Szoftverarchitektúrák

Raktárkezelő rendszer

*Készítette: Kamrás Márton (NEPTUN)*

*Kiss Rebeka Enikő (EVLIR3)*

Tartalom

[Összefoglaló 3](#_Toc436690705)

[Rendszerterv 4](#_Toc436690706)

[Architektúra 4](#_Toc436690707)

[Adatbázis terv 6](#_Toc436690708)

[Biztonság 7](#_Toc436690709)

[Felhasználói felületek 8](#_Toc436690710)

[Bejelentkezési felület 8](#_Toc436690711)

[Raktár menedzselő felület 9](#_Toc436690712)

[Tétel típus kezelő felület 10](#_Toc436690713)

[Sablon menedzselő felület 11](#_Toc436690714)

[Szállítmányozási felület 12](#_Toc436690715)

[Funkciók 13](#_Toc436690716)

[Általános funkciók 13](#_Toc436690717)

[Alkalmazotti funkciók 14](#_Toc436690718)

[Menedzseri funkciók 17](#_Toc436690719)

[Tulajdonosi funkciók 17](#_Toc436690720)

[Telepítési útmutató 23](#_Toc436690721)

[Üzemeltetői 23](#_Toc436690722)

[Felhasználói 23](#_Toc436690723)

[Továbbfejlesztési lehetőségek 24](#_Toc436690724)

[Verziókövetés 26](#_Toc436690725)

[Irodalomjegyzék 26](#_Toc436690726)

# Összefoglaló

A Raktárkezelő rendszer egy böngészőből elérhető alkalmazás, melynek segítségével a cégek a figyelemmel kísérhetik raktárkészleteiket. Az alkalmazás web szerverre való telepítés után bárhonnan elérhető, így nem igényel speciális felszerelést.

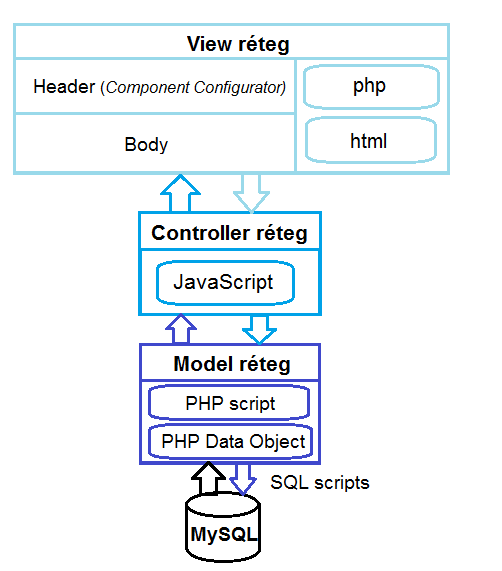
A raktárakat egyszerre, felsorolás szinten is megtekinthetik, vagy kiválasztva az adott raktárt (esetleg egy minta alapján újat hozzáadva) annak készleteit, a raktárak közti készlet mozgásokat monitorozhatják. Az egész napos számítógép előtt ülést elkerülendő, az alkalmazás bizonyos eseményekről értesítést küldhet. Továbbá az örökmozgók kedvéért egy mobilra optimalizált honlapot is biztosítunk.

A biztonságért a jelszavaknál egy egyszerű titkosítás felel, míg a kommunikáció https felett történik.

A felhasználói felület kialakításánál az elsődleges szempont a funkciók könnyű elérése volt, hiszen az alkalmazást használóknak sok adatot kell egyszerre átlátni.

# Rendszerterv

## Architektúra



1. ábra

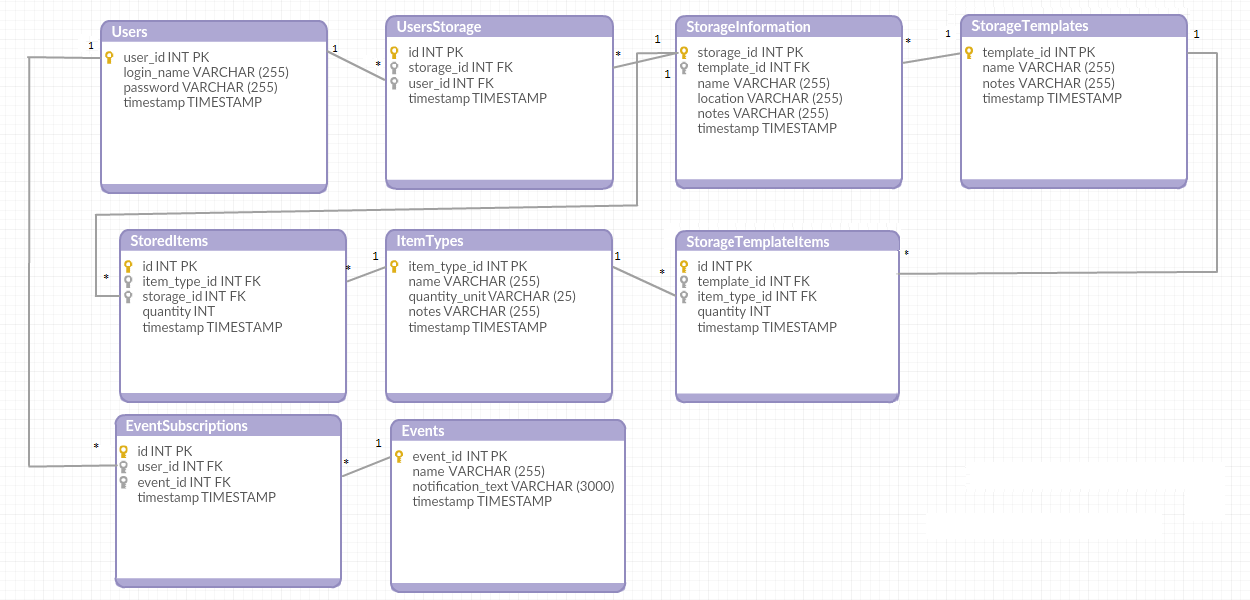
A program a *Model-View-Controller* architektúrát követi, kiegészítve a *Model* réteget egy *Wrapper* mintával, valamint a *View* rétegben egy *Component Configurator* mintát is használ.

A *Model* réteg kommunikál a *MySQL* szerverrel, ahol az adatbázis tárolódik. A *Model* réteg osztályai felelősek az adatok tárolásáért. A *Model* rétegben *Wrapper* mintát követve használtuk a *PHP Data Objects* kiegészítőjét, melynek segítségével az *SQL injection* típusú hibák könnyen elkerülhetőek. A *PDO* utasítások beburkolják az egyszerű SQL utasításokat, átalakítva őket paraméterezett előre elkészített utasításokká, így a paraméterek átadásakor az esetleges utasítások nem hajtódnak végre, mert egyszerű stringként kerül feldolgozásra.

A *Controller* réteg tartalmazza az üzleti logikát, a *Model* réteg osztályainak adatai alapján. Közvetlenül már nem fér hozzá az SQL utasításokhoz, csak a *Model* által publikált metódusokat ismeri. Ide tartoznak a JavaScript file-ok (pl. Events/js/script.js, Statistic/js/script.js)

A *View* réteg pedig csak a megjelenítésért felel, a megjelenítendő adatokat a modellektől kéri el. A felhasználó által módosított mezőket a *Controllernek* adja át, aki elvégzi a szükséges műveleteket. A megjelenítő rétegben további mintaként megjelenik a *Component configurator*. Ez az egyes felhasználók által látott modulok betöltésekor kap szerepet. A betöltendő modulokat a program egy konfigurációs fájlból olvassa fel, s ebben a fájlban van leírva az egyes felhasználói szerepek által látható modulok listája is. Ez futás időben is változtatható. Az egyes modulok betöltésért egy script felel, amely az oldal frissítésekor lefut. Végül pedig a *View* rétegben a szép megjelenítésért a css stíluslapok felelnek.

## Adatbázis terv



2. ábra

Az adatbázis 4 fő részre bontható:

A felhasználókra (*Users* tábla). Ebben a táblában mentődnek el a felhasználó belépési adatai. A felhasználó név egyszerű karaktersorozatként tárolódik, de a jelszónál, csak a jelszóhoz tartozó hash-t mentjük le.

A raktárak (*UsersStorage, StorageInformation, StorageTemplateItems, StorageTemplates* táblák). Ezekben a táblákban találhatóak meg a raktárak leírásai (*StorageInformation*), a felhasználók által használt raktárak (*UsersStorage*), illetve az új raktárak létrehozásához szükséges raktárminták (*StorageTemplates*). A *StorageTemplateItems* tartalmazza a raktárkészlet leírását, raktárminta – termék típus – mennyiség hármasokból felépítve.

