# Dokumentáció

## A program rövid ismertetése

A program egy raktárkezelő alkalmazás, létre lehet hozni felhasználót és megadni, hogy alkalmazott-e. Az alkalmazásban van lehetőséged tárgyak hozzáadására, frissítésére, illetve törlésére attól függően, hogy erre megvan-e az engedélyed.

## Adatbázis

Az adatbázis öt táblát tartalmaz.  
Van egy felhasználó(users) tábla, ami eltárolja a felhasználó nevét és, hogy ott dolgozik-e vagy sem, mivel ha valaki nem ott dolgozó csak korlátozott jogosultságot kap.

A tárgyak(items) tábla tárolja a tárgyak elnevezését, márkáját, egy rövid leírást és hogy mennyi van az adott tárgyból. A tábla meghívja a kategóriákat(categories) ami a kategória nevét tartalmazza.

Az utolsó fő tábla a elhelyezési körlet(locations) tábla ez tárolja a helyszínek nevét és, hogy hány tárgyat lehet odatenni.

Az items és locations táblát a loctem tábla köti össze, ez arra szolgál hogyha több tárgyat egy helyre, illetve egy tárgyat több helyre szeretnénk tenni akkor az megoldható legyen.

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Automatikusan generált leírás

## Adatbázis összekapcsolása

Az adatbázis összekapcsolásához, először letöltöttük, illetve beimportáltuk a MySql.Data library-t. Ezután létrehoztunk egy DataBaseConnect class-t.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

Beállítjuk a DataBaseConnect class Fieldjeit, amik az adatbázis tulajdonságait tárolják.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

Létrehoztunk egy Initialize methodot, ami inicializálja a connection stringet, amit aztán meghív a Constructor.

## Regisztráció/Bejelentkezés

A bejelentkezés és a regisztráció nem sokban különböznek egymástól.

Míg a bejelentkezés csak megnézi, hogy létezik-e az adott néven felhasználó és ha létezik, akkor megnézi, hogy létezik-e employee jogosultsággal.

Viszont a regisztráció abban az esetben, ha a felhasználónév még nem létezik feltölti az adatbázisba a megadott nevet és hogy milyen jogosultsága van utána meg automatikusan bejelentkezteti a korábbi adatokkal.

## SQL Injection elleni védelem

Az SQL injection ellen a projectünkben paraméterezett lekérdezéseket (parameterized queries) használtunk. Ennek a lényege, hogy a lekérdezésbe beillesztett adatokat paraméterként adjuk át. Például:



Ezután paraméterként átadjuk az értékeket:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

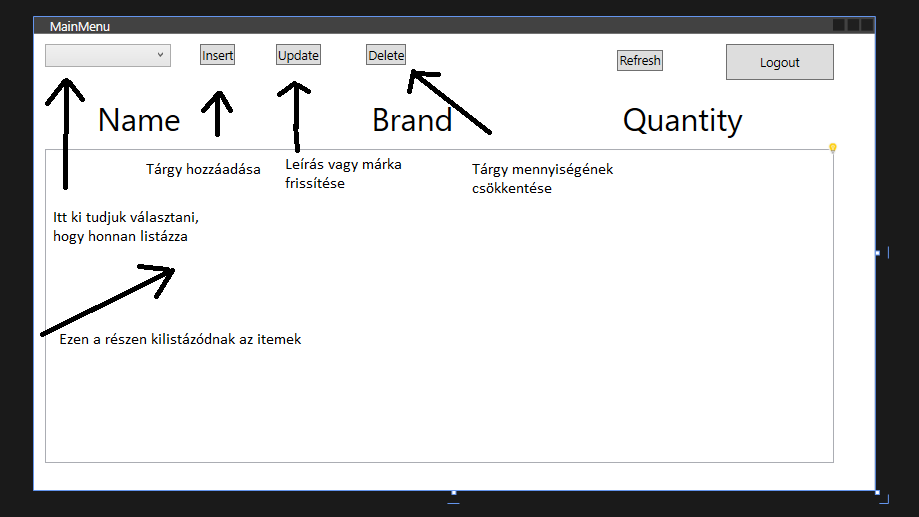
A Query így fog kinézni, amibe a programunk behelyettesít:

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

## Főmenü

A belépés után megnyílik a főmenü



## Tárgy hozzáadása

Tárgyat csak az ott dolgozó adhat hozzá, a hozzáadás működése egészen egyszerű a felhasználó megadja a szükséges információkat és hogy hova szeretné tenni a tárgyat és hogy mennyit.  
Abban az esetben, ha ugyan azon a helyen már van ugyan olyan tárgy akkor csak növeli az ott lévő tárgyak mennyiségét, viszont, ha nincs akkor csak feltölti az adatbázisba az adatokat és a darabszámot.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

