**Vendéglátóipari szoftver**

**Rakonczai Ádám**

**Elkészült funkciók:**

* Manager Functions -> Felhasználó felvétele (regisztráció)
* Manager Functions -> Felhasználó törlése
* Manager Functions -> Logikai nyomtató felvétele
* Manager Functions -> Logikai nyomtató törlése
* Manager Functions -> Logikai nyomtató módosítása (Pl: ki/be kapcsolása)
* Manager Functions -> Áfa felvétele
* Manager Functions -> Áfa törlése
* Manager Functions -> Asztal felvétele
* Manager Functions -> Asztal törlése
* Manager Functions -> Termék felvitele
* Manager Functions -> Termék törlése
* Manager Functions -> Termék módosítása (Pl: ár változás)

Ezekre a funkciókra a Connect osztályban vannak külön metódusok, amik az adatbázisba felvisznek, törölnek vagy módosítanak adatokat.

* Beléptetés.
* Admin és User felhasználói jog. (User nem tud a Manager Functions menübe lépni)
* Szabad és foglalt asztalok kilistázása.
* Termékek felvétele asztalokra.
* Asztal zárása.
* Asztal módosítása. (Plusz termékek felvétele)
* A „Managing” ablakban lehessen a termékekhez darabszámot állítani.
* Vendégszám módosítása.
* Termék készletből levonás, és figyelmeztető ablak dobás egy szám alatt.
* Terméket lehessen törölni az asztalról.
* Az adatbázis alapján napos, hetes, hónapos és éves kimutatásokat lehessen kérni.
* Asztal zárásakor nyomtasson egy asztali összegzést a kasszához.
* Az asztalra felvett termékek mentésekor, azon termékek amiknek ugyan az a nyomtatója, alkossanak listát, ami listát a nyomtató kinyomtat (Például ha van külön egy italpult, és egy konyha, akkor mentéskor az italokat nyomtassa ki egy listában az italpulthoz, az ételeket pedig egy másik listában a konyhához.)
* (esetleges) Termék kereső a „Managing” ablakba.

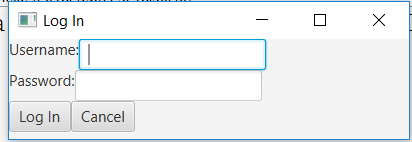
**Elkészítendő funkciók:**

* (esetleges) Két asztal összevonása.
* (esetleges) Asztal ketté bontása.
* (esetleges) Kiegészítő android app írása, amivel web szerveren keresztül vehetnénk fel termékeket az asztalokra. Ez azért lenne hasznos, mert akkor a pincérek egy tablet segítségével vehetnek fel rendeléseket ami rendelések rögtön meg is jelennének a fő alkalmazásban.

