

Tartalomjegyzék

Bevezetés.....	3
1. Feladat specifikáció	6
1.1. Adatszerkezet.....	6
1.1.1. Tárolt adatok.....	6
1.1.2. Adatbázis táblák	9
1.1.3. Táblák közötti kapcsolatok.....	10
1.1.4. Felületnél használt objektumok.....	12
1.1.5. Frame-ek közötti kapcsolatok.....	14
1.1.6. Java osztályok.....	15
1.1.7. Az adatbázis és a felület közötti kapcsolat létrehozása.....	17
2. Felhasználói dokumentáció.....	18
2.1. Menü kategóriái.....	19
2.1.1. Adat feltöltés	19
2.1.1.1. Alkatrész adatainak felvitele	20
2.1.1.2. Új alkatrész adatainak felvétele.....	21
2.1.1.3. Új VTSZ szám felvétele	22
2.1.1.4. Cég adatainak felvétele.....	23
2.1.1.5. Szállító adatainak felvétele.....	23
2.1.1.6. Tartozások rendezése.....	25
2.1.2. Adat módosítás	26
2.1.2.1. Alkatrész adatainak módosítása.....	26
2.1.2.2. VTSZ adatainak módosítása.....	27
2.1.2.3. Cég adatainak módosítása	28
2.1.3. Keresések.....	29
2.1.3.1. Alkatrész keresése	29
2.1.3.2. VTSZ szám keresése	30
2.1.3.3. Cégek adatainak keresése	31
2.1.4. Adat törlés	32
2.1.4.1. Alkatrész adatainak törlése.....	32
2.1.4.2. VTSZ adatainak törlése	33
2.1.4.3. Cég adatainak törlése.....	34
3. A szoftver továbbfejlesztési lehetőségei	35
4. Összegzés	36
5. Irodalomjegyzék	38
Könyvek.....	38

Jegyzetek	38
Web címek.....	38
6. Függelék.....	39
6.2. Ábrajegyzék.....	39
7. Köszönetnyilvánítás.....	40

Bevezetés

Szakedolgozatnak adatbázis-kezelésből, egy alkatrész nyilvántartó alkalmazásfejlesztését választottam. Azért döntöttem az adatbázis-kezelés mellett egyrészt, mert tanulmányaim során ez keltette fel legjobban az érdeklődésemet. Másrészt pedig tapasztalatokat akartam gyűjteni ezen a téren, és ha lehetőség nyílik rá, akkor az egyetem elvégzése után ezen a szakterületen szeretném kamatoztatni a megszerzett tudásomat egy cégnél.

A témám pedig, azért esett egy alkatrész nyilvántartó elkészítésére, mert az édesapám egyéni vállalkozó MTZ, IFA és egyéb gépalkatrészek eladásával foglalkozik, ezért elég sok gondot okoz neki az alkatrészek nyilvántartása. Ezzel a szoftverrel neki szeretnék segítséget nyújtani. Remélem, hogy használatával, kevesebb erőfeszítéssel könnyebbé, egyszerűbbé, gyorsabbá tehetem a munkáját. Ez a program segítséget nyújt többek között az alkatrészek adatainak tárolásában, mennyiségének áttekintésében, a tartozások, szállítók, cégek adatainak és vásárlók adatainak a nyilvántartásában.

Édesapám gyakorlatban szerzett tudása és remek tanácsai sokat segítettek a szoftver megírásában, így jobban átláttam, milyen adatokat kell tárolnom, és milyen problémákat kell megoldanom. Ha pedig valamilyen gondom volt programírás közben, akkor meg tudtam beszélni vele.

Ugyanakkor szerettem volna egy felhasználóbarát könnyen kezelhető, átlátható rendszert létrehozni, amit később a gyakorlatban is használnak. Remélem, hogy ha a programom elkészül, akkor a vállalkozásban megfelelően tudjuk majd használni, és sok felesleges munkát, időt takarít meg.

Amikor már megterveztem a feladatul kitűzött programomat és át kellett térnem a gyakorlati megvalósításra, akkor nagyon tanácstalan voltam. Mert rengeteg szoftver van manapság a piacon, és nem tudtam, hogy melyikkel lenne a legcélszerűbb dolgoznom. Ekkor megkértem a témavezetőmet, akinek sokkal nagyobb rálátása van ezekre a dolgokra, hogy ajánljon néhányat és mondja el róluk a véleményét. Ő kellő körültekintéssel meg is tette ezt, és ezután a döntés csak rajtam állt. Végül azért döntöttem a következő két szoftver mellett, mert elég

széles körben alkalmazzák őket és reméltem, hogy ha szerzek egy kis gyakorlatot a használatuk során, akkor ez segíteni fog a munkavállalásnál.

A szakdolgozat megvalósításában az Oracle 10g Express Edition adatbázis-kezelő rendszert és Oracle JDevelopert használtam. A következőkben, néhány szóban bemutatom a két szoftvert, és szeretném megosztani a személyes tapasztalataimat is, hátha ezzel többen kedvet kapnak a programok kipróbálásához vagy esetleg hosszabb távú alkalmazásához.

Azért Oracle programot választottam az adatok tárolásához és menedzseléséhez, mert ez az egyik vezető relációs adatbázis-kezelő rendszer. Számos olyan helyen használják, ahol fontos az adatok gyors elérése és az adatbiztonság. 1979-től van a piacon, DBMS¹ termékeinek legutóbbi néhány verziója az Oracle7, Oracle8i, Oracle9i és az Oracle10g nevet viselik. A legutóbbi három verzió már objektumrelációs, internetes és osztott adatbázis-kezelési lehetőségekkel is el van látva.

Az Oracle Express Edition egy jól strukturált grafikus felülettel segítette a munkámat. Könnyebbséget jelentett, hogy nem kellett az egyes utasításokat parancssorban megadni, hanem elég volt a megfelelő mezőket kitölteni. De az összes adat megadása után meg tudtam tekinteni a parancsot, amelyet az általam megadott paraméterek alapján hozott létre. Viszont, ha mégis parancssoros felülettel szerettem volna dolgozni, akkor erre is adott lehetőséget.

Az Oracle JDeveloper 10g fejlesztőeszköz előnye többek között az, hogy számos operációs rendszeren teljes mértékben támogatja a nyílt szabványokon alapuló fejlesztést, és beépített funkcióival optimalizálható a Java alkalmazások teljesítménye. Az eszköz egységes, integrált fejlesztőkörnyezetet nyújt a Java, XML és SQL alapú fejlesztéshez, az UML segítségével történő modellezéshez, valamint a J2EE alapú webszolgáltatások létrehozásához.

A felület kialakításánál én azért választottam a JDeveloper-t, mert ez a program a Java nyelvet használja. Én pedig az egyetemi éveim során többek között megtanultam a Java nyelv alapjait és többször alkalmaztam különböző kurzusok során. Ezért a felület létrehozásánál csak a megszerzett ismereteimet kellett felelevenítenem. A JDeveloper-ben nagyon tetszett, hogy az ablakok elkészítésénél a felületet úgy lehetett megtervezni, mintha egy rajzoló programot használnék. Így egyből látszott az összhatása az objektumoknak. A kód beírásánál

¹ Database Management System: adatbázis-kezelő rendszer

pedig csak elég volt egy másik fülön a programban a megfelelő részeket kitölteni, mert az objektumokhoz kapcsolódó általános részeket már automatikusan legenerálta a szoftver.

Az Oracle JDeveloper 10g egyébként ingyenesen letölthető az Oracle fejlesztői közössége, Oracle Technology Network webhelyéről. Ez a [10] pontban megtalálható az irodalomjegyzékben.

A fejlesztés során sok tapasztalatot gyűjtöttem és sok nehézséggel megküzdöttem, amelyek inspiráltak. Remélem, hogy édesapám könnyen fogja tudni kezelni az általam megírt szoftvert, és legalább annyit segít neki a munkájában, mint amennyit segített nekem a tapasztalatszerzésben.

1. Feladat specifikáció

Most nagy vonalakban ismertettem, hogy mit valósítottam meg az alkatrész nyilvántartó programban. A felület elkészítésénél törekedtem az egyszerűsége, átláthatóságra és a felület elemeinek konzisztenciájára.

A szoftverben lehetőség van az alkatrészek, a cégek, a szállító és a VTSZ adatainak felvételére, módosítására, törlésére, keresésére. Az alkatrészek a bolt árukészletét, a cégek a boltban vásárlókat reprezentálják. A VTSZ a vámtarifa számot jelenti, és minden alkatrészhez kapcsolódik egy VTSZ szám.

Egy alkatrész megvásárlásakor egy szállító készül, amely tartalmazza a termékek, a vevő illetve az eladó adatait, a vásárlás dátumát és azt, hogy mely termék lett fizetve. Közben természetesen csökken a raktárkészlet az eladott termékek darabszámával.

Az előzőekben már említett Oracle 10g Express Edition-t alkalmaztam az adatbázis elkészítésénél, mely grafikus felületen keresztül segítette elő a munkámat, a felületet pedig Oracle JDeveloper-ben készítettem.

A továbbiakban először megadom az adatszerkezetet, melyet az adatbázisban használtam az adatok tárolására, utána pedig a felületnél használt objektumokat írom le.

1.1. Adatszerkezet

Az adatszerkezetek kialakításánál törekedtem arra, hogy az adatokat minél kevesebb redundanciával tároljam és kiküszöböljem az inkonzisztenciát.

1.1.1. Tárolt adatok

A mezők létrehozása közben nagy figyelmet fordítottam arra, hogy az alkatrészekről és a vásárláshoz kapcsolódó információkról minden le legyen tárolva az adatbázisban.

Az alkatrészeket cikkszámmal azonosítom, ami nem csak az adatbázisban, hanem a gyakorlatban is használható. Ezt úgy gondolom, hogy az alkatrészekre egy boltban elég mondjuk a cikkszámukat ráírni, és az árat ez alapján ki lehet keresni az adatbázisból. A gyakorlati alkalmazás azért is jó, mert amikor feltöltik a raktárkészletet egy olyan áruval, aminek megváltozott az ára, akkor elég csak az adatbázisban megadni a változást a megfelelő mezőkben. És a raktárban lévő árukon pedig maradhat a cikkszám, mivel az nem változik.

Az alkatrészeknél még fontos dolog a mértékegység megadása, ezt azért tettem egy külön mezőbe bele, mert vannak olyan mennyiségek, amit például darabban, literben, kilóban, m²-ben számolnak és ezt a számlákon és szállítókon is meg szokták adni. Így csak egyszer kell ezt megadni egy új alkatrész adatának felvételekor, és bárhol fel lehet használni, ahol szükséges.

Amit még lényegesnek tartottam az áru adatainak tárolásánál, az alkatrész elhelyezésének megadása, amely inkább a felhasználónak nyújt segítséget. Amikor egy vevő például, keres egy árut, akkor azt ki lehet keresni a program segítségével. Az adatok között ekkor megjelenik a *helye* nevű mezőben egy információ, hogy a raktár melyik szegletében, vagy a bolt melyik polcán található a keresett áru. Ennek a mezőnek a kitöltése egyébként nem szükséges, de célszerű.

Egy másik mező kitöltése is csak ajánlott adatfeltöltéskor, ennek neve *tipus*, ezt akkor célszerű megadni, ha egy alkatrész például egy bizonyos gépjárműhöz tartozik. Ha pedig az előzőekben már említett cikkszámot alkalmazzuk az alkatrészek árának beazonosítására, akkor szintén a cikkszám alapján azt is meg tudjuk állapítani, hogy milyen járműhöz jó. Ez főleg két egymáshoz közelálló típusú alkatrésznél hasznos, mint például, vannak olyan alkatrészek, amelyek MTZ80-ashoz és vannak, amik MTZ50-eshez készülnek és csak kis dolgokban térnek el. Ilyenkor sokkal könnyebb egy cikkszám alapján keresni a számítógépben az alkatrészhez tartozó típust, mint esetleg átlapozni egy könyvet, amely az adott jármű alkatrészeinek képeit tartalmazza.

A *fizetve* mező pedig azért került bele az adatbázisba, mert alkatrészeknél a vásárló nem mindig tudja eldönteni, hogy milyen az általa keresett alkatrész. Ez egy kicsit furcsán hangzik, de mondjuk, ha nem tudja csak a hozzávetőleges méreteit az alkatrésznek (pl.: csapágy vagy szimmering esetén milliméterek is számítanak), akkor nem biztos, hogy egyből meg lesz az a termék, amit keres. Ilyenkor van szerepe a *fizetve* mezőnek, mert nem kell minden alkalommal, ha egy alkatrészt elvisz a vevő a járműbe belepróbálni, azonnal fizetni. Hanem ilyenkor elég csak egy szállítót készíteni, ahol a *fizetve* mezőt csak 'n'-re kell állítani, ami azt jelenti, hogy nem fizették. És ha a próbálkozások után meglett az alkatrész, akkor csak azt kell kifizetni. Ezzel sok időt és felesleges tevékenységeket is meg lehet spórolni.

