SZÁM-PONT Kft.

**Vadászlak** **Vendégház**

Tanár neve Készítő neve

Beck Sándor Czuppon Dávid

**SZOFTVERFEJLESZTŐ**  
2019

Tartalomjegyzék

[Bevezetés 2](#_Toc6593502)

[Témaválasztás indoklása 2](#_Toc6593503)

[Problémafelvetés 2](#_Toc6593504)

[Célkitűzés rögzítése, esetleges előfeltevés felvázolása. 3](#_Toc6593505)

[A választott programnyelv rövid bemutatása 3](#_Toc6593506)

[Célcsoport-ajánlás 4](#_Toc6593507)

[Felhasználói dokumentáció 5](#_Toc6593508)

[Általános leírás-a program célja 5](#_Toc6593509)

[A program indítása 5](#_Toc6593510)

[A program használata 6](#_Toc6593511)

[Jogszabályi háttér 18](#_Toc6593512)

[Fejlesztői dokumentáció 19](#_Toc6593513)

[Rendszerterv 19](#_Toc6593514)

[Biztonsági kérdések 19](#_Toc6593515)

[Navigáció 20](#_Toc6593516)

[Adatbázis 24](#_Toc6593517)

[Ergonómia 27](#_Toc6593518)

[Tesztelés 27](#_Toc6593519)

[Összegzés 29](#_Toc6593520)

[A cél megvalósulása 29](#_Toc6593521)

[Tovább fejlesztési lehetőségek 29](#_Toc6593522)

Bevezetés

Témaválasztás indoklása

Szakdolgozatom témájának egy olyan szoftver létrehozását választottam, amely fontos információk tárolására képes egy vendégházhoz tartozó vendégekről és a hozzájuk tartozó foglalásokról. Akik ilyen típusú vállalkozással foglalkoznak, bevételüket abból szerzik, hogy minél többször kiadják ingatlanjukat más embereknek. Természetesen a vendégeik után származó bevétel még nem a tiszta haszon. Számolniuk kell kiadásokkal is. Minél több információval rendelkeznek, annál nehezebb átlátni a pénzügyi folyamatokat.

Lényegében az eredményességünket követhetjük nyomon. Az alkalmazás olyan fontos adatok visszaadására képes, melyek figyelembe vételével hatékonyabban üzemeltethető a vállalkozás. Azért döntöttem ennek a szoftvernek a megalkotása mellett, mert a későbbiekben szeretném felhasználni saját célra. Van egy vendégházunk, aminek a működtetését szeretném sikeresebbé tenni a program használatával. Mivel nem találtam olyan alkalmazást, ami konkrétan megfelelne az igényeimnek, vettem a bátorságot hogy létrehozzam. Véleményem szerint egy üzlet akkor lehet sikeres, ha vele kapcsolatos pénzmozgás pontosan nyomon követhető. Az alábbiakban részletesen bemutatott szoftver részben ezt a célt hivatott megvalósítani.

A program képes:

* + - * Vendég adatok bevitelére, módosítására, törlésére
      * Vendégekhez tartozó foglalások felvételére, módosítására, törlésére
      * Költségek felvételére, módosítására, törlésére
      * Számos statisztikai adat megjelenítésére

Problémafelvetés

A program egy vendégház üzemeltető válláról veszi le azt a terhet, hogy mindent észben kelljen tartania, amire esélye sem lenne egyetlen embernek sem, főleg ha több évnyi nagy mennyiségű adattal rendelkezik. Sőt mi több olyan extra információkhoz juthatunk a programnak köszönhetően, amiket valószínűleg észre sem vennénk. Megfelelő információkkal a birtokunkban lehetőségünk van vállalkozásunk irányítását hatékonyabbá tenni. Egész biztosan lesznek olyan időszakok a vállalkozásunk során, amikor kevesebb bevételre teszünk szert, és nyilván olyanok is, amelyekben az átlagosnál jobban teljesítünk. Ezek figyelembe vételével javíthatunk a tevékenységünk eredményességén. A program használatával nem maradnak rejtetett költségeink, így a profitunk pontosan nyomon követhető. A programnak a vállalkozás működésének eredményességére kell rávilágítania a következő kérdésekre adott válaszaival.

* Mennyi vendéget sikerült szerezni egy időszak alatt?
* Mennyi lemondott foglalás volt?
* Mely hónapokban volt a legtöbb vendég?
* Hogyan találtak rá a vendégházra?
* Mekkora volt a profit?
* Mennyi időt töltenek el itt a vendégek?
* Vannak e visszatérő vendégek?
* Honnét jöttek a legtöbben?
* Ki a célközönség?

A kérdésekre kapott válaszokból láthatjuk, hogy miben voltunk kevésbé eredményesek, ezután nagyobb eséllyel tudjuk a megfelelő irányba terelni a vállalkozásunkat.

Célkitűzés rögzítése, esetleges előfeltevés felvázolása.

Felhasználó számára könnyen kezelhető, jól átlátható, egyszerű, letisztult, ugyanakkor mégis hatékony program megalkotása volt a cél. Arra törekedtem, hogy a felhasználó a lehető legegyszerűbben vihesse fel az adatait, majd a szükséges műveleteket végezhesse rajtuk. A programom megvalósításánál az alábbi szempontokat vettem figyelembe. Előszőr is a felhasználónak egy jelszóval védett felületet szeretnék létrehozni, egyedi jelszó beállítási lehetőséggel. Bejelentkezést követően legyen lehetősége vendég, foglalás és költség adatok felvitelére, módosítására, törlésére. Lehessen egy vendégekhez több foglalást hozzárendelni, és egy foglaláshoz is több vendéget. A feltöltött adataiból különböző statisztikai adatokat jeleníthet meg, mint például az éves bevétel, havi foglalások, vendégek száma stb. Azokon az oldalakon ahol felvitt adatok láthatók egy listában, ott egy kereső mezővel szűkítheti a kört a releváns adataira. Grafikonon is szemléltetem az adatokból kinyert eredményeket.

A választott programnyelv rövid bemutatása

A szoftver elkészítéséhez a Microsoft által fejlesztett .NET keretrendszer részét képző C# nyelvet választottam, melynek alapjául a Common Language Infrastructure röviden (CLI) szolgál. Ebben a modern objektum orientált nyelvben könnyedén és gyorsan fejleszthetünk asztali alkalmazásokat. Előnyei közé tartozik, hogy általános felhasználású programokat készíthetünk benne, melyeket szinte bármilyen platformon futtathatunk. Fejlesztő környezetnek szintén a Microsoft által fejlesztett Visual Studio-t választottam, mely többek közt a C# -t is támogatja.

Célcsoport-ajánlás

Ajánlom a szoftvert mindenkinek, aki vendégház üzemeltetéssel foglalkozik, és szeretné a vállalkozásának eredményességét nyomon követni, és hatékonyabbá tenni. A program segítségével könnyen átláthatóvá válik, hogy mely időszakokban mennyi vendéget sikerül szerezni. Melyek azok az időszakok, amikor sok vendéget sikerült szerezni, és amikor csak keveset. Mekkora a teljes bevételünk, ebből mennyi hasznunk marad.

Felhasználói dokumentáció

Általános leírás-a program célja

Ez a termék, amelyet most a kezében tart a kedves felhasználó, egy vendégház eredményességét nyomon követő rendszer nyilvántartó szoftver. Vállalkozásunk sikerességéhez fontos hogy minden rendelkezésünkre álló információt feljegyezzünk, melyeket aztán összevetve levonhassuk a megfelelő következtetéseket, hogy kihozhassuk a maximumot vállalkozásunkból. Ehhez nyújt segítséget ez a könnyen kezelhető egyszerű alkalmazás. Alig néhány érték bevitelével számos hasznos információhoz juthatunk.

