

Szakdolgozat

Szimeonov Miklós

Debrecen
2009

Debreceni Egyetem

Informatika Kar

Vizsgáztató rendszer fejlesztése

Témavezető:

Pere Zsolt
számítástechnikai munkatárs

Készítette:

Szimeonov Miklós
informatika tanári-matematika

Debrecen
2009

Tartalomjegyzék

1. Bevezető	4
2. Fejlesztői eszközök.....	6
2.1. LAMP	6
2.2. Zend Framework	8
2.3. JavaScript	11
2.4. Eclipse	11
3. Rendszer felépítése	12
3.1. Adatbázis	12
3.2. Működés	21
4. Más rendszerek megvizsgálása.....	29
4.1. EVR (Elektronikus Vizsgáztató Rendszer)	29
4.2. Moodle.....	29
5. Rendszer bemutatása	31
5.1. Adminisztráció	31
5.2. Tanár.....	33
5.3. Tanuló.....	44
6. Összefoglalás	48
7. Köszönetnyilvánítás.....	50
8. Irodalomjegyzék	51

1. Bevezető

A számítástechnika hardveres és szoftveres eszközei rohamos fejlődésének köszönhetően napjainkban szinte minden háztartásban találkozhatunk számítógéppel. Hazánkban is növekszik az a tendencia, hogy minden olyan háztartásban, ahol számítógép van, rendelkeznek Internet-előfizetéssel. A fejlődésnek köszönhetően a szoftverek kezelése felhasználók számára folyamatosan egyszerűsödött, a kor követelményévé vált a felhasználóbarát szoftverek készítése. Napjainkban az internetes alkalmazások, egyre nagyobb teret hódítanak, az asztali alkalmazásokat felváltják az online használható szoftverek. Manapság ebben az elkényelmesedett világban az Internet csodájának köszönhetően egyre több szolgáltató tartja fontosnak az online ügyintézkést. Vonatjegy vásárlásától kezdve banki ügyintézésen keresztül a zöldségvásárlásig szinte bármit el tudunk intézni Interneten keresztül.

Lassan egy évtized telt el, mióta széles körben megjelentek a közoktatásban a számítógépek. A NAT-nak köszönhetően kötelező tantárggyá vált az informatika oktatása. A kor megköveteli az alapvető informatikai ismereteket, hiszen nehéz olyan szakmát mondani, ahová még nem gyűrűzött volna be. A harmadik évezred szakemberével szemben alapkövetelmény az átalakítható tudás és a továbbfejlődésre, az önmegújulásra való képesség megszerzése. Megnőtt a projekt típusú oktatás szerepe és új képzési modellek kerülnek előtérbe. Napi gyakorlattá váltak a magas szintű interakciós kapcsolatokat megvalósító számítógéppel segített tantermi és egyéni tanulást biztosító stratégiák. Nemcsak a közoktatásban, hanem a felsőoktatásban is elengedhetetlen az informatikai eszközök magas fokú alkalmazása. Az utóbbi évtizedben a felsőoktatásban résztvevő tanulók számának rohamos növekedése elkerülhetetlenné teszi a diákok tudásának számítógéppel történő ellenőrzését.

Ezen okok miatt merült fel bennem egy online vizsgáztató rendszer fejlesztésének az ötlete. Azonban hosszas töprengés után rájöttem, hogy ha egymagam készíteném el a rendszert, az egy embernek óriási munka lenne. Így egy informatika szakos hallgatóval közösen kezdtük el a munkát. Mivel ő nem ebben a félévben fejezi be a tanulmányait, ezért a rendszer nem teljes egészében készült el.

A szakdolgozatomban a rendszer fejlesztésének előkészületeit, a rendszer szerkezetét, fejlesztési fázisait és használatát fogom bemutatni. Ezen folyamatok a következők:

- A fejlesztői eszközök címszó alatt bemutatom a rendszer fejlesztése során használt eszközöket:
 - o Operációs rendszert, webszervert, adatbázisszervert, és a rendszer fejlesztéséhez használt programozási nyelvet, ami a PHP
 - o Zend Framework keretrendszert, amit a program átláthatósága, bővíthetősége miatt vezettem be
 - o JavaScript programozási nyelvet, amivel az oldal működése vált interaktívabbá, felhasználóbarátabbá
 - o A fejlesztéshez használt integrált fejlesztői környezetet, ami az Eclipse kiegészített változata
- A rendszer felépítése résznél megvizsgálom a rendszer szerkezetét:
 - o Elsőként az adatbázis kerül terítékre, a rendszer 19 táblát használ
 - o A működés alatt a programozási oldalról mutatom meg a rendszert
- Más rendszerek megvizsgálása fejezetnél a piacon fellelhető alternatív megoldások kerülnek górcső alá
- Rendszer bemutatása rész három témakör köré csoportosul ezek:
 - o Adminisztráció rész alatt a rendszerbe történő regisztrációt és a belépést mutatom be
 - o Tanár alcím alatt a tanár szemszögéből kerül bemutatásra a rendszer
 - o Tanuló alcím alatt pedig a tanuló szemszögéből kerül bemutatásra a rendszer

A vizsgáztató rendszer elérhető a: <http://examsystem.hu> helyen, itt lehet megtekinteni, és ingyenesen használni.

2. Fejlesztői eszközök

A fejlesztői eszközök, technológiák megválasztásánál szem előtt tartottam azt, hogy az adott eszköz mindenki számára elérhető, ingyenes legyen. Ebben a részben szeretném bemutatni, az általam használt eszközöket, technológiákat.

2.1. LAMP

A LAMP mozaikszó, amely egy megoldásgyűjtemény, általában szabad és nyílt forrású szoftverekből, melyet dinamikus weboldalak és szerverek futtatásához használnak. A név a következőkből származik:

- Linux
- Apache HTTP Server
- MySQL
- PHP.

2.1.1. Linux

A Linux egy operációs rendszer, amely a szabad szoftverek és a nyílt forráskódú programok egyik legismertebb példája. Linux disztribúció alatt egy adott készítőől vagy gyártótól származó összeállítást értünk, amely egy GNU/Linux alaprendszert, és hozzá tartozó, válogatott és konfigurált programokat tartalmaz. Személy szerint néhány disztribúció kipróbálása után a Debian GNU/Linux-ot választottam. Miért esett éppen erre a disztribúcióra a választásom? Következő indok miatt:

- A szerverek többsége is ezt használja, így nem kellett aggódnom a kompatibilitási problémák miatt. Valóban így történt a dolog, amikor kiraktam a rendszert az oldalra, első indításra tökéletesen működött.
- A biztonsági hibákat a szabad szoftveres közösség gyorsan javítja
- Rengeteg alkalmazás elérhető a rendszerhez, így mindent eszközt sikerült elérnem, használnom, amikre szükségem volt a rendszer fejlesztése során.

2.1.2. Apache

Az Apache HTTP Server egy szabadon terjeszthető, nyílt forráskódú webkiszolgáló szoftver, mely létfontosságú szerepet játszott a World Wide Web (WWW) elterjedésében. Az Apache egy robusztus, erőteljes és rugalmas webszerver, amely kompatibilis a HTTP/1.1 (RFC2616) protokollal. A webszerver egy olyan kiszolgáló, mely a helyileg tárolt weblapokat

HTTP protokollon keresztül elérhetővé teszi. Statikus és dinamikus weboldalak közzétételére egyaránt használják. Sok webalkalmazást az Apache által nyújtott környezethez és szolgáltatásokhoz terveznek. A netcraft felmérése szerint a webszerverek részesedését tekintve az Apache vezet 46%-kal [3]. Ez mindent elárul az Apache használhatóságáról, ezek a tények engem is meggyőztek, így az általam tesztelésre használt webszerver az Apache lett.

2.1.3. MySQL

A dinamikus weboldalak sarkalatos pontja a sebesség. A sebesség egyik legmeghatározóbb eleme az adatbázis elérési, kezelési sebessége. További befolyásoló tényezők voltak még a választásomban a biztonság, az adatbázis méret határok. Választásom a MySQL szerverre esett. A MySQL egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver. PHPMyAdmin egy vizuális rendszer az adatbázisok kezeléséhez. Ez írja és megjeleníti az adatbázisok tartalmát a szerveren (vagy a kliensnél), ahol a MySQL telepítve van. Ezen az interfészen keresztül lehet létrehozni új adatbázisokat, módosítani a már létezőket. Segítségével az adatbázisban tárolt adatok manipulálása is egyszerűbbé válik. Mivel PHP alapú, ezáltal platformfüggetlen hozzáférést biztosít az adatbázishoz. Ezen tulajdonságainak köszönhetően választásom a PHPMyAdmin-ra esett.

2.1.4. PHP

A PHP (PHP: Hypertext Preprocessor) szabadon használható, nyílt forráskódú, szkriptnyelv, dinamikus weboldalak készítésére használják. A PHP az ötös verziótól kezdődően teljesen objektumorientált, ezen paradigma előnyeivel és hátrányaival egyaránt. A rendszer fejlesztése során kiaknáztam az objektumorientáltság adta lehetőségeket. A PHP-t jórészt szerver-oldalon használják, bár létezik parancssori interfésze is, illetve önálló, grafikus felületű alkalmazások is létrehozhatóak vele. A PHP több platformra (webszerverre és operációs rendszerre) ingyenesen telepíthető. Napjainkban több mint 21 millió domainen, kb 1,2 millió szerver futtat PHP-t, bár a nyelvet használó oldalak száma 2005 augusztusától kezdve folyamatosan csökken [7]. A PHP emellett az Apache webszerver egyik legnépszerűbb beépülő modulja. Szintaktikailag könnyű programnyelvről van szó, ugyanis amellett, hogy nem kell az utasítások típusával törődni, azon kívül az egész nyelv egyszerű, könnyen megtanulható. Egyetemi tanulmányaim folyamán nem vettem részt PHP-val foglalkozó kurzuson, így autodidakta módon sajátítottam el a nyelvet.

2.2. Zend Framework

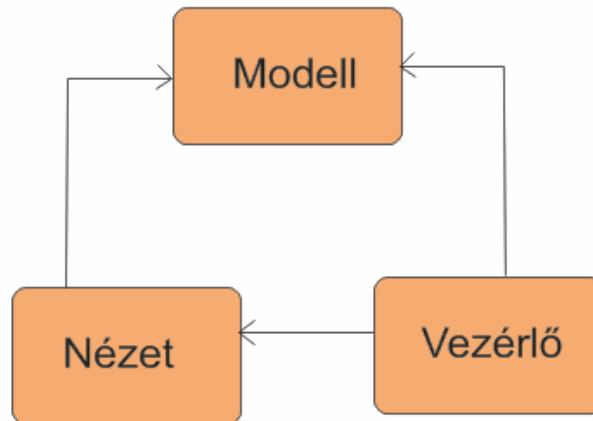
A Zend Framework egy nyílt forrású, objektumorientált webes keretrendszer, amely összetevőket kínál a Modell – Nézet – Vezérlő (MNV) és a Tábla Átjáró (Table Data Gateway) programtervezési mintákhoz. Mivel ezeket, az összetevőket én is használom, ezért mielőtt rátérnék a Zend Framework előnyeire, szeretném tisztázni az MNV és a Tábla Átjáró programtervezési mintákat [8].

2.2.1. MNV (Modell – Nézet - Vezérlő)

Kiseb weblapoknál, nem okoz problémát az, hogy ha HTML és a PHP keverve jelenik meg, mert néhány oldal még átlátható. Valami megoldást kellett találni, amivel egyszerűsödne a karbantarthatóság és a bővíthetőség. Tehát összetett, sok adatot a felhasználó elé táró számítógépes alkalmazásokban gyakori fejlesztői kíváncs az adathoz (modell) és a felhasználói felülethez (nézet) tartozó elemek szétválasztása, hogy a felhasználói felület ne befolyásolja az adatkezelést, és az adatok átszervezhetőek legyenek a felhasználói felület változtatása nélkül. Az MNV tervezési minta (Modell – Nézet – Vezérlő azaz Model – View – Controller Design Pattern) célja különválasztani az adatrepresentációs réteget (Model), a felületi megjelenéstől (View) és az üzleti logikától (Controller). Az MNV jelentősen meghatározza a rendszer felépítését, mint az egy programtervezési mintára jellemző. Nézzük meg az egyes elemek funkcióját:

- Az első a Modell. A rendszer által kezelt adatok, információk reprezentációjáért a felelős. A tartománylogika jelentést ad a puszta adatnak (például kiszámolja, hogy a tanulónak az aktuális dátumhoz képest mikor kezdődik a vizsgája). Sok alkalmazás használ állandó tároló eljárásokat (mint én is ez az adatbázis) adatok tárolásához. Az MNV nem említi külön az adatelérési réteget, mert ezt beleérti a modellbe. Az adatelérési rétegért a Tábla átjáró lesz a felelős.
- A második a Nézet. Megjeleníti a modellt egy megfelelő alakban, mely alkalmas a felhasználói kölcsönhatásra, általában egy felhasználói felületi elem formájában. Különböző célokra különböző nézetek létezhetnek ugyanahhoz a modellhez. Számomra ugye ez a HTML oldal volt.
- A harmadik és egyben legfontosabb a vezérlő. Mint nevéből is adódik ez az elem irányítja a kapcsolatot a modell és a nézet között. Úgy mond ő a felelős a rendszer működéséért. Tehát az eseményeket, jellemzően felhasználói műveleteket dolgozza fel

és válaszol rájuk, illetve a modellben történő változásokat is kiválthat. Például egy linkre kattintva a vezérlő, határozza meg, hogy mi legyen a hatása új rekord, kerüljön be az adatbázisba, esetleg törlődjön egy rekord, stb.