A raktárak leírásainál meg kell adnunk, hogy milyen minta alapján készült a raktár, mi a neve, hol található. Emellett egyéb jegyzetet is megadhatunk. A különböző raktármintáknak nevet adhatunk, s megjegyzést csatolhatunk hozzájuk.

Az egyes felhasználókhoz tartozó raktárakhoz nem csatolunk más információt.

A tárolandó termékek/készletek (*StoredItems, ItemTypes*). Az *ItemTypes* a lehetséges termékek leírását adja meg (név, mértékegység párosokkal, és esetleges megjegyzésekkel), míg a *StoredItems* a konkrét készleteket raktáranként (miből, melyik raktárban, mennyi van).

Végül pedig az események (*Events, EventSubscriptions*). A felhasználók különböző eseményekre iratkozhatnak fel, s ezekről értesítést kaphatnak. A lehetséges eseményeket az *Events* tábla, míg az egyes feliratkozásokat az *EventSubsriptions* tábla tartalmazza. Egy esemény névből és egy értesítési szövegből áll.

Emellett minden tábla tartalmaz egy *timestamp* mezőt, melyben az adott rekord hozzáadásának vagy módosításának időpontja mentődik.

## Biztonság

A felhasználók jelszavainak védelme „*Secure Salted Password Hashing*”-gel történik, ami a következőképpen működik: a felhasználó megadja a jelszavát, a rendszer hozzáad egy előre meghatározott sót, majd ennek a karaktersorozatnak veszi a hash leképezését, s ezt a leképezést hasonlítja össze az adatbázisban tárolttal. Ha egyezik, az authentikáció sikeres és a felhasználó bejelentkeztetése megtörténik.

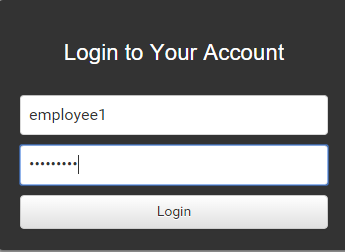
A hálózati forgalom lehallgatása ellen *HTTPS* protokollt használ a program, melyet a konfigurációban határoztunk meg. A *https* széles körben elterjedt protokoll, ezért részletesen most nem ismertetjük[[1]](#endnote-1).

Az adatbázissal rendelkező rendszerek esetén szükség van még *SQL injection*[[2]](#endnote-2) elleni védekezésre is. A program ezt *PDO* (*PHP Data Objects*)[[3]](#endnote-3) alkalmazásával védi ki. A *PDO* egy objektum az adatbázis kapcsolatok, lekérdezések, stb. kényelmes, hatékony, átlátható kezelésére. A *PDO*-val ellátott modellek már nem publikálnak SQL parancsokat a fentebbi rétegeknek, emellett *prepared statement*-eket használ az alkalmazás, ami az elsődleges védelmi vonal az *SQL injection* ellen.

A felhasználó jogosultságai szerint változik az elérhető modulok, gombok láthatósága a felhasználói felületet, így a jogosulatlan hozzáférést igyekszik a program megakadályozni. Az egyes modulokat hozzárendelhetjük szerepkörökhöz, így meghatározva, hogy ki láthatja. Ez konfigurációs fájlból, illetve *on-the-fly* módon is meghatározható a *Component Configurator* mintának köszönhetően.

