1.Bevezetés

A szakdolgozat témáját az informatika tanszéken meghirdetett témák közül elsősorban a JAVA nyelv megismerése és gyakorlati alkalmazása miatt választottam. A JAVA nyelv beépített csomagjaival alkalmas (GUI) grafikus felületű programok készítésére, appletek készítésére, amelyek HTML környezetben, böngészőn keresztül saját gépen futó programok, valamint távoli MySQL adatbázisokkal való kommunikációra is. A gyakorlatban is alkalmazható raktárkészlet nyilvántartó alkalmazás készítése pedig a JAVA és MySQL ismeretek felhasználására egy jó lehetőség. Egyik rokonom építőipari cégnél raktárosként dolgozott, így valóságos raktárkezelési problémakört vettem alapul.

A cég raktárkészlet nyilvántartását kartonok alapján végzik, melyek tartalmazzák cikk kiadása esetén a cikk megnevezését, a cikkszámot, mennyi lett belőle kiadva (mennyiség és egység), és kinek. Tartalmazza a felhasználás helyét, hogy mely gépen, műhelyben vagy telephelyen dolgozik az alkalmazott, ahol a cikk felhasználásra kerül, valamint rajta van a dátum, a karton sorszáma, és az átvevő aláírása. Cikk átvétele esetén szintén tartalmazza a karton sorszámát, a dátumot, a cikk megnevezését, a cikkszámot, tartalmazza a vett mennyiséget, és egységet valamint az egységárat. Raktározási kartonok is készülnek, amelyek tartalmazzák a cikkszámot, a cikk megnevezését, a cikkszámot, is készülnek, amelyek tartalmazzák a cikkszámot, a cikk megnevezését, a ktuális mennyiségét és egységét, valamint a raktározás dátumát.

A raktárkészlet olyan eszközökből és alkatrészekből áll, amelyek a cég termelése során meghibásodás, illetve elhasználódás miatt pótlásra szorulnak (például: munkaruha, csavar, légszűrő, golyóscsapágy, stb...). Itt nem kereskedelmi cég raktárkezeléséről van szó, hanem egy termelő építőipari cég raktárkezeléséről. A raktárkészlet beszállítóktól kerül megvásárlásra, majd kisebb tételekben a dolgozókhoz kerül kiadásra.

A feladat olyan nyilvántartó alkalmazás készítése, amely a raktárkezelési eseményeket rögzíti. A nyilvántartás során szükséges tevékenységeket: adatbevitel, adat törése, javítása lehetővé teszi, megjeleníthető az aktuális raktárkészletet, és a nyilvántartás több helyről, többféle jogosultsággal elérhető. Az adminisztráció számára fontos lehet gazdasági kimutatásokhoz, (költségek, készletérték) és a készlet megrendelések is indíthatók alacsony készlet esetén.

2. Az alkalmazás megvalósítása

Az első lépés a feladat elemzése, a megvalósítás részlépésekre bontása (Analízis).

1. Igényfelmérés Információgyűjtés, annak meghatározza, hogy mely felhasználók, milyen feladatokat akarnak végezni.

2. Felhasználói felület tervezése. A feladat szempontjából célszerű, a felhasználói igényeknek megfelelő, felhasználói felület kialakítása.

3. Az adatbázis tervezése, elkészítése.

4. Futási feladat kialakítása. Az alkalmazásnak a JAVA és a MySQL eszközeinek, lehetőségeinek figyelembe vételével való megtervezése.

5. Megvalósítás JAVA nyelven.

6. Üzembe helyezés, tesztelés.

2.1. Feladatspecifikáció

Az alkalmazással szembeni elvárások: A feladat olyan alkalmazás készítése, amely a raktárkezelési események nyilvántartására alkalmas. A nyilvántartás MySQL adatbázisában tárolni kell a kartonozás során is hasznát adatokat:

Cikk átvétele, vásárlása során a cikkszámot, a cikk megnevezését, a vásárolt mennyiséget, és mennyiségi egységet, az egységárat, a vásárlás dátumát.

Cikk kiadása során a cikkszámot, a cikk megnevezését, a kiadott mennyiséget, és mennyiségi egységet, a kiadás dátumát, hogy ki vette át és az átvevő mely gépen, telephelyen dolgozik. A 2. ábra (az elkészült adatbázis) tartalmazza az adatok típusát és hosszát.

A nyilvántartásban a cikkek aktuális raktármennyiségének naprakészen elérhetőnek kell lennie. Az alkalmazásnak kezelnie kell a raktározási eseményeket. A kiadási, vásárlási eseményeket nyilvántartásba kell vennie. Lehetővé kell tennie az adatok módosítását, törlését, új adatok felvételét. Az adatokat táblázatos formában meg kell jelenítenie, lehetővé kell tennie az adatbázisban való keresést, (valamely szempont szerinti lekérdezést).

A biztonságos adatkezelés érdekében jelszavas beléptetés szükséges, valamint külön jogosultság, amellyel adminisztratív célból az adatok megtekinthetők, de nem változtathatók. Az alkalmazásnak így több gépen futtatva is biztonságosan kezelnie kell ugyanazt az adatbázist. Az alkalmazásnak könnyen kezelhetőnek, stabilnak, kis hardverigényűnek kell lennie.

Futási környezet: Windows 98.

Az 1. Ábra annak szemléltetése, hogy mely felhasználók milyen feladatokat akarnak végezni.



1. ábra aktorok és Use-Casek kapcsolata

2.2. Felhasználói felület tervezése

A program indításakor megjelenő ablak kérje a felhasználó nevét és jelszavát, megakadályozva ezzel az illetéktelen adathozzáférést, és jogosulatlan adatmódosítást. Az itt kért név és jelszó nem a MySQL adatbázis jelszava. A MySQL adatbázis felhasználóneve és jelszava fenti példában "" azaz nincs, mert ez már helyi beállítástól függ. A jogosultság megállapítása után a felhasználó részére meg kell jelenjen egyfajta menü a rendelkezésre álló lehetőségekről. Ugyanezen belépési ablak része további hét nyomógomb, amelyek elérhetősége a jogosultságtól függ. Kezdetben és jogosulatlanság esetén a gombok disable módban Windows módon jelzik, hogy jelen vannak, de jelenleg nem funkcionálnak. Jogosultság esetén viszont az adott gombon, feketén megjelenik a funkciónak megfelelő felirat, a gomb pedig használható. A gombra kattintva újabb ablak kell, hogy megjelenjen, az előző ablaknak pedig záródnia kell vagy a háttérben maradnia. A későbbiekben viszont szükség lehet rá, ha más műveletet kívánunk végezni, így jobb, ha a háttérben marad. A raktárosnak szüksége lehet adatlap megjelenésű ablakra, amely által bevihetjük a cikk kiadását vagy az átvételét.

Kellenek továbbá olyan ablakok, amelyeken az adattáblák adatainak törlése, módosítása elvégezhető. Itt fontos sok adat egyszerre áttekinthetősége, ezért célszerű egyfajta táblázatos forma. Lekérdezések esetén először egy adatlapszerű ablak kell, ahol megadjuk a lekérdezés feltételeit, majd gombnyomásra hívnia kell egy táblázatszerű ablakot, ahol a feltételnek megfelelő adatok megjelennek. Az adminisztráció számára az adattáblák lekérdezésére szintén szükség van.

2.3. Az adatbázis tervezése

A feladat szempontjából külön nyilván kell tartani a készletvásárlás, és a cikk kiadás (kartonait) tábláit, hiszen ezek fontosak a feladat szempontjából, a cikk kiadás, átvétel ezek által történik. Viszont sok közöttük az átfedés, a cikkszámhoz egyértelműen hozzárendelhető a megnevezés, és a mennyiségi egység. Az így létrejövő tábla (aru) indexével, a cikkszámmal kapcsolódhat a másik két táblához, az áru megnevezése, és mennyiségi egysége pedig kihagyandó ezekből. A leltár viszont egyfajta aktuális készletmennyiség lekérdezéseként értelmezhető, és nem kell külön tábla neki, csak az aktuális mennyiséget kell az aru táblában tárolni. Az aktuális mennyiség frissítéséről a programnak kell gondoskodnia, a másik két tábla adatai szerint. Ha áru érkezik, hozzá kell adni a készlethez, ha kivétel, csökkenteni az aktuális mennyiséget.



