

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
PŘÍSLUŠNÝ ÚSTAV

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF

APLIKACE PRO GENEROVÁNÍ TESTOVÝCH OTÁZEK

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

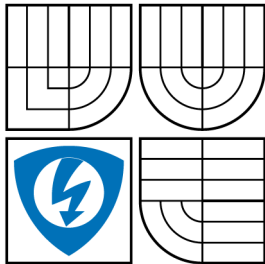
AUTOR PRÁCE
AUTHOR

PETR ŽOLDOŠ

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
PŘÍSLUŠNÝ ÚSTAV

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF

APLIKACE PRO GENEROVÁNÍ TESTOVÝCH OTÁZEK APPLICATION DESIGN FOR TEST QUESTION GENERATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

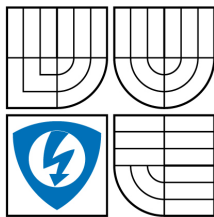
AUTOR PRÁCE
AUTHOR

PETR ŽOLDOŠ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PETRA LAMBERTOVÁ

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Petr Žoldoš
Ročník: 3

ID: 77727
Akademický rok: 2008/2009

NÁZEV TÉMATU:

Aplikace pro generování testových otázek

POKYNY PRO VYPRACOVÁNÍ:

Navrhnete aplikaci v jazyce PHP, pomocí kterého bude možno z dané databáze vygenerovat testové otázky. Aplikace by měla umožňovat členění otázek do různých skupin, přidělovat každé otázce bodové hodnocení, zadat více správných odpovědí k jedné otázce. Uživatel by měl mít možnost exportovat vybrané otázky do formátu XML. Aplikace by měla být zabezpečena proti zneužití.

DOPORUČENÁ LITERATURA:

- [1] WELLING, Luke, THOMSON, Laura. PHP a MySQL - rozvoj webových aplikací. Praha : SoftPress, 2002. 718 s. ISBN 80-86497-20-8.
[2] ULLMAN, Larry. PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek. 1. vyd. Brno : Computer Press, 2004. 534 s. ISBN 80-251-0063-4.

Termín zadání: 9.2.2009

Termín odevzdání: 2.6.2009

Vedoucí práce: Ing. Petra Lambertová

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práve třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

ABSTRAKT

Podstatou bakalárskej práce je navrhnuť a realizovať zabezpečený databázový systém. Jeho úlohou je správa testových otázok v užívateľsky prístupnej aplikácii. Je schopná otázky vkladať a vymazávať, editovať, zadeľovať do skupín, exportovať do externých súborov typu XML.

V teoretickej časti práce sú popísané vlastnosti použitých programovacích jazykov. Druhá polovica práce sa venuje postupu programovania databázového systému a popisu akým spôsobom funguje.

KĹÚČOVÉ SLOVÁ

databáza, testová otázka, Apache, PHP, MySQL, XML, Moodle, DOM

ABSTRACT

The purpose of the Bachelor work is to design and implement secure database management system. It's task is management of the test questions in an user-friendly application. It is able to insert and erase questions, edit them, sort them by groups as well as export them to external XML files.

The theoretical part is dealing with characteristics of used programming languages. Second part is about building the database system and describing the way it works.

KEYWORDS

database, test question, Apache, PHP, MySQL, XML, Moodle, DOM

ŽOLDOŠ P. *Aplikace pro generování testových otázek*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 42 s. Vedoucí bakalářské práce Ing. Petra Lambertová.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „APLIKACE PRO GENEROVÁNÍ TESTOVÝCH OTÁZEK“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

OBSAH

Úvod	9
1 Návrh aplikácie	10
1.1 Jazyk PHP	10
1.1.1 História jazyka PHP	10
1.1.2 Technické fakty o PHP	11
1.1.3 Výhody jazyka PHP	12
1.2 Databáza	13
1.2.1 Delenie databáz	14
1.3 MySQL	15
1.3.1 História MySQL	15
1.3.2 Technické fakty o MySQL	16
1.3.3 Úložiská dát v MySQL	17
1.3.4 Výhody MySQL	17
1.4 XML	18
1.4.1 Vlastnosti XML	18
1.4.2 Formát Moodle	19
1.5 Spolupráca PHP s MySQL a XML	20
1.6 PHP a MySQL	21
1.7 PHP a XML	21
1.7.1 Metóda SAX	21
1.7.2 Metóda DOM	22
1.7.3 Metóda SimpleXML	22
2 Realizácia aplikácie	23
2.1 Server a inštalácia	23
2.2 Adresárová štruktúra	24
2.3 Orientácia vo webovej aplikácii	25
2.3.1 Pripojenie sa k databáze	25
2.3.2 Registrácia	25
2.3.3 Prihlásenie	27
2.3.4 Zoznam testových otázok	27
2.3.5 Nová otázka	29
2.3.6 Editácia otázok	30
2.3.7 Skupiny	31
2.3.8 Dowload	31
2.3.9 Odhlásenie	32

2.4	Štruktúra databázy MySQL	32
2.5	Implementácia	35
3	Záver	36
	Literatura	37
	Seznam symbolů, veličin a zkratk	38
	Seznam příloh	39
A	Prvá příloha	40
A.1	Príklad XML štruktúry vo formáte Moodle	40
B	Druhá príloha	42
B.1	Zdrojové kódy a pomocné súbory	42

SEZNAM TABULEK

2.1	Tabuľka otázok	34
2.2	Tabuľka skupín	34

ÚVOD

V modernom čase 21. storočia je uchovávanie informácií skôr v elektronickej ako v papierovej podobe už samozrejmosťou. Vo svete sa nepretržite hromadia nekonečné množstvá dát a doba kedy sa takéto dáta ukladali a zálohovali v nekonečných skladoch kartoték je už preč. Elektronická forma dokumentov prináša nové možnosti. Je neporovnateľne efektívnejšia, praktickejšia a môžeme povedať, že aj bezpečnejšia, aj keď v tomto smere má všetko svoje pre a proti. Výhod je však nespočetne veľa. K tomu aby sme mohli bezpečne uložiť potrebné informácie prehľadným spôsobom a ďalej s nimi pracovať je potrebné úložisko dát a tiež aplikácia, ktorá bude umožňovať spoluprácu s ním. Systém riadenia báze, alebo databázový systém, je softwarové vybavenie, ktoré túto nevyhnutnú spoluprácu zaisťuje. Tvorí teda rozhranie medzi aplikačnými programami a uloženými dátami.

S databázovými systémami sa stretávame prakticky na každom kroku a v každom obore. Či už je to bankovníctvo, pošta, farmaceutický priemysel alebo telekomunikácie. Výnimkou už nie je ani školstvo. Formou dynamických webových aplikácií je možné zadávať študentom testy, sťahovať študijné materiály, informovať študentov o termínoch skúšok či zápočtov a podobne.

Na nasledujúcich stránkach budú popísané funkcie, vlastnosti a správanie sa webovej aplikácie určenej na generovanie a spracovavanie testových otázok. Práca je rozdelná do viacerých častí. V prvej polovici sú detailne rozobrané technológie a programovacie jazyky, ktoré boli použité na vybudovanie tohto projektu. Sú nimi hlavne jazyky PHP, SQL a XML. Taktiež je rozobraná téma databázových systémov a ich úloha v súčasnom technickom svete informácií.

Druhá polovica sa venuje praktickej časti projektu. Popri analýze a popise samotného databázového systému budú uvedené ukážky niektorých dôležitých príkazov a popis riešenia, ktoré dotvorili efekt výslednej práce.

1 NÁVRH APLIKÁCIE

Pre budovanie tak komplexných a náročných projektov, akými databázové systémy naozaj sú, existuje množstvo variant programovacích jazykov, skriptovacích systémov a databázových serverov. V našom prípade bola použitá dvojica voľne šíriteľných technológií MySQL a PHP. Tie sa stali „chlebom a maslom“ tvorcov webových aplikácií pre ich rýchlosť a dostupnosť.

1.1 Jazyk PHP

Za posledné roky sa z PHP stala jedna z najrozšírenejších webových platforiem na svete. O jej popularite svedčí aj fakt, že je používaná na viac ako tretine webových serverov na celej planéte [1]. Tento rast však nie je len kvantitatívny ale predovšetkým kvalitatívny. Stále viac spoločností, vrátane tých najvýznamnejších, spolieha práve na jazyk PHP aj v najdôležitejších obchodných operáciách.

1.1.1 História jazyka PHP

V roku 1995 začal Rasmus Lerdorf pracovať na vývoji nástroja PHP/FI. Jeho prvá verzia bola nazvaná Personal Homepage Tools/Form Interpreter. Bola to kolekcia skriptov v jazyku Perl, ktorému bola aj veľmi podobná. Perl zabezpečoval spracovávanie odosielania dát z formulárov. Novému nástroju však v tejto dobe chýbalo mnoho základných užitočných jazykových konštrukcií.

V roku 1997 bol nástroj PHP/FI prepracovaný, v tej dobe sa však o vývoj staral iba Lerdorf. Po jeho zverejnení si ho všimli Andi Gutmans a Zeev Suraskim, ktorí pátrali po jazyku, ktorý by zjednodušil vývoj elektronického obchodovania. Následne sa rozhodli skriptovací jazyk prepracovať. Spojili sa s Lerdorfom a spoločne vytvorili novú verziu PHP 3. PHP už viac nebol určený len na osobnú potrebu. Gutmans a Suraskim urobili systém rýchlejšim a tiež navrhli a implementovali rozšírenie rozhrania API, ktoré predstavovalo jednoduchú podporu ďalších rozšírení. Tie sa dali využiť napríklad k spojeniu s databázou, ku kontrole pravopisu a k iným viac ako užitočným procesom. V roku 1998 sa odhadovalo, že bude PHP inštalované na 50 000 doménach. Nakoniec sa ukázalo, že počet domén používajúcich túto prvú verziu mnohonásobne prekonal akékoľvek očakávania. Bola nainštalovaná na viac ako jednom miliónu domén.

