](#_Toc182911919)

[8.2.1 Étel tábla 21](#_Toc182911920)

[8.2.2 ÉtelHozzávaló tábla 21](#_Toc182911921)

[8.2.3 Étlap tábla 22](#_Toc182911922)

[8.2.4 Hozzávaló tábla 22](#_Toc182911923)

[8.2.5 Napirendelés tábla 23](#_Toc182911924)

[8.2.6 Rendelés tábla 23](#_Toc182911925)

[8.2.7 Település tábla 24](#_Toc182911926)

[8.2.8 TételesRendelés tábla 24](#_Toc182911927)

[8.2.9 User tábla 25](#_Toc182911928)

[9 A rendszer fejlesztése 26](#_Toc182911929)

[9.1 MVC minta 26](#_Toc182911930)

[9.2 Néhány kiemelt kódrészlet 26](#_Toc182911931)

[9.3 Tesztelési dokumentáció 29](#_Toc182911932)

[9.3.1 Egységteszt alkalmazása 29](#_Toc182911933)

[9.3.2 Integrációs- és rendszerteszt alkalmazása 29](#_Toc182911934)

[9.3.3 Normál- és extrém tesztesetek kezelése 29](#_Toc182911935)

[10 Felhasználói dokumentáció 30](#_Toc182911936)

[10.1 Rendszerkövetelmények 30](#_Toc182911937)

[10.2 A program általános specifikációja 30](#_Toc182911938)

[10.3 Oktatóanyag a rendszer felhasználói számára 31](#_Toc182911939)

[10.3.1 Első lépések 31](#_Toc182911940)

[10.3.2 Regisztráció 32](#_Toc182911941)

[10.3.3 Rendelés menete 33](#_Toc182911942)

[10.3.4 Alapszintű felhasználó eszközei 35](#_Toc182911943)

[10.3.5 Adminisztrátor eszközei 36](#_Toc182911944)

[10.4 A rendszer bevezetése 44](#_Toc182911945)

[10.5 Fejlesztési lehetőségek 44](#_Toc182911946)

[11 Összegzés 45](#_Toc182911947)

[Irodalomjegyzék 46](#_Toc182911948)

[Mellékletek 47](#_Toc182911949)

**Ábrajegyzék**

[1. ábra Ishikawa-diagram - Forrás: Saját szerkesztés a Lucidchart eszközzel 6](#_Toc182911950)

[2. ábra Hozzávalók, ételek, étlapok USE CASE diagram - Forrás: Saját szerkesztés 11](#_Toc182911951)

[3. ábra Login USE CASE diagram - Forrás: Saját szerkesztés 11](#_Toc182911952)

[4. ábra Rendelés USE CASE diagram - Forrás: Saját szerkesztés 12](#_Toc182911953)

[5. ábra Admin folyamatok USE CASE diagram – Forrás: Saját szerkesztés 13](#_Toc182911954)

[6. ábra phpMyAdmin – Forrás: https://www.phpmyadmin.net 18](#_Toc182911955)

[7. ábra Az adatbázis szerkezete – Forrás: phpMyAdmin 19](#_Toc182911956)

[8. ábra Szoftver követelmények 30](#_Toc182911957)

[9. ábra Menürendszer – Forrás: képernyőkép 31](#_Toc182911958)

[10. ábra Belépés – Forrás: képernyőkép 32](#_Toc182911959)

[11. ábra Regisztráció – Forrás: képernyőkép 33](#_Toc182911960)

[12. ábra Étlapok – Forrás: képernyőkép 34](#_Toc182911961)

[13. ábra Kosár egy tétele – Forrás: képernyőkép 34](#_Toc182911962)

[14. ábra Rendeléseim felület asztali – Forrás: képernyőkép 35](#_Toc182911963)

[15. ábra Rendeléseim felület telefonos – Forrás: képernyőkép 35](#_Toc182911964)

[16. ábra Admin menü – Forrás: képernyőkép 36](#_Toc182911965)

[17. ábra Hozzávalók alfunkció – Forrás: képernyőkép 37](#_Toc182911966)

[18. ábra Ételek alfunkció – Forrás: képernyőkép 37](#_Toc182911967)

[19. ábra Hozzávalók ételekhez rendelése alfunkció – Forrás: képernyőkép 38](#_Toc182911968)

[20. ábra Étel hozzávaló igénye – Forrás: képernyőkép 38](#_Toc182911969)

[21. ábra Település alfunkció – Forrás: képernyőkép 39](#_Toc182911970)

[22. ábra Ideiglenes étlap készítése – Forrás: képernyőkép 39](#_Toc182911971)

[23. ábra Ideiglenes étlap – Forrás: képernyőkép 40](#_Toc182911972)

[24. ábra Elkészített dátumok az étlapon – Forrás: képernyőkép 40](#_Toc182911973)

[25. ábra Rendelések kifizetése – Forrás: képernyőkép 41](#_Toc182911974)

[26. ábra Mai tételek (1) – Forrás: képernyőkép 42](#_Toc182911975)

[27. ábra Mai tételek (2) – Forrás: képernyőkép 42](#_Toc182911976)

[28. ábra Háromheti szükségletek kimutatása – Forrás: képernyőkép 43](#_Toc182911977)

[29. ábra Napi teendők 43](#_Toc182911978)

**Táblázatjegyzék**

[1. táblázat SWOT-analízis jelenlegi - Forrás: Saját szerkesztés 4](#_Toc182911979)

[2. táblázat SWOT-analízis új - Forrás: Saját szerkesztés 14](#_Toc182911980)

[3. táblázat Gantt diagram - Forrás: Saját szerkesztés 16](#_Toc182911981)

[4. táblázat Étel tábla – Forrás: phpMyAdmin 21](#_Toc182911982)

[5. táblázat ÉtelHozzávaló tábla – Forrás: phpMyAdmin 21](#_Toc182911983)

[6. táblázat Étlap tábla – Forrás: phpMyAdmin 22](#_Toc182911984)

[7. táblázat Hozzávaló tábla – Forrás: phpMyAdmin 22](#_Toc182911985)

[8. táblázat Napirendelés tábla – Forrás: phpMyAdmin 23](#_Toc182911986)

[9. táblázat Rendelés tábla – Forrás: phpMyAdmin 23](#_Toc182911987)

[10. táblázat Település tábla – Forrás: phpMyAdmin 24](#_Toc182911988)

[11. táblázat TételesRendelés tábla – Forrás: phpMyAdmin 24](#_Toc182911989)

[12. táblázat User tábla – Forrás: phpMyAdmin 25](#_Toc182911990)

**Kódjegyzék**

1. kódrészlet Struktúra 26

2. kódrészlet Ideiglenes étlap létrehozása 27

3. kódrészlet Rendelések lekérdezés 27

4. kódrészlet Mai rendelések lekérdezés 28

5. kódrészlet User rendelései lekérdezés 28

# Bevezetés

Informatikai tanulmányaimat Tatabányán, a TSZC Bánki Donát - Péch Antal Technikumban kezdtem, itt megismerkedtem a többek között a weboldalak készítésének fortélyaival. Ezt a tudást felhasználva, az egyetemi oktatással kiegészítve, és egy ismerős által felkínált lehetőség alapján esett a választásom a szakdolgozatom témájára.

Témám a mai világban azt gondolom elengedhetetlen, hogyha ma valamire szükségünk van, rákeresünk, megrendelésünket leadjuk, és hamar hozzá is jutunk. A téma kapcsán, melyben az ételkifőzde által készített napi menü és a rendelések kiszállítása hagyományos módszerekkel történne, nem lenne hatékony. Régóta fennálló, jól ismert helynél lehetséges, de egy újonnan induló helynek már reklámozása sem egyszerű, a mérhetetlen papírmunkáról, az adatok hagyományos módon való tárolásának hátrányairól nem is beszélve. Az elmúlt időben figyelemmel követtem, hogy ezt főként egy személyben megoldani hosszútávon lehetetlen feladat, az ügyfélkör növekedésével valamilyen információs rendszer használata megkerülhetetlen az ételkifőzde számára.

A folyamatosan fejlődő technológiai világban és az egyre növekvő igényeknek köszönhetően, ma már szinte elengedhetetlen a digitális adatkezelés, -tárolás, automatizálás. Megoldási javaslatom egy, a mai követelményeknek eleget tevő reszponzív weboldal, az általam tervezett adatbázissal támogatva.

A majd elkészülő rendszer használatának megtanítására adminisztrátor, vagy felhasználói szinten is szükség lehet. Eme probléma áthidalására készítek minden folyamatra részletekig kiterjedő oktató anyagot, felhasználói útmutató formájában.

# Ételkifőzde bemutatása

Az ételkifőzde, egy olyan nem rég indult egyéni vállalkozás, mely célja, hogy adott napra a megrendeléseknek megfelelő mennyiségű ételt előállítson, és azokat házhoz is szállítsa. Előre többnyire egy hétre készít étlapot, melyben egy-egy napra található egy leves, és két különböző főétel, melyekre a megrendelő leadhatja rendelését. Az étlap az ételkifőzdére jellemző dizájnt alkalmazza, emellett látszódik rajta az ételek megnevezése és mellettük egy mező, melybe a rendelt mennyiséget írhatja be, aki rendelni szeretne.

## Egy nap az ételkifőzdében

Reggel korán kezdődik a munka a napi rendelések átnézésével, ezek alapján a leltár áttekintésével. Ezután, hogy ebédidőben házhoz szállításra kerüljenek az ételek, megkezdődik a főzés a konyhán, majd a kiszállítás. A kemény munka délután érne véget, de ilyenkor még hátra van a papírmunka, raktár utánpótlása, és a másnapi megrendelések áttekintése.

A vállalkozásban jelenleg egy személy végzi a rendelések felvételét, kezelését, az ételek készítését, a leltárazást és a kiszállítást is. A jelenlegi rendszerben a megrendelői kör hetente kapja meg a következő heti menüt, majd Facebook üzenet formájában kitöltve adják le rendelésüket. A kiszállítás területe Tatabánya és környéke.

# Üzleti igény definiálása

Az egyetlen jelenlegi dolgozóval és egy rendszeres felhasználóval folytatott megbeszéléseim, és az általuk támasztott elvárások alapján felmértem, hogy melyek azok a paraméterek, funkciók, amelyeket fontosnak tartanak a megrendeléseket és étlapokkal kapcsolatos lekérdezéseket lebonyolító nyilvántartó rendszerben.

A felhasználó által állított probléma az volt, hogy számára kényelmetlen hetente fényképezni és küldeni az étlapot, megfelelőbb lenne egy felület, ahol egyszerűen kiválogathatja, amit és amely napra szeretne rendelni.

Ami nyilvánvalóan érződött, hogy a mostani rendszerben a megrendelések kezelése, húsz, maximum harminc fős ügyfélkörig tűrhető, de használata igen nehézkes több szempontból is. Első fontos szempont az adat vesztés kikerülése, Facebook üzenet formájában, nem egy helyre összegyűjtve kész csoda, hogy nem mindennapos, hogy valakinek esetleg nem-, vagy nem megfelelő étel kerül kiszállításra. Tehát mindenképpen szükség van egy egységes felületre, ahol a napi feladatok láthatóak.

A másik kiemelt szempont, hogy valamilyen módon segíteni kellene a házhozszállítási folyamatot, néhány cím esetén, amelyek esetleg még a konyhához közel is találhatóak, ez nem jelent problémát, de itt előfordul, hogy meg kell tervezni az utat, hogy az ételek ne hidegen érkezzenek házhoz, vagy hogy néhány címnél a fizetés kártyával történik, akkor az eszközt is vinni kell. E célt szolgálva kellene egy felület, ahol egységesen láthatóak a címek a hozzájuk tartozó étellel, fizetési móddal, és ezen felületen lehetne szűrni, városra, vagy akár fizetési módra.

A harmadik fontos terület a leltár, jelenleg a bevásárlás a következő heti tervek alapján történik, majd napi szinten szükség van felügyeletre, a szükségletek alapján, ki kell számolni, hogy adott alapanyagból mennyire van szükségük. Célszerű lenne tehát, egy olyan funkciót létrehozni melyben automatizáltan a rendelések alapján kilistázzuk a napi, és a többheti szükségleteinket is.