1. ábra: A Modell – Nézet – Vezérlő kapcsolatai

Az MNV összetettebb alkalmazásoknál előnyös, egyszerű weblapok elkészítéséhez fölösleges teher. Ennek oka, hogy jelentősen nagyobb erőforrás és időigénye van egy olyan alkalmazásnak, ami MNV-s programtervezési mintát használ, de ha elég komplex az alkalmazás ez háttérbe szorul és sokkal fontosabbak az átláthatóság, a bővíthetőség, a hatékonyság, és az újrafelhasználhatóság.

2.2.2. Tábla Átjáró

Amikor egy adatbázisban viszonylag sok tábla található, akkor felmerül az igény arra, hogy egységesen kellene ezeket, a táblákat kezelni. A másik szempont, amiért érdemes tábla átjáró mintát használni az, hogy ha az alkalmazás funkcióit összekevernénk SQL utasításokkal sok bonyodalomhoz vezetne. A Tábla Átjáró magába foglalja az összes SQL utasítást, ami egy adatbázistábla eléréséhez, manipulálásához szükséges. Tehát a Tábla Átjáró néhány SQL lekérdezésből, rekordok törléséből, módosításból álló interfész. Nézzünk egy példát:

Person Gateway
find (id) : RecordSet findWithLastName(String) : RecordSet update (id, lastname, firstname, numberOfDependents) insert (lastname, firstname, numberOfDependents) delete (id)

2. ábra: Egy tipikus Tábla Átjáró [1]

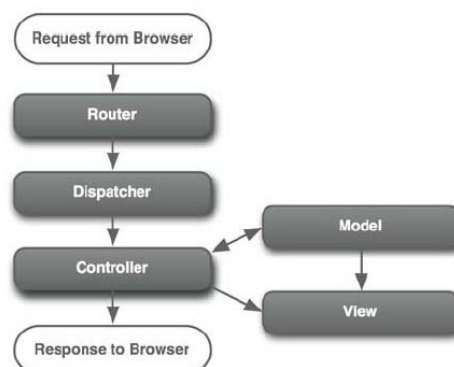
Lekérdező művelet esetén, mint látható a find és a findWithLastName esetén egy rekord halmazzal tér vissza a metódus. A rekordot szintén egy programtervezési mintával szokták megoldani, ez a Sor Átjáró minta. A Sor Átjáró minta segítségével rekordmanipuláló műveleteket végezhetünk el.

Zend Framework fejlesztésének célja a webfejlesztés megkönnyítése, és közben követendő gyakorlatok bevezetése a PHP közösség körébe. Igény szerint használható felépítése lehetővé teszi a fejlesztők számára, hogy újrahasznosítsák az összetevőket, amikor és ahol az alkalmazásukban értelme van, anélkül, hogy egyéb Zend Framework összetevőket követelnének a legalapvetőbb függőségeken túl. A Zend Framework önálló összetevőt nyújt sok más, webalkalmazás fejlesztésekor gyakori követelményhez, beleértve az azonosítást és a jogosultságkezelést hozzáférésvezérlő listákon keresztül, alkalmazásbeállítást, adatok gyorsítótárazását, a felhasználók által küldött adatok szűrését/ellenőrzését a biztonság és az adatok sértetlensége érdekében, nemzetköziesítést, felületeket AJAX funkciókhoz, sok más népszerű webszolgáltatás mellett. A Zend Framework kínál összetevőket a Modell – Nézet – Vezérlő és a Tábla Átjáró programtervezési mintákhoz, melyeket az én alkalmazásom is használ.

2.2.3. Zend framework és az MNV

A Zend Framework-ben: Modellt a Zend_Db valósítja meg; Nézetet az Zend_View; A Vezérlőt Zend_Controller.

Dispatcher és Router a böngészőnek adott URL-t feldolgozzák és átadják a vezérlést a Controller számára. A Vezérlő a Modellbeli adatokkal dolgozik, és ezeknek a kimenetét átadja a View-nak, ami megjeleníti azokat. Majd ha a felhasználó reagál a felületen, például elküldi a formot akkor a vezérlő dolgozza fel a felhasználó utasításait, módosítja a modellt, és értesíti a view-t, hogy jelenítse meg a változásokat.



3. ábra: Zend Framework MNV megvalósítása [2]

2.2.4. Zend Framework és a Tábla Átjáró

Zend Framework-ben a Tábla Átjárót a `Zend_Db_Table_Abstract` osztály segítségével lehet megvalósítani. A táblák modelljeit ettől az osztálytól kell származtatni. A Sor Átjáróért a `Zend_Db_Table_Row` felel. Minden adatbázis táblához két darab osztályt kell megírni, egyet a táblaobjektumnak, egyet a sorobjektumainak.

2.3. JavaScript

A JavaScript programozási nyelv egy objektumalapú szkript nyelv, amelyet weblapokon előszeretettel használnak. Én is használom a rendszerben, űrlapok érvényesítésre, lebegő ablakok létrehozására, kereső rendező funkciókkal ruházom fel a táblákat. JavaScript használata azért célszerű, mert sokkal interaktívabbá tehető segítségével a weblap, sokkal inkább az asztali alkalmazás benyomását keltheti.

2.4. Eclipse

Az Eclipse fejlesztői környezetet használtam a PDT kiegészítővel, melynek segítségével az Eclipse ismerte a PHP szintaxist. Jelentősen megkönnyítette a fejlesztést, sokkal átláthatóbb volt a könyvtárszerkezet, a metódusneveket is felajánlotta, így időt is tudtam nyerni.

3. Rendszer felépítése

3.1. Adatbázis

Először szeretném bemutatni azt, hogy hogyan jutottam el a jelenlegi adatbázistervhez. Kezdetben, nagyon minimális adatbázisban gondolkodtam. Még nem is ismertem semmilyen modellező alkalmazást, így egyszerűen egy képszerkesztőben került be a néhány táblából álló adatbázis, de ahogyan születtek az újabb ötletek, úgy bővült az adatbázis és egyre kezelhetetlenebbé vált a dolog. Ezek után egy kísérlet történt az adatbázisterv JDeveloper alkalmazásba való átültetésére. Sajnos nem hozta meg a várt eredményeket, a program nem az egyszerű adatbázisok kezelésére lett kitalálva, és a MySQL-el és a PHP-val sem működött együtt a kívánt módon. Végül a DBDesigner grafikus adatbázis-tervező felület és fejlesztőeszköze esett a választásom.

Az adatbázis a fejlesztés során folyamatosan változott, fejlődött, átalakult. Jelenleg 19 adattáblát tartalmaz, amit öt nagyobb csoportra lehet felosztani. Ezeket, a csoportokat én régióknak nevezem. A teljesség igénye nélkül szeretnék kitérni a régiókra, és a régiókban található táblákra.

3.1.1. Autentikáció (Authentication)

Az autentikáció régió felelős a felhasználó jogainak és adatainak a tárolásáért. Ez a régió öt darab táblát tartalmaz. A következő táblákról van szó: login_user, person, user_group, user_to_group, acl_role.

3.1.1.1. A login_user tábla

Ebben a táblában tárolódnak a felhasználók azonosításához szükséges információk, valamint a rendszerben legutóbb használt beállítások.

login_user		
Mezőnév	Típus	Leírás
login_user_ID	Integer	A tábla elsődleges kulcsa
default_project_ID	Integer	Az alapértelmezett projektnek az azonosítója
default_institute_ID	Integer	Az alapértelmezett intézménynek az azonosítója
username	Varchar (128)	A felhasználó felhasználói neve
passwd	Varchar (128)	A felhasználó jelszava, md5 kódolással kerül tárolásra
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.1.2. A person tábla

A person táblában a regisztrált felhasználó személyes adatai tárolódnak, login_user_ID mező alapján kapcsolódik a felhasználó (login_user) táblához.

person		
Mezőnév	Típus	Leírás
person_ID	Integer	A tábla elsődleges kulcsa
login_user_ID	Integer	Kapcsolat a felhasználó (login_user) táblával
name	Varchar (128)	A felhasználó valódi neve
EHA	Varchar (10)	A felhasználó egységesen használt azonosítója
email	Varchar (128)	A felhasználó e-mail címe
telephone	Varchar (64)	A felhasználó telefon száma
mobile	Varchar (64)	A felhasználó mobil száma
country	Varchar (128)	A felhasználó címének az ország részére vonatkozó adata
postcode	Varchar (10)	A felhasználó címének az irányítószám részére vonatkozó adata
place	Varchar (128)	A felhasználó címének a város részére vonatkozó adata
street	Varchar (128)	A felhasználó címének az utca, házszám részére vonatkozó adata
note	Text	A felhasználóra vonatkozó egyéb információ
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.1.3. A user_group tábla

Ebben a táblában kerülnek tárolásra a felhasználói csoportok. Felhasználó csoportok között definiálható szülő gyermek kapcsolat. Azaz amihez a szülőnek van joga, ahhoz a gyermeknek is lesz.

user_group		
Mezőnév	Típus	Leírás
user_group_ID	Integer	A tábla elsődleges kulcsa
parent_ID	Integer	Ezzel lehet megvalósítani a szülő gyermek kapcsolatot. Ha van a csoportnak szülője, akkor itt a szülő azonosítója szerepel.
institute_ID	Integer	Az intézmények hozhatnak létre saját csoportot. Az intézmény által létrehozott csoport esetén az intézmény (institute) azonosítója szerepel itt
name	Varchar (128)	A csoport neve
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.1.4. A user_to_group tábla

Kapcsolótábla, segítségével összekapcsolódik a login_user és a user_group tábla.

user_to_group		
Mezőnév	Típus	Leírás
login_user_ID	Integer	Külső kulcs a felhasználó (login_user) táblához
user_group_ID	Integer	Külső kulcs a felhasználói csoport (user_group) táblához
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.1.5. Az acl_role tábla

Az acl_role táblával tudunk a csoportnak jogosultságot adni, illetve megtagadni egy Controller-hez (vezérlőhöz) és Action-höz. Ha nincs megadva Action akkor az egész vezérlőre érvényes a jog.

user_group		
Mezőnév	Típus	Leírás
acl_role_ID	Integer	A tábla elsődleges kulcsa
user_group_ID	Integer	Külső kulcs a felhasználói csoporthoz (user_group)
access	Enum ('allow','deny')	Jogosultság, allow-engedélyez, deny-megtilt
controller	Varchar (128)	A vezérlő (controller) neve amire a jogosultság érvényes
action	Varchar (128)	Az action neve amire a jogosultság érvényes
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.2. Intézmény (Institute)

Az intézmény régió felelős az intézmények, projektek adminisztrálásáért, valamint a felhasználókat kapcsolja össze az intézménnyel. Az intézményre azért van szükség, hogy a rendszerben jobban szeparálni tudjuk a tanárokat, hallgatókat. Mivel egy intézményben még így is sok tantárgy, kurzus fordul elő, ezért még egy bontást találtam célszerűnek, ez lett a projekt. A projekteket lehet úgy értelmezni, mint például egy kar esetleg egy tanszék. A következő táblákról van szó: institute, project, user_to_institute.

3.1.2.1. Az institute tábla

Egy intézmény adatai tárolódnak ebben a táblában. Mint említettem az intézményszeparáló szerepet tölt be a felhasználók között.

Institute		
Mezőnév	Típus	Leírás
institute_ID	Integer	A tábla elsődleges kulcsa
name	Varchar (128)	A intézmény valódi neve
email	Varchar (128)	A intézmény e-mail címe
telephone	Varchar (64)	A intézmény telefon száma
fax	Varchar(64)	Az intézmény fax szám

mobile	Varchar (64)	A intézmény mobil száma
country	Varchar (128)	A intézmény címének az ország részére vonatkozó adata
postcode	Varchar (10)	A intézmény címének az irányítószámra részére vonatkozó adata
place	Varchar (128)	A intézmény címének a város részére vonatkozó adata
street	Varchar (128)	A intézmény címének az utca, házsza részére vonatkozó adata
note	Text	A intézményre vonatkozó egyéb adat
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.2.2. A project tábla

A project tábla az intézményen belüli tantárgyak szeparálására szolgál. A projekt szorosan az intézményhez kapcsolódik. Amikor egy felhasználó regisztrál a rendszerbe létrejön számára egy új projekt.