A program szoftver környezete

Windows 7 64-bit operációs rendszer, vagy az feletti (a zökkenőmentes futtatásához)

Xampp 7.3.3 verzió telepítése, mely tartalmazza az Apache webszervert is, (amely webes irányba való tovább fejlesztés estén jöhet jól) és MySQL adatbáziskezelő legfrissebb verzióit. (ajánlott)

A program -hardver környezete

Processzor: 1GHz

Memória: 1GB

A Vadászlak szoftver tárhely szükséglete: 2,1 MB

Adatbázis tárhelyszükséglete: 5Mb éves szinten elég lehet

A program indítása

A program Windows operációs rendszeren futtatható. Indítás előtt telepítenünk kell egy szervert, és egy adatbázis-kezelőt. A nyílt forráskódú és független Xampp szervercsomagot ajánlom. Gyorsan és könnyen telepíthető. A csomagból nem szükséges mindent telepítenünk, elég csak a számunkra szükséges Apache webszervert és MySQL adatbázis-kezelőt installálni. Telepítés után indítsuk el a Xampp kontroll panel-t, majd a start gombokra kattintva futtassuk az Apache és a MySQL modulokat. Ha mindent jól csináltunk, akkor a felíratok háttérszíne zöldre váltott. Ezután a CD-n a Vadászlak nevű mappában található vadászlak.exe fájlon duplán kattintva elindíthatjuk az alkalmazást. Természetesen fel is másolhatjuk a gépünkre a mappát a teljes tartalmával együtt, majd futtathatjuk.

A program használata

Belépés:

A program a következőképpen működik. Indítás után felugrik a bejelentkezés ablak (1. kép).



1. kép (Bejelentkezés)

A program egy jelszót vár tőlünk. Ami alapból az „121212” van beállítva. Ezt begépelve a jelszó mezőbe nyomhatunk egy enter-t vagy a „Belépés” feliratú gombra kattintva beléphetünk a programba. Biztonsági okokból a jelszó kitöltése közben a beviteli mezőben „\*” karakterek fognak látszani. A jelszót belépés után érdemes megváltoztatni. Ezt a menü „Jelszó” felirata alatt megtehetjük.

A felhasználói felület:

A program fő ablakában két fontosabb területet különböztethetünk meg. Az egyik a fejléc, az ablak felső területe, ahol a program neve „Vadászlak Vendégház”, és a menü is található. Másik rész az ablak fő része ahol a menü kiválasztott részei nyílnak meg. A Vendég oldal az alapértelmezett. Az éppen aktuális menü elem háttérszíne feketére vált. A különböző menüpontok alatt található oldalak az alábbiakban kerülnek részletes bemutatásra.

Vendég oldal:

Belépést követően a „Vendég” (2. kép) oldal fogad minket.

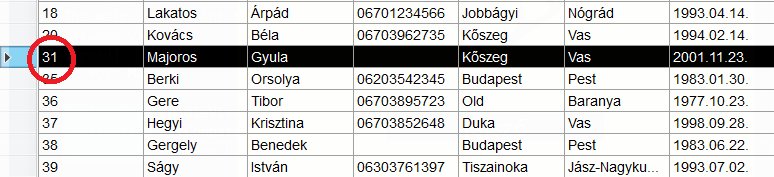


2. kép (Vendég oldal)

Ezen az oldalon előszőr rögzíteni kell vendégek adatait, hogy a későbbiekben foglalásokat tudjunk hozzájuk rendelni. Ezt az „Új vendég ” feliratú gombra kattintva tehetjük meg. Az oldal bal alsó sarkában található a „Vendég űrlap”. Az űrlap kitöltése után a „Mentés” gombra kattintva, az új vendég egyből megjelenik az oldalon található táblázatban. A „Vezetéknév” és „Keresztnév” kötelezően kitöltendő mező. Ide tartozik még a születési dátum beviteli mező is, amely alapból mindig az aktuális napra van állítva, amit ha nem állít át a felhasználó, a program figyelmeztetni fogja, hogy állítson be érvényes születési dátumot. A „Mégse” gomb törli az űrlap tartamlát, és vissza állítja az alapértelmezett értékeket. Abban az esetben, ha hibás adatokat vittünk volna fel a táblázatba, jelöljük ki a módosítani kívánt sort, majd a „Szerkeszt” gombra kattintva vissza kerülnek az értékek az űrlap mezőibe, amik a „Mentés” gomb helyén megjelenő „Módosít” gomb megnyomása után ismét visszakerülnek a táblázatba. Természetesen lehetőségünk van a táblázat sorainak eltávolítására is, amennyiben az „Eltávolít” gombra kattintunk. Ha a program indulásakor már vannak adataink, akkor azok egyből betöltődnek a táblázatba. A táblázat első oszlopa a kijelölt vendég egyedi azonosítója a későbbiekben („ID”) a program által generált mező, aminek az a szerepe, hogy a felhasználó láthassa, hogy későbbiekben bemutatott foglalás, ténylegesen a kijelölt vendéghez tartozik e. A táblázat melletti jobb felső sarokban találunk egy „Szűrés” feliratot. Az alatta lévő lehulló listából lehetőségünk van kiválasztani a táblázat fejlécének egy celláját. Ezt követően az alatta elhelyezkedő „Keresés…” feliratú mezőbe kattintva végezhetjük a táblázat sorainak kiválasztását, a megadott feltételek alapján. Minden egyes billentyű leütés után frissülni fog a táblázat, aszerint hogy van e egyezés a mezőbe bevitt tartalom és a már meglévő vendég adataink között. Ha újra a teljes táblázatot szeretnénk megjeleníteni, akkor egyszerűen csak töröljük ki a mező teljes tartalmát. Bármely oszlop fejlécén kattintva rendezhetjük a tábálázat tartalmát az adott oszlop értékinek ABC szerint növekvő illetve csökkenő sorrendben.

Foglalás felvétele:

Ha egy foglalást szeretnénk rögzíteni, előtte figyeljünk, hogy a táblázat melyik sorában illetve cellájában állunk a Vendégek oldalon, mert az ott kijelölt vendéghez fog tartozni az újonnan rögzített foglalás.(3. kép)

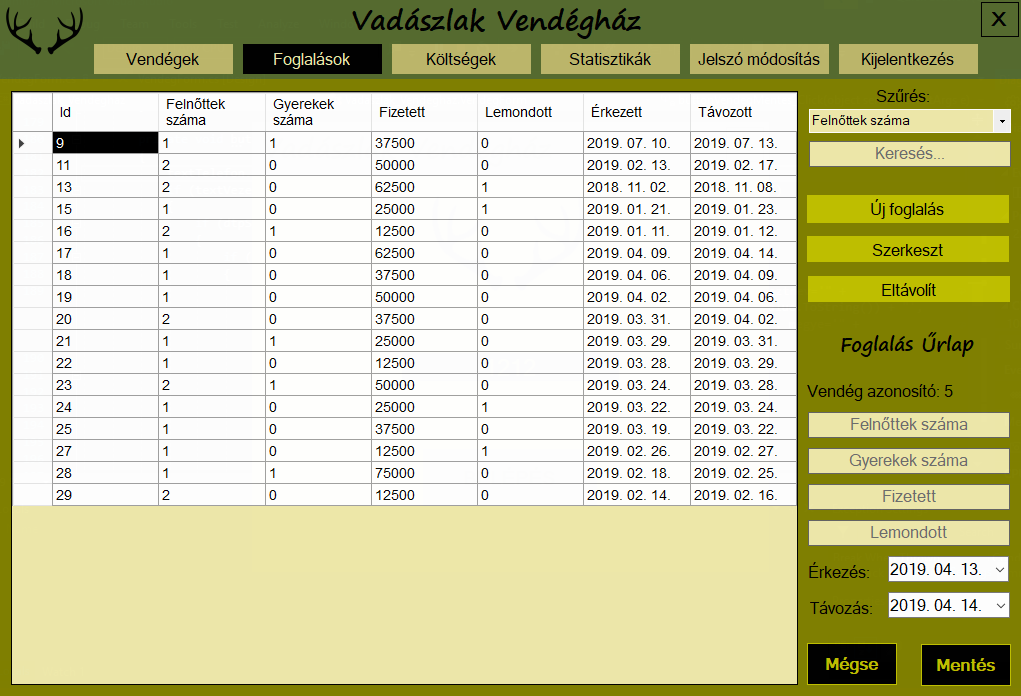


3. kép (Vendég kijelölése)

A kijelölt vendég ID-ja megjelenik a Foglalások oldalon is. Ha több foglalást szeretnénk rögzíteni különböző vendégekhez, akkor Vendégek oldalon először mindig válasszuk ki azt a személyt, akihez a foglalást rögzíteni kívánjuk. Ezután kattintsunk a menü Foglalás menüpontjára. További szükséges műveletek elvégzése a Foglalás résznél olvashatók.

