

Debreceni Egyetem

Informatikai Kar

**Egy Bioenergia alapanyagok begyűjtését  
nyilvántartó rendszer fejlesztése**

Témavezető:

Dr. Szabó István

tanszékvezető

**DE TTK**

**Szilárdtest Fizika Tanszék**

Készítette:

Kovács István

mérnök informatikus

hallgató

Debrecen

2011

## Nyilatkozat

Alulírott Kovács István, a Debreceni Egyetem Informatikai Karának hallgatója ezennel büntetőjogi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy Egy Bioenergia alapanyagok begyűjtését nyilvántartó rendszer fejlesztése című szakdolgozatom **saját, önálló munkám**; az abban hivatkozott nyomtatott és elektronikus szakirodalom felhasználása a szerzői jogok nemzetközi szabályainak megfelelően készült.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Debrecen, 2011. április 29.

aláírás

## Tartalomjegyzék

1. Bevezetés .....	4
2. Elméleti áttekintés .....	6
3. Feladatspecifikáció .....	9
4. Tervezési folyamatok .....	10
4.1 Adatbázis tervezése .....	10
4.2 Felületek tervezése.....	15
5. Megvalósítás.....	16
5.1 Adatbázis kapcsolat megvalósítása .....	16
5.2 Ügyfélkapu megvalósítása .....	16
5.3 Bioenergia alapanyagok feltöltésének megvalósítása.....	20
5.4 Új partner hozzáadásának megvalósítása.....	22
5.5 Partner adatainak módosítása .....	23
5.6 Adatok lekérdezésének megvalósítása.....	24
5.7 Adatok módosításának megvalósítása .....	27
5.8 Regisztráció megvalósítása .....	28
5.9 Belépési statisztika megvalósítása.....	29
5.10 Adatbázis mentésének megvalósítása .....	29
5.11 Statisztikai adatok ábrázolásának megvalósítása .....	30
6. Tesztelés .....	34
9. Irodalomjegyzék.....	37

# 1. Bevezetés

Világunk rohanó fejlődése mellett az energia felhasználásának igénye is jelentősen növekedik. Földünk energia készletei végesek, és egyre nagyobb fontossággal bírnak az alternatív energiaforrások (megújuló energiaforrások). Megújuló energiaforrásnak nevezzük azon természeti jelenségeket, melyekből az energia úgy nyerhető ki, hogy jelentősebb emberi beavatkozás nélkül legfeljebb néhány éven belül az újratermelődik. Az alternatív energiaforrások jelentősége, hogy használatuk összhangban van a fenntartható fejlődés alapelveivel, és nem okoznak környezet szennyezést. Alternatív energiaforrás az az energiahordozó, amelyből a jelenleg használatos szénhidrogének alternatívájaként valamilyen energiát tudunk kinyerni. Egyetlen módon menthetjük és óvhatjuk környezetünket, ha rendületlenül kutatjuk az alternatív energiaforrásokat. Alternatív energiaforrások például bioenergia, napenergia, szélenergia, vízenergia, geotermikus energia. [1]

A Hajdú-Komposzt Bt. 2004-ben az Európai Unió csatlakozását követően alakult meg, melynek fő tevékenysége bioenergia alapanyagok begyűjtése, mint például használt étolaj illetve élelmiszer hulladék. Ezen alapanyagokból biogázt állítanak elő. A biogáz szerves anyagok lebomlása során keletkezett gázelegy, amely levegőtől elzárt nedves körülmények között keletkezik. Kb. 45-70% metánt ( $\text{CH}_4$ ), 30-55% szén-dioxidot ( $\text{CO}_2$ ), nitrogént ( $\text{N}_2$ ), hidrogént ( $\text{H}_2$ ), kénhidrogént ( $\text{H}_2\text{S}$ ) és egyéb maradványgázokat tartalmaz. A biogáz felhasználása előtt tisztításon esik át, majd rövid ideig tárolják, mielőtt egy blokkfűtő erőműben elégetnék, és elektromos áramot, hőt termelnének belőle. [2]

A biogáz technológia fantasztikusan beilleszkedik a fenntartható fejlődés koncepciójába, így kiemelkedő szerepet kapott az Európai Unióban. Nem csak hogy környezetkímélő hatással bír, hanem csökkenti a légkörbe jutó üvegházhatású metángáz mennyiségét. [2]

A begyűjtött bioenergia alapanyagokról naprakész elektronikus nyilvántartást kell vezetni és adatot kell szolgáltatni a 164/2003. (X.18.) Kormányrendelet szerint. Így a cég azzal a kéréssel fordult hozzám, hogy fejlesszek ki egy olyan szoftvert, amely a begyűjtött alapanyagok információit tartja nyilván, illetve ezen adatokat dolgozza fel. A szakdolgozatom elkészítéséhez elengedhetetlen volt a cég működésének alapfokú ismerete. A begyűjtött alapanyagokat korábban Excel táblázatban vezették, amely nem csak, hogy időigényes, de

nem túl nagy hatékonysággal bírt. Céлом az volt, hogy egy olyan hatékony szoftvert fejlesszek, amely megkönnyíti a cég adatainak megfelelő nyilvántartását.

A cég számára fontos szereppel bír, hogy ez a rendszer bárhol elérhető legyen, így biztosítva a partnerei számára, a tőlük elszállított alapanyagok nyomon követését, ezért úgy döntöttem, hogy ezt a szoftvert webes alapokra helyezem. Mivel a PHP, mint programozási nyelv új volt számomra, így alapos elméleti és gyakorlati utánajárás volt szükséges a munka kezdetekor.



1. ábra: Nyírbátori biogáz feldolgozó üzem

Dolgozatom elkészítésének legfőbb lépései a következőképpen foglalható össze:

- feladat bemutatása
- a probléma megoldásához szükséges rövid elméleti áttekintés
- funkciók specifikálása
- tervezési folyamatok lépéseinek ismertetése
- alkalmazás implementálása forráskód szemléltetésével
- tesztelés
- tesztelés folyamán felmerülő problémák javítása

## 2. Elméleti áttekintés

Mi is az a honlap?

A honlap tulajdonképpen egy adott domain névhez tartozó, kódolt formában szerkesztett lapok gyűjteménye, amelyek tárhelyen vannak elhelyezve a szerveroldalon. A felhasználók saját gépükről egy böngésző szoftver segítségével a domain név beírásával kérik le a honlapot. A felhasználói oldalt kliens oldalnak nevezzük.

Alapvetően két féle honlap létezik:

- statikus: általában tájékoztató jellegű honlapok
- dinamikus: általában adatbázis támogatással működő weblapok

A dinamikus oldalak tartalma nem HTML állományban, hanem adatbázisban vannak tárolva. Az adatok motorja pedig a PHP programozási nyelv támogatásával valósul meg. A PHP rendszer első verziója 1994 körül jelent meg. Az első verzió kidolgozása Rasmus Lerdorf nevéhez fűződik. Az 1995-ös évben jelent meg egy javított, a FORM elemeket is kezelő PHP2 verzió. [6] Ekkor a PHP még csak az Apache WEB szerverrel tudott együttműködni. [6] A PHP3-as változat már kereskedelmi terméké is kinőtte magát, s piaci vagy ingyenes változatának is több százezer felhasználója lett világszerte. [6] A 2000-es évben már a PHP4 verzió is megjelent, melybe épült be először a Zend optimalizált értelmező és kibővített WEB szerver kapcsolatrendszer. [6] Ezen programozási nyelv egy szerver oldali scriptnyelv. [6] A válaszlap formátumát leíró utasítások HTML dokumentumba ágyazva helyezkednek el. [6] A PHP programozási nyelv két nagyon fontos programozási lehetőséget nyújt a programozó számára:

- eljárás orientált: forráskód szövege részekre tagolható, melyeket eljárásoknak nevezünk.
- objektum orientált: kielégíti az OOP alapelveket (egységbezárás, öröklés, sokalakúság).

Ahhoz hogy az adatbázisban tárolt adatokkal dolgozni tudjak szükség volt az MYSQL adatkapcsolatra, amely lehetővé teszi, hogy a PHP kódszeleten belül ne csak a saját vagy a környezeti változókat érijük el, hanem más adatbázis adatforrásokból is nyerhessünk ki

adatokat. E adattípus előnye, hogy egyrészt ezen keresztül számos más adatforrás is elérhető lesz. Az adatbázis megtervezése adatmodell segítségével történik. Az adatmodell a valóság leképezése adatokra illetve azok kapcsolataira. A modellnek tartalmazni kell az adatok közötti logikai összefüggést is. Ilyen adatmodell például a CDM (Conceptual Data Model).