A következő táblázatban, néhány szóban leírtam az adatbázisban lévő mezők legfontosabb tulajdonságait.

	Mezőnév	mező típusa	mező hossz	integritási megszorítások	tárolt adatok megnevezése
1.	cikkszam	Varchar2	15	not null	alkatrész cikkszama
2.	megnevezes	Varchar2	50	not null	alkatrész megnevezése
3.	tipus	Varchar2	50		milyen járműhöz tartozik az alkatrész
4.	mennyiség	Number	10:2	not null	alkatrész raktáron lévő mennyisége
5.	mertegys	Varchar2	20	not null	az adott alkatrész mértékegysége
6.	netto	Number	10:2	not null	alkatrész nettó ára
7.	afa	Number	4:2	not null	alkatrész áfája
8.	brutto	Number	10:2	not null	alkatrész bruttó ára
9.	helye	Varchar2	100		az alkatrész helye a boltban
10.	vtsz_szam	Varchar2	15	not null	az alkatrész vámtarifaszama
11.	alktip	Varchar2	50	not null	milyen típusú a VTSZ számhoz tartozó alkatrész
12.	ceg_kod	Varchar2	15	not null	céget azonosító kód
13.	ceg_nev	Varchar2	50	not null	cég neve
14.	ceg_cim	Varchar2	50	not null	cég címe
15.	tel1	Varchar2	10		cég telefonszáma (1)
16.	tel2	Varchar2	10		cég telefonszáma (2)
17.	mobil1	Varchar2	11		cég mobilszáma (1)
18.	mobil2	Varchar2	11		cég mobilszáma (2)
19.	adosz	Varchar2	13		cég adószáma
20.	banksz	Varchar2	20		cég bankszámla száma
21.	aszl_kod	Varchar2	15	not null	a szállító és a vásárolt alkatrészek közötti kapcsolatot azonosítja
22.	mennyi	Number	10:2	not null	a vásárolt alkatrész mennyisége
23.	aszl_netto	Number	10:2	not null	a vásárolt alkatrész nettó ára
24.	aszl_afa	Number	4:2	not null	a vásárolt alkatrész áfája
25.	aszl_brutto	Number	10:2	not null	a vásárolt alkatrész bruttó ára
26.	fizetve	Char	1	not null	fizetve lett a vásárolt alkatrész
27.	sz_kod	Varchar2	15	not null	szállító kódja
28.	v_nev	Varchar2	50	not null	vásárló neve
29.	v_cim	Varchar2	50	not null	vásárló címe
30.	e_nev	Varchar2	50	not null	eladó neve
31.	e_cim	Varchar2	50	not null	eladó címe
32.	v_datum	Date		not null	vásárlás dátuma

1. ábra: Tárolt adatok leírásának táblázata

1.1.2. Adatbázis táblák

A táblák megtervezésénél lényeges volt számomra, hogy átláthatóak legyenek, és minden fontos adatot tároljanak, lehetőleg redundancia nélkül.

Egy kisebb redundanciát mégsem tudtam kiküszöbölni, ez a *Szallito* táblában van. A probléma abból adódik, hogy vevők adatait egy szállítón nem lehet megváltoztatni. Ezért nem egy cég kóddal adom meg a vevők adatait, mert azt a programban meg lehet változtatni a megfelelő ablakban. Hanem még egyszer letárolom a teljes nevet és címet, így a *Ceg* táblában történő módosítások nem befolyásolják a *Szallito* táblában lévő adatokat.

ALKATRESZ(cikkszam – **PK**, megnevezes, tipus, mennyisege, mertegys, netto, afa, brutto, vtsz_szam – **FK**, helye)

VTSZ(vtsz_szam – **PK**, alktip)

CEG(ceg_kod – **PK**, ceg_nev, ceg_cim, tel1, tel2, mobil1, mobil2, adosz, banksz)

ASZL(aszl_kod – **PK**, cikkszam – **FK**, mennyi, aszl_netto, aszl_afa, aszl_brutto, fizetve, sz_kod – **FK**)

SZALLITO(sz_kod – **PK**, v_nev, v_cim, e_nev, e_cim, v_datum)

Megjegyzés: A tábla és mező neveket a dokumentumban dőlt betűvel fogom írni és a tábla nevek kezdő karakterei nagybetűsek a jobb olvashatóság miatt.

Magyarázat:

PK – Primary Key (Elsődleges kulcs): Olyan oszlop(ok) alkotják, amelyben szereplő értékek a tábla sorait egyértelműen azonosítják. Egy vagy több oszlop alkothatja. Értékének (több oszlop esetén azok összevont értékének) egyedinek kell lenni. (Különböző sorokban lévő kulcsértékek különbözőek.) Nem lehet NULL értékű, összetett kulcs esetén egyetlen része sem. Egy táblának csak egy elsődleges kulcsa lehet.

FK – Foreign Key (Külső vagy idegen kulcs): Olyan oszlop(ok) alkotják, melyek egy másik táblával való logikai kapcsolatot valósítják meg. Ezen oszlop(ok) értéke adja meg, hogy az

adott tábla (gyerek tábla) egy sora, a másik tábla (szülő tábla) mely sorához tartozik. Egy vagy több oszlop alkothatja. A külső kulcs a gyerek táblában van és hivatkozik a szülő tábla elsődleges vagy egyedi kulcsára. A hivatkozott szülő tábla általában egy másik tábla, de lehet ugyanaz. A külső kulcs értéke csak létező szülő kulcs értékre hivatkozhat, vagy NULL értékű lehet. A szülő tábla olyan sora nem törölhető és olyan sorának kulcs értéke nem módosítható, melyre hivatkoznak gyerek sorok a gyerek táblában.

1.1.3. Táblák közötti kapcsolatok

A programban sok helyen szükség van a különböző táblákban lévő összetartozó adatokra, ezek egyidejű elérése csak a táblák összekapcsolásával lehetséges.

A táblák között háromféle kapcsolat kialakítása lehetséges: egy–egy, egy–több vagy több–több. (Esetleg ide lehet még sorolni negyediknek, ha a táblák között nincs kapcsolat).

Egy–egy kapcsolatnál, ha tekintünk egy **A** meg egy **B** táblát, akkor az **A** tábla egy sorához a **B** táblából legfeljebb egy sor kapcsolódik., és ez visszafele is igaz. Hétköznapi példát tekintve ez olyan, mint az emberek közötti házastársi viszony. Egy férfinak egy felesége lehet, és egy feleségnek egy férje lehet (a magyarországi szokásokat figyelembe véve).

A több–több kapcsolatnál az előző táblákat tekintve, az **A** tábla egy sorához, a **B** táblából több sor kapcsolódhat. Viszont a **B** tábla egy sorához az **A** tábla legfeljebb egy sora kapcsolódik. Újabb általános példát tekintve ez olyan, mint az anya gyerek kapcsolat. Egy anyának több gyereke is lehet, de minden gyereknek csak egy anyja van.

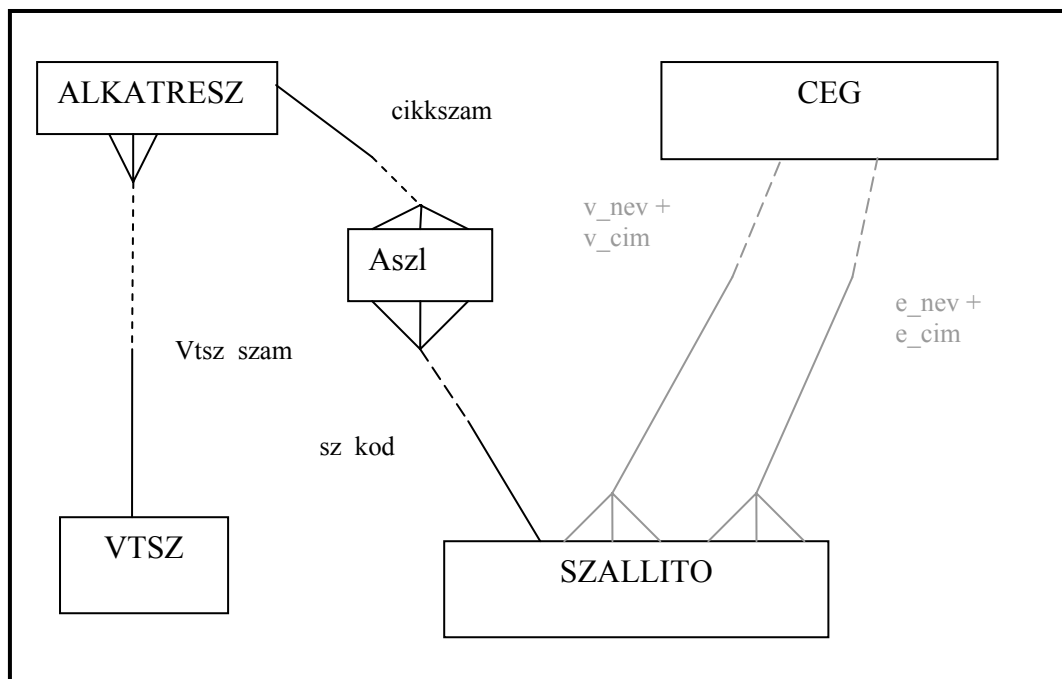
És végül a több–több kapcsolatnál az **A** tábla egy sorához a **B** táblából több is kapcsolódhat és a **B** tábla egy sorához, az **A**-ból is több rekord kapcsolódhat. Ez pedig köznapi példával élve olyan, mint a tanár diák viszony. Egy tanárnak több diákja is lehet, és ez visszafele is igaz, egy diáknak is lehet több tanára.

Az én programomban lévő adatbázis táblák között is van kapcsolat, ahogy azt a későbbiekben látható 2. ábra is mutatja. A kapcsolatok létrehozásában a táblák egy, vagy több mezőjének kombinációja vesz részt.

Az *Alkatresz* tábla a *VTSZ* táblával van egy-több kapcsolatban. A kapcsolat a *vtsz_szam* mező segítségével valósul meg. De az *Alkatresz* tábla egy másik táblával is kapcsolatban van, mégpedig az *Aszl* elnevezésével, itt a kapcsolat a *cikkszam* mező alapján jön létre, és ez is egy-több kapcsolat. Viszont az *Aszl* tábla is kapcsolatban van még egy másikkal a *Szallitoval*, ahol a kapcsolat az *sz_kod* nevű mezővel történik, és szintén egy-több típusú.

És tulajdonképpen a *Ceg* és a *Szallito* tábla között is egy-több kapcsolat van, amelyet a *ceg_kod* elnevezésű mező valósítana meg. De a külső kulcs megszorítás szabályai értelmében, ha egy cég adatait módosítanánk, akkor módosulnának a szállítón szereplő cég adatai is, ahogy ezt már az adatbázis táblák leírásánál is említettem. Viszont egy szállítón szereplő vásárló és eladó adatai a jogszabályok értelmében időközben nem változhatnak. Ezért ez csak redundanciával oldható meg úgy, hogy a *Szallito* táblába a *ceg_kod* helyett a cég neve és címe lesz eltárolva a *v_nev* illetve *v_cim* mezőkben. Az előző két mező tulajdonképpen meg is valósítja a kapcsolatot, de csak addig, amíg nem módosulnak a *Ceg* táblában lévő név és cím bejegyzések kapcsolatban résztvevő értékei.

Az elkövetkezőkben egy ábrán szemléltetem a különböző táblák közötti kapcsolatokat.



2. ábra: Táblák közötti kapcsolatok

1.1.4. Felületnél használt objektumok

A felületek elkészítésénél a Swingben Frameket használtam. A fő célom az volt, hogy minél egyszerűbben, gyorsabban lehessen felvenni az adatokat, és ne kelljen egy adatot többször megadni, ha nem szükséges. Ha valahol az adatbázisban lévő adatokból kell választani, ott listákat használok. Például, ha anyag beszerzés után az adatbázisba be szeretném vinni az alkatrészek adatait, akkor meg tudom nézni, hogy volt-e már ilyen alkatrész raktáron. Ha volt, kikeresem egy listából, és csak a mennyiséget és az árat módosítom (ha szükséges). Ha még nem volt raktáron, akkor felveszem egy másik ablakban a termék minden adatát.