## Összegezve

Összességében tehát az igények alapján egy olyan szolgáltatásra van szükség, amely felhasználóbarát, egyszerűen kezelhető és könnyen elérhető, első sorban az ügyfelek számára, hogy ők jól átlátható étlapokból könnyedén kiválasztva tudják leadni rendelésüket, ami automatikusan bekerül rendszerbe. Emellett adminisztrátori oldalról fő irányelv az átláthatóság és az üzleti folyamatok támogatása, a rendelések kezelésében, a kiszállításban és a leltár figyelemmel tartásában.

## SWOT-analízis a jelenlegi rendszerről

. táblázat SWOT-analízis jelenlegi - Forrás: Saját szerkesztés

|  |  |
| --- | --- |
| **Erősségek** | **Gyengeségek** |
| * Házias ízek * Az adatkezelésnek nincs plusz költsége | * Lassú adatkeresés, adat felvétel * Felesleges adminisztráció * Adatduplikáció, elavult adatok * Rendezettlen adatok |
| **Lehetőségek** | **Veszélyek** |
| * Adatok rendszerezése * Egységesítés * Fejlődés, ügyfélkör bővülés * Több munkatárs bevonása | * Adat vesztés * Humán tényező * Adatok illetéktelen kézbe kerülése * Plusz munka * Hiányos leltár |

A SWOT-analízis célja, hogy a vizsgált rendszer erősségeit, gyengeségeit, lehetőségeit, illetve a rá leselkedő veszélyeket bemutatja jól átlátható módon, megfigyelhető rajta a pozitív vagy negatív oldal túlsúlya. Fent látható az 1. táblázatban a mostani rendszerről készült SWOT-analízis.

A jelenlegi rendszer legnagyobb előnye az egyszerűségében rejlik, viszont ez hosszútávon nem kivitelezhető. Ahogy az itt is látszódik, a SWOT-analízis negatív oldala került túlsúlyba, ez alapján a rendszer nem működik megfelelően, fejlesztésre van szükség, hogy a pozitív oldal kerüljön előnybe.

Esetlegesen bővülő ügyfélkört ez a rendszer nem tudna elbírni, a gyengeségekből kifolyólag, nagyon lassú adatáramlásból például előfordulhat, hogy valaki még leadja a rendelését, de azt sajnos már nem lehet kivitelezni amiatt, hogy a leltár nem naprakész, ez egyből negatív felhasználói élményt eredményez. Gyengeség még az adatduplikáció, ez első sorban a napi teendők áttekintésénél fordulhat elő, például egy rendelést kétszer készítünk el, így feleslegesen. Nagy gyengeség még a lassú adminisztráció, melyből következően rengeteg idő telik el papírmunkával feleslegesen.

A veszélyek meghatározása során jelentős hibák derültek ki a jelenlegi rendszerrel kapcsolatban. Az adatok hanyag tárolása miatt a felhasználók bizalmas adatai nincsenek védve esetleges harmadik féltől, illetve az emberi tényezőből adódó hibák is elkerülhetőek lennének megfelelő automatizált folyamatok bevezetésével.

A lehetőségekből látszódik, hogy a problémák megoldása érdekében az adatok rendezése a cél, és az ügyfélkör bővülése, megfelelő változtatások eszközölésével nem akadály.

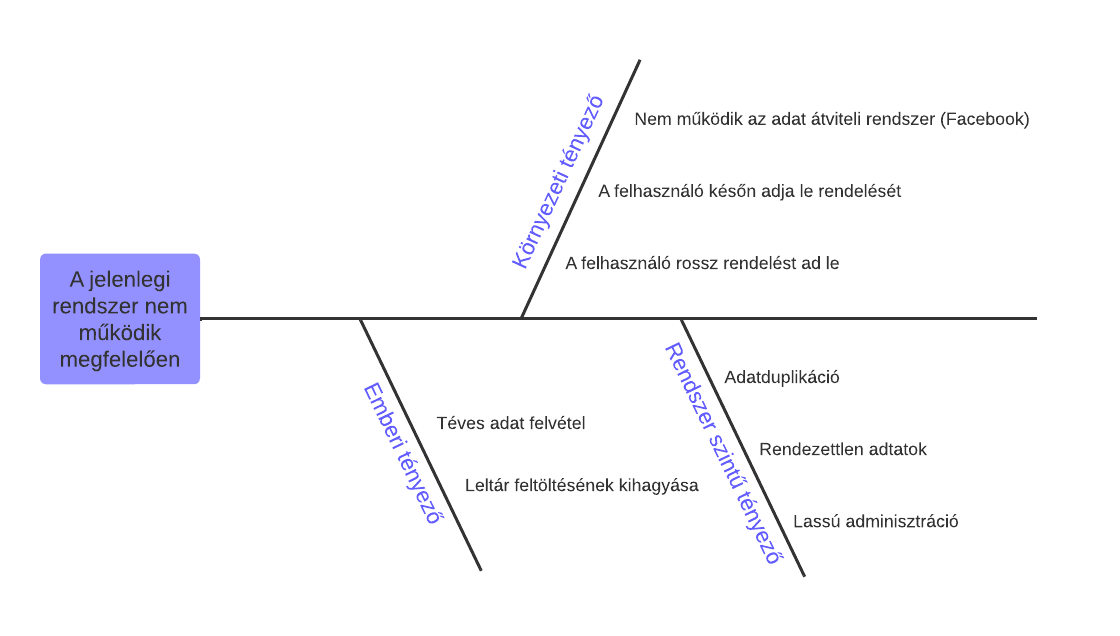
# Az új rendszert előkészítő elemzés

Ebben a fejezetben az új rendszer szükségességét taglalom, a jelenlegi rendszer hibáiból kifolyólag, emellett bemutatok néhány hasonló megoldást a piacon.

## A jelenlegi rendszer hibáinak definiálása

### Ishikawa-diagram

Az Ishikawa-diagram, más néven Halszálka-diagram, egy olyan eszköz, amelyet a rendszer problémáinak azonosítására használnak. Megmutatja, hogyan kapcsolódnak össze az okok és okozatok, és segít elemezni, hogy mi romlik el a rendszerekkel, folyamatokkal és termékekkel. A név Kaoru Ishikawa japán mérnöktől származik, aki az 1960-as években fejlesztette ki a módszert. [1]



. ábra Ishikawa-diagram - Forrás: Saját szerkesztés a Lucidchart eszközzel

Az első ábrán látható módon a problémának a jelenlegi rendszerben több forrása is van, az egyik ilyen az emberi tényező, mely nem helyettesíthető, de munkája könnyíthető és bizonyos folyamatok automatizálásával, ezen hibák minimumra csökkenthetőek.

A következő ilyen tényező a környezeti hatásokat foglalja magában. Ilyen lehet például, egy esetleges Facebook leállás vagy átmeneti kép küldés leállítása a részükről, egy ilyen esemény a jelenlegi rendszerben végzetes hibákhoz vezethet, hiszen az adatáramlás nagy részben ezen a platformon történik. A megrendelők által vétett hibák, környezeti tényezőhöz tartoznak hiszen ezekre nem lehetünk kihatással, mégis veszélyt jelentenek a rendszer gördülékeny működésére nézve, az általuk szolgáltatott téves adat revíziójára nincs, vagy csak túl későn adatik lehetőség.

A harmadik csoport, pedig a konkrét rendszer metodikájából adódó hibákat tartalmazza, az itt felsorolt okok lehetnek, adatduplikáció, rendezettlenség és a lassú adminisztráció, ezen problémákat elkerülendő fejlesztésre van szükség a módszereinken.

## Megoldási javaslatok vizsgálata

Ebben a fejezetben hasonló problémára, mely szerint egy konyhán készített ételeket kiszállítását, előzetes megrendelés alapján kellene lebonyolítani, megoldást nyújtó alkalmazásokat fogok bemutatni.

### Wolt/Foodora

A legközelebb egy egyszerű előfizetéses rendszerhez, ami legkönnyebben beindítható, azok a Wolt vagy a Foodora megoldásai, az étterem, vagy kisvállalkozás egyszerűen regisztrál, majd az ellenőrzés után be is kerülnek a rendszerbe, innentől kezdve érkeznek a rendelések, és futárról sem kell gondoskodni. Az egyértelműen látszódik, hogy ezen eszközök bevezetése egyszerű, viszont nem szolgálnak megoldást a jelenlegi rendszerben előforduló hibák összességére.

Ezzel a megoldással több probléma is adódik, az ételkifőzde módszereibe nem illik bele a rendelés azonnali elkészítése, olyan megoldásra lenne szükség, melynél napokkal előre leadott rendelések hamar látszódnak. Emellett a leltár rendszer támogatására is szükség van.

### iBar leltározó app

Az iBar használatával a jelenlegi rendszer leltár problémáit lehetne orvosolni, könnyedén telefon segítségével a rendszerbe felvihetőek az adatok. Ezen rendszer használatával az a probléma, hogy csak egy aspektusra nyújt megoldást, a leltározás folyamatára, emellett használata havidíjhoz kötött.

### Simpra

Az eddig bemutatott alternatívák közül a Simpra a legkifinomultabb megoldással szolgáló, eszközei közül az adott problémakörhöz legjobban passzoló a vendéglátó szoftver éttermek számára. Ezen módszer választása, amellett, hogy ez is inkább éttermek számára megfelelő, igen költséges választás, viszont nagy előnye a leltár és a rendelések egységesítése és a rendszer telepítése, oktatása.

### Saját rendszer fejlesztése

A bemutatott alternatívákból látszódik, hogy nem igazán található olyan rendszer, mely készen vásárolható és nem éttermi megoldásra terveztek. Akad olyan, mint például a Wolt/Foodora, vagy az iBar, amelyek a problémakör csak egy aspektusára nyújtanak megoldást, vagy fellelhető olyan egészen kifinomult megoldás, mint a Simpra, mely viszont elég költségesnek bizonyul. A fent említett eszközök közül egyik sem kifejezetten az ügyfél igényeihez igazodik, sem árban, sem metodikában.

Megoldási javaslatom, egy saját rendszer készítése, mely ténylegesen, részletekig kiterjedve, a felhasználó igényeihez készül. A probléma minden oldalát egységesen, egy helyen lefedi.

## Hasonló megoldások a piacon

Az alábbiakban néhány általam kiemelt, a készülő rendszerhez hasonló eszközt mutatok be, ezen alkalmazások ügyfélkörben és régióban is eltérnek a vállalkozástól. Ezeken az oldalakon tett látogatásaim során láttam üzleti modelljük hasonlóságát, emellett néhány kiemelten fontos funkciót melyet a készülő rendszerbe is feltudok használni.

### Egészségpajzs

Az Egészségpajzs főként Budapesten és környékén tevékenykedő vállalkozás, mely célja, hogy hazai ízeket vigyenek az otthonokba és mindezt egészséges hazai alapanyagokból készítsék. [2]

### e-food

Az e-food egy országosan több nagyvárosban és környékükön is jelen lévő alkalmazás, mely hasonlóan a készülő rendszerhez konyhák rendelési és kiszállítási rendszerét bonyolítja le. [3]

# Funkcionális és nem funkcionális követelmények meghatározása

## Funkcionális követelmények

Elvárás, hogy a rendszer képes legyen ételeket, étlapokat, felhasználókat és megrendeléseket nyilvántartani, őket egymáshoz rendelni.

A felhasználó csak belépés (regisztráció szükséges) után legyen képes megrendelés leadására, saját rendelés megtekintésre és változtatásra vagy lemondásra. A regisztrációnál szükséges adatok: e-mail, név, telefonszám, jelszó, cím adatok.

Legyen adminisztrátori felhasználói szint, mely képes új étel felvételre, étlap létrehozásra, módosításra, leadott rendelések megtekintése, hozzávaló készlet igénykimutatás megtekintésére.

A következőkben a rendszert funkciók szerint alrendszerekre bontva mutatom be.

### Hozzávalók, ételek, étlapok nyilvántartása

A képen szöveg, diagram, sor, tervezés látható

Automatikusan generált leírás

. ábra Hozzávalók, ételek, étlapok USE CASE diagram - Forrás: Saját szerkesztés

Az alrendszer feladata hozzávalók, ételek, étlapok tárolása, az adat hozzáférhetőségének és módosíthatóságának biztosítása. Amint az a 2. ábrán is jól látható, az itt tárolt adatokon módosításokat csak adminisztrátori jogosultsággal rendelkező személy (később: admin) hajthat végre.