Project		
Mezőnév	Típus	Leírás
project_ID	Integer	A tábla elsődleges kulcsa
institute_ID	Integer	Külső kulcs az intézmény (institute) táblához
name	Varchar (128)	A projekt neve
note	Text	A projekt leírását tartalmazza
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.2.3. A user_to_institute tábla

Kapcsolótábla, segítségével összekapcsolódik a login_user és az institute tábla. Egy felhasználó tetszőleges intézménynek a tagja lehet.

user_to_institute		
Mezőnév	Típus	Leírás
login_user_ID	Integer	Külső kulcs a felhasználó (login_user) táblához
institute_ID	Integer	Külső kulcs az intézmény (institute) táblához
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.3. Tantárgy – Kursus (Subject-Course)

A tantárgy - kurzus régió felelős a tantárgyak, kurzusok adminisztrálásáért, valamint a tanuló felhasználókat kapcsolja össze a kurzussal. A tantárgy alatt egy nagyobb egységet értek, azaz tantárgyat egy intézményben elég egyszer létrehozni, és egy periódusban (például félév) kerülnek meghirdetésre a kurzusok a tantárgyhoz kapcsolódóan. Tehát a kurzus tantárgy nélkül nem létezik, viszont tantárgy előfordulhat lehetséges kurzus nélkül (például az adott félévben nem hirdettek kurzust a tárgyból). A tanuló felhasználó kurzusokra jelentkezik a login_user_to_course tábla létesít kapcsolatot a tanuló és a kurzus között. A következő táblák tartoznak ebbe a régióba: subject, course, login_user_to_course.

3.1.3.1. A subject tábla

A subject táblában a tantárggyal kapcsolatos információk kerülnek tárolásra. A tantárgy összefogja a kurzusokat, így megkönnyítik a felhasználók számára a kurzusok közötti keresést. A tantárgy projekt szintű, azaz egy projektbe tartozó felhasználók látják egymás tárgyait.

subject		
Mezőnév	Típus	Leírás
subject_ID	Integer	A tábla elsődleges kulcsa
project_ID	Integer	Külső kulcs a projekt (project) táblához
name	Varchar(128)	A tantárgy neve
subject_code	Varchar (32)	A tantárgy kódja, ezt a tanár felhasználó szabadon megválaszthatja A keresésnél, adminisztrálásnál hasznos
opened	Enum('true','false')	A tantárgyat lehet aktívvá, vagy inaktívvá tenni, csak aktív tárgy esetén lehet kurzust meghirdetni a tárgy keretein belül
description	Text	A tantárgy leírása
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.3.2. A course tábla

A course táblában a kurzussal kapcsolatos adatok tárolódnak. Egy kurzus szigorúan egy tanár felhasználóhoz kapcsolódik, csak ő láthatja, szerkesztheti. Kurzust csak tantárgy keretein belül lehet meghirdetni.

course		
Mezőnév	Típus	Leírás
course_ID	Integer	A tábla elsődleges kulcsa
subject_ID	Integer	Külső kulcs a tantárgy (subject) táblához, kurzus csak tantárgy keretei között létezhet
login_user_ID	Integer	Külső kulcs a felhasználó (login_user) táblához, minden tanár csak a saját kurzusát látja, szerkesztheti
course_code	Varchar (32)	A kurzus kódja, ezt a tanár felhasználó szabadon megválaszthatja.
name	Varchar (128)	A kurzus neve
term	Varchar (32)	A periódus amikor a kurzus meg van hirdetve, például: 2008/2009 I
headcount	Integer	A kurzusra jelentkező maximális tanulók száma
room	Varchar (128)	A kurzus helye
opened	Enum ('true','false')	A kurzusra való jelentkezés csak akkor lehetséges, ha a kurzus ezen mezőjének az értéke 'true'
description	Text	A kurzusról egyéb információk megadása lehetséges
registry_start	DateTime	A kurzusra jelentkezés kezdő időpontja, ettől a dátumtól tud csak a tanuló jelentkezni a kurzusra
registry_end	DateTime	A kurzusra jelentkezés záró időpontja, eddig a dátumig tud csak a tanuló jelentkezni a kurzusra

date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.3.3. A login_user_to_course tábla

Kapcsolótábla, segítségével összekapcsolódik a login_user és a course tábla. Egy felhasználó tetszőleges számú kurzust vehet fel.

login_user_to_course		
Mezőnév	Típus	Leírás
login_user_ID	Integer	Külső kulcs a felhasználó (login_user) táblához
course_ID	Integer	Külső kulcs a kurzus (course) táblához
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.4. Teszt (Test)

A teszt régióban található meg a vizsgáztató rendszer teszt lapjai és kérdései. A rendszer lehetőséget biztosít kérdés sablonok (question_template) létrehozására, melyekből egyszerűen kérdést (question) lehet készíteni. Egy tesztlaphoz (test) tetszőleges kérdést hozzá lehet rendelni (question_to_test). Tehát négy darab táblából áll a régió ezek: question_template, question, test, question_to_test.

3.1.4.1. A question_template tábla

A tábla a kérdés sablonok tárolására szolgál. A kérdés sablonok projekt szintűek. Maga a sablon egy felparaméterezett HTML forrásként tárolódik, a pontozással együtt.

question_template		
Mezőnév	Típus	Leírás
question_template_ID	Integer	A tábla elsődleges kulcsa
project_ID	Integer	Külső kulcs a projekt (project) táblához
name	Varchar(64)	A sablon neve
template	Text	Maga a sablon, HTML forrása, illetve a pontozás
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.4.2. A question tábla

Ez a tábla tartalmazza a kérdés konkrét szövegét, és a válaszokra adható pontot. A kérdés projektszintű, de csak a létrehozó felhasználója tudja szerkeszteni, használni. A kérdés text mezője voltaképpen egy HTML forrás ebben az esetben is. Kérdés szerkesztésénél tudjuk meghatározni, hogy részpont adható-e.

question		
Mezőnév	Típus	Leírás
question_ID	Integer	A tábla elsődleges kulcsa
login_user_ID	Integer	Külső kulcs a felhasználó (login_user) táblához
project_ID	Integer	Külső kulcs a projekt (project) táblához
name	Varchar(64)	A kérdés neve
subject	Varchar(64)	A kérdéshez rendelhető kérdés tárgya például: geometria
text	Text	Maga a kérdés, HTML forrása
wrong	Enum('true','false')	A kérdést meg lehet jelölni rosszként. Ekkor aki vizsgázott az adott kérdésre maximális pontot kapja
partpoint	Enum('true','false')	Ha részpont adható a kérdésre, akkor az értéke 'true' különben 'false'
answer	Text	A kérdés válaszának kódolása
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.4.3. A test tábla

A test tábla csak általános információkat (text mező) tartalmaz, a tesztel kapcsolatban, ami HTML forrásként kerül tárolásra. A tesztlap projekt szintű, csak a létrehozója láthatja, módosíthatja.

test		
Mezőnév	Típus	Leírás
test_ID	Integer	A tábla elsődleges kulcsa
project_ID	Integer	Külső kulcs a projekt (project) táblához
login_user_ID	Integer	Külső kulcs a felhasználó (login_user) táblához
name	Varchar(64)	A teszt neve
text	Text	A teszt általános információja, HTML forrásban
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.4.4. A question_to_test tábla

Kapcsolótábla, segítségével összekapcsolódik a question és a test tábla. Egy tesztlaphoz tetszőleges számú kérdés tartozhat.

question_to_test		
Mezőnév	Típus	Leírás
question_ID	Integer	Külső kulcs a kérdés (question) táblához
test_ID	Integer	Külső kulcs a teszt (test) táblához
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.5. Vizsga (Exam)

A vizsga régió felelős a vizsgák, adminisztrálásáért, valamint a tanuló felhasználókat kapcsolja össze a vizsgával. A tanár felhasználó meghirdeti a vizsgát (exam tábla), legfeljebb

négy darab tesztlappal (test_to_exam). Egy vizsgára meghatározott számú tanuló felhasználó tud jelentkezni (login_user_to_exam). Amikor egy felhasználó a vizsga során felel egy kérdésre, akkor ez a question_to_login_user_to_exam táblában kerül tárolásra. A következő táblák tartoznak ebbe a régióba: exam, test_to_exam, login_user_to_exam, question_to_login_user_to_exam.

3.1.5.1. Az exam tábla.

Az exam tábla szolgál a vizsgával kapcsolatos információk tárolására. Egy vizsgát kiírhat egy tanár felhasználó. A vizsga kapcsolódhat kurzushoz, de meg lehet kurzus nélkül is hirdetni. A vizsga is projekt szintű és a létrehozója tudja csak módosítani, törölni.

exam		
Mezőnév	Típus	Leírás
exam_ID	Integer	A tábla elsődleges kulcsa
login_user_ID	Integer	Külső kulcs a felhasználó (login_user) táblához
course_ID	Integer	Külső kulcs a kurzus (course) táblához, nem kötelező a megadása. Kurzus keretein kívül is meg lehet vizsgát hirdetni
project_ID	Integer	Külső kulcs a projekt (project) táblához
name	Varchar(128)	A vizsga megnevezése
exam_start	DateTime	A vizsga kezdetének az időpontja
exam_end	DateTime	A vizsga végének az időpontja
headcount	Integer	A vizsgán résztvevők maximális létszáma
room	Varchar(128)	A vizsga helyszíne
description	Text	A vizsga leírása
weight	Integer	Kurzushoz kapcsolva van értelme, ugyanis egy kurzuson belül általában több vizsgát szoktak létrehozni, ekkor a vizsga súlyát jelenti a végelszámolásnál
registry_start	DateTime	A vizsgára való jelentkezésnek a kezdetének az időpontja
registry_end	DateTime	A vizsgára való jelentkezés végének az időpontja
ip_address	Varchar(32)	Az ip tartományt jelent. Ezen tartományon belül lévő gépekről lehet csak megkezdeni a vizsgát
try_number	Integer	Hányszor lehet a vizsgát folytatni, azaz, ha például kifagy a böngésző, vagy a számítógép újraindul, akkor hányszor lehet a vizsgát a félbehagyott helyen folytatni
opened	Enum('true','false')	A vizsgára jelentkezhetőséget szabályozza. Ha 'false' az értéke akkor a vizsgára nem lehet jelentkezni, 'true' érték esetén akkor lehet jelentkezni, ha a jelentkezési időintervallumba esik az aktuális dátum
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.5.2. A test_to_exam tábla

A test_to_exam kapcsolótábla segítségével vizsgához (exam) lehet tesztlapot (test) rendelni. Egy vizsgához legfeljebb négy darab tesztlap rendelhető.

test_to_exam		
Mezőnév	Típus	Leírás
exam_ID	Integer	Külső kulcs a vizsga (exam) táblához
test_ID	Integer	Külső kulcs a test táblához
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.5.3. A login_user_to_exam tábla

Szintén kapcsolótábla, de egyéb fontos információk is tárolásra kerülnek, mint például, az, hogy mikor kezdte meg a vizsgát, illetve hányszor próbálkozott.

login_user_to_exam		
Mezőnév	Típus	Leírás
login_user_to_exam_ID	Integer	A tábla elsődleges kulcsa
login_user_ID	Integer	Külső kulcs a felhasználó (login_user) táblához
exam_ID	Integer	Külső kulcs a vizsga (exam) táblához
exam_start_date	DateTime	A vizsga megkezdésének az időpontja
total_point	Integer	A vizsgán elért pontszám, felesleges lenne minden alkalommal kiszámolni a vizsgán elért pontokat, ezért kerül ez a pontszám itt tárolásra
try_number	Integer	A vizsga hányszor lett elkezdve
valid	Enum('true','false')	A vizsga érvényes-e például nem történt csalás vagy egyéb dolog ami miatt érvényteleníteni kellene a vizsgát.
ip_address	Varchar(32)	A vizsgán használt ip cím
note	Text	A tanár tehet megjegyzést a tanulóhoz.
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.1.5.4. A question_to_test tábla

Kapcsolótábla, segítségével kapcsolódik a felhasználó vizsgajelentkezése (login_user_to_exam) a kérdéshez (question). Ezen kívül még tárolásra kerül a tanuló válasza az adott kérdésre.

question_to_login_user_to_exam		
Mezőnév	Típus	Leírás
question_to_login_user_to_exam_ID	Integer	Külső kulcs az login_user_to_exam táblához
question_ID	Integer	Külső kulcs a kérdés (question) táblához
answer	Text	A kérdésre adott válasz
date_created	DateTime	A rekord táblába kerülésének időpontja
date_updated	DateTime	A rekord módosításának az időpontja

3.2. Működés

Ebben a részben a rendszer programozói szemmel kerül bemutatásra. Mivel a rendszer a keretrendszeren, kívül körülbelül 130 darab PHP és 70 darab JavaScript, CSS állományt tartalmaz, ezért csak az általam érdekesnek, tanulságosnak vélt részekre térek ki. A következő nagyobb csoportokra osztottam fel a rendszer működésbeli bemutatását:

- Könyvtárszerkezet
- Bootstrapping
- Modell osztályok (Tábla Átjárók)
- Vezérlők és nézetek
- Jogosultságkezelés
- Regisztráció
- Rendszer általános működése

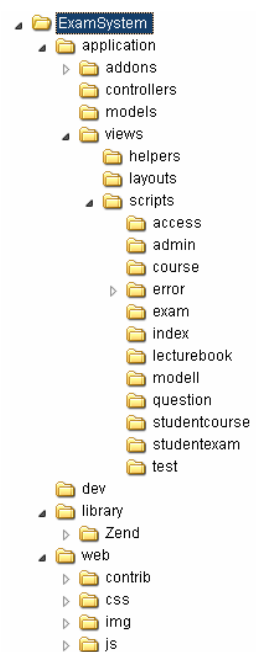
3.2.1. Könyvtárszerkezet

A rendszer eléggé sok könyvtárat tartalmaz, akárcsak a Zend Framework-ben írt alkalmazások. A felbontás segít elkülöníteni a különböző rendszerelemeket.

