

TDK-dolgozat

Szerző: Fibi Ladislav

SELYE JÁNOS EGYETEM

Tanárképző Kar
Informatika Tanszék

A dolgozat címe: On-line szótár informatikából

Témavezető: Prof. Ing. Stoffová Veronika, CSc.
egyetemi tanár

Szerző: Bc. Fibi Ladislav
szakpár: Informatika-Történelem
levelező képzés

Komárom 2012

Becsületbeli nyilatkozat

Alulírott Bc. Fibi László aláírásommal igazolom, hogy a munkámat önállóan, a felhasznált irodalomban feltüntetett forrásokra támaszkodva és egyéb források felhasználása nélkül dolgoztam ki.

Absztrakt

Munkánk célja egy olyan háromnyelvű szótár létrehozása volt, amely a szavak egyik nyelvről a másikra történő átfordításán túl lehetővé teszi bővebb információk, leírások tárolását is mindhárom nyelven. Fontos elvárás volt, hogy a szótár egyazon időben több felhasználó munkáját is lehetővé tegye. Annak elérése érdekében, hogy a szótár rövid idő alatt sok szót tartalmazzon, úgy terveztük az alkalmazást, hogy új bejegyzés létrehozására minden, tehát nem regisztrált és nem bejelentkezett felhasználónak is lehetősége legyen. Hogy a szavak között mégse szerepelhessenek nyelvtanilag, jelentés tanilag vagy logikailag helytelen bejegyzések, minden új szó csak elbírálás és jóváhagyás után kerülhet be a szótár szavai közé. A bírálatot egy regisztrált felhasználó végzi. Alkalmazásunk létrehozásához a webszerver, PHP szkriptnyelv és a MySQL adatbázis kezelő rendszer hármását választottuk. A szakszótár webböngészőben történő megjelenésének befolyásolására a kaszkádolható stíluslapokat használtuk. A hosszú leírásokat XML fájlokban tároljuk.

Munkánk bevezetőjében megkíséreltük megvilágítani, miért fontos egy ilyen többnyelvű szakszótár létrehozása és használata, különösen a mi körülményeink között. Az első fejezetben felvázoltunk egyet a probléma lehetséges megoldásai közül, és leírtuk azokat az eszközöket amelyeket felhasználtunk az alkalmazás megvalósítása során. Rávilágítottunk arra is, milyen előnyökkel jár ezek használata.

A második fejezet az alkalmazás leírását, az adatstruktúrákat és a működést tartalmazza azokkal a problémákkal együtt, amelyekkel szemben találtuk magunkat a fejlesztés során.

Kulcsszavak:

szótár, online alkalmazás, webszerver, honlap, HTML, PHP, XML, kaszkádolható stíluslap, adatbázis, felhasználó

Tartalomjegyzék

Ábrajegyzék.....	9
Táblázatjegyzék	10
Rövidítések, jelölések jegyzéke	11
A szakkifejezések jegyzéke.....	13
Bevezetés	14
1. Online alkalmazások.....	17
1.1. A webserverver	17
1.2. A PHP	18
1.3. Az adatbázis kezelő rendszer.....	18
1.4. Az XML.....	19
1.5. Stíluslapok használata	19
1.6. Minden egyben	20
1.7. Javascript	20
2. Online többnyelvű, bővíthető szótár.....	21
2.1. Felhasznált fejlesztői környezet	21
2.2. Konstansok	22
2.3. Változók	23
2.4. Adatok és adatstruktúrák	24
2.4.1 A MySQL adatbázisban tárolt adatok.....	24
2.4.2 Az XML fájlokban tárolt adatok.....	28
2.4.3 A képfájlok tárolása	28
2.5. Az online alkalmazás megvalósítása	29
2.5.1 A kezdőlap	29
2.5.2 Új szó bevitele a szótárba	33
2.5.3 Összes szó listázása.....	35
2.5.4 Bejelentkezés	36
2.5.5 A szavak helyességének elbírálása	38
2.5.6 Új kategória definiálása, és a kategóriák módosítása	40
2.5.7 Felhasználók kezelése	42
2.5.8 Felhasználók követése	44
2.6. Az alkalmazás felhasználhatóságának lehetőségei.....	44
Összefoglalás.....	47

Felhasznált irodalom jegyzéke.....	49
Melléklet	50

Ábrajegyzék

Ábra 1 Az adatbázis relációi közti kapcsolat.....	27
Ábra 2 A kezdőlap	29
Ábra 3 Részleteket megjelenítő oldal	33
Ábra 4 Új szó bevitelére szolgáló lap	34
Ábra 5 Az összes szót listázó oldal.....	36
Ábra 6 Bejelentkezési felület.....	37
Ábra 7 Értesítés a sikertelen bejelentkezésről	37
Ábra 8 Megváltozott menü	38
Ábra 9 Bírálat lap.....	39
Ábra 10 Szó javítását lehetővé tevő oldal.....	40
Ábra 11 Kategória javítására szolgáló oldal	41
Ábra 12 Felhasználók kezelését biztosító felület.....	43
Ábra 13 Felhasználó adatainak megváltoztatását lehetővé tevő felület	43

Táblázatjegyzék

Tab. 1 A szavak reláció struktúrája	24
Tab. 2 A biralando reláció struktúrája	25
Tab. 3 A kategória reláció struktúrája	25
Tab. 4 A felhasználók reláció struktúrája	26
Tab. 5 A csoport reláció struktúrája.....	27

Rövidítések, jelölések jegyzéke

ASP (Active Server Pages) – a Microsoft Visual Basic alapú webes programozási nyelve.

BMP (bitmap) – bittérkép, a legrégebbi raszteres grafikus fájlformátum

CSS (Cascading Style Sheets) – egymásba ágyazható stíluslapok

DBMS (DataBase Management System) – adatbázis kezelő rendszer

GIF (graphics interchange format) – veszteséges grafikus fájlformátum, raszteres képek és animációk 256 színben történő tárolására

HTML (HyperText Markup Language) – hiperszöveges jelölőnyelv

http (HyperText Transfer Protokoll) – hiperszöveg-átviteli protokoll, alapértelmezés szerint a 80-as porton keresztül. A webböngésző és a webszerver közötti kétirányú kommunikáció nyelve.

https (Secured HyperText Transfer Protokoll) – biztonságos hiperszöveg-átviteli protokoll, alapértelmezés szerint a 8080-as porton keresztül.

IIS (Internet Information Services) – a Microsoft Windows operációs rendszeren futó internet alapú szolgáltatások.

JPG (Joint Photographic Experts Group) – veszteségesen tömörített grafikus fájlformátum

LAN (Local Area Network) – helyi számítógépes hálózat

PHP (Hypertext preprocessor) – hiperszöveges előfeldolgozó, a tools for Personal Home Page rövidítése

PNG (portable network graphics) – veszteségmentesen tömörített raszteres grafikus fájlformátum. Eredetileg a GIF fájlformátum leváltására szántak.

PWS (Microsoft Personal Web Server) – személyes webkiszolgáló.

SQL – (structured query language) struktúrált lekérdező nyelv

W3C – (World Wide Web Consortium) feladata a webes szabványok és iránymutatások kidolgozása¹

XML – (eXtensive Markup Language) kiterjeszthető jelölőnyelv

¹ World Wide Web Consortium. [online]. [2011.2.20]. <<http://www.w3c.org>>.

Dőlt betűvel jeleztük a fájlok neveit vagy azok egy részét, a forráskódban előforduló változók és konstansok neveit, az adatbázis struktúrájában előforduló megnevezéseket, valamint a munkában előforduló programkódokat és kódrészleteket.

A szakkifejezések jegyzéke

Adatbázis (Database) – valamilyen rendszer szerint tárolt adatok halmaza.

Adatbázis kezelő rendszer (database management system) – az adatbázisok létrehozását, tárolását és kezelését lehetővé tevő programrendszer, amely szabályozza az adatokhoz való hozzáférést és gondoskodik az adatbázis belső karbantartásáról is.

cross-platform – különböző platformok között szabadon átjárható, átvihető.

HTML-embedding – HTML kódba ágyazás.

PHP interpreter – PHP értelmező.

Session – munkamenet, kommunikáció két számítógép között, miközben a számítógépek átmenetileg adatokat tárolnak egymásról.

Reláció – táblázatos formában tárolt adatok, ahol a logikailag egymáshoz tartozó adatok a táblázat soraiban (record), a táblázat oszlopaiban (attribútum) pedig az azonos mennyiségekre vonatkozó adatok vannak tárolva.

Szerver – kiszolgáló számítógép vagy egy kiszolgáló számítógépen futó szolgáltatás.

tag – a HTML nyelvben használt speciális karakterek között elhelyezkedő, előre definiált jelentéssel bíró címkék angol megnevezése.

webböngésző – webes információ megjelenítésére alkalmas szoftver.

Bevezetés

Az 17. században, Angliában meginduló, majd a 19. század második felében az egész világra kiterjedő ipari forradalom soha nem látott változásokat idézett elő az ember mindennapi életében. Széles körben ismertek ezeknek a változásoknak a társadalmi, politikai, szociális és demográfiai hatásai. Kevesebbet hallani az iparosodás modern nyelvekre gyakorolt hatásáról. Pedig a 18-19. századi nyelvújító mozgalmak célja nemcsak az egyes nyelvek irodalmivá és egységessé tétele volt. Megjelent bennük az a törekvés, hogy a modern nyelvet az iparosodás és a tudományos vívmányok által generált, egyre erősödő kihívásoknak megfeleltessék. Teljesen új tárgyak, fogalmak és eszmék jelentek meg, amelyeket sok esetben nem lehetett a nevükön nevezni, tekintve, hogy nem volt megnevezésük az adott nyelv szókincsében. Új szakkifejezések, megnevezések bevezetésére került sor, melyek pontosan meghatározták az adott tárgyat, fogalmat. Szükség volt erre az új iparos réteg kialakítása és az újonnan kialakuló tudományok ismerőinek oktatása miatt is. Nélkülük nem lehetett modernizálni az országot és a társadalmat, vagyis az egyre gyorsuló fejlődés elengedhetetlen feltételei voltak. A tudományok fejlődésével a nyelvbe kerülő terminusok vagy szakkifejezések tudományos területek szerint csoportokba rendeződtek. Ha a laikus közönség nem is minden esetben, de az adott tudományág művelői ismerték a saját területük szakkifejezéseit. Folyamatosan kialakult tehát az egyes tudományágak szakszókészlete, szakterminológiája.

A terminológia szó jelentése az *Idegen szavak és kifejezések szótára* szerint “valamely szakterület gyakran használt, a szakma művelőitől általában ismert és a köznyelvhez is közel álló szavainak összessége”.² A meghatározás pontos, tömör és egyértelmű. Gyorsuló világunknak köszönhetően azonban a gyakorlatban használatosak olyan kifejezések, amelyek értelmezése nem mindig és nem mindenki számára azonosítja egyértelműen ugyanazt az objektumot, fogalmat vagy történést. A másik gyakori jelenség, amikor egy és ugyanazt a dolgot mindig újabb és újabb megnevezésekkel illetjük. Az olyan gyorsan fejlődő ipar- és tudományágakban, mint az informatika, kibernetika és telekommunikáció gyakori az idegen, elsősorban angol eredetű kifejezések átvétele, valamint a mozaikszavak előfordulása a nyelvben. Szlovákiai magyar létünkéből adódóan egy további sajátos kihívásnak is meg kell felelnünk. A szlovák nyelvi környezet és a szlovák nyelv gyakori használatának eredményeként előfordul, hogy alkalmazott szókincsünkbe nem az egyetemes magyar terminológiának megfelelő szavak kerülnek be,

² BAKOS F. *Idegen szavak és kifejezések szótára*. Budapest: Akadémiai Kiadó, 2007. ISBN 9789630578752.

hanem a szlovák szakkifejezések tükörfordításából keletkező, a magyar terminológiától teljesen idegen terminusok. Ezek alkalmazása félreértésekre adhat okot, nem beszélve arról, hogy a szókincsben való meggyökeresedésük háttérbe szorítja a terminológiai helyes kifejezéseket.

Az, hogy többnyelvű közegben élünk, több nyelvet beszélünk, lehetővé teszi számunkra az információszerzést különböző nyelveken íródott szakirodalmi forrásokból. Sok esetben ez különböző nézőpontot, más-más hangsúlyozást is jelenthet. Ezt a vitathatatlan előnyt azonban csak akkor tudjuk hatékonyan érvényesíteni, ha birtokában vagyunk a helyes szakszókincsnek, mégpedig mindegyik általunk ismert nyelven. Kézenfekvő tehát az az igény, hogy kezdeményezzük egy olyan szakszótár kialakítását, amely három nyelven kínál minőségi információt az érdeklődő számára. Az egyik nyelv az angol, mint a technika és az üzleti világ nyelve, és nem mellesleg gyakran a terminusok forrásnyelve. A másik a mi anyanyelvünk, a magyar, továbbá a szlovák, vagyis az állam hivatalos nyelve. Az az elképzelés, miszerint egy ilyen szótár mindenki által elérhető és feltölthető legyen, lehetőséget kínál nemcsak a széles körben történő használatra, de nagy mennyiségű adat felhalmozására is. A tömeges és névtelen hozzáférés azonban magában hordozza azt a veszélyt, hogy az információk minőségileg igen eltérőek lehetnek. Ez pedig a szótár használhatóságát kérdőjelezi meg. Szükséges egy olyan ellenőrző mechanizmus beépítése, amely a bejövő információk mindegyikét csak egy alapos szűrést követően engedi a felhasználható adatok közé jutni. Ezt a szűrést nyelvészek és terminológusok vagy egy belőlük felálló munkacsoport végezné. Ezzel a módszerrel, mintegy önszerveződésszerűen hozható létre egy olyan folyamatosan bővülő hasznos adatbázis, amely segédeszközülni szolgálhat mindannyiunk számára.

Bár eddig csak egy szótárról beszéltünk, az elképzelés egy egyszerű szótárnál többet rejt magában. Szakkifejezések, fogalmak rendszerezett tárolásáról van szó, és egy ilyen alkalmazásnak többet kell nyújtania annál, mint szavaknak az egyik nyelvről a másikra, esetleg harmadikra való lefordítása. Szükség van a fogalmak meghatározásainak tárolása is. Így olyan többnyelvű fogalomtárat hozhatunk létre, amely a tudományok különböző diszciplínáiban egyaránt hatékonyan használható.

