Online állatkölcsönző

Rendszerleírás

Szoftverarchitektúrák (VIAUMA06) házi feladat

Készítette:

Pollák Tamás (BABQ9E) és Kincses Gergő (A7SOEA)

2015. november. 24

Tartalom

[A rendszer célja, funkciói és környezete 3](#_Toc436170591)

[Feladatkiírás 3](#_Toc436170592)

[Az alkalmazás által biztosított funkciók 3](#_Toc436170593)

[Állat attribútumai 3](#_Toc436170594)

[Az alkalmazás felülete 4](#_Toc436170595)

[Ügyfél funkciók 4](#_Toc436170596)

[Adminisztrátor funkciók 5](#_Toc436170597)

[Az alkalmazás környezete 6](#_Toc436170598)

[Megvalósítás és architektúra 6](#_Toc436170599)

[Adatbázis 7](#_Toc436170600)

[Metódusok: 7](#_Toc436170601)

[Adat- és adatbázisterv 7](#_Toc436170602)

[Backend 9](#_Toc436170603)

[Metódusok: 9](#_Toc436170604)

[Frontend 9](#_Toc436170605)

[A GUI főbb elemei: 9](#_Toc436170606)

[Az oldal működése 10](#_Toc436170607)

[Regisztráció és bejelentkezés 10](#_Toc436170608)

[Menedzsment 11](#_Toc436170609)

[Böngészés és kölcsönzés 12](#_Toc436170610)

[Telepítési leírás 12](#_Toc436170611)

[Hivatkozások 12](#_Toc436170612)

# A rendszer célja, funkciói és környezete

## Feladatkiírás

Ki ne akart volna már néhány napra szórakozásból hazavinni egy édes kis nyuszikát, hörcsögöt, vagy éppen tyúkot, tehenet? Most valóra válthatják az álmukat! A feladat egy webes rendszer elkészítése, mely lehetőséget kínál állatok regisztrálására és követi a foglalásaikat, kölcsönzésüket.

## Az alkalmazás által biztosított funkciók

A feladat megoldása során a cél egy olyan webes alkalmazás létrehozása, melyen keresztül lehetőség nyílik különböző állatok kikölcsönzésére. Az alkalmazás biztosítja az állatok közti böngészés lehetőségét, ill. a kikölcsönzés utáni visszaadásukat. Az állatok adatait egy adminisztrátor tölti fel az adatbázisba, és ő is tartja karban őket. Egy állatot tetszőleges időtartamra ki lehet kölcsönözni.

## Állat attribútumai

Egy állatot a következő tulajdonságai jellemeznek:

* kép – egy kis kép az állatról
* név – az állat neve
* faj – az állat faja
* fajta – az állat fajtája
* nem – az állat neme
* kor – az állat életkora
* szín – az állat színe
* térfogat – az állat térbeli kiterjedése
* súly – az állat súlya
* kedvenc étel – az állat kedvenc étele
* hobbi – ezt szereti az állat általában csinálni
* elérhető-e – az állat jelenleg ki van-e kölcsönözve
* törött-e – az állat egészségi vagy mentális állapota megengedi-e, hogy kikölcsönözzék

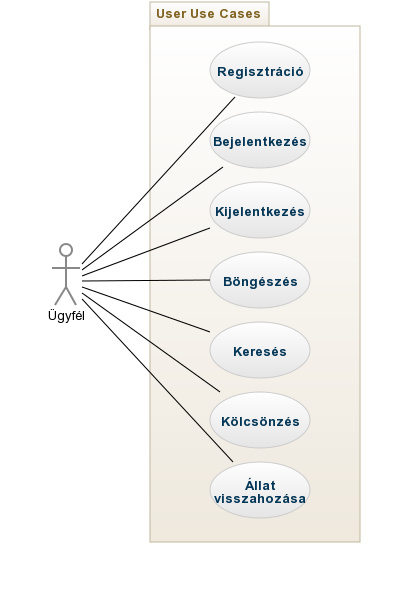
## Az alkalmazás felülete

Az alkalmazásnak alapvetően három féle felülete van:

* Bejelentkező felület: Itt tudnak az ügyfelek és az adminisztrátorok bejelentkezni az alkalmazásba, valamint itt tudnak az ügyfelek regisztrálni. A rendszerből történő kijelentkezés után ide kerülnek a felhasználók.
* Lista felület: Itt tudnak az ügyfelek böngészni és keresni az állatok között. A felületen egymás alatt jelennek meg az egyes állatok minden attribútumukkal.
* Menedzsment felület: Itt tud az adminisztrátor új állatot bevinni a rendszerbe, állatot eltávolítani a rendszerből, ill. módosítani a rendszerben lévő állatok egyes attribútumain.

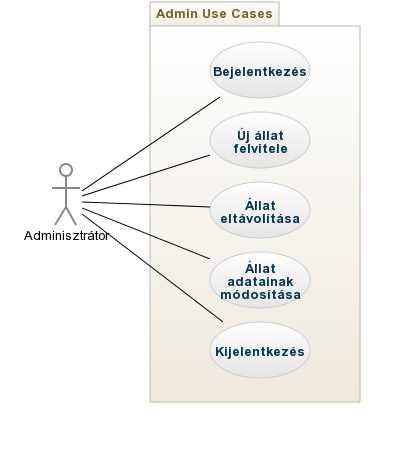
## Ügyfél funkciók

* Az ügyfél a bejelentkező felületen be tud regisztrálni az alkalmazásba.
* Az ügyfél a bejelentkező felületen be tud jelentkezni az alkalmazásba.
* A bejelentkezett ügyfél a lista felületen tud böngészni az állatok között.
* A bejelentkezett ügyfél a lista felületen tud keresni közöttük a fajuk, fajtájuk, elérhetőségük és épségük alapján.
* A bejelentkezett ügyfél a lista felületen ki tud kölcsönözni tetszőleges számú állatot.
* A bejelentkezett ügyfél a lista felületen vissza tud hozni tetszőleges számú kikölcsönzött állatot, amennyiben azok a felhasználónál vannak.
* A bejelentkezett ügyfél a lista felületen ki tud jelentkezni az alkalmazásból.



***1. ábra*** *Ügyfél use case-ek*

## Adminisztrátor funkciók

* Az adminisztrátor a bejelentkező felületen be tud jelentkezni az alkalmazásba, egy előre kialakított profilon keresztül.
* Az adminisztrátor a menedzsment felületen tud új állatot vinni az alkalmazásba.
* Az adminisztrátor a menedzsment felületen tud az alkalmazásból eltávolítani állatokat.
* Az adminisztrátor a menedzsment felületen tud módosítani az alkalmazásban lévő állatok attribútumainak értékein.

***2. ábra*** *Adminisztrátor use case-ek*

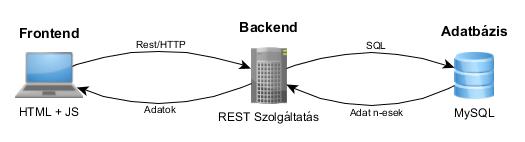
* Az adminisztrátor a menedzsment felületen tud kijelentkezni az alkalmazásból.

# Az alkalmazás környezete

Az alkalmazás egy kliensből és egy szerverből áll. A szervert Java nyelven készítjük el, a kliens HTML felhasználásával áll elő. A kliens REST interfészen éri el a szervert, és a szerver ugyanezen keresztül küld adatot a kliensnek. A szerver egy MySQL adatbázist használ az adatok tárolására. A program további magától nem értetődő követelményeket nem támaszt a futtató számítógép felé.

# Megvalósítás és architektúra

Az alkalmazást három komponens segítségével valósítottuk meg. Elkülönítettünk egy klienst, egy szervert és egy adatbázist. A komponensek és azok kommunikációja a következő ábrán látható.