1. ábra: Ha van ilyen tárgy már az adatbázisban

Természetesen a program megnézi, hogy az adott területen még van-e férőhely, mivel, ha az be van telve vagy be fog telni ezzel a hozzáadással akkor nem engedi, hogy odategyünk annyi tárgyat és visszaadja a felhasználónak, hogy mennyi szabad hely van az adott területen.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

2. ábra: Ha nincs ilyen típusú elem az adatbázisban

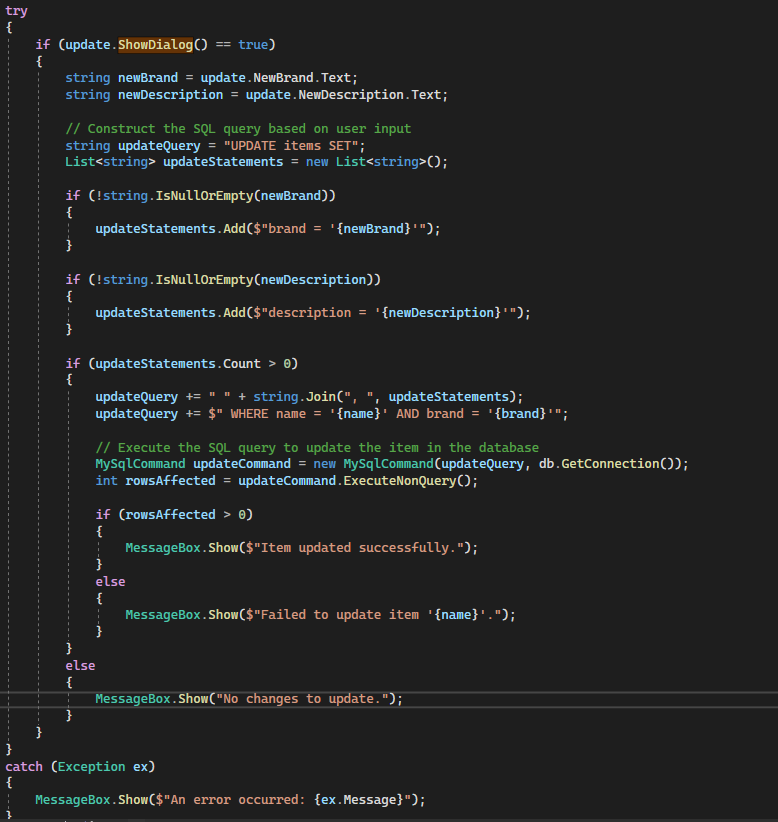
## Tárgy frissítése

A főmenün kiválasztott tárgy leírását és márkáját lehet itt változtatni, ha a felhasználó nem add meg értéket akkor nyilvánvalóan nem fog változás történni. Illetve lehet csak a leírást vagy csak a márkát külön frissíteni.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

3. ábra: Először átkérjük, a user által kiválasztott itemet a listboxbol, illetve ha a user nem választott ki semmit, akkor visszaküldjük és értesítjük, hogy válasszon



4. ábra: Ezt követően felépítjük a query-t az alapján, hogy a felhasználó a márkát, a leírást vagy mindkettőt szeretné frissíteni (updatelni).

## Tárgy törlése

A törlést nem szó szerint kell érteni itt csupán a kiválasztott tárgy mennyiséget lehet csökkenteni az adott helyen, ez mindenki számára engedélyezett.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

5. ábra: Kilistázzuk az elérhető tárgyakat (itemeket) ahonnan lehet törölni

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

6. ábra: Csökkentjük a tárgy mennyiségét a raktárban, illetve a helyen ahol tároltuk, illetve a delete click method fogja meghívni

Lekérjük az item location id-ját a GetLocationId method-dal, amit a Delete\_Click method fog használni.

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, szoftver látható

Automatikusan generált leírás

7. ábra: Legvégül lefut a delete click, ha megnyomjuk a gombot

Tartalom

[Dokumentáció 1](#_Toc149830664)

[A program rövid ismertetése 1](#_Toc149830665)

[Adatbázis 1](#_Toc149830666)

[Adatbázis összekapcsolása 2](#_Toc149830667)

[Regisztráció/Bejelentkezés 3](#_Toc149830668)

[SQL Injection elleni védelem 3](#_Toc149830669)

[Tárgy hozzáadása 4](#_Toc149830670)

[Tárgy frissítése 5](#_Toc149830671)

[Tárgy törlése 7](#_Toc149830672)