A hatékony, szakszerű és következetes ellenőrzés mellett létrejövő és folyamatosan bővülő adatbank létrehozásához szükség van egy olyan keret megalkotásához, amely biztosítja az adatok tárolását és integritását, a felhasználók nyomon követését és felhasználói felületet biztosít. Mindezt úgy kell megoldani, hogy egyszerre többen tudjanak

dolgozni a rendszerrel akár különböző helyeken, nagyobb földrajzi távolságokban elhelyezkedő számítógépekről. Munkánk célja egy ilyen alkalmazás elkészítése volt.

1. Online alkalmazások

Olyan rendszer megalkotásához, amely lehetővé teszi több felhasználó egyidejű munkáját egyazon adatok halmazával, szükség van egy központi, kiszolgáló számítógépre, és egy olyan átviteli médiumra, amelyen keresztül az adatforgalom bonyolódik a kiszolgáló és a felhasználó számítógépe között. Az adatok gyors átvitelére elsősorban a számítógépes hálózatok szolgálnak. Ha az adatcserét egy intézményen belül kívánjuk megoldani, adatátviteli médiumként szolgálhat a lokális hálózat (LAN). Ha a felhasználók köre szélesebb, és a földrajzi elhelyezkedésük sem behatárolható, akkor az internetet használhatjuk, mint hatékony adatátviteli médiumot.

Az interneten vagy hálózaton keresztül hozzáférhető alkalmazások lehetséges módozatai közül a webes alkalmazások vagy weboldalak a leginkább elterjedtebbek. Előnyük, hogy a kliens oldali számítógépeken nem igényelnek szinte semmilyen beavatkozást vagy beállítást. Csupán internetkapcsolatra és egy webböngészőre van szükség. A szerveren természetesen el kell végezni a szükséges telepítéseket és beállításokat ahhoz, hogy egy működő webszervert működtethessünk.

1.1. A webszerver

Weboldalak és webes alkalmazások működtetésének elengedhetetlen kelléke a webszerver. A webszerver egy számítógépen futó szolgáltatás, vagyis egy szoftver. Alapvető feladata webes dokumentumok szolgáltatása http és https protokollokon keresztül a hozzá csatlakozó kliensek webböngészői számára. További feladatai lehetnek a felhasználók nyomon követése, a hibaüzenetek naplózása, a hozzáférések korlátozása.

Weboldalak és webes applikációk készítésére a HTML az elsődlegesen használt nyelv, amely az egyszerű szöveget speciális, a HTML-nyelvre jellemző címkékkel fogja közre. A böngésző a címkék és azok attribútumai alapján dönti el, mit kell tennie a szöveggel, hogyan jelenjenek meg a weboldal egyes elemei, milyen legyen köztük a kapcsolat. Azonban a HTML dokumentumok nem képesek másra, mint a dokumentum statikus megjelenítésére. A világhálóra kerülő weboldalak, különösen az online alkalmazások készítéséhez ez kevés. Szükség van a dinamikus megjelenésre, az interaktív működésre, a különböző forrásokból származó adatok megjelenítésére és a velük való manipulálásra. Mivel a HTML ezen igényeket képtelen kielégíteni, további eszközök használatához kell folyamodnunk.

1.2. A PHP

A PHP a dinamikus webes tartalom gyors összeállítására szolgáló nyelv. Képességei közt szerepel a magas szintű programozási nyelvekre jellemző programozási technikák - mint a változók, tömbök, feltételes elágazások, ciklusok, objektumok - használatának lehetősége, állományok és könyvtárak kezelése. Továbbá biztosítja az adatok kezelését adatbázisokból vagy XML állományokból. Szabadon és ingyen hozzáférhető. A PHP interpretált nyelv, ami annyit jelent, hogy a programozó által megírt forráskód csak a meghíváskor, tehát a program indítását közvetlenül megelőzően kerül lefordításra a webserveren futó PHP értelmező által. A program az elindításáig, majd a futást követően is csupán forráskódban van tárolva. Ez a megoldás lassúbb indítást eredményez, mint a kompilált nyelvek esetében, viszont vitathatatlan előnye, hogy az alkalmazások különböző platformokra átvihetők.

1.3. Az adatbázis kezelő rendszer

Az adatbázis kezelő rendszereket (DBMS – DataBase Management System) kifejezetten adatok tárolására és kezelésére tervezték. Az adatbázis tulajdonképpen adatok hatékonyan szervezett együttesét jelenti. Arról, hogy az adatok mindig hatékonyan legyenek szervezve, az adatbázis kezelő rendszerek gondoskodnak, átláthatóan reprezentálják az adatok struktúráját, és jól kezelhetők. Az adatokat sorokból és oszlopokból felépülő adatbázis-táblákba, más szóval relációkba szervezik. A reláció egyes sorai, más szóval rekordok, az egymással logikai kapcsolatban lévő adatelemeket tartalmazzák. Az oszlopok, vagy attribútumok olyan adatmezőket tartalmaznak, amelyek az egyes rekordokban ugyanazt a jelentést hordozzák, ezért az attribútumok előre meghatározott típusúak. Ez azt is jelenti, hogy a reláció minden egyes sorának azonos struktúrája van.

Az adatbázis több táblából állhat, amelyek egymástól függetlenek. Kapcsolatot köztük az egyes attribútumokon keresztül létesíthetünk. Ezeket a többi relációhoz való kapcsolódásra használt attribútumokat nevezzük kulcsoknak. A kulcs a kapcsolt relációk egyikének egyetlen sorát azonosítja. Az egymáshoz kapcsolt relációk közti integritásról az adatbázis kezelő rendszer hivatott gondoskodni.

A relációs adatbázisok kezelésére használható nyelvek között az SQL vált általánosan elfogadott szabvánnyá. Az SQL nem használ algoritmus szerkezeteket, ahogy fájlkezelésre, vagy képernyőkezelésre vonatkozó utasításokat sem. Kizárólag adatkezelésre szolgál. Halmazorientált nyelv, vagyis a feladatot a relációk kiválasztott sorain hajtja végre.³ Alkalmazások készítése esetén az SQL utasítások valamely 3. generációs nyelvbe ágyazva alkalmazhatók.

1.4. Az XML

Az XML (eXtensive Markup Language) kiterjeszthető jelölőnyelvet eredetileg azért tervezték, hogy az ember által is olvasható lehetőséget teremtsen valamilyen struktúrába szervezett adatok átviteléhez egyik alkalmazásból egy másikba. Kialakítása emellett alkalmassá tette adatok tárolására is, így csakhamar mint tárolási formátum is elterjedté vált. Az XML dokumentumok olvasásakor formai szempontból feltűnő a hasonlóság a HTML dokumentumokkal. Mindkettő címkékkel zárja körül az adatot, azonban míg a HTML rögzített, előre meghatározott címkekészlettel és azok rögzített attribútumaival dolgozik, az XML lehetőséget biztosít egyéni címkekészlet és hozzá tartozó attribútumok használatára. Tulajdonképpen egy saját jelölőnyelv kialakítását teszi lehetővé a fejlesztő számára.

1.5. Stíluslapok használata

A HTML dokumentumok megjelenése a böngésző-ablakban, az egyes elemeket reprezentáló címkék attribútumaiként megadható tulajdonságoknak a beállításával befolyásolható. Így a forráskód jelentős része a megjelenésről és nem a tartalomról szól, a kód jelentősen megnő, bonyolulttá és nehezen olvashatóvá válik.

Stíluslapok használatával elegendő egyszer definiálni a megjelenést valamely HTML elem részére. Ezáltal egyszerűsödik a programkód és kisebb lesz a forráskódot tartalmazó fájl mérete. További előnye a stíluslapok használatának, hogy segítségükkel nemcsak az egyes elemekre hivatkozhatunk, hanem az elemekkel kapcsolatos eseményekre is. Így például, egyszerűen beállíthatjuk, hogy egy nyomógomb megjelenése megváltozzon, ha az egérmutató fölé kerül.

³ [online]. [2011.1.18]. <<http://ikon.inf.elte.hu/~cserep/index.php?fm=1>>.

1.6. Minden egyben

Mint láttuk, egy online alkalmazás fejlesztésének és működtetésének egyik hatékony és igen közkedvelt fajtája valamely webservert, ez leggyakrabban az Apache, a PHP programozási nyelv és valamelyik SQL alapú adatbázis kezelő rendszer - általában a MySQL - együttes használata. Mindhárom összetevő szabadon elérhető a világhálón. A telepítésük nem különösen összetett, a konfigurációjuk viszont okozhat meglepetéseket. Gyakran nem is az egyes komponensek beállításával adódhat probléma, hanem a három összetevő egymással való problémamentes működését nehéz elérni. Ez volt az oka a három összetevőt egy csomagba integráló komplex kis webszerverek megjelenésének. Lényegük az, hogy tartalmazzanak egy Apache webservert, PHP-t és egy MySQL adatbázis kezelő rendszert, mindegyiket minimális konfigurációban. Az egyszerű telepítést követően a szerver egy optimális konfigurációval működik. Kisebb szerverek esetén eredményesen használhatóak.

1.7. Javascript

A Javascript kliens oldali parancsnyelv a felhasználó böngészőjében fut. Mivel a HTML, mint egyszerű szöveges leírónyelv nem tud döntéseket hozni, szükségünk lehet egy parancsnyelv használatára a kliens oldalon is. A Javascript elsősorban a felhasználó által generált események kezelésére, a kliens számítógépének specifikumait figyelembe vevő módosítások elvégzésére használatos. Használatával aktív kód jut a kliens számítógépére, ami biztonsági kockázatot jelent. Mivel az ebben rejlő lehetőségeket sokan kihasználták, az utóbbi években kriminalizálódott. A modern böngészők a használatát erősen korlátozzák.

2. Online többnyelvű, bővíthető szótár

Feladatunk egy olyan alkalmazás elkészítése volt, amelyik adatokkal dolgozik, miközben több felhasználó különböző jogosultságok birtokában egy időben használja. Nagyon leegyszerűsítve két problémát kellett megoldani, az adatok rendszerezett tárolását és a felhasználók nyomon követését. A továbbiakban ismertetjük azokat az eszközöket, amelyek segítségével dolgoztunk, majd pedig azt, hogy milyen megoldásokat használtunk a cél elérése érdekében.

2.1. Felhasznált fejlesztői környezet

Felhasznált hardver eszköz:

- Dell Latitude D630 notebook, CPU Intel Core 2 Duo T71000 1,8 GHz, RAM 3 GB, HDD Toshiba MK6037GSX 250 GB, DVD RW DL, display 14,1” Mobile Intel 965 Express 1280x800, audio Sigmatel High Definition, lan Broadcom NetXtreme Gigabit Ethernet, wifi Dell WLAN minicard, bluetooth, FireWire, 4x USB, COM1, VGA out, S-video out

Igénybevett szoftver:

- Windows XP Professional SP 3 (Build 5.1.2600), angol
- Apache HTTP Server 1.3.29 Released
- PHP 5.2.1
- MySQL 5.1.38
- phpMyAdmin 2.3.3pl1
- XAMPP Lite (Basic Package) version 1.7.3
- Notepad++ v.5.5.1 Portable version, angol
- Paint.NET v3.36 (Final Release build 3.36.3158.38068), angol
- Firefox v.4.0b2 Portable

A hardver konfigurációjának leírásából látható, hogy az alkalmazás fejlesztése és tesztelése egy teljesen hétköznapi számítógépen folyt. Az egyszerűbb átvihetőség érdekében került használatra az XAMPP webes programcsomag alapverziója, amely magában foglalja a következőket: Apache 2.2.14, MySQL 5.1.41, PHP 5.3.1, Minipearl

5.10.1, OpenSSL 0.9.8l, phpMyAdmin 3.2.4. Segítségével elkerülhető a bonyolult telepítés és konfiguráció.

A Notepad++ a HTML, PHP, Javascript és CSS kód szerkesztésére használható hatékony eszköz. Felismeri a kód utasításait, színekkel jelzi a program struktúráját, így figyelmeztet az esetleges szintaktikai hibákra. Továbbá számozza a sorokat, ami figyelembe véve, hogy az esetleges hibajelentések a kód soraira hivatkoznak, elengedhetetlen feltétele a hatékony munkának.

A Paint.NET grafikai szerkesztő lehetőségeit az alkalmazásban előforduló grafikai elemek megalkotásánál vettük igénybe. Igyekeztünk a grafikus elemeket úgy megalkotni, hogy azok a lehető legkisebb tárterületen legyenek tárolhatók. Ez nemcsak azért fontos, hogy spóroljunk a háttértárral, hanem elsősorban azért, hogy csökkentsük ezen fájlokhoz az átviteli idejét a szervertől a kliensig.

A phpMyAdmin interaktív SQL parancsértelmező segítségével grafikus felületen, lényegesen kényelmesebb formában dolgozhattunk az adatbázisokkal, mint ahogy azt a MySQL adatbázis kezelő lehetőségeit igénybe véve megtehettük volna.

Tekintettel arra, hogy a felhasználók különböző böngészőket használnak, és ezek a böngészők nem minden esetben igazodnak maximálisan a W3C által előírt normákhoz, a tesztelés során a Firefox 4.0 verzióját és a Windows XP operációs rendszer beépített böngészőjének, az Internet Explorernek 8.0 verzióját egyaránt használtuk. Bizonyos esetekben a böngészők más-más formában jelenítik meg az oldal egyes elemeit, sőt előfordult, hogy ugyanaz a kód eltérő működést eredményezett. Célunk volt egy olyan optimum elérése, amely a két leggyakoribb böngészőben egyaránt használható kimenetet eredményez.

2.2 Konstansok

A konstansok csak skaláris értékeket tárolhatnak, tömböket vagy objektumokat nem. A PHP programokban bárhol használhatóak, tekintet nélkül a változók hatáskörére.⁴ Az alkalmazás fejlesztésekor a három nyelv megnevezésére a *lang0*, *lang1*, *lang2* konstansokat használtuk. A forráskódban mindig ezekkel a konstansokkal dolgoztunk. A konstansok tényleges értéke a *lang.inc* fájlban került definiálásra. Ezzel a módszerrel egy helyen elég módosítani a nyelvek definícióját, és az a teljes alkalmazásban érvényre jut. Lehetőség nyílik így a szótárnak más nyelveken való felhasználására.

