

Virtuální edukační prostředí

Bc. Jarmila Chudá

Diplomová práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Vložit oficiální zadání bakalářské práce

Za cenné rady a připomínky děkuji svému vedoucímu diplomové práce Ing. Markovi Čandíkovi, Ph.D. a svým přátelům.

Souhlasím s tím, že s výsledky mé práce může být naloženo podle uvážení vedoucího diplomové práce, ředitele ústavu a institutu. V případě publikace budu uvedena jako spoluautor.

Prohlašuji, že jsem na celé diplomové práci pracovala samostatně a použitou literaturu jsem citovala.

Ve Zlíně, 10. 06. 2005

.....

ABSTRAKT

Abstrakt česky

Cílem mé diplomové práce bylo vytvořit pro potřeby studentů distanční formy studia virtuální edukační systém, který by jim pomohl ve vzdělávání. Systém by měl studentům nabízet aktuální studijní materiály a testy ověřující jejich znalosti. Registrovanému uživateli jsou přidělena různá práva. Uploadovat textové a binární soubory, následně je modifikovat mohou pouze vyučující. Všem přihlášeným je ale povoleno vkládat poznámky, upravovat je a navzájem spolu komunikovat.

Web je vytvořen pomocí HTML-kódu. Do stránek jsou vloženy PHP-skripty, které zajišťují jeho dynamičnost. Většina uživatelských dat je spravována pomocí databázového systému MySQL. Struktura celé aplikace byla navržena tak, aby byla maximálně přehledná. Grafická úprava je vytvořena použitím kaskádových stylů.

Abstrakt ve světovém jazyce

The aim of my work was created virtual on-line system for students in distance learning or individual programme to facilitate their study. Registered users are assigned varied authority. We can find texts and tests in system, which can upload on server or modify only teachers. All users can insert and edit notes and communicate each other.

Whole web is programmed in HTML-code. His dynamic ensure PHP-scripts. Most of user's data are controlled by database system MySQL. The structure of the web was projected for maximal lucidity and simplicity. Graphic format is generated using Cascading Style Sheets.

OBSAH

ÚVOD.....	7
1 INTERNET.....	8
2 TRENDY VE VZDĚLÁVÁNÍ NA BÁZI INFORMAČNÍCH TECHNOLOGIÍ	10
2.1 Distanční vzdělávání.....	10
2.1.1 Struktura učebních textů pro distanční vzdělávání	10
2.1.2 Zpracování textů	11
2.2 On-line vzdělávání	13
2.2.1 Prostředky transferu informací ve vzdělávání	14
2.3 E-learning.....	17
3 SOFTWAREVÉ POŽADAVKY K REALIZACI APLIKACE.....	19
3.1 Jazyk HTML	19
3.2 Kaskádové styly (CSS)	19
3.2.1 Použití CSS v HTML kódu	20
3.3 Javascript	21
3.3.1 Syntaxe jazyka a jeho používání v HTML	21
3.3.2 Obsluha formulářů	23
3.4 PHP	23
3.5 Spolupráce s databázemi.....	25
3.5.1 Databázové modely	25
3.5.2 Relační databázový model	25
3.5.3 DBMS - Systém řízení báze dat	28
3.5.4 Jazyk SQL	30
3.5.5 Architektura spolupráce PHP a MySQL s webovým serverem	33
3.6 Instalace serveru a PHP	35
4 SOFTWAREVÁ REALIZACE PROJEKTU	38
4.1 Charakteristika aplikace.....	38
4.2 Grafický vzhled aplikace.....	39
4.3 Jádro systému	40
4.3.1 Přihlášení do systému	41
4.4 Návrh databázového systému	42
4.4.1 Konceptuální schéma.....	42
4.4.2 Převod konceptuálního schéma do databázového modelu	43
4.5 Funkce systému na základě uživatelských práv.....	46

4.6	Popis ostatních částí vzdělávacího on-line systému	47
4.6.1	Registrace nového uživatele	47
4.6.2	Testy	48
4.6.3	Uživatelské rozhraní po vstupu do systému	51
ZÁVĚR.....		53
SEZNAM POUŽITÉ LITERATURY		54
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....		56
SEZNAM OBRÁZKŮ.....		57
SEZNAM TABULEK		58
SEZNAM PŘÍLOH.....		59

ÚVOD

V dnešní době se stal Internet pro mnohé z nás neodmyslitelnou součástí života. Najdeme zde obrovské množství informací, je velkým pomocníkem při jejich vyhledávání, zdrojem zábavy i vzdělání. Pro rychlost a přístupnost nepřehledného množství dat se z Internetu pomalu stává jedno z nejmocnějších médií.

Hlavními výhodami publikování na Internetu je především zveřejňování informací bez omezování rozsahu, redakčních uzávěrek, využívání hypertextových odkazů a v neposlední řadě při stále zvyšující se rychlosti přenosu dat, využívání multimédií.

Nové informační technologie samozřejmě přinesly možnosti jejich využití i v pedagogice. Tento přínos je hlavně viditelný u tvorby a používání distančních vzdělávacích materiálů, při komunikaci mezi studenty, učiteli apodobně.

Model výuky založený na informačních a komunikačních technologiích – on-line výuku – můžeme charakterizovat tím, že nemá prostorové ani časové omezení a může tak současně nabízet obrovské množství informací více studentům najednou. Na druhé straně použití takovýchto technologií vyžaduje zvýšené požadavky na vzdělávání jako jsou například: dostupnost počítače a připojení k Internetu, základní praktické zkušenosti s informačními technologiemi, silnější sebemotivaci ke studiu.

Cílem mé práce bylo vytvořit jednoduchý vzdělávací on-line systém, tak aby studentům usnadnil jejich vzdělávací proces. Systém by měl stále nabízet aktuální studijní materiály, uživatelé by měli mít možnost konzultovat své poznatky s ostatními přihlášenými, ověřovat si své získané znalosti. Součástí je také administrační prostředí pro vyučující. Web je vytvořen pomocí HTML-kódu. Do stránek jsou vkládány PHP-skripty, které zajišťují jeho dynamičnost a důležitá uživatelská data jsou spravována pomocí databáze MySQL. Strukturu stránek jsme se snažila navrhnout tak, aby byla aplikace maximálně přehledná. Jádro systému je navrženo takovým způsobem, aby se projekt mohl dále rozvíjet nebo aplikovat na jiné podobné problémy.

1 INTERNET

Je rozsáhlá celosvětová síť, významný činitel globalizace současného světa se všemi klady a zápory, které tento proces přináší. Má široké využití v mnoha oborech lidské činnosti. Z oblastí, ve kterých Internet přinesl zcela nové možnosti rozvoje a kvalitativního růstu lze jmenovat:

- ***Vzdělávání a vyhledávání informací***
 - *On-line vzdělávání* - zvyšování kvalifikace po Internetu, Internet umožňuje vyhledávání informací a nejnovějších poznatků z vědy, kultury, umění a dalších lidských aktivit.
 - *Vyhledávací služby* – slouží k ulehčení vyhledávání, jedná se o speciální uzly sítě – *vyhledávací servery*. Na základě zadaného textu jsou schopné vyhledat dostupné dokumenty obsahující hledaný text.
 - *Katalogová služba (tríděná)* – chová se jako skutečný katalog, např: Google , Yahoo, Atlas, AltaVista, Seznam.
 - *Netříděná služba* – indexuje výskyt klíčových slov, nalezené dokumenty mohou obsahovat klíčová slova jen okrajově – např. AltaVista, Kompas, Seznam, Google.
- ***Navazování kontaktů a vzájemná celosvětová komunikace*** – E-mail, Chat, videokonference apod.

Pomocí sítě Internet mohou spolu velmi pružně komunikovat lidé a organizace po celém světě. Kromě komerčního efektu to samozřejmě přináší také poznání jiných kulturních prostředí a odstraňování vzájemných předsudků a obav.

- ***Obchodování, prezentace a reklama – E-commerce***

Obchodování prostřednictvím Internetu a to jak maloobchodní prodej nejrozličnějších produktů, tak přímé obchodování a jednání firem. Prostřednictvím Internetu mohou živnostníci vyhledávat obchodní partnery, zadávat zakázky, apod. V současné době je také samozřejmostí zprostředkování bezhotovostního platebního styku. I reklama šířená na Internetu je velmi efektivní.

Tabulka 1 Klasifikace obchodních modelů

	Admin	Business	Customer
Admin	A-A	A-B	A-C
Business	B-A	B-B	B-C
Customer	C-A	C-B	C-C

Vysvětlení: Admin = státní správa

Business = firma, organizace

Customer = zákazník, spotřebitel

Například e-banking je model B-C (Business-to-Customer), označován B2C

2 TRENDY VE VZDĚLÁVÁNÍ NA BÁZI INFORMAČNÍCH TECHNOLOGIÍ

Stále častěji slycháme pojmy jako je distanční vzdělávání, on-line vzdělávání a e-learning. Proto zde ve stručnosti nastíním rozdíly mezi těmito formami výuky.

2.1 Distanční vzdělávání

Je doplňkem klasických forem denního a dálkového studia. Jedná se o samostatné studium podporované speciálně zpracovanými studijními pomůckami. Studenti jsou fyzicky odděleni od vzdělávací instituce, která jejich studium řídí a podporuje. [13]

V rámci distančního studia by texty měly být:

- *samoinstrukční* – měly by poskytnout ucelené poznatky a hlavně obsahovat motivační prvky k tomu, aby studující splnil stanovené cíle studijního programu.
- *strukturované* – aby podněcovaly a umožňovaly systematické a soustavné studium, měly by být přehledné a zprostředkovat požadovanou obsahovou náplň předmětu.

2.1.1 Struktura učebních textů pro distanční vzdělávání

Skripta a učebnice v denním studiu umožňují především zopakování, shrnutí a doplnění znalostí poskytovaných v průběhu přednášek. Cílem distančních textů je hlavně kvalitně nahradit přímou výuku. Učební texty pro distanční vzdělávání jsou již na první pohled výrazně odlišné od běžných učebnic. Nápadné je zejména členění textu, grafická úprava, "odlehčení" pomocí obrázků, schémat, grafů, různých symbolů a jiných názorných prvků.

Výkladová část by měla být rozdělena do kapitol a podkapitol. Při strukturaci kapitol používáme nejvýše tři úrovně nadpisů (při větším počtu se z textu vytrácí logická návaznost a text se stává nepřehledným). Kapitola má mít takový rozsah, aby její obsah průměrný student dokázal vnímat od začátku do konce. Nejčastěji uváděný odhad časové délky potřebné k nastudování patnácti stran studijního materiálu je dvě až tři hodiny. Pokud je do těchto stran zařazen korespondenční úkol, pak je nezbytné čas na jeho vypracování k tomuto odhadu připočítat. Jestliže se jedná o rozsáhlejší dílo (více než čtyřicetistránkový a do více kapitol členěný text), je nutné uvádět údaj o časové náročnosti každé kapitoly. U méně obsáhlých materiálů stačí začlenit do úvodu čas potřebný k nastudování celého textu.

Struktura kapitoly by měla být následující:

- studijní cíle;
- klíčová slova (na začátku nebo konci);
- uvedení do problematiky;
- samotný text (učivo);
- průvodce studiem (textem) - vstup autora do textu, kdy se čtenářem hovoří, radí mu, povzbuzuje jej, motivuje ho ke studiu, odvolává se na vlastní zkušenosti a názory;
- příklady, cvičení, úkoly, otázky (kontrolní složka, vedou k zamyšlení, sebehodnocení);
- případové studie;
- shrnutí;
- úkoly nebo otázky pro zpětnou vazbu, testy;
- další doplňkové zdroje informací a odkazy na ně, doporučená literatura, případně další doplňkové zdroje informací (články, reportáže, osobní zkušenosti, videonahrávky, CD atd.);
- řešení a klíč k sebehodnotícím aktivitám.

Odpovědi na otázky a cvičení, výsledky testů lze zařadit přímo za ně. Častějším způsobem bývá umístit je na konec kapitoly nebo celého studijního textu s označením klíč, výsledky testů, správné odpovědi, řešení apod.

2.1.2 Zpracování textů

Jelikož textové materiály musí nahradit živé přednášky, jejich vyjadřovací styl a jazyk, musí být pečlivě volena grafická úprava textů. Forma, v jaké je text předložen, hraje významnou pedagogickou roli, protože nahrazuje chybějící školní atmosféru [10]. Jestliže chce učitel, aby si studenti odnesli z jeho výuky co nejvíce nových informací, chce-li je zaujmout a podnítit k samostudiu, snaží se např. pomocí gest, modulací hlasu, zdůrazňováním některých pasáží textu, zdůraznit důležité prvky učiva. V distančním studiu přebírá tento úkol různé zvýrazňování textu a grafická úprava odstavců.

Text proto členíme na dva sloupce:

- *hlavní sloupec* - je prostorem pro učivo, hlavní textovou náplní studijního materiálu;
- *popisný sloupec* – jedná se o asymetrický sloupec zabírající v distanční publikaci dvacet až třicet procent šířky tiskové plochy stránky. Plní významnou metodickou funkci. Obsahuje stručné komentáře (glosy) k textu v hlavním sloupci (zvláště k

jednotlivým významnějším odstavcům), může zde být umístěn i stručný popis obrázků, tabulek a text úkolů k obrázkům a k tabulkám.

Nejčastěji se v popisném sloupci objevují výukové symboly, malé obrázky, ikonky apod. Celým studijním textem tak prostupuje permanentní glosátor (komentátor, vykladač), který upřesňuje, oznamuje, upozorňuje, odkazuje, ukládá úkoly a povzbuzuje studujícího.

Popisný sloupec slouží i jako místo pro poznámky, čtenář si do něj může vpisovat další zajímavosti, vlastní postřehy, nápady, připomínky apod.