**Feladatspecifikáció**

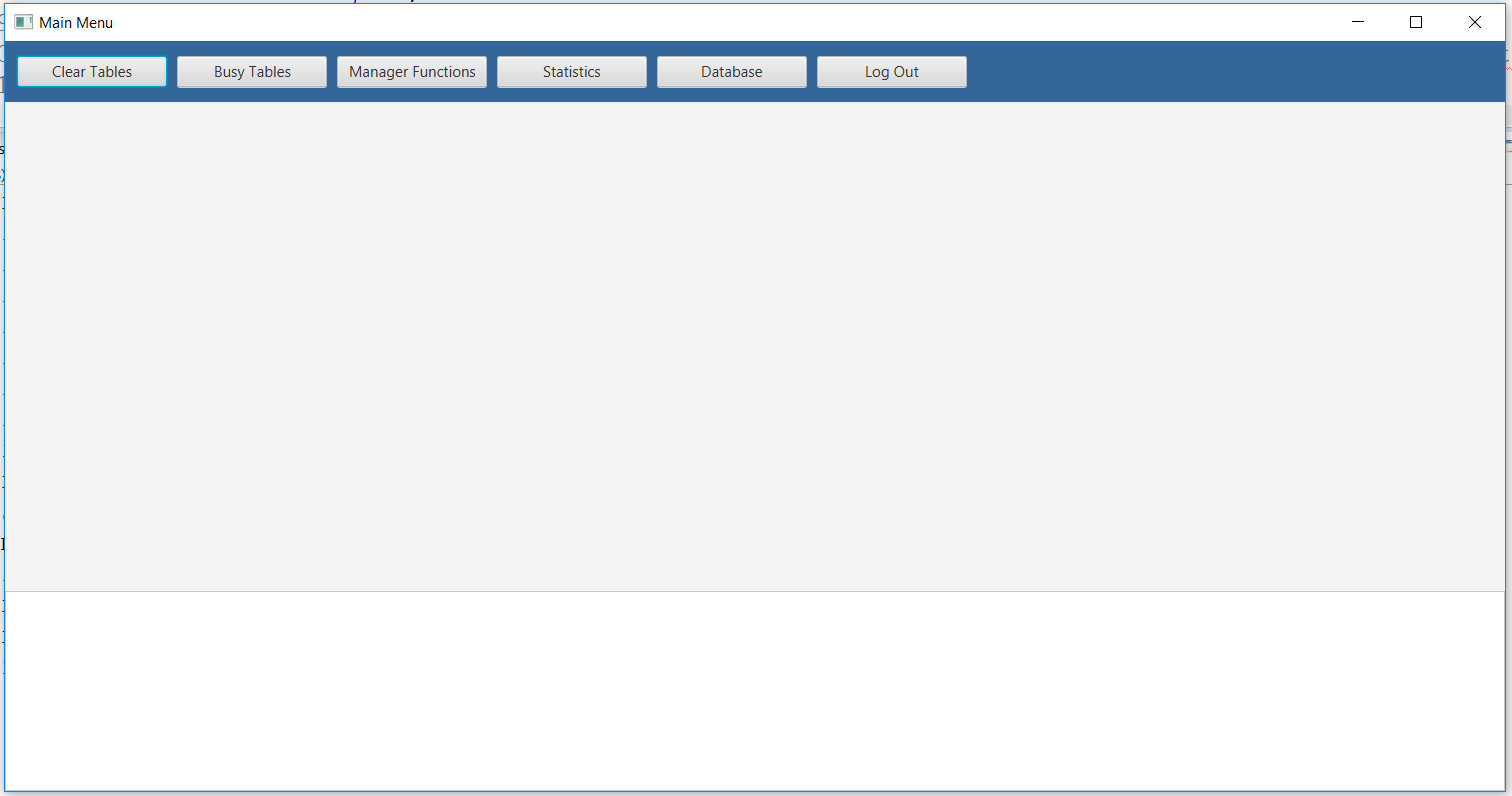
Osztályok:

* **Main.java**: Innen indul a program. Ez az osztály példányosit egy Connect típusú objektumot illetve meghívja LoginFrame osztály konstruktorát.
* **Connect.java**: Kapcsolatot létesít az adatbázissal, elkészíti az adatbázis táblákat, meghívja a pojo osztályok lista generálásához szükséges metódusokat, eldönti a bejelentkező felhasználó jogosultságát (User/Admin) illetve vannak további metódusai amik különböző eseményekkor hívódnak meg és az adatbázist menedzselik valamilyen formában.
* **LoginFrame.java**: Ez az első felületet megjelenítő osztály. Egy kis méretű ablakot hoz létre, ami a fő felületre való bejelentkezésre használatos (1. ábra). Vagy meghívja a MainFrame osztály konstruktorát, vagy ha hibás felhasználónevet/jelszót írunk be, figyelmeztető Alert ablakot hoz létre.