A vásárlók adatait a *Ceg* táblában tárolom, és különböző felületeken, szükség esetén ugyanúgy listából lehet választani. Ha még nem szerepeltek az adatbázisban a megfelelő adatok, akkor egy másik ablakot meghívva fel lehet venni azokat. Ugyanez a helyzet VTSZ számok esetén is, erre azért törekedtem, mert így a program minden területén hasonló műveleteket végezhet a felhasználó az adatokkal, és nem kell olyan sokat tanulnia a program használatához. Ha például a cégek adatainak felvételét megtanulta, akkor az előzőhöz hasonlóan fel tudja venni, az alkatrészek illetve a VTSZ számok adatait is. Továbbá próbáltam az azonos funkciókat végrehajtó elemeket az ablakokon belül azonos helyeken elhelyezni, de közben figyelembe vettem a praktikusságot és az esztétikát is. Egy ablakon belül pedig, elválasztottam a különböző műveleteket végző egységeket egy szeparátorral. (Például, amikor egy ablakon belül lehet keresni és törölni, akkor egy vonal van a két tevékenységet ellátó elemek között.)

Az elkövetkezőkben ismertetem a Frame-eket, amelyeket létrehoztam és használtam a szoftverben:

(Megjegyzés: a Frame-ek neveit Tahoma betűtípussal írtam a szövegben az átláthatóság kedvéért.)

- **Alkatrész felvitele:** Az adatbázisban már szereplő alkatrészek adatait lehet frissíteni (pl.: alkatrész mennyisége, nettó ára, áfája). Ez azért hasznos, mert nem kell minden árubeszerzés után felvinni a már korábban árusított termék állandó adatait, mint például megnevezés, mértékegység, VTSZ szám, stb. Az itt megadott mennyiség

értékével növekszik a raktáron lévő mennyiség, az ár és az áfa pedig felülíródik a beírttal.

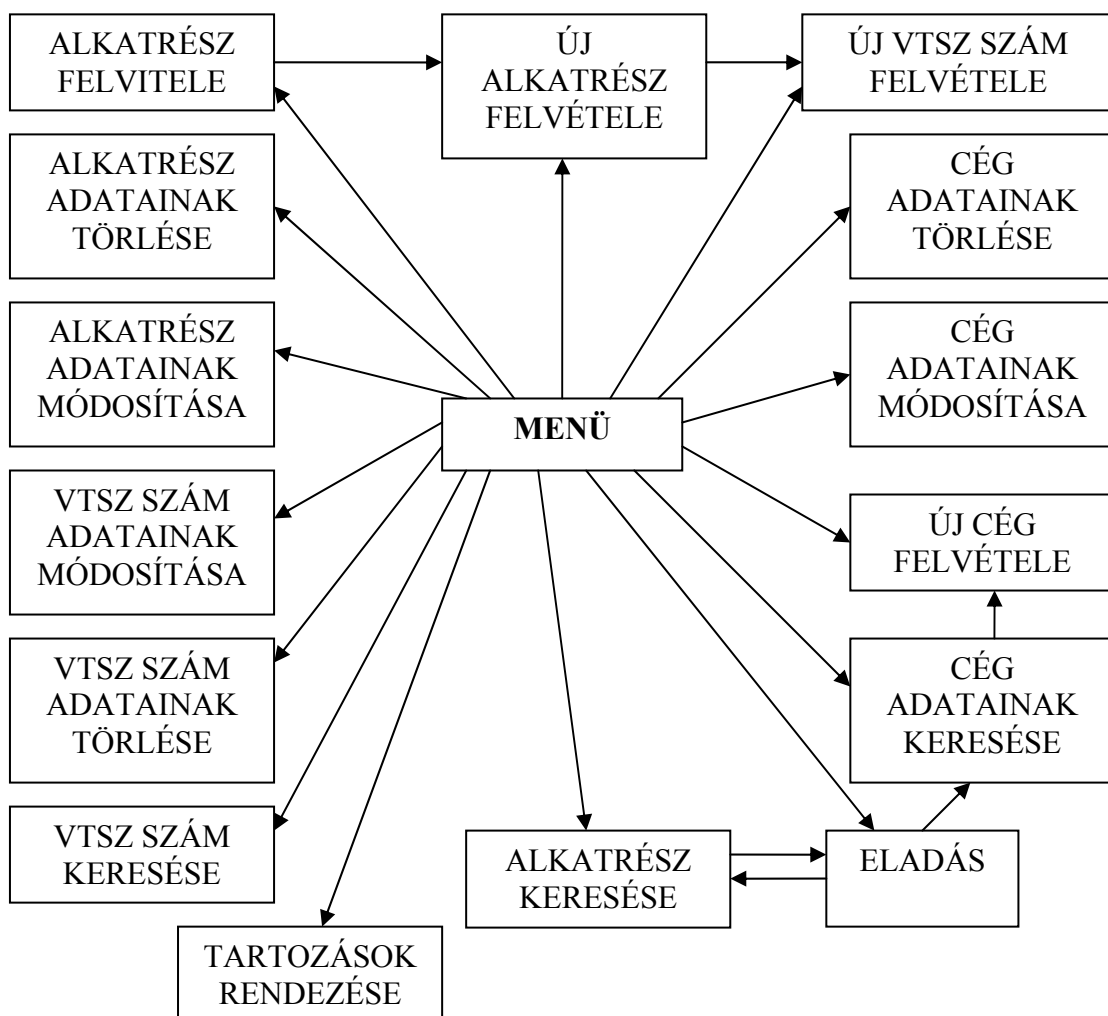
- Alkatrész keresése: Az alkatrészek között lehet keresni cikkszám és név szerint, de lehetőség van egy kiválasztott termék eladására is amely, ezáltal a szállítóra kerül.
- Alkatrész adatainak módosítása: A termékek adatait módosíthatjuk (pl.: a megnevezést, az árat, a raktáron lévő mennyiséget, stb.), és ki lehet keresni név vagy cikkszám szerint.
- Alkatrész adatainak törlése: Itt egy kiválasztott terméket lehet véglegesen törölni az adatbázisból, és természetesen itt is lehet keresni név és cikkszám szerint.
- Cég adatainak keresése: Cégek adatait lehet ki keresni és megtekinteni, név alapján lehet keresni is.
- Cég adatainak módosítása: Cégek adatait lehet kikeresni név szerint és módosítani (pl.: nevet, címet, telefonszámot, stb.). Módosítás során a név és cím adatok, nem maradhatnak üresen, szemben a többivel, amelyek kitöltése opcionális.
- Cég adatainak törlése: Itt cégek végleges törlésére van lehetőség, ha szükséges lehet keresni név szerint. Csak az összes adat törlése lehetséges.
- Eladás: Itt a kiválasztott tételekből lehet a kikeresett névre készíteni a szállítót, amelynek adatai beíródnak az adatbázis megfelelő tábláiba. Az elkészült szállító tartalmazni fogja azt is, hogy mely árucikk lett fizetve.
- Menü: A szoftver különböző tevékenységeit lehet kiválasztani, és hozzájuk kapcsolódó ablakokat meghívni.
- Tartozások rendezése: A szállítón szereplő nem fizetett áruk árát lehet kikeresni és kiegyenlíteni.
- Új alkatrész felvétele: A még eddig nem forgalmazott termékek adatait lehet felvenni.(Pl.: megnevezés, típus, mennyiség, mértékegység, stb.)
- Új cég felvétele: Az adatbázisban még nem szereplő cégek adatait lehet kitölteni és lementeni (Pl.: név, cím, telefonszám, mobilszám, adószám, bankszámlaszám). A név és a cím megadása kötelező, a többi opcionális.
- Új VTSZ szám felvétele: A VTSZ adatait adhatjuk meg (VTSZ szám, alkatrész típusa), minden érték megadása kötelező.
- VTSZ szám keresése: Kikereshetünk egy alkatrész típushoz tartozó VTSZ számot. Ez azért fontos, mert a program nem végzi el a számlázást, de a számlán viszont

szerepelnie kell a vámtarifa számnak, és ha kézzel írják a számlát, akkor ebben az ablakban könnyen kikereshetőek a megfelelő adatok.

- VTSZ adatainak módosítása: Egy kiválasztott VTSZ adatait módosíthatjuk, természetesen itt is lehetőség van alkatrész típus szerinti keresésre.
- VTSZ adatainak törlése: Egy kiválasztott VTSZ szám végleges törlésére van lehetőség, a keresés itt is megoldott.

1.1.5. Frame-ek közötti kapcsolatok

A következő ábra szemléletesen mutatja a kapcsolatokat és azok irányát az egyes ablakok között.



3. ábra: Frame-ek közötti kapcsolatok

Egy program működése során egyes ablakokban szükség lehet olyan információkra, amelyeket még nem tartalmaz az adatbázisunk, és nincs lehetőség adatfelvételre sem az aktuális ablakban. Vagy például a helyhiány miatt, nem férnek rá olyan objektumok az ablakokra, amelyekből a kért információ azonnal kinyerhető. Erre szolgáltat megoldást a Frame-ek (vagyis ablakok) közötti kapcsolatok kialakítása. Az én programomban a kapcsolatokat úgy oldottam meg, hogy az egyik Frame-ből a másikat egy gomb lenyomásával meg lehet hívni.

Lényegesnek tartottam még azt is, hogy minden ablakot el lehessen érni közvetlenül is a menü ablakból. Ez azért jó, mert ha a felhasználó csak egy tevékenységet akar végrehajtani (csak egy VTSZ számot akar felvenni), akkor nem kell neki több ablakon keresztül eljutnia a megfelelőhöz.

1.1.6. Java osztályok

Az adatbázisból érkező információk felületen való kezeléséhez és az adatbázis kapcsolat kiépítéséhez, még használok néhány Java osztályt.

Először olyan Java osztályokat akartam készíteni, amelyeknek az adattagjaik ekvivalensek lettek volna a megfelelő adatbázis táblák minden mezőjével. De később rájöttem, hogy nem minden adatra van szükségem a felületek kezelésénél. Így végül az osztályok kialakításánál csak a legszükségesebb mezőknek megfelelő adattagokat hoztam létre.

A programban használt Java osztályok a következők:

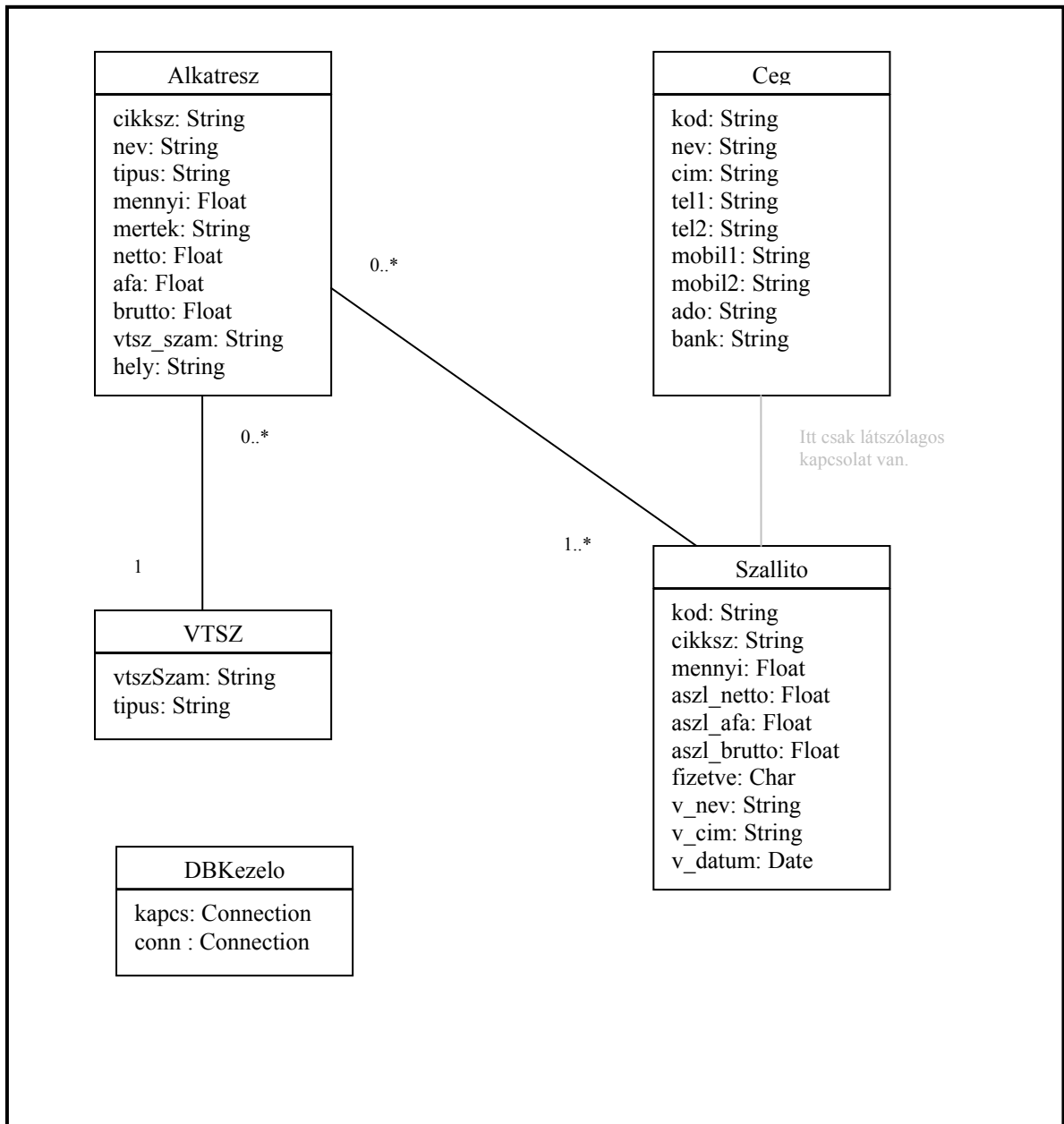
(Megjegyzés: a Java osztályokat is Tahoma betűtípussal adom meg a dokumentumban.)