2. ábra a raktárkezelés adatbázisának felépítése

2.4 A MySQL és a JDBC

Az adatbázis gyakorlati megvalósításához szükséges a MySQL valamely verziója. A MySQL több ingyenesen letölthető verziója, több operációsrendszerre rendelkezésre áll a mysql.com [1] honlapján. A kereskedelmi üzleti célú felhasználása viszont nem ingyenes. Nem feltétlen a legújabb a legjobb az fontosabb, hogy tesztelt, stabil verzió legyen. A kicsomagolt program Docs alkönyvtárban található manual.html és manual_toc.html hasznos információkat tartalmaz angolul, amely segíthet a MySQL alkalmazása során. A bin alkönyvtárban található mysql.exe parancssoros adatbevitelt tesz lehetővé.

A WinMySQLAdmin.exe ablakos formában jeleníti meg a MySQL beállításait. A data alkönyvtárban található egy teszt adatbázis, és ide kerülnek az általunk létrehozott adatbázisok is. A MySQL-ban egy adatbázis egy alkönyvtárnak felel meg, a táblázatokat pedig három fájl alkotja:

tbl_név.frm Tábla leíró fájl tbl_név.MYD Adat fájl és

tbl_név.MYI Index fájl

A mysql.exe elindításkor parancssoros adatbeviteli módon működő program indul el, amely a MYSQL> prompt után várja az SQL parancsokat. Az SQL Structured Query Language (struktúrált lekérdező nyelv) rövidítése. Ez egy IBM által relációs adatbázisokhoz kifejlesztett kommunikációs nyelv. Az SQL egyszerű adatbázis parancsok gyűjteménye, amelyekkel az adatbázisok adatai lekérdezhetők, módosíthatók, de új adatbázisok is létrehozhatóak.

[1]. MySQL Downloads http://dev.mysql.com/get/Downloads/MySQL-3.23/mysql-3.23.58-win-src.zip/from/pick Néhány az SQL parancsok közül:

A rakt adatbázis (könyvtár) létrehozása:

CREATE DATABASE rakt;

Belépés a könyvtárba (adatbázisba):

USE DATABASE rakt;

A vasarolt tábla létrehozása a tábla oszlopainak megadásával:

CREATE TABLE vasarolt(sor int, csz char(12), mennyi int, ar int);

Tábla módosítása táblázathoz hozzáadása:

ALTER TABLE vasarolt ADD (datum date);

Sor bevitele a táblába:

INSERT INTO vasarolt VALUES(5,'kk1',7,6,20020506);

Adatok megjelenítése a vásárolt táblában:

SELECT * FROM vasarolt;

Cikkszám szerint rendezett megjelenítés:

SELECT * FROM vasarolt ORDER BY csz;

Adat felülírás:

UPDATE vasarolt SET csz='kk1' WHERE sor=27;

Sortörlés:

DELETE * FROM vasarolt WHERE sor=28;

Itt csak az adatbázist és a három táblát kell létrehozni néhány adattal. A műveletek többsége a JAVA alkalmazáson keresztül történik.

Az SQL nyelv kvázi szabvánnyá vált, a Java integrált adatbázis támogatása (JDBC) is az SQL-t használja adatbázisokkal történő kommunikációra. A JDBC (Java DataBase Connectivity) osztályok és interfészek gyűjteménye, melyek segítségével kapcsolat építhető fel a Java-program és egy adatbázis között. Az adatbázis-kapcsolat létrehozásához nem elegendő az egyik oldalon a JDBC a másik oldalon pedig az adatbázis. A kapcsolathoz szükség van még egy meghajtóra is, mely a JDBC-parancsokat megérti, lefordítja és továbbítja az adatbázishoz. Az Access adatbázishoz Access JDBC, Sybase adatbázishoz Sybase JDBC –meghajtó kell. JDBC-ODBC híd segítségével a Microsoft által kifejlesztett számos ODBC-meghajtó használható.

A JDBC-ODBC kapcsolat azonban nem biztonságos az ODBC-kód nincs korlátozva, például a helyi merevlemezek hozzáférésétől. Célszerű inkább a MySQL adatbázisokhoz a MySQL JDBC –konnektor használata, amely a MySQL.com címen szintén megtalálható. [2]

A fontosabb JDE	BC osztályok:
Driver:	Az adatbázis megvalósítása és regisztrálása.
DriverManager:	Kapcsolat létrehozása a program és az adatbázis között.
	(Előtte megfelelő meghajtót kell regisztrálni a DriverManagernél.)
Connection:	Az adatbázis-kapcsolatot képviseli.
Statement:	Az SQL-parancsok összeírása és adatbázishoz elküldése.
ResultSet:	Az adatbázisból visszakapott adatokat képviseli.

Példa MySQL kapcsolatra Java-ban MySQL JDBC- konnektorral:

```
import java.sql.*; //SQL kezeléshez importálni kell a java.sql
csomagot.
```

```
public void mennyiseg_korrekcio (String csza, int
uj_raktmennyiseg){
String URL = "jdbc:mysql:///rakt";
String felhasznev="";
String jelszo="";
```

/* A meghajtó betöltése itt történik a try blokkban. A try catch blokkpár alkalmazása ilyen esetben kötelező, mert ezen utasítás hibajelenséget válthat ki, (Exception ex) és ezen esetben a program futása a catch blokkban folytatódik, mint f lese esetén. A catch blokk üresen hagyható, de célszerű itt hibaüzenetet megjeleníttetni. */

```
try {
    Class.forName("com.mysql.jdbc.Driver").newInstance();
// meghajtó betöltése
    }
    catch (Exception ex) {
        megjegyzes.setText("SQLException 1: " + ex.getMessage());
// hibaüzenet
    }
    Statement parancs = null;
    Connection kapcsolat = null;
/* A MySQL kapcsolat felépítése itt történik az URL cím,
felhasználónév és jelszó megadása kell a kapcsolat
létrehozásához.*/
```

```
try { // kapcsolat létrehozása
    kapcsolat = DriverManager.getConnection
(URL,felhasznev,jelszo);
    parancs = kapcsolat.createStatement();
    }
    catch (SQLException ex) {
        megjegyzes.setText("SQLException 2: " + ex.getMessage() +
        SQLState: " +
        ex.getSQLState() + " VendorErr: " + ex.getErrorCode());
    }
```

/* Itt történik a végrehajtandó utasítás kiadása siker esetén a visszatérési érték igaz. A parancs a macskakörmök közötti uj_raktmennyiseg változó értékére változtatja a csza cikkszámú cikk aktuális raktármennyiségét az aru táblában. Majd a kapcsolat zárása következik. */

```
boolean adatk;
 try{
       adtak
                       parancs.execute("UPDATE
                                                          SET
                 =
                                                  aru
aktme="+uj_raktmennyiseg+" WHERE
       csz='"+csza+"';"); // SQL parancs
       kapcsolat.close(); // kapcsolat zárása
  }
  catch (Exception e) {
e.printStackTrace(); //A szabvány kimenetre kiírja a program
hibajelzését.
       megjegyzes.setText("Hiba:
                                     UPDATE
                                                  aru
                                                           SET
aktme="+uj_raktmennyiseg+"
       WHERE csz='"+csza+"';");
  }
}
```

2.5. Az alkalmazás általános felépítése

A JAVA nyelv objektum-orientált. Az objektumok tartalmazzák a függvényeket és a hozzájuk tartozó adatokat. A program működése során több ablakot, kezel ezért célszerű volt, hogy minden ablak egy-egy külön osztályt alkotva külön fájlban szerepeljen.

₿R	aktárke	zelés				_	□×
	Név:	Sanyi	Jelszó:	****	Belépés:	Ok	
		Raktárból áru kiadása :		Kiadás tábla szerkesztése		Lekérdezések: kiadás :	
		Készlet vásárlása :		Vásárolt tábla szerkesztése		Lekérdezések: vásárlás (
				Áru tábla szerkesztése			

3. ábra. A program első ablaka raktárosi jogosultság esetén, mely ablakot a Password. java hozza létre

Raktárosi jogosultság esetén, (3. ábra) választható opciókhoz, gombokhoz kapcsolódó fájlok:

Raktárból áru kiadása:	Adatbevitel_kiadas.java
Készlet vásárlása:	Adatbevitel_vasarolt.java
Kiadás tábla szerkesztése:	Kiadas_tabl_szerkesztese.java (írásjog=true)
Vásárolt tábla szerkesztése:	Vasarolt_tabl_szerkesztese.java (írásjog=true)
Áru tábla szerkesztése:	Aru_tabl szerkesztése.java (írásjog=true)
Lekérdezések kiadás:	Lekerdezes_kiadas.java majd
	Kiadas_tabl_szerkesztese.java
	(A lekérdezés eredménye írásjog=false)
Lekérdezések vásárlás:	Lekerdezes_vasarolt.java majd
	Vasarolt_tabl_szerkesztese.java
	(A lekérdezés eredménye írásjog=false)

ktárke	zelés				
Név:	Gabi	Jelszó:	*****	Belépés:	Ok
	Raktárból áru kiadása 💠		Kiadás tábla megjelenítése		Lekérdezések: kiadás :
	Készlet vásárlása :		Vásárolt tábla megjelenítése		Lekérdezések: vásárlás 👘 :
			Áru tábla megjelenítése		Belépés rendben



Adminisztrátori jogosultság esetén, (4. ábra) választható opciókhoz, gombokhoz kapcsolódó fájlok:

Kiadás tábla megjelenítése:	Kiadas_tabl_szerkesztese.java (írásjog=false)
Vásárolt tábla megjelenítése:	Vasarolt_tabl_szerkesztese.java (írásjog=false)
Áru tábla megjelenítése:	Aru_tabl szerkesztése.java (írásjog=false)
Lekérdezések kiadás:	Lekerdezes_kiadas.java majd Kiadas_tabl_szerkesztese
	.java (A lekérdezés eredménye írásjog=false)
Lekérdezések vásárlás:	Lekerdezes_vasarolt.java majd Vasarolt_tabl_szerkesztese
	.java (A lekérdezés eredménye írásjog=false)

A választható opcióknak gomboknak megfelelő ablakok a melléklet ábráin láthatóak.

