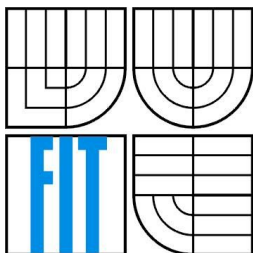


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MODUL ONLINE TESTOVÁNÍ V INFORMAČNÍM SYSTÉMU
ONLINE QUIZZES IN INFORMATION SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL SLÁMA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PETR LAMPA

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Centrum výpočetní techniky

Akademický rok 2006/2007

Zadání diplomové práce

Řešitel: **Sláma Pavel**

Obor: Výpočetní technika a informatika

Téma: **Modul online testování v informačním systému**

Kategorie: Web

Pokyny:

1. Seznamte se s formáty používanými pro import a export testových otázek v různých e-learning systémech (WebCT, Blackboard, Moodle, apod.).
2. Navrhněte strukturu databáze pro uložení testových otázek pokrývají možnosti uvedených systémů.
3. Navrhněte a implementujte moduly pro manipulaci s testy, online testování a automatické hodnocení výsledků testů.
4. Zhodnoťte úplnost navržených modulů vzhledem k uvedeným systémům.

Literatura:

- URL: <http://www.moodle.org/>
- URL: <http://www.webct.com/>

Při obhajobě semestrální části diplomového projektu je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Lampa Petr, Ing.**, CVT FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 22. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Pavel Sláma**
Id studenta: 23160
Bytem: Rovečné 220, 592 65 Rovečné
Narozen: 04. 06. 1983, Nové Město na Moravě
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
diplomová práce

Název VŠKP: Modul online testování v informačním systému
Vedoucí/školicel VŠKP: Lampa Petr, Ing.
Ústav: Centrum výpočetní techniky
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

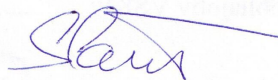
Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel



.....

Autor

Abstrakt

Tato práce se snaží zorientovat v širokém prostoru poměrně nové problematiky e-learningu. E-learning přišel s rychlým rozvojem internetu, jako z pohodlnější, zrychlení vzdělávání a zároveň snížení jeho závislosti na lidských zdrojích. Problematika online testování má před sebou ještě velký kus nutného vývoje, ale základem je ujasnit si pravidla hry a snažit se o jejich sjednocení. První metou k vytvoření plnohodnotného systému schopného nahradit metody stávajícího vzdělávání je zmapování problematiky testování obecně, analýza požadavků a vytvoření formátu pro přenos dat, což je zapotřebí kvůli vzájemné komunikaci. Na těchto základech je možné navrhnout databázi a potažmo celý systém.

Klíčová slova

e-learning, online test, vzdělávání, otázka, internet, php, mysql, databáze.

Abstract

This thesis is trying to find its way in extensive area of relatively new problems of E-learning. E-learning has come together with fast advancement of the Internet trying to make education more comfortable and faster and at the same time trying to lower its dependency on the human resources. A lot of a development of the online tests is still needed. However, it is important to make clear the rules of the "game" and to press for their integration. The first goal to achieve the full-value system which would be able to substitute the conventional methods of education is to find out and solve the issues of testing in general, to analyze the requirements and to create data format for the data portability, which is required because of the mutual communication between different systems. On these bases it is possible to design the database scheme and then the complete system.

Keywords

e-learning, online quiz, education, question, internet, php, mysql, database.

Modul online testování v informačním systému

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Petra Lampy

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Sláma
21. května 2007

Poděkování

Rád bych poděkoval Ing. Petru Lampovi za počáteční inspiraci a všem, kteří mi vytvořili podmínky pro tvůrčí činnost.

© Pavel Sláma, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	8
1 Úvod.....	11
2 Pojem e-learning.....	12
2.1 Proč?.....	12
2.2 Jak?.....	12
2.3 VLE – Virtual Learning Environment.....	13
2.4 Elektronické testy.....	13
3 Technologie.....	15
3.1 Obecně.....	15
3.2 Webový server.....	15
3.3 PHP.....	15
3.4 Databáze.....	16
3.5 Součinnost jednotlivých částí.....	16
4 Existující testovací moduly.....	17
4.1 Typy otázek.....	17
4.1.1 Multiple choice.....	17
4.1.2 Multiple answer.....	17
4.1.3 True/False.....	18
4.1.4 Short answer.....	18
4.1.5 Numerical.....	18
4.1.6 Fill-in-the-blank/Cloze.....	19
4.1.7 Matching question.....	19
4.1.8 Ordering/ranking.....	19
4.1.9 Drag-n-drop.....	20
4.1.10 Paragraph question.....	20
4.2 Formáty pro import/export testových otázek.....	21
4.2.1 Standard IMS QTI.....	21
4.2.2 Formát Aiken.....	21
4.2.3 Formát GIFT.....	22
4.2.4 Formát Cloze (embedded answers).....	22
4.2.5 Ostatní formáty.....	23
4.3 Srovnání jednotlivých systémů.....	23
5 Návrh aplikace.....	25
5.1 Role uživatelů v systému.....	25

5.2	Use-Case diagram.....	26
5.3	E-R diagram	27
5.4	Popis databáze	28
5.4.1	tm_quizz_cat, tm_question_cat	29
5.4.2	tm_quizzes	29
5.4.3	tm_questions	29
5.4.4	tm_qTypes.....	30
5.4.5	tm_quizz_quest_assigns.....	30
5.4.6	tm_users	30
5.4.7	tm_results.....	30
5.4.8	tm_user_answers	31
5.4.9	tm_config	31
5.5	Tabulky odpovědí na jednotlivé typy otázek.....	31
5.5.1	tm_a_multiple_answer.....	31
5.5.2	tm_a_short_answer.....	32
5.5.3	tm_a_numerical	32
5.5.4	tm_a_cloze	32
5.5.5	tm_a_ordering	32
5.6	Rozšíření	33
6	Implementace.....	34
6.1	Použité technologie.....	34
6.1.1	PHP Session	34
6.2	Databázová vrstva.....	35
6.2.1	C_db_common	35
6.2.2	C_field_base.....	36
6.2.3	Zbývající třídy databázové vrstvy	37
6.3	Činnost systému.....	37
6.3.1	Přihlášení.....	37
6.3.2	Role lektor.....	38
6.3.3	Role Administrátor	40
6.3.4	Role Student	40
6.4	Popis jednotlivých programových modulů.....	42
6.4.1	Tvorba XML pomocí DOM rozhraní	42
6.4.2	Tvorba formulářů.....	42
6.4.3	Správa uživatelů a editace profilu	43
6.4.4	Správa otázek a odpovědí	43
6.4.5	Správa testů, přiřazování otázek, výsledky	45

6.4.6	Modul pro vlastní průběh testu.....	47
7	Zhodnocení navrženého systému	49
8	Závěr	50
	Literatura.....	51
	Příloha A – Instalace systému	52

1 Úvod

Již několik let ve světě probíhá vývoj informačních systémů za účelem zjednodušení a zvětšení efektivity procesu vzdělávání. Jelikož pro naprostou většinu studentů a učitelů je přístup k internetu samozřejmostí, ne-li přímo nutností, otvírají se nám tímto široké možnosti využití tohoto dynamického média. Zpočátku dílčí aplikace na různých platformách začaly tvořit komplexnější celky, které jsou dnes schopné zabezpečit po informační stránce celou vzdělávací činnost univerzity.

V další kapitole bych rád shrnul obecné požadavky na tyto systémy a různé související standardy a přístupy k řešení této problematiky. Postupně budu klást důraz na hlavní předmět této práce a to systémy umožňující online testování. Myslí se tím v první řadě příprava, správa, uchovávání, prezentace a hodnocení testů.

Následovat bude jakési shrnutí dostupných technologických možností pro vytvoření a provoz těchto systémů. Převážně se ale bude týkat možností programového vybavení uvedeného v zadání (Apache, PHP5, MySQL). Zmíním klady a obtíže takového řešení spolu se základními koncepty navrhovaného postupu.

Další kapitola bude věnována již existujícím systémům pro e-learning. Zde už se zaměřím na konkrétní část, která se týká testů a veškeré manipulace s nimi. Všimneme si existujících typů otázek a s tím souvisejících různých formátů pro export a import dat. Cílem této práce je navrhnout systém pokrývající co možná nejširší možnosti různých již fungujících projektů a umožnit vzájemnou přenositelnost dat. Pokračování tvoří konkrétní návrhy a koncepty vlastního zpracování. Od návrhu databáze přejdeme k základním požadavkům na aplikační část.

Dále popíšeme zvolenou techniku implementace a objasníme funkce jednotlivých modulů. Na závěr zhodnotíme implementovaný systém ve srovnání s ostatními produkty.

2 Pojem e-learning

Co si vlastně představit pod tímto názvem? Jde o jakýsi způsob učení se převážně s využitím počítačů, popřípadě jiných komunikačních zařízení jako PDA, mobilní telefon, atd. Lze také říci, že jde o učení s využitím sítě a dokumentů na ní se nacházejících. Pro tuto konkrétní oblast se používá označení „online education“. Je tak umožněno studium odkudkoliv a kdykoliv (tzv. asynchronní studium).

2.1 Proč?

Důležité pro samotný návrh každého systému je porozumění motivu jejich vzniku. Se zvětšujícím se počtem lidí, kteří mají přístup k internetu, začíná být efektivnější a pohodlnější prezentovat informace v elektronické podobě na webu, než v podobě papírové.

Předvedeme na ukázce životního cyklu tištěné knihy a její elektronické podoby. Stádium návrhu a vzniku mají obě podoby společné. Dále se už ovšem značně liší. Tištěnou knihu je třeba vytisknout, což je jistě poměrně nákladné jak finančně, tak časově. Potom přichází na řadu distribuce a užívání čtenáři. Již zde se naplno projeví výhody elektronické kopie. Náklady na její vystavení nebo zpřístupnění na webu jsou minimální, přičemž není třeba se zabývat počtem výtisků. Elektronická podoba je přístupná teoreticky neomezenému počtu čtenářů. A dostáváme se k dalšímu důležitému rozdílu a to sice k velmi snadné možnosti aktualizace, kterou skýtá elektronická kniha. Prakticky má čtenář neustále k dispozici aktuální verzi s novými údaji aniž by se o to musel jakkoliv přičinit. Nemluvě o rozdílu ve fyzické trvanlivosti dat v tištěné a elektronické formě. Naopak mezi zápory patří v některých případech chabá ochrana autorských práv.

E-learning samozřejmě neskýtá pouze možnost prezentace elektronických kopií učebnic. Lze například prezentovat audio-vizuální záznamy, nespočet podpůrných dokumentů. Komunikaci učitelů a studentů zrychlují a usnadňují diskusní fóra. Všechny tyto prvky lze samozřejmě spravovat a měnit podle konkrétních potřeb té které školy nebo přímo kurzu. V neposlední řadě nesmíme zapomenout na část stěžejní pro tuto práci a to možnosti testování, které se budou věnovat další kapitoly.

2.2 Jak?

Do elektronického učení se (jak by se dal přeložit pojem e-learning) spadá mnoho různých přístupů.

První velká skupina produktů slouží k tzv. offline education (čili primárně neslouží k použití v síti). K tomuto vzdělávání není za potřebí internet. Data čtenář obdrží formou nějakého média (CD, DVD) většinou spolu se softwarem, který je dovede logicky a přehledně prezentovat. Možnosti těchto

systemů jsou poněkud omezené. Zvláště problematická aktualizace a nemožnost interakce s dalšími systémy a lidmi z nich činí pouze lepší náhradu tištěného materiálu.

