Beregszászi Pál Technikum

SZOFTVERFEJLESZTŐ ÉS -TESZTELŐ TECHNIKUS SZAKMA

Raktárkezelés és Webshop Project

Készítette Serdült Zoltán

Csapattagok Tokai Kiss Máté

Debrecen, 2022.

[Bevezetés 2](#_Toc100590938)

[A Webshop ismertető 2](#_Toc100590939)

[A Raktározás rész ismertetője 3](#_Toc100590940)

[Fejlesztői dokumentáció 4](#_Toc100590941)

[Adatbázis modell felépítés 4](#_Toc100590942)

[A program szerkezete 10](#_Toc100590943)

[A Program részletes ismertetés 11](#_Toc100590944)

[A Form1 ablak elemei 11](#_Toc100590945)

[A form1 Váltózói és Algoritmusai 12](#_Toc100590946)

[A Form\_Foablak ismertetése 14](#_Toc100590947)

[A Form\_Foablak ablak elemei 14](#_Toc100590948)

[A form\_Foablak Váltózói és Algoritmusai 17](#_Toc100590949)

[A Form\_Foablak-ba főbb funkciói 18](#_Toc100590950)

[Termékek megjelenítése 18](#_Toc100590951)

[A termék adatai módosítása az adatbázisból. 20](#_Toc100590952)

[A termék törlése az adatbázisból. 21](#_Toc100590953)

[Felhasználói dokumentáció 22](#_Toc100590954)

[Termékek megjelenítése 22](#_Toc100590955)

[Új termék hozzáadása az adatbázishoz. 22](#_Toc100590956)

[A termék adatai módosítása az adatbázisból. 22](#_Toc100590957)

[A termék törlése az adatbázisból. 23](#_Toc100590958)

[További Fejlesztési lehetőségek 23](#_Toc100590959)

[Nem teljesített funkciók 23](#_Toc100590960)

[További fejleszthető funkciók 23](#_Toc100590961)

[Összegzés 23](#_Toc100590962)

# Bevezetés

A mi csapatunk Tokai-Kiss Máté és Serdült Zoltán egy már nagyon régi esetre próbálunk egy szoftvert csinálni a kereskedelem terén.

A legtöbb üzlet foglalkozik raktározás és eladás részel és a informatika megjelenésével egyre hatékonyabb lett. Kezdetben még csak nyilvántartók és raktárkezelési programok léteztek de a Internet feltűnése nagyba meg változtatta a üzletek világát. Lehetőség nyílt Internetes eladásoknak a Web-shopok létrejöttének.

Mi egy üzlet online kezelését szeretnénk megvalósítani a Termékek felvételtől a interneten való árusításáig (ami egy web-shopként valósulna meg)

Mint ketten a Áru kezelés valamely területén dolgozunk, így van valamely rá láttásunk a kereskedés és áru logisztika működésére és szeretnénk hatékonyabbá tenni azzal hogy virtuális világba ültetjük át, ezzel sokkal könnyebb és kényelmesebb lenne a munkamenet

## A Webshop ismertető

(Írta: Tokai Kiss Máté)

Elektronikus kereskedelem (e-kereskedelem) alatt az árucikkek interneten való kereskedését értjük, aminek a fő alapja az olyan holnap, ahol ez a folyamat végbemegy. Ezeket a honlapokat hívjuk webshopnak (internetes bolt, internetes áruház, online áruház). Eltérően a hagyományos áruházaktól a webshopokban a valós termékekkel fizikailag nem találkozik a vásárló, hanem általában kép formájában tudja megtekinteni. Általában a termékekhez tartozik egy részletes leírás, amely tartalmazza az áru pontos specifikációit. A legtöbb webáruház egy speciális piaci részesedést céloz meg a teljes piaci szegmensből, de természetesen vannak kivételek, mint például az Amazon. A mai napra igen kiforrott technológiai háttérrel rendelkezik a – közel harminc éves – online értékesítés. Rengeteg különböző módszert fejlesztettek ki a kereskedés megoldására, mint például az igen elterjedt kosaras rendszerű webshop, ahol a vevő egy virtuális kosárba tudja pakolni a megvásárolni kívánt termékeket, amelyet tetszés szerint akár ki is vehet a kosárból. A rendelés véglegesítéséhez szükség van egy előzetes regisztrációra a weboldalon, így rögzíteni lehet a szállításhoz kapcsolódó információkat, vagy cégek esetében az adószámot is, amelyek így a jövőbeni vásárlást könnyítik. A regisztrációval az összes vásárlást tárolni lehet, amit így a vásárló bejelentkezés után bármikor meg tud tekinteni.

## A Raktározás rész ismertetője

(Írta: Tokai Kiss Máté)

Raktározásnak az áru tárolására, állagának megóvására, a készletek elhelyezésére szolgáló tevékenységeket nevezzük. Leggyakrabban természetesen kiszolgáló szerepe van: a vállalkozások által megtermelt értékek tárolója. Célja a nyersanyag biztonságos tárolása a felhasználásáig, fizikai tulajdonságainak megőrzése. A raktárakat, raktározás rendszereket elsősorban a különböző munkafolyamatok gazdaságos kapacitási különbségeinek kiegyenlítése teszi szükségessé, mint például a beszerzéskori optimális mennyiség. A raktározás sajátos szerepet tölt be a kitermeléstől a felhasználásig terjedő komplex folyamatrendszerben. A raktárak a részfolyamatok összekötő elemei. A raktározás történhet nyitott és zárt helyen. Zárt helyen tárolják például az elektronikai eszközöket, világítástechnikai termékeket (izzók, LED-szalagok). A raktározás általában valamely üzlet zavartalan és folyamatos működését, valamint az eszközök, nyersanyagok, félkész- és késztermékek szakszerű, veszteségmentes megőrzését szolgálja mindaddig, míg azok felhasználásra, illetve értékesítésre kerülnek. A raktározás folyamata a következő részfolyamatokat foglalja magába: az áru átvétele, az áru tárolása, kezelése, az áru kiadása. Az egész folyamat kezdete az áruátvétel, ahol rögzíteni lehet a beszállítót, az áruátvételt végző személyt, időpontot, pontos mennyiséget. Az áruátvétel után a termékeket raktározni kell.