***3. ábra*** *Architektúra*

## Adatbázis

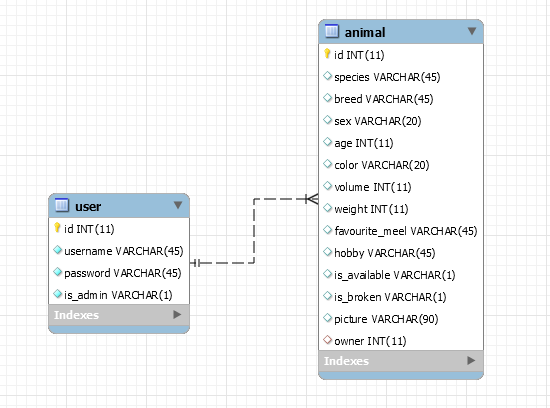
Az adatbázisban tároljuk a weboldalon kikölcsönözhető állatok adatait, a regisztrált felhasználók adatait és az adminisztrátorok adatait. Az adatbázist MySQL segítségével valósítottuk meg, egy ún. Object-Relational Mapping (ORM) [1] eljárás alkalmazásával. A módszer segítségével a kódban lévő adatstruktúrákat (praktikusan osztályokat) egy relációs adatbázis-tábla elemeire (sorokra) lehet leképezni és fordítva. Az ORM-et a Java Persistence API (JPA) segítségével implementáltuk.

### Metódusok:

* insertUser – beilleszt egy felhasználót az adatbázisba
* login – visszaad egy felhasználót az adatbázisból a neve és a jelszava alapján
* checkUserName – ellenőrzi, hogy az adott felhasználónév létezik-e az adatbázisban
* listAnimals – visszaadja az adatbázisban lévő összes állatot
* searchAnimals – a listázott állatokat szűri bizonyos feltételek alapján
* rentAnimal – az adatbázisban egy állat is\_available mezőjét „Y”-ra állítja
* bringBackAnimal - az adatbázisban egy állat is\_available mezőjét „N”-re állítja
* insertAnimal – felvesz egy új állatot az adatbázisba
* updateAnimal – egy az adatbázisban már létező állat adatait módosítja
* deleteAnimal – eltávolít az adatbázisból egy állatot

### Adat- és adatbázisterv

Az adatbázisban két fő tábla van, melyek tartalmazzák a felhasználók (ill. adminisztrátorok) és az állatok adatait.



*4. ábra Adatbázis táblák*

#### User tábla

A User táblában tároljuk a felhasználók és az adminisztrátorok adatait. Egy User elem négy adatból tevődik össze, amik az azonosító (id), egy felhasználónév, egy jelszó és egy is\_admin mező, amely megmondja, hogy az adott felhasználó adminisztrátor-e vagy sem. A User entitás 1:N kapcsolatban áll az Animal entitással, ami azt jelenti, hogy egy User-hez tartozhat több állat, de egy állatnak csak egy kölcsönzője lehet.

#### Animal tábla

Az Animal táblában tároljuk az állatok adatait. Egy állat tizennégy attribútummal rendelkezik: azonosító, faj, fajta, nem, életkor, szín, térfogat, súly, kedvenc étel, hobbi, egy kép az állatról (pontosabban a kép neve, amire majd az oldal hivatkozik), egy idegenkulcs a kölcsönzőre, illetve két további mező melyek megmondják, hogy az állat elérhető-e (is\_available) és, hogy minden rendben van-e vele (is\_broken).

## Backend

A backendet egy Rest szolgáltatás valósítja meg, amely egy WildFly szerveren [2] fut. Ebben a komponensben kerültek implementálásra azok az eljárások, amelyeket a frontend kliens hív. A működése során az egyes metódusai JSON objektumokat küldenek és fogadnak, ill. használják az adatbázis komponensben definiált eljárásokat.

### Metódusok:

* getUser – ellenőrzi, hogy egy adott felhasználó benne van-e az adatbázisban
* regUser – regisztrál egy adott felhasználót, amennyiben még a név még szabad
* getAnimals – kilistázza az adatbázisban lévő állatokat
* rent – egy adott felhasználóhoz rendel egy adott állatot
* bringBackAnimal – egy állatot „elválaszt” a bérlőtől és „visszahelyez” a készletbe
* searchAnimal – szűkíti az állatok listáját bizonyos tulajdonságok alapján
* insertAnimal – beilleszt egy új állatot az adatbázisba
* deleteAnimal – töröl egy állatot az adatbázisból
* updateAnimal – módosítja egy már felvett állat adatait

## Frontend

Az alkalmazás böngészőben történő megjelenítéséhez HTML oldalakat, Javascriptet és CSS-t használunk. Az oldal designja a Bootstrap keretrendszeren alapul.

