

BMSZC Egressy Gábor Két Tanítási Nyelvű Technikum

Nyami Menüztetés

Programozás gyakorlat, Adatbázis kezelés, IKT projektmunka

osztályozó vizsga feladat

2023. április 17.

*Szerző:*

*Ónodi-Kiss Viktor 11/K,*

*Szoftverfejlesztő szakirány,*

*Két tanítási nyelvű informatika osztály,*

*11. évfolyam*

Tartalomjegyzék[[1]](#forr_1)

1. [Bevezetés](#cim_1)
   1. [Témaválasztás indoklása](#cim_1_1)
   2. [Motiváció](#cim_1_2)
   3. [Megoldandó feladat leírása](#cim_1_3)
2. [Felhasználói dokumentáció](#cim_2)
   1. [Ismertető](#cim_2_1)
   2. [Célközönség](#cim_2_2)
   3. [Megvalósítás eszközei](#cim_2_3)
   4. [Rendszerkövetelmények](#cim_2_4)
   5. [Telepítés, futtatás, konfiguráció](#cim_2_5)
   6. [Felhasználói felület](#cim_3)
3. [Fejlesztői dokumentáció](#cim_3)
   1. [Megoldási terv](#cim_3_1)
   2. [Megvalósítás](#cim_3_2)
   3. [Fejlesztés](#cim_3_3)
      1. [Továbbfejlesztés lehetőségei](#cim_3_3_1)
4. [Összegzés](#cim_4)

[Felhasznált források](#cim_forr)

[Ábrajegyzék](#cim_abra)

1. Bevezetés

1.1. Témaválasztás indoklása

A COVID 19 járvány következtében a mindennapi élet legalapvetőbb részei alakultak át érintésmentessé. A Netflix és a hasonló stream szolgáltatások, melyek már ekkor is igen elterjedtek voltak, egyre közkedveltebbek lettek. Sajnos azonban az ételrendelést kicsivel nehezebb érintésmentessé tenni, tekintve, hogy az étel, mint fizikai objektum, már önmagában érintést igényel a fogyasztáshoz.

Ezt a problémát orvosolhatná az, ugyanúgy érintésmentes, telefonos rendelés, aminek viszont a hátránya, hogy folyamatosan lefoglalva tart egy alkalmazottat a rendelés idejéig, illetve túlságosan időigényes.

Az alkalmazásom célja egy olyan felület prototípusának létrehozása, ahol a felhasználó véletlenszerűen összeállított napi menükből választhatja ki az aznapi ebédjét, meg tudja rendelni és összegezni is tudja a rendelését.

1.2. Motiváció

A program megírásában motivált az a tény, hogy a feladatot osztályozó vizsgára kaptam feladatként, melynek sikeres teljesítését követően hetente 7 tanórám szabadul fel önfejlesztő tevékenységekre.

1.3. Megoldandanó feladat leírása

A feladat részletes specifikációját egy pdf-fájlban[[2]](#forr_2) kaptam meg, mely tartalmazta a leírást illetve a megírandó asztali alkalmazás felhasználó felületének elrendezéséhez tartozó mintákat is.

A feladat megbeszélésekor azt a tanácsot kaptam, hogy nem muszáj 100%-osan követnem a feldat leírását, amit én több alkalommal is figyelembe vettem a program megírásakor, olyan feature-ok implementálásakor, melyek alapból nem voltak követelmények, vagy szerintem a saját módszeremmel jobb hatást érnek el. Ilyen például a Tabulátorral történő oldal váltás.

2. Felhasználói dokumentáció

2.1. Ismertető

A program felhasználójának lehetősége van kettő különböző, naponta frissített menüből kiválasztania az aznapi ebédjét. A menükbe lehetségesen bekerülő levesek és főételek egy külső fájlban találhatóak meg, melyet az étterem tud karbantartani szükség szerint.

A felhasználónak továbbá lehetősége van törölni az eddig kiválasztott tételeket a “Mégsem” gombra kattintva, melyet követően egy felugró ablak ad visszajelzést a művelet sikerességéről.

A felhasználó a következőképpen tud rendelni:

1. Kiválasztja, hogy “teljes” vagy “zóna” verziót kíván rendelni az adott menüből.
2. Meghatározza a rendelés darabszámát.
3. Ha elvitelre kéri, akkor bejelöli az “elvitelre” mezőt, melynek ára hozzáadódik a rendeléshez.
4. Megnyomja a “Rendelés” gombot.
5. Felugró ablakban jelzi az alkalmazás a rendelés sikerességét.

A “Rendelések” fülre kattintva (vagy a Tabulátor billentyű megnyomásával) a felhasználónak lehetősége van megtekinteni az eddig leadott rendeléseit.

Itt a felhasználó tudja törölni a kiválasztott rendelést:

1. A felhasználó kiválaszt egy elemet a listából, ha az nem üres.
2. Megnyomja a “Rendelés törlése” gombot.
3. A kiválasztott rendelés kitörlődik a listából.
4. Felugró ablak jelzi a művelet sikerességét.

Illetve tudja törölni az összes rendelést is:

1. A felhasználó megnyomja a “Minden rendelés törlése” gombot.
2. Felugró ablak jelzi a művelet sikerességét.

A felhasználó számára megnyomható gombként szolgál minden lehetőségben a “Minimize” (“-“-el jelölt) gomb az oldal tetején, mely használatával a tálcára tudja küldeni az alkalmazást, a “Close” (“X”-el jelölt) gomb, ugyanúgy az oldal tetején, mely megnyomásával a az alkalmazás bezár.