# Fejlesztői dokumentáció

## A Hardver és Szoftverismertetés

A Hardverem egy régebbi laptop amin a Windows 7 Ultimate található, az éppen irt dokumentumot Az Microsoft Office Word 2010 szoftverrel írom (Ez által lehetnek fórmázási inkompatibilitási hibák ha egy újabb Worddel nyitják meg a dokumentumot. A Vizsgaremek fájljait a Git verzió követőre töltöm fel a projekt munka fájljait

A Szoftverem a Windows operációs rendszer alatt készült mert elég elterjedt és ismert a kezelése is. A fejlesztést Visual Studio Community 2019-cel valósítottam meg C# nyelven .NET 4.7.2 Frameworkkel. Lehetőség lett volna a Opensource vonalon a SharpDevelop fejlesztői környezettel is de az már nincs támogatva 2016 óta, a VC2019 kiváló fejlesztési környezet mert Microsoft termék mint a Windows így nem lehetnek inkompatibilitási hibák. A Programunk adatbázisra támaszkodik amit az én részem csekély részét használja, Az adat bázist a normalizálás szabályai szerint optimalizáltuk. Az adatbázist a XAMMP 3.3.0 csomagban lévő MariaDB 10.4.24-ban futtatom és beépített phpMyAdmin-nal menedzselem. Lett volna még lehetőség a DBforge Studióval is megalkotni de az nem teljesen ingyenes.

## Adatbázis modell

Az adatbázisra az egész szoftverünk támaszkodik mind a C# raktár kezelő részre és a Webes webshop részre is. A adatbázisunk utf8\_hungarian karakterkészletet és az utf8\_hungarian\_ci karakterillesztést használja az alap innodb engine-re épülve.

### dolgozó tábla

Az alkalmazásban fontos a felhasználók megkülönböztetése. A mezők a következők: **dolgozo\_id, felhasznalonev, jelszo, nev**.

A **dolgozo\_id** egy integer típusú autoincrement elsődleges kulcs ami csak azonosításra és illesztésekre használatos

A **felhasznalonev** egy varchar típusú 50 karakter hosszúságú karakterlánc amit a admin ad meg a épp a programot használni akaró dolgozó embernek nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű és szám .

A **jelszo** a felhasznalonev mező párja csak ha a kettő meg egyezik akkor lehet belépni ez egy varchar típusú 50 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű és szám amit a admin add meg a épp a programot használni akaró dolgozó embernek

A **nev** a dolgozo tábla egyedének a értelmezhetőbb beazonosítására van az admin számára ez egy varchar típusú 50 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű és szám .

### beerkezesek tabla

a dolgozo\_id köt össze a beerk\_id vel A mezők a következők: **beerk\_id, beszallito, atvetel\_datum, dolgozo\_id**

A **beerk\_id** egy integer típusú autoincrement elsődleges kulcs ami csak azonosításra és illesztésekre használatos

A **beszallito** a beszállítót tárolja el amely egy varchar típusú 50 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű és szám .

Az **atvetel\_datum** mezö az adott időpontot rögzíti amikor be érkezik a adott árú. Tipusa datetime.

A **dolgozo\_id** a dolgozo táblával való kapcsolatot teszi lehetővé összevan kötve a dolgozo\_id a dolgozo táblába. Ez integer típus.

### beerkezett\_termek tábla

amely az árú mennyiségét számlálja a mezői a következők: **beerk\_id, termek\_id, mennyiseg.**

**beerk\_id** A beerkezesek van összekötetésbe ez a mező amely integer típus. autoincrement elsődleges kulcs ami csak azonosításra és illesztésekre használatos

**termek\_id** ez a mező a termek táblával van összekötve szintén integer típusú.

**mennyiseg** mező a beérkezett termék mennyiségét tartja számon (és majd hozzá adja a készlethez) szintén integer típusú

### A termek tábla

a fő része a rendszerünknek itt tároljuk a termék adatait és tulajdonságait a tartalma**: termek\_id, nev, ar, keszlet, foglalat, forma, teljesimeny, szinhomerseklet, termekkod, ean, allapot, leirás, elerheto, akcio, marka,**

**termek\_id** Ez a elsödlegeskulcs amely a beérkezett termék és a rendelt termék táblákal van összekötve integer típusú.

**nev** A termék nevét tárolja ez a mező varchar típusú 50 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű és szám.

**ar** A termék árát tartalmaza ami integer típusú

**keszlet** Ez a mező a készlet mennyiségét tárolja a típusa integer

**foglalat** Ez a mező a termék foglalat típusát jelöli (ha ez létezik többnyire a világítás technikai termékeket ismerő rendszert fejlesztünk ezt nem kötelező megadni) a típusa a integer.

**forma** ez a mező a termék formáját tárolja a tipusa varchar típusú 20 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű és szám (ha ez létezik többnyire a világítás technikai termékeket ismerő rendszert fejlesztünk ezt nem kötelező megadni).

**teljesitmeny** a termék teljesítményét tárolja ami integer típus (ha ez létezik többnyire a világítás technikai termékeket ismerő rendszert fejlesztünk ezt nem kötelező megadni).

**szinhomerseklet** ez a mező a termék Színhőmérsékletét jelöli ha ez világítás technikai termék integer típusú.

**termekkod** A termékkód az üzlet belső termékkódját tarolja varchar típusú 20 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű és szám.

**ean** Az nemzetközi ean kódot tárolja ami integer tipus.

**allapot** Ebbe a mezőbe a termék állapotát lehet rögzíteni típusa varchar típusú 20 karakter hosszúságú karakterlánc (Lehetet volna egy enum típus is de így sokkal részletesebben le lehet írni az állapotot)

**leiras** A termék leirását tartalmazza a Webshop résznél lesz főbb jelentősége mert ott látszik legjobban ez a kiírás. típusa text.

**elerheto** Ez a mező a termék elérhetőségét jelölné ami int típus. Ha 0 az értéke akkor nem elérhető (Nem látható a webshop résznél) ha 1 az értéke akkor elérhetö

**akcio** Az akcio mező a termék árának az akcióját tárolja el ami int típus.

**marka** A márka a termék márkáját tárolja ami varchar típusú 20 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű és szám.

### rendelt\_termek tábla

amibe a rendelés és a termék ára és mennyisége van számontartva. A mezői: **rendeles\_id, termek\_id, mennyiseg, osszeg.**

**rendeles\_id** Ez a mező a rendelt\_termek és a rendeles táblákat köti össze ami egy integer típusú autoincrement elsődleges kulcs ami csak azonosításra és illesztésekre használatos

**termek\_id** Ez a mező a rendelt\_termek és a termek táblákat köti össze ami egy integer típusú autoincrement.

**mennyiseg** ez a mező a megrendelt termék mennyiségét tárolja ami a termek táblából levonható a típusa integer.

**osszeg** ez a mező a megrendelt termék árát tárolja ami a típusa integer

### rendeles tábla

ahol a termék megrendelése tárolódik le a megrendelt termékkel és rendelővel van összekötetésbe a mezöi: **rendeles\_id, vasarlo\_id, rendeles\_datum, osszeg, megjegyzes.**

**rendeles\_id** Ez a mező a rendeles és a rendelt\_termek táblákat köti össze ami egy integer típusú autoincrement elsődleges kulcs ami csak azonosításra és illesztésekre használatos.