### **CDM alapelvei:**

- a tervezés fogalmi szinten kezdődik
- a fogalmi modell szintjén nem foglalkozunk a fizikai implementálással
- az adatbázis általános logikai struktúráját adja meg, ami teljesen független bármilyen szoftvertől
- grafikus reprezentációját adja egy szervezet adatainak
- lehetőséget ad a tervezés validitásának ellenőrzésére
- lehetővé teszi a fizikai adatmodell generálását (PDM)

[3]

### **CDM objektumai:**

- adatelem: az információ legegyszerűbb része a tervezés folyamán
- domain: egy értékhalmoz, amelyből az adatelem felveheti az értékét
- egyed: bármit reprezentálhat, amely érdekes a probléma szempontjából, melyről információt szeretnénk tárolni
- egyed attribútum: olyan adatelem, amelyet az egyedhez rendelünk, mint az egyed jellemzőjét
- azonosító: az azonosító egy egyed attribútuma, vagy ezek kombinációja, melynek értéke egyértelműen azonosítja az egyedet minden előfordulásánál. Minden egyednek legalább egy azonosítóval rendelkeznie kell.
- elsődleges azonosító: az azonosító, amelyet kiemelünk a többi azonosító közül és a továbbiakban az egyed azonosítására használjuk. Csak egy elsődleges azonosító van.
- kapcsolat: névvel ellátott viszony vagy összefüggés egy vagy több egyed között
- reflexive kapcsolat: amikor az egyed önmagával áll kapcsolatban

[3]

## Mi is az a JavaScript?

Az első 1.0-s verzióját a Netscape dolgozta ki 1995-ben. Azóta szinte évente adnak ki újabb verziókat. 2005-ben az 1.6-s verziója jelent meg.

A JavaScript egy objektumorientált programozási nyelv, amit leginkább webes felületek fejlesztésekor alkalmaznak. Alkalmazásával a weblap felhasználóbarátabbá, interaktívabbá és látványosabbá válik. A JavaScript kódok a felhasználó számítógépén futnak le, azaz kliens oldali programozási nyelv. A kódok által előállított eredmények függenek a böngésző típusától, amiről a programozónak nem szabad megfélekednie. [5]

A szakdolgozatom elkészítéséhez e két fontos programozási nyelvet használtam (JavaScript, PHP5). PHP5 programozási nyelvénél mind a két programozási típust alkalmaztam, hogy szemléltessem mind az eljárás orientált mind pedig az objektumorientált programozást.

### 3. Feladatspecifikáció

Hajdú-Komposzt Bt. két féle bioenergia alapanyagokat gyűjt be (élelmiszer hulladék és használt étolaj). A begyűjtés nagyobb étkeztetéssel foglalkozó intézményekből történik. Kiszállítás során, a cég szállítóleveleken rögzíti a mért súlymennyiségeket digitális mérleg segítségével a helyszínen. A rendszerbe való rögzítés a szállítólevelek alapján történik. Cél egy olyan egyedi szoftver elkészítése, amely a cég által begyűjtött bioenergia alapanyagokat elektronikusan tartja nyilván. Hasonló tevékenységgel foglalkozó cégek e rögzítési módot még nem alkalmazzák, így számomra hatalmas kihívásnak mutatkozott e probléma megoldása. Feladatomból volt minden olyan funkció biztosítása, amely a begyűjtés során felmerülő adatok tárolásához, szerkesztéséhez szükséges. Első lépésem egy tájékoztató jellegű weblap létrehozása volt, amely információt nyújt a cég működéséről a látogatók számára. Következő lépésben lehetőséget kellett biztosítanom az adatok rögzítését, lekérdezését, illetve szerkesztését. Ezen műveletek elérése már nem publikus feladat, így implementáltam egy ügyfélkaput, amely biztosította, hogy megfelelő jogosultsággal rendelkező felhasználók férjenek hozzá. A weblapon szabad regisztráció nem lehetséges, így regisztrációs szándékát a felhasználónak jelezni kell a cég munkatársainak, akik a regisztrációt elvégzik. Regisztrációs igényt a céggel partneri kapcsolatban lévő ügyfelek igényelhetnek. A belépés engedélyezése után a partnerek adataikat nyomon követhetik, illetve ellenőrzés során az adott szervezet fele igazolásként kinyomtathatják, amely nagy dinamizmusra tesz szert. Ez a szolgáltatás a cég és partnerei közötti kapcsolatot szorosabbá tette.

Az adatok tárolása volt az elsődleges cél, amely a későbbiek során módosult. Fontosnak tartottam, hogy a tárolt adatok segítségével statisztikai adatokat állítsak elő és grafikus szemléltetsem, ezzel segítve a cég működésének hatékonyságát.

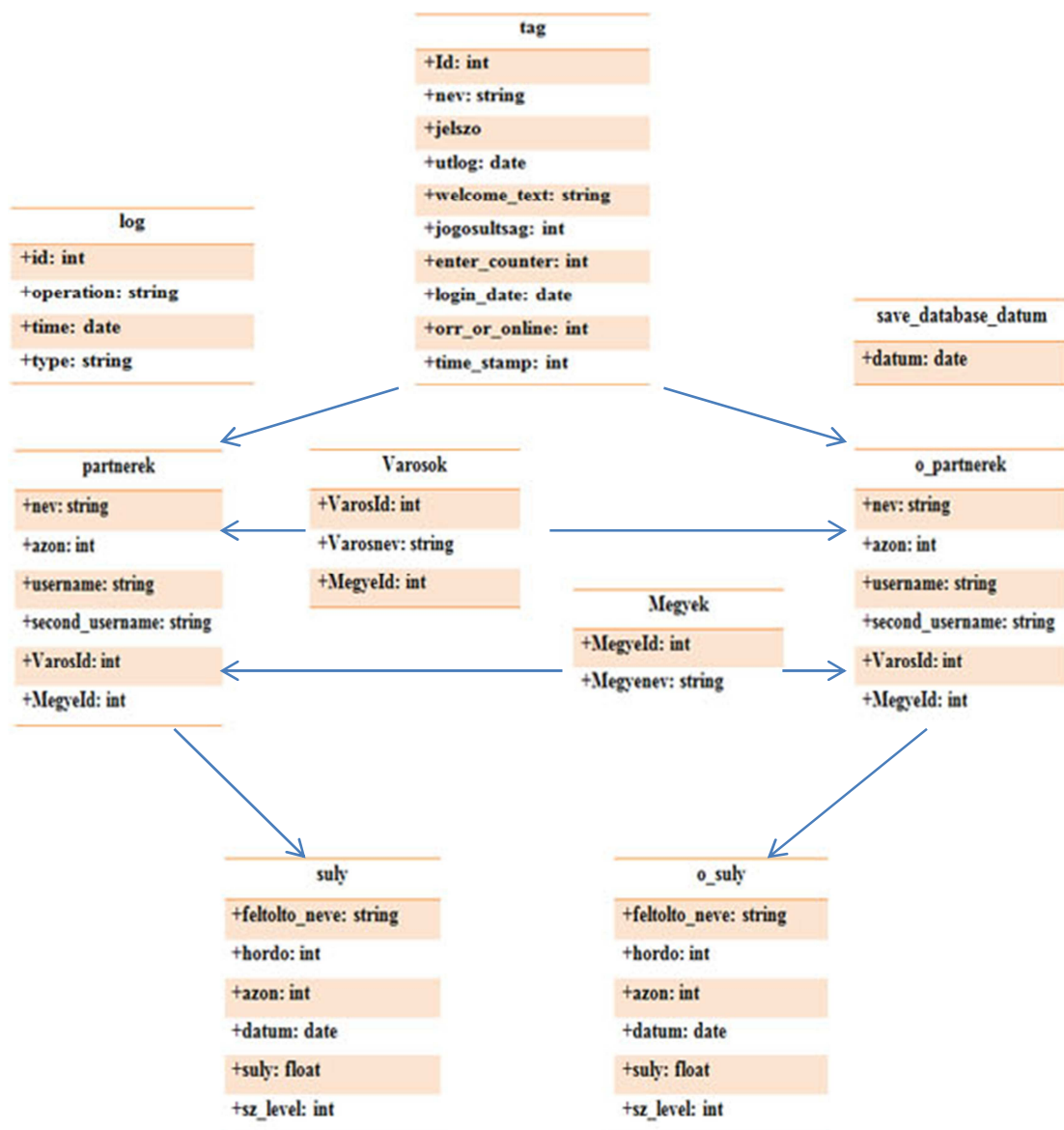
A feladat implementálása közben újabb igények merültek fel, amelyek nem szerepeltek a tervezési fázisban. A cég folyamatos fejlődése megkövetelte, hogy a partnerek ne csak név alapján legyenek rögzítve, mert előfordult, hogy több településen is létezett azonos nevű létesítmény. Minden partnert az adatbázisban egyedi kulcs jellemez, ami szerint nem lenne szükség egyéb adat tárolására, de a moderátor felületen ezek az azonosítók nem láthatóak csak a partner neve. Ezért a problémának orvosolására kibővítettem a partner rögzítését egy megye illetve település megjelölésével is. Ezen adatok alapján egy partner már konkrétan beazonosíthatóvá vált a moderátor számára.

