Veszprémi Szakképzési Centrum Ipari Szakgimnáziuma

Szakképzés neve:

Szoftverfejlesztő

OKJ száma:

54 213 05

SZAKDOLGOZAT

Szakdolgozat címe:

MűhelyApp

Vastag Atila Hanich Kristóf

Témavezető 13. C

Veszprém 2020

Tartalom

[A program általános specifikációja 3](#_Toc35768949)

[Rendszerkövetelmények 3](#_Toc35768950)

[Hardverkövetelmények 3](#_Toc35768951)

[Szoftverkövetelmények 3](#_Toc35768952)

[A program telepítése 4](#_Toc35768953)

[Fejlesztői Dokumentáció 5](#_Toc35768954)

[Témaválasztás indoklás 5](#_Toc35768955)

[Az alkalmazott fejlesztői eszközök 5](#_Toc35768956)

[Microsoft Visual Studio 2019 5](#_Toc35768957)

[Biztonsági kérdések 8](#_Toc35768958)

[Adatbázis modell 8](#_Toc35768959)

[Részletes feladatspecifikáció, algoritmusok 9](#_Toc35768960)

[Tesztelési dokumentáció 9](#_Toc35768961)

[Továbbfejlesztési lehetőségek 9](#_Toc35768962)

[Források 9](#_Toc35768963)

Bevezetés

## Feladat rövid ismertetése

Felhasználói dokumentáció

# A program általános specifikációja

A program egy lézergravírozással foglalkozó cég rendeléseinek rögzítésére szolgál.  
A program felhasználói felületének kifejlesztésekor felhasználóbarát navigációra törekedtem.

Az első lapon rendelések leadására illetve listázására van lehetőség.  
A rendelés leadás egy ügyfélnévből, a vásárolt árucikkből, a vásárolt termék darabszámából, illetve az elkészítés határidejéből áll.  
A második lapon új terméket vihetünk fel az adatbázisba. A feltöltéshez szükséges adatok közé tartozik a termék neve, rövid leírás, ár, jelenleg hány darab van belőle, illetve képet is csatolnunk kell az adott termékről.  
A harmadik lap vásárlók rögzítésére szolgál legfontosabb adataikkal együtt: név, lakcím, telefonszám.   
A negyedik lap a Raktárkészlet módosítására szolgál.  
Az ötödik lapon a raktár található, ahol meg tudjuk tekinteni, minden termékünk összes adatát.

## Rendszerkövetelmények

A program futtatásához szükséges összetevők a következők:

### Hardverkövetelmények

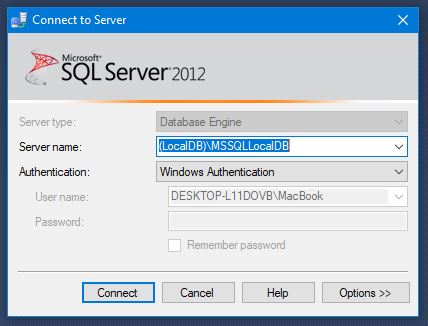
* Billentyűzet, egér vagy más mutató és szövegbeviteli eszköz
* Processzor: 2 x 1.6GHz
* Memória: 2GB
* Merevlemez: 500MB szabad hely
* 1024x768pixel felbontású monitor

### Szoftverkövetelmények

* Program:
  + .Net keretrendszer 4.7.2-os verzió
  + Windows 7
* Adatbázis:
  + Microsoft® SQL Server® 2012 Express
  + Windows 7

# A program telepítése

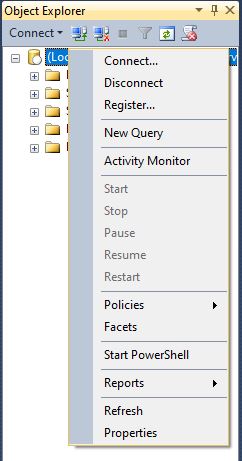
## Csatlakozás az adatbázishoz



1. ábra Kapcsolódás az adatbázishoz

1. Nyissa meg a Microsoft SQL Server Management Studio-t. A Connect to Server (Kapcsolódás a kiszolgálóhoz) párbeszédpanel jelenik meg.
2. Adja meg a következő információkat:

|  |  |  |
| --- | --- | --- |
| Beállítás | Ajánlott érték | Leírás |
| Server type | Database Engine | Csatlakozás típusa |
| Server name | (LocalDB)\MSSQLLocalDB | A teljes kiszolgálónév |
| Authentication | Windows hitelesítés | Hitelesítés típusa |