My se ovšem budeme zabývat hlavně systémy spadajícími pod tzv. online education. Do této skupiny lze mimo jiné zahrnout jakýkoliv naučný text nacházející se na internetu. Může se jednat o text v podobě html stránek (ty se dají jednoduše prohlížet jakýmkoliv www prohlížečem), nebo o dokumenty v nějakém jiném formátu k jejichž zobrazení musí mít čtenář nainstalovaný speciální software. Zatím jsme mluvili o jakýchsi elektronických knihách, přičemž internet slouží k jejich vyhledání, uložení a prezentaci. Elektronické vzdělání ale snadno poskytne i další názornější prvky, které jen obtížně (pokud vůbec) najdou své klasické protějšky. Řeč je o názorných demonstračních experimentů, dodatečných obrazových, multimediálních datech.

2.3 VLE – Virtual Learning Environment

Náš zájem se soustředí na komplexní programové balíčky, které vytváří virtuální vzdělávací prostředí. Dosud jsme se zabývali pouze čistou prezentací informací, vědomostí. Nyní bychom chtěli tyto data nějakým přehledným způsobem členit a spravovat a nejen to. Budeme chtít vytvořit jakousi virtuální školu, která bude mít své učitele, studenty a kurzy. Vše popřípadě ještě můžeme členit na fakulty (už záleží jen na tom, z jakého pohledu na systém nahlížíme). Každá fyzická osoba, která je součástí systému, v něm bude mít svůj virtuální profil. Typ profilu (učitel, student, zvláštní postavení bude mít administrátor) určuje přístupová práva konkrétní osoby ke všem funkcím systému. Každý kurz má své učitele a studenty. Učitelé kurz plní informacemi. Jde o dokumenty týkající se obsahu kurzu, ovšem navíc tu je možnost vytvářet domácí úkoly, testy, sledovat pokroky jednotlivých posluchačů. Studentům jsou přístupná data, která určí učitel. Kurzy by měly obsahovat rovněž diskuzi, chat nebo jinou formu komunikace studentů a učitelů.

Tímto způsobem lze vytvořit úplnou virtuální školu, která nutně nemusí mít skutečný protějšek. Většinou ale toto virtuální prostředí doplňuje skutečnou školu, přičemž představuje zajímavou alternativu ke klasickému systému studia (přednášky, tištěná skripta, daný rozvrh), kterému může asistovat.

2.4 Elektronické testy

Dalším zúžením našeho zaměření jsou moduly VLE systémů pro práci s testy.

Je nutné nějakým způsobem zjišťovat úroveň znalostí, které studenti v kurzu získali. Bohužel individuální ověření u každého zvláště je vzhledem k velkému počtu studentů značně časově náročné a navíc ne vždy úplně spravedlivé. To je důvod zavedení hromadných testů.

Klasický test je vlastně souhrn otázek jež mohou mít různou formu, na které student odpovídá. V závislosti na formě otázky se studentova odpověď vyhodnotí jako špatná, správná nebo „něco mezi“. Celkové ohodnocení je výsledkem souhrnu dílčích ohodnocení otázek.

Proces výroby a provedení takového testu (ať už papírového nebo virtuálního) probíhá následovně:

- § Vytvoření souboru otázek a správných a špatných odpovědí (na jejichž kvalitě často velmi závisí)
- § Zvolení způsobu náhodného výběru otázek (pokud nemají mít všichni studenti stejný test)
- § Provedení testu (můžeme umožnit několik pokusů, penalizace,...)
- § Hodnocení (uložení, statistiky,...)

Z toho plynou požadavky na cíl tohoto projektu.

3 Technologie

3.1 Obecně

Už ze své podstaty je každý konkrétní e-learningový systém určen pro množství lidí, kterým umožňuje jakousi formu sdílení dat. Většina jich je tedy postavena na nějaké variantě architektury klient-server. To znamená, že celý systém sídlí v nějakém počítači a plní požadavky klientů. Jednotlivá provedení se přitom liší hlavně podílem klienta na činnosti systému. Čili jinými slovy jde o to, jestli klient jen zabezpečuje komunikaci a vyměňuje si data se serverem, nebo obsahuje i nějakou aplikační část, popřípadě jak velkou. V krajním případě klient pouze zobrazuje data ze serveru a umožňuje interakci uživatele, v opačném extrému naopak server slouží pouze jako skladiště dat a veškerá aplikační činnost probíhá u klienta. Obě varianty mají svá pro a proti. Druhá možnost klade menší nároky na server, který nemusí vykonávat veškerou činnost, ovšem je nutné instalovat na klientských počítačích speciální software, což s sebou nese problémy s údržbou a provozem.

My se zaměříme na variantu, kdy co možná největší část činností svěříme serveru, který můžeme snadno spravovat, a jako klientský program poslouží webový prohlížeč. Tudíž data budeme prezentovat ve formě webových stránek.

3.2 Webový server

Nejdůležitější programové vybavení serveru. Tento program odpovídá na požadavky klientů a posílá jim odpovídající data. Mezi nejpoužívanější programy tohoto druhu patří Apache, IIS, PWS. Existují speciální variace, které se hodí ke konkrétním účelům (Apache Tomcat). Sám o sobě ovšem neumí nic víc – posílá pouze statická data, jejichž adresu klient předává serveru v požadavku. Mocným se stává až ve spojení s nějakou serverovou skriptovací technologií. My budeme používat server Apache, který nejlépe spolupracuje se skriptovacím jazykem PHP.

3.3 PHP

Interpret tohoto jazyka nainstalovaný na serveru nám dává možnost ovlivňovat data zasílaná klientům. Webový server může ještě před jejich odesláním data předat tomuto interpretu, který je - pakliže jde o kód v PHP - provede a až výsledek tohoto procesu se zašle klientovi. Otevírá se nám tak možnost dynamicky generovat www stránky v závislosti na požadavcích uživatele. Nemá sice takové možnosti jako jiné vyšší programovací jazyky (C++), nicméně pro účely webu jsou dostatečné.

Hlavní přínos těchto technologií spočívá v umožnění spolupráce s databázemi.

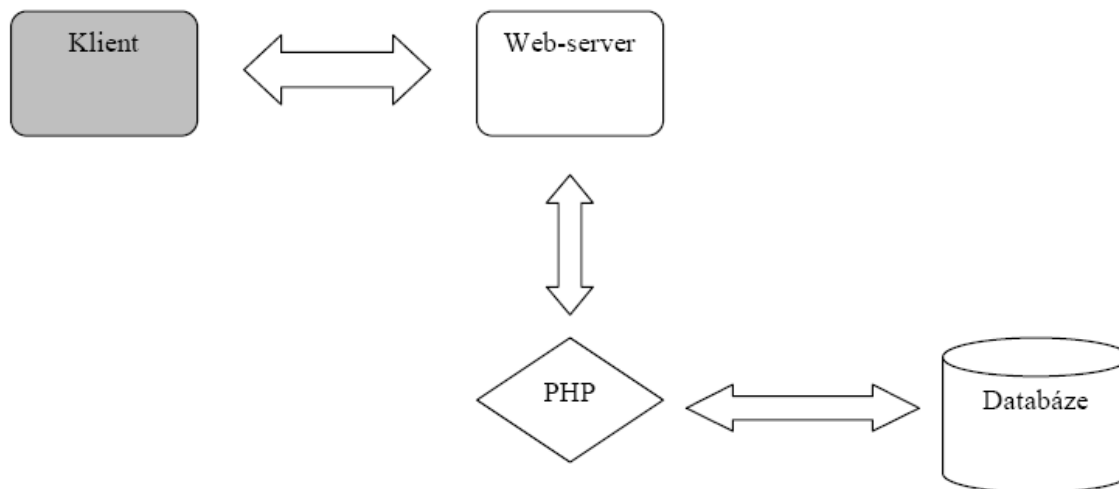
3.4 Databáze

Jedná se o specializovaný software pro práci s daty. Existují různé databázové architektury jejichž kompletní výčet a popis je mimo působnost této práce. Snahy směřují k oddělení dat od aplikací, čímž je umožněna snadná aktualizace obou aniž by se vzájemně ovlivňovala funkce. Oblíbeným databázovým serverem v kombinaci s dvojicí Apache/PHP se stal MySQL. Všechny tyto technologie jsou volně dostupné a MySQL jednoduchým rychlým a pro účely webu ideálním nástrojem. Lze mu vyčíst absenci některých pokročilých funkcí komerčních databází, čímž klade větší nároky na aplikace, které s ním pracují. K obsluze databáze používáme jazyk SQL.

Databázi se vyplatí použít vždy když potřebujeme uložit větší množství dat s podobnou strukturou. Databáze poskytuje pohodlnější prostředí pro práci s daty a oprostí nás od technických detailů samotného fyzického uložení dat.

3.5 Součinnost jednotlivých částí

Sestava tohoto softwaru spolu komunikuje následujícím způsobem:



Obrázek 3-1.: Komunikace jednotlivých částí systému

Klient zasílá serveru požadavky, server odpovídá požadovanými daty. Pokud je to možné, jsou data nejprve zpracována PHP interpretem, který má možnosti komunikovat s databází. PHP také umí reagovat na činnost uživatele a tak ovlivňovat výstup.

4 Existující testovací moduly

V této kapitole vysvětlíme základní pojmy oblasti elektronických testů, projdeme používané formáty pro přenášení dat a pokusíme se shrnout možnosti různých testovacích programů.

4.1 Typy otázek

Jako první je nutné ujasnit si, co vlastně test může obsahovat. Existují různé druhy otázek a vzhledem k nutnosti jejich uchování v databázi je záhodno pokusit se je rozčlenit do podkategorií, které mohou být uloženy ve stejné struktuře.

4.1.1 Multiple choice

Základní testová otázka. Znamená, že odpověď student vybírá vždy jednu z několika možností. Každá možnost je ohodnocena. Všechna hodnocení by měla být pozitivní (včetně nuly) a mohou obsahovat zpětnou vazbu, čili zprávu, kterou student uvidí po zvolení té které odpovědi.

Př.: Kdo je současným českým premiérem (5-2007)?

- a) Václav Klaus
- b) Václav Havel
- c) *Mirek Topolánek*
- d) Miloš Zeman

Jedna z odpovědí je ohodnocena 100% ostatní 0%.

4.1.2 Multiple answer

Tento typ je odvozen od multiple choice. Navíc je možných několik odpovědí. Tentokrát ale mohou odpovědi být i negativně ohodnoceny. Pokud student zvolí takovou kombinaci otázek, která má negativní součet dílčích ohodnocení, výsledek by měl být nula.

Př.: Které z následujících produktů lze označit jako open source?

- a) Linux
- b) Windows XP
- c) Photoshop
- d) PHP

Kde odpovědi a) a d) jsou obě ohodnoceny 50% a b) a c) -50%.

Čili pokud student odpoví například a), b), d), tak získá 50% bodů z maxima možných.

4.1.3 True/False

Jde vlastně o multiple choice, ale počet možných odpovědí je pevně dán na dvě a to **ano** nebo **ne**. Jde o potvrzení nebo vyvrácení daného tvrzení. Jedna z odpovědí vždy musí být ohodnocena 100% a druhá 0%.

Př.: Země je plochá.

a) ano

b) ne

Odpověď ano má 0%, odpověď ne má 100%

4.1.4 Short answer

Tento typ otázky jako odpověď vyžaduje jedno slovo nebo krátké sousloví. Je možné zadat více správných odpovědí, přičemž každou ocenit jinak. V některých systémech je možné využít i zástupné symboly, např. *.

Př.: Tenká membrána, která odděluje vnitřní a vnější ucho se nazývá

Jako správnou odpověď definujeme *bubínek* a ohodnotíme 100%. Čili ať student odpoví jakkoliv, pokud bude odpověď obsahovat toto slovo, bude správná.