## 4. Tervezési folyamatok

### 4.1 Adatbázis tervezése

A munkafolyamatot két lényeges fázisra bontottam.

Első fázisban az adatbázis elméleti megtervezését az elméleti adatmodell (CDM - Conceptual Data Model) segítségével készítettem el. Összegeztem minden olyan lehetséges adatot, amelynek szerepelni kell az adatbázisban, majd a táblák egymáshoz való kapcsolódását.



2. ábra: CDM modell

Az adatbázis fizikai adatmodelljének létrehozása volt a második lényeges lépés. Fizikai tervezés során a logikai szerkezet hatékonyságára koncentráltam. A tervezést phpMyAdmin szoftver segítségével valósítottam meg.

## Az adatbázisban elhelyezkedő táblák

A biogáz nyilvántartó rendszer jelenleg 9 táblát tartalmaz.

A **tag** tábla tartalma:

Mezők neve	Funkciója	Típusa és hossza
id	felhasználó azonosítója	int 11
nev	felhasználói név tárolása	varchar 40
jelszo	felhasználó jelszavának tárolása	varchar 40
utlog	felhasználó utolsó bejelentkezésének ideje	int 11
welcome_text	felhasználó teljes neve	varchar 40
jogosultsag	felhasználó jogosultsági szintje	int 1
enter_counter	felhasználó belépéseinek száma	int 30
off_or_online	felhasználó aktivitása	int 1
timestamp	felhasználó utoljára végrehatott művelete időben	varchar 40

A **tag** táblázat segítségével kerül nyilvántartásra a regisztrált felhasználók. A táblázatban található adatok felhasználásával történik a belépés, a jogosultsági szintek kiosztása, és egyéb statisztikai adatok tárolása.

A **partner** tábla tartalma:

Mezők neve	Funkciója	Típusa és hossza
nev	partner neve	varchar 70
azon	partner azonosítója	int 11
username	partner felhasználó neve	varchar 20
second_username	központi egység felhasználó neve	varchar 20
VarosId	város azonosítója	int 5
MegyeId	megye azonosítója	int 5

A cég azokat a partnereket tartja nyilván ebben a táblázatban, akiktől bioenergia alapanyagokat gyűjt be. A **username** mezőben az ügyfelek felhasználónevének megadásával engedélyezhetjük, hogy hozzáférjenek adataikhoz. Ha egy partner több telephellyel rendelkezik, akkor a **second\_username** mezőbe a központ felhasználóneve kerül, melynek jogosultsága van az összes telephely adatainak megtekintésére.

A **suly** tábla tartalma:

Mezők neve	Funkciója	Típusa és hossza
feltolto_neve	a feltöltést végző moderátor neve	varchar 20
hordo	hordók darabszáma	int 3

azon	partner azonosítója	int 11
datum	szállítás ideje	date
suly	biogáz alapanyag súlya	float (8, 1)
sz_level	szállítólevél száma	int 11

A **suly** táblázat elengedhetetlen, mivel ide kerül rögzítésre az élelmiszer hulladék alapanyagok összes fontosnak vélt adatai, illetve a feltöltést végző moderátor neve, így hibás feltöltés esetén, visszaellenőrizhető a feltöltést végző személy.

A **save\_database\_datum** táblázat tartalma:

Mezők neve	Funkciója	Típusa és hossza
datum	az adatbázis utolsó lementésének időpontja	date

Az adatbázis biztonsági mentésének időtartama található a táblázatban. Az adatok elvesztésének meggátolása érdekében nagyon fontos szereppel bír.

A **log** tábla tartalma:

Mezők neve	Funkciója	Típusa és hossza
id	a műveletet végző személy azonosítója	int 11
operation_data	az az adat, amin a művelet megtörtént	varchar 150
time	módosítás ideje	date
type	művelet típusa	varchar 50

Az adatok manipulációja ebben a táblázatban rögzül. A táblázatnak az elérése csak az adminisztrátor számára lehetséges.

A **VarosID** tábla tartalma:

Mezők neve	Funkciója	Típusa és hossza
VarosID	város azonosítója	int 11
VarosNev	város nevét tartalmazza	varchar 50
MegyeID	megye azonosítója	int 11

A VarosID táblázat Magyarország összes települése található. A táblázat szükségességét a partnerek teszik, a konkrét beazonosítás miatt.

A **MegyeID** tábla tartalma:

Mezők neve	Funkciója	Típusa és hossza
MegyeID	megye azonosítója	int 11
MegyeNev	megye nevét tartalmazza	varchar 20

A MegyeID táblázatban Magyarország megyéi rögzülnek.

A fennmaradó másik két tábla struktúrája teljesen megegyezik az **parter** és **suly** táblázatok struktúrájával. Abban tér el, hogy **o\_partner** és **o\_suly** táblázatokban a használt étolaj bioenergia alapanyag kerül tárolásra.

## 4.2 Felületek tervezése

Az adatbázisban tárolt adatok megjelenítése, illetve az adatok feltöltéséhez jól átlátható felületre volt szükség, aminek a megtervezése újabb feladatot jelentett számomra. Fontos szempont volt, hogy a felületek minden webes szabványnak megfeleljenek, így biztosítva különböző böngészők megjelenítése során a hasonlóságot. A felület megjelenítését CSS (Cascading Style Sheets) segítségével valósítottam meg, amely egy stílusleíró nyelv. Egy CSS stíluslap utasítások felsorolásából áll. A böngészők mindegyike támogatja a CSS-t, de megjelenítési különbségek vannak, ezért a szabványok betartása elengedhetetlen volt. A stíluslapokat általában külön fájlban tároljuk \*.css kiterjesztéssel.

## 5. Megvalósítás

A megvalósítást több fázisra bontottam a hatékonyság érdekében.

### 5.1 Adatbázis kapcsolat megvalósítása

Az adatbázis eléréséhez létre kell hozni az adatbázis kapcsolatot, amelyet egy config.php nevű fájl segítségével hoztam létre. A config.php fájl a következőket tartalmazza

```
<?php
    $felh = "felhasználó_neve";
    $jelszo = "jelszava";
    $base = "adatbázis_neve";
    $host = "host";
    $conn = mysql_connect( $host, $felh, $jelszo ) or die("Hiba"); //Adatbázis csatlakozás
    mysql_select_db( $base, $conn ) or die("Adatbázis hiba" );
?>
```

Az adatok beállítása után létrejön az adatbázis kapcsolat, amelynek segítségével dolgozhatunk az adatbázisban. Amelyik fájlban adatbázis kapcsolatra van szükség, mindenféleképpen be kell szúrni a fájl elejére a következő sort „include("config.php")”, így létesítve a kapcsolatot.

