

RUANDER Oktatási kft.

SZAKDOLGOZAT

Bodor Gyula
Szoftverfejlesztő Okj

Budapest
2021

RUANDER Oktatási Kft.

Event Organizer Application

Konzulens:
Csolti Péter

Készítette:
Bodor Gyula
Szoftverfejlesztő Okj

Budapest, 2021

Tartalomjegyzék

| | |
|--|-----------|
| 1. Bevezetés, a probléma rövid ismertetése | 1 |
| 2. Felhasználói dokumentáció | 2 |
| 2.1. Rendszerkövetelmény: | 2 |
| 2.2. Adminisztrátori feladatok, a program beüzemelése: | 2 |
| 2.3. Felhasználói útmutató: | 2 |
| 2.4. Bejelentkezés: | 3 |
| 2.4.1. Hibaüzenetek: | 3 |
| 2.5. Főmenü: | 4 |
| 2.6. Új árajánlat készítése: | 5 |
| 2.7. Árajánlat megtekintése: | 8 |
| 2.8. Árajánlatok megtekintése: | 10 |
| 3. Fejlesztői dokumentáció | 11 |
| 3.1. Rendszerterv, tervezés: | 11 |
| 3.2. Adatbázis: | 12 |
| 3.2.1. Táblák: | 12 |
| 3.3. Adatkezelés, a program kialakítása. | 16 |
| 3.3.1 Osztályok | 17 |
| 3.3.2 Felhasználó felület, Formok és működés: | 26 |
| 3.4. Tesztelés: | 30 |
| 4.Tovább Fejlesztés: | 32 |
| 5.Összegzés: | 32 |
| 6.Irodalomjegyzék: | 33 |

1. Bevezetés, a probléma rövid ismertetése

A problémát egy rendezvényszervezéssel foglalkozó vállalkozó ismerősöm vetette fel, aki egy rendezvényházat üzemeltet, a konkrét probléma az, hogy egy releváns árajánlatot elkészíteni rengeteg számolással jár az ő szakmájában. A fő szempont hogy az egyéni igényeknek regisztrálását lehetővé téve a program elvégezze a szükséges számításokat és összeállítsa az árajánlatot a kiválasztott ételek és italcsomag eladási árainak összesítésével, az egy főre leosztott árával, a rendezvényház bérleti díjával, a szükséges munkaerő meghatározásával és azoknak bérével, majd az árajánlatot készítse el PDF formátumban a tételek listába szedésével, megformázva és tetszőleges helyre való mentéssel lehetővé téve az árajánlat további használatát. A második fő szempont hogy a beszerzési árak alakulását mind összesített mind egy főre eső részét és a nyereséget, kiszámolja és megjelenítse a program. Mindezekkel a megoldásokkal automatizálunk és központosítunk egy olyan folyamatot, amit alapvetően több program pl.: Excel és Word állományok egyidejű és nehezen nyomon követhető használatával és nem melleleg nagy emberi hibalehetőséggel került elvégzésre.

2. Felhasználói dokumentáció

2.1. Rendszerkövetelmény:

Operációsrendszer: minimum Windows 7 (32-bit vagy 64-bites változat) vagy ennél újabb

Windows operációs rendszer (Windows 8, 10).

Memória: 3-4 GB memória.

Processzor: Pentium(R) Dual-Core CPU E5200/2.5 GHz vagy újabb.

2.2. Adminisztrátori feladatok, a program beüzemelése:

Első lépés: Az adatbázis telepítése. Mindenféléképpen szükségünk lesz egy MySQL adatbázis szerverre, ami ebben az esetben lehet akár egy sima teszt szerver is. 32-bites rendszerek esetében a WAMP szerverkörnyezet telepítését ajánlom, míg 64-bites környezetben (Windows 7.3.27-es verziótól) a XAMPP szerverkörnyezetet.

A linken található videó és leírás segítségével töltsük le és végezzük el a feltelepítés:

- XAMPP: <https://www.youtube.com/watch?v=-f8N4FEQWyY>
- WAMP: <https://www.kelecsenyi.info/joomla/tippek-trukkok-gyakorlat/wampserver-3-0-4-telepitese-windows-10-re>

Ezek után indítsuk el a MySQL szervert (mindig a programhasználatokor futnia kell a szervernek).

Második lépés: Nyissuk meg a phpmyadmin felületet, amit a <http://localhost/phpmyadmin/> linken keresztül érünk el. Bejelentkezéshez szükséges felhasználónév: root , a jelszó mező pedig maradhat üres, majd lépünk be. Bent vagyunk az adatbázisban. Most kattintsunk az importálás fülre, ezután kattintsunk a „Fájl kiválasztása” gombra és rendeljük be a CD-n az EventOrganizerApp/ mappában található „eventorganizer.sql” állományt, majd nyomjuk az Indítás gombra, ha ez nem működik, akkor menjünk az „SQL” fülre a phpmyadmin felületen, nyissuk meg az „eventorganizer.sql” állományt és a benne található kódot másoljuk be a lekérdezés mezőbe majd nyomjunk az „Indítás” gombra. További teendők itt nincs, bezárhatjuk a phpmyadmin felületet.

Megjegyzés: Windows 7 32-bites környezetben nincs mindig megfelelő verziójú .NET keretrendszer telepítve, ha ilyen hibával találkozunk, akkor telepítsük a .NET 4.5.2-es verzióját, az alábbi linkről egyszerűen letölthető és telepíthető.

- <https://www.microsoft.com/en-us/download/details.aspx?id=42642>

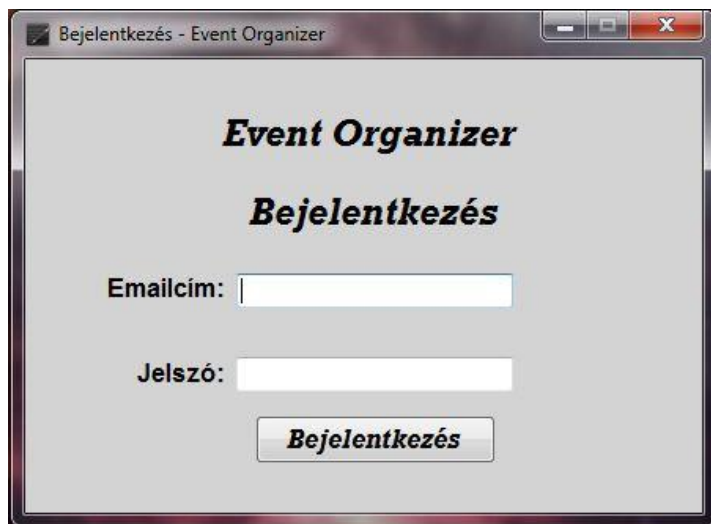
2.3. Felhasználói útmutató:

A sikeres telepítést követően a programot a CD EventOrganizerApp\EventOrganizerApp\bin\Debug\ könyvtárban található

EventOrganizerApp.exe fájlal tudjuk elindítani vagy a CD-ről kimásolhatjuk a merevlemezünkre a EventOrganizerApp mappát és ugyan ezzel az EventOrganizerApp.exe állománnyal elindíthatjuk a programot.

2.4. Bejelentkezés:

A program elindítását követően megnyílik a Bejelentkezés ablak (lásd: 1. ábra).



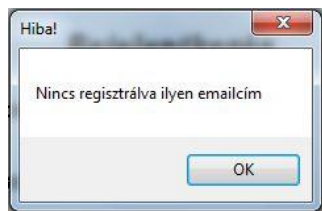
(1. ábra) Bejelentkezés ablak

Alapértelmezett bejelentkezési adatok:

- Emailcím: admin@admin.hu
- Jelszó: Admin@0101

Itt az emailcím és a jelszó mező helyes kitöltése után a bejelentkezés gombra kattintva tudunk belépni a Főmenübe (lásd 7. ábra), amennyiben a mezők helytelenül vagy nem lettek kitöltve vagy nem megfelelőek a bejelentkezési adatok a program erről tájékoztatást ad (lásd: 2, 3, 4, 5, ábra).

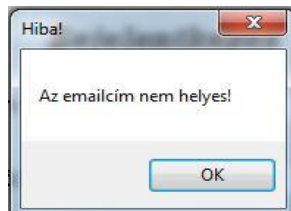
2.4.1. Hibaüzenetek:



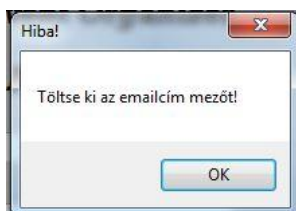
(2. ábra)



(3. ábra)



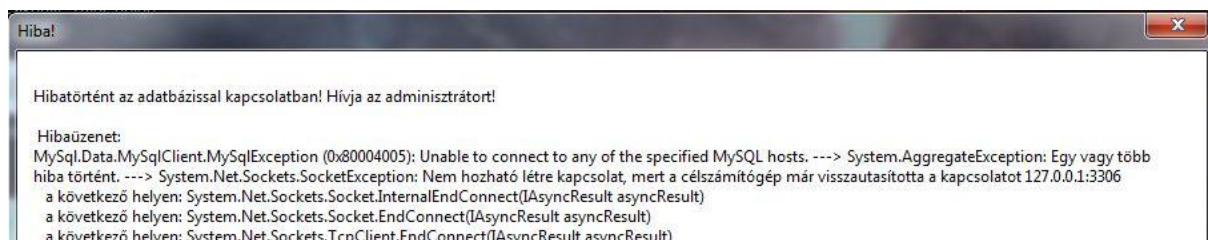
(4. ábra)



(5. ábra)

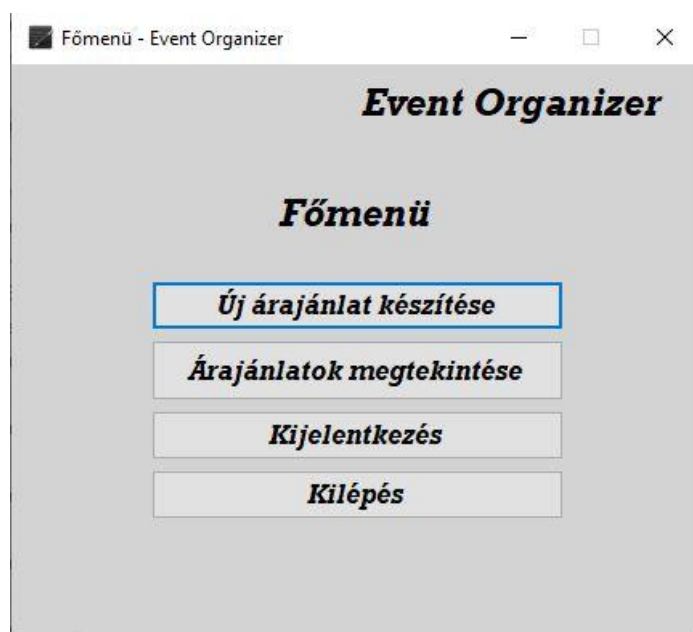
Ezeket a hibaüzeneteket látva egy „OK” gomb megnyomásával el tudjuk tüntetni, és újra meg tudjuk próbálni a bejelentkezést (a program további használata alatt is lesznek ehhez hasonló hibaüzenetek, azoknál is érvényes, hogy az „OK” gomb megnyomása után javíthatjuk a helytelen adatkitöltést.).

Abban az esetben, ha a program hibaüzenete az alábbi módon (lásd 6. ábra) vagy ehhez hasonlóan néz ki, akkor az adminisztrátor segítségét kell kérni a hiba helyreállításához, ez érvényes a további használat alatt felugró ugyan ilyen ablakoknál is!