4.1.5 Numerical

Tento typ se podobá short answer. Odpověď na tuto otázku je však číselná. Oproti ostatním typům lze definovat toleranci, čili rozsah hodnot, které ještě budou považovány za správnou odpověď. Tu lze definovat různými způsoby. Buď absolutní možnou odchylkou, nebo relativní, či geometrickou odchylkou. Relativní odchylka je vhodnější pokud během výpočtu může vzniknout velká chyba vinou například zaokrouhlení. Geometrickou odchylku dovoluje větší chybu směrem nahoru, než dolů. To je výhodné při výpočtech v komplexním oboru, kde při větší toleranci by se nula stala univerzální správnou odpovědí. Dále je možné určit počet porovnávaných signifikantních cifer. Některé systémy umožňují používat v zadáních těchto otázek proměnné a aritmetické funkce. V některých případech je vhodné umožnit i alfanumerickou odpověď, například nekonečno.

Př.: Kolik je $2 + 3$?

Definujeme správný výsledek jako 5, toleranci 0,01 a to relativní (čili tolerujeme 1% odchylku z výsledku). Porovnávat se budou 3 cifry. Za správnou odpověď tedy považujeme číslo z intervalu 4,95 až 5,05 včetně.

4.1.6 Fill-in-the-blank/Cloze

Zde je možné zkombinovat několik různých druhů otázek. Jedná se o textové zadání do něhož se dají vkládat na libovolná místa předchozí typy otázek.

Př.: Arabové většinou čtou z do

V prázdných místech se nachází otázky typu multiple choice s možnostmi „leva“ a „prava“.

4.1.7 Matching question

Jde o přiřazování. Máme skupinu otázek a společnou skupinu promíchaných odpovědí. Existuje jedna správná odpověď pro každou otázku. Student musí odpovědět na všechny otázky, přičemž správné odpovědi na všechny z nich jsou ohodnoceny stejně (dohromady musí dát 100%).

Př.: Doplňte odpovídající úhly (1)-(6):

Otázky	Stupně	Radiány
$\frac{1}{2}$	(1)	(2)
(3)	60°	(4)
(5)	(6)	$\pi/6$ rad

Tabulka 1.: Otázka typu matching

Možnosti: 180° , π rad, $1/6$, $\pi/3$ rad, $1/12$, 30°

Každá možnost bude ohodnocena 16,67%.

4.1.8 Ordering/ranking

Zvláštní typ matching question kdy se k daným možnostem přiřazují přirozená čísla. Jde o seřazení nebo ohodnocení daných odpovědí (vyjádření preferencí). Hodnocení je opět rozděleno rovnoměrně mezi všechny správné dvojice a celkový výsledek tvoří součet správně přiřazených dvojic.

Př.: Seřad'te události od nejstarší k nejmladší:

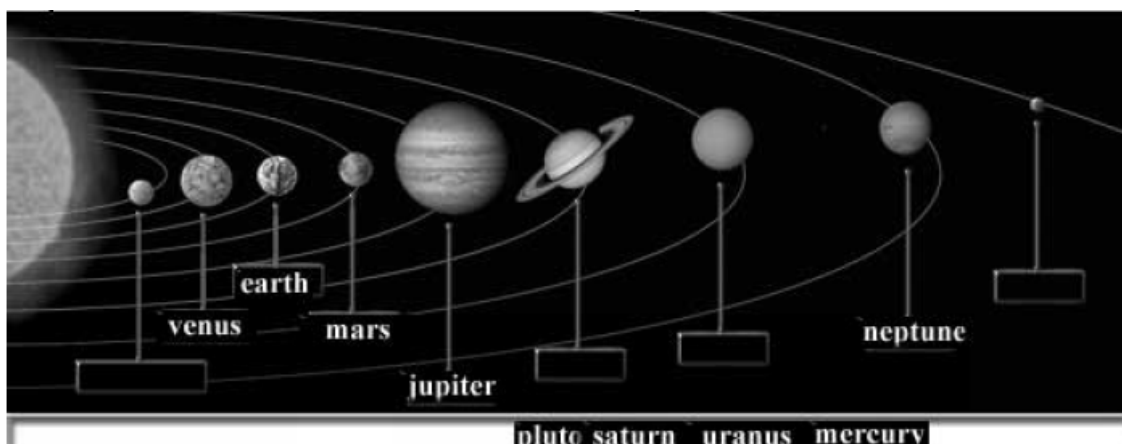
- Rozdělení Československa
- Sametová revoluce
2. světová válka
- První televizní vysílání v Čechách

Máme 6 dvojic (3-4, 3-2, 3-1, 4-2, 4-1, 2-1), správné pořadí je 3, 4, 2, 1. Každá dvojice je ohodnocena 16,67 %. Pokud student seřadí události například takto: 1, 3, 2, 4; obdrží za otázku 33 % (2/6 správných dvojic).

4.1.9 Drag-n-drop

Tento typ poněkud vybočuje. Všechny ostatní formy otázek jsou textové, kdežto zde je nutná grafická podpora (i ostatní typy lze obohatit grafikou, ale není to nutné). Odpověď je založena podobně jako u matching question na přiřazení skupiny odpovědí skupině otázek (jedna odpověď k jedné otázce). Jednotlivé otázky a odpovědi jsou tvořeny grafickými entitami a přiřazení se provede tažením odpovědi myši na odpovídající otázku. Opět za každé správné přiřazení student obdrží poměrný počet procent.

Př.:



Obrázek 4-1.: Ukázka otázky typu drag-n-drop

Zde student myši přetahuje názvy planet do odpovídajících kolonek v zadání.

4.1.10 Paragraph question

Posledním typem je textová otázka. Student odpovídá vlastním libovolně dlouhým textem. Nevýhodou tohoto typu otázek je nemožnost automatického hodnocení. Odpověď musí být ohodnocena ručně. Volitelně lze předepsat text, který se objeví v poli pro odpověď a naznačit tak očekávaný výsledek.

Př.: Stručně vysvětlete směrování v IP sítích.

Do odpovědního formuláře student napíše své vysvětlení. Opravující musí ohodnotit správnost odpovědi a přiřadit dosažená procenta.

4.2 Formáty pro import/export testových otázek

Důležitým parametrem každého systému je jeho schopnost komunikovat s ostatními systémy s podobným zaměřením. Pro tyto účely je nutné stanovit formáty souborů pro uložení všech zmíněných typů otázek. Bohužel se dá říct, že co produkt, to jiný formát. Navíc u komerčních produktů není snadné získat přesné specifikace a proto je obtížné vytvořit software schopný importovat data vytvořené v těchto produktech.

4.2.1 Standard IMS QTI

Sjednocením a standardizací používaných formátů se mimo jiné od roku 1997 zabývá vzdělávací konsorcium IMS Global Learning Consortium, Inc. (Instructional Management Systems). Mezi jeho partnery patří výrobci uznávaných e-learningových a testovacích produktů (WebCT, Blackboard, Questionmark, a další), vývojáři, univerzity a velké firmy v oblasti IT. Toto konsorcium vydalo již několik verzí standardů týkajících se e-learningu. Do oblasti této práce zapadá konkrétně standard **IMS QTI – Question and Test Interoperability Standard**. Jeho první verze byla zveřejněna v lednu 2000. V současnosti (březen 2007) je aktuální verze 2.1.

Filozofie QTI standardu je založena na uchovávání dat ve formátu XML. Vzhledem k existující podpoře a existenci mnoha nástrojů, které pracují s XML je tento standard velmi výhodný. Je definováno i spojení libovolného množství otázek a nejen to, do jednoho XML souboru lze spojit kompletní obsah kurzu a přenášet ho tak libovolně mezi konkurenčními produkty.

Při návrhu našeho systému bude jistě nutné k tomuto standardu prohlížet.

4.2.2 Formát Aiken

Jde o velmi jednoduchý, pro člověka snadno čitelný formát. Lze jím snadno popsat otázky typu multiple choice. Byl navržen aby umožnil jednoduchý přímý zápis otázek, bez nutnosti používat jakýkoliv speciální software.

Příklad zápisu:

Jaký je účel první pomoci?

- A. Zachránit život, předejít dalšímu zranění, zachovat zdraví
- B. Poskytnout léčbu jakékoliv zranění
- C. Předejít dalšímu zranění
- D. Pomoci obětem, které hledají pomoc

ANSWER: A

4.2.3 Formát GIFT

Opět jde o jednoduchý, snadno čitelný formát, který má ovšem poněkud větší možnosti než předchozí typ. Umožňuje popsat otázky typu multiple choice, multiple answer, true/false, short answer, matching, numerical, fill in the blank. Správná odpověď je označena znakem „=“, špatná znakem „~“. Numerická odpověď znakem „#“. Odpovědi ano/ne jsou označeny anglickými TRUE/FALSE (vždy jen správnou variantou). Otázky lze vkládat volně do textu.

Ukázka:

```
Kdo je zakopán v Grantově hrobce?{~Grant ~Jefferson =nikdo}
Grant {~je zakopán =je pohřben ~žije} v Grantově hrobce.
Grant je zakopán v Grantově hrobce.{FALSE}
Kdo je zakopán v Grantově hrobce?{=žádný člověk =nikdo}
Kdy se narodil Ulysses S. Grant?{#1822}
```

4.2.4 Formát Cloze (embedded answers)

Jde o flexibilní formát, který umožňuje do jedné otázky zakomponovat několik různých typů podotázek. Jde vlastně o text, do nějž je vloženo libovolné množství podotázek. Ty mohou být buď typu multiple choice, multiple answer, short answer nebo numerical.

Př.:

```
Tato otázka se skládá z textu a vložené otázky právě zde
{1:MULTICHOICE:Špatná odpověď#Reakce na špatnou odpověď~Další špatná
odpověď#Reakce na další špatnou odpověď~=Správná odpověď#Reakce na
správnou odpověď~%50%Odpověď za polovinu bodů#Reakce na odpověď za
polovinu bodů} a hned po ní se budete muset poprat s další otázkou
{1:SHORTANSWER:Špatná odpověď#Reakce na špatnou odpověď~=Správná
odpověď#Reakce na správnou odpověď~%50%Odpověď za polovinu
bodů#Reakce na odpověď za polovinu bodů} a na konec numerická otázka
{2:NUMERICAL:=23.8:0.1#Reakce na správnou odpověď
23.8~%50%N/A#Reakce na odpověď za polovinu bodů - v blízkém okolí
správné odpovědi}.
```

Všimněte si, že adresy jako www.moodle.org a emotikony :-) fungují:

- a) Líbí se vám to? { :MULTICHOICE:=Ano#Správně~Ne#Máme jiný názor }
- b) Jak byste to oznámkovali? { 3:NUMERICAL:=3:2 }

Všimněme si, že správná odpověď bývá označena znakem „=“ a za znakem „#“ se vyskytují reakce na konkrétní odpovědi. Těmito zprávami systém reaguje na odpovědi studenta.

4.2.5 Ostatní formáty

Na závěr se zmíníme o několika dalších formátech používaných pro export otázek. Ty buď vycházejí z předchozích typů nebo nejsou až tak dalece rozšířeny.

Missing word – podporuje otázky multiple choice, multiple answer. Otázka se vkládá do textu pomocí znaků „{ }“. Odpovědi se oddělují znakem „~“ a správná odpověď se značí „=“.

AON – stejný jako předchozí, ale při importu otázek typu short answer se vždy čtyři převedou na odpovídající matching question. Otázky multiple choice a multiple answer jsou při importu náhodně zamíchány.

Dále by sem patřily různé unikátní formáty používané interně jedním produktem (WebCT, CTM, Moodle – zde dokonce možnost vytvořit si svůj vlastní formát).