Pretože si dvojica vývojárov uvedomovala aký obrovský potenciál sa v tomto jazyku skrýva, pustila sa do ďalších úprav. Skriptovací stroj PHP 3 spracovával skript počas čítania. Verzia 4 už prišla s novým prístupom. Skriptovací stroj už

neprekladá skript do strojového kódu, ale do bajtového kódu, ktorý je potom spracovávaný strojom Zend Engine. Výkon jazyka sa výrazne zvýšil bez toho aby došlo k zásadnému narušeniu kompatibility s jazykom PHP 3. Zdokonalilo sa tiež rozhranie API alebo vrstva serverovej abstrakcie, vďaka ktorej je možné prevádzkovať PHP na väčšine obľúbených webových serveroch. Dôležitým medzníkom bol príchod superglobálnych premenných (napríklad `$_GET` a `$_POST`). Nová verzia 4 bola oficiálne uvoľnená 22.marca v roku 2002 a po troch rokoch ju využívalo viac ako päťnásť miliónov serverov. Krátko nato začali užívatelia požadovať obcenejši objektovo orientovaný prístup, čo viedlo k prepracovaniu objektovo orientovanej časti skriptovacieho stroja Zend. V PHP 5 zostáva základná funkcia nezmenená avšak bola značne rozšírená o ďalšie rysy. Iné funkcie boli pozmenené alebo úplne odstránené. Od najnovšej verzie sa očakáva ďalšie rozšírenie podielu jazyka PHP na poli webového vývoja. Vďaka možnosti jednoduchšej práce s XML sa jazyk radí minimálne na rovnakú úroveň s inými webovými technológiami.

Informácie ohľadne vývoja programovacieho jazyka boli čerpané z publikácie [1].

1.1.2 Technické fakty o PHP

PHP, tiež označované ako PHP: Hypertext Pre-procesor, je skriptovací jazyk fungujúci na strane servera. Je navrhnutý špeciálne na budovanie internetových aplikácií. Často je interpretovaný v spolupráci s HTML, avšak na rozdiel od obyčajného HTML kódu, nie je PHP skript posielaný užívateľovi serverom priamo. Predtým je spracovaný strojom.

Tento fakt má svoje pozitívne ale aj negatívne stránky. Pozitívnu skutočnosťou je, že sa k zdrojovým skriptom nedostane nikto iný okrem autora. V tomto sa líši napríklad od JavaScriptu. Tento je totiž sprístupnený jednoduchým zobrazením zdrojového kódu stránky, ktoré podporujú už prakticky všetky súčasné internetové prehliadače. JavaScript, ako skriptovací jazyk pracujúci na strane klienta, je schopný dynamicky reagovať na udalosti spôsobené návštevníkom www stránky. Toto PHP nedokáže z jednoduchého dôvodu. K uskutočneniu každej novej udalosti musí prehliadač v prípade PHP kódu znovu poslať požiadavku na server. Preto programátori často tieto dva jazyky kombinujú s cieľom vytvoriť živú dynamicky sa meniacu web stránku.

Funkcie, ktoré PHP ponúka, posúvajú hranice možností jednoduchých internetových stránok. Do kódu HTML stránky je vložený PHP kód, ktorý je vykonaný vždy, keď je daná stránka načítavaná. PHP kód je spracovaný na internetovom servery a následne generovaný do HTML alebo iného výstupu, ktorý návštevník vidí. To je princíp dynamického generovania www stránok.

Je to technológia s otvoreným zdrojovým kódom, tzv. „*open source*“. Každý

má k nemu teda prístup. Môže ho ktokoľvek čítať, používať, meniť podľa svojej potreby a šíriť ďalej bez hrozby trestného stíhania. V porovnaní s komerčnými „*closed source*“ produktami sa nemusíme obávať, že ho výrobca stiahne alebo prestane podporovať.

Jedna z príjemných vlastností jazyka je jeho dostupnosť pre Microsoft Windows, veľké množstvo operačných systémov pracujúcich na báze UNIX a pre ktorýkoľvek plnohodnotne pracujúci Web server.

Podporuje ODBC, Open Database Connection standard, čo znamená, že je schopné pripojiť sa ku všetkým databázam podporujúcim tento celosvetový štandard.

Najnovšia verzia jazyka, verzia 5, zahŕňa podporu práce s XML. Tento prídavok je navrhnutý tak inteligentne, že spolupráca programovacieho jazyka s externými súbormi XML už nie je nijak komplikovaná. Jednoduchosť, s akou sa v tomto smere stretávame, robí PHP najlepším nástrojom na spracovanie dokumentov XML.

Tento súhrn faktov je dostatočný na utvorenie si obrazu o vlastnostiach tejto skriptovacej technológie. Akékoľvek ďalšie informácie je možné získať na domovskej stránke PHP¹.

1.1.3 Výhody jazyka PHP

Najväčšiu konkurenciu predstavujú: Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) a Allaire Cold Fusion. V porovnaní s nimi má PHP mnohé silné stránky [5]:

Výkon — PHP je výkonné, pohotové a účelné. Je to stabilný jazyk používajúci jediný nenákladný server. Dokáže obslúžiť milióny príkazov za deň.

Integrácia databáz — PHP umožňuje priame pripojenie k MySQL, PostgreSQL, mSQL, Oracle, dbm, filePro, Hyperwave, Informix, Interbase alebo Sybase databases. Spomenuté systémy sú jedni z najznámejších avšak PHP je schopné spolupracovať aj s mnohými inými. Pretože podporuje štandard otvoreného databázového pripojenia ODBC, zvláda čerpať dáta aj z programov Microsoft Excell či Microsoft Access.

Knižnice — Aj kvôli tomu, že PHP bolo navrhnuté pre internetové účely, obsahuje celý rad zabudovaných funkcií sprostredkujúcich užitočné úlohy spojené s webom. Ako príklad uvediem generovanie GIF obrázkov, pripojenie k službám inej siete, posielanie emailových správ, práca s cookies, generovanie PDF či XML dokumentov. Všetko za cenu pár riadkov textového kódu.

¹<http://www.php.net/>

Cena — Najnovšia verzia je kedykoľvek dostupná na domovskej stránke PHP. Samozrejme bez poplatkov, či registrácie.

Práca s PHP — Začiatky práce s PHP sú relatívne jednoduché. Jedná sa o veľmi prístupný jazyk. Syntax je založená na iných programovacích jazykoch, predovšetkým C a Perl. So skúsenosťami s niektorým z nich, je prechod na PHP prakticky bezproblémový.

Univerzálnosť — Jedná sa hlavne o schopnosť tejto programovacej reči komunikovať s rôznymi operačnými platformami. PHP kód napísaný vo voľne šírenej verzii Unixového systému ako Linux, FreeBSD, bude fungovať rovnako stabilne a spoľahlivo aj na komerčnom Unixovom OS ako Solaris a IRIX či rôznych verziách balíku služieb Microsoft Windows. Jeden kód vo väčšine prípadov funguje na rozdielnych systémoch bez nutnosti jeho modifikácie.

Zdrojový kód — Ako už bolo spomenuté, prístup k zdrojovému kódu je voľný. Na rozdiel od komerčných IT produktov, modifikácia i pridávanie prvkov do samotného jazyka je povolené bez obmedzenia. To je vlastne podstata myslenia aplikácií s otvoreným zdrojovým kódom – zdokonalenie alebo prispôsobenie daného jazyka, programu, systému konkrétnemu užívateľovi.

1.2 Databáza

Podľa zdroja [7] japonský zákon o autorsko-právnej ochrane z roku 1986 definuje databázu ako súbor informácií, tj. znakov, čísiel a podobne, ktorých systematická štruktúra umožňuje, aby tieto informácie mohli byť vyhľadávané pomocou počítača.

V roku 1988 však pán Bancilhan uviedol inú definíciu, výstižnejšiu odbornejšiu. Databázu opísal implicitne týmito slovami: „Databázová technológia sa zaoberá riadením veľkého objemu perzistentných, spoľahlivých a zdieľaných dát“. Tento výklad zahŕňa slová, ktoré túto tematiku exaktne špecifikujú [7]:

Veľký objem znamená, že k ich spracovaniu nestačí len vnútorná pamäť.

Perzistenciu chápeme ako nezávislosť dát na programe, ktorý ich spracováva. Dáta pretrvávajú od jedného spracovania k nasledujúcemu.

Spoľahlivosťou sa rozumie fakt, že po chybe počítača (výpadok napájania, zrušenie systému a iné) je možné dáta opäť zrekonštruovať.

Zdieľané dáta sú dáta, ktoré sú prístupné viacerým užívateľom súčasne.

Poučiek, akými sú tieto banky dát definované, je nekonečné množstvo, avšak základný význam je rovnaký. Z môjho uhla pohľadu je databáza úložiskom dát veľkého objemu, ku ktorej obsluhu je nevyhnutná softwarová aplikácia zabezpečujúca flexibilitu, variabilitu a prehľadnosť.

S databázami sa stretávame denne. S rozvojom sietí a hlavne internetu sa stalo bežnou záležitosťou, že môžu byť prístupné aj vzdialene. Taká databáza umiestnená centrálnie poskytuje dáta mnohým, viac menej nezávislým, odberateľom. Dalo by sa priam povedať, že databázy zažívajú v súčasnom svete zlaté časy. Fakt, že ich existuje celá rada a viacero typov, je toho následkom.

1.2.1 Delenie databáz

Presné rozdelenie je s postupom času stále komplikovanejšie. Jednotlivé kritériá sa začínajú vzájomne prekrývať a mnoho databáň dokáže veľa podobných úkonov.

Skupiny kritérií, ktoré kategorizujú databázové systémy [4]:

Databázové modely — Objektové, hierarchické, sieťové, relačné a iné. Líšia sa podľa spôsobu, akým ukladajú dáta. Vo svete prevláda relačný model.

Jednoužívateľské a viac užívateľské — Podľa počtu užívateľov, ktorí sa môžu k databáze pripojiť. V komerčnej sfére sú jednoužívateľské databáze už ohrozeným druhom aj keď aj tie majú svoj význam.

Súborové a systémové — Posudzované v závislosti na ukladaní dát. V prípade, že sa jedná o ukladanie do jediného súboru, je to súborová databanka. Príkladom je Microsoft Excell. Ak je ukladanie nejakým spôsobom zakomponované do systému, je systémová, napríklad MySQL. Systémový typ tvorí drvivú väčšinu databáz na trhu.