Tento sloupec slouží k tomu, aby zdůraznil význam a důležitost jednotlivých odstavců, doplnil nebo objasnil text, podal vysvětlení, sdělil pokyny, popsal úkol nebo tabulku v hlavním sloupci.

Grafické úpravě studijního materiálu je nezbytné věnovat dostatečnou pozornost. Většina studujících si lépe zapamatuje a následně vybaví nové informace, pokud jsou znázorněné jako obrázky, mapy, grafy, tabulky, symboly, schémata, diagramy, ale i vtipy nebo karikatury.

Následující obrázek ukazuje ikony, které bývají umístěny v popisném sloupci.



Obr. 1 Ukázka rozdělení textu do popisného a hlavního sloupce

Výhody distančního studia: student si určuje čas, místo, rychlost studia.

Nevýhody distančního studia: malá zpětná vazba, nemá ihned komu klást otázky.

2.2 On-line vzdělávání

Jedná se o takový druh vzdělávání, kde komunikace mezi studenty a učiteli, resp. mezi studenty navzájem se mimo jiné realizuje v reálném čase, prostředky počítačových a telekomunikačních technologií.

Výuková metoda, která je specifická pro on-line vzdělávání (všeobecně pro distanční vzdělávání) je tzv. *tutoriál*. Jedná se o nepovinné střetnutí studujících s tutorem. Cílem je získat přehled o studijních povinnostech, nárocích na samostatnou práci a zkoušku, získat informace o závazných termínech studia, a v neposlední řadě zodpovězení individuálních otázek studentů. Hlavním cílem tutoriálů je získat zpětnou vazbu studentů, která povede k dosažení co nejlepších studijních výsledků.

Jsou stanoveny podmínky, jakým způsobem předložit látku v on-line studiu.

Musíme brát v úvahu:

- legislativní, normativní a administrativní aspekty;
- změny v podání obsahu studia;
- u materiálů vystavených na webu se jedná o problém ochrany autorských práv;
- problém autorství výsledků testů prováděných přes Internet.

Role učitele v on-line studiu

Pro označení učitele v on-line vzdělávání se používá termín instruktor (tutor).

Jedná se o speciálně vyškoleného pedagoga, který by měl plnit tyto základní funkce:

- *metodické vedení studentů* - pomoc při výběru programu, předmětů, studijních pomůcek; podpora studujících v průběhu studia; konzultace; průběžné hodnocení studia (zadávání samostatných prací);
- *shromažďování, třídění, zpracování, odesílání a vyhodnocování formulářů, testů a materiál* pro zabezpečení zpětné vazby;
- *zvládání informačních technologií pro přenos informací*;
- *administrativní povinnosti* – zabezpečuje zpětnou vazbu směrem ke vzdělávací instituci, k autorovi textů, k webmastrovi, atd.

Tutor by dále měl:

- povzbuzovat a podporovat do vzdělávání;
- včas reagovat na příspěvky studentů v diskusích a na jejich výsledky v testech a pracích;

- měl by být on-line 5 dní v týdnu;
- mít schopnost dobré komunikace v on-line prostředí, kdy se nevyužívá videokonferencí, ale kde je slovní komunikace nahrazena „klávesnicí“.

Výhody on-line vzdělávání:

- *prostorová decentralizace* – studovat můžeme kdekoli tam, kde existuje vyhovující připojení k Internetu a dostatečné technické vybavení počítače pro studium;
- *časová nezávislost* – on-line vzdělávací kurzy jsou uloženy na serveru, který umožňuje nepřetržitý přístup k vzdělávacím materiálům;
- *možnost studovat vlastním tempem*;
- *možnosti anonymity* – eliminace možných diskriminujících faktorů, jako je rasa, věk, vzhled, tempo výuky apod.;
- *přístup ke zdrojům* – je jednoduché vytvořit odkazy na aktuální zdroje elektronických informací a aktualizovat materiály;
- *komunikace* – používání způsobů komunikace, které jsou založené na využití základních služeb počítačových sítí např. diskuse ve skupinách – tzv. služby dávkového charakteru, diskuse uvnitř skupin – tzv. služby interaktivního charakteru;
- *komplexní podpora samostatného studia* – on-line edukační materiály by měly obsahovat aktivační prvky studia, organizační informace o studiu, manuály pro technickou podporu, návody na způsob on-line studia.

Nevýhody on-line vzdělávání:

- *odlišný přístup k informačním technologiím a technické problémy* – různá rychlost připojení a tím i časově omezená možnost přístupu k informacím, finanční náklady spojené s připojením k Internetu, nestejně technické vybavení počítačů;
- *nedostatek schopnosti samostatného studia* – tato forma vzdělávání umožňuje relativně velkou míru volnosti, což může vést k problémům u studentů s menší sebedisciplínou a odpovědností ke studiu.

2.2.1 Prostředky transferu informací ve vzdělávání

Vzhledem k distančnímu charakteru vzdělávání on-line výuka vyžaduje specifické metody a technické prostředky pro přenos informací od instruktora směrem ke studentům.

On-line vzdělávání disponuje následujícími formami předávání informací:

- **Audio a rozhlasové vysílání** - používané hlavně při výuce jazyků. Finanční náklady jsou nižší než v případě videonahrávek. Audio soubory jsou distribuované i na Internetu. Velkou nevýhodou je to, že těmto informacím chybí vizuální složka.
- **Videonahrávky a televizní programy** – poskytují informace v audiovizuální formě. Kvůli jejich datovému rozsahu vyžadují vyšší nároky na přenosovou rychlost při stahování z Internetu. Ke spuštění vyžadují specializované technické prostředky (televizní přijímač, videopřehrávač, televizní programy mají pevně stanovený vysílací čas).
- **Interaktivní CD média a interaktivní edukační software** – umožňuje studovat vlastním tempem, edukační materiál může obsahovat prvky hypertextu, který představuje hlavní rozdíl mezi elektronickými a tradičními formami zveřejňování informací.
- **Elektronické konference (Mailing Lists)** – jedná se o diskuse na předem určené téma uskutečňované elektronickou poštou. Nové příspěvky dostávají registrovaní účastníci na svůj e-mail. Neregistrovaní účastníci mají možnost vyhledávat informace v archivech diskusí. Nevýhodou je relativně velké množství informací a názorů, které se nahromadí v průběhu střetnutí, a proto je potřeba docela velkého množství času k pročtení těchto zpráv.

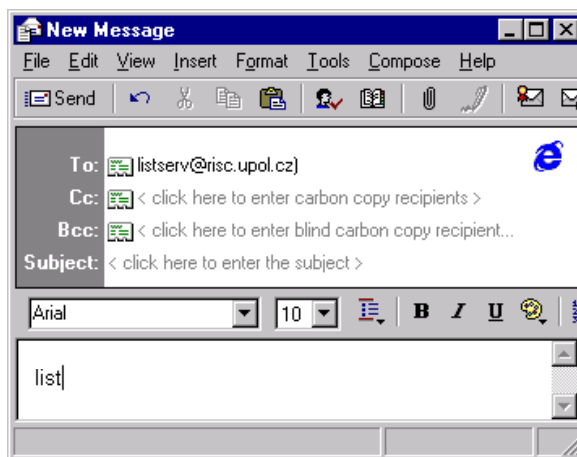
Spravování konferencí zajišťují systémy *listservery* (např. listserv@risc.upol.cz). Chceme-li se zúčastnit nějaké konference, musíme se přihlásit vždy na adresu příslušného listserveru a mít přístup k elektronické poště. Příspěvky se zasílají na adresu příslušné konference. Chceme-li z konference odejít, musíme se odhlásit na adrese příslušného listserveru. Rozlišují se adresy pro práci v konferenci (příspěvky) a pro práci s listservery (přihlášení, odhlášení).

Některé příkazy pro práci s listserverem:

LIST - kompletní seznam konferencí na příslušném listserveru

SUBSCRIBE název_konference - přihlášení do konference

UNSUBSCRIBE název_konference - odhlášení z konference



Obr. 2 ukázka komunikace v elektronické konferenci

- **Diskusní skupiny** – známy také pod názvem Internet News. Jsou analogické elektronickým konferencím s tím rozdílem, že odesílané příspěvky jsou na servery řazené podle jednotlivých tematických okruhů, komunikace se uskutečňuje mimo e-mail a jsou přístupné komukoliv. Velkou nevýhodou je, že se ve velkém množství diskusních materiálů objevují nepřesné nebo nesprávné informace. Diskusní skupiny si můžeme představit jako celosvětové nástěnky, které si čteme. Pro prohlížení příspěvků potřebujeme prohlížečího klienta (např. Outlook Express). Služba nepoužívá protokoly e-mailu, ale protokol NNTP (Network News Transfer Protocol). Diskusní skupiny jsou hierarchicky uspořádány, existuje jich desítky tisíc. Na nejvyšší úrovni je několik základních skupin. Příklady diskusních skupin: *alt* - různé skupiny, *rec* - rekreační aktivity, *sci* - věda, *soc* - společenské vědy, *comp* – počítače, *misc* – různé, *cz* - české skupiny.
- **Chat** – jedná se o synchronní on-line textovou diskusi dvou nebo více účastníků. Poskytuje prostor pro přímý rozhovor ve formě krátkých textových zpráv s možností okamžité reakce na zprávu. Nevýhoda chatu je ta, že pokud je na chatu přihlášeno více diskutujících, komunikace se stává nepřehledná.
- **Videokonference** – jedná se o synchronní on-line obrazovou diskusi alespoň dvou účastníků. Účastníci se navzájem vidí. Nevýhodou jsou zvýšené technické požadavky na počítač (kamera, paměť počítače) a na komunikační linky (vysoká rychlost připojení, stabilita spojení).

- **Články umístěné na Internetu** – studující mohou články ukládat, dále zpracovávat a zveřejňovat. Na Internetu existuje nepřeberné množství dat, jejich nevýhodou je, že informace z některých zdrojů mohou být nespolehlivé.
- **Interaktivní internetová místa** – na některých serverech jsou vyhrazena místa, kde mohou studenti číst aktuální informace související s jejich vzděláváním a mají zde vyhrazený prostor k publikování vlastních prací. Jednou z nevýhod je, že tato činnost může odvádět od jiných povinností.
- **On-line interview** – znamená rozhovor vedený prostřednictvím elektronické pošty, pomocí chatu nebo videokonference s osobami mimo kurz. Studenti komunikují s vybranými osobami, shromažďují potřebné informace, které předávají do on-line kurzu. O interview můžeme požádat kohokoli, ale oslovená osoba samozřejmě nemusí s rozhovorem souhlasit.
- **Případové studie** – tato činnost podporuje myšlení, studujícím je předložen problém, který musí řešit. Případové studie vyžadují vyšší úroveň přemýšlení a dostatečný teoretický základ vědomostí v řešené problematice.
- **Týmové projekty** – skupiny studentů mohou navštívit odborné semináře, pracoviště atd. a vést zde rozhovory, shromažďovat informace a začít pracovat na projektu. Výhodou takového přístupu je, že je možné do on-line výuky implementovat nejnovější informace a poznatky o praktickém využití probírané látky. Nevýhodou je možný nerovnoměrný podíl práce v týmu.

2.3 E-learning

E-learningem rozumíme multimediální interaktivní vzdělávání, praktikované pomocí počítačových kurzů, které jsou distribuované např. na CD-ROMech, Internetu, Intranetu.

Od on-line vzdělávání se odlišuje tím, že obsahuje řídicí složku.

Podle způsobu komunikace mezi lektorem a studenty rozdělujeme e-learning na:

- *asynchronní e-learning* – student samostatně prochází připraveným vzdělávacím programem a záleží na něm samotném, kdy výuku uskuteční. Volí si vlastní tempo průchodem látkou;
- *synchronní e-learning* – nabízí možnost vzájemné komunikace mezi všemi účastníky prostřednictvím tzv. virtuálních učeben. Studenti s lektorem jsou ve stejné době na stejném místě, kde se mohou účastníci pomocí synchronních technologií střetnout i tehdy, když jsou od sebe fyzicky vzdáleni v různých lokalitách.

- *kombinovaný e-learning*

Všechny tři přístupy spojuje využívání počítačů a počítačových sítí pro podporu učení a vzdělávání.

3 SOFTWAREVÉ POŽADAVKY K REALIZACI APLIKACE

Pro tvorbu WWW stránek jsem zvolila kromě HTML-kódu skriptovací jazyk PHP, a to z důvodu existence spousty materiálů, knih, návodů na Internetu a mnoha programů, které usnadňují tvorbu projektů v nich psaných (např. PHP Expert Editor 3.21, PSPad 4.3.2).

Alternativou k PHP je technologie ASP. S tou se můžeme setkat téměř výhradně u placených služeb. Využívají ji především programátoři používající technologie firmy Microsoft. V poslední době se tato firma snaží ASP prosadit tím, že uvolňuje zdarma nástroje pro tvorbu skriptů v tomto jazyce. Užitečnou službou pro softwarové projekty je podpora databází, které slouží jako úložiště uživatelských dat. Existují dvě základní kombinace: PHP s MySQL a ASP s MSSQL. U zdarma dostupných služeb se setkáváme hlavně s první kombinací.

Pro grafickou úpravu stránek jsem využila vlastností CSS stylů.

Jelikož každá z jmenovaných technologií má jinou funkci a používá jiné struktury, v další části stručně nastíním jejich hlavní možnosti a funkce.