- Alkatresz: Ennek példányaiban az *Alkatreszek* tábla rekordjának minden adatát letárolom.
- VTSZ: Ennek példányai pedig a *VTSZ* rekordjainak adatait tartalmazzák.
- Ceg: Ez a *Ceg* tábla rekordjának minden adatát tartalmazza.
- Szallito: Ez egy kicsit komplikáltabb, mert a felület kezelésénél nincs szükség a teljes *Aszl* és a *Szallito* táblában lévő adatokra és azok kapcsolatára. Ezért a *Szallito* osztály példányainak, az attribútumai a két táblában lévő oszlopokból adódnak. De nem

minden oszlopra van szükség. A Szallito osztály példány változói: kod, cikksz, mennyi, aszl_netto, aszl_afa, fizetve, v_nev, v_cim, v_datum.

- DBKezelo: Ez valósítja meg a kapcsolatot az adatbázis és a felület között.

A következő ábra az osztályokat és kapcsolatukat szemlélteti.



4. ábra: Java osztálydiagram

1.1.7. Az adatbázis és a felület közötti kapcsolat létrehozása

Az adatbázis és a felület megtervezésével és létrehozásával még nem fejeződött be a munka, mert ez a két egység nem tudott egymással kapcsolatba lépni és információt cserélni. Szükség volt egy áthidaló megoldásra.

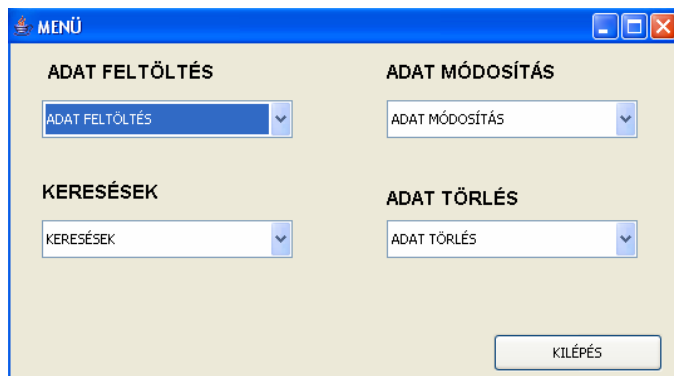
A kapcsolatot JDBC–vel oldottam meg. A JDBC tulajdonképpen egy Java programozói interfész SQL utasítások végrehajtására, amely az egyszerű adatbázis elérési lehetőségeken túl még tetszőleges soralapú adatforrás kezelésére is alkalmas.

A felület ablakaiban sokszor van szükség az adatbázis kapcsolat felépítésére és a megfelelő műveletek után a lebontására. Mivel elég bonyolult lenne minden ablakban felépíteni és lebontani a kapcsolatot, ezért hoztam létre egy külön osztályt, melynek neve *DBKeszelő* (amelyet a Java osztályoknál már megemlítettem), ennek az osztálynak a feladata, a program elindítása után az adatbázis kapcsolat kiépítése és menedzselése, a programban megadott paraméterek alapján. És program bezárása előtt, szintén ennek az osztálynak a feladata a kapcsolat szabályos lebontása.

A JDBC–ről több információt az irodalomjegyzék [1] pontjában megadott könyvben lehet találni.

2. Felhasználói dokumentáció

A program elindítása után, a MENÜ ablak jelenik, meg ahonnan a szoftver összes funkciója elérhető.



5. ábra: Menü ablak kezdeti állapotban

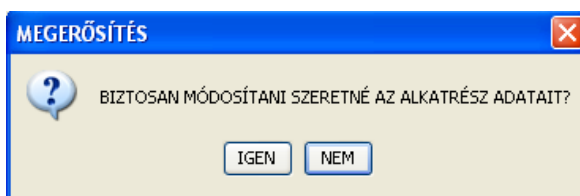
A funkciók négy csoportba vannak sorolva: adat feltöltés, adat módosítás, keresések, adat törlés. A különböző tevékenységek kiválasztásához, a megfelelő felirat alatti legördülő menűben lévő nyílra kell kattintani az egérrel. Ekkor megjelenik a többi tevékenység, ha kiválasztottunk egyet, akkor rákattintva az lesz a legördülő menű, az aktuális felirat. Ezután a legördülő menű feliratára kell kattintani és megjelenik a funkciónak megfelelő ablak. A legördülő menűben a csupa nagybetűvel szedett legelső felirat, csak a kategóriát jelzi, kiválasztás előtt ez az aktív, rákattintva nem történik semmi.

A MENÜ ablakon és mindegyiken, amelyik megjelenik, van KILÉPÉS gomb, ez mindenhol ugyanazt a funkciót tölti be, bezárja az aktuális ablakot és a következő ablak lesz az aktív. Az ablakok jobb felső sarkaiban a Windows-nál már megszokott három ikon van (piros X a teljes program bezárását teszi lehetővé, bármelyik ablakban vagyunk; a négyzet az ablak teljes méretűvé alakítására szolgál, ha pedig, nagyméretűvé alakult, akkor ezzel is lehet vissza állítani az előző méretre; míg _ jel a tálcára rakja le az ablakot) szerepelnek.

Ha az ablakokban nem megfelelően, hiányosan vagy rosszul töltünk ki adatokat, esetleg valami hibát okoztunk, akkor egy piros színű hibaüzenetben figyelmeztet a rendszer. De ha minden megfelelően zajlott és egy objektumhoz rendelt tevékenységet végrehajtotta a program, akkor kék színű üzenetben értesít minket az elvégzett munkáról.

Néhány ablakban, ha egy mező felett tartjuk az egérkurzort, akkor kiír információt a mező kitöltésére vonatkozóan. (Pl.: kereséseknél, ha * karaktert adunk meg keresési feltételnek, akkor a listákban az összes elem meg fog jelenni. Ezt ki is írja, ha a keresés mezők felett tartjuk egy ideig az egérkurzort.)

Minden fontosabb művelet végrehajtása előtt megjelenik egy dialógus ablak, amelyben a rendszer megerősítést vár a tevékenység végrehajtásához. Ezen az ablakon csak egy kérdés van meg két gomb, igen és nem. A két gomb közül valamelyiket ki kell választani, mert addig nem léphetünk tovább. Ha az igen gombot választjuk, akkor végrehajtódik a művelet, ellenkező esetben nem. Ilyen megerősítő ablak például a képen látható, amely alkatrész adatainak felvitelekor jelenik meg.



6. ábra: Megerősítést váró ablak

2.1. **Menü kategóriái**

Adat módosításnál az adatbázisban szereplő alkatrészek, cégek és a VTSZ adatait lehet megváltoztatni.

A keresések kategóriánál a menüben az alkatrészek, a vásárlók, és a VTSZ adatai között lehet keresni az adatbázisban.

Végül pedig az adat törléseknél szintén az alkatrészek, a cégek és a VTSZ szám adatait lehet, eltávolítani az adatbázisból.

A továbbiakban az egyes kategóriák menüpontjait ismertetem részletesen a menüből.

2.1.1. **Adat feltöltés**

A menüben adat feltöltésnél az alkatrészek, a cégek, a szállító, illetve a VTSZ adatait lehet felvenni, és a tartozások kiegyenlítése is itt lehetséges. Ez a kategória hat almenüpontot tartalmaz, ezek a következők: alkatrész adatainak felvétele, új alkatrész adatainak felvétele,

VTSZ adatainak felvétele, cég adatainak felvétele, szállító adatainak felvétele és tartozások rendezése.

2.1.1.1. Alkatrész adatainak felvétele

Ha az adat feltöltésnél választjuk az első menüpontot, amely az alkatrész adatainak felvétele, akkor meghívódik egy ablak, ahol az adatbázisban már szereplő alkatrészek adatait módosíthatjuk. Erre akkor van szükség, ha már az adatbázisban szereplő áru mennyiségét és árát, esetleg áfáját akarjuk megadni anyagbeszerzés után.

The screenshot shows a software window titled "ALKATRÉSZ FELVITELE". At the top, there is a list of components with the following data:

Cikkszám	Megn.	Leírás	db	Nettó	Áfa	Bruttó	VTSZ	Helye
26	csapágy	602 vegyes	19.0	600.0 Ft	20.0%	720.0 Ft	11-22-33	4.palc
88	CSAPÁGY	6203 VEGYES	10.0	1220.0 Ft	20.0%	1464.0 Ft	11-22-33	Hely...
30	gyorscsatlakozó	vegyes	5.0	2500.0 Ft	20.0%	3000.0 Ft	11-22-33	Helye: n...
22	12V 5W IZZÓ	VEGYES	61.0	50.0 Ft	20.0%	60.0 Ft	22-23-66	Helye: null
4	22K45X8 SZIMMERING		20.0	258.0 Ft	20.0%	309.6 Ft	22-33-56	Helye: null
2	6V 5W IZZÓ	VEGYES	60.0	50.0 Ft	20.0%	60.0 Ft	22-23-66	Helye: ELEKT...
28	8-as csavar	vegyes	1.0	100.0 Ft	20.0%	120.0 Ft	11-22-33	Helye: null

Below the list, there is a search section with the following fields and buttons:

- CIKKSZÁM KERESÉSE:
- ALKATRÉSZ NÉV SZERINTI KERESÉSE:
- ALKATRÉSZ MENNYISÉGE:
- ALKATRÉSZ NETTÓ ÁRA:
- ALKATRÉSZ ÁFÁJA:

Buttons: ÚJ ALKATRÉSZ FELVITELE, KERES, MÉGSE, FELVESZ, KILÉPÉS.

7. ábra: Alkatrész felvitele ablak kezdeti állapotban

A beszerzett áru adatainak beviteléhez, először ki kell választani a listából egy alkatrészt, amelyre ha rákattintunk, akkor az ára és az áfája betöltődik a megfelelő mezőkbe. Ezután, meg kell adni, hogy hány darab érkezett és esetleg, ha változott az ár vagy áfája, akkor a mezőkben lehet módosítani. Ha mindent kitöltöttünk, akkor a felvesz gombra kattintva, megjelenik egy megerősítő ablak, amely rákérdez, hogy tényleg végre szeretnénk-e hajtani a műveletet. A megerősítő ablak csak addig van nyitva, amíg nem kattintunk valamelyik gombjára. Ha a nemet választjuk, akkor nem történik semmi, ha pedig az igent választjuk, akkor az adatbázisban frissülnek az adatok. A raktárkészlet növekszik a megadott mennyiséggel, az ár és az áfa pedig felülíródik a mezőkben szereplő értékkel. Ha az árat és az áfát nem módosítottuk a mezőkben, akkor megmarad az eredeti érték az adatbázisban. A mezők nem megfelelő kitöltése esetén vagy esetleg, ha nem aktív a lista egyetlen eleme sem (nem választottunk ki alkatrészt, aminek az adatait frissíthetnénk), akkor egy üzenetben

figyelmeztet a rendszer. Hogy könnyebb legyen a listából kiválasztani a megfelelő alkatrészt, lehetőség van cikkszám és név szerinti keresésre. Egyszerre csak egy keresési feltétel adható meg, erre a rendszer is figyelmeztet. A keresés úgy történik, hogy az egyik (keresési) mezőbe megadjuk a keresési feltételt és a KERES gombra, kattintunk. Ekkor, ha jól lett megadva a paraméter, és van a keresésnek megfelelő eleme, akkor a listában csak a keresésnek megfelelő elemek fognak megjelenni. Keresési feltételként nem szükséges a teljes cikkszámot, vagy nevet megadni, elég csak a kezdő karaktert, vagy karakterek beírása. Ha a keresés után egyszer mégis szeretnénk az alkatrészek teljes listáját látni, akkor bármelyik keresési mezőbe * karaktert ütve, és a KERES gombra kattintva, visszakapjuk a teljes listát.