(6. ábra)

2.5. Főmenü:



(7. ábra)

Miután megtörtént a sikeres bejelentkezés, felugrik a Főmenü ablaka (lásd 7. ábra). Ahol kiválaszthatjuk mit is szeretnénk tenni, Az „Új árajánlat készítése” gomb megnyomására egy teljesen új árajánlatot hozhatunk létre, míg az „Árajánlatok megtekintése” gombra kattintva a már elkészített árajánlatokat tudjuk újra megnézni. A „Kijelentkezés” gomb hatására a program nem záródik be csak visszadob minket a bejelentkezés ablakra, míg a „Kilépés” gomb hatására a program bezárul.

2.6. Új árajánlat készítése:

The screenshot shows the 'Event Organizer' application window with the title 'Új árajánlat készítése - Event Organizer'. The form is divided into several sections:

- Rendezvény adatai** (Event details):
 - *Rendezvény típusa: Esküvő (dropdown)
 - *Vendégek száma: [] fő (input field)
 - *Időpont: 2021. február 26. (calendar icon)
 - Továbbhaladáshoz szükséges a "Vendégek száma" mező kitöltése (note)
- Megrendelő adatai** (Orderer details):
 - *Megrendelő neve: [] (input field)
 - *Megrendelő tel. száma: [] (input field)
 - Megrendelő emailcime: [] (input field)
- Menü / ételcsomag szerkesztése** (Menu / Food package editing):
 - Előétel szerkesztése: Húsleves (dropdown), [] adag, +, - buttons
 - Főétel szerkesztése: Töltött káposzta (dropdown), [] adag, +, - buttons
 - Köret szerkesztése: Párolt jászmin rizs (dropdown), [] adag, +, - buttons
 - Tálak szerkesztése: Sajjtasztal (dropdown), [] adag, +, - buttons
 - Savanyúság szerkesztése: Friss saláta (dropdown), [] adag, +, - buttons
 - Desszert szerkesztése: Rígó Jancsi (dropdown), [] adag, +, - buttons
- Italcsomag szerkesztése** (Beverage package editing):
 - Italok elszámolása: fogyasztás alapú elszámolás (dropdown)
 - A rendezvény végén kerülnek elszámolásra az italok fogyasztás alapján 0 Ft/fő (note)
- Alkalmazottak listája** (Staff list):
 - Table with columns: Munkakör, Darab
 - Buttons: Előzetes számítás, Árajánlat elkészítése
- A Teljes menü lista** (The full menu list):
 - Table with columns: Étel neve, 1 adag ára, Hány adag?
 - Buttons: Főmenü, Kilépés

(8. ábra)

Kezdjük bele az árajánlat megírásába. Értelem szerűen adjuk meg a rendezvény típusát és a vendégek számát (ameddig a vendégek száma nincs megadva addig az ételek és az italcsomag regisztrációja nem elérhető). Kezdjük el hozzáadni az ételeket a menühöz. A legördülő menüből kiválasztjuk az ételt, beírjuk hány adagot szeretnénk elszámolni belőle és rányomunk a „+” gombra (lásd: 9. ábra), így egyből bekerül a teljes menü lista táblázatába (lásd: 10. ábra)

This close-up shows the 'Előétel szerkesztése' section. It features a dropdown menu with 'Brokkoli krémleves' selected, followed by an input field containing the number '35', the word 'adag', and two buttons labeled '+' and '-'.

(9. ábra)

| A Teljes menü lista | | | |
|---------------------|--------------------|------------|------------|
| | Étel neve | 1 adag ára | Hány adag? |
| ▶ | Brokkoli krémleves | 680 | 35 |
| * | | | |

(10. ábra)

Itt megjelenik az 1 adag eladási ára is. Ugyan ezzel a folyamattal ki is lehet vonni a már bekerült étel adaginak számából, csak a „-” gombra kattintva, ha teljesen ki szeretnék törölni egy adott tételt a menüből akkor kiválasztjuk az étel nevét beírjuk hány adag van belőle a menübe és rányomunk a „-” gombra így. A következő lépés az italcsomag kiválasztása. A legördülő menüből válasszuk ki a kívánt italcsomag majd a kiválasztás hatására a program tájékoztat minket annak leírásáról és eladási áráról 1 főre értelmezve (lásd: 11. ábra).

Italcsomag szerkesztése

Italok elszámolása: B - korlátlan italfogyasztás ▼

- Sörök + Borok + Üdítők

4990 Ft/fő

(11. ábra)

Ezzel a lépéssel mindent olyan értéket megadtunk, ami elengedhetetlen az árajánlat kiszámításához. A szükséges alkalmazottak összetételéről is tájékoztat minket a program (lásd: 12. ábra), ezzel is segítve a rendezvényszervezőt.

| Alkalmazottak listája | | |
|-----------------------|-------------|-------|
| | Munkakör | Darab |
| ▶ | pultos | 1 |
| | felhasználó | 4 |
| | szakács | 2 |
| * | | |

(12. ábra)

Ezen a ponton lehetőségünk van elvégeztetni egy előzetes számítást mielőtt mentenénk az árajánlatot, ez jól jöhet abban az esetben, ha a megrendelővel tárgyal a rendezvényszervező és együtt állítják össze valós időben a menüt, mivel itt megjelennek a számítások mind 1 főre és az teljes létszámmal egyaránt, és persze a profit is megjelenítésre kerül (lásd: 13. ábra).

Rendezvény adatai

*Rendezvény típusa: Születésnap

*Vendégek száma: 35 fő

*Időpont: 2021. február 28.

Továbbhaladáshoz szükséges a "Vendégek száma" mező kitöltése

Megrendelő adatai

*Megrendelő neve:

*Megrendelő tel. száma:

Megrendelő emailcime:

Menü / ételcsomag szerkesztése

Előétel szerkesztése: Brokkoli krémleves ☐ adag + -

Főétel szerkesztése: Sült csirkecomb ☐ adag + -

Köret szerkesztése: Hasábburgonya ☐ adag + -

Tálak szerkesztése: Sonkaválogatás ☐ adag + -

Savanyúság szerkesztése: Friss saláta ☐ adag + -

Desszert szerkesztése: Joghurt szelet ☐ adag + -

A Teljes menü lista

| Étel neve | 1 adag ára | Hány adag? |
|--------------------|------------|------------|
| Brokkoli krémleves | 680 | 35 |
| Sült csirkecomb | 990 | 35 |
| Hasábburgonya | 420 | 35 |
| Sonkaválogatás | 1790 | 35 |
| Friss saláta | 390 | 35 |
| Joghurt szelet | 430 | 35 |

Italok elszámolása: B - korlátlan italfogyasztás

- Sörök + Borok + Üdítők
4990 Ft/fő

Alkalmazottak listája

| Munkakör | Darab |
|-------------|-------|
| pultos | 1 |
| felszolgáló | 4 |
| szakács | 2 |

Előzetes számítás **Árajánlat elkészítése**

Előzetes árajánlat számítása

Menü ára 1 főre: 4700.- Ft/fő Menü ára a teljes létszámmal: 164500.- Ft

Italok ára 1 főre: 4990.- Ft/fő Italok ára a teljes létszámmal: 174650.- Ft

Menü és italok ára 1 főre: 9690.- Ft

Menü és italok teljes ára: 339150.- Ft

Munkaerő teljes bére: 109000.- Ft

Helybérleti díj: 39990.- Ft

Rendezvény ára összesen: 488140.- Ft Profit: 270790.- Ft

(13. ábra)

Abban az esetben, ha előzetes számításra használjuk a programot, nem szükséges kitölteni a Megrendelő adatait viszont, ha el szeretnénk menteni az árajánlatot akkor mindenféleképpen ki kell töltenünk a „Megrendelő nevét”, „Megrendelő Telefonszámát” és a „Rendezvény Dátumát”, az emailcím opcionális. Ezután rányomva az „Árajánlat elkészítése” gombra tovább haladhatunk az Árajánlat megtekintése ablakra (lásd: 14. ábra).

Megjegyzés: A „Kilépés” gomb bezárja a programot a „Főmenü” gombbal pedig visszatérhetünk a Főmenübe.

2.7.Árajánlat megtekintése:

Árajánlat megtekintése

Árajánlat

Rendezvény típusa: Születésnap
Vendégek száma: 35
Rendezvény dátuma: 2021-02-28

Event Organizer

Megrendelő neve: Mekk Elek
Megrendelő telefonszáma: 06203145659
Megrendelő emailcime: mekkelek@outlook.hu

Ételek listája:

| Étel neve | Adag |
|--------------------|------|
| Brokkoli krémleves | 35 |
| Sült csirkecomb | 35 |
| Hasábburgonya | 35 |
| Sonkaválogatás | 35 |
| Friss saláta | 35 |
| Iskolai csokoládé | 35 |

Alkalmazottak listája:

| Munkakör | Darab |
|-------------|-------|
| pultos | 1 |
| felszolgáló | 4 |
| szakács | 2 |
| * | |

Italcsomag: B - korlátlan italfogyasztás - Sörök + Borok + Üdítők

Menü eladási ára 1 főre: 4700.- Ft
Menü eladási ára a teljes létszámra: 164500.- Ft
Italok eladási ára 1 főre: 4990.- Ft
Italok eladási ára a teljes létszámra: 174650.- Ft
Menü és italok eladási ára 1 főre: 9690.- Ft
Menü és italok teljes eladási ára: 339150.- Ft
Munkaerő teljes bére: 109000.- Ft
Helybérleti díj: 39990.- Ft

Menü beszerzési ára 1 főre: 2360.- Ft
Menü beszerzési ára a teljes létszámra: 82600.- Ft
Italok ára 1 főre: 3850.- Ft
Italok beszerzési ára a teljes létszámra: 134750.- Ft
Menü és italok beszerzési ára 1 főre: 6210.- Ft
Menü és italok teljes beszerzési ára: 217350.- Ft