3.1 Jazyk HTML

O HTML-kódu je toho napsáno hodně. Ti kdo vytvářeli webovou stránku se s ním zákonitě museli setkat. A pokud jste opravdu začátečníci a nemuseli se hned od začátku učit všechny značky jazyka HTML, existují HTML editory s režimem WYSIWYG. V tomto režimu se rozmisťují prvky stránky podobně jako v aplikaci Word a program za nás sám generuje kód. V takto vytvořeném kódu pak jednoduchou změnou několika znaků upravíme například pozici obrázku a postupem času se takto naučíme velkou část jazyka HTML. Jakmile jednou zvládnete většinu tohoto jednoduchého značkovacího jazyka, režim WYSIWYG už určitě nikdo nepoužije, protože takto tvořený kód je většinou nepřehledný a málo optimalizovaný. Dobrým průvodcem nejen pro začátečníky jsou webové stránky na adrese <http://www.jakpsatweb.cz>.

3.2 Kaskádové styly (CSS)

Původně byl jazyk HTML navržen tak, aby jeho pomocí šlo lehce vyznačit jednotlivé logické části dokumentu. Nebyl přitom kladen důraz na přesnou definici výsledného vzhledu. Webová prezentace se postupně rozšířila i do komerční sféry a vznikly náročnější požadavky na grafický vzhled dokumentu. Tyto požadavky byly částečně

uspokojeny přidáním různých atributů (např. align) a elementů (např. FONT) pro optimálnější ovládání vzhledu a formátování dokumentu. Ale i tak použití mělo velké nevýhody. Za prvé se v HTML dokumentech místo jejich struktury začal vyznačovat spíše grafický vzhled a druhý problém spočíval ve velké pracnosti a neflexibilitě tohoto řešení, kdy se způsob formátování musel nastavit jednotlivě pro každý element dokumentu [1].

Obě tyto nevýhody odstraňují kaskádové styly (CSS - Cascading Style Sheets). Pomocí stylu lze jednoduše definovat druh písma, formát odstavců, odkazů a další vlastnosti elementu. Následně se tato definice použije jednotně v celém dokumentu. V něm se pak zaměříme pouze na strukturu informace, protože grafický vzhled je definován stylem.

Princip CSS je velmi jednoduchý: lze modifikovat výchozí vlastnosti elementů, které již v HTML existují (jako třeba H1, P, DIV, A atd.). Můžeme vytvářet nové styly a přiřazovat jim vlastnosti. Formátovací možnosti jsou velice široké a svou komplexností mají strukturu programovacího jazyka. Mezi algoritmické principy u CSS patří například seskupování a dědičnost definicí, tvorba tříd atd.

3.2.1 Použití CSS v HTML kódu

Jak bylo řečeno, v CSS stylu můžeme:

a) *předefinovat styl přímo u měněného elementu:*

```
<p style="text-align: center">Tady bude následovat text odstavce </p>
```

b) *stylopisem* = seznamem stylů zapsaným mezi <style> </style> ve hlavičce dokumentu

```
<html>
<head><title>Jednoduchý příklad se stylopisem</title>
<style>
h2    {color: blue; font-style: italic}
a: visited {color: red; text-decoration: underline}
</style>
</head>
<body>
<h2>Nadpis druhé úrovně</h2>
<a href="http://diplomka.euweb.cz">navštívený odkaz</a>
</body>
</html>
```

c) externím souborem CSS (externí stylopis)

Tento způsob je výhodný zejména proto, že změnou jednoho souboru měníme vzhled ve všech dokumentech, které mají tento soubor „přilinkován“. Tedy jedinou úpravou můžeme změnit vzhled celého webu.

Propojit soubory s CSS styly a HTML dokument provedeme takto:

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

Touto definicí v hlavičce stránky říkáme prohlížeči, že pro tento dokument bude použit CSS styl, umístěný v souboru definovaném v hodnotě atributu *HREF*.

Více o CSS stylech uvádí již výše uvedený web <http://www.jakpsatweb.cz> nebo například literatura [1] a [3].

3.3 Javascript

JavaScript je skriptovací jazyk, který se používá ve stránkách HTML. Jeho počátky nalezneme v době, kdy světlo světa spatřila druhá verze prohlížeče Netscape Navigator. V něm byla totiž poprvé zabudována podpora pro tento skriptovací jazyk [3].

Skripty se zapisují přímo do HTML kódu stránky a prohlížečem jsou interpretovány na klientském počítači. Proto není pro jejich tvorbu zapotřebí překladače a zdrojové kódy jsou všem přístupné. JavaScript je objektový jazyk. Do objektů je zapouzdřeno okno prohlížeče, obsah stránky a další prvky. Jeho syntaxe vychází z jazyka C++, proto je *case sensitive* (záleží na velikosti písmen v zápisu kódu).

Stejně jako je tomu s interpretací HTML, je tomu i s JavaScriptem. Každý prohlížeč podporuje jinou verzi jazyka, a navíc si ji v mnoha případech interpretuje po svém. Vznikají tak velké nekompatibility mezi prohlížeči, což je jedna z nevýhod. Navíc prohlížeč nemusí JavaScript vůbec podporovat nebo jej může mít uživatelem zakázán ve svém nastavení.

V mé diplomové práci jsem ho využívala u webových formulářů – např. při vyhodnocování testů, kontrole nevyplněných položek formuláře atd.

3.3.1 Syntaxe jazyka a jeho používání v HTML

Celý dokument se vyhodnocuje jako objekt obsahující další objekty až po jejich elementární části. Díky této vlastnosti můžeme přistoupit zvlášť ke každému prvku HTML stránky.

Dále uvádím malou ukázkou zdrojového kódu funkce, která kontroluje, zda bylo ve formuláři vyplněno uživatelské jméno a správný formát e-mailu.

```
<script language="JavaScript">

<!--
function kontrola1(frm){
if (frm.name_formuláře.value=='') {
    alert("Nemáte vyplněno uživatelské jméno.");
    frm.name_formuláře.focus();
    return false;
}
else if (frm.name_formuláře.value=='') {
    alert("Musíte zadat e-mail.");
    frm.name_formuláře.focus();
    return false;
}

if (frm.name_formuláře.value!="")
    if(window.RegExp){
        re=new RegExp("^[^.]*(\\.[^.]*)*@[^[^.]([a-z]{2,4})$");
        if(!re.test(frm.name_formuláře.value)){
            alert("Špatně zadaná e-mailová adresa.");
            frm.name_formuláře.select();
            frm.name_formuláře.focus();
            return false;
        }
    }
}
}
```

Celý tento kód se definuje v hlavičce HTML dokumentu. Skript je uzavřen do párového tagu `<SCRIPT>`, který definuje část, kde bude kód JavaScriptu. Část textu uzavřená do HTML komentáře (`<!-- ... // -->`) říká starší verzi prohlížeče, který JavaScript nepodporuje, že jej nemá interpretovat jako HTML kód. Zbytek textu je kód JavaScriptu. Proměnné začínají definicí slovem *var*. Definovat proměnnou lze kdekoliv během skriptu. Jedná se jazyk beztypový, tzn. že při deklaraci proměnných se nespecifikuje, jakého jsou typu, a konverze hodnot probíhá automaticky.

Funkce se zapisují obdobně jako v C jazyce, tělo funkce pak musí být v bloku kódu (složených závorkách). Textové hodnoty jsou uzavřeny v apostrofech nebo dvojitých uvozovkách. Základní řídicí struktury jsou podobné syntaxi jazyka C++.

3.3.2 Obsluha formulářů

Pro obsluhu formulářových položek a dat se používají objekty *DOCUMENT* (dokument - www stránka) a *FORM* (formulář). Pro přístup k položkám je důležitý atribut *NAME* (jméno). Přes tyto jména jsou prvky jako objekty dostupné. Jejich hodnoty a vlastnosti lze pochopitelně i měnit, aplikovat na ně funkce atd.

JavaScript je často používán pro kontrolu dat zadaných ve formuláři. Ta se provádí po stisku tlačítka pro odeslání. Pro volání funkce nám slouží atributy HTML elementů nazývané *události*.

3.4 PHP

Tato zkratka označuje skriptovací jazyk zabudovaný na straně serveru, který v současnosti patří k nejoblíbenějším a nejrozšířenějším jazykům mezi programátory webových aplikací. Je to nejen díky jeho spolehlivosti, výkonnosti a jednoduchosti, ale také díky tomu, že je volně šiřitelný - Open Source. Právě tyto vlastnosti ho řadí k současné světové špičce. Argumentem pro jeho využití je mimo jiné spolupráce PHP s databázovými systémy. Dnes tento jazyk najdeme v nabídce většiny webových serverů, především pak na webhostingu poskytovaném zdarma (např. <http://www.webzdarma.cz/>).

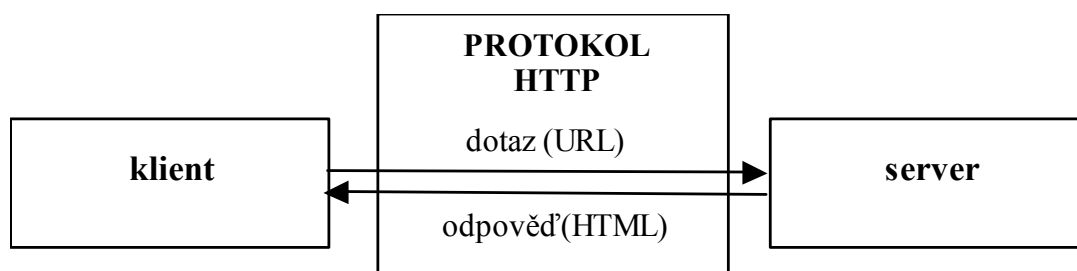
PHP byl vytvořen v roce 1994 a je to původně práce jednoho člověka - Rasmuse Lerdorfa. Vzápětí se ho chytili další talentovaní lidé a prošel velkými předělvkami a v současné době je dostupná verze PHP 5. Zdarma je možné je stáhnout na Internetu z adresy: <http://www.php.net/>, kde nalezneme dokumentaci, manuály a další aktuální informace týkající se PHP.

Pracuje uvnitř dokumentu HTML a dává mu tak schopnost generování požadovaného obsahu. Skripty v PHP jsou HTML-stránky doplněné o výkonné příkazy. Aby server od sebe odlišil HTML-stránku a stránky zapsané v PHP, ukládají se PHP-skripty do souboru s příponou *.php*. Takovýto soubor se musí uložit do adresáře, který je součástí stromu dokumentů WWW-serveru. Tento adresář je na každém serveru jiný.

Znaky '`< ?php`' a '`? >`' slouží k oddělení příkazů PHP od HTML-kódu. Příkazy se píší dovnitř těchto znaků. Je možné kombinovat HTML-kódy s příkazy PHP.

Jakmile WWW-server obdrží požadavek na soubor, který končí na tuto příponu, předá požadavek systému PHP. Ten se postará o provedení příkazů uložených v souboru a výsledek předá zpět serveru. Server výslednou stránku odešle klientovi jako odpověď. Skript se nesmí v prohlížeči otevírat jako soubor přímo z disku, protože takto se nezobrazí

jeho výsledek. Musí být před zasláním interpretován systémem PHP. Interpret je vyvolán pouze v případě, kdy k dokumentu přistupujeme přes WWW-server, tj. pomocí URL adresy, která začíná na http. Další výhodou PHP je to, že není svázán s žádným konkrétním serverem, může běžet na libovolném (např. Apache, IIS4 – Internet Information Server 4.0, PWS – Personal Web Server).



Obr. 3 Průběh komunikace mezi prohlížečem a WWW-serverem

Rozlišujeme tři základní prostředky služby:

- *URL (Uniform Resource Locator)* – jednoznačně určuje identifikaci informačního zdroje.

Můžeme si ji představit jako přesnou adresu dokumentu.

Obecný tvar URL: schema://specifická část

- *HTTP (Hypertext Transfer Protocol)* – protokol ke komunikaci mezi klientem a serverem.
- *HTML (HyperText Markup Language)* – jazyk pro tvorbu WWW-stránek.

3.5 Spolupráce s databázemi

Databázový systém je chápán jako společné označení pro:

- *datové soubory* uchovávané v centrálně zpracovávané struktuře dat zvané *databáze*,
- *obslužné speciální programové vybavení*, které nazýváme *Systém řízení báze dat* (DBMS - Database Management System též SŘBD).

Databázi můžeme tedy specifikovat jako souhrn logicky uspořádaných a uložených dat zpracovávaných odděleně od nadstavbových aplikačních programů, i od informačního systému [5].

Databáze jako taková tedy zahrnuje:

- *datové prvky* - vlastní data;
- *logické vazby* - vztahy mezi datovými prvky;
- *integritní omezení* - podmínky kladené na data;
- *logické schéma* - uživatelský popis struktury dat.

Souhrn postupů, nástrojů a technik využívaných v souvislosti s databázemi je označován jako *databázová technologie*.

3.5.1 Databázové modely

Databázový model definuje model ukládání dat, přičemž každá databáze může využívat jiný model. Během vývoje vyprofilovaly následující databázové modely:

- *souborové*: hierarchické, síťové a relační
- *objektové*: objektově orientované a objektově relační

Souborové modely jsou historicky nejstarší a jejich název vychází z pohledu ukládání dat v souborech.

Objektové modely jsou založeny na vlastnostech objektově orientovaných jazyků, které sebou přináší celou řadu nových logických vztahů mezi objekty, které v souborových systémech nelze běžně implementovat, jako např. dědičnost, souhrnný obsah vlastností daného objektu, vztahy rodič - potomek, atd. Nevýhodou objektových modelů, je že databázové systémy, které tyto modely podporují nemají standardizovaný jazyk DBMS.

Moje aplikace využívá k ukládání dat relační databázový model, který popisují níže.

3.5.2 Relační databázový model

Definoval ho v roce 1970 E. F. Codd.

Relace je pojata z matematického hlediska, tj. jako podmnožina kartézského součinu množin dat. Je to tabulka, které přiřazujeme jméno a je složená z řádků a sloupců. Synonymum pro relaci je také: *instance*, *objekt*, *table*.