Ha az alkatrészek adatainak megadása közben rájöttünk, hogy például, nem jól töltöttünk ki több mezőt, akkor a MÉGSE gombra kattintva visszaáll az ablak eredeti állapotba.

Ezen az ablakon is van kilépés gomb, melynek hatására bezáródik

Lehetséges, hogy olyan áru adatait szeretnénk megadni az alkatrész felvitele ablakban, amelyet még eddig nem forgalmaztunk, ekkor az ÚJ ALKATRÉSZ FELVÉTELE gombra kattintva, megjelenik az Új alkatrész felvétele ablak.

2.1.1.2. Új alkatrész adatainak felvétele

Ezt az ablakot nem csak az alkatrészek felvitele ablakból lehet elérni, hanem közvetlenül meghívható a menü adat feltöltés kategóriájának második menüpontjából.

ALKATRÉSZ MEGNEVEZÉSE:	csapágy 6200
ALKATRÉSZ TÍPUSA:	vegyes
ALKATRÉSZ MENNYISÉGE:	5
ALKATRÉSZ MÉRÉTKEGYSÉGE:	db
ALKATRÉSZ NETTÓ ÁRA:	1500
ALKATRÉSZ ÁFÁJA:	20
ALKATRÉSZ VTSZ SZÁMA:	<input type="text"/>
ALKATRÉSZ HELYE:	<input type="text"/>

Nem jól vannak a mezők kitöltve!

MÉGSE FELVESZ KILÉPÉS ÚJ VTSZ

8. ábra: Új alkatrész felvitele ablak, ahol nem választottunk ki VTSZ számot

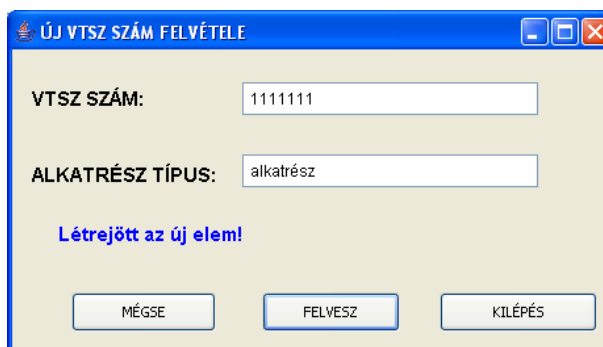
Az új alkatrész adatainak felvétele során meg kell adni a mezők értékeit a címkéjüknek megfelelően, és ki kell választani a legördülő menüből a VTSZ számot. Minden adatot kötelező megadni kivéve az alkatrész típusa és az alkatrész helye értékeket, melyek kitöltése opcionális. Ha az opcionális kitöltésű értékeken kívül elfelejtünk megadni egy értéket vagy nem választjuk ki a VTSZ számot (ahogy az a képen is látszik), akkor egy üzenetben figyelmeztet a rendszer. Ha mindent megfelelően kitöltöttünk, akkor a FELVESZ gombra kattintva az adatbázisba töltődnek az adatok, és egy üzenetben értesít minket az elem létrejöttéről. Ha már sok mezőt kitöltöttünk és közben rájöttünk, hogy valami nem megfelelő, akkor a MÉGSE gombra kattintva az ablak visszaáll eredeti állapotba.

Ha pedig a kitöltés során nincs az új alkatrésznek megfelelő VTSZ szám az adatbázisban, akkor az Új VTSZ gombra kattintva lehet ezt a problémát megoldani, amely meghívja a megfelelő ablakot.

Az ablakot szabályosan bezárni a KILÉPÉS gombbal lehet.

2.1.1.3. Új VTSZ szám felvétele

Ezt az ablakot közvetlenül az adat feltöltés harmadik almenüpontjából tudjuk meghívni.

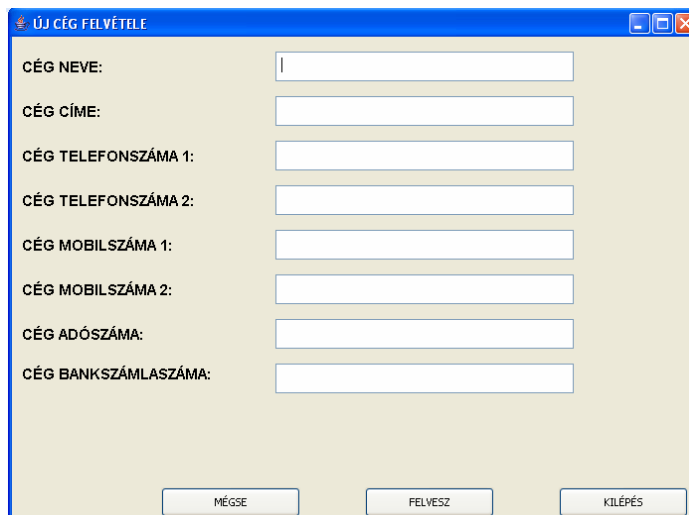


9. ábra: Új VTSZ felvétele ablak, amikor létrehozunk egy új elemet

Ha egy új vámtarifa szám adatait szeretnénk felvenni, akkor mindkét mező kitöltése kötelező a címkékben megadott értékeknek megfelelően. Ha ezt megtettük, akkor a FELVESZ gombra kattintás után, megvizsgálja a rendszer, hogy az adatbázisban szerepel-e a megadott VTSZ szám. Ha nem szerepel, akkor létrejön az új elem, és ezt egy kék színű üzenet tudatja velünk (ezt mutatja az ábra is.). Viszont, ha már benne van az adatbázisban a VTSZ szám, akkor piros üzenet jelzi a hibát. Ezen az ablakon is van MÉGSE és KILÉPÉS gomb, amelyek funkciója megegyezik az előzőekben már leírtakkal.

2.1.1.4. Cég adatainak felvétele

A Menü adat feltöltés funkciójának következő pontja a cég adatainak felvétele, ahol az új vásárlók adatait vehetjük fel.



10. ábra: Új cég felvétele ablak kezdeti állapotban

Ebben az ablakban a cég nevét és címét kötelező megadni, a többi adat kitöltése a felhasználóra van bízva. Ha ezt a két adatot nem adtuk meg, és a FELVESZ gombra kattintunk, akkor hibaüzenetben figyelmeztet minket a rendszer. Viszont ha minden szükséges mezőt kitöltöttünk és az adatbázisba szeretnénk menteni a cég adatait, akkor a FELVESZ gombra kattintva, megtörténik a mentés, és ezt egy üzenetben a gép is kiírja a számunkra.

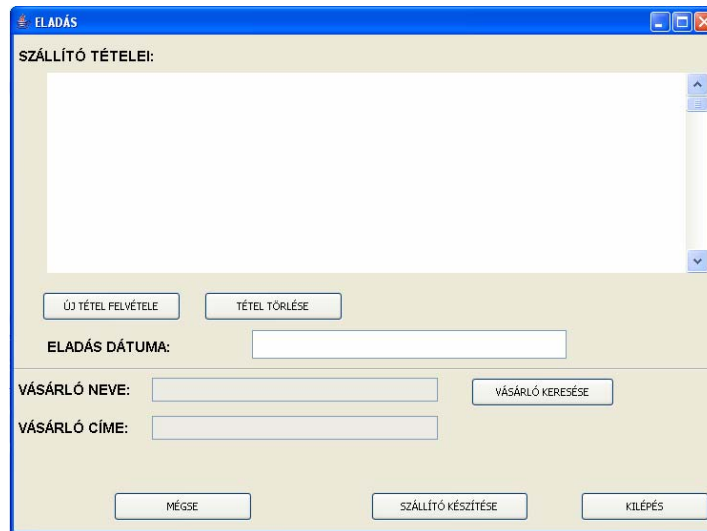
Ha már sok mezőt kitöltöttünk az ablakban, de közben rájöttünk, hogy nem jók az adatok, akkor a már előzőekben is alkalmazott MÉGSE gomb a megoldás, melynek hatására az ablak visszaáll a kezdeti állapotába.

A KILÉPÉS gomb itt is az ablakot zárja be.

2.1.1.5. Szállító adatainak felvétele

Az adat feltöltés legördülő menü utolsó előtti menüpontja a szállító adatainak felvétele. Itt a vásárolni kívánt termékek adatait, a vásárló adatait és a vásárlás dátumát adjuk meg. A szállító tételei listában a szállítóra kerülő alkatrészek adatai lesznek. Ha az ÚJ TÉTEL FELVÉTELE gombra kattintunk, akkor megjelenik az Alkatrész keresése ablak. Ebben ki lehet választani, annak az alkatrésznek az adatait, amelyet a vásárló megvásárol és az ablak bezárása után a kiválasztott árunak az adatait, bekerülnek a szállító tételei listába. Az

alkatrészkereső ablakot közvetlenül is el lehet érni a menüből, ezért a felépítését és a működését a későbbiekben ismertetem, a sorrendiség miatt.



11. ábra: Eladás ablak kezdeti állapotban

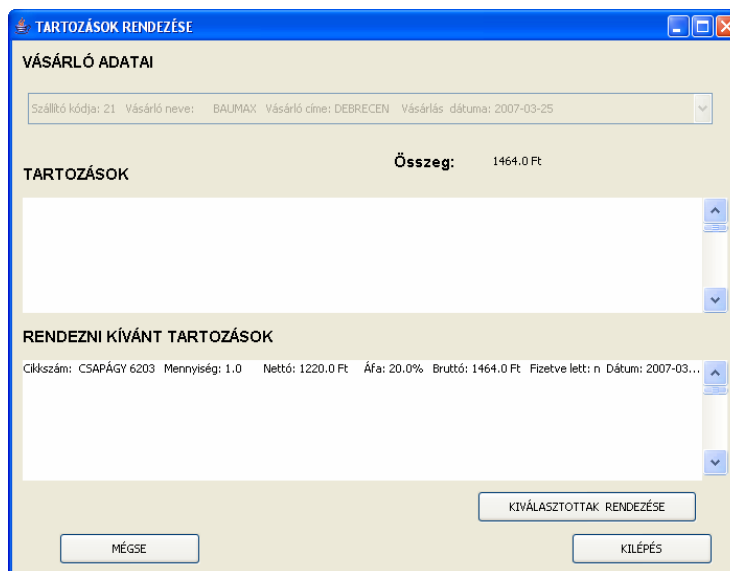
Ha valamelyik tételt, még sem akarjuk a szállítóra írni, mert például a vevő meggondolta magát, és nem szeretné már megvenni. Akkor a listából ki kell választani az elemet és a **TÉTEL TÖRLÉSE** gombra kattintva, egy felirat fogja jelezni, hogy a törlés megtörtént és raktárkészlet visszaáll a kiválasztás előtti darabszámra. Mikor már kiválasztottuk az összes alkatrészt, amelyet a vásárló megvásárol, meg kell adni a vásárlás dátumát (pl.: 07-MÁRC. - 25). A dátumnál fontos a formátum, három részből tevődik össze. Először az évet kell megadni két karakteren, azután a hónapot, rövidítve, betűkkel, ponttal a végén, utána szóköz is kell, azután pedig a napot, és minden rész közé kötőjelet kell írni.

Utána már csak a vásárló adatait kell kiválasztani az adatbázisból, a **VÁSÁRLÓ KERESÉSE** gomb segítségével, ami megnyitja a cégek keresése ablakot. A cégkereső ablak is elérhető közvetlenül a menüből, a későbbiekben ennek a működésére is kitérek a megfelelő menüpontnál. A vásárlók adatit közvetlenül nem lehet beírni vagy kiválasztás után átírni, csak azok a vevők vásárolhatnak, akiknek az adatai az adatbázisban vannak. Az adatbázisban nem tárolt vevők adatait a megfelelő ablakban fel kell venni.

A **MÉGSE** gomb, ugyanúgy működik, visszaállítja az ablakot eredeti állapotba. Ha minden ki lett töltve, akkor a **SZÁLLÍTÓ KÉSZÍTÉSE** gombra kattintva az adatok a rendszerbe kerülnek, miután a megerősítő ablakban az igen gombot választottuk.