Rendezvény ára összesen: 488140.- Ft
Profit: 270790.- Ft

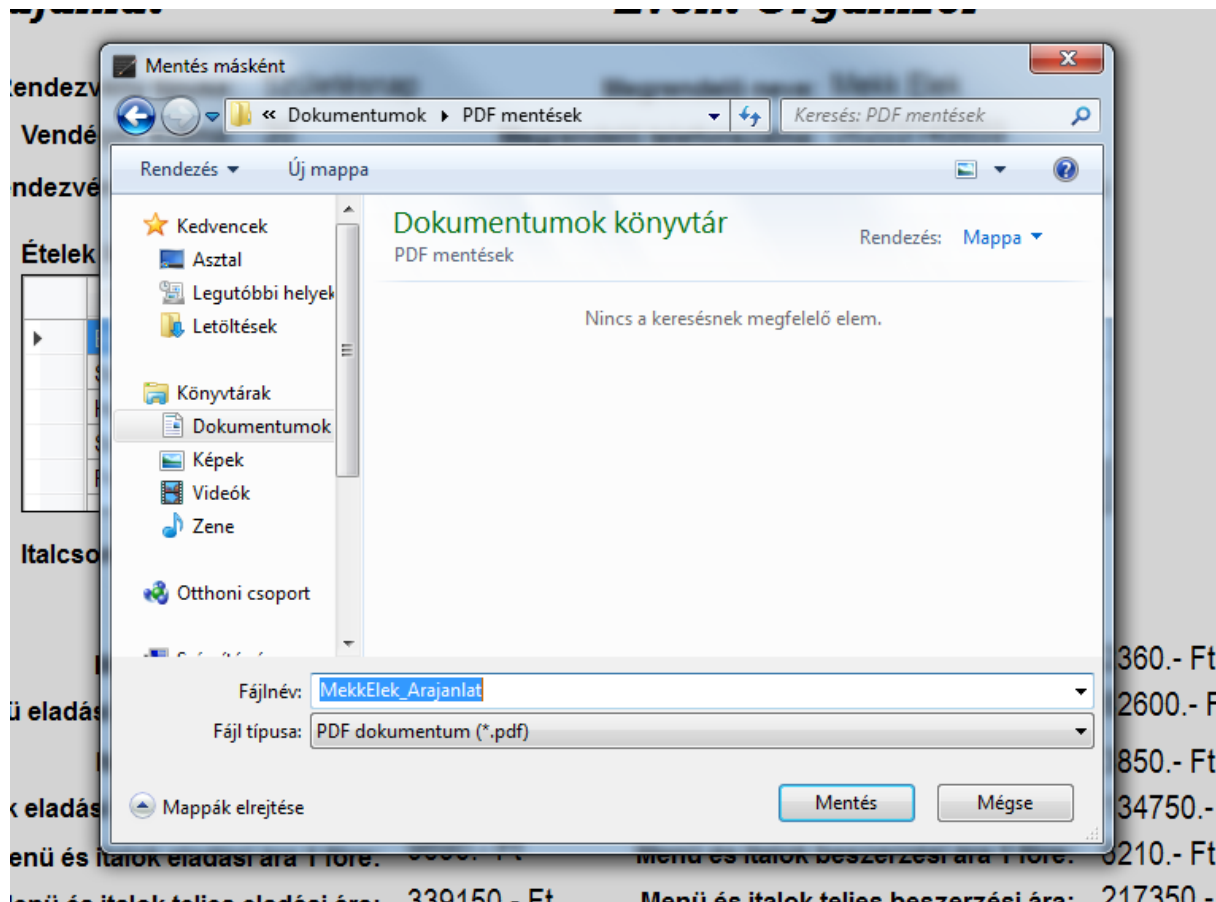
Főmenü
Kilépés

PDF készítése
Új árajánlat készítése
Árajánlatok megtekintése

(14. ábra)

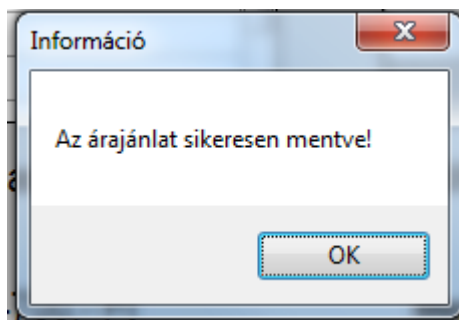
Az árajánlat megtekintése ablak minden számított értéket, adatot megjelenít és leközi a rendezvényszervező számára. Itt a felhasználó újra megnézheti milyen ételekből áll össze a menü, milyen italcsomagot választott az ügyfél, hogy mennyi és milyen típusú alkalmazottakra lesz szüksége a rendezvény lebonyolításához. Természetesen megjeleníti leglényegesebb adatokat úgy, mint az eladási árak részletezése a baloldalon és a beszerzési árak részletezése a jobb oldalon. A „PDF készítése” gomb megnyomása hatására előugrik egy „Mentés Másként”

ablak ahol a „fájlnév” mezőbe csak a PDF állománynak szánt nevet kell megadni, automatikusan hozzárendeli „.pdf” kiterjesztést (lásd 15. ábra).



(15. ábra)

A mentést követően a program tájékoztat, hogy elkészült a PDF fájl sikeres mentése (lásd.: 16. ábra).



(16. ábra)

Azután hogy elkészítettük sikeresen az árajánlatot, megtaláljuk az általunk megadott helyen és máris küldhetjük tovább emailben.

Árajánlat

Megrendelő neve: Mekk Elek

Megrendelő telefonszáma: 06203145659

Megrendelő emailcime: mekkelek@outlook.hu

Rendezvény típusa: Születésnap

Rendezvény időpontja: 2021-02-28

Ételek listája:

- 35 adag -- Brokkoli krémleves
- 35 adag -- Sült csirkecomb
- 35 adag -- Hasábburgonya
- 35 adag -- Sonkaválogatás
- 35 adag -- Friss saláta
- 35 adag -- Joghurt szelet

Italcsomag: B - korlátlan ital fogyasztás

- Sörök + Borok + Üdítők

Ételek ára 1 főre: 4700.- Ft

Ételek ára teljes létszáma: 164500.- Ft

Italok ára 1 főre: 4990.- Ft

Italok ára teljes létszáma: 174650.- Ft

Ételek és italok ára 1 főre: 9690.- Ft

Ételek és italok ára teljes létszáma: 339150.- Ft

Munkaerő bére: 109000.- Ft

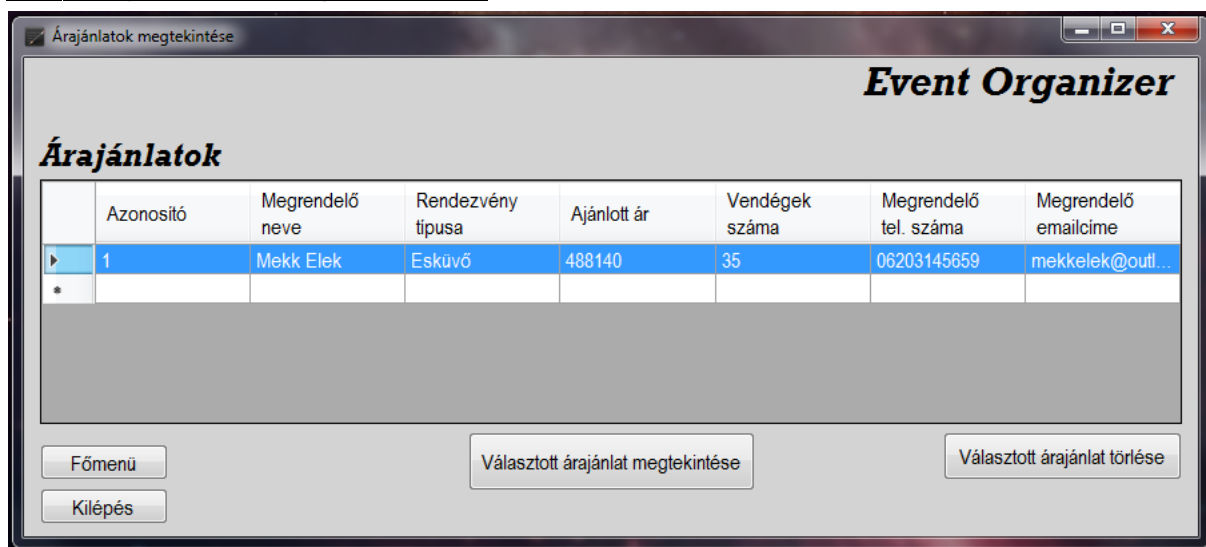
Rendezvényház bérleti díja: 39990.- Ft

A rendezvény teljes megajánlott ára: 488140.- Ft

(17. ábra)

Az árajánlat megtekintése ablakon belül további funkciók vannak a kilépésre, a Főmenübe való visszatérésre, egy újabb árajánlat készítésére és az eddig elkészített árajánlatok megtekintésére.

2.8. Árajánlatok megtekintése:



(18. ábra)

Ez az ablak lehetőséget ad az eddig elkészített árajánlatok újra megtekintésére vagy törlésére. Válasszuk ki a listából az árajánlatot majd kattintsunk értelem szerűen a „Választott árajánlat megtekintésére” gombra (ebben az esetben megjelenik az árajánlat lásd: 14. ábra) vagy a „Választott árajánlat törlése” gombra(ebben az esetben a program rákérdez még egyszer hogy biztosan törölni szeretnék-e az árajánlatot).

3. Fejlesztői dokumentáció

3.1. Rendszerterv, tervezés:

A tervezésnél szükségszerű volt mindennek előtt elkérni az alapvető ételkínálatról, italkínálatról, szóló dokumentumot, és pontosítani, hogy milyen adatok szerepeljenek az árajánlaton, amik alapján megkezdődhetett a tervezés(lásd 19. 20. ábra).

| Név | beszár | elár |
|---|--------|------|
| Húsleves | 465 | 780 |
| Hideg meggyleves | 375 | 680 |
| Brokkoli krémleves | 380 | 680 |
| Zöldségleves | 495 | 680 |
| Palócleves | 585 | 780 |
| Mátrai borzaska | 690 | 1290 |
| Sült csirkecomb | 270 | 990 |
| Rántott csirkemell | 540 | 1290 |
| Natúr csirkemell | 490 | 1290 |
| Rántott Sajt | 290 | 1290 |
| Fűszeres tarja | 640 | 1290 |
| Cigánypecsenye | 590 | 1290 |
| Holstein szelet | 570 | 1290 |
| Töltött káposzta | 325 | 990 |
| Rántott fogasfilé | 610 | 1490 |
| Rigó Jancsi | 215 | 430 |
| Joghurt szelet | 195 | 430 |
| Gyümölcskosár | 240 | 390 |
| Zserbó szelet | 230 | 430 |
| Panna Cotta | 190 | 390 |
| Sajtaszta (Karaván, Pannónia, Márványsajt, Caman) | 1490 | 2190 |
| Sonkaválogatás (Tiroli sonka, Serranó sonka, Párm) | 1320 | 1790 |
| Paraszt tál (Magyar parasztszonda, Parasztkolbász, | 1439 | 2190 |
| Tengeri tál (Garnéla, Füstölt lazacfilé, Feketekag | 1490 | 2490 |
| Vega tál (Grillezett zöldségek: Cukkini, Padlizsán, f | 1390 | 1890 |
| Friss saláta | 135 | 390 |
| Vecsesi savanyúság | 89 | 390 |
| Párolt Jázmin rizs | 60 | 420 |
| Hasábburgonya | 60 | 420 |

(19. ábra)

| Italcsomagok | | beszár | eladár |
|------------------------------|--|--------|--------|
| Fogyasztás alapú elszámolás | A rendezvény végén kerülnek elszámolásra az italok | 0 | 0 |
| A- korlátlan italfogyasztás | Sörök + Borok + Rövid italok + üdítők | 4890 | 6900 |
| B- korlátlan italfogyasztás | Sörök + Borok + üdítők | 3850 | 4990 |
| C - korlátlan italfogyasztás | Üdítők | 1370 | 2990 |

(20. ábra)