4.3 Srovnání jednotlivých systémů

V této podkapitole se pokusíme shrnout možnosti jednotlivých systémů, konkrétně jejich testových modulů. Některé ze systémů jsou samostatné moduly, které obsahují pouze testovací část, naproti tomu stojí velké komplexní e-learningové systémy, přičemž testovací část je pouze jedním z mnoha prvků.

Dále můžeme rozlišovat systémy podle toho, co všechno umožňují uchovávat. Míni se tím například databáze studentů, databáze učitelů, rozřazení testů do témat a podobně. Za nejdůležitější měřítko budeme považovat počet použitelných typů otázek v testu a schopnost komunikovat s ostatními systémy (formáty importu/exportu dat), což se pokusíme shrnout formou tabulky.

Typ otázky/Systém	Moodle	WebCT	Blackboard	Respondus	Hot potatoes
Multiple choice	X	X	X	X	X
Multiple answer	X	X	X	X	X
True/false	X		X	X	
Fill-in-the-blank/Cloze	X		X	X	X
Ordering/Ranking			X	X	
Matching	X	X	X	X	
Short answer/ essay	X	X	X	X	X
Drag-n-drop					
Numerical	X	X			
Paragraph		X			

Tabulka 2.: Srovnání vybraných e-learningových a testovacích systémů

Vidíme, že spektrum záběru jednotlivých produktů se různí. Je ovšem nutno poznamenat, že občas dochází k překrývání typů a ne vždy pro ně všechny systémy používají stejné označení. Může tedy dojít k situaci, že i když systém podle tabulky neumí nějaký typ otázky, tak je tento typ skrytý v typu jiném.

5 Návrh aplikace

5.1 Role uživatelů v systému

V systému se vyskytují dvě základní role – studenti a lektori, přičemž výsadní práva má role administrátora.

Studenti

Uživatelé s nejmenšími právy. Po vstupu do systému mají minimální možnost editace svého profilu (změna hesla) a vidí naplánované testy (označené jako viditelné studentům). Pokud nějaký test zrovna probíhá, mají možnost vstoupit do něj.

Lektori

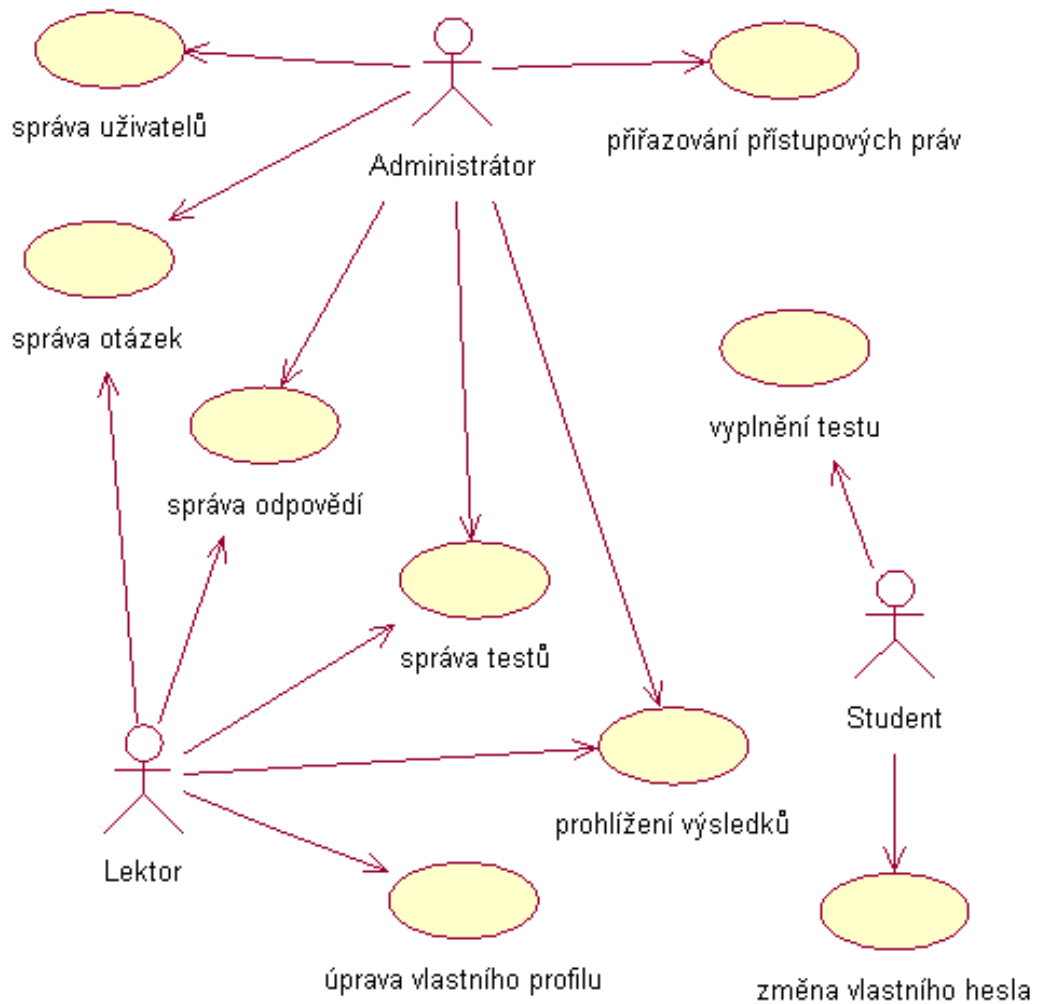
Uživatelé s kompletní možností správy otázek, testů a jejich kategorií. Dále mají možnost upravovat svůj profil (kromě změny práv) a prohlížet výsledky testů.

Administrátor

Stejná práva jako lektori, navíc mohou plně spravovat databázi uživatelů a měnit jejich přístupová práva a role v systému. Administrátorský účet je jeden a nedá se smazat. Lze však přidělit administrátorská práva i jiným uživatelům.

Systém rozliší jednotlivé role při přihlášení. Privilegované operace jsou zabezpečeny proti zneužití (např. podstrčením upravené url adresy – proměnné předávané metodou *get*).

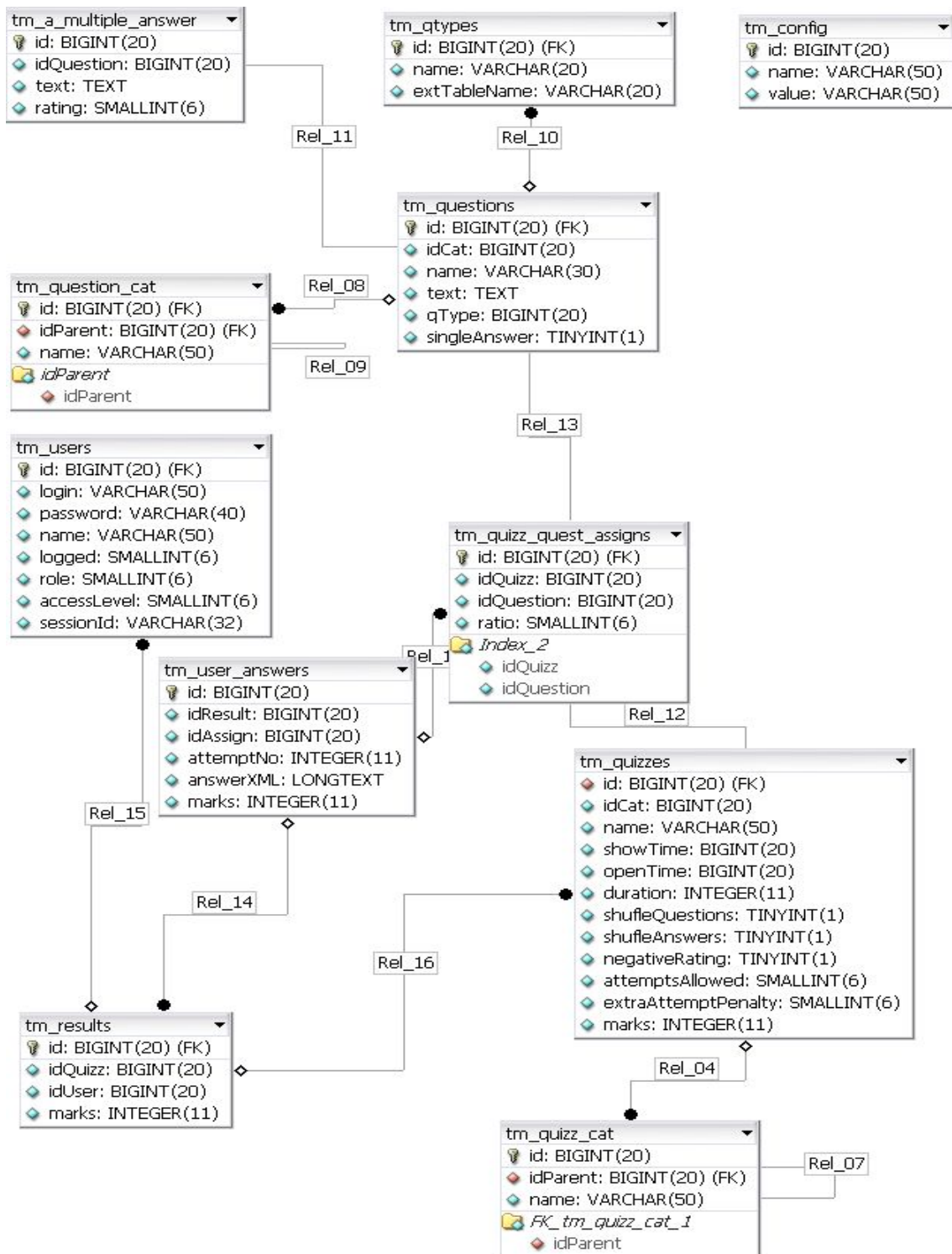
5.2 Use-Case diagram



Obrázek 5-1.: Use-case diagram

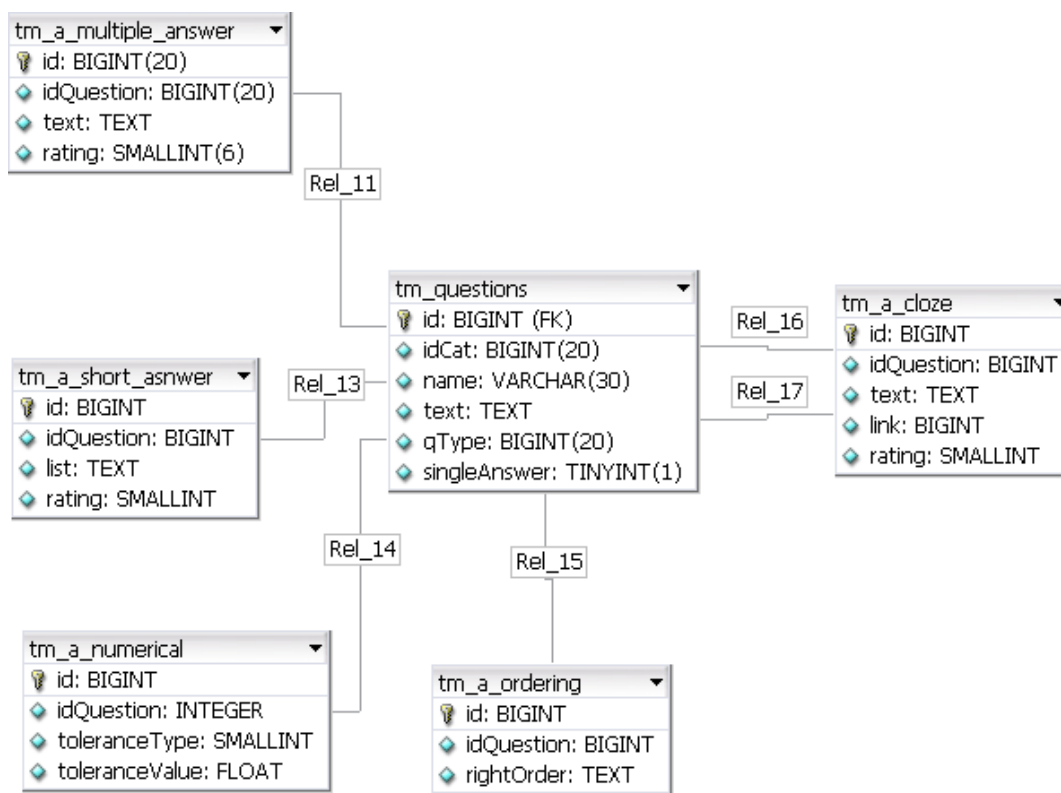
Jak je vidět role administrátora a lektora sdílejí většinu pravomocí.

