

Záródolgozat

Fidrus Márk

Miskolc

2021

Tartalomjegyzék

1.Bevezetés.....	3
1.1. Miről szól a záródolgozat (téma).....	3
2. Fejlesztői környezet.....	4
2.1. Visual Studio (C# windows form).....	4
2.1.1. Mi is az Visual Studio és mi szerepe van a programban.....	4
2.2. MySQL, phpmyadmin, XAMPP.....	10
2.2.1. Mi is az a MySQL és mi szerep van a programban.....	10
2.2.2. Mi is az a PhpMyadmin és mi szerepe van a programban.....	10
2.2.3. Mi is az a XAMPP és mi szerepe van a programban.....	11
2.2.3.1. Apache, MariaDB, PHP, Perl.....	11
3. Program ismertetése és bemutatása	14
3.1. Program rövid bemutatása.....	14
3.2. Formok,.....	14
3.3. Hogyan működik részletezés (felhasználói és fejlesztői perspektívából).....	15
3.3.1. Bejelentkezés (az alap programba).....	15
3.3.2. Főoldal ismertetése és kezelése.....	18
3.3.3. Szerviz menüpont bemutatása.....	19
3.3.4. Ügyfél menüpont bemutatása.....	21
3.3.5. Ügyfélre vár menüpont bemutatása.....	22
3.3.6. Szerviz esemény bemutatása.....	24
3.3.7. Átadva menüpont bemutatása.....	25
3.3.8. Számlára! menüpont bemutatása.....	27

3.3.9. Nyomtatás menüpont bemutatása.....	29
3.3.10. Infó menüpont bemutatása.....	31
3.3.11. Profil menüpont bemutatása.....	31
3.3.12. Almenüpontok bemutatása.....	32
3.3.8.1. Új hozzáadás.....	32
3.3.8.2. Szerkesztés, törlés menüpont bemutatása	34
3.3.8.4. Aktív, inaktív menüpont bemutatása.....	35
3.3.8.5. Kész! menüpont bemutatása.....	35
4. Tovább fejlesztési ötletek.....	37
4.1. Egyedi faviconok.....	37
4.2. Mobil applikáció.....	37
4.3. Raktár program hozzákapcsolása.....	38
5. Források.....	39

Bevezetés

Záródolgozatom témája a szerviz és karbantartó program(ok). A szerviz és karbantartó program(ok)on belül pedig a mobil szerviz és karbantartó program.

Mi is az a szerviz, karbantartás menedzsment program. A program különféle eszközök (berendezések, gépek, járművek, ingatlanok stb.) adatainak nyilvántartását, karbantartásának tervezését, a tervteljesítés nyomon követését, nyilvántartását látja el. A rögzített adatok alapján ellenőrizhetőek a tervezett karbantartások, az eszközökről elektronikus karbantartási napló készülhet, valamint költség és időkimutatás készíthető a tervezett és a tény költségekről, időkről, amikkel a minőségbiztosítási igények is teljesíthetőek. Maga a szerviz és karbantartó programokat nagyon széles körben lehet használni. A program funkciói erősen függenek, hogy mit szeretnénk a program segítségével rendszerezni, nyilvántartani és kezelni. A program nagyon erős előnye a mai gyorsan fejlődő világban és az IT (information technology) világában is egyaránt, hogy a feleslegesen kinyomtatott és rendszerezett papírok helyett egy adott programmal elvégezhető minden könnyen, gyorsan és adott helyen. Amennyiben viszont szükséges kiállítani bizonylatot akkor is csak egy vagy adott esetben két (egyét az ügyfélnek és egyet a cégnek) lapot kell kinyomtatni.

Fejlesztői környezet

2.1. Visual Studio (C# windows form):

Mi is az a Visual Studio ? A **Visual Studio** a Microsoft több programozási nyelvet tartalmazó fejlesztőkörnyezete, amely az évek során egyre több új programnyelvvvel bővült. Jelenleg a F#, C++, C# és Visual Basic programozási nyelveket, valamint az XML-t támogatja. A csomag része a még a MASM (Microsoft Macro Assembler) is, ami részleges assembly támogatást biztosít.

A **C#** (ejtsd: szí-sárp) a Microsoft által a .NET keretrendszer részeként kifejlesztett objektumorientált programozási nyelv. A nyelv alapjául a C++ és a Java szolgált.

Története: A 90-es években a Microsoft a saját Java keretkörnyezetét a saját operációsrendszer-specifikus függvényeivel és szolgáltatásaival bővítette ki, amire viszont nem volt engedélye (a Java licencét birtokoló) Sun Microsystemstől. A Microsoft-féle Javára fejlesztett alkalmazások nem lettek volna futtathatók más rendszereken, ami ellenkezik a Java platform-függetlenségre vonatkozó alapelveivel. A Sun végül beperelte a Microsoftot, és ezt követően a Microsoft, és ezt követően a Microsoft a Java eltávolítására kényszerült a Windows rendszerekből (természetesen a Java-t a továbbiakban is lehet külön rendszerként telepíteni). A Microsoft egy saját keretrendszer fejlesztésébe kezdett. Ez lett a .NET (dotnet), és ehhez a keretrendszerhez adták ki programnyelvként a C# első verzióját. A későbbiekben ezt a nyelvet úgy fejlesztették tovább, hogy meglegyen az egyensúly a programozó nyelvi szabadsága és a gyors alkalmazásfejlesztés lehetősége között. Rengeteg hivatalosan kiadott bővítmény létezik hozzá, amelyek a kódolást könnyebbé teszik.

A C# fejlesztését Anders Hejlsberg vezette, aki a Turbo Pascal tervezője is volt. A .NET osztálykönyvtárait a programozók eredetileg egy sajátos, ún. SMC (Simple Managed C) nyelven írták. 1999-ben azonban Hejlsberg (aki addig a J++ nyelven dolgozott) vezetésével egy csapat alakult egy új programozási nyelv létrehozására, akkoriban COOL (C-like Object Oriented Language, magyarul: C-szerű Objektum Orientált Nyelv) fedőnév alatt. A cél egy olyan nyelv megalkotása volt, mellyel teljesen objektumorientált módon lehet a keretrendszer komponenseit fejleszteni. Egy évvel később, 2000 júliusában a PDC konferencián a .NET-tel, és az ASP.NET-el együtt a C#-ot is bemutatták, immár ezen névvel, hogy az esetlegesen szerzői jog problémákat megelőzzék. Bár sok bírálat érte megjelenésekor, főleg a Java-s

táborból, miszerint a Microsoft pusztán legyártotta a saját Java verzióját, Anders azt nyilatkozta, hogy a nyelvet sokkal inkább a C++-hoz szerették volna közelíteni.

A Microsoft benyújtotta a C# nyelvi specifikációját az ECMA-hoz formális szabványosításra. 2001. decemberében az ECMA kiadta az ECMA-334 C# Language Specification szabványt, 2003-ban pedig ISO szabvány lett (ISO/IEC 23270). Több független megvalósítás is folyamatban van, többek között:

- Mono, a Ximian nyílt forráskódú .NET implementációja.
- dotGNU és a Portable .NET a Free Software Foundationtól.

Bár a Mono Project egy nagyrészt kompatibilis nyílt forráskódú C# fordítót állított elő (lehetővé téve az ilyen kódok futtatását Linux és Macintosh rendszereken), a nyelv Windows operációs rendszereken kívüli használata nem terjedt el, mivel az osztálykönyvtárakat szolgáltató .NET Framework portolása más rendszerek alá még kezdetleges stádiumban van. Ugyanakkor a Mono az utóbbi években rohamos fejlődésnek indult, amit elősegített a Microsoft és a Novell (a Mono „anyacége”) együttműködése a Microsoft Silverlight technológiájának Linux illetve Macintosh rendszerekkel kompatibilis változatának a Moonlightnak a készítésekor.

A C# az a programozási nyelv, ami a legközvetlenebb módon tükrözi az alatta működő, minden .NET programot futtató .NET keretrendszert, valamint erősen függ is attól. Nincsen nem menedzsel, natív módban futó C# program. A primitív adattípusai objektumok, a .NET típusok megfelelői. Szemétgyűjtést használ, valamint az absztrakcióinak többsége (osztályok, interfészek, delegáltak, kivételek...) a .NET futtatórendszert használja közvetlen módon.

A C vagy C++ nyelvhez hasonlítva a C# több korlátozást és továbbfejlesztést is tartalmaz. A lehetőségei közül néhány:

- A Mutatók és a nem ellenőrzött aritmetika csak egy speciális, nem biztonságos módban (unsafe mode) használható. A legtöbb objektum-hozzáférés csak biztonságos hivatkozásokon keresztül tehető meg, és az aritmetikai műveletek debug módban túlcsoordulás szempontjából ellenőrzöttek.
- Az objektumok nem szabadíthatók fel közvetlen módon, ehelyett a szemétgyűjtő szabadítja fel őket, mikor már nincs rájuk hivatkozás. Ez a módszer kizárja a nem létező objektumokra való hivatkozás lehetőségét.

- A destruktorkok (~) elérhetőek. A megfelelően megírt IDisposable interfész (Disposable programozási minta), aminek a lefutását garantálja using blokk, együtt kikényszerítheti az azonnali felszabadítást az osztályon belüli natív erőforrások esetében. A nem natív erőforrások felszabadítását ebben az esetben is a szemétygyűjtő (Garbage Collector) végzi. A finalizerek szintén rendelkezésre állnak, de nem váltanak ki azonnali felszabadítást. Finalizer a Dispose eljárás javasolt implementációs módozatában nem fut le, mivel az ajánlás szerint a szemétygyűjtőt utasítani kell a finalizer hívás kihagyására.
- A nyelv csak egyszeres öröklődést támogat, de egy osztály több interfészt is megvalósíthat
- A C# sokkal típus biztonságosabb, mint a C++. Az egyetlen implicit konverzió a biztonságos konverzió, úgy mint az egészek tágabb intervallumban konvertálása vagy a leszármazott osztályok alaposztályba konvertálása. Nincs implicit konverzió az egészek és a logikai típus (boolean) között, a felsorolás tagok és az egészek között. Nincsenek void mutatók (bár az Object osztályra mutató mutatók hasonlóak), valamint bármely, a felhasználó által definiált implicit konverziót explicit módon meg kell jelölni.
- A felsorolás adattagjai a saját névterükben helyezkednek el.
- A 2.0-s verziótól felfelé már rendelkezik a generikus programozás néhány eszközével.
- Tulajdonságok (Properties) használhatók, amelyek úgy tesznek lehetővé kód futtatást mezők beállításakor és olvasásakor, hogy közben az adattagok szintaxisát használja.