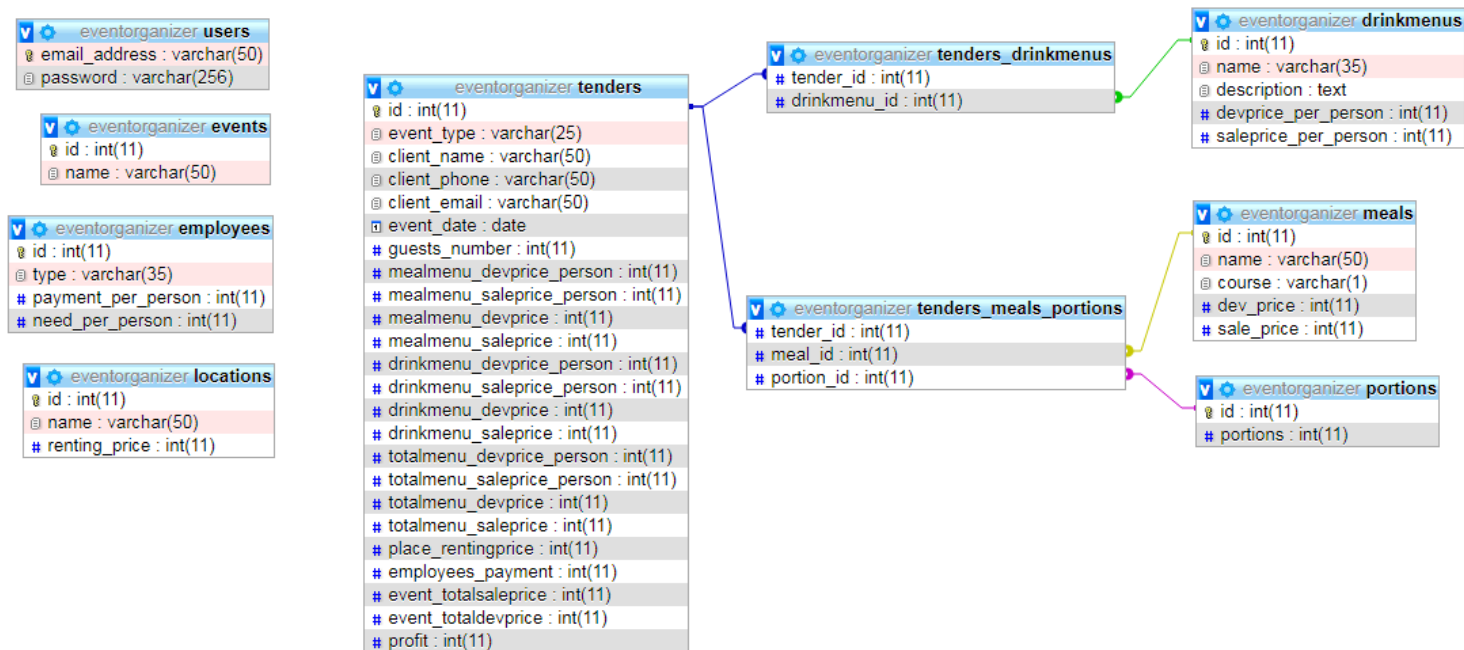
Az árajánlaton szereplő adatok a következők: megrendelő neve, telefonszáma, emailcíme (opcionális), rendezvény típusa, időpontja, a megrendelt ételek listája és ételenként számolt adagok, az italcsomag típusa, ételek ára 1 főre, ételek ára a teljes létszámmra, italok ára egy főre, italok ára a teljes létszámmra, a teljes menü(ételek és italok) ára 1 főre és teljes létszámmra, a munkaerő bére, a rendezvényház bérleti díja és a rendezvény teljes megajánlott ára.

Ezek alapján a tervezést három fő részegységre osztottam, az adattárolás (Adatbázis), az adatkezelés (osztályok, osztályhierarchia) és a felhasználói felület megtervezésére.

3.2. Adatbázis:

Az adattároláshoz a mára már elterjedt relációs adatbázist használtam, pontosabban MySQL adatbázist, ezen belül is InnoDB tárolómotort mivel ez támogatja teljes körűen a relációs szolgáltatásokat. Az adatbázis táblák oszlopainak meghatározásánál ügyeltem arra hogy az általuk felvehető értékek hosszúságát lekorlátozom, ezzel is optimálisabbá téve az adatbázist.

Az alábbi modell került elkészítésre (lásd 21. ábra).



(21. ábra)

3.2.1. Táblák:

Minden tábla alapvetően utf8_hungarian_ci karakterkódolású

users tábla:

email: Emailcím
password: Jelszó

Oszlopok tulajdonságai:

email: varchar(50), NOT NULL, PRIMARY KEY
password: varchar(256), NOT NULL.

A **users** táblában tároljuk a bejelentkezési adatait a felhasználónak. az emailcím elsődleges kulcsként való beállítása biztosítja, hogy ne lehessen kettő ugyan olyan emailcímet megadni. A jelszavakat MD5-ös titkosítással ellátva tároljuk ezzel is védve a felhasználó adatait. Egyik oszlop sem vehet fel NULL értéket.

events tábla:

id: A rendezvény azonosítója.
name: A rendezvény neve.

Oszlopok tulajdonságai:

id: int(11), NOT NULL, AUTO_INCREMENT, PRIMARY KEY.
name: varchar(50), NOT NULL.

Az **events** tábla tárolja a rendezvények típusait, amiket majd a nézetnek fog szolgáltatni.

tenders tábla:

| | |
|--------------------------------|--|
| id: | Árajánlat azonosítója |
| event_type: | Rendezvény típusa |
| client_name: | Megrendelő neve |
| client_phone: | Megrendelő telefonszáma |
| client_email: | Megrendelő emailcíme |
| event_date: | Rendezvény időpontja |
| guests_number: | Vendégek száma |
| mealmenu_devprice_perperson: | Ételmenü beszerzési ára 1 főre számolva. |
| mealmenu_saleprice_perperson: | Ételmenü eladási ára 1 főre számolva. |
| mealmenu_devprice: | Ételmenü beszerzési ára teljes létszámra számolva. |
| mealmenu_saleprice: | Ételmenü eladási ára teljes létszámra számolva. |
| drinkmenu_devprice_perperson: | Italcsomag beszerzési ára 1 főre számolva. |
| drinkmenu_saleprice_perperson: | Italcsomag eladási ára 1 főre számolva. |
| drinkmenu_devprice: | Italcsomag beszerzési ára teljes létszámra számolva. |
| drinkmenu_saleprice: | Italcsomag eladási ára teljes létszámra számolva. |
| totalmenu_devprice_perperson: | Ételek és italmenü összesített beszerzési ára 1 főre. |
| totalmenu_saleprice_perperson: | Étel és italmenü összesített eladási ára teljes létszámra. |
| place_rentingprice: | Rendezvényház bérleti díja. |
| employees_payment: | Szükséges munkaerő teljes fizetése. |
| event_totalsaleprice: | Rendezvény teljes eladási ára. |
| event_totaldevprice: | Rendezvény teljes beszerzési ára. |
| profit: | Haszon. |

Oszlopok tulajdonságai:

| | |
|--------------------------------|---|
| id: | int(11), NOT NULL, AUTO_INCREMENT, PRIMARY KEY. |
| event_type: | varchar(25),NOT NULL |
| client_name: | varchar(50),NOT NULL |
| client_phone: | varchar(50),NOT NULL |
| client_email: | varchar(50),NOT NULL |
| event_date: | date, NOT NULL |
| guests_number: | int(11) NOT NULL |
| mealmenu_devprice_perperson: | int(11) NOT NULL |
| mealmenu_saleprice_perperson: | int(11) NOT NULL |
| mealmenu_devprice: | int(11) NOT NULL |
| mealmenu_saleprice: | int(11) NOT NULL |
| drinkmenu_devprice_perperson: | int(11) NOT NULL |
| drinkmenu_saleprice_perperson: | int(11) NOT NULL |
| drinkmenu_devprice: | int(11) NOT NULL |

| | |
|--------------------------------|------------------|
| drinkmenu_saleprice: | int(11) NOT NULL |
| totalmenu_devprice_perperson: | int(11) NOT NULL |
| totalmenu_saleprice_perperson: | int(11) NOT NULL |
| place_rentingprice: | int(11) NOT NULL |
| employees_payment: | int(11) NOT NULL |
| event_totalsaleprice: | int(11) NOT NULL |
| event_totaldevprice: | int(11) NOT NULL |
| profit: | int(11) NOT NULL |

A **tenders** táblába fogjuk végezni az árajánlat mentését, itt fogjuk tárolni az árajánlat készítése közben számított értékeket és az alapvető az árajánlathoz elengedhetetlen adatokat. Az elsődleges kulcs, mint automatikusan növekvő értékű szám, azért tökéletes választás, mert a későbbiek során látni fogjuk hogy N:1 kapcsolatot hozunk létre a **meals** és a **portions** táblával és 1:1-hez kapcsolatot fogunk létrehozni a **drinkmenüs** táblával.

meals tábla:

| | |
|-------------|---|
| id: | Az étel azonosítója (auto increment) |
| name: | Az étel neve |
| course: | A fogás típusa (lehet f -főétel, e-előétel, d-desszert, s - savanyúság,k-köret) |
| dev_price: | Az étel beszerzési ára 1 főre értelmezve. |
| sale_price: | Az étel eladási ára 1 főre értelmezve. |

Oszlopok tulajdonságai:

| | |
|------------|---|
| id: | int(11), NOT NULL, AUTO_INCREMENT, PRIMARY KEY. |
| name: | varchar(50), NOT NULL. |
| course: | varchar(1), NOT NULL. |
| dev_price: | int(11), NOT NULL |
| sale_price | int(11),NOT NULL |

A **meals** táblának két fontos funkciója is lesz, ide töltjük fel az ételek listáját, A nézet innen fogja lekérni az ételeket, amikből a felhasználó ki tudja választani melyiket akarja hozzáadni az árajánlatához. A másik fontos feladata hogy N:M-hez kapcsolatot fogunk létrehozni a **portions** és a **tenders** táblákkal, ezzel biztosítva, hogy egy árajánlathoz lekérdezhető legyen milyen és hány adag étel tartozik.

portions tábla:

| | |
|-----------|------------------|
| id: | Adag azonosítója |
| portions: | Adagok száma |

Oszlopok tulajdonságai:

| | |
|-----------|---|
| id: | int(11), NOT NULL, AUTO_INCREMENT, PRIMARY KEY. |
| portions: | int(11) NOT NULL |

A **portions** egy segéd tábla, ami kizárólag egy automatikusan növekvő egész számú azonosítót és egy hozzátartozó adag számot tartalmaz, ennek az elsődleges kulcsa 1:1-hez fog

aránylani a **meals** és a **tenders** elsődleges kulcsaival. Tehát 1 árajánlat, 1 fajta ételéhez 1 adagszám tartozhat.

tenders_meals_portions tábla:

tender_id: Adag azonosítója
meal_id: Étél azonosítója
portion_id: Adag azonosítója

Oszlopok tulajdonságai:

tender_id: int(11), NOT NULL, FOREIGN KEY
meal_id: int(11), NOT NULL, FOREIGN KEY
portions_id: int(11), NOT NULL, FOREIGN KEY

A **tenders_meals_portions** tábla tehát egy join tábla, ami arra hivatott szolgálni, hogy konkrétan az ételeket menübe tudjuk rendezni és hozzácsatolni egy árajánlathoz, azaz 1 árajánlathoz tartozhat több fajta étel, viszont 1 árajánlathoz 1 fajta ételből csak egy tartozhat 1 adagszámmal. 1 ételhez tartozhat több adagszám is viszont nem tartozhatnak ugyanazon árajánlathoz.

drinkmenus tábla:

id: Italcsomag azonosítója.
name: Italcsomag Megnevezése.
description: Italcsomag leírása.
devprice_per_person: Beszerzési ár 1 főre értelve.
saleprice_per_person: Eladási ár 1 főre értelve.

Oszlopok tulajdonságai:

id: int(11), NOT NULL, AUTO_INCREMENT, PRIMARY KEY.
name: varchar(35), NOT NULL
description: text, NOT NULL
devprice_per_person: int(11), NOT NULL
saleprice_per_person: int(11), NOT NULL

A **drinkmenus** táblának szintén két szerepe lesz a szoftver szempontjából, egyrészt a felhasználói felület az innen lekérdezett adatokból összeállított italcsomagokból ad majd lehetőséget választani a felhasználónak, másrészt a már elkészült árajánlat elsődleges kulcsával és a kiválasztott italcsomag elsődleges kulcsával 1:1-hez arányú kapcsolatot fogunk létrehozni a **tenders_drinkmenus** táblában.

tenders_drinkmenus tábla:

tender_id: Italcsomag azonosítója.
drinkmenu_id: Italcsomag Megnevezése.