1. Kattintson a Connect gombra. Az Object Explorer ablak nyílik meg.

2. ábra Object Explorer

1. Az adatbázis felvételéhez kattintson jobb egérgombbal a (LocalDB)\MSSQLLocalDB elemre
2. Válassza a New Query opciót.
3. Illessze be ide a program könyvtárban található Adatbázis.sql fájlt.
4. Nyomja meg az Execute (végrehajtás) gombot.
5. Az adatbázis objektumainak megtekintéséhez bontsa ki az Databases csomópontot, majd bontsa ki a gravirozasDB elemet.
6. Az adatbázis elkészítése sikeres.

## MűhelyApp telepítése

# Fejlesztői Dokumentáció

## Témaválasztás indoklás

A témát azért választottam, mivel szabadidőmben lézergravírozással foglalkozom. Ahogy egyre jobban növekedtek a rendeléseink számai, egyre nehezebb volt papíron, illetve telefonon, jegyzetben vezetni a rendeléseket. Gyakran előfordult, hogy közösségi oldalunkon kerestek meg az emberek, emiatt szükségünk volt egy programra, amiben rögzíteni és rendszerezni tudjuk a rendeléseket, így nem fordulhat elő az, hogy egy rendelés elkeveredik és csúszunk a határidővel. Később mivel a programban rögzítjük vásárlóinkat és elérhetőségeiket, szeretném új funkcióval bővíteni a programot, ahol tudunk nyereményjátékot bevezetni, illetve kedvezményeket is tudunk biztosítani törzsvásárlóinknak.

## Az alkalmazott fejlesztői eszközök

### Microsoft Visual Studio 2019

A Microsoft által készített .NET keretrendszer (a .NET Framework) gyors alkalmazásfejlesztést, platformfüggetlenséget és hálózati átlátszóságot támogató szoftverfejlesztői platform. A keretrendszert a korábbi platform, a COM leváltására szánták. Eredetileg a .NET kifejezés nemcsak fejlesztői környezetet jelentett, hanem fejlesztőeszközök, szoftverek, sőt hardvereszközök összességét is. Az évek során a kép kitisztult, így mostanra a .NET alatt a keretrendszert értjük. A .NET Framework eszköztára a szoftverfejlesztés szinte minden aspektusát (kliens-, illetve szerveroldali megoldások, adatbázisok kezelése, játékfejlesztés stb.) lefedi.

A CLI-t úgy tervezték, hogy bármilyen objektumorientált programozási nyelvet támogasson, megosztva egy közös objektum modellt és egy nagy, közös osztálykönyvtárat. A .NET jelenleg több mint 40 programozási nyelvet támogat, melyek többsége ingyenes (a kereskedők fejlesztői környezeteket árulnak). Sok nyelvet jelentősen hangoltak, hogy illeszkedjen a .NET keretrendszerbe. A gyártók ezt kihasználva gyakran egyéb nyelvi funkciókat is módosítottak.