### 5.2 Ügyfélkapu megvalósítása

Megvalósítottam az ügyfélkaput COOKIE alapokra fektetve (6. ábra). A COOKIE a web szerver által a web böngészőnek küldött üzenet, melyet a böngésző a felhasználó saját számítógépének háttértárán egy fájlban tárol. Az üzenet tartalmát az üzenetet küldő szerver bármikor lekérdezheti a böngészőtől. A weboldalak üzemeltetői COOKIE-kat elsősorban a visszatérő felhasználók azonosítására használják. A tesztelések során viszont kiderült, hogy nem volt túl biztonságos. Ha a felhasználó nem jelentkezett ki szabályosan és bezárta a böngészőt kijelentkezés nélkül, böngésző újraindításakor bejelentkezve maradt, így hozzáférhetővé téve az adatokat illetéktelen felhasználók számára. Ezért úgy döntöttem, hogy munkamenetek (SESSION) segítségével oldom meg az ügyfélkaput. Mivel az alkalmazás elég nagy részét SESSION-ökre és azok kezelésre épül, figyelni kell a megfelelő session beállításokra az oldalon. Általában a szerver

gyökerében Temp könyvtárba kerülnek rögzítésre a munkamenetek, de a PHP `save_session_path()` függvényével megadható az a könyvtár, útvonal, ahová a rendszer elhelyezheti a munkamenet-állományokat. A belépést egy felhasználónév és egy jelszó páros megadásával történik, amelyet a rendszer ellenőriz. Helyes páros megadása esetén a rendszer belépteti a felhasználót, ellenkező esetben „**Rossz felhasználónév vagy jelszó!**” hibaüzenettel tér vissza. Belépést követően érhető el a műveletközpont, amely segítségével manipulálható illetve lekérdezhető az adatbázis tartalma. A felhasználói szinteket három részre bontottam, melynek segítségével történnek a jogosultságok kiosztása. Ezek a jogosultsági szintek a regisztráció során rendelődnek az adott felhasználóhoz. A legalsó szinten álnak a felhasználók, akik a céggel partneri kapcsolatban állnak. Ők rendelkeznek a rendszerben a legkevesebb jogosultsággal. Csak a hozzájuk tartozó adatokhoz férnek hozzá lekérdezési vagy nyomtatási szinten. Az adatbázisban, a **tag** táblázatban és **jogosultsag** mezőben 3-as értékkel rendelkeznek.

←T→	id	nev	jelszo	utlog	welcome_text	jogosultsag	enter_counter	login_date	off_or_online	timestamp
<input type="checkbox"/>	16			1302163476	Magyar Posta Zrt.	3	5	2011-04-07	0	2011-04-07 10:07:05

Ez az érték a belépés során Session-ban tárolódik.

```
$_SESSION['jogosultsag'] = $row["jogosultsag"];
```

Az érték hozzárendelése a `log.php` fájlban található.

A második szinten a moderátorok állnak, akik a cég munkatársai. Ők végzik az adatok manipulálását a rendszerben. Azaz sokkal több jogosultsággal rendelkeznek, és 2-es értékkel szerepelnek az adatbázisban.

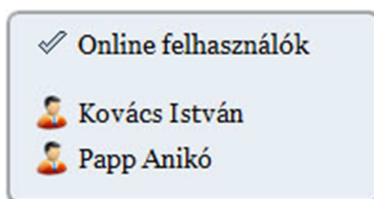
←T→	id	nev	jelszo	utlog	welcome_text	jogosultsag	enter_counter	login_date	off_or_online	timestamp
<input type="checkbox"/>	11			1302184359	Papp Anikó	2	70	2011-04-07	0	2011-04-07 16:14:58

A legfelső szinten az adminisztrátor áll, akik a rendszert megfelelő működését felügyeli. Azaz sokkal több jogosultsággal rendelkeznek, és 1-es értékkel szerepel az adatbázisban.

←T→	id	nev	jelszo	utlog	welcome_text	jogosultsag	enter_counter	login_date	off_or_online	timestamp
<input type="checkbox"/>	1			1302548115	Kovács István	1	346	2011-04-11	1	2011-04-11 21:05:34

Belépés során még rögzítésre kerül a belépések száma, illetve az utolsó belépés dátuma, amely egy jó statisztikai adatot tükröz vissza. Lekérdezésre kerül az **enter\_counter** mező tartalma az aktuális felhasználónál, majd megnövelve eggyel felülírásra kerül a táblázatban. Így nyilvántartva a felhasználók belépését.

Az éppen bejelentkezett felhasználók aktivitását is nyilvántartja a rendszer, ezzel szemléltetve az aktív felhasználókat (3. ábra). Ezt úgy oldottam meg, hogy a felhasználóhoz egy 0 vagy 1 értéket rendeltem, amit az **off\_or\_online** mezőben található. Ha ennek a mezőnek 1 az értéke a felhasználó bejelentkezett állapotban van ellenkező esetben pedig inaktív. Az érték módosítását a kijelentkezés gomb végzi. Azaz ha a felhasználó kijelentkezik, a mező tartalma 0 értéket veszi fel. Ellenkező esetben, ha a felhasználó nem szabályosan jelentkezik ki a rendszerből, azaz bezárja a böngészőt kijelentkezés nélkül, a következő belépő vagy már belépett felhasználó módosítja a mező tartalmát 0-ra, ha valamelyik felhasználó **timestamp** mezőben rögzített ideje régebbi 10 percnél. Így a felhasználók ültetik ki egymást, ha az időkorlát letelt, megvalósítva az online felhasználók hitelességét.



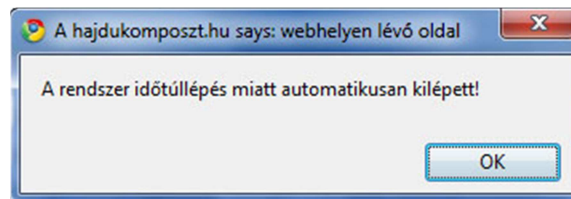
3. ábra: online felhasználók felülete

A **timestamp** mező tartalma csak az index.php fájl meghívása esetén frissül, mivel ezen az oldalon látható az online felhasználók listája. Ezt a timestamp() függvény végzi, ami a következőképpen néz ki.

```
//Időbélyeg
function timestamp()
{
    mysql_query("UPDATE tag SET timestamp = now() where nev = '$_SESSION[username]'");
}
```

Az ügyfélkapuban szükség volt implementálni egy automatikusan kiléptető rendszert időkorláttal az adatok biztonsága miatt. Ami azt jelenti, hogy ha a felhasználó nem végez semmi féle műveletet az adott felületen, 10 perces időkorlát után,

automatikusan kilépteti a rendszer. Ezt az adatot is a **time\_stamp** mezőből nyerem ki. Frissítésre művelet végrehajtáskor kerül sor, a felhasználó így igazolva aktív jelenlétét.

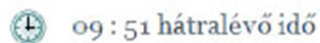


4. ábra: automatikus kijelentkezés figyelmeztető box-ja

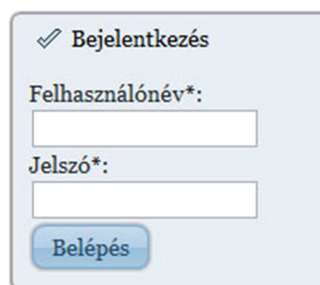
Az időkorlát grafikus szemléltetését csak az index.php web helyen látható (5. ábra). A visszazámlálás dinamizmusa miatt javascript programozási nyelven implementáltam. A kód első két sorában állítható be az időkorlát, perc és másodperc megadásával.

```
var mp = 00; // itt add meg a masodperceket  
var perc = 10; // itt add meg a perceket
```

Az időkorlát lejártával egy alertbox segítségével tájékoztatja a felhasználót az automatikus kiléptetésről (4. ábra).