Oszlopok tulajdonságai:

| | |
|---------------|--------------------------------|
| tender_id: | int(11), NOT NULL, FOREIGN KEY |
| drinkmenu_id: | int(11), NOT NULL, FOREIGN KEY |

A **tenders_drinkmenus** tehát egy join table, ami segít lekérdezhetővé tenni, hogy egy árajánlathoz milyen italcsomag lett kiválasztva. 1:1-hez arányú kapcsolatot létrehozva, azaz 1 árajánlathoz 1 italcsomag tartozhat.

employees tábla:

| | |
|---------------------|--|
| id: | Munkaerő azonosítója. |
| name: | Munkaerő típusa |
| payment_per_person: | Fizetés / alkalmazott |
| need_per_person: | Hány vendégként kell 1 alkalmazott az adott típusból |

Oszlopok tulajdonságai:

| | |
|---------------------|---|
| id: | int(11), NOT NULL, AUTO_INCREMENT, PRIMARY KEY. |
| name: | varchar(35), NOT NULL |
| payment_per_person: | int(11), NOT NULL |
| need_per_person: | int(11), NOT NULL |

Az **employees** táblában tároljuk azokat a fajta adatokat amik az alkalmazott szükséglet és az alkalmazottak összesített béréhez szükséges számításokhoz kellenek.

locations tábla:

| | |
|----------------|-----------------------|
| id: | Helyszín azonosítója |
| name: | Helyszín neve |
| renting_price: | Helyszín bérleti díja |

Oszlopok tulajdonságai:

| | |
|----------------|---|
| id: | int(11), NOT NULL, AUTO_INCREMENT, PRIMARY KEY. |
| name: | varchar(50), NOT NULL |
| renting_price: | int(11), NOT NULL |

A **locations** táblában tároljuk a rendezvényház adatait.

3.3. Adatkezelés, a program kialakítása.

Az EventOrganizerApplication megírásához C# nyelvet és .NET 4.5.2 verziójú keretrendszert használtam, Visual Studio 2015-ös fejlesztői környezetben, továbbá helyi xampp szerverkörnyezetet. Az UML osztálydiagrammokat a <https://www.draw.io/> szerkesztő program segítségével készítettem. A program készítését két fő szekcióra bontottam, az osztály komponensek összetételének, hierarchiájának és egymással való kapcsolatainak megtervezésére és a felhasználó felület, a navigáció, a felhasználói funkciók sorrendjének kialakítására.

3.3.1 Osztályok

Mivel a tervezést az adatbázis kialakításánál kezdtem, a programom szükséges adatainak összegyűjtésével, így a csoportosítással és a csoportok specifikus osztályokba való meghatározásával folytattam az objektum orientált szemlélet módnak megfelelően. A fő feladatunk, hogy releváns árajánlatot készítsünk, és a tervezést is itt kezdtem a **Tender (árajánlat)** osztály és az adatbázis műveletekért felelős **DataBase(adatbázis)** osztály definiálásával. Fontosnak tartottam az adatrejtés elvét követni így a C# adta lehetőséget a „Property” használata helyett, privát adattagokat használtam és publikus Getterek és Setterek segítségével érjük el az adatokat.

Megjegyzés: Az egyes osztályok függvényeinek és metódusainak részletezésénél a funkciókat csak nevükkel és paraméterikkel fogom csak jelölni (funkcióNeve(paraméter)), mivel minden osztályhoz csatolok UML ábrát amin jól látszik a hozzáférhetőség és a funkció típusa.

Külső modulok használata:

- MySql.Data.6.10.9
- itext7.7.1.14 – PDF készítéshez szükséges osztály
- Common.Logging.3.4.1 és Common.Logging.3.4.1 Core –Az itext-hez szükséges kiegészítő csomagok
- Portable.BouncyCastle.1.8.10–Az itext szükséges kiegészítőcsomagok

Adattagok elnevezéseinek segédlete:

Több helyen is megjelennek az alábbi kifejezések, ezért kiemelem egy segédletbe:

- TotalDevPrice Beszerzési ár a teljes létszámra.
- DevPricePerPerson Beszerzési ár egy főre.
- TotalSalePrice Eladási ár a teljes létszámra.
- SalePricePerPerson Eladási ár egy főre.

DataBase (Adatbázis) osztály:

Ez az osztály valósítja meg az adatbázis kapcsolatot, felelős az adatbázishoz való biztonságos hozzáférés biztosításáért és a lekérdezések végrehajtásáért. Logikailag én úgy terveztem a programot, hogy vannak olyan osztályok, amelyek objektumai kizárólag adatbázisban tárolt adatokból jöhetnek létre így azokban az osztályokban el lett helyezve segéd adattagként egy **DataBase** objektum. Az adattagokat abban az esetben fogom magyarázni, ha nem magától érthető az adattag neve és szerepe.

Adattagok és funkciók (lásd 22. ábra):

| DataBase | |
|--|--|
| - conn: MySqlConnection | |
| - da: MySqlDataAdapter | |
| - ds: DataSet | |
| - conString: string | |
| - query: string | |
| <hr/> | |
| + DataBase(): void | |
| + commandExecute(query:string): void | |
| + getRowsLength(): int | |
| + getDataFromDataSet(tableName:string,rowIndex:int,colName:string): object | |
| + getQuery(): string | |
| + setQuery(query:string): void | |
| + getDataSet(): DataSet | |

(22. ábra)

Főbb Funkciók:

- Database(): Bekéri a Properties.Settingsben létrehozott connection stringet , majd beállítja ezt a **conn** objektumnak.
- commandExecute(string query): Végrehajta a lekérdezést majd feltölti a ds(DataSet) objektumot az eredménnyel.
- getRowsLength(): Visszaadja az adattábla sorainak számát.
- getDataFromDataSet(string tableName, int rowindex,string colName): Visszatér a tábla megadott sorának és oszlopának értékével

Tender (árajánlat) osztály:

Ez az osztály fogja össze az összes olyan adategységet, ami jellemzője egy árajánlatnak, konkrétan a megrendelő adatait, a rendezvény típusát, a vendégek számát, az összeválogatott ételmenüt, a kiválasztott italcsomagot, az alkalmazottak listáját, a helyszín információkat, a rendezvény dátumát, eladási és beszerzési árakat és persze a profitot.

Adattagjai és funkciói (lásd 23. ábra):

| Tenders | | Getterek és setterek | |
|---|--|---|--|
| - id: int | | + getId() : int | |
| - clientName: string | | + setId(id: int) : void | |
| - clientPhone: string | | + getEventType() : string | |
| - clientEmail: string | | + setEventType(eventType:string) : void | |
| - eventType: string | | + getClientName() : string | |
| - numberOfGuest: int | | + setClientName(clientName:string) : void | |
| - eventDate: DateTime | | + getClientEmail() : string | |
| * - mealMenu: MealMenu | | + setClientEmail(clientEmail:string) : void | |
| - drinkMenu: DrinkMenu | | + getClientPhone() : string | |
| - menu: Menu | | + setClientPhone(clientPhone:string) : void | |
| - employeeList: EmployeeList | | + getEventDate() : DateTime | |
| - place: Place | | + setEventDate(eventDate:DateTime) : void | |
| - totalSalePrice: int | | + getNumberOfGuests() : int | |
| - totalDevPrice: int | | + setNumberOfGuests(numberOfGuests:int) : void | |
| - profit: int | | + getMealMenu() : MealMenu | |
| - db: DataBase | | + getDrinkMenu() : DrinkMenu | |
| | | + getMenu() : Menu | |
| + Tender(): void | | + getEmployeeList() : EmployeeList | |
| + Tender(id: int): void | | + setEmployeeList(employeeList:EmployeeList) : void | |
| + save(): void | | + getPlace() : Place | |
| + delete(): void | | + getTotalSalePrice() : int | |
| + calculateTenderValues(int numberOfGuests): void | | + setTotalSalePrice(totalSalePrice:int) : void | |
| + calculateProfit() : void | | + getTotalDevPrice() : int | |
| | | + setTotalDevPrice(totalDevPrice:int) : void | |
| | | + getProfit() : int | |

(23. ábra)

Főbb Funkciók:

- `Tender()`: Alapértelmezett konstruktor, beállít minden adattagnak egy alapértelmezett értéket.
- `Tender(int id)`: **id** alapján adatbázisból lekérdezi az árajánlat adatait így hozva létre a `Tender` objektumot.
- `calculateProfit()`: Kiszámolja a profitot és beállítja a teljes eladási és beszerzési árakat.
- `calculateTenderValues(int numberOfGuests)`: Az árajánlat összes számított értékét kikalkuláltatja.
- `save()`: Az árajánlat adatábzisba való beírását(mentését) végzi.
- `delete()`: Az árajánlat adatbázisból való törlését végzi.
- `setNumberOfGuests(int numberOfGuests)`: beállítja a vendégek számát minden olyan objektumnak ami rendelkezik ilyen adattaggal.

Menu (Menü) osztály:

A `menü` osztály önmagában a rendezvény létszáma alapján teszi lehetővé az Eladási/Beszerzési árak számításait és adatkezelését. Ez egy `ősosztály`, leszármazottai a **`MealMenu`** és a **`DrinkMenu`** osztályok.

Adattagjai és funkciói (lásd 24. ábra):

| Menu | |
|---|--|
| - numberOfGuests: int | |
| - devPricePerPerson: int | |
| - salePricePerPerson: int | |
| - totalDevPrice: int | |
| - totalSalePrice: int | |
| <hr/> | |
| + Menu(numberOfGuests): void | |
| + calculateTotalDevPrice(a:int,b:int): int | |
| + calculateDevPricePerPerson(totalDevPrice:int): int | |
| + calculateTotalSalePrice(a:int,b:int): int | |
| + calculateSalePricePerPerson(a:int,b:int): int | |
| + getDevPricePerPerson(): int | |
| + setDevPricePerPerson(devPricePerPerson:int): void | |
| + getSalePricePerPerson(): int | |
| + setSalePricePerPerson(salePricePerPerson:int): void | |
| + getTotalDevPrice(): int | |
| + setTotalDevPrice(totalDevPrice:int): void | |
| + getTotalSalePrice(): int | |
| + setTotalSalePrice(totalSalePrice): void | |
| + getNumberOfGuests(): int | |
| + setNumberOfGuests(numberOfGuests:int): void | |

(24. ábra)

Főbb Funkciók:

- **Menu(int numberOfGuests):** A konstruktor, beállítja a vendégek számát és 0-ra inicializálja a többi adattagot.
- **calculateTotalDevPrice(int a , int b):** Kiszámolja és beállítja a teljes beszerzési ár adattagot, majd ennek értékével tér vissza.
- **calculateDevPricePerPerson(int totalDevPrice):** Kiszámolja és beállítja a beszerzési ár egy főre jutó részét a teljes beszerzési ár és a vendégek száma alapján.
- **calculateTotalSalePrice(int a , int b):** Kiszámolja és beállítja a teljes eladási ár adattagot, majd ennek értékével tér vissza.
- **calculateSalePricePerPerson(int a, int b):** Kiszámolja és beállítja az eladási ár egy főre jutó részét a teljes beszerzési ár és a vendégek száma alapján, majd ennek értékével tér vissza.

