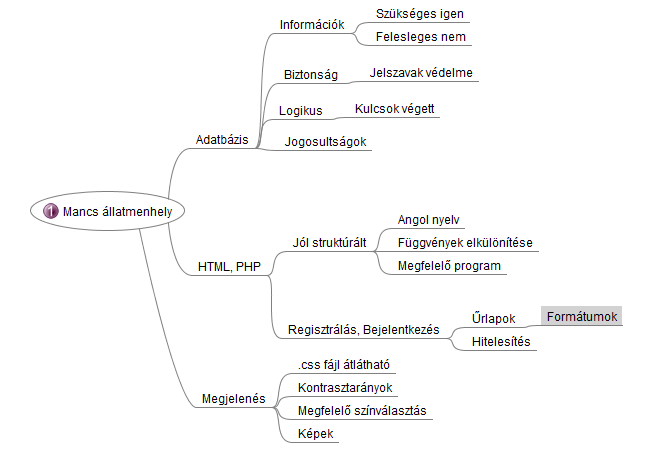
**Mancs-állatmenhely**

**Rendszerterv**

# Bevezetés

Mielőtt bármibe is belekezdenénk érdemes tartani egy ötletbörzét, és ezt grafikus módon is megvalósítani. Erre a mindmap az egyik leghatékonyabb módszer, így mi is ezt tettük, hogy a legfontosabb dolgok ki ne maradjanak.



# Architektúrális követelmények:

A feladat elvégzéséhez elég egy átlagos PC-re vagy egy jobb minőségű laptopra stabil internetkapcsolattal, és a megfelelő programokra (notepad++, netbeans stb.)

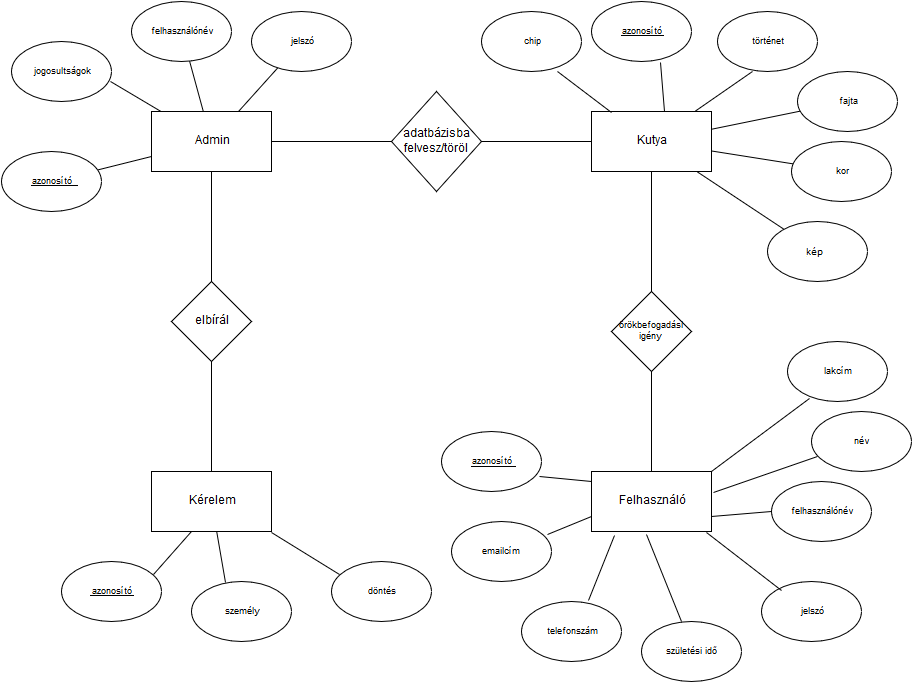
Szükségünk lesz egy adatbázisszerverre, ezt bérelhetjük, vagy üzemeltethetünk sajátot, ez esetben be kell szereznünk egy olyan gépet, amit szerverként üzemeltethetünk (magasabb specifikációkkal, és költségekkel). A mi esetünkben a szervergép az egyetemé, és ezt használjuk.

# Adatbázisterv:

Az **adatbázisokat** hozzuk létre először, a kódunk ugyanis ebből nyeri ki az adatokat.