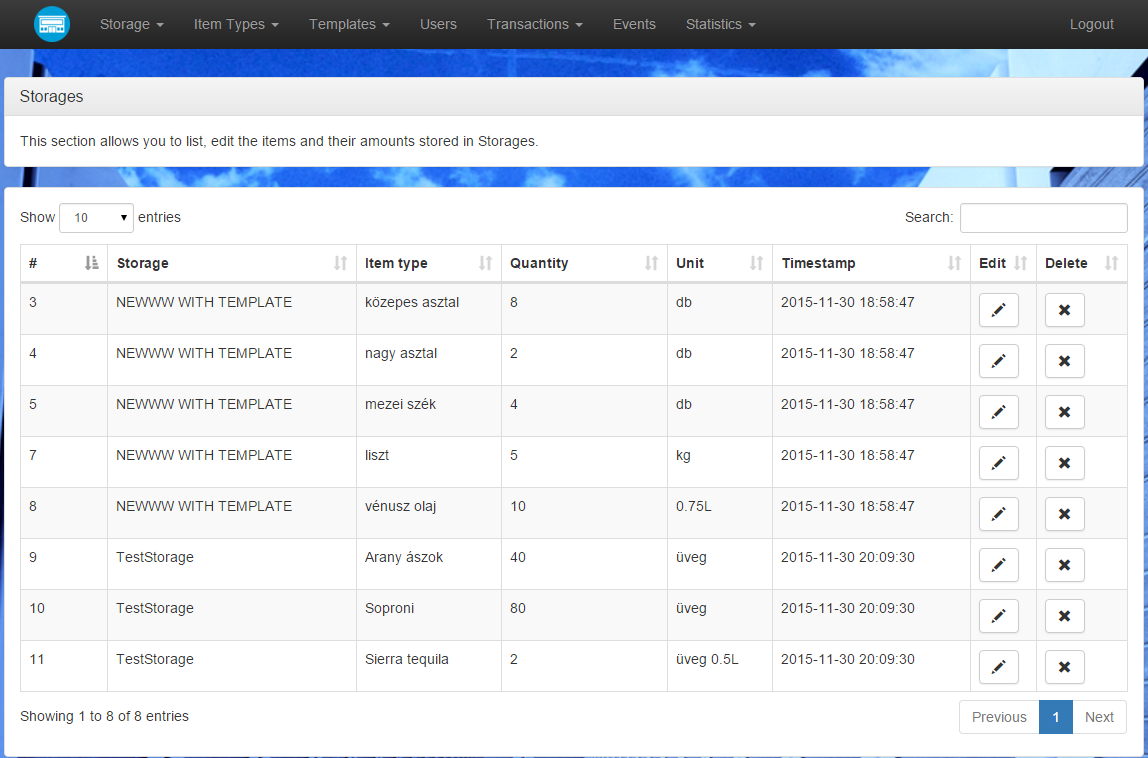
## Felhasználói felületek

### Bejelentkezési felület

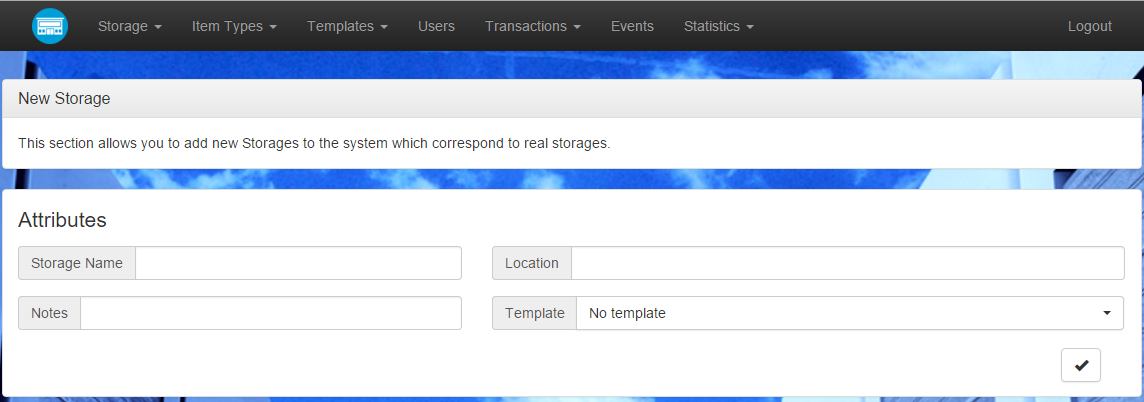


3. ábra

### Raktár menedzselő felület

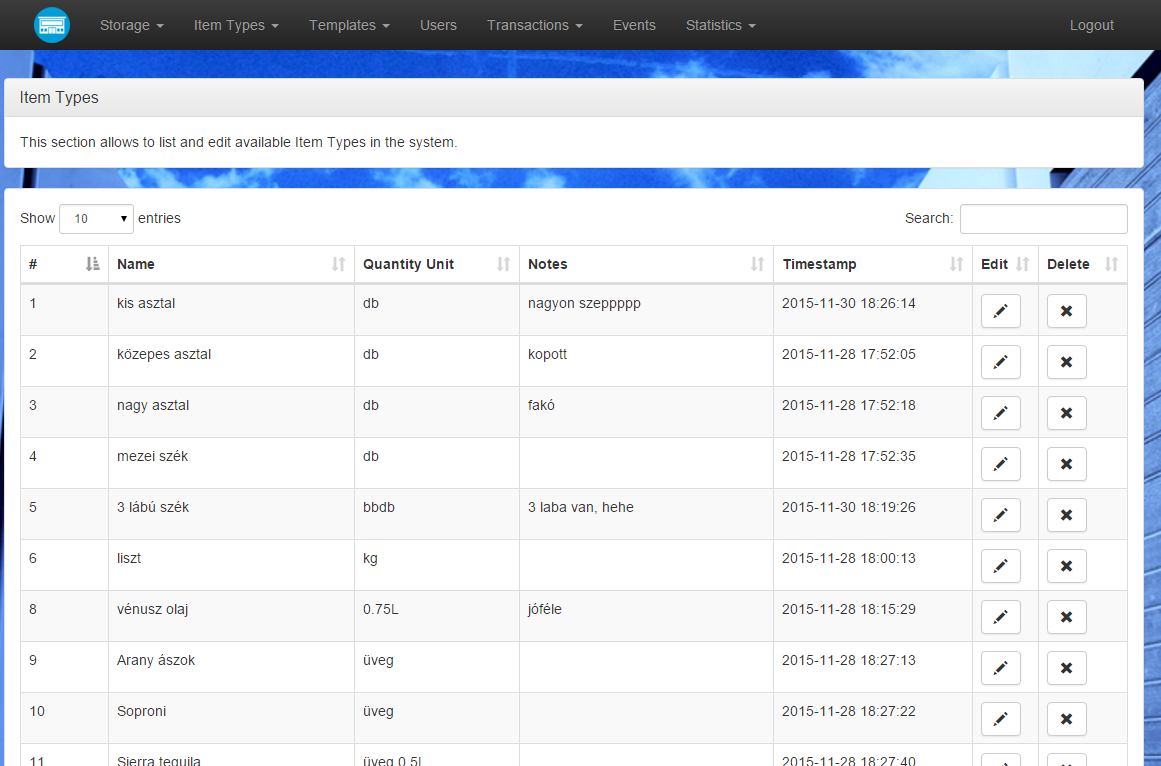


4. ábra

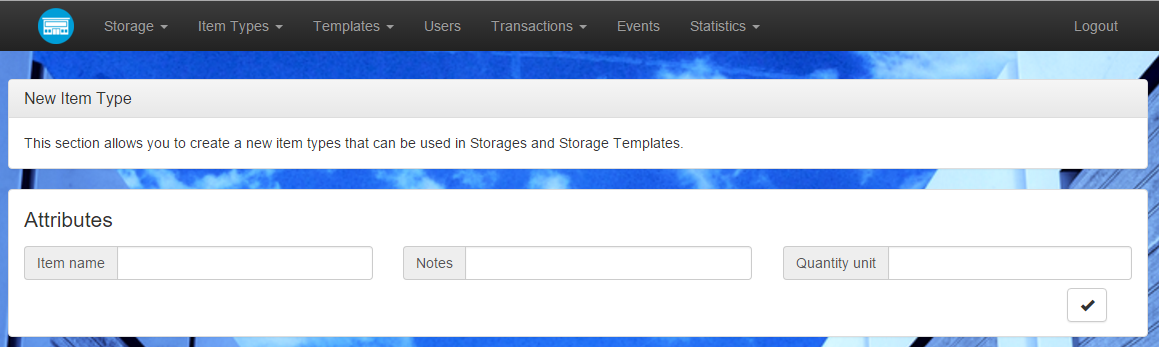


5. ábra

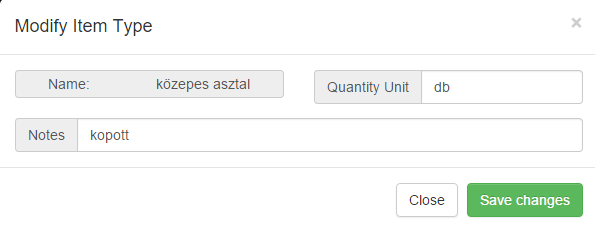
### Tétel típus kezelő felület



6. ábra

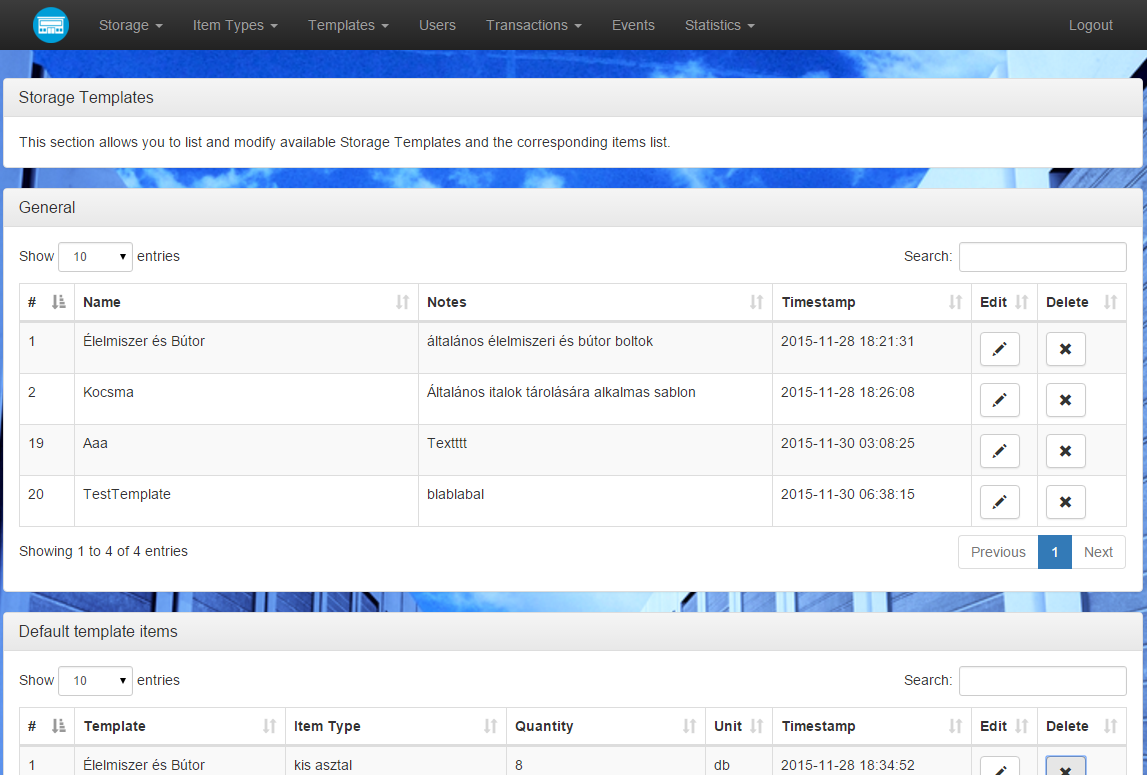


7. ábra

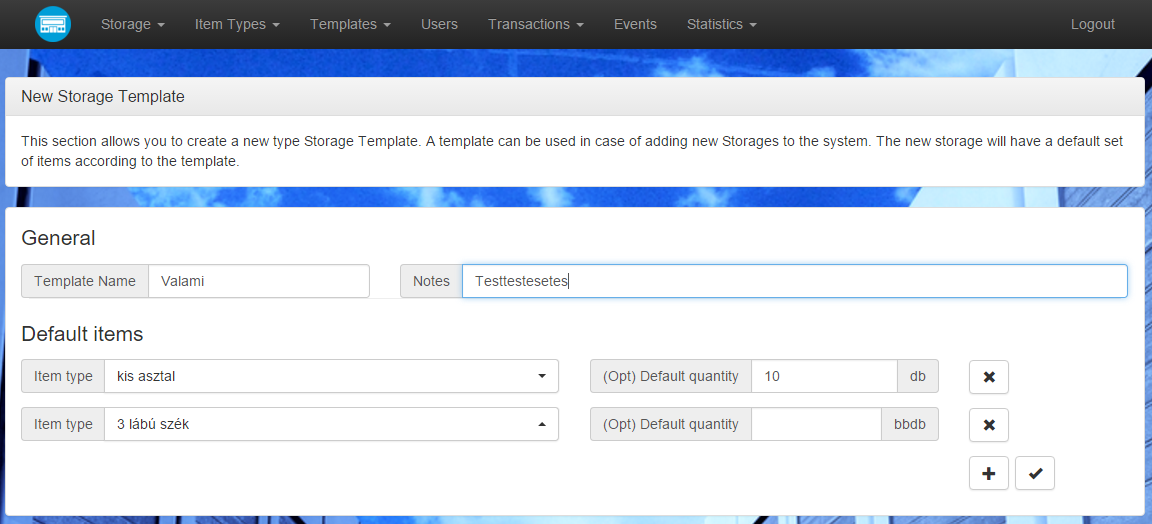


8. ábra

### Sablon menedzselő felület

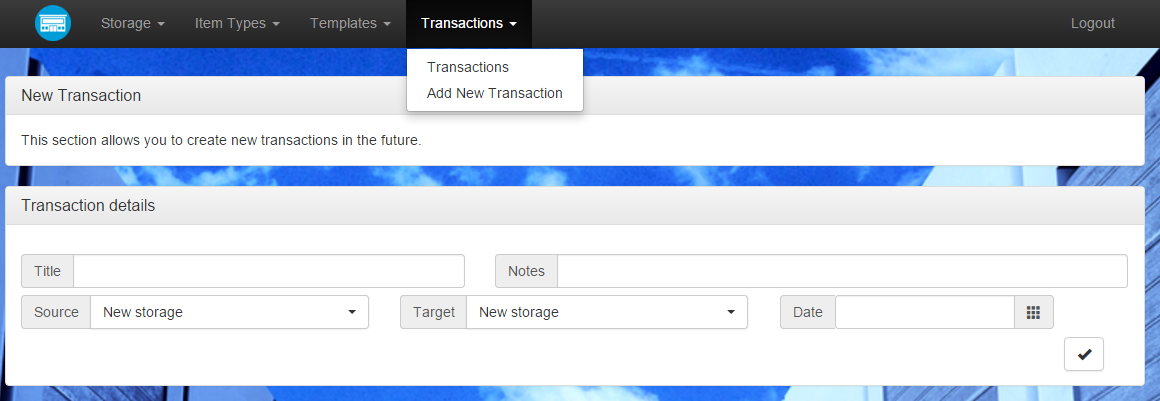


9. ábra



10. ábra

### Szállítmányozási felület

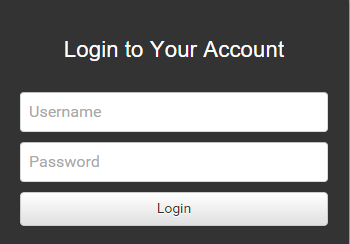


11. ábra

## Funkciók

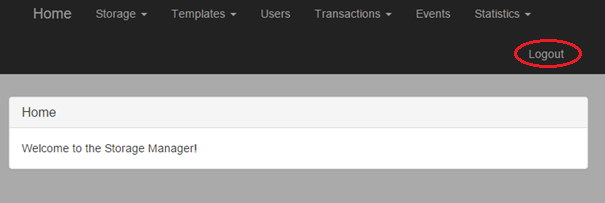
### Általános funkciók

Bejelentkezés: a felhasználó belép a felületre, megadja felhasználó nevét és jelszavát, a rendszer az authentikáció után tovább engedi.



12. ábra – bejelentkezési felület (host.cim/sm/Login)

Kijelentkezés: a felhasználó már korábban belépett a felületre és 10 percig inaktív. A rendszer automatikusan kijelentkezteti, és visszalép a belépő felületre.  
