

## **ZÁRÓDOLGOZAT**

### **Telefon szerviz rendszer**

Szakképesítés neve: Szoftverfejlesztő

OKJ száma: 54 213 05

**Csolti Péter**

Témavezető

**Figler Renáta**

SZF29

**Budapest, 2019**

# Tartalomjegyzék

|   |    |
|---|----|
| Bevezetés .....                                 | 3  |
| Felhasználói dokumentáció.....                  | 4  |
| Általános információk .....                     | 5  |
| A programról .....                              | 5  |
| A program használata .....                      | 6  |
| Alapok .....                                    | 6  |
| Úrlapok menü.....                               | 7  |
| Készletek menü .....                            | 8  |
| Eladás menü .....                               | 10 |
| Statisztikák menü .....                         | 10 |
| Keresés menü.....                               | 11 |
| Kilépés menü.....                               | 12 |
| Elérhetőségek:.....                             | 12 |
| Fejlesztői dokumentáció .....                   | 13 |
| Az alkalmazott fejlesztői eszközök .....        | 14 |
| Adatmodell leírása .....                        | 14 |
| Részletes feladat-specifikáció, forráskód ..... | 18 |
| Tesztelési dokumentáció.....                    | 23 |
| Továbbfejlesztési lehetőségek .....             | 25 |
| Összefoglalás .....                             | 27 |
| Irodalomjegyzék.....                            | 27 |

## Bevezetés

Jelenleg a Huawei szerviz pontján dolgozom, mint ügyfélszolgálatos. Ezáltal beelátok a javítási és készletezési folyamatokba. Ezekre szerettem volna egy egységes, alap rendszert létrehozni, ami később szükség szerint bővíthető újabb funkciókkal. Mivel üzletünk nem rendelkezik még semmilyen egységes programmal, és a folyamatainkra is négy különféle rendszert használunk, ezért gondoltam ezt jó témának.

A programom négy fő részből áll:

1) Készletnyilvántartó: Háromféle dolgot kell készletezni:

- a) alkatrészeket (ezen belül alkatrészek és külön fődarabok)
- b) telefonokat
- c) tartozékokat

Az ide tartozó funkciók: bevételezés, selejtezés; eladáskor vagy javításkor készletcsökkenés, eladva, felhasználva státusszal, dátummal.

2) Szervizes adatlap: A beérkező telefonok rögzítésére szolgál. Két típus lehet: garanciális és fizetős. Javítás során alkatrészeket és fődarabokat kell felvinni az adatlapra, ez le kell kerüljön készletről is. Vezetni kell külön, hogy melyik lett fizetősre és melyik garanciális javításra használva. Szakvéleményt is kell csatolni az adatlaphoz, amennyiben nem javítható a telefon valami miatt.

3) Eladás: Telefon; és tartozék eladása során ezeknek a rendszerből is eladásra kell kerülniük. Ezeket az információkat tárolni is kell, pontos dátummal.

4) Lekérdező felület, statisztikák:

- a) Dátum szerint a felhasznált garanciális alkatrészek, fődarabok áráról szükséges riportot húzni, ezt megtéríti a gyártó + 5% többlet összeget ad.
- b) Dátum szerint a felhasznált fizetős alkatrészek, fődarabok áráról is kell majd riportot készíteni, ez 10% haszonnal működik.
- c) Dátum szerint eladott telefonok, tartozékok áráról, haszonról is kell legyen lehetőség riportot húzni.
- d) Db szerinti riportokra is szükség van, eladott vagy felhasznált telefonra, tartozéokra, alkatrészeire, fődarabra.
- e) Többféle szempont szerinti keresés megoldására is szükség van. (például alkatrész, úrlap, telefon stb.).

# **Felhasználói dokumentáció**

Készítette: Figler Renáta

2019

## **Általános információk**

### **A programról**

#### Ismertető:

A program egy telefon szerviz készletező program. Elsődlegesen a különböző eszközök nyilvántartására szolgál, emellett a teljes szervizelési folyamat adminisztrációjára is alkalmas, a hibás készülék beérkezésétől a javításig.

#### A használatához szükséges környezet:

Windows 7 vagy magasabb operációs rendszer ajánlott a program futtatásához. A program adatbázis háttérét Xampp alkalmazáson keresztül a phpMyAdmin felülete adja. A Xampp az internetről díjmentesen letölthető program. A phpMyAdmin ingyenesen használható felület.

#### Telepítés:

A program használatához nem szükséges internetkapcsolat, telepítést nem igényel. A mellékelt cdről futtatható vagy átmásolható a számítógépre.

#### Használat:

A program használatához felhasználónév és jelszó szükséges. Első alkalommal ez a rendszergazdától kapott adat, a jelszó természetesen megváltoztatható.

## A program használata

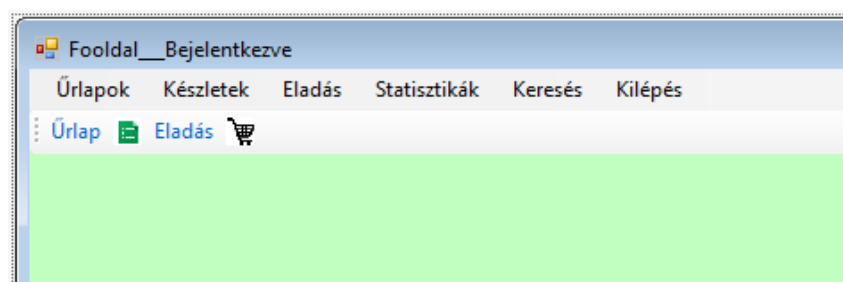
### Alapok

A program megnyitása után az 1a. ábrán látható felület jelenik meg. A jobb felső sarokban lévő gombokkal lehet méretezni, tálcára tenni, illetve bezárni az alkalmazást. A bejelentkező képernyő közepén az éppen aktuális dátum látható. Alatta pedig a felhasználónév és jelszó párost kell beírni, majd a Belépés gomb megnyomásával lehet belépni a programba és használni a funkcióit. Alatta az új jelszó gomb megnyomásával módosítható a jelszó.