DrinkMenu (Italcsomag) osztály:

Az italcsomag osztály leszármazottja a **Menu** osztálynak. A **drinkmenus** adatbázis táblából lekérdezett adatok fogadására lett felkészítve.

Adattagjai és funkciói (lásd 25. ábra):

| DrinkMenu | | | | |
|----------------|----------------------------|--------|--|--|
| - id: | int | | | |
| - name: | string | | | |
| - description: | string | | | |
| - db: | DataBase | | | |
| | | | | |
| + | DrinkMenu(name:string): | void | | |
| + | setByName(name:string): | void | | |
| + | calculateTotalSalePrice(): | int | | |
| + | calculateTotalDevPrice(): | int | | |
| + | getID(): | int | | |
| + | getName(): | string | | |
| + | getDescription(): | string | | |

(25. ábra)

Főbb Funkciók:

- DrinkMenu(int numberOfGuests): A konstruktorhívási láncnak megfelelően bekéri a **Menu** számára fontos vendégek létszámát.
- setByName(string name): Név alapján összeállítja az adatbázis lekérdezést és beállítja a létrejövő objektum adattagjainak értékeit a lekérdezés eredményéből.
- calculateTotalDevPrice(int a , int b): Kiszámolja és beállítja a teljes beszerzési ár adattagot a vendégek száma alapján, majd ennek értékével tér vissza.
- calculateTotalSalePrice(int a , int b): Kiszámolja és beállítja a teljes eladási ár adattagot a vendégek száma alapján, majd ennek értékével tér vissza.

MealMenu (Ételmenü) osztály:

Adattagjai és funkciói (lásd 26. ábra):

| MealMenu | | | | |
|----------|--------------------------------|------------|--|--|
| - meals: | List<Meal> | | | |
| | | | | |
| + | MealMenu(numberOfGuests:int) | | | |
| + | addMeal(meal:Meal): | void | | |
| + | removeMeal(i:int): | void | | |
| + | getMenuLength(): | int | | |
| + | calculateTotalDevPrice(): | int | | |
| + | calculateDevPricePerPerson(): | int | | |
| + | calculateTotalSalePrice(): | int | | |
| + | calculateSalePricePerPerson(): | int | | |
| + | getMenu(): | List<Meal> | | |

(26. ábra)

Főbb Funkciók:

- `MealMenu(int numberOfGuests)`: A konstruktorhívási láncnak megfelelően bekéri a **Menu** számára fontos vendégek létszámát és létrehozza a `List` objektumot.
- `addMeal(Meal meal)`: Egy étel hozzáadása a listához.
- `removeMeal(int i)`: Egy étel törlése a megadott indexen a listából.
- `getMenuLength()`: Lista hosszának/méretének lekérdezését végzi, majd annak értékével tér vissza.
- `calculateTotalDevPrice()`: Kiszámolja és beállítja az étel lista teljes beszerzési árát, majd annak értékével tér vissza.
- `calculateDevPricePerPerson()`: Kiszámolja és beállítja az étel menü egy főre jutó beszerzési árát, majd annak értékével tér vissza.
- `calculateTotalSalePrice()`: Kiszámolja és beállítja az étel lista teljes eladási árát, majd annak értékével tér vissza.
- `calculateSalePricePerPerson()`: Kiszámolja és beállítja az étel menü egy főre jutó eladási árát, majd annak értékével tér vissza.

Meal (Étel) osztály:

Az étel osztály, egy étel adatkezelést végzi. A **meals** és a **tenders_meals_portions** adattáblából érkező adatok fogadására lett felkészítve.

Megjegyzés: A „course” adattag a fogás megjelölésére lett létrehozva értékei lehetnek pl.: „f” mint főétel, „e” mint előétel, a felhasználói felület fejlesztésénél került felhasználásra.

Adattagjai és funkciói (lásd 27. ábra):

| Meal | |
|---|--|
| - id : int | |
| - name : string | |
| - course : char | |
| - dev_price : int | |
| - sale_price : int | |
| - portion : int | |
| - db : DataBase | |
| | |
| + Meal(name:string): void | |
| + Meal(name: string, portion:int): void | |
| + getID(): int | |
| + getName(): string | |
| + getDevPrice(): int | |
| + getSalePrice(): int | |
| + getPortion(): int | |
| + setPortion(portion:int): int | |

(27. ábra)

Főbb Funkciók:

- Meal (string name): A konstruktor a név alapján összeállítja az adatbázis lekérdezést és beállítja a létrejövő objektum adattagjainak értékeit a lekérdezés eredményéből.
- Meal(string name, int portion): A konstruktor hívási lánc elvén alapulva létrejöhet az adag(portion) megadásával is az objektum.

Employee (Alkalmazott) osztály:

Egy adott típusú alkalmazott adatkezelését végző osztály, az **employees** táblából érkező adatok fogadására lett felkészítve, viszont egy másik fontos funkciót is ellát, Kiszámolja a létszám alapján, hogy egy adott típusú alkalmazottból hány emberre lesz szükség és azoknak mennyi lesz az összesített fizetésük.

Megjegyzés: a „paymentPerPerson” adattag az 1 alkalmazott fizetését tárolja, a „numberOfEmployees” azt tárolja, hogy hány alkalmazott van az adott típusból, a „needPerPerson” pedig megmondja, hogy hány vendégnekét kell 1 alkalmazott. a „totalPayment” pedig az alkalmazottak összesített fizetését jelöli.

Adattagjai és funkciói (lásd 28. ábra):

| Employee | |
|---|--|
| - id: int | |
| - type: string | |
| - paymentPerPerson: int | |
| - needPerPerson: int | |
| - numberOfEmployees: int | |
| - totalPayment: int | |
| - db: DataBase | |
| <hr/> | |
| + Employee(type:string, numberOfGuests:int): void | |
| - calculateNumberOfEmployees(numberOfGuests:int): int | |
| - calculateTotalPayment(numberOfGuests:int): int | |
| + getID(): int | |
| + getEmployeeType(): string | |
| + getNumberOfEmployees(): int | |
| + getTotalPayment(): int | |

(28. ábra)

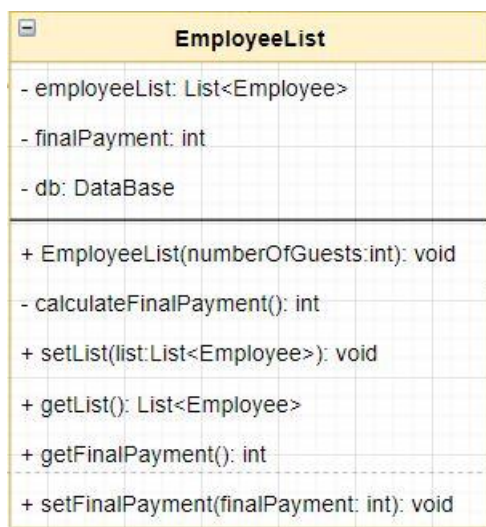
Főbb Funkciók:

- Employee (string type, int numberOfGuests): A konstruktor a típus alapján összeállítja az adatbázis lekérdezést és beállítja a létrejövő objektum adattagjainak értékeit a lekérdezés eredménye és a vendégek száma alapján.
- calculateNumberOfEmployees(int numberOfGuests): Kiszámolja hány alkalmazott kell az adott típusból a vendégek létszáma alapján.
- calculateTotalPayment(int numberOfGuests): Kiszámolja az alkalmazottak teljes fizetését.

EmployeeList (Alkalmazottak listája) osztály:

A különböző típusú alkalmazottak listába rendezését és költségeinek összesítését végző osztály.

Adattagjai és funkciói (lásd 29. ábra):



(29. ábra)

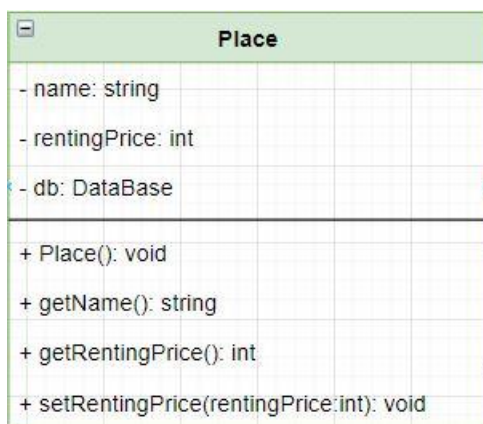
Főbb Funkciók:

- EmployeeList (int numberOfGuests): A konstruktor a rendezvény létszáma alapján feltölti a listát a szükséges alkalmazott objektumokkal.
- calculateFinalPayment(): Kiszámolja az alkalmazottak teljes költségét, és az eredménnyel visszatér.

Place (Helyszín) osztály:

A helyszín osztály, a **locations** táblából érkező adatok fogadására lett felkészítve. Jelenleg mivel ez a program egy konkrét embernek lett készítve, aki egy rendezvényhelyszínnel rendelkezik úgy lett kialakítva, hogy konkrétan annak a helyszínnak az adatait rendelje be, ez könnyen tovább fejleszthető.

Adattagjai és funkciói (lásd 30. ábra):



(30. ábra)

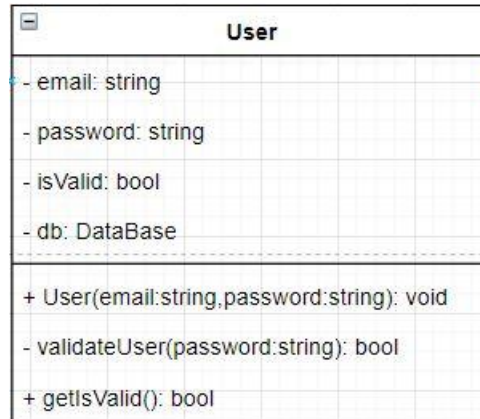
Főbb Funkciók:

- Place(int numberOfGuests):A konstruktor adatbázis lekérdezéssel beállítja az adattagok értékét.

User (Felhasználó) osztály:

A felhasználó osztály tárolja a **users** táblából érkező bejelentkezési adatokat. és legfontosabb funkciója hogy hitelesíti a felhasználói felületről érkező jelszót.

Adattagjai és funkciói (lásd 31. ábra):



(31. ábra)

Főbb Funkciók:

- User(int email): Kizárólag már létező emailcímmel rendelkező **User** objektum jöhet létre ezt a vizsgálatot a Form végzi el.
- validateUser(string password): A Formtól érkező jelszó MD5ös titkosítását és az adatbázisból érkezett titkosított jelszó összehasonlítását végző függvény.