Licencia, cena — Kód systému je buď otvorený alebo uzatvorený. Šírenie je bez obmedzenia alebo podlieha majiteľom stanoveným podmienkam. Na trhu sa vyskytujú komerčné aplikácie, ku ktorým sa dostanete až po zaplatení stanoveného poplatku a sú aj také, za ktorých používanie sa neplatí.

Platformy — Multiplatformové bázové systémy dokážu pracovať na systémoch GNU/Linux, Microsoft Windows, FreeBSD, Sun Solaris, IBM's AIX, Mac OS X, HP-UX, AIX, QNX, Novell NetWare, SCO OpenUnix, SGI IRIX, Dec OSF. Naproti tomu, jednoplatformové pracujú len na niektorom, a to jedinom z nich.

Ďalšie vlastnosti — Výkon, robustnosť, vhodnosť nasadenia, bezpečnosť. Maximálne limity sú určujúce pre počet prihlásených užívateľov, počet súčasne prebiehajúcich procesov, kapacitu a podobne.

Rozhodnúť, ktorý systém na správu dátových úložísk je najlepší, nie je možné. Je nutné zvažovať príliš veľa faktorov, pričom dokonalý systém neexistuje. Zvolenie vhodnej databáze je otázkou kompromisu medzi rýchlosťou softwaru a jeho skutočnými schopnosťami. Zúžiť výber vzhľadom na popularitu medzi užívateľmi vo svete, predstavuje jednoduchšiu úlohu. Navrhnutá aplikácia využíva databázový systém MySQL.

1.3 MySQL

Má vysoký podiel medzi v súčasnosti používanými databázami na trhu. Počet web-hostingových spoločností, ktoré ho ponúkajú pre svoje hosťované weby ako súčasť portfólia svojich služieb je obrovský. Odporcovia tvrdia, že to je jediný dôvod tak veľkej popularity. Sú toho názoru, že keby tomu tak nebolo, obľúbenosť by citeľne klesla. Zástancovia však vedia, že trh si jednoducho MySQL vybral a to dokazuje jeho kvality.

Pravda bude s najväčšou pravdepodobnosťou niekde medzi týmito dvomi názormi.

1.3.1 História MySQL

Švédska firma MySQL AB stála pri zrode tohto technologického prvku. Bola založená v roku 1995 a považuje sa za jednu z najväčších open source spoločností na Zemi. Počiatky vývoja systému siahajú až do roku 1979. Verejnosti prístupný sa stal o takmer dvadsať rokov neskôr, v roku 1996. Podľa referencií spoločnosti bolo v priebehu niekoľkých rokov vykonaných viac ako päť miliónov inštalácií MySQL a v roku 2004 bolo stiahnutých až desať miliónov produktov. 26. februára 2008 sa stala firma MySQL AB vlastníctvom Sun Microsystems a postavenie samotného jazyka MySQL sa ešte upevnilo.

Súčasný jazyk je potomkom pôvodného mSQL. To slúžilo na pripojenie k tabuľkám, pričom používalo rýchle primitívne procesy. Z dôvodu nedostatočnej flexibility a rýchlosti prišli vývojári s novým SQL. Malo za cieľ obsluhovať veľké objemy dát oveľa rýchlejšie ako iné vtedajšie riešenia. Aby bolo zahrnutie tretej strany bezproblémovjšie, rozhranie API zostalo takmer nezmenené ako v prípade technológie mSQL [8].

Nový systém správy databáz prešiel v priebehu vývoja nespočetnými úpravami, pričom sa najväčší dôraz stále kladie na pomer rýchlosti a funkčnosti.

1.3.2 Technické fakty o MySQL

Tento relačný databázový systém si získal priaznivcov po celom svete vďaka jeho rýchlosti, robustnosti, spoľahlivosti. Umožňuje ukladať, triediť, vyhľadávať a získavať dáta veľmi efektívne a jednoducho. V jednom čase s ňou môže pracovať viac užívateľov, pritom je zaručený rýchly prístup a autentifikácia pripojených subjektov. Všetko za spolupráce s MySQL serverom, ktorý kontroluje každý prístup k dátam. Ten je pre tieto schopnosti známy ako multiužívateľský a viacúčelový.

Komunikácia s MySQL prebieha za pomoci SQL. Podobne ako v prípade ostatných SQL databáz sa jedná o dialekt tohto celosvetovo rozšíreného jazyka s niektorými rozšíreniami. MySQL je multiplatformný, open-source, relačný. Teší sa masívnej podpore zo strany aplikácií či programovacích jazykov celého sveta.

Výraz relačný model databázy vyjadruje skutočnosť, že dáta sú ukladané v separovaných tabuľkách so stĺpcami a riadkami. Toto riešenie pridáva na rýchlosti aj pružnosti. Tabuľky sú spojené vopred definovanými vzťahmi, čo v prípade potreby dovoľuje kombinovať dáta z rôznych tabuliek. Skratka SQL, Structured Query Language, je teda používaná opodstatnene.

Tento systém je považovaný za úspešného priekopníka dvojitého licencovania. Je k dispozícii pod bezplatnou licenciou GPL (Všeobecná verejná licencia GNU), ako aj pod komerčne platenou licenciou.

MySQL je ²:

- napísaný v C a C++, testovaný vo viacerých druhoch kompilátorov
- testovaný na komerčnom detektore pretekania pamäte
- funkcie SQL sú implementované cez vysoko optimalizované knižnice, čo zaručuje najvyššiu možnú rýchlosť
- ponúka veľa typov stĺpcov
- bezpečnosť systému je zabezpečená a ponúka možnosť overovania užívateľov, heslá sú zabezpečené kryptovacím mechanizmom
- schopné uchovávať milióny záznamov
- klienti sa pripájajú cez TCP/IP Sokety, Unix Sokety (Unix), či Named Pipes (NT)
- plná podpora rôznych znakových sád
- server je schopný sprostredkovať chybové hlášky vo viacerých jazykoch

²Prebrané z [9]

1.3.3 Úložiská dát v MySQL

V ponuke samotného MySQL sa nachádza niekoľko typov databázových tabuliek, ktoré sa líšia svojimi možnosťami, použitím a spôsobom ukladania dát do súborov [8]:

1. MyISAM — je najpoužívanejšia, bez podpory transakcií
2. InnoDB — s podporou transakcií ³
3. BerkeleyDB (BDB)
4. MEMORY — práca s dátami v pamäti
5. NDB Cluster — úložisko pre clusterované databázy
6. ARCHIVE — komprimované tabuľky, bez podpory indexov
7. CSV — ukladanie v prostých textových súboroch

1.3.4 Výhody MySQL

Výkon — MySQL je dokázateľne rýchle ⁴. Od začiatku bolo optimalizované predovšetkým na rýchlosť a to aj za cenu niektorých zjednodušení. Konkurenčné databázové systémy tohto typu prekonáva v niektorých prípadoch dokonca niekoľkonásobne.

Cena — Rovnako ako platená verzia MySQL, je k dispozícii aj voľne šírená. V prípade platenej komerčnej verzie je cena nízka. Najnovšia verzia je dostupná priamo na stránkach MySQL ⁵.

Práca s MySQL — Inštalácia a nastavenie MySQL sú v zásade jednoduchšie ako v prípade mnoho iných podobných produktov.

Univerzálnosť — Kompatibilitosť je zaručená tak na Unixových systémoch ako aj pod balíkom Microsoft Windows.

Zdrojový kód — Podobne ako u PHP tak aj u MySQL nie je problém získať zdrojové kódy za akýmkoľvek účelom.

Čo sa konkurencie týka, najväčšiu predstavuje Oracle, PostgreSQL a Microsoft SQL [5].

³Databázová tranzakcia je skupina príkazov, ktoré prevedú databázu z jedného konzistentného stavu do druhého

⁴<http://www.mysql.com/why-mysql/benchmarks>

⁵<http://dev.mysql.com/downloads/mysql/5.1.html#downloads>

1.4 XML

Technológia XML⁶ sa v súčasnej dobe stáva stále univerzálnejším jazykom pre komunikáciu medzi rôznymi platformami. Niektorými je považovaná za novú webovú revolúciu. Občas sa dokonca používa aj ako databáza na ukladanie dokumentov. Treba ale spomenúť, že nikdy nebolo cieľom vytvoriť z XML dátové úložisko.

Pôvodný jazyk určený na publikovanie, HTML, prestal vyhovovať hlavne pre jeho zložitosť, ktorá vznikla jeho postupným rozširovaním. Jazyk XML nemá žiadne preddefinované značky. Tagy a názvy elementov si programátor volí svojvoľne, avšak musí dodržiavať presnú syntax, ktorá je prísnejšia ako v prípade HTML.

XML bol vyvinutý špeciálne na predávanie informácií medzi nehomogénnymi systémami v štandardnom formáte. Skratka XML pochádza z anglického eXtensible Markup Language, čiže rozšíriteľný značkovací jazyk.

Jeho silnou stránkou je spôsob akým pracuje. Má vlastnú terminológiu, popisujúcu štruktúru dokumentu z hľadiska vecného obsahu jednotlivých častí. Sám o sebe sa nezaobera vzhľadom dokumentu. To ako má dokument vyzeráť po vizuálnej stránke definujeme pomocou špeciálnych na to určených externých súborov. Skutočné dáta sú umiestnené v označených prvkoch dokumentu a to štruktúrovaným spôsobom. Ak chceme takto uložené dáta získať, musí sa dokument spracovať pomocou syntaktického analyzátora (parseru). Väčšinou je potrebné previesť informácie do iného formátu.

Jednu z jeho hlavných úloh predstavuje publikovanie dokumentov. Je schopný slúžiť ako komunikačný protokol pri práci so vzdialenými službami. Bohužiaľ, výpis všetkých techník použitia jazyka XML v tejto práci nie je možný pre jeho rozsiahlosť.

1.4.1 Vlastnosti XML

Táto programovacia reč je založená na jednoduchom texte a je spracovateľná v akomkoľvek textovom editore. Je kompletne zdarma, čo je rozdiel oproti firemným formátom, ku ktorým nie je k dispozícii žiadna verejná literatúra, navyše to bývajú spravidla značne zložitejšie a systémovo náročnejšie formáty.