A Hiba.java mindkét esetben bármelyik függvényből meghívódhat, ha SQL exception lép fel, vagyis hiba történik az SQL parancs végrehajtása során.

2.6 Az alkalmazás fájljainak elemzése

A program kilenc forrás fájlból áll, melyek mindegyike egy-egy ablakot hoz létre. Ezek között van három, mely egy táblázatot, hoz létre a hozzá tartozó gombokkal (pld.: Kiadas_tabl_szerkesztese.java), van négy, amely adatlapszerű ablakot jelenít meg (pld.: Adatbevitel_kiadas.java) adatbeviteli mezőkkel, és gombokkal. Egy ablak menüszerűen működik (Password.java), és gombokat tartalmaz, amelyekkel más ablakok hívása történik, egy pedig a legegyszerűbb, csak egy ablakot tartalmaz, amely hiba esetén jelenik meg, hibaüzenetet jelenít meg (Hiba.java). A kilenc fájlt tehát négy típusba sorolhatjuk a továbbiakban ezek egy-egy példányát elemezzük.

2.6.1. Password.java

A Password.java láthatóan központi helyet foglal el a programban. Ez a fájl tartalmazza a main függvényt, vagyis a program innen indul. Az ablakban megjelenik két adatbeviteli mező, ahová a felhasználónevet és a jelszót kell írni. A ezeket a program megvizsgálja, majd annak függvényében, hogy az adatbázisban megtalálta ezen adatokat vagy sem, a középen illetve alul levő gombok elérhetővé válhatnak. Ha nem találja az adatpárt, hibaüzenet jelenik meg. A program nem engedi belépni a felhasználót, de új adat (felhasználónév, jelszó) írható be. Ha megtalálta, akkor ellenőrzi, hogy a jogosultság `'r' (csak olvasás), vagy 'w' (írás-olvasásmódosítás törlés). Előbbi esetben csak az adatok megjelenítése és a lekérdezések elérhetőek, míg második esetben az összes funkció választható.

Osztály: Password extends Frame A Password a Frame osztály leszármazottja, így örökli a java.awt.Frame minden adatelemét és metódusát (osztályfüggvényét). Példányváltozók: A ptex1 és ptex2 TextFieldek a helye az adatok név és jelszó bevitelének. Button pasb a felhasználónév és a jelszó vizsgálata ezen gomb benyomása után kezdődik. Label labmegj üzeneteket közvetít a felhasználó felé. Button butt[0]-butt[6] tömb elemei olyan gombok, amelyek megjelenítik a választási lehetőségeket, és a kívánt ablak hívása megtörténik. **Osztály:** WindowFigyelom extends WindowAdapter. A WindowFigyelom a WindowAdapter osztály leszármazottja. Ezen osztály segítségével lehet beállítani, mi történjen az ablak kilépés(x) az átméretezés(\Box) vagy az elrejtés(_) gombra kattintás hatására. Az osztály windowClosing (WindowEvent e) osztályfüggvényét felülírva, System.exit(0) hatására a program összes ablakát bezárja, és a lefoglalt erőforrásokat felszabadítja. A program befejezi a működését, a programból való kilépés itt történik.

Osztályfüggvény: jogosultsag(String felh, String jels) visszatérési típus: String. Ez a függvény ellenőrzi, milyen jogosultság tartozik a felhasználónév, jelszó adatpárhoz. Ha az adatbázis nem elérhető (SQL exception lép fel), a belépés sikertelen, ha az adatbázisban nem található a megadott adatpár, szintén. A helyesen megadott jelszó és felhasználónév esetén a belepo táblából kiolvasásra kerül egy csak olvasási hozzáférést ("r") vagy teljes hozzáférést írás olvasás törlés ("w") jogosultságot jelző String, ami továbbítódik a hívó függvény felé. A függvény megnyitja a belepo táblát, majd egyesével kiolvassa az adatsorokat, és megvizsgálja azonos-e a bemenetben megadott adatpár a táblából kiolvasottal. Ha igen, a **vanusert** igazra állítja jogosultságot a **jogosult** változóba teszi, és visszatérési értékként használja.

Osztály: AkcioFigyelom implements ActionListener. Ez az osztály tartalmazza az actionPerformed(ActionEvent e) osztályfüggvényt, amely kezeli az adatbeviteli mezők és a gombnyomások eseményeit. Ha az Ok gombot nyomtuk meg (pasb) akkor a program meghívja a jogosultsag függvényt. Majd a visszatérési érték alapján megállapítja, milyen menüpontok legyenek elérhetőek. A writejog is ekkor kap true, vagy false értéket. Ez azért kell, mert van három ablak a Kiadas az Aru és a Vasarolt_tabla_szerkesztese, amely mindkét jogosultság esetén meghívódik, de más néven és egy true, vagy false paraméterrel. Ez utóbbi csak megtekintést semmilyen változtatást nem enged a táblákban, szemben a true paraméterrel.

2.6.2. Adatbevitel_kiadas.java

Adatbevitel_kiadas.java egy ablakot jelenít meg, amely segítségével új adatsort vihetünk be a kiadas táblába. Használata akkor célszerű, ha a raktárból cikket visznek ki. Adatbevitel történhet a kiadás tábla szerkesztése menüben is, ez csak egy újabb adatbeviteli lehetőséget ad. Az ez a fájl használja az aru táblát cikkszám ellenőrzéshez és az aktuális raktármennyiség csökkentéséhez, valamint a kiadas táblát új sor beírásához.

Osztály: Adatbevitel_kiadas extends Frame. Az adatbevitel_kiadas a Frame osztály leszármazottja, így örökli a java.awt.Frame minden adatelemét és metódusát.

Osztályfüggvény(metódus): cikkszam_ellenorzes() visszatérés: boolean. Igaz, ha a cikkszám, amely az adatbeviteli mezőben meg van adva, szerepel az adatbázisban, és hamis, ha nem. Ez a függvény tehát a Textfieldben megadott adatot keresi a MySQL adatbázisban. Hogyha megtalálja, az ablak Label típusú objektumain keresztül kijelzi az áru nevét raktári mennyiségét, és mennyiségi egységét.

Meghívó függvény: actionPerformed(ActionEvent e)

Osztályfüggvény: void mennyiseg_korrekcio(String csza, int uj_rakt mennyiseg) visszatérési érték nincs. Ez a függvény a csza Stringben megadott változót keresi a MySQL adatbázisban, és a hozzátartozó raktármennyiség értékét csökkenti az uj_raktmennyiseg értékével. Meghívó függvény: uj_adat_bevitel()

Osztályfüggvény: void uj_adat_bevitel()

Ez a függvény ellenőrzi, hogy szerepel-e a megadott cikkszám az aru táblájában.

Ha szerepel (a cikk létezik, vásárolható, kiadható), megkeresi a legnagyobb sorszámot a kiadas táblában, majd a következő sorszámmal beviszi az ablak TextFieldjeiban szereplő adatokat a kiadas táblába (így utolsó helyre, sorszámra kerül, a kiadási tétel.). Ha nem nem létezik a cikkszám, ezen utóbbi lépések kimaradnak.

Meghívó függvény: actionPerformed(ActionEvent e)

Osztály: WindowFigyelom extends WindowAdapter. Ezen osztály segítségével az ablak viselkedését lehet beállítani. Például mi történjen, ha a felhasználó az ablak bal felső sarkában a kilépés(x) az átméretezés(\Box) vagy az elrejtés(_) gombra kattint. A java.awt-ben ezen függvények már szerepelnek. Az átméretezés, és elrejtés funkciók rendesen működnek, viszont a kilépésre alapban az ablak nem reagál. Ezért az osztály windowClosing függvénye felül van írva, hogy a super.dispose() függvényhívással az ablakot bezárja, és a lefoglalt erőforrásokat felszabadítsa.