**vasarlo\_id** Ez a mező a rendeles és a vasarlo táblákat köti össze ami egy integer típusú autoincrement.

**rendeles\_datum** itt a rendelés pillanatnyi idő pontja tárolódik le Datetime tipus.

**osszeg** itt a meg rendelt termék ára lesz letárolva amit a megrendelő visz be a tipusa integer.

**megjegyzes** Itt a megrendelt termékkel kapcsolatos igények, és egyéb meg jegyzések kerülnek eltárolásra a megrendelő keze által.

### vasarlo tábla

ahol a vásárló az adatai tudja le tárolni mezői: **vasarlo\_id, vnev, knev, email, phone, elfogadva, jelszo, irszam, varos, utca, hazszam.**

**vasarlo\_id** Ez a mező a vasarlo és a rendeles táblákat köti össze ami egy integer típusú autoincrement elsődleges kulcs ami csak azonosításra és illesztésekre használatos.

**vnev** ebbe a mezőbe a vásárló a vezeték nevét tárolja le varchar típusú 20 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű

**knev** ebbe a mezőbe a vásárló a kereszt nevét tárolja le varchar típusú 20 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű

**email** ebbe a mezőbe a vásárló a e-mail címét tárolja le varchar típusú 30 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű

**phone**  ebbe a mezőbe a vásárló a vezeték nevét tárolja le varchar típusú 20 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű és szám is lehet.

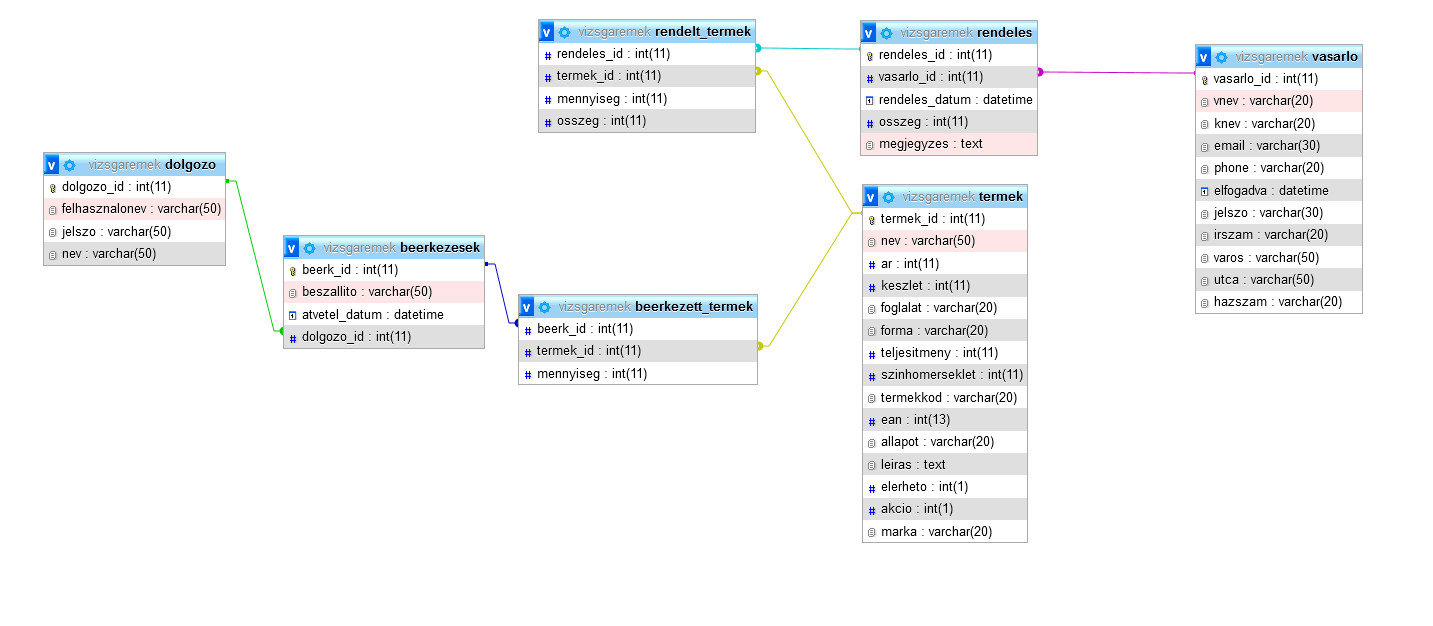
**jelszo** ebbe a mezőbe a vásárló a jelszavát tárolja le varchar típusú 30 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű

**irszam** ebbe a mezőbe a vásárló a vezeték nevét tárolja le varchar típusú 20 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű

**varos** ebbe a mezőbe a vásárló a városa nevét tárolja le varchar típusú 50 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű

**utca** ebbe a mezőbe a vásárló a utca nevét tárolja le varchar típusú 50 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű

**hazszam** ebbe a mezőbe a vásárló a vezeték nevét tárolja le varchar típusú 20 karakter hosszúságú karakterlánc nincs semmi megkötöttség a bevitelhez lehet kis-nagy betű



# A program szerkezete

A programom pár főbb szempontjai

* **Egyszerűség** mert a felhasználót ne terheljük felesleges átláthatatlan funkciókkal amik felesleges időre menne el a munkája során.
* **Hatékonyság** Legyen a program gyors és kevés legyen a futási ideje az adott funkciók közt. Ne legyen feleslegesen megirt algoritmusok és logikátlan felépítések.
* **Átláthatóság** A program legyen egyértelmű a felhasználó számára ne tévessze össze a feleslegesen kitalált funkciók a munkája során az csak idő és pénz kiesést eredményezz számára

A program Szin világa a klasszikus Windows szint tükrözi mert így talán otthonosabban érzi magát a felhasználó a munkája során. Grafikai elemeket nem használtam mert az csak el vonhatja a munkaközbe a felhasználót a dolga alul (Nem is feltétlenül a backend programnak kell olyan látványosnak lenni inkább a webshop rész nézzen ki valahogy.)