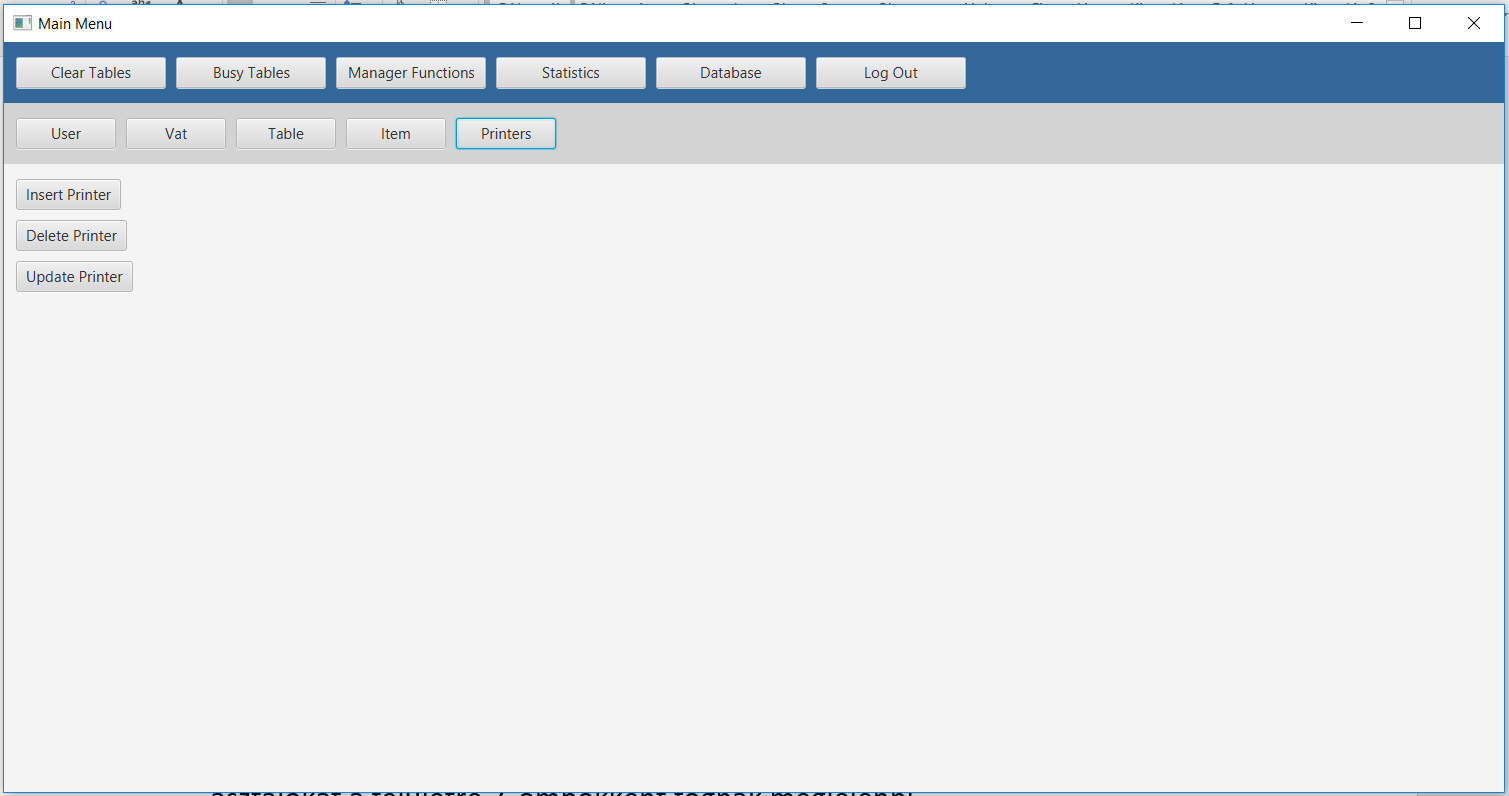
1. ábra: Bejelentkező felület

* **MainFrame.java**: (2. ábra) Ez a szoftver fő felülete. Itt az ablakot felosztom 3 részre. Az első egy menü rész az ablak tetején. A második a kozépső részt. A középső részen megjelenő tartalom változik a menüben található gombok hatására. A harmadik alsó rész pedig egy lista amiben a legutóbb . Ez csak a „Clear Tables” és „Busy Tables” menüpontokban jelenik meg.