A rendszer gyökérkönyvtárában az alábbi mappákat találjuk:

- application
- dev
- library
- web

Az application könyvtár tartalmazza az összes szükséges általam megírt PHP kódot, amit a rendszer felhasznál, ez a mappát böngészőn keresztül nem lehet elérni. A controllers mappában találhatóak a vezérlők, models mappa tartalmazza az adatbázis



táblákhoz megírt árjárókat, a scripts könyvtár pedig a nézeteket tartalmazza vezérlő neve szerinti mappa lebontásban. Tehát a scripts könyvtárnak annyi alkönyvtára lesz, ahány vezérlő van a rendszerben.

A dev mappa nem kapcsolódik a Zend keretrendszerhez, az általam a fejlesztéskor használt fájlok találhatóak itt, ilyenek például az adatbázismodell, vagy akár az adatbázist generáló sql fájl, és a rendszer inicializálásához szükséges sql szkript. Nem elérhető böngészőből.

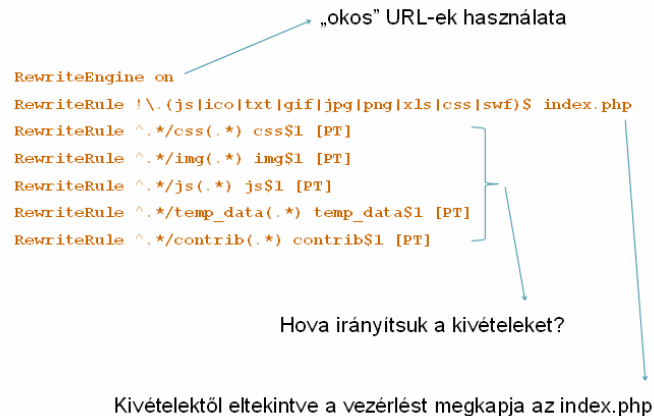
A library könyvtár egyetlen könyvtárat tartalmaz ez a Zend, ez a keretrendszert alkotó PHP fájlok gyűjteménye. Szintén láthatatlan, a böngészőből nem lehet elérni.

A web mappa az egyetlen, amit a böngésző el tud érni. A rendszerben használt képek, stíluslapok, JavaScript-ek, találhatóak itt.

3.2.2. Bootstrapping

A bootstrapping egy kifejezés, amit arra használnak, hogy leírják azt a mechanizmust, ami inicializálja, konfigurálja a rendszert. Tehát úgymond működésre kész állapotba hozza a programot. A Zend-ben általában egy bootstrap állományt használnak. A bootstrap állomány szerepe, hogy betöltse a keretrendszert az alkalmazás számára, és elindítsa a programot. Ahogy említettem a böngészőből, csak a web könyvtár elérhető, itt található az index.php. Az index.php is elláthatná a bootstrap állomány szerepét, viszont ettől logikusabb, hogy ha az application mappába kerül egy bootstrap.php fájlba ugyanis így csak az application mappában található Zend Framework-ös kód, és így sehol másol nincs a szerkezetben. Tehát a web mappában lévő index.php nem tartalmaz mást, mint egy `include('../application/bootstrap.php')` sort. Mielőtt rátérnék a bootstrap fájl felépítésére, szeretném megemlíteni, hogy lehet azt megoldani, hogy ne lehessen elérni a web mappán kívüli könyvtárak elemeit.

A web könyvtárban található egy .htaccess állomány segítségével a webszervert lehet konfigurálni.



4. ábra: A .htaccess állomány felépítése

A „RewriteEngine on” utasítás segítségével a webszerver csak a RewriteRule szabályoknak megfelelő címeket engedi csak át. A PHP állományok eleve nem tölthetők le ugyan, sőt más kiterjesztés is előfordulhat, a fentiek ezeket is levédik, így a rendszer teljes forrása védett és láthatatlan lesz a böngészők számára.

A bootstrap.php fájl négy fontosabb részre tagolható

1. Elsőnek a rendszert alkotó mappák linkjei kerülnek beállításra a PHP számára, a könnyebb kezelhetőség érdekében.
2. A Zend_Loader osztály kerül betöltésre, majd a registerAutoload() statikus metódus kerül meghívásra. Ez az egyetlen require függvényünk a kódban (speciális eseteket kivéve). Minden osztályt egyszerűen példányosítani kell, a loader elintézi helyettünk a többi. Ami fontos, hogy nem az egész keretrendszert tölti be a loader, mindig csak a szükséges állományokat.
3. Ezek után következnek az esetleges Zend_Config (konfigurációs állomány beolvasása, például az adatbázis kapcsolatot tartalmazó információk kiolvasása), Zend_Db (adatbáziskapcsolat létrehozása), Zend_Registry és Zend_Session (session-kezelés) komponensek beállításai.
4. Végül elindul az alkalmazás a Front Controller segítségével:

```

$controller = Zend_Controller_Front::getInstance();
$controller->addControllerDirectory($_PATH.'controller')
->throwException(true)
->dispatch();

```

A Front Controller fogja eldönteni, hogy melyik vezérlő melyik action-jének kell végrehajtódnia.

URL-ek lekezeléséért a Zend Framework lesz felelős az url alakja:
<http://examsystem.hu/controllernev/actionnev/parameter1/ertek1/parameter2/ertek2...>

3.2.3. Modell osztályok (Tábla Átjárók)

Az application könyvtár models akönyvtárában találhatóak a modell osztályok, azaz a Tábla Átjárók. Amint korábban arról már szó volt az adatbázis minden táblájához tartozik tábla és sor átjáró. Egy példán keresztül szeretném bemutatni az átjárók megvalósítását. Legyen a tábla az exam. Az általam használt név konvenció az, hogy minden a táblanéven található szavakat nagybetűvel kezdem. Az exam táblához tartozó tábla átjáró az Exams a sor átjáró pedig az Exam. Tehát a tábla átjáró kap egy „s” vagy „es” végződést ami a többes számra utal. A Zend Framework átjáró megvalósításáról már esett szó, így következzen itt egy részlet az átjárókból:

Exams.php	Exam.php
<pre> <?php /** * Implementation of table "exam" * * @package ModelTable */ class Exams extends addons_Super_Table { /** * @access protected * @var string \$_name Name of the table */ protected \$_name = 'exam'; /** * @access protected * @var string \$_rowClass Name of the row * object */ protected \$_rowClass = 'Exam'; /** * @access protected * @var array \$_dependentTables dependent * tables */ protected \$_dependentTables = array('Courses', 'LoginUser'); ... ?> </pre>	<pre> <?php /** * This class implements the row of "exam" * table * * @package ModelRow */ class Exam extends addons_Super_Record { /** * Retrieve row exam_ID field value * * @return exam_ID column value * @access public */ public function getExamID() { return \$this->exam_ID; } /** * Set row exam_ID value * @param mixed \$value The value for the * exam_ID. * @access public */ public function setExamID(\$value) { \$this->exam_ID = \$value; } /** * Retrieve row course_ID field value * * @return course_ID column value * @access public */ public function getCourseID() { return \$this->course_ID; } ... ?> </pre>

3.2.4. Vezérlők és nézetek

A vezérlők az application könyvtár controllers alkönyvtárában helyezkednek el. A keretrendszer megköveteli, hogy legalább egyetlen vezérlő legyen (IndexController) a vezérlőhöz tartozzon legalább egy action (indexAction). A nézetek az application/views/scripts helyen találhatóak, itt minden egyes vezérlőhöz található egy könyvtár, ami megegyezik a vezérlő nevével. A fentiekből következik, hogy egy könyvtár mindig van (index). A scripts mappában található könyvtárakban általában annyi fájl található, ahány action-je van a controllernek, ezek lesznek a nézetek (view szkriptek). A nézetek phtml fájlok (például: index.phtml), vagyis PHP inserteket tartalmazó HTML oldal. A vezérlőben található action-ökből változókat a „`$this->view->valtozonev = $var;`” értékadással lehet megoldani. A view szkriptben ezt a változót „`<?=$this->valtozonev?>`” módon lehet elérni.

3.2.5. Jogosultságkezelés

A jogosultságkezelés két részre bontható:

- felhasználó azonosítása ezért a Zend_Auth komponens felel,
- jogosultságrendszer megadása, azaz hogy mely vezérlő mely action-jét van joga a felhasználónak elérni, ezzel a Zend_Acl foglalkozik.

A vizsgáztató rendszerben a felhasználók azonosításával az AuthController foglalkozik. A felhasználói jogok az acl_role adatbázistáblában vannak letárolva. Mivel eléggé összetett lenne az egész folyamat leírása és megértése, ezért csak vázlatosan a lényeget kiemelve szeretnék az autentikációra kitérni.

A Zend_Auth csak a felhasználók azonosításával foglalkozik, ehhez a komponensnek szükséges egy „bizonyítvány” (credential). A Zend_Auth osztály egy Singleton mintát megvalósító osztály, csak egyetlen példánya létezik, és ezt statikusan lehet elérni a getInstance() metódussal. Szükséges lesz a login_user táblára ahol a felhasználói nevek és a jelszavak kerülnek md5 kódolással tárolásra. Kell még egy adapter (\$authAdapter), mellyel azonosításhoz szükséges adatok kerülnek beállításra

```
$authAdapter = new Zend_Auth_Adapter_DbTable($dbAdapter);  
$authAdapter->setTableName('login_user');  
$authAdapter->setIdentityColumn('username');  
$authAdapter->setCredentialColumn('passwd');
```

A \$dbAdapter az adatbáziskapcsolatért felel. Ezután bejelentkező oldalról a felhasználó által megadott felhasználónév és jelszó kerül megadásra:

```
$authAdapter->setIdentity($formData['username']);  
$authAdapter->setCredential(md5($formData['password']));
```

Végül el kell végezni a tényleges autentikációt a Zend_Auth segítségével:

```
$auth = Zend_Auth::getInstance();  
$result = $auth->authenticate($authAdapter);
```

Ha a result értéke igaz, akkor sikeres volt a bejelentkezés. A Zend_Auth példány megjegyzi a bejelentkezést, és ezután nem kell vele foglalkozni. A felhasználót, kijelentkezését egy sorral meg lehet oldani:

```
Zend_Auth::getInstance()->clearIdentity();
```

Jogosultságkezelés szempontjából kétféle objektumot lehet megkülönböztetni:

1. Resource - Zend_Acl_Resource (erőforrások, amikhez jogosultságok állíthatók be, tehát olyan objektumok, melyekhez a hozzáférést lehet megadni, korlátozni)
2. Role - Zend_Acl_Role (olyan objektumok, amik igénylik a jogokat például a felhasználók)

Érdemes az egyes controllereket, action-öket Resource-nak választani. A Resource-ok hierarchikusan tárolhatók le, néhány kivétel adódhat, ezeket is meg lehet oldani. A Role-ok is hasonlóképp jöhetnek létre, mint a Resource-ok, azaz szülő-gyermek kapcsolat alakítható ki, így amihez a szülőnek volt joga, a gyermeknek is lesz joga. Mivel a jogosultságok adatbázisban vannak tárolva, ez még jobban megkönnyíti a használatát ezért egy egyszerűbb példán keresztül illusztrálom a lényegét:

```
$acl = new Zend_Acl();  
$acl->addRole(new Zend_Acl_Role('guest'))  
->addRole(new Zend_Acl_Role('student'))  
->addRole(new Zend_Acl_Role('admin'));  
$parents = array('guest', 'student');  
$acl->addRole(new Zend_Acl_Role('teacher'), $parents);  
$acl->add(new Zend_Acl_Resource('someResource'));  
$acl->deny('guest', 'someResource');  
$acl->allow('student', 'someResource'); ...
```

A rendszerben jelenleg négy darab szerepkör van, ezek az: admin, guest, teacher, student. Regisztrációkor dől el, hogy a felhasználó tanár, vagy tanuló szerepkörrel fogja majd a rendszert használni. A rendszert további funkcióval lehetne bővíteni, ha egy intézmény szerepkör kerülne be, valamint a hozzá tartozó felület is megvalósulna.