### Login rendszer

A képen diagram, szöveg, vázlat, minta látható

Automatikusan generált leírás

. ábra Login USE CASE diagram - Forrás: Saját szerkesztés

Az alrendszer feladata, hogy az oldalra érkező jogosultságait akképpen formálhassuk, hogy milyen fiókba jelentkezett be. Regisztráció után a felhasználó beírt adatait mentjük a rendszerbe. Admin jogosultságot nem lehet regisztrációval szerezni, ilyen kiemelt felhasználó csak egy van a rendszerben. Ez az alrendszer kezeli az oldalra való beléptetést, mely során ellenőrizzük a felhasználó által beírt adatokat, ha megfelelő adatokat adott meg, beléptetjük a rendszerbe.

### Rendelés

Amikor a vevő megtekinti az étlapokat automatikusan létrejön az üres „kosara”. Ebbe a virtuális kosárba az étlapokról helyezhet be, három hétre előre létrehozott dátumokra ételeket, minden napra három különböző ételből választhat, akár mindegyikből többet. A kosarát bármikor megtekintheti, változtathatja az ételek mennyiségét, illetve törölhet tételt a kosárból, ha pedig már elégedett a kosárral, rendelését leadhatja.

A leadott, még ki nem szállított tételekkel rendelkező, ki nem fizetett rendeléseit megtekintheti a megrendelő.

A képen diagram, szöveg, sor, minta látható

Automatikusan generált leírás

. ábra Rendelés USE CASE diagram - Forrás: Saját szerkesztés

A 4. ábrán látható módon a rendszerben értelmezhető egy harmadik típusú felhasználó, a „Látogató”, az oldalra nem bejelentkezett személy is rendelkezhet a kosarával, megtekintheti az étlapot, de rendelését csak regisztráció, illetve bejelentkezés után adhatja le.

### Adminisztratív folyamatok támogatása

A képen szöveg, diagram, sor, képernyőkép látható

Automatikusan generált leírás

. ábra Admin folyamatok USE CASE diagram – Forrás: Saját szerkesztés

Az alrendszer feladata, hogy az admin számára megjelenítendő információkat mindig pontosan mutassa ki.

Ilyen felület például a rendelés megtekintés, amely magában foglalja a leadott rendelések vizsgálatát, köztük való keresést szűrő alapján, ha a kiszállítás és kifizetés megtörtént, azt is ezen a felületen lehet felvinni a rendszerbe.

Ezen kívül ide tartozik még a háromheti szükséglet kimutatás, mely jelentősen megkönnyíti a leltározás folyamatát, rendszeren a rendelések alapján automatikusan kiszámolja, hogy melyik alapanyagból mennyire van szükség.

A harmadik fontos admin támogatási terület a napi teendők kimutatása, melyen jól elkülönítve megtalálható, hogy a mai napon mely ételből hány adagot kell elkészíteni és kiszállítani tovább csökkentve ezzel a hibák előfordulásának lehetőségét.

## Nem funkcionális követelmények

Az étlap felületen az ételek nevei mellett fel kell tüntetni az étellel kapcsolatos különböző fontos információkat, hozzávalók, az allergének megjelölve, átlagos tápérték, nettó tömeg.

Biztosítsa, hogy az adatok ne kerülhessenek illetéktelen felhasználó kezébe. Mindenki csak azokat az adatokat láthassa, vagy módosíthassa, ami a feladatköréhez kötődik. Ne tároljon adatvédelmi szempontból nem megengedett adatot. A kezelőfelület használata ne igényeljen informatikai előismereteket.

## SWOT-analízis az új rendszerről

A 2. táblázatban látható új SWOT-analízis azt hivatott bemutatni, hogy az új rendszer elkészülése miként javítaná a régi megoldáshoz képest a rendszer működését.

. táblázat SWOT-analízis új - Forrás: Saját szerkesztés

|  |  |
| --- | --- |
| **Erősségek** | **Gyengeségek** |
| * Jól áttekinthető * Könnyen elérhető, módosítható adatok * Az ügyfél módosíthat rendelésén * Elenyésző adatkezelési probléma * Több platformon elérhető | * Szükséges az új rendszer megtanulása |
| **Lehetőségek** | **Veszélyek** |
| * Fejlődés, ügyfélkör bővülés * Több munkatárs bevonása | * Humán tényező * Személyes adatok rossz kézbe kerülése |

A fent látható 2. táblázatban látható, hogy a fejlesztéseknek és a folyamatok automatizációjának köszönhetően a pozitív oldal került túlsúlyba, így a rendszer megfelelően működik.

Az új rendszer fontos erőssége az áttekinthetőség, és hogy az adatok rendezett módon vannak tárolva elkerülve ezzel az adatvesztést, emellett a megrendelők számára is ez egy felhasználóbarát megoldás, melyben egyszerűen megtekinthetik, vagy módosíthatják rendelésüket is, akár telefonjukról vagy asztali eszközükről egyaránt.

Figyelembe kell venni azonban az új megoldás azon gyengeségét, hogy újdonsága révén a használatát tanulni kell, felhasználói oldalról más rendszerekhez hasonlóan használata intuitív, a szakdolgozatomban később kifejtett oktató anyag erre a területre is részletesen kitér. Másik oldalról viszont, adminisztrátori szinten használata lényegesen bonyolultabb a számos funkciónak köszönhetően, de a használati útmutatóban leírtakat követve, az új funkciók használatával jelentősen javul a rendelések, a leltár és a kiszállítások kezelése is.

A veszélyek vizsgálata során kiemelendő az emberi tényező, ez alól egy rendszer sem kivétel, amelyet emberek használnak, hiszen előfordulhatnak tévedések, hibás adat bevitel, melyek problémákhoz vezethetnek, az új rendszerben alkalmazott fejlesztéseknek köszönhetően ezek száma jelentősen csökken. A másik veszély a személyes adatok harmadik félhez kerülése, jelszóval védett felületen, adminisztrátori jogosultsági szinthez kötött adatokhoz jóval kisebb valószínűséggel juthat hozzá rossz szándékú személy, viszont ez a lehetőség továbbra is adott.

A lehetőségekből látszódik, hogy új munkatárs bevonásával fejlődhet a rendszer, mely személy betanítása az oktató anyagokkal történhet, illetve az új rendszernek köszönhetően, esetleges új felhasználók könnyebben elérhetik az ételkifőzde kínálatát.

# Fejlesztés ütemterve - Gantt diagram

A következő táblázatban látható a megvalósításhoz szükséges ütemterv Gantt digramon ábrázolva. Látható rajta az egyes feladatokhoz tartozó kezdő hét (K) és végső hét (V). Egy ilyen nagy projekt esetén elengedhetetlen a pontos terv elkészítése, még ha ez nem is teljesen kerül betartásra, jó útmutatót ad a munkafolyamat kézben tartása érdekében.

. táblázat Gantt diagram - Forrás: Saját szerkesztés

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tevékenység** | | | | **Hetek** | | | | | | | | | | | | | |
| Megnevezés | K | V | Hossz | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| **Tervezés** | 1 | 5 | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Specifikáció elemzés | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Adatmodell tervezése | 4 | 4 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fejlesztői eszközök kiválasztása | 4 | 4 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Megvalósítás** | 5 | 12 | 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Konzultáció a felhasználóval | 5 | 5 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Konzultáció a felhasználóval | 9 | 9 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Konzultáció a felhasználóval | 12 | 12 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Tesztelés** | 5 | 14 | 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Dokumentáció | 1 | 14 | 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Ahogy az a 3. táblázatban is jól látható a megvalósítás folyamata a tervezéssel kezdődik, a specifikáció alapján kialakul a konkrét megvalósítandó funkciók köre. Ezen tervezési folyamat része az adatmodell kialakítása is melyre épül az adatbázis.

A fejlesztési eszközök kiválasztása után megkezdődhet a fejlesztés, ennek sikeressége a megfelelő kommunikáción múlik a felhasználóval, ennek érdekében a fejlesztés során többször is konzultáció kerül megrendezésre.

A diagramból látszódik, hogy a folyamat szerves része a tesztelés, ez magában foglalja az egység-, iterációs- és rendszer teszteket is a fejlesztés ideje alatt. Ez a lépés elengedhetetlen, hogy végül megfelelő minőségű produktum kerüljön a felhasználókhoz. Megfigyelhető, hogy a tesztelési folyamat egészen az ütemterv végéig tart, ez azért fontos, hogy az elkészült rendszeren is végezzünk felhasználói teszteket, mely célja, hogy minden aspektusában a lehető legjobban kivitelezett eszköz kerüljön a végfelhasználókhoz.

Ami még látható a diagramon, hogy dokumentáció, a teljes folyamat alatt kiemelt szerepet kap, ez azért fontos, hogy ha esetleg, későbbi fejlesztésre lenne szükség, vagy ha valaki kíváncsi a program belső működésére, egyszerűen, jól átlátható módon tanulmányozhassa lényeges aspektusaiban.

# Alkalmazott fejlesztői környezet

## XAMPP

A XAMPP egy nyílt forráskódú szoftvercsomag, melynek általam is használt moduljai az Apache szerver és a MySQL, vagy MariaDB. Ez az eszköz tette lehetővé webes alkalmazásom futtatását, tesztelését, emellett az általam választott adatbázis kezelő használatát is támogatta. [4]

## Visual Studio 2022

A fejlesztés legnagyobb részében ezt alkalmaztam, a megfelelő kiegészítőkkel a feladatnak tökéletesen megfelelt ingyenes megoldás, sok funkciója nagyban megkönnyíti és felgyorsítja a fejlesztés folyamatát. A fejlesztés közben böngészővel párhuzamosan futtatva könnyedén alkalmaztam gyors javításokat hibák estén, jól navigálható átlátható a felülete.

## PhpMyAdmin

A képen vitorlás hajó, hajó, fehér, közlekedés látható

Automatikusan generált leírás

. ábra phpMyAdmin – Forrás: https://www.phpmyadmin.net

A phpMyAdmin egy ingyenes szoftver, melyet MySQL adatbázisom kezelésére használtam, kezelőfelülete intuitív, használata könnyen elsajátítható. A rendszer adatbázisának megalkotása ezen a felületen keresztül történt, emellett a dokumentációs folyamatokban is támogat, adatbázis szerkezet és táblák szintjén is. Lehetőség van SQL parancsok közvetlen futtatására, mely eszközt tesztelésre előszeretettel alkalmaztam. [5]

# Adatmodell tervezése és megvalósítása

Az adatmodell megtervezése, során fel kellett használom az előzetes terveket, hogy pontosan tudjam, hogy milyen adatok tárolására lesz szükségem. A pontos tervezés biztosította, hogy az elkészülő adatszerkezet megfelel a normálformáknak. Emellett a rendszer minden adatszükségletét lefedi

## Az adatbázis szerkezete kapcsolatokkal

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

. ábra Az adatbázis szerkezete – Forrás: phpMyAdmin

Az adatbázis tervezésénél és megvalósításánál kiemelten fontos odafigyelni arra, hogy megfeleljen a normálformáknak, hogyha megfelelő adatbiztonságok szeretnénk és elkerüljük a tárolásból és adatfelvételből adódó hibákat, mint például a redundancia, ineffektivitás, emelett biztosítsuk a skálázhatóságot.

### Normálformák

Az első normálforma alapján, minden tábla atomi (oszthatatlan) attribútumokat tartalmaz. Nincsenek ismétlődő csoportok, ami azt jelenti, hogy minden attribútum csak egyetlen értéket tartalmaz. Minden táblában egyértelműen megjelenik egy elsődleges kulcs (a kulcs szimbólum jelzi az attribútum mellett), amely biztosítja, hogy minden rekord egyedi legyen.

A 2NF követelménye, hogy minden nem kulcs attribútumnak teljesen függenie kell az elsődleges kulcstól. Olyan táblák esetében, mint például a user, rendeles, és etel, minden nem kulcs attribútum láthatóan teljesen függ a saját táblája elsődleges kulcsától. A táblák közötti kapcsolatok idegen kulcsok (pl. tid, eid) segítségével vannak kezelve, ami szintén támogatja a 2NF elvet.

A 3NF azt követeli meg, hogy ne legyenek tranzitív függőségek (azaz a nem kulcs attribútumoknak csak az elsődleges kulcstól kell függeniük).

A táblák közötti kapcsolatok egyértelműek. Például a rendeles tábla a userId segítségével kapcsolódik a user táblához, míg a tetelrend tábla a rendeléseket (napId) köti össze az ételekkel (eid). Az idegen kulcsok használatával biztosított a referenciális integritás, ami megelőzi az árva rekordok megjelenését az adatbázisban.