Továbbá a felhasználónak lehetősége van megnyitni az alkalmazás névjegyét is a “Névjegy” gombra kattintva (vagy az “N” billentyű megnyomásával), amikor az alkalmazás névjegye megjelenik felugró ablakban.

Ugyanúgy az e mellett található “Kilépés” gomb megnyomásával (vagy a “K” billentyű lenyomásával) a felhasználó bezárhatja az alkalmazást.

2.2. Célközönség

Az alkalmazást otthoni felhasználásra szánom, az étterem potenciális vendégei számára a rendelésük kényelmes leadása érdekében.

2.3. Megvalósítás eszközei

A programom megvalósításához a .NET keretrendszer 4.7.2.-es verzióját használtam.

A program megírásában nagy hasznát vettem az SQLite csomagnak, melynek használatával sikerült létrehoznom az alkalmazáshoz tartozó adatbázist.

A MainForm.cs fájl felel az alkalmazás főoldalaként szolgáló Form megjelenítéséért, a Nevjegy.cs pedig a Névjegy oldal megjelenítéséért.

A View.cs tartalmazza a MainForm oldal felépítéséhez használatos kódot.

A Model.cs (és az ahhoz tartozó IModel.cs interfész) tartalmazza a View mögött rejlő logikát.

A Persistence.cs (és az ahhoz tartozó IPersistence.cs interfész) tartalmazza az adatbázist módosító, illetve azzal kapcsolatot teremtő kódot.

Az Items.cs, a Rendelesek.cs és a Menu.cs tartalmazza az azonos nevű struct-okat, melyeket az adatok eltárolására használtam fel.

2.4. Rendszerkövetelmények

1. Operációs rendszer

Az alkalmazás futtatásához Microsoft Windows 10 vagy annál frissebb, pl. MS Windows 11 operációs rendszer szükséges.

|  |  |  |
| --- | --- | --- |
| **Operációs rendszer** | **verzió** | **Architektúra** |
| MS Windows | 10, 11 | 64-bites, x86 |

1. Hardver követelmények

Az alkalmazás hardver követelménye megegyezik azzal, ami az operációs rendszer futtatásához szükséges.

2.5. Telepítés, futtatás, konfiguráció

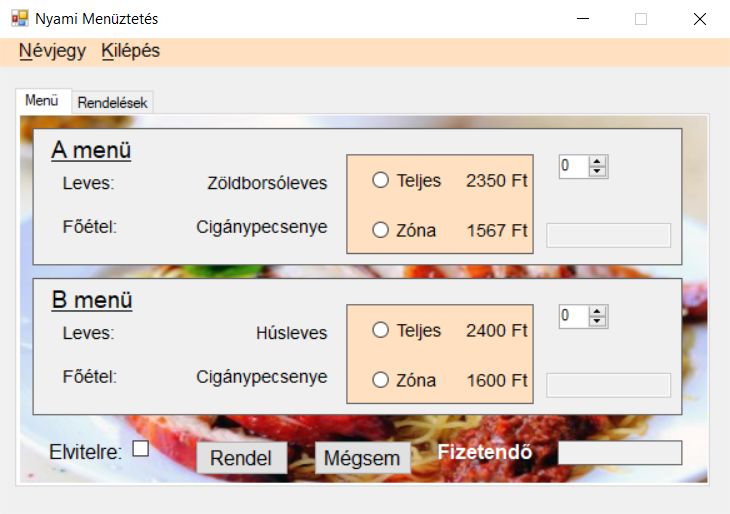
Az alkalmazáshoz nem készítettem telepítőt, ezért a futtatása csak is a Visual Studio-n keresztül valósítható meg, ahol a projekt betöltését követően a “Debug” -> “Start without debugging” menüpontra kattintva (vagy a Ctrl + F5 billentyűkombináció megnyomásával) lehet a programot futtatni.

2.5.1. Külső függőségek

Az alkalmazás rendelkezik külső függőségekkel, melyeket figyelemebe kell venni a program továbbításakor.

Ezek az SQLite-hoz tartozó fájlok, a projekt mappán belül található “forrasok” mappa és annak tartalmai, a “forras.txt”, ami a napimenü összeállításához szükséges adatokat tartalmazza, illetve a “menuztetes.jpg”, mely a hátteréül szolgál a MainForm-on található “TabPage” objektum mind a kettő fülének.

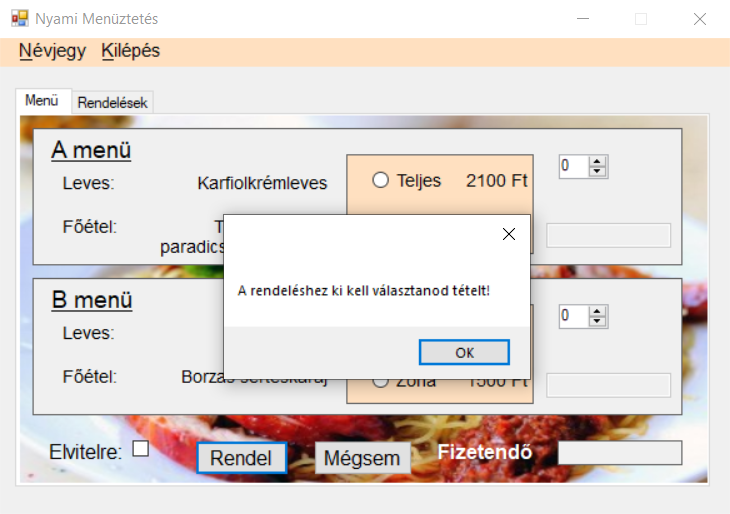
2.6. Felhasználói felület

Amikor a felhasználó elindítja a programot, a legelső oldal, amit meglát, a fő oldal (*ábra 2.1*) a “Menü” füllel kiválasztva és azon belül pedig minden adat bevitel lehetőséggel üresen.