3.2.6. Regisztráció

A regisztráció egy egyszerű folyamat, ugyanis ez nem áll másból, mint három darab táblát fel kell tölteni adatokkal. A regisztrációért az access vezérlő, register action-je a felelős. A regisztrációs űrlap kitöltése után az action-ben beszúr egy-egy sort:

- a login_user táblába, ahol a felhasználói nevek tárolódnak
- a person táblába, amiben a személyes adatok kerülnek
- user_has_group táblába, amivel a felhasználó kapcsolódik egy felhasználói csoporthoz és így megkapja azokat a jogokat, amelyek megilletik.

3.2.7. A rendszer általános működése

A rendszer működését tekintve három nagyobb csoportba sorolható: regisztráció bejelentkezés, tanár vagy diák felhasználói felülete. A regisztrációt, már korábban bemutattam, így ezzel most nem foglalkozok. A vizsgáztató rendszert a tanár fogja többet látni, illetve neki sokkal több tennivalója van, mint egy diáknak, így a tanár szemszögéből az elérhető funkciók sokkal gazdagabbak. Igyekeztem nagyobb egységekre bontani az elérhető funkciókat, és ezekhez hozzárendelni egy-egy menüpontot. Egy almenüt kiválasztva (például a tanár felhasználóval bejelentkezve a kurzust) egy az adott fogalomhoz (kurzus) kapcsolódóan jelenik meg legalább két darab lapfűl (Kurzusok listája, Új kurzus létrehozása). Az egyik fűl a listázást, a másik pedig az új létrehozását jelenti. Ezek a lapfűlek általában egy vezérlőn belüli különböző action-öket jelentenek. Az új elem létrehozása, és az elem szerkesztése úgy vannak megoldva, hogy ugyanazon action végzi el a műveletet, így nem kellett kétszer ugyanazt megírni, csak kezdőértéket kell adni az űrlapon lévő elemeknek. Már említettem hogy rengeteg PHP fájlból áll a rendszer, ezért csak egy tipikus példán keresztül szemléltetném azt, ahogyan egy listázás történik. A rendszerben nyilván vannak olyan oldalak, amelyek nem illeszkednek bele ebbe a mintába, de ezek összetettségük, bonyolultságuk miatt nem férnek el a szakdolgozat keretei között. A tantárgyak listázását szeretném bemutatni.

Az egyik legegyszerűbb listázás az a tantárgyak listázása, mégis jól látszanak a keretrendszer előnyei. A tantárgyak listázásáért a Course vezérlő (CourseController.php) listsubjects action-je (listsubjectAction) felel. Az action-höz a listsubject.phtml nézet tartozik. Tehát nézzük a vezérlő, action-jét először:

<pre> public function listsSubjectsAction() { \$subjectModell = new Subjects(); \$namespace = new Zend_Session_Namespace(); \$subjRows= \$subjectModell->fetchAll(\$subjectModell->select()-> where("project_ID = ?", \$namespace->project)); if(\$subjRows->current()){ \$settings = array(array(name=>"Név", sort=>"default", filter=>"false", search=>"true"), array(name=>"Tantárgy kód", sort=>"default", filter=>"false", search=>"true"), array(name=>"Státusz", sort=>"default", filter=>"true", search=>"false"), array(name=>"Létrehozva", sort=>"date", filter=>"false", search=>"true"), array(name=>"Szerkesztés", sort=>"false", filter=>"false", search=>"false"), array(name=>"Törlés", sort=>"false", filter=>"false", search=>"false")); \$JsTable = new addons_JsTable(\$this->view, Zend_Registry::get('RPath').'/js/JsTable.js',\$settings); \$this->view->JsTable = \$JsTable; \$this->view->tableRows = \$subjRows; } } </pre>	A metódus specifikációja A subject tábla Tábla Átjárójának a példányosítása A namespace az a PHP session megfelelője Az aktuális projekthez tartozó rekordokat kérdezi le, ami Sor Átjáró halmazzal tér vissza. Ha van elem a halmazban, akkor a listázó táblázat beállításait tartalmazó tömb kerül feltöltésre. A táblázat a view szkriptben kerül majd kiírásra, ami egy egyszerű HTML táblázat lesz. A táblázatot kereső, rendező, szűrő funkciók megvalósítása egy JavaScript segítségével történik. A JavaScript ami, a táblázatot átalakítja egy segéd PHP szkripttel kerül beállításra. Ez a PHP szkript kerül itt példányosításra. A táblázat megjelenítését segítő PHP osztály kerül átadásra a view szkriptnek. A táblázat sorai kerülnek átadásra a view szkriptnek.
--	---

Az action-höz tartozó view szkript túlságosan hosszú, így csak egy része kerül bemutatásra:

<pre> ... <?if(\$this->JsTable) :?> <table id="myJsTable" ..."> <thead> <?=\$this->JsTable->getTableHeader() ?> </thead> <tbody> <? foreach(\$this->tableRows as \$row) : ?> <tr class="<?php if(\$currRow++%2==0) echo \$evenRowClass; else echo \$oddRowClass; ?>"> <td><?=\$row->getName();?></td> <td><?=\$row->getSubjectCode();?></td> <td><?=\$row->getOpenedStatus();?></td> ... </tr> <? endforeach; ?> </tbody> <tfoot> <?=\$this->JsTable->getTableFooter() ?> </tfoot> </table> <?endif; ?> ... </pre>	Az action-ben átadott tábla létezik-e, azaz van-e a táblázatnak sora. Ha igen, akkor a táblázat bevezető tagje kerül kiírásra. A táblázat fej részét bevezető nyitó tag A segéd PHP osztállyal a táblázat fejléce kerül ki. A táblázat fejének záró része A táblázat testének nyitó tagja A táblázat sorain végighaladó foreach ciklus kezdete A sorok felváltva más css class-t kapnak, így más színűek lesznek a táblázat sorai. A táblázat mezői a Sor Átjáró megfelelő metódusának meghívásával kerülnek meghívásra. A táblázat sorainak lezárása A foreach ciklus lezárása A táblázat testének lezárása A táblázat lábléce kezdő tagjának a beállítása A segéd PHP osztállyal a táblázat lábléce kerül ki. A táblázat lábléce záró tagjának a beállítása. Táblázat lezárása. Feltételes utasítás lezárása
---	---

Ezen a „rövid” példán is látszik az, hogy átlátható, világos, szeparált, egyszerűen bővíthető kódot eredményez a keretrendszer elveinek megfelelő kódolás.

4. Más rendszerek megvizsgálása

A fejezet célja bemutatni, összehasonlítani néhány jelenleg a szoftverpiacon elérhető vizsgáztató rendszert. A termékek palettája igen sokrétű, az ingyenesről a fizetősig, az egyszerűtől az összetettig, minden fajtát meg lehet találni. Ezért kiválasztottam egy ingyenes és egy fizetős szoftvert.

4.1. EVR (Elektronikus Vizsgáztató Rendszer)

Az Elektronikus Vizsgáztató Rendszer célja, hogy megkönnyítse az oktatók munkáját, illetve leegyszerűsítse és felgyorsítsa a diákok számonkérését. Különösen nagy segítség lehet sok hallgató egyidejű teszt alapú vizsgáztatása esetén [4].

Az oktató előállítja a tetszőlegesen nagy feladathalmazt - a tesztet - amelyből a rendszer vizsgákat fog létrehozni minden bejelentkezőnek. A hallgatók az EVR-ben előre megadott számítógépekről és az előre megadott időben tudnak vizsgát kérni. A munkájuk időről időre elmentésre kerül és kiértékelődik a szerveren, ami folyamatos követhetőséget és nagy rugalmasságot eredményez. Az oktató eközben figyelemmel kísérheti a diákok tevékenységét és tudhatja, hogy azok milyen eredménnyel töltötték ki a vizsgáikat. A rendszert egy szerver oldali és három kliens oldali komponens alkotja: EVR szerver; EVR Tesztszerkesztő kliens; EVR Vizsgáztató kliens; EVR Nézegető.

Az EVR egy jól felépített, átgondolt rendszer. A klienseket Java programozási nyelven írták, így ez platform független megoldás, de azért lehet érezni azt, hogy nem az igazi. Ez alatt azt értem, hogy először le kell tölteni a klienseket, és fel kell őket telepíteni. Ha például 50 gépre fel, kell telepíteni a vizsgáztató klienst akkor az időtényezőn kívül a szoftver függőségi viszonyok megoldása is problémás. Az EVR nem tud intézményt, projektet, kérdéssablont, kurzust kezelni. Az EVR nem alkalmas egy kurzus létrehozására, csak egyszeri vizsgáztatásra, így nem lehet statisztikai, összesítő adatokat megtekinteni. Az ExamSystem negatívuma, szöveges válaszlehetőséggel ellátott kérdéstípust nem lehet létrehozni, viszont az EVR képes erre. Az utolsó, az hogy az EVR használata sajnos nem ingyenes.

4.2. Moodle

A Moodle egy nyílt forráskódú e-Learning keretrendszer (LMS), amely flexibilitásának és a korszerű pedagógiai módszerek támogatásának köszönhetően világszerte egyre népszerűbb [5]. A program neve a Modular Object-Oriented Dynamic Learning Environment kifejezés

rövidítéséből származik. A program szintén PHP nyelvben íródott, és sokan használják. Ez a szoftver, sokkal összetettebb, mint az ExamSystem. Rengeteg funkcióval rendelkezik, és hasonlóan rendelkezik vizsgáztató résszel. De a kérdés összeállító, vizsgáztató része az ExamSystem-nek erősebb, ugyanis az egész rendszer erre van kihegyezve. Bár a Moodle is több kérdés típussal rendelkezik, de az ExamSystem ezen irányú bővítése tervben van.

5. Rendszer bemutatása

5.1. Adminisztráció

Ahogy ellátogat az ExamSystem honlapjára a nyitóoldalt, láthatja. Az üdvözlő képernyőn információt olvashat a szoftver funkcióiról, és átfogó képet kaphat a rendszerszolgáltatásiról. Az egyszerű és letisztult menürendszeren még a gyakorlatlan felhasználó is könnyen eligazodhat.

A baloldali menürendszerben a sötét háttérrel jelölt menük a főmenük (a szövegben, továbbiakban félkövér aláhúzott betűvel szedve), ezekre kattintva előhívhatja az almenüpontokat (a szövegben, továbbiakban félkövéren szedve), melyek a következők:

Nyitólap

Nyitólap

Elérhetőségek

Hozzáférés

Bejelentkezés

Regisztráció

5.1.1. Nyitólap

A **Nyitólap** főmenüponttal a nyitólaphoz tartozó almenüpontokat hívhatja elő.

A **Nyitólap** almenüponttal a bemutatkozó oldal ismételt megtekintését érheti el.

Ha az **Elérhetőségek** almenüpontra kattint, akkor megtekintheti az oldal üzemeltetőinek az elérhetőségi címeit, ezáltal segítve a rendszer karbantartóival való könnyed kapcsolatfelvételt.

5.1.2. Hozzáférés

A **Hozzáférés** főmenüpont a rendszerbe történő bejelentkezés és regisztrációs almenüpont megtekintését segíti elő.

A **Bejelentkezés** almenüpont alatt tudja a rendszer a már korábban regisztrált tagokat bejelentkeztetni, a *Felhasználói név* és *Jelszó* megadásával. Ha Ön már regisztrált felhasználó, akkor itt tud a rendszerbe belépni.

A **Regisztráció** almenüpont a rendszerhez csatlakozni kívánó személyek felvételét segíti egy regisztrációs űrlap kitöltésével, melyek a következő mezőkből tevődnek össze:

*Felhasználói név:	teszt
*Jelszó	••••••
*Jelszó ismét	••••••
*Szerep	tanuló ▾
*Intézmény	DE ▾
*Név	Teszt Elek
EHA	AZON01
E-mail	teszt@elek.hu
Mobil	0620123456
Telefon	0652123456
Ország	Magyarország
Irányítószám	3212
Helység	Debrecen
Utca, házszám	Egyetem sqt 45
Regisztrál	

5. ábra: Kitöltött regisztrációs űrlap

A *-al jelölt mezők kitöltése kötelező, a többi opcionális.

A *Felhasználói név* megadása kötelező, hisz a későbbiekben ezzel tud belépni a rendszerbe.

A *Jelszó* minimális hossza 6 karakter, melynek megadása kötelező.

A *Jelszó ismét* mező funkciója, az esetlegesen hibásan megadott jelszavak kiszűrése.

A *Szerep* legördülő lista segítségével tudja megadni, hogy diákként vagy tanárként szeretne-e Ön regisztrálni a rendszerbe.

Ha az *Intézmény* legördülő listában nem szerepel az Ön által megjelölt intézmény, akkor vegye fel velünk a kapcsolatot, és mi felvesszük őket a listába.

A *Név* mezőben az Ön valódi nevét adja meg. A további mezők kitöltése nem kötelező, de ajánlott.

Az *EHA* (Egységes Hallgatói Azonosító) mezőben megadhatja az Ön intézményében használt hallgatói azonosítót.

Az *E-mail*, *Mobil*, *Telefon*, *Ország*, *Irányítószám*, *Helység*, *Utca*, *házszám* mezőkben a kapcsolattartási elérhetőségeket adhatja meg.

A *Regisztrál* gombbal, véglegesítheti a regisztrációját.