⁴ Mercer-Kent-Nowicki-Mercer-Squier-Chot: PHP5, Panem KFT, Budapest, 2006, 81.o.

A gyakran használt állandóknak egy következő csoportja az, amelyik az adatbázishoz történő kapcsolódáskor és a velük való munkához szükséges. Ebben a csoportban négy konstans definiáltunk. A *DBHOST* az adatbázis kiszolgáló nevét vagy IP címét és annak a portnak a számát tartalmazza, amelyiken keresztül elérhető. A *DBUSER* és *DBPASS* konstansok az adatbázis szerverre történő bejelentkezéshez szükséges felhasználói azonosítót és a hozzá tartozó jelszót tartalmazzák. A *DBNAME* az alkalmazás által használt adatbázis nevét hordozza.

A konstansok harmadik csoportjába azok tartoznak, amelyek a fájlkezeléssel kapcsolatosak. A *XmlUploadDir* és a *PixUploadDir* konstansok az xml és a képfájlok elhelyezésére szolgáló könyvtárak neveit takarják. A *MaxFileSize* állandó, a képfájl maximális méretét határozza meg.

Az adatbázisokkal és a fájlkezeléssel kapcsolatos konstansok a *config.php* fájlban kerültek definiálásra. A php állományban történt elhelyezésnek elsősorban biztonsági okai vannak. A php fájlok a külső felhasználók számára elérhetetlenek, soha nem töltődnek le a kliens számítógépre, így a konstansok és értékeik láthatatlanok maradnak.

2.3 Változók

A PHP gyengén típusos nyelv. Ez azt jelenti, hogy a változókat nem szükséges sem deklarálni, sem inicializálni, sőt adattípusokba sem kell sorolni. Ennek ellenére a változók sok meglepetést okozhatnak a program fejlesztése során. Ez elsősorban a változók hatáskörével függ össze, vagyis azzal, hogy a változó és értéke a program melyik pontjáról érhető el. Általában elmondható, hogy a függvényben definiált változó értéke csak az adott függvényből érhető el, a függvényen kívülről nem. Ha egy függvény a futását befejezte, akkor a változó is megsemmisül, és az értéke sem érhető el. Erre kínál megoldást a *global* kulcsszó, melynek használatával kiterjeszthető a változók hatásköre. Azonban már a PHP 4.0 verziójától a *php.ini* fájlban - ami a php konfigurációs állománya - található *register_globals* változó kikapcsolt állapotban van, elsősorban biztonsági okokból.⁵ Be lehet kapcsolni, de nem ez az ajánlott opció. Mivel előre nem tudhatjuk milyen webszerveren fut majd az alkalmazás, érdemes a *register_globals = Off* beállítással számolni. Ebben az esetben az űrlapváltozókat nem használhatjuk globális változóként.

⁵ MERCEL-KENT-NOWICKI-MERCEL-SQUIER-CHOT: *PHP5*, Panem KFT, Budapest, 2006, 1001.o., ISBN 80-86815-10-2

Egyetlen lehetőségünk a POST, GET és SESSION változók használata marad, így az alkalmazásban átadott űrlapváltozókat mi is a fent említett metódusok segítségével adtuk tovább.

2.4 Adatok és adatstruktúrák

Az adatok tárolására és a velük történő manipulálásra a MySQL adatbázist használtuk. Az adatok többsége a *szotar* nevű adatbázisban került tárolásra. Kivételt képeznek a hosszú, leíró adatok, amelyek hossza meghaladhatja a 255 karaktert. Ezek xml állományokban vannak tárolva.

2.4.1 A MySQL adatbázisban tárolt adatok

A *szotar* adatbázis öt relációt tartalmaz. Az első reláció megnevezése *szavak* és a már elbírált szavak adatait tartalmazza. A struktúrája a Tab. 1 táblázatban látható.

Tab. 1 A szavak reláció struktúrája

Az attribútum neve	Típus	Hossz (karakter)	Nulla érték megengedett	Automatikus értéknövelés
id	int	11	nem	igen
lang0	varchar	60	igen	nem
lang1	varchar	60	igen	nem
lang2	varchar	60	igen	nem
kategoria	varchar	20	igen	nem
megjegyzes	varchar	50	igen	nem
kep	varchar	50	igen	nem
biralta	int	3	igen	nem

Az *id* attribútum az egyes rekordok egyedi azonosítója a relációban. A *lang0*, *lang1* és *lang2* attribútumok a szó megfelelőit tartalmazzák a három különböző nyelven. Hogy melyik nyelvet takarják, nincs előre definiálva az adatbázisban. A nyelvek definíciója a *lang.inc* fájlban, mint konstansok kerültek meghatározásra. Ez azzal az előnnyel jár, hogy a nyelvek definícióját egy három soros fájlban történő javítással elvégezhetjük.

A *kategoria* attribútum azoknak a kategóriáknak a rövidítéseit tartalmazza, amelyekbe a szó beletartozik. Mivel egy szó több kategóriába is tartozhat - miközben nem tudjuk előre mennyibe - a következő megoldást választottuk. A rövidítések három karakterből állnak. Minden rövidítésnek van még egy záró karaktere, ami a ; karakter. Úgy

gondoltuk, hogy egy szó legfeljebb öt kategóriába tartozhat, ezért a *kategoria* attribútum hosszát 20 karakterben határoztuk meg. ($5 * 3 + 5 * 1 = 20$)

A *megjegyzes* attribútum annak az xml fájlnek a nevét tartalmazza, amelyik a szó mint fogalom bővebb leírását tárolja. A *kep* attribútumban a szóhoz tartozó kép fájl nevét tartalmazza.

A *biralta* attribútum annak a felhasználónak az azonosítóját tárolja, aki az adott szót jóváhagyta, és a szótár szavai közé kerülését lehetővé tette.

A második relációnak a *biralando* megnevezést adtuk. Ahogy a neve is mutatja, ez a reláció tárolja azokat a szavakat, amelyek már bekerültek az adatbázisba, de a jóváhagyásukra még nem került sor. A struktúrája a szavak reláció struktúrájához hasonló, mert a *biralando* reláció ugyanolyan adatokat tartalmaz, mint a szavak reláció.

Tab. 2 A biralando reláció struktúrája

Az attribútum neve	Típus	Hossz (karakter)	Nulla érték megengedett	Automatikus értéknövelés
Id	int	11	nem	Igen
lang0	varchar	60	igen	Nem
lang1	varchar	60	igen	Nem
lang2	varchar	60	igen	Nem
Kategoria	varchar	20	igen	Nem
megjegyzes	varchar	50	igen	Nem
Kep	varchar	50	igen	Nem
Biralva	tinyint	1	igen	Nem

Azt is mondhatjuk, hogy a szavak reláció előszobája. Az egyetlen eltérés a *biralva* attribútumban van. Mivel az itt tárolt szavak még nem kerültek jóváhagyásra, ezért nincs is a bírálót tároló oszlop. Helyette a *biralva* attribútum az előzetes bírálatot teszi lehetővé.

A harmadik reláció *kategoria* néven a kategóriák kódját és a megnevezésüket tartalmazza.

Tab. 3 A kategoria reláció struktúrája

Az attribútum neve	Típus	Hossz (karakter)	Nulla érték megengedett	Automatikus értéknövelés
Kat	varchar	3	nem	Nem
Leiras	varchar	30	nem	Nem

A *kat* attribútum a kategória legfeljebb három karakterből álló egyedi kódját tartalmazza. Ez a kód azonosítja a kategóriát a *szavak* illetve a *biralando* relációkban. A leírás attribútum a kategória teljes megnevezését tárolja.

A negyedik és az ötödik reláció a felhasználókkal kapcsolatos. A felhasználó reláció tárolja a regisztrált felhasználók adatait. A felhasználók csoportokba vannak sorolva. A különböző csoportokba sorolt felhasználók különböző jogosultságokkal rendelkeznek. A csoportok a csoport relációban vannak tárolva. Az első csoport a vendég felhasználóké, akik a legalapvetőbb jogosultságokkal rendelkeznek. Külön csoportba vannak besorolva azok a felhasználók, akik a bírálatot végezhetik el. A legtágabb jogosultságokkal az adminisztrátor csoportba sorolt felhasználók rendelkeznek.

Tab. 4 A felhasználók reláció struktúrája

Az attribútum neve	Típus	Hossz (karakter)	Nulla érték megengedett	Automatikus értéknövelés
Id	int	4	nem	Igen
Login	varchar	20	igen	Nem
Pw	varchar	35	igen	Nem
Nev	varchar	25	igen	Nem
Ingroup	int	1	igen	Nem
Disabled	tinyint	1	igen	Nem

A felhasználók tábla *id* attribútuma a felhasználó egyedi azonosítója. Egyedi kulcs, amely a szavak relációban azt a felhasználót azonosítja, aki a szót jóváhagyta. A *login* és a *pw* attribútumok együttesen a felhasználó autentifikációs adatait képezik. A *login* a bejelentkezési azonosító, a *pw* az azonosítóhoz tartozó jelszó. Ha a felhasználó a bejelentkezési oldalon mindkét adatot helyesen adja meg, a bejelentkezése érvényes, egyéb esetben a bejelentkezése sikertelen. A *pw* attribútum az md5 algoritmus szerinti hash értékben kerül tárolásra az adatbázisban. Ez lehetővé teszi, hogy a felhasználó jelszavát ne lehessen visszakeresni még a legmagasabb jogosultsággal rendelkező felhasználónak sem. A jelszó elvesztése vagy elfelejtése esetén az adminisztrátor csak új jelszót tud definiálni.

A *nev* attribútum a felhasználó nevét tárolja. Nem azonos a *login* attribútumban tárolt azonosító névvel. Alkalmas a felhasználó teljes, polgári nevének a tárolására. Az *ingroup* attribútum annak a csoportnak a kódját tartalmazza, amelyikbe a felhasználó be van sorolva. Mindegyik felhasználó csak egy csoportba lehet besorolva. Alapértelmezés szerint az értéke 1, vagyis ha a felhasználó definiálása során nincs beállítva a csoportba besorolás, a felhasználó a legalapvetőbb jogosultságokat biztosító vendég csoportba kerül.

A *disabled* attribútum a felhasználói fiók érvényességét jelzi. A felhasználói fiókok törlését a program nem teszi lehetővé, csak a blokkolásukra ad lehetőséget. Ha a *disabled* attribútum értéke 0, a felhasználó be tud jelentkezni a rendszerbe, ha 1 az értéke a

felhasználó blokkolva van. Alapértelmezés szerint az attribútum értéke 0, vagyis minden újonnan felvett felhasználó be tud jelentkezni a rendszerbe.

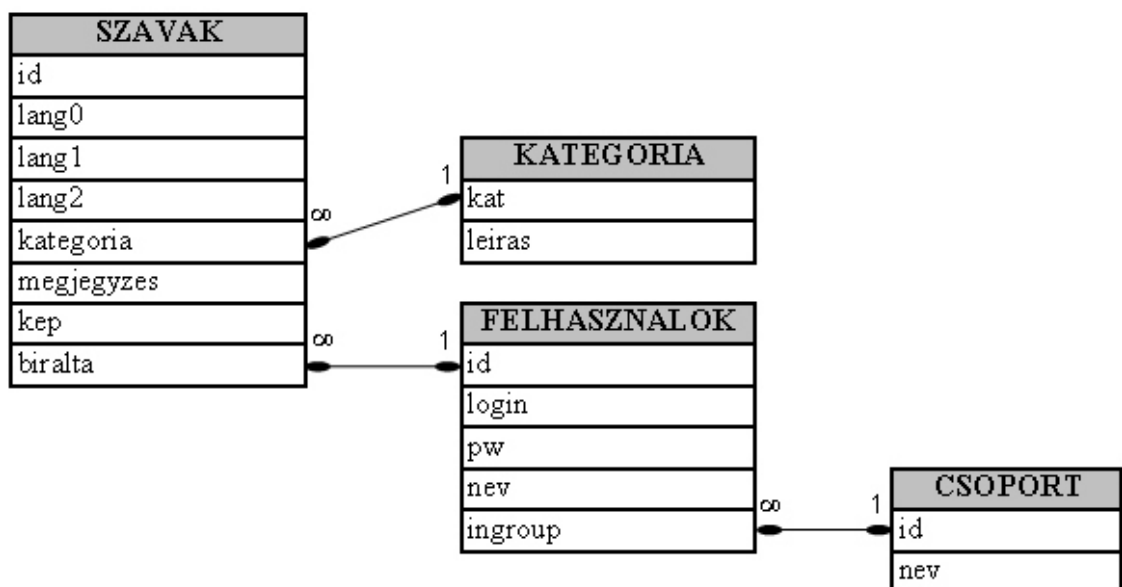
A csoport reláció azoknak a csoportoknak az adatait tartalmazza, amelyekbe a felhasználó besorolása megtörténhet. A csoport tulajdonképpen a hozzáférési lehetőségeket definiálja a felhasználó számára.

Tab. 5 A csoport reláció struktúrája

Az attribútum neve	Típus	Hossz (karakter)	Nulla érték megengedett	Automatikus értéknövelés
Id	int	1	Nem	nem
Nev	varchar	15	Igen	nem

Az *id* attribútum a csoport egyedi azonosítója. Kulcsot képez, amely a *felhasznalo* reláció *ingroup* attribútumában tárolva információt nyújt arról, melyik csoportba van besorolva a felhasználó. A *nev* attribútum a csoport megnevezését tartalmazza.

Az adatbázis relációi közti kapcsolatokat az alábbi ábra szemlélteti.



1. ábra Az adatbázis relációi közti kapcsolat

A *biralando* reláció struktúrája csaknem megegyezik a szavak reláció struktúrájával, ezért a *kategoria* és *felhasznalok* relációkkal meglévő kapcsolatok is egyeznek.