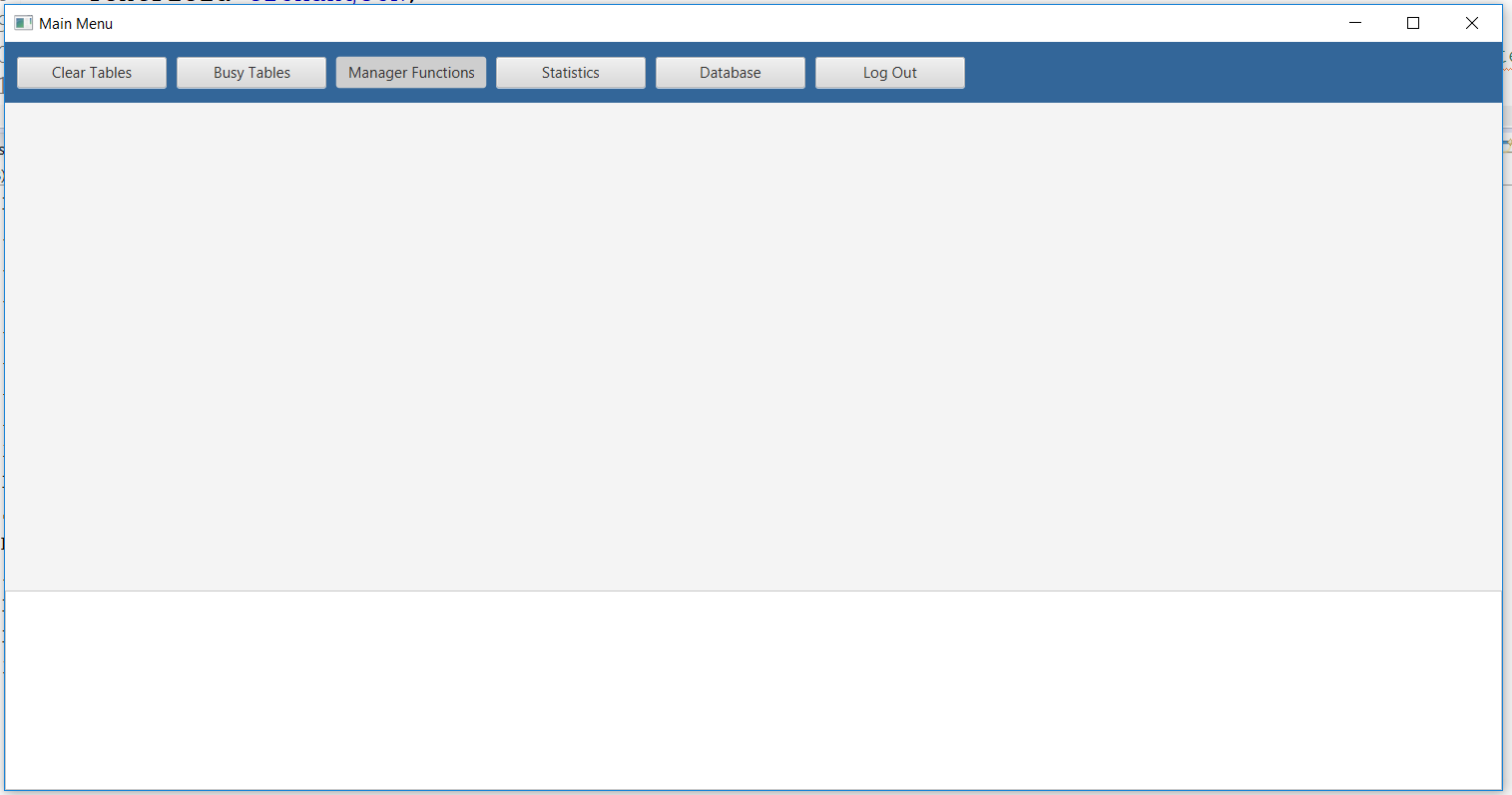


2. ábra: Main Menu

* **ManagerFunctions.java**: Ez az osztály a Manager Functions gomb megnyomásával hívódik meg. Ez alatt a menüpont alatt lehet menedzselni az adatbázisunkat (3. ábra) **.** Ezt csak olyan user nyithatja le aki rendelkezik admin joggal, ellenkező esetben belépés után szürkés színre vált, és megnyomásával semmi sem történik (4. ábra).

