Szoftverarchitektúrák

Raktárkezelő rendszer

*Készítette: Kamrás Márton (NEPTUN)*

*Kiss Rebeka Enikő (EVLIR3)*

Tartalom

[Összefoglaló 3](#_Toc436412866)

[Rendszerterv 3](#_Toc436412867)

[Architektúra 3](#_Toc436412868)

[Adatbázis terv 3](#_Toc436412869)

[Biztonság 3](#_Toc436412870)

[Felhasználói felület 3](#_Toc436412871)

[Funkciók 3](#_Toc436412872)

[Telepítési útmutató 3](#_Toc436412873)

[Üzemeltetői 3](#_Toc436412874)

[Felhasználói 3](#_Toc436412875)

[Verziókövetés 3](#_Toc436412876)

# 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

A program a *Model-View-Controller* architektúrát követi, kiegészítve az adatbázis réteggel, melyet a *Wrapper* mintával kapcsoltunk az alkalmazáshoz. A *Wrapper* mintával az SQL utasításokat akartuk elfedni, így elkerülendő az esetleges elrontott vagy épp rosszindulatú parancsok végrehajtását.

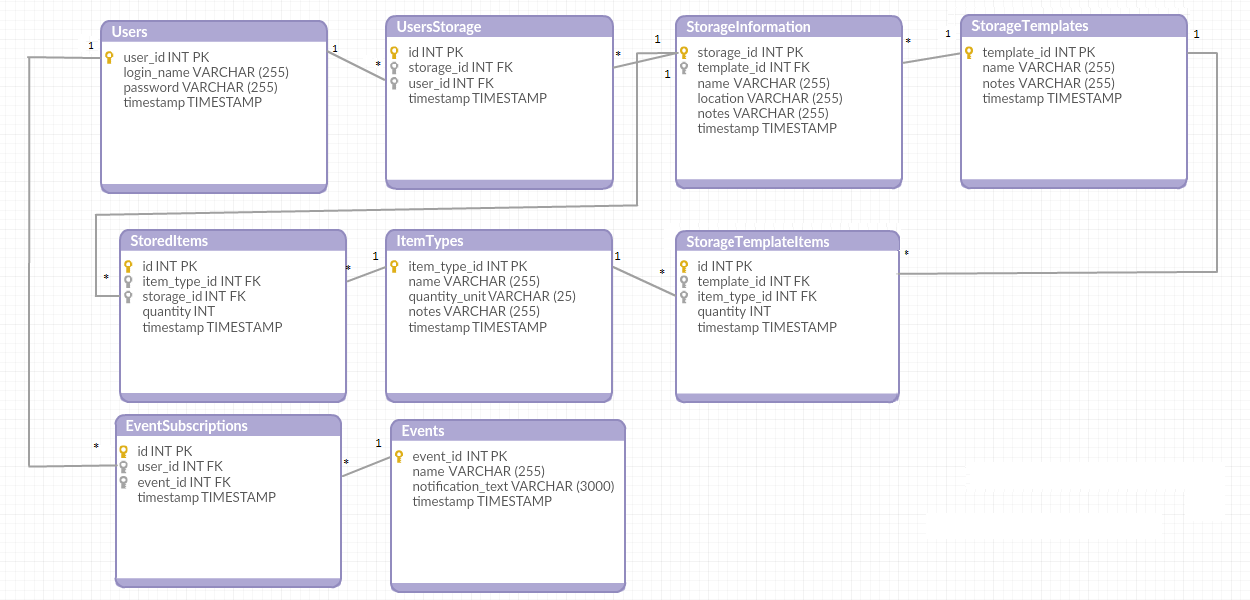
A *Model* réteg kommunikál a MySQL szerverrel, ahol az adatbázis tárolódik. A *Model* réteg osztályai felelősek az adatok tárolásáért.

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.

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.

## Adatbázis terv



. 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* tartalmaz egy raktárkészlet leírást.

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, míg a *StoredItems* a konkrét készleteket raktáranként.

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.

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

## Biztonság

## Felhasználói felület

## Funkciók

# Telepítési útmutató

## Üzemeltetői

## Felhasználói

# 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 |
|  |  |  |  |
|  |  |  |  |