Osztály: AkcioFigyelom implements ActionListener. Ezen osztályba tartozó függvény az actionPerformed(ActionEvent e) hívódik meg, amikor valamely az ablakban található objektumra kattintunk, például gombokra (Button), vagy adatbeviteli mezőkre (TextField). arg=e.getSource(); utasítás hatására megtudjuk mely objektum generálta az eseményt az aktualhash=arg.hashCode(); segítségével pedig kapunk egy integer típusú hash kódot, amely összevethető más objektumok hash kódjával. Az actionPerformed (ActionEvent e) függvényben az adatbeviteli mezők tartalmának vizsgálata történik. Megvizsgálja, hogy a kiadott mennyiség egész szám-e, vagy hogy a cikkszám, amit beírtunk szerepel-e az aru táblában. Megvizsgálja még, hogy az összes beviteli mező ki van-e töltve. Ha hiba történt, hibaüzenet jelenik meg, s ha minden rendben, megtörténik az adatok bevitele az uj_adat_bevitel() függvénnyel.

Adatbevitel_kiadas(String nev) a hasonló nevű osztály konstruktora.

A nev String lesz az ablak nev, a **super(nev)**; függvényhívás által. Itt történik az ablak objektumainak létrehozása. A TextFieldek Buttonok a java.awt hasonló nevű osztályainak példányaiként vannak létrehozva, a **new** paranccsal. Ha az objektumra később hivatkozunk, nevet kell adunk neki, de sok esetben (például a labelek esetében, amelyeket nem módosítunk) teljesen felesleges. De a gombok, és az adatbeviteli mezők tartalma felhasználásra kerülnek más függvényekben, ezért az Adatbevitel_kiadas osztály példányváltozóiként a konstruktoron kívül kell inicializálni ezeket. A Labelek (lab1-6) szolgálnak a cikkszámhoz tartozó áru adatainak megjelenítésére, és a hibaüzenetek megjelenítésére. A TextFieldek értelemszerűen az adatok beviteli helyei: A név, akinek a cikk kiadásra kerül, a gép/telephely, ahol felhasználásra kerül. A **mehet** gomb, pedig az adatok bevitelét indítja. Az objektumok ablakban elhelyezésére a

GridBagLayoutot használjuk, amely egyfajta táblázatos elrendezést biztosít, mint a GridLayout, de megengedi, hogy egyes TextFieldek hosszabbak legyenek, mint mások az oszlopban. c.gridwidth =GridBagConstraints.REMAINDER hatására a következő objektum kitölti a maradék helyet. c.fill = GridBagConstraints. BOTH; hatására függőleges és vízszintes irányban is kitölti az objektum a rendelkezésre álló helyet. Ez a panel kerül középre, míg a jobb és a bal oldali panel üres.

Az Adatbevitel_vasarolt.java működése nagyon hasonló az Adatbevitel_kiadas.java működéséhez. A fő eltérés, hogy ez az ablak egy másik táblába, a vasarolt táblába visz be adatokat. Ennek megfelelően más adatokat kér ez az ablak.

2.6.3. Kiadas_tabla_szerkesztese.java

Ez a fájl egy ablakot jelenít meg, amely a kiadas tábla és az aru tábla adatait mutatja egy táblázatban. Ez a táblázat több esetben is felhasználásra kerül, raktárosi jogosultság esetén kiadas tábla szerkesztése funkció hatására írásjoggal szerkeszthető, adminisztrátori jogosultság esetén kiadas tábla megjelenítése funkció hatására az adatokat megjeleníti az ablak, de csak rendezni lehet, adatokat változtatni nem. A kiadas tábla lekérdezése funkció estén, a lekérdezés feltételeinek megadása után szintén ez az ablak kezd működni, de már szűkített tartalommal, amely szintén nem módosítható. A táblázat alatt található a rendezések kiválasztásához egy legördülő menü, mellette egy a megjegyzések megjelenítésére használatos TextField, majd három gomb melyekkel a jobbra-balra lapozás, és a frissítés történik.

Osztály CegerFigyelom a MouseListenerben regisztrált objektumok egéreseménykezelése itt történik.

Osztályfüggvény: void mouseClicked(MouseEvent e) Ha a regisztrált objektumon kattintunk, ez a függvény hívódik meg. A táblázat kijelölt elemének színe zöldre vált, (frissítés esetén visszanyeri eredeti fekete színét). Ha viszont nem módosítható, a megjegyzés rovatban megjelenik, hogy az adott oszlop nem szerkeszthető. Osztályfüggvény: boolean mouseUp(Event esemeny, int x, int y) CegerFigyelom által nem kezelt objektumok egéreseményeit itt kezelni lehet. Ebben az esetben például az ablak jobb szélén kattintva, a módosított adatok változtatás előtti értékeinek visszaírása történik, vagyis a változtatások visszavonása.

Osztály: ItemFigyelom implements ItemListener Az ItemListener által a choice elem eseménykezelése itt történik. Choice típusú elemet alkalmazunk a rendezés kiválasztására, ami egy legördülő menü, amelynek több sora közül egyet lehet választani.

Osztályfüggvény: void itemStateChanged(ItemEvent e) Ha a choice objektumban aktuális elemet választunk, (kijelöljük a rendezés feltételét), ez a függvény működik. Itt a rend nevű String típusú változóba kerül a MySQL adatbázis megfelelő mezőjének neve, amely a SELECT SQL parancs részeként meghatározza a rendezés módját. Majd a lapmegjelenites függvényt hívja, amely az adatok megjelenítését végzi. A rendezés előtt megvizsgáljuk, történt-e változtatás és mentjük, amit menteni kell, mielőtt a rendezéssel más adatok kerülnek megjelenítésre.

Osztály: AkcioFigyelom implements ActionListener

Osztályfüggvény void actionPerformed(ActionEvent e) Az ablak gombjain való kattintások, TextFieldekben nyomott Enterek, mint események kezelését végzi. A gombok azonosítása hash kód alapján történik. Jobbra gomb esetén az inx változót növeli az oldalra kerülő sorok számával. Majd hívja a lapmegjelenites függvényt, amely az inx változó szerinti értéktől jeleníti meg a sorokat. A lapmegjelenites függvény hívása előtt szintén megtörténik a változások MySQL adatbázisba mentése. Balra gomb esetén a különbség, hogy az inx változó csökkentése történik.

Ha az esemény hash kódja az otex hashkódja (oldalszámot tartalmazó TextField) akkor megvizsgáljuk, a beírt adat lehet-e az új oldalszám. Ha igen inx-et módosítjuk majd a lapmegjelenites() függvény hívása következik.

A frissítés gomb esetén szintén a változások mentése történik, majd a lapmegjelenítés, az **inx** viszont nem változik. Majd ellenőrzés következik, mely megjegyzésként kijelzi, ha van hiányos sor. A bal oszlopban a törlésgombok ellenőrzése a ciklusban történik szintén hashkód szerint. A törlés gombra kattintva a **sortorles** függvényt hívjuk meg.

Osztályfüggvény: **boolean frissites()** Ezen függvénynél vizsgáljuk, történt-e változás a táblázatban. Összehasonlításokat végzünk a tárolt értékekkel, valamint az üres Stringgel (""). Ha a sorban nincs sorszám, de a sorban van adat, új sort kell létrehoznunk, az **uj_sor(i)** függvény által. Ha viszont a sorban van sorszám és adatváltozás is történt, akkor a sort felül kell írni a **sor_feluliras(i)** függvénnyel.

Osztályfüggvény: **boolean joe()** A táblázat sorait ellenőrzi, vizsgálja, van-e valamely sorban adathiány vagy 0-adat. Ha rossz formátumban történt az adatbevitel, akkor, a dátumba 0000-00- 00 String kerül, rossz mennyiség vagy mennyiség megadása hiányában a sor bevitele során 0 mennyiség kerül, String típusoknál pedig "" üres String. Ami jelzi, hogy az adat nem volt megadva, ilyen esetekben hibajelzést jelenít meg a megjegyzés területen. Mivel a függvény boolean false visszatérési értéket ad, ha volt hiba, ezt a továbbiakban fel lehet használni.

Osztályfüggvény: void mennyiseg_korrekcio(String csza, int uj_raktmennyiseg) Ez a függvény a csza cikkszámú cikk cikkszámának aktuális raktármennyiségét uj_raktmennyiseg értékre változtatja. Vagyis megnyitja a MySQL aru tábláját és kiadja az UPDATE aru SET aktme=" +uj_raktmennyiseg +" WHERE csz='"+csza+"'; parancsot.