## Adatbázistáblák bemutatása

A következő táblázatokból, melyeket a már létrehozott táblákról állítottam össze a phpMyAdmin megtekintési funkciója segítségével, fontos információk nyerhetők ki. Balról haladva a „Name” oszlopban az azonosítók és tulajdonságok megnevezésé látható, a következő a „Type”, vagyis típus, mely a tárolt adatok típusára vonatkozik. A „Collation” oszlop a karakteres típusú mezők kódolását jelöli. A „Null” oszlop azt hivatott bemutatni, hogy az adott attribútumnál engedélyezett-e null érték.

Ezen kívül kiemelném még a „Default” és az „Extra” oszlopokat, előbbi alapértelmezett értéket állít be, ha esetleg adatfeltöltésnél ilyen értékkel ellátott új értéket veszünk fel, oly módon, hogy ezen értéknek nem határozunk meg semmit, akkor az alapértelmezett értékkel töltjük fel. Ezt például a 9. táblázatban láthatjuk élesben, ahol ugyanis rendeléseket tárolunk. Rendelés mentésénél nem a felhasználótól várjuk, hogy beírja a pontos időt, ennek alapértelmezett értéke a jelenlegi idő. Az utóbbi, „Extra” oszlopban pedig látszódik, hogy bizonyos azonosítóknál automatikusan növeljük az értéket, ennek is új adat felvételnél van létjogosultsága.

### Étel tábla

. táblázat Étel tábla – Forrás: phpMyAdmin

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

Az Étel tábla a rendszerben található ételeket tárolja, elnevezésük és nettó tömeg adatokkal.

### ÉtelHozzávaló tábla

. táblázat ÉtelHozzávaló tábla – Forrás: phpMyAdmin

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

Az ÉtelHozzávaló tábla ételeket kapcsol hozzávalókhoz, oly módon, hogy egy ételhez annyi sor tartozik ahány hozzávaló kell hozzá, mindegyiknél feltüntetve a mennyiség.

### Étlap tábla

. táblázat Étlap tábla – Forrás: phpMyAdmin

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

Az Étlap tábla minden egyes étlapra előkészített dátumot tartalmazza, dátumokhoz rendelünk háromfajta ételt, amiből a megrendelők választhatnak.

### Hozzávaló tábla

. táblázat Hozzávaló tábla – Forrás: phpMyAdmin

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A Hozzávaló tábla néhány információt tartalmaz az egyes hozzávalókról, mely szükséges fogyasztó védelmi szempontból, például „allergen” meghatározza, hogyan kell feltűntetnük az adott hozzávalót. Illetve leltár szempontjából, például „megys” ez a hozzávaló mértékegységét tartalmazza, mely fontos megjelenítés szempontjából a leltár támogató felületen.

### Napirendelés tábla

. táblázat Napirendelés tábla – Forrás: phpMyAdmin

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

A Napirendelés tábla a rendeléseket napokra bontja, ennek az a célja, hogy tudjuk, hogy adott megrendelőnek melyik napra kell kiszállítást végezni rendelése alapján.

### Rendelés tábla

. táblázat Rendelés tábla – Forrás: phpMyAdmin

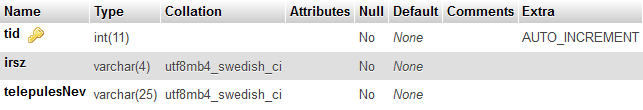
A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

A Rendelés tábla a felhasználó által leadott rendelést reprezentálja, felhasználóhoz kapcsol, dátumot , fizetési adatokat tárol. A rendszerben egy rendelés úgy épül fel, hogy mindegyikhez tartoznak napok, a napokhoz pedig tételek, ezeket a Napirendelés, illetve TételesRendelés tábláknál kifejtem.

### Település tábla

. táblázat Település tábla – Forrás: phpMyAdmin



A Település táblának a regisztrációnál van szerepe, csak olyan településre lehet regisztrálni, amely már megtalálható a rendszerben, ennek kiszállítás szempontjából vannak korlátai azért kellett meghatározni.

### TételesRendelés tábla

. táblázat TételesRendelés tábla – Forrás: phpMyAdmin

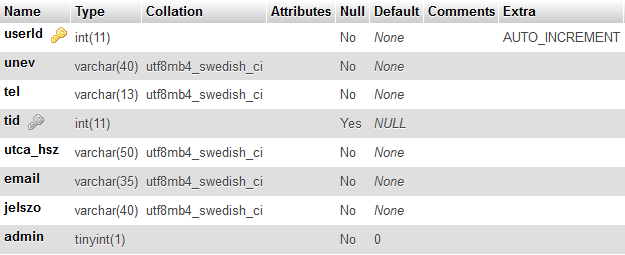
A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

A TételesRendelés tábla határozza meg, hogy adott rendeléshez, adott napra konkrétan milyen ételeket és mennyit kell kiszállítani. Minden sor egy rendelés, egy napjának, egy ételét reprezentálja.

### User tábla

. táblázat User tábla – Forrás: phpMyAdmin



A User tábla, felhasználóink regisztrációs adatait tárolja, melyeknek több célja van.

Autentikációs célok, ilyen az „email”, „jelszo” és „admin”, az admin attribútum határozza meg, hogy egy felhasználó admin-e.

Kisszállítási célok, „tid” és „utca\_hsz”, a „tid”-re itt azért volt szükség, mert a magyar irányítószám rendszert kikerülve tároljuk a településeinket, azzal a céllal, hogy egyértelműen azonosítsuk a településeinket.

Leíró adatok, „unev” és „tel” ezeknek a felhasználó elérésében, van szerepük, ha valamilyen változás van, vagy fontos információ, mely érint egy felhasználót könnyen fel tudjuk venni vele a kapcsolatot.

# A rendszer fejlesztése

## MVC minta

A Model-View-Controller irányelveit alkalmaztam a fejlesztés során, mely lényegében három területre bontja a rendszert. Működés szempontjából a felhasználói kérdésekre egy controller válaszol a megfelelő view választva és feltölti a szükséges model adatokkal.

Ez a felbontás kódolás szempontjából megkönnyíti a skálázhatóságot, komplikáltság kezelését, könnyeb a tesztelés, hibakeresés, ha ezen területeken jól elkülönülnek. Például, ha a megjelenítés és üzleti logika egy objektumban található, akkor, ha felületet változtatni szeretnék ebbe az objektumba kell belenyúlni, ezáltal hibák adódhatnak az üzleti logikában egy minimális változtatásnak köszönhetően, ez elkerülhet az MVC pontos alkalmazásával.

Az elkészült rendszer megfelel az MVC irányelveinek, a model reprezentálja az üzleti logikát, a view feladata a megjelenítés, csak minimális logika található ezekben és a controller végzi a view feltöltését adatokkal. [6]

## Néhány kiemelt kódrészlet

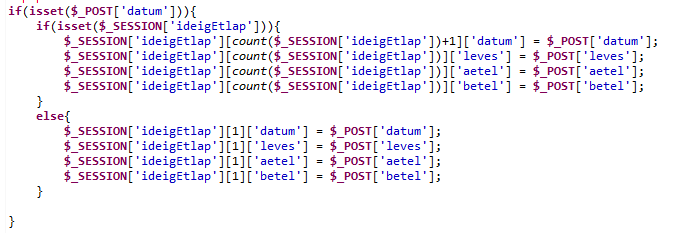
. kódrészlet Struktúra

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

Az MVC minta alapján a controller mappában lévő fájlok az első kódrészleten látható módon kerülnek meghívásra, „$action”-ként tároljuk a felhasználók lehetőségeit, itt kezeljük az adminisztrátor megkülönböztetését is.

. kódrészlet Ideiglenes étlap létrehozása



A második kódrészletben látható, hogy az első bejövő étlap elem bekerül a tömb első helyére, megfelelő tulajdonságok a megfelelő helyre onnantól pedig következő elemek folytatólagosan kerülnek bele a tömbbe. A tömb az oldal elhagyásával, vagy rendelés leadással törlődik.

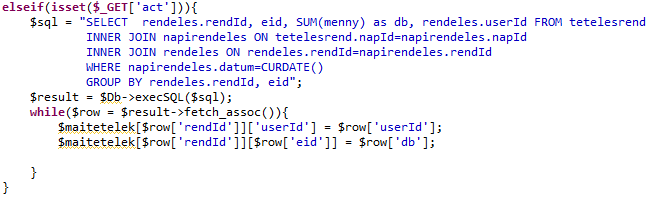
. kódrészlet Rendelések lekérdezés

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

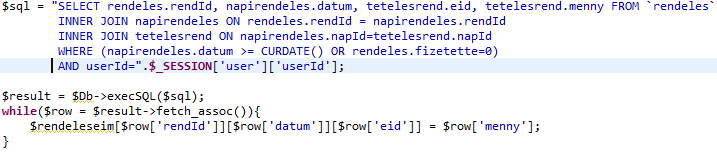
Ha rendelések kifizetése felületet szeretnénk megjeleníteni az adatbázison a harmadik ábrán látható SQL lekérdezést kell lefuttatnunk, ha keresőmező is be van állítva akkor azt a feltételt is itt fűzzük hozzá. A kapott értéket mentjük a tömbbe így a view fájlon az már könnyen kezelhető.

. kódrészlet Mai rendelések lekérdezés



Ha felhasználó a mai rendeléseket szeretné megtekinteni a fent látható SQL lekérdezés az összes számára megjelenítendő információval visszatér ezeket mentjük a kódrészlet alsó részén látható módon a tömbbe.

. kódrészlet User rendelései lekérdezés



A követelmény specifikációban meghatározottak alapján az alapszintű felhasználónak szüksége van egy felületre, melyen megtekintheti és lemondhatja eddigi rendeléseit. A ötödik kódrészleten látható SQL lekérdezés visszatér az adott user jelenleg futó, ki nem fizetett rendelésivel, ezeket rendezzük a „rendeleseim” tömbbe, ebben tároljuk a rendelésre vonatkozó adatokat és az ahhoz tartozó dátumokat és a dátumokhoz tartozó tételeket.

## Tesztelési dokumentáció

A tesztelés minden fejlesztési folyamat elengedhetetlen része, mivel ez biztosítja a rendszer hibamentes működését és a kívánt funkcionalitás elérését. Egy nem megfelelően tesztelt rendszer számos problémát eredményezhet, például hibás működést, biztonsági réseket, vagy rossz felhasználói élményt, amelyek végső soron a rendszer korai elavulásához vagy használaton kívül helyezéséhez vezethetnek.

A tesztelési folyamat célja, hogy a fejlesztés során keletkező esetleges hibákat még az éles üzembe helyezés előtt feltárja és kijavítsa. Ez nemcsak a rendszer megbízhatóságát növeli, hanem a hosszú távú fenntarthatóságot is biztosítja. A tesztelési dokumentáció fontos szerepet játszik abban, hogy a tesztelés folyamata átlátható legyen, és bármikor visszakövethető legyen az elvégzett tesztek eredménye.

A következő alfejezetekben bemutatom az általam alkalmazott tesztelési eljárásokat, beleértve az egységteszteket, az integrációs- és rendszerteszteket, valamint a normál és extrém tesztesetek kezelését. Ezek mind hozzájárultak ahhoz, hogy a rendszer megbízhatóan és hibamentesen működjön a különböző felhasználói forgatókönyvek során.

### Egységteszt alkalmazása

A rendszer fejlesztési életciklusában először megjelenő tesztelési forma az egységteszt melyet a komplex program fontosabb moduljaihoz kell elkészíteni.

Az egységteszt a metódusokat teszteli, adott paraméterekre ismerjük a metódus visszatérési értékét (vagy mellékhatását), megvizsgálja, hogy a tényleges visszatérési érték megegyezik-e az elvárttal. Az általam alkalmazott egységtesztek nem tértek vissza nem várt értékekkel, így elmondható, hogy a tesztek sikeresek voltak. Ezeket a teszteket a regressziós tesztelési elvnek megfelelően, újabb modulokkal való együttes tesztelés előtt újra lefuttattam metódusaimon.

### Integrációs- és rendszerteszt alkalmazása

Az integrációs teszt alkalmazásánál több együtt működő modellt tesztelve nem alakult ki futást gátló hiba, a folyamatok nem generáltak egymást gátló hibát, ugyan ez elmondható rendszertesztnél, mely során a teljes rendszert együtt működtetve nem történt semmi rendellenes a folyamatok tesztelése közben.