5. ábra: időkorlát megjelenítési stílusa



6. ábra: bejelentkezési felület

## 5.3 Bioenergia alapanyagok feltöltésének megvalósítása

Ezen a felületen kerülnek rögzítésre a begyűjtött bioenergia alapanyagok. A szállítólevélen elhelyezkedő összes adat rögzítésre kerül, és még egyéb plusz információ. Az adatok az alábbiakból tevődnek össze: begyűjtött nettó súlymennyiség, szállítás dátuma, melyik partnertől történt a szállítás, mennyi hordóban helyezkedett el az adott súlymennyiség, biogáz alapanyag típusa, szállítólevél száma illetve a feltöltést végző személy neve. Fontos hogy az adatbázisba ne kerülhessenek valótlan adatok, ezért több mezőben lekezelésre kerültek a hibás adatok megadása. Mivel ezt dinamikusan figyelni kell a gépelés folyamán, ezért javascript segítségével oldottam meg a problémát. A szállítólevélszám mindig egyedi és csak számokból összetevődő adat, ezért írtam egy olyan függvényt, ami a szám karaktereken kívül nem enged egyéb más karakterek bevitelét. Ez a metódus az `only_number.js` fájlban található meg.


```
//Csak számot bevitelnek figyelése
function isNumberKey(evt,adat)
{
    var charCode = (evt.which) ? evt.which : event.keyCode
    return (charCode==8 || (charCode >= 48 && charCode <= 57 && adat.value.length<6))
}
```

A mennyiség mezőben float típusú számok kerülnek bevitelre, így ebben a mezőben nem csak a szám, hanem a pont vagy vessző karaktert is engedélyeznem kellett. Ezt az ellenőrzést az `only_float_number.js` fájl végzi. A float típus pont elválasztásával történik, de a moderátorok munkáját megkönnyítve mivel a numerikus billentyűn csak vessző található, így vessző esetén a program automatikusan kicseréli pontra, ezzel segítve a moderátor munkáját.

```
//A súly bevitelnél a , karaktert .-ra cseréli
function karakter_csere($szo)
{
    for($i = 0; $i <= strlen($szo); $i++)
    {
        if($szo[$i] == ",")
        {
            $szo[$i] = ".";
        }
    }
    return $szo;
}
```

Következő a hordó mező, amely alapértelmezetten egyes értékkel rendelkezik, mert többségében ez a domináló szám a szállítóleveleken. Az utolsó mező a dátum kiválasztása. Itt nagyon lényeges volt, hogy ne lehessen valótlan dátumot megadni pl. 2011.02.30. Ezt a legordulo\_datum.js fájl hatja végre. Az év legördülő menüben mindig az aktuális év kerül, hogy ne lehessen manipulálni visszamenőleg az adatokat. A hónap legördülő menüben mindig az aktuális hónapra van beállítva alapértelmezetten, így elősegítve a moderátor munkáját. A mennyiség mezőben a bruttó súlymennyiség kerül kitöltésre. Az alkalmazás annyiszor vonja le az 1.6 kg-ot (hordó tiszta súlya), ahány hordó bevételre került a hordó mezőben, így a nettó súly kerül tárolásra. Az adatok feltöltése mind a két biogáz alapanyag esetén ugyan úgy történik, csak más adatbázis táblázatba kerülnek rögzítésre.

### Élelmiszer hulladék feltöltése



Partner:	<input type="text" value="Admirális Panzió"/>
Szállítólevél száma:	<input type="text" value="22547"/>
Mennyiség:	<input type="text" value="28,9"/>
Hordók száma:	<input type="text" value="1"/>
Szállítás dátuma:	<input type="text" value="2011"/> <input type="text" value="Április"/> <input type="text" value="1"/>

Vissza

7. ábra: feltöltést végző felület

## 5.4 Új partner hozzáadásának megvalósítása

A cég folyamatos fejlődése mellett a partneri kör is állandó növekedést mutat. Ezért kihagyhatatlan volt az új partnerek feltöltésének lehetősége. A partnerek azonosítása ID alapján történik az adatbázisban. Az ID mezőt lehetett volna auto\_increment módban használni, de írtam egy saját függvényt ennek megvalósítására. A függvény lekérdezi az adatbázisban a legnagyobb ID értékét majd ezt eggyel megnövelve rendeli hozzá az új partner ID-jához.

```
mysql_query("SET NAMES 'utf8'");  
$azon_get = @mysql_query("SELECT Max(azon) FROM ".$table."  
$azon = intval(mysql_result($azon_get, 0, 0));  
$azon = $azon + 1;
```

Partner hozzáadását követően töltheti fel a moderátor a partnerhez tartozó bioenergia alapanyagokat. Partnert csak moderátor vagy adminisztrátor adhat hozzá. A moderátorok a felületen három adat megadásával adhatnak új partnert a rendszerben. A partner neve, melyik megyében helyezkedik el majd utoljára hogy melyik városban. A partner hozzáadását végző forráskód a left\_over\_partner\_add.php-ban található.

### Új Partner hozzáadása(élelmiszer hulladék)

Partner neve:

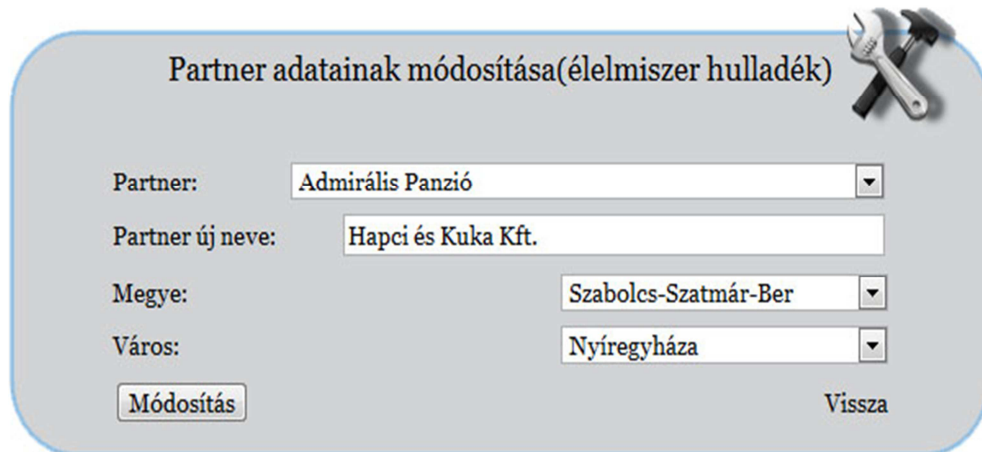
Megye:

Város:

8. ábra: új partner hozzáadását végző felület

## 5.5 Partner adatainak módosítása

Ezen a felületen a már meglévő partnerek adatainak módosítása végezhető. A partner adata három részből tevődik össze név, megye, város. Nagyon sok esetben módosul az ügyfél adata, így biztosítani kell a módosítás lehetőségét. Először a moderátornak ki kell választani, melyik cég adatait kell módosítani, majd az új név, megye és város megadásával módosíthatja a kívánt adatot. Módosítás után megvizsgálja a rendszer az adatbázisban, hogy megtalálhatóak az új adatok. Ha megtalálhatóak, akkor kiírja, hogy a „Partner módosítása sikeres” egyébként „Módosítás sikertelen”. A forráskód a `left_over_partner_modify.php` fájlban található.



Partner adatainak módosítása(élelmiszer hulladék)

Partner: Admirális Panzió

Partner új neve: Hapci és Kuka Kft.