A Microsoft bár úgy fejlesztette ezt a nyelvet, mint a saját Java implementációját, bizonyos tulajdonságokban azonban eltérő szemléletmódot követ a két rendszer. Ilyen különbségek például:

- A java alapértelmezett láthatósága mindig csomagon (package) belül publikus, a C#-nál pedig mindig privát, a protected jelentése eltér, illetve támogatja az internal és a protected internal láthatóságot.
- A C# támogatja a parciális osztályokat (2.0-s verziótól felfelé), és az operátor-felüldefiniálást.
- A metódusok a Javában alaptól virtuálisak, ellentétben a C#-pal.

- A Java ellenben például lehetővé teszi a fordításidejű kivételkezelést.
- A Java támogatja az egzisztenciális generikusokat, míg a C# a ko- és kontravarianciát (4.0-s verziótól) a generikusoknál.
- A C# támogatja az azonos nevű, de eltérő verziójú szerelvények betöltését, ezzel szemben a Java-ban nincs ilyen támogatás.
- Java esetén nincsenek tulajdonságok, amelyek megkönnyítenék az objektumok belső változóinak az ellenőrzött lekérését és módosítását. Helyette ez a funkció Getter és Setter metódusokkal van megvalósítva.
- A C# támogatja a delegált metódusokat, amely hasonló funkciót valósít meg, mint a C++ esetén a függvény mutatók.
- A C# esetén van unsafe (nem biztonságos) kontextus, amelyben létezik mutató típus. (optimalizációt segíti)
- A C# esetén van unchecked (nem ellenőrzött) kontextus, amiben a változók túlsordulása és helyes típus konvertálása felülbírálnak (optimalizációt segíti)

Szintaxisban a két nyelv ugyanabba a nyelvcsaládba (C/C++ alapú nyelvek) tartozik, tehát mindkettőre a kapcsos zárójeles tagolás, sor végi pontosvessző a jellemző, ezért a kód nagyban hasonlít.

A legtöbb programozási nyelvtől eltérően a C# megvalósítások nem rendelkeznek önálló, eltérő osztály vagy függvénykönyvtárakkal. Ehelyett a C# szorosan kötődik a .NET keretrendszerhez, amittől a C# kapja a futtató osztályait és függvényeit. A .NET keretrendszer osztálykönyvtárát tartalmaz, ami a .NET nyelvekből felhasználható egyszerű feladatok (adatrepresentáció és szövegmanipuláció) végrehajtásától kezdve a bonyolult (dinamikus ASP.NET weblapok generálása, XML feldolgozás és reflexió) feladatokig. A kód névterekbe van rendezve, mely a hasonló funkciót ellátó osztályokat fogja össze. Például System.Drawing a grafikai, System.Collections az adatstruktúra és System.Windows.Forms a windows ablakos megjelenítésért felelős funkciókat fogja össze.

További rendezési szint az assembly (szerelvény). Egy assembly állhat egy fájlból, vagy több összelinkelt fájlból (az al.exe segítségével), ami több névteret és objektumot tartalmazhat. A különböző feladatokhoz szükséges osztályokat assemblyk (például System.Drawing.dll, System.Windows.Forms.dll) hivatkozásával vagy a központi könyvtár (mscorlib.dll a Microsoft megvalósításában) használatával érhetik el a programok.

A .NET keretrendszer támogatja azonos nevű, de eltérő assemblyk betöltését, amelyek verziózott implementációkat tartalmazhatnak, így könnyedén lehetővé válik egy alkalmazáson belül a visszafelé kompatibilitás eltérő implementációk betöltésével és példányosításával.

A **Windows Forms** egy ingyenes és nyílt forráskódú grafikus (GUI) osztálykönyvtár, amely a Microsoft .NET Framework vagy a Mono Framework részeként található, platformot kínál gazdag kliens alkalmazások írásához asztali, hordozható és táblagépekhez. Noha a korábbi és összetettebb C++ alapú Microsoft Foundation Class Library helyettesítőjeként tekintenek rá, nem kínál összehasonlítható paradigmát és csak a felhasználói felület szintjének platformjaként működik egy többszintű megoldásban.

A Microsoft Connect eseményen, 2018. december 4-én a Microsoft bejelentette, hogy a Windows Forms-t nyílt forráskódú projektként bocsátja ki a GitHubon. MIT licenc alatt jelenik meg. Ezzel a kiadással a Windows Forms elérhetővé vált a .NET Core keretrendszert célzó projektek számára. A keret azonban továbbra is csak a Windows platformon érhető el, és a Mono a Windows Forms hiányos megvalósítása marad az egyetlen platformokon átívelő megvalósítás.

Csakúgy, mint az Abstract Windows Toolkit (AWT), az egyenértékű Java API, a Windows Forms is egy korai és egyszerű módja annak, hogy grafikus felhasználói felület-összetevőket biztosítsunk a .NET-keretrendszerben. A Windows Forms a meglévő Windows API-ra épül, és egyes vezérlők csupán a mögöttes Windows-összetevőket burkolják. Egyes módszerek közvetlen hozzáférést tesznek lehetővé a Win32 visszahívásokhoz, amelyek nem Windows operációs rendszereken nem érhetők el. A .NET Framework 2.0-ban a Windows Forms gazdagabb elrendezés-vezérlőket, Office 2003 stílusú eszköztár-vezérlőket, többszörös összetevőt, gazdagabb tervezési és adatkötési támogatást, valamint a ClickOnce webalapú telepítést kapott. A .NET 3.0 kiadásával a Microsoft kiadott egy második, párhuzamos API-t a GUI-k megjelenítésére: DirectX-alapú Windows Presentation Foundation (WPF) valamint az XAML nevű GUI deklaratív nyelv.

A záródolgozathoz a C# és Windows formos fejlesztő környezetet és az eszközöket a Visual Studio 2019 segítségével sikerült összekötnöm az adatbázissal, elkészíteni a program jelenlegi kinézetét és maga a programot teljesen elkészíteni.

2.2. MySQL, phpmyadmin, XAMPP

Mi is az a MySQL ? A MySQL egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver. A szoftver eredeti fejlesztője a svéd MySQLAB cég, amely kettős licenccel tette elérhetővé a MySQL-t. Választható módon vagy a GPL szabad szoftver licenc, vagy egy zárt (tulajdonosi) licenc érvényes a felhasználásra. 2008 januárjában a Sun felvásárolta az Oracle Corporation, így a MySQL is az Oracle tulajdonába került. A MySQL az egyik legelterjedtebb adatbázis-kezelő, aminek egy oka lehet, hogy a teljesen nyílt forráskódú LAMP (Linux-Apache-MySQL-PHP) összeállítás részeként költséghatékony és egyszerűen beállítható megoldást ad dinamikus webhelyek szolgáltatására. Egyedi illesztőfelületekkel az adatbázis-kezelő elérhető C, C++, C#, Delphi, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python, Ruby és Tcl programozási nyelvvel. Egy MyODBC nevű ODBC interfész további, ODBC-t kezelő nyelvek számára is hozzáférhetővé teszi az adatbázis-kezelőt. A MySQL számára az ANSI C a natív nyelv. A MySQL adatbázisok adminisztrációjára a mellékelt parancssori eszközöket (mysql és mysqladmin) használhatjuk. A MySQL honlapjáról grafikus felületű adminisztráló eszközök is letölthetők: MySQL Administrator és MySQL Query Browser. Széles körben elterjedt és népszerű adminisztrációs eszköz a PHP nyelven írt, nyitott forráskódú phpMyAdmin. A phpMyBackupPro (amelyet szintén PHP-ban írták) adatbázisok (akár időzített, ismétlődő) mentésére szolgál eszközként. A MySQL különböző platformokon futtatható: AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, OpenVMS, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP és a Windows frissebb verziói. Már Mac OS X 10.4-re is elérhető. A MySQL az adatbázis (phpMyAdmin) és C# windows form-os környezet közötti kapcsolatot teszi lehetővé a szerviz programban.

Mi is az a phpMyAdmin ? A phpMyAdmin egy nyílt forrású eszköz, amit PHP-ban írtak a MySQL menedzselésére az interneten keresztül. Jelenleg képes készíteni és eldobni adatbázisokat, készíteni/eldobni/módosítani táblákat, törölni/módosítani/hozzáadni mezőket, SQL parancsokat futtatni és a mezőkön kulcsokat kezelni. Képes az egész MySQL szerver kezelésére (szuper-felhasználói igényel) épp úgy, mint egyetlen adatbázisra. Az utóbbi megvalósításához be kell állítani a MySQL felhasználót, hogy csak a kívánt adatbázist tudja írni/olvasni.

Tobias Ratschiller, akkoriban IT-tanácsadó, majd a Maguma szoftveres cég alapítója, 1998-ban kezdett el dolgozni egy PHP-alapú webes MySQL-kezelőfelületen, melyhez Peter Kuppelwieser MySQL-Webadmin című munkája adta az ötletet. 2000-ben időhiány miatt abbahagyta a projektet (a phpAdsNew hirdetőszervert is, melynek ugyancsak ő volt az eredeti szerzője).