Az MVC minta alapján történő fejlesztés során, az inkrementális módszer volt a legalkalmasabb a fejlesztésre, hiszen a külön egységek egymástól függetlenül fejleszthetőek és egy-egy újonnan elkészített és letesztelt funkciót egyből integrálhatjuk a rendszerbe és tesztelhetjük az együttműködést.

### Normál- és extrém tesztesetek kezelése

Az egyes modulokon lefutatott tesztelés során a beérkező adatokat megfelelően kezelte, felhasználó nem tud rendszer meghibásodást okozó adatot megadni, illetve, ha a program által kezelhető adatot szolgáltat, azt az oldal megfelelően kezelni tudja.

### Összegezve

A tesztelési folyamat során számos olyan hibát fedeztem fel, melyeket nem, vagy csak a fejlesztés sokkal későbbi szakaszában vettem volna figyelembe így ezen technikáknak köszönhetően a fejlesztési költségek nagyban lecsökkentek, hiszen minnél később derül fény egy adott hibára annál több egység kapcsolódhat rá, használhatja fel így ezek javítása megelőzhető.

Ebben a folyamatban megtanultam, hogy a funkcionális, illetve felhasználói interfész tesztelése egyaránt, elengedhetetlen ahhoz, hogy megfelelő minőséget tudjunk garantálni a későbbi felhasználók számára.

# Felhasználói dokumentáció

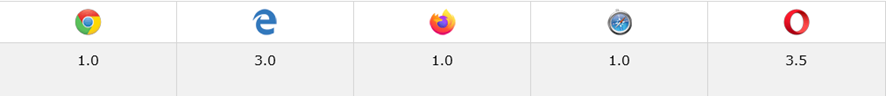
Ebben a fejezetben részletesen kitérek az új rendszer elkészült elemeire, mechanizmusainak elsajátítására.

## Rendszerkövetelmények

Minimális hardver követelmény:

* CPU: Intel® Core™ 2 Duo E6600 vagy AMD Phenom™ X3 8750
* RAM: 2GB
* OS: Windows 7

Szoftver követelmények:



. ábra Szoftver követelmények

Az oldal reszponzív, tehát mobil-, illetve asztali eszközön is megfelelően működik minden jól látható módon van megjelenítve. Később az oktató anyagban is látható képernyőképeken.

## A program általános specifikációja

A weboldal képes lebonyolítani rendelések fogadását regisztrált felhasználóktól, az alap szintű felhasználók rendelésüket leadhatják az admin által létrehozott étlapokon, ezen kívül saját eddigi megrendeléseiket megtekinthetik, illetve lemondhatják.

Admin felhasználói szinten az oldalon adatbázisához felvehető: hozzávaló, étel, étlap, ételhez rendelhető hozzávaló és kép. Valamint megtekinthetőek a felhasználók által leadott rendelések, ugyan ezen a felületen látható a következő háromheti alapanyag szükséglet, kilistázhatók a teendők bármilyen dátumra a rendelések alapján és a mai kiszállítási címek az oda tartozó tételekkel együtt. Rendszerbe fel lehet venni további településeket egy dedikált felületen.

## Oktatóanyag a rendszer felhasználói számára

Ahhoz, hogy egy új, az ezt megelőző rendszertől ennyire eltérő, eszközt a felhasználók, tényleg alkalmazzanak, szükséges az új eszközök használatának elsajátítása. A régi, jól megszokott metodikák itt máshol, más formában elérhetőek, elengedhetetlen tehát, hogy megismerjék a program használatát. Az új funkciók garmadája elsőre rémisztőnek tűnhet, de ahhoz, hogy a rendszer használata gördülékenyen menjen és ezen funkciók kihasználásra kerüljenek, célszerű ezeket elsajátítani, felhasználói és adminisztrátori szinten egyaránt.

### Első lépések

Az oldalra látogatót először a start felület fogadja, ahol a kiszállításon felüli szolgáltatásokról olvashat információkat, illetve elkészült fogásokról láthat illusztrációkat.

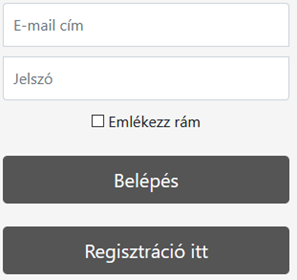
A képen Betűtípus, Grafika, Grafikus tervezés, tervezés látható

Automatikusan generált leírás

. ábra Menürendszer – Forrás: képernyőkép

Erről a felületről a 9. ábrán látható menürendszeren keresztül navigálhat tovább. Az ikonra kattintva a felhasználó visszakerül a start felületre. Az „Étlap” menüpontot választva a felhasználó megtekintheti a következő három heti étlapot.

„Belépés” fülön pedig a látogató bejelentkezhet saját, már létrehozott fiókjába, megadva bejelentkezési adatait a megjelenő ablakon. Az „Emlékezz rám” jelölő négyzetet bepipálva az oldal újbóli felkeresésekor bejelentkezve marad fiókjába.



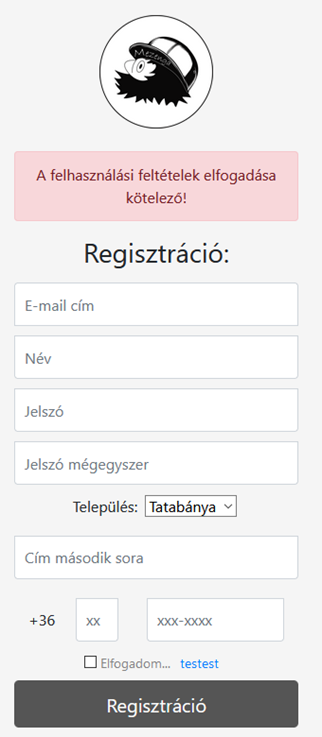
10. ábra Belépés – Forrás: képernyőkép

### Regisztráció

A felhasználó a belépési felületen lévő „Regisztráció itt” gombra kattintva, navigálhat át a 11. ábrán látható regisztrációs felületre. A felső ikonra kattintva vissza kerülhet a látogató a start felületre.

A regisztrációs adatok megadása kötelező a következő sorrendben: e-mail cím, vezetéknév és keresztnév, jelszó, jelszó még egyszer, település a legördülő listából kiválasztva, a cím második sora, azaz utca, házszám, emelet stb., majd telefonszám két szekcióban, első szolgáltatóhoz tartozó két számjegy, a második a telefonszám befejezése, ezek megadása a jelölt formátumban kötelező. Ezután a felhasználási feltétek elfogadása is kötelező.

Ha sikeres a regisztráció a felhasználó a belépési felületre kerül majd a megfelelő jelszó, e-mail cím párossal bejelentkezhet az újonnan létrejött fiókba. Sikeres bejelentkezés után a menüben megjelenik egy új funkció a „Rendeléseim”, melyről a későbbiekben lesz szó.



. ábra Regisztráció – Forrás: képernyőkép

### Rendelés menete

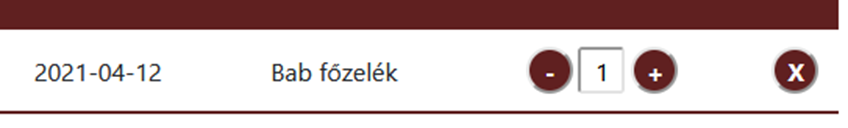
A rendelés menete a következő, először a felhasználónak a menürendszerben korábban látott módon most az „Étlapok” lehetőséget választva kell navigálnia az alábbi felületre. Ahol a megfelelő napon, a megfelelő ételnél a „+”, illetve „-” helyezhet, távolíthat el tételeket a kosárból.

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás

. ábra Étlapok – Forrás: képernyőkép

Ahol az étlapon a „szünnap” felirat látható arra a napra előre láthatólag nem lehet rendelést leadni. Amely napnál pedig az „Előzőnap 16 óráig lehet rendelni” felirat látható, arra napra már nem lehet rendelést leadni, mivel rendelést véglegesíteni csak a kiszállítást megelőző nap 16 óráig lehet.



. ábra Kosár egy tétele – Forrás: képernyőkép

Miután összeállította kosarát a felhasználó a menürendszerben megjelenő „Kosár” gombbal megtekintheti és módosíthatja a „+”, „-” gombokkal vagy törölhet tételt az „x” gombbal.

Ha elégedett a rendelésével véglegesítheti azt. Figyelem, rendelést csak regisztrált, bejelentkezett felhasználó adhat le, ha ennek ellenére elkészíti kosarát az bejelentkezés után veszik el, de csak utána véglegesítheti.

### Alapszintű felhasználó eszközei

Bejelentkezett alapszintű felhasználó a rendeléseim menüpontban megtekintheti, illetve lemondhatja korábbi rendeléseit a 14. és 15. ábrán látható asztali vagy telefonos felületen.

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás

. ábra Rendeléseim felület asztali – Forrás: képernyőkép

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

. ábra Rendeléseim felület telefonos – Forrás: képernyőkép

Ha eddigi megrendelés nem található a „Jelenleg nincs kifizetendő, illetve kiszállítandó tételekkel lévő megrendelése” szöveg jelenik meg ezen a felületen, a navigációs menüvel haladhat tovább a felhasználó. A „Kilépés” gombot választva pedig kijelentkezhet az oldalról.

### Adminisztrátor eszközei

A képen szöveg, képernyőkép, Betűtípus, névjegykártya látható

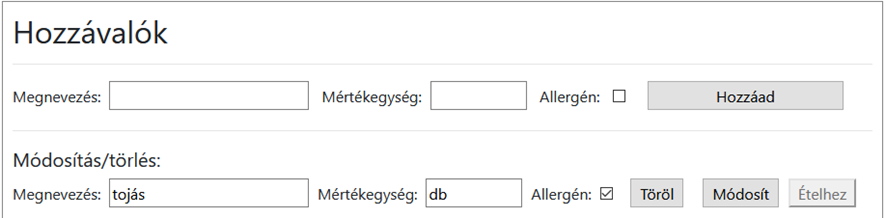
Automatikusan generált leírás

. ábra Admin menü – Forrás: képernyőkép

Ha a bejelentkezett felhasználónak vannak adminisztrátori jogai akkor a 16. ábrán látható módon a menüben megjelenik két új terület, „Admin”, melyhez tartozik 2 alterület és „Rendelések”, melyen az admin felhasználó a futó rendelésekről értesülhet és leltár információk is itt elérhetőek. Ha az admin felhasználó a „Hozzávalók-Ételek” felületre navigál négy alfunkciót ér el.

#### Hozzávalók alfunkció

A „Hozzávalók-Ételek” felületen elérhető első alfunkció a „Hozzávalók”.

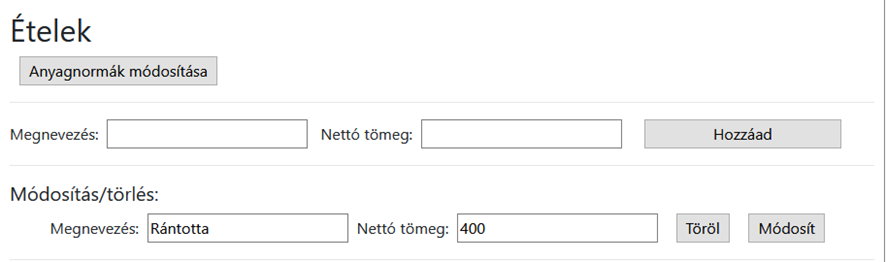


17. ábra Hozzávalók alfunkció – Forrás: képernyőkép

Hozzávaló felvétele a rendszerbe a 17. ábrán látható panel segítségével végezhető, megnevezés, kívánt mértékegység, majd allergén adatok megadása után a „Hozzáad” gombra kattintva.

A funkció felületén látható a rendszerben fellelhető összes alapanyag, ezek tulajdonságainak megváltoztatása után a módosít gombra kattintva menthető minden változtatás. A „Töröl” gomb törli a kiválasztott hozzávalót.

#### Ételek alfunkció



18. ábra Ételek alfunkció – Forrás: képernyőkép

Étel felvétele a rendszerbe a 18. ábrának felső részén lévő kérdőíven keresztül történik a megnevezés, valamint a nettó tömeg tulajdonságok megadása után a „Hozzáad” gombra kattintva.