Osztályfüggvény: int cikkszam_szerinti_raktmenny(String csza) Szintén az aru táblát nyitja meg és a csza cikkszám alapján megkeresi a megfelelő sort, és a cikkhez tartozó aktuális mennyiséget adja meg visszatérési értékként. Ha nem talált, akkor hibajelzés lép fel, a visszatérési érték pedig 0.

Osztályfüggvény: void sortorles(int aktsor, int i) Az i. sorban található aktsor sorszámú (a MySQL adatbázisban) sor törlése. Először az adatkapcsolatot hozza létre az adatbázissal, majd kiadja a DELETE FROM kiadas WHERE sor= "+aktsor+"; SQL parancsot melynek hatására a kívánt sor törlődik az adatbázisból. Ezután sorcsere történik, ha a törlés sikeres volt, és nem az utolsó sor törlése történt. Ez esetben a legutolsó sor megkapja a törölt sor cikkszámát, hogy ne legyenek kihagyások a sorszámokban, a sorszámok ugyanis itt nem szerkeszthetők. Ezután a cikk aktuális

mennyisége növekszik meg a törölt sorban volt kiadott mennyiséggel, az adatbevitelek és törlések így a cikk mennyiségét a valós eseményeknek megfelelően korrigálják.

Osztályfüggvény: void uj_sor(int aktsor) ez az osztályfüggvény új sort visz be az adatbázisba. Az adatbázissal létrehozza a kapcsolatot, majd az INSERT INTO kiadas SET sz='"+tex2[aktsor].getText()+"',mennyi="+tex4[aktsor] .getText()+",kinek='" +tex6[aktsor].getText()+"',gep='"+tex7[akt sor].getText()+"',datum='"+tex8 [aktsor].getText()+"',sor="+k+"; SQL parancsot adja ki, ahol a beírandó adatok a képernyő táblázatának aktsor sorszámú sorának TextFieldjeinek értékei. Ha nem történt hiba, a cikk aktuális mennyisége lesz csökkentve a kiadott mennyiséggel.

Osztályfüggvény: void sor_feluliras(int aktsor) ez az osztályfüggvény egy meglévő sort ír felül az adatbázisban. Az adatbázissal létrehozza a kapcsolatot, majd az UPDATE kiadas SET csz='"+tex2[aktsor].getText()+"',mennyi=" +tex4[aktsor].getText()+", kinek='"+tex6[aktsor].getText()+"', gep='"+tex7[aktsor].getText()+"',datum='"+tex8[aktsor].getText() +"' WHERE sor="+x0[aktsor]+"; SQL parancsot adja ki, ahol a beírandó adatok a képernyő táblázatának aktsor sorszámú sorának TextFieldjeinek értékei.

A kulcs az kiadas tábla sor változójának értéke az **x0[aktsor]**. Ha nem történt hiba, a cikk aktuális mennyiségét is változtatni kell. Több eset van, ha sem a kiadott mennyiség, sem a cikkszám nem változott, akkor nem kell a cikk aktuális mennyiségét változtatni, de ha a cikkszám vagy a kiadott mennyiség, esetleg mindkettő változott, akkor kell, és előtte meg kell határozni az új aktuális mennyiséget.

Osztályfüggvény: void lapmegjelenites() A változások után ez a függvény tölti fel az ablak táblázatát a MySQL adatbázis adataival. Az oldalszám beállítása után az adatkapcsolat létrehozása és a SELECT utasítás meghatározása következik. A rendezés feltétel rend értéke szerint, itt az elágazások sorával választjuk a megfelelő utasítást. A rendezések közül egyesek viszont két táblát is alkalmaznak, így nem lehet a rend String az SQL utasítás része, mint például a selfelt változó. SELECT kiadas.* FROM kiadas, aru "+selfelt+" and aru.csz=kiadas.csz ORDER BY aru.aktme"+fordit+"; A selfelt a lekérdezések esetén egy WHERE szűkítő utasításrészlet. A fordit String pedig a rendezés irányát jelöli, értéke lehet "DESC" vagy "". Ezután inx változóig kiolvassuk az adatokat a kiadas táblából, hogy a megfelelő oldalszámtól kezdődjön a megjelenítés, majd tovább olvasunk a megjeleníthető sorok számának megfelelő mennyiségű sort. Amely adatok már megjelenítésre, és az x tömbben tárolásra kerülnek. Ha nincs a szükséges számú sor, a fennmaradó sorokba üres String kerül. A szín is feketére vált a zöld kijelölő színről. Ezután az aru tábla adatai kerülnek kiolvasásra a megjelenített táblába, pedig azok kerülnek melyek cikkszáma, egyezik a kiadas táblából olvasottakkal. Előtte viszont törölni kell az oszlopokat, hogy régi adat ne maradjon benne.

Kiadas_tabl_szerkesztese(String nev, boolean writejog, String selectfelt). Az ablak konstruktora. A writejog dönti el, hogy megjelenítés, vagy szerkesztés-e a feladat, a **selectfelt**, pedig a SELECT utasítás WHERE része, amelynek lekérdezések esetén van szerepe, egyébként üres String. Ezen változók globálissá tétele után a panelek és azok részeinek létrehozása történik. Az m2 alsó panelban, sorban kerülnek felvételre a rendezés a megjegyzés és a gombok. A középső panel összetettebb, az m1 középső panelba a pd, p0-p8 panelek kerülnek, amelyek a különböző szélességű oszlopokhoz kellenek. Az m1 pedig a scroll scrollpanelba kerül mely a görgethetőséget teszi lehetővé, majd az m0 panelba kerül az oszlopnevek sora melyek TextField típusú objektumok. A fontbeállítás után sorperlap számú TextField sort hozunk létre a p panelek elemeiként. A MouseListenerbe és ActionListenerbe történő felvétel, valamint a szín és fontbeállítások is itt történnek, a megjelenítés vagy szerkesztés üzemmód szerint más módon. Ezután a kész táblázat adatokkal feltöltésre kerül, mint a lapmegjelenites() esetén, tehát a kiadas tábla sorai vannak kiolvasva először majd, az aru tábla sorai hozzárendelve cikkszám szerint, valamint a legnagyobb sorszám (maxsor) meghatározása és a sorok megszámolása (hossz) történik, az oldalszám megjelenítéséhez. Végül a panelek felvétele történik az ablak elemeiként.

2.6.4. Hiba.java

Hiba.java SQL hiba esetén előreugró ablakként jelenik meg.

Osztály: WindowFigyelom extends WindowAdapter. Az ablak átméretezését kezeli, és a bezárását végzi a void windowClosing(WindowEvent e) osztályfüggvény segítségével.

Osztály: AkcioFigyelom implements ActionListener. A gombnyomás hatására az ablak bezárása történik.

Hiba(String nev, int hibakod) Az ablak konstruktora. Az ablak neve is hibajelzés lesz, A hibakód egy feltétel paramétere, mely magyarázó szöveget jelenít meg az ablakban, de az ablak neve is egy hibajelzés lesz, mert a nev változó tartalma is egy hibaüzenet.

3. Felhasználói dokumentáció

3.1 Belépés



5. ábra. Raktárkezelés ablak.

A program indítása után egy kisebb ablak jelenik meg címe: Raktárkezelés. Ez az ablak a felhasználónevet, és a jelszót kéri, amely alapján meghatározza a jogosultságot az adatbázishoz való hozzáféréshez. A jelszót * jelek takarják, hogy mások ne láthassák, a felhasználónév azonban jól látható. Az adatbevitelhez nem Entert kell nyomni, hanem a bal egérgombbal egyet kell kattintani az Ok gombon.

Ha az általunk megadott jelszó vagy felhasználónév nem megfelelő, akkor a bal alsó sarokban a "Hiba: Nincs nyilvántartva" felirat jelenik meg. A helyes adatokat beírva a probléma megoldható, a próbálkozások száma nincs korlátozva. Előfordulhat, hogy egy vagy több kisablak jelenik meg Hiba történt a MySQL adatbázishoz való csatlakozáskor felirattal, ez azt jelenti, hogy a program a MySQL adatbázist nem tudja elérni, (például a hálózati kapcsolat nincs rendben, vagy a program telepítése során történt hiba) a hiba a programon belül nem orvosolható, a hibaüzenetek ablakai a 🗵 gombbal bezárhatók.