Ábra 2. 1 - Az alkalmazás elindításakor megjelenő ablak

Itt a felhasználó ki tudja választani, hogy hogyan szeretné megrendelni a menüjét. A kettő darab menüt (“A menü” és “B menü”) követően találhatóak a rádiógombok, melyek lenyomásával ki tudja jelölni, hogy az adott menüből “Teljes” (100% ár) vagy “Zóna” (66.6% ár) adagot szeretne-e rendelni. Valamelyiket mindenképpen ki kell választani a rendeléshez. Abban az esetben, ha ez nem történik meg, a program hibaüzenetet dob felugró ablakban (*ábra 2.2*).

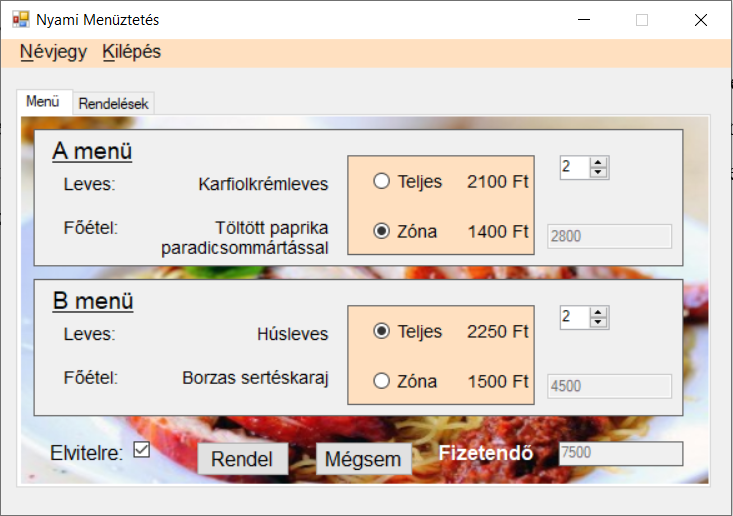
A sikeres rendeléshez a felhasználónak ezt követően ki kell választania egy számot a nyilak segítségével az adott menü szegmensének a jobb felső sarkában található beviteli mezőn keresztül. Ha ezt elmulasztja a felhasználó, azaz a mező értéke 0 marad, az alkalmazás hibaüzenetet dob egy felugró ablakban (*ábra 2.2*).

A “Menü” fülön lehetőség van arra is, hogy a fogyasztás módját kiválasszuk az “Elvitelre” felirat mellett található jelölőnégyzetre kattintva, aminek következtében az elvitel ára (200 Ft) hozzáadódik a végső rendeléshez.

Ábra 2. 2 - A rendelés elemeinek hibás kiválasztásakor felugró hibaüzenet

Abban az esetben, ha akármelyik menün belül az adag kiválasztás és az adag szám megadása sikeres, az adott menü szegmensének a jobb alsó sarkában megváltozik a szövegdoboz tartalma a fizetendő árra, illetve az egész “Menü” fül jobb alsó sarkában található mezőben megjelenik a kiválasztott menü(k) összesített ára, elvitellel együtt (*ábra 2.3*).

Graphical user interface, application, PowerPoint

Description automatically generatedEzt követően a “Rendel” gombra kattintva a program felugró ablakban értesíti a felhasználót a rendelés sikerességéről (*ábra 2.4*).

Ábra 2. 3 - A rendelés sikerességét jelző ablak

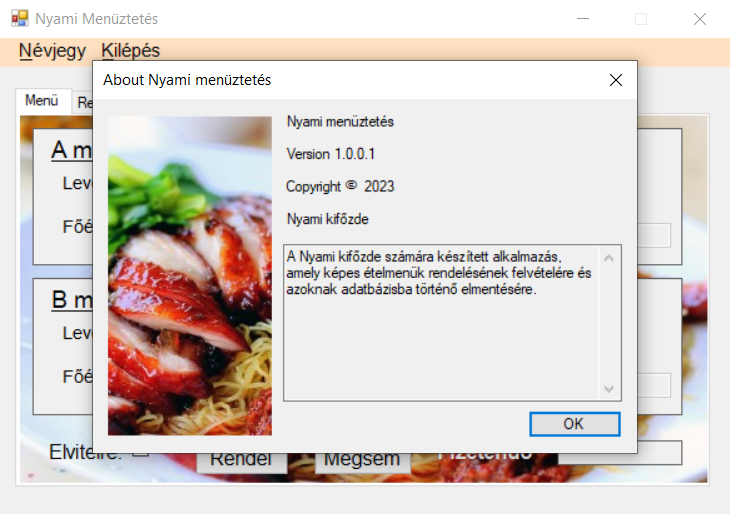
Ábra 2. 4 - A kiválasztott tételek összesített ára megtalálható a jobb alsó sarokban, részlet ár pedig az adott menü szövegdobozában

Graphical user interface, application, PowerPoint

Description automatically generatedA “Mégsem” gombra kattintva a felhasználó törölni tudja az eddig kiválasztott elemeket. A művelet sikerességéről felugró ablak értesíti a felhasználót (*ábra 2.5*).