Tvorcovia hneď od počiatku mysleli na potreby iných jazykov, ako je angličtina. Ako znaková sada sa implicitne používa Unicode (ISO 10646), preto je možné vytvárať texty v mnohých jazykoch súčasne a odpadáva nutnosť konvertovať text z jedného kódovania do iného. Pomocou značiek, tzv. tagov, vyznačujeme význam jednotlivých častí textu. Dokument je tak prehľadnejší a obsahuje viac informácií, ako keby sa značkovanie používalo k definovaniu písma, odsadenia a podobne. XML súbory sú informačne bohatšie, neobsahujú nepotrebné nadbytočnosti.

⁶<http://www.w3.org/XML>

Pri používaní XML dokumentu potrebujeme samozrejme dokument aj zobraziť. XML samo o sebe žiadne prostriedky pre definíciu vzhľadu neponúka. Existuje niekoľko štýlových jazykov, ktoré definujú, ako sa majú jednotlivé elementy zobraziť. S veľkou pravdepodobnosťou je najznámejším CSS — kaskádové štýly [3]. Hoci som ich taktiež v práci použil, nebudem sa s nimi bližšie zaoberať, pretože nie sú podstatou tejto práce.

Jednoduchosť celkovej štruktúry má ale svoju daň a tou je, že XML nedisponuje žiadnym mechanizmom pre kontrolu chýb pri vytváraní dokumentu. K tomu je potrebné DTD a parser. DTD (Document Type Definition) je definičný jazyk, kde nadefinujeme jednotlivé tagy, a parser je program, ktorý vykoná kontrolu.

DTD nie je jediným definičným jazykom pre texty písané v XML. Neobsahujú možnosť kontrolovať typy dát. To je vlastnosť, ktorá citeľne chýba pri spracovaní dát do databázového charakteru. V súčasnej dobe sa pracuje na vytvorení jednotného štandardu, ktorý tieto kontroly umožní. Pre rôzne štandardné aplikácie, boli postupne vytvorené schémy pre konkrétne typy dokumentov. Príkladom je DocBook, ktorý definuje štruktúry pre vytváranie kníh, článkov, vedeckých publikácií. Inou ukážkou takej schémy je formát pre Moodle.

1.4.2 Formát Moodle

Moodle⁷ je e-learningový softwarový balíček určený na tvorbu výukových systémov a elektronických kurzov na internete. Je poskytovaný zdarma ako open source aplikácia patriaca pod obecnú verejnú licenciu GNU. Bežne dodávané moduly dokážu do online kurzov vkladať napríklad:

- študijné materiály vo forme HTML stránok, súborov k stiahnutiu, Flash animácií, štruktúrovaných prednášok,...
- diskusné fóra
- úlohy pre účastníkov kurzu
- automaticky vyhodnocované testy zložené z rôznych typov testových úloh
- ankety
- a iné

Práve Moodle formát testových otázok je použitý v aplikácii tejto práce. Aby bolo súbor, do ktorého sú testové otázky exportované, možné použiť v iných aplikáciách slúžiacich na výukové účely, tj. aby bola zaistená kompatibilita, mal by

⁷http://docs.moodle.org/en/Moodle_XML

jeho textový obsah dodržiavať určitý všeobecne uznávaný štandard. Keďže má byť kompatibilný so školským výukovým prostredím E-learning, podporuje práve model Moodle XML.

1.5 Spolupráca PHP s MySQL a XML

V bakalárskej práci tieto tri technológie doposiaľ neboli vzájomne konfrontované zámerne. Bol uvedený detailný popis avšak vždy pre každú zvlášť. Nasledujúce riadky sú venované práve ich vzájomnej spolupráci.

Internetové komerčné stránky sú v dnešnom svete tak rozšírené, že ich nie je možné prehliadnuť. Nezainteresovaný užívateľ si všimne rozdiel vo vizuálnej stránke, ktorá je vidieť na prvý pohľad. Taktiež určite pocíti zmenu v celkovej organizácii prezeraných stránok. Programátor však musí myslieť na nespočetne viac iných vecí, dôležitých vecí. Musí ísť pod povrch. Do úvahy berie zabezpečenie aplikácie, funkčnosť, flexibilitu, jednoduchosť užívateľského rozhrania, spoľahlivosť. Musí brať ohľad na požiadavky klienta, pre ktorého takýto projekt navrhuje. Klienti si málokedy uvedomujú, čo všetko sa skrýva za zrealizovaním aplikácie, ktorú si objednali. Málokedy si uvedomujú, aké zložité a komplexné riešenia musia byť použité. Za zmenou funkcie alebo za pridaním novej, neraz stoja hodiny práce programátora strávené za klávesnicou počítača. Je to tak aj napriek tomu, že sa výsledný efekt môže javiť ako banálna maličkosť.

Stránky vyžadujúce implementáciu databázového systému sú toho názorným príkladom. V zásade platí, čím viac funkcií a možností aplikácia ponúka, tým je zdrojový kód širší, komplikovanejší, náchylnejší k chybám, útokom a podobne. Databázové systémy tiež vyžadujú stabilitu aplikácie. Je neprípustné, aby sa uprostred práce s dátami aplikácia zasekla, zamrzla či spadla a to z dôvodu akéhokoľvek.

Organizácie vkladajú neskutočnú dôveru do nástrojov, ktoré disponujú možnosťou vzdialeného prístupu, cez internet. Tým vkladajú dôveru aj do samotných technikov, vďaka ktorým takéto nástroje existujú. Je to až desivá predstava, keď sa zamyslíme nad rizikami. Tými sú útoky z vonka. Útočník, ktorý pozná ako daná technológia pracuje, predstavuje nezanedbateľné nebezpečenstvo. Aby sa predišlo úniku dát či iným hrozbám, sú do zabezpečenia vkladané nemalé finančné prostriedky. Rizikom je taktiež nespoľahlivosť softwarového ako aj hardwarového vybavenia.

Ako príklad skutočne kritického databázového systému predstavuje bankový sektor a jeho internetové bankovníctvo, ktoré už patrí k bežnému modernému životu ľudí. Tu musí byť chyba na sto percent vylúčená. Jedná sa tu nie len o peniaze bankovej inštitúcie ale aj o peniaze samotných klientov a následky zneužitia alebo zlyhania systému sú v každom prípade katastrofálne.

Toto všetko a viac musia mať tvorcovia na zreteli. Tlak na nich, čo sa zodpovednosti, koncentrovanosti a spoľahlivosti týka, je vskutku opodstatnený.

Dôležitým bodom je vybrať programovacie nástroje, ktoré budú schopné realizovať všetky požiadavky pri budovaní aplikácie. Spolupráca programovacieho jazyka PHP s relačným databázovým systémom MySQL je mnohými oceňovaná a zároveň neoceniteľná súčasť programovacieho sveta dynamických internetových stránok.

1.6 PHP a MySQL

Ich kombináciou vznikajú nekonečné možnosti. Oba sú open-source a voľne šírené. Pre začiatočníka nie je problémom naučiť sa základy a začať pracovať prakticky okamžite. Pre ľudí so skúsenosťami z iných jazykov je prechod taktiež bezproblémový. PHP dokáže pracovať aj s inými databázami (Oracle, Postgree, MS Access). Ale databázu MySQL poskytuje aj mnoho freewebov, zatiaľ čo ostatné zatiaľ asi žiadny. Bezproblémová dostupnosť, jednoduchá inštalácia a nenáročnosť sú základnými kameňmi popularity PHP a MySQL.

1.7 PHP a XML

V PHP 4 chýbalo mnoho funkcií v orientovo objektovom programovaní, našli sa nedostatky v práci s MySQL aj XML. Nová piata verzia prichádza s rozšírením, ktoré spomenuté problémy opravuje, v prípade zlepšenia práce s XML, viac ako uspokojivo. PHP pracuje s XML dokumentom ako s celkom a efektívne spracováva dáta. Dokáže čítať a modifikovať dokumenty prehľadne a nekomplikovane. K dátam pristupuje ako do poľa alebo k objektom. To je presne to, čo spĺňa požiadavky môjho projektu. Schopnosť dostať dáta z databázových tabuliek do XML dokumentu.

Ku spracovaniu dokumentov XML používa PHP dve tradičné techniky: SAX (Simple API for XML) a DOM (Document Object Model) [1].

1.7.1 Metóda SAX

Analyzátor prechádza dokument a pre každú začiatočnú a koncovú značku, prípadne pre každý iný prvok nájdený v dokumente, generuje príslušnú udalosť. Prvým krokom je vytvorenie objektu analyzátoru (parseru):

```
$analyzátor = xml_parser_create ('UTF-8');
```

Druhým definovanie obslužných rutín udalostí:

```
xml_set_element_handler($analyzátor, 'start_handler', 'end_element');  
xml_set_character_data_handler($analyzátor, 'charakter_handler');
```

Uvedené príkazy deklarujú obslužné rutiny a to `start_handler`, `end_handler` a `charakter_handler`. Tie ešte musíme definovať, aby skript vedel, čo má vykonať ak narazí na počiatočnú značku (`start_handler`), koncovú značku (`end_handler`) a čo sa má zobrazíť vo výstupe (`charakter_handler`). Spracovanie súboru XML:

```
xml_parse ( $analyzátor, file_get_contents ( 'file.xml' ) );
```

Konečný výstup je v prípade tejto metódy rozhádzaný. Obsahuje veľa prázdneho miesta, pretože rutina pre ošetrovanie textu je volaná pre každý znak. To bol kameň úrazu pri rozhodovaní, ktorú techniku zvolíť.

1.7.2 Metóda DOM

Ak je použité rozhranie DOM, je najprv celý dokument prevedený na stromovú štruktúru následne uloženú v pamäti. Tá je potom prechádzaná v PHP pomocou dostupných funkcií ako celok. Má to za následok zvýšenie spotreby pamäte, ale na druhej strane, nie je potrebné volať žiadne metódy, pretože len čítame atribúty jednotlivých prvkov. Načítanie koreňového prvku dokumentu DOM:

```
$dom = new DomDocument ();  
$dom-> load ( 'file.xml' );  
$koreň = $dom-> documentElement;
```

Zobrazenie textového obsahu:

```
if ( $prvok-> nodeType == XML_TEXT_NODE )  
echo trim ( $prvok-> nodeValue )
```

Konečný výstup je prehľadný, pretože si prakticky tvoríme vlastnú stromovú štruktúru.