### A GUI főbb elemei:

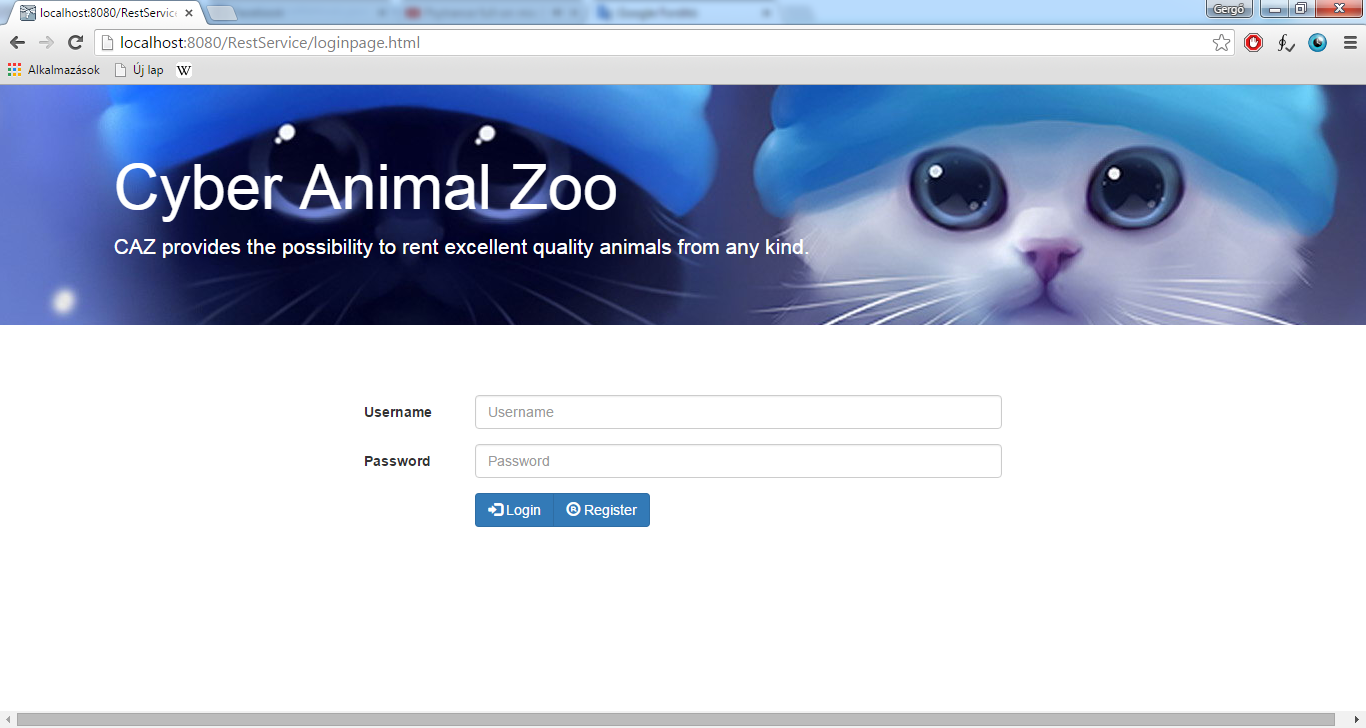
* Bejelentkező oldal – ezen keresztül lehet bejelentkezni, ill. regisztrálni
* Management oldal – az adminisztrátorok ezen keresztül menedzselhetik az alkalmazást
* Lista oldal – ez a fő felület, a felhasználók itt válogathatnak és kölcsönözhetnek az állatok közül

A fenti három GUI elemet (a designtól eltekintve) a következő egységek valósítják meg:

* Bejelentkező oldal – loginpage.html, loginpage.js
* Management oldal – managementpage.html, managementpage.js
* Lista oldal – listpage.html, listpage.js