Ábra 2. 5 - A kiválasztott menük törlésének sikerességét jelző ablak

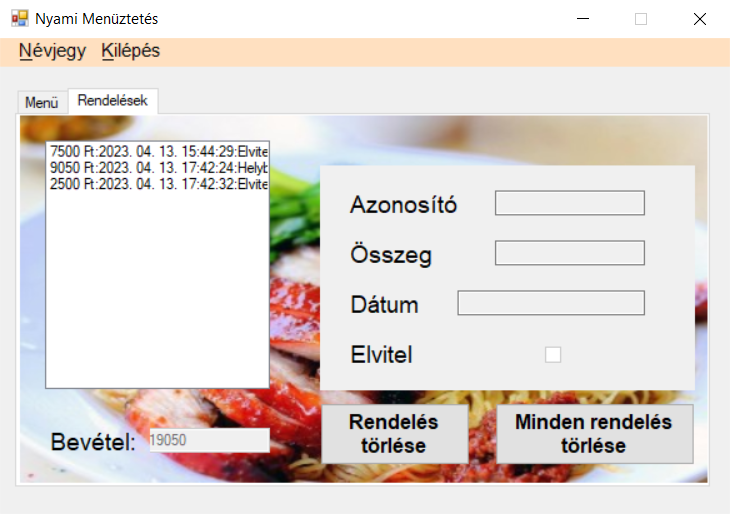
A felhasználónak bármelyik oldalon lehetősége van megnyitni az alkalmazás névjegy oldalát az “N” billentyű megnyomásával, vagy a “Névjegy” gombra kattintva, amit követően megjelenik a névjegy oldal (*ábra 2.6*).

A Névjegy ablakot a felhasználó be tudja zárni az “OK” gombra, vagy az ablak címsorába épített “X”-szel jelölt bezárás gombra kattintva, amelyek bár a Névjegy ablakot bezárják, az eredeti oldalt nem.

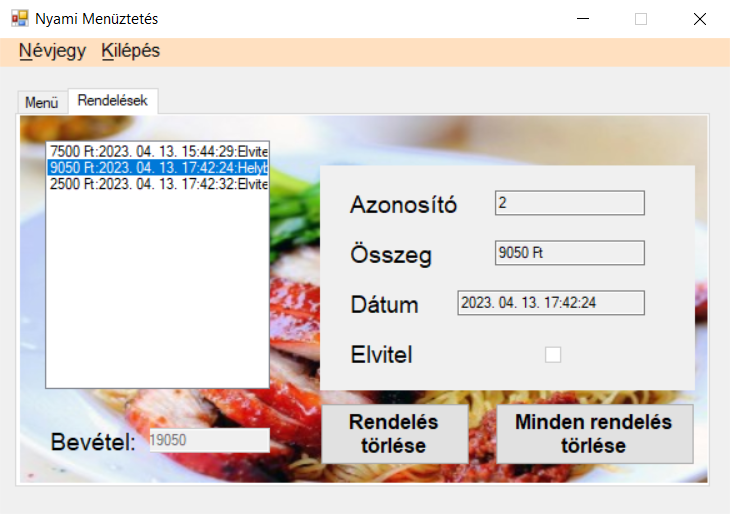
Ábra 2. 6 - Az alkalmazáshoz tartozó névjegy oldal

Ugyanúgy lehetősége van a felhasználónak bezárni az “Nyami Menüztetés” ablakot a “K” billentyű megnyomásával, a “Kilépés” gomb megnyomásával, vagy az ablak címsorába beépített “X”-szel jelölt bezárás gombra kattintva.

A “Rendelések” fülre kattintva megjelenik az eddig ezen a gépen leadott rendeléseket ábrázoló oldal (*ábra 2.7*).

A “Menü” fülön keresztül leadott sikeres rendelések automatikusan rögzítésre kerülnek az adatbázisba, hozzáadódnak a listához és a listában található rendelések árainak összege, amely a lap bal-alsó sarkában, a “Bevétel” felirat melletti szövegdobozban található, frissül.