5.2. Tanár

A tanár felhasználóval történő belépés után egy üdvözlő képernyő fogadja, azaz a személyes kezdőlapja, ahol hasznos információkat talál a rendszerrel kapcsolatban. A bal oldali menüsor segítségével navigálhat az oldalon. A menüsor a következő elemekből áll:

Kezdőlap

Kezdőlap

Elérhetőségek

Kurzus

Tantárgy

Kurzus

Teszt

Kérdéssablon

Kérdés

Tesztlap

Vizsga

Vizsga

Adminisztráció

Intézmény

Projekt

Hozzáférés

Adatmódosítás

Kijelentkezés

5.2.1. Kezdőlap

A **Kezdőlap** főmenüpontnak két darab almenüpontja van ezek a **Kezdőlap**, és az **Elérhetőségek**.

A **Kezdőlap** almenüpont alatt éri el a személyes kezdőlapját, itt hasznos információt talál a rendszer működéséről.

Az **Elérhetőségek** almenüpontra kattintva, megtekintheti az oldal üzemeltetőinek az elérhetőségi címeit.

5.2.2. Hozzáférés

A **Hozzáférés** főmenüpont alatt két darab menü pontot talál, ezek az **Adatmódosítás** és a **Kijelentkezés**.

Az **Adatmódosítás** alatt tudja módosítani a regisztrációkor megadott adatait, illetve jelszavát. Az alábbi adatait tudja megváltoztatni:

A *Régi Jelszó* mező kitöltése akkor kötelező, ha szeretné megváltoztatni a jelszavát.

Jelszó az új jelszó megadása, legközelebb ezzel a jelszóval tud majd bejelentkezni a rendszerbe

Jelszó ismét mező funkciója, az esetlegesen hibásan megadott jelszavak kiszűrése.

A *Név* mezőben az Ön valódi nevét adja meg. A további mezők kitöltése nem kötelező, de ajánlott.

Az *EHA* (Egységes Hallgatói Azonosító) mezőben megadhatja az Ön intézményében használt hallgatói azonosítót.

Az *E-mail*, *Mobil*, *Telefon*, *Ország*, *Irányítószám*, *Helység*, *Utca*, *házzám* mezőkben a kapcsolattartási elérhetőségeket adhatja meg.

A *Regisztrál* gombbal, véglegesítheti a kívánt változtatásokat.

A **Kijelentkezés** menüpontra kattintva kilép a rendszerből, és visszakerül a **Bejelentkezés** oldalra

5.2.3. **Kurzus**

A **Kurzus** főmenüpont két almenüpontot tartalmaz ezek a **Tantárgy** és a **Kurzus**. Tantárgy létrehozására azért van szükség, hogy jobban el tudja különíteni kurzusait. Tehát ha, van olyan kurzusa, amelyet félévenként (esetleg évenként) újrahirdet, akkor érdemes ezeket egy tantárgy keretein belül megtenni. Vegyük példaként egy „HTML szerkesztés” nevű kurzust, megszeretné hirdetni, de sejti, hogy nem csak egyszeri lesz meghirdetve, akkor érdemes egy „HTML szerkesztés” tantárgyat készíteni. A kurzus meghirdetésénél kiválasztani, hogy a „HTML szerkesztés” tantárgy keretei között van meghirdetve HTML 2009/2010 tavasz néven. Kurzus létrehozása akkor célszerű, ha több darab vizsgát szeretne meghirdetni egy adott témában. Kurzus segítségével nyomon tudja követni hallgatói előrehaladását a leckekönyv segítségével.

Tantárgy almenüpontra kattintva a Tantárgyak listája oldal jelenik meg, amelyen két darab lapfül található. A két lapfül a Tantárgyak listája és az Új tantárgy létrehozása. Alapértelmezetten a Tantárgyak listája lapfül be van nyomva.

A Tantárgyak listája oldalon egy táblázat jelenik meg amely a projektben korábban létrehozott tantárgyakat tartalmazza. Ha még nem hozott létre egyet sem akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket,

figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetlegesen szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a tantárgy neve, amit létrehozáskor megadott
- Tantárgy kód – a tantárgy kódja, amit szerkesztéskor adott meg
- Státusz – két értéket vehet fel: aktív, és inaktív jelzi, hogy tantárgy keretein belül lehet-e kurzust meghirdetni
- Létrehozva – a tantárgy léthozásának az időpontja
- Szerkesztés – a tantárgyat szerkesztheti vele
- Törlés – a tantárgyat törölheti, ha még nem hirdetett meg kurzust a tantárgy keretein belül

Az Új tantárgy létrehozása lapfűlön tud új tantárgyat felvenni. Új tantárgy felvételéhez az alábbi adatokat kell megadnia:

Név, azaz a tantárgy neve, megadása kötelező.

A *Tantárgy kód* megadása nem kötelező, de általában minden intézményben van valamilyen kódja a tantárgyaknak, így megadása ajánlott, már csak a keresés megkönnyítése miatt is.

A *Státusz* megadásával szabhatja meg, hogy lehetséges-e a tantárgy keretien belül kurzust meghirdetni.

Leírás mező megadásával egyéb a tantárggyal kapcsolatos információt adhat meg.

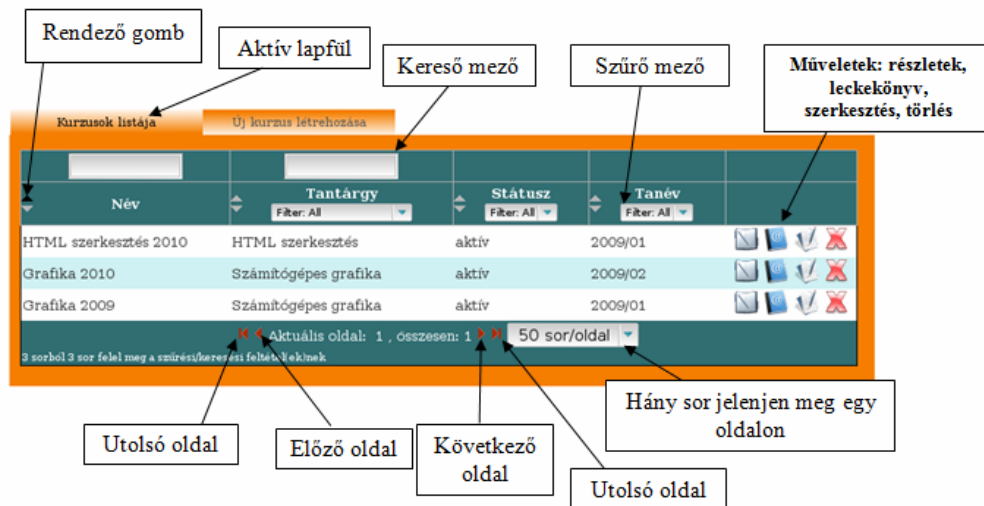
A *Mentés* gomb megnyomásával mentheti el a tantárgyat.

Kurzus almenüpontra kattintva a Kurzusok listája oldal jelenik meg, amelyen két darab lapfűl található. A két lapfűl a Kurzusok listája és az Új kurzus létrehozása. Alapértelmezetten a Kurzusok listája lapfűl be van nyomva.

A Kurzusok listája oldalon egy táblázat jelenik meg amely a projektben Ön által korábban létrehozott kurzusokat tartalmazza. Ha még nem hozott létre egyet sem akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetlegesen szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a kurzus neve, amit létrehozáskor megadott
- Tantárgy kód – a tantárgy kódja, amit szerkesztéskor választott ki, ha nem adta meg, akkor üres lesz a mező

- Státusz – két értéket vehet fel: aktív, és inaktív jelzi, hogy kurzusra a hallgatók jelentkezhetnek-e
- Tanév – a tanév, amelyben a kurzus aktív, létrehozáskor adhatja meg
- Részletek – a kurzussal kapcsolatos további információkat tekintheti meg egy új lapfűlön
- Leckekönyv – a kurzust hallgatók listája illetve a vizsgák eredményeit tekintheti meg
- Szerkesztés – a kurzust szerkesztheti vele
- Törlés – a kurzust törölheti vele, ha még nem jelentkezett egyetlen hallgató sem



6. ábra: Kurzusok listája

Az Új kurzus létrehozása lapfűlön tud új kurzust felvenni. Új kurzus felvételéhez az alábbi adatokat kell megadnia:

Név, azaz a kurzus neve, megadása kötelező.

A *Tantárgy* megadása nem kötelező, de általában kurzust egy tantárgy keretein belül hirdetnek meg, így megadása ajánlott, már csak a keresés megkönnyítése miatt is.

A *Státusz* megadásával szabhatja meg, hogy lehetséges-e a kurzusra jelentkezés.

A *Létszám* mező megadása kötelező, segítségével megszabhatja azt, hogy hány hallgató jelentkezhet a kurzusra.

Terem a kurzus helye, tehát ahol a kurzust hallgatni lehet.

A *Jelentkezés kezdete* mező segítségével adhatja meg, hogy a tanulók mely időponttól jelentkezhetnek a kurzusra. Az időpont megadásának, megkönnyítését egy a mezőtől balra

levő naptár gomb segíti, ha rákattint, akkor egy felugró naptár jelenik meg, ahol könnyedén megadhatja a dátumot órára pontosan.

A *Jelentkezés vége* mező segítségével adhatja meg hogy a tanulók mely időpontig jelentkezhetnek a kurzusra. Az időpont megadásának, megkönnyítését egy a mezőtől balra levő naptár gomb segíti, ha rákattint, akkor egy felugró naptár jelenik meg ahol, könnyedén megadhatja a dátumot órára pontosan.

Leírás mező megadásával egyéb a kurzussal kapcsolatos információt adhat meg.

A *Mentés* gomb megnyomásával mentheti el a kurzust.

5.2.4. Teszt

A **Teszt** főmenüpont három almenüpontot tartalmaz ezek a **Kérdéssablon**, **Kérdés** és a **Tesztlap**. Kérdéssablon segítségével tudja meghatározni a kérdéseinek a kinézetét, típusát illetve a pontozást is itt tudja megadni. Használja a kérdéssablonok adta lehetőségeket arra, hogy minél tetszetősebb, változatosabb kérdéseket tudjon létrehozni. Kérdés szerkesztésekor a létrehozott kérdéssablonokat tudja betölteni, majd létrehozni az új kérdést, és megszabni azt, hogy adható-e részpont a kérdésre. Kérdés létrehozásánál használja a kategória megadását, hogy később könnyebben meg tudja keresni az egy csoportba tartozó kérdéseket. Ha már létrehozta kérdéseit, akkor hozhat létre tesztlapokat, ahol egy általános leírás után összeválogathatja a teszthez tartozó kérdéseket.

Kérdéssablon almenüpontra kattintva a Sablonok listája oldal jelenik meg, amelyen két darab lapfül található. A két lapfül a Sablonok listája és az Új sablon létrehozása. Alapértelmezetten a Sablonok listája lapfül be van nyomva.

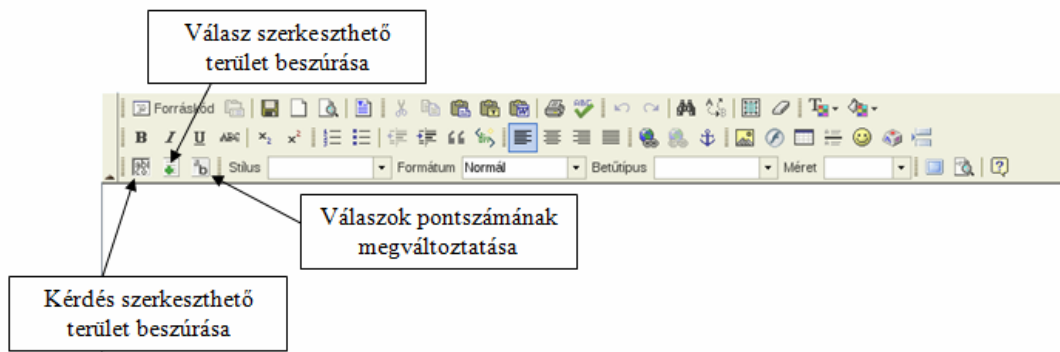
A Sablonok listája oldalon egy táblázat jelenik meg amely a projektben Ön által korábban létrehozott sablonokat tartalmazza. Ha még nem hozott létre egyet sem akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a sablon neve, amit létrehozáskor megadott
- Létrehozás időpontja – az a dátum, amikor a sablont létrehozta
- Szerkesztés – a sablont szerkesztheti vele
- Törlés – a sablont törölheti vele

Az Új sablon létrehozása lapfülön tud új sablont létrehozni. Új sablon létrehozásához az alábbi adatokat kell megadnia:

Név, azaz a sablon neve, megadása kötelező.

A név mező alatt megjelenő WYSIWYG („What You See Is What You Get”, azaz „amit láatsz, azt kapod”) editor segítségével tudja megszerkeszteni sablonját.