2.4.2 Az XML fájlokban tárolt adatok

Az adatok azon halmazát, melyek hossza nagyon megnövekedhet, nem az adatbázisban, hanem külső fájlokban tároljuk. A szavak pontosabb és bővebb leírását tartalmazó adatok tartoznak ebbe a halmazba. A kétféle tárolási mód között a kapcsolatot egyfelől a külső fájl neve, másfelől a *szavak* és *biralando* relációk *megjegyzes* attribútumai alkotják. Mindegyik szó leírása egy XML fájlba kerül. Mivel mindegyik szó leírása három nyelven kerül tárolásra, erre alkalmas struktúrát kellett kialakítani. A XML struktúra a következő:

```
<?xml version="1.0" encoding="windows-1250" ?>
<nyelvek>
  <nyelv>
    <lang0 />
    <lang1 />
    <lang2 />
  </nyelv>
</nyelvek>
```

A struktúra első sora az XML szabványnak megfelelően az XML verziójának leírását, és a karakterek kódolásának a definícióját tartalmazza. A *nyelvek* címke a struktúra gyökér eleme. Nyitótagja a struktúra elején, zárótagja a végén található, tehát minden egyéb adatot tartalmaz. A *nyelv* konténer tartalmazza a három nyelv mindegyikét. A nyelveket a *lang0*, *lang1* és *lang2* címkék reprezentálják. Nyitó- és zárótagjaik között helyezkedik el a szó leírása az adott nyelven. Hogy melyik címke melyik nyelvet takarja, a *lang.inc* fájlban van definiálva. Az alkalmazás minden egyes XML fájlja a *config.php* fájlban definiált *XmlUploadDir* konstans értékének megfelelő könyvtárban található.

2.4.3 A képfájlok tárolása

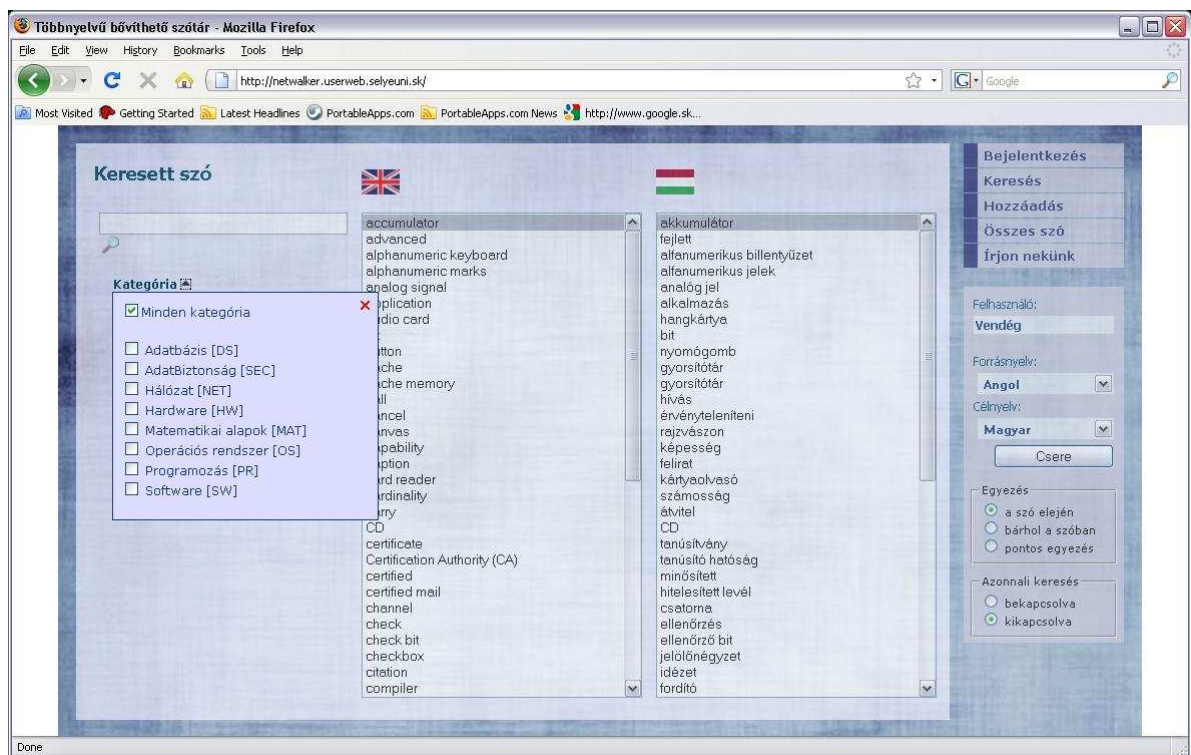
A szótár lehetővé teszi az adott szóhoz tartozó, a fogalom könnyebb megértését szolgáló kép tárolását is. A képeket tartalmazó fájl helyét a fájlrendszerben a *config.php* fájlban definiált *PixUploadDir* konstans határozza meg. Mivel mindegyik képfájl ugyanabban a könyvtárban van tárolva, mindegyik fájl nevének egyedinek kell lennie. A megengedett képformátumok a jpg, jpeg, gif, png, tif, tiff és bmp. A képeket érintő további megkötés a fájl méretére vonatkozik. Maximális méretüket a *config.php* fájlban definiált *MaxFileSize* konstans értéke határolja felülről, jelenlegi értéke szerint 1 MB-ban.

2.5 Az online alkalmazás megvalósítása

Hogy az alkalmazásunk az adatok tárolásán kívül egyéb szolgáltatásokat is nyújtson a felhasználónak, szükség van olyan mechanizmusok megalkotására, amelyek lehetővé teszik az adatokkal történő manipulációt, a felhasználók nyomon követését, és felületet biztosítanak a felhasználónak a programmal történő munkához. Ezeket a feladatokat az általunk megírt kódok valósítják meg. A kódok megalkotásához többségében a PHP-t választottuk, kisebb részben a Javascript-et. Az alkalmazás megjelenését stíluslapok használatával igyekeztük egységessé tenni. A stílusok definíciója a *dict.css* fájlban található.

2.5.1 A kezdőlap

Az alkalmazás betöltődését követően a böngészőben a kezdőlap jelenik meg. A kezdőlapot generáló forráskód az *index.php* fájlban található.



2. ábra Kezdőlap

A kezdőlap a szavak idegen nyelvű megfelelőinek keresésére szolgál. Munkapanelében egy szövegbeviteli mező, és két listadoboz jelenik meg. A szövegbeviteli mezőbe egy karakterláncot vár. Ezt a karakterláncot keresi a forráshelyként definiált nyelv szavai között. Az első listadobozban a szövegbeviteli mező tartalmának megfelelő szavakat jeleníti meg abc sorrendben, miközben a kurzort az első találatra állítja. A második

listadoboz az első listadobozban található szavak célnyelvi megfelelőit tartalmazza. Az első listadobozban a kurzor elmozdítására a második listadoboz kurzorja is reagál, jelezve ezzel a célnyelvi megfelelőt. A második listadoboz kurzorának a mozgása csak az első listadoboz kurzorával befolyásolható. A szövegbeviteli mező alatt található kategória felirat melletti nyílra kattintva a felhasználónak lehetősége nyílik az adatbázisban található összes szóból kiszűrni azokat, amelyek bizonyos kategóriákba vannak szervezve. A szűrés beállítására szolgáló felületet egy előugró panel biztosítja, amelyen az adatbázisban definiált kategóriák jelennek meg. Az egyes kategóriák kiválasztását jelölőnégyzetek segítségével végezheti el a felhasználó. Alapértelmezésben mindegyik kategória ki van választva. A kategória panel működését a „divhandle.js” állományban tárolt Javascript kód valósítja meg, megjelenését stíluslapként a „divhandle.css” állományban definiáltuk. Az egyes kategóriákat jelölőnégyzetek reprezentálják. A jelölőnégyzetek egy általános tárolóban vannak elhelyezve. Az általános tároló megjelenését vagy rejtett voltát a stíluslapban definiált display tulajdonság adja meg. A felhasználó által generált eseményeket a javascript kód kezeli, ami nem tesz mást, mint a stíluslapban beállított display tulajdonságot a megfelelő értékre állítja. A felhasználó beállítja a jelölőnégyzeteket, majd a keresést elindítva a jelölőnégyzetek értékeinek megfelelően a szavak kategória attribútumuk alapján kerülnek be az eredményhalmazba.

A forrásnyelv és a célnyelv a jobb oldalon elhelyezkedő beállítási panel „forrásnyelv”, illetve „célnyelv” feliratú legördülő listájával választható ki. A nyelvek beállítását lehetővé tevő legördülő listák alatt elhelyezett „csere” feliratú nyomógomb a forrás- és célnyelvek gyors megcserélését teszi lehetővé. A kívánt eredményhalmaznak a keresés kulcsának megfelelő szavakat kell tartalmaznia forrás- és célnyelven egyaránt a forrásnyelv szerint rendezve, és az így kapott eredményhalmazt még meg kell szűrni a kategóriák beállításainak megfelelően. Mivel ezek a feltételek a működés közben változnak, az adatbáziskezelőhöz intézett kérést több darabból állítjuk elő a következő PHP kódba ágyazott SQL utasítással:

```
$query1 = "select id, $slang, $dlang, kategoria from szavak where ($slang like '$key') ";  
if ($checkedkats=="0") $query2 = "";  
else {  
    $query2 = "and ";  
    for ($i=0; $i<count($filterkat); $i++) {  
        if ($i==0) $query2 = $query2."(kategoria like '%" . $filterkat[$i] . '%" ";
```

```

else $query2 = $query2."or kategoria like '%" . $filterkat[$i] . "%' ";
if ($i==count($filterkat)-1) $query2 = $query2.") ";
}
}
$query3 = "order by $slang, $dlang";
$query = $query1.$query2.$query3;

```

ahol a *query* változó a kívánt eredményhalmazt reprezentálja, az *slang* a forrásnyelvet, a *dlang* a célnyelvet és a *key* változó a keresés kulcsát tartalmazza. A *checkedkats* változó a kategóriákat reprezentáló jelölőnégyzetek értékeiből előállított nullákból és egyesekből álló karakterlánc, a *filterkat* változó a kategória relációból kinyert kategóriakódokat tartalmazó tömb. A kérés első része a forrásnyelv, a célnyelv és a keresés kulcsa alapján szűri a szavakat. A második rész a kategóriák szerint szűri az eredményhalmazt. Ha az összes kategória opció volt megjelölve, a kérésnek ez a része üresen marad, mivel ekkor nem kell kategóriák szerinti szűrést alkalmazni. Az első és a második rész között logikai szorzás történik. A kérés harmadik része a rendezést vezérli.

Az alkalmazás a keresésre többféle lehetőséget is kínál. Ezeknek a beállítására az egyezés és az azonnali keresés címkéjű rádiógombok szolgálnak. Az „egyezés” feliratú rádiógombok segítségével beállítható, hogy a keresett karakterláncot pontos egyezés alapján, a szó elején, vagy bárhol a szóban keresse az alkalmazás. Alapértelmezés szerint a szavak elején történő keresés van beállítva. A programkódban az egyezést a keresés kulcsát tartalmazó *key* változó modifikálásával valósítottuk meg oly módon, hogy helyettesítő karakterrel bővítettük ki a megfelelő helyen.

```

if ((isset($_POST["key"])) && ($_POST["key"]!="")) {
    switch ($mode) {
        case 0: $key=$_POST["key"]."%%"; break;
        case 1: $key="%".$_POST["key"]."%%"; break;
        case 2: $key=$_POST["key"]; break;
    }
}

```

Az azonnali keresés lehetővé teszi azt a keresési és megjelenítési módot, ahol a szövegbeviteli mezőbe történő íráskor minden változásra egy új keresés indul el. Ilyenkor nincs szükség a betöltött karakterlánc után az enter billentyű vagy az egér által jóváhagyni

a bevitt szöveget. Igaz ez nagyon kényelmes és hatásos működést eredményez, azonban tekintve, hogy minden billentyű megnyomása után új keresés indul, ami egyben egy új kérés a szervertől, lassúbb működéshez vezethet túlterhelt szerver vagy lassúbb hálózati kapcsolat esetén. További hátránya, hogy az újra és újra fókuszba kerülő szövegbeviteli mező értékét egy javascript kódnak mindig ki kell olvasnia, majd újra beírnia a mezőbe, így biztosítva, hogy a kurzor mindig a megfelelő helyen legyen. Ez további idővesztést jelent. Ennek okán ez a keresési lehetőség bizonyos esetekben nem megfelelő. Ilyenkor hatékonyabb az a keresési mód, amikor csak a teljes karakterlánc beírása után indul el a keresés. Ebben az esetben az alkalmazás az enter billentyű leütését, vagy az ilyenkor megjelenő nagyító ikonra való kattintást várja a felhasználótól. Az azonnali keresés alapértelmezés szerint kikapcsolt állapotban van. A forráskódban ezt a funkciót úgy oldottuk meg, hogy a rádiógombok állásának függvényében változtattuk meg mikor küldje el az oldalt a keresési kulcsot tartalmazó szövegbeviteli mező.