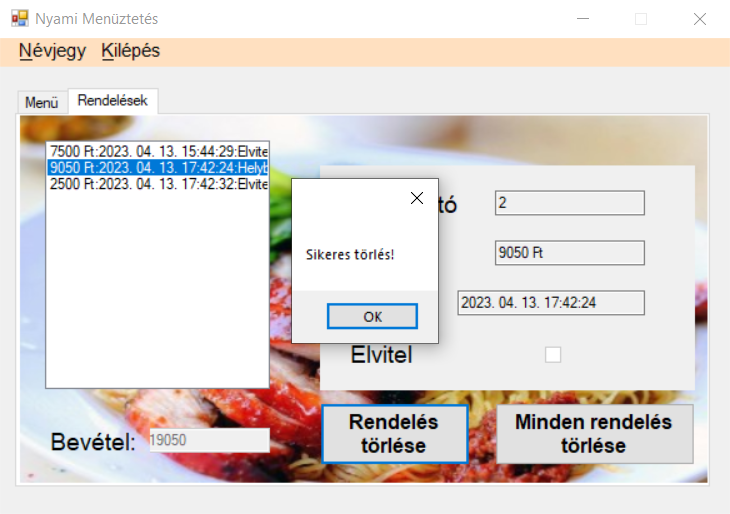
Ábra 2. 7 - A "Rendelések" fül tartalma

A listában megtalálhatóak a rendelések összegei, a rendelés leadásának pillanata, illetve, hogy hol fogyasztja a felhasználó kettősponttal (“:”) elválasztva. Ebben a listában “Elvitel” jelzi a bejelölt “elvitel” mezőt, a “Helyben” pedig a nem jelölt “Elvitel” mezőt. A lista bármelyik elemének kiválasztásával (abban az esetben, ha a lista nem üres) a lap jobb oldalán, a szürke hátterű területen található “Azonosító”-hoz, “Összeg”-hez, “Dátum”-hoz tartozó szövegdobozok és az “Elvitel”-hez tartozó jelölőnégyzet frissül, a kiválasztott rendelés azonosítójával (nem ismétlődő, egyedi szám), Összegével, a rendelés dátumával, és azzal, hogy elvitelre kérte-e az adott rendelést a felhasználó (*ábra 2.8*).

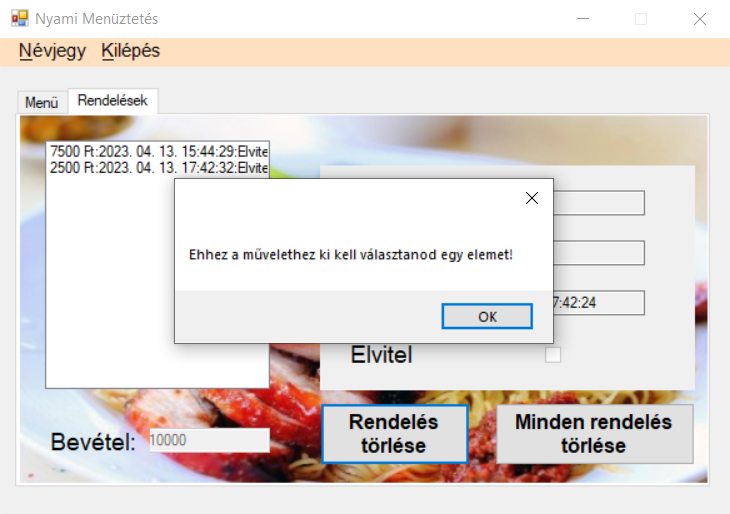
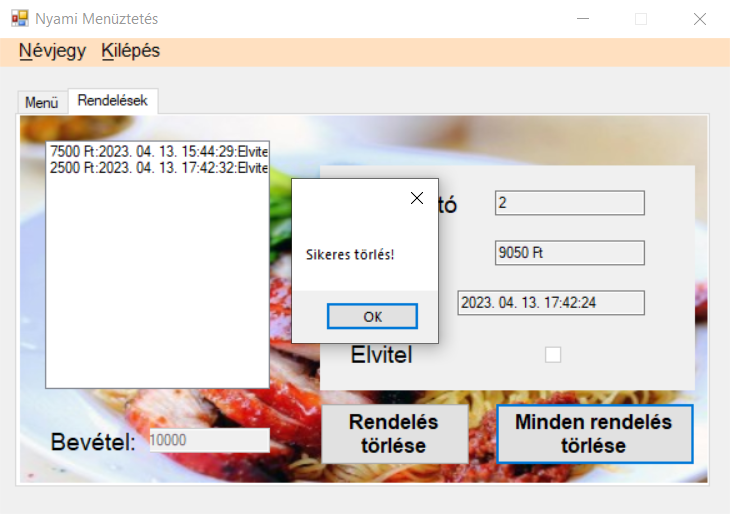
Ábra 2. 8 - A második rendelés kiválasztva

A felhasználó számára kettő gomb áll rendelkezésre a “Rendelések” fülön.

Az első gomb, a “Rendelés törlése”, melynek megnyomásával a felhasználó törli a kiválasztott lista elemet, és a művelet sikerességéről visszajelzést kap egy felugró ablakban (*ábra 2.9*).

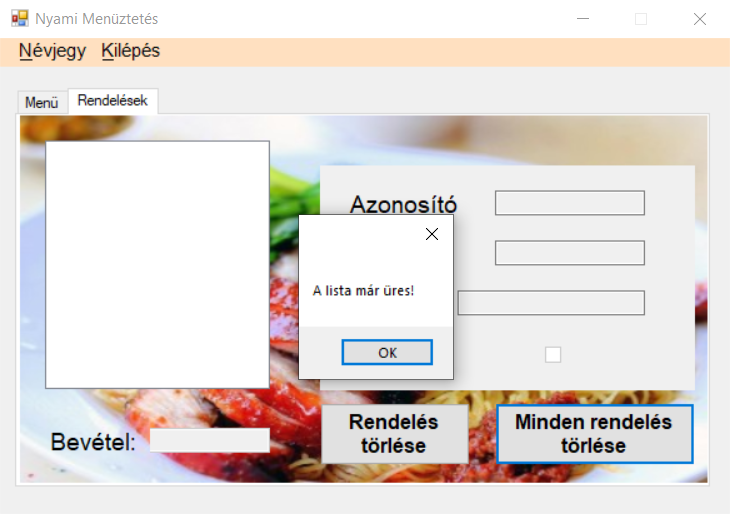
Abban az esetben, ha a felhasználó nem választott ki egy rendelést sem, akkor az alkalmazás hibaüzenetet dob egy felugró ablakban (*ábra 2.10*).