7. ábra: WYSIWYG szerkesztő

A sablonba használhatóság érdekében úgynevezett szerkeszthető területeket kell beleraknia. Két fajta szerkeszthető területet lehet megkülönböztetni. Szerkeszthető terület lehet a kérdés szövege, ebből egy darabot érdemes elhelyezni. Kérdés szövegét az editor menüjének harmadik sorának első elemével tud elhelyezni. Ha sikerült beszúrní, akkor az editorba bekerül a következőhöz hasonló szöveg: „{{{*A kérdés szövege*}}}". Kérdés létrehozásakor a „{{{*}}” és a „*}}}" közötti részt tudja szerkeszteni. A másik fajta szerkeszthető terület az a lehetséges válasz. Választ az editor menüjének harmadik sorának a második gombjával tud elhelyezni, itt kell megadnia a kérdés pontszámát. A válasz hasonló megjelenésű lesz, azzal a kivétellel, hogy itt megjelenik előtte egy a kérdés típusának megfelelő jelölő rádiógomb, vagy kiválasztó négyzet. Választ tetszőlegesen sokat adhat meg. Az editor harmadik sorának harmadik gombjával a beszúrt válaszok pontszámát tudja megváltoztatni.

A *Többszöri választás*, *Egyszeri választás* rádiógommbal tudja kiválasztani a kérdés típusát. *Többszöri választás* esetén a tanuló több választ is bejelölhet, míg *Egyszeri választáskor* csak egyet tud kijelölni a lehetséges válaszok közül

A *Mentés* gomb megnyomásával mentheti el a sablonját.

Kérdés almenüpontra kattintva a Kérdések listája oldal jelenik meg, amelyen két darab lapfűl található. A két lapfűl a Kérdések listája és az Új kérdés létrehozása. Alapértelmezetten a Kérdések listája lapfűl be van nyomva.

A Kérdések listája oldalon egy táblázat jelenik meg amely a projektben Ön által korábban létrehozott kérdéseket tartalmazza. Ha még nem hozott létre egyet sem akkor a „Jelenleg üres

a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a kérdés neve, amit létrehozáskor megadott
- Kategória – a kérdés mely kategóriába tartozik
- Létrehozás időpontja – az a dátum, amikor a kérdést létrehozta
- Előnézet – a kérdést egy felugró ablakban tekintheti meg
- Szerkesztés – a kérdést szerkesztheti vele
- Törlés – a kérdést törölheti vele

Az Új kérdés létrehozása lapfülön tud új kérdést létrehozni. Új kérdés létrehozásához az alábbi adatokat kell megadnia:

Név, azaz a kérdés neve, megadása kötelező.

A Név mező alatt megjelenő fehér téglalapba (továbbiakban kérdésszerkesztő) tudja a kérdését létrehozni. Ehhez először a terület alatt lévő *Sablonok listájából* be kell töltenie egy sablont. Ha betöltött egy megfelelően elkészített sablont, akkor az megjelenik a kérdésszerkesztőben. A kérdés szövegét úgy tudja szerkeszteni, hogy kattint a „{{{*}}” és a „*}}}}” közötti részre, ekkor megjelenik egy WYSIWYG editor, ahol tudja a kérdését szerkeszteni. A válaszok pontszámát megtekintheti, ha a válaszok előtt álló jelölő négyzetre (körre) ráviszi az egérmutatót, ekkor a kérdés szerkesztő alatt található sárga területen megjelenik a kérdés pontszáma.

8. ábra: Kérdés szerkesztése

A *Részpont adható* jelölőnégyzettel adhatja meg, hogy megengedi-e azt, hogy részpontot kapjon a tanuló a válaszára.

A *Mentés* gomb megnyomásával mentheti el a kérdését.

Teszt almenüpontra kattintva a Tesztlapok listája oldal jelenik meg, amelyen két darab lapfül található. A két lapfül a Tesztlapok listája és az Új Teszt létrehozása. Alapértelmezetten a Tesztlapok listája lapfül be van nyomva.

A Tesztlapok listája oldalon egy táblázat jelenik meg, amely a projektben Ön által korábban létrehozott tesztlapokat tartalmazza. Ha még nem hozott létre egyet sem akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a tesztlap neve, amit létrehozáskor megadott
- Létrehozás időpontja – az a dátum, amikor a kérdést létrehozta
- Kérdések – segítségével a tesztlaphoz tartozó kérdéseket tekintheti meg illetve új kérdéseket vehet fel a tesztlaphoz, bővebb információt az új tesztlap létrehozása részben olvashat.
- Előnézet – a tesztlapot tekintheti meg
- Szerkesztés – a tesztlapot szerkesztheti vele
- Törlés – a tesztlapot törölheti vele

Az Új tesztlap létrehozása lapfülön tud új tesztlapot létrehozni. Új tesztlap létrehozásához az alábbi adatokat kell megadnia:

Név, azaz a tesztlap neve, megadása kötelező.

A Név mező alatt megjelenő WYSIWYG editorba, tudja formázva megadni a tesztlaphoz tartozó általános információkat. Ezt a vizsgázó a vizsga folyamán bármikor megtekintheti.

A *Mentés* gomb megnyomásával mentheti el a kérdését.

A *Mentés és kérdések hozzáadása* gombra kattintva a kérdés mentésre kerül, és egy olyan oldalra kerül át, amelyen két darab lapfül jelenik meg. A két darab lapfül az a Teszt kérdések listája és a Kérdések felvétele. Alapértelmezetten a Kérdések felvétele lap füle be van nyomva. A kérdések felvétele oldalon szerepelnek az Ön által létrehozott kérdések egy táblázatban. Ne felejtse azok a kérdések jelennek itt meg, amelyeket még nem vett fel a tesztlapjához. A táblázat a következő oszlopokat tartalmazza:

- Név – a kérdés neve
- Kategória – a kérdés mely kategóriába tartozik
- Előnézet – a kérdést egy felugró ablakban tekintheti meg
- Kijelöl – azokat a kérdéseket jelölje ki, amelyeket szeretne a tesztlaphoz adni

Az oldalon található *Mentés* gombbal rendelheti hozzá a kijelölt kérdéseket a tesztlapához.

A Teszt kérdések listája lapfűlön a tesztlaphoz már hozzárendelt kérdései szerepelnek táblázatos formában. A táblázat felépítése megegyezik a Kérdések felvétele lapfűlön lévővel, csak az utolsó oszlop különbözik. Az utolsó oszlopban egy Törlés gomb található, segítségével eltávolíthatja a kérdést a tesztlap kérdései közül.

5.2.5. Vizsga

A **Vizsga** főmenüpont egy almenüpontot tartalmaz ez a **Vizsga**. Ez alatt a menüpont alatt tud új vizsgát meghirdetni, és a meglévőket megtekinteni, valamint a vizsga eredményei is elérhetőek. Egy vizsgához legfeljebb négy darab tesztlap rendelhető. A vizsgához rendelt tesztlapok véletlenszerűen kerülnek kiosztásra a vizsgázó tanulók között. A tesztlapon belüli kérdések is véletlenszerűen kerülnek feltevéssre. Hogy a tanulók ne jegyezzék meg a kérdésekre adható válaszok sorrendjét ezért még a kérdésekre adható válaszok is véletlenszerű sorrendben jelennek meg.

Vizsga almenüpontra kattintva a Vizsgák listája oldal jelenik meg, amelyen két darab lapfűl található. A két lapfűl a Vizsgák listája és az Új vizsga létrehozása. Alapértelmezetten a Vizsgák listája lapfűl be van nyomva.

A Vizsgák listája oldalon egy táblázat jelenik, meg amely a projektben Ön által korábban létrehozott vizsgákat tartalmazza. Ha még nem hozott létre egyet sem akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a vizsga neve, amit létrehozáskor megadott
- Kursus – a kurzus neve, ami alatt a vizsga meghirdetésre került, ha nem adta meg akkor üres lesz ez a mező
- Státusz – két értéket vehet fel: aktív, és inaktív jelzi, hogy vizsgára a hallgatók jelentkezhetnek-e
- Részletek – a tantárggyal kapcsolatos további információkat tekintheti meg egy új lapfűlön

- Eredmények – a vizsgára jelentkező hallgatókat illetve az eredményeket tekintheti meg egy új lapfűlön
- Szerkesztés – a vizsgát szerkesztheti vele
- Törlés – a vizsgát törölheti vele, ha még nem jelentkezett egyetlen hallgató sem

Az Új vizsga létrehozása lapfűlön tud új vizsgát létrehozni. Új vizsga létrehozásához az alábbi adatokat kell megadnia:

Név, a vizsga neve, megadása kötelező.

Kurzus megadása nem kötelező, de általában vizsgát egy kurzus keretein belül hirdetnek meg, így megadása ajánlott

A *Státusz* megadásával szabhatja meg, hogy lehetséges-e a kurzusra jelentkezés.

A *Próbálkozások száma* mezővel megszabhatja azt, hogy hányszor kezdheti el a felhasználó a vizsgát, a megkezdés csakis be nem fejezett vizsgára érvényes. Ez azt jelent, hogy valamilyen hiba miatt (például leállt a számítógép) a hallgató folytatni tudja esetleg máshol a vizsgáját.

Az *IP tartomány megadása* mezővel egy IP cím tartományt határozhat meg (például: 172.14.151.10-172.14.151.40), így csakis azokon a gépeken tudják a hallgatók megkezdni a vizsgát, amelyek beleesnek ebbe a tartományba.

A *Létszám* mező megadása kötelező, segítségével megszabhatja azt, hogy hány hallgató jelentkezhet a vizsgára.

Terem a vizsga helye, tehát ahol a vizsgát el lehet kezdeni.

A *Tesztlapok kiválasztása* hivatkozásra kattintva egy táblázat jelenik meg, ahol ki tudjuk választani a vizsgán használni kívánt tesztlapokat. *Vissza a vizsga szerkesztéséhez* hivatkozással tud visszatérni a megkezdett vizsga szerkesztéséhez.

A *Jelentkezés kezdete* mező segítségével adhatja meg, hogy a tanulók mely időponttól jelentkezhetnek a vizsgára. Az időpont megadásának, megkönnyítését egy a mezőtől balra levő naptár gomb segíti, ha rákattint, akkor egy felugró naptár jelenik meg, ahol könnyedén megadhatja a dátumot órára pontosan.

A *Jelentkezés vége* mező segítségével adhatja meg, hogy a tanulók mely időpontig jelentkezhetnek a vizsgára. Az időpont megadásának, megkönnyítését egy a mezőtől balra levő naptár gomb segíti, ha rákattint, akkor egy felugró naptár jelenik meg ahol, könnyedén megadhatja a dátumot órára pontosan.

A *Vizsga kezdete* mező segítségével adhatja meg, hogy a vizsga mikor kezdődik. Az időpont megadásának, megkönnyítését egy a mezőtől balra levő naptár gomb segíti, ha rákattint, akkor egy felugró naptár jelenik meg, ahol könnyedén megadhatja a dátumot órára pontosan.

A *Jelentkezés vége* mező segítségével adhatja meg, hogy a vizsgának mikor van vége. Az időpont megadásának, megkönnyítését egy a mezőtől balra levő naptár gomb segíti, ha rákattint, akkor egy felugró naptár jelenik meg ahol, könnyedén megadhatja a dátumot órára pontosan.

A *Leírás* mező megadásával egyéb a vizsgával kapcsolatos információt adhat meg.

A *Mentés* gomb megnyomásával mentheti el a vizsgát.

5.2.6. Adminisztráció

Az **Adminisztráció** főmenüpont két almenüpontot tartalmaz ezek az **Intézmény** és a **Projekt**. A regisztrációjakor megadott intézménybe kerül be alapértelmezetten. De lehetősége van Intézményt váltani, és más intézményben is tudja használni a vizsgáztató rendszert. Hasonlóan Projektet is tud váltani és létrehozni is.

Intézmény almenüpontra kattintva az Intézmények listája oldal jelenik meg, amelyen egy darab lapfűl az Intézmények listája található.

Az Intézmények listája oldalon egy táblázat jelenik, amelyben a rendszerben regisztrált intézmények listája jelenik meg. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – az intézmény neve
- E-mail – az intézmény E-mail címe
- Részletek – az intézmény részletes leírását tekintheti meg
- Aktivál – csatlakozhat az intézményhez ekkor az intézményen belül a legelső projekt közül kiválasztásra, amit a **Projekt** oldalon tud megváltoztatni.

Projekt almenüpontra kattintva a Projektek listája oldal jelenik meg, amelyen két darab lapfűlet talál. A két lapfűl a Projektek listája és az Új projekt létrehozása. Alapértelmezetten a Projektek listája lapfűl be van nyomva.

A Projektek listája oldalon egy táblázat jelenik, amelyben a rendszerben az aktív intézményben található projektek szerepelnek. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat

fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a projekt neve
- Aktivál – csatlakozhat a projekthez ekkor, ha például új tantárgyat hoz létre, akkor azt ebben a projektben fogja csak látni
- Szerkesztés – az Ön által létrehozott projektet tudja csak szerkeszteni
- Törlés – az Ön által létrehozott projektet tudja csak törölni

Az Új projekt létrehozása lapfülön tud új projektet létrehozni. Új projekt létrehozásához az alábbi adatokat kell megadnia:

Név, a projekt neve, megadása kötelező.