Belépés után a főoldalon felül helyezkednek el a menüpontok. Ez látszik az 1b. ábrán.



1a. ábra

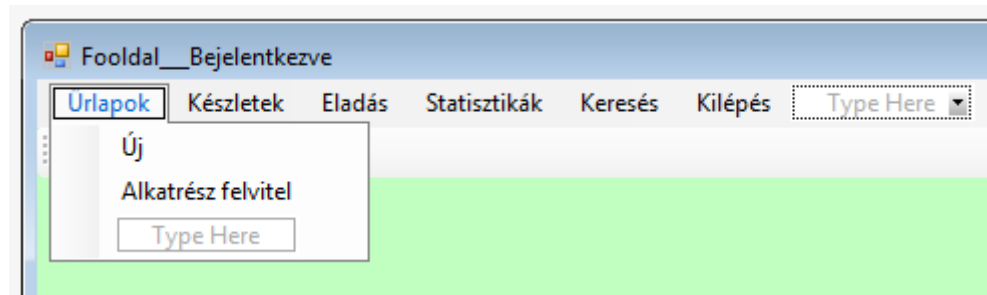


1b. ábra

## Űrlapok menü

Kettő almenüt tartalmaz: (2. ábra)

- Új
- Alkatrész felvitele



2. ábra

Új: Új űrlapot hoz létre, melyen a rossz telefont lehet felvenni javításra. Kötelezően kitöltendő az összes mezője. Az első négy pont az ügyfél adatokat kéri el (név, cím, telefonszám, e-mail cím), ezután listából ki kell választani a telefon típusát, meg kell adni a készülék imei azonosítóját, a hiba leírást, és a javítás fajtáját. Ha minden kész a mentés gombbal az adatok eltárolásra kerülnek. (3. ábra)

A screenshot of a web form titled 'Új' (New) for creating a new report. The form is set against a green background and contains the following fields and controls:

- Név (Name): Text input field.
- Cím (Address): Text input field.
- Telefonszám (Phone Number): Text input field.
- E-mail: Text input field.
- Típus (Type): Dropdown menu with options: P10 fekete, P20 lite vörös, P20 rózsaszín, P8 lite fekete.
- Imei (IMEI): Text input field.
- Hiba (Error Description): Text input field.
- Javítás fajtája (Repair Type): Radio buttons for 'Garanciaváltás' (Warranty replacement) and 'Fizetős' (Paid).
- Dátum (Date): Text input field.
- Mentés (Save): Button at the bottom right.

3. ábra

Alkatrész felvitel: Itt a javítás során felhasznált alkatrészeket és fődarabokat lehet regisztrálni, illetve ha nem lehetett javítani a készüléket, akkor a sablonok közül fel kell tölteni a megfelelő jegyzőkönyvet. Felül a javításra váró telefonok listájából választható ki, amelyikkel éppen dolgozni szeretnénk. A kiválasztott imei típusa és a javítás típusa automatikusan töltődik. A felhasznált fődarabnál és alkatrésznél azokat kell kiválasztani, amelyek beépítésre kerültek. Ha minden kész a mentés gombbal tároljuk el az adatokat. (4. ábra)

|                          |             |
|--------------------------|-------------|
| Javításra váró telefonok | 345678912   |
| Kiválasztott imei típusa | P10 fekete  |
| Javítás típusa           | Garanciális |
| Felhasznált Fődarab      | 745644456   |
| Felhasznált Alkatrész    | p10kameraf  |
|                          | 89          |
| Javítási eredmény        | Javítva     |
| Dátum                    | 2019.01.01. |
| Felvités                 |             |
|                          | Mentés      |

4. ábra

## Készletek menü

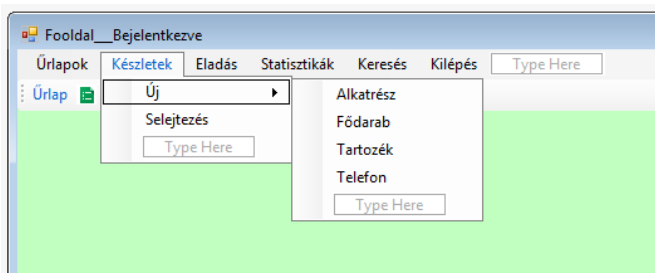
Kettő almenüt tartalmaz:

- Új
- Selejtezés

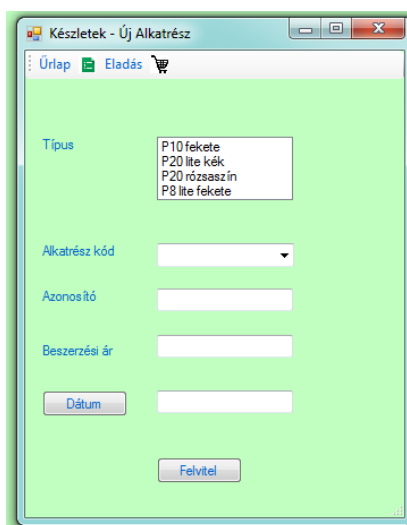
Új: Itt tudjuk raktárkészletre venni az új eszközöket. A felület mind a négy fajta készlet esetén hasonló (6. ábra). A megfelelő típust kell csak kiválasztani, beírni az egyedi azonosítót, illetve a megfelelő árak bruttó értékeit. A felvitel gombbal tudjuk eltárolni az űrlap adatait. Egyszerre egy darab eszköz vihető fel.



