

# Követelmény specifikáció

---

## *Raktárkezelő rendszerhez*

### Vezetői összefoglaló

A program egy nagyobb vállalat raktárainak összefogását, a raktárak tartalmának követését hivatott segíteni, webes felületű megjelenítéssel, amely mobil eszközökön is olvasható, áttekinthető felületet biztosít a felhasználóknak.

### Szoftver rendeltetése

1. Raktárak általános adminisztrálása
  - Különböző raktártípusok kezelése, ezekhez sablonok biztosítása
  - Raktár típusonkénti tételek meghatározása
  - Raktárak létrehozása, tételek hozzáadása, módosítása
  - Raktárankénti beszerzések támogatása sablonokkal
  - Raktárankénti tételfogyás rögzítése periódikusan (felhasználó végzi, de értesítést küld a rendszer, ha elmarad pl. konfigurálhatóan bizonyos eseményeknél automatikus email küldés)
2. Raktárankénti adminisztráció
  - Beszállítások kezelése (alapvetően sablon alapján, de a módosításhoz külön felületet ad a program)
  - A kimenő tételek követése, rögzítése
  - Raktárak közti tranzakció könnyített elérése (előző két pontot összefogva)
3. Monitorozás
  - Raktárak aktuális állapota
  - Generálható statisztikák (pl. fennmaradó szabad helyek, összesített nézet az egyes tételekből)

### Szerepkörök

1. *Alkalmazott* – általános felhasználó, a program leggyakoribb szereplője. Raktárak készleteit frissíti, figyeli a készletmozgásokat, jelentéseket készít. Csak a tételeket módosíthatja, a raktárat nem. Beszerzéseket kezdeményezhet, raktárak közti szállításokat is módosíthatja.
2. *Tulajdonos* – az adatbázis adminisztrátora, tulajdonosa, hozzáadhat vagy törölhet raktárat, konfigurálhatja az automatikus email küldést, sablonokat hozhat létre. Az egyes raktárak készleteit általában nem figyeli, de az ellenőrizhetőség miatt látja. Az általános felhasználói jogokkal is rendelkezik.
3. *Menedzser* – email értesítéseket kap az általa kért eseményekről, a generálható statisztikákat figyeli. Csak lekérdező jogosultságai vannak.

## Use-case-k

### Általános Use-case-k

- Bejelentkezés – belép a felhasználó a rendszerbe
- Kijelentkezés – kilép a felhasználó a rendszerből

### Alkalmazotti Use-case-k

- Összes raktár lekérdezése – megjelennek a felhasználó számára elérhető raktárak
- Adott raktárhoz tartozó információk lekérése – megjelennek a kiválasztott raktár részletes adatai
- Új tételek felvétele – a felhasználó új termékeket visz be a rendszerbe
- Tételek törlése – a felhasználó töröl már meglévő termékeket
- Tételek mennyiségének módosítása – a termékek raktáron lévő mennyiségének módosítása (csökkentés, növelés)
- Be- és kiszállítások adatai - melyik raktárba, melyik raktárból, mit, mennyit
- Be- és kiszállítások felvétele – új szállítmány bevitele a rendszerbe
- Be- és kiszállítások törlése – már meglévő szállítmány törlése a rendszerből

### Menedzseri Use-case-k

- Összes raktár lekérdezése - megjelennek a felhasználó számára elérhető raktárak
- Adott raktárhoz tartozó információk lekérése – megjelennek a kiválasztott raktár részletes adatai
- Be- és kiszállítások adatai - melyik raktárba, melyik raktárból, mit, mennyit

### Tulajdonosi Use-case-k

- Összes raktár lekérdezése - megjelennek a felhasználó számára elérhető raktárak
- Adott raktárhoz tartozó információk lekérése – megjelennek a kiválasztott raktár részletes adatai
- Új raktár felvétele a rendszerbe – új raktár bevitele a rendszerbe a megfelelő információk segítségével
- Raktár törlése – már meglévő raktár eltávolítása a rendszerből
- Raktár adatainak módosítása
- Sablonok létrehozása – az egyéni raktárakhoz tartozó új sablonok bevitele a rendszerbe
- Sablonok módosítása – már meglévő sablonok módosítása
- Sablonok törlése – már meglévő sablonok eltávolítása a rendszerből
- Be- és kiszállítások adatai - melyik raktárba, melyik raktárból, mit, mennyit
- Be- és kiszállítások felvétele – új szállítmány felvétele
- Be- és kiszállítások törlése – már meglévő szállítmány törlése
- Tételtípusok hozzáadása – új tételtípusok létrehozása a rendszerben
- Tételtípusok módosítása – már meglévő tételtípusok módosítása
- Tételtípusok törlése – már meglévő tételtípusok törlése a rendszerből

