**Bevezetés**

Szakdolgozatom célja egy termelő- és kereskedő vállalatok számára specializálható, grafikus felhasználói felülettel rendelkező adatbáziskezelő szoftver létrehozása volt, amellyel az említett vállalkozások raktározási és logisztikai ügyeit egyszerűen, átláthatóan, informatikai előképzettség nélkül lehessen kezelni.

A szoftverrel szemben támasztott alapvető követelmények:

* grafikus megjelenési felület,
* könnyen értelmezhető, felhasználóbarát megjelenés,
* gombok és menük alkalmazása, parancssor mellőzése
* egy adatbázissal történő valós idejű, oda-vissza irányú kommunikáció,
* az adatbázishoz kapcsolódás felhasználónévvel és jelszóval történő védelme (beléptetőrendszer),
* felkész- és késztermékek közös táblában történő tárolása, megkülönböztetésük egy mező értékével,
* bekerülési (bruttó) árból számított nettó, eladási és akciós ár számítása előre meghatározott matematikai képletekkel,
* a számított értékek védettsége,
* tetszés szerint választható „akciós” termékek nyilvántartása és lehetőség csak ezen termékek megjelenítésére,
* események részletes naplózása,
* napló védelme (utólag nem módosítható, nem törölhető),
* a megjelenített adatok közötti, egyszerre több érték szerinti szűrésének lehetősége,
* keresés az adatbázis elemei közt, egyszerre több mező értékének figyelembevételével,
* a megjelenített adatok tetszés szerinti rendezésének lehetősége,
* a megjelenített táblázatok futási időben történő, tetszés szerinti átrendezése,
* a felhasználó által bevitt adatok ellenőrzése mind adatbázis, mind szoftver oldalról,
* esetleges helytelen/hibás beviteli adatok esetén a hibás mező(k) jelölése, segítség a helyes szintaktikához,
* a program, az adatbázis és az adatkapcsolat által létrejött hibák kezelése.

Az alkalmazás elméleti jelentősége az iránymutatás a vállalatok részére, hogy a jelenleg is zajló, negyedik ipari forradalom jegyében lépjenek előre, és zárkózzanak fel a digitális technológia vívmányaival karöltve a hatékonyabb és kevesebb emberi munkavégzés érdekében, a logisztika, raktározás és a valós időben történő nyomkövetés használatával, akár személyi számítógépeken, akár mobil eszközökön. Megelőző kutatásaim és személyes tapasztalataim alapján tisztában vagyok vele, hogy digitális bevándorlóként sokkal nehezebb átlátni és megérteni egy számítógépes program működését, de éppen ezért támasztottuk a programmal szemben azt az alapvető követelményt, hogy könnyen értelmezhető, felhasználóbarát kezelőfelülettel rendelkezzen, úgy, hogy egy informatikában nem jártas személy is megfelelően tudja használni.

A szoftver gyakorlati jelentősége az említett elméleti irány megvalósítása, tehát jelen szoftver alkalmazása vállalati szinten az előrehaladás és felzárkózás érdekében, gazdasági és technológiai okokból kifolyólag, az Ipar 4.0 elveit követve, jelen esetben nem a gyártáskövetésre értelmezve, hanem a készleten lévő áruk raktározásának és esetleges mozgásának követésére.

Szakdolgozatom elméleti és gyakorlati jelentősége rám tekintve, hogy az eddig tanult, jellemzően fiktív adatokra épülő szoftveres megoldásokat már létező, valós problémakörre tudtam kiterjeszteni. Ehhez nagyban hozzájárult az, hogy az egyetemi alapképzés kereteiben megfelelő szintű tudást sajátítottam el ahhoz, hogy valós adatokkal dolgozó szoftvereket készítsek, amiken gyakran nem kis felelősség van.

Az adatbáziskezelő szoftver jelenleg – mint a témában is jelölve van, – a termelő- és kereskedő vállalatok részére specializált, de az alapvető üzleti logika és szoftveres implementáció nem csak ezen területnek felel meg, igény szerint más területen is alkalmazható, többek közt szolgáltató, pénzügyi, fuvarozó, vendéglátó és logisztikai vállalatoknál is.