Megye: Szabolcs-Szatmár-Ber

Város: Nyíregyháza

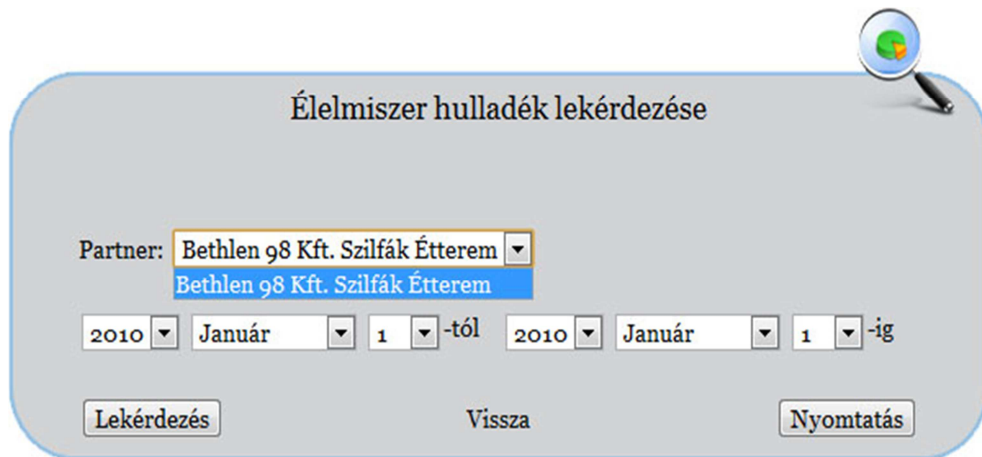
Módosítás Vissza

9. ábra: partner adatainak módosítását hozzáadását végző felület

## 5.6 Adatok lekérdezésének megvalósítása

A felület minden felhasználó számára elérhető, de más felhasználási jogosultsággal. Ezen a felületen tarthatják számon az ügyfelek a tőlük elszállított biogáz alapanyagokat. Adatvédelmi okokból meg kellett valósítani, hogy mindenki csak a rá vonatkozó adatokat láthassa. Ezt úgy valósítottam meg, hogy belépéskor az ügyfél felhasználónevét eltárolom egy SESSION-ba. A lekérdezés ablak megnyitásakor a SESSION-ban eltárolt adat alapján végignézi a rendszer, hogy a partner tábla user vagy second\_user mezőben található ilyen felhasználónév. Ha található, akkor a partner selectet csak azzal a partnerrel tölti fel, ahol egyezés volt. Így az ügyfél csak saját adataihoz fér hozzá.

```
//Partnerek lekérdezése
mysql_query("SET NAMES 'utf8'");
$partnerek_get = @mysql_query("SELECT nev FROM partnerek WHERE username = '". $_SESSION['username']."'");
$sorok = @mysql_num_rows($partnerek_get);
if($sorok == 0)
{
    $partnerek_get = @mysql_query("SELECT nev FROM partnerek WHERE second_username = '". $_SESSION['username']."'");
}
```



10. ábra: ügyfél adatlekérdezési felülete

Az időköz megadásával lekérdezheti az adatokat. Az adatok lekérdezése egy táblázatban jelenik meg napi lebontásban. Az utolsó sorban összegzi a lekérdezett időintervallum összes súlyát. Az ügyfél nyomtathatja ezeket az adatokat igazolásként. A nyomtatást CSS segítségével oldottam meg. Az volt a cél, hogy ne az kerüljön kinyomtatásra ami a

képernyőn látszik, hanem egy formanyomtatvány a lekérdezett adatokkal. Ebben az esetben a nyomtatóhoz és a képernyőhöz külön megjelenítési stílusokat rendelhetünk hozzá CSS segítségével. Így nyomtatáskor nem a képernyőn megjelenített stílusban kerül nyomtatásra. A képernyő megjelenítéséhez a style.css alkalmaztam (11. ábra).

```
<link rel="stylesheet" href="style.css" type="text/css" media="screen" />
```

Lekérdezett adatok			
EWC 20 01 08			
Partner: Agrár Étterem			
Időintervallum: 2010-1-1-től 2011-1-1-ig			
Szállítás dátuma	Szállítólevél száma	Hordó	Nettó súlymennyiség
2010-08-19	17457		22.9 kg
2010-08-25	17385		17.5 kg
2010-08-27	17315		28.1 kg
2010-08-31	17256		31.5 kg
2010-09-03	17165		27.5 kg
2010-09-07	17148		16.4 kg
2010-09-10	17948		25.9 kg
2010-09-14	17226	1	16.4 kg
2010-09-16	17430	1	33.3 kg
2010-09-21	17571	1	20.2 kg
2010-09-24	17666	1	30.7 kg
2010-09-28	17733	1	31.1 kg
Összesen			301.5 kg

11. ábra: lekérdezett adatok

A nyomtatás megjelenítéséhez pedig a print.css használtam (12. ábra).

```
<link rel="stylesheet" href="print.css" type="text/css" media="print" />
```

2011.04.24

Hajdú - Komposzt Bt. \\\



4251 Hajdúsámson, Rednői u. 1  
E-mail: [info@hajdukomposzt.hu](mailto:info@hajdukomposzt.hu)  
Web: [www.hajdukomposzt.hu](http://www.hajdukomposzt.hu)

### Lekérdezett adatok

EWC 20 01 08

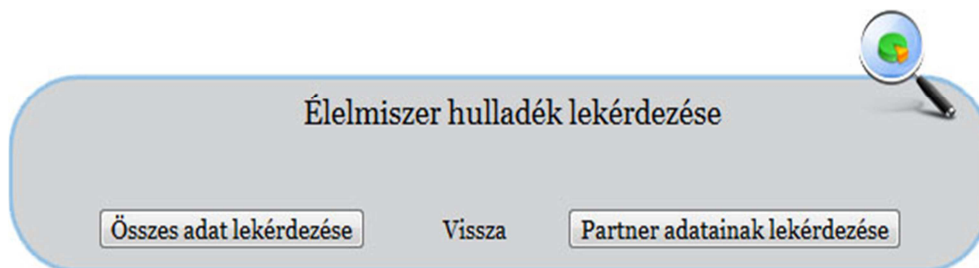
Partner: Agrár Étterem

Időintervallum: 2010-1-1-től 2011-1-1-ig

Szállás dátuma	Szállólevél száma	Hordó	Nettó súlymennyiség
2010-08-19	17457		22.9 kg
2010-08-25	17385		17.5 kg
2010-08-27	17315		28.1 kg
2010-08-31	17256		31.5 kg
2010-09-03	17165		27.5 kg
2010-09-07	17148		16.4 kg
2010-09-10	17948		25.9 kg
2010-09-14	17226	1	16.4 kg
2010-09-16	17430	1	33.3 kg
2010-09-21	17571	1	20.2 kg
2010-09-24	17666	1	30.7 kg
2010-09-28	17733	1	31.1 kg
<b>Összesen</b>			<b>301.5 kg</b>

12. ábra: lekérdezett adatok nyomtatott formátumban

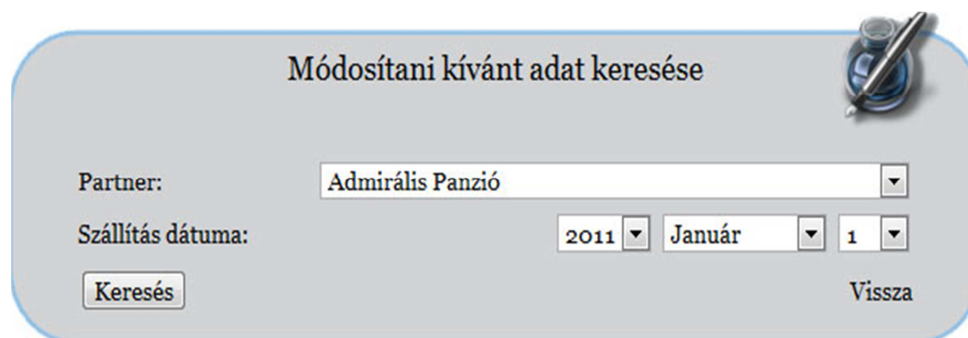
A moderátor és az adminisztrátor felülete (13. ábra) annyiban tér el, hogy ők jogosultak az összes partner adatainak lekérdezésére, illetve összesített adatok eléréséhez. Az összes adat az összes partner súlyának összege a kívánt időintervallumban.



13. ábra: moderátor lekérdezésének felülete

## 5.7 Adatok módosításának megvalósítása

Erre a felületre azért volt nagy szükség, mert a moderátor feltöltés során hibázhat, és a hibát pedig ezen a felületen lehet korrigálni. A módosítás az elrontott adat megkeresésével kezdődik. Itt meg kell adnunk, hogy melyik partnernél és melyik napon történt a szállítás (14. ábra). Ezen két adat alapján a rendszer megkeresi az adatbázisban a kívánt adatot.



14. ábra: adat módosításának kereső felülete

Ha nincs találat, akkor egy „**Nem található adat a kiválasztott időben!**” hibaüzenettel tér vissza a rendszer. Ha pedig a kiválasztott adatok alapján talál adatot az adatbázisban, megjelenik a felületen egy módosító ablak (15. ábra). A rendszer manipulációjának kizárása miatt, mindig csak az aktuális év adatai módosíthatóak. Az aktuális év meghatározása a szerver idejének lekérdezésével történik.

### Módosítani kívánt adat keresése

Partner:

Szállítás dátuma:

---

### Módosítani kívánt adatok

Partner:

Szállítólevél száma:

Mennyiség:

Hordók száma:

Szállítás dátuma:

15. ábra: módosító felület

Ha létezett ilyen adat az adatbázisban, akkor a talált adatok alapján kitölti a módosító felületet ezzel megkönnyítve a moderátor munkáját. Így csak az elrontott adatot kell módosítani. Az adat törlésére is lehetősége van a moderátornak, amelyet a törlés gomb segítségével vihet végbe.