A programnak van 2 ablaka a Form1 nevezető (Ami a klasszikus a VC2019-be) és a Form\_Foablak ablak ami egymásból nyílnak majd. A Form1-gyen levő indítás gombra kattintva el indul a Form\_Foablak amire a Program.cs fájlba osztály váltózóként hivatkozók rájuk és az adott funkció által meg nyílnak

A program egy beléptető rendszerrel fogadja az aktuális dolgozót mindenkinek van egy egyedi jelszava és felhasználói neve a vizsgaremek adatbázis dolgozo táblájába. amihez egy a dolgozó rendes neve társul a könnyebb beazonosításhoz Ezenkívül az elsődleges kulcs általi sorozat számmal is lehet egyszerűbb adatbeolvasást végezni és ezzel újabb funkciókat meg alkotni.

A program a Mysql.Data függvény könyvtárat használja amit a Oracle fejlesztése más alternatíva is lenne (de ez az egyetlen ami az eredeti sql nyelvhez van legközelebb) a Nuget kiegészítővel szereztem be a Visual Studio 2019 fejlesztői programon keresztül a projectbe társítva (nem lett külön letöltve a Windowsba).

# A Program részletes ismertetés

A Form1 nevű ablaknál történik a dolgozó bejelentkezése a programba felhasználó és jelszó megadással.



## A Form1 ablak elemei

* Az ablak kis ikonja (Faviconja) egy kis villanykörte jelez ezzel a program jellegét az ikon a iconarchive.com-ról lett le töltve a linkje a következő <https://iconarchive.com/show/office-icons-by-vexels/bulb-icon.html>.
* A kezelő név alatti szöveg dobozba kerül a dolgozó neve amit az admin add meg majd. És összehasonlítja a vizsgaremek nevű adatbázis dolgozó tábla felhasznalo mezőjével. A mezőbe kerülhet kis és nagy betű szám, a karakter hossza nincs le korlátozva
* A Jelszó alatti szöveg dobozba kerül a dolgozó jelszava amit az admin add meg. És összehasonlítja a vizsgaremek nevű adatbázis dolgozó tábla jelszo mezőjével. A mezőbe kerülhet kis és nagy betű szám, a karakter hossza nincs le korlátozva
* A bejelentkezés feliratú gombra kattintva megjelenik a Program indítása feliratú gomb ha meg egyeznek a Kezelő név és a Jelszóban beirt adatok.
* A programindítása gombra kattintva elindul a fő ablak vagyis a program.

## A form1 Váltózói és Algoritmusai

A Form1 nevezetű ablakon található kettő Textbox a Textbox\_Belepesnev és a Textbox\_Belepesjelszo. A Textboxokba beirt adatokat a program ellenőrzi hogy üres-e a két textbox tartalma ezt a IsNullOrEmpty tulajdonsággal ellenőrzöm (Lehetet volna isnullorwhitespace tulajdonsággal is ). Ha üres akkor a program kiírja egy Messageboxba mind két esett be hogy hiányzik az adott adat és vissza tér az eredeti állapotába hogy a felhasználó újra meg adja az adatait. Ezután az adatotokat két string változóba lementi a program a későbbi műveletek miatt az input\_nevbe kerül a Textbox\_Belepesnev adatai az input\_jelszoba a Textbox\_Belepesjelszo adatai.

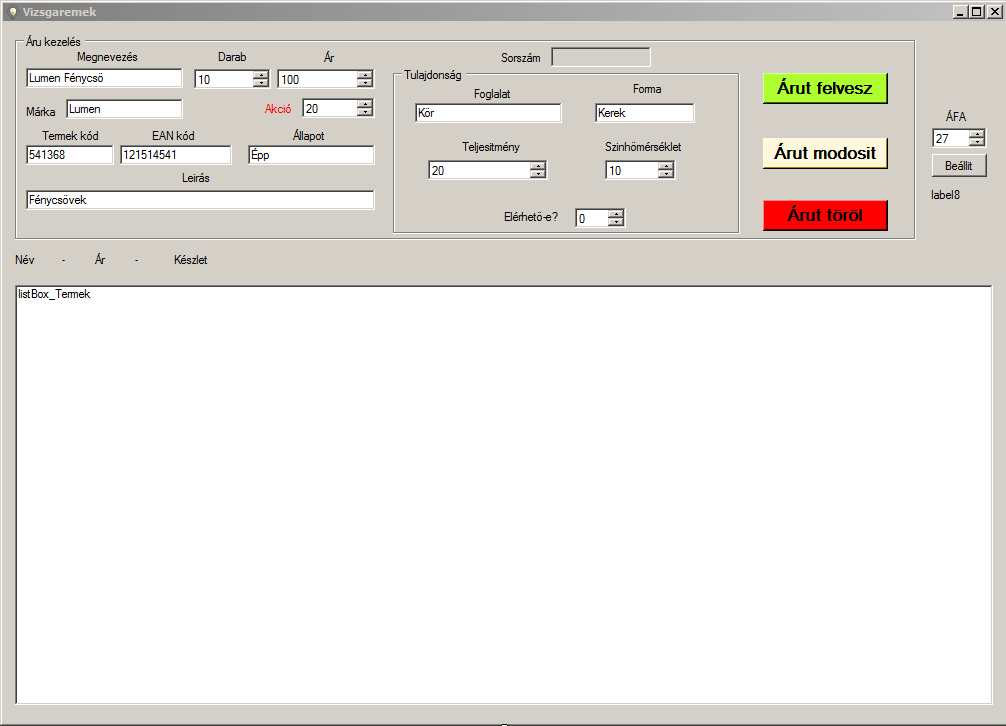
A program a kapcsolat alapú adatelérést használja, a kapcsolat nélkülivel szembe itt nem történhetnek összeütközések adatok között sokkal jobban kezelhetőbb az adat forgalom. Sok módja lenne a Connectionstring létrehozásnak de én a MySqlConnectionStringBuilder osztályt használtam jól átlátható és könnyű az adatait meg változtatni. Ebből lett példányosítva egy sb példány És a kötelező adatokat megadtam Szervernek a localhost, a UserID a root, a jelszó ami éppen nincs. az adatbázis nevet is biztos ami biztos alapon a karakterkódolást utf8 írtam be Ezután létre hoztam egy con nevű példányt a Mysqlconnection függvény könyvtárból amihez paraméternek a sb paramétereit átadom szöveg formába a Tostring átalakítással.