Az alkalmazás C# nyelven készült, Microsoft .NET <<#>> használatával, az adatbázist pedig az Oracle Express Edition <<#>> biztosítja. A szoftver alkalmas mind helyi, mind távoli adatkapcsolat létrehozására és használatára, valamint az események részletes, védett naplózására is.

**Elmélet**

**C#**

A program készítésekor C# nyelvet használtam, választásom azért esett erre, mert már egyetemi tanulmányaim előtt is használtam ezt a nyelvet, és a BSc alatt is ez volt a választott objektum-orientált programozási nyelvem, valamint a témavezetőm és általam kijelölt célok megvalósítására is alkalmas.

A C# egy erősen típusos, normatív, objektum-orientált programozási nyelv, amelynek alapjául a C++ és a Java szolgált. A fejlesztéskor a Microsoft a C++ hatékonyságát, a Visual Basic kezelhetőségét és a Java platform-függetlenségét próbálta ötvözni, amely sikerült is, 2000-ben, a Professional Development Conference-en mutatták be először a nyilvánosság előtt, és megjelenése óta már a hetedik verziónál jár. Minden verzió számos újítással érkezett, többek között a párhuzamos programozás támogatásával. Megjegyzendő, hogy a C# fejlesztésének vezetője a kezdetektől az az Anders Hejlsberg, aki a Turbo és Borland Pascal, valamint a Delphi létrehozásakor főmérnökként dolgozott a Borlandnál. A nyelv olyan széles körben alkalmazott, hogy néhány hónappal megjelenése után, 2001 decemberében már szabványosították, erről az Európai informatikai és kommunikációs rendszerek szabványosítási szövetsége ECMA-334 kódnévvel ellátott publikációjában olvashatunk, valamint 2003 óta ISO szabvány is lett, erről pedig a Nemzetközi Szabványügyi Szervezet ISO/IEC 23270:2006 szabványának hivatalos kiadványában találunk bővebb információt.

**.NET**

A szoftver futtatásához szükséges, feljebb említett keretrendszer a Windows Update-ből, és a MSDN-ről is elérhető, valamint a mellékelt lemezen is megtalálható a ##### könyvtárban.

A már említett, 2000-es Professional Development Conference-en a C#-pal együtt a .NET keretrendszer is megjelent, szükséges a C#-ban írt programok futtatásához. A Visual Studio 6.0 megjelenése óta a Microsoft nem a Win32 környezetben, hanem a .NET környezetben futtatja programjait, utalva ezzel a hálózati munka integrálására, amit az általam készített program is használ. A keretrendszer a használt fejlesztőkörnyezetnek fordítási idejű szolgáltatásokat végez, és az így lefordított alkalmazásoknak futási idejű környezetet biztosít, valamint az alkalmazások számára a már nem használt objektumok memóriabeli felszabadítását automatikusan elvégzi a Garbage Collector (szemétgyűjtő) használatával. A fordító a forráskódot nem natív, hanem egy köztes kódra fordítja le. Ez a köztes kód az MSIL (Microsoft Intermediate Language), ezt továbbfejlesztve, a Visual Studio 2015-ben hivatalosan is megjelent a .NET Compiler Platform (Roslyn), ami a Visual Basic-hez és C#-hoz készült új, fordítói platform, ami közvetlenül a kódanalízist szolgálja, valamint készíthető vele különálló Form alkalmazás, amely a forráskódkezelő eszközökhöz kapcsolódva forráskódok megfigyelésére, statisztikáinak létrehozására használható.

A keretrendszer az alkalmazások típusi szerint több csoportba sorolja az osztálykönyvtárakat, most csak azokat említem, amelyeket érint az alkalmazásom:

* ADO.NET: Adatbázis alkalmazások, és a hozzájuk tartozó driverek menedzselése és használata, valamint az adatkapcsolat kiépítése és működtetése
* Windows Form: Windows alapú felhasználói könyvtár, grafikus felhasználói felülettel rendelkező ablakok létrehozásához

A .NET keretrendszert is szabványosították, ez az Európai informatikai és kommunikációs rendszerek szabványosítási szövetségénél az ECMA-335 kódnevet kapta és a Nemzetközi Szabványügyi Szervezet az ISO/IEC 23271:2012 kódszámmal jelölte, amikből egyértelműen látszik, hogy a C# és .NET szoros kapcsolatban állnak, kódszámaik közvetlenül egymást követik.