1.7.3 Metóda SimpleXML

Nové rozšírenie v piatej verzii jazyka PHP.

V tejto práci bolo vybrané rozhranie DOM, ktoré vyhovovalo nárokom a predstavám. SimpleXML je totiž síce najjednoduchší ale na modifikáciu a načítavanie. V aplikácií bola potreba XML dokument vytvoriť. A na tento účel je najvhodnejším riešením použitie techniky DOM. Rozhranie SAX bolo zavrhnuté z už spomenutého dôvodu.

Z laického hľadiska by sa dalo povedať, že PHP predstavuje akýsi virtuálny most na prepravu dát od užívateľa k databáze a od nej do XML dokumentu.

2 REALIZÁCIA APLIKÁCIE

2.1 Server a inštalácia

K tomu, aby mohol programátor začať dlhú cestu k naprogramovaniu databázového systému, musí vybrať hardware pre Web server, bez ktorého sa PHP, ako jazyk pracujúci na strane servera, nezaobíde. V prípade www stránok už zverejnených na internete tento hardware sprostredkovaný hostingovou spoločnosťou. Pre súkromné použitie a pre vývoj aplikácie na osobnom počítači je vhodné siahnuť po servery, ktorý pracuje na softwarovej úrovni.

Veľmi obľúbenou a často nasadzovanou kombináciou je MySQL a PHP v spolupráci so serverom Apache. Táto kombinácia zosobňuje základný software webového serveru. Apache je najpoužívanejší http server s otvoreným zdrojovým kódom. Je ale témou tak rozsiahlou, že sa ním zaoberá nejedna odborná kniha. V tejto práci bude spomenutý len okrajovo. V našom prípade slúži len na simuláciu skutočného servera.

Samostatne ho nainštalovať a nastaviť nie je jednoduché, s úmyslom pridať MySQL a PHP sa inštalácia komplikuje ešte väčšími.

Problém sa dá riešiť jednoduchým postupom. PHP, MySQL a Apache sú dostupné v jedinom inštaláčnom balíku. Na stránkach priaznivcov produktu Apache ¹ nájdeme distribúciu XAMPP, ktorá obsahuje všetky tri prvky. Stačí stiahnuť a inštalácia už nepredstavuje problém. Na stránke sú na výber inštaláčné súbory pre Linux, Windows, Mac OS X, Solaris. Ďalej tu nájdeme iné technické informácie a tiež varovanie, že verzia nie je vhodná pre produkčné prostredie. Jej zabezpečenie nie je pre také postačujúce. Distribútor neposkytuje žiadne reklamačné záruky.

Inštaláčny balíček XAMPP nie je jediným svojho druhu. Ďalšou voľbou môže byť EasyPHP.

XAMPP sa nainštaluje do predvoleného adresára xampp. Ten by sme v priečinku nainštalovaných programov Program Files nenašli, pretože sa nainštaluje priamo na jeho úroveň v adresárovej štruktúre diskovej jednotky. Tj. priamo ako podzložka lokálneho disku C.

Po nainštalovaní distribúcie pre operačný systém Microsoft Windows XP sú všetky súčasti balíka zapnuté. Je to kvôli pohodliu. Zápornou stránkou veci je, že týmto pádom klesá bezpečnostná úroveň. Preto sa odporúča vypnúť všetky funkcie, ktoré nie sú pre prácu nevyhnutné. K zvýšeniu zabezpečenia taktiež prispejeme, keď v textovom konfiguračnom súbore zakážeme používanie globálnych premenných

¹<http://www.apachefriends.org/en/xampp.html>

tým, že hodnotu parametra `register_globals` zmeníme na `OFF`. XAMPP si dokonca nastavil vlastné automatické spustenie pri štarte operačného systému. Ak teda nechceme aby nám bežal server spolu s pripojením k databáze nepretržite, mali by sme ho odobrať zo zoznamu automaticky spúšťaných programov.

Otvorením dialógového okna kontrolného panelu aplikácie XAMPP vidíme, ktoré súčasti sú momentálne zapnuté. Je dovoľené ich zastaviť, spustiť, získať stav vybraných súčastí. Pre naše účely bude stačiť, keď status *”Running”* bude svietiť pri položke Apache a MySQL.

Čo sa týka MySQL, nastavenie databázového stroja je naozaj inštinktívne cez nástroj `phpmyadmin`.

2.2 Adresárová štruktúra

Textové súbory, ktoré obsahujú textové inštrukcie na to, ako má program fungovať, čo má robiť, ako sa má správať a ako má vyzeráť sa ukladajú do priečinka `C:\xampp\htdocs\Bc.praca`, odkiaľ sú načítavané internetovým prehliadačom cez adresu `<http://localhost/Bc.praca>`.

V rámci prehľadnosti a z dôvodu vysokého počtu textových súborov, ktorých je dovedna cca. 30, sú zdrojové kódy rozdelené ešte do ďalších priečinkov. **Bc.praca** je koreňovým adresárom. Adresár **UZIVATEL** a **Sablony** predstavujú jeho podadresáre. Okrem nich je tu ešte zložka **ADMIN**, ktorá je ale pre naše účely nedôležitá. Vytvoril som ju pre prípad, keď by bolo nutné doplniť rolu s inými právami ako má užívateľ.

V adresáry **UZIVATEL** sa nachádzajú súbory s koncovkou `.php`, ktorých vykonanie je reálne až po overení prihlasovacích údajov. Nachádza sa tu ešte podpriečinkov s názvom **xml_files**. Slúži na ukladanie súborov s koncovkou `.xml` potom, ako sú exportované z databázy. Odtiaľto sa aj dané súbory načítajú vo forme zoznamu a s možnosťou downloadu z internetovej aplikácie.

V súboroch priečinka **Sablony** sú uložené informácie o vizuálnej stránke projektu. To znamená kaskádový štýl **kaskadovy_stylcss.css**, záhlavie **zahlavie.html**, a päta stránky **zapatie.html**. **Prava.php** obsahuje zdrojový kód stránky, na ktorú bude návštevník presmerovaný v prípade, že sa bude snažiť zobrazíť obsah určitej konkrétnej stránky, pričom nie je prihlásený alebo nemá potrebné práva.

Obrázkový objekt **clbg.bmp** je univerzálnym pozadím, webovej aplikácie.

Iné súbory koreňového adresára, ktoré nie sú rozdelené do zvláštnych adresárových množín sú: **Home.php**, **login.php**, **logout.php**, **register.php**. Tieto sú určené návštevníkom pred prihlásením. Slúžia na prihlásenie a odhlásenie registrovaných užívateľov a na registráciu nových.

2.3 Orientácia vo webovej aplikácii

Pri budovaní projektu boli, ako pomocný zdroj vedomostí, použité publikácie zaoberajúce sa PHP jazykom [1], SQL jazykom [2] a online seriál PHP [4].

Po zadaní domovskej webovej adresy <http://127.0.0.1/Bc.praca/home.php>² sa nám zobrazí úvodná stránka, kde je možné v menu, umiestnenom v ľavom paneli, vybrať medzi prihlásením a registráciou. Pokiaľ užívateľ už má vytvorený účet, stačí vyplniť prihlasovacie údaje vo formulári na adrese prihlásenia. Ak na doméne ešte registrovaný nie je, musí kliknúť na odkaz *Registrácia* a vytvoriť si nový účet.

2.3.1 Pripojenie sa k databáze

Možnosť vzájomnej komunikácie medzi databázou a aplikáciou poskytuje **connect.php**. Obsahuje iba dva riadky textu:

```
mysql_connect("localhost","root","");  
mysql_select_db("Database");
```

Vďaka prvému príkazu sa pripojíme do danej databázy ako špecifický užívateľ. Tretím parametrom funkcie `mysql_connect()` je heslo. Závisí na poskytovateľovi služby, aký stupeň zabezpečenia zvolí. Druhá funkcia vyberá konkrétnu databázovú tabuľku.

Tento súbor je nutné implementovať v každom skripte, ktorý potrebuje pracovať s databázou. Konáme tak prostredníctvom funkcie `include(sablony/connect.php)`. Pri prechode na internetovú doménu, kde sú prihlasovacie údaje na pripojenie k databáze iné, stačí zmeniť údaje len raz a to v tomto súbore.

2.3.2 Registrácia

Tento úkon zabezpečuje skript **register.php**. Po jeho načítaní je zobrazený formulár s editovacími políčkami pre Login, heslo a E-mail. Formulár je ošetrený proti zadávaniu chybných údajov. To znamená, že ak užívateľ zadá napríklad do editovacieho pola pre emailovú adresu textový reťazec, ktorý nebude mať správny formát, potom ho na to skript upozorní hlásením:

*„Zadali ste nesprávnu emailovú adresu!
Späť na registráciu.“*

V tomto prípade máme dve možnosti. Buď kliknúť na odkaz vedúci späť na registráciu alebo na domovskú stránku.

²127.0.0.1 je IP adresou servera localhost

Login, čiže prihlasovacie meno by malo byť pre každého rozdielne. Kontrola unikátnosti teraz prebieha na strane databázy, kde má pole *Login* nastavenú vlastnosť UNIQUE. Ak je snaha registrovať sa s prihlasovacím menom, ktoré si už niekto registroval skôr, MySQL to zistí a údaje neuloží.

Skript registrácie ďalej sleduje, či boli vyplnené všetky polia a to jednou z podmienok príkazu `switch`:

```
case (empty ($_POST['login']) || empty ($_POST['heslo']) ||
      empty ($_POST['potvrđH']) || empty ($_POST['mail'])):
    echo ("Nevyplnili ste všetky potrebné údaje");
break;
```

Heslo musí byť zadané dvakrát, aby sme ušetrili registrujúcemu nepríjemnosti, ktoré by nastali preklepnutím. Samozrejme, že vďaka typu editačného poľa *password*, sa písaný text nezobrazuje. Kontrola hesla spočíva v porovnaní textových polí *heslo* a *potvrđH*. V prípade, že zadané údaje nie sú rovnaké, zobrazí sa hláška, ktorá na to upozorní:

„Nesprávne zadané heslo!“

Ak sú správne vyplnené všetky kolonky, informácie o novom užívateľovi sa odošlú a uložia do databázy. Heslo sa kvôli bezpečnosti predáva šifrovane cez funkciu `sha1()`. Znamená to, že ani v databáze žiadne heslo nenájdem. Uloží sa ako zašifrovaný kód zložený zo znakov a čísel.