A funkció felületén látható a rendszerben található összes étel, ezek tulajdonságainak megváltoztatása után a „Módosít” gombra kattintva menthetjük változtatásainkat az adott hozzávalón. Az anyagnormák módosítása gombra kattintva az ételek tulajdonságainak listája kibővül a hozzájuk tartozó kép módosításával a tallózás gombra kattintva kiválaszthatjuk a feltöltendő képet, a módosítás gombbal menthetjük a változtatást. A „Töröl” gomb törli a választott ételt.

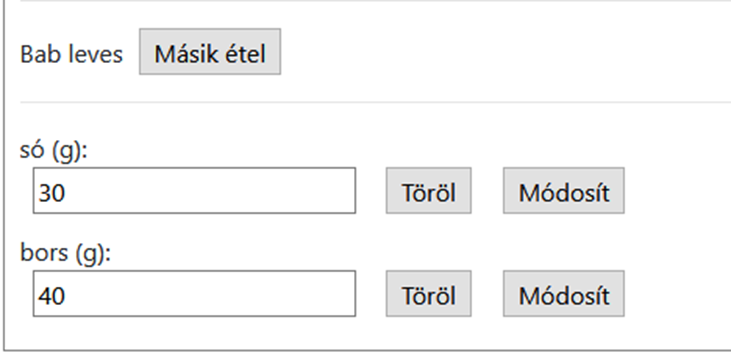
#### Hozzávalók ételekhez rendelése

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

19. ábra Hozzávalók ételekhez rendelése alfunkció – Forrás: képernyőkép

Hozzárendelés menete a következő, a 19. ábrán látható legördülőstából kiválasztjuk a megfelelő ételt, ezután megjelennek a kiválasztott ételhez eddig rendelt hozzávalók és mennyiségük.



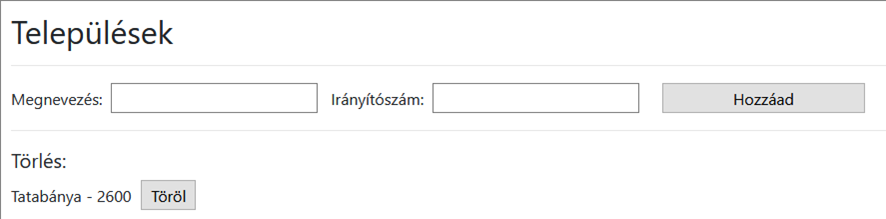
. ábra Étel hozzávaló igénye – Forrás: képernyőkép

Ha az ételhez már van hozzávaló rendelve, akkor azok mennyisége módosítható, illetve hozzávaló eltávolítható az előző funkciókhoz hasonló módon.

Hozzávaló kapcsolása pedig, úgy történik, hogy a 17. ábrán látható „Ételhez” gombra kattintunk a „Hozzávaló” alfunkción, így a hozzávaló hozzákapcsolódik az ételhez.

#### Települések alfunkció

A negyedik és egyben utolsó alfunkció, amely ezen a felületen található a „Település”, mely segítségével az admin felhasználó esetleges új kiszállítási területet vehet fel a rendszerbe.



21. ábra Település alfunkció – Forrás: képernyőkép

Település felvétele a rendszerbe a 21. ábrának felső részén lévő panelen történik a megnevezés, valamint az irányítószám tulajdonságok megadása után a „Hozzáad” gombra kattintva. A „Töröl” gomb törli a választott települést.

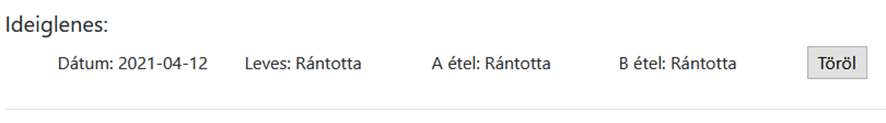
#### Étlap készít/módosít



22. ábra Ideiglenes étlap készítése – Forrás: képernyőkép

Erre a felületre az admin felhasználó szintén a menürendszerben megjelent „Admin”, azon belül pedig az „Étlap készít/módosít” lehetőségen keresztül juthat el.

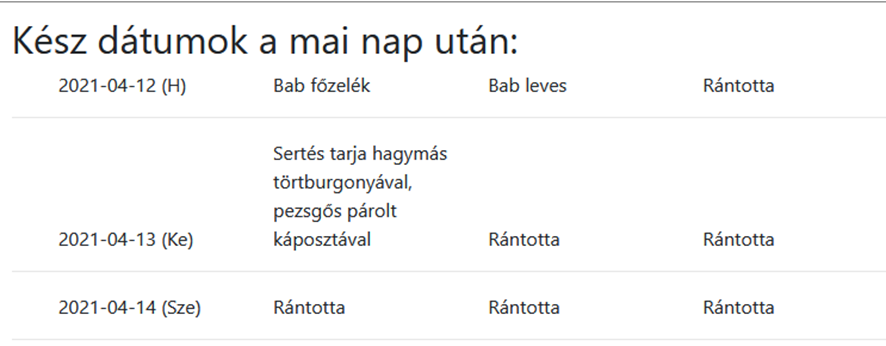
Ideiglenes étlap elemet a 22. árán látható űrlapot kitöltve hozhatunk létre a megfelelő dátumra a megfelelő ételeket hozzáadva, amint felveszünk elemeket az ideiglenes étlap láthatóvá válik.



23. ábra Ideiglenes étlap – Forrás: képernyőkép

Ez az étlap még nem végleges, elemei törölhetők, ha elégedettek vagyunk az elkészült étlappal rányomhatunk a „lead” gombra. Figyelem leadás után az étlap nem módosítható!

A funkció felületének alsó táblázatában, a 24. ábrán láthatóan, a mai nap után már beállított dátumok látszódnak.

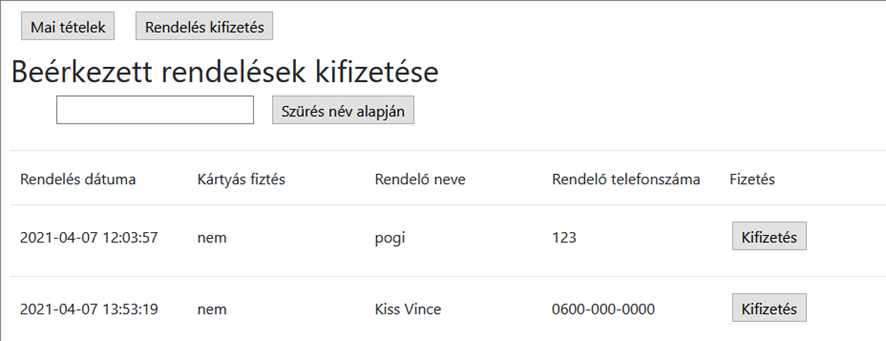


24. ábra Elkészített dátumok az étlapon – Forrás: képernyőkép

Az első oszlopban látható a dátum, amihez már van kapcsolt étlapelem, attól jobbra pedig az ételek: leves, „A” menü, „B” menü sorrendben. Ha véletlenül hétvégi dátumot veszünk fel a rendszerbe az, az „Étlapok” felületen nem fog megjelenni, ha pedig a következő három hétben valamelyik hétköznap itt nem kerül beállításra, akkor az „Étlapok” felületen annál a dátumnál a „szünnap” felirat jelenik meg.

#### Rendelések

A rendelések felület, a menürendszer „Rendelések” gombjára kattintva érhető el.



. ábra Rendelések kifizetése – Forrás: képernyőkép

Admin felhasználói jogosultságokkal rendelkező felhasználónak a rendelések felület kezdő képernyőjének középső táblázatában, látható a 25. ábrán, megjelenő felületen nyomon követheti az összes jelenleg futó megrendelést, emellett a már lezárult, de ki nem fizetett megrendelések is itt jelennek meg. A táblázatban bal oldali oszlopaiban a megrendelés információi láthatóak, jobb szélén pedig a kifizetés gomb, mellyel kezelhető a kiszállítás helyszínén történő kifizetések nyilvántartása. Ezen a felületen több beérkezett megrendelés esetére név alapján kereshetünk a felső kereső mezőbe írva a keresett személy nevét, majd a „Szűrés név alapján” gombra kattintva a beírt névhez tartozó rendelések jelennek meg.

A felület felső részén a „Mai tételek” menü pontot választva a 26. és 27. ábrán látható felület jelenik meg, melyen a mára elkészítendő összes tétel kiszállítási címmel, megrendelő telefonszámmal együtt megjelenik, ha esetleg adott napra nem található megrendelés a „Mára nincs teendő” felirat jelenik meg. Ezek mellett látható megrendelő által választott fizetési mód, így ebben az oszlopban lehet: kártyás, készpénzes vagy kifizetve. A „Rendelés kifizetés” gombra kattintva pedig visszatérhetünk a 25. ábrán látható felületre.



. ábra Mai tételek (1) – Forrás: képernyőkép

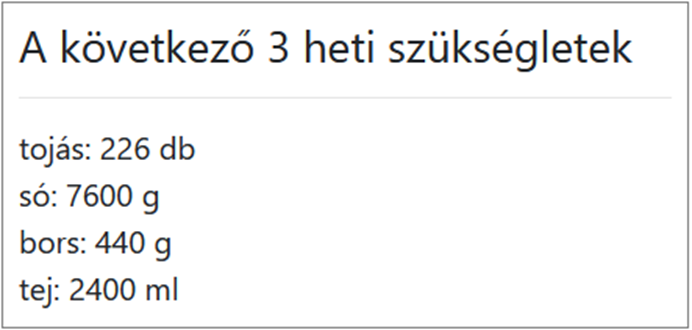
A képen szöveg, képernyőkép, nyugta, Betűtípus látható

Automatikusan generált leírás

. ábra Mai tételek (2) – Forrás: képernyőkép

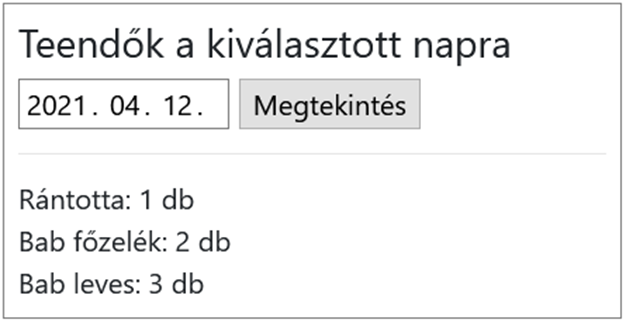
A 27. ábráról személyes adatok védelme érdekében töröltem az információkat, normál esetben a táblázatban megjelennének a megrendelő adatai.

Ezeken túl a rendelések funkció felületén két, fontos információkat megjelenítő kimutatás látható.



. ábra Háromheti szükségletek kimutatása – Forrás: képernyőkép

A 28. ábrán látható kimutatáson megfigyelhető, az összesített hozzávaló szükséglet a rendelések alapján a következő 3 hétre kivetítve, megkönnyítve ezzel a leltár karbantartását.



. ábra Napi teendők

A 29. ábrán látható kimutatásban pedig kiválasztható egy dátum, a „Megtekintés” gombra kattintva pedig megjelenik a választott naphoz tartozó összes elkészítendő tétel egyesítve.

Összesítve a rendelés felület három területét fedi le a konyhában futó folyamatoknak, segít a kiszállításban fizetés nyomon követésben. Emellett jól látható, hogy melyik tételből mennyit kell adott napra készíteni és segít felkészülni más napok teendőire. Végül a leltár folyamatokat is nagyban segíti a háromheti szükségletek kimutatásával.

## A rendszer bevezetése

A rendszer bevezetése A fejlesztés életciklusának egy fontos mérföldköve, ez a folyamat, könnyedén haladt, az általam elkészített rendszert, a felhasználó könnyedén elérte a webszolgáltató által kínált webcímen, oda bejelentkezett az adminisztrátori jogosultságokat nyújtó adatokkal mobil, illetve asztali eszközén egyaránt. A rendszer használatát elsajátította az általam készített oktatóanyag segítségével.

Amint a felhasználók is értesültek az oldal indulásáráról jelentős részük azonnal áttért a használatára emellett számos új megrendelő is regisztrált az oldalra.