A felhasználó korábban belépett a felületre, még nem telt el 10 percnyi inaktivitás, s a „Logout” gombra kattint. A rendszer kijelentkezteti, és visszalép a belépő felületre.



13. ábra – A bejelentkezés utáni felület

### Alkalmazotti funkciók

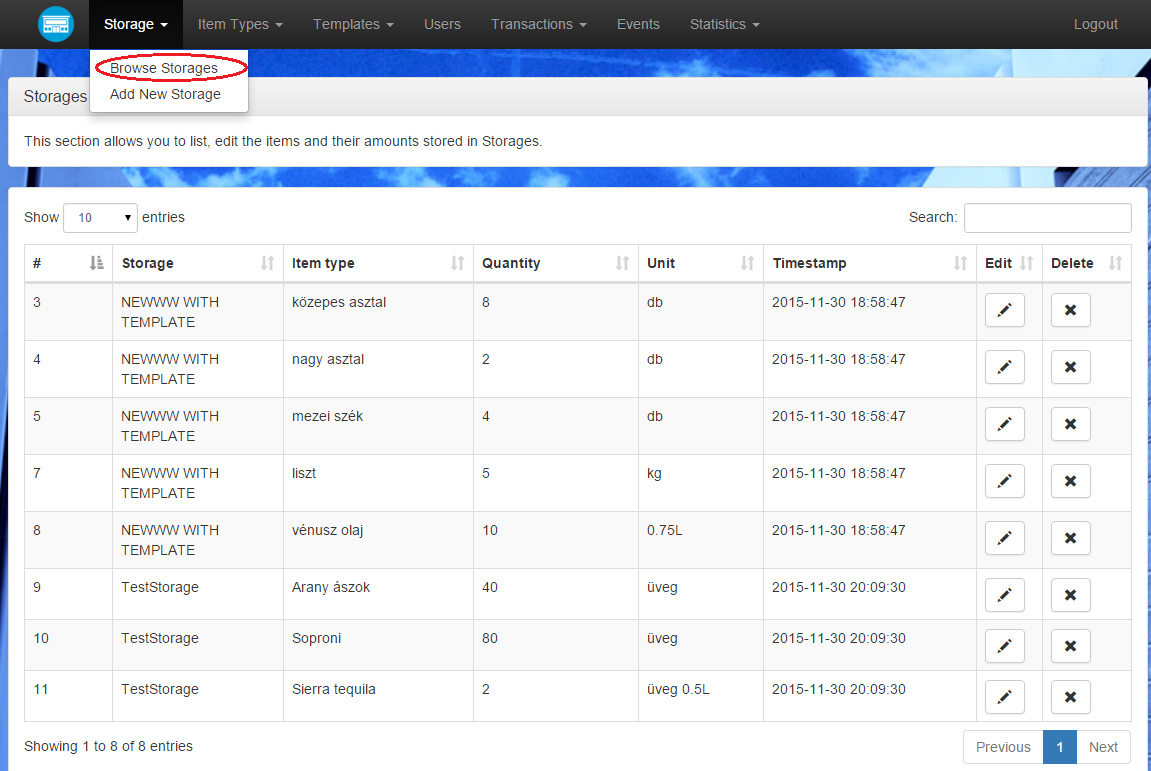
*A következő funkcióknál feltételezzük, hogy a felhasználó bejelentkezett és alkalmazott (employee) szintű jogosultságai vannak.*

Összes raktár lekérdezése: a felhasználó kiválasztja a *Browse Storages* menüpontot.



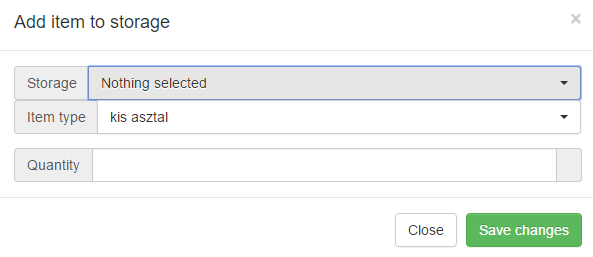
14. ábra – Raktárak listázása

Adott raktárhoz tartozó információk lekérése: (raktárban lévő tételek neve, készlet): a felhasználó kiválasztja a *Browse Storages* menüpontot, majd kikeresheti a számára releváns raktárat, s megtekintheti a benne tárolt tételeket.



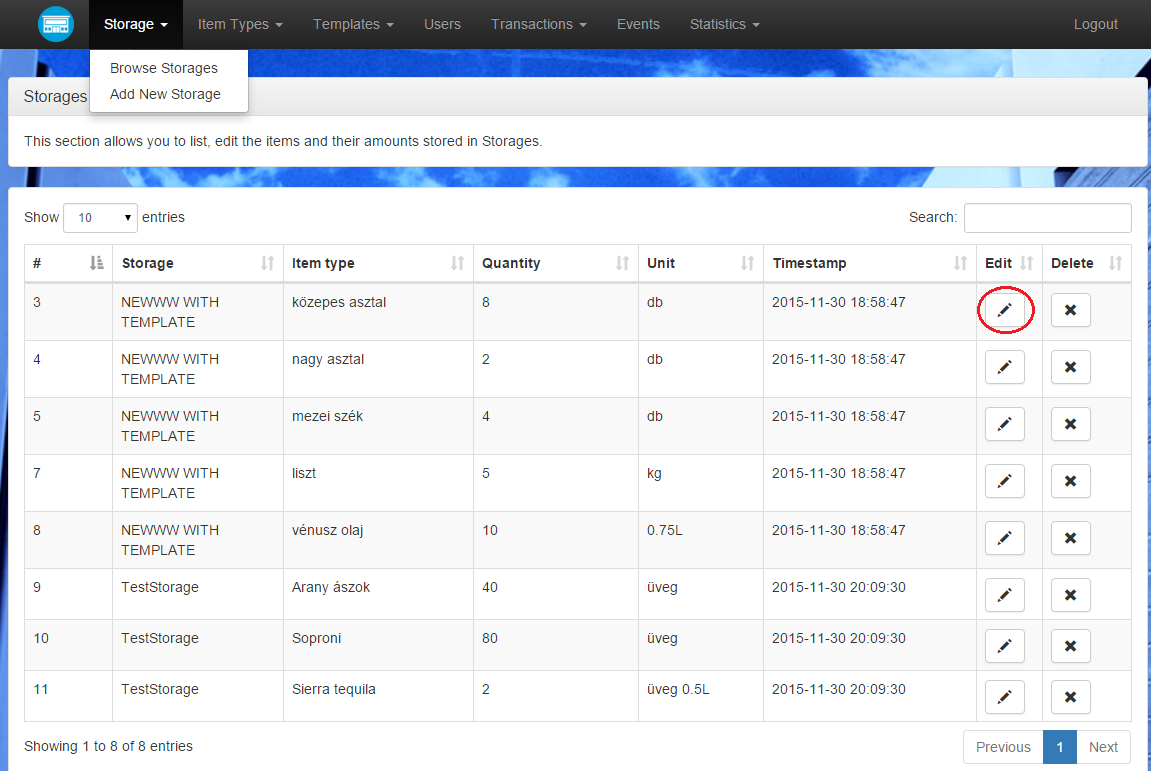
15. ábra – Raktárak adatainak megtekintése

Új tételek felvétele: a felhasználó a *Browse Storages* menüpontot választja, majd az oldal alján található hozzáadás (+) gombra kattint. A felugró ablakban kitölti a megfelelő adatokat és elmenti a változtatásokat.