**Microsoft Visual Studio 2015**

Elsősorban az objektum-orientált szemléletmód és az Oracle adatbázissal történő megfelelő, robosztus működés volt a szempont a fejlesztői környezet kiválasztásakor. A környezet és a driverek megfelelő beállításainak elvégzése után a kapcsolat szilárd és a Visual Studio is megfelelően kezel minden, Oracle-lel kapcsolatos metódust.

A Visual Studio 2015 már több, mint 15 nyelvet támogat, és már nem csak Windows platformra lehet fejleszteni, hanem többek közt Azure, Android, iOS rendszerekre is, amellyel a alkalmazásom továbbgondolása esetén, és a jelenleg zajló negyedik ipari forradalom jegyében az adatbázis (legyen szó konkrétan készletezésről, leltárról, vagy bármilyen logisztikai műveletről) egy mobil eszközről elérhető úgy, hogy közben a felhasználó fizikailag ott van a raktárban, és az eszköz segítségével, egyedi azonosítókat használva valós időben, valós adatokkal dolgozhat, ide értve a készletfeltöltést – akár vonalkód szkenneléssel –, ellenőrzést, keresést, kivételezést.

**Oracle**

A tanulmányaim alatt az Adatbáziskezelés I-II. tantárgyakból az Oracle Express Edition adatbáziskezelő rendszert használtam, így számos funkcióját megismerhettem, megtanultam a kliens oldal használatát, különböző eljárásokat. A szakdolgozatomhoz szükséges szerver oldal konfigurációját pedig magam kellett elsajátítanom, a rendelkezésre álló dokumentációból (megjegyzendő, hogy az Oracle dokumentációi kiemelkedően jók és részletesek), így végül egy jól működő rendszert tudtam használni úgy, hogy mind a szerver, mind a kliens oldal működését a megfelelő szinten elsajátítottam.

Az Express Edition az Oracle egyik ingyenes adatbáziskezelője, amely nem csak konzolos, hanem webes interfésszel is rendelkezik, így a fejlesztés is könnyebb, és a felhasználók is relatíve könnyen, egyszerűen kezelhetik az adatbázist. Néhány dologhoz szükséges némi parancssori ismeret, de mivel a szoftver nagyon jól dokumentált, az esetleges nehézségek is megoldhatóak. Mivel az Oracle rendelkezik olyan illesztőprogramokkal, amelyek a fejlesztőkörnyezettel kommunikálnak, így a webes felület elhagyható, elegendő egy ablakban történő kliens-szerver kapcsolat is, amely mellett - többek közt – a jóval kisebb memóriahasználat és az alacsonyabb erőforrásigény szól.

<<< ide egy benchmark web vs alkalmazás >>>

**Dokumentáció**

**Összefoglalás**

<<ide az ipar4.0-hoz kapcsolódó fejlesztési lehetőségeket>>

**Irodalomjegyzék**

Illés Zoltán: *Programozás C# nyelven*, Jedlik Oktatási Stúdió, Budapest, 2005

ECMA International: *ECMA-334 C# Language Specification*, **5** (2017)

ECMA International: *ECMA-335 Common Language Infrastructure*, **6** (2012)

International Standard Organization: *ISO/IEC 23270:2006(E)*, 2nd Edition, 2006

International Standard Organization: *ISO/IEC 23271:2012(E)*, 3rd Edition, 2006

Oracle: *Oracle Database Express Edition Getting Started Guide*, 2014

Microsoft Docs: *C# Guide*, https://docs.microsoft.com/en-us/dotnet/csharp

Microsoft Docs: *.NET Guide*, https://docs.microsoft.com/en-us/dotnet/standard