Az adatbázisok szerkezetét a következő diagram szemlélteti:

## Egyed/kapcsolat diagram



Az egyed/kapcsolat modellen szemléltetjük az adatbázisunk struktúráját. Adatbázisunk implementálásához feltétlenül szükséges az egyed/kapcsolat diagram, a diagramot transzformáljuk a kívánt nyelvre így megkapjuk az adatbázist.

Jelen esetben 4 egyedünk van :

**1. Admin :** az állatmenhely adminisztrátora különleges jogokkal rendelkezik.

- elbírálhat kérelmeket : elutasíthat illetve elfogadhatja azokat

- új kutyákat vihet be az adatbázisba, vagy törölhet onnan

az adminisztrátor egyed attribútumai:

felhasználónév: az egyed felhasználóneve, ennek segítségével jelentkezik be

jelszó: az egyed jelszava, ennek segítségével jelentkezik be

azonosító: a tábla kulcsa, ez határozza meg egyértelműen az egyedet

jogosultságok: az egyed mindennemű jogosultságát tároljuk itt

**2. Felhasználó:** az állatmenhely áltanos felhasználói tartoznak ide

- örökbefogadási igényt adhat le választott kutyá(k)ra

egyed attribútumai:

azonosító: a tábla kulcsa, ez határozza meg egyértelműen az egyedet

az egyed többi attribútuma mind vagy regisztrációnál, vagy később az oldalon kitölthető.

**3. Kutya :** az állatmenhelyen lévő kutyák tartoznak ide

- fel lehet venni az adatbázisba/ki lehet törölni az admin által

- örökbefogadási igényt lehet rá leadni

azonosító: a tábla kulcsa, ez határozza meg egyértelműen az egyedet

a kutya egyed attribútumait az adminisztrátor állítja be, amikor felveszi az adatbázisba.

**4. Kérelem:** a kérvényekről tartalmaz információt

- kérelmet ellehet bírálni admin által

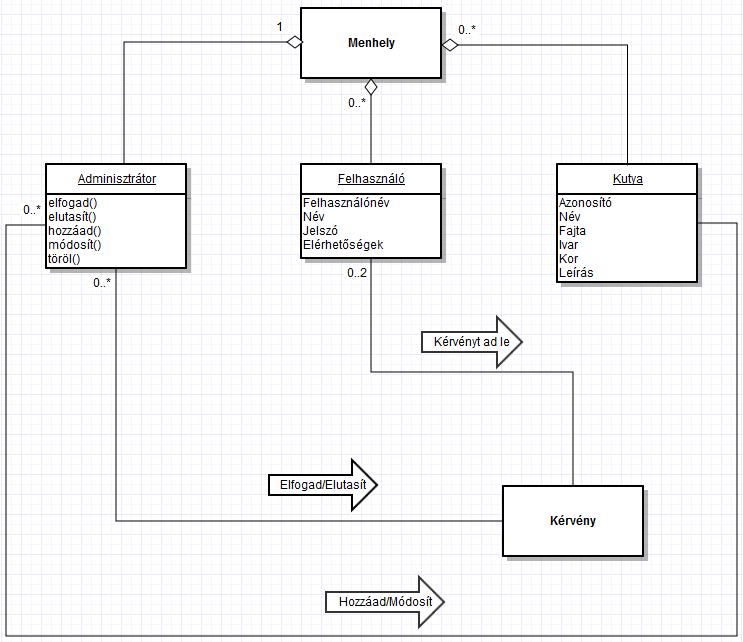
azonosító: a tábla kulcsa, ez határozza meg egyértelműen az egyedet

személy: aki az örökbefogadási igényt beadta

döntés: következő lehetséges paraméterek lehetnek : igen/nem/döntésre vár(alapértelmezett)

Egy másik diagrammal szemléltetve:

## UML diagram



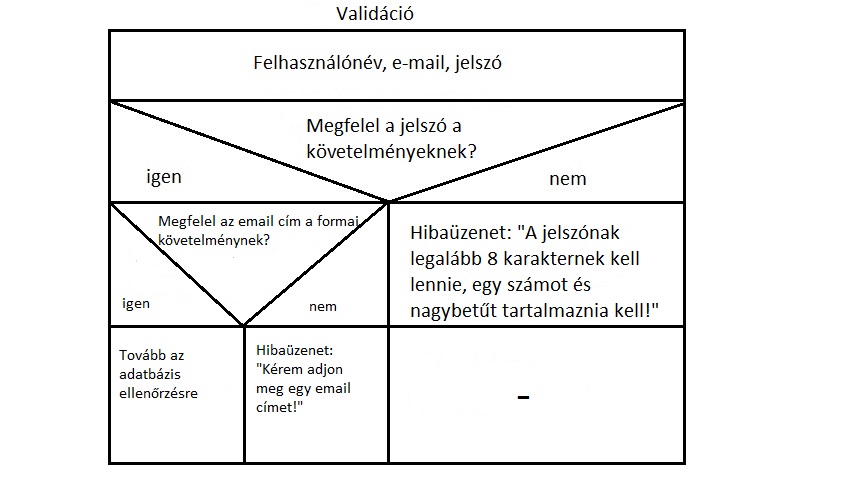
Az ábrán láthatjuk a Mancs állatmenhely portál UML diagramját, amelyből a következők olvashatók le:

* A menhelynek egy adminisztrátora van, aki különböző jogosultságokkal rendelkezik
  + elfogad() 🡪kérvényt
  + elutasít() 🡪kérvényt
  + hozzáad() 🡪kutyát
  + módosít() 🡪kutya adatait
  + töröl() 🡪kutyát
* A menhelynek tetszőleges számú regisztrált felhasználója lehet, akik maximum két kérvényt adhatnak le és az alábbi attribútumokkal rendelkeznek:
  + Felhasználónév
  + Név
  + Jelszó
  + Elérhetőségek
* A menhelyen tetszőleges számú kutya található, amik az alábbi attribútumokkal rendelkeznek:
* Azonosító
* Név
* Fajta
* Ivar
* Kor
* Leírás

# Felhasználói felület:

A munka legbonyolultabb, és leghosszabb része a **program megírása**. Mivel webes felületet tervezünk mi php és html kódolást használunk. A feladat elvégzésére minden funkció megtalálható bennük, mindemellett nem túl bonyolult, logikus nyelv.

Elsőként létrehozunk fő oldalt. A böngésző ezt tölti be az URL feldolgozásakor. Erre az oldalra rakunk akciógombokat, mint bejelentkezés, regisztráció. Egy felső sávon lesznek a menüpontok: Rólunk, Kutyáink, Elérhetőség stb. Az oldal középső részére egy rövid bemutatkozást írunk. A menü elemei linkek lesznek, amik a megfelelő oldalakra mutatnak (Rólunk és az Elérhetőség lehet html, a Kutyáink oldalnak viszont phpnak kell lennie, mert abban adatbázis hívások is szerepelni fognak.   
 Következő a regisztrációs oldal. Itt minden olyan adatot kérünk kell, amilyen rekord szerepel az adatbázisban. A hitelesség szempontjából figyelnünk kell a formátumokra. Fontos, hogy a telefonszámot egységesen adják meg, az e-mail-nak e-mail formája legyen stb. miután minden lépéssel végeztünk, teszteljük, hogy a függvényeink elmentik-e az felhasználók adatbázisba a regisztrált felhasználót. Sikeres regisztráció esetén kiíratunk egy üzenetet „sikeres regisztráció!” felirattal. Hiba esetén (pl.: e-mail, felhasználónév ütközés, hibás formátum) az annak megfelelő üzenetet íratjuk ki.   
A regisztráció szemléltetésére struktogrammokat használtunk:

****

A felhasználó mikor meglátogatja először az oldalt azzal fog szembesülni hogy felhasználói fiók nélkül nem tud örökbe fogadni. Egyszerű regisztrációt szeretnénk nyújtani leendő felhasználóinknak.

A regisztrációs struktogrammot 2 részre bontva:

**1.**

3 adatot kell megadni :