## Fejlesztési lehetőségek

A program életciklusa ott nem áll meg, hogy átadásra kerül, a későbbiekben, használat során esetleges hibák kiderülhetnek, illetve új funkciókra lehet igény, ezekkel kapcsolatban ugyan úgy kell eljárni, ahogy a már meglévőknél látható volt, pontos tervezés, fejlesztés és tesztelés elengedhetetlen. A következőkben néhány felmerült fejlesztési lehetőségről teszek említést.

A probléma felvetésben meghatározott követelményeknek az elkészült rendszer sikeresen eleget tesz funkciók szintjén, a megrendelés és azok megtekintése különösebb informatikai előképzettség nélkül megvalósítható. Emellett az oldalon biztosítva van, hogy mindenki csak azokat adatokat láthatja, ami feladatköréhez kapcsolódik.

Skálázhatóság szempontjából a rendszer és az azt támogató adatbázis jól felvan készítve, amit kiemelnék, hogy ha jövőben olyan igény merülne fel, hogy egy naphoz több ételt szeretnénk rendelni, mint a mostani három kisebb változtatások árán az is elérhető, de ez a közel jövőben nem várható, hacsak nem bővül a vállalkozás.

A konzultációk során felmerült még egy blog indítása az oldalon, mely által más akár közösségi oldalakról is újabb felhasználókat lehetne az oldalra csalogatni. Ezeken kívül online fizetésre nincs jelenleg lehetőség az oldalon keresztül, de ennek bevezetésére is felvan készítve a rendszer.

# Összegzés

A tervezés folyamata, elméletben ismert technológiákat ültetett át gyakorlatba, SWOT analízis és Ishikawa diagram területén, illetve use-case modellek használatában egyaránt hasznos ismereteket mélyítettem el.

A fejlesztés a pontos tervezésnek köszönhetően rendben haladt, minden a követelmény specifikációban, illetve a feladatkiírásban meghatározott követelmény teljesült. A folyamatsorán előforduló nehézségekkel való megküzdés segített, hogy jobb fejlesztővé váljak, egyetemi tanulmányaimat is tudtam kamatoztatni bizonyos megoldások kapcsán, de nagyon sokat tanultam konkrétan ebből a feladatból is.

Elmélyültem az MVC minta követésében, mely eleinte bonyolultnak tűnt, de a végére annyira leegyszerűsítette új funkciók beépítését, hogy úgy gondolom jól kihasználtam az általa kínált lehetőségeket. Megkönnyítette a fejlesztés, tesztelés, hibajavítás folyamatát.

A rendszer oktató anyagának elkészítése segített közelebb kerülni a felhasználókhoz, érdekes tapasztalat volt egy teljesen más szemszögből nézni a rendszerre.

Összességében tehát azt gondolom, hogy ez a fejlesztési folyamat hozzájárult, hogy jobb fejlesztővé váljak, releváns tudást adott számos területen és abban, hogy egy nagyobb projekt, miként valósulhat meg elejétől a végéig.

# Irodalomjegyzék

[1] „Ishikawa Diagram: A Guide on How to Use It”, SafetyCulture. Elérés: 2024. április 20. [Online]. Elérhető: https://safetyculture.com/topics/ishikawa-diagram/

[2] „Egészségpajzs étel házhozszállítás - egészséges ételrendelés Budapesten és környékén”. Elérés: 2024. április 21. [Online]. Elérhető: https://egeszsegpajzs.hu/

[3] „Cégünkről | e-food - Van, aki finoman szereti!”, e-food. Elérés: 2024. április 21. [Online]. Elérhető: https://www.e-food.hu/cegunkrol/

[4] „Az XAMPP megismerése annak funkcióival és fontos részeivel az XAMPP-ben”. Elérés: 2024. november 6. [Online]. Elérhető: https://altitudetvm.com/hu/website/3096-pengertian-xampp-beserta-fungsi-dan-bagian-bagian-penting-pada-xampp.html

[5] phpMyAdmin contributors, „phpMyAdmin”, phpMyAdmin. Elérés: 2024. november 6. [Online]. Elérhető: https://www.phpmyadmin.net/

[6] ardalis, „Overview of ASP.NET Core MVC”. Elérés: 2024. november 6. [Online]. Elérhető: https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0

[7] „Abmodell.pdf”. Elérés: 2024. november 19. [Online]. Elérhető: http://www.sze.hu/~szorenyi/sz03/segitseg/adatmodell/Abmodell.pdf

[8] „Normálformák | Adatbázisok gyakorlat”. Elérés: 2024. november 19. [Online]. Elérhető: https://www.inf.u-szeged.hu/~gnemeth/adatbgyak/exe/Normalizalas\_web/normlformk.html

[9] CodeProject, „CodeProject”. Elérés: 2024. november 19. [Online]. Elérhető: https://www.codeproject.com

[10] „Stack Overflow - Where Developers Learn, Share, & Build Careers”, Stack Overflow. Elérés: 2024. november 19. [Online]. Elérhető: https://stackoverflow.com/

[11] Martin Kleppmann: Designing Data-Intensive Applications; „Könyv:” O'Reilly Media, Inc, USA

# Mellékletek

Az alábbiakban megjelenítek néhány általam kiemeltnek tartott részletet az elkészült kódbázisból.

<https://github.com/khgfkufigzufg/SzakdolgozatPHP> linken elérhető a teljes kódbázis.

**Index.php**

1. <?php
2. session\_start();
3. //-------------osztályok---------------
4. require 'model/db.inc.php';
5. require 'model/Hozzavalo.php';
6. require 'model/Telepules.php';
7. require 'model/Rendeles.php';
8. require 'model/Etel.php';
9. require 'model/EtlapSor.php';
10. require 'model/EtelHozzavalo.php';
11. require 'model/NapiRendeles.php';
12. require 'model/User.php';
13. require 'model/Tetel.php';
14. //-------------osztályok---------------
15. if(isset($\_COOKIE['user'])){//BELÉPTETÉS----------------------------
16. $sql = "SELECT \* FROM user WHERE email LIKE '".$\_COOKIE['user']."'";
17. $result = $Db->execSQL($sql);
18. if($row = $result->fetch\_assoc()){
19. $\_SESSION['user'] = $row;
20. }
21. }
22. ?>
23. <!DOCTYPE html>
24. <html lang="hu">
25. <head>
26. <!-- Head -->
27. <meta charset="utf-8">
28. <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
29. <!-- Bootstrap CSS -->
30. <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">
31. <link rel="stylesheet" type="text/css" href="style/main.css">
33. <?php
34. if(isset($\_REQUEST['action'])){
35. if($\_REQUEST['action'] == 'belep' || $\_REQUEST['action'] == 'regist'){
36. echo'<link rel="stylesheet" type="text/css" href="style/style.css">';
37. }
38. if($\_REQUEST['action'] == 'etlap\_rendeles' || $\_REQUEST['action'] == 'kosar'){
39. echo'<link rel="stylesheet" type="text/css" href="style/etlapok.css">';
40. }
41. }
42. if(isset($\_SESSION['user']['admin']) && $\_SESSION['user']['admin'] == 1){
43. echo'<link rel="stylesheet" type="text/css" href="style/admin.css">';
44. }
45. ?>
46. <link rel="stylesheet" type="text/css" href="style/tooltip.css">
47. <link rel="stylesheet" type="text/css" href="style/kosarRend.css">
48. <title>Mezenga</title>
49. </head>
51. <?php
52. //-------------ACTION---------------
53. $action = 'start';
54. $adminact = array('hozza','delhozz','hozzavaloAdd', 'updHozz', 'etelhozzr', 'etlapCRUD', 'etelCRUD');
56. if(!empty($\_REQUEST['action'])) {
57. if(in\_array($\_REQUEST['action'], $adminact)){
58. if(isset($\_SESSION['user'])){
59. if($\_SESSION['user']['admin']==1){
60. if(file\_exists('controller/'.$\_REQUEST['action'].'.php')) {
61. $action = $\_REQUEST['action'];
62. }
63. }
64. }
65. else{
66. $action = 'start';
67. }
68. }
69. elseif(file\_exists('controller/'.$\_REQUEST['action'].'.php')) {
70. $action = $\_REQUEST['action'];
71. }
72. }
73. //-------------ACTION---------------
74. //-------------NAVBAR---------------
75. if(!($action == 'belep' || $action == 'regist')){
76. echo '<body>';
77. echo '<nav class="navbar navbar-expand-xl navbar-dark bg-dark align-bottom">
78. <a class="navbar-brand" href="index.php?action=start"><img src="img/mezenga.png" alt="" width="60" height="60"> <span class="szep">Mezenga</span></a>
79. <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarsExample06" aria-controls="navbarsExample06" aria-expanded="false" aria-label="Toggle navigation">
80. <span class="navbar-toggler-icon"></span>
81. </button>
83. <div class="collapse navbar-collapse" id="navbarsExample06">
84. <ul class="navbar-nav mr-auto">
85. <li class="nav-item">
86. <a class="nav-link" href="index.php?action=etlap\_rendeles">Étlapok</a>
87. </li>';
88. if(isset($\_SESSION['kosar']['isset']) && $\_SESSION['kosar']['isset']){
89. echo '
90. <li class="nav-item">
91. <a class="nav-link" href="index.php?action=kosar">Kosár</a>
92. </li>';
93. }
95. if (!isset($\_SESSION['user'])) {
96. echo '
97. <li class="nav-item">
98. <a class="nav-link" href="index.php?action=belep">Belépés</a>
99. </li>';
101. }
102. else {
104. if($\_SESSION['user']['admin']==1){
106. echo '<li class="nav-item dropdown">
107. <a class="nav-link dropdown-toggle" href="https://example.com" id="dropdown06" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Admin</a>
108. <div class="dropdown-menu" aria-labelledby="dropdown06">
109. <a class="dropdown-item" href="index.php?action=hozza">Hozzávalók-Ételek</a>
110. <a class="dropdown-item" href="index.php?action=etlap">Étlap készít/módosít</a>
112. </div>
113. </li>';
114. echo '
115. <li class="nav-item">
116. <a class="nav-link" href="index.php?action=rendelesMegtekint&act=kif">Rendelések</a>
117. </li>';
119. }
120. else{
121. echo '
122. <li class="nav-item">
123. <a class="nav-link" href="index.php?action=rendelesMegtekint">Rendeléseim</a>
124. </li>';
125. }
126. echo '<li class="nav-item">
127. <a class="nav-link" href="index.php?action=logout">Kilépés</a>
128. </li>';
129. }

132. echo '
134. </ul>
135. </div>
136. </nav>';
138. //-------------NAVBAR---------------
139. }
140. else{
141. echo '<body class="text-center">';
142. }
144. require 'controller/'.$action.'.php';//action meghívása
145. ?>
146. <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
147. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" crossorigin="anonymous"></script>
148. <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous"></script>
149. </body>
150. </html>

**DB osztály**

1. <?php
2. // Db osztály
3. class Db {
4. //bkTlk8QOX4vzjP5E --- adminszakd --- c31f --- afw!ZXfS4fK2
5. private $servername = "localhost";
6. private $username = "c31f";
7. private $password = "afw!ZXfS4fK2";
8. private $dbname = "szakdoga";
10. private $conn;
12. public function \_\_construct() {
13. // Create connection
14. $this->conn = new mysqli($this->servername,
15. $this->username, $this->password, $this->dbname);
17. // Check connection
18. if ($this->conn->connect\_error) {
19. die("Connection failed: " . $this->conn->connect\_error);
20. }
22. /\*else{
23. echo 'Sikeres csatlakozás';
24. }\*/
25. }
27. public function execSQL($sql) {
28. if(!$result = $this->conn->query($sql)) {
29. echo("Error description: " . mysqli\_error($this->conn)." in $sql");
30. }
31. return $result;
32. }
34. public function insertSQL($sql) {
35. $result = $this->execSQL($sql);
36. return $this->conn->insert\_id;
37. }
39. }
40. // Db példány
41. $Db = new Db();
42. ?>