A kapcsolat egy try catch ágba van elhelyezve mert lehetséges hogy nem működik az adat bázis kapcsolódás ez esetbe a catch ág a következő hiba üzenetet irja ki ” Keresse fel a rendszergazdát” És kiírja az aktuális Mysql hibát az ex nevű Mysqlexception váltózóval. Az esetben ha mégis sikerül a kapcsolat akkor a Mysqlcommand osztályból létre jön egy cmd példány amely a con mysqlconnection példány CreateCommand tulajdonsága hajtódik végbe. Ezek után be állítódik a cmd (Mysqlcommand példány) Commandtext-je amely a: **SELECT `felhasznalonev`,`jelszo` FROM `dolgozo** parancsot kapja meg amit csak annyit csinál hogy a dolgozo táblából a felhasznalonev és jelszo mezőket keresi meg. Itt nyitódik meg a con nevű váltózó (Mysqlconnection példány) az open metódussal. Most már itt ténylegesen nyitva van az adatkapcsolat. Létre hoztam egy MysqlDatareader példányt amit dr a neve és egy using részben helyeztem el hogy ha befejeződött az adat kapcsolat biztosan kapcsolódjon le az adat áramlásról. A dr-nek (MysqlDatareader példány) atadom a cmd Execurereader tulajdonságát ezzel kiolvas az adatbázisból a dr idegenleges táblába. Ezután a dr végig olvasódik egy while ciklussal a read tulajdonsággal beolvasva a felhasználonevet az adatbázisból és ki mentve egy felh nevü string váltózóba és kiolvasva a jelszot és egy jelszo string váltózóba lementve szintúgy. Ezek után egy if elagazás megvizsgálja hogy a input\_nev (Amit a felhasználó visz be a textBox\_Belepesnev textbox révén) egyenlő-e a adatbázisból beolvasott felh váltózóba levő karaktersorozattal és a input\_jelszo (Amit a felhasználó visz be a textBox\_Belepesjelszo textbox révén) egyenlő-e a adatbázisból beolvasott jelszo váltózóba levő karaktersorozattal. És ha nem egyenlő akkor egy messagebox figyelmezteti a felhasználót hogy Rossz adatott adott meg és ki törli a textBox\_Belepesnev és a textBox\_belepesjelszó Text tulajdonságát (Ezt a funkciót lehetöség lenne ki hagyni de akkor a messageboxot egy NO és OK gombosra kéne változtatni hogy kitörli a textbox tartalmát vagy ,nem olyan esetben lenne jó ez a működés ha hibásan bevitt adat kicsit tér el a helyestől pl: Felhasznalonev helyett Felhasznaev). Ha viszont megfelelő az if feltétele akkor Meg változik a Form1 Grafikus felülete. A button\_Programindito Viseble tulajdonsága True vált (ami azt eredményezi hogy látható lesz a Program indítás lehetőség) És a textbox\_Belepesnev és a textBox\_Belepesjelszo viseble tulajdonsága false-re vált a hozzájuk tartózó label-kel együtt (Ezzel eltűnnek az ablakról) A label1 text értéke a „Bejelentkezve” értéket kapja meg (Igy jelezve a felhasználónak hogy be van jelentkezve).

Ez után megjelent a button\_Programindito gomb amivel a form\_Foablak nevű ablakot jeleníthetjük meg .

# A Form\_Foablak ismertetése

A sikeres bejelentkezés után megjelenik a program fő ablaka ami a Form\_foablak nevet kapta. Itt történik a Termek tábla megjelenítése egy listboxba. A Termékhez adhatunk új cikket ami az adatbázisba kerül, módosíthatjuk is a tulajdonságait és el is dobhatjuk az adatbázisból. Egy Áfa érték numericupdown is van a ablakon azon keresztül tudja kiíratni a program az árú áfás árát.

****

## A Form\_Foablak ablak elemei

* Az ablak kis ikonja (Faviconja) egy kis villanykörte jelez ezzel a program jellegét az ikon a iconarchive.com-ról lett le töltve a linkje a következő <https://iconarchive.com/show/office-icons-by-vexels/bulb-icon.html>.
* Az árukezelés gruopbox fogja közbe vizuálisan a termékkezelés funkciót (Termék felvétel, Termék módosítás, Terméktörlés).
* A megnevezés textboxba a termék neve kerül. Nincs semmilyen feltétel hozzá rendelve kerülhetnek bele kis és nagybetűk, és számok is a hossz érték nincs korlátozva.
* A darab nevü Numericupdown mezőbe a termék mennyiségét lehet szabályozni (ha mínusznak érzékeli a program a mennyiségét akkor azt a Form\_Foablak betöltésnél egy messageboxba jelzi az árú nevével.
* Az ár nevü Numericupdown mezőbe a termék árát lehet megadni ami nem lehet mínusz érték (Numericupdown minimum értéke 0)
* A márka textboxba a termék márkája kerül. Nincs semmilyen feltétel hozzá rendelve kerülhetnek bele kis és nagybetűk, és számok is a hossz érték nincs korlátozva.
* Az akció numericupdown a termék árának az akcióját állítja be (minimum értéke 0 a maximum 100).
* A termékkód textboxba a termék belsö kódja kerül vagyis az üzlet által használt kód amit maga a üzlet talál ki. Lehet algoritmusokat írni rá miként generálja le majd a kódot Pl: DEB0723 város kezdőbetűi termék besorolás stb.
* EAN
* Az állapot textboxba a termék állapota kerül ami lehet több féle Pl épp, sérült ,gyári hibás (Az állapotot lehetet volna egy listboxként meg adni Enum adatként de az túl kötött így sokkalta részletesebben le lehet írni a termék állapotát) Nincs semmilyen feltétel hozzá rendelve kerülhetnek bele kis és nagybetűk a hossz érték nincs korlátozva.
* A leírás textboxba a termék leírása irható ami egy hosszabb szöveg lenne. Nincs semmilyen feltétel hozzá rendelve kerülhetnek bele kis és nagybetűk, és számok is a hossz érték nincs korlátozva.
* A sorszám textbox csak olvasható ítt csak a felvett termék sorszáma látható így meg könnyítve az adott termék meg találását (az adatbázis a termek\_id ami automatikusan növekszik.
* A tulajdonság gruopboxba lévő adatokat nem kötelező kitölteni csak az égőtestekre érvényes.
  + A foglalat textbox a termék (égövest) foglalatát lehet rögzíti. Nincs semmilyen feltétel hozzá rendelve kerülhetnek bele kis és nagybetűk a hossz érték nincs korlátozva.
  + A forma textbox a termék (égövest) formáját lehet rögzíti. Nincs semmilyen feltétel hozzá rendelve kerülhetnek bele kis és nagybetűk a hossz érték nincs korlátozva.
  + A teljesítmény numericupdown-ba a egötest teljesítménye kerül ami aminek a minimális értéke 0.
  + A Színhőmérséklet  numericupdownba a égőtest Színhőmérséklet kerül. Aminek a minimális értéke a 0.
* Az elérhető-e numericupdown-ba a termék elérhetősége kerül. Ha nulla a benne levő értéke akkor nem elérhető ha egy akkor elérhető (Ezt egy checkbox által is meg lehetne valósítani úgy még talán kisebb lenne a memória használat mert igaz vagy hamis értéket adna vissza, de numericupdownba több érték is kerülhet például ha valami időszakosan érhető el lehetne a 2.érték kiárusítás 3.érték készlet kisöprés.
* Az Árut felvesz gomb az adatok kitöltése után rákattintva bekerülnek az adatbázisba és a alább látható listboxba meg jelenek a legfontosabb értékek (Név Ár Készlet Áfás ár). A gomb zöld színe a gyarapodást növekedést kívánná kifejezni
* Az Árut módosít gomb a listboxba megjelenítet termékekből változtatásokat hajt végre a termék kiválasztása után a termék. A termék kiválasztás hatására az adatai bekerülnek a beviteli mezőkbe módosítás hatására bekerülnek az adatbázisba
* Az Árut töröl gomb megnyomására az adatbázisból a termék táblából eldobódik egy rekord. A listboxból kiválasztás szükséges természetesen. A gomb piros színe a veszélyre hívná fel a figyelmet mert ha valamit törlünk utána nem hozható vissza
* Az Áfa nevű numericupdown a termék árának áfa értékét állítja be a beállít gomb hatására ezt a értéket elmenti a program és a következő indításkor betölti és kiírja egy labelbe hogy mennyi az éppen aktuális áfa érték.
* Az ablakon levő listboxba a termék legfontosabb jellemzői kerülnek kiírásra a név, az ár, a készlet mennyiség, és az ár áfa értékei. Ha részletesebben kívánnánk meg nézni a termék adatait akkor azt kiválasztva azok betöltöttnek a beviteli mezőkbe és onnan módosíthatok is akár (listbox mellett azért döntöttem mert így kicsit hagyományosan néz ki a jegyzetezés mintha egy noteszbe írnánk fel az adott árút. Másik alternatíva a DataGridview lenne ami sókkal átláthatóbb de én ez mellett döntöttem)