Foglalás oldal:

A felső menüben található „Foglalások” gombra kattintva, a Foglalások oldalra érkezünk (4. kép).



4. kép (Foglalások)

Amennyiben már rendelkezünk foglalási adatokkal, akkor azok azonnal be is töltődnek az oldalon található táblázatba. Az „Új foglalás” gombra kattintva felvihetjük a kiválasztott vendéghez tartozó foglalást az oldal bal alsó sarkában elhelyezett „Foglalás űrlap” kitöltésével. Nem muszáj minden adatot rögzítenünk, csupán a „Felnőttek száma” és a „Fizetett” mezők kitöltése kötelező. Amennyiben nem töltjük ki a kötelező mezőket az alkalmazás erre figyelmeztetni fog minket. Ha a „Lemondott” mezőt üresen hagyjuk akkor automatikusan „0” értéket fog kapni, ami azt jeleni, hogy nem számít lemondottnak a foglalás. Természetesen mi magunk is adhatunk neki 0 vagy 1-es értéket. Utóbbi érték azt jelenti, hogy a foglalást visszamondták. Egy foglalás akkor számít lemondottnak ha „1”-es értéket kap. Előfordulhat, hogy egy korábban felvitt foglalást időközben lemondanak, ebben az esetben lehetőségünk van a későbbiekben is módosítani. A „Mentés” feliratra kattintva az űrlap tartalma eltárolásra kerül a táblázatba. A „Szerkeszt” feliratú gombra kattintva lehetőségünk van megváltoztatni a táblázatunkban megjelenő értékeket. Jelöljük ki a szerkeszteni kívánt sort, vagy a sornak egy celláját, majd a gombra kattintva az adatok visszakerülnek a „Foglalás űrlap” mezőibe, melyeken lehetőségünk van megváltoztatni, végül a „Módosít” gombra kattintva rögzíthetünk. A táblázatból az „Eltávolít” gombbal törölhetünk. A „Szűrés” résznél itt is van lehetőségünk szűkíteni a tartalmat a már említett módon.

Költségek oldal:

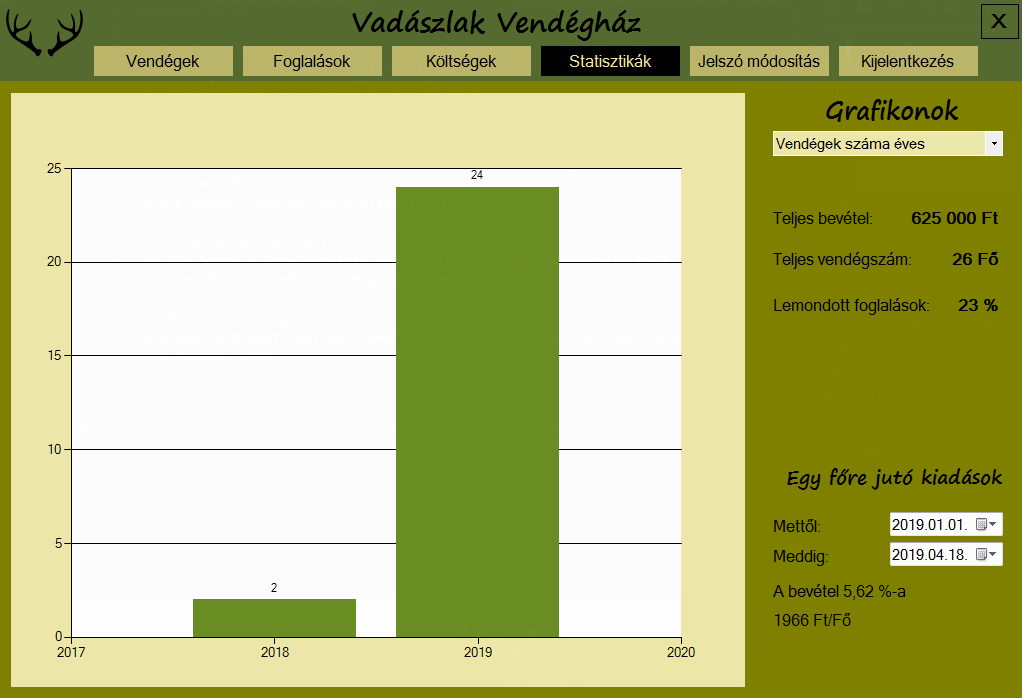
A felső menüben lévő „Költségek”–re klikkelve juthatunk a költségek oldalra, ahol szintúgy, mint eddig, ha már vannak meglévő adataink, akkor megjelennek az oldalon lévő táblázatban. Az „Új költség” gombbal tudjuk kiadásainkat felvinni a táblázatba. Itt arra kell figyelni, hogy legalább egy költség típus ki legyen töltve, különben a program nem rögzít semmit. A törlés a már ismertetett módon történik. Szintén van lehetőségünk meglévő adataink szűkítésére.

Statisztikák oldal:

A menüsor „Statisztikák” elemén kattintva érhetjük el a meglévő adatainkból kimutatható, különböző statisztáinkat. Az oldal nagyrészét a grafikon foglalja el. A jobb felső sarokban a „Grafikonok” résznél választhatjuk ki egy legördülő listából azt a grafikont, amelyikre kíváncsiak vagyunk. Tíz féle grafikon áll rendelkezésünkre. melyek az alábbiakban kerülnek ismertetésre.

Vendégek száma éves:

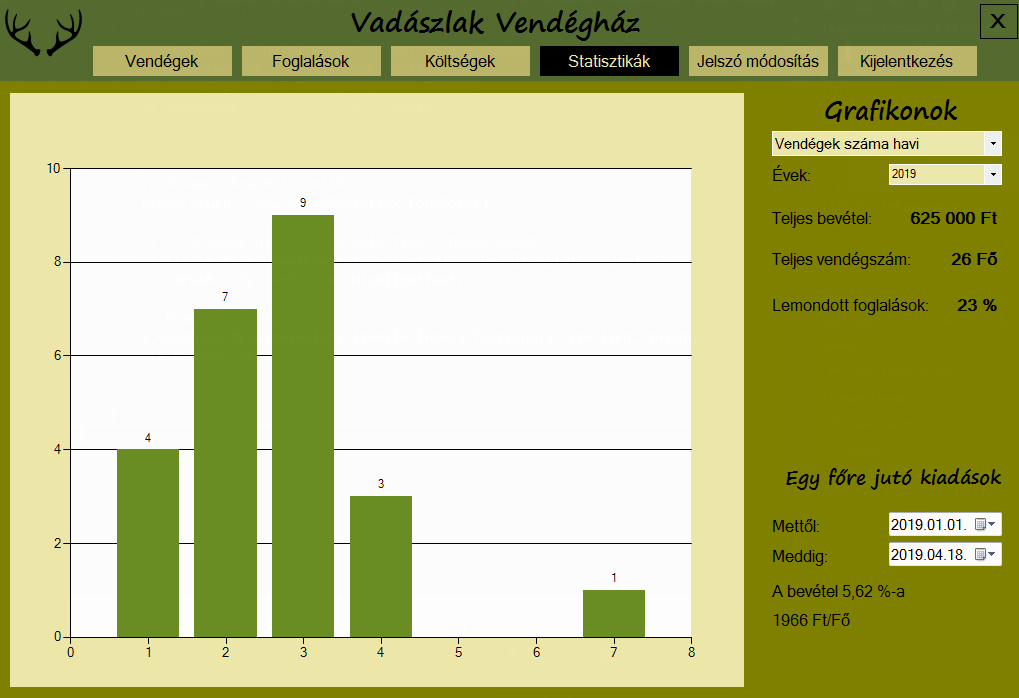
Alapértelmezetten ez a grafikon jelenik meg. Mint ahogy az elnevezéséből is könnyen kiderül, a vendégek számát jeleníti meg évek szerint összegezve azokat.