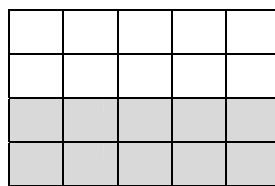
Relační databázi můžeme tedy charakterizovat jako soustavu jednoduchých relací, přičemž jsou definovány vztahy mezi relacemi [4]. Nad relacemi se provádějí databázové operace.

Nevýhodnou relačních databází je, že při požadavku na maximální nezávislosti dat, odstranění redundance a zajištění konzistence, musí být zpracovávána data rozložena na mnoho entit, což se výrazně projevuje na vzdalování se jednoduchosti dosažení dat uživatelem.

Entita

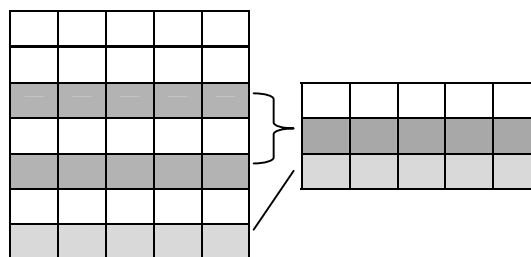
Entita představuje řádek v relaci, tzn. entity představují datové údaje uložené v databázi.

Výběr entit z relace se označuje jako *selekce* či *restrikce*. V rámci selekce může být porovnáváno několik domén, přičemž lze jednoduché formule sdružovat pomocí *logických spojek* and, or, not, implies.



Obr.4 Selekcce v relaci

Sdružování entit do jedné entity se nazývá *agregace* (Obr. 5). Vlastní agregace se provádí na základě srovnání hodnoty či hodnot z domény (*dom()*) pomocí porovnávacích predikátů a logických spojek, stejně jako při selekci. Hlavním účelem agregace je možnost vyčíslit nějaký údaj z agregovaných entit. Mezi nejzajímavější a běžně implementované agregované údaje patří např: počet agregovaných hodnot *COUNT*, součet hodnot *SUM* a výpočet průměrné hodnoty *AVG* z hodnot *dom()* agregovaných entit, zjištění min. či max. hodnoty *MIN*, *MAX*.

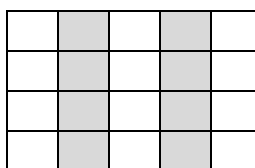


Obr. 5 Zobrazení agregace v relaci

Atribut

Atributem označujeme sloupec v relaci. Atribut (sloupec, pole) popisuje jaké údaje jsou v jednotlivých entitách za sebou uloženy. Každý atribut má definovaný *název* a *formát*. V rámci definice atributu se vymezuje rozsah platných hodnot ukládaných do entit - *integrita*.

Výběr některých atributů z relace je nazývá *projekce*.



Obr. 6 Projekce v relaci

Doménou rozumíme seznam všech hodnot v jednom daném atributu. Bývá označována jako $\text{dom}(A)$, kde A je název atributu. V rámci RDM (relačního datového modelu) je požadováno, aby všechny hodnoty $\text{dom}()$ byly stejného formátu.

Klíč

Mezi entitami neexistuje žádná hierarchie, avšak jakýkoliv údaj v ní může být použit pro přístup k jiné entitě relace. Klíč je atribut, který jednoznačně identifikuje danou entitu v relaci. Jednoznačnou identifikaci entity určuje *primární klíč* (*PRIMARY KEY*), tzn. klíč musí mít pro každou entitu v relaci jinou hodnotu (např. id uživatele).

Primární klíč v relaci plní tyto základní funkce:

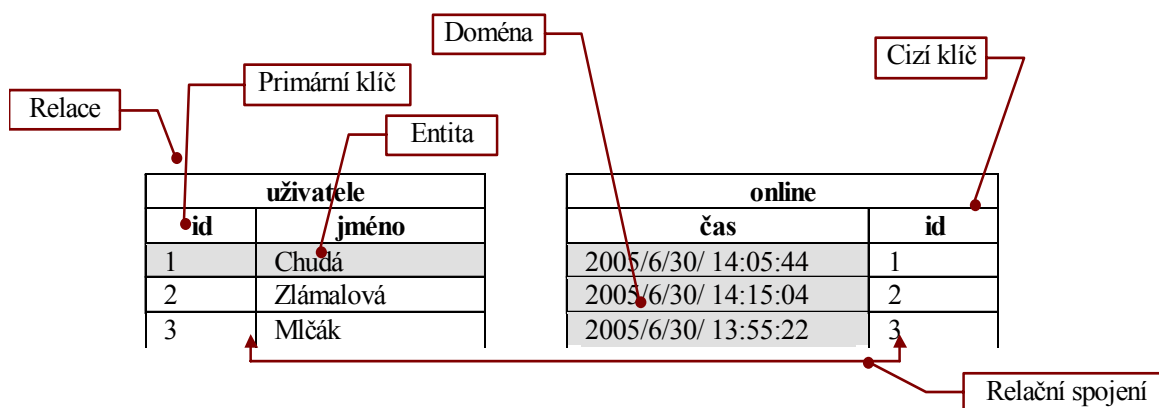
- slouží k efektivnímu vyhledávání záznamů v relaci;
- slouží k vytváření relačního spojení mezi relacemi;
- zabezpečuje referenční integritu mezi relacemi, nesmí obsahovat prázdnou hodnotu.

Při tvorbě primárního klíče je vhodné volit takové atributy v entitě, pomocí kterých budou entity nejčastěji vyhledávány nebo nejčastěji provázány s entitami v jiné relaci.

V RDM existují klíče:

- *vlastní klíč* - tj. klíč, který je uložen v téže relaci, ve které identifikuje entitu;
- *cizí klíč* - tj. klíč, který je uložen v jiné relaci než v té, kde identifikuje entitu. Slouží k relačnímu spojení. V RDM je přípustné, že cizí klíč může být i prázdný.

Pomocí nevlastních a vlastních klíčů je realizováno *relační spojení*, a je možné provádět sjednocení relací do jedné relace. Tabulka kde je definován vlastní klíč se nazývá master, nadřazená. Relace která obsahuje cizí klíč, se nazývá závislá.



Obr. 7 Základní pojmy v relační databázi

Integrita databáze

Integritou označujeme vlastnost, která vymezuje hodnoty atributů a relačních spojení. V rámci integrity může být omezeno vkládání, smazání nebo změna entity v závislosti na již vložených entitách nebo vkládaných hodnotách atributů.

Zvláštním typem integrity je *referenční integrita*, tzn. jakých hodnot může nabývat relační spojení nebo jaké činnosti jsou definovány v rámci relačního spojení. Např. do relace výrobky, nelze vložit výrobek, v případě, že ještě nebyl definován výrobce (referenční integrita kontroluje existenci výrobce), a v případě, že referenční integrita je nastavena na funkci automatického mazání, a bude-li smazán konkrétní výrobce, budou smazány pomocí relačních spojení i všechny výrobky spojené s tímto výrobcem.

Integritu zajišťuje DBMS vlastními prostředky. Integritu zpravidla definujeme při tvorbě databáze a je součástí DDL.

3.5.3 DBMS - Systém řízení báze dat

SŘBD se stará o přístup k údajům uloženým v databázích. Mezi takové programy patří například Oracle, MS SQL Server, Sybase, Informix, Progress. Nejedná se o programy

nikterak levné, jejich cena se může pohybovat v desítkách tisíc korun. I zde existují programy šířené zdarma jako freeware. Jde například o MySQL používající jazyk SQL, jehož možnosti jsem využila při tvorbě svého projektu.

DBMS zabezpečuje vztah mezi abstrakcemi pohledů na data ve 3. úrovních:

- *fyzickou úroveň* - úroveň jak jsou data fyzicky uložena v paměti (disk, páska, RAM);
- *logickou úroveň* - úroveň jak jsou data reprezentována, tj. co vlastně představují, jak jsou data logicky strukturována;
- *náhledovou úroveň* - úroveň která, skrývá detaily např. typ dat, přístupové mechanismy (oprávnění).

Musí zabezpečit tyto funkce:

- vstup/výstup dat do databáze;
- aktualizaci dat;
- vyhledávání dat;
- třídění dat;
- množinové operace s relacemi (jako součin, průnik, doplněk průniku atd.);
- matematické operace nad daty (jako např. vytváření součtů nad polem).

Charakteristickými vlastnostmi současných DBMS jsou:

- *transakční zpracování* - každé zpracování dat je realizováno jako posloupnost operací, převádějící databázi z jednoho konzistentního stavu do druhého;
- *zotavení z chyb* - pokud se nepodaří transakci úspěšně ukončit, musí být všechny její efekty odstraněny;
- *řízení souběžného přístupu* více uživatelů;
- *ochrana dat* - před neoprávněným přístupem, změnou nebo poškozením formou speciálních uživatelských práv zpravidla chráněných heslem (autorizace);
- *variabilní architektura úschovy dat*:
 - *centralizované* uložení dat - data jsou fyzicky uložena na jednom počítači;
 - *distribuované* uložení dat - data jsou fyzicky uložena na více počítačích;
- *zpracování požadavků v reálném čase* - umožňuje ovlivňovat procesy běžící v reálném čase (lze definovat čas odezvy);
- *práce s multimediálními daty* - text, zvuk, obraz, video apod;

V souvislosti se DBMS se rozlišují dva typy jazyků, které zpracovávají:

- **Jazyk pro definici dat** (DDL - Data Definition Language) - pro definici fyzické úschovy uživatelských dat. Jazyk definuje data, integritní omezení, způsob úschovy relací, údaje o právech přístupu k datům.
- **Jazyk pro manipulaci dat** (DML - Data Manipulation Language)- vybírá a manipuluje s daty dle zadaných kritérií. Ten může obsahovat dvě třídy jazyků a to:
 - *procedurální* - určují jakým způsobem a jaká data z databáze budou získány;
 - *neprocedurální* - specifikují jaká data požadujeme ale ne jakým způsobem je získáme. Narozdíl od jazyků typu C, Pascal atd. nepopisují, jak se má úkol provést, ale pouze výsledek kterého se má dosáhnout.

Každý SQL-server má svůj vlastní protokol, pomocí kterého s ním může klient komunikovat. Pokud chceme, aby klient komunikoval s různými servery, musí podporovat více protokolů. A proto na platformě Windows vzniklo rozhraní ODBC. To slouží jako prostředník mezi klientskou aplikací a databázovým serverem. Rozhraní ODBC se volá jednotně a jeho ovladač pak požadavek předá databázovému serveru pomocí správného protokolu. Výhodou tohoto přístupu tedy je, že stejným způsobem můžeme přistupovat k libovolné databázi. ODBC bylo původně vyvinuto pro platformu Windows, ale dnes jeho implementace existuje i pro všechny významnější verze operačního systému Unix.

Skripty PHP umožňují vedle přístupu přes rozhraní ODBC také nativní přístup k datům. Např. pro připojení k databázi se používá funkce `xxx_connect`, kde `xxx` určuje název databázové platformy (např. `MySQL_connect`) [2].

MySQL je velmi rychlý a robustní databázový systém, označuje se jako multi-uživatelský a multi-threadový server. Databáze umožňuje efektivně ukládat, hledat, třídit a získávat data. MySQL server se stará o to, aby k databázi mohlo přistupovat více uživatelů zároveň a zajišťuje, aby se jednalo pouze o autorizované uživatele.

MySQL je veřejnosti přístupná od roku 1996, ale její kořeny sahají až do roku 1979. Je možné ho používat na různých platformách a stejně jako u PHP i u MySQL můžeme upravovat zdrojové kódy.

3.5.4 Jazyk SQL

Stal se nejpoužívanějším jazykem pro přístup k relačním databázovým systémům. SQL jazyk je strukturovaný, neprocedurální, deklarativní jazyk. Obsahuje nejen DDL a DML, ale i další příkazy, např. pro udílení práv uživatelům.

Historie SQL sahá až do sedmdesátých let, kdy se vyvinul z jazyka SEQUEL, který byl vyvinut v laboratořích IBM. Existuje ANSI standard pro SQL, který musí obsahovat každý databázový systém.

Samotný jazyk SQL může být použit jako:

- dotazovací jazyk pro relační databáze;
- část hostitelského jazyka pro vývoj databázových aplikací;
- prostředek komunikace mezi různými zdroji dat.

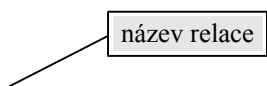
Datové typy použitelné v SQL se příliš neliší od typů, které se užívají v jiných jazycích. Existují typy pro celá i reálná čísla, pro textové řetězce, pro datum a čas apod. V následující tabulce uvádím přehled nejpoužívanějších typů. [2]

Tabulka 2 Přehled datových typů SQL

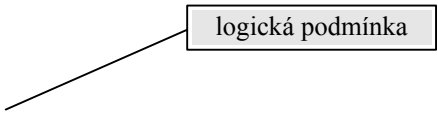
Typ	Popis
int	celé číslo v rozsahu od -2 147 483 648 do 2 147 483 648
tinyint	celé číslo v rozsahu od 0 do 255
smallint	celé číslo v rozsahu od -32 768 do 32 767
float	číslo s pohyblivou řádovou čárkou
char (n)	textový řetězec o délce n (maximálně však 255 znaků)
varchar (n)	textový řetězec o délce maximální délce n (maximálně však 255 znaků)
decimal (p,d)	desetinné číslo s p platnými číslicemi a s d desítnými místy
money	peněžní částka (můžeme ho nahradit např. decimal (12,2))
datetime	údaj o čase a datu ve formátu RRR-MM-DD HH:MM:SS
time	údaj o čase HH:MM:SS
date	údaj o datu ve formátu RRR-MM-DD
timestamp	1970-01-01 00:00:00 - kolem roku 2037, časová známka, vhodná pro zaznamenávání transakcí
blob, image	Pro uložení dlouhých binárních dat (každý server používá vlastní typ)

Základním a nejpoužívanějším příkazem jazyka SQL je příkaz *SELECT*, který slouží k zápisu požadavků na výběr dat z databáze. Umožňuje zapsat operace relační algebry - kartézský součin, sjednocení, rozdíl, projekci relace na množinu atributů a selekci na základě logické podmínky *WHERE*. Jazyk SQL je tedy relačně úplný.