2.1.1.6. Tartozások rendezése

Az adat felvételnél az utolsó menüpont a tartozások rendezése. Itt a szállítóra került, de még nem rendezett árukat lehet kifizetni.



12. ábra: Tartozás rendezése ablak, a rendezni kívánt tartozások kiválasztása után

Először ki kell választani a vásárlónak az adatait a legördülő menüből, itt csak azok a vásárlók szerepelnek, akiknek van olyan szállítójuk, amelyiken valamelyik tételt nem fizették ki. Ha kiválasztottunk egy vevőt, akkor egérrel rákattintva a tartozások listába bekerülnek azoknak a termékeknek az adatai, amelyeknek az árával tartozik. Ezek után ki lehet választani azokat az alkatrészeket a rendezni kívánt tartozások listából, amelyeket ténylegesen fizetni szeretne. Ezt azért oldottam meg két listával, mert ha egy vevő csak néhány tételt tud rendezni egy szállítójának a tételeiből, akkor ezt az első listából kiválasztva meg lehet tenni. Amikor a tartozások listából az első elem átkerül a rendezni kívánt listába, akkor a vevő adatait tartalmazó legördülő menü inaktívvá válik, hogy ne lehessen a szállítók adatait összekeverni. Ha úgy gondoljuk, hogy mégsem ennek a vevőnek a tartozásaival szeretnénk dolgozni, akkor a MÉGSE gombra kattintva visszaáll az ablak eredeti állapotába.

Ha minden rendezni kívánt tételt átraktunk a második listába, akkor a KIVÁLASZTOTTAK RENDEZÉSE gombra kell kattintani és a megfelelő tételek mezőiben regisztrálva lesz a fizetés.

Azt, hogy az ügyfélnek mekkora összeget kell fizetnie, az ablak jobb felső részében lévő összeg felirat melletti számjegyek mutatják.

Az ablakot bezárni természetesen itt is a KILÉPÉS gombbal lehet.

2.1.2. Adat módosítás

Ebben a legördülő menüben is ugyanúgy lehet kiválasztani az almenüpontokat, mint az előzőben. Itt csak három pont kiválasztása lehetséges: az alkatrészek módosítása, VTSZ módosítása és a cégek módosítása (természetesen a legelső csupa nagybetűs felirat itt is a kategória nevét mutatja, az átláthatóság kedvéért).

2.1.2.1. Alkatrész adatainak módosítása

Nézzük először az egyes pontban kiválasztható alkatrészek módosítása ablak működését.

The screenshot shows a software window titled "ALKATRÉSZ ADATAINAK MÓDOSÍTÁSA". At the top, there is a list of parts with columns for Cikkszám, Megn., description, DB, Nettó, Áfa, Bruttó, VTSZ, and Helye. Below the list are several input fields for modification: ALKATRÉSZ MEGNEVEZÉSE, ALKATRÉSZ TÍPUSA, ALKATRÉSZ MENNYISÉGE, ALKATRÉSZ MÉRTÉKEGYSÉGE, ALKATRÉSZ NETTÓ ÁRA, ALKATRÉSZ ÁFÁJA, VTSZ SZÁM (with a dropdown menu and "ÚJ VTSZ SZÁM" button), ALKATRÉSZ ELHELYEZÉSE (with a "MÉGSÉ" button), KERESÉS CIKKSZÁMSZERINT, and KERESÉS NÉVSZERINT (with a "KERES" button). At the bottom right, there are "MÓDOSÍT" and "KILÉPÉS" buttons.

13. ábra: Alkatrész adatainak módosítása ablak kezdeti állapotban

Itt a listából kell kiválasztani egy alkatrészt, melynek az adatai betöltődnek a megfelelő mezőkbe, ahol azokat a kívánt értékekre átírhatjuk, illetve VTSZ esetén beállíthatjuk. A listából való kiválasztást elősegítik a keresés mezők, melyek használata megegyezik az előző ablakokban is alkalmazott keresést megvalósító mezők működésével. Egyszerre csak egy feltétel szerint lehet keresni, és a lista elemei a keresést kielégítő alkatrészek lesznek. Így rövidebb lesz a lista és könnyebb lesz kiválasztani a módosítani kívánt elemet.

A MÉGSE gomb itt is ugyan azt a funkciót tölti be, mint az előzőekben, visszaállítja az ablakot a kiindulási állapotba.

Az adatok módosításánál figyelembe kell venni a mezők típusait, mert ha egy numerikus adatot váró mezőbe betűket írunk (pl.: alkatrész nettó ára: ötszáz) akkor az, hibát fog eredményezni. Még fontos az is, hogy az alkatrész típusa és az alkatrész elhelyezése mezők maradhatnak csak üresen, az összes többit ki kell tölteni, egyébként hibaüzenetet kapunk.

Ha mindent átírtunk, amit szeretnénk, és a módosítást az adatbázisban is végre akarjuk hajtani, akkor a MÓDOSÍT gombra kell kattintani. Ekkor megjelenik a fentebb már említett megerősítő dialóg ablakhoz hasonló és rákérdez, hogy tényleg végre szeretnénk-e hajtani a műveletet. Ha az igen gombot választjuk és minden adatot rendben talált a rendszer, akkor végrehajtja a módosítást, ellenkező esetben pedig nem.

Itt is lehetőség van új vámtarifa szám felvételére, ha nem találunk megfelelőt a listában, az ÚJ VTSZ SZÁM feliratú gomb segítségével, amely meghívja a megfelelő ablakot.

Az ablak becsukását most is a KILÉPÉS gomb végzi el.

2.1.2.2. VTSZ adatainak módosítása

A következő menüpont az adatok módosítása kategóriában a VTSZ adatainak módosítása. Ennek az a célja, hogyha a vámtarifaszám adataiban változás történik, akkor az adatbázisban lévő adatokat is ennek megfelelően módosítani tudjuk.

VTSZ ADATAINAK MÓDOSÍTÁSA

VTSZ ADATAI:

- VTSZ: 11-22-33 AlkTip.: alkatrész
- VTSZ: 22-23-66 AlkTip.: izzó
- VTSZ: 11-22-34 AlkTip.: csap
- VTSZ: 22-33-95 AlkTip.: szimmering
- VTSZ: 22-25-66 AlkTip.: lámpa
- VTSZ: 14-14-14 AlkTip.: MTZ
- VTSZ: nnnnnn AlkTip.: TEA

VTSZ SZÁM:

ALKATRÉSZ TÍPUS:

KERESÉS ALKATRÉSZ TÍPUS SZERINT:

14. ábra: VTSZ adatainak módosítása ablak kezdeti állapotban

Itt is egy listában vannak felsorolva a VTSZ adatai, és ebből lehet választani. A kiválasztást itt is segíti az alkatrész típusa szerinti keresés, amely hasonlóan működik az eddigiekhez. Ha keresni szeretnénk, akkor be kell írni a keresési mezőbe a feltételt, és a keres gombra kattintva, csak a feltételt kielégítő elemek maradnak a listában. Ha keresési feltételként viszont egy * karaktert adunk akkor a listába újra az összes elem belekerül.

Amikor már rákattintottunk a módosítani kívánt elemre, akkor a megfelelő adatai betöltődnek a mezőkbe. Itt kedvünkre módosíthatjuk azokat, de egyik mező sem lehet üres, amikor a MÓDOSÍT gombra, kattintunk (ez azért fontos, mert egyik érték sem lehet NULL).

Ha aktiváltuk a MÓDOSÍT gombot, akkor egy megerősítő ablak jelenik meg, ahhoz hasonló, mint amit már a kategóriák leírása előtt megadtam. Ebben egy kérdés van, hogy tényleg módosítani szeretnénk-e az adatokat, ha igen akkor az igen gombra kattintva a módosítások elmentődnek az adatbázisba. Ha nem gombot választjuk, akkor pedig a módosítás nem történik meg.

Bármilyen műveletet is végzünk az ablakban, a rendszer a tevékenységek végrehajtását és az esetleges hibákat egy üzenetben visszajelzi a felhasználó számára.

Az ablakot a KILÉPÉS gombbal zárhatjuk be.

2.1.2.3. Cég adatainak módosítása

Végül pedig az utolsó pont az adatszámítás kategóriában a cégek adatainak módosítása.

The screenshot shows a software window titled "CÉG ADATAINAK MÓDOSÍTÁSA". The window is divided into several sections. On the left side, there are eight input fields for entering company data, each with a label: "CÉG NEVE:", "CÉG CÍME:", "CÉG TELEFONSZÁMA 1:", "CÉG TELEFONSZÁMA 2:", "CÉG MOBILSZÁMA 1:", "CÉG MOBILSZÁMA 2:", "CÉG ADÓSZÁMA:", and "CÉG BANKSZÁMLASZÁMA:". Below these fields is a "MÉGSE" button. On the right side, there is a list box titled "CÉGEK ADATAI:" which contains a list of company records with columns for code, name, address, and phone numbers. At the bottom of the window, there is a search section with a label "KERESÉS NÉVSZERINT:", a search input field, and a "KERES" button. To the right of the search section are two buttons: "ADATOK MÓDOSÍTÁSA" and "KILÉPÉS".

15. ábra: Cégek adatainak módosítása ablak kezdeti állapotban

Mint a módosításoknál az eddigiekben, itt is listából lehet kiválasztani a módosítandó cégek adatait. Itt is lehetőség van keresésre a cégek neve alapján, és ugyanazzal a technikával történik egy feltételnek megfelelően a lista leszűkítése, ahogy az előző ablakoknál is leírtam.

Ha megvan a módosítandó elem, akkor egy ráirányuló egérgattintás hatására betöltődnek a megfelelő adatok a mezőkbe az előző módosításokhoz analóg módon.

Tetszőlegesen módosíthatunk minden adatot, de a végén a cég nevét és címét tartalmazó mezők nem maradhatnak üresen (ezek kitöltése kötelező).

Ennél az ablaknál is rákérdez, hogy tényleg végre szeretnénk-e hajtania a módosítást, igenlő válasz esetén, módosulnak az adatok, egyébként nem.

Itt is a KILÉPÉS gombbal lehet az ablakot bezárni.

2.1.3. Keresések

A következő kategória a keresések. Ebben a lenyíló menüben is három almenüpont van: az alkatrészek keresése, a VTSZ keresése és a cégek adatainak keresése.

2.1.3.1. Alkatrész keresése

Először nézzük az első almenüpontot, amely az alkatrészek keresése.

Cikkszám	Megn.	Nettó	Áfa	Bruttó	VTSZ	Helye
4	Megn.: 22X45X8 SZIMMERING	19,0Db	Nettó: 258,0 Ft	Áfa: 20,0%	Bruttó: 309,6 Ft	VTSZ: 22-33-56 Helye: null
28	Megn.: 8-as csavar vegyes	1,0db	Nettó: 100,0 Ft	Áfa: 20,0%	Bruttó: 120,0 Ft	VTSZ:11-22-33 Helye: null
26	Megn.: csapágy 602 vegyes	20,0db	Nettó: 600,0 Ft	Áfa: 20,0%	Bruttó: 720,0 Ft	VTSZ:11-22-33 Helye: 4.palc
88	Megn.: CSAPÁGY 6203 VEGYES	15,0DB	Nettó: 1220,0 Ft	Áfa: 20,0%	Bruttó: 1464,0 Ft	VTSZ:11-22-33 Helye: 4. POLC
24	Megn.: csapágy 702 vegyes	2,0db	Nettó: 500,0 Ft	Áfa: 20,0%	Bruttó: 600,0 Ft	VTSZ:11-22-33 Helye: 4.palc
30	Megn.: gyorscsatlakozó vegyes	5,0db	Nettó: 2500,0 Ft	Áfa: 20,0%	Bruttó: 3000,0 Ft	VTSZ:11-22-33 Helye: null
22	Megn.: 12V 5W IZZÓ VEGYES	60,0DB	Nettó: 50,0 Ft	Áfa: 20,0%	Bruttó: 60,0 Ft	VTSZ: 22-23-66 Helye: null
2	Megn.: 6V 5W IZZÓ VEGYES	50,0DB	Nettó: 60,0 Ft	Áfa: 20,0%	Bruttó: 60,0 Ft	VTSZ: 22-23-66 Helye: ELEKTROMOS POLC

16. ábra: Alkatrész keresése ablak, egy alkatrész kiválasztása után

Alkatrészkeresésnél is a tárolt adatok egy listában kerülnek elhelyezésre. A keresés kétféleképpen történhet az alkatrészek között, mégpedig név szerint és cikkszám szerint. De

egyszerre csak egyfajta keresés lehetséges, ez úgy történik, hogy valamelyik mezőben meg kell adni a keresési feltételt. A feltétel megadásánál nem szükséges a teljes cikkszámot vagy nevet megadni, de minimum egy karakter megadása kötelező. Ha kitöltöttük a megfelelő kereső mezőt, akkor a KERES gombra, kell kattintani, és a lista leszűkül azokra az elemekre, melyek eleget tesznek a feltételnek. De ha a keresés után újra az összes alkatrész adatára van szükségünk, akkor bármelyik kereső mezőben feltételként egy * karaktert megadva a KERES gomb aktivizálása után, az adatbázisban tárolt alkatrészeknek minden sora megjelenik a listában.