Összegezve a **Tender** osztály tartalmaz minden az árajánlat megírásához szükséges osztályt (kivéve **User** osztályt, ami csak a bejelentkezéshez szükséges). Az osztályok a logikailag hozzájuk tartozó saját adatcsoportjaikon végzik el a szükséges számításokat, a **Tender** osztály a saját számítást igénylő értékein kívül az adatbevitel elvének megfelelően csak az alosztályokon

Oszálydiagramm

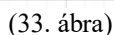
```

classDiagram
    class DrinkMenu {
        -id: int
        -name: string
        -description: string
        +db: Database
        +DrinkMenu(name: string): void
        +setName(name: string): void
        +calculateTotalDevPrice(): int
        +getDevPrice(): int
        +getName(): string
        +getDescription(): string
    }
    class Menu {
        +numberOfGuests: int
        +devPricePerPerson: int
        +setDevPricePerPerson(int): void
        +totalDevPrice: int
        +totalSalePrice: int
        +Menu(numberOfGuests): void
        +calculateTotalDevPrice(int, int): int
        +calculateDevPricePerPerson(totalDevPrice: int): int
        +calculateTotalSalePrice(int, int): int
        +calculateTotalDevPricePerPerson(int, int): int
        +getDevPricePerPerson(): int
        +setDevPricePerPerson(devPricePerPerson: int): void
        +setSalePricePerPerson(salePricePerPerson: int): void
        +getTotalDevPrice(): int
        +setTotalDevPrice(totalDevPrice: int): void
        +getTotalSalePrice(): int
        +setTotalSalePrice(totalSalePrice: int): void
        +getNumberOfGuests(): int
        +setNumberOfGuests(numberOfGuests: int): void
    }
    class MealMenu {
        -meal: List<Meal>
        +MealMenu(numberOfGuests: int): void
        +addMeal(meal: Meal): void
        +removeMeal(): void
        +getMealLength(): int
        +calculateTotalDevPrice(): int
        +calculateDevPricePerPerson(): int
        +calculateTotalSalePrice(): int
        +getMeal(): List<Meal>
    }
    class Meal {
        -id: int
        -name: string
        -course: char
        -dev_price: int
        -sale_price: int
        -portion: int
        +db: Database
        +Meal(name: string): void
        +Meal(name: string, portion: int): void
        +getId(): int
        +getName(): string
        +getDevPrice(): int
        +getSalePrice(): int
        +getPortion(): int
        +setPortion(portion: int): void
    }
    class Place {
        -name: string
        -rentingPrice: int
        +db: Database
        +Place(): void
        +getName(): string
        +getRentingPrice(): int
        +setRentingPrice(rentingPrice: int): void
    }
    class Database {
        +conn: MySqlConnection
        +db: MySQLDataAdapter
        +db: DataSet
        +connString: string
        +query: string
        +Database(): void
        +commandExecute(query: string): void
        +getRowLength(): int
        +getConnectionDataSet(tableName: string, columnIndex: int, columnName: string): object
        +getQuery(): string
        +setQuery(query: string): void
        +getConnection(): DataSet
    }
    class User {
        -email: string
        -password: string
        -isValid: bool
        +db: Database
        +User(email: string, password: string): void
        +validateUser(password: string): bool
        +getIsValid(): bool
    }
    class Tenders {
        -id: int
        -clientId: string
        -clientPhone: string
        -clientEmail: string
        -eventTypo: string
        -numberOfGuests: int
        -eventDate: DateTime
        -mealMenu: MealMenu
        -drinkMenu: DrinkMenu
        -menu: Menu
        -employeeList: EmployeeList
        -place: Place
    }
    DrinkMenu "1" --> "0..*" MealMenu
    MealMenu --|> Menu
    Menu "1" --> "0..*" Meal
    Place "1" --> "0..*" Database
    Database "1" --> "0..*" User
    Database "1" --> "0..*" Tenders
    EmployeeList "1" --> "3" Employee
    
```

3.3.2 Felhasználó felület, Formok és működés:

frm Login(Bejelentkezés):

Adattagjai és funkciói (lásd: 33. ábra)



Főbb funkciói:

- btn_Login_Click(object sender, EventArgs e): A bejelentkezési adatok ellenőrzését és a főmenübe léptetést végző „klikk” esemény.

frm MainMenu (Főmenü):

A főmenü form egy egyszerű navigációt létesít a program különböző szegmensei között, konkrétan az új árajánlat készítése form és az árajánlatok megtekintése form között, ezen kívül a kijelentkezést is a főmenüben lehet megtenni, a kilépésre a program minden formon ad lehetőséget, szintűgy a főmenübe való visszatéréshez is.

| frm_MainMenu | |
|---|--|
| - newEventTender: frm_NewEventTender | |
| - frmTenders: frm_Tenders | |
| - frmLogin: frm_Login | |
| + frm_MainMenu(): void | |
| - btn_NewEvent_Click(sender:object,e:EventArgs): void | |
| - btn_Tenders_Click(sender:object,e:EventArgs): void | |
| - btn_LogOut_Click(sender:object,e:EventArgs): void | |
| - btn_Exit_Click(sender:object,e:EventArgs): void | |

Adattagjai és funkciói (lásd: 34. ábra)

(34. ábra)

Főbb funkciói:

- btn_NewEvent_Click(object sender, EventArgs e): Az új árajánlat form megnyitására szolgáló eseményvezérlő.
- btn_Tender_Click(object sender, EventArgs e): Az árajánlatok megtekintése form megnyitására szolgáló eseményvezérlő.
- btn_LogOut_Click(object sender, EventArgs e): Kijelentkezésre szolgáló eseményvezérlő.
- btn_Exit_Click(object sender, EventArgs e): A program bezárásra szolgáló eseményvezérlő.

frm NewEventTender(Új árajánlat készítése):

Az új árajánlat megírását teszi lehetővé és végzi el a szükséges adatkezelést, itt hajtjuk végre az adatbázisba való feltöltést is a **Tender** objektum és egy gomb eseménykezelő segítségével. Nagyon fontos minden beviteli mezőnél, ellenőrzéseket végezni. Megfelelő szövegformátum vagy, hogy megfelelő számok kerüljenek az adott mezőkbe ezzel elkerülve az

Adattagjai és funkciói (lásd: 35. ábra):

```

frm_NewEventTender
- frmCheckTender: frm_CheckTender
- frmMainMenu: frm_MainMenu
- newTender: Tender
- db: DataBase
- query: string

+ frm_NewEventTender(): void
- frm_NewEventTender_Load(sender:object,e:EventArgs): void
- txt_NumberOfGuests_TextChanged(sender:object,e:EventArgs): void
- cbx_EventType_SelectedValueChanged(sender:object,e:EventArgs): void
- cbx_DrinkServiceType_SelectedValueChanged(sender:object,e:EventArgs): void
- btn_PreCalculation_Click(sender:object,e:EventArgs): void
- btn_AddStarter_Click(sender:object,e:EventArgs): void
- btn_RemoveStarter_Click(sender:object,e:EventArgs): void
- btn_AddMainCourse_Click(sender:object,e:EventArgs): void
- btn_RemoveMainCourse_Click(sender:object,e:EventArgs): void
- btn_AddSideDishes_Click(sender:object,e:EventArgs): void
- btn_RemoveSideDish_Click(sender:object,e:EventArgs): void
- btn_AddPickles_Click(sender:object,e:EventArgs): void
- btn_RemovePickles_Click(sender:object,e:EventArgs): void
- btn_AddDessert_Click(sender:object,e:EventArgs): void
- btn_RemoveDessert_Click(sender:object,e:EventArgs): void
- btn_AddPlatters_Click(sender:object,e:EventArgs): void
- btn_RemovePlatters_Click(sender:object,e:EventArgs): void
- btn_SavePreTender_Click(sender:object,e:EventArgs): void
- btn_MainMenu_Click(sender:object,e:EventArgs): void
- btn_Exit_Click(sender:object,e:EventArgs): void
- addMealOnClickEvent(txt:TextBox, cbx:Combobox): void
- removeMealOnClickEvent(txt:TextBox, cbx:Combobox): void
- generateMealDataGridView(menu:MealMenu, dtgv: DataGridView): void
- fillComboBox(txt:TextBox, cbx:Combobox): void

```

(35. ábra)

Főbb funkciói:

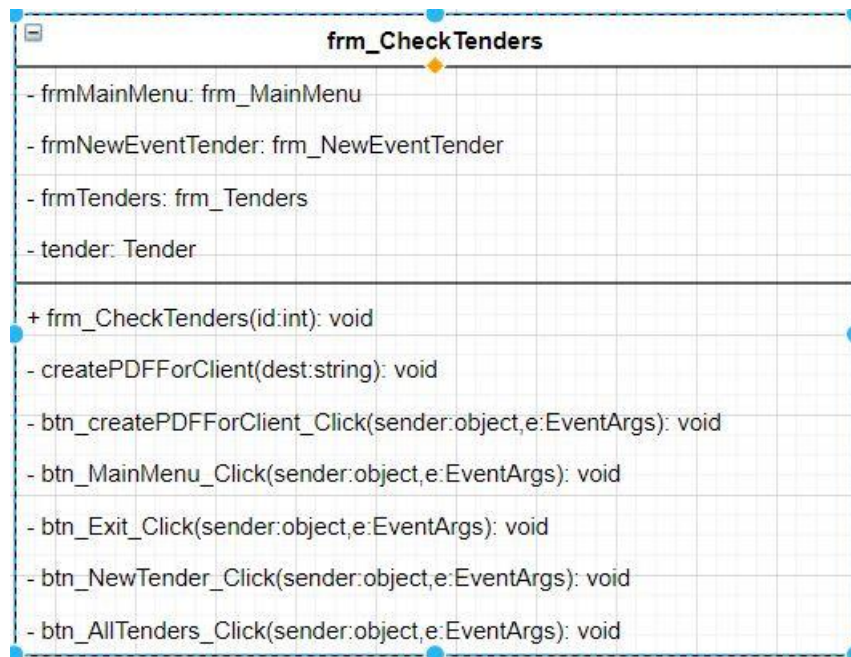
- **frm_NewEventTender():** létrehoz az alapértelmezett **DataBase** és **Tender** objektumokat.
- **frm_NewEventTender_Load():** A form betöltése alatt lekérdez és a feltölti A rendezvény típusa ComboBoxot, és a az Italcsomag ComboBoxot adatbázis lekérdezés alapján a fillCombox() funkció segítségével.
- **txt_NumberOfGuests_TextChanged():** Ez az esemény a vendégek számának helyes kitöltése után, engedélyezi a rendezvény további adatainak regisztrálását, azaz a menü összeállítását, az italcsomag kiválasztását, és kiszámítja a munkaerőszükségletet, amit meg is jelenít.
- **cbx_EventType_SelectedValueChanged():** A rendezvénytípusokat tartalmazó Combobox kiválasztott értékének változásánál feltölti az ételeket tartalmazó comboboxokat.
- **cbx_DrinkServiceType_SelectedValueChanged():** A választott italcsomag megjelenítését végiz.
- **btn_AddStarter_Click()** és **btn_RemoveStarter_Click** ugyan ezen _Click események az ételek hozzáadását és törlését végzik az ételmenü összeállítás során.

- `btn_PreCalculation_Click()`: kiszámoltatja a **Tender** objektumon segítségével az árajánlat jelenlegi állását, de csak az eladási árakat, a kiajánlott összeírt árat és a profitot. Itt még nem történik mentés.
- `btn_SavePreTender()`: Leellenőrzi, hogy a megrendelő adatai és rendezvény időpontja helyesen ki lettek-e töltve. Az árajánlat számításait elvégzi a **Tender** objektumon keresztül, majd elvégzi az adatbázisba való mentést. és átirányít Az „Árajánlat megtekintése” formra.
- `addMealOnClickEvent(TextBox txt, ComboBox cbx)`: Az ételek menühöz való hozzáadását végző metódus.
- `removeMealOnClickEvent(TextBox txt, ComboBox cbx)`: Az ételek menüből való törlését végző metódus.
- `generateMealDataGridView(Mealmenu menu, DataGridView dtgv)`: Legenerálja az összeállított **MealMenu** objektum alapján a kívánt DataGridView táblázatot.
- `fillComboBox(string query, ComboBox cbx, string displayMember, string valueMember)`: Feltölti a kívánt comboboxot, adatbázis lekérdezés alapján.

frm_CheckTender(Árajánlat megtekintése):

Ez a form arra szolgál, hogy megjelenítse az árajánlat összes adatát, mind a beszerzési árakat mind az eladási árakat és a profitot. Az alkalmazottak összeállítását, a bérüket, a választott italcsomagot, és hogy melyik ételből hány adagot kérnek a rendezvényhez. Ezen kívül itt történik az árajánlat PDF-be generálása és mentése a felhasználó által kiválasztott helyre. mindezt az **iText** API teszi lehetővé, hogy .NET környezetben PDF állományt hozzunk létre programkódból tetszőleges formában.