## A form\_Foablak Váltózói és Algoritmusai

A program kezdetlegesen objektum orientált, található benne egy Termék osztály ami az adatbázis termek táblát felelteti meg.

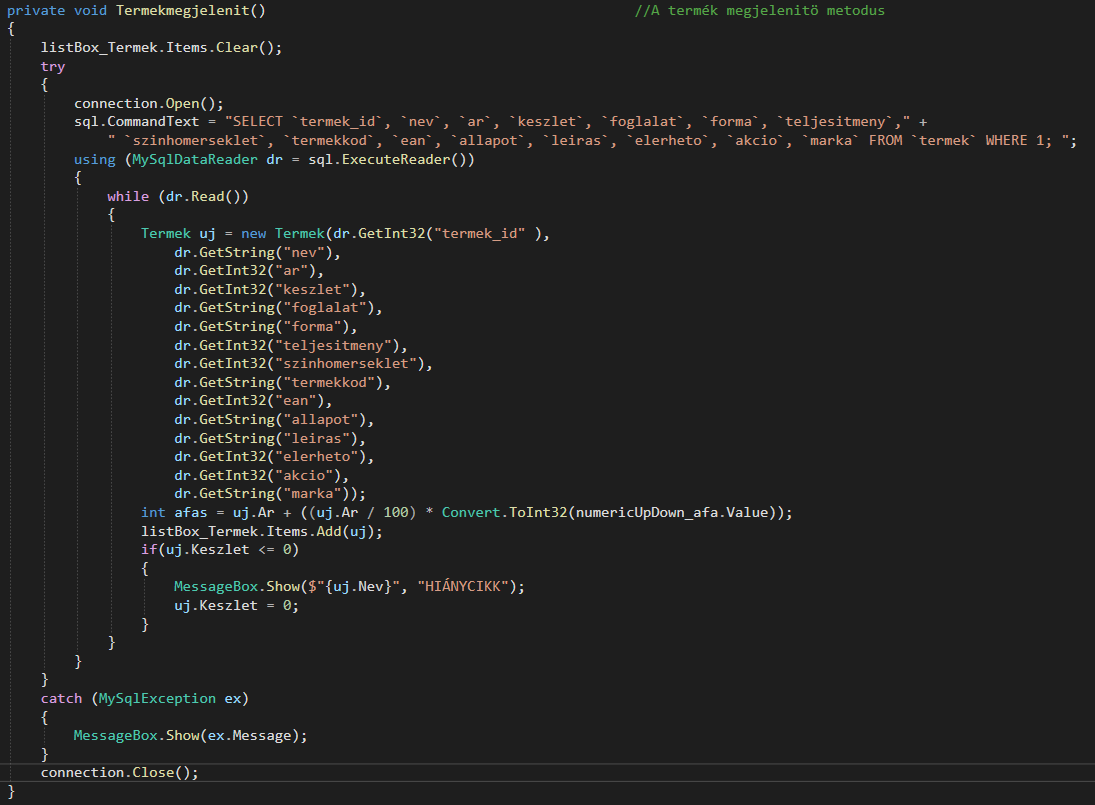
**Termék osztály objektumai:**

* **termek\_id** – Csak olvasható string típusú A termék egyedi azonosítója fontos lesz a termék törlésnél
* **nev** - string típusú váltózó a termék nevét tárolja
* **ar** – int típusú a termék árát tárolja nem tud mínusz érték lenni a numericupdown minimum értéke nem engedi meg (ezt a set részbe is meg lehetne határozni)
* **keszlet** - egy int tipusó változó a termék mennyiségét (darabszámát) tárolja.
* **foglalat** – A foglalat egy string típusú változó a világító test foglalat típusát kívánja eltárolni
* **forma** – A forma egy string típus a világító test formáját tárolja el.
* **teljesitmeny** – A teljesitmeny egy int típusú változó a világító test teljesítményét tárolja le
* **szinho** – A szinho egy int típusú változó a világító test színhőmérsékletét tárolja le.
* **termekkod** – A termekkod a az árú belső kódját tárolja amit az üzlet kérése szerinti feltételeknek lehet generálni. tipusa string.
* **ean** – Az Ean a nemzetközi kódot tárolja ami int tipús
* **allapot** – Az allapot a termék állapotát írja le típusa string (Ezt lehetet volna egy Enum-ként tárolni vagy tömbként)
* **leiras** – A leirás váltózóba kerül a termék leírása ez lehet hosszabb karakter sorozat típusa string.
* **elerheto** – Az elerheto váltózóba a termék elérhetősége tárolható le ami int tipús ha nulla az értéke akkor nem elérhető ha pedig egy akkor elérhető (több értéket is lehet felvetetni benne pl 2 végkiárusítás esetén 3 készlet kisöprés.
* **akcio** – az akció az termék akciós árát szabályozza ami int típus.
* **marka** – A marka a termék márkáját tárolja ami egy string tipús.