## 5.8 Regisztráció megvalósítása

Regisztrációt csak moderátor vagy adminisztrátor hajthat végre. Moderátor csak felhasználót regisztrálhat, míg az adminisztrátor bármely jogosultsági szintű felhasználót létrehozhat a rendszerben. Regisztráció folyamán 5 adat megadása szükséges (16. ábra). Minden adat kitöltése kötelező illetve a jelszó mezőknek egyezni kell. A jelszó tárolása az adatbázisban md5 kódolással történik.

```
$jelszo = md5($_POST["jelszo"]);
```

A felhasználónév ellenőrzésre kerül, hogy esetleg valaki nem e regisztrált már a megadott felhasználónévvel. Ha a rendszer talál az adatbázisban ugyan olyan felhasználónevet,

akkor egy „**Foglalt név!**” hibaüzenettel jelzi a már meglévőséget. A forráskód a reg.php fájlban található.



The image shows a registration form titled "Regisztráció" with a key icon in the top right corner. The form contains the following fields and elements:

- Felhasználónév\*:
- Jelszó\*:
- Jelszó újra\*:
- Üdvözlő üzenet:
- Jogosultság: Adminisztrátor (dropdown menu)
- Buttons: Regisztráció and Vissza

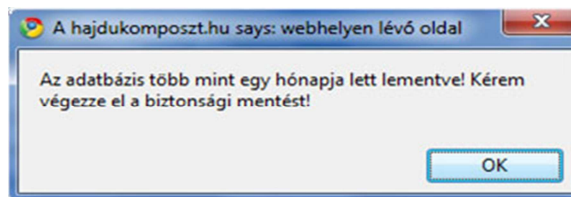
16. ábra: regisztrációs felület

## 5.9 Belépési statisztika megvalósítása

A belépési statisztika segítségével nyomon követhető a rendszerbe való belépések száma illetve megtekinthető, az utolsó belépés dátuma, amely tükrözi a rendszer kihasználtságát. Ezen adatok rögzítése belépéskor tárolódnak.

## 5.10 Adatbázis mentésének megvalósítása

Az adatbázis biztonsági mentése kihagyhatatlan feladat a rendszer alapvető felépítésében. A mentés elvégzése a moderátorok feladata. A rendszer havonta biztonsági mentést kér a moderátortól. Amíg a moderátor nem végzi el a biztonsági mentést, a rendszer nem enged egyéb tevékenység végrehajtását. A moderátorok belépését követően az aktuális dátumot összehasonlítja a save\_database\_datum táblázatban tárolt dátummal. Ha ennek a különbsége nagyobb, mint egy hónap, akkor figyelmezteti a moderátort, hogy az adatbázis több mint egy hónapja lett lementve és mentse le (17. ábra).



17. ábra: figyelmeztető ablak

Lementéskor az adatbázis egy sql kiterjesztésű fájlba történik és a mentés dátuma módosul a save\_database\_datum táblázatban. Illetve e-mailben értesíti az adminisztrátort, hogy ki, mikor mentette le a teljes adatbázist, így növelve az adatok biztonságát.

```
//Email értesítés
$site = "http://hajdukomposzt.hu";
$content = "Tisztelt Hajdú-Komposzt Bt. !";
$content .= "<br>\n<br>\n".$_SESSION['enter_user_name']." letöltötte a teljes adatbázist!";
$content .= "<br>\nletöltés ideje :".akt_datum()."";
$content .= "<br>\n<br>\nA " . $site . " csapata";
```

A fájl neve árulkodik a letöltés idejéről így könnyen megmondható a fájl lementésének ideje. A fájl struktúrája úgy van kialakítva, hogy egy egyszerű adatbázis importálással visszanyerhető a teljes adatbázis. Az adminisztrátor bármikor lementheti a teljes adatbázist. Az adatbázis mentését az save\_database.php fájl végzi.

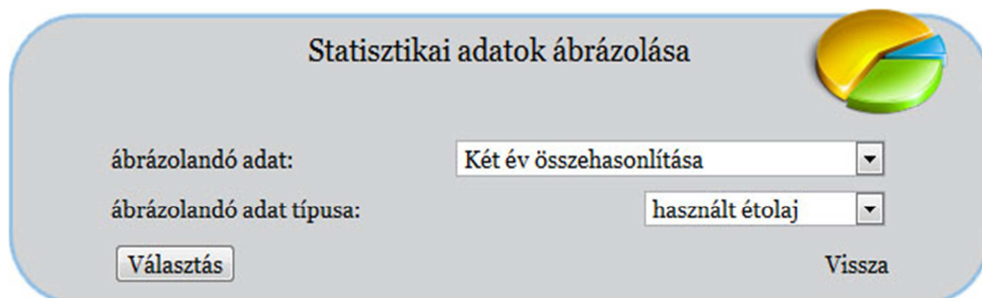
## 5.11 Statisztikai adatok ábrázolásának megvalósítása

Az adatok ábrázolása kéttípusú grafikon segítségével történik. Ez a funkció nagyon hasznos a vezetőség számára, mivel évek összehasonlítását és partnerek adatainak ábrázolását teszi lehetővé. Első lépésben a moderátornak ki kell választani az ábrázolandó adatot illetve, hogy melyik biogáz alapanyagot szeretné ábrázolni (18. ábra). Ezek után az ábrázolás módját, amely lehet vonal vagy oszlop diagram. Rajzolás során a függőleges tengelyen a hónapok, míg a vízszintes tengelyen pedig a súlyok helyezkednek el. Mivel a súlymennyiség dinamikusan változik, így a függőleges tengely beosztásának is dinamikusan kell változnia. Ezt úgy valósítottam meg, hogy lekérdeztem az év legmagasabb súlymennyiségét azt adott lekérdezésnél, és azt állítottam be a legmagasabb beosztásnak, majd ezt osztottam 20 egyenlő részre. A vízszintes tengely mindig 12 részre

kellett osztani a hónapok miatt. A rajzol gomb megnyomásával a digram.php átadja az kiválasztott adatokat a diagram\_rajz.php-nek, amely az adatok alapján elkészíti a diagramot.

```
<img usemap='#headerbanner'  
src='diagram_rajz.php?p_nev=$_POST[select_nev]&ev=$_POST[year]&abrazolas=$_PO  
ST[select_mod]'
```

Ábrázoláskor a diagram nem kerül tárolásra képfájlban így megelőzve direkt link segítségével a megnyitást.



Statisztikai adatok ábrázolása

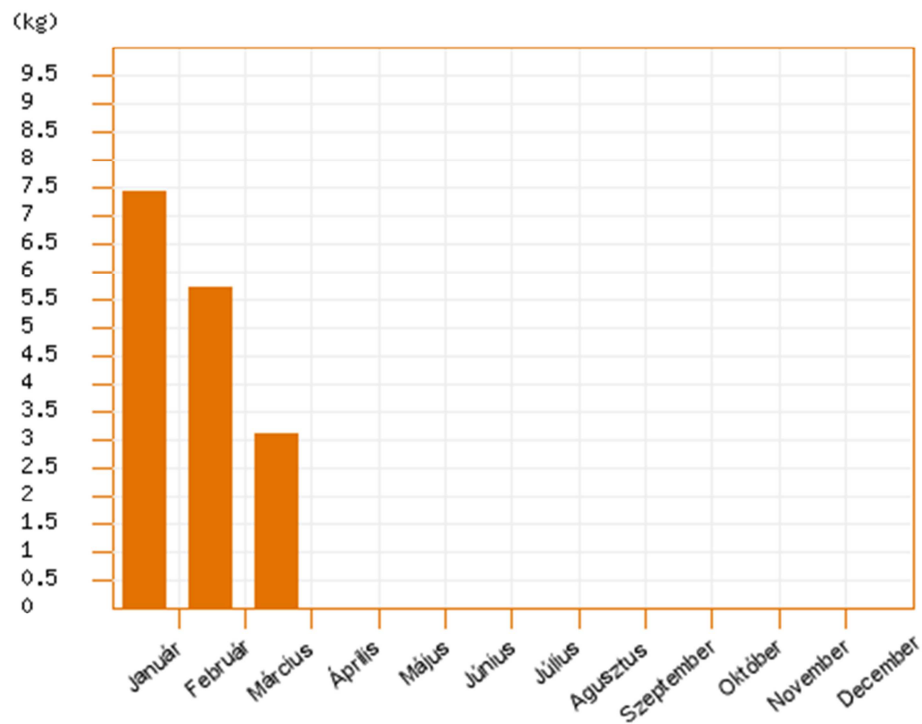
ábrázolandó adat: Két év összehasonlítása