# Az oldal működése

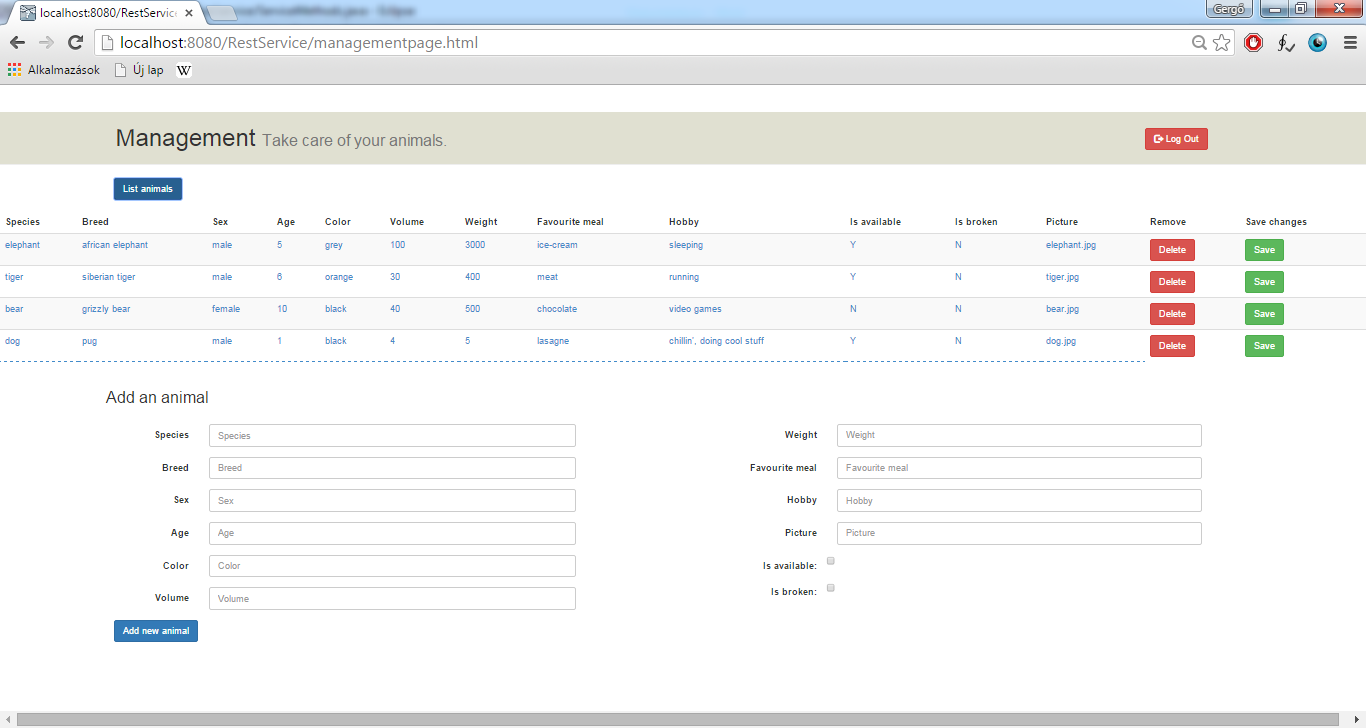
## Regisztráció és bejelentkezés



*5. ábra Bejelentkező oldal*

A felhasználók az ábrán látható felületen regisztrálhatnak a kölcsönző online felületére. Amennyiben már van regisztrációjuk, itt tudnak bejelentkezni. A kölcsönző adminisztrátorai is itt tudnak belépni a rendszerbe. Amennyiben hibás adatokkal próbál valaki belépni, a rendszer hibaüzenetet küld.