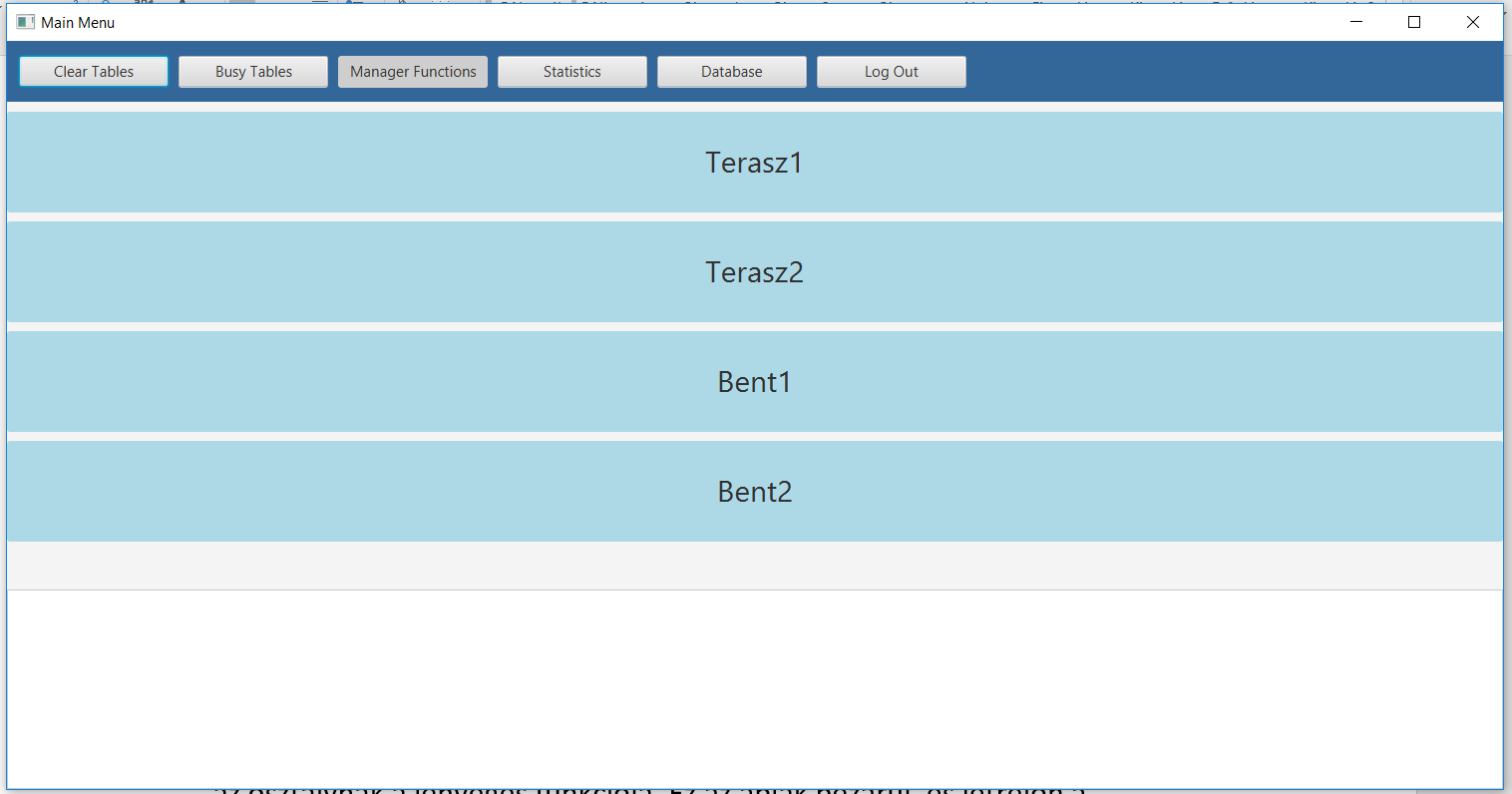


**3**. ábra: Admin joggal bejelentkezve



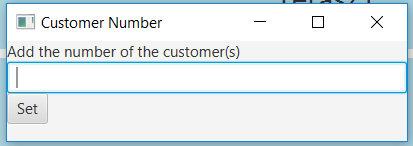
**4**. ábra: User joggal belépve

* **TableManaging.java**: Ez az osztály a „Clear Tables” és egyaránt a „Busy Tables” gomb megnyomásával is meghívódik. Ennek a feladata kilistázni az asztalokat a felületre, melyek gombokként fognak megjelenni (5. ábra).



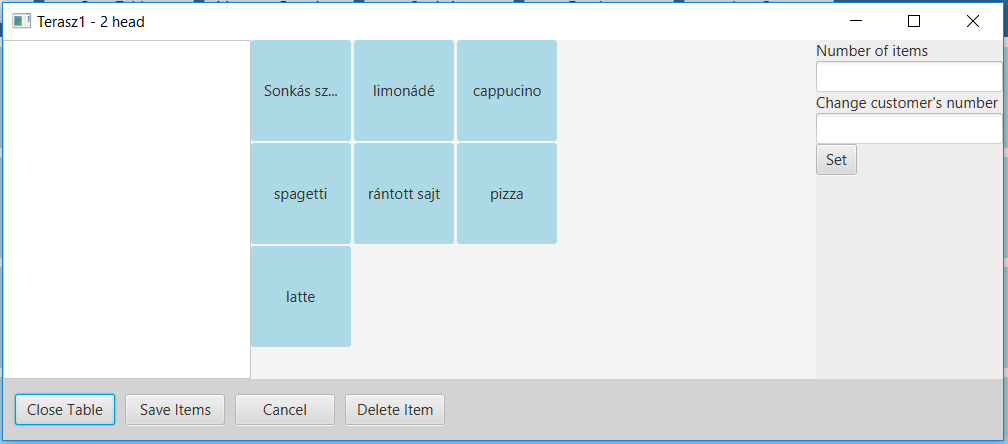
**5**. ábra: Asztalok kilistázása

* **ManageFrame.java**: Ennek az osztálynak a konstruktora, akármelyik asztal gombjára kattintva meghívódik. Ha a „Clear Tables” menü pontból választott asztalra kattintottunk akkor először megkérdezi a program egy kis ablak segítségével, hogy hányan ültek le az adott asztalhoz (6. ábra).