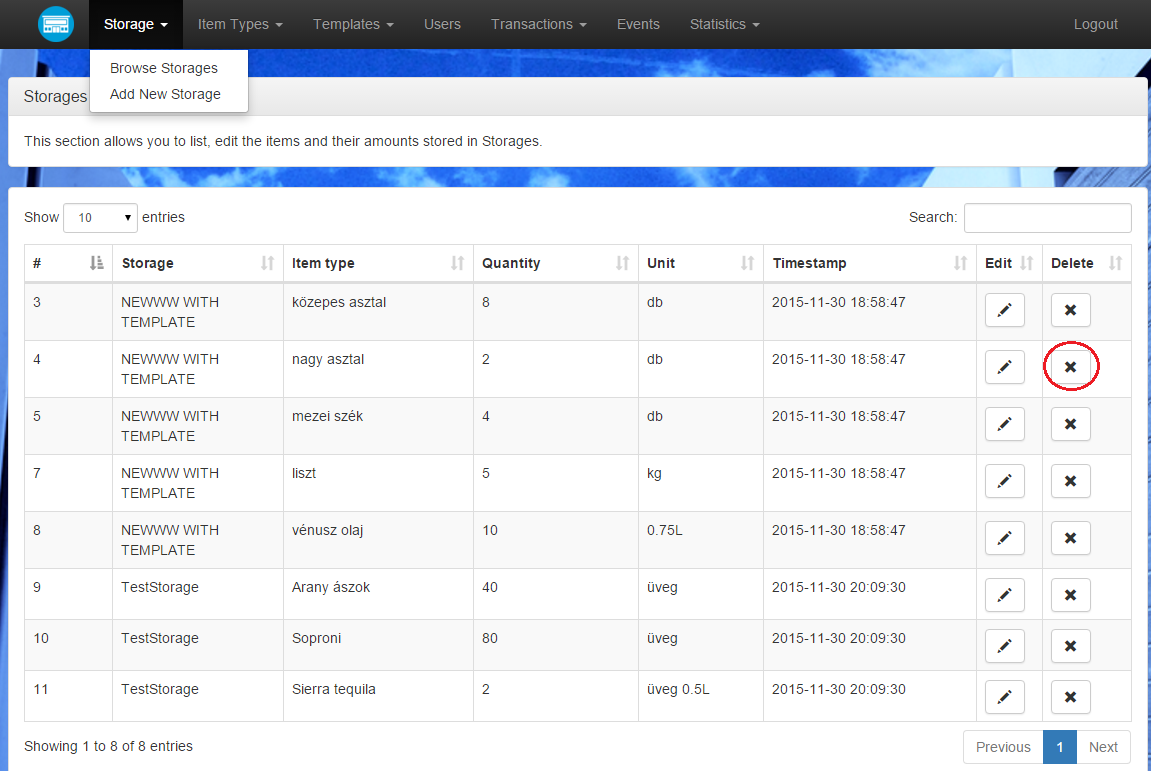
16. ábra – Új tétel hozzáadásakor felugró ablak

Tételek mennyiségének módosítása: a felhasználó a kikeresett tétel mennyiségét a szerkesztés gombbal módosíthatja. A gombra kattintás után a felugró ablakban beállítható az új mennyiség, majd a „*Save Changes*” gombbal elmenthető.



17. ábra – Elemek szerkesztése

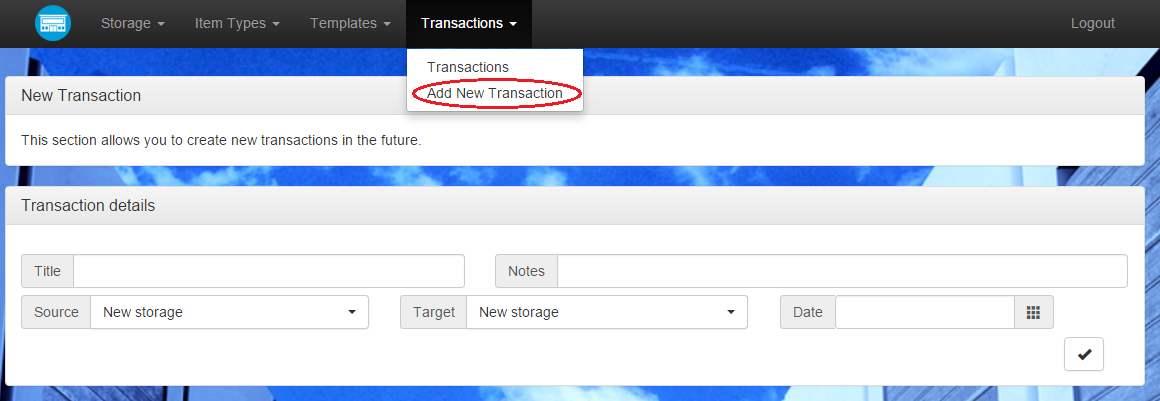
Tételek törlése: a felhasználó kiválasztja a listából az általa törölni kívánt tételt, és a sor szélén található törlés gombbal eltávolítja.



18. ábra – Tételek törlése

Be- és kiszállítások adatai (melyik raktárba, melyik raktárból, mit, mennyit): a felhasználó kiválasztja a *Transaction* menüből a *Transactions* menüpontot és kikeresi a megjelenő listából a számára lényeges elemeket

Be- és kiszállítások felvétele: a felhasználó a *Transaction->Create New Transaction* lehetőséget választja, majd kitölti az űrlapot a megfelelő adatokkal (Source: Küldő raktár, Target: Fogadó raktár, Date: dátum, dátumválasztóval)



19. ábra – Új szállítmány felvétele

Be- és kiszállítások törlése: a felhasználó a *Transaction* menüpontot választja, majd a megjelenő listában kikeresi a törlendő bejegyzést. Végül a sor végén található törlés gombbal eltávolítja azt

### Menedzseri funkciók

*A következő funkcióknál feltételezzük, hogy a felhasználó bejelentkezett és menedzser (manager) szintű jogosultságai vannak. Továbbá a korábban ismertetett funkciókat csak megemlítjük, további részletekért lásd hivatkozások*

Összes raktár lekérdezése: *lásd Alkalmazotti funkciók*

Adott raktárhoz tartozó információk lekérése: *lásd Alkalmazotti funkciók*

Be- és kiszállítások adatai: *lásd Alkalmazotti funkciók*

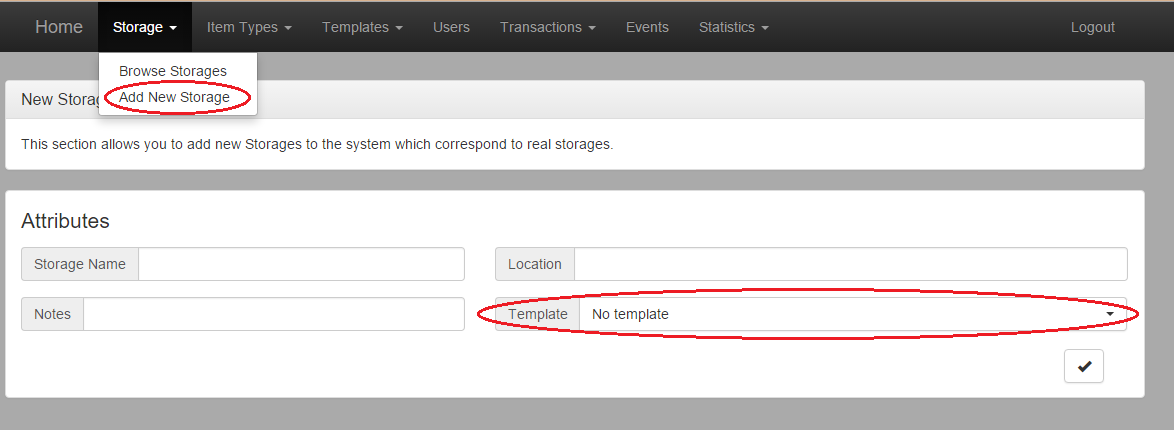
### Tulajdonosi funkciók

*A következő funkcióknál feltételezzük, hogy a felhasználó bejelentkezett és tulajdonos (owner) szintű jogosultságai vannak. Továbbá a korábban ismertetett funkciókat csak megemlítjük, további részletekért lásd hivatkozások*