Ha a megadott felhasználónév és jelszó helyes, a jobb alsó sarokban a "Belépés rendben" felirat jelenik meg. A többi hét eddig halvány feliratú gomb felirata sötét lesz, ha raktárosi jelszavunk (jogosultságunk) van. Abban az esetben viszont, ha adminisztrátori jelszavunk (jogosultságunk) van, a baloldali két gomb a továbbiakban sem elérhető. A továbblépéshez a sötét feliratú gombok egyikére egyet kell kattintani a bal egérgombbal. A jobb alsó sarokban a "Kérem várjon" felirat jelenik meg, majd rövidesen megjelenik, a gombhoz rendelt újabb ablak, az előző pedig a háttérbe kerül.

A Raktárkezelés ablak bal felső sarkában a kilépés \boxtimes gomb hatására a program bezáródik az összes többi ablakkal együtt, amely a program működése során lett megnyitva. A többi ablak esetén, viszont a kilépés \boxtimes gomb hatására csak az az egy ablak záródik be.

3.2. Táblák adatainak megjelenítése

Sorszám:	Rakt.menr	Cikkszám:	Megnevezés: 1	Kiad.me	Egység:	Kinek:	Gép/Telephely:	Dátum:
1	2229	kk6	flex korong	2	db	Tóth Béla	Kamaz 3	2004-01-08
2	19	m10	csavar 10	1	kg	Kovács Pál	mühely	2003-01-06
3	39	m21	csavar 21	1	kg	Nagy Károly 🕴	Ford 5	2002-11-29
4	20	aa7	gumiabroncs Ifa	2	db	Fehér Zsolt	lfa 4 ASF-119	2002-05-09
5	2229	kk6	flex korong	1	db	Molnár Péter	Kamacsu 1	2002-11-24
6	38	f5p	festek piros	3	liter	Kerekes Sándor	Telep2	2003-01-25
7	62	f5s	festék sárga	2	liter	Sánta Lajos	Porta	2002-12-16
8	39	m21	csavar 21	3	kg	Telek Henrik	mühely	2003-02-24
9	1397	a2	gyertya	6	db	Molnár Péter	Kamacsu 1	2002-11-23
10	1397	a2	gyertya	6	db	Szabó Ferenc	Cat 3 /1	2003-12-18
11	39	m21	csavar 21	1	kg	Német György	Scania 1	2003-02-09
12	2229	kk6	fléx korong	2	db	Hegedüs András	Kamacsu 2	2002-12-14
13	606	kk1	géprongy	2	zsák	Tóth Kálmán	Kamaz 4	2003-11-28
14	85	f5z	csapágy lfa	2	db	Aradi József	lfa 2 (thj-634)	2003-02-02
15	362	hp11	heggesztöpáka	4	db	Balog Róbert	Tatra 4 (FFS-547)	2002-12-20
16	40	gcs	gumicsö 7cm	5	méter	Horváth Aladár	mühely	2002-11-09
17	2229	kk6	flex korong	4	db	Eröss Tibor	Kamaz 3	2002-12-11
18	1397	a2	gyertya	2	db	Rác Péter	Tatra 5	2003-01-09
19	141	es21	ékszíj lfa 21mm	2	db	Farkas Gábor	Kamaz 1	2003-02-28
20	2229	kk6	flex korong	1	db	Kiss Jóska	Lada 1500	2003-01-05
21	1397	a2	gyertya	5	db	Fekete Péter	Targonca 4	2002-10-30
22	20	aa7	gumiabroncs Ifa	4	db	Fehér Tibor	Konverter	2003-02-22
23	2229	kk6	flex korong	1	db	Faragó Lajos	Hitachy vontató	2002-11-09
24	20	aa7	gumiabroncs Ifa	2	db	Nagy Ferenc	Targonca 3	2002-09-01
25	141	es21	ékszíj lfa 21mm	5	db	Kiss Jakab	Furgon (ASD-156)	2002-12-16
26	362	hp11	heggesztöpáka 🦽	7	db	Varga Lajos	Gépjavító	2003-01-28
27	606	kk1	géprongy	1	zsák	Berkes András	Scania 6	2002-11-09
Gorszám	- Megjeg	yzés:			ţ.		Oldal: 1/2	< >

6.ábra. Kiadás tábla megjelenítése ablak.

Kiadás tábla megjelenítése. A táblázat jelenik meg, mely tartalmazza, hogy mely cikk mikor és kinek lett kiadva, tartalmazza a cikk nevét és cikkszámát, mennyi lett belőle kiadva, valamint az aktuális raktármennyiséget, a cikk kiadási egységét, a felhasználás helyét, vagyis a cikket felvevő mely gépen, telephelyen dolgozik, és a kiadás sorszámát (mint egy kartonszám). Mindegyik sor egy-egy ilyen kiadási eseményt jelent, egyszerre 50 sor jeleníthető meg a táblázatban. A felső sorban az oszlopok nevei az alsóban pedig a táblázat kezelésére alkalmas funkciók vannak.

A sorbarendezéshez a bal alsó sarokban található választómenü lefele mutató nyíl gombjára kell kattintani. Ekkor megjelenik egy kis lista, ebből kell kiválasztani, mi szerint legyen rendezve a táblázat. Sor, Mennyiség, Kinek, Gép, Dátum, stb. szerint növekvő sorrendben. A fordítva opció hatására, pedig a rendezés iránya megfordul, a rendezési feltétel viszont változatlan marad. A listaelemre kattintás után azonnal kezdődik a sorbarendezés. Mellette a megjegyzés sor található, amely a program üzeneteit jeleníti meg, például hibaüzeneteket, vagy az előző esetben például, hogy Rendezés. Egyszerre 50 sor jelenik meg az oldalon, a jobb szélen levő csúszkával lehet fel-le mozgatni ezt az oldalt, hogy az összes adat a lapról láthatóvá váljon.

Az adatbázis természetesen jóval több adatot tartalmazhat, a többi adatot a jobbra lent jobbra-balra mutató nyílgombokkal érhetjük el. Egy kattintás hatására a jobb egérgombbal a jobbra mutató nyílon, a következő oldal jelenik meg, például a megjelenített 50 adat után következő 51-100 adat jelenik meg, ha van ilyen. A balra mutató gombbal visszatérni lehet az előző 50 adathoz, vagyis az előző laphoz. A megjelenítendő oldal számát, az aktuális oldalszámot tartalmazó mezőbe beírva elérhető szintén a keresett oldal. Az egérrel a táblázatban is kattintani lehet, a kurzor megjelenik, de az adatok itt nem módosíthatóak. Az oszlopnév vagy a megjegyzés nem minden esetben fér el a neki fenntartott ablakban ilyenkor a bal egérgombbal odakattintva, majd a jobb kurzort nyomva a rubrika teljes tartalma megjeleníthető. A jobb felső is gombbal lehet az ablakból kilépni, ilyenkor nem az egész program záródik, hanem csak ez az ablak.

Vásárolt tábla megjelenítése. Szintén táblázat jelenik meg, mely tartalmazza, hogy mely cikkből mikor és mennyi lett vásárolva, milyen egységáron tartalmazza, a cikk nevét és cikkszámát, az aktuális raktármennyiséget, a cikkvásárlási egységét, és a vásárlás sorszámát. Az ablak működése teljesen megegyezik az előző ablak a kiadás tábla megjelenítése leírásánál mondottakkal. Csak itt a vásárlási kiadásokat összegzi, és a laponkénti végösszeget is megjeleníti.

Áru tábla megjelenítése. Ez az ablak az áru tábla adatait tartalmazza, tehát melyik árunak, mi a megnevezése, cikkszáma, aktuális raktármennyisége, mennyiségi egysége. Működése a fentebb leírtak szerint.

Ez a három tábla adminisztrátori jelszóval érhető el. Raktárosi jelszónál az adott tábla megjelenítése helyett szerkesztése szó van, amely kissé eltérő megjelenésű, és bővített funkciójú ablakokat jelent.

3.3. Táblák lekérdezése

🖉 Lekérdezések: kiadás	
ŀ	Ziadás tábla lekérdezése
Név =	
Gép/Telephely =	
Cikkszám =	kk1
Cikk megnevezése:	géprongy Cikk adatai
	<= Kiadott mennyiség <=
Raktári mennyiség:	606 zsák Lekérdezés két dátum között
2003 <	<= Dátum <= 2004.02.05
Adat bevitele:	Start — Lekérdezés végrehajtása
Lekérdezés tábla hívá	sa folyamatban!!
WHERE csz='kk1' and	datum>='20030000' and datum<='2004.02.05'

7. ábra. Kiadás tábla lekérdezése.