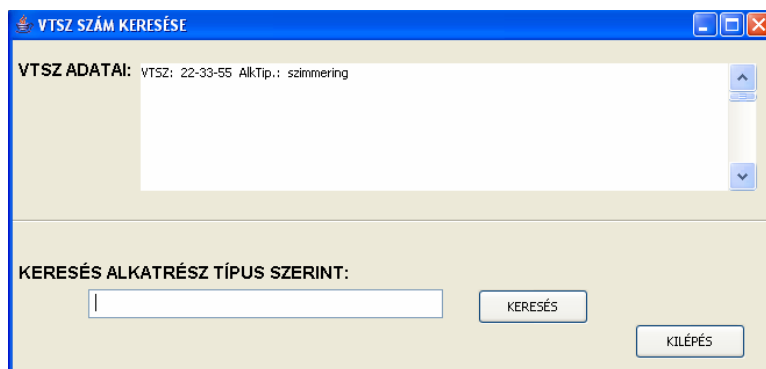
Ha már kikerestünk egy alkatrészt, és a vevő úgy gondolja, hogy meg szeretné vásárolni, akkor a listában az egérrel rákattintunk. Ekkor a termék ára betöltődik a nettó ár mezőbe, ez a mező az áru nettó eladási árát reprezentálja. Ezt meg is lehet változtatni, de mindenképpen tartalmaznia kell egy értéket. Ezután, meg kell adni, hogy mekkora mennyiséget kíván vásárolni a vevő a termékből, ha az elvitelkor fizeti, akkor be kell pipálni (egérkattintással) az „alkatrész fizetve lesz?” kiválasztó négyzetet. Ha a vásárláskor nem lesz fizetve az áru, akkor a kiválasztó négyzetben nem szerepelhet pipa.

Ha az értékeket kitöltöttük, illetve beállítottuk, és minden rendben van, akkor az ELADÁS gombra kell kattintani. Ekkor két dolog történik, először az árukészlet csökken a termék eladott mennyiségével, azután pedig aktivizálódik az eladás ablak (melynek működése az adat feltöltés megfelelő menüpontjánál, már ismertetésre került), és a szállító tételei közé bekerülnek az áru adatai, végül pedig bezárul az alkatrész keresése ablak.

Ha mondjuk a vevő meggondolta magát, mielőtt az ELADÁS gombot aktivizáltuk, akkor a MÉGSE gombra kattintva az ablak visszaáll kezdeti állapotba. Az ablak bezárására itt is a KILÉPÉS gomb szolgál.

2.1.3.2. VTSZ szám keresése

A második menüpont a keresések kategóriában a VTSZ szám keresése. Természetesen itt is egy listában vannak a vámtarifa számok adatai megjelenítve. Alkatrész típusa szerint van lehetőség keresésre. A keresés az eddigiekhez hasonlóan történik, minimum egy karakter megadása kötelező keresési feltételnek és a KERESÉS gombra kattintva a listában csak feltételnek megfelelő elemek maradnak. Ha * karakter a feltétel, akkor a lista az összes vámtarifa számot tartalmazza.



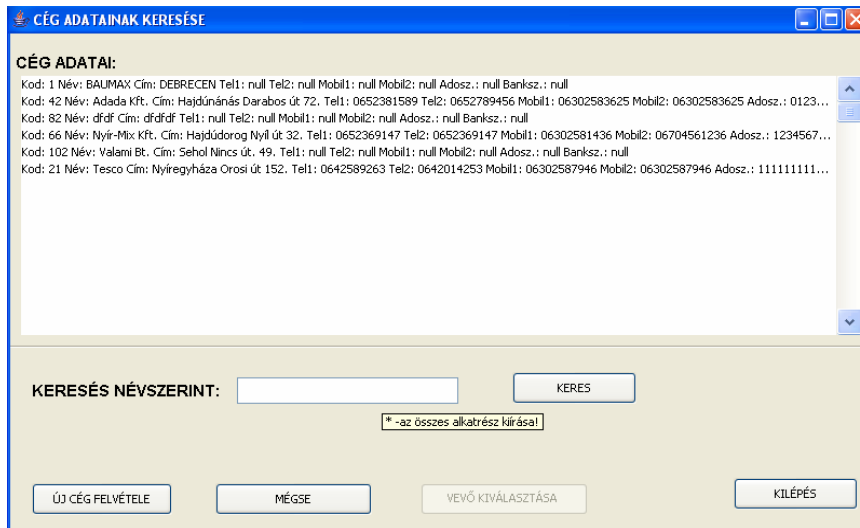
17. ábra: VTSZ szám keresése ablak, alkatrész típusaszerinti keresés után

Ezt az ablakot azért tartottam fontosnak a szoftverben létrehozni, mert a számlákon fel kell tüntetni a VTSZ számokat, és mivel ez a program nem rendelkezik számlázási funkcióval, ezért kézzel kell a számlát kitölteni, ahol nagyon fontosak a pontos adatok.

Az ablakból kilépni a KILÉPÉS gomb aktivizálásával lehet.

2.1.3.3. Cégek adatainak keresése

Az utolsó menüpont a keresések kategóriánál a cégek adatainak a keresése.



18. ábra: Cég adatainak keresése ablak kezdeti állapotban

Ez a következőképpen történik, a listából ki lehet választani a cég adatait. Hogy elősegítsük a kiválasztást, név szerint kereshetünk, a keresés úgy történik, mint az összes többi ablakban. Megadunk legalább egy karaktert keresési feltételnek, és a KERES gombra kattintunk, ekkor a lista leszűkül a feltételt kielégítő elemekre. Ha vissza akarjuk kapni a teljes listát, akkor itt is keresési feltételnek a * karaktert kell megadni.

Ha nem találjuk a megfelelő céget, akkor az ÚJ CÉG FELVÉTELE gomb aktivizálásával megnyílik az adatok felvételéhez segítséget nyújtó ablak. (A megnyíló ablak a cég adatainak felvétele címet viseli, amelyet az adat feltöltés kategória negyedik menüpontjából is el lehet érni közvetlenül. Ezért a használata a kategória megfelelő menüpontjának leírása alatt olvasható.)

Ha cég adatainak keresése ablakot az eladás ablakból hívtuk meg, és kiválasztjuk a keresett céget, akkor aktívvá válik a VEVŐ KIVÁLASZTÁSA gomb. Ha erre a gombra rákattintunk az egérrel, akkor bezáródik a keresés ablak és kiválasztott cég adatai betöltődnek az eladás ablak megfelelő mezőibe.

Végül, pedig ha a MÉGSE gombot választjuk, akkor a kezdeti állapotba állítjuk vissza az ablakot.

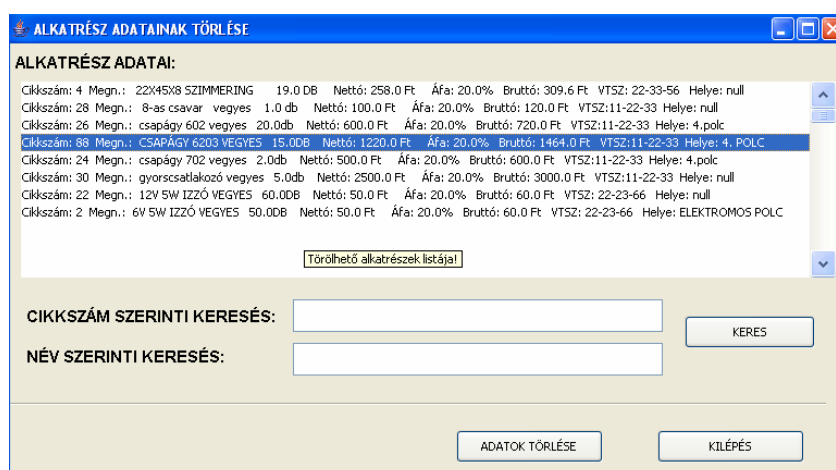
A szabályos ablak bezárást itt is a KILÉPÉS gomb valósítja meg.

2.1.4. Adat törlés

Adat törlés kategóriában is három menüpont van. Az első az alkatrészek adatainak törlése, a második a VTSZ adatainak a törlése, és az utolsó pedig a cégek adatainak a törlése.

2.1.4.1. Alkatrész adatainak törlése

Nézzük először az első menüpont által meghívható alkatrész adatainak törlése ablak működését.



19. ábra: Alkatrész adatainak törlése ablak kezdeti állapotban

Már megszokott módon az alkatrészek adatait ebben az esetben is egy lista tartalmazza, amelyből egérekattintással kell kiválasztani a törölni kívántat. Hogy elősegítsük a kiválasztást, cikkszám szerinti vagy név szerinti kereső feltételt lehet megadni, a kereső mezőkben. Egyszerre csak egyféleképpen lehet keresni, és nem szükséges a teljes cikkszámot vagy nevet beírni, de legalább egy karaktert meg kell adni. Miután megadtuk a feltételt, a KERES gombra kattintva a listából, eltűnnek azok az elemek amelyek, a keresést nem elégítik ki. A * keresési feltétel itt is visszatölti az összes elemet a listába.

Ha kiválasztottuk az elemet, amit törölni akarunk, akkor egy egérekattintással aktívvá kell tenni a listában (ez azt jelenti, hogy miután rákattintottunk az egérrel egy kék színű csíkban lesznek kiemelve az adatai, ahogy az ábrán is látszik), utána pedig az ADATOK TÖRLÉSE gombra kell kattintani. Ekkor a listából és az adatbázisból is végleg törlődnek a kiválasztott alkatrész adatai.

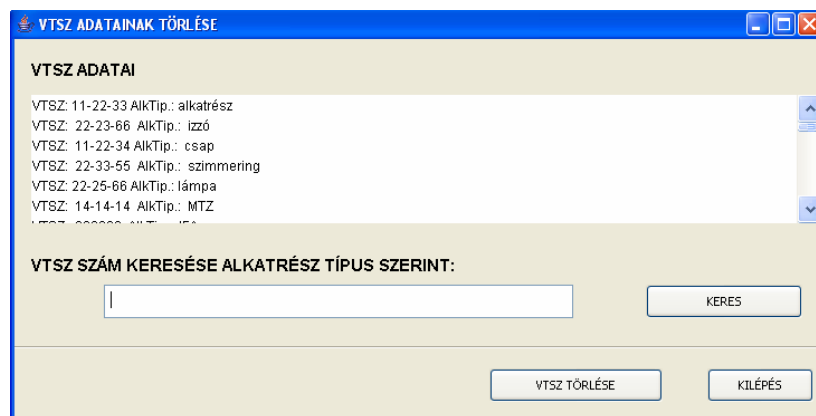
A törölt adatokat nem lehet visszaállítani, ezért törlés előtt kell végig gondolni a szükségességüket.

Ha mégsem szeretnénk adatot törölni, akkor a KILÉPÉS gombbal tudjuk bezárni ezt az ablakot.

2.1.4.2. VTSZ adatainak törlése

VTSZ adatainak törlésénél a kiválasztott vámtarifaszámot lehet véglegesen törölni, az adatbázisból.

Itt is listában vannak megadva a vámtarifaszám adatai, és innen kell kiválasztania a törlendő.