Příklady s příkazem SELECT:

 `SELECT * FROM `uzivatele`;`

Výběr všech entit z relace.

 `SELECT jmeno FROM `uzivatele` WHERE id==1;`

Výběr uživatelského jména z tabulky *uzivatele*, kde identifikační číslo uživatele je rovno číslu jedna.

Složitější dotaz v jazyku SQL lze obvykle vyjádřit více způsoby v závislosti na způsobu uvažování programátora a případně na snaze o optimalizaci dotazu uvnitř DBMS. Cílem optimalizace dotazu je obvykle co nejefektivnější vyhodnocení požadované operace z hlediska času a potřebného diskového prostoru.

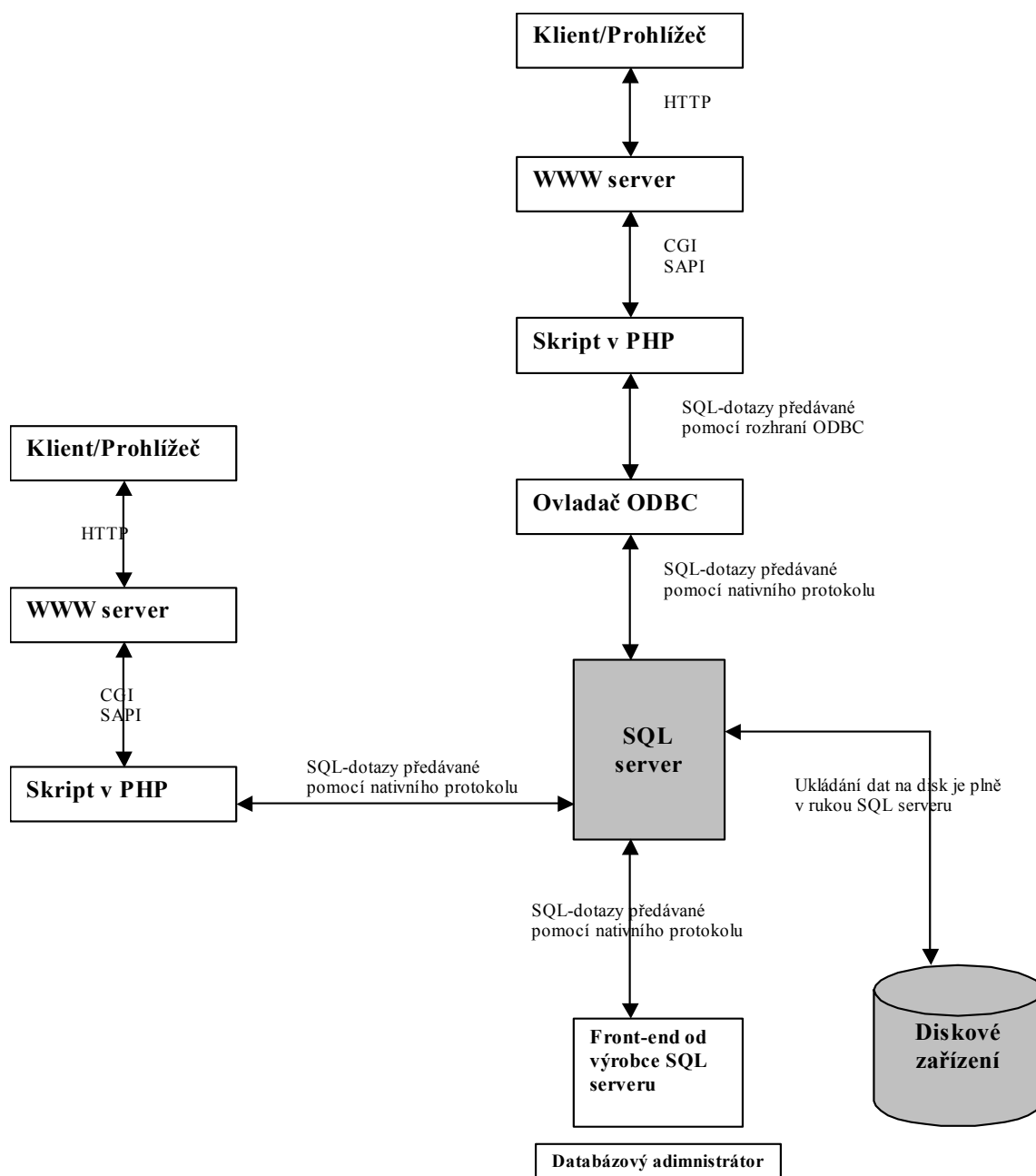
Moderní databázové systémy jsou vybaveny možností automaticky vyvolat SQL příkaz nebo celou proceduru při vzniku určité události *trigger* - *spoušt*. Takovou aktivující událostí může být například přidání, vymazání a změna entity v relaci.

SQL je používán v databázových systémech jako MySQL, Oracle, PostgreSQL, Sybase a mezi jinými také Microsoft SQL Server. V rámci DBMS existují i další manipulační jazyky jako např. *4GL* (jazyk čtvrté generace). Základní vlastností těchto jazyků je výrazně zjednodušují tvorbu uživatelských rozhraní v aplikacích. Pro tyto jazyky neexistuje žádný standard a jednotlivé databázové systémy používají vlastní jazyk.

3.5.5 Architektura spolupráce PHP a MySQL s webovým serverem

Následující obrázek (*Obr. 8*) ukazuje tři způsoby komunikace s SQL-serverem. Aplikace mohou se serverem komunikovat přes rozhraní ODBC, nebo přímo pomocí nativního protokolu [2].

- a) uživatel ve svém prohlížeči provede akci, jejímž výsledkem bude požadavek na určitou stránku;
- b) prohlížeč se spojí s WWW-serverem a pošle přes HTTP požadavek;
- c) WWW-server spustí interpret PHP na požadovaný skript a předá mu parametry zaslané uživatelem;
- d) PHP začne provádět skript. Uvnitř nalezne příkaz k připojení k databázi a provedení dotazu. PHP si vytvoří spojení s MySQL a odešle požadovaný dotaz;
- e) MySQL přijme dotaz, zpracuje a pošle zpět PHP;
- f) PHP dokončí provádění skriptu a naformátované HTML vrátí webovému serveru;
- g) ten pak zasílá zpět odpověď klientovi;
- h) na obrázku je zachycena front-end aplikace, která slouží ke správě SQL-serveru (vytváření databází, správa uživatelských účtů, pravidelné zálohy dat apod.). Pro tyto účely se ke každému serveru dodává speciální aplikace.



Obr. 8 Spolupráce SQL-serveru s aplikacemi

3.6 Instalace serveru a PHP

Na svém počítači si můžeme snadno vytvořit serverové prostředí pro provoz jednoduchých webových aplikací. Instalaci WWW-serveru (Apache), PHP a MySQL serveru a jejich následnou konfiguraci bych začátečníkům přece jen nedoporučovala. Nejen pro takové uživatele vzniklo několik balíčků, které instalaci serverů a podpory PHP plně automatizují [9]. Já sama jsem vyzkoušela a vřele doporučuji instalační balíček *EasyPHP*, který je zdarma. Balíček najdete nejen na přiloženém CD, ale je volně dostupný například na internetové adrese: <http://sourceforge.net/projects/quakeasyphp>.

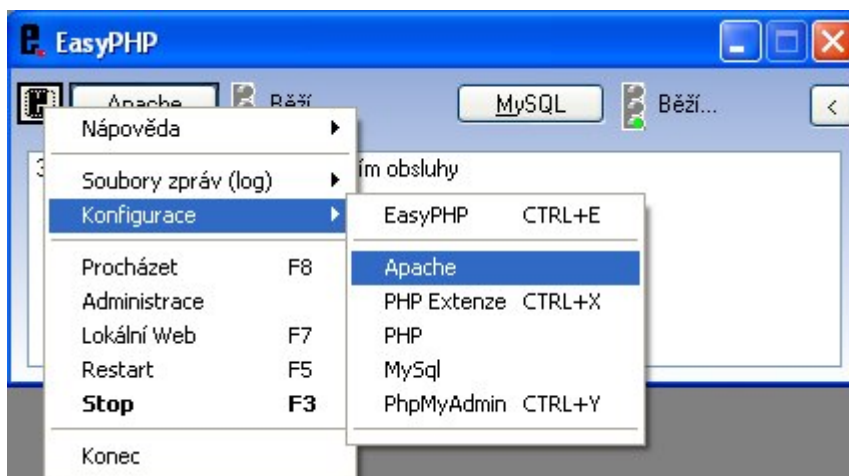
Po spuštění instalačního souboru, po odsouhlasení licence a zvolení instalační složky dojde ke zkopírování souborů do počítače. EasyPHP obsahuje přehledný administrační nástroj, kterým můžeme ovládat webový i databázový server.



Obr. 9 Administrační nástroj EasyPHP

Nástroj obsahuje dvě tlačítka, která po kliknutí zobrazí nabídku s možnostmi pro spuštění, restart a ukončení serverů Apache a MySQL. O jejich stavu informuje semafor. Hlavní nabídka se zobrazí po klepnutí na tlačítko s logem programu. Nabídka *Konfigurace* umožňuje nastavit spuštění programu ihned po startu operačního systému, měnit nastavení WWW nebo databázového serveru, PHP. I bez nutnosti přepisování konfiguračních souborů nebo jiného nastavení můžeme s webovým serverem ihned pracovat. Složku, do které můžeme zkopírovat své projekty, nalezneme v hlavní složce EasyPHP jako *www*. V případě výchozího instalačního nastavení to bude *C:\Program Files\EasyPHP1-8\www*. Při volbě *Konfigurace/Apache* se otevře textový soubor s nastavením serveru. Text je od nastavení oddělen značkou *#*. Pokud například vyhledáme položku *DocumentRoot* "*#{path}/www*", tak ta značí cestu ke složce, kde má být webová prezentace uložena. Proměnná *path* zastupuje cestu k instalaci EasyPHP. Pokud chceme, stránky spouštět z jiné

složky, a změníme cestu např. na “ $\{\text{path}\}/\text{www}/\text{diplomka}$ ”. Webová prezentace pak z adresy <http://localhost> bude zobrazovat sčrtipy, uložené ve složce diplomka.

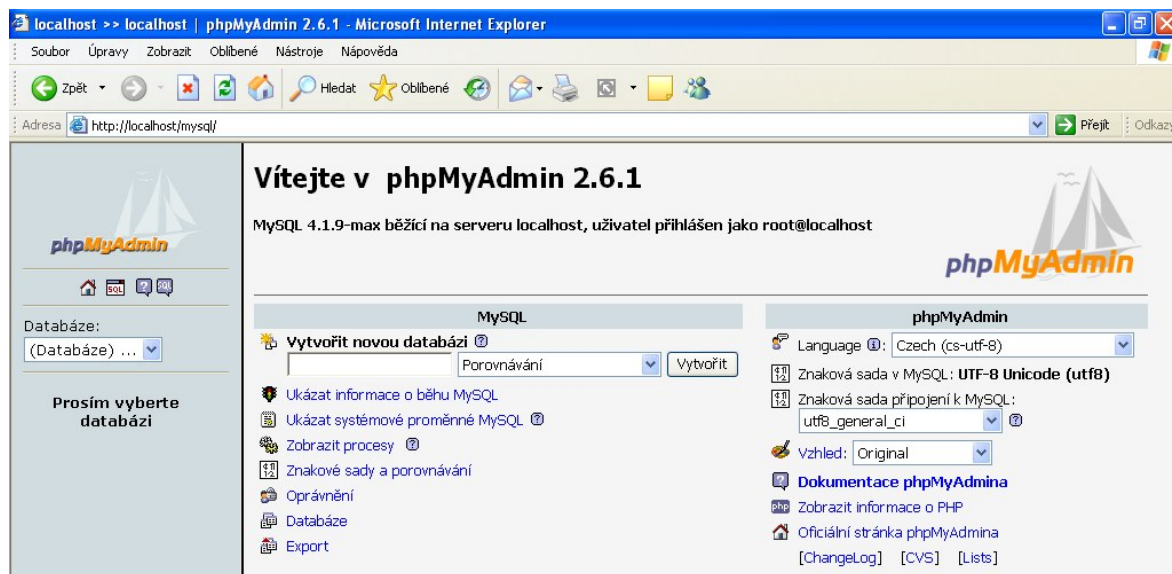


Obr. 10 Nabídka konfigurace programu EasyPHP

Plánujeme-li použití některých PHP modulů, které nemusí být ve výchozím stavu aktivovány, můžeme je povolit pod položkou *PHP Extenze*.

Jakmile budeme chtít zobrazit obsah svých stránek na serveru a naše skripty budou uloženy v adresáři `www` (programu EasyPHP), spustíme EasyPHP a do prohlížeče napíšeme <http://localhost/> nebo <http://127.0.0.1/> s názvem naší stránky.

Na adrese <http://localhost/mysql/> (<http://127.0.0.1/mysql/>) nalezneme nástroj pro správu databází – *phpMyAdmin*, jenž je lokalizován do čeština. S tímto nástrojem se máme možnost setkat i u poskytovatelů webového prostoru.