Kiadás tábla lekérdezése. Ez a menüpont arra szolgál, hogy a cikk kiadások közül csak adott feltételeknek megfelelő adatokat jelenítsen meg a tábla. Például adott cikkszámú áruból mikor, kinek, mennyi lett kiadva, egy adott évben. Az első sorba a cikket felvevő nevét kell írni pontosan, vagy lehet csak vezetéknevet megadni, például: Kovács. Alatta a Gép/Telephely a munkavégzés helye következik, majd a cikkszám. Rossz cikkszámot nem fogad el az alkalmazás, a cikkszámot beírva majd Entert nyomva a cikkszámhoz tartozó adatok, a cikk neve, egysége, raktári mennyisége, megjelenik. A program a kiadott mennyiséget is ellenőrzi, és csak pozitív egész számot fogad el. Az ellenőrzés történik a rublikában Entert nyomva, és a Start gomb benyomásakor is.

A dátumot többféle módon megadhatjuk, lehet megadni csak az évszámot (pld.: 2004), vagy egybeírva év hó nap formában, (pld.: 20040327) kötőjelekkel vagy pontokkal elválasztva (pld.:2004-03-27, 2004.03.27). A kiadott mennyiséget és a dátumot intervallumban is meg lehet adni, a jobb és baloldali rublikát egyaránt kitöltve, vagy lehet csak az egyiket kitöltve a másikat üresen hagyva. Adott számértékkel való egyenlőséget úgy adhatunk meg, dátum és kiadott mennyiség esetén, hogy jobb és a bal oldalra egyenlő értéket írunk. Természetesen több vagy akár az összes mező üresen hagyható.

Ha a lap adatai rendben vannak, a Start gombra kattintva kezdődik az adatok értelmezése, majd megjelenik egy táblázat mely a fent leírt kiadás tábla megjelenítése nevű táblázattal azonos, de csak a feltételeknek megfelelő adatokat tartalmazza. A hibaüzenetek az alsó sorban jelennek meg. Ha minden rendben, akkor a "Lekérdezéstábla hívása folyamatban" üzenet, és az SQL keresési feltétel jelenik meg, a Start gomb nyomása után pedig az eredménytábla megjelenik. Az eredménytábla bezárása után is ezen ablak nyitva marad, így újabb keresési feltétel adható meg.

Vásárolt tábla lekérdezése. Ez a menüpont arra szolgál, hogy a cikkvásárlások közül csak adott feltételeknek megfelelő adatokat jelenítsen meg a táblázat. Például adott cikkszámú áruból mennyi lett vásárolva, egy adott évben. Az első sorban a cikkszámot kell megadni. Rossz cikkszámot nem fogad el, a cikkszámot beírva majd Entert nyomva a cikkszámhoz tartozó adatok a cikk neve egysége raktári mennyisége megjelenik. A további hat mezőbe a kiadott mennyiséget, egységárat, és a dátumot lehet megadni. A program a vásárolt mennyiséget és egységárat is ellenőrzi, és csak pozitív egész számot fogad el. A vásárolt mennyiséget, az egységárat és a dátumot intervallumban is meg lehet adni, a jobb és baloldali rublikát egyaránt kitöltve, vagy lehet csak az egyiket kitöltve a másikat üresen hagyva. Adott számértékkel való egyenlőséget úgy adhatunk meg dátum és kiadott mennyiség esetén, hogy jobb és a bal oldalra egyenlő értéket írunk. Természetesen több, vagy akár az összes mező üresen hagyható. Ha a lap adatai rendben vannak, a Start gombra kattintva kezdődik az adatok értelmezése, majd megjelenik egy táblázat, mely a fent leírt vásárolt tábla megjelenítése nevű táblázattal azonos, de csak a feltételeknek megfelelő adatokat tartalmazza.

A lekérdezések mind raktárosi mind adminisztrátori jelszóval elérhetőek.

3.4. Cikk átvétele és kiadása

Adatbevitel raktárból áru kiad	ása 💶 🗵 🗶
Ra	ktárból cikk kiadása
Sorszám:	54 ————————————————————————————————————
Név:	Kovács Pál
Gép/Telephely:	Telep 2.
Cikkszám:	ккб
Cikk megnevezése:	flex korong Cikk adatai
Kiadott mennyiség:	6 db
Raktári mennyiség:	2229 db
Dátum:	2004.04.22
Funkció :	Adatbevitel: Laptörlés: Nyomtatás:
Adatbevitel: az adatok a adatbázisba kerülnek	az Beírt adatok törlése a lapon, de nem az
új sor az adatbázisban	: 54 adatbázisban
Adatbevitel!	

8. ábra. Raktárból cikk kiadása.

Raktárból áru kiadása. Ez az ablak arra szolgál, hogy új sort vigyünk be a készlet kiadásokat rögzítő adatbázisba. Valaki jön, például és ki akar vinni valamit a raktárból, ilyen esetben egy az adatbázisban nyilvántartott sorszámozott lap készíthető. Az ablakban a név a telephely a cikkszám a kiadott mennyiség és a dátum rublikák mindegyikét ki kell tölteni. A fentiekkel egyezően a program csak pozitív egész számot fogad el kiadott mennyiségként, a cikkszámot ellenőrzi, és kiírja a cikkszámhoz tartozó áru adatait, az adatbevitel gomb benyomásának hatására. A hibaüzenetek szintén az alsó sorba kerülnek.

Ha nem történt hiba, akkor azt jelzi, hogy új sor az adatbázisban és, hogy hányadik sor. Ez a sorszám fent is megjelenik, mikor az adatok az adatbázisba kerülnek, az adatbevitel gomb inaktívvá válik, a nyomtat gomb pedig előtűnik. A nyomtat gombra kattintva a megjelenik a nyomtatás menü, ha van csatlakoztatva nyomtató, majd a lapon megjelenítettek nyomtatásra kerülhetnek. Nyomtatáskor az alsó két sor üzenete törölve van. A laptörlés gombra kattintva az ablakba írt adatok kitörlődnek, az adatbevitel gomb pedig újra aktívvá válik. Ha nem akarunk nyomtatni csak új sort, bevinni, a laptörlés gombbal tehető a program újra alkalmassá ismételt adatbevitelre.

Készlet vásárlása. Ez az ablak arra szolgál, hogy új sort vigyünk be a készletvásárlásokat rögzítő adatbázisba. Szállítmány érkezik például, amelyet nyilvántartani kell, ilyen esetben egy az adatbázisban nyilvántartott sorszámozott lap készíthető. Ezen a lapon a cikkszámot a vásárolt mennyiséget, az egységárat és a dátumot kell beírni, a működés, a funkciók a fent leírtakkal (Raktárból áru kiadása) megegyező.

A fenti két ablak raktárosi jogkörrel elérhető.

3.5. Táblák adatainak szerkesztése

tábla szerke	sztése	0	Sciopilev			Njelot, va		rapiazat	. nugguleges gorg
Sortörlés	Sorszám:	Rakt.menr	Cikkszäm:	Megnevezés:	Kiad.me	Egység:	Kinek:	Gép/Telephely:	Dátum:
Törlés ->	1	2223	kk6	flex korong	2	db	Tóth Béla	Kamaz 3	2004-01-08
Törlés ->	2	19	m10	csavar 10	1	kg	Kovács Pál	mühely	2003-01-06
Törlés ->	3	39	m21	csavar 21	1	kg	Nagy Károly	Ford 5	2002-11-29
Törlés ->	4	20	aa7	gumiabroncs Ifa	2	db	Fehér Zsolt	lfa 4 ASF-119	2002-05-09
Törlés ->	5	2223	kk6	flex korong	1	db	Molnár Péter	Kamacsu 1	2002-11-24
Törlés ->	6	38	f5p	festek piros	3	liter	Kerekes Sándor	Telep2	2003-01-25
Törlés ->	7	62	f5s	festék sárga	2	liter	Sánta Lajos	Porta	2002-12-16
Törlés ->	8	39	m21	csavar 21	3	kg	Telek Henrik	mühely	2003-02-24
Törlés ->	9	1397	a2	gyertya	6	db	MolnárPéter	Kamacsu 1	2002-11-23
Törlés ->	10	1397	a2	gyertya	6	db	Szabó Ferenc	Cat 3 /1	2003-12-18
Törlés ->	11	39	m21	csavar 21	1	kg	Német György	Scania 1	2003-02-09
Törlés ->	12	2223	kk6	flex korong	2	db	Hegedüs András	Kamacsu 2	2002-12-14
Törlés ->	13	606	kk1	géprongy	2	zsák	Tóth Kálmán	Kamaz 4	2003-11-28
Törlés ->	14	85	15z	csapágy Ifa	2	db	Aradi József	lfa 2 (thj-634)	2003-02-02
Törlés ->	15	362	hp11	heggesztöpáka	4	db	Balog Róbert	Tatra 4 (FFS-547)	2002-12-20
Törlés ->	16	40	gcs	gumicsö 7cm	5	méter	Horváth Aladár	mühely	2002-11-09
Törlés ->	17	2223	kk6	flex korong	4	db	Eröss Tibor	Kamaz 3	2002-12-11
Törlés ->	18	1397	a2	gyertya	2	db	Rác Péter	Tatra 5	2003-01-09
Törlés ->	19	141	es21	ékszíj lfa 21mm	2	db	Farkas Gábor	Kamaz 1	2003-02-28
Törlés ->	20	2223	kk6	flex korong	1	db	Kiss Jóska	Lada 1500	2003-01-05
Törlés ->	21	1397	a2	gyertya	5	db	Fekete Péter	Targonca 4	2002-10-30
Törlóc >	22	20	aa7	gumiabroncs Ifa	4	db	Fehér Tibor	Konverter	2003-02-22
Rakt.menn		2223	kk6	flex korong	1	db	Faragó Lajos	Hitachy vontató	2002-11-09
Cikkszám		20	aa7	gumiabroncs Ifa	2	db	Nagy Ferenc	Targonca 3	2002-09-01
Megnevezé: Kiad menny	s F	141	es21	ékszíj lfa 21mm	5	db	Kiss Jakab _{//}	Furgon (ASD-156)	2002-12-16
Egység	· _	362	, hp11	heggesztöpáka	7	db	Varga Lajos	Gépjavító	2003-01-28
Kinek Gép/Telep		606	kk1	géprongy 🔥	1	zsák	Berkés András	, Scania 6	2002-11-09
Sorszám	 Megj 	egyzés: ki	ijelölés 10. so	orban			010	ial: 1/2 <	Frissít >