Beépített nyelvek

* C#, a C++-hoz és a Sun Java nyelvéhez hasonló objektumorientált nyelv
* JScript .NET, Microsoft JScript nyelvének fordított verziója
* J#, a Java és a J++ programozók átmeneti nyelve a .NET keretrendszer felé
* Managed C++, a C++ egy változata a .NET platformra
* Visual Basic .NET, a klasszikus Visual Basic egy továbbfejlesztett, objektumorientált, többszálas verziója

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

(<https://hu.wikipedia.org/wiki/Microsoft_Visual_Studio>)

A program elkészítéséhez a következő beépülő modulok voltak szükségesek:

* System.Data.SqlClient v4.8.1
* MaterialDesignColors v1.2.2
* MaterialDesignThemes v3.0.1
* NETStandard.Library v2.0.3

### .NET keretrendszer

A Microsoft által készített. NET keretrendszer (.NET Framework) gyors alkalmazásfejlesztést, és hálózati átlátszóságot támogató szoftverfejlesztői platform. A .NET Framework eszköztára a szoftverfejlesztés szinte minden színterét (kliens-, szerveroldali megoldások, adatbázisok kezelése, játékfejlesztés stb.) lefedi.

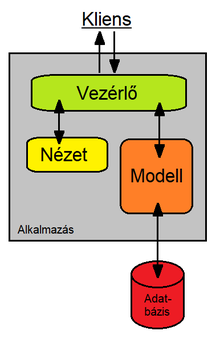
(<https://hu.wikipedia.org/wiki/.NET_keretrendszer>)

### GitHub Desktop

A GitHub egy elosztott verziókezelő rendszer. Más központosított rendszerekhez képest (pl. SVN) itt nem egy központi repository-ba dolgoznak a fejlesztők. E helyett mindenkinek saját lokális repository-ja van. Ide teszi bele a módosításokat, majd ha kész van egy részfeladat, a munkáját beszinkronizálja egy központi repository-ba.

### Advanced Installer

Az Advanced Installer egy Windows Installer készítő eszköz. Ennek a programnak köszönhetően egy könnyen használható felülettel egyszerűen készíthetünk programunknak telepítőt akár EXE, akár MSI kiterjesztéssel.

A modell-nézet-vezérlő (MNV) (angolul model-view-controller) a szoftvertervezésben használatos szerkezeti minta. Összetett, sok adatot a felhasználó elé táró számítógépes alkalmazásokban gyakori fejlesztői kívánalom az adathoz (modell) és a felhasználói felülethez (nézet) tartozó dolgok szétválasztása, hogy a felhasználói felület ne befolyásolja az adatkezelést, és az adatok átszervezhetők legyenek a felhasználói felület változtatása nélkül. A modell-nézet-vezérlő ezt úgy éri el, hogy elkülöníti az adatok elérését és az üzleti logikát az adatok megjelenítésétől és a felhasználói interakciótól egy közbülső összetevő, a vezérlő bevezetésével.

Gyakori egy alkalmazás több rétegre való felbontása: megjelenítés (felhasználói felület), tartománylogika és adatelérés. Az MNV-ben a megjelenítés tovább bomlik nézetre és vezérlőre. Az MNV sokkal inkább meghatározza egy alkalmazás szerkezetét, mint az egy programtervezési mintára jellemző. Modell Az alkalmazás által kezelt információk tartomány-specifikus ábrázolása. A tartománylogika jelentést ad a puszta adatnak (pl. kiszámolja, hogy a mai nap a felhasználó születésnapja-e, vagy az összeget, adókat és szállítási költségeket egy vásárlói kosár elemeihez). Sok alkalmazás használ állandó tároló eljárásokat (mint mondjuk egy adatbázis) adatok tárolásához. Az MNV nem említi külön az adatelérési réteget, mert ezt beleérti a modellbe. Nézet Megjeleníti a modellt egy megfelelő alakban, mely alkalmas a felhasználói interakcióra, jellemzően egy felhasználói felületi elem képében. Különböző célokra különböző nézetek létezhetnek ugyanahhoz a modellhez. Vezérlő Az eseményeket, jellemzően felhasználói műveleteket dolgozza fel és válaszol rájuk, illetve a modellben történő változásokat is kiválthat. Az MNV gyakran látható webalkalmazásokban, ahol a nézet az aktuális HTML oldal, a vezérlő pedig a kód, ami összegyűjti a dinamikus adatokat és létrehozza a HTML-ben a tartalmat. Végül a modellt a tartalom képviseli, ami általában adatbázisban vagy XML állományokban van tárolva. Habár az MNV-nek sok értelmezése létezik, a vezérlés menete általánosságban a következőképp működik:

* A felhasználó valamilyen hatást gyakorol a felhasználói felületre (pl. megnyom egy gombot).
* A vezérlő átveszi a bejövő eseményt a felhasználói felülettől, gyakran egy bejegyzett eseménykezelő vagy visszahívás útján.
* A vezérlő kapcsolatot teremt a modellel, esetleg frissíti azt a felhasználó tevékenységének megfelelő módon (pl. a vezérlő frissíti a felhasználó kosarát). Az összetett vezérlőket gyakran alakítják ki az utasítás mintának megfelelően, a műveletek egységbezárásáért és a bővítés egyszerűsítéséért.
* A nézet (közvetve) a modell alapján megfelelő felhasználói felületet hoz létre (pl. a nézet hozza létre a kosár tartalmát felsoroló képernyőt). A nézet a modellből nyeri az adatait. A modellnek nincs közvetlen tudomása a nézetről.
* A felhasználói felület újabb eseményre vár, mely az elejéről kezdi a kört.

A modell és a nézet kettéválasztásával az MNV csökkenti a szerkezeti bonyolultságot, és megnöveli a rugalmasságot és a felhasználhatóságot.

* Microsoft SQL Server Management Studio 2012

## Biztonsági kérdések

A program háromrétegű architektúrára épül, ennek köszönhetően biztonságosan kommunikál a felhasználói felület egy szerveren keresztül az adatbázissal.

A háromrétegű architektúra egy szoftvertervezési minta, a szoftverarchitektúrán felül. John J. Donovantól származik. A modell legnagyobb előnye, hogy lehetővé teszi az egyes rétegek egymástól függetlenül történő fejlesztését, sőt, akár teljes cseréjét is, lépést tartva így a folyamatosan változó követelményekkel és az egyre újabb technológiákkal. Ez biztonságosan megtehető, mert egy réteg módosítása nincs hatással a többi réteg működésére. Egymástól független modulokként tartalmazza a felhasználói felületet, az üzleti logikát és az adatbázist a szükséges hozzáférési műveletekkel. Például ha a megjelenítésért felelős szerverre új operációs rendszer kerül, akkor elég csak a megjelenítést vezérlő modulokat frissíteni az alkalmazásunkban, az új körülményeknek megfelelően. Ennek a fejlesztési technikának köszönhetően később mobilalkalmazás is készíthető a már meglévő adatbázishoz és backendhez.

## Adatbázis modell

A program elkészítéséhez Microsoft SQL nyelvet használtam.

A rendelések eltárolásához 3 táblára van szükség.  
A bal oldali táblában eltároljuk az árucikk fontos adatait.

|  |  |  |
| --- | --- | --- |
| Mezőnév | Típus | Leírás |
| Id | INT | Áru egyedi azonosítója |
| Nev | NVARCHAR(255) | Áru neve |
| Leiras | TEXT | Áru leírása néhány mondatban |
| Mennyiseg | INT | Jelenleg hány darab van raktáron |
| Ar | INT | A termék ára |
| Kep | NVARCHAR(255) | A termék képének elérési útja |

A jobb oldali tábla az ügyfelek adatainak tárolására szolgál.

|  |  |  |
| --- | --- | --- |
| Mezőnév | Típus | Leírás |
| Id | INT | Az ügyfél egyedi azonosítója |
| Nev | NVARCHAR(255) | Ügyfél neve |
| Cim | NVARCHAR(255) | Ügyfél lakcíme |
| Telefonszam | NVARCHAR(20) | Ügyfél telefonszáma |

A középső tábla egy vásárlás adatait tárolja.

|  |  |  |
| --- | --- | --- |
| Mezőnév | Típus | Leírás |
| ID | INT | A rendelés azonosítója |
| UgyfelID | INT | Vásárló azonosítója |
| AruID | INT | Áru azonosítója |
| Datum | DATETIME | Vásárlás dátuma |
| Hatarido | DATETIME | Mikorra kell a rendelésnek elkészülnie |
| Darab | INT | Rendelt mennyiség |
| TeljesAr | INT | Fizetendő összeg |

## Részletes feladatspecifikáció, algoritmusok

A programom alkalmas ügyfelek, termékek és rendelések tárolására és kezelésére.