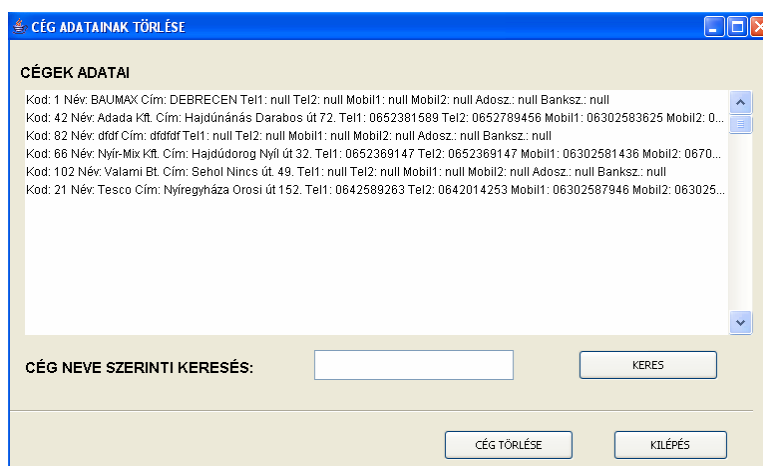
20. ábra: VTSZ adatainak törlése ablak kezdeti állapotban

A kiválasztást az alkatrész típusa szerinti keresés segíti, ahol a kereső mezőben kell megadni a keresési feltételt, amely legalább egy karakter kell, hogy legyen. Azután a KERES gomb aktivizálásával a lista csak a feltételt kielégítő adatokat fogja tartalmazni, így könnyebbé válik a kiválasztás. Ha kiválasztottuk a törlendő elemet (rákattintottunk egérrel és aktív lett), akkor a VTSZ TÖRLÉSE gombra kattintva végleg el tudjuk távolítani az adatbázisból. A törlés után a törölt elem eltűnik a listából.

Ha minden elemet töröltünk, amit akartunk, akkor a KILÉPÉS gombbal tudjuk bezárni az ablakot.

2.1.4.3. Cég adatainak törlése

Adat törlés menürészben az utolsó menüpont a cégek adatainak törlése, ahol az adatbázisból végleg törölni kívánt cégek adatait lehet kiválasztani és a tevékenységet végrehajtani.



21. ábra: Cég adatainak törlése ablak kezdeti állapotban

Ebben az ablakban is az adatokat tartalmazó listából kell kiválasztani a törölni kívánt céget. Cég neve szerinti keresés megoldott, mégpedig úgy, hogy a keresési mezőben megadunk legalább egy karaktert. Ezután a KERES gombra kattintva, a mezőben megadott feltételnek megfelelő cégek adatait fogja csak a listában elhelyezni. Így a kevesebb elemből könnyebb lesz a kiválasztás.

Ha kiválasztottuk a megfelelő céget, és biztosak vagyunk a törlésben, akkor a CÉG TÖRLÉSE gombra kattintva a cég törlődni fog az adatbázisból és a listából.

Ha minden művelet végrehajtottunk, amit itt szerettünk volna, akkor a KILÉPÉS gomb segítségével zárhatjuk be az ablakot.

3. A szoftver továbbfejlesztési lehetőségei

A program csak az alkatrészek a cégek, és a hozzájuk kapcsolódó adatok nyilvántartását valósítja meg és eltárolja az eladott alkatrészekhez fűződő információkat.

Hasznos lenne, ha bizonyos adatokról lehetne listát készíteni és kinyomtatni. Például, ha az áruk raktárban lévő mennyisége, egy megadott érték alá csökken (esetleg nullára), akkor felkerülnének egy listára, aminek lehetne a neve hiány lista. Vagy ha egy vevő szeretné tudni, hogy a bolt milyen áron forgalmazza a termékeket, akkor ki lehetne nyomtatni egy árlistát. Esetleg jó lenne, ha a boltnak tartozó ügyfelekről is lehetne egy kimutatást készíteni, amely tartalmazná a vásárló nevét, azt hogy mely termékeket nem fizetett ki és azt, hogy mikor vásárolta azokat.

De szükséges volna az esetleges félreértések elkerülésére, a szállító kinyomtatására két példányban. Az egyik példány a bolt tulajdonában maradna a másikat pedig megkapná a vásárló, így mindenki látná a vásárlási paramétereiket, és az adatok papíron mindig meglennének, az adatbázis megsérülése esetén is.

Lehetne még az alkatrészekről tárolt adatokat kiegészíteni képekkel, amelyek a termékekről készülnének. Így ha egy vevő nem tudja pontosan az áru nevét, akkor a kép alapján mégis ki lehetne keresni az adatbázisból.

Sokszor láttam, hogy milyen gondot okoz egy bolt esetében a kifogyott áru újabb beszerzése. Egy kezdőnek nehéz észben tartani, hogy melyik árut, melyik vállalattól szerezte be. Ezért az is hasznos lenne, ha tárolni lehetne, hogy a termékeket, melyik cégtől vásároltuk meg.

4. Összegzés

Amikor tanulmányaim során elkezdtem programozni, és már néhány nagyobb programot elkészítettem, akkor rájöttem, hogy teljesen kész program nincs. Ezt azért merem kijelenteni, mert egy szoftvert mindig lehet bővíteni újabb funkciókkal, és esetleg a már meglévőket is lehet finomítani. Ezzel az alkatrész nyilvántartó programmal is ugyan ez a helyzet.

Mikor elgondoltam, hogy milyen szakdolgozati témát szeretnék választani, akkor sok terv bolyongott a fejemben és nehéz volt döntenem. A választásnál az a cél vezérelt, hogy minél nagyobb gyakorlatot szerezzek, és sok tapasztalatot gyűjtsek. De a legfontosabb az volt számomra, hogy a sok munkának legyen valami haszna is, és mivel édesapámnak úgymint szüksége volt egy programra, ez eldöntötte a kérdést.

Kezdetben több funkciót szerettem volna megvalósítani a szoftveremben, de az időhiány és technikai problémák miatt, szűkíteni kellett a megoldandó feladatokat. Az adatbázis megtervezése és létrehozása elég könnyen ment. Még a felület megtervezésén is hamar túl jutottam, a legnagyobb gondot a felületi objektumok összehangolt munkájának létrehozása jelentette.

Kezdetben egy kicsit nehezen tudtam megszokni az Oracle és a JDeveloper használatát, habár már tanulmányaim során külön-külön mind a kettővel kapcsolatba kerültem. Először létrehoztam az adatbázis-kezelőben a táblákat, azután pedig a felületet készítettem el. A felület létrehozásánál fő szempont volt az adatok gyors elérése és a logikusság. Próbáltam arra törekedni, hogy minél kevesebb adatot kelljen kézzel begépelni, ezért használtam a legtöbb helyen listákat, és adtam lehetőséget keresésre.

Az adatbázis és a felület között, akkor hoztam létre a kapcsolatot, amikor már a felületen lévő objektumok majdnem tökéletesen tudtak együtt működni, ezt a konzulens tanárom javasolta nekem.

Nehezen tudtam a szoftverem határait meghúzni, mindig eszembe jutottak újabbnál újabb tevékenységek, amiket szerettem volna még megvalósítani benne. A terveim között szerepelt, hogy a számlázást is megvalósítom, amely az elkészült szállítók alapján jött volna létre azokból a tételekből, amelyeket az ügyfél már kifizetett. Jó lett volna még, ha a szoftverrel listákat is lehetett volna készíteni. Ezt azért szerettem volna megvalósítani, mert a felhasználónak ennyivel is kevesebb papírmunkát kellett volna végeznie. (Ezeket, a terveimet bővebben kifejtettem a szoftver tovább fejlesztését megcélzó fejezetben.) Az időhiány, ami a legjobban meggátolt az előbbieken leírt program részek létrehozásában. De azért törekedtem

arra, hogy az elkészült részek is egy egészt alkossanak, és ne legyen hiányérzete a szoftvert használó embernek.

Azért a későbbiekben, ha lesz rá időm és alkalmam, akkor az államvizsga után szeretném a továbbfejlesztési lehetőségeknél leírtakat a gyakorlatban is megvalósítani.

A szoftver elkészítésének fázisai közben rengeteget tanultam, és a már megszerzett tudásomat is több helyen alkalmaztam. Sokat segített a témavezetőm is, ha megakadtam valamelyik fázisban, és rengeteg apróságra felhívta a figyelmemet. A csoporttársaim is adtak egy-két ötletet, ami elősegítette a munkámat.

Remélem, hogy a program továbbfejlesztésével kapcsolatban kitűzött céljaimat lesz lehetőségem megvalósítani, és a szoftver használatba helyezése után sok munka alól fogja mentesíteni a felhasználót.

5. Irodalomjegyzék

Könyvek

- [1] Nyékyné Gaizler Judit (szerk.): J2EE útikalauz programozóknak, ELTE TTK Hallgatói Alapítvány, 2002
- [2] Nyékyné Gaizler Judit (szerk.): Java 2 Útikalauz programozóknak I. : 1.3 A kávé, ELTE TTK Hallgatói Alapítvány, 2001
- [3] Nyékyné Gaizler Judit (szerk.): Java 2 Útikalauz programozóknak II. : 1.3 A hab, ELTE TTK Hallgatói Alapítvány, 2001
- [4] Nyékyné Gaizler Judit (szerk.): Java 2 referencia programozóknak: 1.3 A csésze, ELTE TTK Hallgatói Alapítvány 2001
- [5] Gábor András – Juhász István: PL/SQL programozás, Panem, 2007
- [6] Kevin Loney: Oracle Database 10g Teljes referencia, Panem, 2006
- [7] Robert A. Maksimchuk, Eric J. Naiburg: UML földi halandóknak, Kiskapu 2006
- [8] Gábor András, Gunda Lénárd, Juhász István, Kollár Lajos, Mohai Gábor, Vágner Anikó: Az Oracle és a web, Panem, 2003

Jegyzetek

- [9] L. Nagy Éva: Haladó Oracle ismeretek 1. órai jegyzet,
http://delfin.unideb.hu/~lnagyeva/PLSQL/PLSQL_roviden.doc

Web címek

- [10] <http://otn.oracle.com> - Oracle Technology Network webhelye
- [11] <http://www.prog.hu/>

6. Függelék

6.1. Ábrajegyzék

1. ábra: Tárolt adatok leírásának táblázata	8
2. ábra: Táblák közötti kapcsolatok	11
3. ábra: Frame-ek közötti kapcsolatok	14
4. ábra: Java osztálydiagram	16
5. ábra: Menü ablak kezdeti állapotban	18
6. ábra: Megerősítést váró ablak	19
7. ábra: Alkatrész felvitele ablak kezdeti állapotban	20
8. ábra: Új alkatrész felvitele ablak, ahol nem választottunk ki VTSZ számot	21
9. ábra: Új VTSZ felvétele ablak, amikor létrehozunk egy új elemet	22
10. ábra: Új cég felvétele ablak kezdeti állapotban	23
11. ábra: Eladás ablak kezdeti állapotban	24
12. ábra: Tartozás rendezése ablak, a rendezni kívánt tartozások kiválasztása után	25
13. ábra: Alkatrész adatainak módosítása ablak kezdeti állapotban	26
14. ábra: VTSZ adatainak módosítása ablak kezdeti állapotban	27
15. ábra: Cégek adatainak módosítása ablak kezdeti állapotban	28
16. ábra: Alkatrész keresése ablak, egy alkatrész kiválasztása után	29
17. ábra: VTSZ szám keresése ablak, alkatrész típusaszerinti keresés után	31
18. ábra: Cég adatainak keresése ablak kezdeti állapotban	31
19. ábra: Alkatrész adatainak törlése ablak kezdeti állapotban	32
20. ábra: VTSZ adatainak törlése ablak kezdeti állapotban	33
21. ábra: Cég adatainak törlése ablak kezdeti állapotban	34

7. Köszönetnyilvánítás

Ebben a fejezetben szeretném megköszönni témavezetőmnek Kollár Lajosnak azt a sok segítséget és tanácsot, amellyel elősegítette ennek a szakdolgozatnak a megírását. Megmutatta a fejlesztésben alkalmazott szoftverek hatékony kezelését, programozás technikai tanácsokat adott a problémás részeknél. De nem utolsó sorban szeretném még azt a sok biztatást is megköszönni, amellyel kitartásra ösztönzött a nehézségek áthidalásánál.

Köszönet illeti még meg a szüleimet és a testvéreimet is, akik érdeklődésükkel, okos tanácsaikkal, anyagi támogatásukkal és rengeteg türelmükkel ösztönöztek és segítettek a nehézségek során.

Hálás vagyok még nagymamámnak, aki minden vizsgánál és nehézségnél imádkozik értem és biztat engem.

Köszönöm még a barátomnak és a legjobb barátnőmnek, hogy ha kellett meghallgattak, a nehézségeknél tanácsokkal láttak el, és izgultak értem a vizsgáimnál.

Az egyetemi évek alatt sok csoporttársammal ismerkedtem meg, akik a nehéz percekben is mellettem álltak, és segítettek engem, ha pedig kellett tanácsot adtak, ezért nekik is köszönetet mondok.

És végül, de nem utolsó sorban köszönöm az oktatóimnak, hogy az évek alatt próbáltak minél több tudást és információt átadni nekem, hogy ezzel is segítsék az életben való boldogulásomat.