ábrázolandó adat típusa: használt étolaj

Választás Vissza

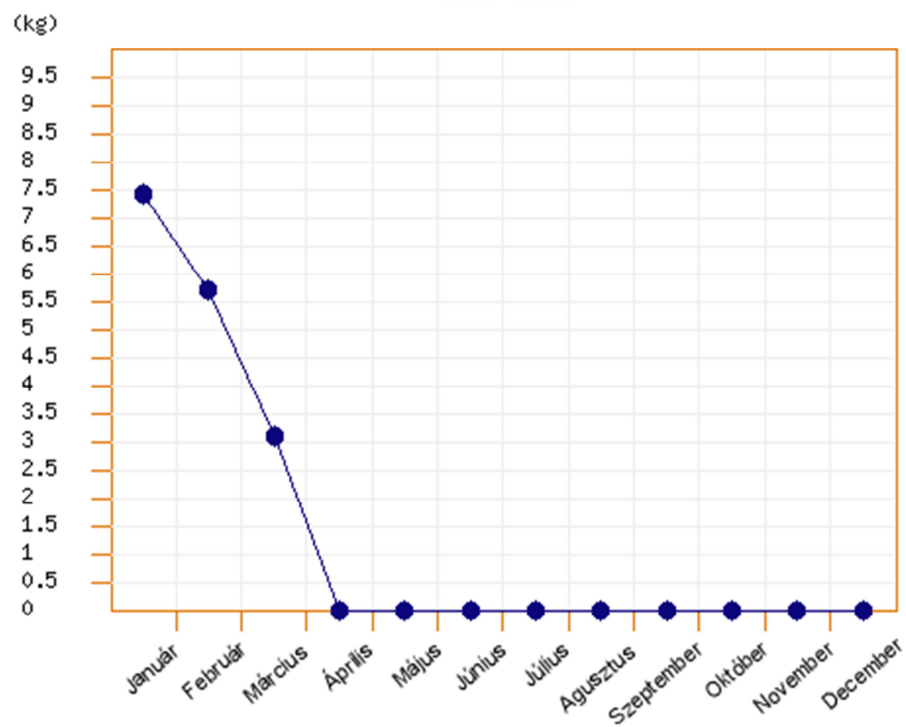
18. ábra: statisztikai ábrázolás felülete

Admirális Panzió leadott élelmiszerhulladékának havi lebontása  
2010 évben



19. ábra: statisztikai ábrázolás oszlop diagram segítségével

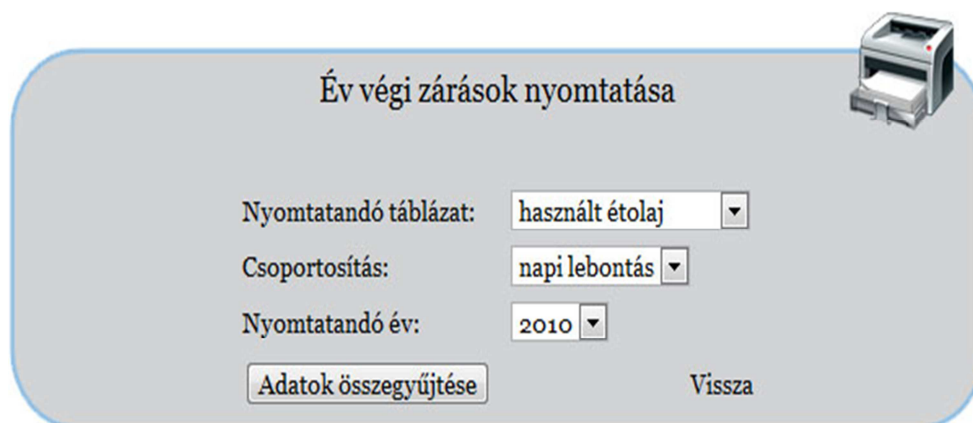
Admirális Panzió leadott élelmiszerhulladékának havi lebontása  
2010 évben



20. ábra: statisztikai ábrázolás vonal diagram segítségével

## 5.12 Év végi zárások nyomtatásának megvalósítása

Év végén az adatok zárásra kerülnek, amelyeket nyomtatott formában is tárolni kell. Így létrehoztam egy olyan funkciót, amely segítségével egyetlen gombnyomással az adatbázisban található összes adat nyomtatásra kerül. Lehetőség van napi vagy havi lebontás nyomtatására a már rögzített években (21. ábra)



Év végi zárások nyomtatása

Nyomtatandó táblázat:

Csoportosítás:

Nyomtatandó év:

Vissza

21. ábra: év végi zárások nyomtatásának felülete

## **6. Tesztelés**

A tesztelés nagyon fontos fázisa egy rendszer felépítésében. A rendszer tesztelése kb. 1 hónapon keresztül folytatódott, ami alatt folyamatos figyelmet igényelt, hogy megbizonyosodjak a rendszer helyes működéséről. A tesztelés során elég sok probléma előkerült, amelyre a tervezés során nem is gondoltam. Nagyon sok időt elvett a szakdolgozatom elkészítése folyamán a tesztelés alatt fellépő problémáknak korrigálása. Ilyen probléma volt például a partnerek mezőben nem helyesen rendelte az ID-kat a partnerekhez, amely az azonosítás folyamán galibákat okozott.

## **7. Köszönetnyilvánítás**

Köszönettel tartozom Dr. Szabó István tanszékvezetőnek, a szakdolgozat elkészítése folyamán, nyújtott támogatásáért, a folyamatos felügyeletért, valamint azért, hogy ötleteket adott nekem a sikeres szakdolgozat elkészítéséhez. Segítsége, irányítása és ötletei nélkül nem készülhetett volna el a dolgozatom.

## 8. Összefoglalás

Szakedolgozatom témája Egy Bioenergia alapanyagokat nyilvántartó rendszer fejlesztése, működésének megtervezése illetve megvalósítása volt. Azért választottam ezt a témakört, mert a mai világban egyre nagyobb energiaszükségletre van igénye az emberiségnek, és úgy gondoltam, hogy így egy kicsit hozzájárulhatok az alternatív energiahordozók nyilvántartásának könnyebb felhasználhatóságában. Munkám eredményeként létrejött egy olyan alkalmazás, melynek segítségével egyszerűbbé és hatékonyabbá, illetve átláthatóbbá tette a cég bioenergia alapanyagok nyilvántartását.

Az alkalmazással kapcsolatban felmerültek további igények, amelyek későbbiekben további célkitűzéseim lesznek (pl. felhasználók statisztikai ábrázolásának elkészítése, profil szerkesztése, jelszó módosítása).

Hátrányként említeném meg, hogy nem minden böngésző jeleníti meg tökéletesen az alkalmazást. Próbáltam a legáltalánosabb böngészőkre fejleszteni az szoftvert, így hatékonytalan maradt néhány népszerűtlen böngésző.

## 9. Irodalomjegyzék

- [1] Magyar Biogáz Kft.: [www.biogaskft.hu](http://www.biogaskft.hu)
- [2] Alternatív energia: [www.alternativenergia.hu](http://www.alternativenergia.hu)
- [3] CMD adatmodell: <http://alka.web.elte.hu/3resz.pdf>
- [4] Biogáz üzem: <http://www.batorcoop.hu/?q=node/3>
- [5] JavaScript fogalma: <http://weblabor.hu/cikkek/oojsafelszinfolt>
- [6] Fábrián Zoltán, PHP kezdőknek:  
[http://fzolee.hu/download/download.php?fname=./PHP\\_programoz%E1s\\_kezdoknek.pdf](http://fzolee.hu/download/download.php?fname=./PHP_programoz%E1s_kezdoknek.pdf)
- [7] PHP: Documentation: <http://php.net/docs.php>
- [8] Matt Zandstra: Tanuljuk meg a PHP4 használatát 24 óra alatt. Kiskapu kiadó 2002.