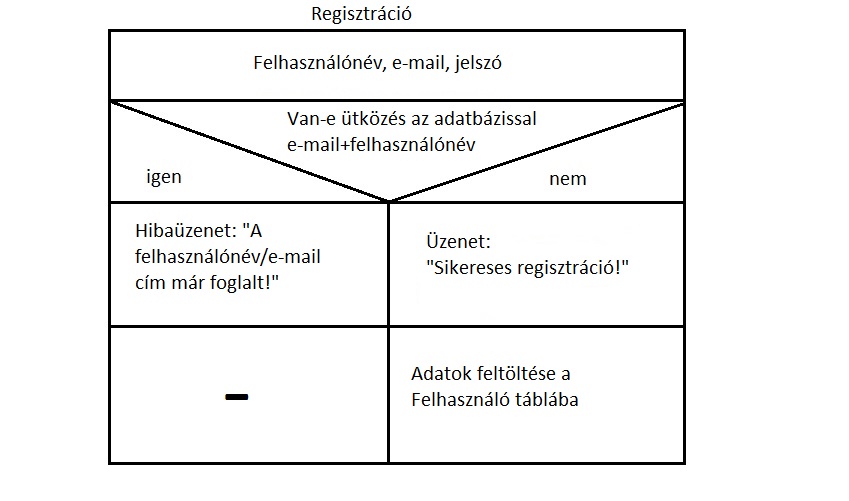
-felhasználónév

- e-mail

- jelszó

A validáció résznél ellenőrizzük a bevitt adatok helyességét, hogy pl. valós e-mail formátum lett-e megadva, kellően biztonságos jelszó lett-e megadva.

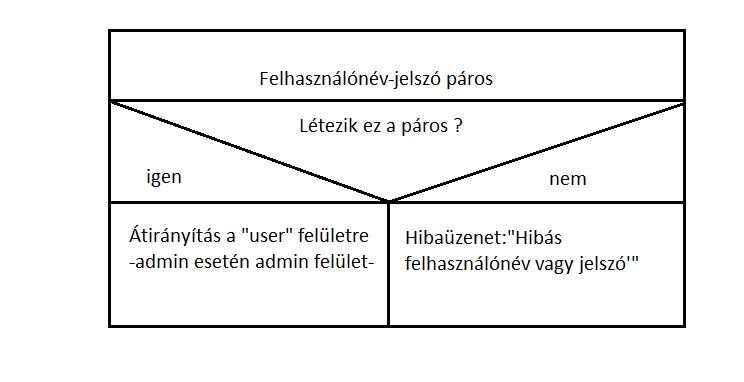
Ha a validáció részén átmegy a felhasználó akkor következik az adatok összehasonlítása az adatbáziséval, ütközéseket keresve.

****

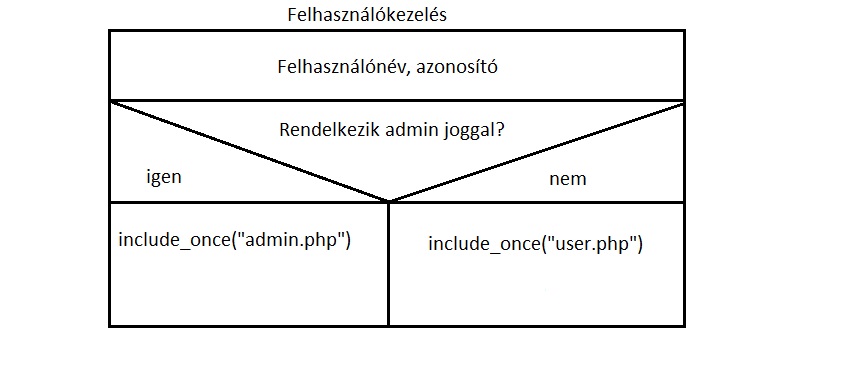
**2.**

Amint a felhasználó lenyomja a submit gombot átfésüli a rendszer az adatbázist duplikációkat keresve, ha talál azonos e-mail címet vagy felhasználót, a regisztráció meghiúsul, ha nincs akkor sikeres a regisztráció.

A regisztrálóoldal után következhet a bejelentkező oldal. A bejelentkezéshez szükséges a felhasználónév és a jelszó. Ezeket a program a felhasználók táblából olvassa ki, ez alapján azonosít. Hiba esetén a „hibás felhasználónév vagy jelszó” üzenet jelenik meg. Ha mindent jól adunk meg, akkor a gomb átirányít minket az úgynevezett „user” felületre (admin adatok esetén admin felületre).   
Struktogrammon ábrázolva:



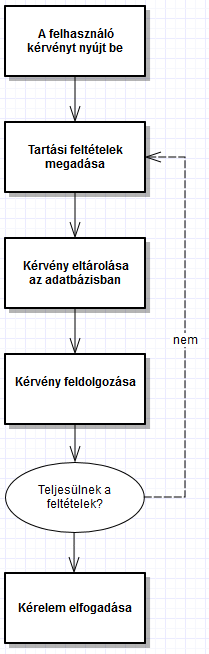
Bejelentkezés nagyon egyszerűen zajlik, a felhasználónév-jelszó páros ellenőrzésre kerül a Felhasználó táblában, ha nincs egyezés hibaüzenettel szembesül a felhasználó; ha van egyezés: beléptetjük a felhasználót és átirányításra kerül a felhasználói felületre.