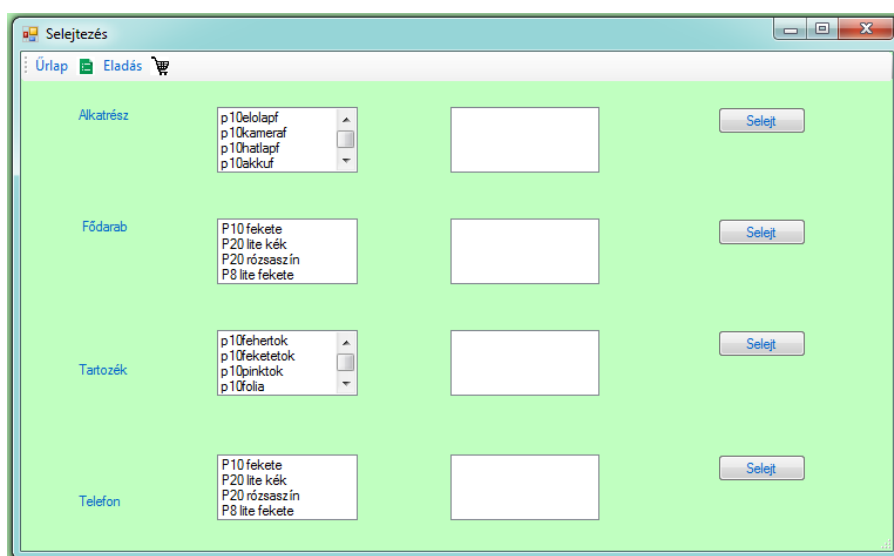
Selejtezés: Ezen a menüponton keresztül lehet törölni a készletről elemeket, amik valami miatt hibásak, vagy használhatatlanná váltak (7. ábra).



5. ábra



6. ábra



7. ábra

## Eladás menü

Ebben a menüpontban lehet regisztrálni a tartozék- és készülékértékesítéseket. A felületen az ügyfél nevét, címét kell megadni, majd kiválasztani az eladott termék típusát. A fizetendő mező gombnyomásra automatikusan töltődik. A termékeket egyesével el is kell adni a rendszerből. A kész gomb megnyomásával az adott termék lekerül készletről, az eladásról txt állomány készül, amely tárolható.

8. ábra

## Statisztikák menü

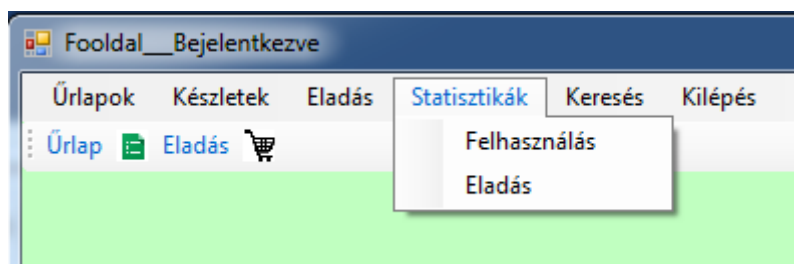
Kettő almenüt tartalmaz:

- Felhasználás
- Eladás

Felhasználás: Az alkatrész és fődarab felhasználást lehet lekérdezni különböző szempontok szerint. Meg kell adni a kívánt időintervallumot, és ki lehet választani, hogy a fizetős vagy a garanciális fogyásról szeretnénk statisztikát. A rendszer darabszámot, vagy forintban kifejezett összeget ad válaszul. A haszon gombbal garanciális javításnál a

gyártó által fizetett, darabonként + 5% hasznot kérdezhetjük le forintban; fizetős javításnál darabonként 10% haszonnal kell számolnunk (10. ábra).

Eladás: Az előzőhöz teljesen hasonlóan működő felület a telefon és alkatrész eladási statisztikáinak lekérdezésére.



9. ábra

10. ábra

## Keresés menü

Öt almenüt tartalmaz:

- Űrlapok
- Alkatrész
- Fődarab
- Tartozék
- Telefon

Ezekon a felületeken tudunk keresni az adatbázisokban, az űrlapok között és a készleten lévő eszközök között, különböző szempontok szerint. Az eredmény mindig az oldal alján jelenik meg, táblázatos formában.

The screenshot shows a web application window titled 'Kereses\_alkatresz'. It has a navigation bar with 'Úrlap' and 'Eladás' links. The main area is green and contains search filters: 'Telefon típusa' with a dropdown menu showing 'P8Lite fekete', 'P20 rozsaszin' (selected), 'P10 fekete', and 'P20Lite kek'; 'Alkatrész kód' with a dropdown menu showing 'p20kamerar'; and 'Keresés eredménye' with a text input showing '2'. Below the filters is a table with 5 columns: 'azonosito', 'tipus', 'alkatreszkod', and 'beszerzesiar'. The table contains two rows of data.

|   | azonosito | tipus         | alkatreszkod | beszerzesiar |
|---|-----------|---------------|--------------|--------------|
| ▶ | 11        | P20 rozsaszin | p20kamerar   | 1111         |
| * | 0         | P20 rozsaszin | p20kamerar   | 256          |

11. ábra

## Kilépés menü

Ezzel tudunk kilépni a programból, ha befejeztük a munkánkat.

## Elérhetőségek:

Készítette: Figler Renáta

Kapcsolattartási e-mail cím: [renatafigler85@gmail.com](mailto:renatafigler85@gmail.com)

## **Fejlesztői dokumentáció**

Készítette: Figler Renáta

2019

## Az alkalmazott fejlesztői eszközök

Az alkalmazás megtervezésekor olyan szoftverekre próbáltam törekedni, amelyek nyílt forráskódúak, illetve törvényesen elérhetőek. A programomat Microsoft Visual Studio 2015-ös változatában írtam, C# nyelven. A Visual Studio a Microsoft több programozási nyelvet tartalmazó fejlesztőkörnyezete. Az adatbázis kapcsolat létrehozásához Xampp alkalmazást telepítettem. A Xampp egy szabad és nyílt forrású, platformfüggetlen webszerver-szoftvercsomag. Legfőbb alkotóelemei az Apache webszerver, a MariaDB adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői. A Xampp egyetlen tömörített állományba van csomagolva, telepítéséhez mindössze ezt az állományt kell letölteni és futtatni. A telepítés elvégzi az alapbeállításokat, ezután a rendszer készen áll a webszerver és a mintaalkalmazások futtatására. A Xampp-csomagot rendszeresen frissítik, így az mindig az Apache, MariaDB, PHP és Perl legújabb változatát tartalmazza, valamint más kiegészítőket is, mint például a phpMyAdmin. Az adatbázisom létrehozásához ezért a phpMyAdmin programot alkalmaztam, ami egy SQL alapú, relációs adatbázis-kezelő program. (forrás: <https://hu.wikipedia.org/wiki/XAMPP>).