Ábra 2. 9 - A lista kiválasztott elemének sikeres törlését jelző ablak

A második gomb, a “Minden rendelés törlése”, melynek megnyomásával a felhasználó törli az összes rendelést a listából, illetve az adatbázisból is, és a művelet sikerességét követően azt az alkalmazás felugró ablakban jelzi (*ábra 2.11*).

Ábra 2. 11 – A lista kiválasztott elemének sikeres törlését jelző ablak

Ábra 2. 10 – A lista elemének sikertelen törlését jelző ablak

Abban az esetben, ha a lista már üres a gomb lenyomásakor, akkor ezt az alkalmazás felugró ablakban, egy hibaüzenet formájában jelzi a felhasználó számára (*ábra 2.12*).

Ábra 2. 12 – A rendelések sikertelen törlését jelző felugró ablak

3. Fejlesztői dokumentáció

3.1. Megoldási terv

1. Felhasználói esetek

A problémám fel volt vázolva az előre megkapott feladat specifikációban, melyben különösebb problémát nem jelentettek előre láthatólag a kért feature-ok.

A tervem a következő volt:

1. A MainForm létrehozása, a projekt konfigurálása.
2. A View osztály létrehozása, a MainForm felépítése.
   1. Az egyes oldalakhoz tartozó speciális osztályok létrehozása.
   2. Ideiglenes adatok beírása a specifikációban található mintához való összehasonlítás érdekében.
3. A Model osztály és az IModel interfész létrehozása, az eddig a View osztályba elhelyezett logikai elemek/metódusok áttelepítése.
4. A Névjegy Form létrehozása a beépített “about box” sablon alapján.
5. Az alkalmazásban felhasznált adatok eltárolására használható struct-ok (Items/Menu/Rendelesek) létrehozása, a hozzájuk tartozó statikus mezőkkel együtt.
6. A Persistence osztály és az IPersistence interfész létrehozása, az eddig a Model osztályba elhelyezett adatbeolvasó metódusok áthelyezése.
   1. Az alkalmazáshoz tartozó adatbázis és annak a tábláinak a létrehozása.
7. Az Objektum-Orientált programozás alapelveit követő osztály architektúra megvalósítása.

Az így létrehozott alkalmazásban a felhasználónak lehetősége van a következőkre (*ábra 3.1*):

* Választás több menü lehetőség között.
* A menükből a kívánt adagot kiválasztani.
* A menükből a kívánt számot kiválasztani.
* Annak jelzésére, hogy elvitelre kéri-e.
* A rendelés leadására.
* Az eddig kiválasztott elemek kitörlésére, az oldal “reset”-elésére.
* Az oldalról kilépésre.
* A névjegy oldal megnyitására.
* Az eddig leadott rendelések megtekintésére.
* Tetszés szerinti elem törlésére az eddig leadott rendelések közül.
* Az összes elem törlésére az eddig leadott rendelésekből.

Diagram

Description automatically generated

Ábra 3. 1 - Az alkalmazás use case UML diagrammja

1. Felhasználói felület

A felhasználói felület mintáját megkaptam a feladat specifikációban (*ábra 3.2, ábra 3.3*).

Ábra 3. 2 - A feladat leírásban található "Rendelések" oldal mintája

Ábra 3. 3 - A feladat leírásban található "menük" oldal mintája

Diagram

Description automatically generatedGraphical user interface

Description automatically generatedAz alkalmazás felépítéséhez továbbá felhasználtam egy saját készítésű képernyő tervet, mely bár kevésbé részletes, mint a megadott példák, ábrázolja az oldal felépítését (*ábra 3.4, ábra 3.5*).

Ábra 3. 5 - A rendelések oldal képernyőterve

Ábra 3. 4 - A Menü oldal képernyőterve

1. Követett alapelvek[[4]](#forr_4)

A program megírásakor elsődleges szempontom volt az általam írt kód leegyszerűsítése, illetve átláthatóvá tétele.

A feladatspecifikációban leírtakat követve továbbá fontos alapelvem volt az *Objektum elvű programozás*, melyet a leírásban meghatározott *View-Model-Persistence* architektúra felhasználásával, polimorfizmussal, és öröklődés igénybe vételével, illetve interfészek használatába vételével valósítottam meg[[3]](#forr_3).

*Azonosítónevek átláthatósága:* Az átláthatóság érdekében a program írása közben folyamatosan figyeltem arra, hogy az azonosítónevek (osztályok, struct-ok, változók) érthetőek legyenek.

*Lépésenkénti finomítás*: A programban a megírás után, illetve közben, külön figyelmet fektettem arra, hogy az általam már megírt kódot áttekintsem újra, lehetséges egyszerűsítési módszereket keresve. Ebben nagy segítségemre volt a program részeinek feladat szerinti tagolása.

1. Használt fejlesztési eszközök

Az alkalmazást Visual Studio 2022 IDE-ben írtam meg, MS Windows 10 operációs rendszeren. A program megírásához a C# programozási nyelvet használtam fel, melyet azért választottam, mert ezt a programozási nyelvet vettük igénybe az idei tanév során, melyből az általam írt osztályozó vizsgát is teszem, ezért ésszerű volt a választás. A .NET 7.0.4-es verzióját használtam a program megírásakor, mivel az volt a legfrissebb verzió.

A form megalkotásához használatba vettem a .NET Framework 4.7.2 verzióját, mely a legtöbb támogatást nyújtja a feladat elkészítésének időpontjában.