Celý formulár registrácie, tak ako všetky formuláre v aplikácií, eliminuje nebezpečenstvo napadnutia domény prostredníctvom JavaScriptu. Ten pracuje tak, že potencionálny útočník zadá javascriptový kód do formulárového pola a ako každý iný údaj ho uloží, čím sa kód uloží ako textový reťazec na server. Problém nastáva pri jeho opätovnom načítaní. Vtedy sa nebezpečný textový reťazec načíta, ale namiesto toho, aby sa použil len ako obyčajný text, prehliadač ho preloží ako súčasť HTML kódu. Efektom, ktorý spôsobí príkaz javascriptového kódu, môže byť nevinné zobrazenie irelevantnej hlášky vo vyskočenom okne, alebo v horšom prípade infikácia vírusom, krádež informácií z databázy, či jej úplné vymazanie. Možností sú potom stovky. Našťastie, tak ako primitívny je tento postup, tak je primitívne aj riešenie. Stačí ak nikomu neumožníme taký textový reťazec vôbec zadať. Napríklad textové pole, v ktorom očakávame iba numerické hodnoty, najprv typovo skontrolujeme a až potom pošleme ďalej. Nezabúdame tiež na kontrolu dĺžky textového reťazca, formátu textového reťazca a podobne. V praxi nastáva aj situácia, kedy takáto kontrola nie je možná. Jedná sa o vkladanie komentárov, diskusné fóra. Aj tu existuje riešenie. Pri ukladaní textu odstránime všetky znaky zosobňujúce začiatky a konce javascriptových tagov. Sú nimi znaky `<` a `>` a funkciou, ktorá zabezpečí ich ignoráciu, je funkcia `strip_tags()`.

2.3.3 Prihlásenie

Textový súbor **login.php** slúži na overenie oprávnenosti užívateľa prezerat obsah stránok, ktoré nemajú byť prístupné každému. Po zadaní prihlasovacieho mena a hesla sú premenné `$_POST ['login']` a šifrované `$_POST ['heslo']` uložené do superglobálnych premenných `$_SESSION ['login']` a `$_SESSION ['heslo']`. Následne sú porovnané s hodnotami z databázovej tabuľky. Ak dotaz SQL príkazu

```
SELECT role.Rola, users.* FROM role, users
WHERE (Login ='" .$_POST['login'] ."'
AND Heslo ='" .sha1($_POST['heslo']) ."'")
AND (users.ID_rola=role.ID_rola)
```

vráti práve jeden riadok informácií z tabuľky, potom je prihlásenie úspešné a stránka je automaticky presmerovaná na domovskú stránku prihlásených **UZIVATEL/home.php**. Ak funkcia `mysql_num_rows()` vráti nulu, znamená to, že daná kombinácia mena a hesla sa v tabuľke užívateľov **users** nenachádza. Návštevník si môže vytvoriť nový účet alebo sa pokúsiť prihlásiť znova.

Po úspešnom prihlásení, o ktorom nás informuje aj text v pravom hornom rohu s odkazom na odhlásenie, sú v ponuke menu tri odkazy. Najdôležitejším z nich je odkaz *Správa otázok*. Po kliknutí naň je zobrazený celý zoznam testových otázok načítaných z databázovej tabuľky **otazky**.

2.3.4 Zoznam testových otázok

Aktuálna stránka ponúka možnosť vyhľadávania podľa znenia otázok alebo identifikačného čísla otázok. V menu sa nachádzajú odkazy, ktoré umožňujú zoradenie otázok podľa tých istých parametrov, ako v prípade vyhľadávania. Ďalším elementom sprehľadňujúcim výpis otázok je rolovací zoznam. Ten je filtrom, ktorý rozdeľuje zoznam otázok do skupín. Znenie testových úloh predstavuje linky vedúce k detailu každej z nich.

V riadkoch jednotlivých otázok je zaškrtávacie políčko, takzvaný checkbox. Ním vyberieme otázky, ktoré chceme buď vymazať alebo exportovať do externého súboru.

Export

Musíme začať vybraním otázok, ktoré potrebujeme odoslať do XML formátu. V opačnom prípade sa nám zobrazí varovanie. Kliknutím na potvrdzovacie tlačítko *Export do XML* sa spustí celý rad príkazov. Vytvorí sa strom modelu DOM, ktorý sa odošle na disk. Prvé, čo urobíme je, že zavedieme objekt DOM a jeho koreňový uzol `quiz`:

```

$dom = new DomDocument('1.0', 'UTF-8');
$quiz = $dom->createElement('quiz');
$dom-> appendChild($quiz);

```

Všetky prvky XML stromu vytvárame volaním metódy `createElement()` inštancie triedy `DomDocument` a v prípade textových uzlov metódy `createTextNode()`. Metóde je predaný názov prvku. Následne metódou `appendChild()` pripojíme k prvku `quiz` prvok `question`, k nemu prvok `name`, k nemu textový prvok `text` a tak ďalej:

```

$question= $dom->createElement('question');
$question->setAttribute("type","multichoice");
$quiz->appendChild($question);

$name= $dom->createElement ('name');
$question->appendChild ($name);

$text= $dom->createElement ('text');
$text->appendChild($dom->createTextNode($riadok['Text_otazky']));
$name->appendChild($text);

```

Riadok `$question->setAttribute("type","multichoice");` nastaví hodnotu atribútu elementu `question`.

Funkcia `$appendChild($dom->createTextNode($riadok['Text_otazky']));` zase uloží medzi tagy `text` premennú `$riadok['Text_otazky']`, ktorá bola načítaná pomocou SQL dotazu na príkaz `SELECT`.

Toto je ukážka časti kódu, vďaka ktorému doslova vyrastie strom uzlov kompletnej XML štruktúry.

Keď máme strom vybudovaný, potrebujeme ho odoslať do externého súboru. Názov súboru je v našom prípade daný premennou `$nazov`, ktorá sa skladá z prihlasovacieho mena a času, kedy bol strom vygenerovaný a samozrejme koncovky `.xml`. Takto presne vieme, kto a kedy súbor vytvoril. Zdrojové kódy, ktoré nasledujú, sú ukážkou, ako tento proces pracuje.

```

$datum = StrFTime("%d%m%y_%H-%M-%S", Time());
$nazov = $_SESSION['login']."_".$datum;

```

Súbory sa ukladajú do priečinka `xml_files` :

```

$dom->save('xml_files/' . $nazov . '.xml');

```

Obsah vzniknutého súboru pozostáva z niekoľkých značiek identických pre každú otázku. Kompletná testová štruktúra je zasadená medzi začiatočnú a koncovú značku `quiz`. Je tu tag určujúci začiatok a tag určujúci koniec vlastností otázky s parametrom, ktorý určuje, či je správnych odpovedí jedna alebo viac. Jeho vetvami sú tagy pre znenie otázky, veľkosť penalizácie v prípade zlej odpovede, hodnota `hidden` a tag `single`, ktorého hodnota je `true` alebo `false`. V strome ďalej nasledujú informácie opisujúce vlastnosti každej jednej odpovede. Ohraničuje ho párová značka `answer` a jeho parameter `fraction` vyjadruje bodovú hodnotu odpovede. Text otázky a komentár `feedback` sú dátami, ktoré sa nachádzajú medzi začiatočnou a koncovou značkou `answer`. Všetky tieto parametre sa opakujú v rámci každej testovej otázky XML modelu Moodle. Názorný príklad takej XML štruktúry možno nájsť v prílohe A.1.

Mazanie

Druhým tlačítkom formulára je tlačítko na vymazávanie otázok. Maže dáta, ako z tabuľky `otazky`, tak aj z tabuľky `moznosti`, na základe `id_otazky`. V tom nám pomáha MySQL dotaz `DELETE`:

```
mysql_query ("DELETE FROM otazky
  WHERE id_otazky='".$$_POST['hidden'].$i.'"");
mysql_query ("DELETE FROM moznosti
  WHERE id_otazky='".$$_POST['hidden'].$i.'"");,
```

pričom superglobálna premenná `$_POST['hidden'].$i.'` obsahuje číselnú hodnotu identifikačného čísla otázok, ktoré boli označené v úmysle vymazania.

Toto všetko sa deje po preložení kódu uloženého v súbore `zobraz_vsetky_otazky_SQL.php`.

`zobraz_otazky_SQL.php` poskytuje podobný scenár, ktorého adresa umiestnenia je tiež v ľavom panely stránky. Tento scenár je ešte obohatený o stránkovaný výpis otázok. Ak je teda počet otázok vyšší ako hodnota 15, potom sa zoznam rozdelí. Položky v menu sú pre obidve adresy prakticky rovnaké. Okrem doposiaľ spomenutých, sú tu odkazy na založenie novej otázky, tvorbu novej skupiny testových otázok a zobrazenie existujúcich skupín. Internetové odkazy *Všetky otázky / Stránkovaný náhľad* slúžia na pohyb medzi kompletným a stránkovaným zoznamom.

2.3.5 Nová otázka

Prvým krokom k zrodeniu novej otázky je rozhodnutie, koľko možností bude mať. Jednoduchá stránka `potvrd_pocet_moznosti.php` obsahuje 6 rádiových tlačítko a `Submit` button. Každý `Radio` button vyjadruje počet možných odpovedí od

dvoch do šesť. Výber potvrdíme, čím sa dostaneme do formulára **nova_otazka.php**. Formulár je zabezpečený proti vkladaniu chybného a nebezpečného textu. Vyplnenie položiek *Znenie otázky*, *Penalizácia*, *Možnosť 1 až X* je povinné. Nepovinné sú kolonky *body* a *komentár*. Je tu aj ponuka výberu, do akej skupiny otázok má nová otázka patriť. Ďalej sa v XML štruktúre Moodle vyskytuje parameter **hidden** avšak ten je konštantne nulový, takže nebolo potrebné ho vo formulári vyžadovať a do databáze sa prenáša prostredníctvom skrytého elementu HTML. *Penalizácia* sa tiež nemení, je štandardne nastavená na jednu desatinu a *komentár* odpovedí sa v praxi prakticky nepoužíva. Do aplikácie sú napriek tomu zahrnuté. Sú buď preddefinované alebo nepovinné, takže nijak tvorcu otázky neobťažujú. Mojmým úmyslom bolo poskytnúť užívateľovi čo najväčšie rozpätie možností.