Obr. 11 phpMyAdmin – nástroj pro správu databází

Pokud chceme zjistit, jakým způsobem je Apache spojen s modulem PHP, najdeme v konfiguračním souboru Apache řádek `LoadModule php4_module "${path}/php/php4apache.dll"`. Na tomto místě můžeme změnit modul na novější verzi PHP. Nové nastavení se projeví po uložení souboru a restartu serveru Apache.

4 SOFTWAREVÁ REALIZACE PROJEKTU

4.1 Charakteristika aplikace

Jedná se o víceuživatelský systém, proto se prostředí mění na základě práv přihlášeného účastníka. Strukturu i vzhled aplikace jsem se snažila navrhnout pro co největší přehlednost a jednoduchost, aby se v ní i nově registrovaný uživatel okamžitě zorientoval a načítání stránek netrvalo při pomalém připojení příliš dlouho.

Úvodní stránka obsahuje kromě formuláře k přihlášení a novou registraci také seznam testů, které jsou po vyplnění ihned vyhodnoceny. Uživatel si tak okamžitě může ověřit své znalosti z dané oblasti. Obsah testů jsem čerpala z počítačových časopisů Computer [6], [7] a [8].

Po přihlášení se dostáváme do jádra aplikace a prostředí se mění na základě toho, zda je uživatel přihlášen s právy učitele nebo zda do systému vstupuje student.

VZDĚLÁVACÍ ON-LINE SYSTÉM

Univerzita Tomáše Bati ve Zlíně
Univerzitní institut

Přihlášení do systému:

jméno:
heslo:
přihlásit

registre

Vítejte na stránkách věnujících se on-line vzdělávání.

Na úvod jsme pro Vás připravili několik testů, které ověřují Vaše znalosti. A pokud se chcete dozvědět další zajímavé informace, je to snadné. Stačí se k nám zaregistrovat.

TESTY	:: třídít podle názvu	:: podle velikosti	:: podle data
test_mobil_a_pc.html umíte spojit počítače s mobily?	9.03 kB	29.05.2005	
test_site.html vznáte se v počítačových sítích?	8.95 kB	29.05.2005	
test_tisk.html rozumíte tiskovým technologiím?	9.08 kB	30.05.2005	
test_video.html vznáte se ve videu?	8.87 kB	27.05.2005	

Obr. 12 Úvodní stránka aplikace

Uživatelská data jsou rozdělena a uložena ve formě tabulek v MySQL databázi, přičemž většina z nich je vázána na konkrétního registrovaného a následně přihlášeného uživatele. Účastníci systému jsou rozděleni na dvě skupiny, z nichž každá má právo vstupu do určitých částí aplikace. Podrobnější popis jednotlivých částí bude uveden dále.

Projekt je umístěn na bezplatném webhostingu, a to na adrese <http://diplomka.euweb.cz/>. Tento server poskytuje podporu PHP i dostatečný prostor pro MySQL databázi.

4.2 Grafický vzhled aplikace

Design každého webu je prestižní záležitostí a také na něm závisí, zda se na něj budou v budoucnu uživatelé rádi vracet. Mnoho stránek na Internetu je možnou inspirací a samozřejmě záleží na vkusu každého programátora. Při tvorbě jsem se snažila, aby byl výsledný web co nejpřehlednější a aby se alespoň trochu přiblížil ke stylu stránek Univerzitního institutu. Následující tabulky znázorňují logické rozdělení webových stránek. Hlavička a menu v levé části zůstává vždy stejné, obsah stránek se mění na základě volaných skriptů. Pro editaci vzhledu bylo použito CSS stylů, které jsem zmínila v kapitole 3.2.

Tabulka 3 Logické členění stránek HTML nepřihlášených uživatelů

hlavička	
menu	obsah stránky

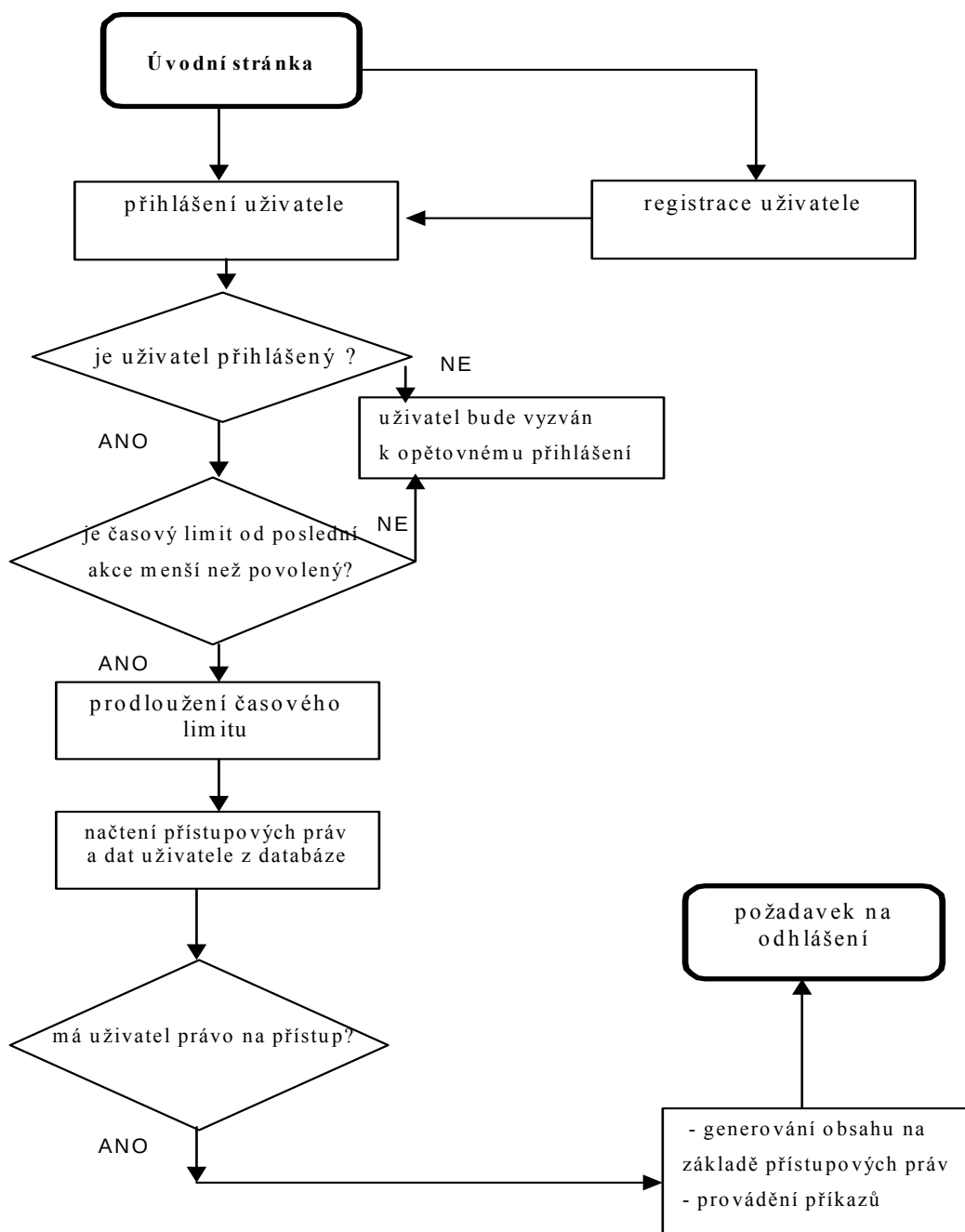
Tabulka 4 Logické rozdělení stránky HTML, pokud je uživatel přihlášen

hlavička	
obsah stránky	seznam on-line uživatelů

4.3 Jádru systému

Základem systému je užívání přístupových práv k jednotlivým skriptům. To je zaručeno vložení autorizační části do každého skriptu. Celý proces si lze představit jako cyklus, který je schematicky zobrazen na obrázku (Obr. 13).

Veškerou činnost spojenou s autorizací přístupu vykonává skript *checklogin.php*.



Obr. 13 Schéma činnosti jádra systému

4.3.1 Přihlášení do systému

Přihlášení do systému obstarává script *login.php*, o odhlašování se stará soubor *logout.php*. Jejich hlavní funkcí je autorizace uživatele.

Autorizace je zajištěna pomocí sessions. Tento mechanismus je založen na proměnných, které si předává automaticky prohlížeč s webovým serverem, buď na bázi cookies nebo pomocí parametru URL. Programátor zajistí start mechanismu session funkcí *session_start* a potom už jen ukládá potřebné proměnné a pracuje s nimi. Tímto máme zajištěno, že dokud uživatel nezavře aktuální okno, neodhlásí se anebo session není zrušena programátorem, je uživatel pomocí unikátního čísla, které přiděluje server autorizován.

Postup při přihlašování

V případě tohoto systému je postup při přihlašování následující:

1. uživatel, který se chce přihlásit má vytvořeno session spojení;
2. při přihlášení je odesláno jeho uživatelské jméno a heslo přihlašovacímu skriptu;
3. jméno a heslo je porovnáno s údaji v databázi (tabulka *uzivatele*);
4. pokud login a heslo souhlasí, je do session proměnné *auth* vloženo *id* (tedy unikátní číslo uživatele z databáze);
5. uživatel je přesměrován na požadovanou stránku.

V další části autorizace je kontrolována existence proměnné *auth*, a pokud existuje, jsou data o uživateli načtena z příslušného řádku tabulky *uzivatele* (Tabulka 5).

Odhlašování je proces, kdy je na začátku ověřena existence proměnné *auth* a v případě, že existuje je session zrušena funkcí *session_destroy*.

Může se stát, že se uživatel správně neodhlásí ze systému, ale pouze uzavře prohlížeč. V případě, že tak učiní, existuje možnost neoprávněného přístupu do jeho osobního nastavení. Tento problém je vyřešen podobně jako ve většině aplikací na Internetu, a to časovým limitem. V případě že po určenou dobu nevykazuje uživatel v systému známky činnosti, je automaticky odhlášen. Tento problém řeší databázová tabulka *ONLINE*, která zaznamenává přihlášené uživatele a čas jejich poslední akce (např. načtení stránky, chatování).

4.4 Návrh databázového systému

Návrh databázového systému se opírá o datovou analýzu, která se uplatňuje shora - dolů, tzn. postupně upřesňuje datovou strukturu databázového systému formou vytváření hierarchie popisů schémat lišících se mírou abstrakce:

- *konceptuální schéma* - vyjádření konceptuálního modelu,
- *databázové schéma* - vyjádření databázového modelu,
- *realizace databázového schématu* - pomocí konkrétního DBMS resp. pomocí jeho implementovaného jazyku zabezpečující interface do DBMS,
- *fyzická implementace databázového schématu* - např. indexy pro rychlejší přístup k datům, atd.

4.4.1 Konceptuální schéma

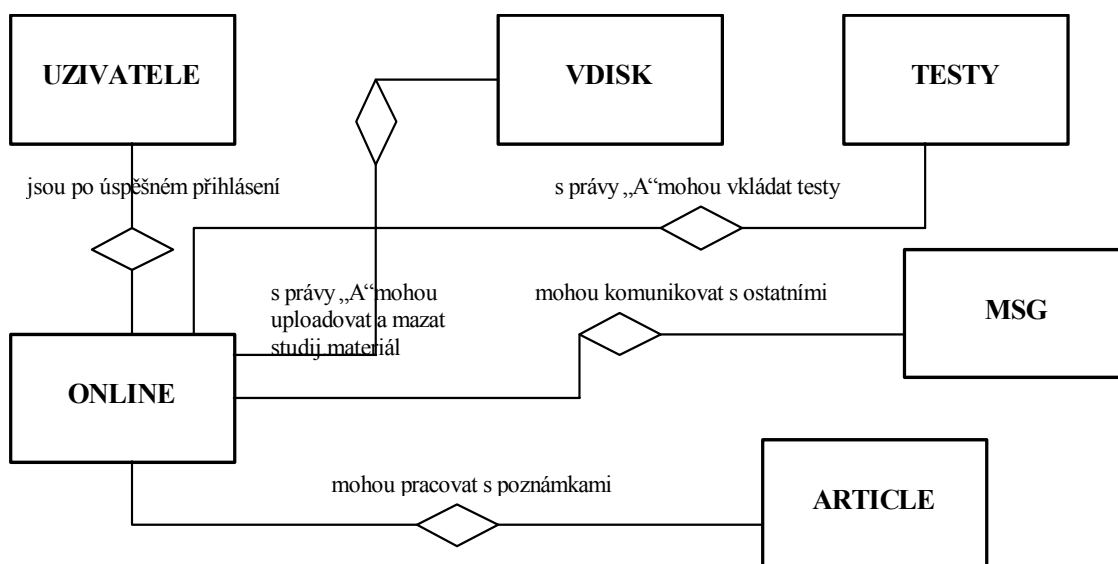
Konceptuální schéma popisuje data v databázi na abstraktní úrovni nezávisle na jejich fyzickém uložení, přičemž má co nejvýstižněji zachycovat pohled člověka na datové údaje. Konceptuální schéma se v další etapě návrhu databázového systému konkretizuje do podoby databázového schématu.

Při mé práci jsem použila konceptuální E-R model (Entity Relationship model), který dnes tvoří v podstatě světový standard. ERM obsahuje:

- *jednotlivé entity* (Entity Types), což jsou jednoznačně profilované a nezávisle existující údaje,
- *relace* (Relationship Types), popisují vztahy mezi entitami (relace zde vystupuje ve významu spojení).

V rámci zobrazení grafického modelu lze použít jakýkoliv způsob grafického zobrazení (a je běžné, že v různé literatuře se používá rozdílného zobrazení), za předpokladu, že dostatečně objasníme použité symboly. V standardizovaném grafickém modelu se pak entity nejčastěji vyjadřují obdélníky, relace kosočtverci.