Az adatbázis kezeléshez az SQLite könyvtárat használtam fel, melyet a beépítettsége és a lekérdezési nyelve (SQL alapú) miatt választottam.

1. .NET Framework 4.7.2.[[5]](#forr_5)

Az alkalmazásomban használatba vettem a .NET Framework 4.7.2-es verzióját, mely a program megírásakor a legtöbb lehetőséget és a legtöbb támogatást nyújtó verzió. Továbbá ez a verzió az alapértelmezett verzió a Visual Studio 2022-ben.

1. SQLite[[6]](#forr_6)

Az adatbázis kezelést az SQLite könyvtárral oldottam meg, mely lehetőséget nyújt a lokális adatbázis tárolására, miközben ugyanakkora védelmet nyújt, mint a szerver adatbázissal dolgozó társai. További előny számomra a teljesítménye, mely más DBMS-ekkel szemben nem a memóriában tárolja az összes adatot. Az SQLite általam használt verziója a 3-as verzió volt, mely felhasználásával a kettő táblámat és annak tartalmait egy “data.sqlite” elnevezésű fájlban tároltam el.

1. Architektúra

Diagram

Description automatically generatedA program felépítésekor, a specifikációban leírtakat követve, View-Model-Persistence architektúrát használtam (*ábra 3.6*).

Ábra 3. 6 - A VMP architektúra komponens diagramja

1. Adatbázis felépítése

Az SQLite-en keresztül megvalósított adatbázisban a kettő tábla nem áll semmilyen kapcsolatban, egymástól teljesen függetlenek és önálló azonosítóval rendelkeznek (*ábra 3.7*).

Az *items* tábla a következő mezőket tartalmazza:

id – Egy maximum 20 karakteres szöveg, mely az adott elem egyedi azonosítóját tárolja el.

name – Maximum 20 karakteres szöveg, mely az adott elem nevét tárolja el.

cat – Maximum 20 karakteres szöveg, mely az adott elem kategóriáját (leves/főétel) tartalmazza.

price – Egész szám, az adott elem árát tartalmazza.

A *rendelesek* tábla a következő mezőket tartalmazza:

id – Egy egész szám, mely az adott rendelés egyedi azonosítóját tartalmazza.

price – Egész szám, mely az adott rendelés összegét tárolja.

date – Maximum 20 karakteres szöveg, mely a rendelés leadásának időpontját tárolja szöveg formátumban.

Graphical user interface

Description automatically generated with medium confidenceelvitel – Egész szám, mely egy bináris (0/1) értéket tárol attól függően, hogy a rendelést elvitelre kérték-e.

Ábra 3. 7 – Az adatbázisban használt táblák

3.2. Megvalósítás

Diagram

Description automatically generatedA megvalósításkor nem volt szükség a tervezéskor leírtakhoz képest különösebb változásokat eszközölnöm. A megvalósításkor csakis az általam figyelembe vett alapelvek követése érdekében változtattam az eredeti terv szerinti felépítésen, a változó (és osztály) neveken és az általam használt architektúrán belül felépített osztályok metódusain (*ábra 3.8*).

Ábra 3. 8 - Az alkalmazáshoz tartozó class UML diagram

3.3. Fejlesztés

Maga a projekt rengeteg fejlesztési lehetőséget tartalmaz önmagában, ebben a dokumentációban egyszer már hivatkoztam rá “prototípusként”, amely téves. Az általam megírt alkalmazás egy kis kiegészítéssel megállná helyét egy Vendég-Étterem felépítésű projektben a Vendég kliensként.

3.3.1. Továbbfejlesztés lehetőségei

Az alkalmazás tovább fejleszthető egy külső szerveren tárolt adatbázissal, melyben naponta frissül a napi menü, az eddig az SQLite-ben eltárolt rendeléseket tárolja, illetve egy külön táblán keresztül bejelentkezést tesz lehetővé. Ezen túl a projektet kiegészíteném egy admin felülettel is, amely a szerveren található rendeléseket tudja megtekinteni és kezelni. Továbbá, ahogy említettem, a vendég felület kiegészíthető felhasználói profillal (mely a rendeléseket és a rendelési adatokat őrzi meg), ahova a felhasználó be tud jelentkezni.

4. Összegzés

A projekt elkezdésekor egy célom volt, hogy egy olyan munkát tudjak bemutatni, ami a lehető legjobban reprezentálja a tudásomat melyet az év során az iskolán kívül és belül szereztem meg.

Összegezve a projektet, ez egy érdekes projekt volt számomra. Ez az első alkalom, hogy ekkora kiterjedésű dokumentációt írok, melynek szerkezeti felépítésében nagy hasznomra volt Beke Ágnes tanárnő szakdolgozata, melynek felhasználására szóbeli engedélyt adott, abban az esetben, ha felhasználáskor megjelölöm forrásként[[1]](#forr_1). Ezen kívül a projekt megírásakor még nem ismertem tervezési folyamatokat, kevés fogalmam volt a tervezési eszközökről, mint az UML, de szerencsére a végére sikerült egész jó tapasztalatot szereznem belőlük a Budapesti Műszaki Egyetem Országos Műszaki Információs Központ és Könyvtárának (BME OMIKK) rendszeres látogatásával. Az Objektum-Orientált programozási elmélet elsajátításában pedig nagy segítségemre volt Dr. Gregorics Tibor, aki az ELTE Informatikai Karának tanszékvezető egyetemi docense. Ő segített megérteni az OOP architektúrák, illetve az interfészek használatának mibenlétét.