A Xampp elindításakor az Apache és MySQL modulokat elég elindítani a start gomb megnyomásával, majd a böngészőbe a <http://localhost/phpmyadmin/> elérési út beütésével lehet az adatbázishoz kapcsolódni.

## Adatmodell leírása

Az adatbázisomat a phpMyAdmin felületen hoztam létre. Külön adatbázis szolgál a belépéshez, „belepes” néven és külön adatbázisban tároltam a programhoz szükséges minden egyéb táblázatot „szervizkeszletezorendszer” néven.

A kapcsolódást a következő kódokkal oldottam meg:

```
static string MySqlConnectionString = "Datasource=localhost;  
SslMode port=3306; username=figlerr; password=Napsugar;  
database= belepes; =none";  
MySqlConnection con= new MySqlConnection(MySqlConnectionString);  
  
static string MySqlConnectionString = "Datasource=localhost;  
port=3306; username=figlerr; password=Napsugar;  
database=szervizkeszletezorendszer ; SslMode=none";  
MySqlConnection con= new MySqlConnection(MySqlConnectionString);
```

Az adatbázis kapcsolatot ezután a `con.Open()` paranccsal nyitom meg, a `con.Close()` parancs pedig megszünteti a kapcsolatot az adatbázissal. Ez a művelet mindenféleképpen nagyon lényeges mozzanat, mivel ha nem szakítjuk meg a kapcsolatot a helytelen működés nem lesz kizárva.

A kapcsolódáshoz nélkülözhetetlenek a következő adatok:

- `datasource = localhost;`
- `port = 3306;`
- `username = figlerr;`
- `password = Napsugar;`
- `database = belepes/szervizkeszletezorendszer;`
- `SslMode = none;`

A felhasználónevet és jelszót a phpMyAdmin felületen a felhasználói fiókok menüpontban hoztam létre, majd a jogosultságok szerkesztése menüponton belül az adatbázis fülön hozzáadtam a programom adatbázisait.

A táblákat a program telepítésekor létre kell hozni. A „szervizkeszletezorendszer” adatbázis hat darab táblát tartalmaz. Ezek fontosabb jellemzői és mezői a következők:

- „telefontipusok” tábla: ez tartalmazza az összes telefontípust, amivel dolgoznunk kell, illetve egy-egy típushoz tartozóan minden fontos adatot.

| Oszlop    | Típus       |
|-----------|-------------|
| ID        | int(11)     |
| tipus     | varchar(20) |
| elolapkod | varchar(20) |
| kamerakod | varchar(20) |
| hatlapkod | varchar(20) |
| akku      | varchar(20) |
| fehertok  | varchar(20) |
| feketetok | varchar(20) |
| pinktok   | varchar(20) |
| folia     | varchar(20) |

- „adatlapok” tábla: ez tartalmazza a javításra beérkezett készülékek és tulajdonosaik adatait, a szervizelés kimenetelét, az esetleges javítás eredményét és a felhasznált alkatrészeket.

| Oszlop               | Típus         |
|----------------------|---------------|
| adatlaponosito       | int(11)       |
| nev                  | varchar(200)  |
| cim                  | varchar(200)  |
| telefonszam          | int(11)       |
| email                | varchar(100)  |
| tipus                | varchar(20)   |
| imei1                | varchar(9)    |
| hiba                 | varchar(1000) |
| garancialisfizetos   | varchar(12)   |
| datum                | date          |
| felhasznaltalkatresz | varchar(20)   |
| felhasznaltfodarab   | varchar(20)   |
| javitasieredmeny     | varchar(20)   |
| sablonszoveg         | varchar(1000) |
| javitasdatuma        | date          |

- „alkatreszek” tábla: Ez tartalmaz az alkatrészekkel kapcsolatban minden fontos információt.

| Oszlop             | Típus       |
|--------------------|-------------|
| sorszam            | int(11)     |
| azonosito          | int(9)      |
| tipus              | varchar(20) |
| alkatreszkod       | varchar(20) |
| beszerzesiar       | int(11)     |
| beszerzesdatuma    | date        |
| felhasznalasdatuma | date        |
| statusz            | varchar(20) |
| felhasznalas       | varchar(20) |
| selejt             | date        |



- „fodarabok” tábla: A fődarabokkal kapcsolatos információkat tárolom ebben a táblában.

| Oszlop             | Típus       |
|--------------------|-------------|
| sorszam            | int(11)     |
| tipus              | varchar(20) |
| imei2              | varchar(9)  |
| beszerzesiar       | int(11)     |
| beszerzesdatuma    | date        |
| felhasznalasdatuma | date        |
| statusz            | varchar(20) |
| felhasznalas       | varchar(20) |
| selejt             | date        |

- „tartozekok” tábla: Az eladásra szánt tartozékok adatait tárolom ebben a táblában. A beszerzési és eladási árakat, illetve a készletezéshez kapcsolódó információkat.

| Oszlop            | Típus        |
|-------------------|--------------|
| sorszam           | int(11)      |
| megnevezes        | varchar(200) |
| tartozekazonosito | varchar(9)   |
| beszerzesiar      | int(11)      |
| eladasiar         | int(11)      |
| beszerzesdatuma   | date         |
| statusz           | varchar(20)  |
| eladasdatuma      | date         |
| selejt            | date         |