**6**. ábra: Vendégszám megadása

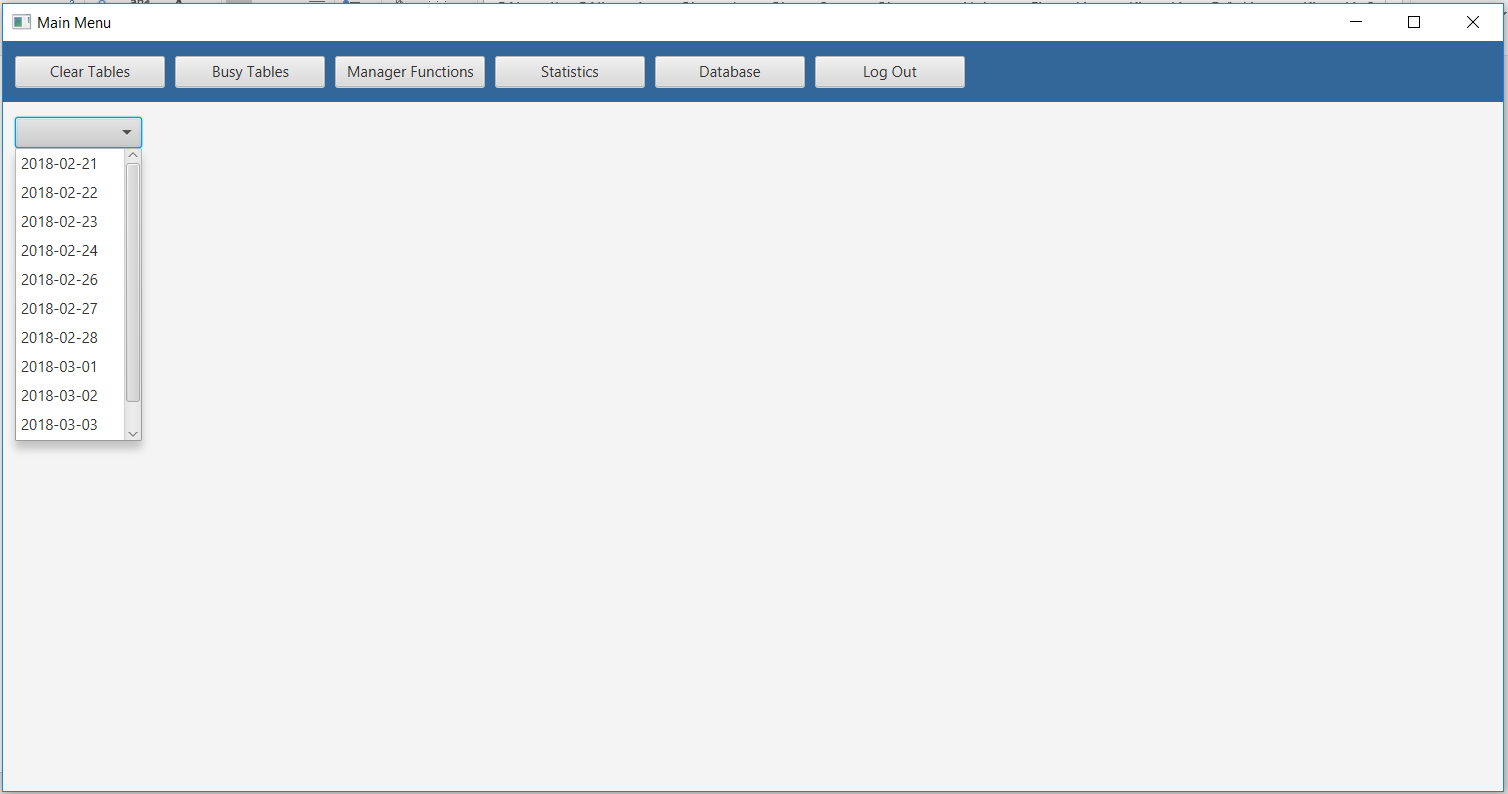
Ha ezt beírtuk, és rányomtunk a „Set” gombra, akkor utána jelenik meg ennek az osztálynak a lényeges funkciója. Ez az ablak bezárul, és létrejön a „Managing” ablak ahol a kiválasztott asztalra tudunk termékeket felvenni és törölni, illetve az asztal vendégszámát módosítani (7. ábra).