A phpMyAdmin addigra az egyik legnépszerűbb PHP-alkalmazás és MySQL-kezelő eszköz lett, nagy felhasználói és közreműködői közösséggel. Sok Linux-disztribúció is tartalmazza. A növekvő számú javítások koordinálása céljából három fejlesztőből, Olivier Müller, Marc Delisle és Loic Chapeaux, álló csoport bejegyezte a The phpMyAdmin Project projektet a SourceForge.net-en, és 2001-ben átvette a fejlesztést.

A 2008. november 28-án megjelent 3.1.0-s verzió új telepítőt kapott, valamint BLOBstreaming és Swekey hardveres hitelesítés támogatással rendelkezik.

2002. decemberben a hónap projektje lett a SourceForge.net-en. Háromszor nyerte el a SourceForge.net Community Choice Awards díját.:

- 2006. A SysAdmin és az Adatbázis kategória nyertese.
- 2007. A rendszergazdák legjobb segédeszköze.
- 2008. Most Likely to Be the Next \$1B Acquisition.

A phpMyAdmin szerepe a programban a felvitt rögzíteni kívánt adatok rögzítése és tárolása és könnyed kezelése.

Mi is az XAMPP ? A XAMPP egy szabad és nyílt forrású platformfüggetlen webserverszoftvercsomag, amelynek legfőbb alkotó elemei az Apache webserverver, a MariaDB (korábban a MySQL) adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői (végrehajtó rendszerei). Ez a szoftvercsomag egy integrált rendszert alkot, amely webes alkalmazások készítését, tesztelését és futtatását célozza, és ehhez egy csomagban minden szükséges összetevőt tartalmaz. A rendszer egyik nagy előnye az összehangolt elemek könnyű telepíthetősége.

A XAMPP egy betűszó ill. rövidítés, betűi a következő kifejezésekből származnak:

- X – eredeti olvasatban az angol cross-platform szót helyettesíti, amely a platformfüggetlenséget jelenti
- A – Apache webserverver

- M – MariaDB adatbázis-kezelő (korábban MySQL)
- P – PHP szerveroldali szkriptnyelv
- P – Perl általános célú szkriptnyelv

Hivatalosan a XAMPP tervezői az eszközt egy fejlesztőrendszernek szánták, amellyel a web-tervezők és programozók internetes kapcsolat nélkül fejleszthetik és tesztelhetik alkalmazásaikat. Ennek érdekében több fontos biztonsági funkció alapértelmezésben ki van kapcsolva a csomagban, ennek ellenére a XAMPP szoftver valódi webes szolgáltatóként is használják. A csomag egy külön eszközt tartalmaz a legfontosabb részek jelszavas védelmének beállítására.

A XAMPP többféle adatbázis-kezelő használatát is támogatja, ilyenek pl. a MySQL és az SQLite és mások.

A XAMPP telepítése után a helyi gép (a localhost) hálózati gépként is hozzáférhetővé válik, pl. FTP kliensprogrammal elérhető. Alkalmazható pl. FileZilla fájlkezelő, telepíthető tartalomkezelő rendszerek, mint a WordPress vagy a Joomla!. A localhost közvetlenül is kezelhető egyes HTML vagy általános szövegszerkesztőkből, az FTP protokoll használatával.

Az alapbeállítás szerinti FTP felhasználónév a „newuser”, jelszava „wampp”. Az alapbeállítás szerinti MySQL felhasználónév a „root”, jelszó nélkül.

Mi is azaz Apache HTTP Server ? Az Apache http Server (röviden Apache) nyílt forráskódú webkiszolgáló alkalmazás, szabad szoftver, mely kulcsfontosságú szerepet játszott a World Wide Web elterjedésében. A projekt célja olyan webszerver program létrehozása, karbantartása és fejlesztése, amely megfelel a gyorsan változó Internet követelményeinek, biztonságos, üzleti, vállalati felhasználásra is megfelelő és szabadon használható. Az Apache a régi NCSA HTTPd szerverre épül, az Apache Szoftver Licenc feltételei alatt terjesztik.

Az Apache robosztus, erőteljes és rugalmas webszerver, amely kompatibilis a HTTP/1.1 (RFC2616) protokollal. Az Apache projekt koordinálását az Apache Software Foundation végzi. Néhány vezető és több száz fejlesztő van e projekt mögött.

Az Apache volt az első használható alternatíva a Netscape Communications Corporation webszerverrel szemben (melynek mai neve Sun Java System Web Server). A későbbiekben továbbfejlődött és más unix alapú webszerverekkel is felvette a versenyt funkcionalitást és teljesítmény tekintetében.

Többek között a következő operációs rendszerekhez készítették el az Apache-t: Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X és Microsoft Windows.

Mi is az a MariaDB ? A MariaDB egy többfelhasználós, tábszálú, SQL-alapú relációsadatbázis-kezelő szerver, egy MySQL-fork. Mivel kompatibilis a MySQL-lel, a MariaDB is lehet a LAMP adatbázisszervere.

Mi is az a PHP ? A PHP egy általános szerveroldali szkriptnyelv dinamikus weblapok készítésére. Az első szkriptnyelvek egyike, amely külső fájl használata helyett HTML oldalba ágyazható. A kódot a webszerver PHP feldolgozómodulja értelmezi, ezzel dinamikus weboldalakat hozva létre. Rasmus Lerdorf 1995-ben indította útjára. Ma a The PHP Group tartja fenn és fejleszti. A PHP szabad szoftver, de licence nincs csereszabatos a GNU licenccel, mivel megkötéseket tartalmaz a PHP név használatára.

A PHP születésekor csupán egy makrókészlet volt személyes honlapok karbantartására. Innen jön az eredeti név is: Personal Home Page Tools. A rövidítés jelentése később PHP Hypertext Preprocessor lett, így rekurzívva vált. Később a PHP képességei bővültek, így egy önállóan használható programozási nyelv alakult ki, amely képes nagyméretű webes adatbázisalapú alkalmazások működtetésére is.

Mi is az a Perl ? A Perl egy általános célú, magas szintű, interpretált, dinamikus programozási nyelv, melynek első verzióját Larry Wall 1987. december 18-án tette közzé.

Stílusában és funkcionalitásában sokat merít a C, sed, awk és sh nyelvekből. A Perl egyik legfontosabb része a reguláris kifejezések széles körű támogatása, mely által kiválóan alkalmas nagy méretű szöveg- vagy adatfájlok egyszerű feldolgozására. Az 5-ös változat megjelenésével a 90-es években elterjedt a dinamikus, CGI alapú weboldalak nyelveként, de jellemzőbb felhasználási területei a már említetteken túl a rendszeradminisztráció, a hálózati programozás, a pénzügyi informatika és a bioinformatika.

A XAMPP segítségével vált lehetővé, hogy a program, a MySQL és a phpMyAdmin közötti kapcsolat létrejöhessen.

Program bemutatása

Az alábbi program egy szervíz és karbantartó menedzsment program. Ezen belül egy mobil telefon készülékekre szakosodott szervíz és karbantartó program. Ez a program a hibásan beérkező szereltetni kívánt mobiltelefon készülékeket hivatott adminisztrálni és nyilvántartani.

Az alábbi adatokat hivatott adminisztrálni:

- Beérkező eszköz márkáját, típusát, színét.
- Eszköz IMEI (egyedi) azonosítóját
- Beérkező eszköz, beérkezésének idejét rögzíteni és elvállalt határidőt rögzíteni.
- Ügyfél nevét, település irányítószámát
- Ügyfél e-mail címét (nem kötelező)
- Ügyfél telefonszámát (nem kötelező)
- A szervízelés státuszát (aktív, inaktív)
- Munkafolyamat rögzítése.
- Elkészült eszközök (Ügyfélre vár)
- Átadott eszközök (Átadva)
- Alap céges adatok

3.2. Formok és tulajdonságaik:

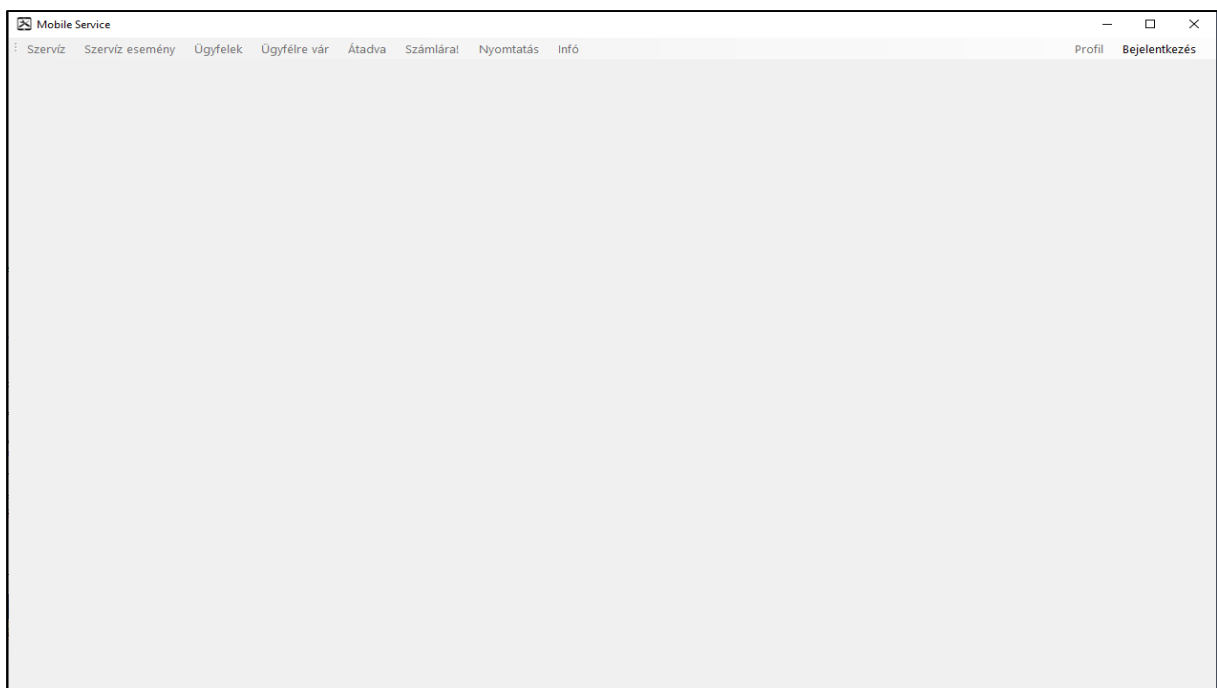
- **Form 1** – Főoldal (Részletes leírás lejjebb található)
- **Form 2** – Bejelentkezés (Az alap programba)
- **Form 3** – Új hozzáadás (Új eszköz felvitele a rendszerbe)
- **Form 4** – Szervízben lévő eszköz adatainak szerkesztése
- **Form 5** – Ügyfél adatainak szerkesztése
- **Form 6** – Ügyfél adatlap
- **Form 7** – Munkalap elkészítése (Amikor kész az adott eszköz)
- **Form 8** – Bizonylat elkészítése
- **Form 9** – Nyomtatás (bizonylat)
- **Form 10** – Munkafolyamat adatlap
- **Form 11** – Ügyfélre váró eszközök adatlap