A felhasználónév és azonosító által megvizsgáljuk hogy a felhasználó "egyszerű" felhasználó vagy pedig egyedi jogokkal rendelkező egyén (admin), ennek kimenetele alapján használjuk a fent látható kódokat.

A user felületen a felhasználók már leadhatnak örökbefogadási igényt az állatokra ezáltal válnak potenciális örökbefogadókból tényleges örökbefogadókká. Ezen kérelmeket az adminisztrátorok bírálják el, de egy külön menüpont alatt ezek döntések végkimenetelét a felhasználók is megtekinthetik.   
 Ha adminként jelentkeztünk be, akkor egy másik oldal fogad minket. Hasonló mint az előző, de egy adminnak lehetősége van új állatot feltölteni. Ki kell tölteni az adatokat, melyek a kutyákra vonatkozó adatbázisban szerepelnek (mindegyikre lehetőséget adunk). Ezen kívül el tudja bírálni a kutyákra leadott kérelmeket a megfelelő gombra kattintva. Eszerint lehet „elfogad”, „elutasít”, amíg nem döntött akkor pedig a „még nem döntött” (tehát ez alapértelmezetten beíródik).

A kérelmek leadását és elbírálását mutatja a következő ábra:



A felhasználó maximum két kutyára adhat le örökbefogadási kérelmet. A kérvényezése menete az ábrán látható és a következő módon zajlik:

* A felhasználó a regisztrációt követően a kérvények leadását lehetővé tevő menüpontra navigál
* A kizárólagosan erre a célra kialakított interfészen megadja az általa biztosított tartási feltételeket
* A kérvény eltárolása az adatbázisban a felhasználó adataival és a tartási feltételekkel együtt
* Az adminisztrátor, a bejelentkezést követően –az erre a célra létrehozott – felületen keresztül megkezdi a leadott kérelmek feldolgozását.

-Az örökbe fogadni kívánt kutya szükségleteinek, és a felhasználó által megadott tartási feltételek összevetése

-Amennyiben nem felel meg a felhasználó által megadott tartási körülmény, abban az esetben lehetősége van változtatni azon, különben elutasításra kerül a kérvénye.

* Ha a megadott tartási feltételek megfelelnek (vagy a javítást követően megfelelnek), akkor a kérelem elfogadásra kerül.

# Design:

Ha az adatbázisaink készen vannak, és a kódjaink is megfelelően működnek, akkor már csak a weboldalunk kinézetének megformázása maradt hátra. A weblapunk kinézete nagyban befolyásolja a róla alkotott véleményeket, így körültekintőnek kell lennünk ennél a folyamatnál is.   
Először is a színekről. Egy kellemes barnás árnyalatot választottunk a menüsornak. Ez a szín egy alsó, felső szegélyt fog eredményezni. A köztes részt fehér alapon hagyjuk, így a szem számára sem megterhelő böngészni az oldalt.   
Másodszor fontos a betűtípus megválasztása. Ha túl csicsás, akkor nehezen olvasható, de ha túl egyszerű akkor esetleg nem vennék komolyan. Ez mi projektünkben a szövegtörzsek Cambria betűtípust használnak, a menüpontok, és a fejlécfeliratok esetében ettől eltérünk.  
A harmadik fontos dolog design szempontjából pedig a képek használata. Ne árasszuk el a felhasználót sok képpel, elég csak néhány meggyőző, esetünkben azért, hogyha valaki idelátogat, regisztráljon, és fogadjon örökbe egy kutyát.

Mindezt összevetve egy kezdetleges skiccet rajzoltam, hogy hogyan nézhetne ki ez a weboldal:



Tartalom

[Bevezetés 2](#_Toc466570502)

[Architektúrális követelmények: 3](#_Toc466570503)

[Adatbázisterv: 3](#_Toc466570504)

[Egyed/kapcsolat diagram 3](#_Toc466570505)

[UML diagram 5](#_Toc466570506)

[Felhasználói felület: 6](#_Toc466570507)

[Design: 11](#_Toc466570508)