^{9.} ábra. Kiadás tábla szerkesztése.

Kiadás tábla szerkesztése. Ez az ablak a cikkek kiadása során létrejött adatbázis módosíthatására szolgál. Itt az adatok rendezetten megjeleníthetőek, a sorok közül törölni lehet, új sor bevihető, és a régi adat felülírható. A Táblázatban kilenc adatokat tartalmazó oszlop van, bal oldalon pedig a sortörléshez használható, törlés feliratú gombok. Egyszerre 50 sort jelenít meg a táblázatban. A felső sorban az oszlopok nevei az alsóban pedig a táblázat kezelésére alkalmas funkciók vannak. A sorbarendezéshez a bal alsó sarokban található választómenü lefele mutató nyíl gombjára kell kattintani. Ekkor megjelenik egy kis lista, ebből kell kiválasztani mi szerint, legyen rendezve a táblázat. Sor, Mennyiség, Kinek, Gép, Dátum, Megnevezés, Kiadott mennyiség, Egység szerint növekvő, csökkenő sorrendben.

A listaelemre kattintás után azonnal kezdődik a sorbarendezés. Mellette a megjegyzés sor található, amely a program üzeneteit jeleníti meg, például hibaüzeneteket, vagy az előző esetben például, hogy Rendezés. Egyszerre 50 sor jelenik meg az oldalon, a jobb szélen levő csúszkával lehet fel-le mozgatni ezt az oldalt, hogy az összes adat a lapról láthatóvá váljon. Az adatbázis természetesen jóval több adatot tartalmazhat, a többi adatot a jobbra lent jobbra-balra mutató nyílgombokkal érhetjük el. Egy kattintás hatására a jobb egérgombbal a jobbra mutató nyílon a következő oldal jelenik meg, a megjelenített 50 adat után következő 51-100 adat jelenik meg, ha van ilyen. A balra mutató gombbal visszatérni lehet az előző 50 adathoz, vagyis az előző laphoz, de oldalszámot írva a jobb alsó oldalszámot kijelző mezőbe az adott oldalt jeleníti meg, feltéve, ha van ilyen.

Az adatsorok a táblázatban módosíthatók, a módosítás a frissítés gomb benyomásának hatására kerül be az adatbázisba, az egérrel a táblázatban kattintva, a kijelölt mező tartalma zölddé válik. Nem minden adat szerkeszthető, amelyek nem szerkeszthetőek azok kék színűek, amelyek igen azok feketék a táblázatban. A sorszám automatikusan számítódik, az egység, a cikk megnevezése, raktári mennyisége, pedig az áru táblában szerkeszthető. Az adatbázisba új sor bevihető, csak a táblázat egy üres sorába kell az adatokat beírni.

A változtatások a módosítás vagy új sor beírása, akkor még csak a táblázatban jelenik meg, az adatbázisba, csak frissítés után kerül. Frissítés történhet, a frissítés gombra kattintva. Az egérrel a jobb szélen kattintva a változtatásokat mentés helyett visszavonhatjuk, de automatikus frissítés történik, például lapozáskor, rendezéskor, és kilépéskor. A frissítés után adatellenőrzés történik. A cikkszám létezik-e, a beírott mennyiség pozitív egész-e, hiányzik-e valamely adat. Ilyenkor hibaüzenet jelenhet meg, mely az adathiba, vagy az adathiány helyét, sorszámát, megadja. Rossz dátummegadás esetén 0000-00-00 érték lesz dátumként tárolva, amit szintén hibaként érzékel. A dátum megadható kötőjellel, ponttal elválasztva vagy egybeírva. Pld.: 20040405, 2004-04-05, 2004.04.05.

A törlés viszont azonnal történik. A törölt sor sorszáma nem marad kihagyva, hanem a legutolsó sor megkapja a kitörölt sor sorszámát. Ha a mező tartalma nem fér el a neki fenntartott területen ilyenkor a bal egérgombbal odakattintva, majd a jobb kurzort nyomva a rubrika teljes tartalma megjeleníthető. A bal felső 🖾 gombbal lehet az ablakból kilépni, ilyenkor nem az egész program záródik, hanem csak ez az ablak. A Raktárkezelés ablakból újabb ablak nyitható.

Vásárolt tábla szerkesztése. Ez az ablak, a cikkek vásárlása során létrejött adatbázis módosíthatására szolgál. Itt az adatok rendezetten megjeleníthetőek, a sorok közül törölni lehet, új sor bevihető, és a régi adat felülírható. Ezen tábla működése megegyezik a Kiadás tábla szerkesztése: táblánál leírtakkal. A táblában szerepelnek az áru adatai, megnevezés cikkszám egység, vásárolt mennyiség, aktuális raktármennyiség, egységár, sorszám, dátum, és az összeg a vásárolt mennyiség és az egységár szorzataként.

Áru tábla szerkesztése. Ez a táblázat a fenti két táblában szereplő cikkek adatainak módosítására szolgál. Ennek a táblának a működése is a Kiadás tábla szerkesztése pontban leírtaknak megfelelő, de itt csak négy oszlop van és ezek mindegyike szerkeszthető.

A program működése során az ablakok megjelenése gépfüggő, az operációs rendszer szín, képernyő beállításai hatással vannak a programra. Előfordulhat, hogy nincs elég memória, ilyenkor az ablak nem töltődik be, vagy csak részlegesen, ilyenkor célszerű más programokból kilépni, illetve a program más nem használt ablakait bezárni, és újra próbálkozni. Nem ajánlott továbbá a türelmetlen többszöri kattintgatás sem, mert hasonló helyzetet okozhat. Az ablakok betöltése kisebb teljesítményű gépek esetén néhány másodpercet igénybe vehet.

4. Összefoglalás

Az elkészült alkalmazás termelő cég, üzem raktárkészlet nyilvántartásának megkönnyítésére szolgál. Képes távoli MySQL adatbázissal való kapcsolattartásra, egyszerre több gépen futtatva kezelheti ugyanazt az adatbázist, különböző jogosultsággal. Az alkalmazás három táblában kezeli a raktározás adatait, egy negyedikben pedig a jogosultságokat. Raktárosi jogosultsággal az adatbázishoz adatok hozzáírhatók, az adatok megjeleníthetők, lekérdezhetők, rendezhetők, törölhetők, módosíthatók. Adminisztrátori jogosultsággal, az adatok csak megjeleníthetők, lekérdezhetők, rendezhetők, rendezhetők.

A program futtatásához ajánlatos egy legalább Pentium II (kb. 400MHz, 128MB RAM) teljesítményű számítógép. Alacsony teljesítmény esetén szükséges lehet más programokból való kilépés, és az ablakok is lassabban töltődnek be. Az alkalmazás JDK 1.2.2 JDeveloper-ben készült a MySQL JDBC kapcsolathoz a mysql-connector-java-3.1.10-stable-bin.jar-t használja.

Jövőbeni fejlesztési feladat hogy, a táblák, lekérdezések adatainak textfájlba mentését, nyomtatását, meg kell oldani. Az alkalmazás küllemét lehetne még javítani swing elemek alkalmazásával.