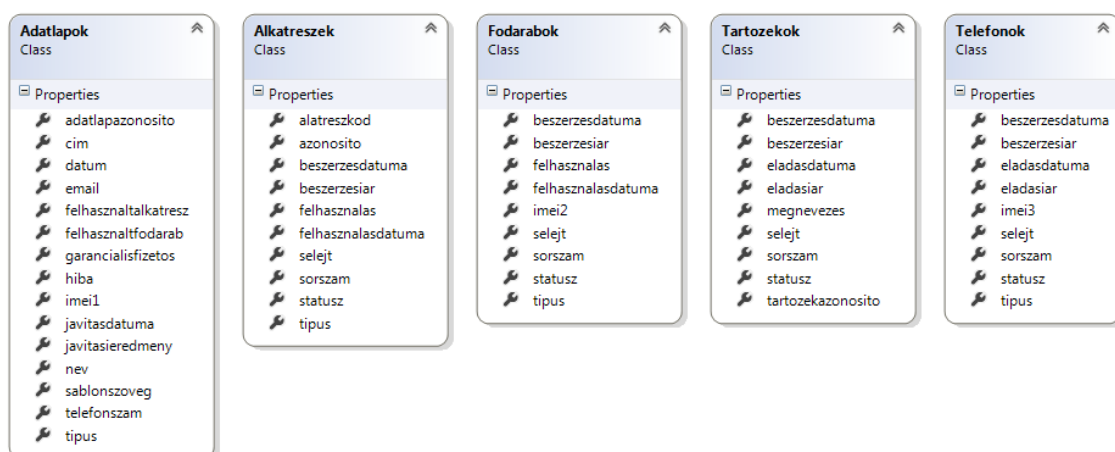
- „telefonok” tábla: Az eladásra szánt telefonok adatait tárolom ebben a táblában. A beszerzési és eladási árakat, illetve a készletezéshez, azonosításhoz kapcsolódó információkat.

| Oszlop         | Típus       |
|----------------|-------------|
| sorszam        | int(11)     |
| tipus          | varchar(20) |
| imei3          | varchar(9)  |
| beszerzesiar   | int(11)     |
| eladasiar      | int(11)     |
| beszerzesdatum | date        |
| statusz        | varchar(20) |
| eladasdatum    | date        |
| selejt         | date        |

## Részletes feladat-specifikáció, forráskód

A program tervezésekor az elsődleges célom az volt, hogy a tanultakat alkalmazva egy minél egyszerűbben kezelhető, átláthatóbb alkalmazást hozzak létre, olyan kezelőfelülettel, amelynek a használata nem igényel informatikai előismereteket.

Szerkezet: Az adatbázis táblázataira alapozva öt darab osztályt hoztam létre. Szükség szerint ezek segítségével a kiolvasott adatok eltárolhatóak és a keresett elem egyszerűen elérhető. A program jelen fázisában alap funkciókkal működik, így ez listákkal is egyszerűen megoldható, de a későbbiekben a tovább fejlesztés során az osztályok használata célszerűbb.

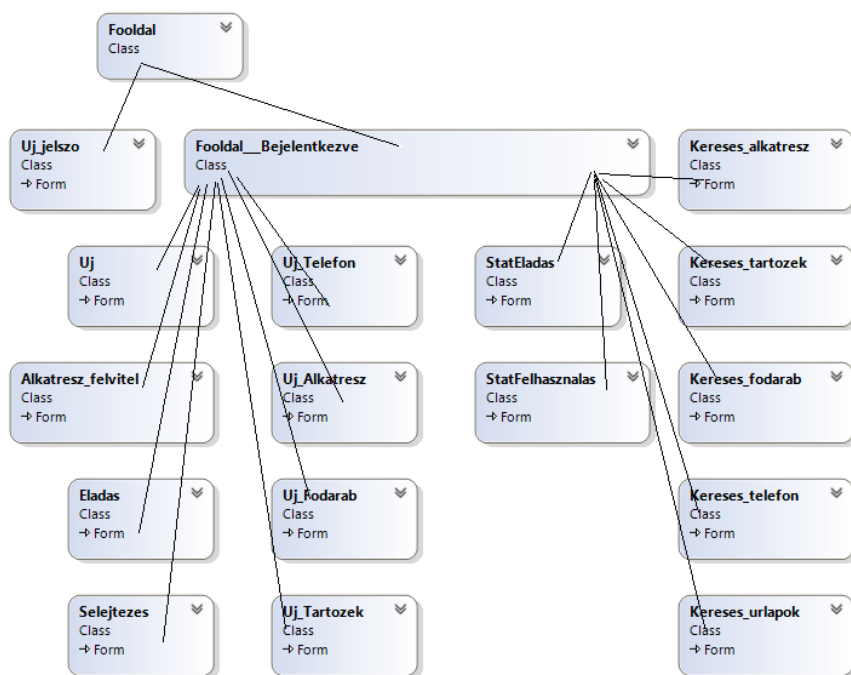


A program főbb részei a következők:

1. Bejelentkezés modul: A program csak bejelentkezés után használható. A bejelentkezést úgy oldottam meg, hogy az adatbázis létrehozásakor előre definiált

felhasználókat adunk meg, egy hozzá tartozó jelszóval. Ezt biztonsági okokból terveztem így. A programból a felhasználók nem módosíthatók, törölhetők, nem hozhatók létre újak sem. A jelszót viszont lehet változtatni, a régi jelszót és a felhasználónevet viszont tudni kell hozzá. Az eltárolt jelszavakat titkosítottam, erre az MD5 kódolást használtam. Az MD5 kódolás bármilyen adatból – függetlenül a méretétől, vagy a típusától – egy 16 bájt hosszú bináris hasht eredményez (32 hexadecimális számjeggyel írható le). A kódolás egyirányú, így nem lehet visszafejteni. Éppen emiatt alkalmas eredetiség ellenőrzésre. A gyakorlatban a hasht-t nem magából a jelszóból számoljuk ki, hanem egy véletlenszerű adat és a jelszó összekapcsolásával előállított stringre, hogy megnehezítsük a szótáralapú támadásokat. (forrás: <https://hu.wikipedia.org/wiki/MD5>). A programomban ezt a MD5CryptoServiceProvider és TripleDesCryptoServiceProvider osztályok használatával valósítottam meg.