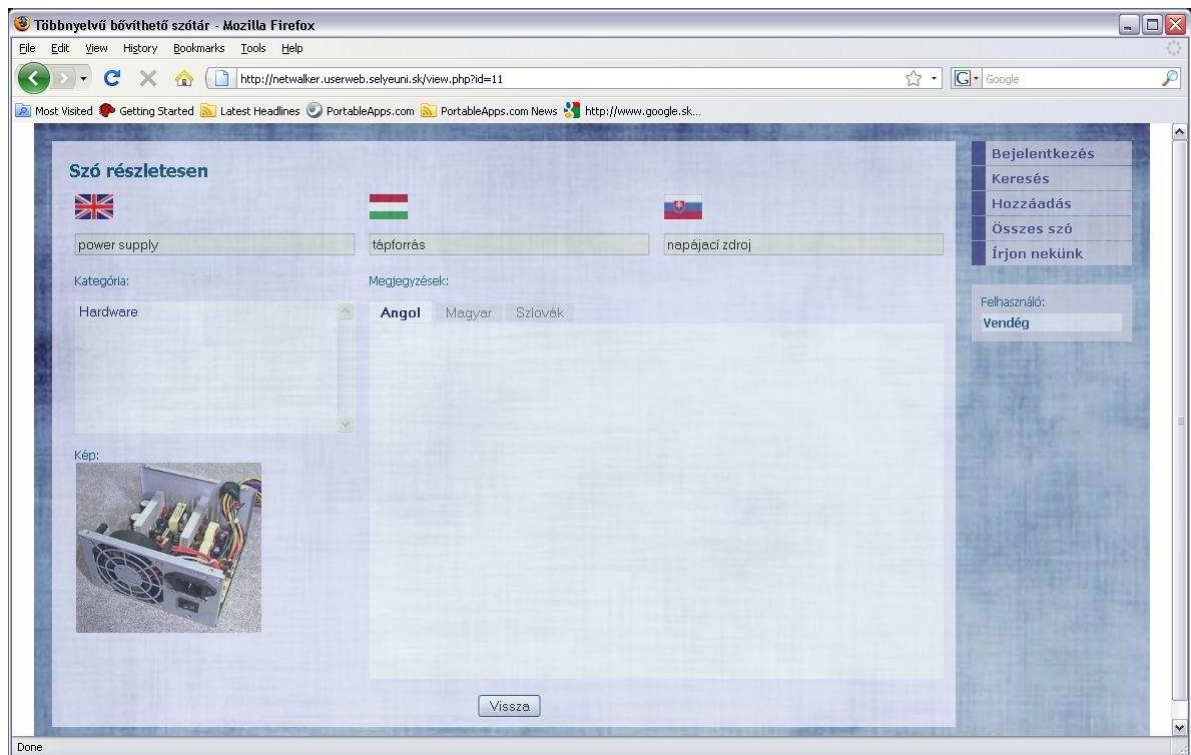
```
<input type="text" name="key" ...
```

```
<?php echo $typeOfSearch==1 ? 'onKeyUp' : 'onChange' ?> ="this.form.submit()"
```

A szavak közti keresést az alkalmazás minden felhasználónak lehetővé teszi. A felhasználó kilétét a beállítási panelen jelzi számunkra. Nem bejelentkezett felhasználó esetén a „Vendég” feliratot kapjuk, mint felhasználót.

A forrásnyelv szavait tartalmazó listadoboz valamelyik sorára kattintva kiválaszthatjuk a kívánt szót. Ilyenkor a cél nyelv szavait tartalmazó listadoboz is azonnal reagál, és a kurzort a forrásnyelvi listadobozban kiválasztott szó cél nyelvi megfelelőjére állítja. Ebben a nézetben csak a szavak szótári megfelelőjét kapjuk meg az alkalmazástól. A szó kiválasztása után duplakattintással kérhetjük le a kiválasztott szó részleteit. Mivel az alkalmazásnak ez a része a szavak keresését teszi lehetővé, semmilyen lehetőségünk nincs a szavak modifikációjára, csupán megtekinthetjük azokat. A részleteket megjelenítő lapot a *view.php* fájlban lévő forráskód generálja, az őt meghívó, ebben az esetben a keresési oldaltól kapott paraméter segítségével. A *view.php* paramétereként a keresési oldalon kiválasztott szó azonosítója szerepel. Ez az azonosító a *szotar* adatbázis szavak relációjának *id* attribútuma az adott szó sorában. A paraméter átadása GET módszerrel történik, vagyis a részleteket megjelenítő oldalra történő hivatkozás ki van bővítve legalább egy paraméterrel. Az alábbi ábrán a *http://localhost/szotar/view.php* hivatkozás a *?id=11* karakterlánccal van kibővítve, ahol a *?* jelzi, hogy paraméter következik, az *id* a

változó neve, a 11 pedig a változó értéke. Ezen az oldalon a szóhoz tartozó minden információ megjelenésre kerül.



3. ábra Részleteket megjelenítő oldal

A felső sorban található a szó mindhárom nyelven. A nyelv identifikációjára kis zászlócskák szolgálnak. A baloldalon látható, hogy mely kategóriákba van besorolva az adott szó. A kategóriák alatt a szóhoz tartozó kép jelenik meg. A „megjegyzések” felirat alatt a szó leírása szerepel. Tekintettel arra, hogy a leírás 255 karakternél hosszabb is lehet, egy időben csak az egyik nyelven jelenik meg a leírás. Ha valamelyik másik nyelven kívánjuk olvasni, a szöveg felett lévő fülek segítségével kiválaszthatjuk a kívánt nyelvet. Azt, hogy melyik nyelv az aktív, a fülek megjelenése teszi világossá a felhasználó számára. Az aktív nyelvet reprezentáló fül enyhén nagyobb és világosabb a többi fülnél. A visszatérést a keresés lapra a „Vissza” feliratú nyomógommbal, vagy a menü „Keresés” pontjának kiválasztásával realizálhatjuk. A kezdőlapról való navigációt az alkalmazás egyéb részeibe a jobb oldalon elhelyezkedő menü teszi lehetővé.

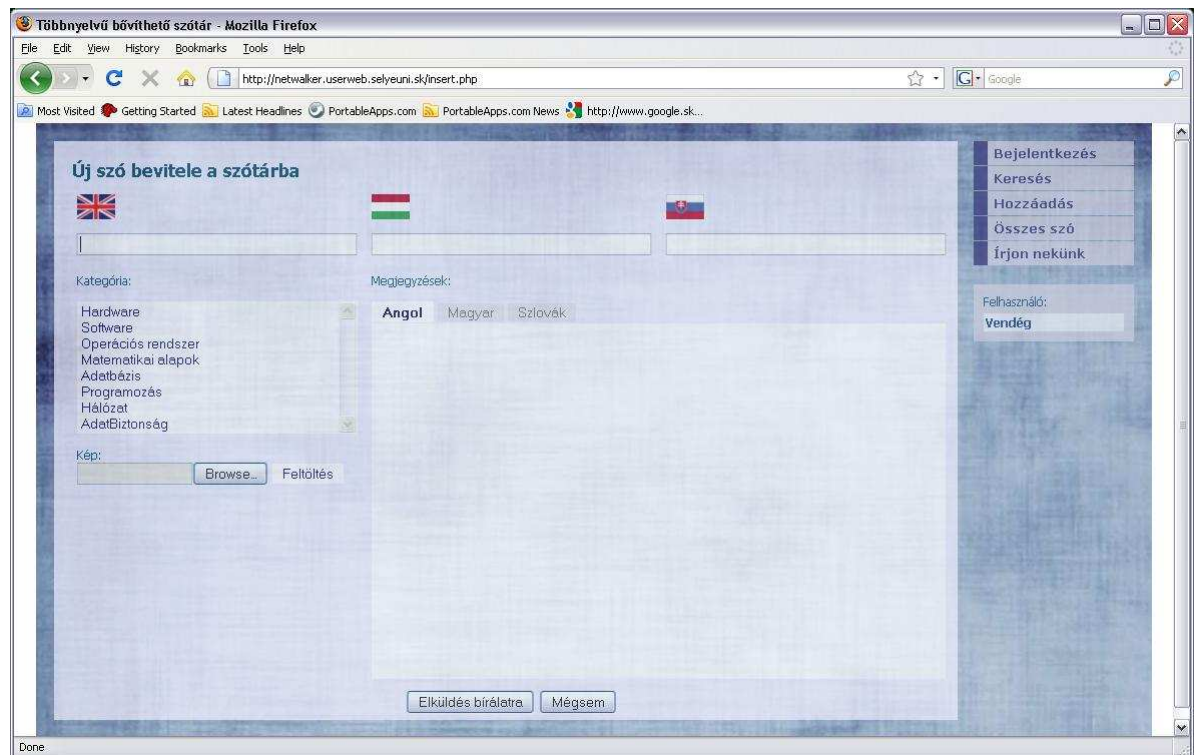
2.5.2 Új szó bevitel a szótárba

Új szó bevitelére a szótárba bármelyik felhasználónak lehetősége nyílik. Az újonnan a szótárba kerülő szavak nem kerülnek be azonnal a szavakat tartalmazó adatok közé, vagyis a bevitel után nem visszakereshetőek. Előbb egy a *cenzura* csoportba tartozó

regisztrált, vagyis a *felhasznalok* relációban jegyzett és legalább a 8 csoportba tartozó felhasználónak jóvá kell azt hagynia.

Az új szó bevitelére szolgáló lap a menü „Hozzáadás” pontjára kattintva érhető el. Egy a részleteket megjelenítő lapra hasonló oldalra jutunk. A különbség az, hogy az új szó bevitelénél az oldal elemei a kategória lista kivételével mind üresek, valamint, hogy a felhasználónak lehetősége adódik az adatok megváltoztatására.

Az új szó bevitelére szolgáló forráskód az *insert.php* fájlban található. A zászlókkal jelölt szövegbeviteli mezőkbe a szót kell beírni a zászlónak megfelelő nyelven. A kategóriákat egy többértékű listaboxban kell megjelölni az egér kattintásával. Amennyiben a szó több kategóriába is besorolható, akkor a Ctrl+bal egérgomb kombinációval több érték is kijelölhető. A szóhoz tartozó kép feltöltéséhez először meg kell keresni a képfájlt a háttértárolón. Erre a szövegbeviteli mező mellett elhelyezkedő nyomógomb szolgál. A felirata a kliens gépre telepített operációs rendszer lokalizációjának függvényében változik. Magyar operációs rendszer esetén ez a „listázás” felirat, angol esetén „browse”, szlovák esetén „vyhladat”. A fájl kiválasztása után a képet fel kell tölteni a kiszolgálóra, a „Feltölt” nyomógomb segítségével. Ilyenkor a fájl a szerveren egy az ideiglenes fájloknak fenntartott könyvtárba kerül.



4. ábra Új szó bevitelére szolgáló lap

A szövegmező felett a három nyelvet reprezentáló jelölőfül helyezkedik el. Ezek a fülek tulajdonképpen nyomógombok, csak a megjelenésük módosítva van stílussal. A többsoros szövegmezőbe az aktív fül feliratának megfelelő nyelvű leírás kerül. A fülek váltása esetén az előző fül beállításának megfelelő szöveg egy ideiglenes xml fájlba kerül. A következő váltáskor az újonnan bekerült szöveg is ebben a fájlban tárolódik, mint ahogy az azt követő minden további változás is. Az oldal alján két nyomógomb található. A Vissza nyomógomb értényteleníti a bevitt adatokat, és visszavigáz az oldalt meghívó lapra. Az Elküldés bírálatra nyomógomb hatására az adatok a *biralando* táblába kerülnek mentésre. A képfájl a *config.php* fájlban definiált könyvtárba kerül, és a fájl neve bekerül a *biralando* tábla *kep* attribútumába. Az ideiglenes xml fájl tartalma egy egyedi nevű, végleges xml fájlba kerül. Az xml fájl neve szintén mentésre kerül a *biralando* tábla *megjegyzes* attribútumába. A mentési folyamat befejeztével visszakerülünk az előző lapra.

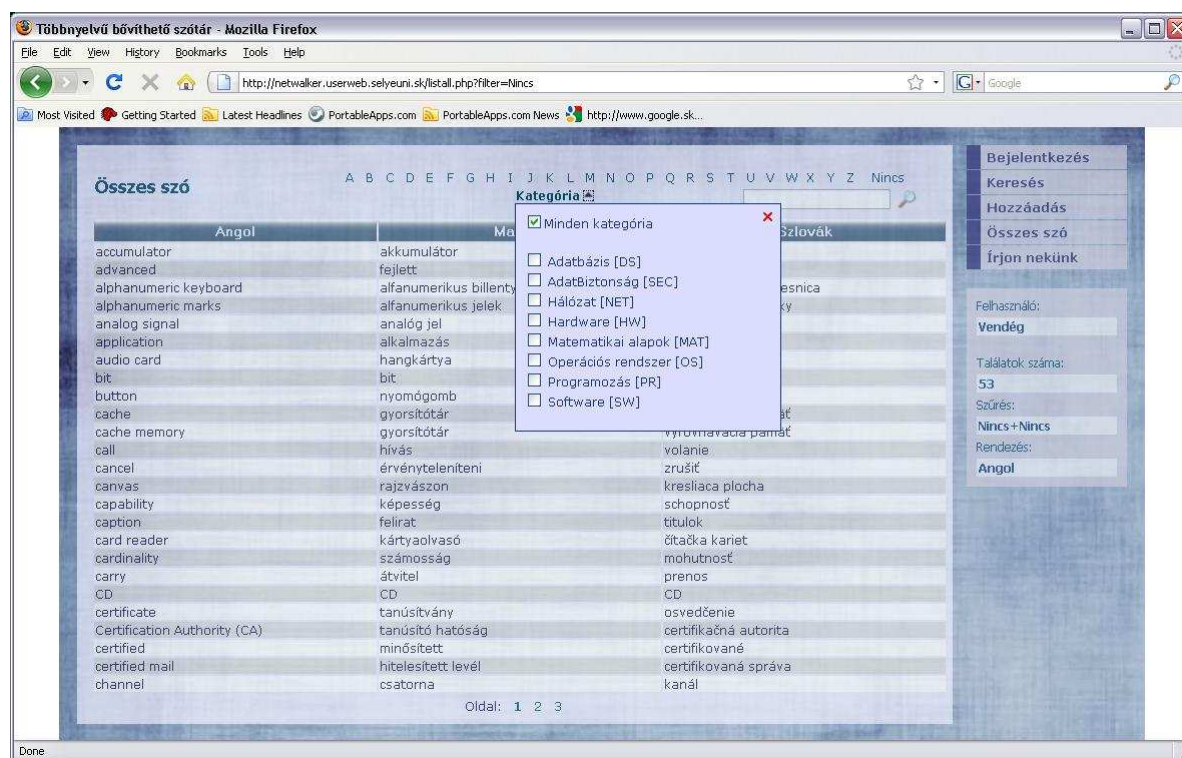
2.5.3 Összes szó listázása

Az összes szó listázása lap eredeti funkciója szerint az összes szó megjelenítését teszi lehetővé mindhárom nyelven egy időben. Kiterjesztett funkcióinak köszönhetően hasonló lehetőségeket biztosít, mint a keresés lap. Az oldal generálását végző forráskód a *listall.php* fájlban található.