Leírás mező megadásával egyéb a projekttel kapcsolatos információt adhat meg.

5.3. Tanuló

A tanuló felhasználóval történő belépés után egy üdvözlő képernyő fogadja, azaz a személyes kezdőlapja, ahol hasznos információkat talál a rendszerrel kapcsolatban. A bal oldali menüsor segítségével navigálhat az oldalon. A menüsor a következő elemekből áll:

Kezdőlap

Kezdőlap

Elérhetőségek

Kurzus

Kurzusok

Felvett kurzusok

Vizsga

Vizsgák listája

Felvett vizsgák

Kezdődő vizsgák

Adminisztráció

Intézmény

Hozzáférés

Adatmódosítás

Kijelentkezés

A **Kezdőlap**, **Adminisztráció**, **Hozzájárás** főmenüpontok megegyeznek a Tanár felhasználó **Kezdőlap**, **Adminisztráció**, **Hozzájárás** főmenüpontok bemutatásánál leírtakkal.

5.3.1. **Kurzus**

A **Kurzus** főmenüpont két almenüpontot tartalmaz ezek a **Kurzusok** és a **Felvett kurzusok**. A kurzusok oldalon el tudja érni az intézményben szereplő kurzusokat, a felvett kurzusok oldalon pedig a már korábban felvett kurzusokat érheti el.

Kurzusok almenüpontra kattintva a Kurzusok listája oldal jelenik meg, amelyen két darab lapfül található. A két lapfül a Kurzusok listája és a Felvett kurzusok. Alapértelmezetten a Kurzusok listája lapfül be van nyomva. A **Felvett kurzusok** menüpontra kattintva pedig a Felvett kurzusok oldalra juthat el.

A Kurzusok listája oldalon egy táblázat jelenik meg amely az intézményben korábban meghirdetett kurzusokat tartalmazza. Ha még az intézmény egyetlen kurzust sem hirdetett meg akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- **Kurzus** – a kurzus neve
- **Tantárgy** – a tantárgy neve, aminek keretein belül a kurzus meg lett hirdetve
- **Státusz** – két értéket vehet fel: aktív, és inaktív, csak aktív kurzust vehet fel
- **Tanév** – a kurzus mely félévben van meghirdetve
- **Jelentkezés kezdete** – az időpont, amelytől jelentkezhet a kurzusra
- **Részletek** – a kurzusról tekinthet meg bővebb információt
- **Felvétel** – rákattintva jelentkezhet a kurzusra

A Felvett kurzusok oldalon egy táblázat jelenik meg amely az Ön által korábban felvett kurzusokat tartalmazza. Ha még nem jelentkezett egyetlen kurzusra sem akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- **Kurzus** – a kurzus neve
- **Tantárgy** – a tantárgy neve, aminek keretein belül a kurzus meg lett hirdetve
- **Státusz** – két értéket vehet fel: aktív, és inaktív, csak aktív kurzust vehet fel
- **Tanév** – a kurzus mely félévben van meghirdetve

- Jelentkezés vége – az időpont, ameddig lejelentkezhet a kurzusról
- Részletek – a kurzusról tekinthet meg bővebb információt
- Leckekönyv – a kurzuson elért eredményeket tekintheti meg
- Leadás – rákattintva jelentkezhet le a kurzusról

5.3.2. Vizsga

A **Vizsga** főmenüpont három almenüpontot tartalmaz ezek a **Vizsgák listája**, **Felvett vizsgák** és a **Kezdődő vizsgák**. A vizsgák listája oldalon el tudja érni az intézményben meghirdetett vizsgákat, a felvett vizsgák oldalon pedig a már korábban felvett vizsgákat láthatja. A kezdődő vizsgák oldalon a most folyó, vagy öt percen belül kezdődő vizsgákat tekintheti meg.

Minden egyes menüponthoz tartozik egy lapfűl, és a menüpontra kattintva az a menüpont nevével megegyező lapfűl lesz benyomva.

A **Vizsgák listája** oldalon egy táblázat jelenik meg amely az intézményben korábban meghirdetett vizsgákat tartalmazza. Ha még az intézményben egyetlen vizsgát sem hirdettek meg akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a vizsga neve
- Kurzus – a kurzus neve, aminek keretein belül a vizsga kiírásra került
- Státusz – két értéket vehet fel: aktív, és inaktív, csak aktív vizsgára jelentkezhet
- Időpont – a vizsga végét jelző dátum
- Létszám – a vizsgára jelentkezők maximális száma
- Részletek – a vizsgáról tekinthet meg bővebb információt
- Felvétel – rákattintva jelentkezhet a vizsgára

A **Felvett vizsgák** oldalon egy táblázat jelenik meg, amely az Ön által korábban felvett vizsgákat tartalmazza. Ha még nem jelentkezett egyetlen vizsgára sem akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a vizsga neve
- Kurzus – a kurzus neve, aminek keretein belül a vizsga kiírásra került
- Státusz – két értéket vehet fel: aktív, és inaktív, csak aktív vizsgára jelentkezhet
- Időpont – a vizsga végét jelző dátum
- Részletek – a vizsgáról tekinthet meg bővebb információt
- Eredmények – a vizsgán elért eredményét tekintheti meg
- Leadás – rákattintva lejelentkezhet a vizsgáról

A Kezdődő vizsgák oldalon egy táblázat jelenik meg, amely az Ön által korábban felvett vizsgák közül azokat jeleníti meg, amelyek elkezdődtek, vagy öt percen belül el fognak kezdődni. Ha nincs kezdődő vizsga akkor a „Jelenleg üres a lista” szöveget olvashatja. A táblázat legelső sora tartalmazza a kereső mezőket, figyeljen arra, hogy a keresés kis-nagybetű érzékeny. A következő sorban a táblázat fejléce található esetleges szűrőmezőkkel ellátva. A táblázat a következő oszlopokat tartalmazza:

- Név – a vizsga neve
- Kurzus – a kurzus neve, aminek keretein belül a vizsga kiírásra került
- Időpont – a vizsga végét jelző dátum
- Részletek – a vizsgáról tekinthet meg bővebb információt
- Kezdés – rákattintva elkezdheti a vizsgát

Vizsga megkezdésére kattintva egy új oldalra, a Tesztlap kitöltésére kerül. Ezen az oldalon látja az információkat a vizsgával kapcsolatban, és kiválaszthatja a lapozást is, azaz hány kérdés jelenjen meg egyszerre a teszt kitöltése során egy oldalon. A kezdés gombra kattintva tudja megkezdni a vizsgát. Vizsgát megkezdve minden oldalon az oldal felső részében megjelenik az *Információk megjelenése* hivatkozás, melyre rákattintva a következő információkat tudja megjeleníteni:

- teszt általános leírását
- valamint egy visszaszámláló mezőt, amely a vizsgából hátralévő időt mutatja
- Vizsga befejezése gombot, mellyel a vizsgát befejezheti

A tesztkések alatt láthatja a lapozó gombokat, minden egyes lapozással menti a bejelölt válaszokat. Ha letelt az idő, azaz befejeződött a vizsga, de még nem fejezte be, akkor a lapozás hatására vizsga mentésre kerül, és befejeződik.

6. Összefoglalás

Ebben a fejezetben szeretném összefoglalni a korábbi fejezetekben részletezett, fontosabb és érdekesebb részeket.

A bevezetőben kitértem arra, hogyan fogalmazódott meg az ötlet, hogy miért választottam egy vizsgáztató rendszer fejlesztését. Tömören az oktatás tömegesedése és a számítástechnika begyűrűződése a társadalom életébe adta azt az ötletet, hogy miért ne lehetne egy olyan rendszert készíteni, amivel meg lehetne könnyíteni az oktatók és a tanulók dolgát.

A fejlesztői eszközök részben bemutattam azokat az eszközöket, technológiákat, amelyeket a fejlesztés során használtam. Igyekeztem platform függetlenül megválasztani az eszközöket, sőt minden eszköz ingyenes, szabadon használható, így bárki számára elérhető.

A LAMP egy elterjedt és kiválóan használható gyűjtemény, egyes elemekről szeretnék véleményt mondani:

- Linux: a Windows operációs rendszerről áttérve, először kicsit idegenként hatott az újdonság, de már annyira a megszerettem, hogy ha bekapcsolom a gépet, már Linuxot indítok, nem Windows-t.
- Apache: Gyors, megbízható webszerver, nem volt problémám vele.
- MySQL: Szintén kiválóan működött, és megfelelően gyors volt.
- PHP: nagyon megszerettem a PHP-t, egyszerűen sikerült elsajátítanom. Még mindig vitatkoznak a szakértők, hogy van-e értelme objektumorientáltan programozni weboldal készítésénél. Számomra a kérdés egyértelmű, el se tudnám képzelni objektumorientált eszközök nélkül a rendszer működését

A szakdolgozat írásának első félévében megpróbáltam keretrendszer nélkül a rendszert megvalósítani. Kudarcba fulladt a próbálkozás, nem tudtam nyomon követni az oldalakat, keveredett a PHP-s kód HTML-el, egyszerűen szörnyű volt egy osztályra ránézni. Zend Framework és az MVC minta segítségével bármikor fejleszthetem a rendszert, ugyanis jól elkülönített helyen vannak az egyes elemek.

A JavaScript egy kis szeletével foglalkoztam csak, és sokat bajlódtam az egyes elemekkel, de egyszerűbben kezelhetőbbé vált a rendszer.

Az Eclipse használatával gyorsabban ment a kódolás, ugyanis előszeretettel használtam az automatikus befejezés eszközöket.

A Rendszer működése fejezet két részből áll.

Bemutattam az adatbázist, minden táblára kitértem és részleteztem az adatbázis mezőket is. Az adatbázis terv folyamatosan változott a rendszer fejlesztése közben. A feleslegesnek vélt ötletek kikerültek, például szerepelt egy `test_template` tábla is, amellyel a tesztlapok megjelenését lehetett volna szabályozni. Valószínű még a későbbiekben is változni fog, mert a szakdolgozat megírásakor is hiányoltam néhány funkciót, ilyen például intézmény, vagy projekt váltását lehetne jelszóval védeni.

A rendszer működése volt a másik rész, ahol egy kis szeletét mutattam meg a rendszer kódjának, logikájának. A Zend Framework kiváló eszközöket nyújt az úgynevezett gyors alkalmazásfejlesztéshez, gondolok itt például a rendszer autentikációs megoldására. A tábla és sor átjárók sok segítséget nyújtottak a lekérdezések összeállításához és végrehajtásához.

Mielőtt hozzákezdttem volna a rendszer fejlesztéséhez, először megvizsgáltam az akkor elérhető rendszereket, hogy nehezen véletlenül egy meglévő rendszert készítsék el újból. Úgy érzem, sikerült egy olyan egyedi alkalmazást készíteni, melyet kiegészítve akár nagyobb intézmények is alkalmaznának.

A Rendszer bemutatása fejezet lényegében egy felhasználói kézikönyv. Szeretném ezt a részt sűgőként elhelyezni az oldalon. A fejezetben bemutatott funkciók nagy többsége már meg van valósítva, tehát a rendszer 90%-ban készen van. Az eredmények csoportos megjelenítése és a vizsgázás folyamatának finomítása maradt csak hátra. A rendszer azért nem készült el teljesen, mert a szakdolgozatot másodmagammal kezdtem el készíteni, de akivel készítettem, az ebben a félévben nem fejezi be a diplomamunkát és sajnos nem tudott olyan mértékben segíteni a fejlesztésben, ahogy azt előre elterveztük. A feladat megvalósításában 50%-ot vállaltam el, de túlteljesítettem a részem, mert a rendszert én fejlesztettem a jelenlegi állapotra.

A rendszer fejlesztése során rengeteget tanultam és sok gyakorlatra tettem szert, amit remélem a későbbiekben kamatoztatni, tudok majd az egyetemi tanulmányaim után. Remélem, hogy lesz olyan intézmény, vagy tanulók egy csoportja, aki majd tudja használni a rendszert.

7. Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, Pere Zsoltnak, hogy megosztotta velem tapasztalatait, valamint dolgozatom megírásához nyújtott útmutató tanácsokat.

8. Irodalomjegyzék

- [1] Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford: *Patterns of Enterprise Application Architecture*. Addison Wesley, 2002
- [2] Rob Allen, Nick Lo, Steven Brow: *Zend Framework in Action*.
Manning Publications, 2008
- [3] Apache szerver statisztika:
http://news.netcraft.com/archives/2009/04/06/april_2009_web_server_survey.html
- [4] Elektronikus Vizsgáztató Rendszer:
<http://www.bibl.u-szeged.hu/evr/>
- [5] Moodle e-learning keretrendszer:
<http://moodle.org/>
- [6] PHP hivatalos oldal:
<http://php.net/>
- [7] PHP-t használó szerverek száma:
<http://www.php.net/usage.php>
- [8] Zend Framework bemutatása:
<http://framework.zend.com/about/overview>
- [9] Zend Framework hivatalos oldal:
<http://framework.zend.com/>