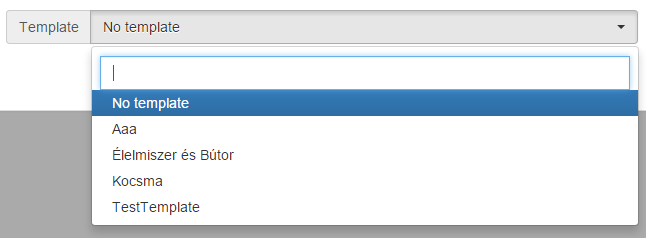
Összes raktár lekérdezése: *lásd Alkalmazotti funkciók*

Adott raktárhoz tartozó információk lekérése: *lásd Alkalmazotti funkciók*

Új raktár felvétele a rendszerbe: a felhasználó kiválasztja a *Storage* menüpont alatt található *Add new storage* lehetőséget, majd kitölti az űrlapot a raktár információival. A felhasználható minták közül egy legördülő listából választhat, vagy „*No template*”-re állítva nem lesz minta csatolva. Végül a „pipa” gombbal elmenti a rendszerben. Ha volt minta csatolva, akkor az alapértelmezett tételeket a rendszer hozzáfűzi a raktárhoz.

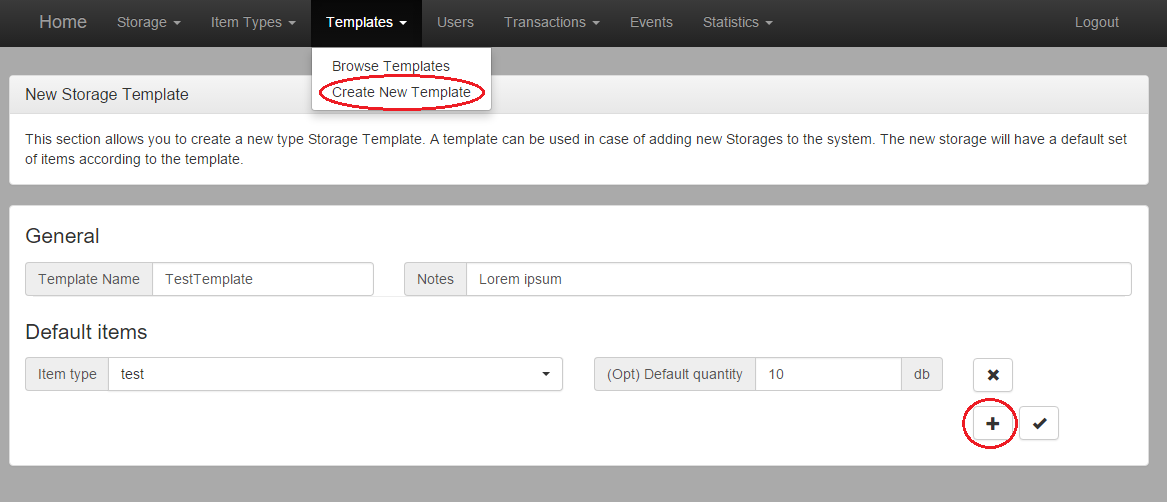


20. ábra – Új raktár hozzáadása



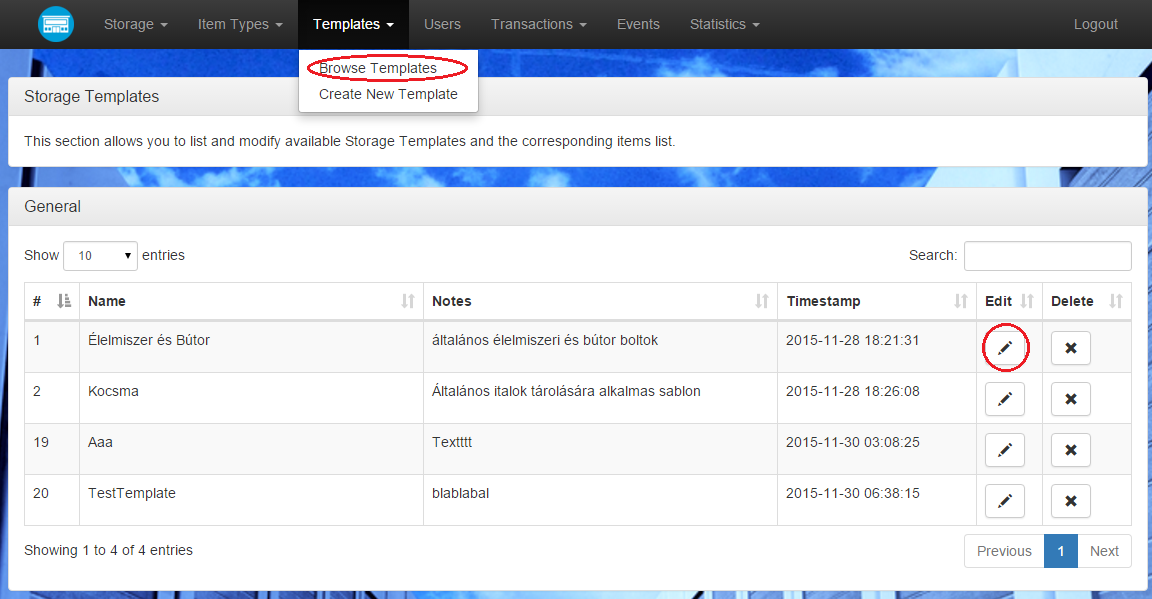
21. ábra – Mintaraktár beállítása

Sablonok létrehozása: felhasználó kiválasztja a *Template* legördülő menüből a *Create New Template* menüpontot, majd kitölti az űrlapot, végül hozzáadja a megfelelő tételeket a listából a „plusz” jel segítségével, opcionálisan még a hozzáadott tétel alapértelmezett mennyiségét is megadhatja. A sablon mentéséhez a „pipa” gombot kell megnyomni.



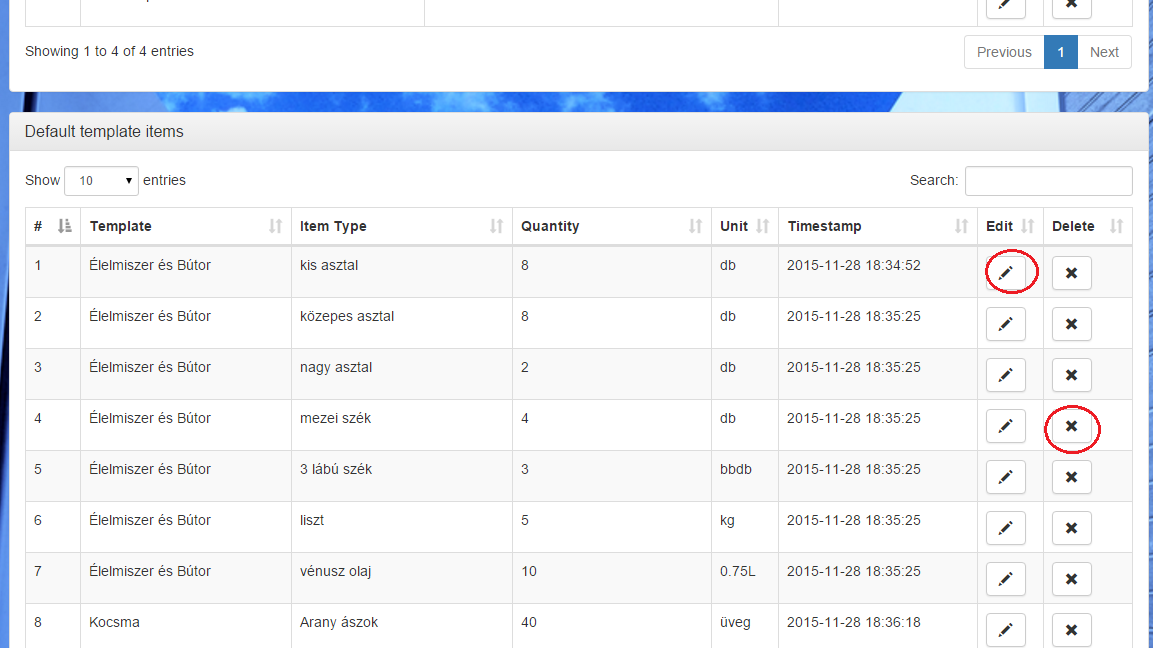
22. ábra – Új sablon létrehozása

Sablonok módosítása: felhasználó kiválasztja a *Template* legördülő menüből a *Browse Templates* menüpontot, majd kikeresi a módosítani kívánt mintát kézzel, vagy a keresőmezővel. Ezután a mintához tartozó szerkesztés gombra kattint, módosítja a felugró ablakban a megfelelő mezőket, végül a *Save Changes* gombbal elmenti a változásokat.