Adattagjai és funkciói (lásd: 36. ábra):



(36. ábra)

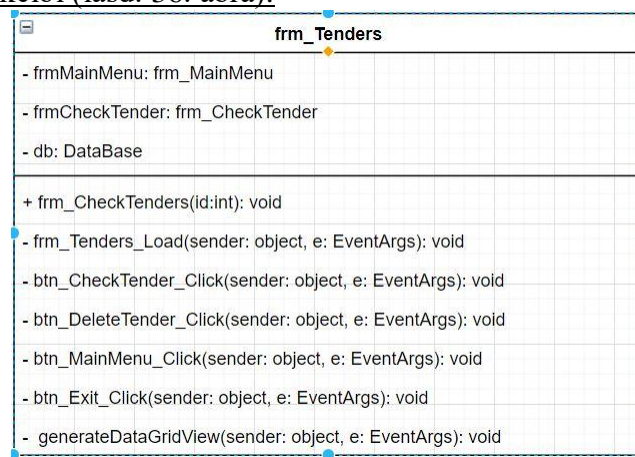
Főbb funkciói:

- frm_CheckTender(tenderID int): Amikor létrejön a form akkor a bekért **tenderID** alapján a **Tender** objektum segítségével lekérdezi az adott adatbázis azonosítón lévő Árajánlat adatait, amiket meg is jelenít.
- createPDFForClient(string dest): A konkrét PDF állományt szerkeszti meg és állítja össze a **Tender** objektum adatai alapján és menti el a bekért elérési útvonalra.
- btn_createPDFForClient_Click(): A PDF állomány elkészüléséhez szükséges elérési útvonal kiválasztását végzi SaveFileDialog nézetkezelő segítségével.
- btn_NewTender_Click(): Visszatér az „Új árajánlat készítése” form-ra.
- btn_AllTenders_Click(): Visszatér az „Árajánlatok megtekintése” form-ra.

frm Tenders(Árajánlatok megtekintése):

Az árajánlatok megtekintése form teszi lehetővé hogy az adatbázisban tárolt, árajánlatokat újra meg tudjuk nézni, vagy törölni tudjuk őket.

Adattagjai és funkciói (lásd: 36. ábra):



(36. ábra)

Főbb funkciói:

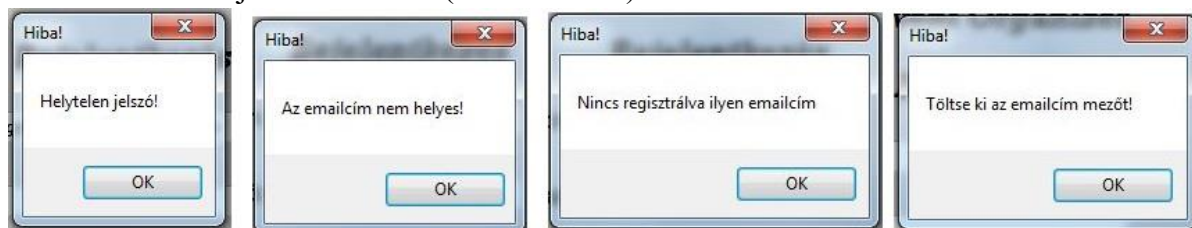
- frm_Tenders_Load(): A form betöltésekor lekérdezi az eddig összes mentésre került árajánlatot és megjeleníti azokat a generateDataGridView() metódussal.
- generateDataGridView(): Az árajánlatokat DataGridViewban megjelenítő metódus.
- btn_CheckTender_Click(): A táblázat kiválasztott sorában szereplő árajánlatot jeleníti meg a frm_CheckTender segítségével teljes egészében.
- btn_DeleteTender(): a táblázat kijelölt sorában lévő árajánlatot törli az adatbázisból.

3.4. Tesztelés:

Az alkalmazás fejlesztése alatt folyamatosan teszteltem az algoritmusokat, a számításokat végző függvények eredményeit, az objektumok egymással való kommunikációját tesztesetekkel így megbizonyosodtam róla hogy az alapvető függvények metódusok és logikák jól működni, Így a következőkben az alkalmazás funkcionális tesztjét fogom végrehajtatni Formonként.

Bejelentkezés:

Megadtam helyesen a bejelentkezés adatokat, emailcím: admin@admin.hu jelszó: Admin@0101, ezután átengedett a Főmenü formra. Amikor helytelen adatokat adtam meg az alábbi hibaüzenetek tájékoztattak róla (lásd: 37. ábra).



(37. ábra)

Főmenü:

A főmenüben rányomtam az „Új árajánlat készítése” gombra működött, ugyan így tettem az „Az árajánlatok megtekintése” gombbal a „Kijelentkezéssel”, és a „Kilépéssel” minden menügomb a megfelelő formra vezetett át a Kilépés bezárta a programot.

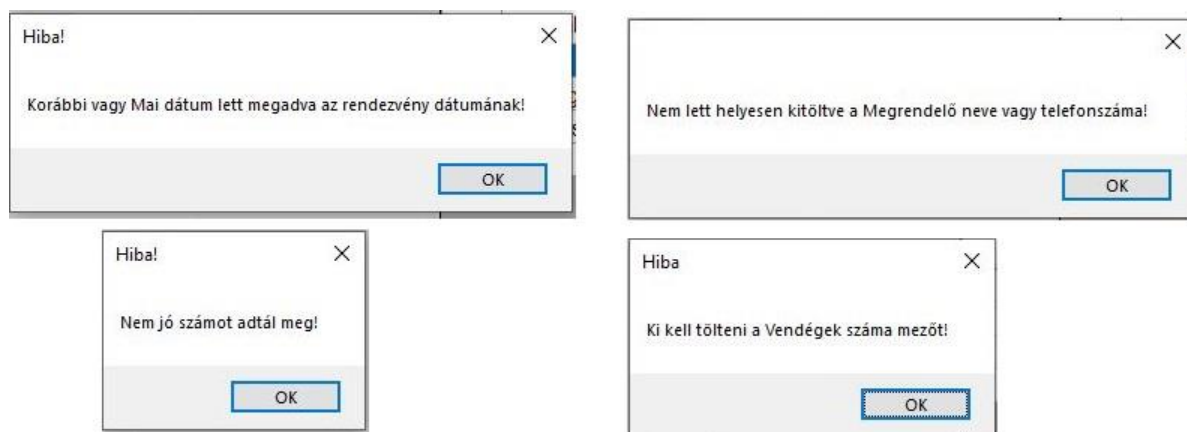
Új árajánlat készítése:

Teszteltem a Főmenü gombot, visszaléptem a főmenübe, teszteltem a Kilépés gombot, bezárta a programot.

A rendezvény típusánál kiválasztottam a Születésnap lehetőséget, a vendégek számához beírtam hogy 50, az időpontnál megadtam egy időpontot 2021. május 29-et. Megadtam a megrendelő adatainál a nevét, tel. számát, emailcímét, Kovács János, +36203144990, kovacsjani@outlook.hu

A vendégek számát beírva elérhetővé vált a Menü / ételcsomag szerkesztése az Italcsomag szerkesztése és a program helyesen kiszámolta az Alkalmazottak listáját. A menü szerkesztésénél kiválasztottam minden fogásból 50et majd kivontam mindenből 50et és aztán újra hozzá adtam, a gombok megnyomására a menü sikeresen változott. Kiválasztottam az Italcsomagok közül az „A korlátlan italfogyasztást” ezután a program tájékoztatót a kiválasztott italcsomag tartalmáról és áráról. Az előzetes számítás gombra kattintva megjelent az „Előzetes árajánlat számítása” blokk amiben helyesen kiszámolt adatok szerepeltek. Ezután rányomtam az Árajánlat készítése gombra, ami átvezetett az „Árajánlat megtekintése” formra.

Leteszteltem hibás adatok megadásával, a kötelező megrendelői adatok kihagyásával, rossz dátum megadással a következő hibaüzenetek jöttek elő sikeresen(lásd 38. ábra).



(38. ábra)

Árajánlat megtekintése:

A Főmenü gombot megnyomva visszatértem a főmenübe, a Kilépés gombot megnyomva kilép a programból

Az előbb elkészített árajánlatomnak minden értékét és számítását helyesen megjelenítette a PDF készítése gombra kattintva felugrott egy párbeszédablak, amiben kiválaszthattam hova akarom menteni az elkészülő PDF állományt, megadtam neki nevet és a kívánt helyre mentette, a sikeres mentésről tájékoztatást adott. Az Új árajánlat készítése gombra kattintva megnyitottam az Új árajánlat készítése formot, az Árajánlatok megtekintése gombra kattintva megnyitotta az Árajánlatok megtekintése formot.

Árajánlatok megtekintése:

A Főmenü gomb megnyomásával visszatértem a főmenübe, a Kilépés gomb hatására bezárta az alkalmazást. Kiválasztottam Az előbb megcsinált azt a sort, amiben Kovács János szerepel, majd Rákattintottam az Választott árajánlat megtekintése formra amibe megnyitottam a Kovács Jánosnak készült árajánlatot az Árajánlat megtekintése formon.

Ezután, kipróbáltam, hogy kiválasztom szintén ugyan ezt a sort és a Választott árajánlat törlésére nyomtam, sikeresen megnyitotta a Törlés validálására szolgáló üzenetet hogy „Biztosan törölni szeretném e..”, Nem gombra nyomva nem törölte, majd újra próbálva az Igen gombra nyomva kitörölte a sorból is és az Adatbázisból is sikeresen az árajánlatot, Ezek után tájékoztatott az alkalmazás, hogy Sikeresen törölve lett a kívánt sor.

4.Tovább Fejlesztés:

Itt meg kell jegyezni, hogy számos funkció nem került bele az alkalmazásba, amit én előzetesen megterveztem, de a leadási határidőig a programot ilyen szintre tudtam fejleszteni, és azt szerettem volna, ha egy olyan programot adok le aminek van eleje és vége.

Lehetőségek:

- Az ételek és italcsomag, a helyszín, a munkavállalók bérének szerkesztési lehetőségét kidolgozni.
- A legenerált PDF kiterjesztésű árajánlatot emailben való tovább küldését kidolgozni.

5.Összegzés:

Sok tapasztalatra és élményre tettem szert az alkalmazás fejlesztése során, megtanultam milyen előre eltervezni, és leképezni akár csak vázlatosan is egy egész alkalmazást pontról pontra, hogy mennyi időt vesz igénybe egy-egy modul kidolgozása, és hogy miért ne tervezzek több funkciót és modult mint amennyi az időmbe belefér. Végérvényben az alkalmazás jól működik és teljes mértékben ellátja feladatát. Örülök hogy ezt a témát választottam.

6.Irodalomjegyzék:

- Reiter István: C# programozás lépésről lépésre Jedlik Oktatási Stúdió Budapest, 2012
- https://www.w3schools.com/sql/sql_ref_mysql.asp
- <https://kb.itextpdf.com/home/it7kb/ebooks/itext-7-jump-start-tutorial-for-java/chapter-1-introducing-basic-building-blocks>
- <https://docs.microsoft.com/hu-hu/>