5. kép (Vendégek száma éves)

Vendégek száma havi:

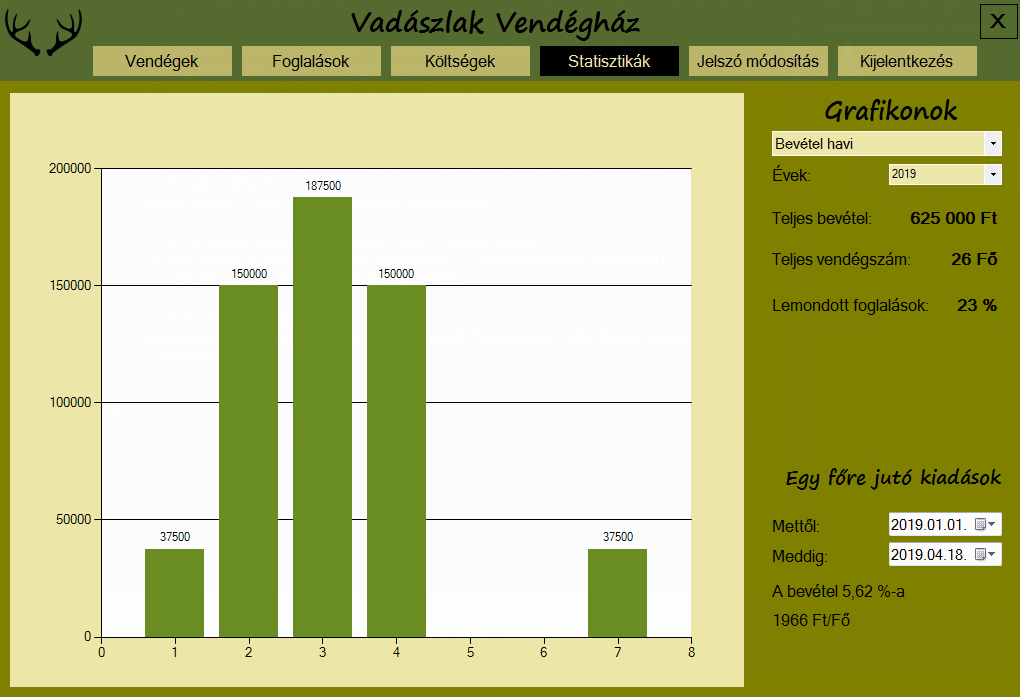
A vendégek számát összegezi havi szinten. Ha ezt az opciót válasszuk és a későbbiekben is olyan grafikont ahol havi szintű adatokat akarunk megjeleníteni, ott felugrik egy újabb legördülő lista, amiben választhatunk azok közül az évek közül, melyekben már voltak adataink. Mindig az aktuális év jelenik meg alapból.



6. kép(Vendégek száma havi)

Bevétel éves / Bevétel havi:

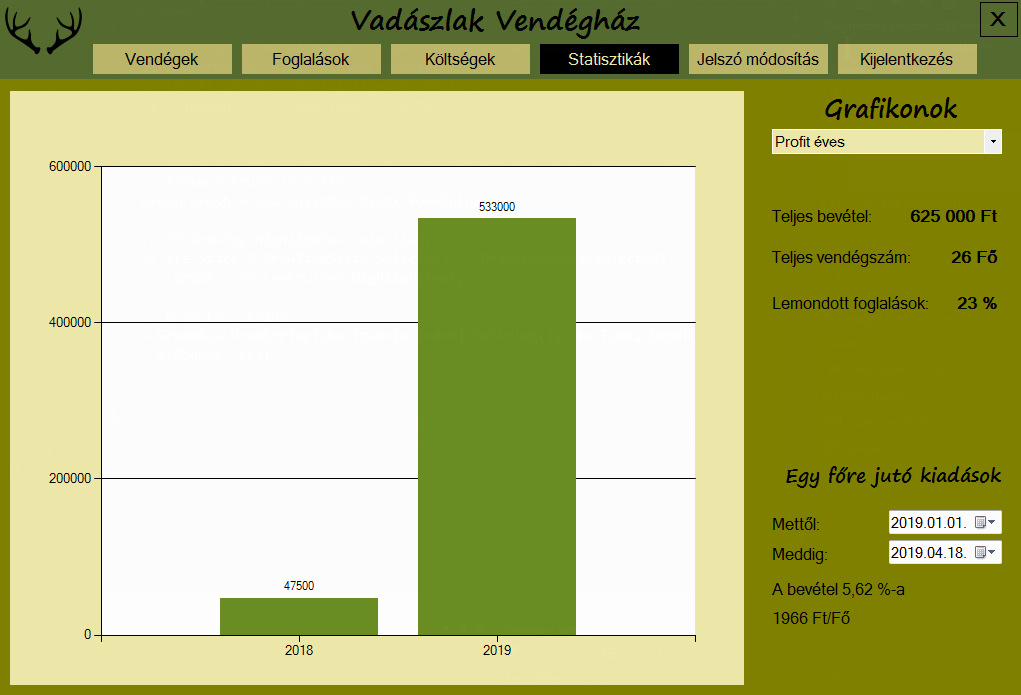
A vendégház bevételeit összegezi évek szerint csoportosítva, ebből még nem jönnek le a költségek. A már említett módon választhatunk az évek közül.



7. kép (Bevétel havi)

Profit éves:

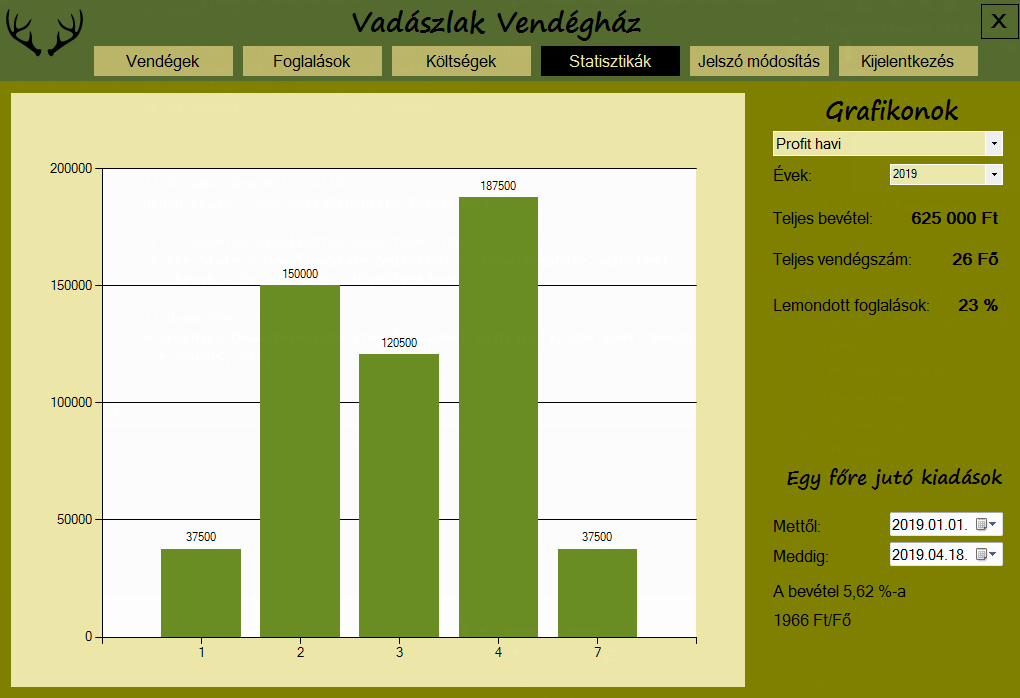
A profit éves annyiban tér el a „Bevétel éves”-től, hogy ebből már lejönnek az adott időszakra vonatkozó költségeket.