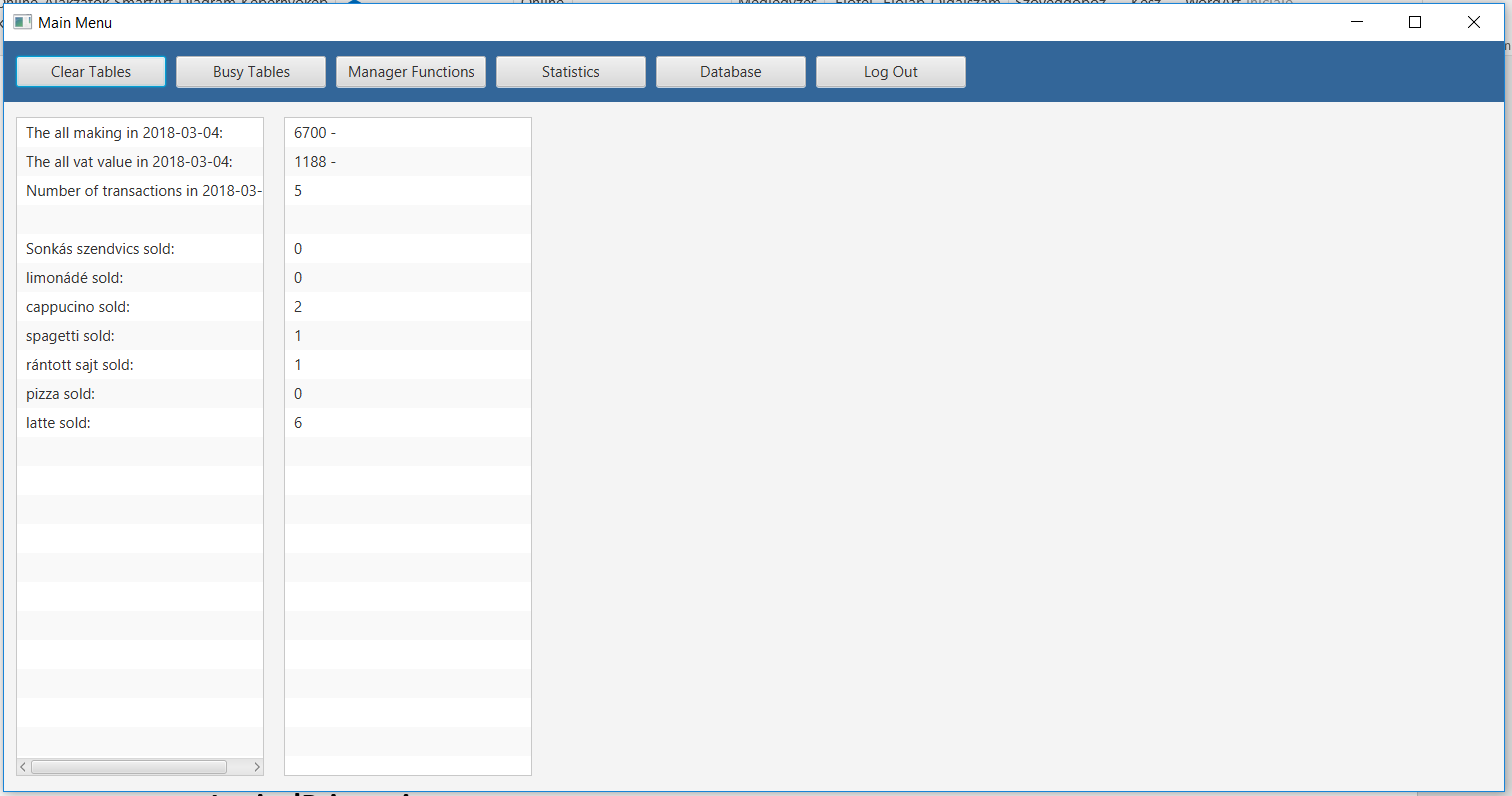
7. ábra: Termékek felvétele

Mikor a „Busy Tables” asztalok közül választunk, csak ez utóbbi ablak jön létre.

* **Statistics.java**: A „Statistics” menüpontra kattintva hívódik meg. Ennek az osztálynak köszönhetően napos, havi vagy éves statisztikai adatokat tekinthetünk meg (összes bevétel, összes adó, tranzakciók száma, egyes termékekböl hány darab fogyott) (8,9. ábra).