Po potvrdení vyplnenia formulára prebehne kontrola odosielaných dát, pričom sa berie do úvahy aj súčet bodového hodnotenia, ktorý musí byť 0 alebo 100. Hodnota 0 vyjadruje, že ani jedna odpoveď nie je považovaná za správnu. Hodnota 100 znamená, že je správna odpoveď jedna alebo ich je viac. Ak nastáva situácia, že je formulár vyplnený korektne, dáta sa uložia do MySQL tabuliek **otazky** a **moznosti**. Príklad vloženia nových riadkov do tabuliek databázového úložiska je MySQL dotaz:

```
mysql_query("INSERT INTO otazky VALUES (0,
    '". $_POST['skupina'] ."', '". strip_tags($_POST['otazka']) ."',
    '". $_POST['penalty'] ."', '". $_POST['hidden'] ."', 'true')");
```

Hodnoty medzi hranatými zátvorkami premennej `$_POST` sú dáta získané z formulárových políčk a ako vidíme, tam kde je to potrebné, sú ešte ošetrené funkciou `strip_tags()`.

Nami vytvorenú otázku zobrazíme kliknutím na odkaz *viď detail otázky*. Sme presmerovaný na stránku súboru **detail_otazky_SQL.php**. Detaily konkrétnej otázky, ktorú chceme vidieť, sú z databázy načítané vďaka ID otázky. Tá je prenášaná z predchádzajúcej adresy a uložená v premennej `$_GET[id_ot]`.

2.3.6 Editácia otázok

Časť aplikácie **uprav_otazku.php** funguje prakticky rovnako, ako keby sme tvorili novú otázku. Rozdiel je v tom, že všetky polia sú už predvyplnené. Ná základe premennej `$_GET[id_ot]` sa vopred načítajú riadky tabuliek, kde figuruje práve táto hodnota, špecifická pre každú testovú otázku. Druhý a posledný rozdiel je v spôsobe ukladania dát späť do databázy. Nedeje sa to za pomoci SQL príkazu **INSERT** ale príkazu **UPDATE**:

```
UPDATE otazky SET Text_otazky='". strip_tags($_POST['question']) ."',
    ID_skupiny='". $_POST['skupina'] ."',Penalty='". $_POST['penalty'] ."',
```

```
single='true' WHERE ID_otazky='".$_GET[id_ot] ."'");
```

Spomenutý príklad teda v tabuľke otázky modifikuje všetky položky v riadku, ktorého `ID_otazky` sa rovná identifikačnému číslu upravovanej otázky. Ak sme niektorú z nich needitovali vôbec, uloží sa údaj bezo zmeny. Stránka nám ponúkne prehľad, ktoré údaje zostali nezmenené, a ktoré boli modifikované. Okrem tlačítka na potvrdenie a odoslanie formulára sú tu ešte dve predvolené tlačítka. Tlačítko na vykonanie SQL dotazu `DELETE` a tlačítko na obnovenie pôvodných hodnôt.

2.3.7 Skupiny

Zadanie bakalárskej práce vyžaduje, aby sa otázky mohli členiť do skupín a mohli byť podľa nich filtrované. Práca so skupinami ponúka približne rovnaké možnosti a flexibilitu ako v prípade práce s otázkami. To znamená, že umožňuje ich vytváranie, upravovanie, mazanie a tiež dokáže zobrazíť zoznam otázok, ktoré skupina obsahuje. Súbory zdrojových kódov zapojených do práce so skupinami otázok a stručný popis ich významu:

skupiny.php — Zoznam skupín so zaškrtačiacimi políčkami. Zaškrtnuté skupiny je možné mazať len v prípade, že neobsahujú žiadne otázky, inak nás skript varuje a zobrazí skupiny, ktoré je nutné vyprázdniť

nova_skupina.php — Formulár s jedným editovacím políčkom pre zadanie názvu novej skupiny. Ukladá sa do databázovej tabuľky **skupiny**, názov musí byť unikátny

detail_skupiny.php — Zobrazí sa zoznam otázok, ktorých `ID_skupiny` sa rovná hodnote premennej `$_GET[ID_skupiny]`

uprav_skupinu.php — Taktiež sa používa hodnota `$_GET[ID_skupiny]`. V jedinom editačnom políčku je predvolená hodnota pôvodného názvu skupiny. Problém nastane len v jedinej situácii a to ak zadáme rovnaký názov ako názov už existujúcej skupiny

2.3.8 Dowload

Ako som už spomínal, otázky generované z databázového systému MySQL nie je problém dostať do externého súboru vo formáte XML. Súbory sa automaticky ukladajú do priečinka **xml_files**. Na dolovanie týchto súborov je určený textový súbor **download.php**. Veľmi efektívne a elegantne zobrazuje celý zoznam súborov, ktoré sú uložené v **xml_files** a končia koncovkou `.xml`. Kód na rozpoznanie XML súborov stojí za uvedenie:

```

$dir= "../uzivatel/xml_files/";
if (is_dir($dir))
    if ($dh=opendir($dir)) {
        while (($file=readdir($dh))!=false) {
            $ext=end(explode('.', $file));
            if ($ext=="xml")
                echo "<a href=\"xml_files/$file\">$file</a><br>";
            }
        closedir($dh);
    }

```

Do premennej `$dir` sa uloží adresárová cesta, v podmienke `if` sa súbor otvorí a následne sa pomocou cyklu `while` načítava. Podstatou je zložená funkcia `end(explode('.', $file))`. Tá uloží do premennej `$ext` znaky nachádzajúce sa za poslednou bodkou v názve súboru. Vo väčšine prípadov znaky za poslednou bodkou symbolizujú typ súboru. Posledná podmienka `if` zisťuje, či sa znaky v premennej rovnajú 'xml'. Ak áno, názov súboru sa vypíše, ak nie, súbor je ignorovaný. Každý súbor má na stránke svoj odkaz, takže má informáciu o svojom umiestnení v adresárovej štruktúre. To predstavuje ohromnú pomoc, pretože z tohto bodu za nás prácu koná operačný systém. Teraz už len stačí na odkaz kliknúť pravým tlačítkom myši a jeho cieľ podľa potreby otvoriť alebo uložiť.

2.3.9 Odhlásenie

Posledným súborom, ktorý nebol zmiený je **logout.php**. Plní funkciu rušiteľa, čiže maže všetky uložené superglobálne premenné:

```

$_SESSION = array();
session_destroy();

```

Týmto štýlom sa užívateľ odhlasuje. Pre opätovné prihlásenie musí navštíviť adresu **login.php** a inicializovať premenné `$_SESSION` znova.

2.4 Štruktúra databázy MySQL

Dostávame sa k poslednej kapitole, v ktorej sa zaoberám miestom, na ktoré sa dáta ukladajú a odkiaľ sa dáta získavajú.

Kompletné nastavenie a všetko potrebné k ovládaniu databáze nájdeme vo výkonnom nástroji **phpmyadmin**. Dovoľuje nám vytvoriť si viacero databáz, ja sa však budem zaoberať len jednou. Tou je databáza **database**. Ako býva zvykom,

databáza obsahuje viac tabuliek, ktoré sú vzájomne previazané. Naša obsahuje päť tabuliek a ich prehľad spolu so stĺpcami, ktoré obsahujú prináša nasledujúci zoznam:

- **moznosti**
 - ID_opt (primárny kľúč)
 - ID_otazky (cudzí kľúč)
 - Text_opt (UNIQUE)
 - Fraction
 - Feedback

- **otazky**
 - ID_otazky (primárny kľúč)
 - ID_skupiny (cudzí kľúč)
 - Text_opt (UNIQUE)
 - Penalty
 - Hidden
 - Single

- **role**
 - ID_rola (primárny kľúč)
 - Rola (UNIQUE)

- **skupiny**
 - ID_skupiny (primárny kľúč)
 - Nazov_skupiny (UNIQUE)

- **users**
 - ID_login (primárny kľúč)
 - ID_rola (cudzí kľúč)
 - Login (UNIQUE)
 - Heslo
 - Mail

Tab. 2.1: Tabuľka otázok

ID_otazky	ID_skupiny	Text_otazky	Penalty	Hidden	Single
130	11	K řízení spojení...	0.1	0	true
131	11	Který z následov...	0.1	0	true

Tab. 2.2: Tabuľka skupín

ID_skupiny	Nazov_skupiny
11	Audio
12	Video

Príkladmi databázových tabuliek sú tabuľky 2.1 a 2.2. Sú typu InnoDB, tak ako všetky ostatné v tomto projekte, a vzájomne spolupracujú.

Stĺpec `ID_otazky` predstavuje identifikátor jednotlivých riadkov. Je primárnym kľúčom tabuľky **otazky**. Ako primárny kľúč by mal byť čo najkratší a jedinečný. Tieto vlastnosti sú nastavené priamo v MySQL. Pridaním novej otázky sa automaticky inkrementuje číselná hodnota, je teda vylúčené aby mali dve otázky rovnaký identifikátor.

Cudzí kľúč `ID_skupiny` v tabuľke otázok je primárnym kľúčom tabuľky **skupiny**. Vďaka nemu vzniká väzba medzi oboma tabuľkami. Je akýmsi odkazom na jeden riadok v odkazovanej tabuľke. Dalo sa to riešiť aj iným spôsobom. Mohli sme v každom riadku namiesto prvku `ID_skupiny` rovno vložiť `Nazov_skupiny` a druhá tabuľka by vôbec nemusela existovať. Tento štýl je značne nepraktický. Ak by sme chceli názov skupiny modifikovať, museli by sme tak spraviť pre každý jeden riadok. V našom prípade stačí, keď názov zmeníme len jedenkrát v tabuľke **skupín**.

`Text_otazky`, rovnako ako `Nazov_skupiny`, je unikátny – vlastnosť `UNIQUE`. Nie je teda možné, aby sa v tabuľke vyskytli dve otázky s identickým znením. `Penalty` je číselnou hodnotou typu `DECIMAL`(desatinné číslo) s jedným desatinným číslom – dĺžka 10,1. `Hidden` je celé číslo `INT`, ktorého dĺžka nie je väčšia ako 11. A nakoniec stĺpec `Single` je textovým reťazcom `VARCHAR`. Keďže jeho hodnota bude vždy len **TRUE** alebo **FALSE**, jeho maximálna dĺžka je 5.