Na následujícím obrázku je znázorněno relační spojení všech objektů aplikace



Obr. 14 ER – diagram databázových relací

4.4.2 Převod konceptuálního schéma do databázového modelu

Převod konceptuálního modelu do databázového schématu v podobě návrhu jednotlivých relací lze provedeme takto:

- entitám přiřadíme tabulky, název entity přeneseme do názvu tabulky,
- tabulky budou obsahovat sloupce (atributy), které odpovídají jednotlivým atributům entity,
- relace se realizují pomocí vlastních a cizích klíčů.

Záznam pak odpovídá řádku tabulky. V RDM se neprovádí rozlišení relace a relačního spojení, rozlišení se provádí na úrovni algoritmů v jazyce pro manipulaci s daty.

Po základním převodu konceptuálního modelu do databázového modelu v rámci RDM je vhodné provést *normalizaci relací*. Normalizace má za úkol odstranit duplicitu ukládání dat v relacích a eventuálně zlepšit efektivitu manipulace s relacemi.

V mé aplikaci je propojeno a navzájem spolupracuje šest tabulek. Údaje o registrovaných uživateli se nachází v tabulce *UZIVATELE*.

Tabulka 5 Struktura tabulky UZIVATELE

název	datový typ	funkce
id	int (11)	unikátní číslo uživatele, primární klíč
jmeno	varchar (15)	uživatelské jméno (login)
celejmeno	varchar (50)	jméno a příjmení uživatele
heslo	varchar (50)	heslo zašifrované algoritmem MD5
prava	char (1)	práva uživatele, default „U“
email	varchar (30)	e-mailová adresa uživatele

Defaultně je atribut „prava“ nastaven na hodnotu U, což znamená, že přihlášený uživatel má právo studenta. Pokud bude chtít v systému pracovat jako vyučující (tomu je umožněno vkládat a mazat studijní materiály), musí po zaregistrování kontaktovat administrátora, který ověří, zda se opravdu jedná o učitele a následně mu zajistí požadovaný přístup do systému.

Tabulka *ONLINE* zaznamenává přihlášené uživatele a čas jejich poslední akce.

Tabulka 6 Struktura tabulky ONLINE

název	datový typ	význam
uz_id	int (11)	číslo registrovaného uživatele, cizí klíč
cas	int (15)	čas poslední provedené akce
sent_on	timestamp	aktuální čas (důležité pro chat)

Proměnná *cas* je typu *timestamp*, značí tzv. časové razítko - počet sekund od 1.1. 1970, položka *uz_id* obsahuje stejné číslo, jaké má uživatel v tabulce *UZIVATELE*. Při každé akci je volán skript s autorizací, který vyhodnocuje hodnotu času z tabulky *ONLINE* pro danou session (číslo uživatele) a kontroluje časový rozdíl od aktuálního času. V případě, že je menší než nastavený, je hodnota času v tomto řádku aktualizována právě tímto aktuálním časem. V případě, že časový limit nečinnosti vypršel, je uživatel odhlášen a upozorněn chybovou hláškou pro opětovné přihlášení. Pokud je uživatel odhlášen, je jeho *uz_id* (celý řádek) z tabulky *ONLINE* odstraněn.

Každý přihlášený uživatel může na svém účtu vkládat, mazat a upravovat poznámky. Ty se po vytvoření vkládají do tabulky *ARTICLE* (Tabulka 7).

Tabulka 7 Struktura tabulky ARTICLE

název	datový typ	funkce
id_article	int (11)	unikátní číslo poznámky, primární klíč
title	varchar (100)	předmět poznámky
message	text	text poznámky
uzivatele	int (11)	unikátní číslo uživatele, cizí klíč
publish_dat	int (11)	datum vložení poznámky

Tabulka MSG slouží k dočasnému ukládání zpráv při chatování uživatelů. Po 24 hodinách se záznamy v tabulce automaticky odstraní.

Tabulka 8 Struktura tabulky MSG

název	datový typ	funkce
id	int (11)	unikátní číslo zprávy, primární klíč
uz_id	int (11)	identifikační číslo on-line uživatele, cizí klíč
msg	text	text zprávy
color	char (1)	určuje barvu textu zprávy
sent_on	timestamp	čas vložení textové zprávy

Do tabulky TESTY a tabulky VDISK mají právo vkládat informace o souborech pouze vyučující.

Tabulka 9 Struktura tabulky TESTY

název	datový typ	funkce
testname	varchar (50)	název testu, primární klíč
comment	text	stručný komentář k testu
filesize	int(20)	velikost testu
filetime	datetime	datum vložení testu

Tabulka 10 Struktura tabulky VDISK

název	datový typ	funkce
filename	varchar (50)	název studijního materiálu, primární klíč
comment	text	stručný komentář souboru
filesize	int(20)	velikost souboru
dloaded	int(11)	počet stahování souboru
filetime	datetime	datum vložení testu

4.5 Funkce systému na základě uživatelských práv

Aplikace je navržena pro výuku, ale zároveň musí obsahovat i funkce k úpravě, konfiguraci a administraci. Je zde uplatněna hierarchie práv – *student*, *učitel*, *správce systému*. Každé právo v sobě zahrnuje určité možnosti.

Běžný uživatel

Jedná se o nepřihlášeného uživatele, který zavítá na úvodní stránku. Získá základní informaci o tom, komu by měl vzdělávací systém sloužit a má možnost otestovat své znalosti v připravených testech.

Student

Pokud chceme systém využívat, musíme se nejdříve zaregistrovat. Pokud tak učiníme, je nám automaticky přiděleno právo v systému vkládat osobní poznámky, komunikovat s přihlášenými uživateli (chatovat) a studovat z materiálů, které jsou v systému uloženy.

Učitel

Toto právo se týká vyučujících. Je jim umožněno vykonávat nejen to, co mohou uživatelé s právy – *student*, ale dále mohou uploadovat studijní materiály, mazat je a jinak modifikovat. Pokud vkládají na server soubor s názvem, kdy počáteční sekvence znaků obsahuje řetězec `test_`, soubor se po uploadu na server automaticky zobrazí na úvodní stránce aplikace.

Správce systému

Toto právo umožňuje rozsáhlejší úpravy v systému. Organizace struktury webu, úpravy práv uživatelů, zálohování dat, přístup ke scriptům atd.

4.6 Popis ostatních částí vzdělávacího on-line systému

V příloze je uveden seznam všech skriptů se stručným popisem, které systém obsahuje. Následující kapitoly se budou věnovat každé části systému samostatně.

4.6.1 Registrace nového uživatele

Zaregistrovat se může jen nepřihlášený uživatel. Registrační proces se skládá ze tří kroků:

1. Návštěvník vyplní své uživatelské jméno, pod kterým se bude do systému hlásit a svůj e-mail. Pokud jméno není registrováno, je na jeho e-mail poslán *registrační potvrzovací kód*, což je náhodné číslo zašifrované metodou MD5.

Registrace nového uživatele:

1. krok v registraci

Pro úspěšné zaregistrování zvolte uživatelské jméno a e-mail, kde Vám bude poslán ověřovací registrační kód.

login:

e-mail:

Obr. 15 První krok v registraci

2. Po obdržení e-mailu a opsání registračního kódu, uživatel vyplní své jméno a příjmení a zvolí si heslo. Pokud je registrační kód správný, heslo a jeho potvrzení jsou shodné, je uživateli odeslán e-mail s úspěšným dokončením registrace. Ten pro kontrolu obsahuje přihlašovací jméno a heslo uživatele. Poté se může okamžitě přihlásit.

Registrace nového uživatele:

2. krok

Login byl potvrzen. Nyní pokračujte zadáním registračního kódu a hesla.

Právě Vám byl zaslán e-mail s registračním kódem. Opište kód z e-mailu a potom si zvolte heslo, pro kontrolu zadejte heslo 2x.

registrační kód: 3f31b4a09973690c3ad2c86de5ab

heslo:

heslo znovu:

jméno a příjmení: Jiří Novák

pokračovat ->

Obr. 16 Druhý krok v registraci

3. Každému uživateli je automaticky přiděleno právo studenta. Může tedy studovat z připraveného materiálu, vkládat si k němu poznámky nebo komunikovat s ostatními přihlášenými pomocí chatu. Pokud chceme využívat práva studenta, musíme kontaktovat správce systému.

4.6.2 Testy

Titulní stránka aplikace obsahuje řadu testů, které jsou přístupné každému návštěvníkovi úvodní stránky. Testy do systému vkládají učitelé. Jako ukázkou jsem zvolila testy z oblasti informatiky. Po jeho vyplnění je zapomocí Javascriptu zajištěno vyhodnocení. Špatně zodpovězené otázky jsou vyznačeny červeně, jejich správná varianta je označena tučně. Obsahovou náplň testů, varianty odpovědí a jejich celkové vyhodnocování jsem čerpala z počítačových časopisů Computer [6], [7] a [8].

```
function ukaz_hodnoceni() {  
    var pocet_bodu = 0;  
    for(var i=0; i<10; i++){  
        pocet_bodu = pocet_bodu + parseInt(odpovedi[i]);  
        if (odpovedi[i] == 0) {  
            document.getElementById('o' + (i+1)).style.color = 'red';  
        }  
        else {  
            document.getElementById('o' + (i+1)).style.color = 'black';  
        }  
        document.getElementById('dobre' + (i+1)).style.color = 'green';  
        document.getElementById('dobre' + (i+1)).style.fontWeight = 'bolder';  
    }  
    var hlaska = "Váš bodový zisk v tomto testu je "+pocet_bodu+" z 10.\n";  
    if (pocet_bodu <10){  
        hlaska = hlaska + "Chybné odpovědi jsou označeny červeně.\n\n";  
    }  
    if (pocet_bodu <= 3){  
        alert(hlaska+vysledek1);  
    }  
    else if (pocet_bodu >= 8){  
        alert(hlaska+vysledek3);  
    }  
    else {  
        alert(hlaska+vysledek2);  
    }  
}  
//-->  
</script>
```

Obr. 17 Zdrojový kód funkce (v editoru PSPad), která zajišťuje vyhodnocení testu

7. Tzv. IP adresa se skládá ze čtyř:

- ☐ a) trojčiferných čísel
- ☒ b) osmibitových čísel
- ☐ c) osmičkových čísel
- ☐ d) prvočísel od 0 do 255

8. Maximální přenosová rychlost sítě 802.11b je:

- ☐ a) 1 Mb/s
- ☒ b) 11 Mb/s
- ☐ c) 100 Mb/s
- ☐ d) 1 024 Mb/s

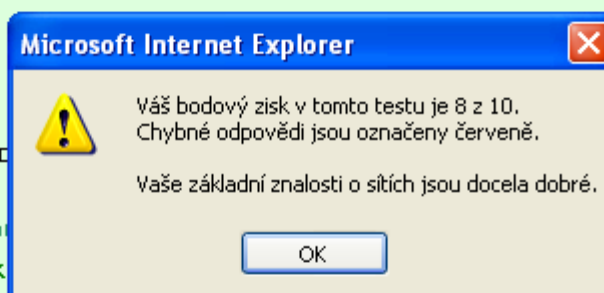
9. K propojení dvou počítačů je potřeba:

- ☐ a) dvou kabelů, dvou počítačů a dvou sítí
- ☒ b) dvou síťových karet, dvou počítačů a dvou sítí
- ☐ c) tří síťových karet, dvou kabelů a specializovaného serveru
- ☐ d) dvou dvojic síťových karet a dvou kabelů zapojených křížem

10. Výrazům hub/switch/router odpovídají české výrazy:

- ☐ a) rozdvojka/vypínač/brána
- ☐ b) hříbek/přepínač/brána
- ☒ c) rozbočovač/přepínač/směrovač
- ☐ d) rozdělovač/spínač/směrovač

Hodnocení

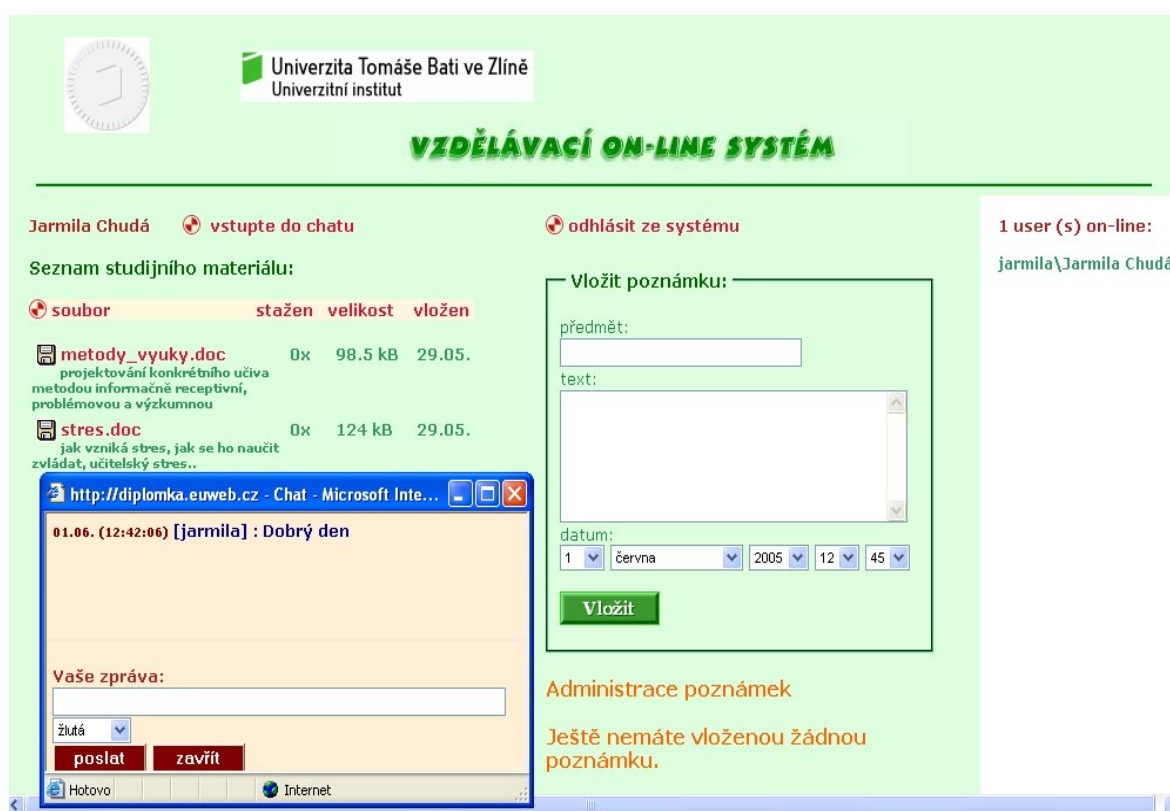


nejméně:

Obr. 18 Ukázka testu z oblasti počítačových sítí

4.6.3 Uživatelské rozhraní po vstupu do systému

Uživatel s právy *student* vstupuje do uživatelského rozhraní, které je zobrazeno na obrázku (Obr. 19). Struktura stránky je rozdělena do tří částí. Vlevo systém nabízí seznam studijního materiálu, uprostřed formulář, pomocí kterého může vkládat, mazat a upravovat osobní poznámky. V pravé části systému je vypsán seznam právě přihlášených uživatelů. Těsně pod hlavičkou dokumentu jsou umístěné dva odkazy, jeden slouží ke vstupu do chatu a druhý k řádnému odhlášení ze systému.