8. kép (Profit éves)

Profit havi:

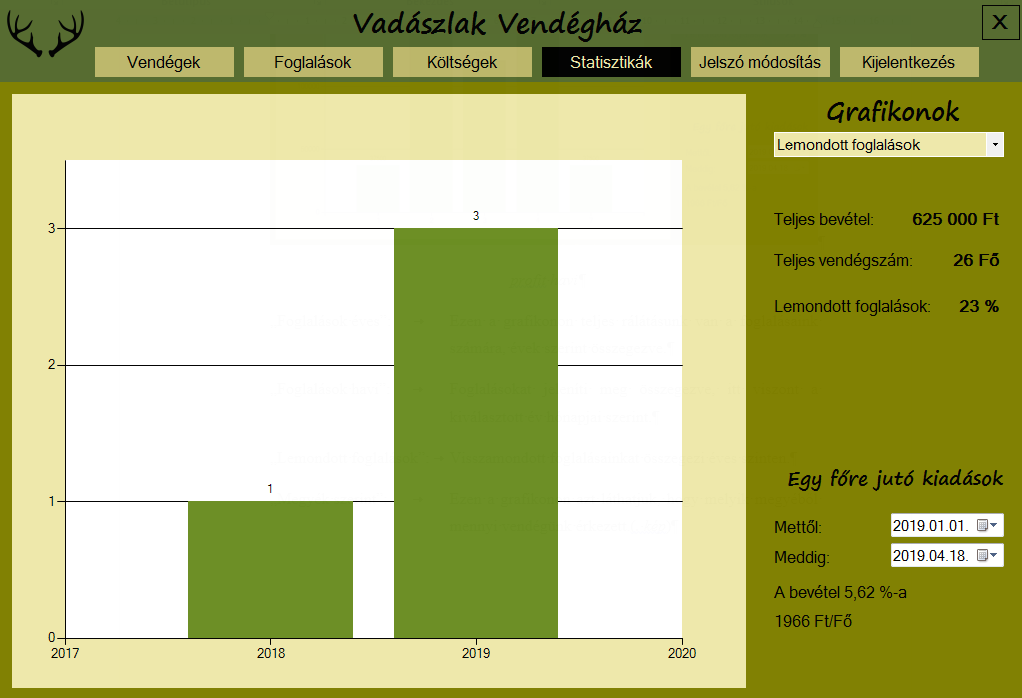
Lényegében havi szinten vizsgálhatjuk Apartmanunk profitját, a kiválasztott év alapján.



9. kép (Profit havi)

Lemondott foglalások:

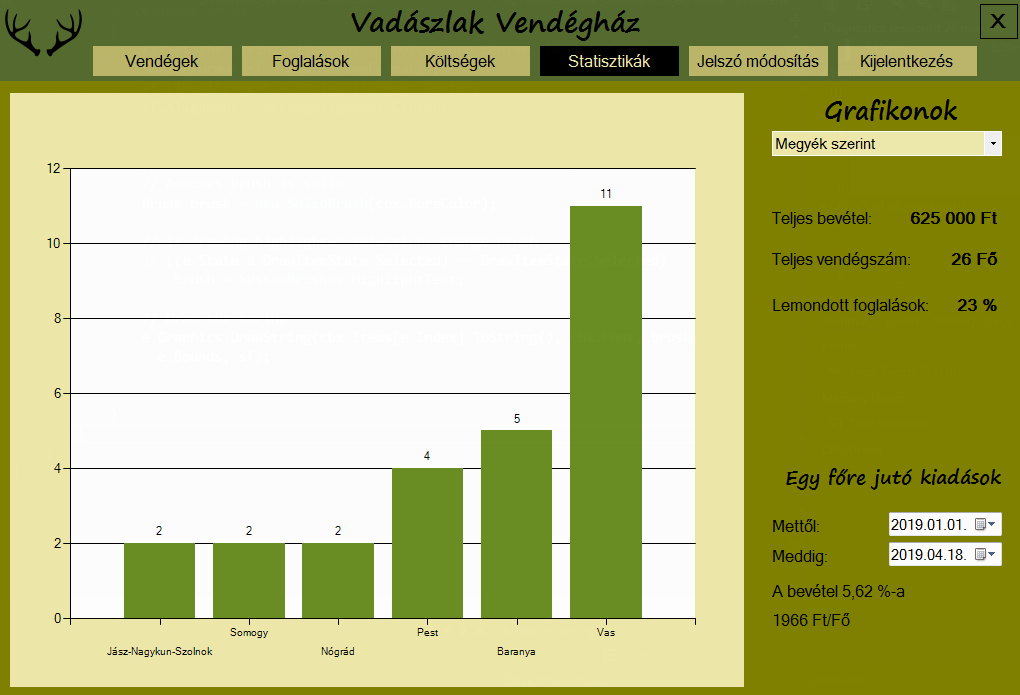
Visszamondott foglalásainkat összegezi éves szinten.



10. kép (Lemondott foglalások)

Megyék szerint:

Ezen a grafikonon azt láthatjuk, hogy melyik megyéből mennyi vendégünk érkezett.



11. kép (Megyék szerint)

Foglalások éves:

Ezen a grafikonon teljes rálátásunk van a foglalásaink számára, évek szerint összegezve.

Foglalások havi:

Foglalásokat jeleníti meg összegezve, itt viszont a kiválasztott év hónapjai szerint.

A Statisztikák oldal jobb oldalán található még néhány kimutatás, ami nem került bele a grafikonba. Itt jelenik meg a „Teljes bevétel”, ami mindenkori bevételeinket összegezi. Ez alatt látható a teljes eddigi vendégeink száma. Ezt követi a „Lemondott foglalások”, ami a százalékosan jeleníti meg a lemondásokat, az összes foglaláshoz képest. A bal alsó sarokba kerültek az egy főre jutó kiadások. Letőségünk van beállítani az időszakot, amit látni szeretnénk. Ha ezt nem tesszük meg, a „Mettől” után mindig az aktuális év első hónapjának első napja fog szerepelni a „Meddig” után pedig az aznapi dátum. A dátum beállítása után automatikusan frissülnek az adatok, melyek százalékos formában is megjelennek, alatta pedig forintban is látható az egy főre jutó kiadás.

Jelszó módosítás:

A menüsorban található a „Jelszó módosítás” gomb. Már az első belépés után érdemes ide kattintani és megváltoztatni a megadott jelszót.(12. kép)



12. kép (Jelszó módosítása)

Az új jelszó létrehozásához előszőr meg kell adnunk a régit, majd egy tetszőlegeset kétszer egymás után. Arra azért van szükség hogy a felhasználó megbizonyosodjon róla hogy valóban azt a jelszót adta meg amit szeretett volna, (nem történt e elgépelés). Ha régi jelszó hibás, vagy az új jelszavak nem egyeznek meg vagy kevesebb, mint négy karakterből állnak, sikertelen a jelszó módosítása. Erre figyelmeztetni is fog minket a program. Ha minden feltételt teljesítünk, akkor megjelenik a „A jelszó módosítása sikeres!” felirat.