5.3 E-R diagram



Obrázek 5-2.: Návrh databáze - ER diagram

Dále je možné tento návrh rozšiřovat podle požadovaných typů otázek následovně:



Obrázek 5-3.: Rozšíření databáze pro různé typy otázek

5.4 Popis databáze

Na základě shrnutých možností různých současných systémů bylo nutné vytvořit databázi, která bude schopná v první řadě uchovat všechny typy otázek. Dále je nutné uchovávat jejich bodové hodnocení, které se může různit v závislosti konkrétním testu (otázka může patřit do více testů různých autorů zároveň). A v neposlední řadě musíme samozřejmě myslet na ukládání hodnocení studentů. Tento návrh je jakýmsi konceptem, který dále může být obohacován o další funkce (několik pokusů na jeden test, používání proměnných v otázkách, atd.). Nám zde jde především a nastínění základního principu. Návrh je inspirován systémem Moodle.

Postupně popíšeme jednotlivé tabulky a účel, kterému slouží. Každá tabulka má primární klíč – sloupec **id**, který proto dále nebudu zmiňovat.

5.4.1 tm_quizz_cat, tm_question_cat

Tyto tabulky umožňují rozdělit testy a otázky do kategorií. Každá kategorie obsahuje odkaz na svou nadřazenou kategorii. Kořenová položka je předem vytvořena při instalaci a nemůže být smazána.

- **idParent** – vazba na nadřazenou kategorii
- **name** – zobrazované jméno kategorie

5.4.2 tm_quizzes

Zde uchováváme údaje o jednotlivých testech v systému. Každý test patří do nějaké kategorie, při jejímž odstraňování se test přiřadí kategorii nadřazené.

- **idCat** – vazba na kategorii
- **name** – název testu
- **showTime** – datum, od kterého je test viditelný pro studenty
- **openTime** – datum a čas, kdy se test otevře pro vyplnění
- **duration** – doba v minutách, po kterou bude test přístupný pro vyplnění
- **shuffleQuestions** – zamíchat otázky – každý student bude vyplňovat otázky v jiném, náhodném pořadí
- **shuffleAnswers** – zamíchat odpovědi – v otázkách s několika nabízenými odpověďmi se tyto zobrazí v náhodném pořadí
- **negativeRating** – určuje, zda má potencionální negativní výsledek otázky ovlivňovat výsledek testu (jinak se každý dílčí záporný výsledek počítá jako nula)
- **attemptsAllowed** – počet povolených pokusů na každou otázku
- **extraAttemptPenalty** – pokud je povoleno více pokusů, je možné další pokusy navíc penalizovat
- **marks** – počet bodů, které lze v testu získat

5.4.3 tm_questions

Tato tabulka slouží k uchovávání informací o otázkách společných všem druhům otázek.

- **idCat** – vazba na kategorii otázek
- **name** – název otázky
- **text** – text otázky
- **qType** – odkaz na typ otázky
- **singleAnswer** – povolení pouze jediné odpovědi

5.4.4 tm_qTypes

Obsahuje záznamy o typech otázek použitelných v aplikaci. Váže jednotlivé typy na speciální tabulky s rozšiřujícími položkami pro každý typ.

- **name** – název typu
- **extTableName** – název tabulky s rozšiřujícími položkami

5.4.5 tm_quizz_quest_assigns

Tato tabulka realizuje relaci n:m mezi tabulkami **tm_quizzes** a **tm_questions**. Jako parametr této relace přidáváme podíl konkrétní otázky na hodnocení konkrétního testu.

- **idQuizz** – vazba na přiřazovanou otázku
- **idQuestion** – vazba na přiřazovaný test
- **ratio** – podíl otázky na hodnocení testu v procentech

5.4.6 tm_users

Zde ukládáme informace o uživatelích využívajících službu aplikace spolu s informacemi o přihlašování.

- **login** – jméno používané při přihlašování
- **password** – heslo pro přihlášení uložené ve formě hashe sha1
- **name** – příjmení, jméno uživatele
- **logged** – informace o tom, zda-li je uživatel právě přihlášen
- **role** – role uživatele v systému (student, lektor, administrátor)
- **accessLevel** – oprávnění uživatele v systému – může se lišit od role (např. lektor může mít navíc administrátorská práva)
- **sessionId** – session identifikátor přihlášeného uživatele – předchází bezpečnostním rizikům

5.4.7 tm_results

Tato tabulka slouží k ukládání celkových výsledků jednotlivých testů dosažených každým studentem. Tuto informaci lze sice získat ze součtu jednotlivých odpovědí, nicméně v rámci zrychlení aplikace je vhodné tuto redundanci použít. Navíc záznamy v této tabulce plní důležitou funkci při rozhodování, zda-li již daný uživatel nevyplňoval konkrétní test.

- **idQuizz** – vazba na hodnocený test
- **idUser** – vazba na uživatele, jehož hodnocení v bodech se záznam týká
- **marks** – bodové hodnocení, jehož uživatel v testu dosáhl

5.4.8 tm_user_answers

Zde uchováváme informace o odpovědích studentů na jednotlivé otázky včetně historie odpovědí pokud je povolena opakovaná odpověď.

- **idResult** – vazba celkové hodnocení (informace o uživateli, kterého se odpověď týká)
- **idAssign** – vazba na pár otázka – test (určuje které otázky a testu se odpověď týká)
- **attemptNo** – číslo pokusu o odpověď (historie pokusů)
- **answerXML** - odpověď studenta uložená ve formátu XML – liší se v závislosti na typu otázky
- **marks** – bodové hodnocení každého pokusu

5.4.9 tm_config

Slouží k uložení konfiguračních proměnných.

- **name** – název proměnné
- **value** – hodnota proměnné

5.5 Tabulky odpovědí na jednotlivé typy otázek

Jelikož každý druh otázky má specifické požadavky na uložení odpovědí, je výhodnější pro každý typ vytvořit zvláštní tabulku, která bude odpovídat daným požadavkům bez zbytečných nevyužitých sloupců. Název této tabulky je pro každý typ uveden v tabulce typů qTypes. Společně pro všechny odpovědi budou následující vlastnosti:

- **idQuestion** – vazba na otázku, které se
- **rating** – procentuální vyjádření správnosti odpovědi, součet těchto hodnot u všech odpovědí na jednu otázku musí být 100 %. Odpovědi mohou mít i záporné hodnocení, přičemž součet všech záporných hodnocení musí dát -100 %.

Dál následuje výčet tabulek specifických vlastností pro odpověď na každý typ otázky.

5.5.1 tm_a_multiple_answer

Zde je možné ukládat odpovědi pro otázky typu *Multiple choice*, *Multiple answer*, *True/False*. Jedná se o klasické testové otázky, kde odpovědí je zvolený textový řetězec. Proto tabulka **tm_a_multiple_answer** obsahuje navíc tuto položku:

- **text** – řetězec obsahující text odpovědi

5.5.2 **tm_a_short_answer**

Tato tabulka z hlediska uložení dat odpovídá předchozímu typu, rozdílná je však logika zpracování. Zde každá odpověď obsahuje seznam slov, která pokud se vyskytnou v odpovědi studenta, jsou hodnocena dle odpovídajícího pole **rating**. Zde součet těchto polí nemusí být roven 100 % a při hodnocení studentovy odpovědi se vždy vybere nejlépe vycházející varianta, jejíž dílčí slovo se nachází ve studentově odpovědi.

5.5.3 **tm_a_numerical**

Zde se kromě správné číselné odpovědi na otázku uchovává také tolerance (typ a hodnota):

- **toleranceType** – typ tolerance (absolutní, relativní, geometrická)
- **toleranceValue** – hodnota tolerance

5.5.4 **tm_a_cloze**

Tento formát poněkud vybočuje, jedná se v podstatě o kontejner pro ostatní otázky, které sdružuje do jedné. Proto i tabulka odpovědí fakticky neobsahuje odpovědi jako takové, ale odkazy na otázky jiných typů spolu s nepovinným komentářem:

- **text** – komentář (text dílčí otázky je přístupný přes vlastnost **link** z tabulky otázek), který může upřesňovat a vyjasňovat požadovanou odpověď v daném kontextu
- **link** – vazba na otázku

5.5.5 **tm_a_ordering**

Odpověď na tento typ otázky je vždy jen jedna správná. Parametr **singleAnswer** dané otázky určuje, zda je požadována přesně správná odpověď nebo má logika hodnocení uvažovat i částečně správnou odpověď, která se hodnotí podle počtu vzájemně relativně správně seřazených odpovědí. V tomto případě probíhá hodnocení autonomně. Řazené prvky jsou součástí otázky, odpověď obsahuje správnou posloupnost klíčů.

- **rightOrder** – správné pořadí klíčů

5.6 Rozšíření

Takto navržená databáze je schopná pojmout celé spektrum otázek, kromě drag-n-drop, jejíž implementace není triviální (na zvolené platformě v podstatě nemožná – je nutná další technologie jako Java nebo Flash).

Samozřejmě, že existují široké možnosti rozšíření, které již byly zmíněny. Nicméně jejich zahrnutí do schématu ho činí již značně nepřehledným a neposkytují nic, co by nešlo provést jiným způsobem, pouze zvyšují uživatelský komfort. O zapracování některých z nich lze v budoucnu uvažovat.

6 Implementace

V této kapitole budou popsány použité technologie a způsob implementace.

6.1 Použité technologie

Programovacím jazykem bylo PHP 5 (5.1.4). Oproti předchozím verzím poskytuje mimo jiné zásadně vylepšenou podporu objektově orientovaného programování. Jako webový server byl použit Apache (2.0.58). Dále využíváme databázi MySQL 5 (5.0.22), nicméně systém není vázaný na konkrétní databázový server a databázová vrstva je připravena pracovat pod jakýmkoliv serverem přístupným z PHP. Data se balí do formy XML a ještě na serveru se pomocí šablony XSL převádí do XHTML. Výhodou tohoto řešení je snadná změna struktury stránky (vzhledu) pouhou změnou šablony bez nutnosti jakkoliv upravovat skripty.

6.1.1 PHP Session

Mechanismus session byl vyvinut z důvodu bezstavovosti protokolu http. Jinak řečeno http požadavky jsou zpracovávány bez jakékoliv vzájemné návaznosti nebo závislosti. Session (relace) slouží k uchování dat mezi jednotlivými požadavky. Totéž je možné uskutečnit pomocí metod GET a POST, to ale není příliš bezpečné, jelikož veškerá data neustále putují po síti. Session – proměnné se ukládají na serveru a předává se pouze identifikátor konkrétní relace. K tomuto účelu je po nastartování session zpřístupněna konstanta SID, která se může připojit k odkazům na další stránky. Tento způsob využívá metodu GET a předává identifikátor v adrese dokumentu. Je také možné využít cookies, pokud jsou povoleny. Pak je konstanta SID rovna prázdnému řetězci. Praktické využití najdou sessions například v identifikaci uživatelů.