- **Form 12** – Átadott eszközök adatlap
- **Form 13** – Céges profil bejelentkezés
- **Form 14** – Céges profil szerkesztése
- **Form 15** – Munkafolyamat szerkesztése
- **Form 16** – Ügyfélre váró eszköz adatainak szerkesztése
- **Form 17** – Átadott eszköz adatainak szerkesztése
- **Form 18** – Súgó (Programhoz segítség)
- **Form 19** – Szervizben lévő eszköz adatlap

3.3. Hogyan működik ?

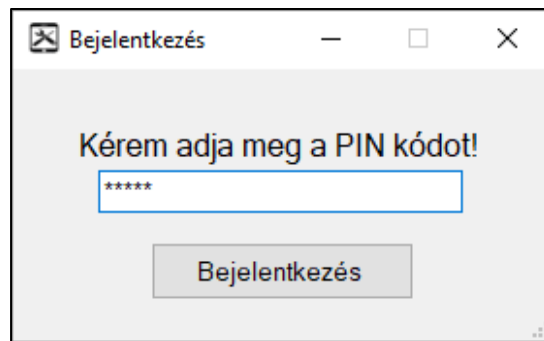
3.3.1. Bejelentkezés:

A szerviz program elindításakor a főoldal jelenik meg majdnem teljesen üresen. Ilyenkor a bejelentkezés előtt egyedül csakis a „Bejelentkezés” menüpont érhető el. Az egyéb menüpontok „disabled” változatban vannak azaz nem elérhető változatban.



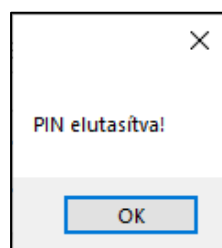
(főoldal bejelentkezés előtt)

Amikor rákattintunk a „Bejelentkezésre” elő jön egy újabb ablak „Bejelentkezés” néven. Itt egy szöveg mező látható, ahol meg kell adni a hozzátartozó PIN kódot.

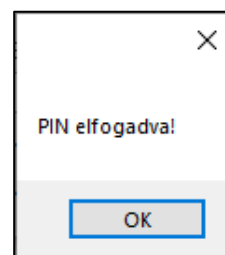


(bejelentkezés panel)

A szöveg mező csak számot fogad el. Amennyiben tartalmaz a szöveg mező nem szám formátumú elemet a bejelentkezés gombra kattintva jelezni fog a rendszer, hogy „Nem megfelelő a bemeneti formátum!”, ilyenkor elég az „OK” gombra menni és újra megadható a jelszó. Amennyiben helyesen lett megadva a jelszó (a programban PIN néven van megadva) a program beenged, eltűnik a bejelentkezés ablak és elérhetővé válnak a további menüpontok, mint például a „Szerviz”, „Szervizesemény”, „Ügyfél”, „Ügyfélre vár”, „Átadva” és a „Profil”. Viszont vannak olyan menüpontok amik fix helyzetben érhetőek el csak, mint például a „Számlára!” és a „Nyomtatás”. A „Számlára!” menüpont csak az „Ügyfélre vár” menüpontból érhető el, míg a „Nyomtatás” menüpont, csak az „Átadva” menüpontból érhető el. Amennyiben a PIN kód hibásan lett megadva úgy a program teljes egészében kilép és újra el kell indítani a programot. Csak hibás PIN esetén lép ki a program teljesen.



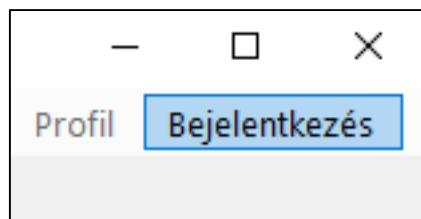
(elutasítva)



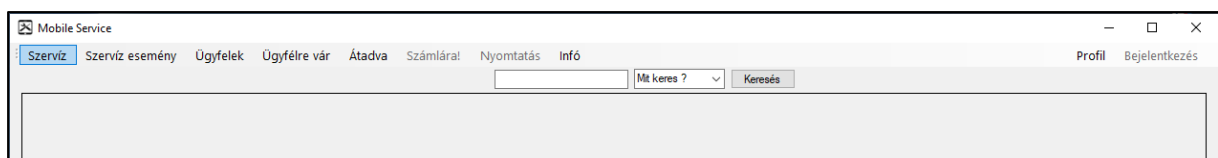
(elfogadva)

A bejelentkezés előtt a form1 azaz a főoldalon a második menüsáv látszódik csak. A második menü sávból csak a bejelentkezés elérhető.

A bejelentkezésre azért van szükség, hogy az illetéktelen személyek ne tudjanak hozzáférni a rendszerben tárolt adatokhoz. A bejelentkezés menüpontra kattintva megjelenik a form2 vagyis a bejelentkezés felület. Amennyiben van hozzáférésünk a programhoz meg kell adni a PIN-t a kijelölt szöveg mezőbe. A „buttonclick” (gombnyomás) esemény első sorban lekérdezi az adatbázisból a programhoz tartozó PIN kódot amit tárol egy integer változóban. A lekérdezés után egy „if” elágazás elkezd vizsgálni, hogy az integeré konvertált szöveg doboz/mező (textbox) azonos-e a lekérdezett PIN kóddal. Amennyiben azonos úgy a program egy üzenet (messagebox lásd.:) formájában jelzi a sikeres bejelentkezést és az „OK” gombra kattintva belép a program. Viszont amennyiben a PIN nem egyezik, úgy a program szintén egy üzenet formájában válaszol, hogy sikertelen a bejelentkezés és az „OK” gombra kattintva kilép teljesen a program. Bejelentkezéskor, ha simán csak a form 2 ablakban lévő jobb felső „X”-re rányomtunk akkor egyszerűen kilépett a form 2 és tovább engedett a form 1-re és elérhetővé vált minden. Ez javítva lett egy egyszerű form bezárás (form closing) eseménnyel. A form closing eseményhez létre lett hozva egy eldöntendő (bool) változó, ami azért felel, hogy a bejelentkezés sikeres (true) vagy sikertelen (false) volt-e. Ez szintén a „buttonclick” eseménynél dől el. Amennyiben sikeres (true) a bejelentkezés, úgy egyszerűen bezáródik a form 2 és kezelhetővé válik a form 1 (főoldal). Viszont amennyiben sikertelen (false) a bejelentkezés úgy a „form closing” eseményen belül igazzá válik az „if” ág (acceptLogin == false) és végrehajtódik a program teljes bezárása.



(bejelentkezés menüpont)



(sikeres bejelentkezés után)

3.3.2. Főoldal:

A főoldalon történnek az eseményeknek több mint fele és ez a kiinduló form innen indul minden esemény. A főoldalon keresztül elérhetőek a Szerviz, Szervizesemény, Ügyfelek, Ügyfélre vár, Átadva, Számlára!, Nyomtatás, Profil menüpontok. De itt elérhetőek az Új, Szerkesztés, Törlés, Aktív, Inaktív, Kész! menüpontok is és itt láthatóak az adatbázisok is (formban → dataGridView1-6). Ezek mellett található a formon egy könnyed, egyszerű kereső rendszer is. Ez az egyedüli form ami rugalmasan mozgatható. A Szerviz, Szervizesemény, Ügyfelek, Ügyfélre vár és az Átadva menüpontokra kattintva láthatóvá válnak a hozzájuk tartozó adatbázisok. Minden alkalommal a menüponthoz tartozó adatok jelennek meg. Az alsó menüsávban az Aktív, Inaktív és a Kész! menüpontok, csakis a Szerviz menüpont alatt elérhetőek.

A formon kiválasztott adatbázissal rendelkező menüpontoknál lehetőség van elemek szerkesztésére, törlésére és részletes adatlap megnyitására. A részletes adatlapok (lásd.:) a jobb átláthatóság miatt készültek. Az elemek tetszőlegesen a sorra kattintással kiválaszthatóak. Dupla kattintással pedig az adott elem részletes adatlapja jelenik meg. Hogyan is történik a kiválasztás pontosan ? (lásd.:) A program figyeli, hogy melyik adatbázis van megnyitva jelenleg és annak megfelelően az adott elemre kattintva (az adott sorban akárhová) figyeli, hogy az „ID” oszlopon belül hányadik sorban lett kijelölve a kiválasztott elem és ez alapján az értéket a programban „id” néven létrehozott változóban tárolja el. Kivétel a Szervizeseménynél, mivel ott több elem is lehet azonos egy „ID”-n, így ott egy külön változó lett nekik létrehozva sorszám néven és ebben a változóban is vannak eltárolva. A külön formon megnyíló lehetőség, mint például a Szerkesztések, a Kész! vagy a Számlára! menüpontoknak a form 1 külön tárol egy adatbázis frissítő eseményt, így frissülnek az adatok anélkül, hogy újra rá kellene menni az adott menüpontra az újonnan betöltéshez. A frissítések az adott form bezáródásához van igazítva azaz amikor a form 1 észleli, hogy egy másik form bezáródott, akkor újra tölti a megfelelő adatbázist a form (lásd.:).