### Funkcionális követelmények

* Rendelések felvétele: lehessen rögzíteni ki, mikor, mit vásárol milyen mennyiségben, mikorra kell rendelésének elkészülnie, illetve átvételkor mennyit fog fizetni.
* Miután rögzítettünk egy rendelést, fontos, hogy tudjunk róla listát készíteni, ahol könnyen áttekinthető minden, ami a rendeléssel kapcsolatos.
* Új termék felvitele: Rögzítenünk kell a termék fontos tulajdonságait: Termék neve, rövid leírás, termék ár, raktárkészleten levő mennyiség, illetve egy kép az adott termékről, hogy megkönnyítse a tájékozódást, melyik termékről van is szó.
* Kép tallózásakor csak .jpg illetve .png kiterjesztésű képeket tudunk tallózni, más kiterjesztésű állományt nem.
* Ügyfél felvitelekor az ügyfél három alapvető tulajdonságát tároljuk: név, lakcím, telefonszám.
* Az ügyfelek listázására ezen a lapon is szükség van, amennyiben esetleg egy ügyfél nem jön a rendeléséért, el tudjuk érni.
* A raktárkészlet feltöltése menüpontban lehetőségünk van kiválasztani egy terméket és a darabszámán változtatni.
* A raktár menüpontban megtekinthetjük az összes termékünket az összes információjával és képével együtt.

### Nem funkcionális követelmények

* Rendelés leadáskor minden mező ellenőrizve van, például nem lehet rendelést úgy leadni, hogy nem választunk ki ügyfelet, vagy megrendelt terméket. Emellett nem lehet a rendelés napjától régebbi dátumot sem kiválasztani határidőnek.
* Termék felvitelkor sem lehet egy mezőt sem üresen hagyni, minden mező ellenőrizve van, illetve darabszámnak és árnak sem lehet megadni negatív értéket, vagy szöveget. Hiba fellépése esetén hibaüzenet tájékoztat, és pirosan jelöli a hibás mezőt.
* A telefonszám alapértelmezett értéke a +36, ezzel is megkönnyíti a program a telefonszám beírását.
* Új rendelés vagy ügyfél felvitelekor a listába azonnal feltöltődik a frissen felvitt adat.

## Tesztelési dokumentáció

A program minden fülön, ahol adatot viszünk be, vagy legördülő listából választunk, ellenőrzi, hogy minden adat megvan-e adva, illetve ki van-e választva. Továbbá a bevitt adatokat is ellenőrzi, például nem vihetünk be rendeléskor negatív értéket vagy nullát darabszám kiválasztásakor.

## Továbbfejlesztési lehetőségek

A program felépítése úgy lett megalkotva, hogy viszonylag egyszerűen, kevés módosítással lehessen belőle webalkalmazást készíteni a Model View Controller architektúrának köszönhetően.

## Források

<https://stackoverflow.com> (megtekintve: 2020. 03. 28.)

[https://www.wpf-tutorial.com](https://www.wpf-tutorial.come) (megtekintve: 2020. 03. 22.)

<https://hu.wikipedia.org/wiki/Modell-nézet-vezérlő> (megtekintve: 2020. 03. 22.)

<https://support.office.com/> (megtekintve: 2020. 03. 22.)

<https://hu.wikipedia.org/wiki/Microsoft_Visual_Studio> (megtekintve: 2020. 03. 22.)

<https://docs.microsoft.com/hu-hu/azure/sql-database/sql-database-connect-query-ssms> (megtekintve: 2020. 03. 22.)

<https://github.com/MaterialDesignInXAML/MaterialDesignInXamlToolkit> (megtekintve: 2020. 03. 22.)

<https://www.nuget.org/packages/System.Data.SqlClient> (megtekintve: 2020. 03. 22.)