**Controller**

1. <?php
2. //-----------------------Rendelés megtekintés user/admin---------------------------
3. if(isset($\_SESSION['user'])){
4. if($\_SESSION['user']['admin'] == 1){ //Admin
6. if(isset($\_GET['act']) && $\_GET['act'] == 'kif'){ //kifizetendő vagy futó megrendelések
7. $sql = "SELECT DISTINCT rendeles.rendId FROM `rendeles`
8. INNER JOIN napirendeles ON rendeles.rendId = napirendeles.rendId";
9. $sql .= " INNER JOIN user ON rendeles.userId=user.userId
10. WHERE fizetette = 0 OR napirendeles.datum>=CURDATE()";
11. if(isset($\_POST['knev'])){
12. $sql .= " AND unev LIKE '%".$\_POST['knev']."%'";
13. }
15. $result = $Db->execSQL($sql);
17. while($row = $result->fetch\_assoc()){
18. $rendelesek[$row['rendId']] = $Rendeles->getRendeles($row['rendId'], $Db);
19. }
20. }
21. elseif(isset($\_GET['act'])){ //mai tételek
22. $sql = "SELECT  rendeles.rendId, eid, SUM(menny) as db, rendeles.userId FROM tetelesrend
23. INNER JOIN napirendeles ON tetelesrend.napId=napirendeles.napId
24. INNER JOIN rendeles ON rendeles.rendId=napirendeles.rendId
25. WHERE napirendeles.datum=CURDATE()
26. GROUP BY rendeles.rendId, eid";
27. $result = $Db->execSQL($sql);
28. while($row = $result->fetch\_assoc()){
29. $maitetelek[$row['rendId']]['userId'] = $row['userId'];
30. $maitetelek[$row['rendId']][$row['eid']] = $row['db'];
32. }
33. }
34. //háromheti szükségletek
35. $sql = "SELECT eid, SUM(menny) as db FROM tetelesrend
36. INNER JOIN napirendeles ON tetelesrend.napId=napirendeles.napId
37. WHERE napirendeles.datum>=CURDATE()
38. GROUP BY eid";
39. $result = $Db->execSQL($sql);
41. if($result->num\_rows>0){
42. while($szukseglet = $result->fetch\_assoc()){
43. $sql2 = "SELECT hozzavalo.hid, mennyi, megys FROM hozzavalo
44. INNER JOIN etelhozzav ON hozzavalo.hid=etelhozzav.hid WHERE eid = ".$szukseglet['eid'];
45. $rs = $Db->execSQL($sql2);
46. if($rs->num\_rows>0){
47. while($hozz = $rs->fetch\_assoc()){
48. if(!isset($haromheti[$hozz['hid']][$hozz['megys']])){
49. $haromheti[$hozz['hid']][$hozz['megys']] = ($hozz['mennyi']\*$szukseglet['db']);
50. }
51. else{
52. $haromheti[$hozz['hid']][$hozz['megys']] += ($hozz['mennyi']\*$szukseglet['db']);
53. }
54. }
55. }
56. }
57. }
58. //kereső alapján napi teendők
59. if(isset($\_POST['kivdatum'])){
60. $napiEteleksql = "SELECT eid, SUM(menny) as db FROM tetelesrend
61. INNER JOIN napirendeles ON tetelesrend.napId=napirendeles.napId
62. WHERE napirendeles.datum='".$\_POST['kivdatum']."'
63. GROUP BY eid";
64. $napiEtelek = $Db->execSQL($napiEteleksql);
65. if($napiEtelek->num\_rows>0){
66. while($napiEtel = $napiEtelek->fetch\_assoc()){
67. $nap[$napiEtel['eid']] = $napiEtel['db'];
68. }
69. }
70. }
71. }
73. else{ //user saját rendelési adatai/
74. $sql = "SELECT rendeles.rendId, napirendeles.datum, tetelesrend.eid, tetelesrend.menny FROM `rendeles`
75. INNER JOIN napirendeles ON rendeles.rendId = napirendeles.rendId
76. INNER JOIN tetelesrend ON napirendeles.napId=tetelesrend.napId
77. WHERE (napirendeles.datum >= CURDATE() OR rendeles.fizetette=0)
78. AND userId=".$\_SESSION['user']['userId'];
80. $result = $Db->execSQL($sql);
81. while($row = $result->fetch\_assoc()){
82. $rendeleseim[$row['rendId']][$row['datum']][$row['eid']] = $row['menny'];
83. }
85. if(isset($\_GET['delId'])){
86. $Rendeles->delRendeles($Db, $\_GET['delId']);
87. header("Refresh:0 url=index.php?action=rendelesMegtekint");
88. }
90. require 'view/user/rendelesek.php';
91. }

94. if(!empty($\_GET['kif'])){
95. $Rendeles->fizetes($Db, $\_GET['kif']);
96. header("Refresh:0 url=index.php?action=rendelesMegtekint&act=kif");
97. }
98. }
99. else{
100. header('Location:index.php?action=belep');
101. exit;
102. }
103. if(isset($\_GET['act']) && $\_GET['act']== 'kif'){
104. require 'view/user/rendelesek.php';
105. }
106. elseif(isset($\_GET['act'])){
107. require 'view/rendelesMegtekint/maiTetelek.php';
108. }
109. ?>
110. <?php
111. //--------------------KOSÁR----------------------majd vissza kerülünk arra a felütre amelyikről érkeztünk étlap/kosár(ha kosár üres=>étlapra jutunk)
112. if(isset($\_GET['lead'])){ //RENDELÉS MENTÉS
113. if(isset($\_SESSION['user'])){
114. if(isset($\_POST['kartyase'])){
115. $Rendeles->setRendeles($\_SESSION["user"]['userId'], 1);
116. $Rendeles->addRendeles($Db);
117. }
118. else{
119. $Rendeles->setRendeles($\_SESSION["user"]['userId'], 0);
120. $Rendeles->addRendeles($Db);
121. }
122. for($i=0;$i<$\_SESSION['dbtetel'];$i++){

125. $j=$i;
126. $NapiRendeles->setNapi($Rendeles->getRendID(), $\_POST['datum'.$i]);
127. $NapiRendeles->addNapi($Db);
128. while($j<$\_SESSION['dbtetel'] && $\_POST['datum'.$i] == $\_POST['datum'.$j]){
130. $Tetel->setTetel($NapiRendeles->getNapiID(), $\_POST['etel'.$j], $\_POST['menny'.$j]);
131. $Tetel->addTetel($Db);
132. $j++;
133. }
134. $i=$j-1;
135. }
137. unset($\_SESSION['kosar']);
138. header('Location:index.php?action=etlap\_rendeles');
139. exit;
140. }
141. else{
142. header('Location:index.php?action=belep&vasarlas=TRUE');
143. exit;
144. }
145. }
146. elseif(!empty($\_GET['datum']) && !empty($\_GET['eid']) && !empty($\_GET['act'])){
147. if(isset($\_SESSION['kosar'][$\_GET['datum']][$\_GET['eid']])){
148. if($\_GET['act'] == 'del'){//GETbe mentett datum eid alapján töröljük a kosár elemet
149. $\_SESSION['kosar'][$\_GET['datum']][$\_GET['eid']] = 0;
150. header('Location:index.php?action=kosar');
151. exit;
152. }
153. elseif($\_GET['act'] == 'minus'){//GETbe mentett datum eid alapján csökkentjük a kosár elem mennyiségét
154. if($\_SESSION['kosar'][$\_GET['datum']][$\_GET['eid']]>0){
155. $\_SESSION['kosar'][$\_GET['datum']][$\_GET['eid']]--;
156. if(isset($\_GET['kosar'])){
157. header('Location:index.php?action=kosar');
158. exit;
159. }
160. else{
161. if(isset($\_GET['tabla'])){
162. header('Location:index.php?action=etlap\_rendeles#'.$\_GET['datum'].''.$\_GET['eid']);
163. exit;
164. }
165. else{
166. header('Location:index.php?action=etlap\_rendeles');
167. exit;
168. }
169. }
170. }
171. else{
172. if(isset($\_GET['kosar'])){
173. header('Location:index.php?action=kosar');
174. exit;
175. }
176. elseif(isset($\_GET['tabla'])){
177. header('Location:index.php?action=etlap\_rendeles#'.$\_GET['datum'].''.$\_GET['eid']);
178. exit;
179. }
180. else{
181. header('Location:index.php?action=etlap\_rendeles');
182. exit;
183. }
184. }
185. }
186. elseif($\_SESSION['kosar'][$\_GET['datum']][$\_GET['eid']]<127){//GETbe mentett datum eid alapján csökkentjük a kosár elem mennyiségét
187. $\_SESSION['kosar'][$\_GET['datum']][$\_GET['eid']]++;
188. $\_SESSION['kosar']['isset'] = TRUE;
189. if(isset($\_GET['kosar'])){
190. header('Location:index.php?action=kosar');
191. exit;
192. }
193. elseif(isset($\_GET['tabla'])){
194. header('Location:index.php?action=etlap\_rendeles#'.$\_GET['datum'].''.$\_GET['eid']);
195. exit;
196. }
197. else{
198. header('Location:index.php?action=etlap\_rendeles');
199. exit;
200. }
201. }
202. }
203. else{
204. if(isset($\_GET['kosar'])){
205. header('Location:index.php?action=kosar');
206. exit;
207. }
208. elseif(isset($\_GET['tabla'])){
209. header('Location:index.php?action=etlap\_rendeles#'.$\_GET['datum'].''.$\_GET['eid']);
210. exit;
211. }
212. else{
213. header('Location:index.php?action=etlap\_rendeles');
214. exit;
215. }
216. }
218. }

221. require 'view/etelek/kosar.php';
222. ?>
223. <?php
224. if(isset($\_POST['datum'])){ //admin étlap létrehozás közben a sorokat itt tároljuk
225. if(isset($\_SESSION['ideigEtlap'])){
226. $\_SESSION['ideigEtlap'][count($\_SESSION['ideigEtlap'])+1]['datum'] = $\_POST['datum'];
227. $\_SESSION['ideigEtlap'][count($\_SESSION['ideigEtlap'])]['leves'] = $\_POST['leves'];
228. $\_SESSION['ideigEtlap'][count($\_SESSION['ideigEtlap'])]['aetel'] = $\_POST['aetel'];
229. $\_SESSION['ideigEtlap'][count($\_SESSION['ideigEtlap'])]['betel'] = $\_POST['betel'];
230. }
231. else{
232. $\_SESSION['ideigEtlap'][1]['datum'] = $\_POST['datum'];
233. $\_SESSION['ideigEtlap'][1]['leves'] = $\_POST['leves'];
234. $\_SESSION['ideigEtlap'][1]['aetel'] = $\_POST['aetel'];
235. $\_SESSION['ideigEtlap'][1]['betel'] = $\_POST['betel'];
236. }
238. }
239. if(isset($\_SESSION['ideigEtlap'])){ //kitörölt elemek a tömbben maradnak null értékkel
240. $nullos=1;
241. while($nullos<count($\_SESSION['ideigEtlap']) && $\_SESSION['ideigEtlap'][$nullos]==null){
242. $nullos++;
243. }
244. echo $nullos;
245. }
246. if(!empty($\_GET['delID'])){
247. if($nullos == $\_GET['delID']){
248. $\_SESSION['ideigEtlap'][$nullos]=NULL;
249. $nullos++;
250. }
251. else{
252. $sv = $\_SESSION['ideigEtlap'][$nullos];
253. $\_SESSION['ideigEtlap'][$nullos]=NULL;
254. $\_SESSION['ideigEtlap'][$\_GET['delID']]= $sv;
255. $nullos++;
256. }
257. echo '  '.$nullos;
258. }
259. if(count($\_SESSION['ideigEtlap'])>1){ //ideiglenes étlap sorba rendezése
260. for($i=$nullos;$i<count($\_SESSION['ideigEtlap']);$i++){
261. $min=$i+1;
262. for($j=$i;$j<count($\_SESSION['ideigEtlap'])+1;$j++){
263. if($\_SESSION['ideigEtlap'][$j]['datum']<$\_SESSION['ideigEtlap'][$min]['datum']){
264. $min=$j;
265. }
266. }
267. $sv = $\_SESSION['ideigEtlap'][$i];
268. $\_SESSION['ideigEtlap'][$i] = $\_SESSION['ideigEtlap'][$min];
269. $\_SESSION['ideigEtlap'][$min] = $sv;
270. }
271. }
272. if(!empty($\_GET['lead'])){ //etlapSorok mentése az adatbázisba
273. foreach($\_SESSION['ideigEtlap'] as $etlapSor){
274. if($etlapSor != NULL){
275. $beEtlapSor->setEtlapSor($etlapSor['leves'], $etlapSor['aetel'], $etlapSor['betel'], $etlapSor['datum']);
276. $beEtlapSor->addSor($Db);
277. unset($\_SESSION['ideigEtlap']);
278. }
279. }
280. }
281. header('Location:index.php?action=etlap');
282. exit;
283. ?>