Az oldal betöltődése után a szavak táblázatos formában jelennek meg a munkapanel felületén. A lap görgetését elkerülendő az oldalra egyszerre 25 szó kerül. A táblázat alatt helyezkedik el az oldalszámláló, ami félkövér betűstílussal jelzi az aktív oldal számát. Az oldalak számának nemcsak jelzőfunkciója van. Mindegyik oldal száma egy hivatkozás is, így a többi szó listázására is lehetőséget nyújt. A szavakat tartalmazó táblázat fejlécére kattintva választhatjuk ki, melyik oszlop, vagyis melyik nyelv szerint legyen rendezve a táblázat tartalma.

A munkapanel első sorában az abc betűi találhatók. Valamelyik betűre kattintva aktiválhatjuk a szűrőt, és a táblázatban csak a kiválasztott betűvel kezdődő szavak jelennek meg. A kiválasztás arra a nyelvre vonatkozik, amelyik szerint rendezve van a szavak listája. A „nincs” hivatkozásra kattintva a szűrő kikapcsolható. Az abc betűi alatt található szövegmező egy másfajta szűrést tesz lehetővé. Azok a szavak kerülnek kiíratásra, amelyekben a szövegmezőbe beírt karakterlánc bárhol megtalálható. A szűrés harmadik lehetőségét az előzőekben már ismertetett kategória felirat és a hozzá tartozó nyíl szolgáltatja. A három szűrési mód között logikai szorzás érvényesül, tehát a felhasználónak lehetősége van megadni melyik kategóriákban, milyen betűre kezdődő és milyen

karakterláncot tartalmazó szavakat keres. A jobb áttekinthetőség érdekében a jobb oldali beállítási panelen tájékozódhat a felhasználó a rendezés és a szűrő állapota felől. A szűrésnek megfelelő találatok számáról is itt kap tájékoztatást.



5. ábra Az összes szót listázó oldal

Valamelyik szóra duplán kattintva lehívható a szó részletes adatait tartalmazó oldal. Mivel az alkalmazásnak ez az ága is szabadon, bármelyik felhasználó számára hozzáférhető, itt is csak az adatok megtekintésére nyílik lehetőség, a változtatásukra nem.

2.5.4 Bejelentkezés

Az alkalmazás eddig ismertetett részeit minden felhasználó szabadon használhatja. Kereshet a szavak között, és új szavakat vihet be az adatbázisba. A továbbiakban ismertetésre kerülő funkciók azonban már csak a *felhasznalo* relációban bejegyzett felhasználók számára hozzáférhetőek. Mindaddig, amíg a felhasználó nem azonosította magát, vendég felhasználóként kezeli a rendszer. Az azonosítás a menü „Bejelentkezés” menüpontjával történik. Mindegyik regisztrált felhasználó rendelkezik egy azonosítóval és egy, az azonosítóhoz tartozó jelszóval. A két adat egyidejű helyes megadása teszi lehetővé a bejelentkezést. Miután a bejelentkezés megtörtént, az adatbázisban regisztrált adatok alapján a felhasználó valamelyik csoport tagjaként tevékenykedik tovább az alkalmazásban. A rendszer ezt a csoportazonosítót teszteli minden olyan esetben, amikor a külső, vendég felhasználóknak tiltott programrészre kerül. Maga az alkalmazás a nem

bejelentkezett felhasználónak nem nyújt felületet a számára tiltott programrészek elérésére, erre az ellenőrzésre mégis szükség van. Előfordulhat, hogy a böngésző címsorába beírva, vagy a kedvencek közül kiválasztva próbál a vendég felhasználó valamelyik programrészre navigálni. Ebben az esetben az alkalmazás automatikusan átirányít a kezdőlapra.



6. ábra Bejelentkezési felület

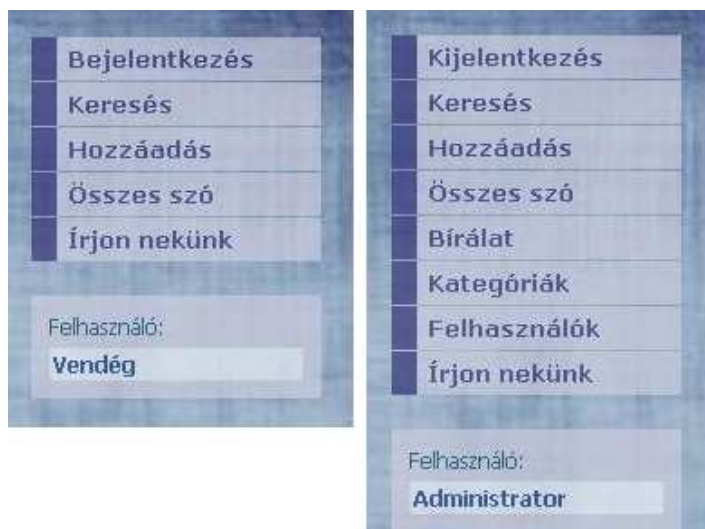
A „Bejelentkezés” menüpontra kattintva megjelenik a bejelentkezési felület. Két szövegmezőt és két nyomógombot tartalmaz. Az első szövegmezőbe a felhasználó azonosítóját várja, a másodikba az azonosítóhoz tartozó jelszót. A jelszó szövegmezőbe íráskor a beírt karakterek nem láthatóak, pontokkal helyettesítve jelennek meg. A két szövegmező kitöltését követően a bejelentkezés nyomógombra kattintva történik meg a bejelentkezés. Amennyiben a szövegmezőkben található adatokkal a *felhasznalo* relációban nincs egyezés, vagy a felhasználó *blocked* attribútuma nem 0, a bejelentkezés sikertelen. Erről a felhasználó a felső sorban, piros mezőben kap értesítést.



7. ábra Értesítés a sikertelen bejelentkezésről

Sikertelen bejelentkezés esetén a felhasználó módosíthatja a bejelentkezési adatokat, és újra próbálkozhat a bejelentkezéssel. A mégsem nyomógomb használatával a bejelentkezés elmarad, és a program visszatér a bejelentkezést meghívó lapra.

Sikeres bejelentkezés esetén a felhasználó lehetőségei kitágulnak. Ennek vizuális jele a menüben látható. Attól függően, hogy melyik felhasználói csoportba van besorolva új menüpontok jelennek meg, miközben az első, „Bejelentkezés” menüpont kijelentkezésre vált. A menü alatt elhelyezkedő beállítási panelen a felhasználó címke alatt megjelenik a felhasználó neve.



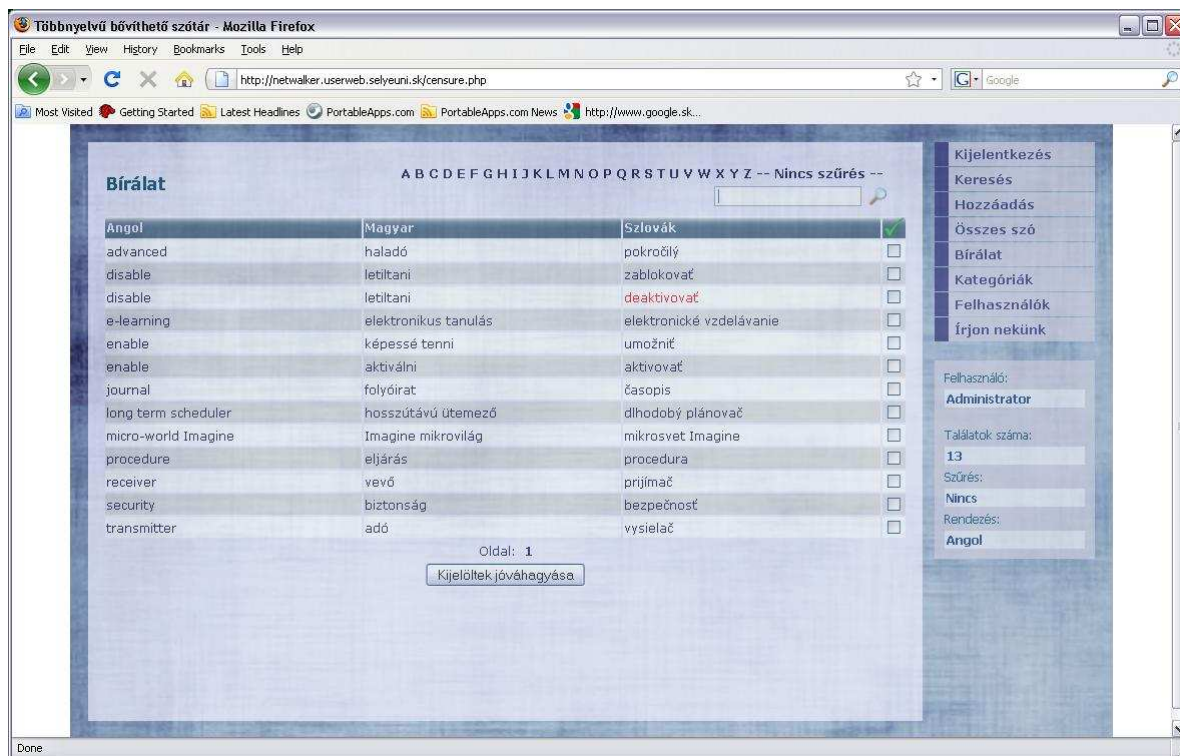
8. ábra Megváltozott menü

A „Kijelentkezés” menüpontra kattintva a felhasználó visszatér a vendég csoportba és elveszti a regisztrált felhasználói mivoltából adódó privilégiumokat. Ennek a lehetőségnek a használata mégis ajánlott, elsősorban abban az esetben, ha a felhasználó kénytelen távozni a számítógépétől. A be- és kijelentkezést a *login.php* illetve a *logout.php* fájlokban található kód vezérli.

2.5.5 A szavak helyességének elbírálása

A 8. és a 9. csoportba sorolt felhasználóknak, tehát azon felhasználóknak, akikről bejegyzés található a *felhasznalo* relációban és az *ingroup* oszlopban 8 vagy 9 van, lehetőségük nyílik a *biralando* relációban lévő szavak elbírálására, és a szavak relációba juttatására. A bírálat két lépcsőben történhet. A „Bírálat” menüpontra kattintva a bírálat lap nyílik meg. Első látásra felfedezhető a hasonlóság az összes szót megjelenítő lappal. A legfőbb különbség abban rejlik, hogy a bírálat lapon az elbírálásra váró szavak jelennek meg. Ezen kívül minden sor végén van egy jelölőnégyzet, és az oldal alján található egy „Kijelöltek jóváhagyása” feliratú nyomógomb. A rendezés és a szűrés lehetősége ezen a lapon is segíti a felhasználót a kívánt információ megtalálásában. A pozitív elbírálás a legegyszerűbb esetben úgy történik, hogy a felhasználó a szavak mellett található jelölőmezőt kijelöli, majd a „Jóváhagyás” nyomógombra kattintva a szavak relációba

juttatja a kijelölt szavakat. Amennyiben a nyomógombra kattintás elmarad, a szavak kijelölt voltát a rendszer megjegyzi. Ez a lehetőség akkor hasznos, ha a bíráló a szavakat előzetesen helyesnek ítélte meg, de még meg szeretné konzultálni valakivel, vagy bármilyen egyéb okból kifolyólag a döntése még nem végleges.

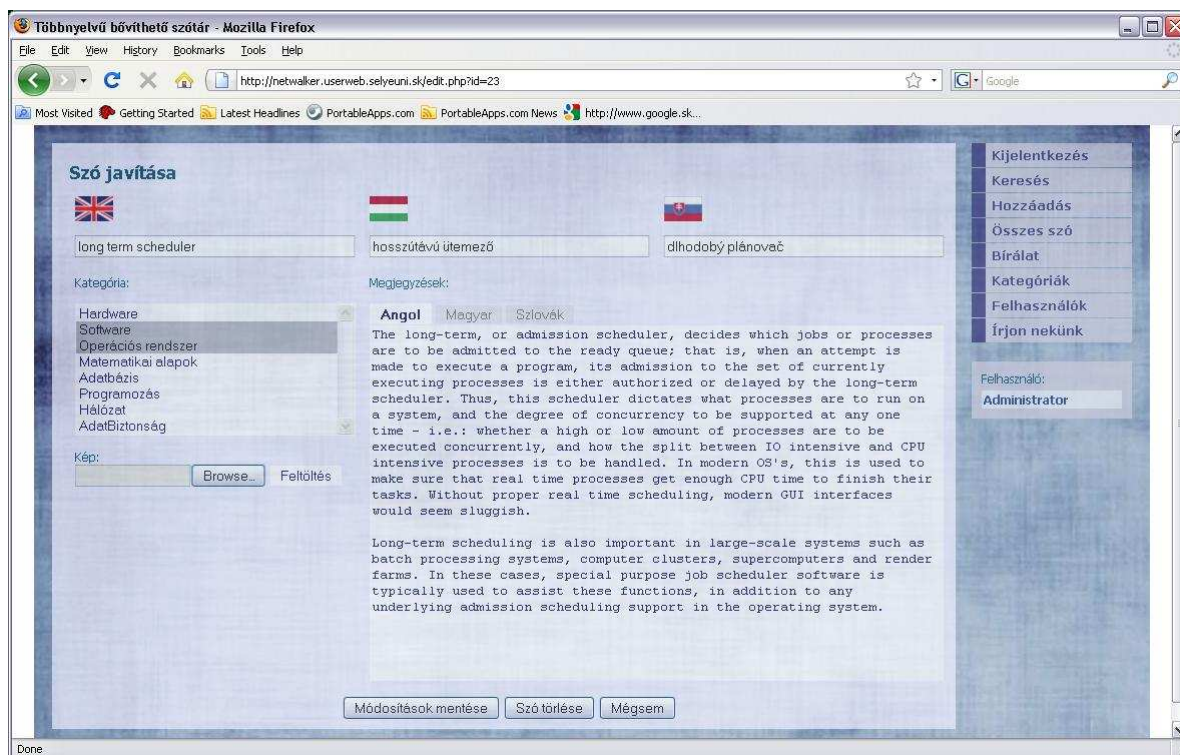


9. ábra Bírálat lap

Az esetek többségében a bírálat lapon található információk nem elegendőek ahhoz, hogy a bíráló felelősségteljesen véleményt nyilvánítson a szavakról. Nem beszélve arról, hogy a bírálónak még nem tettük lehetővé, hogy a hibás szavakat kijavítsa, vagy törölje a rendszerből. Ezért a bírálat lapon található szavak hivatkozásként is működnek. A szóra duplán kattintva a szó részletes lapja jelenik meg.