Kijelentkezés:

A menü utolsó eleme a „Kijelentkezés”. Erre klikkelve a Bejelentkezés Ablak ugrik elő. A fő ablak elérhetetlenné válik, viszont a program továbbra is fut.

Kilépés a programból:

A programból való kilépés nagyon egyszerűen a megszokott módon történik, amivel már sokféle programban találkozhattunk. A bejelentkezés és a fő ablak jobb felső sarkában találhatunk egy-egy „X”-et. Bármelyiken kattintva a teljes alkalmazás bezárul.

Hibajegyzék

Az adatbázist úgy építettem fel, hogy nem muszáj megadni a vendég születési dátumát, ez a programozás során elkerülte a figyelmem, tehát kötelező beviteli mező lett belőle.

A „Textbox”-oknál nem kezeltem le, hogyha a bemeneti értéknek számot vár, akkor ne lehessen bele más karaktereket írni, ha meg szöveget vár, ne lehessen bele számot írni. Továbbá hogyha pozitív számot vár beviteli mező, ne lehessen bevinni negatív számot. Ha pedig tört számot szeretne eltárolni a felhasználó, akkor ne egy kerekített számot tároljon el a program, hanem írjon ki egy hiba üzenetet. Be kell állítani, hogy a „Lemondott” mező csak „0” vagy „1” értéket kaphasson.

Jogszabályi háttér

Szerzői jogok

A szoftvereket a szerzői jog védi. Szerzői jogomat fenntartom, a szerzői jogról szóló 1999. évi LXXVI.törvény alapján! Tilos a szoftvert engedély nélkül másolni, sokszorosítani, visszafejteni. Engedélyezett egy darab biztonsági másolat készítése, arra az esetre, ha elveszítené a CD, vagy használhatatlanná válna a lemez. Üzletszerű tovább értékesítése nem megengedett. Amennyiben a felhasználó nem kívánja tovább használni a terméket, a használati jog értékesíthető. Ezt megelőzően kötelező a szerzőt írásban tájékoztatni. A szoftver jogosulatlan használata eljárást von maga után.

Fejlesztői dokumentáció

Rendszerterv

A Visual Studioban C# nyelven írt szoftver nem igényel különösebb hardvert. Könnyedén futtatható egy egyszerűbb irodai számítógépen.

A program szoftver környezete

Windows 7 64-bit-es operációs rendszer (a zökkenőmentes futtatásához)

Xampp 7.3.3 verzió telepítése, mely tartalmazza az Apache webszerver, és MySQL adatbáziskezelő legfrissebb verzióit. (ajánlott)

Microsoft Visual Studio Community 2017 15.9.14-es verzió

Microsoft .NET Framework 4.7.03062–es verzió

A program -hardver környezete

Processzor: 1GHz

Memória: 1GB

A Vadászlak szoftver tárhely szükséglete: 2,1 MB

Adatbázis tárhelyszükséglete: 5Mb éves szinten elég lehet

Biztonsági kérdések

A szoftverben md5 (Message-Digest algorithm 5) jelszó titkosítást használtam, ami egy 16 bájt hosszú „hash”-t hoz létre. Ez egy 128 bit-es egyirányú kódolási algoritmus. Ebben a formában tárolja az adatbázis a jelszót, nem pedig a felhasználó által megadott alakban. A végeredmény egy visszafejthetetlen jelszó. Emellett a jelszó mezők PasswordChar tulajdonságát állítottam be „\*” karakterre.(12. kép)

A programban SQL műveletek elvégzéséhez „try-catch” blokkot használtam. A program futása közben keletkező hibák kiszűrésének eszköze. Azoknál a program részeknél érdemes használni ahol gyakran léphetnek fel hibák. A szerverhez való kapcsolódás ilyennek számít. A „try” blokkba kell helyezni azt a program részt, ahol a hiba előfordulhat. A „catch” ágban kezelhetjük az esetlegesen fellépő hibákat.

A programban számos helyen kellett kezelni azokat az eseteket, amikor a felhasználótól kérünk be adatokat. Ezekre különböző bemeneti mezőket alkalmaztam. Az alábbiakban bemutatom őket, és felhívom a figyelmet a kezelendő esetekre.

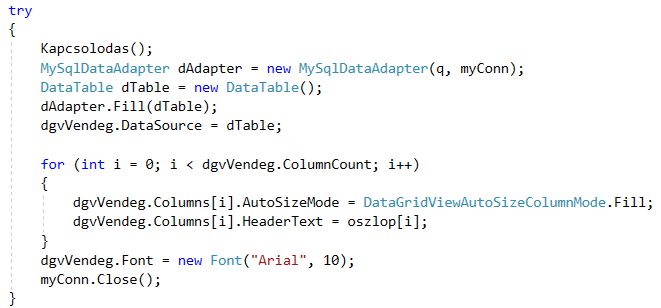
Navigáció

Beviteli mezők:

TextBox: Szöveges beviteli mező. A felhasználó a perifériás eszközéről bármilyen karaktert bejuttathat ezen keresztül a programba. Problémát okozhat, ha olyan karakterek kerülnek bele, amik nem felelnek meg például az adatbázisban adat tárolás céljából létrehozott mezők típusával. Jó példa rá, hogy amennyiben a keresés résznél a ComboBox SelectedItem-je születési dátumra van állítva, tehát a felhasználó a táblázatban megjelenő dátum formátum alapján szeretné szűkíteni a találatokat, nem értené, hogy miért nem jelenik meg semmi a táblázatban, amikor láthatta, hogy van a keresési feltételnek megfelelő tartalom. Ennek a problémának a hátterében az áll, hogy az SQL-ben tárolt „DATE” típusú értékek kötőjellel tagoltak, míg a DataGridView-ban pont választja el egymástól az évet, hónapot, és a napot. Ezt karakterek cseréjével oldottam a „Replace” függvény segítségével. A „DateTimePicker” beviteli mezőnél ugyanez a helyzet. A legtöbb helyen belépési és elhagyási eseményfigyelőket állítottam be a TextBoxoknak, Ami nem csak jól néz ki, hanem hely takarékosság szempontjából is fontos.

Adatok feltöltése:

Következő képpen történik: Adatbázishoz való kapcsolódás után, egy új MySqlDataAdapter példányt feltöltök értékekkel, az SQL parancs lefuttatásával a kapcsolaton keresztül. Létrehozok egy DataTable példányt is, aminek átadom az adapter értékeit. Végül a DataGridView DataSource-a kapja meg a DataTable értékeit (13. kép).



13. kép (Adatfeltöltés)

Az osztály elején deklaráltam egy string oszlop nevű tömböt, amit statikus értékekkel töltöttem fel. Ezek az értékek kerülnek bele a DataGridView fejlécébe, ez azért jó, mert így nem az adatbázisban tárolt ékezet nélküli oszlopnevek jelennek meg. Ezt a metódust meg tudom hívni a textKeres Textbox TextChanged esemény meghívásakor, mivel az SQL utasítást úgy írtam meg, hogy textKeres mező tartalmával megegyező adatokat töltsön be, amennyiben ez üres akkor minden adatot betölt. A keresés mező fölött elhelyezett ComboBox érétkeit szintén az előbb említett oszlop tömbből kapja, ezért egy Switch-Case kapcsolóval vissza állítom a ComboBox ékezetes értékeit, az ékezet nélküli mezőneveknek megfelelőre.

Adatok törlése:

A „vendeg” és a „foglalas” táblákból az idegenkulcs megkötés miatt az adatokat először a szallas nevű (kapcsoló) táblából kell törölni, utána lehet csak az említett táblákból. Fontos hogy használjuk „where” megkötést, különben az egész tábla tartalma törlésre kerül (7. kép).