## Adatstruktúra

**Tétel típus:** A raktárak legkisebb logikai egysége. A tétel nevéből és mértékegységéből áll, valamint megjegyzéseket tartalmazhat.

**Szállítás:** Több tétel típusból álló egység, mely a raktárba hozott vagy a raktárból eltávolított tételeket foglalja magában. A raktár egyes tételeinek mennyiségét befolyásolja

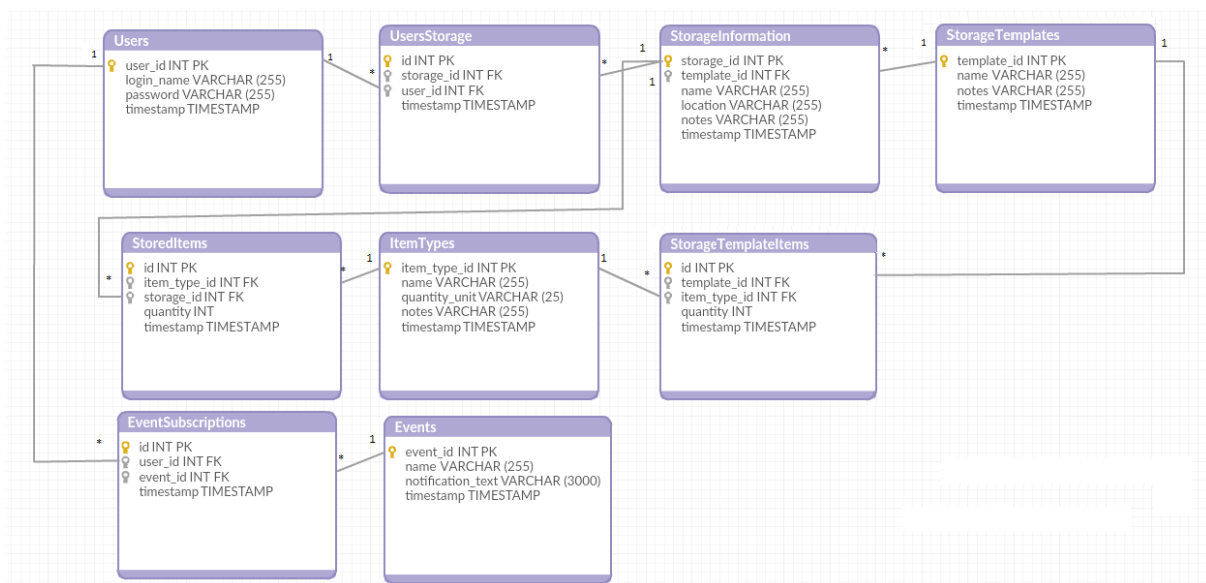
**Raktár:** A tételeket (tétel típus és mennyiség párok) és szállításokat tartalmazó egység, valamilyen raktár típussal ellátva