## A Form\_Foablak-ba főbb funkciói

* Termékek megjelenítése
* Új termék hozzáadása az adatbázishoz.
* A termék adatai módosítása az adatbázisból.
* A termék törlése az adatbázisból.

### Termékek megjelenítése

A program elején példányosítva van a connection a Mysqlconnection függvény osztályból és a sql a Mysqlcommand függvény osztályból amihez nem adunk semmi értéket később még több helyen lesz használva ezért tetűnk így. Az ablak betöltésékor a jön létre az adatbázis kapcsolat Az kapcsolat a kapcsolat alapú adatelérést használja, a kapcsolat nélkülivel szembe itt nem történhetnek összeütközések adatok között sokkal jobban kezelhetőbb az adat forgalom. Sok módja lenne a Connectionstring létrehozásnak de én a MySqlConnectionStringBuilder osztályt használtam jól átlátható és könnyű az adatait meg változtatni. Ebből lett példányosítva egy sb példány És a kötelező adatokat megadtam Szervernek a localhost, a UserID a root, a jelszó ami éppen nincs. az adatbázis nevet is. A connection példánynak átadótik a sb szöveges adatai. Ezekután a connection példánynak a CreateCommand tulajdonságát használva át kerül a sql példányba. Itt jön egy Termekmegjelenit nevű metódus erre több helyen szükség lesz (jegyezzük meg). A Termekmegjelenit metódus a Listbox kitörlésével kezdődik azért mert mindig a friss és nem a régi adatokat adja vissza. Egy try catch blokk-ba kerül a beolvasás. Ha sikeres a müvelet akkor nyitóm a connection-t az open tulajdonsággal. Ezután a sql példánynak beállítom a Commandtextjét ami: SELECT `termek\_id`, `nev`, `ar`, `keszlet`, `foglalat`, `forma`, `teljesitmeny`, `szinhomerseklet`, `termekkod`, `ean`, `allapot`, `leiras`, `elerheto`, `akcio`, `marka` FROM `termek` WHERE 1; ". Ez az adatbázis termék táblája lekérdezése. utána egy using részbe teszem a MysqlDatareader dr példányát az sql ExecuteReader-rel. Majd egy While ciklussal végig megyek de read tulajdonságával a Termék osztályból létrehozott uj nevű példányán át adva a lekérdezés eredményeit a uj példány paramétereinek. Ezután a adatokkal feltöltött uj példány hozzá adom a list\_Termek elemeihez. Itt csináltam egy elágazást ha a Keszlet kisebb mint nulla akkor írja ki a éppen aktuális Terméknek a nevét egy messageboxba (és állítsa nullára az értéket ez nem egy jó lekezelés mert az adatbázisba mínuszba kerülnek a készlet elemek és rossz értékeket látunk majd). Ha viszont az adatbázis nem kapcsolódik akkor a catch ág egy a Mysqlexceptionból ex váltózóba kimentett parancsot kiír messageboxba. És itt zár a connection példány a close tulajdonsággal.

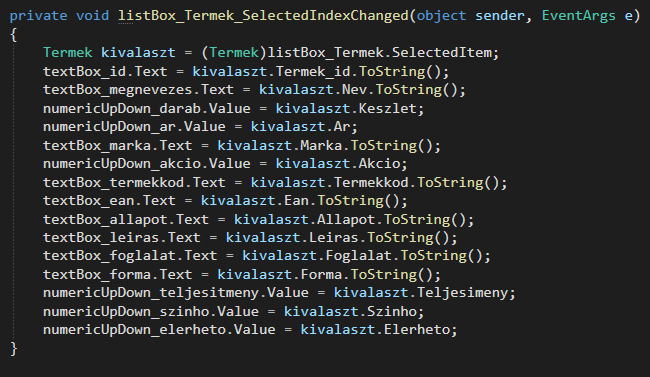
*A Termekmegjelenit metódus*

**Új termék hozzáadása az adatbázishoz**

Az Árut felvesz gomb a button\_felvesz nevet viseli ennek a click eseménye az Ellenoriz metódussal kezdődik ami le ellenőrzi a beviteli mezők üresek-e if elágazásokkal ha üres akkor a felhasználót figyelmezteti egy messageboxal hogy töltse ki az üres mezőt az és visszatér a foablakra az ellenőrzött mezők: textbox\_megnevezes, textbox\_marka, textbox\_termekkod, textbox\_ean, textbox\_allapot, textbox\_leiras. A tulajdonságok lévő gruopboxba nem kötelező ellenőrizni mert az értéküket se kötelező megadni. Ezután a sql példányak beállitom a parancsát ami a következő : "INSERT INTO `termek` (`termek\_id`,`nev`, `ar`, `keszlet`, `foglalat`, `forma`, `teljesitmeny`, `szinhomerseklet`, `termekkod`, `ean`, `allapot`, `leiras`, `elerheto`, `akcio`, `marka`) " +

"VALUES (null, @nev, @ar, @keszlet, @foglalat, @forma, @teljesitmeny, @szinhomerseklet, @termekkod, @ean, @allapot, @leiras, @elerheto, @akcio, @marka);". Ez a parancs paraméteresen van meg írva hogy ne közvetlen az adatbázisba kerüljenek Különféle műveletek (Ez az sql injekció kivétése). Ez után a beviteli mezők átadják az értéküket a paraméteresen az adatbázisnak. Utána jön egy try catch rész ha nem sikerül az adatbázisba írni akkor a program figyelmezteteti a felhasználót egy Messageboxal a hibára egy adatbázis hibakóddal. A sikeres művelet esetén a connection példány meg nyílik és a sql Executenonquery végrehajtódik (Megtörténik a új termék felvétele az adatbázisba. a művelet végzése után le zár a connection példány a close tulajdonsággal és a Termegmegjelenit metódus lefut a listbox friss tartalma miatt.