S tímto systémem jsou ale přesto spojena jistá bezpečnostní rizika. Jedním z nich je tzv. session-stealing. Pokud někdo získá cizí session-identifikátor, může v systému vystupovat jako jeho pravý majitel. Toto lze ošetřit například ukládáním IP adresy do databáze a její kontrolou při každém přístupu na chráněné stránky nebo autorizačním klíčem, který se neustále mění.

Session využijeme k uchování dat a serializovaných (zjednodušeně – převedených na binární data) objektů.

6.2 Databázová vrstva

PHP poskytuje pro práci s databází sadu funkcí specifických vždy pro určitý databázový server. Práce s databází potom probíhá zasíláním jednotlivých SQL příkazů ve formě řetězce. Výsledku dotazů vrací po zpracování buď jako pole nebo objekty. Tento způsob se při větších projektech stává nepoužitelným pro svou pracnost, časté opakování již napsaného kódu, špatnou udržitelnost a hlavně nepřenositelnost.

Pro odstranění těchto nevýhod se používá styčná vrstva funkcí/tříd, které se starají o práci s databází a poskytují pohodlné rozhraní pro práci s databází. Častým přístupem je pro každou tabulku v databázi vytvářet vlastní třídu, která se potom stará o práci s určenou tabulkou. Přičemž tyto třídy využívají pro práci s databází sadu obecných metod, jejichž konkrétní implementace se liší právě podle použitého typu serveru. Tak lze dopředu připravit několik tříd s těmito metodami uzpůsobenými pro různé servery a vždy vybírat tu nejvhodnější. Tímto způsobem odstíníme pro zbytek aplikace rozdíly v databázových serverech a učiníme ji tak nezávislou na jednom typu.

Nadále ovšem zůstává „nešikovná“ manipulace s databází, kdy v naprosté většině případů potřebuji pouze jednoduché typy dotazů pro základní operace, které ovšem musím pro každou tabulku uzpůsobit. Dále nám vadí, že v PHP při dotazu typu SELECT dostáváme jednotlivé záznamy výsledku buď jako objekt nebo pole, přičemž můžeme data pouze číst. Zápis do databáze si žádá další specifický dotaz. Tato „nepohodlnost“ je vyřešena pomocí dvou základních tříd, které jsou natolik flexibilní, že jsou schopny poskytnout základní operace nad jakoukoliv tabulkou (jejíž název dostanou jako parametr konstruktoru) a nejen to. Pokud dotazem typu SELECT vybíráme z databáze maximálně jeden prvek, pak mi toto rozhraní vrátí stejný objekt jako standardní funkce (`fetch_object`) ale navíc s možností zapisovat přes něj zpět do databáze. Nyní si tyto dvě třídy popíšeme podrobněji.

6.2.1 C_db_common

Této třídě v konstruktoru předáváme jméno tabulky, se kterou má nový objekt pracovat. Tímto získáme základní rozhraní s danou tabulkou. Jedná se především o metody:

- **create** – vytvoří nový záznam v databázi a vrátí speciální objekt s rozšířenými možnostmi správy tohoto záznamu
- **delete** – maže záznam
- **deleteAll** – maže všechny záznamy (potažmo seznam záznamů)
- **getObject** – vrací speciální objekt podle předané hodnoty *id*
- **getObjectKey** – vrací speciální objekt vybraný podle hodnoty jakéhokoliv sloupce, nebo odkaz na výsledek standardní funkce (pokud čekáme více vybraných záznamů než jeden nastavíme odpovídající parametr metody)

- **update** – očekává jako parametr speciální objekt pro správu záznamu a s využitím sebereflexe objektů zjistí jaké vlastnosti jsou potřeba uložit do databáze a provede operaci UPDATE

Kdykoliv potřebujeme tuto základní sadu metod obohatit o speciální metody pro práci s konkrétní tabulkou, použijeme tuto třídu jako předka a doplníme pouze potřebné metody.

Dalším parametrem konstruktoru je název předka onoho speciálního objektu, který mohou vracet výše zmíněné metody. Věc vyjasníme v následující podkapitole.

Spojení s databází tvoří statická proměnná **\$db**, která odkazuje na objekt přímo komunikující s databází.

6.2.2 C_field_base

Jak je zmíněno výše, standardně dostáváme data z databáze ve formě struktur, které již ale nelze použít také pro úpravu záznamů jejichž data obsahují. Tato třída obsahuje mechanismy, které toto umožňují.

Instance této třídy vznikají zásadně prostřednictvím metod třídy **C_db_common** a jejich potomků. Při vzniku instance třídy **C_db_common** se vytvoří deklarace nové třídy, která dědí potřebné metody z třídy **C_field_base** a přidává datovou strukturu podle obsluhované tabulky tak, aby výsledná třída obsahovala vše potřebné pro obsluhu jednoho záznamu v dané tabulce.

Pokud navíc potřebujeme speciální operace se záznamem, deklarujeme potomka třídy **C_field_base**, kde ony operace definujeme a jméno tohoto potomka potom předáme třídě **C_db_common** (popřípadě jejímu potomkovi), která pak bude vytvářet žádané objekty opět obohacené o datovou strukturu pro uložení dat záznamu.

Tato datová struktura navíc uchovává informaci, které vlastnosti byly změněny (systémem příznaků změny) a tuto pak využívá objekt při zpětném zápisu svých dat do databáze. Třída **C_field_base** obsahuje:

- **\$parent** - odkaz na svého „stvořitele“
- **get(\$vlastnost)** – vrací hodnotu požadované vlastnosti
- **set(\$vlastnost, \$hodnota)** – nastavuje vlastnost na danou hodnotu, pokud to znamená změnu hodnoty, tak aktualizuje příslušný příznak změny
- **flush()** – tato metoda uloží všechny změny do databáze, používá k tomu metodu **update** svého „stvořitele“, které předá jako parametr odkaz na sama sebe (objekt jehož je členem)

6.2.3 Zbývající třídy databázové vrstvy

Pokud jsou pro kteroukoliv tabulku potřeba pokročilejší činnosti závislé na konkrétní tabulce, je pro ni vytvořena zvláštní třída jako potomek `C_db_common`, do které se potřebné metody přidají. Stejně tak pokud je potřeba speciální operace se záznamy, využije se stejným způsobem třída `C_field_base`. Tento způsob řešení databázové vrstvy je použitelný bez velkých úprav v libovolné databázi na libovolné tabulky.

Jedná se o tyto třídy:

- `C_answers` – správa odpovědí
- `C_categoriesTools` – společné funkce pro kategorie
- `C_question_cat` – správa kategorií otázek
- `C_questions` – správa otázek
- `C_quizz_cat` – správa kategorií testů
- `C_quizzes` – správa testů
- `C_results` – správa celkových výsledků
- `C_user_answers` – správa odpovědí studentů
- `C_users` – správa uživatelů

6.3 Činnost systému

Další detaily implementace budou názorně vysvětleny současně s popisem činnosti systému, což umožní čtenáři získat lepší představu funkci jednotlivých modulů.

6.3.1 Přihlášení



Obrázek 6-1.: Přihlašovací dialog

System umožňuje činnost pouze přihlášeným uživatelům. Přihlášení proběhne zadáním loginu a hesla. Pokud odpovídající uživatel v systému existuje, pokračuje se v závislosti na roli uživatele. Do databáze se uloží identifikátor session, příznak přihlášení a v neposlední řadě čas vypršení platnosti přihlášení. Ten se nastavuje v konfiguračním skriptu. Zároveň se do session uloží serializovaný objekt uživatel pro použití na dalších stránkách. Na každé další stránce se potom kontroluje zda je uživatel

řádně přihlášený a zda má pro požadovanou činnost patřičná oprávnění. Pokud se zjistí, že oprávnění je nedostatečné, je to známkou útoku na systém, jelikož systém přímo nikomu nenabídne činnost, pro kterou by řádně přihlášený uživatel neměl oprávnění.

Při neúspěšném přihlášení je zobrazeno upozornění a nabídnuto opětovné přihlášení.

6.3.2 Role lektor

Pokud je uživatel rozpoznán jako lektor, je mu nabídnuto odpovídající rozhraní.

Správa uživatelů mu nabídne seznamy registrovaných uživatelů s upozorněním na právě přihlášené uživatele a úpravu vlastního profilu. Lektor nemůže měnit svou roli ani práva v systému.

Přihlášen jako: Ostrý František (Odhlásit)

Uživatelé

- Lektoři
- Studenti
- Můj profil

Otázky

-- Vše --

- Testovky [e] [x]

-- nová kategorie --

Testy

-- Vše --

- Laboratoře [e] [x]
- Pulsesemestrálky [e] [x]
- Semestrálky [e] [x]

-- nová kategorie --

XML

- Export databáze
- Import databáze

Testy - kategorie: Vše

-- Nový záznam -- [x] [- odznačit vše]

Lab 2	[x]	[+ přiřazení]
Lab 3	[x]	[+ přiřazení]
Lab 4	[x]	[+ přiřazení]
Půlsesemestrální test 2007	[x]	[+ přiřazení]
Semestrální test 2005	[x]	[+ přiřazení]
Semestrální test 2006	[x]	[+ přiřazení]
Semestrální test 2007	[x]	[+ přiřazení]

Test ME! - Pavel Sláma 2007

Obrázek 6-2.: Rozhraní systému pro roli Lektor

Správa otázek nabídne stromové menu s kategoriemi otázek s možností editace. Při smazání kategorie otázek se obsažené otázky přiřadí kategorii, která je nadřazená smazané. Vrcholem je kořenová kategorie, která smazat nelze. Po kliknutí na kategorii se zobrazí editovatelný seznam otázek. Otázky lze plně editovat a přiřazovat k libovolným kategoriím.

Každá otázka nabízí přístup ke správě odpovědí. Zde systém neustále kontroluje součet hodnocení všech odpovědí a přepočítává je tak aby zachovávaly vzájemný poměr avšak součet odpovídal logice.

Správa testů má podobné rozhraní jako správa otázek. Důležitou funkcí je **přiřazování otázek k testům**. To lze několika způsoby:

- Hromadně spoluprací správy otázek a správy testů – každá položka na seznamu testů a otázek obsahuje možnost „[+/- přiřadit]”. Systém přiřazování pak funguje tak, že lektor nejprve otevře pro přiřazování testy (lze víc najednou) a potom přiřazuje otázky do všech otevřených testů najednou. Stejným způsobem lze otázky ze všech otevřených testů najednou odstraňovat. Touto cestou lze snadno a rychle přiřadit velké množství otázek velkému množství testů.
- Druhý způsob je využití části správy testů, která umožňuje přesně vybírat přiřazení otázek ke každému jednotlivému testu. Rozhraní nabídne formulář s dvěma okny, v jednom jsou přiřazené otázky k onomu testu, v druhém nepřiřazené. Lektor zaškrtně otázky, které si přeje přesunout z jednoho okna do druhého a zvolí směr. Jedině tímto způsobem je možné upravovat vliv jednotlivých otázek na celkové hodnocení testu. Systém opět po každé změně přepočítává hodnocení otázek tak aby zůstal zachován zadaný poměr a součet odpovídal 100 %. Přiřadit lze pouze otázky s nenulovým počtem odpovědí.

Důležitou vlastností je čas zviditelnění testu studentům, těm se od daného dne bude test zobrazovat jako naplánovaný, a čas otevření testu spolu s dobou trvání. Dále pokud se jedná o test, který již proběhnul, jsou k dispozici výsledky. Zde jsou vidět jak celková hodnocení jednotlivých studentů tak podrobnější výpis průběhu testu u každého testovaného. Konkrétně výsledek každého pokusu u každé otázky.