A form 1 tartalmazza az aktív vagy inaktív színének kiválasztását a Szerviz menüpont adatbázisának. A piros szín a listában lévő inaktív elemeket jelzi, míg a zöld a listában lévő aktív elemek jelöli. Noha az adatbázisban van külön oszlop az „Aktív” és „Inaktív”

menüpontoknak, viszont kényelemből tekintve sokkal egyszerűbb és kényelmesebb látni, hogy melyik elem aktív és melyik inaktív a listában.

Ezek mellett kezelésre került a program teljes bezárása is, mivel eredetileg teljesen nem tudott leállni magától a program, ha a jobb felső „X”-re mentünk. Így egy egyszerű esemény kezeléssel vagyis egy „form closed” (lásd.:) eseménnyel teljesen sikerült bezárni a programot.

A form 1-en továbbra elérhető egy „Infó” néven futó menüpont ami röviden egy súgó a programhoz.

3.3.3. Szerviz menüpont:

A Szerviz menüponthoz tartozó adatbázis tárolja a szerelni kívánt eszköz adatait és a hozzátartozó ügyfél adatait, mint például.:

- Eszköz márkája (pl.: Huawei, Samsung)
- Eszköz típusát
- Eszköz egyedi azonosítóját (15 számjegyű IMEI azonosító)
- Eszköz színe
- Hiba
- Ügyfél neve
- Ügyfél telefonszáma (nem minden esetben)
- Ügyfél e-mail címe (nem minden esetben)
- Egyedi azonosító (ID)
- Megjegyzés(ek)
- Szerelés elvállalásának ideje
- Szerelés elvállalt határideje
- Aktív vagy Inaktív

Adatot felvinni az „Új” (lásd.:) nevezetű almenüponttal lehetséges. Tetszőlegesen kiválasztott elemet lehetséges szerkeszteni és törölni.

Szintén ezen a menüponton elérhető csak a „Kész!” menüpont is, mivel a szerelés nyilvántartása lépésről-lépésre megy a rendszerben.

s_id	s_brand	s_type	s_IMEI	s_c_name	s_c_pnumber	s_date_record	s_date_promised	s_color	s_problem	s_comment	s_active	s_inactive
2	Microsoft	LUMIA 950	765392847389123	Szendrei Árpád János	+36(20)3248476	2021-03-15	2021-03-17	Fehér	Készülék kijelzője hosszában megrepedt, a kép lá...		0	1
3	Samsung	GALAXY S8	345745231374995	Szabó Ivett	-	2021-03-15	2021-03-18	Sötét szürke	Ismeretlen okokból nem kapcsol be a készülék.		0	1

(Szerviz menüponthoz tartozó adatbázis)

A phpMyAdminban tárolt adatok láthatóak a fenti képen. Az azonosítók előtt az „s” betű jelöli a „Service” szót magyarul Szerviz. Az s_active és az s_inactive boolean változójú elemek a „0” jelenti a „false”-t és az „1” pedig a „true”-t. A programban amikor „true”-ra vagy „false”-ra kell hivatkozni akkor mindenféle kiegészítő jel nélkül (pl.: idéző jelek és stb.:) egyenlőség jel mögé oda írható a „true/false”, mint például (active = true), viszont a phpMyAdmin 0 és 1-ét úgy kellett kezelni, hogy (active = „True”) itt lényeges volt, hogy idézőjelbe legyen a true vagy false és az első betűjének nagynak kellett lennie lekérdezéskor.

Adatbázisban szereplő oszlopok nevei és jelentéseik:

- s_id => service identifier (-szervíz- azonosító)
- s_brand => service brand (-szervíz- márka)
- s_type => service type(-szervíz- típus)
- s_IMEI => service IMEI (-szervíz- IMEI azonosító)
- s_c_name => service customer name (-szervíz- ügyfél neve)
- s_c_pnumber => service customer phonenummer (-szervíz- ügyfél telefonszáma)
- s_date_record => service date record (-szervíz- dátum rögzítés – eszköz rögzítésének az ideje)
- s_date_promised => service date promised (-szervíz- dátum ígért – eszköz elvállalt elkészülésének ideje)
- s_color => service color (-szervíz- szín – készülék színe)
- s_problem => service problem (-szervíz- probléma – készülék hibája)
- s_comment => service comment (-szervíz- megjegyzés)
- s_active => service active (-szervíz- aktív)
- s_inactive => service inactive (-szervíz- inaktív)

3.3.4. Ügyfél menüpont:

Az ügyfél menüponthoz tartozó adatbázishoz tárolja az eszközhöz rendelt ügyfelek adatait. Viszont nincs befolyással a szerelési folyamat zavartalan menetében. Tartalmaz más adatokat amiket a Szerviz adatbázis nem, de nem elhanyagolható adatok.

Az adatbázisban tárolt adatok:

- Ügyfél teljes neve
- Ügyfél telefonszáma
- Ügyfél e-mail címe
- Ügyfél irányítószáma
- Ügyfél rögzítésének ideje

Lehetséges az ügyfél adatainak módosítása és törlése egyaránt.

Az ügyfél adatai közül csak az irányítószám ami innen lesz lekérdezve.

c_id	c_name	c_email	c_pnumber	c_postcode	c_date_record
2	Szendrei Árpád János	-	+36(20)3248476	3526	2021-03-15
3	Szabó Ivett	-	-	3527	2021-03-15
4	Erdei Péter	erdei.peter0101@gmail.com	-	3528	2021-03-13

(Ügyfélhez tartozó adatbázis)

Adatbázisban szereplő oszlopok nevei és jelentéseik:

- c_id => customer identifier (-ügyfél- azonosító)
- c_name => customer name (-ügyfél- neve)
- c_email => customer email address (-ügyfél- email címe)
- c_pnumber => customer phonenumber (-ügyfél- telefonszáma)
- c_postcode => customer postcode (-ügyfél- irányítószáma)
- c_date_record => customer date record (-ügyfél- dátum rögzítése – ügyfél rögzítésének dátuma)

Az ügyfél azonosítója (ID) az új hozzáadásakor generált azonosítót adja át. A phpMyAdmin-ban szereplő „customer” (magyarul: ügyfél) tábla oszlop neveiben szereplő „c” a customert jelöli, magyarul ügyfelet. Egyedi részletes adatlap elérhető. Az irányítószámhoz tartozik egy külön adatbázis ahol Magyarország majdnem összes postai irányítószáma elérhető település névvel együtt. Az irányítószámhoz tartozó település megjelenik mindenhol ahol fel van tüntetve az irányítószám.

id_postcode	num_postcode	settlement_postcode
1	2234	Maglód
2	2000	Szentendre
3	2009	Pilisszentlászló
4	2011	Budakalász

(Irányítószámok és településeik)

Adatbázisban szereplő oszlopok nevei és jelentéseik:

- id_postcode => identifier postcode (azonosító -irányítószám-)
- num_postcode => number postcode (szám -irányítószám- - az postai irányítószámok)
- settlement_postcode => ~||~ (település -irányítószám- - irányítószámhoz tartozó települések nevei)

3.3.5. Ügyfélre vár menüpont:

Az ügyfélre vár menüpont adatbázisa tárolja az olyan eszközöket, amik elkészültek és az ügyfélre várnak. A Kész! menüpontra kattintva és a szervizeseményt kitöltve kerülhet át ebbe az adatbázisba az adott készülék. Amennyiben sikeresen átkerült, úgy a Szerviz menüpont adatbázisából törlődik az elem. Itt is elérhető a szerkesztés, mivel lehetséges az is, hogy egy esetleges elírás később kerül szembe és így lehetséges a javítása vagy törlése (amennyiben töröljük az adott elemet az végleg elfog veszni). Egyedi részletes adatlap elérhető.

Az egyedi azonosító (ID) itt is öröklődik. Az egyedi azonosító teljesen a műveltek végéig öröklődik.

Az adatbázisban tárolt adatok:

- Egyedi azonosító (ID)
- Eszköz márkája

- Eszköz típusa
- Eszköz egyedi azonosítója (15 számjegyű IMEI azonosító)
- Ügyfél teljes neve
- Ügyfél telefonszáma (amennyiben meg van adva)
- Készülék színe
- Eszköz felvételének ideje
- Eszköz elvállalt határideje
- Eszköz elkészülésének ideje
- Eszköz hiba
- Megjegyzés(ek)
- Felelős szerelő teljes neve

d_id	d_brand	d_type	d_c_name	d_c_pnumber	d_IMEI	d_color	d_date_record	d_date_promised	d_date_done	d_problem	d_comment	d_m_name
11	Huawei	ENJOY 10	Szabadszabó János	+36(20)4233636	723426346845456	Fehér	2021-03-18	2021-03-22	2021-03-22	1. Kijelző sérült, érintő panel működik, kép látsz...	Nincs megjegyzés... További megjegyzés:	Szerelő István
17	CAT	B15Q	Gergely Balázs	-	765432196473825	Fehér	2021-03-18	2021-03-20	2021-03-25	1. Kijelző sérült, érintő panel működik, kép látsz...	Szerkesztett változat További megjegyzés: Munkadíj...	Szerelő Ottó

(Ügyfélre vár adatbázis)

A phpMyAdmin-ban szereplő „device_done” (magyarul: eszköz/készülék_kész – kész eszköz/készülék) nevezetű tábla oszlopainak nevében szereplő „d” betű a done-t (magyarul: kész) rövidítése.