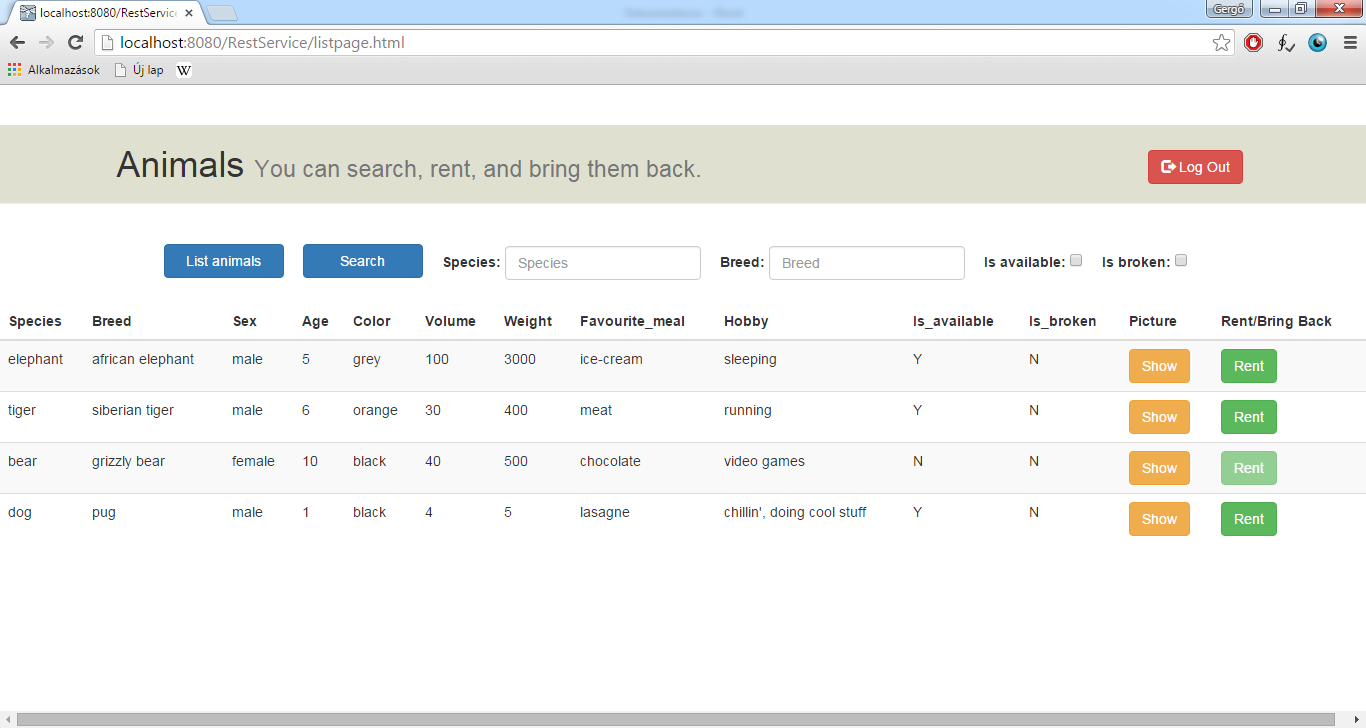
## Menedzsment



*6. ábra Menedzsment felület*

A kölcsönző adminisztrátorai ezen a felületen tudják a rendszerben lévő állatok adatait menedzselni, ill. új állatot felvinni a rendszerbe. A felső táblázat lehetőséget nyújt az egyes állatok adatainak mezőnkénti módosítására, az alul látható beviteli mezők pedig egy új kisállat bevitelére a kölcsönző adatbázisába.

## Böngészés és kölcsönzés



*7. ábra Állatok listája*

A felhasználó bejelentkezés után tud böngészni az állatok között egy lista nézetben. Ebben a listában tud szűrni specifikációban rögzített négy attribútum alapján. Az állatokról elérhetők a képeik, melyek a „Show” gomb megnyomása után jelennek meg. Ha megtalálta az igényeinek megfelelő állatot (vagy állatokat), akkor a ”Rent” gomb megnyomásával kikölcsönözheti. Ha a felhasználó vissza kívánja hozni az állatot, akkor ezt a „Rent” gomb helyén megjelenő „Bring back” gomb használatával teheti meg.

# Telepítési leírás

TODO

# Hivatkozások

[1] - <https://en.wikipedia.org/wiki/Object-relational_mapping>

[2] - <http://wildfly.org/>