2. Menürendszer: Belépés után érhető csak el a főoldalon. Az egyes menüpontokhoz tartozó adatlapok új ablakban nyílnak meg. Az űrlapok és eladás menüpontok viszont nem csak a főoldalról, hanem minden oldalról elérhetőek, mivel ezeket kell leggyakrabban használni. Fentieket a MenuStrip és ToolStrip eszközök segítségével valósítottam meg. A programot összesen 18 form segítségével valósítottam meg. Ezeket szemlélteti az alábbi ábra:



3. Űrlapok (új és alkatrész felvitel): A két menüpont alatt az „adatlapok” táblába tudunk rekordokat létrehozni. A kódban figyeltem rá, hogy a felhasználó minden kötelező mezőt kitöltsön és megfelelő adatokkal, ellenkező esetben hibaüzenetet kap. Ehhez a telefonszám mezőnél például a `MaskedTextBox` eszközt használtam. A `MaskedTextBox` módosított változata a szövegdobozoknak. Segítségükkel előre meghatározott formátumnak megfelelően adhatjuk meg az adatokat a program számára. A `MaskedTextBox` tulajdonságértéke tartalmazza, hogy milyen karaktereket, és azokat opcionálisan vagy kötelező jelleggel kell-e megadni. (forrás: Achs Ágnes–Szendrői Etelka Programozás 2. II. kötet Windows form alkalmazások). A következő, alkatrész felvitel menüpontban csak a kezeletlen adatlapok jelennek meg, amit az adatlapok táblából olvasunk be. A feltöltés gombbal előre létrehozott sablon szövegeket tudunk megjeleníteni. A felhasznált fődarab és alkatrész menüpontban a készleten lévő eszközöket látjuk, amely adatokat a „fodarabok” és „alkatreszek” táblákból vesz a program. A mentés gombbal nem csak az „adatlapok” táblába kerülnek új információk, hanem a kiválasztott alkatrészek és fődarabok státusza is változik a megfelelő táblákban, illetve bekerül a felhasználás dátuma is. Ezeket nagyrészt az „update” sql utasítás segítségével valósítottam meg.
4. Készletek felvitel és selejtezés: A „telefontípusok” nevű tábla értékeit olvastam be a felviteli felületekre. Ezek határozzák meg, hogy milyen készletet tudunk regisztrálni. Például az új/telefon menüpontban a választható telefonok típusát adja meg. A kitöltött űrlap adatait az „insert into” sql paranccsal tárolom el a megfelelő készlet táblában, az új telefon adatait például a „telefonok” táblában. A selejtezés felületen, a készleten lévő állományt olvassa be a program az adatbázisból a kiválasztott készlettípusonként, majd a selejtezendőre rányomva a megfelelő táblában a státusz selejtre vált, és a selejtezés dátumát is tárolom.
5. Eladás modul: Ez a modul arra szolgál, hogy az eladott telefonok és tartozékok regisztrálva legyenek és lekerüljenek készletről. A megnevezéshez az oldalra lépéskor egyből betöltődnek a „telefontípusok” táblából a megfelelő típusok, amiből majd ki lehet választani, mit akarunk eladni a rendszerből. Ugyanígy a tartozékok esetében is A készleten mezőnél a megnevezésnél kiválasztott típusból elérhető készletek jelennek meg, melyet a `CheckedListBox` elemmel és annak `ItemCheck` eseményével oldottam meg. A létrejövő listából a jelölőnégyzetbe

tett pipával lehet bejelölni, amelyik elemet tényleg el akarjuk adni a rendszerből. A kijelölt készlet be is kerül az eladás készletről mezőbe. A megnevezés mezőnél több típust is kiválaszthatunk, és a készleten mezőnél is többet kijelölhetünk. Az így kijelölt elemek eladási árát egy listában gyűjtöm, külön a készülékeket és külön a telefonokat. Ezek segítségével a fizetendő ár egy összegzés tételével számolható. Az adatokat az alábbi sql lekérdezéssel választom ki, melyben paraméteres értékadást használok:

```
cmd.CommandText = "select eladasiar from telefonok where  
imei3=@imei3";  
cmd.Parameters.AddWithValue("@imei3",  
Eladasmegnevezes6.SelectedItem);
```

Az alábbi kód a telefonok esetében példa a fizetendő ár kiszámolására.

```
private void Eladasfizetendo1_Click(object sender, EventArgs e)  
{  
    osszeg1 = 0;  
    osszeg2 = 0;  
    for (int i = 0; i < telefon.Count; i++)  
    {  
        osszeg1 = osszeg1 + telefon[i];  
    }  
    for (int i = 0; i < tartozek.Count; i++)  
    {  
        osszeg2 = osszeg2 + tartozek[i];  
    }  
    fizetendo = osszeg1 + osszeg2;  
    Eladasfizetendo2.Text = (fizetendo + "");  
}
```

A mentés gombra kattintva, a felvitt adatokat el tudjuk menteni txt, vagy csv állományba. Jelenleg csak az ügyféladatokat, a fizetendő összeg és a keltezés mentésére van szükség, így ezeket menti a program a következő kóddal:

```
private void Eladaskesz_Click(object sender, EventArgs e)  
{  
    string[] kiiras = new string[5];  
    kiiras[0] = "Név: ;" + Eladasnev2.Text + " ;"; ;
```

```

kiiras[1] = "Cím: ;" + Eladascim2.Text + ";";
kiiras[2] = "Fizetendő: ;" + Eladasfizetendo2.Text + "Ft;";
kiiras[3] = "Dátum: ;" + Eladasdatum2.Text + ";";
kiiras[4] = Environment.NewLine;
if (saveFileDialog1.ShowDialog() == DialogResult.OK)
{
    File.AppendAllLines(saveFileDialog1.FileName, kiiras,
    Encoding.Default);
}
}