Az ügyfél eszköz átvételekor innen van lehetőség a bizonylat elkészítésére/kiállítására. (lásd.:). Amennyiben elkészült a bizonylat úgy innen törlődni fog az adott elem a listából és átkerül az Átadott menüpont adatbázisába. Ebben a menüpontban nem elérhető az aktív, inaktív és a Kész! menüpont sem, úgy ahogy a Nyomtatás sem.

Adatbázisban szereplő oszlop nevek és jelentéseik:

- d_id => done identifier (-kész- azonosító)
- d_brand => done brand (-kész- márka)
- d_type => done type (-kész- típus)
- d_c_name => done customer name (-kész- ügyfél neve)
- d_c_pnumber => done customer phonenumber (-kész- ügyfél telefonszáma)
- d_IMEI => done IMEI (-kész- IMEI azonosító)
- d_color => done color (-kész- szín)

- d_date_record => done date record (-kész- dátum rögzítés – eszköz rögzítésének az ideje)
- d_date_promised => done date promised (-kész- dátum ígért – eszköz elvállalt határideje)
- d_date_done => done date done (-kész- dátum kész – eszköz szervizelésének elkészülés ideje)
- d_problem => done problem (-kész- probléma – eszköz hibája)
- d_comment => done comment (-kész- megjegyzés)
- d_m_name => done mechanic name (-kész- szerelő neve)

3.3.6. Szervizesemény menüpont:

A Szervizesemény menüponthoz tartozó adatbázisban az elkészült eszköz során elvégzett munkákat lehetséges adminisztrálni és tárolni. Lehetőség van esetleges hibák esetén szerkeszteni és törölni is. Az adott elem kiválasztása itt nem azonosító (ID) alapján történik, hanem az adott elemek sorszáma alapján. Az egyedi részletes adatlap itt is elérhető és a részletes adatlap megjelenítése is a sorszám alapján történik.

Az adatbázisban tárolt adatok:

- Egyedi azonosító (ID)
- Sorszám (szerviz esemény egyedi azonosítója, ennek segítségével lehetséges egyesével hivatkozni az elemekre)
- Rövid megnevezése az eseménynek
- Alkatrész megnevezése (amennyiben van)
- Mennyiség (amennyiben szükséges)
- Ár
- Részletes leírás
- Megjegyzés(ek)

A phpMyAdmin-ban szereplő „workflow” (magyarul: munkafolyamat) nevezetű tábla oszlopainak nevében szereplő „wf” betűk workflow-t rövidítik.

wf_sn	wf_id	wf_IMEI	wf_label	wf_s_parts	wf_quantity	wf_price	wf_description	wf_comment
3	11	758493847584782	Akkumulátor csere	Huawei P30 akkumulátor	1	17500	Készülék túl gyors merülése az akkumulátor meghibá...	Nincs megjegyzés... További megjegyzés:
4	11	758493847584782	Munkadíj	-	1	8500	Szét- és összeszerelés, átvizsgálás, kitakarítás, ...	Nincs megjegyzés... További megjegyzés:

(Munkaesemény adatbázis)

Itt sem elérhető az aktív, inaktív és a Kész! menüpont sem, viszont nem elérhető se a Számlára!, se a Nyomtatás menüpont sem.

Az adatbázisban szereplő oszlop nevek jelentései:

- wf_sn => workflow serial number (-munkafolyamat- sorszám)
- wf_id => workflow identifier (-munkafolyamat- azonosító)
- wf_IMEI => workflow IMEI (-munkafolyamat- IMEI azonosító)
- wf_s_parts => workflow service parts (-munkafolyamat- szerviz alkatrész – a munkafolyamathoz szükséges alkatrész)
- wf_quantity => workflow quantity (-munkafolyamat- mennyiség – az adott alkatrészhez tartozó mennyiség)
- wf_price => workflow price (-munkafolyamat- ár – az adott alkatrészhez tartozó ár)
- wf_description => workflow description (-munkafolyamat- leírás)
- wf_comment => workflow comment (-munkafolyamat- megjegyzés)

3.3.7. Átadva menüpont:

Az Átadva menüponthoz tartozó adatbázisban (invoice - számla) azok az eszközök kerülnek nyilvántartásba és adminisztrálásra amelyeket már az ügyfél visszakapta és bizonylat készült róla. A Számlára! kattintva és a kért adatok kitöltésével kerülnek át ide az eszköz adatai az Ügyfélre vár menüpont adatbázisából. Az adott elem azonosítója (ID) itt is öröklődik, úgy ahogy a többi folyamatnál a legelejétől. A szerkesztés és törlés lehetősége ahogy mindenhol úgy itt is elérhető esetleges hibák korrigálása érdekében. Szintén egyedi részletes adatlappal van ellátva a megtekinteni kívánt elem, amit dupla kattintással lehet megtekinteni. A menüponthoz tartozik egy egyedileg csak itt elérhető menüpont is, még pedig a Nyomtatás (lásd.:).

Az adatbázisban tárolt adatok:

- Egyedi azonosító (ID)
- Bizonylat egyedi azonosító
- Eszköz márka
- Eszköz típus
- Eszköz egyedi azonosító (IMEI azonosító)
- Ügyfél teljes neve
- Ügyfél telefonszáma
- Eszköz felvételének ideje
- Eszköz elvállalt elkészülés határideje
- Eszköz szerelés elkészülés ideje
- Eszköz átadásának/bizonylat kiállításának ideje
- Felelős szerelő
- Eszköz hiba
- Megjegyzés(ek)

Szintén nem elérhető itt sem az aktív, inaktív, Kész! almenüpontok.

A phpMyAdmin adatbázisban lévő „invoice” (magyarul: számla) nevezetű táblában szereplő oszlopok neveiben szereplő „i” betűk az „invoice” rövidítése.

i_id	i_u_id	i_brand	i_type	i_IMEI	i_c_name	i_c_pnumber	i_date_record	i_date_promised	i_date_done	i_date_taken	i_m_name	i_problem	i_comment
14	SZ0000014/2021	Huawei	GR5	324563456748556	Maradi Emő	+36(20)1242623	2021-03-17	2021-03-18	2021-03-18	2021-03-23	Szerelő Péter		
15	SZ0000015/2021	NOKIA	4.2	854633956734684	Szegedi Árpád István	-	2021-03-17	2021-03-19	2021-03-18	2021-03-19	Miskolci Endre		

(Átadva adatbázis)

Az adatbázisban szereplő oszlop neveik és jelentéseik:

- i_id => invoice identifier (-számla- azonosítót)
- i_u_id => invoice unique identifier (-számla- egyedi azonosító – bizonylat azonosító)
- i_brand => invoice brand (-számla- márka – készülék márkája)
- i_type => invoice type (-számla- típus – készülék típusa)
- i_c_name => invoice customer name (-számla- ügyfél név – az ügyfél neve)
- i_c_pnumber => invoice customer phonenumber (-számla- ügyfél telefonszám)

- i_date_record => invoice date record (-számla- dátum rögzítés – Eszköz rögzítésének dátuma)
- i_date_promised => invoice date promised (-számla- dátum ígért – Eszköz elkészülésének határideje)
- i_date_done => invoice date done (-számla- dátum kész – Eszköz elkészülésének ideje)
- i_date_taken => invoice date taken (-számla- dátum átadva – Eszköz átadásának dátuma)
- i_m_name => invoice mechanic name (-számla- szerelő neve – Felelős szerelő teljes neve)
- i_problem => invoice problem (-számla- probléma – Eszköz hibája)
- i_comment => invoice comment (-számla- megjegyzés)

3.3.8. Számlára! menüpont:

A számlára! menüpont egy olyan egyedi menüpont ami kivétel nélkül csak az Ügyfélre vár menüpontron érhető el. Amennyiben kiválasztottunk egye elemet úgy rátudunk kattintani a Számlára!. A Számlára! kattintás után egy új ablak nyílik meg, ahol egy pár adat megadása után elkészül a bizonylat.

A bizonylat kiállításakor megjelenő adatok a formon:

- Cég neve
- Cég adatai:
 - Elérhetőségek (telefonszám, email cím, stb.:)
 - Egyéb adatok (bankszámlaszám, adószám, stb.:)
- Ügyfél neve
- Ügyfél adatai:
 - Elérhetőség (telefonszám, email cím)
 - Egyéb adatok (irányítószám)
- Eszköz azonosítója (ID)
- Eszköz márkája
- Eszköz típusa
- Eszköz egyedi azonosítója (IMEI azonosító)

- Eszköz felvételének rögzítésének ideje
- Eszköz elvállalt elkészülésének határideje
- Eszköz elkészülésének ideje
- Felelős szerelő teljes neve

Továbbra megjelennek az adminisztrált szervizesemények is egy dataGridViewban. Az egyedi bizonylat azonosító automatikusan generálódik. Megadható adatok a bizonylat kiállítás/átadva dátumának megadása/rögzítése. A dátum megadása nem kötelező mivel mindig az aktuális napot mutatja. Ilyenkor az alap adatok egyike sem szerkeszthető vagy törölhető, csak áttekintést szolgál a form és a bizonylat kiállításának idejét megadni.

Amennyiben a kész gombra megyünk a rendszer az ügyfélre várból (phpMyAdminba device_done néven elérhető. Device_done = kész eszköz) kitörli az adott elemet és áthelyezi az átadvába (phpMyAdminba invoice néven elérhető. Invoice = Számla). Az oldal egy fix form nem lehet rugalmasan mozgatni.

Bizonylat kiállítás

Kiállító adatai:

Cég neve:

tesztCégSZERK Kft.

Adatok:

Telefon: +36(20)7654321

E-mail: tesztCégSZERK@cegmail.com

Cím: 3527 Miskolc, Valahol út 7.