Ez az oldal tulajdonképpen a szó és a hozzá tartozó adatok javítását teszi lehetővé. A felső sorban maga a szó szerepel mindhárom nyelven. A kategória listában mindegyik kategória megjelenik, de csak a szó bevitelkor megjelölt sorok vannak kijelölve. A kép felirat alatt a szóhoz tartozó kép is megjelenik, ha van. A megjegyzése alatt a szó bővebb leírása a nyelveket reprezentáló fülekkel váltható át. Minden adat a szó bevitelénél ismertetett módon változtatható meg.

A oldal alján három nyomógomb helyezkedik el. A „Mégsem” feliratú visszanyitja a bírálat oldalra, miközben az esetleges változtatásokat elveti. A módosítások mentése nyomógomb elmenti a módosításokat, majd ugyancsak visszatér a bírálat oldalra.



10. ábra Szó javítását lehetővé tevő oldal

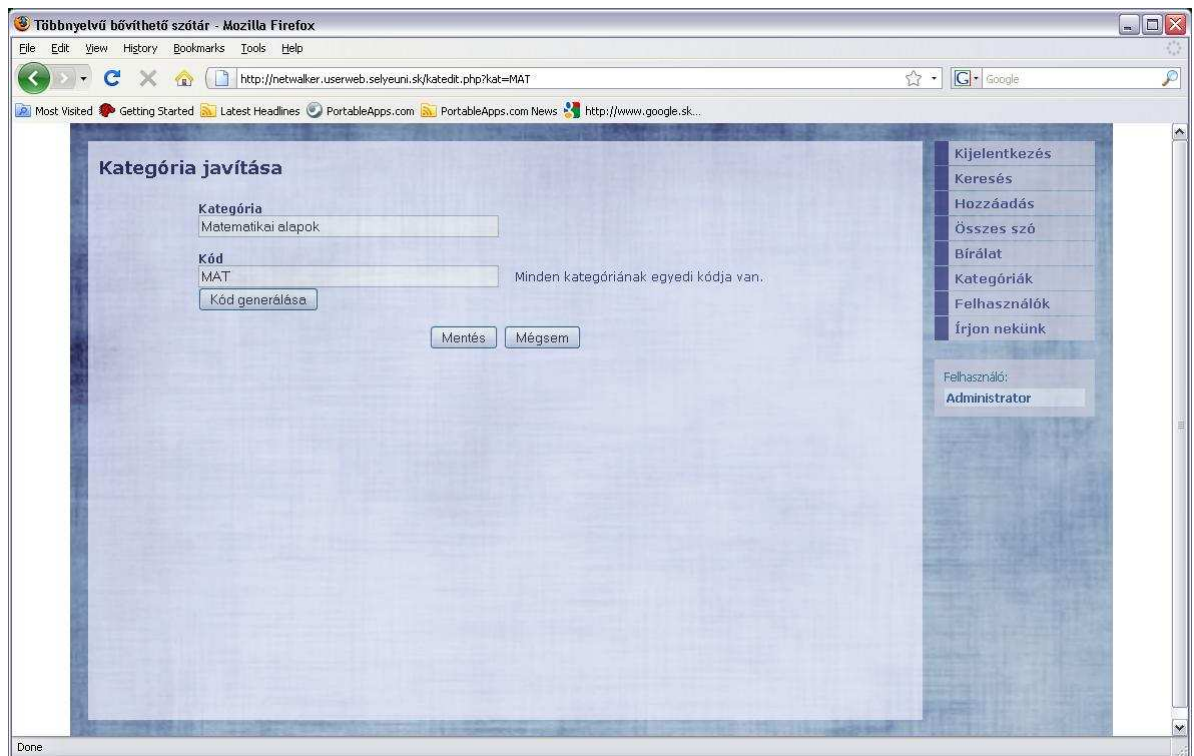
Fontos megjegyezni, hogy a módosítások ekkor még mindig csupán a *biralando* relációba kerülnek mentésre. Vagyis az itt végrehajtott módosítások nem jelentik a szó jóváhagyását is. Az csak a bírálat oldalon a szó kijelölése, majd jóváhagyása után történik meg. A szó törlése nyomógomb nyújt lehetőséget a szó végleges törlésére a rendszerből.

2.5.6 Új kategória definiálása, és a kategóriák módosítása

A szótárban a szavak kategóriákba vannak besorolva, miközben egy szó több, de legfeljebb öt kategóriába is besorolható. A szavak bevitelekor a felhasználó csak a már definiált kategóriák közül választhat. Új kategóriák definiálására és a meglévők modifikálására a bíráló és az adminisztrátor csoportokba tartozó felhasználóknak van lehetősége.

A menü „Kategóriák” pontjára kattintva megjelenik az összes kategóriát listázó oldal. Látható rajta az egyes kategóriák legfeljebb háromjegyű, egyedi kódja, valamint a kategória megnevezése. A kód nagybetűs formában van tárolva. Megadhatja a felhasználó, vagy generáltathatja a programmal. A kód generálása a kategória nevéből történik. A nagybetűs formára alakítást és a duplicitás ellenőrzését az alkalmazás végzi el. Duplicitás esetén a felhasználó értesítést kap. A lista alatt az új kategória definiálását lehetővé tevő nyomógomb helyezkedik el. A listában felsorolt kategóriák bármelyikére duplán kattintva lehetőség nyílik a kategória adatainak megváltoztatására.

A kategória kódjának megváltoztatása esetén szükséges a kód megváltoztatása minden olyan szó bejegyzésében is, amelyik az adott kategóriába van besorolva. Ez a változtatás a szavak és a *biralando* relációkat egyaránt érinti. Mivel mindegyik szó több kategóriába is be lehet sorolva, a szavak és *biralando* relációk *kategoria* attribútumában a kódok pontosvesszővel elválasztva szerepelnek.



11. ábra Kategória javítására szolgáló oldal

Egy kategória megváltoztatásánál ezt a karakterláncot a pontosvesszők mentén fel kell darabolni, megváltoztatni az adott kategória kódját, majd újra összerakni és menteni az adatbázisba. Ezt a műveletet a szavak reláción az alábbi kód hajtja végre,

```
$query = "select ID, kategoria from szavak";
$result_kat = mysql_query($query)
    or die ("Hiba az utasítás végrehajtásában. Kód: 1015 ".mysql_error());
while ($record = mysql_fetch_array($result_kat)) {
    $xszavakid = $record['ID'];
    $kategoria = "";
    $kats=explode(';', $record['kategoria']);
    foreach ($kats as $value) {
        if ($value != $xkat) {
```

```

        if ($value != "")
            $kategoria .= $value.'\';
    }
    else
        $kategoria .= $kat.'\';
}
$query = "update szavak set kategoria='$kategoria' where ID='$xszavakid'";
$result_updateszavak = mysql_query($query)
    or die ("Hiba az utasítás végrehajtásában. Kód: 1015 ".mysql_error());
}

```

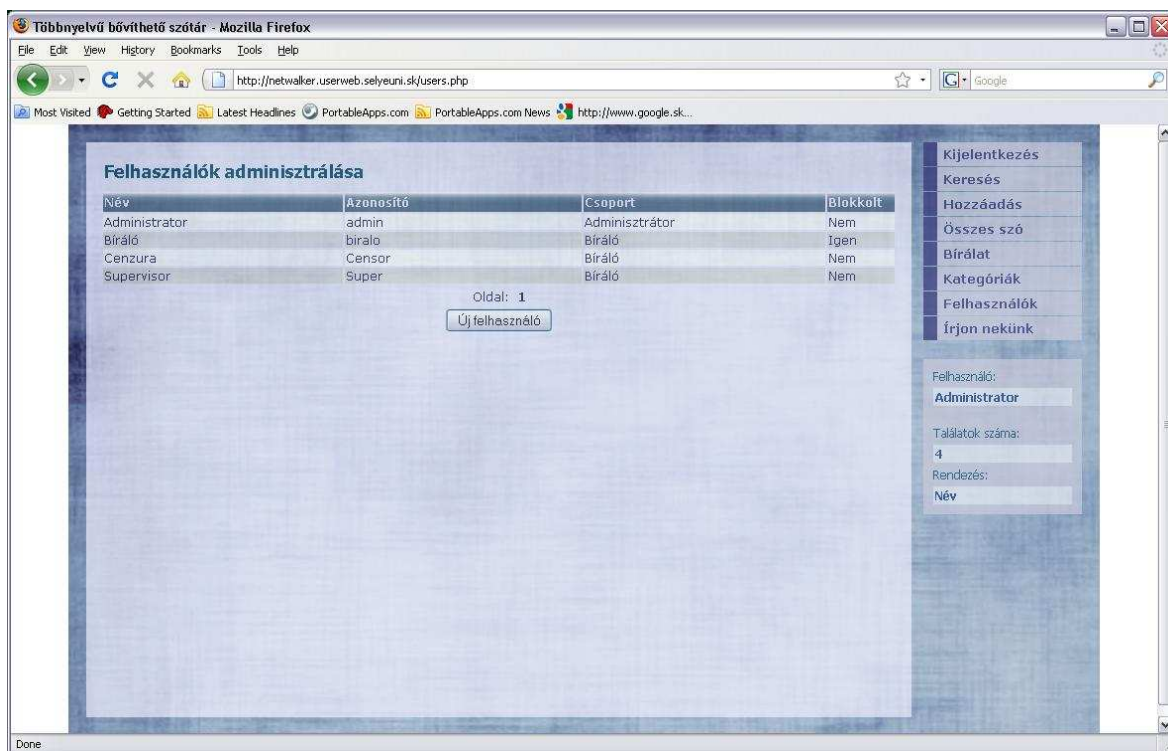
ahol az *xkat* változó a kategória get vagy post metódussal érkező kódját tartalmazza annak függvényében, hogy az oldal önmagát hívta meg, vagy egy szülő oldaltól érkezett a hívása.

A kategóriákat kezelő forráskódok a *kats.php*, *newkat.php* és *katedit.php* fájlokban találhatók.

2.5.7 Felhasználók kezelése

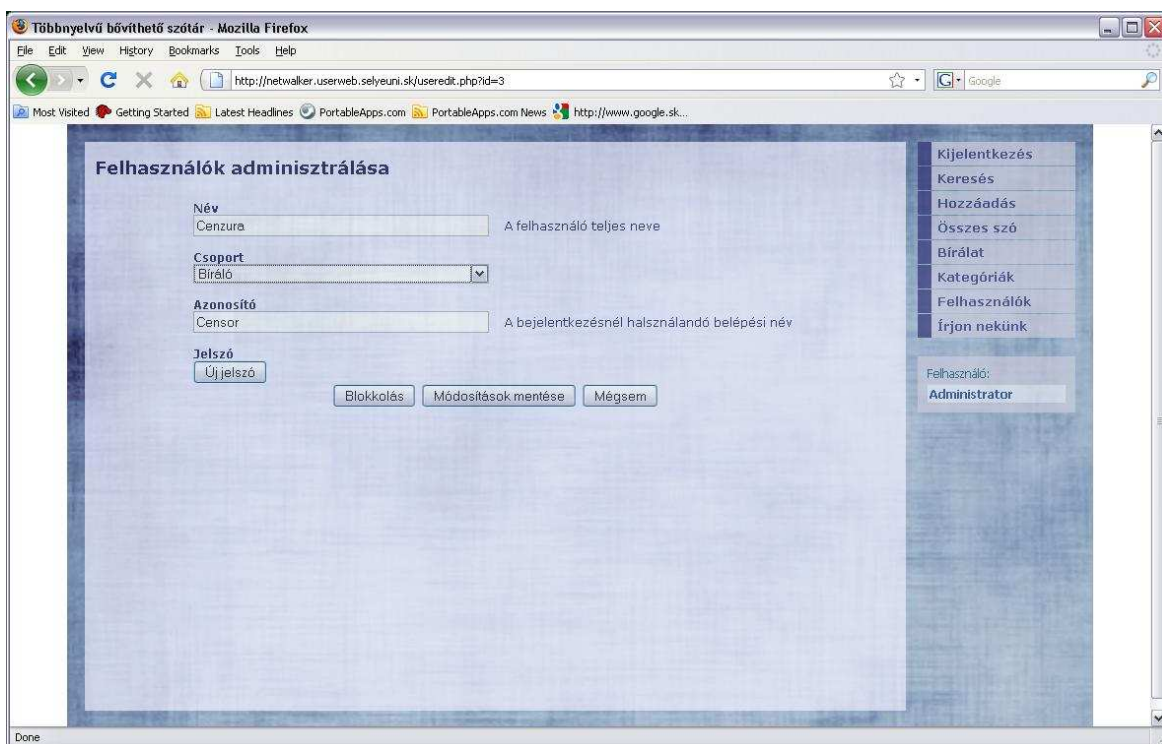
A felhasználói fiókok módosítására csak az adminisztrátor csoportba tartozó felhasználóknak van lehetősége. Ő definiálhat új felhasználót, változtathatja meg a nevüket és a bejelentkezési adatokat, csoportokba sorolhatja, ezzel együtt a jogosultságait is meghatározhatja. A felhasználók kezelését lehetővé tevő felhasználói felület a „Felhasználók” menüpontra keresztül érhető el.

Az új felhasználó nyomógombra kattintva új felhasználót vehetünk fel a rendszerbe. A felhasználóra duplán kattintva lehetőség nyílik a felhasználó adatainak megváltoztatására. A név és az azonosító megváltoztatásához elég a megfelelő szövegmező tartalmának felülírása. A csoportba sorolás modifikációja egy legördülő listán keresztül történhet. Itt az előre definiált értékekből választhatunk. A jelszó megváltoztatásához az „Új jelszó” feliratú nyomógombra kell kattintani. Ekkor megjelenik két szövegmező. Ezekbe kell begépelni az új jelszót. Hogy a jelszó a beíráskor is rejtve maradjon, mégis a felhasználó visszajelzést kapjon a szövegmező tartalmának a változásáról, a begépelte szöveg helyett pontok jelennek meg. A jelszóváltás sikeres, ha a két szövegmező tartalma megegyezik. Ellenkező esetben a felhasználó értesítést kap a változtatás sikertelenségéről és annak okáról.