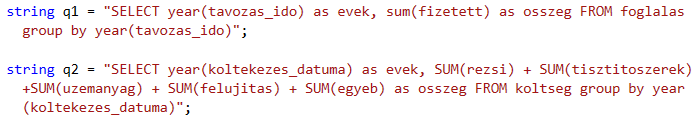
14. kép (Törlés)

Új felvétel /módosítás:

A két művelet elvégzésére egy gombot használtam. A gomb felirata szerint hajtja végre a program a megfelelő utasítást. Ehhez a vizsgálathoz az „Equals()” függvényt használtam, ami két string-et hasonlít össze. Mindkét esetben csak akkor fut le a megfelelő parancs, ha legalább a vezetéknév, keresztnév és a DateTimePicker value-ja meg van adva. A white-space-k eltávolítására a Trim() függvényt használtam. Ügyelni kell rá, hogy a gombok feliratát a megfelelő helyen átállítsuk. Új felvétel esetén először is kapcsolódtam az adatbázishoz, majd egy új „myComm1” nevű MySqlCommand példányt hoztam létre, ami két paramétert vár, az SQL utasítást és a kapcsolatot. Ezután a MySqlDataReader példányának adtam át a „myComm1”-et a rajta meghívott ExecuteReader() nevű függvénnyel, ami lefuttatja azt és az értékek bekerülnek az adatbázisba. A folyamat végeztével a kapcsolaton meghívott Close() függvénnyel le kell zárni azt. A módosítás hasonló képpen történik, két dolgot emelnék ki, amire figyelni kell. Ne felejtsük el megadni az utasításnál a módosítani kívánt sor azonosítóját, különben az egész tábla tartalma módosításra kerül, és állítsuk vissza a gomb textjét „Mentésre”.

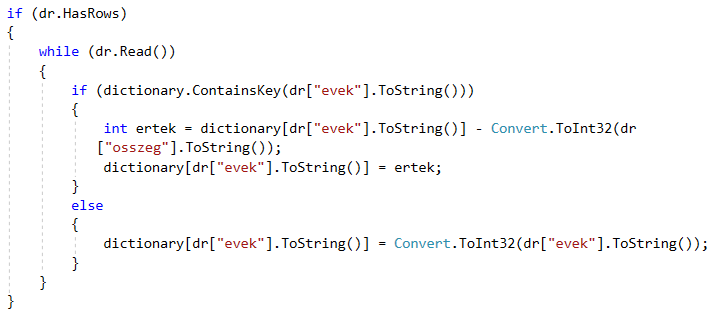
Profit

Ez egy érdekesebb metódus, ami grafikonon jeleníti meg nyereséget évek szerint csoportosítva. Kell két SQL utasítás, aminek segítségével hozzájutunk a szükséges adatokhoz az adatbázisból, az egyik a bevételeket kéri le, a másik meg a költségeket évek szerint csoportosítva (8. kép).



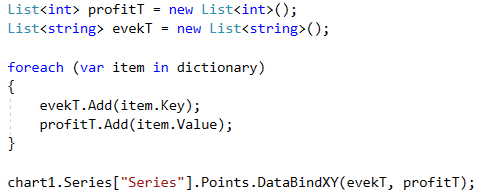
15. kép (SQL utasítások)

Az utasításokat lefuttatva a MySqlDataReader „dr” nevű példányán keresztül érjük el az adatokat. A HasRows() függvény „true” értékkel tér vissza, ha kapunk vissza értéket. Ebben az esetben tudjuk a Read() függvénnyel olvasni. Az olvasás egy While ciklus feltételébe kerül, így a program addig olvas, amíg van mit. Az adatok tárolására Dictionary-t használtam. Ez a megoldás azért volt hasznos, mert kulcs-érték párokat tárol. Mivel „dr” is asszociatív indexek alatt tárolja az értékeket, a megfelelő indexek értékeinek összehasonlítása után, elvégezhetők a műveletek. Ha van olyan év, ahol van bevétel és költség is, akkor levonásra kerül a költség a bevételből (9. kép).



16. kép (Dictionary feltöltése)

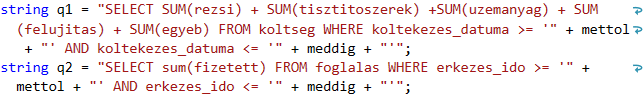
Ezután két listát hoztam létre. A listákat a metóduson belül kell létrehozni, mert különben, a függvény minden egyes meghívásakor hozzá fűzné az értékeket a már meglévőkhöz. Az evekT listát az évekkel a profitT-t pedig az adó levonása után a profittal töltöttem fel. Végül a grafikon a DataBindXY() függvénytől megkapja a listák értékeit (10. kép).



17. kép (Grafikon feltöltése)

Egy főre jutó kiadások

Az érdekessége abban rejlik, hogy a felhasználó két DateTimePickerben (dtpMettol, dtpMeddig) állíthatja be a számára érdekes idő intervallumot. Majd azok Value-ja kerül bele az SQL utasításba, a változók értékeiben a már korábban említett módon le kell cserélni a karaktereket. A függvény Value Changed esemény hatására kerül meghívásra (11. kép).



18. kép(SQL egy főre jutó kiadások)

Adatbázis

MySQL (MariaDB) adatbázis-kezelőt használtam a rendszer felépítéséhez. Az adatbázist HeidiSQL-ben hoztam létre a localhost-on, aminek a „vendeghaz” nevet adtam, amennyiben szükséges a megfelelő IP cím beállításával távoli szerverről is elérhetővé tehető. „latin2\_hungarian\_ci” karakter kódolást választottam, mivel ez az egyik leghatékonyabb karakter kódolás. Hat darab táblát vettem fel, melyek a következők: foglalas, jelszo, koltseg, szallas, vendeg.

**foglalas** tábla: A foglalásokhoz kapcsolódó adatokat tárol. Mezőinek nevei: id, felnottek\_szama, gyerekek\_szama, fizetett, lemondott, erkezes\_ido, tavozas\_ido;

**id**: Ez a kulcs, egyedi automatikusan növekvő érték. Típusa int(integer), előjel nélküli, és 11 karakter hosszú lehet.

**felnottek\_szama:** Felnőttek darab számát jelöli. Típusa int, 11 karakter hosszú, előjel nélküli és nem lehet üresen hagyni a mezőt.

**gyerekek\_szama:** Gyerekek darabszámát jelöli. Típusa int, 11 karakter hosszú, és lehet üres a mező.

**fizetett:** Foglalásért fizetett összeg értékét tárolja. Típusaik int, és 11 karakter hosszú lehet, előjel nélküli, lehet üres a mező.

**lemondott:** Lemondott foglalások követésére hoztam létre. Típusa tinyint, 4 karakter hosszú lehet, előjel nélküli szám, lehet üres a mező. A 0 érték jelöli hogy nem volt lemondás, az 1 pedig hogy volt.

**erkezes\_ido, tavozasi\_ido:**

Vendég érkezési és távozási idejét tárolja. Típusuk date, ami az évet, hónapot, és a napot jeleníti csak meg.

**jelszo tábla:** A jelszót tárolja. Mezőnevei: id, jelszo.

**id:** Ez a kulcs. Típusa tinyint, 3 karakter hosszú, egyedi automatikusan növekvő érték.

**jelszo:** Belépési azonosításra szolgál. Típusa varchar, 50 karakter hosszú.

**koltseg** tábla**:** vendégház üzemeltetése során keletkező költségeket tárolja. Mezőnevei: id, rezsi, tisztitoszerek, uzemanyag, felujitas, egyeb, koltekezes\_datuma.

**id:** Ez a kulcs. Típusa int, 11 karakter hosszú, előjel nélküli, egyedi automatikusan növekvő érték.

**rezsi, tisztitoszerek, uzemanyag, felujitas, egyeb:**

Költségek fajtáinak megkülönböztetésére szolgálnak. Típusaik int, 11 karakter hosszúak lehetnek, előjel nélküliek, üresen hagyható mezők.

**koltekezes\_datuma:** A költség felvételének idejét tárolja. Típusa datetime, Alapértelmezetten kapja értékül az éppen aktuális időt(current\_timestamp) .