A projekt megírásakor kihívást jelentett számomra a túlkomplikálás elkerülése, mely az utóbbi időben egyre több projektemet fertőzte meg, de úgy érzem a végére sikerült egy olyan kódot írnom, mely egyszerűségében versenyzik egy átlagos OOP programéval. További kihívás volt számomra, hogy ugyanebben az időszakban az érettségire, illetve az angol osztályozóvizsgára készültem, melyeket kénytelen voltam előnyben részesíteni. Így ugyan kevesebb időm maradt a projekt tényleges elkészítésére, de amikor mégis tudtam dolgozni rajta, azt már egy megfontoltabb, türelmesebb állapotban tudtam, a vizsgákat körbe ölelő stressztől megszabadulva.

Az alkalmazás elkészítése magában hordozta a tanulás (vagy tudás felelevenítés) izgalmát, mely egészen szórakoztatóvá tette a rajta való munkát. Ezért is ilyen hosszú a dokumentáció, mellyel valószínűleg kicsit túllőttem a célon, mert élveztem dolgozni rajta. A tervezésnek sajnos túl későn kezdtem neki, ezért nem volt időm szekvencia, illetve állapotgép diagrammot készíteni, de azért így is büszke vagyok a tervezésre szánt órákra, melyek közül egyet sem érzek elpazaroltnak.

A 3.3-as fejezetben megemlített fejlesztési lehetőségek már szinte az első elolvasáskor felvillantak a fejemben, sajnos azonban úgyszintén az időhiány miatt nem tudtam semelyiket eszközölni. Elismerem, van olyan a lehetséges funkciók közül, melynek a megvalósításához szükséges tudással nem rendelkezek még, de ebből tanulva tervezem megtanulni ezeket az elkövetkezendő időszakban. Ilyen például a C++ nyelv, a más operációs rendszerekhez való kompatibilitáshoz, az ahhoz tartozó csomagok, a szerveren található adatbázis elérése, hálózati protokollok (melyeket 10. évfolyamon már ugyan tanultunk, de kénytelen vagyok feleleveníteni) és a C# projekt teszteléséhez felhasználható eszközök mint a xUnit, MsUnit, NUnit, JetBrains Aqua és hasonlók. Továbbá az alkalmazáshoz szerettem volna telepítőt is írni a Wix használatával, de kénytelen voltam kihagyni.

Felhasznált források

[1] – Beke Ágnes – Rendeléselvétel és üzletmanagement a vendéglátóiparban

[2] – [nyami\_feladat.pdf](file:///C:\Users\okvik\Documents\Programozás%20projektek\C%23\2022-23\osztalyozo_23_OKV\nyami_feladat.pdf)

[3] – [Dr. Gregorics Tibor docens, ELTE IK Programozáselmélet és Szoftvertechnológia Tanszék](https://people.inf.elte.hu/gt/)

[4] - [ELTE IK - Programozás alapok, Programkészítési elvek](http://progalap.elte.hu/downloads/seged/eTananyag/lecke24_lap1.html)

[5] - [.NET Framework (4.7.2.) dokumentáció](https://learn.microsoft.com/en-us/dotnet/framework/)

[6] - [SQLite dokumentáció](https://www.sqlite.org/docs.html)

Ábrajegyzék

*[Ábra 2. 1 - Az alkalmazás elindításakor megjelenő ablak](#abra_2_1)*

[*Ábra 2. 2 - A rendelés elemeinek hibás kiválasztásakor felugró hibaüzenet*](#abra_2_2)

[*Ábra 2. 3 - A kiválasztott tételek összesített ára megtalálható a jobb alsó sarokban, részlet ár pedig az adott menü szövegdobozában*](#abra_2_3)

[*Ábra 2. 4 - A rendelés sikerességét jelző ablak*](#abra_2_4)

[*Ábra 2. 5 - A kiválasztott menük törlésének sikerességét jelző ablak*](#abra_2_5)

[*Ábra 2. 6 - Az alkalmazáshoz tartozó névjegy oldal*](#abra_2_6)

[*Ábra 2. 7 - A "Rendelések" fül tartalma*](#abra_2_7)

[*Ábra 2. 8 - A második rendelés kiválasztva*](#abra_2_8)

[*Ábra 2. 9 – A lista kiválasztott elemének sikeres törlését jelző ablak*](#abra_2_9)

[*Ábra 2. 10 – A lista elemének sikertelen törlését jelző ablak*](#abra_2_10)

[*Ábra 2. 11 – A rendelések sikeres törlését visszajelző felugró ablak*](#abra_2_11)

[*Ábra 2. 12 – A rendelések sikertelen törlését jelző felugró ablak*](#abra_2_12)

[*Ábra 3. 1 - Az alkalmazás use case UML diagrammja*](#abra_3_1)

[*Ábra 3. 2 - A feladat leírásban található "Rendelések" oldal mintája*](#abra_3_2)

[*Ábra 3. 3 - A feladat leírásban található "menük" oldal mintája*](#abra_3_3)

[*Ábra 3. 4 - A Menü oldal képernyőterve*](#abra_3_4)

[*Ábra 3. 5 - A rendelések oldal képernyőterve*](#abra_3_5)

[*Ábra 3. 6 - A VMP architektúra komponens diagramja*](#abra_3_6)

[*Ábra 3. 7 – Az adatbázisban használt táblák*](#abra_3_7)

[*Ábra 3. 8 - Az alkalmazáshoz tartozó class UML diagram*](#abra_3_8)