**Raktár típus:** A raktárba felvehető lehetséges tételeket definiálja

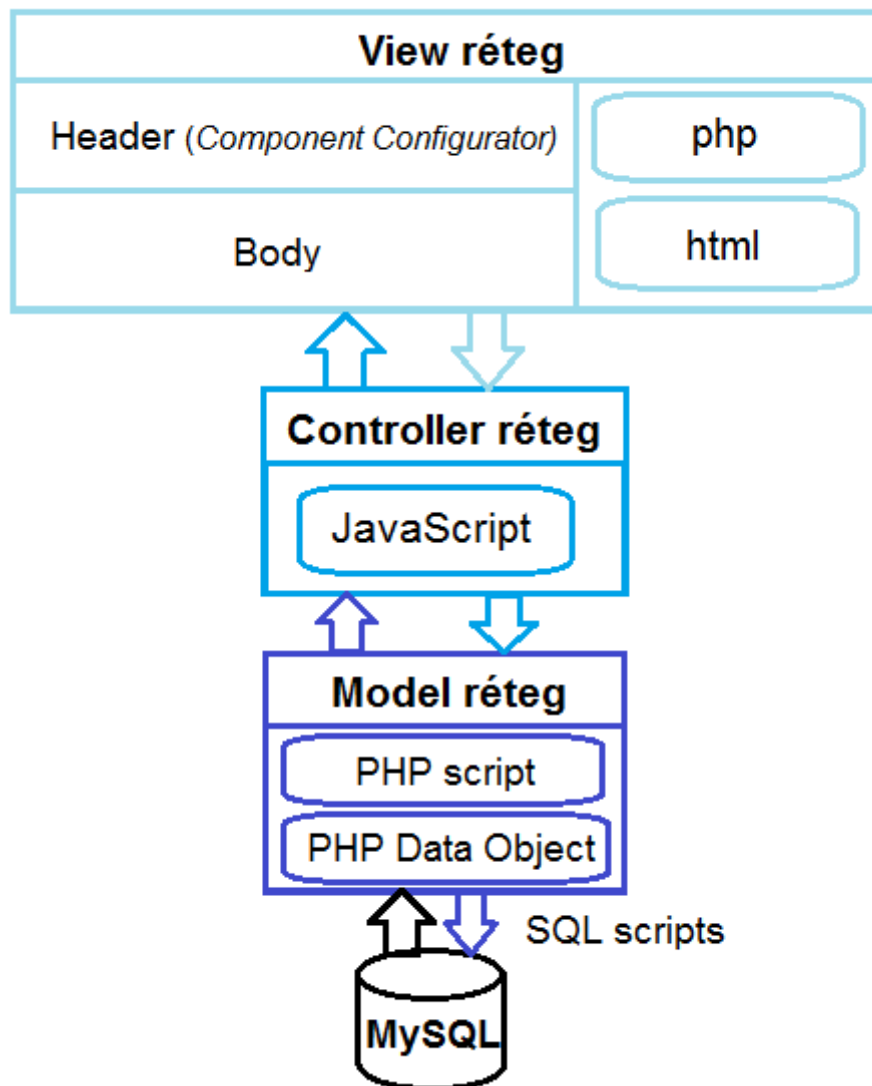
**Esemény:** A raktár állapotában beálló változások, illetve a periodikusan ismétlődő változások (pl. napok) logikai entitása.

**Felhasználó:** Egy valós személy programbeli megtestesítője, mely valamilyen felhasználói típussal rendelkezik

**Felhasználói típus:** Jogosultságok logikai egysége



### Architekturális tervek:



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 elkerülhetők. 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 *Controller*nek 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.

Az adatbázis 4 fő részre bontható:

A felhasználókra (Users tábla). Ebben a táblában mentődnek el a felhasználó belépési adatai, felhasználó név és jelszó.

A raktárak (UsersStorage, StorageInformation, StorageTemplateItems, StorageTemplates táblák). Ezekben a táblákban találhatóak meg a raktárak leírásai (név, hely) (StorageInformation), a felhasználók által használt raktárak (UsersStorage), illetve az új raktárak létrehozásához szükséges raktárminták (StorageTemplates). A StorageTemplateItems tartalmazza a raktárkészlet leírását.

A tárolandó termékek/készletek (StoredItems, ItemTypes). Az ItemTypes a termékek leírását adja meg (név, mértékegység), míg a StoredItems a konkrét készleteket írja le (miből, hol, mennyi).

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 EventSubscriptions tábla tartalmazza.

## Technológiai követelmények

A programnak elérhetőnek kell lennie egy böngésző segítségével, amely böngészőt indíthatják asztali/hordozható számítógépes, illetve mobil környezetből is. Mindkét esetben könnyen áttekinthetőnek kell lennie.

A megvalósítás MySQL adatbázissal, apache szerveren fog történni, PHP illetve JavaScript kódokkal. A biztonságos elérés miatt a szerver https protokollal lehet elérni.