### A termék adatai módosítása az adatbázisból.

A termékmódosításhoz ki kell választani egy elemet a listboxból és annak az elemnek a tulajdonságai bekerülnek a beviteli mezők üres helyeire. Ez úgy történik hogy a Termék osztályból létre hozzunk létre egy példányt ez a kiválaszt névvel lesz el látva. és a listBox\_Termek.SelectedItemet rákényszerítjük a Termék osztályt így a kiválaszt példány lesz a kiválasztott elem Termek osztálybeli megfelelője. Ez után a beviteli mezőkbe betöltjük a kivalaszt példányokat az osztálybeli megfelelő váltózóival szövegesen amit a Tostringkel érünk el.

A Termék kiválasztásának és a beviteli mezőkbe betöltésének algoritmusa

Most már ha kiválasztunk egy elemet a listboxból akkor azok adatai betöltöttnek a beviteli mezőkbe. De ha nincs kiválasztva egy elem se akkor az is le van kezelve úgyhogy ha a listbox\_termek\_selectedindexe kisebb mint nulla (Vagyis az első elem előtti elem lenne kiválasztva ami nem lehetséges) akkor egy messagebox kiírja hogy Nincs kiválasztva termék és vissza a ablakra. Utána jön az Ellenoriz metódus és minden beviteli mezőbe betöltött adatot kimentünk egy váltózóba a megfelelő tipús szerint. Azután beállítjuk a sql példány commandtextjét itt is paraméteresen feleltetjük meg az adatokat. Mielőtt feltöltenék az adatbázisba a beviteli mezekből az adatokat előtte tisztítsuk ki a paramétereket a clear tulajdonsággal. Utána jön egy try catch rész ha nem sikerül az adatbázisba írni akkor a program figyelmezteti a felhasználót egy Messageboxal a hibára egy adatbázis hibakóddal. A sikeres művelet esetén a connection példány meg nyílik és a sql Executenonquery végrehajtódik (Megtörténik a új termék módosítása az adatbázisba. a művelet végzése után le zár a connection példány a close tulajdonsággal és a Termegmegjelenit metódus lefut a listbox friss tartalma miatt.

### A termék törlése az adatbázisból.

A törlés egy elem kiválasztásánál történik ha nincs ki választva elem akkor a nincs kiválasztva termék messagebox ugrik fel az algoritmusa a következő ha a listbox\_termek\_selectedindexe kisebb mint nulla akkor nincs kiválasztva termék és vissza tér az ablakra az ok gombra kattintva utána lefut a Elllenoriz metódus. Az adatbázisba a id alapján törli ki az adott elemet ez biztos egyedi adat. a programba csinálunk egy int id váltózót amibe a listbox selected itemje ki van kényszerítve a Termek osztály Termek\_id eleminek. Ezután a sql példánynak beállítom a parancsát ami a következő : "DELETE FROM `termek` WHERE `termek\_id`=@id" Ez a parancs paraméteresen van meg irva hogy ne közvetlen az adatbázisba kerüljenek Különféle műveletek (Ez az sql injekció kivétése). Ez után a beviteli mezők átadják az értéküket a paraméteresen az adatbázisnak. Utána jön egy try catch rész ha nem sikerül az adatbázisból törölni akkor a program figyelmezteti a felhasználót egy Messageboxal a hibára egy adatbázis hibakóddal. A sikeres művelet esetén a connection példány meg nyílik és a sql Executenonquery végrehajtódik (Megtörténik a termék törlése az adatbázisból. a művelet végzése után le zár a connection példány a close tulajdonsággal és a Termegmegjelenit metódus lefut a listbox friss tartalma miatt.

# Felhasználói dokumentáció

A program windows rendszer kompatibilis (Windows 7, 8, 10) Hardver követelmények ennek megfelelőek.

A program indulásnál a belépés ablak jelenik meg ami a nevet és a jelszót kéri öntől amit a rendszergazda ad meg a programnak ha nem tudja adatait akkor forduljon hozzá.

Ha sikeresen bejelentkezet akkor indíthatja a programot. A program funkciói a következők.

* Termékek megjelenítése
* Új termék hozzáadása az adatbázishoz.
* A termék adatai módosítása az adatbázisból.
* A termék törlése az adatbázisból.

## Termékek megjelenítése

A főablak lista területe jeleníti meg az árukat ha valami hiba adótik kiírja a hibakódot és a következő szöveget *forduljon a rendszergazdához*. A lista ablakba jelenek a jelenlevő termék neve ára és mennyisége ha az ablak indításnál az árú mennyisége nulla vagy az alatti akkor kiírja a program hiánycikként kérem jegyezze meg és módosítsa vagy törölje a terméket.

## Új termék hozzáadása az adatbázishoz.

Az új termék hozzáadásához töltse ki a termék kezelés kereten belüli mezőket az EAN kód mezőnél csak számot adjon erre ügyeljen. Sorszám mezőnél nem adható érték az csak a termék felvétel szerinti sorszáma abból könnyen vissza kereshető a tulajdonság kereten belüli mezőket nem kötelező kitölteni mert ezek csak a égitestekre érvényesek. Az Áru felvételhez nyomja meg Árut felvesz gombot.

## A termék adatai módosítása az adatbázisból.

A termék módosításához válaszon ki egy elemet a listából ez által meg jelenik a termék összes adata a kitöltési mezőkbe amit átírhat ez után nyomja meg az Árut módosít gombot. És már módosult is az áru tulajdonságai az adatbázisba ha nem sikerült a művelet hiba üzenetet kap vissza

## A termék törlése az adatbázisból.

A terméktörléshez válasszon ki egy elemet a listából és nyomja meg az árut töröl gombot. Ha valami gond adódna a törlés folyamat közbe hiba üzenetet kap vissza

# További Fejlesztési lehetőségek

## Nem teljesített funkciók

* A beerkezesek tábla feldolgozása beszállítók meg jelenítése az adott idö pont lerögzítése a beérkezések mennyiségének felvétele majd a készlethez való hozzáadása

## További fejleszthető funkciók

* A felhasználók külön kezelése a bejentkezve marad idő számon tartása

# Összegzés

Most szembesültem először az informatika mélyebb szintjeivel és azt tapasztaltam ide már egy sajátságos látásmód szükséges, ami meg tanulható. Ezzel az új gondolkodás móddal amit a szoftverfejlesztéssel szereztem gondolok a objektum orientált szemlélet módra változik az ember szemlélete a világra ezáltal jobban átlátja a dolgokat az ember sokkal kisebb szeleteket és át lát. Ez a vizsgaremek munkám tudom nem ki dolgozott még hiányságok is vannak benne de próbálkoztam eleget tenni a kitűzött célnak.