```

6. Statisztikák modul: Ezeken a felületeken a négy készlettípus felhasználásáról tudunk információkat szerezni. Lekérdezhetjük a felhasznált darabszámot, amelyet összesen is kiír a program és ki tudjuk választani, hogy a garanciális vagy fizetős felhasználás érdekel minket az alkatrészek és fődarabok esetében. A következő gombbal a felhasználást forintban is lekérdezhetjük, a telefonok és tartozékok esetében külön a beszerzési és külön az eladási áron. Végül az utolsó funkcióval a haszon kérdezhető le. A megfelelő időintervallumot naptár segítségével választhatjuk ki, melyhez a DateTimePicker beépített lehetőséget használtam. A statisztikák létrehozásához adatbázis lekérdezésekkel fértem hozzá a megfelelő adatokhoz, amiket listában tároltam, majd az összegzés tételét használtam. A haszon kiszámolásánál az alkatrészek és fődarabok esetén a következő kódokat használtam:

```

private void Stathaszon1_Click(object sender, EventArgs e)
{
    double haszonfiz1 = (osszegfiz1 * 0.1);
    double haszongar1 = (osszeggar1 * 0.05);
}

```

Fizetős javítás esetén 10% a haszon, míg garanciális javítás esetén 5% haszonnal kell számolni. Az osszegfiz1 és osszeggar1 a felhasznált készletek beszerzési áron való összegzései. A telefonok és tartozékok esetében az eladási árak összegeiből kell kivonni a beszerzési árak összegeit.

7. Keresés modul: A készleten lévő elemekre tudunk szűrni típus szerint, illetve az űrlapokra imei szerint. Az eredményt minden esetben, táblázatos formában a

DataGridView eszközzel jelenítettem meg, miután kiolvastam az adatbázisból. Ezt szemlélteti a következő minta kód:

```
MySqlCommand cmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "select nev, cim, tipus, imeil, hiba,
    garancialisfizetos, felhasznaltalkatresz,
    felhasznaltfodarab,javitasieredmeny, sablonszoveg from
    adatlapok where adatlapazonosito=@adatlapazonosito";
cmd.Parameters.AddWithValue("@adatlapazonosito",
    Keresesazonosito2.SelectedItem);
DataTable dt = new DataTable();
MySqlDataAdapter da = new MySqlDataAdapter(cmd);
da.Fill(dt);
Kereseseredmenylistal.DataSource = dt;
```

A program egészében igyekeztem figyelni az egységes kinézetre. Ezért a listákat egységesen ComboBox és ListBox segítségével oldottam meg. A színezés és a feliratozás is egységes az alkalmazás minden formján. Az egyes modulok az átláthatóság és egyszerűbb kezelhetőség érdekében új ablakban nyílnak meg.

## Tesztelési dokumentáció

A szoftverfejlesztés fontos része a tesztelés. Ilyenkor ellenőrzött körülmények között, minta adatokkal figyeljük a program futását és a kapott eredményeket vizsgáljuk. A tesztelés célja, hogy a hibás működést észre vegyünk és megoldjuk. A tesztelésnek alapvetően kétféle módja lehet: fekete doboz (black box) vagy átlátszó doboz (clear box) megközelítés. A black box mód esetén nem foglalkozunk a belső objektumokkal, csak a kinyert adat értékeket vizsgáljuk, azaz azt teszteljük, egy megadott kérdésre az elvárt válasz jön-e vagy sem. Ez ugyan egyszerűbb tesztelési eszközöket és ismeretek kíván, viszont rendszerint nagyobb időköltséggel jár és kevesebb belső részlet fedhető le vele. Ennek oka, hogy a tesztesetek nem fedik le a teljes állapotteret, összetett bemenő adatok esetén nem tér ki a teszt minden lehetséges esetre. A clear box megközelítésben a program kódját is elérjük. Mivel ekkor az egyes egységek tesztelésénél jelentősen kisebb állapottérrel lehet dolgozni, nagyobb az esélye, hogy lefedettebb lesz a tesztelés is. Ezen módszer az egyes objektumok kódolásnak és működésének alapos ismeretét igényli.

(forrás:

[https://www.tankonyvtar.hu/hu/tartalom/tamop425/0046\\_szoftverteszteles/ch10.html](https://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_szoftverteszteles/ch10.html)).

A program írásakor folyamatosan teszteltem a létrehozott függvényeket, illetve az alkalmazás funkcióit is. A végleges tesztelés során a tényleges működésre voltam kíváncsi, vagyis, hogy az egyes funkciók jól működnek-e, jó eredményeket kapok-e, illetve ha a felhasználó valamit nem, vagy rosszul tölt ki, hibaüzenetet kap-e, ahogy terveztem. Ez a fekete doboz tesztelésnek felel meg, mert ilyenkor a felhasználó szemszögéből vizsgáljuk a program működését és összevetjük a specifikációban leírtakkal. A bevitt adatokat és a várt eredményeket ismerjük, hogy hogyan lett kódolva, azt ennél a módszernél nem vesszük figyelembe.

Fiktív adatokat használtam a teszteléshez. A fejlesztés közben használt adatbázisból a teszt adatokat letöröltem és próbáltam abban a sorrendben haladni, ahogy a programot is valószínűsíthetően használni fogják, így elsőre a belépéssel és jelszómódosítással kezdtem, és mind helyes, mind hibás adatokkal megfigyeltem a működést. Az űrlapokon is ellenőriztem az adatbevitelt jó és rossz adatokkal egyaránt, hogy ha például rossz karaktert üt be a felhasználó, vagy nem választ ki semmit, valóban kap-e hibaüzenetet. Megvizsgáltam az egyes funkciókhoz tartozó gombokat is, elvégzik-e a feladatukat, jó eredményt adnak-e.