Obr. 19 Uživatelské rozhraní s právy student

Jak už bylo zmíněno, struktura systému pro vyučující se o něco málo liší. Jsou zde zabudovány prvky pro odstranění a upload studijního materiálu a testů. Ostatní funkce systému jsou totožné jako v předcházejícím případě. Se studijními dokumenty mohou pracovat pouze přihlášení uživatelé. Pokud je soubor označen s názvem test, je zobrazován na úvodní stránce systému.

J. Chudá [vstupte do chatu](#) [odhlásit ze systému](#)

Seznam studijního materiálu:

soubor	stažen	velikost	vložen
metody_vyuky.doc projektování konkrétního učiva metodou informačně receptivní, problémovou a výzkumnou	0x	98.5 kB	29.05.
stres.doc jak vzniká stres, jak se ho naučit zvládat, učitelský stres..	0x	124 kB	29.05.

Vložit soubor na server:

Soubor: [Procházet...](#)

Popis:

[Upload](#)

Vložit poznámku:

předmět:

text:

datum: 1 června 2005 12 33

[Vložit](#)

Poznámka byla přidána.

Administrace poznámek

metody výuky
do 12. 6. - nezapomenout uložit slíbený dokument
J. Chudá /1. 6. 2005 12:32/
[Upravit](#) [Smazat](#)

1 user (s) on-line:
chuda\J. Chudá

Obr. 20 Uživatelské rozhraní s právy učitel

ZÁVĚR

Cílem mé diplomové práce bylo usnadnění přístupu k informacím studentům distanční formy studia. Byl kladen důraz na jednoduchou a přehlednou strukturu zobrazení. Cílem bylo, aby studenti měli přístup nejen ke studijním textům a jiným dokumentům, ale aby si při jejich prostudování mohli na jednom místě vpisovat poznámky, postřehy, nápady, připomínky apod. Při této formě studia je velice důležitá zpětná vazba, proto jedním z požadavků na systém byla možnost komunikace mezi on-line uživateli. Přihlášení účastníci mohou využívat při vzájemné komunikaci jednoduchý chat. Do budoucna je důležité, aby systém obsahoval kvalitní studijní texty vytvořené speciálně pro distanční nebo on-line studium.

V teoretické části jsou popsány všechny technologie, kterých navržený systém využívá. Jedná se o jazyk HTML, Javascript, kaskádové styly, scriptovací jazyk PHP, databázový systém MySQL s jazykem SQL. V práci je popsána i instalace a možná konfigurace freewarového programu EasyPHP. Jedná se o balík, který obsahuje webový server Apache, databázový server a nástroje pro podporu PHP.

V praktické části je rozebrána činnost webových stránek, jsou zde podrobně rozepsány jednotlivé části aplikace, včetně použitých datových struktur. Projekt webových stránek je navržen tak, aby splňoval požadavky na jednoduchou aktualizaci, ale zároveň zajistil bezpečnost přístupu a bylo možné kontrolovat data zadaná uživateli systému.

Díky přístupnosti přes celosvětovou síť Internet je systém všem dostupný kdekoliv a kdykoliv. To z něj činí vysoce výkonný prostředek k výměně informací a znalostí, i k administrativním účelům.

Veškeré zdrojové soubory jsou uloženy na CD-ROM médiu. Jeho obsahem jsou také volně šiřitelné produkty PHP, MySQL, webserver Apache a referenční příručky HTML a CSS.

SEZNAM POUŽITÉ LITERATURY

- [1] KOSEK, J. *HTML - Tvorba dokonalých WWW stránek, podrobný průvodce*. 1. vyd. Praha: Grada Publishing s.r.o., 1998. 296 s. ISBN 80-7169-608-0.
- [2] KOSEK, J. *PHP – tvorba interaktivních internetových aplikací, podrobný průvodce*. 1. vyd. Praha: Grada Publishing s.r.o., 1998. 492 s. ISBN 80-7169-373-1.
- [3] SCHURMAN, Erik M., PARDI, William J. *Dynamické HTML v akci*. 1. vyd. Praha: Computer Press, 2000. 397 s. ISBN 80-7226-401-X.
- [4] LACKO, L. *Web a databáze*. 1. vyd. Praha: Computer Press, 2001. 250 s. ISBN 80-7226-555-5.
- [5] MASLAKOWSKI, M. *Naučte se MySQL za 21 dní*. 1. vyd. Praha: Computer Press, 2001. 478 s. ISBN 80-7226-448-6.
- [6] Test. Umíte spojit počítače s mobily? Computer. 2004, ročník XI, č. 18/04, str. 94.
- [7] Test. Rozumíte tiskových technologiím? Computer. 2005, ročník XII, č. 2/05, str. 81.
- [8] Test. Umíte pracovat s videem na počítači? Computer. 2005, ročník XII, č.7/05, CD.
- [9] Polzer, J. *Rozběhejte si server*. Computer. 2005, ročník XII, č.7/05, str. 13.
- [10] Chudý, Š., Švec, V. *Pedagogika v teorii a praxi*. UTB - Academia centrum Zlín, 2004, ISBN 80-7318-192-4.
- [11] Čandík, M. *Teleedukácia - nová forma vzdelávania*. In: Sborník příspěvků z mezinárodní konference Modernizace vysokoškolské výuky technických předmětů, Gaudeamus, Hradec Králové, 2000, s. 37-39, ISBN 80-7041-723-4.
- [12] Čandík, M. *Pedagogické požiadavky na tvorbu výukových multimediálnych aplikácií*. In: Sborník příspěvků z mezinárodní konference Modernizace vysokoškolské výuky technických předmětů, Gaudeamus, Hradec Králové, 2000, str. 34-36, ISBN 80-7041-723-4.
- [13] Čandík, M. *Nové trendy v dištančnom vzdelávaní*. In: Sborník příspěvků z mezinárodní konference Trendy technického vzdělávání 2000, Olomouc, 2000, str.178-179, ISBN 80-244-0107-X.
- [14] Čandík, M. *O použití multimediálnych materiálov vo vzdelávaní*. In: Sborník příspěvků z mezinárodní konference Trendy technického vzdělávání 2000, Olomouc, 2000, s.180-182, ISBN 80-244-0107-X.

- [15] <http://www.kosek.cz>
- [16] <http://www.builder.cz>
- [17] <http://www.interval.cz>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

WWW	World-Wide-Web – oblast Internetu přístupná webovými prohlížeči
HTML	HyperText Markup Language – standard pro tvorbu webových stránek
PHP	Hypertextový PreProcesor – skriptovací jazyk pro webové stránky
HTTP	HyperText Transfer Protocol – přenosový protokol webové služby
MySQL	system relačních databází
CSS	Cascading Style Sheets – kaskádové styly
ASP	Active Server Pages – skriptovací jazyk pro webové stránky
URL	Uniform Resource Locator – jedinečné označení objektu Internetu
CGI	Common Gateway Interface – rozhraní pro komunikaci serveru a skriptu
SSL	Secure Socket Layer – programová vrstva síťového protokolu pro bezpečnou a šifrovanou síťovou komunikaci
SQL	Structured Query Language – jazyk sloužící ke komunikaci s databází
DDL	Data Definition Language - jazyk pro definici dat
DML	Data Manipulation Language - jazyk pro manipulaci s daty
RDM	Relační databázový model – typ databázového modelu
ODBC	Open DataBase Connectivity – programové rozhraní, pomocí něhož může aplikace přistupovat k datům ve zdroji dat pomocí jazyka SQL
MD5	Message Digest – kontrolní součet (hash) souboru či řetězce
Apache	osobní webserver operačního systému Windows, Unix

SEZNAM OBRÁZKŮ

<i>Obr. 1 Ukázka rozdělení textu do popisného a hlavního sloupce.....</i>	<i>12</i>
<i>Obr. 2 ukázka komunikace v elektronické konferenci</i>	<i>16</i>
<i>Obr. 3 Průběh komunikace mezi prohlížečem a WWW-serverem</i>	<i>24</i>
<i>Obr. 4 Selekcce v relaci</i>	<i>26</i>
<i>Obr. 5 Zobrazení agregace v relaci.....</i>	<i>27</i>
<i>Obr. 6 Projekce v relaci.....</i>	<i>27</i>
<i>Obr. 7 Základní pojmy v relační databázi</i>	<i>28</i>
<i>Obr. 8 Spolupráce SQL-serveru s aplikacemi</i>	<i>34</i>
<i>Obr. 9 Administrační nástroj EeasypHP.....</i>	<i>35</i>
<i>Obr. 10 Nabídka konfigurace programu EasyPHP.....</i>	<i>36</i>
<i>Obr. 11 phpMyAdmin – nástroj pro správu databází.....</i>	<i>37</i>
<i>Obr. 12 Úvodní stránka aplikace.....</i>	<i>38</i>
<i>Obr. 13 Schéma činnosti jádra systému</i>	<i>40</i>
<i>Obr. 14 ER – diagram databázových relací</i>	<i>43</i>
<i>Obr. 15 První krok v registraci.....</i>	<i>47</i>
<i>Obr. 16 Druhý krok v registraci</i>	<i>48</i>
<i>Obr. 17 Zdrojový kód funkce (v editoru PSPad), která zajišťuje vyhodnocení testu</i>	<i>49</i>
<i>Obr. 18 Ukázka testu z oblasti počítačových sítí.....</i>	<i>50</i>
<i>Obr. 19 Uživatelské rozhraní s právy student</i>	<i>51</i>
<i>Obr. 20 Uživatelské rozhraní s právy učitel</i>	<i>52</i>

SEZNAM TABULEK

<i>Tabulka 1 Klasifikace obchodních modelů</i>	<i>9</i>
<i>Tabulka 2 Přehled datových typů SQL</i>	<i>31</i>
<i>Tabulka 3 Logické členění stránek HTML nepřihlášených uživatelů</i>	<i>39</i>
<i>Tabulka 4 Logické rozdělení stránky HTML, pokud je uživatel přihlášen</i>	<i>39</i>
<i>Tabulka 5 Struktura tabulky UZIVATELE</i>	<i>44</i>
<i>Tabulka 6 Struktura tabulky ONLINE</i>	<i>44</i>
<i>Tabulka 7 Struktura tabulky ARTICLE</i>	<i>45</i>
<i>Tabulka 8 Struktura tabulky MSG</i>	<i>45</i>
<i>Tabulka 9 Struktura tabulky TESTY</i>	<i>45</i>
<i>Tabulka 10 Struktura tabulky VDISK</i>	<i>46</i>

SEZNAM PŘÍLOH

Abecední seznam použitých skriptů a jejich funkce

config.inc.php	konfigurační soubor systému, ovlivňuje vzhled webu
error.php	vypisuje informace o chybných akcích v systému
form.php	formulář pro poznámky
function.php	obsahuje pomocné funkce systému (např. pro zobrazování poznámek)
chat.php	struktura stránky chat
checklogin.php	skript vkládaný do všech ostatních na začátek. Obsahuje autorizaci uživatele a načtení informací o uživateli s jeho nastavením.
index.php	úvodní stránka systému pro neregistrovaného uživatele
index.php	je uložen ve složce admin, určuje strukturu pro přihlášené uživatele
info.php	informace o obsahu stránek a obsluze systému
login.php	skript pro obsluhu přihlášení uživatele
logout.php	skript pro obsluhu odhlášení uživatele
msg.php	výpis oznamovacích informací v systému (např. poznámka byla vložena, byl jste odhlášen ze systému)
reg.php	registrace nových uživatelů
sdelete.php	odstraňování souborů ze serveru
sdownload.php	stahování studijního materiálu
sendmsg.php	formulář pro odesílání textových zpráv do chatu
soubor.php	administrace studijního materiálu, formulář pro upload souboru
sqlconnect.php	připojení k SQL databázi
talk.php	editace informací o chatujících
testdownload.php	stahování testů
update.php	modifikace poznámek
upload.php	upload souborů na server
users.php	editace a správa komunikujících uživatelů v chatu