Podobný vzťah, ako je medzi tabuľkou **otazky** a tabuľkou **skupiny**, majú aj tabuľky **users** a **role** a **možnosti** a **otazky**. Každá tabuľka má svoj primárny kľúč, ktorý môže slúžiť ako cudzí kľúč tabuľke inej. V tabuľke **možnosti** sa nachádza pre viac odpovedí jeden cudzí kľúč z tabuľky **otazky**. To vyjadruje skutočnosť, že otázky majú viac odpovedí. Tie majú zase ďalšie určujúce prvky. Je ním unikátny textový reťazec `Text_opt` s dĺžkou maximálne 255 znakov, číselná hodnota `Fraction`, ktorej

hodnota je celým číslom od 0 do 100, takže počet znakov nebude nikdy väčší ako 3, a posledný je stĺpec **Feedback** pre komentár odpovede – typ **VARCHAR** a maximálny počet znakov 255. V oddelení **users** je zoznam registrovaných užívateľov a ich emailová adresa, zašifrované heslo k prihláseniu do systému a login. K identifikácii tabuľky slúži primárny kľúč **ID_login** a ako odkaz na tabuľku **role**, vyjadrujúcu aké práva užívateľ má, slúži prvok **ID_rola**.

2.5 Implementácia

Po napísaní posledného riadku zdrojového kódu aplikácie a jej praktickom testovaní na servere osobného počítača bol program umiestnený na internetovú freehostingovú doménu. Domovská stránka aplikácie je <<http://www.database.kx.cz/home.php>>. Po uložení súborov na server cez ftp spojenie bolo nutné ešte importovať tabuľky databázy. Na to aby bol program kompatibilný s databázou, musel byť pozmenený zdrojový kód súboru **connection.php**. Prihlasovacie meno a heslo, ktoré boli stanovené poskytovateľom domény, sa totiž líšilo od údajov, ktoré boli nastavené na doméne localhost. Doladenie vlastností prvkov v tabuľkách bolo tiež potrebné, pretože sa nepreniesli z importovaných tabuliek databázy.

Za účelom poskytnutia možnosti skúšania aplikácie bez nutnosti registrácie bol vytvorený účet. Prihlasovacie meno testovacieho užívateľa je *login* a heslo *sha1heslo*. Po zadaní týchto prihlasovacích údajov je možné sa pohybovať v aplikácii ako bežný užívateľ.

3 ZÁVER

Bakalárska práca „APLIKÁCIA PRE GENEROVANIE TESTOVÝCH OTÁZOK“ je názornou ukážkou ako databázové systémy ovládané vzdialene pracujú v praxi. K vybudovaniu systému bolo potrebné vykonať všetky úkony od základnej inštalácie servera, navrhnutia databázových tabuliek a relácií, cez navrhnutie aplikácie a jej naprogramovanie, až po jej samotnú implementáciu. Všetky konkrétne kroky boli napokon popísané a vysvetlené. Spolu s popisom použitých technológií a príkladmi dôležitých príkazov a riešení bolo cieľom vystihnúť zásady databázových systémov ako takých.

Technológie, ktoré dopomohli vytvoriť konečný efekt, sú jedni z mnohých. Dôvody, kvôli ktorým boli použité práve tieto, sú taktiež detailne opísané.

Aplikácia pre generovanie testových otázok môže byť použitá ako užitočná pomôcka pre zamestnancov ktorejkoľvek fakulty. Vyučujúci už viac manuálne neprepisujú testové otázky z papiera na papier. Je to neefektívne, stráca sa totiž flexibilita upravovania otázok ako aj prehľadnosť. Nástroj pre prácu s testovými otázkami, tj. hlavne ukladanie, editácia, generovanie, mazanie, vyhľadávanie či zadeľovanie otázok do skupín, je v porovnaní s metódou ”pero/papier” naozaj neoceniteľnou praktickou pomôckou.

Počas práce na praktickej časti sa vyskytli určité komplikácie. Problém sa vyskytol po umiestnení aplikácie na internet. Použil som freehostingovú doménu Internet Centrum. Po praktickej skúške aplikácie sa ukázalo, že server neposkytuje dostatok práv k ukladaniu exportovaných otázok do nového XML súboru. Cieľom je uložiť súbor s exportovanými otázkami na server, odkiaľ bude aj načítavaný a sťahovaný. S varovaním „*permission denied*“ to však nebolo možné. Všetky ostatné časti projektu pracovali spoľahlivo a správali sa podľa očakávaní. Problém s vytváraním súborov je s najväčšou pravdepodobnosťou možné odstrániť zmenou hostingového servera, ktorý ponúka viac oprávnení.

LITERATURA

- [1] GUTMANS, A.; BAKKEN, S.; RETHANS, D. *Mistroství v PHP 5*. 1. vyd. Brno: Computer Press a.s., 2007. 655s. 013147419X.
- [2] ŠIMŮNEK, M. *SQL kompletní kapesní průvodce*. 1. vyd. Praha: GRADA Publishing, 1999. 248s. ISBN 80-7169-692-7.
- [3] STANÍČEK, P.; LEŠETICKÝ, M.;BIEN, J.; POLAKOVIČ, J. *CSS hotová řešení*. 1. vyd. Brno: Computer Press a.s., 2006. 267s. ISBN 80-251-1031-1.
- [4] ZAJÍC, P. *Linuxsoft.cz*[seriál online]. 27.05.2004, poslední revízia 31.01.2005 [cit.2009-04-27]. <<http://www.linuxsoft.cz/php/>>.
- [5] WELLING, Luke, THOMSON, Laura. *PHP and MySQL Web Development*. 1. vyd. Indianapolis, Indiana (USA) : Sams Publishing, 2001. 893 s. 0-672-31784-2.
- [6] ULLMAN, Larry. *PHP a MySQL — Názorný průvodce tvorbou dynamických WWW stránek*. 1. vyd. Brno: Computer Press, 2004. 534 s. ISBN 80-251-0063-4.
- [7] POKORNÝ, J. *Techniky zpracování dat v databázích*. Zápisy z přednášky zpracoval Jan Šerák. 24.04.1995.
- [8] *MySQL – Wikipedie, otevřená encyklopedie* [online]. 15.05.2009 [cit.2009-05-03]. Dostupné z: <<http://cs.wikipedia.org/wiki/MySQL>>.
- [9] *MySQL Reference Manual*. Finland: TcX AB, Detron HB, MySQL Finland AB, 2001. 716 s.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

PHP PHP: Hypertext Preprocessor

XML eXtensible Markup Language – Rozšířitelný značkový jazyk

SAX Simple API for XML – Jednoduché API rozhranie pre XML

DOM Document Object Model – Objektovo orientovaný dokument

DTD Document Type Definition – Definičný jazyk pre XML

CSS Cascading Style Sheets – Kaskádové štýly

SQL Structured Query Language – Štruktúrovaný dotazovací jazyk

FTP File Transfer Protocol – Protokol prenosu súborov

ODBC Open Database Connection Standart

SEZNAM PŘÍLOH

A Prvá příloha	40
A.1 Příklad XML struktúry vo formáte Moodle	40
B Druhá příloha	42
B.1 Zdrojové kódy a pomocné súbory	42

A PRVÁ PRÍLOHA

A.1 Príklad XML štruktúry vo formáte Moodle

```
<?xml version="1.0"?>
<quiz>

  <question type="multichoice">
    <name><text>K řízení spojení neslouží:</text>
    </name>
    <questiontext format="html">
      <text>
        &lt;p class="MsoNormal"&gt;K řízení
          spojení neslouží:&lt;/p&gt;
      </text>
    </questiontext>
    <image></image>
    <penalty>0.1</penalty>
    <hidden>0</hidden>
    <single>true</single>
    <answer fraction="0">
      <text>
        H.450
      </text>
      <feedback>
        <text></text>
      </feedback>
    </answer>
    <answer fraction="100">
      <text>
        H.263
      </text>
      <feedback>
        <text></text>
      </feedback>
    </answer>
    <answer fraction="0">
      <text>
```

```
    H.245
  </text>
  <feedback>
    <text></text>
  </feedback>
</answer>
<answer fraction="0">
  <text>
    H.235
  </text>
  <feedback>
    <text></text>
  </feedback>
</answer>
</question>

</quiz>
```

B DRUHÁ PRÍLOHA

B.1 Zdrojové kódy a pomocné súbory

Všetky zdrojové kódy potrebné k prevádzke aplikácie sa nachádzajú na priloženom kompaktnom disku. V priečinku **Bc.praca** sú tri ďalšie adresáre a päť textových súborov. Jedným z nich je súbor obsahujúci exportované dátové úložisko. Je ním súbor **exportDB.txt**. Zvyšok slúži na chod časti aplikácie, kedy užívatelia nemusia byť prihlásení.

Adresár **Sablony** obsahuje pomocné zdrojové kódy. Sú používané za účelom dotvorenia jednotného vizuálneho efektu, pripojenia ku konkrétnej databáze alebo zobrazenia varovania o nedostatku práv.

UZIVATEL zahŕňa okrem zdrojových kódov slúžiacich na ovládanie aplikácie po prihlásení ešte jednu zložku. Tou je **xml_files**, kde sa ukladajú exportované otázky vo forme XML súborov. Z tohto umiestnenia sú následne aj načítavané.

Poslednou položkou v priečinku **Bc.praca** je položka **ADMIN**. V aplikácii nie je používaná. Existuje pre prípad, keď by bolo vyžadované, aby aplikáciu mohol ovládať ešte iný typ užívateľov s rozdielnymi právami.

Ukážka programu je k dispozícii online na internetovej adrese <<http://www.database.kx.cz/home.php>>. Bližšie informácie sú uvedené v sekcii 2.5.

Aplikácia bola testovaná na operačnom systéme Microsoft Windows XP. Používaný internetový prehliadač — Mozilla Firefox, softwarový webový server — Apache HTTP Server a internetový server — Internet Centrum¹.

¹<http://www.ic.cz>