Nakonec je možné exportovat nebo importovat celou databázi do formátu XML za účelem migrace systému na jiný server nebo zpracování dat jiným systémem.

6.3.3 Role Administrátor

Přihlášen jako: Administrátor (Odhlásit)

Uživatelé
Lektoři
Studenti
Můj profil

Otázky
-- Vše --
■ Testovky [e] [x]
-- nová kategorie --

Testy
-- Vše --
■ Laboratoře [e] [x]
■ Pulsestrálky [e] [x]
■ Semestrálky [e] [x]
-- nová kategorie --

XML
■ Export databáze
■ Import databáze

Profil: Administrátor

Login:
Heslo:
Změnit heslo:
Jméno:
Role v systému: student lektor administrátor
Přístupová práva: student lektor administrátor

Test MEI - Pavel Sláma 2007

Obrázek 6-3.: Administrátorské rozhraní - úprava profilu

Administrátor má stejná oprávnění jako lektor, obohacená o kompletní správu uživatelů. Administrátor jediný může vytvářet a mazat uživatelské účty a upravovat příslušná oprávnění. Administrátorský účet je pouze jeden, lze však přidělit tato práva i jinému uživateli (např. nějakému lektorovi). Původní administrátorský účet ovšem nelze smazat.

6.3.4 Role Student

Student má ze všech rolí nejmenší privilegia.

Přihlášen jako: Novák Josef (Odhlásit)

Čas
28.05.2007 6:35

Uživatelé
Lektoři
Studenti
Můj profil

Plánované testy
■ Semestrální test 2007 (2.6.2007 14:00, 60 min)

Vítejte v systému - nejbližší test:

Semestrální test 2007 - TEST UZAVŘEN

Čas otevření: 02.06.2007 14:00

Test MEI - Pavel Sláma 2007

Obrázek 6-4.: Rozhraní role Student

Ze svého profilu může měnit pouze své přístupové heslo.

Správa testů zobrazí seznam naplánovaných testů spolu s datem a časem otevření a délkou trvání. Zároveň je student upozorněn na nejbližší naplánovaný test. Toto všechno platí v případě, že momentálně není žádný test otevřený. Pokud ano, tak je tento zobrazen červeně spolu se zbývajícím časem do ukončení. Student může vstoupit do vyplňování testu.

The screenshot shows a web interface for a student. At the top right, it says 'Přihlášen jako: Novák Josef (Odhlásit)'. On the left is a sidebar with a blue header and several menu items: 'Čas' (28.05.2007 6:47), 'Uživatelé' (Lektoři, Studenti, Můj profil), and 'Plánované testy' (Semestrální test 2007 (zbývající čas: 0:12)). The main content area has a blue header 'Vítejte v systému - nejbližší test:' and a large red heading 'Semestrální test 2007 - TEST OTEVŘEN'. Below this, it shows 'Čas otevření: 28.05.2007 6:00' and 'Zbývá do konce: 0:12'. At the bottom, it says 'Test MEI - Pavel Sláma 2007'.

Obrázek 6-5.: Rozhraní role Student - otevřený test

Ještě než se studentovi zobrazí první otázka, systém zjistí podrobnosti o testu. Nejdřív předem připraví sekvenci otázek, kterou v případě volby „zamíchat otázky“ náhodně přeskládá. Vstoupením do testu se zobrazí nové okno s první otázkou. V tabulce celkových výsledků **C_results** se vytvoří záznam, který od této bude bránit novému vstupu do testu. Čili i kdyby se nyní student odhlásil, tak již mu nebude umožněno znovu vstoupit do testu. Studentovi se s každou otázkou zobrazuje zbývajícím čas, číslo otázky, číslo pokusu, počet bodů za aktuální otázku a počet doposud získaných bodů). Potom co student vybere odpověď a potvrdí výběr se zobrazí hodnocení jeho odpovědi. Pokud je v testu povoleno více pokusů na otázku, je mu nabídnuto spolu s postupem k další otázce opakování stávající. Až student vyčerpá všechny pokusy, je nucen přejít k další otázce. Po poslední otázce se test ukončí a studentovi se zobrazí dosažené hodnocení spolu s výpisem průběhu testu. V průběhu testu se hojně využívá proměnných v session, čímž se minimalizuje zatížení databáze. Po dokončení testu je dále hodnocení k dispozici lektorům. Jednou vyplněný test již v rozhraní pro studenta nefiguruje, pokud se do něj pokusí vstoupit podruhé ještě v době kdy je otevřený pak se mu pouze zobrazí dosažené výsledky.



Obrázek 6-6.: Probíhající test

6.4 Popis jednotlivých programových modulů

V této kapitole projdeme jednotlivé součásti systému.

6.4.1 Tvorba XML pomocí DOM rozhraní

Pro účely tvorby XML využíváme objektové rozhraní v PHP na bázi standardu DOM (Dokument Object Model). Celý výsledný dokument chápeme jako strom objektů, které vyjadřují jednotlivé XML uzly. Pod pojmem XML uzel (node) se skrývají veškeré značky, stejně tak atributy značek a textový obsah. Každý tento objekt patří nadřazenému objektu, čímž vzniká jasně daná hierarchie (např. uzel atribut má nadřazený uzel značka, ke které patří). Vrcholem tohoto stromu jsou kořenové značky a ty jsou potomky přímo dokumentu.

Práce s tímto rozhraním je velmi jednoduchá a zbavuje programátora nutnosti neustále hlídat dodržování přísných standardů XML. O to se stará samotný mechanismus DOM. Jednotlivé uzly vždy tvoříme volením metod **createElement**, **createAttribute** instance třídy **DOMDocument**. K navázání uzlů slouží metoda **appendChild**, kterou lze ke každému elementu (uzlu typu značka) navázat potomka. Může jít buď o atribut nebo o další vnořenou značku.

Na závěr využijeme metodu **saveXML** třídy **DOMDocument**, která vrátí XML dokument dle vytvořeného stromu objektů. Tento dokument dále pomocí XSL šablony transformujeme do požadované formy – v našem případě XHTML.

6.4.2 Tvorba formulářů

K tvorbě formulářů využíváme třídu **C_form** a její potomky. Hlavní metoda **getNode** vrací XML uzel s obsahem formuláře. Každý potomek modifikuje metodu **addSpecials**, která podle účelu formuláře přidává potřebné vstupní prvky. Vychází se přitom z předaného datového objektu, případně názvu tabulky, jejíž editaci má formulář umožňovat. Právě v metodě **addSpecials** specifikujeme speciální požadavky na to které datové pole. Standardně se využívá vstupní pole typu *text*. Pokud pro

nějakou datovou položku potřebujeme speciální vstupní pole využívá se metod **getRadioNode**, **getSelectNode**. Těm předáme data a jako výsledek obdržíme XML kód pro odpovídající vstupní pole. Názvy těchto polí se generují automaticky podle názvu sloupce v záznamu tabulky.

Před zpracováním formulářů se volá metoda **checkForm**, do které lze umístit nutné kontroly a ošetřit proměnné předané z formuláře metodou POST. Pokud jsou formulářová data v pořádku, voláme metodu **saveData**. Ta je specifická pro každý formulář a stejně jako v metodě **addSpecials** definujeme speciální požadavky na zobrazení vstupních polí, tak zde definujeme způsob práce s předanými proměnnými a způsob uložení do databáze.

6.4.3 Správa uživatelů a editace profilu

Studenti	
-- Nový záznam --	
Hradil David	[x]
Novák Josef(online)	[x]
Sedlář Jan	[x]

Obrázek 6-7.: Správa uživatelů

Tento modul zpřístupňuje operace s uživateli. Seznam obsahuje informaci, kteří uživatelé jsou právě přihlášení. Přidávat a mazat účty může jen administrátor. Lektoři mohou prohlížet seznamy uživatelů a editovat pouze svůj profil. Studenti mohou pouze editovat svůj profil a to změnou hesla. Formulář pro editaci profilu je na obrázku č. 6-8.

6.4.4 Správa otázek a odpovědí

Otázky - kategorie: Testovky

-- Nový záznam -- [x] [- smazat přiřazení]		
Dotaz 1(2)	[x]	[+ přiřazení]
Počty 1(3)	[x]	[+ přiřazení]
Zápočtová(4)	[x]	[+ přiřazení]

Obrázek 6-9.: Správa otázek

Lektorům a administrátorovi tento modul nabídne seznam otázek zvolené kategorie. Čísla v závorce udávají počet odpovědí asociovaných s otázkou. Přiřazením lze otázku přidat do testů, které jsou momentálně otevřené pro přiřazování. Kliknutím na jméno se otevře editace otázky:

Otázka: Počty 1

Kategorie:

Jméno:

Text:

Typ otázky: [Upravit odpovědi](#)

[< zpět na kategorii <](#)

Obrázek 6-10.: Editace otázky

Zde vidíme příklad speciálního zpracování pole kategorie třídou **C_dbForm**, kde místo skutečné hodnoty cizího klíče do tabulky kategorií se objevuje formulářový prvek *select* umožňující výběr z existujících kategorií. Ke každé otázce je nutné přiřadit množinu odpovědí. K tomu slouží správa odpovědí:

Otázka Počty 1 (Testová otázka) - Odpovědi

1 + 1 = ?

Povolit pouze jedinou odpověď

Odpověď	Hodnocení [%]
<input type="text" value="1"/>	<input type="text" value="-50"/> [x]
<input type="text" value="2"/>	<input type="text" value="100"/> [x]
<input type="text" value="3"/>	<input type="text" value="-50"/> [x]
<input type="text" value="4"/>	<input type="text" value=""/>

[< zpět na otázku <](#)

Obrázek 6-11.: Správa odpovědí

Zde je možné zadat kromě textu nabízených odpovědí rovněž hodnocení jejich správnosti. Podmínkou je, že součet pozitivních hodnocení musí být vždy 100 %, stejně tak součet negativních

hodnocení musí být -100 %. Toto zajišťuje vnitřní logika modulu, která se snaží udržovat i vzájemné poměry hodnocení jednotlivých odpovědí. Pokud je možná pouze jediná správná odpověď, měla by tato mít hodnocení 100 % a ostatní odpovědi buď 0 % nebo záporné hodnocení.

6.4.5 Správa testů, přiřazování otázek, výsledky

Správa testů poskytuje rozhraní podobné správě otázek:

Testy - kategorie: Vše

-- Nový záznam --	<input type="checkbox"/>	[- odznačit vše]
Lab 2	<input type="checkbox"/>	[+ přiřazení]
Lab 3	<input type="checkbox"/>	[+ přiřazení]
Lab 4	<input type="checkbox"/>	[+ přiřazení]
Půlsemestrální test 2007	<input type="checkbox"/>	[+ přiřazení]
Semestrální test 2005	<input type="checkbox"/>	[+ přiřazení]
Semestrální test 2006	<input type="checkbox"/>	[+ přiřazení]
Semestrální test 2007	<input type="checkbox"/>	[+ přiřazení]

Obrázek 6-12.: Správa testů

Neustále se zobrazují testy, které patří do zvolené kategorie testů a do jejich podkategorií. Funkce přiřazení zde povoluje přiřazování a mazání otázek daným testům pomocí rozhraní správy otázek. Po kliknutí na název testu se dostáváme do editace údajů o testu:

Test: Půlsemestrální test 2007 [Otázky](#) [Výsledky](#)

Kategorie: ▼
Jméno:
Ukázat studentům (d.m.rrrr):
Otevřít (d.m.rrrr h:mm):
Doba trvání [min]:
Promíchat otázky
Promíchat odpovědi
Povolit záporné hodnocení otázky
Počet pokusů na otázku (0 - neomezeno):
Penalizace opakovaných pokusů [%]:
Maximální počet bodů:

[< zpět na kategorii <](#)

Obrázek 6-13.: Editace testu

Zde je nutné zadat zobrazené položky. Data jsou kontrolována regulárními výrazy a pokud zadaný řetězec neodpovídá, odmítne modul aktualizovat příslušnou položku. Tuto kontrolu vykonává metoda **checkForm** daného formuláře. Datum se v databázi ukládá ve formě unixového časového razítka. Zde je nutné podotknout, že z neznámých důvodů na systémech windows v závislosti na nastavení času v BIOSu a v operačním systému, může docházet k chybám o celý počet hodin. Zřejmě se jedná o problém s časovými pásmy a přechodem mezi letním/zimním časem. Tento problém je vyřešen od PHP 5.1.4, kdy je potom nutné při instalaci nastavit používané časové pásmo.