12. ábra Felhasználók kezelését biztosító felület

A mégsem nyomógombra történő kattintást követően a rendszer visszatér a felhasználókat listázó oldalra, miközben az esetleges változtatásokat elveti. A módosítások mentése nyomógomb a változtatások mentésére szolgál.



13. ábra Felhasználó adatainak megváltoztatását lehetővé tevő felület

Mivel a *szavak* reláció *biralta* attribútumában hivatkozás van az adott szót elbíráló felhasználóra, a felhasználók törlésére az alkalmazás nem ad lehetőséget. Amennyiben valamelyik felhasználónak lehetetlenné kívánjuk tenni a bejelentkezést, a blokkolás nyomógommbal érvényteleníthető a felhasználói fiók. Blokkolt felhasználó esetén a blokkolás nyomógomb felirata kiblokkolásra vált, és lehetőséget nyújt a felhasználói fiók újbóli érvényesítésére. A felhasználó adatainak megváltoztatása nem okoz zavart az adatbázisban, mert a felhasználók elsődleges azonosítója a *felhasznalo* reláció *id* attribútuma, amely nem változik.

2.5.8 Felhasználók követése

Alkalmazásunk online applikáció, vagyis egy időben több felhasználó munkáját is lehetővé teszi. Hogy az azonos időben az alkalmazást használó felhasználók adatai ne keveredjenek, szükséges az oldalkérések (page request) között megőrizni a felhasználóra és a munkafolyamatra vonatkozó adatokat. A felhasználók nyomon követésére a fejlesztés során a *session* változókat használtuk. Minden szkript elején meghívásra kerül a *session_start()* funkció, ami megpróbálja a felhasználótól átvenni a munkamenet azonosítóját. Ha sikerül megszereznie, azonosítja a felhasználót, ha nem, akkor új munkamenetet hoz létre. Ezt követően meghívásra kerül a *session_register()* funkció, ami lehetővé teszi azoknak a változóknak a mentését, amelyek a munkamenet során keletkeznek.⁶ Munkánk során a *session_register("login")* funkcióval a bejelentkezett felhasználó azonosítóját, a *session_register("ingroup")* funkcióval pedig a felhasználó jogosultságait teszteltük. A *session_register("back")* funkció segítségével az adott oldalt meghívó szülő oldal nevét tartja számon az alkalmazás.

2.6 Az alkalmazás felhasználhatóságának lehetőségei

Minden tantárgy oktatásának eredményességét növelni lehet megfelelő tansegédeszközök hatékony használatával. Az információs-kommunikációs technológia (IKT) megjelenése az oktatásban jelentősen megnövelte a tansegédeszközök körét. Flexibilis, interaktív, a diák aktív részvételét lehetővé tevő eszközök segítik a tanárok és diákok munkáját. Az eszközkészlet oldaláról tekintve az oktatásban felhasználható a számítógép, az internet, az interaktív tábla, különböző vetítéstechnikai eszközök és oktatási

⁶ BRÁZA, J: *PHP5*, Grada Publishing, a.s., Praha, 2005, 203.o., ISBN 80-247-1146-X

szoftverek. Tartalmi értelemben az e-learning vagy blended learning is tansegédeszköznek tekinthető.

Az oktatás folyamatában az információs-kommunikációs technológia több területen is felhasználható. Elsőként a tananyag demonstrációjára használt tansegédeszközök említhetők. Feladatuk a tananyag szemléltetése, akár több szempontból is megközelítve azt. A sztatikus képi megjelenésen túl ide tartoznak a számítógépes animációk és szimulációk is. Az információ percepciójának tekintetében lehetnek vizuális, auditív vagy audiovizuális taneszközök. Hatékony segítséget nyújtanak különböző fejlődési folyamatok, ciklikus változások, történelmi csaták vagy éppen gépek működésének bemutatására. A demonstrációs tansegédeszközökről elmondható, hogy a leggyakrabban használtak, de mivel a tananyag bemutatásán túl további funkcióval nem rendelkeznek, kevésbé hatékonyak. Másodikként az elsajátított tudás ellenőrzésére szolgáló tansegédeszközök említhetők, melyek leggyakrabban tesztek formájában jelennek meg. Összetettebb változataik valamilyen elképzelt szituációba helyezik a tanulót. Az adott helyzetben a diákot döntések elé állítják, majd e döntések alapján további szituációkat generálnak, fokozatosan terelve a tanulót a megfelelő döntések meghozatala felé.

Külön kategóriát képeznek a sajátos nevelési igényű gyermekek oktatására készült tansegédeszközök és tansegédeszköz rendszerek. Talán ezen a területen mutatható fel a legtöbb eredmény.

Az elektronikus enciklopédiák, lexikonok, szótárak, példatárak és archívumok is hatékony segítséget jelentenek az oktatói munkában. Elsősorban forrásként használhatóak. Jellemzően kulcsszavakra épülnek, általuk gyorsan nyerhető ki a kívánt információ. Különböző szempontok szerint a fogalmak rendszerezve vannak, esetleg kategóriákba sorolva is megtalálhatók. A keresést, valamint az összefüggések bemutatását hivatkozások segíthetik.

Az elektronikus tansegédeszközök a fent említetteken kívül hozzájárulnak a digitális írástudás elterjesztéséhez is. A természettudományok, matematika, informatika és a zene oktatásában fejleszthetik a konstruktív gondolkodás készségét. Lehetővé teszik az információ több kommunikációs csatornán történő közlését, ezzel is növelve az oktatás hatékonyságát, és ami talán a legfontosabb előnyükként említhető, a játszva tanulás lehetőségét is biztosítják.

Az általunk készített alkalmazás szótár és fogalomár, tehát elsősorban forrásként használható. Mivel minden felhasználónak lehetővé teszi új fogalmak felvételét, az alkotás örömét is biztosítani képes. Abból adódóan, hogy a szótárba a szűrést követően csak a

helyes bejegyzések kerülhetnek, sikerélményt is nyújthat a diákoknak. Tudva azt, hogy a saját bejegyzése a továbbiakban nemcsak neki, de a társainak is segítséget nyújthat, valamint tudva azt is, hogy a mások bejegyzéseit ő is sikerrel használhatja, erősítheti az összetartozás élményét.

Tartalmi szempontból vizsgálva a lehetőségeket kijelenthető, hogy bármely tantárgy esetében használható az alkalmazás, hiszen mindegyik tudományág rendelkezik szakszókészlettel. Ha több tantárgy fogalmainak gyűjtésére kívánjuk használni, alapvetően két megvalósítási formával számolhatunk. Az első esetben minden bejegyzést egyetlen alkalmazás tart nyilván. Ilyenkor ajánlott a kategóriák számának növelése és részletezése. Ennek a megoldásnak az az előnye, hogy minden egy helyen elérhető. Hátrányként az adatbázis megnövekedésével együtt járó sebességcsökkenés említhető. A másik megvalósítási forma szerint mindegyik tantárgyhoz külön alkalmazás társul, ezzel együtt külön adatbázis is a rendelkezésre áll. Ebben az esetben az egyes tantárgyak szótárai külön címeken érhetők el. Több kisebb adatbázissal dolgozunk, így az alkalmazás reakcióideje csökken. Megjegyzendő, hogy a kiszolgáló reakcióidejére ez nem feltétlenül igaz, hiszen több alkalmazást kell futtatnia. Az új bejegyzések elbírálása egyszerűbbé válik, hisz elég minden alkalmazás esetében egy szaktanár jóváhagyása. A tantárgyak közti szabad átjárhatóság előnyét viszont elveszítjük.

Az egy intézményen belüli felhasználáson túlmenően elképzelhetőnek tartjuk az alkalmazás olyan működtetését is, mely minden iskola számára elérhető. Lehet ez általános vagy tantárgyak szerint rendszerezett háromnyelvű szakszótár, melynek fejlesztését szakemberek koordinálják, felügyelik és garantálják a szakmai helyességét. Az üzemeltetést valamelyik országos hatáskörrel rendelkező intézmény vagy alapítvány biztosíthatná. Konkrétan a somorjai Biblioteca Hungarica, a nyelvi tanácsadással foglalkozó dunaszerdahelyi székhelyű Gramma Nyelvi Iroda említhetőek potenciális üzemeltetőkként. További lehetőségként a Szlovákiai Magyar Pedagógusok Szövetségének állandó jelleggel működtetett honlapján elhelyezett alkalmazás említhető. A szövetség minden magyar iskolával tartja a kapcsolatot, ismert a pedagógusok előtt. Egy központi fogalomtár hasznos segítséget jelenthetne minden pedagógus számára, de pl.a TUDOK-versenyekben résztvevő diákok kutató és tudományos tevékenységét is segíthetné.

Jelenleg az alkalmazás a Selye János Egyetem Informatika Szolgáltató Központja által rendelkezésünkre bocsátott tárhelyen, a <http://netwalker.userweb.selyeuni.sk> címen érhető el. A teljes funkcionalitást biztosító adminisztrátori bejelentkezéshez az *admin*, *admin1234* felhasználói név - jelszó párost használva lehet bejelentkezni.

Összefoglalás

Munkánk célja egy olyan szótárprogram megalkotása volt, amely három nyelven tárolja a szavakat, azonos időben több felhasználó munkáját is lehetővé teszi és folyamatosan bővíthető. További elvárás volt, hogy az alkalmazás fogalomtárként is használható legyen, tehát tartalmazza a szavak bővebb leírását, képét is. Azért, hogy folyamatosan bővíthető legyen, a szótárt az interneten keresztül hozzáférhető weboldalként terveztük meg, és a szavak hozzáadását mindenki számára lehetővé tettük. Az újonnan megadott szavakat egy ideiglenes tárolóként funkcionáló adatstruktúrában tároljuk egészen addig, amíg egy regisztrált és megfelelő jogosultságokkal felruházott felhasználó jóvá nem hagyja. A jóváhagyást követően az új szó bekerül a szótárba. Ha az új szót a bíráló nem találja helyesnek, lehetősége van javítani, vagy törölni azt.

A szótár jelen formájában angol, magyar és szlovák nyelvekre van beállítva. Igyekeztünk az alkalmazást úgy tervezni, hogy ez a három nyelv a lehető legkisebb beavatkozással változtatható legyen. Ennek érdekében egy szövegfájlban definiáltuk a nyelveket reprezentáló konstansokat. A mindössze három soros szövegfájl módosításával átállítható a szótár más nyelvekre.

Az egyes szavak adatait a fejlesztés kezdetétől adatbázisban kívántuk tárolni. Ettől az elképzeléstől két adat esetében tértünk el a későbbiekben. Ez a két adat a szónak, mint fogalomnak a bővebb leírása mindhárom nyelven, és a szót reprezentáló kép. A bővebb leírás hosszabb szöveget tartalmazhat, a kép mérete pedig a jelenlegi beállítások alapján elérheti az 1 MB-ot. Ilyen nagy adatokat nem kívántunk az adatbázisban tartani, mivel tartottunk tőle, hogy ez negatívan befolyásolná az egész alkalmazás sebességét. Ezért a két adatot külső fájlokban tároljuk, az adatbázisban pedig ezekre a fájlokra hivatkozunk. Tisztában vagyunk vele, hogy adatok külső fájlban történő tárolása lassúbb, mint az adatbázisban való tárolás, de figyelembe véve, hogy ezek az adatok csak a szavak részletes megjelenítése során jelennek meg, ezt a megoldást találtuk megfelelőbbnek.

A szótár fejlesztésénél a gazdaságosság is fontos szempont volt. Igyekeztünk olyan eszközöket igénybe venni, amelyek szabadon és térítésmentesen hozzáférhetőek. Mivel webes alkalmazást kívántunk létrehozni, választásunk az Apache webszerver, PHP szkriptnyelv és a MySQL adatbázis kezelő rendszerre esett. Ezt az eszköztárat bővítettük az XML jelölőnyelvvvel, JavaScript parancsnyelvvvel és a kaszkádolható stíluslapokkal.

Következő fontos szempontnak az alkalmazás felhasználóbaráttá tételét tekintettük. Igyekeztünk a programban található vezérlő és megjelenítő elemeket olyanra elkészíteni,

hogy mindenkinek első látásra világos legyen, mi mire való, és az egyes grafikai elemek a képernyőn szimbiózisban legyenek egymással.

Meggyőződésünk szerint sikerült egy jól használható, kis hardverigényű, olcsó alkalmazást készítenünk. Az alapelképzelésből adódóan egy önszerveződésszerűen bővülő, mégis hatékonyan ellenőrzött, hasznos alkalmazás kereteit hoztuk létre. A szótár eredetileg az informatika szakszótárának készült. Azonban mivel minden tudomány rendelkezik a maga szakszókinccsével, lehetőség van bármely más tudományág keretein belül történő felhasználására.

Felhasznált irodalom jegyzéke

BAKOS F. *Idegen szavak és kifejezések szótára*. Budapest: Akadémiai Kiadó, 2007. ISBN 9789630578752.

BRÁZA, J. *PHP5*. Praha: Grada Publishing, a.s., 2005, ISBN 80-247-1146-X

GUTMANS, A. – BAKKEN, S. – RETHANS, D. *Mistrovství v PHP 5*. Brno: Vydavatelství a nakladatelství CP Books, a.s., 2005, ISBN 80-251-0799-X

LACKO, L. *Web a databáze*. Praha: Computer Press, 2001, ISBN 80-7226-555-5

MERCER, D. – KENT, A. – NOWICKI, S. – MERCER, D. – SQUIER, D. – CHOI, W. *PHP5*. Budapest: Panem KFT, 2006, ISBN 963 545 450 3

WEINMANN, L. *Velká kniha webdesignu*. Brno: Zoner Press, 2004, ISBN 80-86815-10-2

Dr. SIKI Z. *Adatbázis kezelés és szervezés*. [online]. [2011.1.16].

<<http://www.agt.bme.hu/szakm/adatb/db3.html>>

[online]. [2011.1.18]. <<http://ikon.inf.elte.hu/~cserep/index.php?fm=1>>.

[online]. [2012.1.4]. <<http://www.stackoverflow.com/>>.

Word Wide Web Consortium. [online]. [2011.2.20]. <<http://www.w3c.org>>.

Melléklet

Melléklet A:

- Forráskódok, adatállományok és képállományok
- Az állományok jegyzéke
- Telepítési útmutató