Banszámlaszám: 12345678-12341234-00000000

Adószám: 12345678-1-12

Ügyfél adatai:

Ügyfél neve:

Szabadföldi János

Adatok:

Telefonszám: +36(20)4233636

Irányítószám: 3526 Miskolc

E-mail cím: elemerlakatos@freemail.hu

Munka adatai:

Azonosító:

11

Márka:

Huawei

Típus:

ENJOY 10

IMEI azonosító:

723426346845456

Rögzítés ideje:

2021. március 18., csütö

Vállalt elkész. ideje:

2021. március 22., héti

Elkészülési ideje:

2021. március 22., hét

Felelős szerelő:

Szerelő István

Munka folyamata/alkatrészek

Sorszám	ID	Cím	IMEI	Menny	Ár	Rész. leírás
3	11	Akkumulátor csere	758493847584782	1	17500	Készülék túl gyor
4	11	Munkadíj	758493847584782	1	8500	Szét- és összesz.

Bizonylat adatai:

Bizonylat azonosító:

SZ0000011/2021

Bizonylat kiáll. ideje:

2021. március 28., vasárnap

Kész

(Számlára! menüponthoz tartozó form)

28

A form bezáródása után a form 1-ben lefut egy esemény ami újra tölti az érintett adatbázisokat. Ez a pillanatnyi frissítést szolgálja, ennek segítségével nem kell újra rálépni a menüpontra, mivel egyből frissül. A munkafolyamati dataGridView-ban ahogy mindegyik másik dataGridView-nál is lehetőség van megnyitni és megnézni a részletes adatlapot. Az egyedi bizonylat azonosító a készülék szervizben kapott azonosítójából, az adott évből és egy „SZ” betűből áll, mint számla.

3.3.9. Nyomtatás menüpont:

A Nyomtatás menüpont egy olyan egyedi menüpont ami kivétel nélkül csak az Átadva menüponton keresztül elérhető. A Nyomtatás menüpontra kattintva egy új form jelenik meg, ami a Nyomtatás elő nézete.

Bizonylat nyomtatása

Bizonylat

Biz. azonosító: SZ0000014/2021

Kiállító adatai: Cég neve: tesztcégsZERK Kft. Telefon: +36(20)7654321 E-mail: tesztcégsZERK@cegmail.com Cím: 3527 Miskolc, Valahol út 7. Banszámlaszám: 12345678-12341234-00000000 Adószám: 12345678-1-12		Ügyfél adatai: Neve: Maradi Ernő Telefonszáma: +36(20)1242623	
Bizonylat kelte:	Felvétel ideje:	Elv. határidő:	Elkész. ideje:
2021. 03. 23.	2021. 03. 17.	2021. 03. 18.	2021. 03. 18.

Eszköz márkája: Huawei	Eszköz típus: GR5
IMEI azonosító: 324563456748556	Felelős szerelő: Szerelő Péter

Sorsz.	ID	Cím	Menny.	Ár	Részletes leírás
13	14	Hiba diagnosztizálása	0	0	Eszköz szétszerelése és átvizsgálása
14	14	Akkumulátor csere	1	8500	Huawei GR5 készülék akkumulátor cseréje, meghibásodás miatt.
15	14	Munkadíj	1	4500	Szét- és összeszerelés, átvizsgálás, hiba diagnosztizálása, alkatrész cseréje stb.

Fizetendő: Szerelési költség + ÁFA (28%)

Összesen: 16 640,00 Ft (nettó)

Átadó aláírása

Átvevő aláírása

(Bizonylat nyomtatásának elő nézete)

A Nyomtatás form megjelenésekor minden fontos adat megjelenik.

Nyomtatáskor megjelenő adatok:

- Kiállító adatai:
 - Cég neve
 - Cég adatai (elérhetőség, egyéb adatok)
- Ügyfél adatai:
 - Ügyfél neve
 - Ügyfél adatai (elérhetőség, egyéb adatok)
- Eszköz rögzítésének ideje
- Eszköz elvállalt határideje
- Eszköz elkészülésének ideje
- Eszköz átadásának/bizonylat kiállításának ideje
- Eszköz márkája
- Eszköz típusa
- Eszköz egyedi azonosítója (IMEI azonosító)
- Felelős szerelő teljes neve
- Munkaesemények.
- Az aktuális manuálisan beállított ÁFA (általános forgalmi adó)
- Egy összesített ár (nettóba)
- Bizonylat egyedi azonosítója

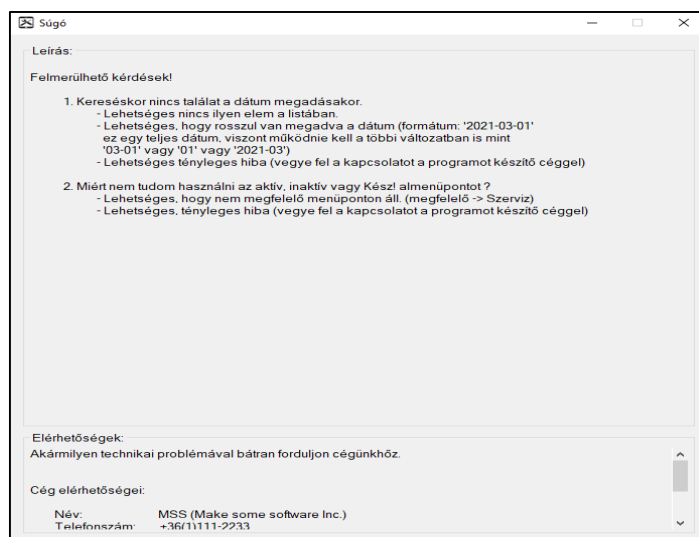
Ezek mellett van rajta két hely az aláírandó személyeknek is és egy nyomtatás gomb.

Itt szerkesztés vagy törlésre lehetőség nincs, mivel ez csak egy egyszerű elő nézet.

Amennyiben módosítani kívánunk egy elemet, úgy ki kell lépni ebből a formból és az Átadva menüpont alatt kell kiválasztani a szerkeszteni vagy törölni kívánt elemet. Az ár összesítő úgy működik, hogy a program a munkaesemény adatok betöltése után elkezd vizsgálni az „ár” nevezetű oszlopot és ahogyan halad elemről elemre úgy mindig hozzáadja az újabb elemeket. Amikor kijön a teljes összeg kiszámolásra kerül az ÁFA (általános forgalmi adó) és hozzáadásra kerül az alap összeghez. Végül kiírásra kerül.

3.3.10. Infó menüpont:

Az infó menüpontra kattintással előjön egy újabb form, amin esetleges kérdésekkel vagy hibákkal kapcsolatban kereshetünk és találhatunk megoldást. Az infó menüpont röviden egy súgó ablak. Amennyiben a kérdésre nem sikerült megfelelő vagy egyáltalán nem sikerült választ kapni úgy a kiállító/programot készítő céggel kapcsolatos adatok/elérhetőségek is nyilván vannak tartva.



(Infó panel (súgó))

Ilyen céges (program készítő) adatok pl.:

- Cég neve
- Telefonszáma
- Cég helyének címe
- E-mail címe
- Egyéb adatok...

3.3.11. Profil menüpont:

A profil menüpontra kattintással egy bejelentkező felület fog fogadni. A helyesen megadott PIN kód (nem ugyan az mint az alap programhoz tartozó) után megjelenik a rendes form, ami segítségével az egyedi céges profil adatokat tudjuk megadni és módosítani, emellett tudjuk módosítani az ÁFÁ-t (Általános Forgalmi Adó) is. Amennyiben a kívánt módosítások végre lettek hajtva úgy elmenthető az adat vagy adatok.

Szerkesztés

Cég adatai:

Cég neve: Irányítószám: Cím:

Telefonszám: Formátum: +36(20)1234567 E-mail cím:

Bankszámlaszám: Adószám:

Általános forgalmi adó (ÁFA): %

(Céges profil szerkesztése)

Szerkeszthető adatok:

- Cég neve
- Irányítószáma, helyének címe
- Telefonszáma
- E-mail címe
- Bankszámlaszáma
- Adószáma
- Általános Forgalmi Adó (ÁFA)

Az aktuális adatokat minden alkalommal beolvassa a program.

3.3.12. Almenüpontok:

3.3.8.1. Új hozzáadás:

A menüpont mindenhol elérhető. Az „új” menüpontra kattintva megjelenik egy új form. Ez az új form tartalmazza az adat mezőket amik kitöltésével felvihetők a rendszerbe a szervizelésre szánt eszközök adatai. Ez az egész eseménye sorozat legelső tagja. Ezen a formon generálódik az egyedi azonosító.

Új ügyfél/eszköz megadása

Adatok megadása:

Ügyfél teljes neve: E-mail cím: Telefonszám: Formátum: +36(30)1234567
☐ Nincs megadva ☐ Nincs megadva

Felvétel dátuma: Terv. elkészülés dát.: Irányítószám: Miskolc

IMEI azonosító: 15 Szín: Márka: Típus:
☐ Nincs a listában

Hiba megadás listából: Márka: Típus:
☐ Egyénileg adom meg

Probléma/hiba megadása: Megjegyzés(ek):

Kész!

(Új hozzáadása form)

Formon megadható adatok:

- Ügyfél teljes neve
- Ügyfél e-mail címe (nem szükséges)
- Ügyfél telefonszáma (nem szükséges)
- Eszköz és ügyfél felvételének dátuma
- Tervezett elkészülésének ideje
- Ügyfél irányítószám
- Eszköz egyedi azonosítója (IMEI azonosító)
- Eszköz színe
- Eszköz márkája
- Eszköz típusa
- Hiba megadása
- Megjegyzés(ek)

Az e-mail címet és telefonszámot nem szükséges megadni, viszont amennyiben mégis meglennének adva úgy egy ellenőrzésen mennek keresztül. A program vizsgálja, hogy az e-mail és a telefonszám is tartalmazza-e az adott formátum karaktereket, amennyiben nem azt a

program jelzi a felhasználónak. Az egyedi készülék azonosítóhoz az IMEI azonosítóhoz tartozik egy karakter számláló az egyszerűség miatt és amennyiben nem megfelelő a karakter hossza, úgy jelez szintén a felhasználónak. Az eszköz márkájának és típusának kiválasztására lehetőség van listából vagy egyénileg megadott módon. Szintén lehetőség van a hiba fixen listából megadott lehetőségére vagy egyénileg megadására is. Az irányítószám szöveg doboza mögött van egy címke, ami ha beírjuk a rendszerben tárolt irányítószámok egyikét akkor írja a hozzátartozó település nevét is. (Magyarország postai irányítószámai) A form bezáródásakor a Szerviz menüpont adatbázisa frissítésére kerül sor.