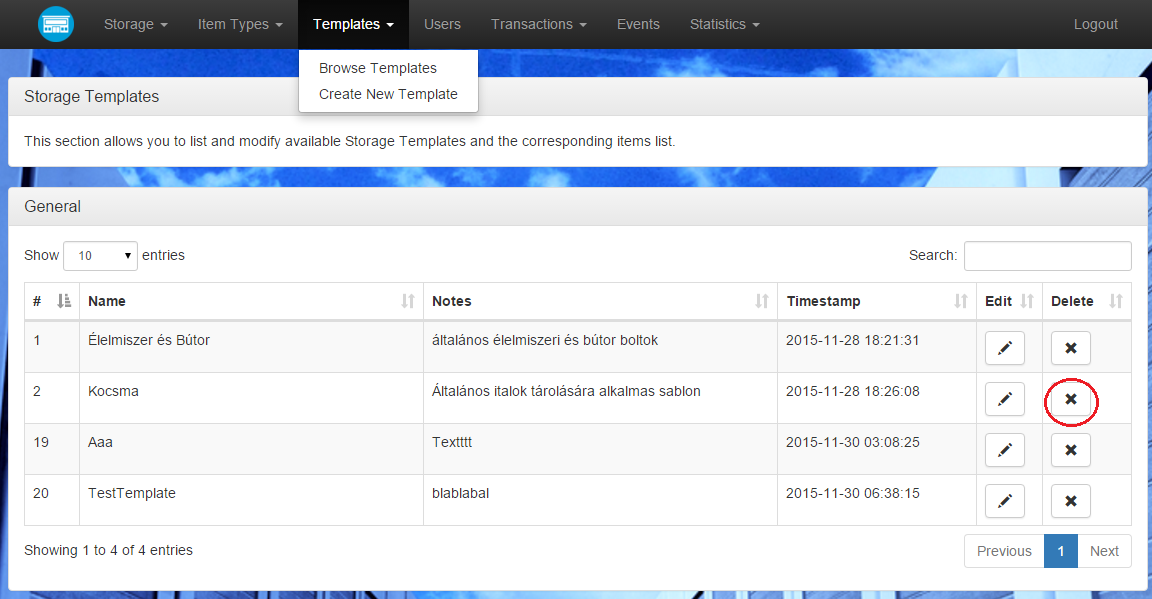
23. ábra – Sablonok szerkesztése

A sablon alapértelmezett tételeit szintén ezen menüpontban, a General beállítások alatt módosíthatjuk. A Szerkesztés gombbal az alapértelmezett mennyiséget módosíthatjuk, a Törlés gombbal pedig a tételtípust törölhetjük, akkor, ha azt nem használja egy konkrét raktár se.



24. ábra – Tétel típusok szerkesztése a sablonokban

Sablonok törlése: felhasználó kiválasztja a *Template* legördülő menüből a *Browse Templates* menüpontot, majd kikeresi a módosítani kívánt mintát kézzel, vagy a keresőmezővel. Ezután a mintához tartozó törlés gombbal törölheti a mintát.



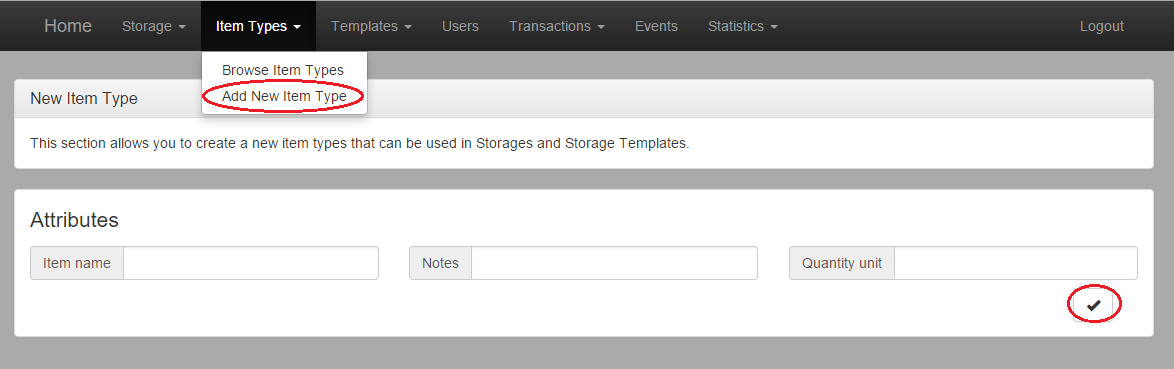
25. ábra – Sablonok törlése

Új tételek felvétele: *lásd Alkalmazotti funkciók*

Tételek törlése: *lásd Alkalmazotti funkciók*

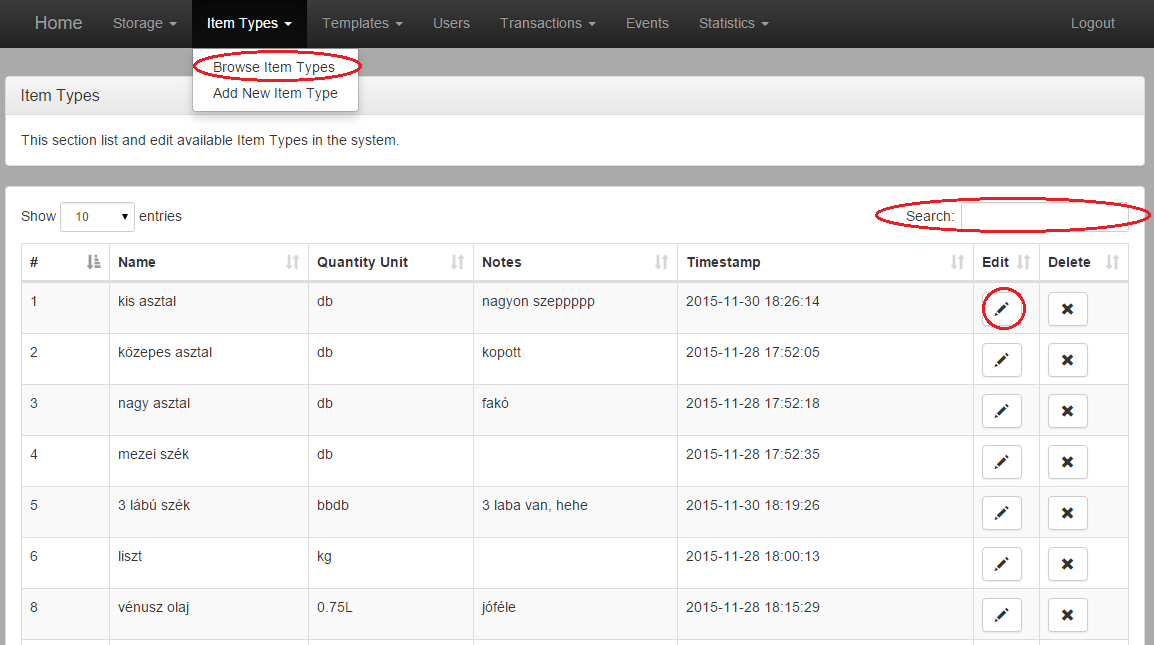
Tételek mennyiségének módosítása: *lásd Alkalmazotti funkciók*

Tételtípus felvétele: a felhasználó az *Item Types* menüpontjai közül kiválasztja az *Add new Item Type* lehetőséget, kitölti a megjelenő űrlapot a megfelelő adatokkal, majd a „pipa” gombra kattintva elmenti.