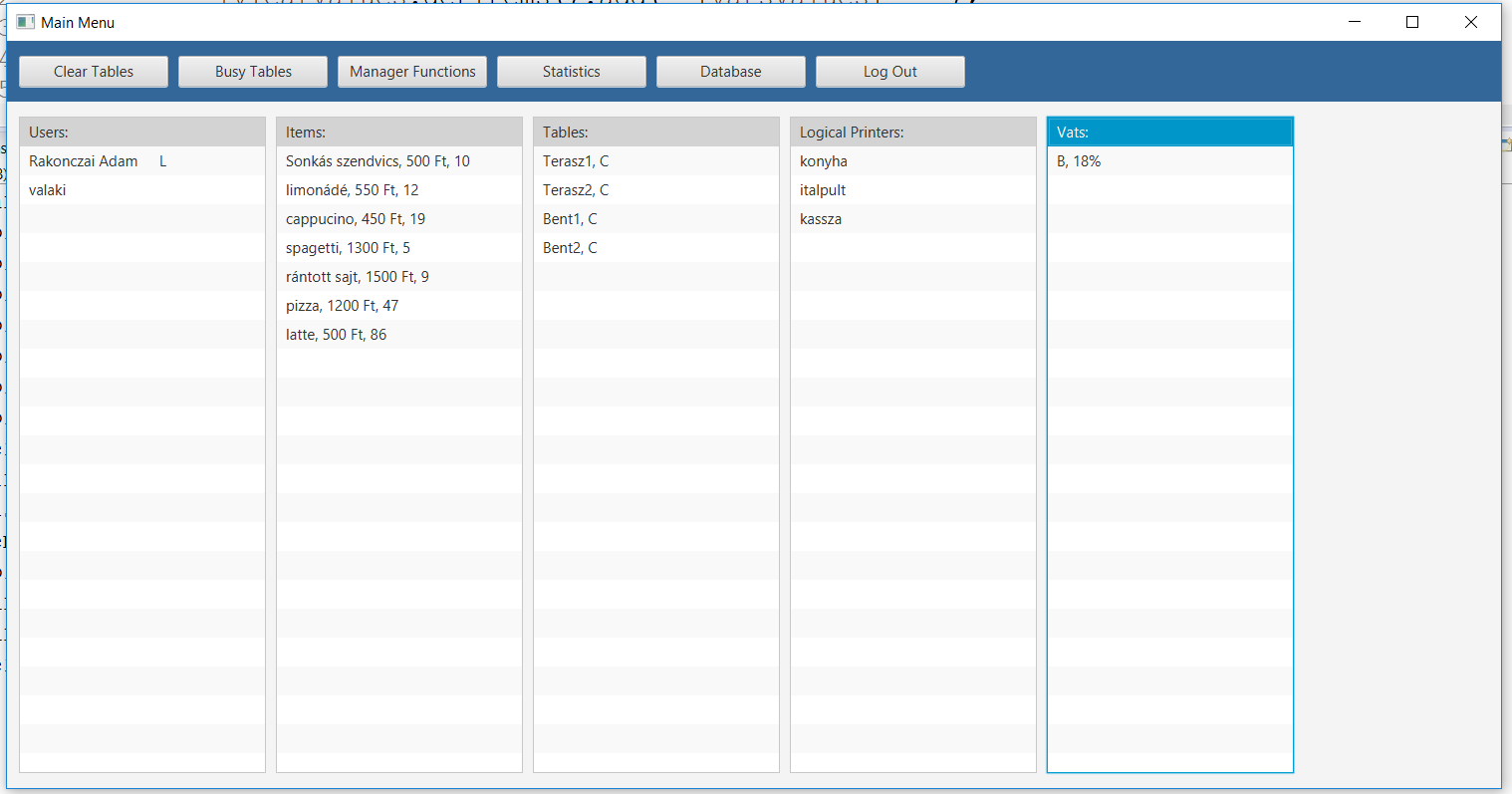
****

**8**. ábra: Nap kiválasztása



9. ábra: Statisztikai adatok

* **Database.java**: A „Database” menüpont megnyomásával hívódik meg. Fontosabb éppen aktuális adatokat jelenít meg az adatbázisról (10. ábra). Az aktuális felhasználó neve mellett lesz egy „L” betű jelzésként, a termékeknek látható az áruk és az aktuális készletük, az asztalok nevei mellett pedig a státuszuk látható (A szabad asztalok mellett „C”, míg a foglaltak mellett „B” betű).



10. ábra: Adatbázis adatok

A „Log Out” gomb lenyomásával ki tudunk jelentkezni, aminek hatására újra a LoginFrame fog megjelenni.

(További szükséges osztályok)

* **User.java**
* **LogicalPrinter.java**
* **Vat.java**
* **Table.java**
* **Item.java**
* **Transaction.java**
* **TransactionItem.java**

Adatbázis táblák:

(Az adatbázist MySQL szerveren hoztam létre.)

* + **Users** (user\_id, user\_name, login\_name, password, is\_admin)
  + **Logical\_Printers** (printer\_id, printer\_name, share, is\_on)
  + **Vats** (vat\_id, vat\_name, value)
  + **Tables** (table\_id, table\_name, status)
  + **Items** (item\_id, name, price, (fk)vat, (fk)logical\_printer, hoard, unit, edge);
  + **Transactions** (transaction\_id, (fk)table\_used, date\_open, date\_close, customer\_number, (fk)user, transaction\_close)
  + **Transaction\_Item** (ti\_id, (fk)transaction, (fk)item, price, value, vat\_value)