3.3.8.2. Szerkesztés, törlés:

Lehetőség van mindegyik elemet külön-külön szerkeszteni és törölni is. A szerkesztés mindegyik menüpontnál egyedi. Egy elem kiválasztása után a szerkesztés menüpontra kattintva megjelenik egy a főbb menüponthoz (Szerviz, Szervizesemény, Ügyfelek, Ügyfélre vár, Átadva) tartozó formok egyike. Amikor az adott form megnyílik a kiválasztott elem adatai a megfelelő helyre betöltésre kerülnek.

Szerkesztés

Eszköz adatai:

Ügyfél neve: Szabadi Sándor

Telefonszám: - Formátum: +36(30)1234567

☒ Nincs megadva telefonszám

Márka: BlackBerry

Típus:

Felvétel dátuma: 2021. március 22., hétfő

☐ Kikeresem a listából

Márka: Microsoft

Típus: LUMIA 735

Terv. elkészülés dát.: 2021. március 24., szerda

IMEI azonosító: 823439845534574

Szín: 15 Piros

Hiba megadás listából: Válasszon egy elemet a listából

☒ Egyénileg adom meg

Probléma/hiba megadása: 3. Kijelző NEM sérült, érintő panel NEM működik, kép látszódik.

Megjegyzés(ek): Kezden el gépelni...

Kész

(Szerviz szerkesztés form)

A kész gombra kattintva a form bezáródik és az adott elemek és a főbb menüpont adatbázisa frissítésre kerül.

3.3.8.3. Aktív, inaktív:

Ezek a menüpontok szabályozzák a Szerviz menüpont adatbázisának színezését. Az újonnan felvett elemek a rendszerbe piros színnel kerülnek rögzítésre azaz inaktív állapotban vannak. Az inaktív állapot annyit jelent, hogy fent van a rendszerben, de még nincs szerelés alatt. Amennyiben az Aktív menüponttal átváltjuk az értékét Aktívra úgy zöld színre változik az adott elem, ezeknek a segítségével könnyebben kezelhetőek az Aktív vagy Inaktív elemek. Amennyiben inaktív egy elem, az nem mehet a további lépésekre. Esetleges félre kattintások okából.

3.3.8.4. Kész!:

Amikor egy elem az adatbázisban aktív és elkészül a szervizeléssel, úgy a Kész! menüpontra kattintva megjelenik egy új form, ahol a szervizesemény adminisztrálása után átkerülhet az Ügyfélre vár menüpont adatbázisába.

Munkalap elkészítése

Jelenlegi eszköz adatai:

Azonosító: 33 Márka: Huawei Típus: Y6 Tulajdonos neve: Fidrus Márk Telefonszáma: +36(30)5594311

IMEI azonosító: 634574568543456 Szín: Fekete Rögzítés ideje: 2021. 03. 22. 0:00:00 Váll. határidő: 2021. 03. 22. 0:00:00

Hiba leírása:
1. Kijelző sérült, érintő panel működik, kép látszódik

Megjegyzés:
Tartozák: Huawei Y6 sima fekete tok

Munka folyamata:

Munka adatai:

Munka rövid megnevezése:

Alkatrész megnevezése:

Mennyiség: 0 db Ár: 0 Ft

Egyéb adatok:

Felelős szerelő:

Elkészülés dátuma: 2021. március 22., hétfő

Részletes leírás:
Tartozák: Huawei Y6 sima fekete tok

Megjegyzés:

Hozzáadás Törölés

Kész

(Kész! form)

Formon szereplő adatok:

- Eszköz azonosítója (ID)
- Eszköz márkája
- Eszköz Típusa
- Ügyfél teljes neve
- Ügyfél telefonszáma
- Eszköz egyedi azonosítója (IMEI azonosító)
- Eszköz színe
- Rögzítésének ideje
- Elvállalt határideje
- Hiba leírása
- Megjegyzés(ek)

Megadható adatok:

- Munkaesemény rövid megnevezése (többször lehetséges)
- Alkatrész megnevezése (többször lehetséges)
- Alkatrész mennyisége (többször lehetséges)
- Alkatrész ára (többször lehetséges)
- Részletes leírás (többször lehetséges)
- Felelős szerelő teljes neve
- Eszköz elkészülésének dátuma
- Megjegyzés(ek)

A kész gombra kattintva a Szerviz menüpont adatbázisából törlésre kerül az elem és az Ügyfélre vár adatbázisába pedig rögzítésre. A formon megadott munkaesemények szintén rögzítésre kerülnek az eszközhöz tartozó azonosító alatt a Szervizesemények menüpont adatbázisában.

Tovább fejlesztési ötletek

4.1. Egyedi faviconok:

Jelenleg csak egy darab ikonnal rendelkezik a program és a formokon ez a form van megjelenítve. Szerkezetileg nem egy teljesen lényeges ötlet, viszont minimálisan látványosabb amennyiben minden résznek egyedi ikonja van, mint például a szerkesztéseknél egy ceruza vagy egy fogaskerék, a törlésnél egy jellegzetes „x”, stb.: Az esztétikus windows keret érdekében nem lett módosítva a magasság, így egy bizonyos mértékben nyomottabb lett a kép. Ez természetesen kérésre korrigálható dolog.



(favicon)

4.2. Mobil applikáció:

A mai pörgő és gyorsan fejlődő világban lehetséges, hogy alkalmas lenne egy ilyen műveletre egy okos telefon vagy egy tablet is. Az alapvető funkciókat teljes mértékben képes lenne ellátni egy okos telefon/tablet. Egyetlen dolog ami nem lenne lehetséges vagy legalábbis nem esztétikus kivitelezni az adat tábla megjelenítése. Ehelyett egy kereső rendszer lenne kifejlesztve az applikációnak amivel lehetséges lenne kiválasztani, hogy mi alapján szeretne keresni és egy kulcsszó megadásával könnyedén megkeresni azt. A másik szembetűnő dolog, hogy hogyan állapítja meg a szerelő, hogy melyik eszköz aktív és melyik nem vagy hogy melyiknek van közelebb a határideje. Egy hamburger menü segítségével lennének kezelve a menüpontok. A menüpontok között lenne olyan menüpont, hogy „Mutasd”. A Mutasd egy kereső menüpont lenne, viszont fix egyedi menüpontokkal személyre szabva, mint például aktív, inaktív, sürgetett. Logikusan az aktív menüpont tartalmazza az aktív készülékeket, az inaktív az inaktív készülékeket, a sürgetett pedig az olyan eszközöket tartalmazza ami közel jár a lejártához. Minél közelebb az elvállalt határidő annál lényegesebb az adott eszköz

szervizelése. Erre egy másik ötlet is adódott, hogy amelyik elemnek közelebb van a lejárat ideje az előrébb van a sorban. Az eszközök pedig egymás alatt sorakoznának és egyszerűen egy azonosító alapján lenének besorolva. Amennyiben az egyik elemre a listában rányomunk megjelenne egy új oldalon a részletes adatlapja. Hatékonyabbnak tűnik ez a telefonos applikáció abból a tekintetből, hogy nem muszáj oda menni a géphez kikeresni az adott eszköz adatait vagy nem foglal plusz helyet mint egy laptop, hanem csak egyszerűen és hatékonyan tárolható zsebben is akár. Amennyiben valamire szükség van az eszközzel kapcsolatban úgy csak előveszi a zsebéből és az applikáció segítségével kész is.

Lehetséges hátrány a kisebb kijelző miatt következhet be.

4.3. Raktár készlet nyilvántartó program:

Amennyiben van a cégnek helyszínen is alkatrésze és nem az adódó szervizeléssel rendelik meg azokat, úgy egy raktár nyilvántartó program beolvasztásával számon lehetne tartani, hogy egy adott készülékhez szükséges alkatrészt meg kell-e rendelni vagy van készleten. Az elképzelésem a programomba olvasztást illetően úgy nézne ki, hogy amikor megadja az illető a munka folyamatot és alkatrész névhez egy kötőjelet ír (-) akkor a program figyelmen kívül hagyja a dolgot, viszont amennyiben elkezd gépelni egy eszköz alkatrész úgy egyben elkezdene keresni is a program és ha van lehetséges találta kiírná amire rákattintással beillesztené a szöveg mezőbe. A beillesztést követően egy címkén kiírná a szövegdoboz mögött, hogy található-e ilyen alkatrész a raktáron.

Záródolgozathoz felhasznált források

Microsoft Visual Studio

https://hu.wikipedia.org/wiki/Microsoft_Visual_Studio

C#

https://hu.wikipedia.org/wiki/C_Sharp

Windows Form

https://en.wikipedia.org/wiki/Windows_Forms

PHP

<https://hu.wikipedia.org/wiki/PHP>

phpMyAdmin

<https://hu.wikipedia.org/wiki/PhpMyAdmin>

XAMPP

<https://hu.wikipedia.org/wiki/XAMPP>

MySQL

<https://hu.wikipedia.org/wiki/MySQL>

MariaDB

<https://hu.wikipedia.org/wiki/MariaDB>

Apache

https://hu.wikipedia.org/wiki/Apache_HTTP_Server

Perl

[https://hu.wikipedia.org/wiki/Perl_\(programoz%C3%A1si_nyelv\)](https://hu.wikipedia.org/wiki/Perl_(programoz%C3%A1si_nyelv))