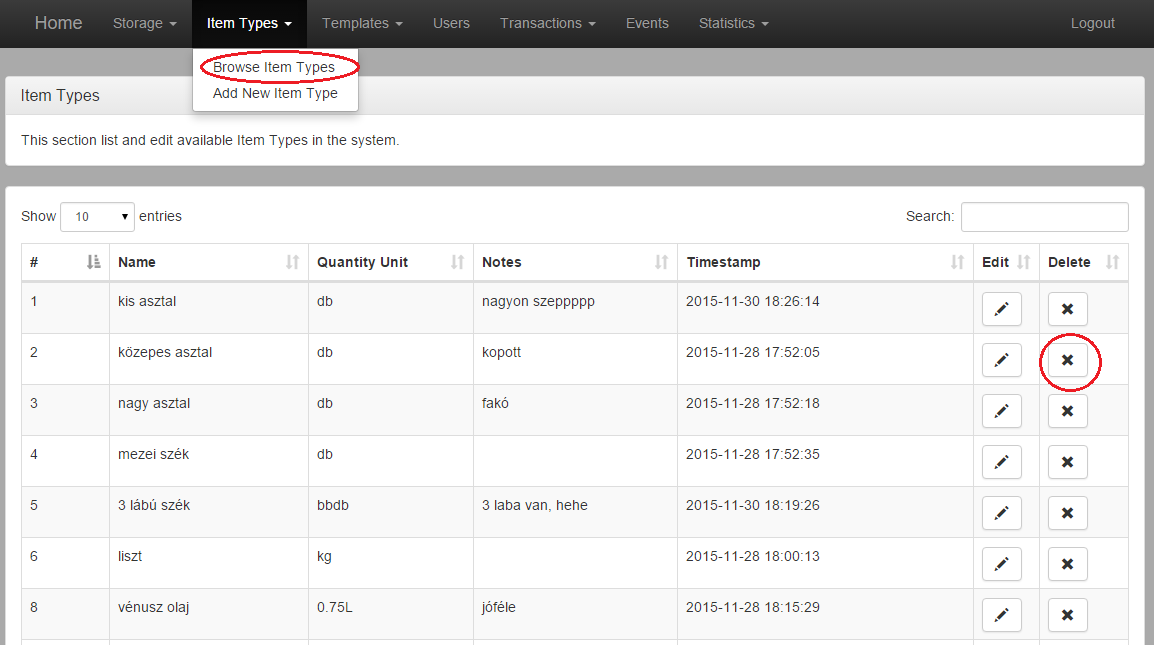
26. ábra – Új tételtípus hozzáadása

Tételtípusok módosítása: a felhasználó az *Item Types* pontjai közül a *Browse Item Types* menüt választja. Megkeresi a módosítani kívánt típust vagy a kereső mezővel, vagy kézzel, majd a tétel sávjában lévő szerkesztés gombra kattint. A felugró ablakban módosítja a megfelelő mezőket, majd a „*Save Changes*” gombbal elmentheti.



27. ábra – tételtípus módosítása

Tételtípus törlése: a felhasználó kiválasztja az *Item Types* alatt megjelenő *Browse Item Types* menüpontot, majd kikeresi az előbbi funkcióban leírtaknak megfelelően a törölni kívánt tételtípust. Ezután a típus jobb szélén látható „X” törlő gombbal eltávolíthatja azt.



28. ábra – Tételtípus törlése

Be- és kiszállítások adatai: *lásd Alkalmazott funkciók*

Be- és kiszállítások felvétele: *lásd Alkalmazott funkciók*

Be- és kiszállítások törlése: *lásd Alkalmazott funkciók*

# Telepítési útmutató

### Üzemeltetői

Előfeltételek:

* egy működő szerver, szervergép, -park, (főként apache szerveren tesztelve)
* működőképes MySQL szerver

Ha az előfeltételek teljesültek, akkor:

* az *apache szerver* adminisztrátori felületén deployolja az alkalmazást (pontos utasítások a Tomcat dokumentációjában[[4]](#endnote-4)),
* hozzon létre egy új adatbázist a *MySQL* szerveren „*StorageManager*” néven[[5]](#endnote-5),
* importálja az adatbázisba az alkalmazás db mappájában található *StorageManager.sql* –t,
* vegyen fel új felhasználókat, s töltse fel az adatbázist a szükséges adatokkal

### Felhasználói

Előfeltételek:

* egy már működő számítógép vagy mobiltelefon, operációs rendszerrel, a felhasználó számára hozzáférhető felhasználói fiókkal,
* internet hozzáférés,
* weblapok megtekintésére alkalmas böngésző
* az üzemeltető publikálta az alkalmazást
* az üzemeltető létrehozott az alkalmazásban egy felhasználót

Ha az előfeltételek teljesültek, akkor:

* Nyisson meg egy új ablakot a böngészőben
* Navigáljon az üzemeltető által megadott honlapra
* Jelentkezzen be az üzemeltetőtől kapott felhasználónévvel és jelszóval
* használja a rendszert.

# Továbbfejlesztési lehetőségek

A programot több szempont szerint megvizsgáltuk, ezen szempontok szerint a következő lehetőségeket találtuk.

*Biztonsággal kapcsolatos fejlesztés:*

Az authentikációhoz tartozó jelszó tárolását egyszerű hasheléssel oldottuk meg („salt” hozzáadásával). Ez egy alapszintű titkosítás, az egyik lehetőség ennek a titkosításnak a lecserélése egy bonyolultabbra.

*Felhasználói felület fejlesztései:*

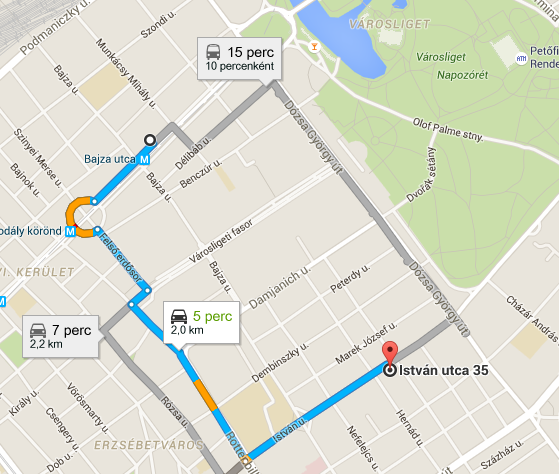
Értesítési események konfigurálása: bizonyos eseményeknél a rendszer értesítéseket (felugró ablak, email, sms) küldhet a felhasználó részére, ha az igényli

Statisztikák lekérdezése: az egyes raktárakhoz, vagy a raktárak összességéhez tartozó statisztikák lekérdezése (pl. kihasználtság, szállítások min./max. száma naponta, hetente)

Továbbá jelenleg a felhasználók hozzáadása és eltávolítása nincs kivezetve a felületre, így azt csak az adatbázis adminisztrátora képes kezelni, ami nagyon biztonságos, de nem túl kényelmes.

A raktárak helyszíneit lehetséges lenne térképen is megjeleníteni (pl. Google Maps), ezzel segítve a felhasználót a raktárak helyzeteinek áttekintését.

A raktárak közötti szállítási idő nem szerepel az adatbázisban, s nem is számítható jelenleg. A Google Maps útvonaltervező szolgáltatását integrálva, viszont ez lehetségessé válik, bár a kötelező pihenő időt nem adja a szolgáltatás hozzá, ezzel ki kell egészíteni a pontosabb számításokhoz.



29. ábra

Ezzel a fejlesztéssel tervezhetőbb lenne a kamionok lepakolása, illetve a raktár telítettségének korlátozása.

A raktárak telítettségét a felhasználói felületen egy dedikált ábra jelezhetné dinamikusan. Az ábra egy lehetséges megvalósítása:

* 0-40% 
* 40-80% 
* 80-100% 

# Verziókövetés

|  |  |  |  |
| --- | --- | --- | --- |
| Szerkesztő neve | Szerkesztés ideje | Módosítások | Új verziószám |
| Kiss Rebeka | 2015. 10. 01. | Dokumentum létrehozása | v0.1 |
| Kiss Rebeka | 2015. 10. 27. | Összefoglaló hozzáadása | v1.0 |
| Kamrás Márton | 2015. 10. 28. | Előzetes architektúra tervek | v1.1 |
| Kiss Rebeka | 2015. 11. 03. | Előzetes adatbázis tervek | v1.2 |
| Kiss Rebeka | 2015. 11. 04. | Funkciók leírása | v1.3 |
| Kamrás Márton | 2015. 11. 05. | Telepítési útmutató kitöltése | v1.4 |
| Kamrás Márton | 2015. 11. 10. | Felhasználói felület leírása | v1.5 |
| Kiss Rebeka | 2015. 11. 10. | Dokumentáció hiányosságainak pótlása | v1.6 |
| Kiss Rebeka | 2015. 11. 28. | Dokumentum formázás, véglegesítés | v2.0 |

# Irodalomjegyzék

1. <https://en.wikipedia.org/wiki/HTTPS> [↑](#endnote-ref-1)
2. <https://en.wikipedia.org/wiki/SQL_injection> [↑](#endnote-ref-2)
3. <http://php.net/manual/en/book.pdo.php> [↑](#endnote-ref-3)
4. <http://tomcat.apache.org/tomcat-7.0-doc/> [↑](#endnote-ref-4)
5. <http://dev.mysql.com/doc/refman/5.7/en/creating-database.html> [↑](#endnote-ref-5)