A tesztelési eredményeimet a lenti táblázatban foglaltam össze:

| Teszt eset   | Elvárás                      | Kapott érték              | Végeredmény       |
|--|------------------------------|---------------------------|-------------------|
| <b>Bejelentkezés</b>                                 | sikeres belépés              | sikeres belépés           | Elfogadva         |
| <b>Jelszó módosítása</b>                             | sikeres módosítás            | sikeres módosítás         | Elfogadva         |
| <b>Bejelentkezés (téves adatokkal)</b>               | hibaüzenet, felugró ablak    | hibaüzenet, felugró ablak | Elfogadva         |
| <b>Adatbázis kapcsolat</b>                           | létrejön                     | létrejött                 | Elfogadva         |
| <b>Adatbevitel vizsgálata</b>                        | csak helyes karaktert enged  | szóközt is enged          | jövőben javítandó |
| <b>Űrlap/Eladás gyorsmenü minden oldalon működik</b> | működik                      | működik                   | Elfogadva         |
| <b>Űrlapok/Új/Mentés</b>                             | adatok adatbázisba mentődnek | működik                   | Elfogadva         |



|   |  |   |  |
|---|--|---|--|
| <b>Űrlapok/Alkatrész felvitel/Mentés</b>                    | „adatlapok” tábla megfelelő sora módosul     | működik   | Elfogadva                                      |
| <b>Készletek/Új</b>   | megfelelő táblába bekerülne az adatok        | működik   | Elfogadva                                      |
| <b>Selejtezés</b>   | a megfelelő táblában a státusz selejtre vált | státusz sikeresen módosul                             | Elfogadva                                      |
| <b>Eladás/Fizetendő</b>                                     | helyes összeg                                | helyes összeg   | Elfogadva                                      |
| <b>Eladás/Eladás készletről</b>                             | megfelelő táblában a státusz eladottra vált  | sikeres státuszváltás                                 |  |
| <b>Eladás/Kész</b>  | eladási adatok txt/csv-be mentődnek          | első alkalommal a választott formátumot meg kell adni | jövőben javítandó (legyen egy alapértelmezett) |
| <b>Statisztikák</b>   | helyes értékek                               | helyes értékek  | Elfogadva                                      |
| <b>Keresés</b>  | működik                                      | működik   | Elfogadva                                      |
| <b>Kilépés</b>  | teljes alkalmazást bezárja                   | teljes alkalmazást bezárja                            | Elfogadva                                      |
| <b>Főoldal-Bejelentkező: jobb felső bezárás ikon</b>        | teljes alkalmazást bezárja                   | csak az adott oldalt zárja be                         | jövőben javítandó                              |
| <b>Főoldal bejelentkezés előtt: jobb felső bezárás ikon</b> | teljes alkalmazást bezárja                   | teljes alkalmazást bezárja                            | Elfogadva                                      |

## Továbbfejlesztési lehetőségek

A program jelenleg a legfontosabb alap funkciókat tartalmazza. A későbbiekben több új funkcióval bővíthető, illetve a most meglévők is módosíthatók szükség szerint. Az alábbi fejlesztéseket lenne célszerű megvalósítani:

- A bejelentkezési felületen egy regisztrációs felületet lehetne kialakítani, hogy amennyiben szükség van rá, új felhasználót lehessen felvinni. Ehhez két jogosultsági szintet is létre kell hozni, hogy csak az illetékes személy adhasson hozzá a rendszerhez újabb felhasználót. Jelenleg az adatbázis létrehozásakor kell

abba közvetlenül felvinni a felhasználókat, programból ez nem módosítható, csak a jelszó változtatható biztonsági okokból.

- A készletről eladásokat jelenleg txt állományban tároljuk, mivel ezt csak a napi záraskori eltérések ellenőrzésére használjuk, de ezeket az adatokat a későbbiekben lehetne pdf-ben, formázva tárolni és nyomtatási lehetőséget is biztosítani hozzá.
- A statisztika menüpontban a felhasználások alatt ki lehetne bővíteni a lekérdezést a selejt készletekre is. Illetve dátumra szűrés mellett meg lehetne valósítani a típusra szűrést is. Az eladás alatt ugyanígy.
- A keresés menük is szükség szerint bővíthetők újabb szempontok alapján, illetve a készleten lévők helyett a már felhasznált, vagy selejt eszközök között is meg lehetne oldani a keresést a későbbiekben.
- Az űrlapokat szükség szerint lehet bővíteni új mezőkkel, dokumentum scannelési lehetőséggel.

## Összefoglalás

A szakdolgozatom során olyan programot próbáltam létrehozni, amellyel a tanult fejlesztői környezeteket jobban megismerhettem, azokban elmélyülhettem. Az alkalmazás fejlesztése során sok új ismerettel is gazdagodtam, illetve betekintést nyerhettem a gyakorlatban is a szoftverfejlesztésbe.

A létrehozott szerviz rendszer egy kisebb üzletet képes kiszolgálni. A legfontosabb funkciók kidolgozására volt időm a szakdolgozat alatt, de a további fejlesztésekkel jó alapja lehet egy komplex szervizkészletező rendszer kialakításának.

## Irodalomjegyzék

- <https://hu.wikipedia.org/wiki/XAMPP>
- [https://hu.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://hu.wikipedia.org/wiki/Microsoft_Visual_Studio)
- <https://dev.mysql.com/>
- [https://www.tankonyvtar.hu/hu/tartalom/tamop425/0046\\_szoftvertesztes/ch01s02.html](https://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_szoftvertesztes/ch01s02.html)
- [https://www.tankonyvtar.hu/hu/tartalom/tamop425/0046\\_szoftvertesztes/ch10.html](https://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_szoftvertesztes/ch10.html)
- Reiter István: C# programozás lépésről-lépésre, Jedlik Oktatási Stúdió, Budapest, 2012
- Illés Zoltán: Programozás C# nyelven, Jedlik Oktatási Stúdió, Budapest, 2005
- <https://www.youtube.com/watch?v=P7aaaLvWys&t=0s&index=12&list=WL>
- <https://www.youtube.com/watch?v=qPAvz8Jdi0Q>
- <https://www.youtube.com/watch?v=ysxC6-AFEYg>
- <https://hu.wikipedia.org/wiki/MD5>
- Achs Ágnes – Szendrői Etelka: Programozás 2., II. kötet Windows form alkalmazások, Pécsi Tudományegyetem Műszaki és Informatikai Kar, Pécs, 2015