Důležitá je volba povolení záporného hodnocení, v tom případě může mít každá otázka záporný vliv na celkový součet hodnocení testu. V opačném případě je každá záporně hodnocená odpověď studenta na otázku hodnocena nulou. Další vlastností je počet pokusů na jednu otázku, s čímž souvisí penalizace každého opakovaného pokusu. Ta udává o kolik procent bude snížen potenciaální bodový zisk z opakované odpovědi.

Rozhraní dále poskytuje možnost přesného přiřazení otázek k testu:

Test: Půlsestrální test 2007 << zpět

Přiřazené otázky + Hodnocení [%]		Nepřiřazené otázky
<input type="checkbox"/> Dotaz 1(2)	25	<input type="checkbox"/> Logika 2(2)
<input type="checkbox"/> Zápočtová(4)	51	<input type="checkbox"/> Logika 1(0)
<input type="checkbox"/> Počty 1(3)	24	

Obrázek 6-14.: Editace testu - přiřazování otázek

V levé části formuláře jsou otázky přiřazené k danému testu. Čísla v závorkách opět znamenají počet přiřazených odpovědí ke každé otázce. Hodnocení vyjadřuje procentuální význam otázky na hodnocení celého testu. Tyto hodnoty opět udržuje vnitřní logika ve stálém vzájemném poměru, přičemž součet dává dohromady 100 %. Může se stát, že součet kvůli chybám v zaokrouhlení nebude přesný a bude o jeden procentní bod přesahovat. To ale na funkci vyhodnocení nebude mít vliv, neboť nedovolí přesáhnout maximální bodové ohodnocení testu.

V pravé části vidíme otázky nepřiřazené k tomuto testu.

Pomocí zatrhávacích boxů označíme otázky, se kterými chceme manipulovat a příslušným tlačítkem je přesuneme do požadované polohy. K testu lze přiřadit pouze otázky s nenulovým počtem přiřazených odpovědí. Pokud není zvolena vlastnost testu *promíchat otázky*, budou všichni studenti odpovídat na otázky v tomto pořadí.

Po vyplnění testu studenty jsou k dispozici souhrnné výsledky:

Test: Půlsestrální test 2007 - výsledky

[<< zpět](#)

Student	Body (max 50)
Sedlář Jan	0
Hradil David	41

Obrázek 6-15.: Souhrnné výsledky testu

Po kliknutí na studenta získáme podrobný výpis průběhu testu, čili seznam všech pokusů ke všem otázkám spolu s hodnocením.

6.4.6 Modul pro vlastní průběh testu

Vstoupením do vyplňování testu student spustí modul **runQuizz.php**. Celý průběh testu tento skript po každé akci studenta volá znovu sám sebe, přičemž podle předaných formulářových hodnot pozná v jaké fázi se vyplňování nachází a co je třeba vykonat dále. Nejdříve je vygenerována posloupnost otázek v závislosti na parametrech testu a do pole *session* se uloží veškeré informace nutné pro běh testu:

- Pole identifikátorů otázek
- Objekt reprezentující záznam v tabulce celkových výsledků, pokud takový záznam týkající se konkrétního studenta a konkrétního testu existuje již před započítáním testu, znamená to, že student již test absolvoval – v tom případě se přeskočí pokládání otázek a studentovi se zobrazí výsledek dosažený v předchozím absolvování
- Objekt reprezentující test
- Pořadí právě pokládané otázky v testu
- Spočítaný maximální bodový zisk za pokládanou otázku (na základě maximálního bodového hodnocení testu a procentuálního podílu dané otázky na výsledku)
- Číslo pokusu o odpověď na položenou otázku
- Objekt reprezentující právě položenou otázku
- Příznak konce testu – nastavuje se po vyčerpání všech otázek

Po celou dobu testu se střídají dva dialogy:

- Buď systém čeká na vyplnění odpovědi na otázku, po jejím odeslání spočítá dosažený bodový zisk, uloží proběhlý pokus do databáze a zobrazí výsledek studentovi spolu s dalším dialogem
- Systém zobrazuje výsledek předchozího pokusu. Pokud studentovi zbývá ještě nějaký pokus, je mu nabídnuto opakování otázky (za cenu penalizace). Dále student může postoupit k další otázce. V tom případě se aktualizuje celkový dosažený počet bodů v testu a systém nabízí další otázku. Pokud jsou všechny otázky vyčerpány přechází se na zobrazení historie průběhu testu spolu s dosaženými výsledky

Samozřejmě student nesmí v průběhu testu zavřít prohlížeč ani mazat nebo zakazovat předtím povolené cookies. Jinak dojde ke ztrátě dat v session a tím pádem k odhlášení, tudíž i k ukončení vyplňování testu a vzhledem k existenci záznamu o výsledku testu v databázi, který je vytvořen ihned po spuštění testu, již není možné absolvovat test znovu. Na tuto skutečnost jsou studenti upozorněni.

7 Zhodnocení navrženého systému

V této kapitole shrneme možnosti navrženého systému v porovnání s existujícími komerčními i volně dostupnými produkty. Srovnávané systémy poskytují celkovou podporu pro elektronické vzdělávání. Testovací moduly tak tvoří pouze jejich část. Rovněž tyto systémy fungují v kontextu použití jediného systému pro celou školu. Naproti tomu navržený systém uvažuje kontext jednoho předmětu/kurzu. Toto ale nemá vliv na aplikační část týkající se samotné manipulace s testy a testování.

Navržené databázové schéma je svou strukturou schopné uchovat kompletní sadu potřebných záznamů. Lze samozřejmě jednotlivé entity obohatit o další vlastnosti, které by potencionálně bylo třeba uchovávat. Toto ale nemá vliv na aplikační logiku, která tímto změn nedozná, pokud již, tak pouze kosmetických. V případě použití jiných databázových serverů než MySQL umožňujících pokročilejší operace s daty na úrovni databázového stroje (např. lepší možnosti triggerů – spouštění triggerů i z událostí vyvolaných podmínkami CASCADE) by bylo možné přesunout část databázových operací z PHP skriptů přímo do databáze. Tímto sice získáme jistě vyšší výkon, nicméně cenou bude závislost na daném databázovém software.

Implementovaná administrace umožňuje sice jednoduchou, ale plně funkční a dostačující správu uživatelů a testů. Správa otázek byla omezena na typy **multiple choice**, **multiple answer**, **true/false** – čili všechny typy klasických testových otázek. Pokud by bylo třeba rozšířit tento modul, znamenalo by to přidat třídy pro tvorbu a zpracování editačního formuláře odpovědí pro každý další typ otázky. Dále potom je třeba přidat třídy pro tvorbu formuláře při vyplňování otázky a hodnotící logiku. Návrh databáze obsahuje vše potřebné.

Dalším problémem je komunikace systému s okolím. Je zahrnuta možnost exportu databáze do XML. Jelikož se výrobci neshodují v používaných formátech, jediné řešení je vytvořit podporu pro co nejširší paletu používaných formátů. To lze vyřešit dodatečným vytvořením speciální XSL šablony pro každý formát, která bude převádět exportované XML na žádaný styl. Pro opačný směr bude třeba složitější parser, který dokáže sofistikovaně skrýt rozdíly mezi jednotlivými formáty a sjednotit je do potřebné databáze.

8 Závěr

Spektrum existujících e-learningových projektů je velmi rozmanité. Proto je obtížné hledat společné rysy, které by umožnily implementaci systému poskytujícího všechny možnosti. Navíc je zde klasický problém komerčních produktů, totiž nesnadný, ne-li nemožný přístup k přesnějším specifikacím a tudíž obtíže při návrhu systému schopného komunikace s ostatními prostřednictvím výměny dat. Můžeme si jen přát aby dál vznikaly univerzální a všemi dodržované standardy pro výměnu dat, které by vývojářům velice usnadnily práci (například IMS QTI).

Tato práce shrnuje základní poznatky o tvorbě a používání testů. Mnoho vlastností je společných jak testům čistě papírovým tak elektronickým (ať jsou pouze elektronicky vytvářeny nebo i vypracovávány). Samozřejmě elektronická podoba má svoje mnohá specifika a problémy, které je třeba vyřešit, než bude možné takovýto systém testování zavést hromadně v praxi (už jen otázka autentizace a podvodů by jistě vydala na samostatnou práci).

Podářilo se vytvořit zajímavé univerzální řešení databázové vrstvy. Systém je tak snadno rozšiřitelný. Díky použití XML bude i potencionální úprava pro jiná zobrazovací zařízení jednoduchou úpravou šablony

Mě osobně práce přinesla mnoho poznatků v oblasti, která má dobré předpoklady k růstu a vývoji, a jistě vedla ke zlepšení mých programátorských dovedností.

Literatura

- [1] Internetové stránky UCI Distance Learning Center, dostupné z adresy:
<http://learn.uci.edu/cms/help.php?module=quiz&file=import.html> - formáty pro import a export testových otázek (březen 2007).
- [2] Internetové stránky: systém Moodle, dostupné z adresy: <http://www.moodle.org> - inspirace, základní principy e-learningových informačních systémů (březen 2007)
- [3] Internetové stránky: IMS Global Learning Consortium, dostupné z adresy:
<http://www.imsglobal.org/question/> - IMS QTI, standardy pro přenos testových otázek (březen 2007)
- [4] Internetové stránky: Survey Konsole, dostupné z adresy: <http://www.surveyconsole.com/sample/> - příklady otázek (březen 2007)
- [5] Internetové stránky: systém WebCT dostupné z adresy: <http://www.webct.com> (březen 2007)
- [6] Internetové stránky: systém Blackboard, dostupné z adresy: <http://www.blackboard.com> (březen 2007)
- [7] Internetové stránky: UCLA Office of Instructional Development, dostupné z adresy:
<http://www.oid.ucla.edu/units/tec/tecresources/> - srovnání testovacích systémů (březen 2007)
- [8] Gutmans, Andi; Bakken, Stig; Rethans, Derick: Mistrovství v PHP5. Brno. CP Books, a.s., 2005, 655 stran. ISBN 80-251-0799-X

Příloha A – Instalace systému

Instalace předpokládá přístupný prostor s příslušnými právy na www serveru a interpretem PHP 5.1.4 a vyšším. Dále vytvořenou databázi typu MySQL 5.0.22 a vyšším. Klienti musí disponovat webovým prohlížečem. Systém je optimalizován pro Mozilla Firefox 2 a Internet Explorer 6.

Zdrojové kódy zkopírujte do příslušného adresáře na serveru a do prohlížeče zadejte adresu projektu. Spustí se instalace, která bude vyžadovat zadání parametrů připojení k databázovému serveru. Po jejich správném zadání se vytvoří potřebné tabulky a systém je připraven k používání.

Počáteční jméno a heslo je admin / admin. Toto doporučujeme okamžitě po přihlášení změnit z bezpečnostních důvodů.

Jako manuál lze použít kapitolu 6.3 této zprávy.