**szallas** tábla: Egy kapcsoló tábla, ami tárolja a vendég azonosítóját és a foglalás azonosítóját. Segítségével követhető hogy melyik vendéghez melyik foglalás(ok) és melyik foglaláshoz melyik vendég(ek) tartoznak. Csak létező vendéghez lehet felvenni foglalást. Mezőnevei: „id”, „v\_id”, „f\_id”.

**id:** Szállás azonosítója. Ez a kulcs. Típusa „int”, 11 karakter hosszú, előjel nélküli, egyedi automatikusan növekvő érték.

**v\_id:** Ez egy idegenkulcs, egyeznie kell a „vendeg” tábla azonosítójával. Típusa „int”, 11 karakter hosszú, előjel nélküli, nem lehet üres a mező.

**f\_id:** Ez egy idegenkulcs, egyeznie kell a „foglalas” tábla azonosítójával. Típusa „int”, 11 karakter hosszú, előjel nélküli, nem lehet üres a mező.

**vendeg** tábla: A vendégek adatait tárolja. Mezőnevei: id, vezeteknev, keresztnev, telefonszam, lakcim, megye, szuletesi\_datum.

**id:** A vendég azonosítója. Ez a kulcs. Típusa int, 11 karakter hosszú, előjel nélküli, egyedi automatikusan növekvő érték.

**vezeteknev:** A vendég vezetékneve. Típusa varchar, hossza 50 karakter. Nem lehet üres a mező.

**keresztnev:** A vendég keresztneve. Típusa varchar, hossza 50 karakter. Nem lehet üres a mező.

**telefonszam:** A vendég telefonszáma. Típusa varchar, hossza 20 karakter. A mező lehet üres.

**lakcim:** A vendég pontos lakcíme. Típusa varchar, hossza 50 karakter. A mező lehet üres.

**megye:** A megye ahonnét a vendég jött. Típusa varchar, hossza 30 karakter. A mező lehet üres.

**szuletesi\_datum:** A vendég születési ideje. Típusa date, ami az évet, hónapot, és a napot jeleníti csak meg. A mező lehet üres.

Ergonómia

Ergonómiát tekintve igyekeztem megfelelni az általános felhasználói igényeknek. Nagy hangsúlyt fordítottam könnyű kezelhetőségre, átláthatóságra. Az oldal felső részén helyeztem el a fő menüt, elkülönítve a többi tartalomtól. A gomboknak akkora mértet adtam, hogy könnyedén lehessen kattintani rajtuk, kellő távolságra legyenek egymástól. Több féle betűméretet használva próbáltam a különböző tartalmak fontossági sorrendjét érzékeltetni. Színek összeállításnál arra figyeltem, hogy ne használjak túl sok féle színt, inkább azok különböző árnyalatait. Ügyeltem a szövegek olvashatóságára.

Tesztelés

A szoftver tesztelésének elsődleges célja a szoftver megfelelő minőségének biztosítása. Másképpen kell tesztelni, egy Atomerőműnek szánt programot és egy beadandót is. Sok esetben nem lehet minden lehetséges bemeneti kombinációt tesztelni, és nem is érdemes a számtalan beviteli variáció miatt. A magas kockázatot okozó értékekre érdemes fektetni a hangsúlyt. Érdemes már a kisebb kódrészletek is tesztelni, nem csak kész projektet. Ezáltal a hibák könnyebben fellelhetők, és gyorsabban javíthatók. Dolgozatom készítése közben számtalan tesztet végeztem. Legtöbb esetben egy-egy kód részlet megszületése után, buildeléssel teszteltem az eredményt. Ha a tesztelni kívánt programrész elindul, akkor a cél, a lehető legtöbb szélsőérték bejuttatása futása közben. Érdemes más embereket is megkérni, hogy próbálják ki a programunkat.

Gyakran előfordultak olyan estek is, amikor az alkalmazás futtatása közben lép fel a probléma. Jobb esetben kapunk hibaüzenetet, amiből gyorsan rálelhetünk a probléma okára. Egyébként a Debugolás is a segítségünkre lehet. Úgynevezett breakpoint-okat kell elhelyezni a program azon részeibe, ahol a hibát feltételezzük, majd lépésről lépésre futtatni azt. Opció lehet egy-egy „teszt” szöveg vagy változó értékének kiíratása is, amiből kiderül hogy az adott pontig lefut e a program, vagy megkapja e a változó az értéket.

Összegzés

A cél megvalósulása

Szakdolgozatom elkészítése során rengeteg tapasztalatot szereztem. Ha csak arra gondolok, hogy tanfolyam elindulását követően, fogalmam sem volt, hogy hogyan leszek képest önállóan megírni egy működő szoftvert, akkor már egész jól látható, hogy milyen messzire jutottam, az első lépések megtétele óta. Nagyszerű lehetőség volt abból a szempontból is, hogy C# nyelvet jobban megismerhettem. Ötleteim nagyrészét sikerült megvalósítanom, de mint tudjuk, kész szoftver nincs, csak használható. Maradtak még benne javításra váró hibák, amiket idővel szeretnék helyreállítani, és látok még benne fejlesztési lehetőségeket is.

Tovább fejlesztési lehetőségek

Sok féle irányba lehetne fejleszteni az alkalmazást. Lehetne tökéletesíteni azokat a részit, amiket elkezdtem fejleszteni. De meg van a lehetősége annak is, hogy új dolgokat vigyen bele a fejlesztő. Néhány javaslatot említenék az alábbiakban.

* Költségeket lehetne például csoportosítva is összegezni, így még hatékonyabban lehetne kezelni a kiadásokat.
* Több adat egyidejű törlése.
* Jó lenne, ha csak annyi vendéghez. tartozhatna egy foglalás, ahány személy szerepel egy foglalásban.
* Hasznos lenne egy sikerességi mutató. Arra gondoltam, hogy százalékban meglehetne jeleníteni, hogy az adott év eltelt napjaihoz képest hányszor volt kiadva az épület.
* Lehetne akár távoli szerverről hozzáférhető az adatbázis.
* A Vendég tábla lakcím mezőjének tartalma jelenleg azon kívül, hogy megjelenik a táblázatban, nincs más szerepe. Ha egészében szeretnénk látni az oszlop adatait szélesebre kell állítani az oszlopszélességet. Úgy persze nem tűnik túl praktikusnak, mivel sok helyet foglal. Vagy ki kellene venni a DataGridView-ból, vagy fontos szerepet kellene adni neki.
* Jó lenne ha, a táblázatok egy-egy sorainak törlése után a törölt sor előtti sor kapná meg a kijelölést.
* A Rezsi oldalon a keresőjét lehetne megcsinálni akár úgy is, hogy bizonyos összeg feletti, illetve alatti eredmények jelenjenek meg.
* A Vendégek oldalra szerettem volna tenni egy „Új foglalás” gombot. Ennek lett volna a funkciója, hogy a vendég kiválasztása után átnavigáljon minket a Foglalások ablakra. Lényegében azt nem sikerült megoldanom, hogy a fő ablaknak a fő paneljében létrehozott vendegform példányának egy gombján kattintva, a foglalasform jelenjen meg a fő ablak fő paneljében.
* Nincs megvalósítva a programban annak a lehetősége, hogy ne csak egy vendéghez lehessen felvenni több foglalást, hanem egy foglaláshoz is tartozhasson több vendég is.

Irodalom jegyzék

Szakkönyvek: Reiter István C# programozás lépésről lépésre, Jedlik Oktatási Stúdió Kft., 1215 Budapest, Ív u. 8-12., 2012

URL hivatkozások: <https://hu.wikipedia.org/wiki/MD5>

<https://stackoverflow.com/>

<https://www.youtube.com/channel/UC8inCnD25Es0VLokfmhko5g/videos>

<http://szerzoijog.com/szoftverjog/>

<http://njt.hu/cgi_bin/njt_doc.cgi?docid=41066>